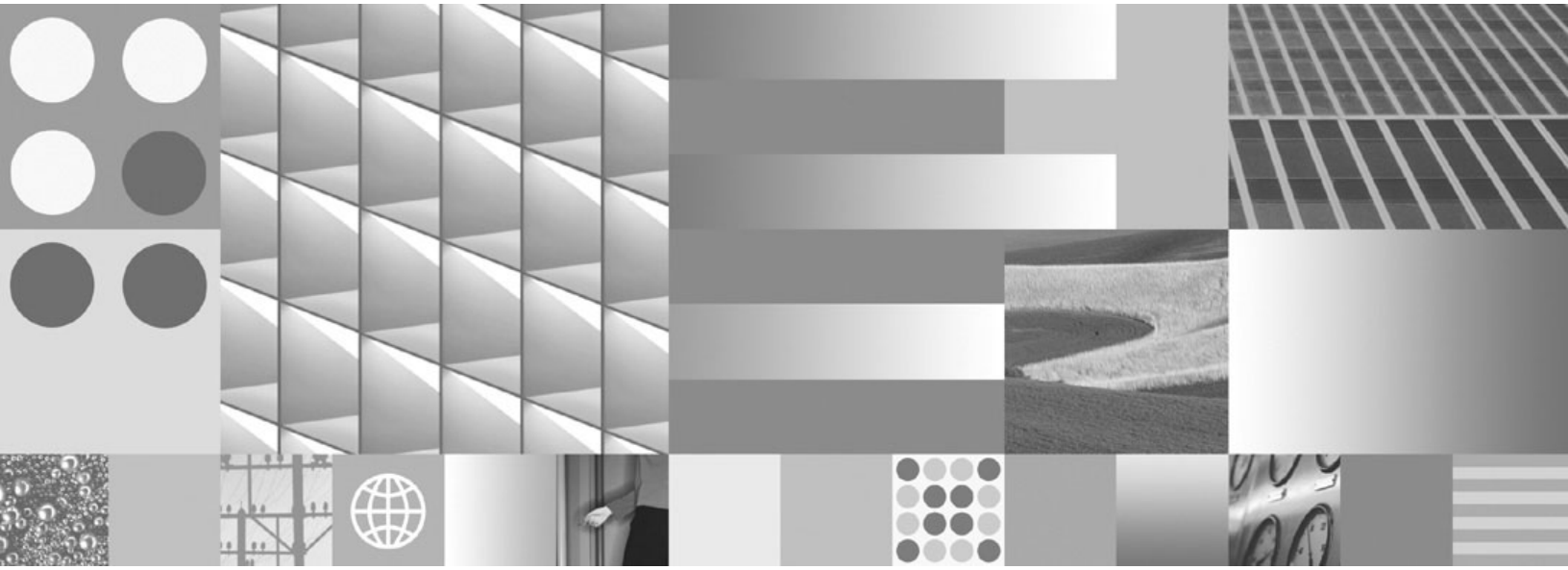
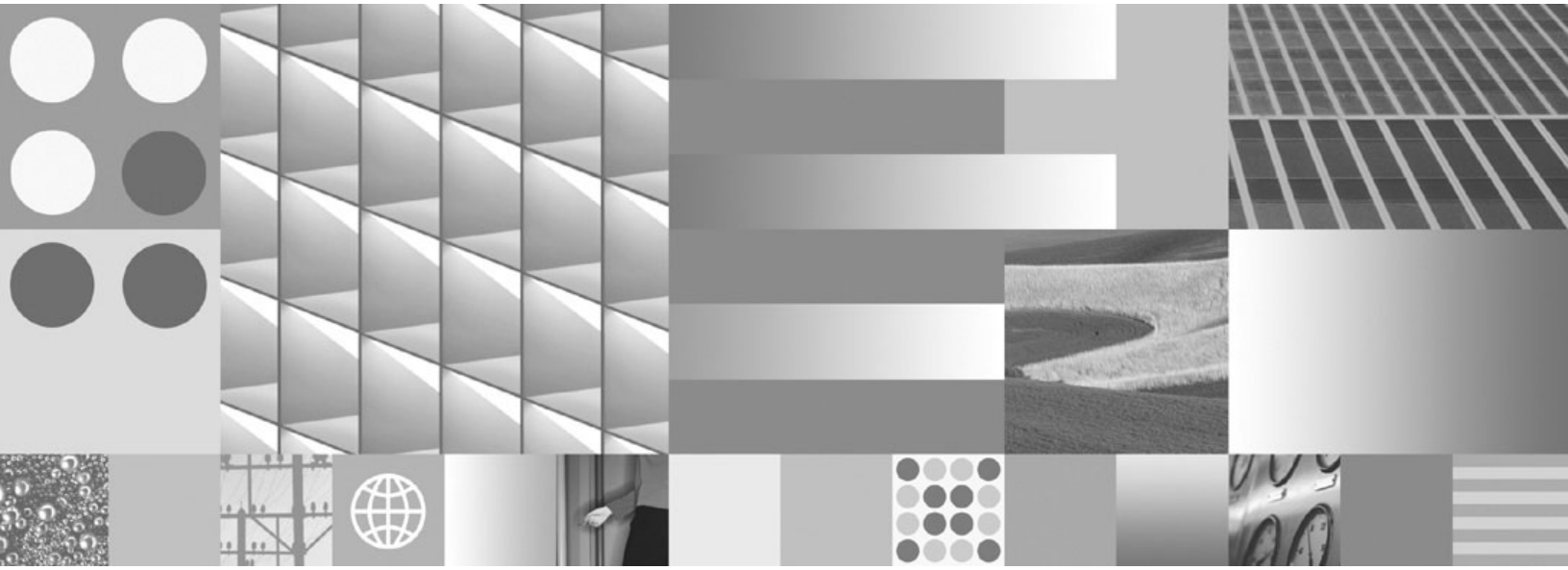
Linux, UNIX 및 Windows용
IBM DB2 9.7



데이터베이스 보안 안내서

SA30-3971-00

Linux, UNIX 및 Windows용
IBM DB2 9.7



데이터베이스 보안 안내서

SA30-3971-00

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 347 페이지의 부록 B『주의사항』의 정보를 읽으십시오.

개정판 주의사항

이 문서에는 IBM에서 소유하고 있는 정보가 있습니다. 이는 라이센스 계약에 따라 제공한 것이며 저작권의 보호를 받습니다. 이 책의 정보에는 제품 보증이 포함되지 않으며, 이 매뉴얼에서 제공된 어떠한 문장도 이와 같이 해석할 수 없습니다.

온라인으로 IBM 서적을 주문하거나 로컬 IBM 담당자를 통해 서적을 주문할 수 있습니다.

- 온라인으로 서적을 주문하려면 IBM Publications Center(www.ibm.com/shop/publications/order)로 이동하십시오.
- 로컬 IBM 담당자를 찾으려면 IBM Directory of Worldwide Contacts(www.ibm.com/planetwide)로 이동하십시오.

미국 또는 캐나다의 DB2 Marketing and Sales에서 DB2 서적을 주문하려면 1-800-IBM-4YOU (426-4968)로 전화하십시오.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

목차

이 책에 대한 정보

데이터베이스 보안 안내서는 데이터베이스 설치에 필요한 보안 레벨을 구현 및 관리하는 DB2® 보안 기능 사용 방법에 대해 설명합니다.

데이터베이스 보안 안내서에는 다음에 대한 자세한 정보가 있습니다.

- DB2 데이터베이스에 액세스할 수 있는 사용자의 인증 관리
- 데이터베이스 오브젝트 및 데이터에 대한 사용자 액세스 제어를 위한 인증 설정

제 1 장 DB2 보안 모델

두 개의 보안 모드가 DB2 데이터베이스 시스템 데이터 및 함수에 대한 액세스를 제어합니다. DB2 데이터베이스 시스템에 대한 액세스는 DB2 데이터베이스 시스템 외부에 있는 기능에 의해 관리되지만(인증), DB2 데이터베이스 시스템 내에서의 액세스는 데이터베이스 관리 프로그램에 의해 관리됩니다(권한 부여).

인증

인증은 시스템에서 사용자의 ID를 확인하는 프로세스입니다. 사용자 인증은 보안 플러그인 모듈을 통해 DB2 데이터베이스 시스템 외부에 있는 보안 기능에 의해 수행됩니다. 디폴트 인증 보안 플러그인 모듈은 운영 체제 기반 인증에 의존하며, DB2 데이터베이스 시스템을 설치할 때 포함됩니다. 사용자 편의를 위해 DB2 데이터베이스 관리 프로그램도 Kerberos 및 LDAP(Lightweight Directory Access Protocol)용 인증 플러그인 모듈을 제공합니다. 사용자의 구체적인 인증 요구를 보다 유연하게 충족할 수 있도록 고유한 인증 보안 플러그인 모듈을 빌드할 수 있습니다.

인증 프로세스를 수행하면 DB2 권한 부여 ID가 생성됩니다. 인증 시 사용자에 대한 그룹 멤버쉽 정보도 얻을 수 있습니다. DB2 데이터베이스 시스템을 설치할 때 포함되는 운영 체제 기반 그룹 멤버쉽 플러그인 모듈에 따라 디폴트로 얻을 수 있는 그룹 정보가 달라집니다. 원하는 경우 특정 그룹 멤버쉽 플러그인 모듈(예: LDAP)을 사용하여 그룹 멤버쉽 정보를 얻을 수 있습니다.

권한 부여

사용자가 인증되면, 데이터베이스 관리 프로그램은 해당 사용자가 DB2 데이터나 자원에 액세스할 수 있는지 여부를 판별합니다. 권한 부여는 DB2 데이터베이스 관리 프로그램이 인증된 사용자에 대한 정보를 얻는 프로세스로서, 사용자가 수행할 수 있는 데이터베이스 조작, 사용자가 액세스할 수 있는 데이터 오브젝트를 나타냅니다.

권한 부여 ID에 사용할 수 있는 사용 권한의 종류는 다음과 같습니다.

1. 기본 사용 권한: 권한 부여 ID에 직접 부여되는 사용 권한입니다.
2. 보조 사용 권한: 권한 부여 ID가 구성원으로 속해 있는 그룹과 역할에 부여되는 사용 권한입니다.
3. 공용 사용 권한: PUBLIC에 부여되는 사용 권한입니다.
4. 컨텍스트 인식 사용 권한: 트러스트된 컨텍스트 역할에 부여되는 사용 권한입니다.

권한 부여는 다음 범주에 있는 사용자에게 제공할 수 있습니다.

- 시스템 레벨 권한 부여

시스템 관리자(SYSADM), 시스템 제어(SYSCTRL), 시스템 유지보수(SYSMAINT) 및 시스템 모니터(SYSMON) 권한은 인스턴스 레벨 함수를 여러 가지 수준에서 제어할 수 있습니다. 권한은 특권을 그룹화하고 인스턴스, 데이터베이스 및 데이터베이스 오브젝트에 대한 유지보수 및 유틸리티 조작을 제어하는 방법을 제공합니다.

- 데이터베이스 레벨 권한 부여

  보안 관리자(SECADM), 데이터베이스 관리자(DBADM), 액세스 제어(ACCESSCTRL), 데이터 액세스(DATAACCESS), SQL 관리자(SQLADM), 워크로드 관리자(WLMADM) 및 설명(EXPLAIN) 권한은 데이터베이스 내에서의 제어를 제공합니다. 기타 데이터베이스 권한에는 LOAD(데이터를 테이블로 로드하는 권한) 및 CONNECT(데이터베이스에 연결하는 권한)가 있습니다.

- 오브젝트 레벨 권한 부여

  오브젝트 레벨 권한 부여에는 오브젝트에서 조작이 수행될 때 특권을 확인하는 것이 포함됩니다. 예를 들어, 테이블에서 선택하려면 최소한 테이블에 대한 SELECT 특권이 사용자에게 있어야 합니다.

- 컨텐츠 기반 권한 부여

  뷰를 통해 특정 사용자가 읽을 수 있는 테이블의 컬럼이나 행을 제어할 수 있습니다. 레이블 기반 액세스 제어(LBAC)를 사용하면 개별 행 및 개별 컬럼에 대해 읽기 및 쓰기 액세스 권한을 보유한 사용자를 판별할 수 있습니다.

이러한 기능을 DB2 감사 기능과 함께 사용하여 액세스를 모니터링하고 데이터베이스 설치에 필요한 보안 레벨을 관리할 수 있습니다.

인증

인증은 DB2 데이터베이스 시스템 외부에 있는 보안 기능을 사용하여 수행됩니다. 보안 기능은 운영 체제의 일부 또는 별도의 제품일 수 있습니다.

보안 기능은 사용자를 인증하기 위해 사용자 ID와 암호를 요구합니다. 사용자 ID는 보안 기능에 대한 사용자를 식별합니다. 사용자 및 보안 기능만 알고 있는 올바른 암호 정보를 제공해야 사용자의 신원(사용자 ID)이 검증됩니다.

주: 비루트 설치의 경우, db2rfe 명령을 실행하여 운영 체제 기반 인증을 사용 가능하게 해야 합니다.

인증 후:

- SQL 권한 부여 이름 또는 *authid*를 사용하여 DB2가 사용자를 식별해야 합니다. 이 이름은 사용자 ID 또는 맵핑된 값과 동일할 수 있습니다. 예를 들어, UNIX® 운영 체제에서 디폴트 보안 플러그인 모듈을 사용하는 경우, DB2 *authid*는 DB2 이름 지정 규칙에 따라 UNIX 사용자 ID를 대문자로 변환하여 파생됩니다.

- 사용자가 속한 그룹 목록이 확인됩니다. 그룹 멤버쉽은 사용자에게 권한을 부여할 때 사용할 수 있습니다. 그룹은 DB2 권한 부여 이름에 맵핑되어야 하는 보안 기능 엔티티입니다. 이 맵핑은 사용자 ID에 사용된 방식과 비슷한 방식으로 실행됩니다.

DB2 데이터베이스 관리 프로그램이 보안 기능을 사용하여 사용자를 인증할 때 다음 두 가지 방법 중 하나를 사용합니다.

- 신원 확인의 증거로 성공적인 보안 시스템 로그인이 사용되며 다음을 허용합니다.
  - 로컬 명령을 사용하여 로컬 데이터에 액세스
  - 서버가 클라이언트 인증을 트러스트할 때 리모트 연결 사용
- 신원 확인의 증거로 보안 기능을 사용하여 사용자 ID와 암호의 유효성을 확인하며 다음을 허용합니다.
  - 서버에서 인증 증거를 요구하는 리모트 연결 사용
  - 로그인에 사용된 ID와 다른 ID로 사용자가 명령을 실행하는 조작 사용

주: 일부 UNIX 시스템의 경우, DB2 데이터베이스 관리 프로그램이 운영 체제에서 잘못된 암호를 시도한 횟수를 로그하며, 클라이언트가 LOGINRETRIES 매개변수에 지정된 허용되는 로그인 시도 횟수를 초과하는 경우를 감지할 수 있습니다.

권한 부여

권한 부여는 DB2 기능을 사용하여 수행됩니다. DB2 테이블 및 구성 파일이 각 권한 부여 이름과 연관된 사용 권한을 기록하는 데 사용됩니다.

인증된 사용자가 데이터에 액세스하려고 하면 기록된 사용 권한과 다음에 해당하는 사용 권한이 비교됩니다.

- 사용자의 권한 부여 이름
- 사용자가 속해 있는 그룹
- 그룹 또는 역할을 통해 사용자에게 직접 또는 간접적으로 부여된 역할
- 트러스트된 컨텍스트를 통해 획득한 사용 권한

DB2 서버는 이 비교 작업을 토대로 요청된 액세스를 허용할지 여부를 결정합니다.

기록된 사용 권한의 유형은 특권, 권한 레벨 및 LBAC 증명서입니다.

*특권*은 사용자가 데이터베이스 자원을 작성하거나 액세스할 수 있도록 권한 부여 이름에 대해 하나의 사용 권한을 정의합니다. 특권은 데이터베이스 카탈로그에 저장됩니다.

*권한 부여 레벨*에서는 특권을 그룹화하고 데이터베이스 관리 프로그램 조작을 제어할 수 있습니다. 데이터베이스 고유 권한은 데이터베이스 카탈로그에 저장됩니다. 시스템 권한은 그룹 멤버쉽과 연관되고, 그룹 이름은 권한 레벨과 연관되며, 주어진 인스턴스용 데이터베이스 관리 프로그램 구성 파일에 저장됩니다.

*LBAC 증명서*는 레이블 기반 액세스 제어(LBAC)에 의해 보호되는 데이터에 대한 액세스를 허용하는 LBAC 보안 레이블 및 LBAC 규칙 면제입니다. LBAC 증명서는 데이터베이스 카탈로그에 저장됩니다.

그룹은 각 사용자에 대해 개별적으로 권한을 부여하거나 취소할 필요 없이 사용자 콜렉션에 대한 권한 부여를 수행하는 편리한 방법입니다. 다르게 지정되지 않는 한, 그룹 권한 부여 이름은 권한 부여 목적에 맞게 권한 부여 이름을 사용하는 곳에서 사용할 수 있습니다. 일반적으로 그룹 멤버쉽은 동적 SQL과 데이터베이스가 아닌 오브젝트 권한 부여(예: 인스턴스 레벨 명령 및 유틸리티)용으로 간주되며, 정적 SQL용으로는 간주되지 않습니다. 단, 특권이 PUBLIC으로 부여될 경우는 예외입니다. 이것은 정적 SQL이 처리될 때 고려됩니다. 그룹 멤버쉽이 적용되지 않는 특정한 경우에 대해서는 해당하는 DB2 문서에 설명되어 있습니다.

역할은 하나 이상의 특권이 그룹화된 데이터베이스 오브젝트로, GRANT문을 사용하여 사용자, 그룹, PUBLIC 또는 다른 역할에 지정하거나, CREATE TRUSTED CONTEXT 또는 ALTER TRUSTED CONTEXT문을 사용하여 트러스트된 컨텍스트에 지정할 수 있습니다. 워크로드 정의의 SESSION_USER ROLE 연결 속성에 역할을 지정할 수도 있습니다. 역할을 사용하는 경우 데이터베이스 오브젝트에 대한 액세스 권한을 역할과 연관시키십시오. 이러한 역할의 구성원인 사용자는 데이터베이스 오브젝트에 액세스할 때 사용하는 역할에 정의된 특권을 갖습니다.

역할은 그룹과 유사한 기능을 제공하며, 각 사용자에 대해 개별적으로 권한을 부여하거나 취소할 필요 없이 사용자 콜렉션에 대한 권한 부여를 수행합니다. 역할의 장점 중 하나는 DB2 데이터베이스 시스템에 의해 관리된다는 점입니다. 역할에 부여된 사용 권한은 그룹에 부여된 사용 권한과 달리, 뷰, 트리거, 구체화된 쿼리 테이블(MQT), 패키지 및 SQL 루틴에 대한 권한 부여 프로세스 중에 고려됩니다. 그룹에 부여된 사용 권한은 뷰, 트리거, MQT, 패키지 및 SQL 루틴에 대한 권한 부여 프로세스 중에 고려되지 않습니다. 왜냐하면 DB2 데이터베이스 시스템에서는 그룹 멤버쉽이 변경되는 시기를 알지 못하여 위의 오브젝트를 무효화할 수 없기 때문입니다.

주: 그룹에 부여된 역할에 대해 부여된 사용 권한은 뷰, 트리거, MQT, 패키지 및 SQL 루틴에 대한 권한 부여 프로세스 중에 고려되지 않습니다.

SQL문 처리 중 DB2 권한 부여 모델에서 고려하는 사용 권한은 다음과 같습니다.

1. SQL문과 연관된 1차 권한 부여 ID에 부여된 사용 권한
2. SQL문과 연관된 2차 권한 부여 ID(그룹 또는 역할)에 부여된 사용 권한
3. PUBLIC에 부여된 사용 권한(다른 역할을 통해 PUBLIC에 직접 또는 간접적으로 부여된 역할 포함)
4. 트러스트된 컨텍스트 역할에 부여된 사용 권한(해당하는 경우)

DB2 데이터베이스 관리 프로그램 설치 및 사용 시 보안 고려사항

보안 고려사항은 제품이 설치된 순간부터 DB2 관리자에게 중요합니다.

DB2 데이터베이스 관리 프로그램의 설치를 완료하려면, 사용자 ID, 그룹 이름 및 암호가 필요합니다. GUI 기반의 DB2 데이터베이스 관리 프로그램 설치 프로그램은 다른 사용자 ID 및 그룹에 대한 디폴트값을 작성합니다. Linux® 및 UNIX 또는 Windows® 플랫폼 중 어느 플랫폼에 설치하는지에 따라 다른 디폴트값이 작성됩니다.

- UNIX 및 Linux 플랫폼에서 인스턴스 설정 창에 DB2 인스턴스를 작성하도록 선택한 경우 DB2 데이터베이스 설치 프로그램은 디폴트로 DAS(dasusr), 인스턴스 소유자(db2inst) 및 분리 사용자(db2fenc)마다 다른 사용자를 작성합니다. 선택적으로 다른 사용자 이름을 지정할 수 있습니다.

  DB2 데이터베이스 설치 프로그램은 아직 존재하지 않는 사용자 ID를 작성할 때까지 디폴트 사용자 이름에 1-99의 번호를 추가합니다. 예를 들어, db2inst1 및 db2inst2 사용자가 이미 존재하면, DB2 데이터베이스 설치 프로그램은 사용자 db2inst3을 작성합니다. 10보다 큰 숫자가 사용되면 이름의 문자 부분이 디폴트 사용자 ID에서 절단됩니다. 예를 들어 사용자 ID db2fenc9가 이미 존재하면, DB2 데이터베이스 설치 프로그램은 사용자 ID에서 c를 절단한 다음 10을 추가합니다(db2fen10). 숫자 값이 디폴트 DAS 사용자에 추가되는 경우에는 절단되지 않습니다(예: dasusr24).
- Windows 플랫폼에서 DB2 데이터베이스 설치 프로그램은 디폴트로 DAS 사용자, 인스턴스 소유자 및 분리 사용자에 대해 사용자 db2admin을 작성합니다(원하는 경우, 설치 시 다른 사용자 이름을 지정할 수 있습니다). Linux 및 UNIX 플랫폼과 달리 사용자 ID에 숫자 값이 추가되지 않습니다.

관리자 이외의 사용자가 디폴트를 알아내 부적절하게 데이터베이스 및/또는 인스턴스에 액세스하는 위험성을 최소화하기 위해서는 디폴트 사용자 ID 이름 및/또는 암호를 변경해야 합니다.

주: 응답 파일 설치에는 사용자 ID 또는 그룹 이름에 디폴트값이 사용되지 않습니다. 이러한 값은 응답 파일에 지정해야 합니다.

암호는 사용자를 인증할 때 매우 중요합니다. 운영 체제 레벨에서는 인증을 요구하도록 설정되어 있지 않지만 데이터베이스가 운영 체제를 사용해 사용자를 인증할 경우 사용자는 연결됩니다. 예를 들어, Linux 및 UNIX 운영 체제에서 미정의 암호는 NULL로 처리됩니다. 이러한 상황에서 정의된 암호가 없는 임의의 사용자는 널(NULL) 암호를 갖는 것으로 간주됩니다. 운영 체제의 관점에서는 암호가 일치하는 것으로 인식되고 사용자는 유효성이 확인되어 데이터베이스에 연결할 수 있습니다. 운영 체제에서 데이터베이스에 대해 사용자를 인증하도록 하려면 운영 체제 레벨의 암호를 사용하십시오.

Linux 및 UNIX 운영 체제 환경에서 DB2 데이터베이스 파티션 기능(DPF)으로 작업할 경우 DB2 데이터베이스 관리 프로그램은 디폴트로 rsh 유틸리티(HP-UX에서는 remsh)를 사용하여 리모트 노드에서 일부 명령을 실행합니다. rsh 유틸리티는 네트워크를 통해 명백한 텍스트로 암호를 전송하며, DB2 서버가 보안 네트워크에 없는 경우 보안이 노출될 수 있습니다. DB2RSHCMD 레지스트리 변수를 사용하여 리모트 쉘 프로그램을 이러한 보안 노출이 없는 보다 안전한 방법으로 설정할 수 있습니다. 보다 안전한 방법의 한 예로는 ssh가 있습니다. 리모트 쉘 구성 제한사항에 대해서는 DB2RSHCMD 레지스트리 변수 문서를 참조하십시오.

DB2 데이터베이스 관리 프로그램을 설치한 후에는 사용자에게 부여된 디폴트 특권을 검토한 후 (필요한 경우) 변경하십시오. 디폴트로 설치 프로세스는 각 운영 체제에서 다음 사용자에게 시스템 관리(SYSADM) 특권을 부여합니다.

Linux 및 UNIX 플랫폼

인스턴스 소유자의 기본 그룹에 속하는 유효한 DB2 데이터베이스 사용자 이름.

Windows 환경

- 로컬 관리자 그룹의 구성원.
- DB2 데이터베이스 관리 프로그램이 사용자가 정의된 위치에서 사용자의 그룹을 열거하도록 구성되어 있는 경우 도메인 제어기에서 관리자 그룹의 구성원. **DB2_GRP_LOOKUP** 환경 변수를 사용하여 Windows 플랫폼에서 그룹 열거를 구성하십시오.
- Windows 확장 보안을 사용할 경우 DB2ADMNS 그룹의 구성원. DB2ADMNS 그룹의 위치는 설치 중 결정됩니다.
- LocalSystem 어카운트.

관리자는 데이터베이스 관리 프로그램 구성 매개변수 **sysadm_group**을 갱신하여 SYSADM 특권을 갖는 사용자 그룹을 제어할 수 있습니다. 아래의 지침을 따라 DB2 데이터베이스 설치, 후속 인스턴스 및 데이터베이스 작성 둘 모두에 대한 보안 요구사항을 완료해야 합니다.

시스템 관리 그룹으로 정의된 그룹(**sysadm_group**을 갱신하여)이 존재해야 합니다. 이 그룹의 이름은 인스턴스 소유자를 위해 작성된 그룹임을 쉽게 식별할 수 있도록 지정되어야 합니다. 이 그룹에 속한 사용자 ID 및 그룹은 각 인스턴스에 대해 시스템 관리자 권한을 갖습니다.

관리자는 어떤 인스턴스와 연관되는지 쉽게 알 수 있도록 인스턴스 소유자의 사용자 ID를 작성해야 합니다. 이 사용자 ID는 위에서 작성한 SYSADM 그룹의 이름을 해당 그룹의 ID로 가져야 합니다. 또는 이 인스턴스 소유자 사용자 ID를 인스턴스 소유자 그룹의 구성원으로만 사용하고 다른 그룹에서는 사용하지 않는 것도 좋은 방법입니다. 인스턴스를 수정할 수 있는 사용자 ID 및 그룹의 확산을 제어합니다.

작성된 사용자 ID는 인스턴스 내의 데이터 및 데이터베이스에 액세스하기 전에 인증을 제공하는 암호와 연관되어야 합니다. 암호 작성시 사용자 조직의 이름 지정 지침을 따르는 것이 좋습니다.

주: 인스턴스 구성 또는 다른 파일을 우연히 삭제하거나 겹쳐쓰지 않게 하려면, 관리자는 서버에서 직접 수행된 일일 관리 태스크에 인스턴스 소유자와 동일한 1차 그룹에 속하지 않는 또다른 사용자 어카운트 사용을 고려해야 합니다.

인스턴스 및 데이터베이스 디렉토리에 대한 파일 사용 권한 요구사항

DB2 데이터베이스 시스템에서는 인스턴스 및 데이터베이스 디렉토리가 최소 수준의 사용 권한을 가지고 있어야 합니다.

주: 인스턴스 및 데이터베이스 디렉토리가 DB2 데이터베이스 관리 프로그램에 의해 작성되면, 권한이 정확하고 변경되지 않아야 합니다.

UNIX 및 Linux 머신에 있는 인스턴스 디렉토리 및 NODE000*x*/sqldbdir 디렉토리의 최소 권한은 u=rwx 및 go=rx여야 합니다. 여기에 사용된 문자가 의미하는 바는 다음 표와 같습니다.

문자	의미
u	사용자(소유자)
g	그룹
o	기타 사용자
r	읽기
w	쓰기
x	실행

예를 들어, /home에 있는 db2inst1 인스턴스에 대한 사용 권한은 다음과 같습니다.

```
drwxr-xr-x  36 db2inst1  db2grp1         4096 Jun 15 11:13 db2inst1
```

데이터베이스가 있는 디렉토리의 경우 NODE000*x*를 포함한 각 디렉토리 레벨에는 다음과 같은 사용 권한이 필요합니다.

```
drwxrwxr-x  11 db2inst1  db2grp1         4096 Jun 14 15:53 NODE0000/
```

예를 들어, 데이터베이스가 /db2/data/db2inst1/db2inst1/NODE0000에 있는 경우 /db2, /db2/data, /db2/data/db2inst1, /db2/data/db2inst1/db2inst1 및 /db2/data/db2inst1/db2inst1/NODE0000 디렉토리에는 drwxrwxr-x가 필요합니다.

NODE000x 내의 sqldbdir 디렉토리에는 다음과 같이 drwxrwxr-x 사용 권한이 필요합니다.

```
drwx------   5 db2inst1  db2grp1          256 Jun 14 14:17 SAMPLE/
drwxr-x---   7 db2inst1  db2grp1         4096 Jun 14 13:26 SQL00001/
drwxrwxr-x   2 db2inst1  db2grp1          256 Jun 14 13:02 sqldbdir/
```

주의:
파일 보안을 유지하려면 DB2 데이터베이스 관리 프로그램이 *DBNAME* 디렉토리(예: SAMPLE) 및 SQL*xxxx* 디렉토리를 작성할 때 지정된 사용 권한을 그대로 유지하십시오.

인증 세부사항

서버의 인증 방법

인스턴스 또는 데이터베이스로의 액세스에는 사용자가 *인증이 된 상태*여야 합니다. 각 인스턴스에 대한 *인증 유형*은 사용자가 확인되는 방법과 확인될 장소를 결정합니다.

인증 유형은 서버의 구성 파일에 저장되며 인스턴스가 작성될 때에 처음 설정됩니다. 인스턴스마다 하나의 인증 유형이 있으며, 해당 데이터베이스 서버와 제어 하에 있는 모든 데이터베이스로의 액세스를 포함합니다.

페더레이티드 데이터베이스에서 데이터 소스에 액세스하려는 경우, 페더레이티드 인증 유형에 대한 데이터 소스 데이터베이스 처리 및 정의를 고려해야 합니다.

주: SERVER_ENCRYPT 인증 사용 시 사용자 ID 및 암호와 DATA_ENCRYPT 인증 사용 시 사용자 ID, 암호 및 사용자 데이터의 암호화를 수행하기 위해 DB2 데이터베이스 관리 시스템이 사용하는 암호화 루틴의 인증 정보는 http://www.ibm.com/security/standards/st_evaluations.shtml 웹 사이트를 참조하십시오.

명시적으로 트러스트된 연결에서 사용자 전환

CLI/ODBC 및 XA CLI/ODBC 응용프로그램의 경우 인증이 필요한 사용자 전환 요청을 처리할 때 사용되는 인증 메커니즘은 트러스트된 연결 자체를 원래 설정하는 데 사용된 메커니즘과 동일합니다. 따라서 명시적으로 트러스트된 연결의 설정 중 사용되는 기타 조정된 보안 속성(예: 암호화 알고리즘, 암호화 키 및 플러그인 이름)은 해당 트러스트된 연결에서 사용자 전환 요청에 필요한 인증과 동일한 것으로 가정합니다. JAVA 응용프로그램을 통해 사용자 전환 요청에서 인증 방법을 변경할 수 있습니다(데이터 소스 특성 사용).

트러스트된 연결에서 사용자 전환 시 인증을 *요청하지 않는* 방식으로 트러스트 컨텍스트 오브젝트를 정의할 수 있으므로 명시적으로 트러스트된 연결 기능에서 사용자 전환을 충분히 활용하려면 사용자 작성 보안 플러그인이 다음을 수행할 수 있어야 합니다.

- 사용자 ID 전용 토큰 승인
- 해당 사용자 ID의 유효한 DB2 권한 부여 ID 리턴

주: 인증의 CLIENT 유형이 적용되는 경우 명시적으로 트러스트된 연결을 설정할 수 없습니다.

인증 유형 제공

다음 인증 유형이 제공됩니다.

SERVER

해당 구성에 적용되는 보안 메커니즘을 통해(예: 보안 플러그인 모듈을 통해) 서버에서 인증이 발생하도록 지정합니다. 디폴트 보안 메커니즘은 연결 또는 접속 시도 중 사용자 ID 및 암호가 지정된 경우 서버로 보내어 서버에서 유효한 사용자 ID 및 암호 조합과 비교하고 사용자가 인스턴스에 액세스할 수 있는지 여부를 판별하는 것입니다.

주: 서버 코드는 연결이 로컬인지 리모트인지를 감지합니다. 로컬 연결의 경우, 인증이 SERVER이면, 사용자 ID와 암호가 인증에 필요하지 않습니다.

SERVER_ENCRYPT

서버가 암호화된 SERVER 인증 스킴을 승인하도록 지정합니다. 클라이언트 인증이 지정되지 않으면, 클라이언트는 서버에서 선택된 방법을 사용하여 인증됩니다. 사용자 ID 및 암호가 네트워크를 통해 클라이언트에서 서버로 전송된 경우 사용자 ID 및 암호가 암호화됩니다.

클라이언트와 서버 사이에 조정된 결과 인증 방법이 SERVER_ENCRYPT인 경우 AES(Advanced Encryption Standard) 256비트 알고리즘을 사용하여 사용자 ID와 암호를 암호화하도록 선택할 수 있습니다. 이 경우 **alternate_auth_enc** 데이터베이스 관리 프로그램 구성 매개변수를 설정하십시오. 이 구성 매개변수에는 세 가지 설정이 있습니다.

- NOT_SPECIFIED(디폴트)는 클라이언트가 제안한 암호화 알고리즘(AES 256 비트 알고리즘)을 서버가 승인함을 의미합니다.
- AES_CMP는 연결 중인 클라이언트가 DES를 제안하지만 AES 암호화를 지원하는 경우 서버가 AES 암호화로 재조정함을 의미합니다.
- AES_ONLY는 서버가 AES 암호화만을 승인함을 의미합니다. 클라이언트가 AES 암호화를 지원하지 않는 경우 연결이 거부됩니다.

AES 암호화는 클라이언트와 서버 간에 조정된 인증 방법이 SERVER_ENCRYPT인 경우에만 사용할 수 있습니다.

CLIENT

운영 체제 보안을 사용하여 응용프로그램이 호출된 데이터베이스 파티션에서 인증이 발생함을 지정합니다. 연결 또는 접속 시도 중 지정된 사용자 ID 및 암호를 클라이언트 노드의 유효한 사용자 ID 및 암호 조합과 비교하여 사용자 ID가 인스턴스에 액세스할 수 있는지 여부를 판별합니다. 데이터베이스 서버에서 인증은 더 이상 발생하지 않습니다. 일반적으로 이를 단일 사인온이라고 합니다.

사용자가 로컬 또는 클라이언트 로그인을 수행하면, 사용자는 해당 로컬 클라이언트 워크스테이션에만 알려집니다.

리모트 인스턴스가 CLIENT 인증을 가지고 있으면 다른 두 매개변수(*trust_allclnts* 및 *trust_clntauth*)가 최종 인증 유형을 판별합니다.

트러스트된 클라이언트 전용 CLIENT 레벨 보안

트러스트된 클라이언트는 신뢰 받는 로컬 보안 시스템을 갖는 클라이언트입니다.

CLIENT의 인증 유형이 선택되면, 추가 옵션이 선택되어 운영 환경에 고유의 보안이 없는 클라이언트를 보호합니다.

보안이 없는 클라이언트를 보호하려면, 관리자는 *trust_allclnts* 매개변수를 NO로 설정하여 트러스트된 클라이언트 인증을 선택할 수 있습니다. 이는 모든 트러스트된 플랫폼이 서버를 대신하여 사용자를 인증할 수 있음을 나타냅니다. 트러스트되지 않은 클라이언트는 서버에서 인증을 받고 사용자 ID와 암호를 제공해야 합니다. *trust_allclnts* 구성 매개변수는 사용자가 트러스트된 클라이언트인지 나타내기 위해 사용합니다. 매개변수의 디폴트값은 YES입니다.

주: 모든 클라이언트(*trust_allclnts*가 YES임)가 인증에 대한 원시(native) 안전 보안 시스템이 없는 클라이언트로 신뢰 받을 수 있습니다.

트러스트된 클라이언트라 하더라도 서버에서 완전한 인증을 받아야 할 경우가 있습니다. 트러스트된 클라이언트의 유효성을 확인하는 장소를 나타내기 위해 *trust_clntauth* 구성 매개변수를 사용합니다. 이 매개변수의 디폴트값은 CLIENT입니다.

주: 트러스트된 클라이언트만을 위해, CONNECT 또는 ATTACH 시도 중에 사용자 ID 또는 암호가 명시적으로 제공되면, 사용자의 유효성이 클라이언트에서 확인됩니다. *trust_clntauth* 매개변수는 USER 또는 USING절에서 제공된 정보의 유효성을 확인하는 곳을 판별하는 데에만 사용됩니다.

z/OS®, OS/390®, VM, VSE 및 System i®의 원시 DB2 클라이언트를 제외하고 z/OS 및 System i의 JCC 유형 4 클라이언트를 포함한 모든 클라이언트로부터의 인증되지 않은 액세스를 제한하려면 *trust_allclnts* 매개변수를 DRDAONLY로 설정하십시오. 이 클라이언트만이 클라이언트측 인증을 수행하도록 신뢰 받을 수 있습니다. 기타 모든 클라이언트는 서버에 의해 인증되려면 사용자 ID와 암호를 제공해야 합니다.

trust_clntauth 매개변수는 위 클라이언트가 인증되는 위치를 판별하는 데 사용됩니다. *trust_clntauth*가 "client"이면 클라이언트에서 인증됩니다. *trust_clntauth*가 "server"이면, 인증은 사용자 ID와 암호가 제공되었을 때 클라이언트에서 발생하고, ID와 암호가 제공되면 서버에서 발생합니다.

표 1. TRUST_ALLCLNTS 및 TRUST_CLNTAUTH 매개변수 조합을 사용한 인증 모드

TRUST_ ALLCLNTS	TRUST_ CLNTAUTH	트러스트되지 않은 비 DRDA® 클라이언트 인증 (사용자 ID 및 암호 없음)	트러스트되지 않은 비 DRDA 클라이언트 인증 (사용자 ID 및 암호 있음)	트러스트되지 않은 비 DRDA 클라이언트 인증 (사용자 ID 및 암호 없음)	트러스트되지 않은 비 DRDA 클라이언트 인증 (사용자 ID 및 암호 있음)	DRDA 클라이언트 인증 (사용자 ID 및 암호 없음)	DRDA 클라이언트 인증 (사용자 ID 및 암호 있음)
YES	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT
YES	SERVER	CLIENT	SERVER	CLIENT	SERVER	CLIENT	SERVER
NO	CLIENT	SERVER	SERVER	CLIENT	CLIENT	CLIENT	CLIENT
NO	SERVER	SERVER	SERVER	CLIENT	SERVER	CLIENT	SERVER
DRDAONLY	CLIENT	SERVER	SERVER	SERVER	SERVER	CLIENT	CLIENT
DRDAONLY	SERVER	SERVER	SERVER	SERVER	SERVER	CLIENT	SERVER

DATA_ENCRYPT

서버가 암호화된 SERVER 인증 스킴 및 사용자 데이터의 암호화를 승인합니다. 인증은 SERVER_ENCRYPT에 표시된 내용과 동일한 방법으로 작동합니다. 사용자 ID 및 암호가 네트워크를 통해 클라이언트에서 서버로 전송된 경우 사용자 ID 및 암호가 암호화됩니다.

이 인증 유형을 사용할 경우 다음과 같은 사용자 데이터가 암호화됩니다.

- SQL 및 XQuery문
- SQL 프로그램 변수 데이터
- SQL 또는 XQuery문의 서버 처리의 출력 데이터 및 데이터에 대한 설명
- 쿼리 결과 생성된 응답 세트 데이터의 일부 또는 전부
- 대형 오브젝트(LOB) 데이터 스트림
- SQLDA 디스크립터

DATA_ENCRYPT_CMP

서버가 암호화된 SERVER 인증 스킴 및 사용자 데이터의 암호화를 승인합니다. 추가로, 이 인증 유형은 DATA_ENCRYPT 인증 유형을 지원하지 않는 하위 레벨 제품과의 호환성을 허용합니다. 해당 제품은 사용자 데이터를 암호화하지 않고 SERVER_ENCRYPT 인증 유형을 사용하여 연결할 수 있습니다. 새 인증 유형을 지원하는 제품은 이 유형을 사용해야 합니다. 이 인증 유형은 서버의 데이터베이스 관리 프로그램 구성 파일에서만 유효하며 CATALOG DATABASE 명령에서 사용할 경우 유효하지 않습니다.

KERBEROS

DB2 클라이언트 및 서버 둘 모두 Kerberos 보안 프로토콜을 지원하는 운영 체제에 있을 때 사용됩니다. Kerberos 보안 프로토콜은 공유 비밀 키를 작성하기 위해 일반 암호를 사용하여 써드 파티 인증 서비스로서 인증을 수행합니다. 이 키는 사용자 증명되며 로컬 또는 네트워크 서비스가 요청될 때 모든 경

우에서 사용자 식별을 검증하는 데 사용됩니다. 키는 명확한 텍스트로서 네트워크에서 사용자 이름 및 암호를 전달하는 필요성을 줄입니다. Kerberos 보안 프로토콜을 사용하면 리모트 DB2 데이터베이스 서버로의 단일 사인온을 사용할 수 있습니다. KERBEROS 인증 유형은 다양한 운영 체제에서 지원됩니다. 자세한 정보는 관련 정보 절을 참조하십시오.

Kerberos 인증은 다음과 같이 동작합니다.

1. 도메인 어카운트를 사용하여 클라이언트 머신에 로그온하는 사용자는 도메인 제어기의 Kerberos 키 분산 센터(KDC)에 인증을 요청합니다. 키 분산 센터는 클라이언트에 TGT(ticket-granting ticket)를 발행합니다.
2. 연결의 첫 번째 단계 동안, 서버는 DB2 데이터베이스 서버 서비스의 서비스 어카운트 이름인 목표 핵심부 이름을 클라이언트로 보냅니다. 서버의 목표 핵심부 이름 및 TGT를 사용하여 클라이언트는 도메인 제어기에 있는 TGS(Ticket-Granting Service)에 서비스 티켓을 요청합니다. 클라이언트의 TGT와 서버의 목표 핵심부 이름이 둘 다 유효하면 TGS가 클라이언트에 서비스 티켓을 발행합니다. 데이터베이스 디렉토리에 기록된 핵심부 이름을 name/instance@REALM으로 지정할 수 있습니다(이 형식은 Windows에서 승인되는 DOMAIN₩userID 및 userID@xxx.xxx.xxx.com 형식의 추가 사항임).
3. 클라이언트는 통신 채널(예: TCP/IP)을 사용하여 서버에 이 서비스 티켓을 보냅니다.
4. 서버는 클라이언트 서버 티켓의 유효성을 확인합니다. 클라이언트의 서비스 티켓이 유효하면 인증이 완료됩니다.

클라이언트 머신에서 데이터베이스를 카탈로그화하고 명시적으로 서버의 목표 핵심부 이름과 함께 Kerberos 인증 유형을 지정할 수 있습니다. 그러면 연결의 첫 번째 단계를 생략할 수 있습니다.

사용자 ID 및 암호를 지정하면 클라이언트는 해당 사용자 어카운트에 대한 TGT를 요청하여 이것을 인증에 사용합니다.

KRB_SERVER_ENCRYPT

서버가 KERBEROS 인증 또는 암호화된 SERVER 인증 스킴을 승인하도록 지정합니다. 클라이언트 인증이 KERBEROS이면, 클라이언트는 Kerberos 보안 시스템을 사용하여 인증됩니다. 클라이언트 인증이 SERVER_ENCRYPT이면, 클라이언트는 사용자 ID 및 암호화된 암호를 사용하여 인증됩니다. 클라이언트 인증이 지정되지 않으면, 클라이언트는 사용 가능한 경우 Kerberos를 사용하며 그렇지 않은 경우 암호 암호화를 사용합니다. 기타 클라이언트 인증 유형에 대하여 인증 오류가 리턴됩니다. 클라이언트의 인증 유형을 KRB_SERVER_ENCRYPT로 지정할 수 없습니다.

주: Kerberos 인증 유형은 특정 운영 체제에서 실행 중인 클라이언트 및 서버에서 지원됩니다. 자세한 정보는 관련 정보 절을 참조하십시오. Windows 운영 체제의 경우 클라이언트 및 서버 머신이 모두 동일한 Windows 도메인에 속하거나 트러스트된 도메인에 속해야 합니다. 서버가 Kerberos를 지원하고 일부 클라이언트 머신이 Kerberos 인증을 지원할 때 이 인증 유형을 사용해야 합니다.

GSSPLUGIN

서버가 GSS-API 플러그인을 사용하여 인증을 수행하도록 지정합니다. 클라이언트 인증을 지정하지 않을 경우, 서버는 *srvcon_gssplugin_list* 데이터베이스 관리 프로그램 구성 매개변수에 나열된 임의의 Kerberos 플러그인을 포함한 서버 지원 플러그인 목록을 클라이언트에 리턴합니다. 클라이언트는 목록에서 클라이언트 플러그인 디렉토리에 있는 첫 번째 플러그인을 선택합니다. 클라이언트가 목록에 있는 플러그인을 지원하지 않은 경우, Kerberos 인증 스킴(리턴된 경우)을 사용하여 인증됩니다. 클라이언트 인증이 GSSPLUGIN 인증 스킴인 경우, 클라이언트는 목록의 첫 번째 지원 플러그인을 사용하여 인증됩니다.

GSS_SERVER_ENCRYPT

서버가 플러그인 인증 또는 암호화된 서버 인증 스킴을 승인하도록 지정합니다. 클라이언트 인증이 플러그인을 통해 발생할 경우, 클라이언트는 서버 지원 플러그인 목록의 첫 번째 클라이언트 지원 플러그인을 사용하여 인증됩니다.

클라이언트 인증을 지정하지 않고 내재된 연결을 수행 중인 경우(즉, 클라이언트가 연결 수행 시 사용자 ID 및 암호를 제공하지 않을 경우), 서버는 서버 지원 플러그인의 목록, Kerberos 인증 스킴(목록의 플러그인 중 하나가 Kerberos 기반일 경우) 및 암호화된 서버 인증 스킴을 리턴합니다. 클라이언트는 클라이언트 플러그인 디렉토리에 있는 첫 번째 지원 플러그인을 사용하여 인증됩니다. 클라이언트가 목록에 있는 임의의 플러그인을 지원하지 않을 경우, Kerberos 인증 스킴을 사용하여 인증됩니다. 클라이언트가 Kerberos 인증 스킴을 지원하지 않을 경우, 클라이언트는 암호화된 서버 인증 스킴을 사용하여 인증되며 암호가 누락된 경우 연결이 실패합니다. 운영 체제에 대해 DB2 지원 Kerberos 플러그인이 존재하거나 *srvcon_gssplugin_list* 데이터베이스 관리 프로그램 구성 매개변수에 대해 Kerberos 기반 플러그인이 지정된 경우, 클라이언트가 Kerberos 인증 스킴을 지원합니다.

클라이언트 인증을 지정하지 않고 명시적 연결을 수행 중인 경우(즉, 사용자 ID 및 암호를 모두 제공한 경우), 인증 유형은 SERVER_ENCRYPT입니다. 이 경우 사용자 ID 및 암호를 암호화하는 데 사용되는 암호화 알고리즘의 선택은 **alternate_auth_enc** 데이터베이스 관리 프로그램 구성 매개변수의 설정에 따라 다릅니다.

주:

1. 구성 파일 자체로의 액세스는 구성 파일의 정보에 의해 보호되므로, 인증 정보를 변경하려면 인스턴스로부터 부주의하게 사용자를 잠그지 마십시오. 다음 데이터베이스 관리 프로그램 구성 파일 매개변수는 다음 인스턴스로의 액세스를 제어합니다.
   - AUTHENTICATION *
   - SYSADM_GROUP *
   - TRUST_ALLCLNTS
   - TRUST_CLNTAUTH
   - SYSCTRL_GROUP
   - SYSMAINT_GROUP

   *는 두 개의 가장 중요한 매개변수를 표시합니다.

   이러한 상황이 발생하지 않도록 다음을 수행할 수 있습니다. 실수로 DB2 데이터베이스 시스템으로부터 사용자 자신을 잠근 경우, 모든 플랫폼에서 장애 안전 옵션을 사용할 수 있습니다. 이 옵션을 사용하면 특권이 높은 로컬 운영 체제 보안 사용자를 사용하여 데이터베이스 관리 프로그램 구성 파일을 갱신하도록 보통 DB2 데이터베이스 보안 점검을 겹쳐쓸 수 있습니다. 이 사용자는 데이터베이스 관리 프로그램 구성 파일을 갱신하여 문제점을 수정하는 특권을 *항상* 가지고 있습니다. 그러나 이 보안 생략은 데이터베이스 관리 프로그램 구성 파일의 로컬 갱신으로 제한됩니다. 리모트로 또는 기타 모든 DB2 데이터베이스 명령에 대해 장애 안전 사용자를 사용할 수 없습니다. 이 특수 사용자는 다음과 같이 식별됩니다.
   - UNIX 플랫폼: 인스턴스 소유자
   - Windows 플랫폼: 로컬『관리자』그룹에 속한 사용자
   - 기타 플랫폼: 기타 플랫폼에는 로컬 보안이 없으므로, 모든 사용자는 로컬 보안 검사를 전달합니다.

리모트 클라이언트의 인증 고려사항

리모트 액세스를 위해 데이터베이스를 카탈로그화할 경우 데이터베이스 디렉토리 항목에 인증 유형을 지정할 수 있습니다.

인증 유형은 필요하지 않으며 지정하지 않은 경우 클라이언트는 디폴트로 SERVER_ENCRYPT를 사용합니다. 그러나 서버가 SERVER_ENCRYPT를 지원하지 않으면, 클라이언트는 서버에서 지원하는 값을 사용하여 재시도하려고 합니다. 서버가 여러 인증 유형을 지원할 경우, 클라이언트는 이들 중에서 선택하지 않고 대신 오류를 리턴합니다. 올바른 인증 유형이 사용되도록 하기 위해서 오류가 리턴됩니다. 이 경우, 클라이언트는 지원되는 인증 유형을 사용하는 데이터베이스를 카탈로그화해야 합니다. 인증 유형을 지정하면, 지정된 값이 서버의 값과 일치하는 한 인증이 즉시 시작됩니다. 불일치가 발견되면 DB2 데이터베이스에서는 복구를 시도합니다. 복구 작업으로 차이를 조

정하는 데 보다 많은 플로우가 발생할 수 있으며 DB2 데이터베이스가 복구할 수 없는 경우에는 오류가 발생합니다. 불일치의 경우, 서버의 값은 올바른 것으로 간주합니다.

DATA_ENCRYPT_CMP 인증 유형은 데이터 암호화를 지원하지 않는 이전 릴리스의 클라이언트가 DATA_ENCRYPT 대신 SERVER_ENCRYPT 인증을 사용하여 서버에 연결할 수 있도록 합니다. 다음에 해당하는 경우, 이 인증이 작동하지 않습니다.

- 클라이언트 레벨이 버전 7.2임.
- 게이트웨이 레벨이 버전 8 FixPak7 이상임.
- 서버가 버전 8 FixPak 7 이상임.

위 내용에 모두 해당하는 경우 클라이언트는 서버에 연결할 수 없습니다. 연결할 수 있도록 하려면 클라이언트를 버전 8 이상으로 업그레이드하거나 게이트웨이 레벨이 버전 8 FixPak 6 이하여야 합니다.

연결할 때 사용되는 인증 유형은 데이터베이스 카탈로그 항목으로 게이트웨이에 적절한 인증 유형을 지정하여 결정됩니다. DB2® Connect™ 시나리오와 클라이언트가 DB2NODE 레지스트리 변수를 설정한 파티션된 데이터베이스 환경의 클라이언트 및 서버 모두에 적용됩니다. 카탈로그 파티션에서 적절한 파티션에 『홉』하기 위해 인증 유형을 카탈로그합니다. 이 시나리오에서는 단지 클라이언트와 서버 간의 조정이기 때문에 게이트웨이에서의 인증 유형 카탈로그가 사용되지 않습니다.

다른 인증 유형을 사용하는 클라이언트가 필요한 경우 게이트웨이에서 다른 인증 유형을 사용하여 다중 데이터베이스 별명을 카탈로그해야 합니다. 게이트웨이에서 어떤 인증 유형을 카탈로그할지 결정할 때, 클라이언트와 서버에서 사용한 동일한 인증 유형을 보존하거나 디폴트값이 SERVER인 NOTSPEC 인증 유형을 사용할 수 있습니다.

파티션된 데이터베이스 인증 고려사항

파티션된 데이터베이스에서 데이터베이스의 파티션마다 정의된 동일한 사용자 및 그룹 세트가 있어야 합니다. 정의가 동일하지 않으면, 사용자는 다른 파티션에 다른 사항을 실행하도록 권한이 부여됩니다.

모든 파티션에 걸쳐서 일관성이 권장됩니다.

Kerberos 인증 세부사항

DB2 데이터베이스 시스템은 AIX®, Solaris, Linux IA32 및 AMD64 Windows 운영 체제에서 Kerberos 인증 프로토콜을 지원합니다.

Kerberos 지원은 서버 및 클라이언트 인증 플러그인 둘 다로 사용되는 『IBMkrb5』라는 GSS-API 보안 플러그인으로 제공됩니다. UNIX 및 Linux의 경우 `sqllib/security{32|64}/plugin/IBM/{client|server}` 디렉토리에, Windows의 경우 `sqllib/security/plugin/IBM{client|server}` 디렉토리에 라이브러리가 있습니다.

주: 64비트 Windows의 경우, 플러그인 라이브러리를 IBMkrb564.dll이라고 합니다. 또한, UNIX 및 Linux 플러그인에 대한 실제 플러그인 소스 코드인 IBMkrb5.C는 the sqllib/samples/security/plugins 디렉토리에 있습니다.

DB2 데이터베이스 시스템에서 Kerberos 인증을 사용하려고 시도하기 전에 Kerberos 사용 및 구성에 대해 잘 이해할 것을 권장합니다.

Kerberos 설명 및 소개

Kerberos는 보안되지 않은 네트워크 환경에서 사용자를 안전하게 인증하기 위해 공유 비밀 키 시스템을 채택하는 써드 파티 네트워크 인증 프로토콜입니다. 응용프로그램 서버 및 클라이언트 간에 텍스트 사용자 ID 및 암호 쌍이 아닌 암호화된 티켓(Kerberos 키 분산 센터 또는 줄여서 KDC라고 하는 별도의 서버에서 제공하는)이 교환되는 3계층 시스템을 사용합니다. 이들 암호화된 서비스 티켓(*증명서*라고 함)의 수명은 유한하며 클라이언트 및 서버에서만 이해할 수 있습니다. 이로써 누군가 티켓을 네트워크에서 가로챈 경우에도, 보안 위험이 줄어듭니다. 각각의 사용자 또는 Kerberos 용어인 *핵심부*는 KDC와 공유되는 개인용 암호화 키를 소유합니다. KDC에 등록된 핵심부 및 컴퓨터 세트를 모두 *범주*라고 합니다.

Kerberos의 주요 기능은 사용자가 Kerberos 범주에서 자원에 대한 본인의 ID를 한 번만 검증하면 되는 단일 로그온 환경을 허용합니다. 이것은 DB2 데이터베이스에서 작업할 때, 사용자가 ID 또는 암호를 제공하지 않고 DB2 데이터베이스 서버에 연결하거나 접속할 수 있음을 의미합니다. 또다른 이점은 핵심부용 중앙 저장소가 사용되므로 사용자 ID 관리가 단순화된다는 점입니다. 마지막으로, Kerberos는 클라이언트가 서버 ID의 유효성을 확인할 수 있게 해주는 상호 인증을 지원합니다.

Kerberos 설정

DB2 데이터베이스 시스템 및 Kerberos 지원은DB2 데이터베이스에 관여하기 전에 관련 머신에 올바르게 설치 및 구성된 Kerberos 계층에 의존합니다. 여기에는 다음과 같은 요구사항이 포함되며, 추가 요구사항도 있을 수 있습니다.

1. 클라이언트와 서버 머신 및 핵심부가 동일한 범주 또는 트러스트된 범주(Windows 용어로 트러스트된 도메인)에 속해야 합니다.
2. 적절한 핵심부를 작성해야 합니다.
3. 해당하는 경우, 서버 키탭(keytab) 파일을 작성해야 합니다.
4. 포함된 모든 머신의 시스템 시계를 동기화해야 합니다(Kerberos는 일반적으로 5분의 시간 오차를 허용하며, 그렇지 않을 경우 증명서를 확보할 때 사전 인증 오류가 발생할 수도 있습니다).

Kerberos 설치 및 구성에 관한 세부사항은 설치된 Kerberos 제품과 함께 제공된 문서를 참조하십시오.

DB2 데이터베이스 시스템의 유일한 관심사는 연결 중인 응용프로그램이 제공하는 증명서(즉, 인증)를 기반으로 Kerberos 보안 컨텍스트가 성공적으로 작성되었는지 여부입니다. 서명 또는 메시지 암호화 같은 기타 Kerberos 기능은 사용되지 않습니다. 또한 사용 가능한 경우 상호 인증이 지원됩니다.

Kerberos 전제조건은 다음과 같습니다.

- AIX, Solaris 운영 환경 및 Linux 플랫폼의 경우 IBM® NAS(Network Authentication Service) Toolkit v1.4 이상이 필요합니다. NAS Toolkit은 https://www6.software.ibm.com/dl/dm/dm-nas-p에서 다운로드할 수 있습니다.
- Windows 플랫폼의 경우 전제조건이 없습니다.

Kerberos 및 클라이언트 핵심부

핵심부는 2파트 또는 다중 파트 형식일 수 있습니다(즉, *name@REALM* 또는 *name/instance@REALM*). 『name』 부분은 권한 부여 ID(AUTHID) 맵핑에서 사용되므로 이름이 DB2 데이터베이스 이름 지정 규칙을 준수해야 합니다. 이는 이름이 최대 30자여야 하며 사용되는 문자의 선택사항에 관한 기존 제한사항을 준수해야 함을 의미합니다. (AUTHID 맵핑은 나중에 나오는 주제에서 설명됩니다.)

주: Windows는 Kerberos 핵심부를 도메인 사용자와 직접 연관시킵니다. 이는 Kerberos 인증이 도메인 또는 범주와 연관되지 않은 Windows 머신에서는 사용할 수 없음을 의미합니다. 또한, Windows는 두 부분 이름(즉, *name@domain*)만을 지원합니다.

핵심부 자체가 대상 데이터베이스에 서비스 티켓을 요청하고 수신할 수 있는 아웃바운드 증명서를 확보할 수 있어야 합니다. 이는 UNIX 또는 Linux에서 보통 kinit 명령으로 수행되며, Windows에서 로그온할 때는 내재적으로 수행됩니다.

Kerberos 및 권한 부여 ID 맵핑

존재의 범위가 일반적으로 단일 머신으로 제한되는 운영 체제 사용자 ID와 달리, Kerberos 핵심부는 자신의 소유가 아닌 범주에서 인증을 받을 수 있습니다. 중복된 핵심부 이름의 잠재적 문제점은 범주 이름을 사용하여 핵심부를 완전히 규정함으로써 방지됩니다. Kerberos에서 완전한 핵심부는 `name/instance@REALM` 양식을 취하며 여기서 instance 필드는 실제로 『/』로 분리된 복수의 인스턴스(즉, `name/instance1/instance2@REALM`)이거나 전부 생략할 수도 있습니다. 명백한 제한사항은 범주 이름이 네트워크에 정의된 모든 범주에서 고유해야 한다는 점입니다. DB2 데이터베이스 관리 프로그램이 핵심부에서 AUTHID로의 단순 맵핑을 제공하려면 핵심부 이름, 즉 완전한 핵심부의 『이름』과 AUTHID 간 일대일 맵핑이 바람직합니다. AUTHID는 DB2 데이터베이스 관리 프로그램에서 디폴트 스키마로 사용되며 쉽고 논리적으로 파생되어야 하므로 단순 맵핑이 필요합니다. 따라서 데이터베이스 관리자는 다음 사항을 숙지해야 합니다.

- 범주는 달라도 이름이 같은 핵심부는 동일한 AUTHID에 맵핑됩니다.

- 다른 인스턴스에 있더라도 이름이 같은 핵심부는 동일한 AUTHID에 맵핑됩니다.

위의 사항을 고려할 때, 다음과 같은 권장사항이 작성됩니다.

- DB2 데이터베이스 서버를 액세스할 모든 트러스트된 범주에서 이름에 대하여 고유한 이름 스페이스를 유지보수하십시오.
- 인스턴스에 관계없이, 동일한 이름을 갖는 모든 핵심부는 동일한 사용자에게 속해야 합니다.

Kerberos 및 서버 핵심부

UNIX 또는 Linux에서 DB2 데이터베이스 인스턴스에 대한 서버 핵심부 이름은 `<instance name>/<fully qualified hostname>@REALM`이어야 합니다. 이 핵심부로 Kerberos 보안 컨텍스트를 승인할 수 있어야 하고, 초기화 수행 시 플러그인이 서버 이름을 DB2 데이터베이스에 보고하기 때문에 DB2 데이터베이스 인스턴스를 시작하기 전에 핵심부가 존재해야 합니다.

Windows에서 서버 핵심부는 DB2 데이터베이스 서비스가 시작된 도메인 어카운트로 간주됩니다. 이에 대한 예외는, 인스턴스가 LocalSystem 어카운트에 의해 시작할 수도 있다는 점입니다. 이 경우, 서버 핵심부 이름은 `host/<hostname>`으로 보고됩니다. 이 이름은 클라이언트 및 서버가 모두 Windows 도메인에 속할 경우에만 유효합니다.

Windows에서는 2파트 이하의 이름만 지원합니다. 이로 인하여 Windows 클라이언트가 UNIX 서버에 연결하려고 할 때 문제점이 발생합니다. 따라서, UNIX Kerberos와의 상호 작동성이 필요한 경우 Windows 도메인에서 Windows 어카운트에 대한 Kerberos 핵심부 맵핑을 설정해야 할 수도 있습니다. (관련 지시사항은 해당하는 Microsoft® 문서를 참조하십시오.)

UNIX 및 Linux 운영 체제에서 DB2 서버가 사용하는 Kerberos 서버 핵심부 이름을 겹쳐쓰기할 수 있습니다. DB2_KRB5_PRINCIPAL 환경 변수를 원하는 완전한 서버 핵심부 이름에 설정하십시오. 서버 핵심부 이름은 **db2start**가 실행된 이후에 DB2 데이터베이스 시스템에서만 인식되기 때문에 인스턴스를 재시작해야 합니다.

Kerberos 키탭(keytab) 파일

보안 컨텍스트 요청 승인을 원하는 UNIX 또는 Linux의 모든 Kerberos 서비스는 *키탭(keytab)* 파일에 증명서를 배치해야 합니다. 이는 DB2 데이터베이스에 의해 서버 핵심부로 사용되는 핵심부에 적용됩니다. 디폴트 키탭 파일에서만 서버 키를 검색합니다. 키탭 파일에 키를 추가하는 방법에 관한 지시사항은 Kerberos 제품에 제공되는 문서를 참조하십시오.

Windows에는 키탭 파일의 개념이 없으며, 시스템이 핵심부에 대한 인증서 핸들 저장 및 획득을 자동으로 처리합니다.

Kerberos 및 그룹

Kerberos는 그룹의 개념이 없는 인증 프로토콜입니다. 따라서, DB2 데이터베이스는 로컬 운영 체제에 의존하여 Kerberos 핵심부에 대한 그룹 목록을 얻습니다. UNIX 또는 Linux의 경우, 각각의 핵심부에 대하여 동등한 시스템 어카운트가 있어야 합니다. 예를 들어, `name@REALM` 핵심부의 경우 DB2 데이터베이스는 운영 체제 사용자 *이름*이 속해 있는 모든 그룹 이름을 로컬 운영 체제에서 조회하여 그룹 정보를 수집합니다. 운영 체제 사용자가 존재하지 않을 경우, AUTHID는 PUBLIC 그룹에만 속합니다. 한편, Windows는 도메인 어카운트를 Kerberos 핵심부에 자동으로 연관시키며 별도의 운영 체제 어카운트를 작성하기 위한 추가 단계가 필요하지 않습니다.

클라이언트에서 Kerberos 인증 사용

clnt_krb_plugin 데이터베이스 관리 프로그램 구성 매개변수를 사용 중인 Kerberos 플러그인의 이름으로 갱신해야 합니다. 지원되는 플랫폼에서는 이를 `IBMkrb5`로 설정합니다. 이 매개변수는 AUTHENTICATION 매개변수가 KERBEROS 또는 KRB_SERVER_ENCRYPT로 설정된 경우 연결 및 로컬 인스턴스 레벨 조치에 Kerberos를 사용할 수 있음을 DB2 데이터베이스에 알립니다. 그렇지 않은 경우, 클라이언트측 Kerberos 지원을 가정하지 않습니다.

주: Kerberos 지원이 사용 가능한지를 검증하는 점검은 수행되지 않습니다.

선택적으로 클라이언트에서 데이터베이스를 카탈로그화할 경우, 다음과 같이 인증 유형을 지정할 수 있습니다.

```
db2 catalog db testdb at node testnode authentication kerberos target
principal service/host@REALM
```

그러나 인증 정보를 제공하지 않을 경우, 서버는 클라이언트에 서버 핵심부 이름을 전송합니다.

서버에서 Kerberos 인증 사용

srvcon_gssplugin_list 데이터베이스 관리 프로그램 구성 매개변수를 서버 Kerberos 플러그인 이름으로 갱신해야 합니다. 이 매개변수가 지원되는 플러그인 목록만을 포함할 경우에도, 하나의 Kerberos 플러그인만을 지정할 수 있습니다. 그러나 필드가 공백이고 AUTHENTICATION이 KERBEROS 또는 KRB_SERVER_ENCRYPT로 설정된 경우, 디폴트 Kerberos 플러그인(`IBMkrb5`)을 가정하고 사용합니다. Kerberos 인증을 사용할 경우 항상 사용할 지 또는 수신 연결에만 사용할 지 여부에 따라 AUTHENTICATION 또는 SVRCON_AUTH 매개변수를 KERBEROS 또는 KRB_SERVER_ENCRYPT로 설정해야 합니다.

Kerberos 플러그인 작성

Kerberos 플러그인을 작성할 때 다음과 같은 몇 가지 고려사항이 있습니다.

- 초기화 기능에서 DB2 데이터베이스에 리턴되는 함수 포인터 배열의 *plugintype*을 DB2SEC_PLUGIN_TYPE_KERBEROS로 설정해야 하는 경우를 제외하고 Kerberos 플러그인을 GSS-API 플러그인으로 기록하십시오.
- 특정 조건에서 서버가 클라이언트에 서버 핵심부 이름을 보고할 수도 있습니다. 이와 같이 DRDA가 핵심부 이름을 GSS_C_NT_USER_NAME 형식(`server/host@REALM`)으로 규정하므로, 핵심부 이름을 GSS_C_NT_HOSTBASED_SERVICE 형식(`service@host`)으로 지정하지 않아야 합니다.
- 일반적인 상황에서 디폴트 키탭 파일은 KRB5_KTNAME 환경 변수로 지정할 수 있습니다. 그러나 서버 플러그인이 DB2 데이터베이스 엔진 프로세스에서 실행되므로, 이 환경 변수를 액세스할 수 없습니다.

zSeries® 및 System i 호환성

zSeries 및 System i와의 연결을 위해서는 AUTHENTICATION KERBEROS 매개변수를 사용하여 데이터베이스를 카탈로그해야 하며, TARGET PRINCIPAL 매개변수 이름을 명시적으로 지정해야 합니다.

zSeries 및 System i 둘 다 상호 인증을 지원하지 않습니다.

Windows 관련 문제

Windows 플랫폼에서 Kerberos를 사용할 때 다음과 같은 문제가 있음을 숙지하고 있어야 합니다.

- Windows가 오류를 발견하고 보고하는 방식 때문에 다음과 같은 조건으로 예상하지 못한 클라이언트 보안 플러그인 오류(SQL30082N, rc=36)가 발생합니다.
  - 만기된 어카운트
  - 유효하지 않은 암호
  - 만기된 암호
  - 관리자가 강제한 암호 변경
  - 사용 불가능한 어카운트

  또한 모든 경우에 DB2 관리 로그나 db2diag 로그 파일에 ″로그온 실패″ 또는 ″로그온 거부″가 표시됩니다.
- 도메인 어카운트 이름도 로컬로 정의되었을 경우, 명시적으로 도메인 이름을 지정하는 연결과 암호는 로컬 보안 권한에 접속할 수 없음 오류와 함께 실패합니다.

  이 오류는 로컬 사용자를 먼저 찾는 Windows 때문에 발생합니다. 연결 문자열에 완전한 사용자 이름을 지정하여 해결할 수 있습니다. 예: `name@DOMAIN.IBM.COM`
- @ 문자는 DB2 Kerberos 플러그인에서 도메인 구분자로 간주되기 때문에 Windows 어카운트에 @ 문자가 포함될 수 없습니다.

- Windows가 아닌 플랫폼과 상호 조작할 경우, 모든 Windows 도메인 서버 어카운트와 모든 Windows 클라이언트 어카운트가 DES 암호화를 사용할 수 있도록 구성되었는지 확인하십시오. DB2 서비스를 시작하는 데 사용된 어카운트가 DES 암호화를 사용하도록 구성되지 않은 경우, DB2 서버는 Kerberos 컨텍스트 승인에 실패합니다. 특히, DB2는 예상하지 못한 서버 플러그인 오류로 실패하고, AcceptSecurityContext API가 SEC_I_CONTINUE_NEEDED를 리턴(0x00090312L)했다고 로그합니다.

  Windows 어카운트가 DES 암호화를 사용하도록 구성되었는지 판별하려면 **사용 중 디렉토리**에서 **어카운트 등록 정보**를 확인하십시오. 어카운트 등록 정보가 변경되면 재시작을 필요로 할 수도 있습니다.
- 클라이언트 및 서버가 모두 Windows에 있으면 DB2 서비스가 LocalSystem 어카운트에서 시작될 수 있습니다. 그러나 클라이언트와 서버가 서로 다른 도메인에 있으면 연결이 유효하지 않은 목표 핵심부 이름 오류로 실패합니다. 완전한 서버 호스트 이름과 완전한 도메인 이름을 사용하여 클라이언트의 목표 핵심부 이름을 *server hostname@server domain name* 형식으로 명시적으로 카탈로그하면 일시적으로 이 문제를 해결할 수 있습니다.

  예: `host/myhost.domain.ibm.com@DOMAIN.IBM.COM`

  그렇지 않으면, DB2 서비스를 유효한 도메인 어카운트에서 시작해야 합니다.

서버에 암호 유지보수

암호 유지보수 태스크를 수행해야 하는 경우도 있습니다. 이 태스크는 일반적으로 서버에서 수행해야 하는데 대부분의 사용자가 서버 환경에 작업을 못하거나 서버 환경에 익숙하지 않기 때문에, 이러한 태스크 수행에 상당한 어려움이 따를 수 있습니다. DB2 데이터베이스 시스템은 서버에 있지 않고도 암호를 갱신하고 검증하는 방법을 제공합니다.

AIX 및 Windows 운영 체제에 설치된 DB2® Universal Database™ 버전 8, Linux 운영 체제에 설치된 DB2 버전 9.1 Fixpack 3 이상, z/OS 버전 7용 DB2, i5/OS® V6R1용 DB2에 있는 데이터베이스에 연결하는 경우 새 암호를 지정할 수 있습니다.

예를 들어, SQL1404N『암호 만기』또는 SQL30082N『원인 1(PASSWORD EXPIRED)로 인해 보안 처리 실패』와 같은 오류 메시지가 표시되는 경우, 다음과 같이 CONNECT문을 사용하여 암호를 변경하십시오.

```
CONNECT TO database USER userid USING
    password NEW new_password CONFIRM new_password
```

DB2 구성 지원 프로그램(CA)의 **암호 변경** 대화 상자 및 ATTACH 명령을 사용하여 암호를 변경할 수도 있습니다.

권한 부여, 특권 및 오브젝트 소유권

사용자(권한 부여 ID로 식별됨)에게 지정된 기능을 수행할 수 있는 권한이 있는 경우에만 사용자가 조작을 실행할 수 있습니다. 테이블을 작성하려면 사용자는 테이블 작성 권한을 부여받아야 하고, 테이블을 변경하려면 사용자는 테이블 변경 권한을 부여받아야 합니다.

데이터베이스 관리 프로그램에서 각 사용자는 특정 태스크를 수행하는 데 필요한 각 데이터베이스 함수를 사용할 수 있도록 특정 권한을 부여 받아야 합니다. 사용자는 자신의 사용자 ID에 대한 권한 부여를 통해 또는 권한을 보유하고 있는 역할 또는 그룹의 멤버쉽을 통해 필요한 권한을 얻을 수 있습니다.

권한 부여의 종류에는 *관리자 권한*, *특권* 및 *LBAC 증명서*라는 세 가지가 있습니다. 또한 오브젝트의 소유권은 작성된 오브젝트에 대한 권한 부여 등급과 함께 제공됩니다. 이러한 권한 부여의 종류는 아래에서 설명합니다.

관리자 권한

관리자 권한을 갖는 사람은 데이터베이스 관리 프로그램을 제어하는 태스크의 의무가 부여되고 데이터의 안정성 및 무결성을 책임집니다.

시스템 레벨 권한

시스템 레벨 권한은 인스턴스 레벨 함수에 대해 다양한 수준의 제어를 제공합니다.

- SYSADM(시스템 관리자) 권한

  SYSADM(시스템 관리자) 권한은 데이터베이스 관리 프로그램에 의해 작성되고 유지보수되는 모든 자원에 대한 제어를 제공합니다. 시스템 관리자는 SYSCTRL, SYSMAINT 및 SYSMON의 모든 권한을 보유합니다. SYSADM 권한을 가진 사용자는 데이터베이스 관리 프로그램을 제어하고 데이터의 무결성 및 보안을 보장할 수 있습니다.
- SYSCTRL 권한

  SYSCTRL 권한은 시스템 자원에 영향을 주는 조작에 대한 제어를 제공합니다. 예를 들어, SYSCTRL 권한이 있는 사용자는 데이터베이스를 작성, 갱신, 시작, 중지 또는 삭제할 수 있습니다. 또한 이 사용자는 인스턴스를 시작 및 중지할 수도 있지만 테이블 데이터에 액세스할 수는 없습니다. SYSCTRL 권한이 있는 사용자는 SYSMON 권한을 보유할 수도 있습니다.
- SYSMAINT 권한

  SYSMAINT 권한은 인스턴스와 연관된 모든 데이터베이스에서 유지보수 조작을 수행하기 위해 필요한 권한을 제공합니다. SYSMAINT 권한이 있는 사용자는 데이터베이스 구성을 갱신하고, 데이터베이스 또는 테이블 스페이스를 백업하며, 기존 데이

터베이스를 복원하고, 데이터베이스를 모니터할 수 있습니다. SYSCTRL과 마찬가지로 SYSMAINT는 테이블 데이터에 대한 액세스를 제공하지 않습니다. SYSMAINT 권한이 있는 사용자는 SYSMON 권한을 보유할 수도 있습니다.

- SYSMON(시스템 모니터) 권한

  SYSMON(시스템 모니터) 권한은 데이터베이스 시스템 모니터를 사용하는 데 필요한 권한을 제공합니다.

데이터베이스 레벨 권한

데이터베이스 레벨 권한은 데이터베이스 내에서의 제어를 제공합니다.

- DBADM(데이터베이스 관리자)

  DBADM 권한 레벨은 단일 데이터베이스에 대한 관리 권한을 제공합니다. 이 데이터베이스 관리자는 오브젝트 작성 및 데이터베이스 명령 발행에 필요한 특권을 보유하고 있습니다.

  SECADM 권한을 보유한 사용자만 DBADM 권한을 부여할 수 있습니다. DBADM 권한은 PUBLIC에 부여할 수 없습니다.

- SECADM(보안 관리자)

  SECADM 권한 레벨은 단일 데이터베이스에 대한 보안 관리 권한을 제공합니다. 보안 관리자 권한을 보유한 경우, 데이터베이스 보안 오브젝트(데이터베이스 역할, 감사 규정, 트러스트된 컨텍스트, 보안 레이블 구성요소 및 보안 레이블)를 관리하고 모든 데이터베이스 특권과 권한을 부여 및 취소할 수 있습니다. SECADM 권한을 보유한 사용자는 자신이 소유하지 않은 오브젝트의 소유권을 양도할 수 있으며, AUDIT 문을 사용하여 서버에 있는 특정 데이터베이스 또는 데이터베이스 오브젝트와 감사 규정을 연관시킬 수도 있습니다.

  SECADM 권한에는 테이블에 저장된 데이터에 액세스할 수 있는 고유 특권이 없습니다. 이 권한은 SECADM 권한을 가진 사용자만 부여할 수 있습니다. SECADM 권한은 PUBLIC에 부여할 수 없습니다.

- SQLADM(SQL 관리자)

  SQLADM 권한 레벨은 단일 데이터베이스 내에 있는 SQL문을 모니터하고 조정할 수 있는 관리 권한을 제공하며, ACCESSCTRL 또는 SECADM 권한을 가진 사용자가 부여할 수 있습니다.

- WLMADM(워크로드 관리자)

  WLMADM 권한은 워크로드 관리 오브젝트(예: 서비스 클래스, 작업 조치 세트, 작업 클래스 세트 및 워크로드)를 관리할 수 있는 관리 권한을 제공하며, ACCESSCTRL 또는 SECADM 권한을 가진 사용자가 부여할 수 있습니다.

- EXPLAIN(설명 권한)

  EXPLAIN 권한 레벨은 데이터 액세스 권한을 부여하지 않고 쿼리 계획을 설명할 수 있는 관리 권한을 제공하며, ACCESSCTRL 또는 SECADM 권한을 가진 사용자만 부여할 수 있습니다.

- ACCESSCTRL(액세스 제어 권한)

  ACCESSCTRL 권한 레벨은 다음 GRANT 및 REVOKE문을 발행할 수 있는 관리 권한을 제공하며, SECADM 권한을 보유한 사용자만 ACCESSCTRL 권한을 부여할 수 있습니다. ACCESSCTRL 권한은 PUBLIC에 부여할 수 없습니다.

  – GRANT(데이터베이스 권한)

    ACCESSCTRL 권한을 보유한 사용자는 ACCESSCTRL, DATAACCESS, DBADM 또는 SECADM 권한을 부여할 수 없습니다. SECADM 권한을 보유한 사용자만 이러한 권한을 부여할 수 있습니다.

  – GRANT(전역 변수 특권)
  – GRANT(인덱스 특권)
  – GRANT(모듈 특권)
  – GRANT(패키지 특권)
  – GRANT(루틴 특권)
  – GRANT(스키마 특권)
  – GRANT(시퀀스 특권)
  – GRANT(서버 특권)
  – GRANT(테이블, 뷰 또는 별칭 특권)
  – GRANT(테이블 스페이스 권한)
  – GRANT(워크로드 특권)
  – GRANT(XSR 오브젝트 특권)

- DATAACCESS(데이터 액세스 권한)

  DATAACCESS 권한 레벨은 다음과 같은 특권 및 권한을 제공하며, SECADM 권한을 보유한 사용자만 부여할 수 있습니다. DATAACCESS 권한은 PUBLIC에 부여할 수 없습니다.

  – LOAD 권한
  – 테이블, 뷰, 별칭 및 구체화된 쿼리 테이블에 대한 SELECT, INSERT, UPDATE, DELETE 특권
  – 패키지에 대한 EXECUTE 특권
  – 모듈에 대한 EXECUTE 특권
  – 루틴에 대한 EXECUTE 특권

예외가 적용되는 감사 루틴: AUDIT_ARCHIVE, AUDIT_LIST_LOGS, AUDIT_DELIM_EXTRACT

- 데이터베이스 권한(비관리자)

  테이블 또는 루틴의 작성과 같은 활동을 수행하거나 데이터를 테이블로 로드하려면 특정 데이터베이스 권한이 필요합니다. 예를 들어, load 유틸리티를 사용하여 데이터를 테이블로 로드하려면 LOAD 데이터베이스 권한이 필요합니다. 이 경우 사용자는 테이블에 대한 INSERT 권한도 보유하고 있어야 합니다.

특권

특권이란 특정 조치 또는 태스크를 수행하는 데 필요한 권한을 말합니다. 권한 부여된 사용자는 오브젝트를 작성하고, 자신이 소유하는 오브젝트에 액세스하며, GRANT문을 사용하여 자신의 오브젝트에 대한 특권을 다른 사용자에게 전달할 수 있습니다.

특권은 특정 사용자, 그룹 또는 PUBLIC에 부여될 수 있습니다. PUBLIC은 향후 사용자를 포함하는 모든 사용자로 구성된 특수 그룹입니다. 그룹의 구성원인 사용자는 해당 그룹에 권한 부여된 특권을 간접적으로 이용합니다.

CONTROL 특권: 오브젝트에 대한 CONTROL 특권을 보유하면 해당 데이터베이스 오브젝트에 액세스하고 해당 오브젝트에 대한 특권을 다른 사용자에게 권한 부여하거나 권한 취소할 수 있습니다.

주: CONTROL 특권은 테이블, 뷰, 별칭, 인덱스 및 패키지에만 적용됩니다.

다른 사용자에게 해당 오브젝트에 대한 CONTROL 특권이 필요한 경우, SECADM 또는 ACCESSCTRL 권한을 보유한 사용자가 해당 오브젝트에 대한 CONTROL 권한을 부여할 수 있습니다. CONTROL 특권을 오브젝트 소유자로부터 권한 취소할 수 없지만, TRANSFER OWNERSHIP문을 사용하여 오브젝트 소유자를 변경할 수 있습니다.

개별적 특권: 사용자가 특정 오브젝트에 대해 특정 태스크를 수행할 수 있도록 개별적 특권을 부여할 수 있습니다. 관리 권한(SYSADM 또는 SECADM) 또는 CONTROL 특권을 보유한 사용자는 사용자에게 권한을 부여하거나 사용자로부터 권한을 취소할 수 있습니다.

개별적 특권 및 데이터베이스 권한은 특정 함수를 사용할 수 있지만 기타 사용자에게 동일한 특권이나 권한을 부여하는 권한은 가지고 있지 않습니다. 다른 사용자에게 테이블, 뷰, 스키마, 패키지, 루틴 및 시퀀스 특권을 부여하는 권한은 GRANT 명령문에서 WITH GRANT OPTION을 통해 다른 사용자에게 확장될 수 있습니다. 그러나 WITH GRANT OPTION으로 일단 부여된 권한을 취소할 수 있는 특권을 사용자가 부여할 수는 없습니다. 권한을 취소하려면 SECADM 권한, ACCESSCTRL 권한 또는 CONTROL 권한이 있어야 합니다.

패키지 또는 루틴의 오브젝트에 대한 특권: 사용자에게 패키지 또는 루틴을 실행하는 특권이 있는 경우, 패키지 또는 루틴에서 사용하는 오브젝트의 특정 특권을 필요로 하지 않습니다. 패키지 또는 루틴에 정적 SQL 또는 XQuery문이 포함된 경우 패키지 소유자의 특권이 이들 명령문에 사용됩니다. 패키지 또는 루틴에 동적 SQL 또는 XQuery 명령문이 포함된 경우, 동적 쿼리 명령문을 발행하는 패키지의 **DYNAMICRULES** BIND 옵션의 설정 및 패키지가 루틴 컨텍스트에서 사용되고 있을 때 해당 명령문이 발행되었는지 여부에 따라 특권 검사에 사용되는 권한 부여 ID가 달라집니다.

사용자 또는 그룹은 개별적 특권 또는 권한의 임의 조합에 대해 권한 부여될 수 있습니다. 특권이 오브젝트와 연관되는 경우에는 해당 오브젝트가 존재해야 합니다. 예를 들어 사용자는 이전에 테이블이 작성된 경우에만 테이블에 대한 SELECT 특권을 부여할 수 있습니다.

주: 사용자 또는 그룹을 나타내는 권한 부여 이름에 권한 및 특권을 부여했는데 해당 이름으로 작성된 사용자 또는 그룹이 없는 경우 주의해야 합니다. 일정 시간이 경과하면 사용자 또는 그룹이 이 이름으로 작성되어 이 권한 부여 이름과 연관된 모든 권한 및 특권을 자동으로 받을 수 있습니다.

REVOKE 명령문은 이전에 권한 부여된 특권을 취소하기 위해 사용됩니다. 권한 부여 이름의 특권 취소는 모든 권한 부여 이름으로 부여된 특권을 취소합니다.

권한 부여 이름의 특권을 취소해도 이 권한 부여 이름으로 특권을 부여한 임의의 기타 권한 부여 이름의 동일한 특권은 취소되지 않습니다. 예를 들어, CLAIRE가 SELECT WITH GRANT OPTION을 RICK에게 권한 부여한 후에 RICK이 SELECT를 BOBBY 및 CHRIS에게 권한 부여한다고 가정하십시오. CLAIRE가 RICK의 SELECT 특권을 취소해도 BOBBY과 CHRIS는 여전히 SELECT 특권을 갖습니다.

LBAC 증명서

레이블 기반 액세스 제어(LBAC)를 사용하면 보안 관리자가 개별 행 및 개별 컬럼에 대해 쓰기 액세스 권한 및 읽기 액세스 권한을 갖고 있는 사용자를 정확히 판별할 수 있습니다. 보안 관리자는 보안 규정을 작성하여 LBAC 시스템을 구성합니다. 보안 규정에는 어떤 사용자가 어떤 데이터에 액세스할 수 있는지를 결정하는 데 사용되는 기준이 기술되어 있습니다. 임의의 한 테이블을 보호하는 데는 하나의 보안 테이블만 사용할 수 있으나 다른 테이블은 다른 보안 규정을 사용하여 보호할 수 있습니다.

보안 규정을 작성한 후 보안 관리자는 해당 규정에 포함될 보안 레이블 및 면제라는 데이터베이스 오브젝트를 작성합니다. 보안 레이블에는 특정 보안 기준 세트가 기술되어 있습니다. 면제를 보유한 사용자가 해당 보안 규정으로 보호된 데이터에 액세스하는 경우, 면제를 통해 보안 레이블을 비교하는 규칙이 해당 사용자에게는 적용되지 않도록 할 수 있습니다.

작성되면, 보안 레벨을 개별 테이블의 행 및 컬럼에 연관시켜 여기에 데이터가 보류되는 것을 방지할 수 있습니다. 보안 레이블에 의해 보호되는 데이터를 보호 데이터라고 합니다. 보안 관리자는 사용자에게 보안 레이블을 부여함으로써 사용자가 보호 데이터에 액세스하게 할 수 있습니다. 사용자가 보호 데이터에 액세스를 시도하면 사용자의 보안 레이블을 데이터를 보호하는 보안 레이블과 비교합니다. 보호 레이블은 일부 보안 레이블은 차단하고 일부 레이블은 차단하지 않습니다.

오브젝트 소유권

오브젝트가 작성되면, 하나의 권한 부여 ID가 오브젝트의 *소유권*에 지정됩니다. 소유권은 사용자가 적용 가능한 SQL 또는 XQuery문의 오브젝트를 참조할 수 있는 권한이 있음을 의미합니다.

오브젝트가 스키마 내에서 작성될 때, 명령문의 권한 부여 ID가 내재적 또는 명시적으로 지정된 스키마에 오브젝트를 작성하기 위한 필수 특권을 가져야 합니다. 즉, 권한 부여 이름은 스키마의 소유자이거나 스키마에 관한 CREATEIN 특권을 소유해야 합니다.

주: 이 요구사항은 테이블 스페이스, 버퍼 풀 또는 데이터베이스 파티션 그룹을 작성할 때 적용되지 않습니다. 이들 오브젝트는 스키마에서 작성되지 않습니다.

오브젝트가 작성될 때 명령문의 권한 부여 ID는 해당 오브젝트의 정의자이며, 오브젝트가 작성된 후에는 디폴트로 오브젝트 소유자가 됩니다.

주: 한 가지 예외가 존재합니다. AUTHORIZATION 옵션을 CREATE SCHEMA문에 대해 지정할 경우, CREATE SCHEMA 조작의 파트로 작성되는 모든 기타 오브젝트는 AUTHORIZATION 옵션에 의해 지정되는 권한 부여 ID에 의해 소유됩니다. 그러나 초기 CREATE SCHEMA 조작 후 스키마에 작성된 오브젝트는 특정 CREATE 문과 연관된 권한 부여 ID에서 소유합니다.

예를 들어 명령문 `CREATE SCHEMA SCOTTSTUFF AUTHORIZATION SCOTT CREATE TABLE T1 (C1 INT)`는 SCOTT이 소유하는 `SCOTTSTUFF` 스키마와 `SCOTTSTUFF.T1` 테이블을 작성합니다. BOBBY 사용자에게 `SCOTTSTUFF` 스키마에 관한 CREATEIN 특권이 부여되었고 `SCOTTSTUFF.T1` 테이블에 관한 인덱스를 작성한다고 가정하십시오. 인덱스는 스키마 이후에 작성되므로, BOBBY는 `SCOTTSTUFF.T1`에 관한 인덱스를 소유합니다.

특권은 다음과 같이 작성 중인 오브젝트 유형을 기반으로 오브젝트 소유자에게 지정됩니다.

- CONTROL 특권은 새로 작성된 테이블, 인덱스 및 패키지에 내재적으로 부여됩니다. 이 특권을 사용하여 오브젝트 작성자는 데이터베이스 오브젝트를 액세스하고, 해당 오브젝트에 관하여 다른 사용자에게/로부터 특권을 부여하고 취소할 수 있습니다. 다른 사용자에게 해당 오브젝트에 대한 CONTROL 특권이 필요한 경우,

ACCESSCTRL 또는 SECADM 권한을 보유한 사용자가 해당 오브젝트에 대한 CONTROL 권한을 부여해야 합니다. CONTROL 특권을 해당 오브젝트 소유자가 취소할 수 없습니다.

- 오브젝트 소유자에게 뷰 정의에서 참조하는 모든 테이블, 뷰 및 별칭에 관한 CONTROL 특권이 있을 경우 CONTROL 특권이 새로 작성된 뷰에 내재적으로 부여됩니다.
- 트리거, 루틴, 시퀀스, 테이블 스페이스 및 버퍼 풀과 같은 기타 오브젝트에는 연관된 CONTROL 특권이 없습니다. 그러나 오브젝트 소유자는 오브젝트와 연관된 각각의 특권을 자동으로 수신하며, 이러한 특권에는 WITH GRANT OPTION(지원되는 경우)이 포함됩니다. 따라서 오브젝트 소유자는 GRANT문을 사용하여 이러한 특권을 다른 사용자에게 제공할 수 있습니다. 예를 들어, USER1이 테이블 스페이스를 작성한 경우, USER1은 자동으로 이 테이블 스페이스에 대해 GRANT OPTION을 포함하는 USEAUTH 특권을 갖게 되며 USEAUTH 특권을 다른 사용자에게 부여할 수 있습니다. 추가로 오브젝트 소유자는 오브젝트를 변경하거나, 오브젝트에 관한 주석을 추가하거나 삭제할 수 있습니다. 이러한 권한 부여는 오브젝트 소유자에 대하여 내재적이며 취소할 수 없습니다.

테이블 변경과 같은 오브젝트에 대한 특정 특권은 소유자가 부여할 수 있으며, ACCESSCTRL 또는 SECADM 권한을 보유한 사용자가 소유자로부터 이러한 특권을 취소할 수 있습니다. 테이블에 주석을 처리하는 것과 같은 오브젝트에서의 특정 특권은 소유자가 권한 부여 및 권한 취소할 수 없습니다. TRANSFER OWNERSHIP문을 사용하여 이러한 특권을 다른 사용자에게 이동시키십시오. 오브젝트가 작성될 때 명령문의 권한 부여 ID는 해당 오브젝트의 정의자이며, 오브젝트가 작성된 후에는 디폴트로 오브젝트 소유자가 됩니다. 그러나 BIND 명령을 사용하여 패키지를 작성하고 **OWNER** *authorization id* 옵션을 지정한 경우 패키지에 있는 정적 SQL문에 의해 작성된 오브젝트의 소유자는 *authorization id* 값입니다. 또한 AUTHORIZATION절이 CREATE SCHEMA문에서 지정된 경우 AUTHORIZATION 키 이후에 지정된 권한 이름이 스키마의 소유자입니다.

보안 관리자 또는 오브젝트 소유자는 TRANSFER OWNERSHIP문을 사용하여 데이터베이스 오브젝트의 소유권을 변경할 수 있습니다. 따라서 관리자는 권한 부여 ID를 규정자로서 사용하여 오브젝트를 작성한 후 TRANSFER OWNERSHIP문을 사용하여 관리자가 오브젝트에 대해 갖는 소유권을 권한 부여 ID로 전송함으로써 권한 부여 ID 대신 오브젝트를 작성할 수 있습니다.

권한 개요

인스턴스 레벨 및 데이터베이스 레벨에 다양한 관리 권한이 있습니다. 이러한 관리 권한은 특정 특권 및 권한과 함께 그룹화되므로 해당 데이터베이스 설치의 태스크를 담당하는 사용자에게 관리 권한을 부여할 수 있습니다.

인스턴스 레벨 권한

인스턴스 레벨 권한을 통해 인스턴스 차원의 기능을 수행할 수 있습니다(예: 데이터베이스 작성 및 업그레이드, 테이블 스페이스 관리, 인스턴스에서의 활동 및 성능 모니터링). 인스턴스 레벨 권한은 데이터베이스 테이블의 데이터에 대한 액세스 권한을 제공하지 않습니다. 다음 다이어그램은 각 인스턴스 레벨 관리 권한이 제공하는 기능을 요약한 것입니다.

- SYSADM – 인스턴스 전체를 관리하는 사용자를 위한 권한
- SYSCTRL – 데이터베이스 관리 프로그램 인스턴스를 관리하는 사용자를 위한 권한
- SYSMAINT – 인스턴스 내에서 데이터베이스를 유지하는 사용자를 위한 권한
- SYSMON – 인스턴스 및 해당 데이터베이스를 모니터링하는 사용자를 위한 권한

높은 레벨의 권한을 가진 사용자는 낮은 레벨의 권한이 제공하는 권한도 가질 수 있습니다. 예를 들어, SYSCTRL 권한이 있는 사용자는 SYSMAINT 및 SYSMON 권한이 있는 사용자의 기능도 수행할 수 있습니다.

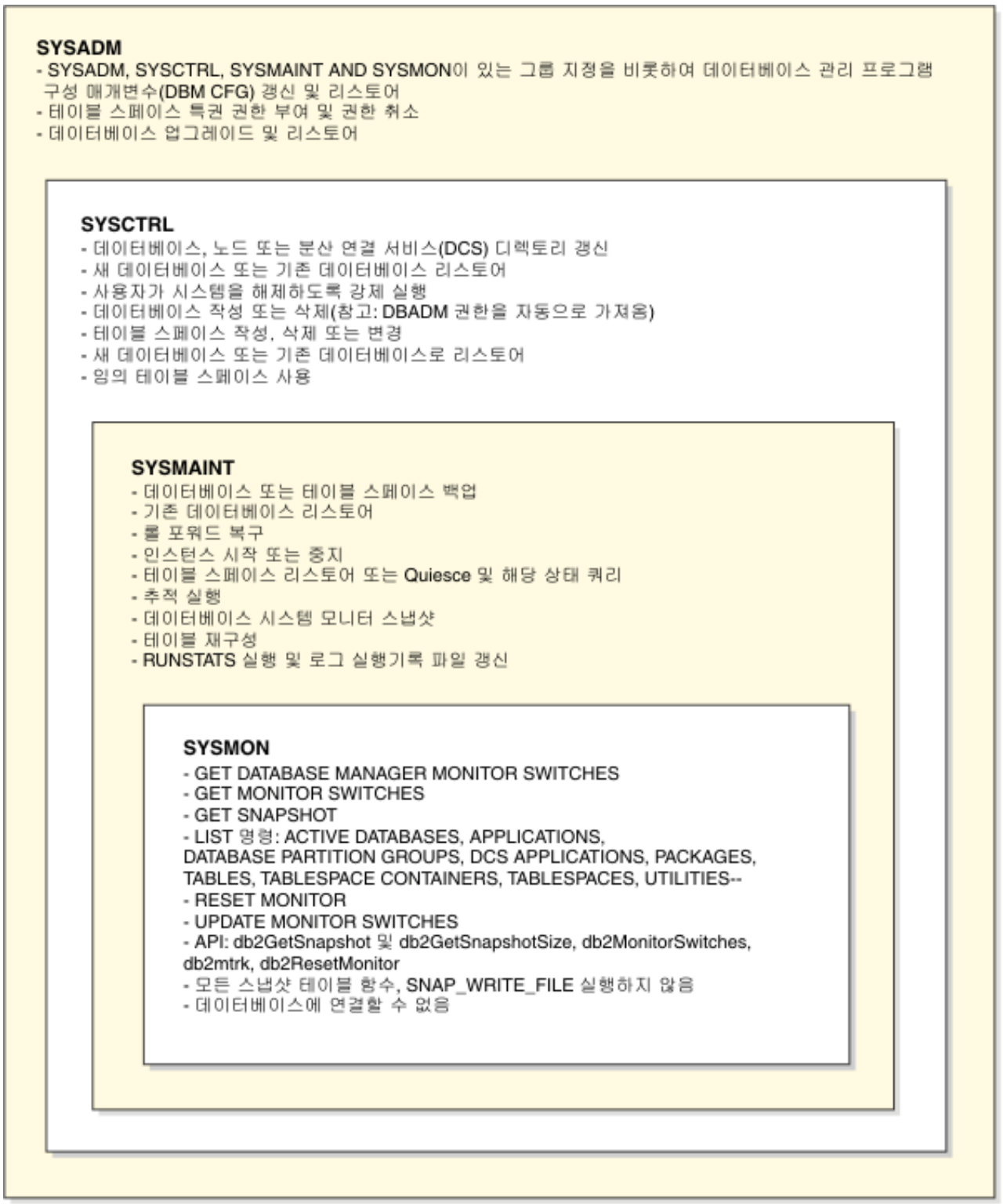


그림 1. 인스턴스 레벨 권한

데이터베이스 레벨 권한

데이터베이스 레벨 권한을 통해 특정 데이터베이스 내에서 기능을 수행할 수 있습니다(예: 특권 부여 및 취소, 데이터 삽입, 선택, 삭제 및 갱신, 워크로드 관리). 다음 다이어그램은 각 데이터베이스 레벨 권한이 제공하는 기능을 요약한 것입니다. 관리 데이터베이스 권한은 다음과 같습니다.

- SECADM – 데이터베이스 내에서 보안을 관리하는 사용자를 위한 권한
- DBADM – 데이터베이스를 관리하는 사용자를 위한 권한
- ACCESSCTRL – 권한과 특권을 부여 및 취소해야 하는 사용자를 위한 권한(SECADM, DBADM, ACCESSCTRL 및 DATAACCESS 제외, SECADM 권한은 이러한 권한을 부여하고 취소하는 데 필요함)
- DATAACCESS – 데이터에 액세스해야 하는 사용자를 위한 권한
- SQLADM – SQL 쿼리를 모니터링하고 조정하는 사용자를 위한 권한
- WLMADM – 워크로드를 관리하는 사용자를 위한 권한
- EXPLAIN – 쿼리 계획을 설명해야 하는 사용자를 위한 권한(EXPLAIN 권한은 데이터 자체에 대한 액세스 권한을 부여하지 않음)

다음 다이어그램은 하위 레벨 권한이 제공하는 기능이 포함되어 있는 상위 레벨 권한을 보여줍니다(있는 경우). 예를 들어, DBADM 권한이 있는 사용자는 SQLADM 및 EXPLAIN 권한이 있는 사용자의 기능과 WLMADM 권한이 있는 사용자의 모든 기능(워크로드에서 USAGE 특권을 부여하는 기능 제외)을 수행할 수 있습니다.

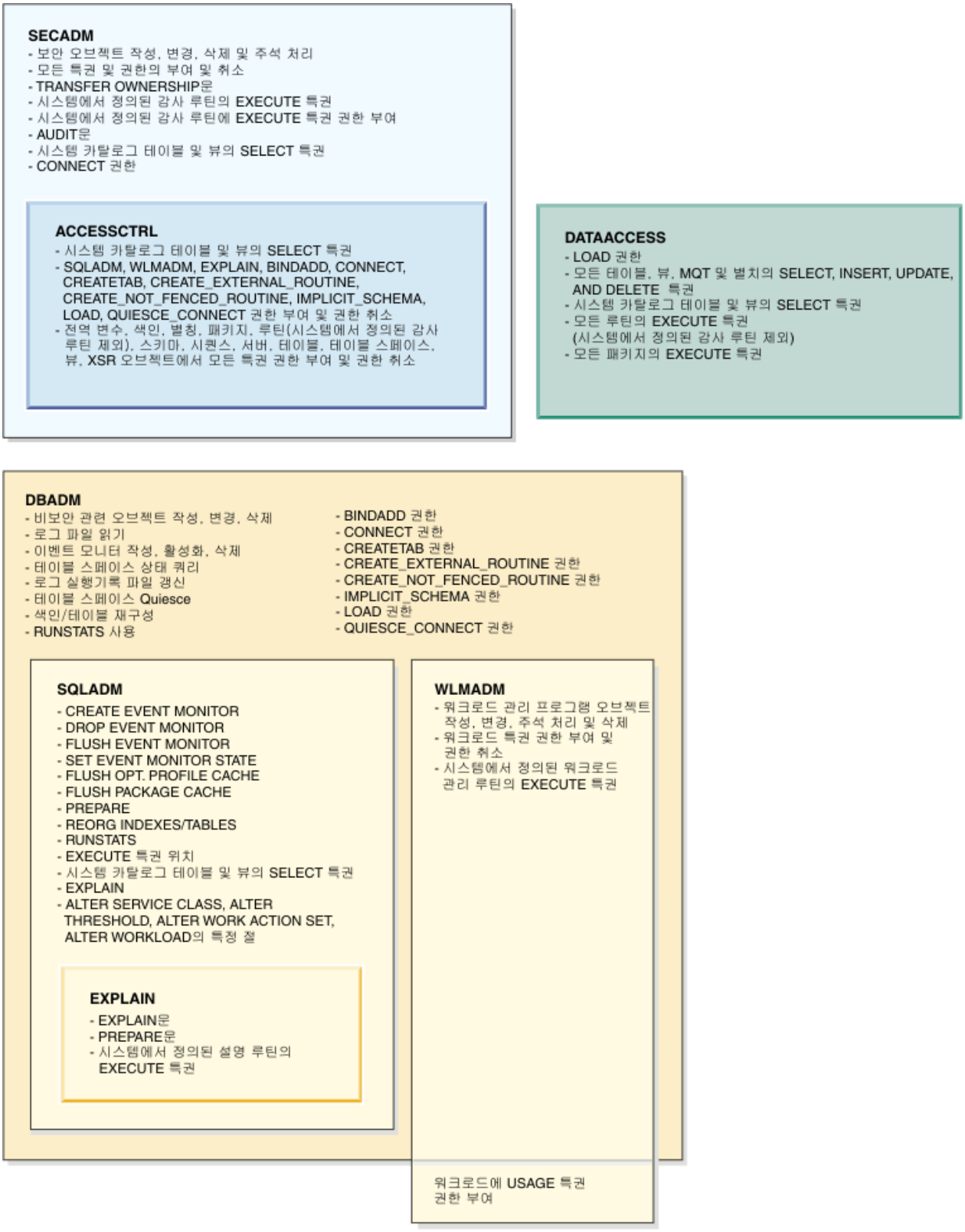


그림 2. 데이터베이스 레벨 권한

인스턴스 레벨 권한

시스템 관리 권한(SYSADM)

SYSADM 권한 레벨은 인스턴스 레벨에서 최상위 관리 권한 레벨입니다. SYSADM 권한을 보유한 사용자는 일부 유틸리티를 실행하고 인스턴스 내의 일부 데이터베이스 및 데이터베이스 관리 프로그램 명령을 발행할 수 있습니다.

SYSADM 권한은 **sysadm_group** 구성 매개변수에서 지정한 그룹에 지정됩니다. 해당 그룹의 멤버쉽은 플랫폼에서 사용하는 보안 기능을 통해 데이터베이스 관리 프로그램 외부에서 제어됩니다.

SYSADM 권한을 가진 사용자만이 다음 기능을 수행할 수 있습니다.

- 데이터베이스 업그레이드
- 데이터베이스 리스토어
- 데이터베이스 관리 프로그램 구성 파일 변경(SYSADM, SYSCTRL, SYSMAINT 또는 SYSMON 권한을 갖는 그룹 지정 포함)

SYSADM 권한을 보유한 사용자는 테이블 스페이스 특권을 부여 및 취소할 수 있으며 모든 테이블 스페이스를 사용할 수도 있습니다.

주: SYSADM 권한을 보유한 사용자가 데이터베이스를 작성하면 데이터베이스에 대한 ACCESSCTRL, DATAACCESS, DBADM 및 SECADM 권한이 해당 사용자에게 자동으로 부여됩니다. 해당 사용자가 데이터베이스에 데이터베이스 관리자 또는 보안 관리자로 액세스하지 못하게 하려면 해당 사용자로부터 이러한 데이터베이스 권한을 명시적으로 취소해야 합니다.

버전 9.7 이전 릴리스에서는 SYSADM 권한에 명시적 DBADM 권한이 포함되어 있어 모든 권한과 특권을 부여 및 취소할 수 있었습니다. 버전 9.7에서는 시스템 관리자, 데이터베이스 관리자 및 보안 관리자의 역할이 명백하게 구분되도록 DB2 권한 부여 모델이 갱신되었습니다. 이러한 개선의 결과로 SYSADM 권한을 통해 제공되던 권한이 축소되었습니다.

버전 9.7에서는 SECADM 권한만 모든 권한과 특권을 부여 및 취소할 수 있는 기능을 제공합니다.

SYSADM 권한을 보유한 사용자가 버전 9.5에서와 동일한 기능(SECADM 권한 부여 기능 제외)을 얻으려면 보안 관리자가 DBADM 권한 및 새 DATAACCESS/ACCESSCTRL 권한을 해당 사용자에게 명시적으로 부여해야 합니다. 이러한 새 권한은 GRANT DBADM ON DATABASE문을 WITH DATAACCESS 및 WITH ACCESSCTRL 옵션(디폴트 옵션)과 함께 사용하여 부여할 수 있습니다. DATAACCESS 권한은 특정 데이터베이스 내에 있는 데이터에 액세스할 수 있도록 허

용하는 권한이고, ACCESSCTRL 권한은 사용자가 특정 데이터베이스 내에서 특권 및 비관리 권한을 부여 및 취소할 있도록 허용하는 권한입니다.

Windows LocalSystem 어카운트에 대한 고려사항

Windows 시스템에서 **sysadm_group** 데이터베이스 관리 프로그램 구성 매개변수가 지정되지 않은 경우, LocalSystem 어카운트가 시스템 관리자(SYSADM 권한 보유)로 간주됩니다. 버전 9.7에서는 SYSADM 권한 범위 내에서 변경된 사항이 LocalSystem에 의해 실행되는 DB2 응용프로그램에 영향을 줍니다. 이러한 응용프로그램은 일반적으로 Windows 서비스 형태로 작성되며 LocalSystem 어카운트에서 서비스 로그온 어카운트로 실행됩니다. 이러한 응용프로그램이 SYSADM 범위에 해당하지 않는 데이터베이스 조치를 수행해야 하는 경우, 필요한 특권 또는 권한을 LocalSystem 어카운트에 부여해야 합니다. 예를 들어, 응용프로그램에 데이터베이스 관리자 기능이 필요할 경우 GRANT(데이터베이스 권한)문을 사용하여 LocalSystem 어카운트에 DBADM 권한을 부여하십시오. LocalSystem 어카운트에 대한 권한 부여 ID는 SYSTEM입니다.

시스템 제어 권한(SYSCTRL)

SYSCTRL 권한은 최상위 레벨의 시스템 제어 권한입니다. 이 권한은 데이터베이스 관리 인스턴스와 데이터베이스에 대해 유지보수 및 유틸리티 조작을 수행할 수 있는 기능을 제공합니다. 이러한 조작은 시스템 자원에 영향을 미칠 수는 있으나, 데이터베이스 내의 데이터에 직접 액세스하지는 못합니다.

시스템 제어 권한은 사용자가 sensitive 데이터가 들어 있는 데이터베이스 관리 프로그램 인스턴스를 관리하기 위한 목적으로 설계되었습니다.

SYSCTRL 권한은 **sysctrl_group** 구성 매개변수로 지정한 그룹에 지정됩니다. 그룹이 지정되면, 해당 그룹 내의 멤버쉽은 사용자의 플랫폼에서 사용되는 보안 기능을 통해 데이터베이스 관리 외부에서 제어됩니다.

SYSCTRL 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 데이터베이스, 노드 또는 분산 연결 서비스(DCS) 디렉토리 갱신
- 사용자가 시스템을 강제로 끄도록 함
- 데이터베이스의 작성 또는 제거
- 테이블 스페이스의 제거, 작성 또는 변경
- 테이블 스페이스 사용
- 새 데이터베이스 또는 기존 데이터베이스로 리스토어

추가로 SYSCTRL 권한이 있는 사용자는 시스템 유지보수 권한(SYSMAINT) 및 시스템 모니터 권한(SYSMON)이 있는 사용자의 기능을 수행할 수 있습니다.

SYSCTRL 권한을 갖는 사용자는 데이터베이스에 연결할 수 있는 내재적 특권도 갖게 됩니다.

주: SYSCTRL 권한을 보유한 사용자가 데이터베이스를 작성하면 데이터베이스에 대한 명시적인 ACCESSCTRL, DATAACCESS, DBADM, 및 SECADM 권한이 사용자에게 자동으로 부여됩니다. SYSCTRL 그룹에서 데이터베이스 작성자를 제거하고 작성자가 해당 데이터베이스에 관리자로 액세스하지 못하게 하려면 이 네 권한을 명시적으로 취소하십시오.

시스템 유지보수 권한(SYSMAINT)

SYSMAINT 권한은 시스템 제어 권한 중 두 번째로 높은 레벨입니다. 이 권한은 데이터베이스 관리 인스턴스와 데이터베이스에 대해 유지보수 및 유틸리티 조작을 수행할 수 있는 기능을 제공합니다. 이러한 조작은 시스템 자원에 영향을 미칠 수는 있으나, 데이터베이스 내의 데이터에 직접 액세스하지는 못합니다.

시스템 유지보수 권한은 사용자가 중요한 데이터가 들어 있는 데이터베이스 관리 인스턴스 내의 데이터베이스를 유지보수하기 위한 목적으로 설계되었습니다.

SYSMAINT 권한은 **sysmaint_group** 구성 매개변수로 지정한 그룹에 지정됩니다. 그룹이 지정되면, 해당 그룹 내의 멤버쉽은 사용자의 플랫폼에서 사용되는 보안 기능을 통해 데이터베이스 관리 외부에서 제어됩니다.

SYSMAINT 이상의 시스템 권한을 가진 사용자만이 다음을 수행할 수 있습니다.

- 데이터베이스 또는 테이블 스페이스 백업
- 기존의 데이터베이스로 리스토어
- 롤 포워드 복구 수행
- 인스턴스 시작 또는 중지
- 테이블 스페이스 리스토어
- db2trc 명령을 사용하여 추적 실행
- 데이터베이스 관리 프로그램 인스턴스 또는 해당 데이터베이스의 데이터베이스 시스템 모니터 스냅샷 생성

SYSMAINT 권한을 보유한 사용자는 다음을 수행할 수 있습니다.

- 테이블 스페이스의 상태 쿼리
- 로그 실행기록 파일 갱신
- 테이블 스페이스 Quiesce
- 테이블 재구성
- RUNSTATS 유틸리티를 사용하여 카탈로그 통계 수집

SYSMAINT 권한을 갖는 사용자는 데이터베이스에 연결할 수 있는 내재적 특권도 갖게 되며, 시스템 모니터 권한(SYSMON)을 갖는 사용자의 기능을 수행할 수 있습니다.

시스템 모니터 권한(SYSMON)

SYSMON 권한은 데이터베이스 관리 프로그램 인스턴스 또는 해당 데이터베이스의 데이터베이스 시스템 모니터 스냅샷을 취하는 기능을 제공합니다.

SYSMON 권한은 **sysmon_group** 구성 매개변수로 지정한 그룹에 지정됩니다. 그룹이 지정되면, 해당 그룹 내의 멤버쉽은 사용자의 플랫폼에서 사용되는 보안 기능을 통해 데이터베이스 관리 외부에서 제어됩니다.

SYSMON 권한을 사용하여 사용자는 다음과 같은 명령을 실행할 수 있습니다.

- GET DATABASE MANAGER MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET SNAPSHOT
- LIST(일부 명령):
  - LIST ACTIVE DATABASES
  - LIST APPLICATIONS
  - LIST DATABASE PARTITION GROUPS
  - LIST DCS APPLICATIONS
  - LIST PACKAGES
  - LIST TABLES
  - LIST TABLESPACE CONTAINERS
  - LIST TABLESPACES
  - LIST UTILITIES
- RESET MONITOR
- UPDATE MONITOR SWITCHES

SYSMON 권한을 사용하여 사용자는 다음과 같은 API를 사용할 수 있습니다.

- db2GetSnapshot - 스냅샷 가져오기
- db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 필요한 크기 측정
- db2MonitorSwitches - 모니터 스위치 가져오기/갱신
- db2mtrk - 메모리 추적 프로그램
- db2ResetMonitor - 모니터 재설정

SYSMON 권한을 사용하여 사용자는 다음과 같은 SQL 테이블 함수를 사용할 수 있습니다.

- 이전에 SYSPROC.SNAP_WRITE_FILE을 실행하지 않은 모든 스냅샷 테이블 함수

  SYSPROC.SNAP_WRITE_FILE은 스냅샷을 취하여 해당 내용을 파일에 저장합니다. 널(NULL) 입력 매개변수를 사용하여 스냅샷 테이블 함수를 호출할 경우, 실시간 시스템 스냅샷 대신 파일 내용이 리턴됩니다.

데이터베이스 권한

각 데이터베이스 권한은 이를 보유하는 권한 부여 ID가 데이터베이스 전체에서 일부 특정 유형의 조치를 수행할 수 있게 합니다. 데이터베이스 권한은 특권과는 다릅니다. 특권은 테이블 또는 인덱스와 같은 특정 데이터베이스 오브젝트에서만 특정 조치를 취할 수 있습니다.

데이터베이스 권한은 다음과 같습니다.

ACCESSCTRL

권한 보유자가 ACCESSCTRL, DATAACCESS, DBADM, SECADM 권한 및 감사 루틴에 대한 특권을 제외한 모든 특권과 데이터베이스 권한을 부여 및 취소할 수 있도록 허용합니다.

BINDADD

보유자가 데이터베이스의 새 패키지를 작성할 수 있게 합니다.

CONNECT

보유자가 데이터베이스에 연결할 수 있게 합니다.

CREATETAB

보유자가 데이터베이스에 새 테이블을 작성할 수 있게 합니다.

CREATE_EXTERNAL_ROUTINE

보유자가 응용프로그램 및 기타 데이터베이스 사용자가 사용할 프로시저를 작성할 수 있게 합니다.

CREATE_NOT_FENCED_ROUTINE

보유자가 *비분리* UDF(User-Defined Function) 또는 프로시저를 작성할 수 있게 합니다. CREATE_EXTERNAL_ROUTINE은 CREATE_NOT_FENCED_ROUTINE이 부여된 모든 사용자에게 자동으로 부여됩니다.

주의: 데이터베이스 관리 프로그램은 비분리 프로시저 또는 UDF로부터 스토리지 또는 제어 블록을 보호하지 못합니다. 이 권한을 가진 사용자는 UDF를 비분리로 등록하기 전에 세심하게 테스트해야 합니다.

DATAACCESS

권한 보유자가 데이터베이스 테이블에 저장된 데이터에 액세스할 수 있도록 허용합니다.

DBADM

권한 보유자가 데이터베이스 관리자 역할을 할 수 있도록 허용합니다. 특히 권한 보유자에게 ACCESSCTRL, DATAACCESS 및 SECADM을 제외한 다른 모든 데이터베이스 권한을 부여합니다.

EXPLAIN

권한 보유자가 쿼리 계획에 참조된 테이블의 데이터에 액세스할 수 있는 특권 없이 해당 쿼리 계획을 설명할 수 있도록 허용합니다.

IMPLICIT_SCHEMA

모든 사용자가 아직 존재하지 않는 스키마 이름을 가진 CREATE문을 사용하여 오브젝트를 작성함으로써 내재적으로 스키마를 작성할 수 있게 합니다. SYSIBM은 내재적으로 작성된 스키마의 소유자가 되며, PUBLIC에는 이 스키마에서 오브젝트를 작성할 특권이 부여됩니다.

LOAD

보유자가 데이터를 테이블로 로드할 수 있게 합니다.

QUIESCE_CONNECT

보유자가 Quiesce 상태에서 데이터베이스에 액세스할 수 있게 합니다.

SECADM

권한 보유자가 데이터베이스에 대한 보안 관리자 역할을 할 수 있도록 허용합니다.

SQLADM

권한 보유자가 SQL문을 모니터하고 조정할 수 있도록 허용합니다.

WLMADM

권한 보유자가 워크로드 관리자 역할을 할 수 있도록 허용합니다. 특히 WLMADM 권한 보유자는 WLM(Workload Manager) 오브젝트 작성 및 삭제, WLM(Workload Manager) 특권 부여 및 취소, WLM(Workload Manager) 루틴 실행 등을 수행할 수 있습니다.

SECADM 권한을 보유한 권한 부여 ID만 ACCESSCTRL, DATAACCESS, DBADM 및 SECADM 권한을 부여할 수 있습니다. 다른 모든 권한은 ACCESSCTRL 또는 SECADM 권한을 보유한 권한 부여 ID를 사용하여 부여할 수 있습니다.

PUBLIC에서 데이터베이스 권한을 제거하려면 ACCESSCTRL 또는 SECADM 권한을 보유한 권한 부여 ID가 명시적으로 이를 취소해야 합니다.

보안 관리 권한(SECADM)

SECADM 권한은 특정 데이터베이스에 대한 보안 관리 권한으로, 사용자가 보안 관련 데이터베이스 오브젝트를 작성 및 관리하고, 모든 데이터베이스 권한 및 특권을 부여 및 취소할 수 있도록 허용합니다. 또는 보안 관리자는 감사 시스템 루틴을 실행할 수 있으며, 감사 시스템 루틴을 실행할 수 있는 사용자를 관리할 수 있습니다.

SECADM 권한이 있으면 카탈로그 테이블 및 카탈로그 뷰에서 선택할 수 있지만 사용자 테이블에 저장된 데이터에 액세스할 수는 없습니다.

SECADM 권한은 보안 관리자(SECADM 권한 보유)에 의해서만 부여될 수 있으며 사용자, 그룹 또는 역할에 부여할 수 있습니다. PUBLIC은 SECADM 권한을 직접 또는 간접적으로 얻을 수 없습니다.

SECADM 권한이 있는 사용자는 다음 조작을 수행할 수 있습니다.

- 다음 항목 작성, 변경, 삭제 및 주석 표시
  - 감사 규정
  - 보안 레이블 구성요소
  - 보안 규정
  - 트러스트된 컨텍스트
- 다음 항목 작성, 삭제 및 주석 표시
  - 역할
  - 보안 레이블
- 데이터베이스 특권 및 권한 부여 및 취소
- 다음 감사 루틴을 사용하여 지정된 태스크 수행
  - SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저 및 테이블 함수가 감사 로그를 아카이브할 수 있습니다.
  - The SYSPROC.AUDIT_LIST_LOGS 테이블 함수를 사용하면 원하는 로그를 찾을 수 있습니다.
  - SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저는 분석할 데이터를 구분된 파일로 추출합니다.

  또한 보안 관리자는 이러한 루틴에 대한 EXECUTE 특권을 부여 및 취소할 수 있으므로 원하는 경우 이러한 태스크를 위임할 수 있습니다. 보안 관리자만 이러한 루틴에 대한 EXECUTE 특권을 부여할 수 있습니다. EXECUTE 특권의 WITH GRANT OPTION은 이러한 루틴에 대해 부여할 수 없습니다(SQLSTATE 42501).
- AUDIT문을 사용하여 서버에 있는 특정 데이터베이스 또는 데이터베이스 오브젝트와 감사 규정 연관
- TRANSFER OWNERSHIP문을 사용하여 명령문의 권한 부여 ID가 소유하지 않은 오브젝트 전송

다른 권한은 이러한 기능을 제공하지 않습니다.

인스턴스 소유자는 디폴트로 SECADM 권한을 보유하지 않습니다.

보안 관리자만 다른 사용자, 그룹 또는 역할에 ACCESSCTRL, DATAACCESS, DBADM 및 SECADM 권한을 부여할 수 있습니다.

버전 9.7에서는 시스템 관리자, 데이터베이스 관리자 및 보안 관리자의 역할이 명백하게 구분되도록 DB2 권한 부여 모델이 갱신되었습니다. 이러한 개선의 결과로 SECADM

권한을 통해 제공되는 권한이 확대되었습니다. 버전 9.7 이전 릴리스에서는 SECADM 권한이 모든 특권과 권한을 부여 및 취소할 수 있는 기능을 제공하지 않았으며, 역할 또는 그룹이 아닌 사용자에게만 부여할 수 있었습니다. 또한 감사 시스템에서 정의한 프로시저 및 테이블 함수에 대한 EXECUTE 특권을 다른 사용자에게 부여하는 기능도 제공하지 않았습니다.

데이터베이스 관리 권한(DBADM)

DBADM은 특정 데이터베이스에 대한 관리 권한입니다. 데이터베이스 관리자는 오브젝트 작성 및 데이터베이스 명령 발행에 필요한 특권을 보유하고 있습니다. 또한 DBADM 권한을 보유한 사용자는 시스템 카탈로그 테이블 및 뷰에 대한 SELECT 특권을 가지고 있으며 모든 시스템 정의 DB2 루틴(감사 루틴 제외)을 실행할 수 있습니다.

DBADM 권한은 보안 관리자(SECADM 권한 보유)에 의해서만 부여 또는 취소될 수 있으며 사용자, 그룹 또는 역할에 부여할 수 있습니다. PUBLIC은 DBADM 권한을 직접 또는 간접적으로 얻을 수 없습니다.

데이터베이스에 대한 DBADM 권한을 보유한 사용자는 해당 데이터베이스에 대해 다음 조치를 수행할 수 있습니다.

- 비보안 관련 데이터베이스 오브젝트 작성, 변경 및 삭제
- 로그 파일 읽기
- 이벤트 모니터 작성, 활성화 및 삭제
- 테이블 스페이스의 상태 쿼리
- 로그 실행기록 파일 갱신
- 테이블 스페이스 Quiesce
- 테이블 재구성
- RUNSTATS 유틸리티를 사용하여 카탈로그 통계 수집

SQLADM 권한 및 WLMADM 권한은 DBADM 권한의 서브세트입니다. WLMADM 권한은 워크로드에 대한 USAGE 특권을 부여할 수 있는 추가 권한을 제공합니다.

DBADM 권한을 사용하여 DATAACCESS 권한 부여

보안 관리자는 데이터베이스 관리자가 데이터베이스 내의 데이터에 액세스할 수 있는지 여부를 지정할 수 있습니다. DATAACCESS 권한은 특정 데이터베이스 내에 있는 데이터에 액세스할 수 있도록 허용하는 권한입니다. 보안 관리자는 GRANT DBADM ON DATABASE문의 WITH DATAACCESS 옵션을 사용하여 관리자에게 이 권한을 제공할 수 있습니다. WITH DATAACCCESS 또는 WITHOUT DATAACCCESS를 지정하지 않으면 디폴트로 DATAACCESS 권한이 부여됩니다.

데이터베이스 관리자에게 권한을 부여할 때 DATAACCESS 권한을 제공하지 않으려면 SQL문에 GRANT DBADM WITHOUT DATAACCESS를 사용하십시오.

DBADM 권한을 사용하여 ACCESSCTRL 권한 부여

보안 관리자는 데이터베이스 관리자가 데이터베이스 내에서 특권을 부여 및 취소할 수 있는지 여부를 지정할 수 있습니다. ACCESSCTRL 권한은 사용자가 특정 데이터베이스 내에서 특권 및 비관리 권한을 부여 및 취소할 있도록 허용하는 권한입니다. 보안 관리자는 GRANT DBADM ON DATABASE문의 WITH ACCESSCTRL 옵션을 사용하여 관리자에게 이 권한을 제공할 수 있습니다. WITH ACCCESSCTRL 또는 WITHOUT ACCCESSCTRL을 지정하지 않으면 디폴트로 ACCESSCTRL 권한이 부여됩니다.

데이터베이스 관리자에게 권한을 부여할 때 ACCESSCTRL 권한을 제공하지 않으려면 SQL문에 GRANT DBADM WITHOUT ACCESSCTRL을 사용하십시오.

DBADM 권한 취소

보안 관리자가 DATAACCESS 또는 ACCESSCTRL 권한을 포함하는 DBADM 권한을 부여했는데 이러한 권한을 취소하려면 DATAACCESS 또는 ACCESSCTRL 권한을 명시적으로 취소해야 합니다. 예를 들어, 보안 관리자가 다음과 같이 사용자에게 DBADM 권한을 부여할 경우

```
GRANT DBADM ON DATABASE TO user1
```

디폴트로 DATAACCESS 및 ACCESSCTRL 권한도 user1에게 부여됩니다.

나중에 보안 관리자가 다음과 같이 user1로부터 DBADM 권한을 취소할 수 있습니다.

```
REVOKE DBADM ON DATABASE FROM user1
```

이제 user1에게 DBADM 권한은 없지만 DATAACCESS 및 ACCESSCTRL 권한은 그대로 남아 있습니다.

남아 있는 권한을 취소하려면 보안 관리자가 다음과 같이 이 권한을 명시적으로 취소해야 합니다.

```
REVOKE ACCESSCTRL, DATAACCESS ON DATABASE FROM user1
```

이전 릴리스에 비해 DBADM 권한에 대해 달라진 점

버전 9.7에서는 시스템 관리자, 데이터베이스 관리자 및 보안 관리자의 역할이 명백하게 구분되도록 DB2 권한 부여 모델이 갱신되었습니다. 이러한 개선의 결과로 DBADM 권한을 통해 제공되던 권한이 변경되었습니다. 버전 9.7 이전 릴리스에서는 DBADM 권한에 데이터에 액세스하고 데이터베이스에 대한 특권을 부여/취소할 수 있는 권한이 자동으로 포함되었습니다. 버전 9.7에서는 자동으로 포함되던 이러한 권한이 앞에서 설명한 DATAACCESS 및 ACCESSCTRL이라는 새 권한을 통해 제공됩니다.

또한 버전 9.7 이전 릴리스에서는 DBADM 권한을 부여하면 다음과 같은 권한도 자동으로 부여되었습니다.

- BINDADD
- CONNECT
- CREATETAB
- CREATE_EXTERNAL_ROUTINE
- CREATE_NOT_FENCED_ROUTINE
- IMPLICIT_SCHEMA
- QUIESCE_CONNECT
- LOAD

버전 9.7 이전에는 DBADM 권한이 취소될 때 위 권한은 취소되지 않았습니다.

버전 9.7에서는 위 권한이 DBADM 권한에 포함되어 버전 9.7에서 DBADM 권한이 취소되면 위 권한도 손실됩니다.

그러나 버전 9.7로 업그레이드할 때 사용자가 DBADM 권한을 보유하고 있었던 경우에는 DBADM 권한을 취소하더라도 위 권한이 손실되지 않습니다. 즉, 사용자가 버전 9.7에서 부여한 DBADM 권한을 보유하고 있는 경우에만 버전 9.7에서 DBADM 권한을 취소할 때 위 권한이 손실됩니다.

액세스 제어 관리 권한(ACCESSCTRL)

ACCESSCTRL 권한은 특정 데이터베이스 내의 오브젝트에 특권을 부여하거나 취소하는 데 필요한 권한입니다. ACCESSCTRL 권한은 테이블(카탈로그 테이블 및 뷰 제외)에 저장된 데이터에 액세스할 수 있는 고유한 특권이 없습니다.

ACCESSCTRL 권한은 SECADM 권한을 가지고 있는 보안 관리자만 부여할 수 있습니다. 이 권한은 사용자, 그룹 또는 역할에 부여할 수 있습니다. PUBLIC에는 직접 또는 간접적으로 ACCESSCTRL 권한을 부여할 수 없습니다. ACCESSCTRL 권한은 다음과 같은 조작을 수행할 수 있는 권한을 사용자에게 제공합니다.

- 다음과 같은 관리 권한 부여 및 취소:
  - EXPLAIN
  - SQLADM
  - WLMADM
- 다음과 같은 데이터베이스 권한 부여 및 취소:
  - BINDADD
  - CONNECT
  - CREATETAB
  - CREATE_EXTERNAL_ROUTINE

  - CREATE_NOT_FENCED_ROUTINE
  - IMPLICIT_SCHEMA
  - LOAD
  - QUIESCE_CONNECT
- 특권을 부여한 사람과 관계없이, 다음과 같은 오브젝트에 모든 특권 부여 및 취소:
  - 전역 변수
  - 인덱스
  - 별칭
  - 패키지
  - 루틴(감사 루틴 제외)
  - 스키마
  - 순서
  - 서버
  - 테이블
  - 테이블 스페이스
  - 뷰
  - XSR 오브젝트
- 시스템 카탈로그 테이블 및 뷰에서 SELECT 특권

이 권한은 보안 관리자(SECADM) 권한의 서브세트입니다.

데이터 액세스 관리 권한(DATAACCESS)

DATAACCESS는 특정 데이터베이스 내에서 데이터에 액세스할 수 있도록 허용하는 권한입니다.

DATAACCESS 권한은 SECADM 권한을 가지고 있는 보안 관리자만 부여할 수 있습니다. 이 권한은 사용자, 그룹 또는 역할에 부여할 수 있습니다. PUBLIC에는 직접 또는 간접적으로 DATAACCESS 권한을 부여할 수 없습니다.

모든 테이블, 뷰, 구체화된 쿼리 테이블 및 별칭에 대해서는 다음과 같은 권한과 특권을 제공합니다.

- 데이터베이스에서 LOAD 권한
- SELECT 특권(시스템 카탈로그 테이블 및 뷰 포함)
- INSERT 특권
- UPDATE 특권
- DELETE 특권

또한, DATAACCESS 권한은 다음과 같은 특권을 제공합니다.

- 모든 패키지에서 EXECUTE
- 모든 루틴에서 EXECUTE(감사 루틴 제외)

SQL 관리 권한(SQLADM)

SQLADM 권한은 SQL문을 모니터링하고 조정하는 데 필요한 권한입니다.

SQLADM 권한은 SECADM 권한을 가지고 있는 보안 관리자 또는 ACCESSCTRL 권한을 가지고 있는 사용자가 부여할 수 있습니다. SQLADM 권한은 사용자, 그룹, 역할 또는 PUBLIC에 부여할 수 있습니다. SQLADM 권한은 다음과 같은 기능을 수행할 수 있는 권한을 사용자에게 제공합니다.

- 다음과 같은 SQL문 실행 권한:
  - CREATE EVENT MONITOR
  - DROP EVENT MONITOR
  - EXPLAIN
  - FLUSH EVENT MONITOR
  - FLUSH OPTIMIZATION PROFILE CACHE
  - FLUSH PACKAGE CACHE
  - PREPARE
  - REORG INDEXES/TABLE
  - RUNSTATS
  - SET EVENT MONITOR STATE
- 다음과 같은 WLM(Workload Manager) SQL문의 특정 절 실행 권한:
  - 다음과 같은 ALTER SERVICE CLASS문의 절:
    - COLLECT AGGREGATE ACTIVITY DATA
    - COLLECT AGGREGATE REQUEST DATA
    - COLLECT REQUEST METRICS
  - 다음과 같은 ALTER THRESHOLD문의 절
    - WHEN EXCEEDED COLLECT ACTIVITY DATA

    .
  - 작업 조치를 변경할 수 있는 다음과 같은 ALTER WORK ACTION SET문의 절:
    - ALTER WORK ACTION ... COLLECT ACTIVITY DATA
    - ALTER WORK ACTION ... COLLECT AGGREGATE ACTIVITY DATA
    - ALTER WORK ACTION ... WHEN EXCEEDED COLLECT ACTIVITY DATA
  - 다음과 같은 ALTER WORKLOAD문의 절:

  - COLLECT ACTIVITY METRICS
  - COLLECT AGGREGATE ACTIVITY DATA
  - COLLECT LOCK TIMEOUT DATA
  - COLLECT LOCK WAIT DATA
  - COLLECT UNIT OF WORK DATA
- 시스템 카탈로그 테이블 및 뷰에서 SELECT 특권
- 시스템에서 정의된 모든 DB2 루틴에서 EXECUTE 특권(감사 루틴 제외)

SQLADM 권한은 데이터베이스 관리자(DBADM) 권한의 서브세트입니다.

EXPLAIN 권한은 SQLADM 권한의 서브세트입니다.

Workload 관리 권한(WLMADM)

WLMADM 권한은 특정 데이터베이스의 워크로드 오브젝트를 관리하는 데 필요한 권한입니다. 이 권한을 통해 WLM(Workload Manager) 오브젝트를 작성, 변경, 삭제, 주석 처리하고 액세스 권한을 부여하거나 권한을 취소할 수 있습니다.

WLMADM 권한은 SECADM 권한을 가지고 있는 보안 관리자 또는 ACCESSCTRL 권한을 가지고 있는 사용자가 부여할 수 있습니다. WLMADM 권한은 사용자, 그룹, 역할 또는 PUBLIC에 부여할 수 있습니다. WLMADM 권한은 다음과 같은 조작을 수행할 수 있는 권한을 사용자에게 제공합니다.

- 다음과 같은 WLM(Workload Manager) 오브젝트를 작성, 변경, 주석 처리 및 삭제할 수 있는 권한:
  - 막대 그래프 템플리트
  - 서비스 클래스
  - 임계값
  - 작업 조치 세트
  - 작업 클래스 세트
  - 워크로드
- 워크로드 특권 권한 부여 및 권한 취소
- 시스템에서 정의된 워크로드 관리 루틴 실행

WLMADM 권한은 데이터베이스 관리자(DBADM) 권한의 서브세트입니다.

Explain 관리 권한(EXPLAIN)

EXPLAIN 권한은 특정 데이터베이스의 데이터에 대한 액세스 권한 없이 쿼리 계획을 설명하는 데 필요한 권한입니다. 이 권한은 데이터베이스 관리자 권한의 서브세트로서 테이블에 저장된 데이터에 액세스할 수 있는 고유한 특권이 없습니다.

EXPLAIN 권한은 SECADM 권한을 가지고 있는 보안 관리자 또는 ACCESSCTRL 권한을 가지고 있는 사용자가 부여할 수 있습니다. EXPLAIN 권한은 사용자, 그룹, 역할 또는 PUBLIC에 부여할 수 있습니다. 이 권한을 통해 다음과 같은 SQL문을 실행할 수 있습니다.

- EXPLAIN
- PREPARE
- SELECT문 또는 XQuery문 출력에서 DESCRIBE

EXPLAIN 권한은 시스템에서 정의된 explain 루틴에 EXECUTE 특권도 제공합니다.

EXPLAIN 권한은 SQLADM 권한의 서브세트입니다.

LOAD 권한

테이블에 대한 INSERT 특권 및 데이터베이스 레벨의 LOAD 권한이 있는 사용자는 LOAD 명령을 사용하여 테이블에 데이터를 로드할 수 있습니다.

주: DATAACCESS 권한을 보유하면 LOAD 명령에 대한 전체 액세스 권한이 사용자에게 부여됩니다.

테이블에 대한 INSERT 특권 및 데이터베이스 레벨의 LOAD 권한이 있는 사용자는 이전 로드 조작이 데이터를 삽입하는 로드인 경우 LOAD RESTART 또는 LOAD TERMINATE를 수행할 수 있습니다.

테이블에 관한 INSERT 및 DELETE 특권과 데이터베이스 레벨의 LOAD 권한이 있는 사용자는 LOAD REPLACE 명령을 사용할 수 있습니다.

이전 로드 조작이 로드 바꾸기였으면 해당 사용자에게 DELETE 특권도 부여해야만 사용자가 LOAD RESTART 또는 LOAD TERMINATE를 수행할 수 있습니다.

로드 조작의 일부로서 예외 테이블이 사용되면 사용자는 예외 테이블에 대한 INSERT 특권이 있어야 합니다.

이 권한이 있는 사용자는 QUIESCE TABLESPACES FOR TABLE, RUNSTATS 및 LIST TABLESPACES 명령을 수행할 수 있습니다.

내재된 스키마 권한(IMPLICIT_SCHEMA) 고려사항

새 데이터베이스가 작성되면 PUBLIC에 IMPLICIT_SCHEMA 데이터베이스 권한이 부여됩니다. 단, CREATE DATABASE 명령에 **RESTRICTIVE** 옵션이 지정된 경우는 예외입니다.

IMPLICIT_SCHEMA 권한이 있는 사용자는 오브젝트를 작성하고 아직 존재하지 않는 스키마 이름을 지정하여 스키마를 작성할 수 있습니다. SYSIBM은 내재적으로 작성된 스키마의 소유자가 되며, PUBLIC에는 이 스키마에서 오브젝트를 작성할 특권이 부여됩니다.

데이터베이스에서 스키마 오브젝트를 내재적으로 작성할 수 있는 사용자를 제어할 필요가 있을 경우, IMPLICIT_SCHEMA 데이터베이스 권한을 PUBLIC에서 권한 취소하십시오. 일단 이렇게 되면, 스키마 오브젝트를 작성할 수 있는 방법은 다음 세 가지 밖에 없습니다.

- 모든 사용자는 CREATE SCHEMA문에 자신의 권한 부여 이름을 사용하여 스키마를 작성할 수 있습니다.
- DBADM 권한을 갖는 모든 사용자는 아직 존재하지 않는 모든 스키마를 명시적으로 작성할 수 있으며, 또다른 사용자를 스키마 소유자로 선택적으로 지정할 수 있습니다.
- DBADM 권한을 보유한 사용자는 IMPLICIT_SCHEMA 데이터베이스 권한이 있어, 다른 데이터베이스 오브젝트를 작성할 때 원하는 이름으로 스키마를 내재적으로 작성할 수 있습니다. SYSIBM은 내재적으로 작성된 스키마의 소유자가 되며, PUBLIC에는 스키마에서 오브젝트를 작성할 수 있는 특권이 부여됩니다.

특권

권한 부여 ID: SETSESSIONUSER

권한 부여 ID 특권에는 권한 부여 ID에 대한 조치가 포함됩니다. 현재 이러한 특권은 SETSESSIONUSER 하나뿐입니다.

SETSESSIONUSER 특권은 한 사용자 또는 그룹에게 부여될 수 있으며 특권 보유자는 식별을 특권이 부여된 임의의 권한 부여 ID로 전환할 수 있습니다. 식별 전환은 SET SESSION AUTHORIZATION이라는 SQL문을 사용하여 수행됩니다. SECADM 권한을 보유하는 사용자만이 SETSESSIONUSER 특권을 부여할 수 있습니다.

주: 버전 8 데이터베이스를 버전 9.1로 업그레이드하는 경우, 해당 데이터베이스에 대해 명시적 DBADM 권한을 보유한 권한 부여 ID에는 자동으로 PUBLIC에 대한 SETSESSIONUSER 권한이 부여됩니다. 이 권한은 세션 권한 부여 ID를 임의의 권한 부여 ID로 설정할 수 있는 DBADM 권한을 가진 권한 부여 ID에 의존하는 응용프로그램이 중단되지 않게 합니다. 권한 부여 ID가 SYSADM 권한은 갖고 있으나 명시적으로 DBADM 권한이 부여되지 않은 경우에는 이런 일이 발생하지 않습니다.

스키마 특권

스키마 특권은 오브젝트 특권 범주 내에 있습니다.

오브젝트 특권은 그림 3에 나와 있습니다.

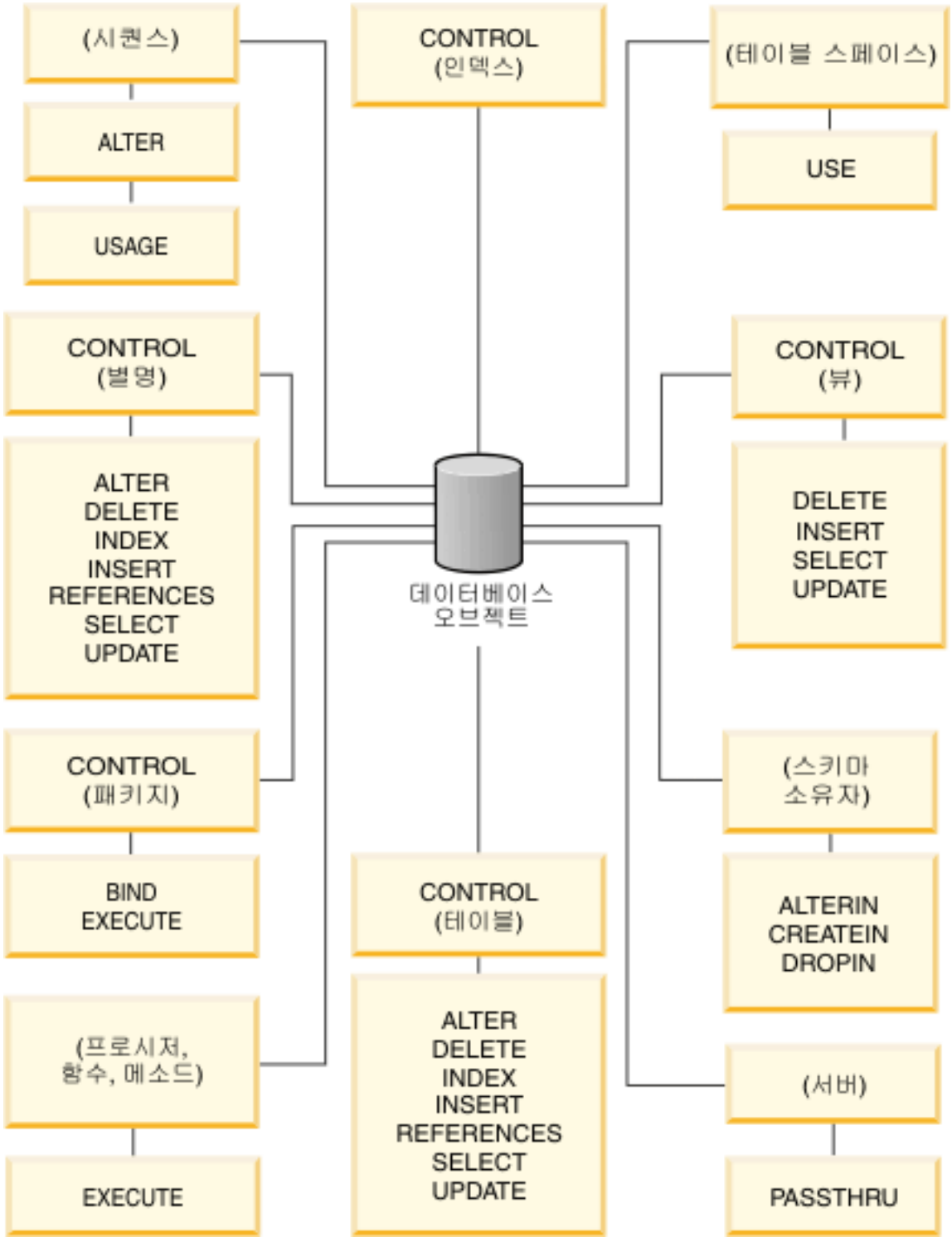


그림 3. 오브젝트 특권

스키마 권한은 데이터베이스의 스키마에 조치를 취할 수 있는 권한입니다. 사용자, 그룹, 역할 또는 PUBLIC에는 다음 특권 중 하나를 부여할 수 있습니다.

- CREATEIN은 사용자가 스키마에서 오브젝트를 작성하도록 합니다.
- ALTERIN은 사용자가 스키마에서 오브젝트를 변경하도록 합니다.
- DROPIN은 사용자가 스키마에서 오브젝트를 제거하도록 합니다.

스키마 소유자는 모두 이러한 특권과 기능을 다른 사용자에게 권한 부여할 권한을 가집니다. 스키마 오브젝트에서 조정된 오브젝트에는 테이블, 뷰, 인덱스, 패키지, 데이터 유형, 함수, 트리거, 프로시저, 별명이 있습니다.

테이블 스페이스 특권

테이블 스페이스 특권은 데이터베이스에서 테이블 스페이스에 대한 조치를 포함합니다. 사용자는 테이블 스페이스에 대한 USE 특권을 부여한 다음 해당 테이블 스페이스 내에 테이블을 작성하도록 허용할 수 있습니다.

테이블 스페이스 소유자에게는 테이블 스페이스 작성 시 테이블 스페이스에 대한 WITH GRANT OPTION이 있는 USE 특권이 부여됩니다. 또한 SECADM 또는 ACCESSCTRL 권한을 보유한 사용자는 테이블 스페이스에 대한 USE 특권도 부여할 수 있습니다.

SYSADM 또는 SYSCTRL 권한을 보유한 사용자는 모든 테이블 스페이스를 사용할 수 있습니다.

디폴트로, 데이터베이스 작성 시 테이블 스페이스 USERSPACE1에 대한 USE 특권이 PUBLIC에 부여되지만, 이 특권은 취소할 수 있습니다.

USE 특권은 SYSCATSPACE 또는 임의의 시스템 임시 테이블 스페이스에서 사용될 수 없습니다.

테이블 및 뷰 특권

테이블 및 뷰 특권에는 데이터베이스의 테이블 또는 뷰에 대한 조치가 포함됩니다.

사용자가 다음과 같은 특권을 사용하려면 데이터베이스에 대해 CONNECT 필터링합니다.

- CONTROL은 테이블 또는 뷰를 제거할 권한을 포함하여 테이블 또는 뷰에 대한 모든 특권을 제공하고, 개별 테이블 특권을 권한 부여하거나 권한 취소합니다. CONTROL 특권을 부여하려면 ACCESSCTRL 또는 SECADM 권한이 필요합니다. 테이블의 작성자는 테이블에 대해 자동으로 CONTROL 특권을 가지게 됩니다. 뷰 작성자가 뷰 정의에 언급된 모든 테이블, 뷰 및 별칭에 대한 CONTROL 특권을 보유하고 있는 경우에만 자동으로 CONTROL 특권을 부여받습니다.
- ALTER는 사용자가 테이블을 수정할 수 있도록 합니다(예를 들어, 테이블에 컬럼 또는 고유한 제한사항 추가). ALTER 특권을 갖는 사용자는 또한 테이블 또는 테이블 컬럼에서 COMMENT ON을 수행할 수도 있습니다. 테이블에서 수행할 수 있는 가능한 수정사항에 관한 정보는 ALTER TABLE 및 COMMENT문을 참조하십시오.

- DELETE는 사용자가 테이블 또는 뷰에서 행을 제거할 수 있도록 합니다.
- INDEX는 사용자가 데이터베이스에 인덱스를 작성할 수 있도록 합니다. 인덱스의 작성자는 인덱스에 대해 자동으로 CONTROL 특권을 가지게 됩니다.
- INSERT는 사용자가 테이블 또는 뷰에 행을 삽입하고 IMPORT 유틸리티를 실행할 수 있도록 합니다.
- REFERENCES는 사용자가 테이블을 관계의 상위로 지정하여, 외부 키를 작성하거나 제거할 수 있도록 합니다. 사용자는 특정 컬럼에 대해서만 이 특권을 가질 수도 있습니다.
- SELECT는 사용자가 테이블 또는 뷰에서 행을 검색하고, 테이블에 뷰를 작성하며, EXPORT 유틸리티를 실행할 수 있도록 합니다.
- UPDATE는 사용자가 테이블의 한 항목, 뷰 또는 테이블이나 뷰의 하나 이상의 특정 컬럼을 변경할 수 있도록 합니다. 사용자는 특정 컬럼에서만 이 특권을 가질 수 있습니다.

이러한 특권을 다른 사용자에게 권한 부여하는 특권은 GRANT문에서 WITH GRANT OPTION을 사용하여 권한 부여될 수도 있습니다.

주: 사용자 또는 그룹에 테이블에 대한 CONTROL 특권이 권한 부여될 경우, 해당 테이블의 다른 모든 특권에는 자동으로 WITH GRANT OPTION이 권한 부여됩니다. 나중에 사용자로부터 테이블에 대한 CONTROL 특권을 권한 취소해도, 사용자는 자동으로 권한 부여된 다른 특권을 계속 보유하게 됩니다. CONTRL 특권으로 권한 부여된 모든 특권을 권한 취소하려면, 각 특권을 명시적으로 권한 취소하거나, REVOKE문에서 ALL 키워드를 지정하십시오. 예를 들어, 다음과 같습니다.

```
 REVOKE ALL
ON EMPLOYEE FROM USER HERON
```

유형이 지정된 테이블에 대한 작업을 하는 경우, 테이블 및 뷰 특권과 관련하여 고려할 사항이 있습니다.

주: 테이블 계층의 모든 레벨에서 특권이 독립적으로 부여될 수 있습니다. 그 결과, 입력된 테이블 내의 슈퍼 테이블에서 특권을 권한 부여받은 사용자는 하위 테이블에도 간접/적접적으로 영향을 줄 수 있습니다. 그러나 해당 하위 테이블에 필수 특권이 설정되어 있는 경우, 에 직접적으로 조작을 보류하십시오.

테이블 계층 구조 내 테이블 간의 슈퍼 테이블/서브테이블 관계는 SELECT, UPDATE 및 DELETE와 같은 조작 시 대상 테이블 및 모든 서브테이블(있는 경우)의 행에 영향을 미침을 의미합니다. 이 작동을 *대체 가능성*이라 부를 수 있습니다. 예를 들어, 유형이 Manager_t인 서브테이블 Manager로 유형이 Employee_t인 Employee 테이블을 작성했다고 가정합니다. 구조화 유형 Employee_t 및 Manager_t 간의 유형/부속 유형 관계와 테이블 Employee 및 Manager 간의 테이블/서브테이블 관계로 표시되는 것과 같이, 관리자도 사원의 (특수) 유형입니다. 즉, 다음 SQL 쿼리는

```
SELECT * FROM Employee
```

사원과 관리자 모두에 대한 오브젝트 ID 및 Employee_t 속성을 리턴합니다. 이와 마찬가지로, 다음 갱신 조작은

```
UPDATE Employee SET Salary = Salary + 1000
```

정규 사원뿐 아니라 관리자의 급여를 1000달러 인상합니다.

Employee의 SELECT 특권을 가진 사용자는 Manager에 대한 명시적인 SELECT 특권을 가지고 있지 않더라도 이 SELECT 조작을 수행할 수 있습니다. 그러나 이러한 사용자는 Manager 하위 테이블에 직접 SELECT 조작을 수행할 수 없기 때문에, Manager 테이블의 상속되지 않은 컬럼에는 액세스할 수 없습니다.

마찬가지로, Employee에 대한 UPDATE 특권을 가진 사용자는 Manager에 대해 UPDATE 조작을 수행할 수 있기 때문에, Manager 테이블에 대한 명시적 UPDATE 특권을 가지고 있지 않더라도 정규 사원과 관리자 모두에게 영향을 줍니다. 그러나 이러한 사용자는 Manager 하위 테이블에 직접 UPDATE 조작을 수행할 수 없기 때문에, Manager 테이블의 상속되지 않은 컬럼은 갱신할 수 없습니다.

패키지 특권

패키지란 특정 응용프로그램에 대해 가장 효과적인 방법으로 데이터에 액세스하기 위해 데이터베이스 관리 프로그램이 필요로 하는 정보가 들어 있는 데이터베이스 오브젝트입니다. 패키지 특권은 사용자가 패키지를 작성하고 조작할 수 있도록 합니다.

사용자는 다음 특권을 사용하기 위해 데이터베이스에 대해 CONNECT 권한을 가지고 있어야 합니다.

- CONTROL은 패키지를 리바인드, 제거 또는 실행할 능력과 이러한 특권을 다른 사용자에게 확대할 수 있는 능력을 제공합니다. 패키지의 작성자는 자동으로 이 특권을 받게 됩니다. CONTROL 특권이 있는 사용자에게는 BIND 및 EXECUTE 특권이 부여되며, GRANT 명령문을 사용하여 다른 사용자에게도 이러한 특권을 부여할 수 있습니다. (WITH GRANT OPTION으로 특권이 부여된 경우, BIND 또는 EXECUTE 특권을 받은 사용자는 이 특권을 다른 사용자에게 차례로 부여할 수 있습니다.) CONTROL 특권을 부여하려면 사용자에게 ACCESSCTRL 또는 SECADM 권한이 있어야 합니다.
- 패키지에 대한 BIND 특권은 사용자가 해당 패키지를 리바인드 또는 바인드하고 패키지 이름 및 작성자가 동일한 새 패키지 버전을 추가할 수 있도록 합니다.
- EXECUTE는 사용자가 패키지를 실행할 수 있도록 합니다.

주: 모든 패키지 특권은 동일한 패키지 이름 및 작성자를 공유하는 모든 버전에 적용됩니다.

이러한 패키지 특권 외에도, 사용자는 BINDADD 데이터베이스 권한을 사용하여 데이터베이스에 새 패키지를 작성하거나 기존 패키지를 리바인드할 수 있습니다.

별칭으로 참조되는 오브젝트는 오브젝트를 포함하는 데이터 소스에 인증 점검을 전달해야 합니다. 뿐만 아니라, 패키지 사용자는 데이터 소스에 있는 데이터 소스 오브젝트에 대해 적합한 특권 또는 권한 레벨을 가지고 있어야 합니다.

DB2 데이터베이스가 DB2 계열 데이터 소스와 통신할 때 동적 쿼리를 사용하므로 별칭을 포함하는 패키지에는 추가 권한 부여 단계가 필요할 수 있습니다. 데이터 소스에서 패키지를 실행하는 권한 부여 ID는 해당 데이터 소스에서 동적으로 패키지를 실행할 권한을 가지고 있어야 합니다.

인덱스 특권

인덱스 또는 인덱스 스펙 작성자는 인덱스에 대해 자동으로 CONTROL 특권을 갖게 됩니다. 인덱스에 대한 CONTROL 특권은 인덱스를 제거할 권한이라 할 수 있습니다. 인덱스에 대한 CONTROL 특권을 부여하려면 사용자에게 ACCESSCTRL 또는 SECADM 권한이 있어야 합니다.

테이블 레벨 INDEX 특권은 사용자가 해당 테이블에 대한 인덱스를 작성할 수 있도록 합니다.

별칭 레벨 INDEX 특권은 사용자가 해당 별칭에 대한 인덱스 스펙을 작성할 수 있도록 합니다.

시퀀스 특권

시퀀스의 작성자는 자동으로 시퀀스에 관한 USAGE 및 ALTER 특권을 받게 됩니다. 시퀀스를 위한 NEXT VALUE 및 PREVIOUS VALUE 표현식을 사용하려면 USAGE 특권이 필요합니다.

기타 사용자가 NEXT VALUE 및 PREVIOUS VALUE 표현식을 사용하도록 허용하려면 시퀀스 특권을 공용(PUBLIC)으로 권한 부여해야 합니다. 이로써 모든 사용자가 지정된 시퀀스로 표현식을 사용할 수 있게 됩니다.

사용자는 시퀀스에 관한 ALTER 특권으로 시퀀스 재시작 또는 후속 시퀀스 값에 대한 증분 변경과 같은 태스크를 수행할 수 있습니다. 시퀀스의 작성자는 ALTER 특권을 다른 사용자에게 부여할 수 있으며, WITH GRANT OPTION을 사용할 경우 해당 사용자가 차례로 이들 특권을 다른 사용자에게 부여할 수 있습니다.

루틴 특권

실행 특권에는 데이터베이스 내의 모든 유형의 루틴(예: 함수, 프로시저 및 메소드)에 대한 조치가 포함됩니다. EXECUTE 특권이 있으면, 사용자는 루틴을 호출하고, 해당 루틴을 소스로 하는 함수를 작성하며(함수에만 적용), CREATE VIEW 또는 CREATE TRIGGER 등의 DDL문에서 루틴을 참조할 수 있습니다.

외부 스토어드 프로시저, 함수 또는 메소드를 정의하는 사용자는 EXECUTE WITH GRANT 특권을 받습니다. EXECUTE 특권이 WITH GRANT OPTION을 통해 또 다른 사용자에게 부여될 경우, 해당 사용자는 EXECUTE 특권을 또다른 사용자에게 차례로 부여할 수 있습니다.

워크로드에 대한 사용 설정

워크로드를 사용할 수 있도록 ACCESSCTRL, SECADM 또는 WLMADM 권한을 보유한 사용자가 GRANT USAGE ON WORKLOAD문을 사용하여 해당 워크로드에 대한 USAGE 특권을 사용자, 그룹 또는 역할에 부여할 수 있습니다.

DB2 데이터베이스 시스템은 일치하는 워크로드를 찾으면 세션 사용자가 해당 워크로드에 대해 USAGE 특권을 가지고 있는지 확인합니다. 세션 사용자가 워크로드에 대해 USAGE 특권을 가지고 있지 않을 경우, DB2 데이터베이스 시스템은 정렬된 목록에서 다음으로 일치하는 워크로드를 검색합니다. 즉, 세션 사용자가 USAGE 특권을 가지고 있지 않은 워크로드는 존재하지 않는 워크로드로 간주됩니다.

USAGE 특권 정보는 카탈로그에 저장되며 SYSCAT.WORKLOADAUTH 뷰를 통해 볼 수 있습니다.

USAGE 특권은 REVOKE USAGE ON WORKLOAD문을 사용하여 취소할 수 있습니다.

ACCESSCTRL, DATAACCESS, DBADM, SECADM 또는 WLMADM 권한을 보유한 사용자는 암시적으로 모든 워크로드에 대한 USAGE 특권을 갖습니다.

SYSDEFAULTUSERWORKLOAD 워크로드 및 USAGE 특권

데이터베이스가 RESTRICT 옵션을 사용하지 않고 작성된 경우, SYSDEFAULTUSERWORKLOAD에 대한 USAGE 특권은 데이터베이스 작성 시 PUBLIC에 부여됩니다. 그렇지 않은 경우,ACCESSCTRL, WLMADM 또는 SECADM 권한을 보유한 사용자가 USAGE 특권을 명시적으로 부여해야 합니다.

세션 사용자가 SYSDEFAULTUSERWORKLOAD를 비롯한 워크로드에 대한 USAGE 특권을 가지고 있지 않은 경우 SQL 오류가 리턴됩니다.

SYSDEFAULTADMWORKLOAD 워크로드 및 USAGE 특권

SYSDEFAULTADMWORKLOAD에 대한 USAGE 특권은 사용자에게 명시적으로 부여할 수 없습니다. SET WORKLOAD TO SYSDEFAULTADMWORKLOAD 명령을 발행하고 ACCESSCTRL을 보유한 세션 권한 부여 ID를 사용하는 사용자만 이 워크로드를 사용할 수 있습니다.

GRANT USAGE ON WORKLOAD 및 REVOKE USAGE ON WORKLOAD문은 SYSDEFAULTADMWORKLOAD에 영향을 주지 않습니다.

여러 컨텍스트에서의 권한 부여 ID

권한 부여 ID는 식별 및 권한 부여 검사에 사용됩니다. 예를 들어, 세션 권한 부여 ID는 초기 권한 부여 검사에 사용됩니다.

특정 컨텍스트에서 권한 부여 ID의 사용을 참조하는 경우 아래에 표시된 컨텍스트를 식별하기 위해 권한 부여에 대한 참조가 규정됩니다.

권한 부여 ID에 대한 컨텍스트 참조
정의

시스템 권한 부여 ID
CONNECT 처리 중 CONNECT 특권을 검사하는 것과 같이, 초기 권한 부여 검사를 수행하는 데 사용되는 권한 부여 ID입니다. CONNECT 처리 중 인증 프로세스의 일부로 DB2 이름 지정 요구사항과 호환되는 권한 부여 ID가 생성되는데, 이는 DB2 데이터베이스 시스템 내에서 외부 사용자 ID를 나타내기 위한 것입니다. 시스템 권한 부여 ID는 연결을 작성한 사용자를 나타냅니다. SYSTEM_USER 특수 레지스터를 사용하면 시스템 권한 부여 ID의 현재 값을 확인할 수 있습니다. 연결에 대한 시스템 권한 부여 ID는 변경할 수 없습니다.

세션 권한 부여 ID
CONNECT 처리 중에 수행된 초기 검사 이후에 수행되는 세션 권한 부여 검사에 사용되는 권한 부여 ID입니다. 세션 권한 부여 ID의 디폴트값은 시스템 권한 부여 ID의 값입니다. SESSION_USER 특수 레지스터를 사용하면 세션 권한 부여 ID의 현재 값을 확인할 수 있습니다. USER 특수 레지스터는 SESSION_USER 특수 레지스터의 동의어입니다. 세션 권한 부여 ID는 SET SESSION AUTHORIZATION문을 사용하여 변경할 수 있습니다.

패키지 권한 부여 ID
패키지를 데이터베이스에 바인드할 때 사용되는 권한 부여 ID입니다. 이 권한 부여 ID는 BIND 명령의 **OWNER** *authorization id* 옵션을 통해 얻을 수 있습니다. 패키지 권한 부여 ID는 패키지 바인더 또는 패키지 소유자라고도 합니다.

루틴 소유자 권한 부여 ID
호출된 SQL 루틴의 소유자로 시스템 카탈로그에 나열되는 권한 부여 ID입니다.

루틴 호출자 권한 부여 ID
SQL 루틴을 호출한 명령문에 대한 명령문 권한 부여 ID입니다.

명령문 권한 부여 ID
특정 SQL문과 연관된 권한 부여 ID로, 오브젝트 소유권(해당하는 경우)을 판별하는 데 사용될 뿐 아니라 권한 부여 요구사항에 사용됩니다. 값은 SQL문의 유형에 따라 해당하는 소스 권한 부여 ID에서 가져옵니다.

- 정적 SQL

  패키지 권한 부여 ID가 사용됩니다.
- 동적 SQL(비루틴 컨텍스트)

  다음 표는 경우에 따라 사용되는 권한 부여 ID를 보여줍니다.

패키지 발행 시 **DYNAMICRULES** 값	사용된 권한 부여 **ID**
RUN	세션 권한 부여 ID
BIND	패키지 권한 부여 ID
DEFINERUN, INVOKERUN	세션 권한 부여 ID
DEFINEBIND, INVOKEBIND	패키지 권한 부여 ID

- 동적 SQL(루틴 컨텍스트)

  다음 표는 경우에 따라 사용되는 권한 부여 ID를 보여줍니다.

패키지 발행 시 **DYNAMICRULES** 값	사용된 권한 부여 **ID**
DEFINERUN, DEFINEBIND	루틴 소유자 권한 부여 ID
INVOKERUN, INVOKEBIND	루틴 호출자 권한 부여 ID

CURRENT_USER 특수 레지스터를 사용하면 명령문 권한 부여 ID의 현재 값을 확인할 수 있습니다. 명령문 권한 부여 ID는 직접적으로 변경할 수 없습니다. 각 SQL문의 특성을 반영하기 위해 DB2 데이터베이스 시스템에 의해 자동으로 변경합니다.

데이터베이스 작성이 허용되는 디폴트 특권

데이터베이스를 작성하면 이 데이터베이스 내에서의 디폴트 데이터베이스 레벨 권한 및 디폴트 오브젝트 레벨 특권이 부여됩니다.

부여되는 권한 및 특권은 이 권한과 특권이 기록되어 있는 시스템 카탈로그 뷰에 따라 나열됩니다.

1. SYSCAT.DBAUTH
   - 데이터베이스 작성자에게는 다음과 같은 권한이 부여됩니다.
     - ACCESSCTRL
     - DATAACCESS
     - DBADM
     - SECADM
   - 비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 다음과 같은 권한이 부여됩니다.
     - CREATETAB

- BINDADD
- CONNECT
- IMPLICIT_SCHEMA

2. SYSCAT.TABAUTH

   비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 다음과 같은 특권이 부여됩니다.

   - 모든 SYSCAT 및 SYSIBM 테이블에 대한 SELECT
   - 모든 SYSSTAT 테이블에 대한 SELECT 및 UPDATE
   - 스키마 SYSIBMADM에 있는 다음과 같은 뷰에서 SELECT:
     - ALL_*
     - USER_*
     - ROLE_*
     - SESSION_*
     - DICTIONARY
     - TAB

3. SYSCAT.ROUTINEAUTH

   비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 다음과 같은 특권이 부여됩니다.

   - 스키마 SQLJ에 있는 모든 프로시저에서 GRANT로 EXECUTE
   - 스키마 SYSFUN에 있는 모든 기능과 프로시저에서 EXECUTE with GRANT
   - 스키마 SYSPROC에 있는 모든 기능과 프로시저에서 GRANT로 EXECUTE(감사 루틴 제외)
   - 스키마 SYSIBM에 있는 모든 테이블 함수에서 EXECUTE
   - 스키마 SYSIBM에 있는 다른 모든 프로시저에서 EXECUTE
   - 스키마 SYSIBMADM에서 다음과 같은 모델에서 EXECUTE:
     - DBMS_JOB
     - DBMS_LOB
     - DBMS_OUTPUT
     - DBMS_SQL
     - DBMS_UTILITY

4. SYSCAT.PACKAGEAUTH
   - 데이터베이스 작성자에게는 다음과 같은 특권이 부여됩니다.
     - NULLID 스키마에 작성된 모든 패키지에 대한 CONTROL

- NULLID 스키마에 작성된 모든 패키지에 대한 BIND with GRANT
- NULLID 스키마에 작성된 모든 패키지에 대한 EXECUTE with GRANT

• 비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 다음과 같은 특권이 부여됩니다.
- NULLID 스키마에 작성된 모든 패키지에 대한 BIND
- NULLID 스키마에 작성된 모든 패키지에서 EXECUTE

5. SYSCAT.SCHEMAAUTH

   비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 다음과 같은 특권이 부여됩니다.
   • SQLJ 스키마에 대한 CREATEIN
   • NULLID 스키마에 대한 CREATEIN
6. SYSCAT.TBSPACEAUTH

   비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 테이블 스페이스 USERSPACE1에 USE 특권이 부여됩니다.
7. SYSCAT.WORKLOADAUTH

   비제한적인 데이터베이스에서, 특수 그룹인 PUBLIC에는 SYSDEFAULTUSERWORKLOAD에 USAGE 특권이 부여됩니다.

비제한적인 데이터베이스는 CREATE DATABASE 명령에 RESTRICTIVE 옵션을 사용하지 않고 작성된 데이터베이스입니다.

액세스 권한 부여 및 권한 취소

특권 권한 부여

대부분의 데이터베이스 오브젝트에 특권을 권한 부여하려면, 해당 오브젝트에 대해 ACCESSCTRL 권한, SECADM 권한 또는 CONTROL 특권을 가지고 있거나 WITH GRANT OPTION 특권을 가지고 있어야 합니다. SYSADM 또는 SYSCTRL 권한이 있는 사용자가 테이블 스페이스 특권을 권한 부여할 수 있습니다. 기존 오브젝트에 대해서만 특권을 권한 부여할 수 있습니다.

그 밖의 다른 사용자에게 CONTROL 특권을 권한 부여하려면, ACCESSCTRL 또는 SECADM 권한이 있어야 합니다. ACCESSCTRL, DATAACCESS, DBADM 또는 SECADM 권한을 부여하려면 SECADM 권한이 있어야 합니다.

GRANT문은 권한 부여된 사용자가 특권을 권한 부여할 수 있도록 합니다. 특권을 하나의 명령문으로 하나 이상의 권한 부여 이름에 권한 부여하거나 PUBLIC에 권한 부여하여, 모든 사용자가 특권을 사용하게 할 수 있습니다. 권한 부여 이름은 개별 사용자 또는 그룹이 될 수 있습니다.

사용자와 그룹이 동일한 이름으로 존재하는 운영 체제에서 사용자 또는 그룹에 특권을 권한 부여할 것인지 여부를 지정하십시오. GRANT 및 REVOKE문은 둘 다 키워드 USER, GROUP 및 ROLE을 지원합니다. 이들 선택적 키워드가 사용되지 않는 경우, 데이터베이스 관리 프로그램은 운영 체제 보안 기능을 검사하여 권한 부여 이름이 사용자 또는 그룹을 식별하는지 여부를 판별합니다. 또한 역할 유형의 권한 부여 ID에 동일한 이름이 있는지 여부도 검사합니다. 데이터베이스 관리 프로그램이 권한 부여 이름이 사용자, 그룹 또는 역할을 나타내는지 여부를 판별할 수 없는 경우에는 오류가 리턴됩니다. 다음 예에서는 HERON 사용자에게 EMPLOYEE 테이블에 대한 SELECT 특권을 권한 부여합니다.

```
GRANT SELECT
 ON EMPLOYEE TO USER HERON
```

다음 예에서는 HERON 그룹으로 EMPLOYEE 테이블에 대한 SELECT 특권을 권한 부여합니다.

```
GRANT SELECT
ON EMPLOYEE TO GROUP HERON
```

제어 센터에서 스키마 특권 노트북, 테이블 스페이스 특권 노트북 및 뷰 특권 노트북을 사용하여 이러한 데이터베이스 오브젝트에 대한 특권을 권한 부여하고 취소할 수 있습니다. 이 노트북을 열려면 다음 단계를 따르십시오.

1. 제어 센터에서, 작업하려는 오브젝트가 있는 폴더(예: **Views** 폴더)를 찾을 때까지 오브젝트 트리를 펼치십시오.
2. 폴더를 누르십시오.

   이 폴더에 있는 모든 기존 데이터베이스 오브젝트가 컨텐츠 영역에 표시됩니다.
3. 컨텐츠 영역에서 관심 오브젝트에 마우스 오른쪽 단추로 누른 다음 팝업 메뉴에서 **특권**을 선택하십시오.

   적절한 특권 노트북이 열립니다.

특권 취소

REVOKE문은 권한 부여된 사용자가 다른 사용자에게 이미 권한 부여된 특권을 권한 취소할 수 있도록 합니다.

데이터베이스 오브젝트에 대한 특권을 권한 취소하려면, 해당 오브젝트에 대한 ACCESSCTRL 권한, SECADM 권한 또는 CONTROL 특권을 가지고 있어야 합니다. SYSADM 및 SYSCTRL 권한이 있는 사용자가 테이블 스페이스 특권의 권한을 취소할 수도 있습니다. WITH GRANT OPTION 특권만으로는 특권을 권한 취소할 수 없다는 점을 유의하십시오. 다른 사용자로부터 CONTROL 특권을 권한 취소하려면, ACCESSCTRL 또는 SECADM 권한을 가지고 있어야 합니다. ACCESSCTRL, DATAACCESS, DBADM 또는 SECADM 권한을 취소하려면 SECADM 권한이 있

어야 합니다. SYSADM 또는 SYSCTRL 권한이 있는 사용자만 테이블 스페이스 특권을 권한 취소할 수 있습니다. 특권은 기존 오브젝트에 대해서만 권한 취소될 수 있습니다.

주: ACCESSCTRL 권한, SECADM 권한 또는 CONTROL 특권을 가지고 있지 않은 사용자는 WITH GRANT OPTION을 사용하여 권한 부여한 특권을 권한 취소할 수 없습니다. 또한 권한 취소된 사용자에게 특권을 권한 부여 받은 사용자의 권한을 연쇄적으로 권한 취소할 수 없습니다.
DBADM 권한을 가지고 있는 사용자가 명시적으로 권한 부여된 테이블(또는 뷰) 특권을 권한 취소한 경우, 해당 테이블에 정의된 다른 뷰에서 특권을 권한 취소할 수 없습니다. 이는 뷰 특권이 DBDAM 권한을 통해 사용 가능하고, 기초가 되는 테이블에 대한 명시적인 특권에 종속되지 않기 때문입니다.

특권이 동일한 이름을 갖는 사용자, 그룹 또는 역할에 권한 부여되면 이 특권을 권한 취소할 때 GROUP, USER 또는 ROLE 키워드를 지정하십시오. 다음 예에서는 사용자 HERON으로부터 EMPLOYEE 테이블의 SELECT 특권을 취소합니다.

```
REVOKE SELECT
ON EMPLOYEE FROM USER HERON
```

다음 예에서는 그룹 HERON으로부터 EMPLOYEE 테이블의 SELECT 특권을 권한 취소합니다.

```
REVOKE SELECT
ON EMPLOYEE FROM GROUP HERON
```

그룹에서의 특권 취소가 해당 그룹의 모든 구성원에서 특권을 권한 취소하는 것이 아님에 유의하십시오. 특권을 개별 이름에 직접 권한 부여한 경우, 해당 특권이 직접 권한 취소될 때까지 이름을 보존합니다.

테이블 특권이 사용자로부터 권한 취소되면, 해당 사용자가 작성한 뷰에 대한 특권도 권한 취소된 테이블 특권에 종속하는 것이므로 권한 취소됩니다. 그러나 시스템에 의해 권한 부여된 특권만이 권한 취소됩니다. 또 다른 사용자에 의해 뷰에 대한 특권을 직접 권한 부여 받은 경우, 특권이 계속 보유됩니다.

테이블 특권이 사용자로부터 권한 취소되면, 해당 사용자가 작성한 뷰에 대한 특권도 권한 취소된 테이블 특권에 종속하는 것이므로 권한 취소됩니다. 그러나 시스템에 의해 권한 부여된 특권만이 권한 취소됩니다. 또 다른 사용자에 의해 뷰에 대한 특권을 직접 권한 부여 받은 경우, 특권이 계속 보유됩니다.

특권을 그룹으로 권한 부여(GRANT)한 다음, 그룹의 한 구성원으로부터 권한을 취소(REVOKE)하려는 상황이 있을 수 있습니다. 다음과 같은 오류 메시지 SQL0556N을 받지 않고 그렇게 하기 위해서는 오직 두 가지 방법만이 있습니다.

- 그룹에서 구성원을 제거하거나, 더 적은 수의 구성원이 있는 그룹 하나를 새로 작성하고 새 그룹에게 특권을 권한 부여(GRANT)할 수 있습니다.

- 그룹으로부터 특권을 권한 취소(REVOKE)한 다음, 사용자에게 개별적으로 특권을 권한 부여합니다(권한 부여 ID).

주: 테이블 또는 뷰의 사용자가 CONTROL 특권을 권한 취소하더라도, 사용자는 여전히 특권을 다른 사용자에게 권한 부여할 수 있는 권한을 지닙니다. 사용자에게 CONTROL 특권이 주어질 경우, 다른 모든 특권 WITH GRANT OPTION도 권한 부여됩니다. 일단 CONTROL이 권한 취소되면, 다른 모든 특권이 명시적으로 권한 취소될 때까지 WITH GRANT OPTION이 남아 있습니다.

권한 취소된 특권에 종속적인 모든 패키지는 올바르지 않음으로 표시되지만, 해당 권한을 갖는 사용자에 의해 리바인드되면 다시 유효성이 확인될 수 있습니다. 특권이 연속적으로 응용프로그램 바인더에 다시 권한 부여될 경우 패키지도 재빌드될 수 있으며, 응용프로그램을 실행하면 내재적 리바인드가 성공적으로 트리거됩니다. 특권이 PUBLIC으로부터 권한 취소되면, PUBLIC 특권에 근거해야만 바인드 가능한 사용자가 바인드한 모든 패키지는 유효하지 않습니다. DBADM 권한이 사용자로부터 권한 취소되면, 데이터베이스 유틸리티와 연관된 패키지를 포함하여 해당 사용자가 바인드한 모든 패키지가 올바르지 않게 됩니다. 올바르지 않음으로 표시된 패키지를 사용하려고 시도하면 시스템이 패키지를 리바인드하려고 합니다. 이러한 리바인드 시도가 실패할 경우, 오류가 발생합니다(SQLCODE-727). 이 경우, 다음과 같은 권한을 갖는 사용자가 패키지를 명시적으로 리바인드해야 합니다.

- 패키지를 리바인드할 수 있는 권한
- 패키지 내에서 사용한 오브젝트에 대한 해당 권한

이들 패키지는 특권이 권한 취소될 때마다 리바인드되어야 합니다.

하나 이상의 특권에 근거하는 트리거를 정의하고 해당 특권 중 하나 이상을 손실한 경우, 트리거 또는 SQL 기능을 사용할 수 없습니다.

오브젝트를 작성 및 삭제하여 내재적 권한 부여 관리

데이터베이스 관리 프로그램은 데이터베이스 오브젝트(예: 테이블 또는 패키지)를 작성할 수 있는 특정 특권을 사용자에게 내재적으로 권한 부여합니다. DBADM 권한을 가진 사용자가 오브젝트를 작성할 때에도 특권이 권한 부여됩니다. 마찬가지로, 특권은 오브젝트가 제거될 때 제거됩니다.

작성된 오브젝트가 테이블, 별칭, 인덱스 또는 패키지인 경우, 사용자는 오브젝트에 대한 CONTROL 특권을 받게 됩니다. 오브젝트가 뷰이면, 뷰 정의에서 참조된 모든 테이블, 뷰 및 별칭에 대해 사용자가 CONTROL 특권을 가지고 있는 경우에만, 뷰에 대한 CONTROL 특권이 내재적으로 부여됩니다.

명시적으로 작성된 오브젝트가 스키마일 경우, 스키마 소유자에게는 ALTERIN, CREATEIN 및 DROPIN 특권 WITH GRANT OPTION이 부여됩니다. 내재적으로 작성된 스키마는 PUBLIC에 권한 부여된 CREATEIN을 갖습니다.

패키지의 소유권 설정

BIND 및 PRECOMPILE 명령은 응용프로그램 패키지를 작성하거나 변경합니다. 두 경우 중 하나에서 **OWNER** 옵션을 사용하여 결과 패키지의 소유자 이름을 지정하십시오.

패키지 소유권의 이름 지정을 위한 간단한 규칙이 있습니다.

- 사용자는 자신을 소유자로서 지정할 수 있습니다. **OWNER** 옵션이 지정되지 않은 경우 이것이 디폴트값입니다.
- DBADM 권한이 있는 사용자 ID는 **OWNER** 옵션을 사용하는 소유자와 마찬가지로 모든 권한 부여 ID의 이름을 지정할 수 있습니다.

DB2 데이터베이스 제품을 사용하여 패키지를 바인드할 수 있는 모든 운영 체제가 **OWNER** 옵션을 지원하는 것은 아닙니다.

패키지를 통한 내재된 특권

데이터베이스 내의 데이터에 대한 액세스는 대화식 워크스테이션 세션에서 작업하고 있는 사용자뿐만 아니라, 응용프로그램에서도 요구할 수 있습니다. 패키지에는 사용자가 수 많은 데이터베이스 오브젝트에 대해 다양한 조치를 수행할 수 있도록 하는 명령문이 들어 있습니다. 각각의 이들 조치에는 하나 이상의 특권이 필요합니다.

패키지를 바인드하는 개인과 PUBLIC에 부여되는 특권 및 개인과 PUBLIC에 부여되는 역할은 정적 SQL 및 XQuery문이 바인드될 때 권한 부여 검사용으로 사용됩니다. 그룹을 통해 부여되는 특권과 그룹에 부여되는 역할은 정적 SQL 및 XQuery문이 바인드될 때 권한 부여 검사용으로 사용되지 않습니다.

패키지를 바인드할 때 VALIDATE RUN이 지정되지 않은 경우, 패키지를 바인드하는 유효한 권한 부여 ID를 가진 사용자는 다음과 같은 특권을 보유해야 합니다.

- 패키지의 정적 SQL 또는 XQuery문을 실행하는 데 필요한 모든 특권이 부여되어야 합니다.
- 다음 중 하나 이상의 멤버쉽을 통해 필요한 특권을 획득해야 합니다.
  - PUBLIC
  - PUBLIC에 부여된 역할
  - 사용자에게 부여된 역할

바인드 시 VALIDATE RUN이 지정된 경우, 이 패키지 내에서 정적 SQL 또는 XQuery문에 대한 모든 권한 부여가 실패하더라도 바인드가 실패하지 않으며 이러한 SQL 또는 XQuery문의 유효성이 런타임 시 재확인됩니다. PUBLIC, 그룹, 역할, 사용자 특권은 패키지를 바인드할 수 있는 적합한 권한(BIND 또는 BINDADD 특권)이 사용자에게 있는지 검사할 때 모두 사용됩니다.

패키지는 정적 및 동적 SQL 및 XQuery문을 모두 포함할 수 있습니다. 정적 쿼리로 패키지를 처리하려면 사용자가 패키지에 대해 EXECUTE 특권만 가지고 있으면 됩니다. 그러면 사용자는 패키지에 적용된 제한 범위 내에서 패키지에 있는 정적 쿼리에 대한 패키지 바인더 특권을 암시적으로 얻을 수 있습니다.

패키지에 동적 SQL 또는 XQuery문이 포함될 경우, 패키지 사전 컴파일 또는 바운드 시 **DYNAMICRULES**에 지정된 값에 따라 필요한 값이 달라집니다. 자세한 정보는 동적 쿼리에서 **DYNAMICRULES**의 결과에 대해 설명하는 주제를 참조하십시오.

별칭이 포함된 패키지를 통한 간접 권한

패키지에 별칭에 대한 참조가 들어 있으면, 패키지 작성자와 패키지 사용자에 대한 권한 부여 처리는 다소 복잡합니다.

패키지 작성자가 별칭이 들어 있는 패키지를 바인드하는 경우, 패키지 작성자는 데이터 소스에서 별칭이 참조하는 테이블과 뷰에 대해 인증 검사 또는 특권 검사를 통과할 필요는 없습니다. 그러나 패키지 실행자는 데이터 소스에서 인증 및 권한 부여 검사를 통과해야 합니다.

예를 들어, 패키지 작성자의 .SQC 파일에 여러 SQL 또는 XQuery문이 들어 있다고 가정합니다. 하나의 정적 명령문은 로컬 테이블을 참조합니다. 또 다른 동적 명령문은 별칭을 참조합니다. 패키지가 바인드되면, 패키지 작성자의 인증 ID(authid)는 로컬 테이블 및 별칭에 대한 특권을 검증하는 데 사용됩니다. 그러나 별칭이 식별하는 데이터 소스 오브젝트에 대해서는 어떠한 검사도 수행되지 않습니다. 또 다른 사용자가 패키지를 실행하면, 해당 패키지에 대해 EXECUTE 특권을 가지고 있고 사용자가 테이블을 참조하는 명령문에 대한 추가 특권 검사를 통과할 필요가 없는 것으로 가정합니다. 그러나 별칭을 참조하는 명령문의 경우, 패키지를 실행하는 사용자는 데이터 소스에서 인증 검사 및 특권 검사를 통과해야 합니다.

.SQC 파일에 동적 SQL 및 XQuery문과 테이블 및 별칭 참조의 혼합만이 포함된 경우, 로컬 오브젝트 및 별칭에 대한 DB2 데이터베이스 권한 부여 점검이 유사합니다. 패키지 사용자는 명령문 내의 로컬 오브젝트(테이블, 뷰)에 대한 특권 검사를 통과해야 하며, 또한 별칭 오브젝트에 대한 특권 검사도 통과해야 합니다(패키지 사용자는 별칭이 식별하는 오브젝트가 들어 있는 데이터 소스에서 인증 및 특권 검사를 통과해야 합니다). 어느 경우에든, 패키지 사용자는 EXECUTE 특권을 가지고 있어야 합니다.

패키지 실행자의 권한 부여 ID와 암호는 모든 데이터 소스 인증 및 특권 처리에 사용됩니다. 이 정보는 사용자 맵핑을 작성하여 변경될 수 있습니다.

주: 별칭은 정적 SQL 및 XQuery문에서 지정될 수 없습니다. **DYNAMICRULES** 옵션(BIND)은 별칭이 포함된 패키지와 함께 사용하지 마십시오.

DB2 데이터베이스가 DB2 계열 데이터 소스와 통신할 때 동적 SQL을 사용하므로 별칭을 포함하는 패키지에는 추가 권한 부여 단계가 필요할 수 있습니다. 데이터 소스에서 패키지를 실행하는 권한 부여 ID는 해당 데이터 소스에서 동적으로 패키지를 실행할 권한을 가지고 있어야 합니다.

뷰에서 데이터에 대한 액세스 제어

뷰를 통해 테이블에 대한 액세스를 제어하거나 테이블에 대한 특권을 확장할 수 있습니다.

뷰에서는 테이블에 대한 액세스를 다음과 같이 제어할 수 있습니다.

- 테이블의 지정된 컬럼에 대해서만 액세스

  테이블의 특정 컬럼에 대한 액세스만이 필요한 사용자와 응용프로그램의 경우에, 권한 부여된 사용자가 뷰를 작성하여 필요한 컬럼에만 주소지정되도록 컬럼을 제한할 수 있습니다.
- 테이블 행의 서브세트만을 액세스

  뷰 정의의 서브쿼리에 WHERE절을 지정함으로써, 권한 부여된 사용자가 뷰를 통해 주소지정되는 행을 제한할 수 있습니다.
- 데이터 소스 테이블이나 뷰에 있는 행 또는 컬럼의 서브세트에만 액세스. 별칭을 사용하여 데이터 소스에 액세스하면, 별칭을 참조하는 로컬 DB2 데이터베이스 뷰를 작성할 수 있습니다. 이들 뷰는 하나 이상의 데이터 소스에서 별칭을 참조할 수 있습니다.

  주: 둘 이상의 데이터 소스에 대한 별칭 참조사항이 들어 있는 뷰를 작성할 수 있으므로, 사용자는 하나의 뷰에서 다중 데이터 소스에 있는 데이터에 액세스할 수 있습니다. 이들 뷰를 *다중 위치 뷰*라고도 합니다. 이러한 뷰는 분산 환경을 통해 중요 테이블의 컬럼에서 정보를 조인할 때 또는 각 사용자에게 특정 오브젝트에 대한 데이터 소스에 필요한 특권이 부족할 때 유용합니다.

뷰를 작성하려면 사용자가 뷰 정의에 참조된 테이블, 뷰 또는 별칭 각각에 대해 DATAACCESS 권한이나 CONTROL 또는 SELECT 특권을 가지고 있어야 합니다. 또한 사용자는 뷰에 대해 지정된 스키마에서 오브젝트를 작성할 권한도 가지고 있어야 합니다. 즉, 기존 스키마에 대한 DBADM 권한, CREATEIN 특권 또는 데이터베이스에 대한 IMPLICIT_SCHEMA 권한(스키마가 아직 존재하지 않을 경우)을 가지고 있어야 합니다.

별칭을 참조하는 뷰를 작성하는 경우, 뷰에서 별칭이 참조되는 데이터 소스 오브젝트(테이블 및 뷰)에 대해 추가 권한이 필요하지 않습니다. 그러나 뷰 사용자는 뷰를 액세스할 때 기초 데이터 소스 오브젝트에 대해 SELECT 권한 또는 동등한 권한 부여 레벨을 가지고 있어야 합니다.

사용자가 기초 오브젝트(테이블 및 뷰)에 대한 데이터 소스에서 적합한 권한을 가지고 있지 않으면, 다음을 수행할 수 있습니다.

1. 데이터 소스 테이블에서 사용자가 액세스할 수 있는 컬럼에 대해 데이터 소스 뷰를 작성합니다.
2. 이 뷰에 대한 SELECT 특권을 사용자에게 권한 부여합니다.
3. 뷰를 참조하는 별칭을 작성합니다.

그런 다음, 새 별칭을 참조하는 SELECT문을 발행하여 컬럼에 액세스할 수 있습니다.

다음 시나리오에서는 뷰가 정보에 대한 액세스를 제한하기 위해 사용되는 방법의 상세한 예를 제공합니다.

많은 사용자가 서로 다른 이유로 STAFF 테이블의 정보를 액세스해야 할 경우가 있습니다. 예를 들어, 다음과 같습니다.

- 인사부에서는 전체 테이블을 갱신하고 볼 수 있어야 합니다.

  이러한 요구사항은 STAFF 테이블에 대한 SELECT 및 UPDATE 특권을 그룹 PERSONNL에 권한 부여함으로써 쉽게 충족될 수 있습니다.

  ```
  GRANT SELECT,UPDATE ON TABLE STAFF TO GROUP PERSONNL
  ```

- 부서 관리자는 각각 부하 직원에 대한 급여 정보를 보아야 합니다.

  이러한 요구사항은 각 부서 관리자에 대한 뷰를 작성함으로써 충족될 수 있습니다. 예를 들어, 다음과 같은 뷰가 부서 번호 51의 관리자에 대해 작성될 수 있습니다.

  ```
  CREATE VIEW EMP051 AS
        SELECT NAME,SALARY,JOB FROM STAFF
        WHERE DEPT=51
  GRANT SELECT ON TABLE EMP051 TO JANE
  ```

  권한 부여 이름 JANE을 갖는 관리자는 STAFF 테이블처럼 EMP051 뷰를 쿼리합니다. STAFF 테이블의 EMP051 뷰에 액세스하면, 이 관리자는 다음과 같은 정보를 볼 수 있습니다.

이름	SALARY	JOB
Fraye	45150.0	Mgr
Williams	37156.5	Sales
Smith	35654.5	Sales
Lundquist	26369.8	Clerk
Wheeler	22460.0	Clerk

- 모든 사용자는 다른 사원을 찾을 수 있어야 합니다. 이 요구사항은 STAFF 테이블의 NAME 컬럼과 ORG 테이블의 LOCATION 컬럼에 대한 뷰를 작성하여, DEPT 및 DEPTNUMB 컬럼 각각에 대해 두 개의 테이블을 조인함으로써 충족될 수 있습니다.

```
CREATE VIEW EMPLOCS AS
      SELECT NAME, LOCATION FROM STAFF, ORG
      WHERE STAFF.DEPT=ORG.DEPTNUMB
GRANT SELECT ON TABLE EMPLOCS TO PUBLIC
```

사원 위치 뷰에 액세스하는 사용자는 다음과 같은 정보를 볼 수 있습니다.

이름	**LOCATION**
Molinare	New York
Lu	New York
Daniels	New York
Jones	New York
Hanes	Boston
Rothman	Boston
Ngan	Boston
Kermisch	Boston
Sanders	Washington
Pernal	Washington
James	Washington
Sneider	Washington
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta
Abrahams	Atlanta
Koonitz	Chicago
Plotz	Chicago
Yamaguchi	Chicago
Scoutten	Chicago
Fraye	Dallas
Williams	Dallas
Smith	Dallas
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver
Davis	Denver
Edwards	Denver
Gafney	Denver

데이터베이스 관리자(DBA)에 대한 액세스 제어

데이터베이스 관리자(DBADM 권한을 보유한 사용자)의 데이터 액세스를 모니터, 제어 또는 방지할 수 있습니다.

데이터 액세스 모니터링

DB2 감사 기능을 사용하여 데이터베이스 관리자의 액세스를 모니터할 수 있습니다. 이를 위해서는 다음 단계를 수행하십시오.

1. DBADM 권한을 보유한 사용자에 대해 캡처할 이벤트를 모니터하는 감사 규정을 작성하십시오.
2. 이 감사 규정을 DBADM 권한과 연관시키십시오.

데이터 액세스 제어

트러스트된 컨텍스트를 역할과 함께 사용하면 데이터베이스 관리자의 액세스를 제어할 수 있습니다. 이를 위해서는 다음 단계를 수행하십시오.

1. 역할을 작성한 다음 DBADM 권한을 해당 역할에 부여하십시오.
2. 트러스트된 컨텍스트를 정의한 다음 해당 역할을 이 트러스트된 컨텍스트의 디폴트 역할로 설정하십시오.

   역할 멤버쉽을 권한 부여 ID에 명시적으로 부여하지 마십시오. 그러면 이 트러스트된 컨텍스트를 통해서만 역할을 사용할 수 있으며 사용자가 트러스트된 컨텍스트 범위 내에 있는 경우에만 DBADM 권한을 얻을 수 있습니다.
3. 다음 두 가지 방법으로 사용자가 트러스트된 컨텍스트에 액세스하는 방식을 제어할 수 있습니다.
   - 내재된 액세스: 사용자마다 고유한 트러스트된 컨텍스트를 작성합니다. 사용자가 트러스트된 컨텍스트의 속성과 일치하는 일반 연결을 설정하면 암시적으로 트러스트되어 역할에 액세스할 수 있게 됩니다.
   - 명시적 액세스: WITH USE FOR절로 트러스트된 컨텍스트를 작성하여 액세스할 수 있는 모든 사용자를 정의합니다. 사용자가 데이터베이스를 요청하는 데 사용할 수 있는 응용 프로그램을 작성합니다. 응용프로그램은 명시적으로 트러스트된 연결을 설정하므로 사용자가 요청을 발행하면 응용프로그램이 해당 사용자 ID로 전환되어 데이터베이스에서 해당 사용자로 요청을 실행합니다.

이 트러스트된 컨텍스트의 사용을 모니터하려면 관심 있는 이벤트를 캡처하는 감사 규정을 이 트러스트된 컨텍스트의 사용자에 대해 작성하십시오. 그런 다음 이 감사 규정을 트러스트된 컨텍스트와 연관시키십시오.

데이터 액세스 방지

테이블 데이터에 대한 액세스를 방지하려면 다음 옵션 중 하나를 선택하십시오.

- 모든 테이블의 데이터에 대한 액세스를 방지하려면 DBADM 사용자, 역할이나 그룹의 DATAACCESS를 취소하십시오. 또는 DATAACCESS 옵션을 사용하지 않고 원하는 사용자, 역할 또는 그룹에 DBADM을 부여할 수 있습니다.
- 하나의 특정 테이블의 데이터에 대한 액세스를 방지하려면 다음 두 단계를 수행하십시오.
  - 테이블의 모든 컬럼에 보안 레이블을 지정하십시오.
  - 해당 보안 레이블을 역할에 부여하십시오.
  - 테이블에 정당하게 액세스할 수 있는 모든 사용자(또는 역할)에게 해당 역할을 부여하십시오.

  해당 역할의 구성원이 아닌 한, 사용자의 권한과 관계없이 어떠한 사용자도 해당 테이블의 데이터에 액세스할 수 없습니다.

간접적인 방법으로 데이터에 대한 액세스 부여

보안을 성공적으로 관리하기 위해서는 사용자가 간접적인 방법으로 데이터에 액세스할 수 있는 점을 파악하고 있어야 합니다.

다음은 권한이 부여되지 않은 데이터에 액세스할 수 있는 간접적인 방법입니다.

- **카탈로그 뷰:** DB2 데이터베이스 시스템 카탈로그 뷰에는 데이터베이스 오브젝트에 대한 메타데이터 및 통계가 저장됩니다. 카탈로그 뷰에 대해 SELECT 액세스 권한이 있는 사용자는 자신들이 규정되지 않은 데이터에 대한 정보를 얻을 수 있습니다. 보안 향상을 위해 규정된 사용자만이 카탈로그 뷰에 대한 액세스 권한을 갖고 있는지 확인하십시오.

  주: DB2 Universal Database 버전 8 이하 버전에서는 디폴트로 카탈로그 뷰에 대한 SELECT 액세스 권한이 PUBLIC에게 부여되었습니다. DB2 버전 9.1 이상의 데이터베이스 시스템에서는 CREATE DATABASE 명령에 새 RESTRICTIVE 옵션을 사용하여 사용자가 카탈로그 뷰에 대한 SELECT 액세스 권한을 PUBLIC에게 부여할지 여부를 선택할 수 있습니다.
- **Visual Explain:** Visual Explain에서는 특정 쿼리를 위해 쿼리 옵티마이저가 선택한 액세스 플랜을 표시합니다. Visual Explain 정보에는 쿼리에서 참조되는 컬럼에 대한 통계도 포함되어 있습니다. 이 통계는 테이블 내용에 대한 정보를 제공합니다.
- **Explain 스냅샷:** Explain 스냅샷은 SQL 또는 XQuery문을 explain할 때 수집되는 압축된 정보입니다. 이 정보는 EXPLAIN_STATEMENT 테이블에 BLOB(Binary Large Object)로 저장되며 테이블 데이터에 대한 정보를 제공하는 컬럼 통계를 포함합니다. 보안 향상을 위해 규정된 사용자에게만 Explain 테이블에 대한 액세스 권한을 부여해야 합니다.
- **로그 판독기 기능:** 로그를 읽는 기능을 실행할 권한이 있는 사용자가 로그 레코드의 형식을 이해하지 못할 경우, 권한이 부여되지 않은 데이터에 대한 액세스 권한을 얻

을 수 있습니다. 다음 기능을 통해 로그를 읽을 수 있습니다.

함수	기능을 실행하는 데 필요한 권한
db2ReadLog	SYSADM 또는 DBADM
db2ReadLogNoConn	없음.

- **복제:** 데이터를 복제하면, 대상 위치에 보호되는 데이터도 재생됩니다. 보안 향상을 위해, 대상 위치의 보안이 최소한 소스 위치의 보안만큼 안전한지 확인하십시오.
- **예외 테이블:** 데이터를 테이블로 로드하는 중에 예외 테이블을 지정하면 예외 테이블에 대한 액세스 권한을 가진 사용자가 권한이 부여되지 않은 정보를 얻을 수 있습니다. 보안 향상을 위해, 권한이 부여된 사용자에게만 예외 테이블에 대한 액세스 권한을 부여하고 작업을 마치면 바로 예외 테이블을 삭제하십시오.
- **테이블 스페이스 또는 데이터베이스 백업:** 백업 명령을 실행할 수 있는 권한을 가진 사용자는 보호되는 데이터를 비롯하여 데이터베이스 또는 테이블 스페이스의 백업을 수행한 후 어디로든지 데이터를 리스토어할 수 있습니다. 백업에는 사용자가 액세스 권한을 갖고 있지 않은 데이터도 포함됩니다.

  백업 명령은 SYSADM, SYSCTRL 또는 SYSMAINT 권한을 갖고 있는 사용자가 실행할 수 있습니다.
- **세션 권한 부여 설정:** DB2 Universal Database 버전 8 이하 버전에서는 DBADM 권한을 보유한 사용자가 SET SESSION AUTHORIZATION SQL문을 사용하여 세션 권한 부여 ID를 모든 데이터베이스 사용자로 설정할 수 있습니다. DB2 버전 9.1 이상의 데이터베이스 시스템에서는 사용자가 세션 권한 부여 ID를 설정하기 전에 GRANT SETSESSIONUSER문을 통해 사용자에게 명시적으로 권한을 부여해야 합니다.

  그러나 기존의 버전 8 데이터베이스를 DB2 버전 9.1 이상의 데이터베이스 시스템으로 업그레이드할 때, 기존의 명시적 DBADM 권한(예: SYSCAT.DBAUTH에서 부여됨)을 보유한 사용자는 세션 권한 부여를 모든 데이터베이스 사용자로 설정할 수 있는 권한을 그대로 유지합니다. 따라서 기존의 응용프로그램은 작업을 계속할 수 있습니다. 세션 권한 부여를 설정할 수 있다면 모든 보호 데이터에 대한 액세스가 허용됩니다. 보안을 보다 강화하기 위해 REVOKE SETSESSIONUSER SQL문을 실행하여 이 설정을 겹쳐쓸 수 있습니다.
- **명령문 및 교착 상태 모니터링:** WITH VALUES절을 지정할 경우, DB2 데이터베이스 관리 시스템의 교착 상태 모니터링 활동의 일부로서 매개변수 표시문자에 연관된 값은 모니터링 출력에 기록됩니다. 모니터링 출력에 대한 액세스 권한을 가진 사용자는 권한이 부여되지 않은 정보에 대한 액세스 권한을 얻을 수 있습니다.
- **추적:** 추적에 테이블 데이터가 포함될 수 있습니다. 추적에 대한 액세스 권한이 있는 사용자는 권한이 부여되지 않은 정보에 대한 액세스 권한을 얻을 수 있습니다.

- **덤프 파일:** DB2 데이터베이스 제품은 특정 문제점을 디버그하는 데 도움이 되도록 sqllib₩db2dump 디렉토리에 메모리 덤프 파일을 생성합니다. 이러한 메모리 덤프 파일에는 테이블 데이터가 포함될 수 있습니다. 데이터가 포함된 경우, 파일에 대한 액세스 권한을 가진 사용자는 권한이 부여되지 않은 정보에 대한 액세스 권한을 얻을 수 있습니다. 보안 향상을 위해 sqllib₩db2dump 디렉토리에 대한 액세스 권한을 제한해야 합니다.
- **db2dart** db2dart 도구는 데이터베이스를 검토하여 찾은 모든 구조적 오류를 보고합니다. 도구는 테이블 데이터에 액세스할 수 있으며 DB2는 해당 액세스 권한에 대한 액세스 제어를 실시하지 않습니다. db2dart 도구를 실행할 권한이 있거나 db2dart 출력에 대한 액세스 권한이 있는 사용자는 권한이 부여되지 않은 정보에 대한 액세스 권한을 얻을 수 있습니다.
- **REOPT 바인드 옵션:** REOPT 바인드 옵션이 지정된 경우, 각 재활용 가능 증분식 바인드 SQL문에 대한 Explain 스냅샷 정보는 실행시 Explain 테이블에 배치됩니다. EXPLAIN은 입력 데이터 값도 표시합니다.
- **db2cat:** db2cat 도구는 테이블의 압축된(packed) 디스크립터를 덤프하는 데 사용됩니다. 테이블의 압축된(packed) 디스크립터에는 테이블 내용에 대한 정보를 제공할 수 있는 통계가 포함되어 있습니다. db2cat 도구를 실행하거나 출력에 대한 액세스 권한을 갖고 있는 사용자는 권한이 부여되지 않은 정보에 대한 액세스 권한을 얻을 수 있습니다.

데이터 암호화

DB2 데이터베이스 시스템은 스토리지에 있는 데이터와 네트워크를 통해 전송 중인 데이터를 암호화하는 몇 가지 방법을 제공합니다.

스토리지에 있는 데이터 암호화

스토리지에 있는 데이터는 다음 방법으로 암호화할 수 있습니다.

- 데이터베이스 테이블 내에 있는 데이터를 암호화하려면 암호화 및 암호 해독 내장 함수 ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR 및 GETHINT를 사용할 수 있습니다.
- 기본 운영 체제 데이터 및 백업 파일을 암호화하려면 IBM Database Encryption Expert를 사용할 수 있습니다.
- AIX 운영 체제에서 DB2 Enterprise Server Edition 시스템을 실행 중인데, 파일 레벨 암호화에만 관심이 있는 경우 EFS(Encrypted File System)를 사용하여 운영 체제 데이터와 백업 파일을 암호화할 수 있습니다.

전송 중인 데이터 암호화

클라이언트와 DB2 데이터베이스 사이에 전송 중인 데이터를 암호화하기 위해 DATA_ENCRYPT 인증 유형 또는 SSL(Secure Socket Layer)의 DB2 데이터베이스 시스템 지원을 사용할 수 있습니다.

ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR 및 GETHINT 함수 사용

ENCRYPT 내장 함수는 암호 기반 암호화 메소드를 사용하여 데이터를 암호화합니다. 또한 이러한 함수를 사용하여 암호 힌트를 캡슐화할 수 있습니다. 암호 힌트는 암호화된 데이터에 임베디드(embedded)됩니다. 암호화된 후에 데이터를 암호 해독하는 유일한 방법은 올바른 암호를 사용하는 것입니다. 이러한 함수를 사용할 것을 선택하는 개발자는 잊어버린 암호 및 사용 불가능한 데이터의 관리를 계획해야 합니다.

ENCRYPT 함수의 결과는 VARCHAR FOR BIT DATA입니다(한계는 32631임).

CHAR, VARCHAR 및 FOR BIT DATA만 암호화될 수 있습니다.

DECRYPT_BIN 및 DECRYPT_CHAR 함수는 암호 기반 암호 해독을 사용하여 데이터를 암호 해독합니다.

DECRYPT_BIN은 항상 VARCHAR FOR BIT DATA를 리턴하는 반면 DECRYPT_CHAR은 항상 VARCHAR을 리턴합니다. 첫 번째 인수가 CHAR FOR BIT DATA 또는 VARCHAR FOR BIT DATA일 수 있으므로, 결과가 첫 번째 인수와 같지 않은 경우가 있습니다.

결과의 길이는 다음 8바이트 경계까지의 바이트 수에 따라 다릅니다. 결과의 길이는 선택적인 힌트 매개변수가 지정될 때 데이터 인수의 길이 더하기 40 더하기 다음 8바이트 경계까지의 바이트 수일 수 있습니다. 또는 결과의 길이는 선택적인 힌트 매개변수가 지정되지 않을 때 데이터 인수의 길이 더하기 8 더하기 다음 8바이트 경계까지의 바이트 수일 수 있습니다.

GETHINT 함수는 캡슐화된 암호 힌트를 리턴합니다. 암호 힌트는 데이터 소유자가 암호를 기억하는 데 도움이 되는 구문입니다. 예를 들어 단어 『Ocean』을 암호 ″Pacific″을 기억하기 위한 힌트로 사용할 수 있습니다.

데이터를 암호화하는 데 사용되는 암호는 두 가지 방법 중 하나로 판별됩니다.

- 암호 인수. 암호는 ENCRYPT 함수가 호출될 때 명시적으로 전달되는 문자열입니다. 데이터는 제공된 암호를 사용하여 암호화 및 암호 해독됩니다.
- 암호화 암호 특수 레지스터. SET ENCRYPTION PASSWORD문은 암호 값을 암호화하며 암호화된 암호를 데이터베이스 관리 프로그램으로 전송하여 특수 레지스터에 저장합니다. 암호 매개변수 없이 호출되는 ENCRYPT, DECRYPT_BIN 및

DECRYPT_CHAR 함수는 ENCRYPTION PASSWORD 특수 레지스터의 값을 사용합니다. ENCRYPTION PASSWORD 특수 레지스터는 암호화된 양식으로만 저장됩니다.

특수 레지스터에 대한 초기 또는 디폴트값은 비어 있는 문자열입니다.

암호의 유효한 길이는 6 - 127입니다. 힌트의 유효한 길이는 0 - 32입니다.

DB2 인스턴스에서 SSL(Secure Socket Layer) 지원 구성

DB2 데이터베이스 시스템은 SSL을 지원하는데, 이것은 역시 SSL을 지원하는 DB2 클라이언트 응용프로그램이 SSL 소켓을 사용하여 DB2 데이터베이스에 연결할 수 있음을 의미합니다. CLI, CLP 및 .Net Data Provider 클라이언트 응용프로그램과 JDBC 및 SQLJ(유형 4 연결)에 대한 IBM Data Server 드라이버를 사용하는 응용프로그램이 SSL을 지원합니다.

SSL 지원을 구성하기 전에 다음 단계를 수행하십시오.

- IBM GSKit(Global Security Kit)에 대한 경로가 Windows 플랫폼에서는 **PATH** 환경 변수, Linux 및 UNIX 플랫폼에서는 **LIBPATH**, **SHLIB_PATH** 또는 **LD_LIBRARY_PATH** 환경 변수에 나타나는지 확인하십시오. GSKit은 DB2 데이터베이스 시스템을 설치할 때 자동으로 포함됩니다.

  Windows 32비트 플랫폼에서 GSKit 라이브러리가 C:¥Program Files¥IBM¥GSK8¥lib에 있습니다. 이 경우에 시스템 **PATH**에 C:¥Program Files¥IBM¥GSK8¥lib가 포함되어야 합니다. Windows 64비트 플랫폼에서 64비트 GSKit 라이브러리는 C:¥Program Files¥IBM¥GSK8¥lib64에 있으며 32비트 GSKit 라이브러리는 C:¥Program Files (x86)¥IBM¥GSK8¥lib에 있습니다.

  UNIX 및 Linux 플랫폼에서는 GSKit 라이브러리가 sqllib/lib에 위치합니다. 그러므로 **LIBPATH**, **SHLIB_PATH** 또는 **LD_LIBRARY_PATH** 환경 변수가 sqllib/lib를 포함해야 합니다.

  Windows가 아닌 플랫폼에서는 DB2 데이터베이스 관리 프로그램이 GSKit을 로컬로 설치하고, 주어진 인스턴스의 경우 GSKit 라이브러리는 sqllib/lib 또는 sqllib/lib64에 있습니다. 인스턴스를 가져오기 위해 GSKit의 다른 사본을 전역 위치에 설치할 필요는 없습니다. GSKit의 전역 사본이 존재하는 경우, 로컬 GSKit보다 높거나 동일한 버전으로 전역 GSKit의 버전을 유지할 것을 권장합니다.
- 연결 집중기(connection concentrator)가 활성화되지 않았어야 합니다. 연결 집중기(connection concentrator)가 실행 중인 경우 SSL 지원은 DB2 인스턴스에서 사용할 수 없습니다.

  연결 집중기(connection concentrator)가 활성화되었는지 여부를 판별하려면 GET DATABASE MANAGER CONFIGURATION 명령을 발행하십시오. 구성 매개변

수 **max_connections**가 **max_coordagents**의 값보다 큰 값으로 설정되는 경우, 연결 집중기(connection concentrator)가 활성화됩니다.

SSL 통신은 항상 FIPS 모드에 있습니다.

DB2 Connect에 대한 SSL 지원

DB2 클라이언트를 호스트나 System i 데이터베이스에 연결하기 위해 중간 서버 컴퓨터에서 System i용 DB2 Connect, System z®용 DB2 Connect 또는 DB2 Enterprise Server Edition을 사용 중인 경우, SSL 지원은 다음 구성 중 하나에서 사용 가능합니다.

- 클라이언트와 DB2 Connect 서버 사이
- DB2 Connect 서버와 서버 사이
- 클라이언트와 DB2 Connect 서버 사이 및 DB2 Connect 서버와 서버 사이

주: SSL 자원이 구성의 모든 경로에서 사용 가능하기 위해서는 각 클라이언트나 서버가 SSL 지원에 대한 모든 요구사항을 이행해야 합니다. 예를 들어 DB2 Connect 연결 집중기(connection concentrator)가 켜진 경우, DB2 Connect 서버에 대한 인바운드 요청은 SSL을 사용할 수 없습니다. 그러나 목표 서버에 대한 아웃바운드 요청은 SSL을 사용할 수 있습니다.

고가용성 재해 복구(HADR) 시스템에 대한 SSL 지원

클라이언트와 HADR 기본 서버 사이에서 SSL이 지원됩니다. SSL을 사용하여 HADR 기본 서버에 연결하는 클라이언트는 SSL을 사용하여 HADR 대기 데이터베이스로 리라우트할 수 있습니다. 그러나 HADR 기본 및 대기 서버 사이에서는 SSL이 지원되지 않습니다.

GSKit 도구: GSKCapiCmd의 문서

GSKit 도구 GSKCapiCmd에 대한 정보는 ftp://ftp.software.ibm.com/software/webserver/appserv/library/v61/ihs/GSK7c_CapiCmd_UserGuide.pdf에 있는 *GSKCapiCmd User's Guide*를 참조하십시오.

SSL 지원 구성

SSL 지원을 구성하기 위해 먼저 디지털 인증서를 관리할 키 데이터베이스를 작성합니다. 이러한 인증서 및 암호화 키가 SSL 연결 설정에 사용됩니다. 두 번째, DB2 인스턴스 소유자가 SSL 지원을 위한 DB2 인스턴스를 구성해야 합니다.

1. 키 데이터베이스를 작성하고 디지털 인증서를 설정하십시오.

a. GSKCapiCmd 도구를 사용하여 키 데이터베이스를 작성하십시오. CMS(Certificate Management System) 유형 키 데이터베이스여야 합니다. GSKCapiCmd는 비Java 기반 명령행 도구입니다(이 도구를 사용하기 위해 시스템에 Java™를 설치할 필요가 없음).

*GSKCapiCmd User's Guide*에서 설명하는 대로 gskcapicmd 명령을 사용하여 GSKCapiCmd를 호출합니다. 명령 경로는 Linux 및 UNIX 플랫폼에서 sqllib/gskit/bin, 32비트 및 64비트 Windows 플랫폼에서 C:₩Program Files₩IBM₩GSK8₩bin입니다. (64비트 플랫폼에, 32비트 GSKit 실행 파일 및 라이브러리도 제공됨. 이 경우 명령 경로는 C:₩Program Files (x86)₩IBM₩GSK8₩bin).

예를 들어, 다음 명령은 mydbserver.kdb라고 하는 키 데이터베이스와 mydbserver.sth라고 하는 은닉 파일을 작성합니다.

```
gsk8capicmd_64 -keydb -create -db "mydbserver.kdb" -pw
"myServerPassw0rdpw0" -stash
```

-stash 옵션은 키 데이터베이스와 동일한 경로에 파일 확장자가 .sth인 은닉 파일을 작성합니다. 인스턴스 시작 시에, GSKit은 은닉 파일을 사용하여 키 데이터베이스에 대한 암호를 얻습니다.

주: 은닉 파일에서 강력한 파일 시스템 보호를 사용해야 합니다. 디폴트로 인스턴스 소유자만 이 파일에 액세스할 수 있습니다(읽기 및 쓰기 액세스).

키 데이터베이스는 작성될 때 자동으로 Verisign 같은 소수의 인증 기관(CA)의 서명자 인증서로 채워집니다.

b. 서버의 인증서를 키 데이터베이스에 추가하십시오. 서버는 SSL 응답 확인 방식 중에 클라이언트로 이 인증서를 전송하여 서버의 인증을 제공합니다. 인증서를 얻기 위해 GSKCapiCmd를 사용하여 새 인증서 요청을 작성하여 이를 서명할 CA에 제출하거나 테스트 목적으로 자체 서명된 인증서를 작성할 수 있습니다.

예를 들어, myselfsigned 레이블이 지정된 자체 서명된 인증서를 작성하려면 다음 예에 표시된 대로 GSKCapiCmd 명령을 사용하십시오.

```
gsk8capicmd_64 -cert -create -db "mydbserver.kdb" -pw "myServerPassw0rdpw0"
       -label "myselfsigned" -dn "CN=myhost.mycompany.com,O=myOrganization,
       OU=myOrganizationUnit,L=myLocation,ST=ON,C=CA"
```

c. 방금 작성한 인증서를 파일로 추출하여 사용자의 DB2 서버로의 SSL 연결을 설정할 클라이언트를 실행하는 컴퓨터에 분배할 수 있습니다.

예를 들어, 다음 GSKCapiCmd 명령은 인증서를 mydbserver.arm라고 하는 파일에 추출합니다.

```
gsk8capicmd_64 -cert -extract -db "mydbserver.kdb" -pw "myServerPassw0rdpw0"
      -label "myselfsigned" -target "mydbserver.arm" -format ascii -fips
```

2. DB2 서버를 SSL 지원을 위해 설정하려면 DB2 인스턴스 소유자로서 로그인하고 다음 구성 매개변수 및 **DB2COMM** 레지스트리 변수를 설정하십시오.

   a. **ssl_svr_keydb** 구성 매개변수를 키 데이터베이스 파일의 완전한 경로로 설정하십시오. 예를 들어, 다음과 같습니다.

   ```
   db2 update dbm cfg using SSL_SVR_KEYDB /home/test/sqllib/security/keystore/key.kdb
   ```

   ssl_svr_keydb가 널(NULL)(설정되지 않음)인 경우 SSL 지원이 사용 가능하지 않습니다.

   b. **ssl_svr_stash** 구성 매개변수를 은닉 파일의 완전한 경로로 설정하십시오. 예를 들어, 다음과 같습니다.

   ```
   db2 update dbm cfg using SSL_SVR_STASH /home/test/sqllib/security/keystore/mydbserver.sth
   ```

   ssl_svr_stash가 널(NULL)(설정되지 않음)인 경우 SSL 지원이 사용 가능하지 않습니다.

   c. **ssl_svr_label** 구성 매개변수를 1단계에서 추가한 서버의 디지털 인증서 레이블로 설정하십시오. **ssl_svr_label**이 설정되지 않은 경우 키 데이터베이스의 디폴트 인증서가 사용됩니다. 키 데이터베이스에 디폴트 인증서가 없는 경우 SSL은 사용 불가능합니다. 예: `db2 update dbm cfg using SSL_SVR_LABEL myselfsigned`, 여기서 *myselfsigned*는 샘플 레이블입니다.

   d. **ssl_svcename** 구성 매개변수를 DB2 데이터베이스 시스템이 SSL 연결에 대해 대기(listen)할 포트로 설정하십시오. TCP/IP 및 SSL이 둘 다 사용 가능한 경우(**DB2COMM** 레지스트리 변수가 'TCPIP, SSL'로 설정됨), **ssl_svcename**을 **svcename**이 설정되는 포트와는 다른 포트로 설정해야 합니다. **svcename** 구성 매개변수는 DB2 데이터베이스 시스템이 TCP/IP 연결에 대해 대기(listen)하는 포트로 설정합니다. **ssl_svcename**을 **svcename**과 동일한 포트로 설정하는 경우 TCP/IP 및 SSL이 사용 불가능합니다. **ssl_svcename**이 널(NULL)(설정되지 않음)인 경우 SSL 지원이 사용 가능하지 않습니다.

   주: HADR 환경에서는 기본 또는 대기 데이터베이스 시스템의 **hadr_local_svc**를 **ssl_svcename**에 대해 설정하는 것과 동일한 값으로 설정하십시오. 또한 **hadr_local_svc**를 **svcename**, 또는 **svcename** 더하기 1과 같은 값으로 설정하지 마십시오.

   주: DB2COMM 레지스트리 변수가 'TCPIP,SSL'로 설정될 때, TCPIP 지원이 예를 들어 널(null)로 설정되는 **svcename** 구성 매개변수로 인해 적절하게 사용 가능하지 않은 경우 SQL5043N 오류가 리턴되고 SSL 지원이 사용 가능하지 않습니다.

   e. (선택적) 서버가 사용할 수 있는 암호 제품을 지정하려는 경우 **ssl_cipherspecs** 구성 매개변수를 설정하십시오. **ssl_cipherspecs**를 널(NULL)(설정되지 않음)로 두면 GSKit이 클라이언트와 서버 둘 다 지원하는 가장 강한 사용 가능한 암호

제품을 선택할 수 있습니다. 사용 가능한 암호 제품에 대한 정보는 82 페이지의 『지원되는 암호 제품』을 참조하십시오.

f. 값 SSL을 **DB2COMM** 레지스트리 변수에 추가하십시오. 예를 들어, 다음과 같습니다.

```
db2set -i db2inst1 DB2COMM=SSL
```

여기서 *db2inst1*은 DB2 인스턴스 이름입니다. 데이터베이스 관리 프로그램은 여러 프로토콜을 동시에 지원할 수 있습니다. 예를 들어 TCP/IP 및 SSL 통신 프로토콜을 둘 다 사용 가능하게 하려면 다음을 수행하십시오.

```
db2set -i db2inst1 DB2COMM=SSL,TCPIP
```

g. DB2 인스턴스를 재시작하십시오. 예를 들어, 다음과 같습니다.

```
db2stop
db2start
```

비Java DB2 클라이언트에서 SSL(Secure Socket Layer) 지원 구성

DB2 서버와의 통신을 위해 SSL(Secure Socket Layer)을 지원하도록 DB2 데이터베이스 클라이언트(예: CLI, CLP 및 .Net Data Provider 클라이언트)를 구성할 수 있습니다.

주: 버전 9.7 DB2 클라이언트 또는 DB2 Connect 게이트웨이가 z/OS V1.8, V1.9 또는 V1.10 시스템에서 z/OS 서버용 DB2에 SSL 연결을 설정하는 경우, 적절한 APAR PK72201용 PTF를 z/OS IP 서비스용 통신 서버에 적용해야 합니다.

클라이언트에 대한 SSL 지원을 구성하기 전에 다음 단계를 수행하십시오.

- 클라이언트와 서버 모두가 동일한 실제 컴퓨터에 있는 경우, GSKit이 DB2 서버와 함께 자동으로 설치되기 때문에 GSKit을 설치할 필요가 없습니다.

  Windows가 아닌 플랫폼에서는 DB2 데이터베이스 관리 프로그램이 GSKit을 로컬로 설치하고, 주어진 인스턴스의 경우 GSKit 라이브러리는 `sqllib/lib` 또는 `sqllib/lib64`에 있습니다. GSKit의 다른 사본을 전역 위치에 설치할 필요는 없습니다. GSKit의 전역 사본이 존재하는 경우, 로컬 GSKit보다 높거나 동일한 버전으로 전역 GSKit의 버전을 유지할 것을 권장합니다.
- ″C″ 기반 클라이언트의 경우 클라이언트가 별도의 컴퓨터에 설치될 경우, 클라이언트가 SSL을 사용하여 서버와 통신하는 경우 GSKit을 설치해야 합니다. SSL 기능 사용에 필요한 IBM DB2 지원 파일 DVD에서 GSKit 라이브러리를 설치할 수 있습니다. 또는 Passport Advantage®에서 다운로드한 이미지에서 설치할 수 있습니다.
  - IBM GSKit(Global Security Kit)에 대한 경로가 Windows에서는 **PATH** 환경 변수, Linux 및 UNIX에서는 **LIBPATH**, **SHLIB_PATH** 또는 **LD_LIBRARY_PATH** 환경 변수에 나타나는지 확인하십시오. 예를 들어 Windows에서는 GSKit `bin` 및 `lib` 디렉토리를 PATH 환경 변수에 추가하십시오.

```
set PATH="C:¥Program Files¥ibm¥gsk8¥bin";%PATH%
set PATH="C:¥Program Files¥ibm¥gsk8¥lib";%PATH%
```

GSKit 도구: GSKCapiCmd의 문서

GSKit 도구 GSKCapiCmd에 대한 정보는 ftp://ftp.software.ibm.com/software/webserver/appserv/library/v61/ihs/GSK7c_CapiCmd_UserGuide.pdf에 있는 *GSKCapiCmd User's Guide*를 참조하십시오.

SSL 통신은 항상 FIPS 모드에 있습니다.

SSL 지원 구성

DB2 클라이언트에서 SSL 지원을 구성하려면 다음을 수행하십시오.

1. 클라이언트의 서버 디지털 인증서의 서명자 인증서를 설정하십시오. 자체 서명 인증서나 인증 기관(CA)이 서명한 인증서 중 하나를 서버 인증서로 사용할 수 있습니다.
   - 서버 인증서가 자체 서명 인증서인 경우, 이 인증서를 서버 컴퓨터의 파일에 추출한 후 해당 서버에 SSL 연결을 설정하려는 클라이언트를 실행하고 있는 컴퓨터에 분배해야 합니다. 인증서를 파일에 추출하는 방법에 대한 정보는 71 페이지의 『DB2 인스턴스에서 SSL(Secure Socket Layer) 지원 구성』을 참조하십시오.
   - 서버 인증서가 알려진 CA에서 서명한 경우, 해당 서버 인증서를 서명한 CA 인증서가 클라이언트 키 데이터베이스에 이미 있을 수 있습니다. 이 인증서가 없는 경우, CA 인증서를 가져와야 합니다. CA 웹 사이트를 방문하여 가져올 수 있습니다.
2. DB2 클라이언트 컴퓨터에서, GSKCapiCmd 도구를 사용하여 CMS 유형의 키 데이터베이스를 작성하십시오. GSKCapiCmd 도구는 비Java 기반 명령행 도구입니다(이 도구를 사용하기 위해 시스템에 Java를 설치할 필요가 없음).

   *GSKCapiCmd User's Guide*에서 설명하는 대로 gskcapicmd 명령을 사용하여 GSKCapiCmd를 호출합니다. 명령 경로는 Linux 및 UNIX 플랫폼에서 `sqllib/gskit/bin`, 32비트 및 64비트 Windows 플랫폼에서 `C:¥Program Files¥IBM¥GSK8¥bin`입니다. (64비트 플랫폼에, 32비트 GSKit 실행 파일 및 라이브러리도 제공됨. 이 경우 명령 경로는 `C:¥Program Files (x86)¥IBM¥GSK8¥bin`).

   예를 들어, 다음 명령은 `mydbclient.kdb`라고 하는 키 데이터베이스와 `mydbclient.sth`라고 하는 은닉 파일을 작성합니다.

   ```
   gsk8capicmd_64 -keydb -create -db "mydbclient.kdb" -pw "myClientPassw0rdpw0" -stash
   ```

-stash 옵션은 키 데이터베이스와 동일한 경로에 파일 확장자가 .sth인 은닉 파일을 작성합니다. 연결 시에, GSKit는 은닉 파일을 사용하여 키 데이터베이스의 암호를 얻습니다.

3. 클라이언트 키 데이터베이스에 서명자 인증서 추가

   예를 들어, 다음 gsk8capicmd 명령은 mydbserver.arm 파일에서 mydbclient.kdb라고 하는 키 데이터베이스로 인증서를 임포트합니다.

   ```
   gsk8capicmd_64 -cert -add -db "mydbclient.kdb" -pw "myClientPassw0rdpw0"
         -label "dbselfsigned" -file "mydbserver.arm" -format ascii -fips
   ```

4. 클라이언트 응용프로그램의 경우, 해당 클라이언트에 적합한 예에 표시된 것처럼 적합한 연결 문자열이나 구성 매개변수를 설정하십시오.

예

CLP 및 Embedded SQL 클라이언트

CLP 클라이언트와 Embedded SQL 클라이언트는 CATALOG TCPIP NODE 명령을 사용하여 노드 카탈로그에 추가된 리모트 호스트의 데이터베이스에 연결할 수 있습니다. "SSL"로 설정된 SECURITY 키워드와 함께 CATALOG TCPIP NODE 명령을 발행하여 해당 연결에 대한 SSL을 지정하십시오.

다음 예는 CLP 클라이언트가 SSL 연결을 사용하여 연결할 수 있도록 노드 및 데이터베이스를 카탈로그하는 방법을 보여줍니다.

첫 번째, 노드 및 데이터베이스를 카탈로그하여 클라이언트 응용프로그램이 SSL 연결을 설정할 수 있도록 하십시오.

```
catalog TCPIP NODE mynode  REMOTE  127.0.0.1  SERVER 50001  SECURITY SSL

catalog DATABASE sample  AS myssldb  AT NODE mynode  AUTHENTICATION SERVER
```

다음으로, **ssl_clnt_keydb** 및 **ssl_clnt_stash** 구성 매개변수를 사용하여 클라이언트 키 데이터베이스 및 은닉 파일을 지정하십시오. **ssl_clnt_keydb** 구성 매개변수를 키 데이터베이스 파일(.kdb)의 완전한 경로로, **ssl_clnt_stash** 구성 매개변수를 은닉 파일의 완전한 경로로 설정합니다.

```
db2 update dbm cfg using SSL_CLNT_KEYDB /home/test1/sqllib/security/keystore/clientkey.kdb
SSL_CLNT_STASH /home/test1/sqllib/security/keystore/clientstore.sth
```

ssl_clnt_keydb 또는 **ssl_clnt_stash** 구성 매개변수가 널(NULL)인 경우(설정되지 않은 경우), 연결은 실패하며 오류 SQL30081N을 리턴합니다.

그런 다음 CLP 클라이언트에서 서버에 연결하십시오.

```
db2 connect to mynode user user1 using password
```

다른 방법으로, Embedded SQL 응용프로그램은 다음 명령문을 사용하여 연결할 수 있습니다.

```
Strcpy(dbAlias,"myssldb");
 EXEC SQL CONNECT TO :dbAlias USER :user USING :pswd;
```

CLI/ODBC 클라이언트 응용프로그램

CLI 응용프로그램을 실행 중인 환경에 따라서, 연결 문자열 매개변수 (ssl_client_keystoredb 및 ssl_client_keystash) 또는 DB2 구성 매개변수(**ssl_clnt_keydb** 및 **ssl_clnt_stash**)를 사용하여 클라이언트 키 데이터베이스 및 은닉 파일에 대한 경로를 지정합니다.

- ODBC 및 CLI에 대해 IBM Data Server 드라이버를 사용 중인 경우, 다음 예에서 보는 것처럼 연결 문자열 매개변수를 사용합니다.

  SECURITY=SSL 키워드를 포함하는 연결 문자열과 함께 SQLDriverConnect를 호출하십시오. 예를 들어, 다음과 같습니다.

  ```
  "Database=sampledb; Protocol=tcpip; Hostname= myhost; Servicename=50001;
  Security=ssl; Ssl_client_keystoredb=/home/test1/keystore/clientstore.kdb;
  Ssl_client_keystash=/home/test1/keystore/clientstore.sth;"
  ```

  이 경우에 Security=ssl이 지정되기 때문에 ssl_client_keystoredb 및 ssl_client_keystash 연결 문자열 매개변수가 설정되어야 하며, 그렇지 않으면 연결은 실패합니다.

- IBM Data Server Client 또는 IBM Data Server Runtime Client를 사용 중인 경우, 연결 문자열 매개변수 또는 DB2 구성 매개변수를 사용하여 클라이언트 키 데이터베이스 및 은닉 파일에 대한 경로를 설정할 수 있습니다. ssl_client_keystoredb 및 ssl_client_keystash 연결 문자열 매개변수가 설정되는 경우 **ssl_clnt_keydb** 또는 **ssl_clnt_stash** 구성 매개변수에 의해 설정되는 모든 값을 겹쳐씁니다.

  이 예는 db2cli.ini 파일을 사용하여 연결 문자열 매개변수를 설정합니다.

  ```
  [sampledb]
  Database=sampledb
  Protocol=tcpip
  Hostname=myhost
  Servicename=50001
  Security=ssl
  SSL_client_keystoredb=/home/test1/keystore/clientstore.kdb
  SSL_client_keystash=/home/test1/keystore/clientstore.sth
  ```

  이 예는 **FileDSN** CLI/ODBC 키워드를 사용하여 데이터베이스 연결성 정보를 포함하는 DSN 파일을 식별하는데, 이것은 연결 문자열 매개변수를 설정합니다. 예를 들어 DSN 파일은 다음과 비슷할 수 있습니다.

  ```
  [ODBC]
  DRIVER=IBM DB2 ODBC DRIVER - DB2COPY1
  UID=user1
  AUTHENTICATION=SERVER
  PORT=50001
  HOSTNAME=myhost
  PROTOCOL=TCPIP
  ```

```
DATABASE=SAMPLEDB
SECURITY=SSL
SSL_client_keystoredb=/home/test1/keystore/clientstore.kdb
SSL_client_keystash=/home/test1/keystore/clientstore.sth
```

이러한 경우에 Security=ssl이 지정되기 때문에 ssl_client_keystoredb 및 ssl_client_keystash 연결 문자열 매개변수가 설정되지 않고 **ssl_clnt_keydb** 및 **ssl_clnt_stash** 구성 매개변수도 설정되지 않는 경우 연결은 실패합니다.

DB2 .Net Data Provider 응용프로그램

DB2 .Net Data Provider 응용프로그램은 연결 문자열 매개변수 SSLClientKeystoredb 및 SSLClientKeystash를 정의하여 클라이언트 키 데이터베이스 및 은닉 파일에 대한 경로를 지정하여 데이터베이스에 대한 SSL 연결을 설정할 수 있습니다. 연결 문자열은 Security=SSL도 포함해야 합니다. 예를 들어, 다음과 같습니다.

```
String connectString = "Server=myhost:50001;Database=sampledb;Security=ssl;
SSLClientKeystoredb=/home/test1/keystore/clientstore.kdb;
SSLClientKeystash=/home/test1/keystore/clientstore.sth";
```

그런 다음, 다음 C# 코드 조각에서 보는 것처럼 데이터베이스에 연결하려면 이 connectString을 DB2Connection 컨스트럭터로 전달하고 DB2Connection 오브젝트의 Open 메소드를 사용하여 connectString에서 식별되는 데이터베이스에 연결하십시오.

```
DB2Connection conn = new DB2Connection(connectString);
Conn.Open();
Return conn;
```

연결 문자열 매개변수 SSLClientKeystoredb 또는 SSLClientKeystash 중 하나가 널(NULL)(설정되지 않음)인 경우 연결은 실패하며 오류 SQL30081N을 리턴합니다.

SSL(Secure Socket Layer)

DB2 데이터베이스 시스템은 SSL(Secure Socket Layer) 및 후속 버전인 TLS(Transport Layer Security)의 사용을 지원하여 클라이언트가 서버를 인증할 수 있도록 하고 암호화를 통해 클라이언트와 서버 간의 개인용 통신을 제공합니다. 인증은 디지털 인증서의 교환에 의해 수행됩니다.

주: 이 주제에서 SSL을 언급할 때, 별도로 언급되지 않으면 동일한 정보가 TLS에 적용됩니다.

암호화가 없으면 정보 패킷이 액세스 권한이 있는 누구나 볼 수 있는 네트워크를 통해 이동합니다. SSL을 사용하여 TCP/IP를 사용하는 모든 네트워크에서 전송 중인 데이터를 보호할 수 있습니다(SSL 연결을 보안 TCP/IP 연결로 생각할 수 있음).

클라이언트와 서버는 ″SSL 응답 확인 방식″을 수행하여 보안 SSL 연결을 설정합니다.

SSL 응답 확인 방식의 개요

SSL 응답 확인 방식 중에, 대개 RSA인 공용 키 알고리즘이 클라이언트와 서버 사이에 디지털 서명과 암호화 키를 안전하게 교환하는 데 사용됩니다. 이 ID와 키 정보가 클라이언트와 서버 간의 세션에 대한 보안 연결을 설정하는 데 사용됩니다. 보안 세션이 설정된 후, 클라이언트와 서버 간의 데이터 전송은 AES 같은 대칭 알고리즘을 사용하여 암호화됩니다.

클라이언트와 서버는 SSL 응답 확인 방식 중에 다음 단계를 수행합니다.

1. 클라이언트가 SSL 연결을 요청하고 지원되는 암호 제품을 나열합니다.
2. 서버가 선택된 암호 제품으로 응답합니다.
3. 서버가 클라이언트로 디지털 인증서를 전송합니다.
4. 클라이언트가 인증 목적을 위해 서버 인증서의 유효성을 검증합니다. 서버 인증서를 발행한 트러스트된 인증 권한을 사용하거나 고유한 키 데이터베이스를 점검하여 이를 수행할 수 있습니다.
5. 클라이언트와 서버가 세션 키 및 메시지 인증 코드(MAC)를 안전하게 협상합니다.
6. 클라이언트와 서버가 선택된 키와 MAC를 사용하여 안전하게 정보를 교환합니다.

주: DB2 데이터베이스 시스템은 SSL 응답 확인 방식 중에 클라이언트의 (선택적) 인증을 지원하지 않습니다.

DB2 인증과 함께 SSL 암호화 사용

모든 기존 DB2 인증 메소드(예: KERBEROS 또는 SERVER)와 함께 SSL 암호화를 사용할 수 있습니다. DBM 구성 매개변수에 있는 인스턴스에 대한 인증 유형을 사용자가 선택하는 인증 메소드로 설정하여 일반적인 경우와 같이 이를 수행합니다.

디지털 인증서와 인증 기관

디지털 인증서는 클라이언트나 서버 같은 엔티티의 ID를 검증하기 위해 인증 기관이라는 트러스트된 당사자에 의해 발행됩니다.

디지털 인증서는 두 가지 목적에 이용됩니다. 소유자의 ID를 검증하고 소유자의 공용 키를 사용 가능하게 만듭니다. 만기일과 함께 발행되며, 만기일 이후에는 인증 기관(CA)이 더 이상 인증서를 보증하지 않습니다.

디지털 인증서를 얻으려면 사용자가 선택하는 CA(예: Verisign 또는 RSA)에 요청을 전송합니다. 요청에는 사용자의 식별 이름, 공용 키 및 시그니처가 들어 있습니다. 식별 이름(DN)은 인증서를 적용하려는 각 사용자 또는 호스트에 대한 고유한 ID입니다. CA는 공용 키를 사용하여 사용자의 시그니처를 점검하고 사용자 ID의 몇 가지 검증 레벨을 수행합니다(이것은 CA에 따라 다름). 검증 후, CA가 사용자에게 사용자의 식별 이름과 공용 키, CA의 식별 이름 및 인증 기관의 시그니처를 포함하는 서명된 디지털 인증서를 전송합니다. 이 서명된 인증서를 사용자의 키 데이터베이스에 저장합니다.

이 인증서를 수신자에게 보내면 수신자는 2단계를 수행하여 사용자의 ID를 검증합니다.

1. 인증서와 함께 제공되는 공용 키를 사용하여 디지털 시그니처를 확인합니다.
2. 인증서를 발행한 CA가 적법하고 신뢰할 수 있는지 검증합니다. 이를 위해 수신자는 CA의 공용 키가 필요합니다. 수신자가 이미 자신의 키 데이터베이스에 CA 공용 키의 보증된 사본을 보유할 수 있지만, 그렇지 않은 경우 수신자는 CA의 공용 키를 얻기 위해 추가 디지털 인증서가 필요할 수 있습니다. 이 인증서는 다시 다른 CA의 디지털 인증서에 의존할 수 있습니다. 각각이 다음 CA의 유효성에 의존하는 다중 CA가 발행하는 인증서의 계층 구조가 있을 수 있습니다. 그러나 수신자는 최종적으로 루트 *CA*의 공용 키가 필요합니다. 루트 CA는 계층 구조의 맨 위에 있는 CA입니다. 루트 CA의 디지털 인증서의 유효성을 신뢰하기 위해서는 공용 키 사용자가 인증된 서버로부터의 다운로드를 통해, 신뢰할 수 있는 소스로부터 수신된 사전 로드된 소프트웨어를 사용하여 또는 안전하게 전달된 디스켓을 통하는 등의 안전한 방식으로 디지털 인증서를 수신해야 합니다.

디지털 인증서를 수신자에게 전송하는 많은 응용프로그램은 자신의 인증서뿐 아니라 루트 CA 인증서까지의 인증서의 계층 구조를 검증하기 위해 필요한 모든 CA 디지털 인증서를 전송합니다.

디지털 인증서를 전적으로 신뢰할 수 있기 위해서는 디지털 인증서의 소유자는 예를 들어 컴퓨터의 하드 드라이브에서 인증서를 암호화하여 개인용 키를 주의깊게 보호해야 합니다. 개인용 키가 손상된 경우, 부정 사용자가 디지털 인증서를 오용할 수 있습니다.

테스트 목적으로 자체 서명 디지털 인증서를 사용할 수 있습니다. 자체 서명 디지털 인증서에는 사용자의 식별 이름, 공용 키 및 시그니처가 들어 있습니다.

공용 키 암호 해독법

SSL은 공용 키 알고리즘을 사용하여 인증을 위해 암호화 키 정보와 디지털 인증서 정보를 교환합니다. 공용 키 암호 해독법(비대칭 암호 해독법이라고도 함)은 두 가지 암호화 키를 사용하는데, 하나는 데이터를 암호화하는 공용 키이고 하나는 암호를 해독하기 위한 연관된 개인용 키입니다.

반대로 대칭 키 암호 해독법은 하나의 키만 사용하는데, 이 키는 보안 통신에 관련되는 모든 당사자가 공유합니다. 이 비밀 키가 정보를 암호화 및 암호 해독하는 데 모두 사용됩니다. 키는 모든 당사자에게 안전하게 분배되고 당사자가 안전하게 저장해야 하는데, 이것은 보증하기 어렵습니다. 공용 키 암호 해독법을 사용할 때 공용 키는 비밀이 아니지만, 공용 키가 암호화하는 메시지는 연관된 개인용 키를 사용해서만 암호 해독될 수 있습니다. 개인용 키는 예를 들어 사용자의 키 데이터베이스에서 안전하게 저장되거나 컴퓨터의 하드 드라이브에서 암호화되어야 합니다.

공용 키 알고리즘 단독으로는 보안 통신을 보증하지 않으므로, 사용자와 통신하는 누구든 당사자의 ID를 검증해야 합니다. 이 인증을 수행하기 위해 SSL은 디지털 인증서를 사용합니다. 디지털 인증서를 누군가에게 전송할 때 인증서는 수신자에게 사용자의

공용 키를 제공합니다. 개인용 키를 사용하여 인증서를 디지털로 서명했으며 따라서 통신 수신자가 사용자의 공용 키를 사용하여 사용자의 시그니처를 검증할 수 있습니다. 디지털 인증서 자체의 유효성은 해당 인증서를 발행한 인증 기관(CA)에 의해 보증됩니다.

지원되는 암호 제품

SSL 응답 확인 방식 중에 클라이언트와 서버는 데이터를 교환하는 데 사용할 암호 제품을 협상합니다. 암호 제품은 인증, 암호화 및 데이터 무결성을 제공하는 데 사용되는 알고리즘 세트입니다.

DB2 데이터베이스 시스템은 FIPS 모드에서 실행 중인 GSKit을 사용하여 SSL 지원을 제공합니다. GSKit은 다음 암호 제품을 지원합니다.

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

각 암호 제품의 이름은 인증, 암호화 및 데이터 무결성 검사에 사용하는 알고리즘을 지정합니다. 예를 들어 암호 제품 TLS_RSA_WITH_AES_256_CBC_SHA는 인증에는 RSA, 암호화 알고리즘에는 AES 256비트 및 CBC, 데이터 무결성에 대한 해시 함수에는 SHA-1을 사용합니다.

SSL 응답 확인 방식 중에 DB2 데이터베이스 시스템은 자동으로 클라이언트와 서버 모두가 지원하는 가장 강력한 암호 제품을 선택합니다. 서버가 하나 이상의 특정 암호 제품만을 허용하기 원하는 경우 **ssl_cipherspecs** 구성 매개변수를 다음 값 중 하나로 설정할 수 있습니다.

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- 이러한 세 값의 모든 조합. 다중 값을 설정하려면 각 값을 쉼표로 구분하지만 값 사이에 공백을 넣지 마십시오.
- 널(NULL). 이 경우에는 가장 강력하고 사용 가능한 알고리즘이 자동으로 선택됩니다.

선택되는 암호 제품에 우선순위를 지정할 수 없습니다. **ssl_cipherspecs** 구성 매개변수를 설정하는 경우, DB2 데이터베이스 시스템은 가장 강력한 사용 가능한 암호 제품을 선택합니다. 이 선택은 **ssl_cipherspecs**를 설정할 때 암호 제품을 지정하는 순서에 의존하지 않습니다.

GSKit 리턴 코드

일부 DB2 데이터베이스 관리 프로그램 메시지는 IBM GSKit(Global Security Kit)의 리턴 코드를 표시할 수 있습니다.

일반 GSKit 리턴 코드

표 2. GSKit 일반 리턴 코드

리턴 코드(16진수)	리턴 코드(10진수)	상수	설명
0x00000000	0	GSK_OK	태스크가 성공적으로 완료되었습니다. 성공적으로 완료된 모든 함수 호출에 의해 발행됩니다.
0x00000001	1	GSK_INVALID_HANDLE	환경 또는 SSL 핸들이 유효하지 않습니다. 지정된 핸들이 성공적인 열기 함수 호출의 결과가 아닙니다.
0x00000002	2	GSK_API_NOT_AVAILABLE	동적 링크 라이브러리(DLL)가 언로드되었으며 사용할 수 없습니다.(Windows만 해당.)
0x00000003	3	GSK_INTERNAL_ERROR	내부 오류입니다. 서비스에 이 오류를 보고하십시오.
0x00000004	4	GSK_INSUFFICIENT_STORAGE	조작을 수행하기 위해 사용할 수 있는 메모리가 부족합니다.
0x00000005	5	GSK_INVALID_STATE	핸들이 핸들에 대한 init 조작을 두 번 수행하는 것과 같이 유효하지 않은 조작 상태에 있습니다.
0x00000006	6	GSK_KEY_LABEL_NOT_FOUND	지정된 키 레이블이 키 파일에 없습니다.
0x00000007	7	GSK_CERTIFICATE_NOT_AVAILABLE	인증서가 상대로부터 수신되지 않았습니다.
0x00000008	8	GSK_ERROR_CERT_VALIDATION	인증서 유효성 확인 오류입니다.
0x00000009	9	GSK_ERROR_CRYPTO	암호 해독법 처리 오류입니다.
0x0000000a	10	GSK_ERROR_ASN	증명서에서 ASN 필드 유효성 확인 오류입니다.
0x0000000b	11	GSK_ERROR_LDAP	LDAP 서버에 연결 오류입니다.
0x0000000c	12	GSK_ERROR_UNKNOWN_ERROR	내부 오류입니다. 서비스에 이 오류를 보고하십시오.
0x00000065	101	GSK_OPEN_CIPHER_ERROR	내부 오류입니다. 서비스에 이 오류를 보고하십시오.
0x00000066	102	GSK_KEYFILE_IO_ERROR	키 파일을 읽는 입출력 오류입니다.
0x00000067	103	GSK_KEYFILE_INVALID_FORMAT	키 파일의 내부 형식이 유효하지 않습니다. 키 파일을 다시 작성하십시오.
0x00000068	104	GSK_KEYFILE_DUPLICATE_KEY	키 파일에 동일한 키의 두 항목이 있습니다. iKeyman 유틸리티를 사용하여 중복 키를 제거하십시오.
0x00000069	105	GSK_KEYFILE_DUPLICATE_LABEL	키 파일에 동일한 레이블의 두 항목이 있습니다. iKeyman 유틸리티를 사용하여 중복 레이블을 제거하십시오.

표 2. GSKit 일반 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수	설명
0x0000006a	106	GSK_BAD_FORMAT_OR_ INVALID_PASSWORD	키 파일 암호가 무결성 검사에 대해 사용됩니다. 키 파일이 손상되었거나 암호 ID가 올바르지 않습니다.
0x0000006b	107	GSK_KEYFILE_CERT_EXPIRED	키 파일의 디폴트 키가 만기된 인증서를 보유합니다. iKeyman 유틸리티를 사용하여 만기된 인증서를 제거하십시오.
0x0000006c	108	GSK_ERROR_LOAD_GSKLIB	GSKit 동적 링크 라이브러리 중 하나를 로드하는 중에 오류가 발생했습니다. GSKit이 올바르게 설치되었는지 확인하십시오.
0x0000006d	109	GSK_PENDING_CLOSE_ERROR	GSK_ENVIRONMENT_CLOSE_ OPTIONS가 GSK_DELAYED_ ENVIRONMENT_CLOSE로 설정되었고 gsk_ environment_close() 함수가 호출된 후 GSKit 환경에서 연결을 작성하려고 시도 중임을 표시합니다.
0x000000c9	201	GSK_NO_KEYFILE_PASSWORD	암호 또는 stash-file 이름이 지정되지 않았으므로 키 파일을 초기화할 수 없습니다.
0x000000ca	202	GSK_KEYRING_OPEN_ERROR	키 파일이나 Microsoft Certificate Store를 열 수 없습니다. 경로가 잘못 지정되었거나, 파일 권한이 파일이 열리도록 허용하지 않거나, 파일 형식이 올바르지 않습니다.
0x000000cb	203	GSK_RSA_TEMP_KEY_PAIR	임시 키 쌍을 생성할 수 없습니다. 서비스에 이 오류를 보고하십시오.
0x000000cc	204	GSK_ERROR_LDAP_NO_SUCH_OBJECT	찾을 수 없는 사용자 이름 오브젝트가 지정되었습니다.
0x000000cd	205	GSK_ERROR_LDAP_INVALID_ CREDENTIALS	LDAP 쿼리에 사용되는 암호가 올바르지 않습니다.
0x000000ce	206	GSK_ERROR_BAD_INDEX	LDAP 서버의 장애 복구 목록에 대한 인덱스가 올바르지 않습니다.
0x000000cd	207	GSK_ERROR_FIPS_NOT_SUPPORTED	GSKit을 FIPS 모드로 하려는 시도가 실패했습니다.
0x0000012d	301	GSK_CLOSE_FAILED	GSKit 환경 닫기 요청이 제대로 처리되지 않았음을 표시합니다. 이것은 대부분 gsk_close_environment() 호출 후에 gsk_secure_socket*() 명령이 시도되기 때문입니다.
0x00000191	401	GSK_ERROR_BAD_DATE	시스템 날짜가 유효하지 않은 값으로 설정되었습니다.
0x00000192	402	GSK_ERROR_NO_CIPHERS	SSLV2 또는 SSLV3를 사용할 수 없습니다.

표 2. GSKit 일반 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수	설명
0x00000193	403	GSK_ERROR_NO_CERTIFICATE	필수 인증서가 상대로부터 수신되지 않았습니다.
0x00000194	404	GSK_ERROR_BAD_CERTIFICATE	수신된 인증서가 잘못 형식화되었습니다.
0x00000195	405	GSK_ERROR_UNSUPPORTED_ CERTIFICATE_TYPE	수신된 인증서 유형이 지원되지 않습니다.
0x00000196	406	GSK_ERROR_IO	데이터 읽기 또는 쓰기 조작에서 입출력 오류가 발생했습니다.
0x00000197	407	GSK_ERROR_BAD_KEYFILE_LABEL	키 파일의 지정된 레이블을 찾을 수 없습니다.
0x00000198	408	GSK_ERROR_BAD_KEYFILE_ PASSWORD	지정된 키 파일 암호가 올바르지 않습니다. 키 파일을 사용할 수 없습니다. 키 파일이 손상되었을 수도 있습니다.
0x00000199	409	GSK_ERROR_BAD_KEY_LEN_ FOR_EXPORT	제한된 암호 해독법 환경에서 키 크기가 지원하기에는 너무 큽니다.
0x0000019a	410	GSK_ERROR_BAD_MESSAGE	상대로부터 잘못 형식화된 SSL 메시지가 수신되었습니다.
0x0000019b	411	GSK_ERROR_BAD_MAC	메시지 인증 코드(MAC)가 성공적으로 검증되지 않았습니다.
0x0000019c	412	GSK_ERROR_UNSUPPORTED	지원되지 않는 SSL 프로토콜 또는 지원되지 않는 인증서 유형입니다.
0x0000019d	413	GSK_ERROR_BAD_CERT_SIG	수신된 인증서에 올바르지 않은 시그니처가 들어 있습니다.
0x0000019e	414	GSK_ERROR_BAD_CERT	상대로부터 잘못 형식화된 인증서가 수신되었습니다.
0x0000019f	415	GSK_ERROR_BAD_PEER	유효하지 않은 SSL 프로토콜이 상대로부터 수신되었습니다.
0x000001a0	416	GSK_ERROR_PERMISSION_DENIED	서비스에 이 내부 오류를 보고하십시오.
0x000001a1	417	GSK_ERROR_SELF_SIGNED	자체 서명 인증서가 유효하지 않습니다.
0x000001a2	418	GSK_ERROR_NO_READ_FUNCTION	읽기 조작에 실패했습니다. 서비스에 이 내부 오류를 보고하십시오.
0x000001a3	419	GSK_ERROR_NO_WRITE_FUNCTION	쓰기 조작에 실패했습니다. 서비스에 이 내부 오류를 보고하십시오.
0x000001a4	420	GSK_ERROR_SOCKET_CLOSED	프로토콜이 완료하기 전에 상대가 소켓을 닫았습니다.
0x000001a5	421	GSK_ERROR_BAD_V2_CIPHER	지정된 V2 암호가 유효하지 않습니다.
0x000001a6	422	GSK_ERROR_BAD_V3_CIPHER	지정된 V3 암호가 유효하지 않습니다.
0x000001a7	423	GSK_ERROR_BAD_SEC_TYPE	서비스에 이 내부 오류를 보고하십시오.
0x000001a8	424	GSK_ERROR_BAD_SEC_ TYPE_COMBINATION	서비스에 이 내부 오류를 보고하십시오.
0x000001a9	425	GSK_ERROR_HANDLE_ CREATION_FAILED	핸들을 작성할 수 없습니다. 서비스에 이 내부 오류를 보고하십시오.

표 2. GSKit 일반 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수	설명
0x000001aa	426	GSK_ERROR_INITIALIZATION_FAILED	초기화에 실패했습니다. 서비스에 이 내부 오류를 보고하십시오.
0x000001ab	427	GSK_ERROR_LDAP_NOT_AVAILABLE	인증서를 유효성 확인할 때 지정된 LDAP 디렉토리에 액세스할 수 없습니다.
0x000001ac	428	GSK_ERROR_NO_PRIVATE_KEY	지정된 키에 개인용 키가 없습니다.
0x000001ad	429	GSK_ERROR_PKCS11_LIBRARY_NOTLOADED	지정된 PKCS11 공유 라이브러리를 로드하려는 시도가 실패했습니다.
0x000001ae	430	GSK_ERROR_PKCS11_TOKEN_LABELMISMATCH	PKCS #11 드라이버가 호출자가 지정한 토큰을 찾지 못했습니다.
0x000001af	431	GSK_ERROR_PKCS11_TOKEN_NOTPRESENT	PKCS #11 토큰이 슬롯에 없습니다.
0x000001b0	432	GSK_ERROR_PKCS11_TOKEN_BADPASSWORD	PKCS #11 토큰에 액세스하기 위한 암호/핀이 유효하지 않습니다.
0x000001b1	433	GSK_ERROR_INVALID_V2_HEADER	수신된 SSL 헤더가 제대로 SSLV2 형식화된 헤더가 아닙니다.
0x000001b2	434	GSK_CSP_OPEN_ERROR	하드웨어 기반 암호화 서비스 제공업체(CSP)에 액세스할 수 없습니다. 주어진 CSP 이름이 시스템에 등록되지 않았거나 지정된 CSP 이름이 등록되었지만 인증서 저장소를 열지 못했습니다.
0x000001b3	435	GSK_CONFLICTING_ATTRIBUTE_SETTING	PKCS11, CMS 키 데이터베이스 및 Microsoft Crypto API 사이의 속성 설정 충돌이 있습니다.
0x000001b4	436	GSK_UNSUPPORTED_PLATFORM	요청된 함수가 응용프로그램이 실행 중인 플랫폼에서 지원되지 않습니다. 예를 들어 Microsoft Crypto API는 Windows 2000 이외의 플랫폼에서는 지원되지 않습니다.
0x000001b5	437	GSK_ERROR_INCORRECT_SESSION_TYPE	세션 유형 재설정 콜백 함수로부터 올바르지 않은 값이 리턴되었습니다. GSKit GSK_SERVER_SESSION 또는 GSK_SERVER_SESSION_WITH_CL_AUTH만 허용됩니다.
0x000001f5	501	GSK_INVALID_BUFFER_SIZE	버퍼 크기가 음수이거나 영(0)입니다.
0x000001f6	502	GSK_WOULD_BLOCK	비블록화 입출력과 함께 사용됩니다.
0x00000259	601	GSK_ERROR_NOT_SSLV3	SSLV3가 reset_cipher에 필요하며, 연결이 SSLV2를 사용합니다.
0x0000025a	602	GSK_MISC_INVALID_ID	gsk_secure_soc_misc 함수 호출에 대해 유효하지 않은 ID가 지정되었습니다.

표 2. GSKit 일반 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수	설명
0x000002bd	701	GSK_ATTRIBUTE_INVALID_ID	함수 호출에 유효하지 않은 ID가 있습니다. 이것은 또한 SSL 연결에 대한 핸들을 사용해야 할 때 환경 핸들을 지정하여 유발될 수 있습니다.
0x000002be	702	GSK_ATTRIBUTE_INVALID_LENGTH	속성이 음수 길이를 가지며, 이것은 유효하지 않습니다.
0x000002bf	703	GSK_ATTRIBUTE_INVALID_ ENUMERATION	열거 값이 지정된 열거 유형에 대해 유효하지 않습니다.
0x000002c0	704	GSK_ATTRIBUTE_INVALID_ SID_CACHE	세션 ID(SID) 캐시 루틴 교체를 위한 유효하지 않은 매개변수 목록입니다.
0x000002c1	705	GSK_ATTRIBUTE_INVALID_ NUMERIC_VALUE	숫자 속성을 설정할 때 지정된 값이 설정될 특정한 속성에 대해 유효하지 않습니다.
0x000002c2	706	GSK_CONFLICTING_VALIDATION_ SETTING	추가 인증서 유효성 확인에 대해 상충하는 매개변수가 설정되었습니다.
0x000002c3	707	GSK_AES_UNSUPPORTED	지정한 암호에 실행 시스템에서 지원되지 않는 AES 암호가 포함되었습니다.
0x000002c4	708	GSK_PEERID_LENGTH_ERROR	피어 ID의 길이가 올바르지 않습니다. 16바이트보다 작거나 같아야 합니다.
0x000002c5	709	GSK_CIPHER_INVALID_WHEN_ FIPS_MODE_OFF	제공된 암호는 FIPS 모드가 해제될 때 허용되지 않습니다.
0x000002c6	710	GSK_CIPHER_INVALID_WHEN_ FIPS_MODE_ON	FIPS 모드에서 FIPS 승인 암호가 선택되지 않았습니다.
0x00000641	1601	GSK_TRACE_STARTED	추적이 시작되었습니다.
0x00000642	1602	GSK_TRACE_STOPPED	추적이 중지했습니다.
0x00000643	1603	GSK_TRACE_NOT_STARTED	이전에 추적 파일이 시작되지 않았으므로 중지할 수 없습니다.
0x00000644	1604	GSK_TRACE_ALREADY_STARTED	추적 파일이 이미 시작했으므로 다시 시작할 수 없습니다.
0x00000645	1605	GSK_TRACE_OPEN_FAILED	추적 파일을 열 수 없습니다. gsk_start_trace()의 첫 번째 매개변수는 유효한 전체 경로 파일 이름이어야 합니다.

키 관리 리턴 코드

표 3. 키 관리 리턴 코드

리턴 코드(16진수)	리턴 코드(10진수)	상수
0x00000000	0	GSKKM_ERR_OK
0x00000000	0	GSKKM_ERR_SUCCESS
0x00000001	1	GSKKM_ERR_UNKNOWN
0x00000002	2	GSKKM_ERR_ASN
0x00000003	3	GSKKM_ERR_ASN_INITIALIZATION

표 3. 키 관리 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수
0x00000004	4	GSKKM_ERR_ASN_PARAMETER
0x00000005	5	GSKKM_ERR_DATABASE
0x00000006	6	GSKKM_ERR_DATABASE_OPEN
0x00000007	7	GSKKM_ERR_DATABASE_RE_OPEN
0x00000008	8	GSKKM_ERR_DATABASE_CREATE
0x00000009	9	GSKKM_ERR_DATABASE_ALREADY_EXISTS
0x0000000a	10	GSKKM_ERR_DATABASE_DELETE
0x0000000b	11	GSKKM_ERR_DATABASE_NOT_OPENED
0x0000000c	12	GSKKM_ERR_DATABASE_READ
0x0000000d	13	GSKKM_ERR_DATABASE_WRITE
0x0000000e	14	GSKKM_ERR_DATABASE_VALIDATION
0x0000000f	15	GSKKM_ERR_DATABASE_INVALID_VERSION
0x00000010	16	GSKKM_ERR_DATABASE_INVALID_PASSWORD
0x00000011	17	GSKKM_ERR_DATABASE_INVALID_FILE_TYPE
0x00000012	18	GSKKM_ERR_DATABASE_CORRUPTION
0x00000013	19	GSKKM_ERR_DATABASE_PASSWORD_ CORRUPTION
0x00000014	20	GSKKM_ERR_DATABASE_KEY_INTEGRITY
0x00000015	21	GSKKM_ERR_DATABASE_DUPLICATE_KEY
0x00000016	22	GSKKM_ERR_DATABASE_DUPLICATE_ KEY_RECORD_ID
0x00000017	23	GSKKM_ERR_DATABASE_DUPLICATE_ KEY_LABEL
0x00000018	24	GSKKM_ERR_DATABASE_DUPLICATE_ KEY_SIGNATURE
0x00000019	25	GSKKM_ERR_DATABASE_DUPLICATE_ KEY_UNSIGNED_CERTIFICATE
0x0000001a	26	GSKKM_ERR_DATABASE_DUPLICATE_KEY_ ISSUER_AND_SERIAL_NUMBER
0x0000001b	27	GSKKM_ERR_DATABASE_DUPLICATE_KEY_ SUBJECT_PUBLIC_KEY_INFO
0x0000001c	28	GSKKM_ERR_DATABASE_DUPLICATE_KEY_ UNSIGNED_CRL
0x0000001d	29	GSKKM_ERR_DATABASE_DUPLICATE_LABEL
0x0000001e	30	GSKKM_ERR_DATABASE_PASSWORD_ ENCRYPTION
0x0000001f	31	GSKKM_ERR_DATABASE_LDAP
0x00000020	32	GSKKM_ERR_CRYPTO
0x00000021	33	GSKKM_ERR_CRYPTO_ENGINE
0x00000022	34	GSKKM_ERR_CRYPTO_ALGORITHM
0x00000023	35	GSKKM_ERR_CRYPTO_SIGN

표 3. 키 관리 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수
0x00000024	36	GSKKM_ERR_CRYPTO_VERIFY
0x00000025	37	GSKKM_ERR_CRYPTO_DIGEST
0x00000026	38	GSKKM_ERR_CRYPTO_PARAMETER
0x00000027	39	GSKKM_ERR_CRYPTO_UNSUPPORTED_ ALGORITHM
0x00000028	40	GSKKM_ERR_CRYPTO_INPUT_GREATER_ THAN_MODULUS
0x00000029	41	GSKKM_ERR_CRYPTO_UNSUPPORTED_ MODULUS_SIZE
0x0000002a	42	GSKKM_ERR_VALIDATION
0x0000002b	43	GSKKM_ERR_VALIDATION_KEY
0x0000002c	44	GSKKM_ERR_VALIDATION_DUPLICATE_ EXTENSIONS
0x0000002d	45	GSKKM_ERR_VALIDATION_KEY_WRONG_ VERSION
0x0000002e	46	GSKKM_ERR_VALIDATION_KEY_ EXTENSIONS_REQUIRED
0x0000002f	47	GSKKM_ERR_VALIDATION_KEY_VALIDITY
0x00000030	48	GSKKM_ERR_VALIDATION_KEY_VALIDITY_ PERIOD
0x00000031	49	GSKKM_ERR_VALIDATION_KEY_VALIDITY_ PRIVATE_KEY_USAGE
0x00000032	50	GSKKM_ERR_VALIDATION_KEY_ISSUER_ NOT_FOUND
0x00000033	51	GSKKM_ERR_VALIDATION_KEY_MISSING_ REQUIRED_EXTENSIONS
0x00000034	52	GSKKM_ERR_VALIDATION_KEY_BASIC_ CONSTRAINTS
0x00000035	53	GSKKM_ERR_VALIDATION_KEY_SIGNATURE
0x00000036	54	GSKKM_ERR_VALIDATION_KEY_ROOT_KEY_ NOT_TRUSTED
0x00000037	55	GSKKM_ERR_VALIDATION_KEY_IS_REVOKED
0x00000038	56	GSKKM_ERR_VALIDATION_KEY_AUTHORITY_ KEY_IDENTIFIER
0x00000039	57	GSKKM_ERR_VALIDATION_KEY_PRIVATE_KEY_ USAGE_PERIOD
0x0000003a	58	GSKKM_ERR_VALIDATION_SUBJECT_ ALTERNATIVE_NAME
0x0000003b	59	GSKKM_ERR_VALIDATION_ISSUER_ ALTERNATIVE_NAME
0x0000003c	60	GSKKM_ERR_VALIDATION_KEY_USAGE
0x0000003d	61	GSKKM_ERR_VALIDATION_KEY_ UNKNOWN_CRITICAL_EXTENSION

표 3. 키 관리 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수
0x0000003e	62	GSKKM_ERR_VALIDATION_KEY_PAIR
0x0000003f	63	GSKKM_ERR_VALIDATION_CRL
0x00000040	64	GSKKM_ERR_MUTEX
0x00000041	65	GSKKM_ERR_PARAMETER
0x00000042	66	GSKKM_ERR_NULL_PARAMETER
0x00000043	67	GSKKM_ERR_NUMBER_SIZE
0x00000044	68	GSKKM_ERR_OLD_PASSWORD
0x00000045	69	GSKKM_ERR_NEW_PASSWORD
0x00000046	70	GSKKM_ERR_PASSWORD_EXPIRATION_TIME
0x00000047	71	GSKKM_ERR_THREAD
0x00000048	72	GSKKM_ERR_THREAD_CREATE
0x00000049	73	GSKKM_ERR_THREAD_WAIT_FOR_EXIT
0x0000004a	74	GSKKM_ERR_IO
0x0000004b	75	GSKKM_ERR_LOAD
0x0000004c	76	GSKKM_ERR_PKCS11
0x0000004d	77	GSKKM_ERR_NOT_INITIALIZED
0x0000004e	78	GSKKM_ERR_DB_TABLE_CORRUPTED
0x0000004f	79	GSKKM_ERR_MEMORY_ALLOCATE
0x00000050	80	GSKKM_ERR_UNSUPPORTED_OPTION
0x00000051	81	GSKKM_ERR_GET_TIME
0x00000052	82	GSKKM_ERR_CREATE_MUTEX
0x00000053	83	GSKKM_ERR_CMDCAT_OPEN
0x00000054	84	GSKKM_ERR_ERRCAT_OPEN
0x00000055	85	GSKKM_ERR_FILENAME_NULL
0x00000056	86	GSKKM_ERR_FILE_OPEN
0x00000057	87	GSKKM_ERR_FILE_OPEN_TO_READ
0x00000058	88	GSKKM_ERR_FILE_OPEN_TO_WRITE
0x00000059	89	GSKKM_ERR_FILE_OPEN_NOT_EXIST
0x0000005a	90	GSKKM_ERR_FILE_OPEN_NOT_ALLOWED
0x0000005b	91	GSKKM_ERR_FILE_WRITE
0x0000005c	92	GSKKM_ERR_FILE_REMOVE
0x0000005d	93	GSKKM_ERR_BASE64_INVALID_DATA
0x0000005e	94	GSKKM_ERR_BASE64_INVALID_MSGTYPE
0x0000005f	95	GSKKM_ERR_BASE64_ENCODING
0x00000060	96	GSKKM_ERR_BASE64_DECODING
0x00000061	97	GSKKM_ERR_DN_TAG_NULL
0x00000062	98	GSKKM_ERR_DN_CN_NULL
0x00000063	99	GSKKM_ERR_DN_C_NULL
0x00000064	100	GSKKM_ERR_INVALID_DB_HANDLE
0x00000065	101	GSKKM_ERR_KEYDB_NOT_EXIST

표 3. 키 관리 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수
0x00000066	102	GSKKM_ERR_KEYPAIRDB_NOT_EXIST
0x00000067	103	GSKKM_ERR_PWDFILE_NOT_EXIST
0x00000068	104	GSKKM_ERR_PASSWORD_CHANGE_MATCH
0x00000069	105	GSKKM_ERR_KEYDB_NULL
0x0000006a	106	GSKKM_ERR_REQKEYDB_NULL
0x0000006b	107	GSKKM_ERR_KEYDB_TRUSTCA_NULL
0x0000006c	108	GSKKM_ERR_REQKEY_FOR_CERT_NULL
0x0000006d	109	GSKKM_ERR_KEYDB_PRIVATE_KEY_NULL
0x0000006e	110	GSKKM_ERR_KEYDB_DEFAULT_KEY_NULL
0x0000006f	111	GSKKM_ERR_KEYREC_PRIVATE_KEY_NULL
0x00000070	112	GSKKM_ERR_KEYREC_CERTIFICATE_NULL
0x00000071	113	GSKKM_ERR_CRLS_NULL
0x00000072	114	GSKKM_ERR_INVALID_KEYDB_NAME
0x00000073	115	GSKKM_ERR_UNDEFINED_KEY_TYPE
0x00000074	116	GSKKM_ERR_INVALID_DN_INPUT
0x00000075	117	GSKKM_ERR_KEY_GET_BY_LABEL
0x00000076	118	GSKKM_ERR_LABEL_LIST_CORRUPT
0x00000077	119	GSKKM_ERR_INVALID_PKCS12_DATA
0x00000078	120	GSKKM_ERR_PKCS12_PWD_CORRUPTION
0x00000079	121	GSKKM_ERR_EXPORT_TYPE
0x0000007a	122	GSKKM_ERR_PBE_ALG_UNSUPPORT
0x0000007b	123	GSKKM_ERR_KYR2KDB
0x0000007c	124	GSKKM_ERR_KDB2KYR
0x0000007d	125	GSKKM_ERR_ISSUING_CERTIFICATE
0x0000007e	126	GSKKM_ERR_FIND_ISSUER_CHAIN
0x0000007f	127	GSKKM_ERR_WEBDB_DATA_BAD_FORMAT
0x00000080	128	GSKKM_ERR_WEBDB_NOTHING_TO_WRITE
0x00000081	129	GSKKM_ERR_EXPIRE_DAYS_TOO_LARGE
0x00000082	130	GSKKM_ERR_PWD_TOO_SHORT
0x00000083	131	GSKKM_ERR_PWD_NO_NUMBER
0x00000084	132	GSKKM_ERR_PWD_NO_CONTROL_KEY
0x00000085	133	GSKKM_ERR_SIGNATURE_ALGORITHM
0x00000086	134	GSKKM_ERR_INVALID_DATABASE_TYPE
0x00000087	135	GSKKM_ERR_SECONDARY_KEYDB_TO_OTHER
0x00000088	136	GSKKM_ERR_NO_SECONDARY_KEYDB
0x00000089	137	GSKKM_ERR_CRYPTOGRAPHIC_TOKEN_ LABEL_NOT_EXIST
0x0000008a	138	GSKKM_ERR_CRYPTOGRAPHIC_TOKEN_ PASSWORD_REQUIRED

표 3. 키 관리 리턴 코드 (계속)

리턴 코드(16진수)	리턴 코드(10진수)	상수
0x0000008b	139	GSKKM_ERR_CRYPTOGRAPHIC_TOKEN_ PASSWORD_NOT_REQUIRED
0x0000008c	140	GSKKM_ERR_CRYPTOGRAPHIC_TOKEN_ LIBRARY_NOT_LOADED
0x0000008d	141	GSKKM_ERR_CRYPTOGRAPHIC_TOKEN_ NOT_SUPPORT
0x0000008e	142	GSKKM_ERR_CRYPTOGRAPHIC_TOKEN_ FUNCTION_FAILED
0x0000008f	143	GSKKM_ERR_LDAP_USER_NOT_FOUND
0x00000090	144	GSKKM_ERR_LDAP_INVALID_PASSWORD
0x00000091	145	GSKKM_ERR_LDAP_QUERY_ENTRY_FAILED
0x00000092	146	GSKKM_ERR_INVALID_CERT_CHAIN
0x00000093	147	GSKKM_ERR_CERT_ROOT_NOT_TRUSTED
0x00000094	148	GSKKM_ERR_CERT_REVOKED
0x00000095	149	GSKKM_ERR_CRYPTOGRAPHIC_OBJECT_ FUNCTION_FAILED
0x00000096	150	GSKKM_ERR_NO_AVAILABLE_CRL_ DATASOURCE
0x00000097	151	GSKKM_ERR_NO_TOKEN_PRESENT
0x00000098	152	GSKKM_ERR_FIPS_NOT_SUPPORTED
0x00000099	153	GSKKM_ERR_FIPS_CONFLICT_SETTING
0x0000009a	154	GSKKM_ERR_PASSWORD_STRENGTH_FAILED

저장된 데이터 암호화를 위한 IBM Database Encryption Expert

IBM Database Encryption Expert는 기본 DB2 보안과 함께 사용할 때 광범위한 위협에 대해 데이터 및 데이터베이스 응용프로그램의 효과적인 보호를 제공하는 포괄적인 소프트웨어 데이터 보안 솔루션입니다.

Database Encryption Expert는 조직이 비공개 및 기밀 데이터가 강력하게 보호되고 규정 및 법규를 준수하는지 확인하는 데 도움이 됩니다. Database Encryption Expert의 핵심 이점은 다음과 같습니다.

- DB2 데이터베이스 시스템을 위한 입증되고 강력한 데이터 보안
- 사용 중인 파일, 구성 파일, 로그 파일 및 백업 데이터의 보호
- 응용프로그램, 데이터베이스 및 스토리지 환경에 대한 투명성
- 온라인 및 오프라인 환경 모두에서의 데이터 보호를 위한 통합된 규정 및 키 관리
- 성능 요구사항 충족

Database Encryption Expert를 사용하여 오프라인 데이터베이스 백업을 암호화하고 온라인(″라이브″) 데이터베이스 파일을 암호화할 수 있습니다. 이것은 네트워크를 통해 이동 중인 ″인플라이트(inflight) 데이터″와는 반대로 때로는 ″저장된 데이터″라는 디스크에 있는 데이터의 보호입니다.

- 백업의 경우 데이터는 백업되는 중에 암호화되므로 백업 디바이스의 데이터가 암호화됩니다. 데이터를 복구해야 하는 경우, 복구 서버가 데이터가 암호화됨을 인식하고 데이터를 암호화 해제합니다.
- 데이터베이스 파일의 경우 DB2 데이터베이스의 데이터를 포함하는 운영 체제 데이터 파일이 암호화됩니다. 이것은 “원시” 데이터베이스 파일을 읽으려는 권한이 없는 사용자로부터 데이터 파일을 보호합니다.

Database Encryption Expert는 사용자, 데이터베이스, 응용프로그램 및 스토리지에 투명합니다. 코드 변경 또는 기존 기반 구조에 대한 변경은 필요하지 않습니다. Database Encryption Expert는 모든 스토리지 환경에서 데이터를 보호할 수 있으며, 그 동안 사용자는 이전과 동일한 방법으로 계속 데이터에 액세스할 수 있습니다.

Database Encryption Expert는 실행할 수 있는 파일, 구성 파일, 라이브러리 등에 대한 변경을 예방하여 응용프로그램에 대한 공격을 막으므로, 데이터베이스 응용프로그램을 보호할 수 있습니다.

Database Encryption Expert의 아키텍처

Database Encryption Expert는 웹 기반 사용자 인터페이스 및 명령행 유틸리티를 사용하여 사용자가 관리하는 에이전트 및 서버 소프트웨어 패키지 세트입니다. Database Encryption Expert 관리자는 보안 및 암호화가 구현되는 방법을 관리하는 보안 규정을 구성합니다.

이러한 보안 규정이 정의되는 방법에 따라서, Database Encryption Expert 백업 에이전트가 DB2 백업을 암호화하고, Database Encryption Expert 파일 시스템 에이전트가 DB2 데이터 파일을 암호화합니다.

Encryption Expert Security Server가 보안 규정, 암호화 키 및 이벤트 로그 파일을 저장합니다. 보안 규정에는 액세스를 허용하거나 거부하기 위해 충족되어야 하는 보안 규칙 세트가 들어 있습니다. 각 보안 규칙은 보호되는 데이터에 액세스하는 사용자, 종류, 시기 및 방법을 평가하며, 이러한 기준이 일치하면 Security Server가 액세스를 허용하거나 거부합니다.

94 페이지의 그림 4은 Database Encryption Expert의 아키텍처를 보여줍니다.

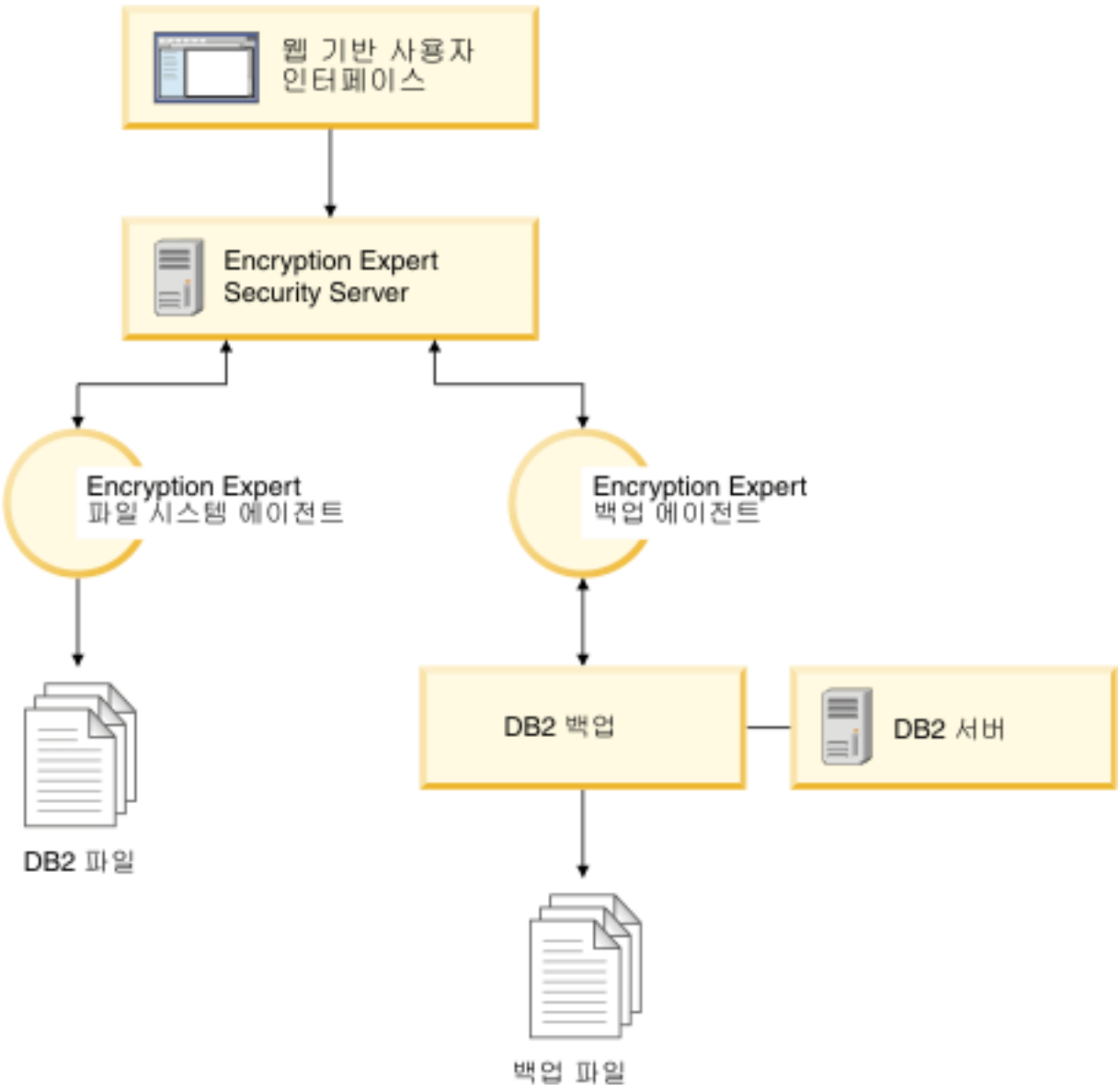


그림 4. Database Encryption Expert의 아키텍처

파일 시스템 에이전트

Database Encryption Expert 파일 시스템 에이전트 프로세스는 항상 백그라운드에서 실행 중입니다. 에이전트는 보호하고 있는 데이터 파일, 디렉토리 또는 실행 파일에 액세스하려는 모든 시도를 인터셉트합니다. Database Encryption Expert 파일 시스템 에이전트는 액세스 시도를 보안 서버로 포워드하며, 적용되는 규정을 기초로 보안 서버가 시도된 액세스를 허용하거나 거부합니다.

Database Encryption Expert 보호는 단순히 파일에 대한 액세스를 허용 또는 거부하는 것 이상으로 확장하므로 사용자가 파일을 암호화할 수도 있습니다. 파일 컨텐츠만 암호화되고 파일 메타데이터는 그대로 있습니다. 그러므로 단지 이름, 시간소인, 파일 유형 등을 보기 위해 암호화된 파일을 암호 해독할 필요는 없습니다. 이것은 데이터 관리

응용프로그램이 파일 컨텐츠를 노출하지 않고 기능을 수행할 수 있도록 합니다. 예를 들어 백업 관리 프로그램은 컨텐츠를 볼 수 있지 않아도 특정 데이터를 백업할 수 있습니다.

암호화된 파일을 권한이 없는 사용자가 액세스하는 경우, 적합한 보안 서버 승인 및 암호화 키가 없으면 파일의 컨텐츠는 쓸모없습니다. 그러나 올바른 규정 및 권한이 있는 사용자는 암호화 및 암호 해독이 발생 중인지 모릅니다.

백업 에이전트

일반적으로 DB2 백업 API 시스템에 의해 수행되는 모든 데이터베이스 백업 기능은 원시(native) 데이터베이스 압축을 포함하여 Database Encryption Expert 서버에 의해 지원됩니다. 추가 명령행 인수 외에, DB2 백업 운영자는 Database Encryption Expert 개입을 모릅니다. Database Encryption Expert가 정적인 저장된 데이터 및 사용 중인 온라인 데이터를 백업하고 리스토어합니다.

기본 백업 및 리스토어 구성이 지원됩니다. 기본 구성에서 데이터는 하나의 서버와 다중 에이전트를 사용하여 암호화 및 백업됩니다. 데이터는 원래 백업을 작성하는 데 사용되었던 동일한 서버와 함께 구성되는 에이전트에서 암호 해독 및 리스토어됩니다.

단일 사이트 및 다중 사이트 구성도 백업 및 리스토어를 위해 지원됩니다. 단일 사이트 시나리오에서 구성 데이터는 단일 데이터 센터에서 다중 보안 서버 사이에 미러링됩니다. 다중 사이트 시나리오에서는 백업이 다른 데이터 센터의 다른 Encryption Expert 서버에서 리스토어됩니다.

감사 로그

Database Encryption Expert 에이전트 활동은 중앙 집중된 감사 로그 기능을 통해 밀접하게 모니터 및 로그됩니다. 백업, 리스토어 및 보안 관리 조작을 포함한 모든 감사 가능한 이벤트를 로그할 수 있습니다. 여기에는 초기화, 종료 및 재시작 같은 Database Encryption Expert 시스템 이벤트가 포함되며, 네트워크는 서로 다른 Database Encryption Expert 구성요소 사이에서 연결하고 연결을 끊습니다.

Database Encryption Expert 문서

Database Encryption Expert에 대한 자세한 정보는 http://publib.boulder.ibm.com/infocenter/mptoolic/v1r0/topic/com.ibm.db2tools.eet.doc.ug/eetwelcome.htm 웹 페이지를 참조하십시오.

AIX 암호화된 파일 시스템(EFS)을 사용하여 데이터베이스 암호화

AIX 운영 체제에서 실행되는 DB2 Enterprise Server Edition은 AIX 암호화된 파일 시스템(EFS)을 사용하여 암호화된 데이터베이스를 설정할 수 있는 옵션을 제공합니다. EFS에 대한 자세한 정보는 AIX 문서를 참조하십시오.

주: 데이터베이스 파티션(DPF) 환경에서 작업하는 경우, EFS를 사용하려면 데이터베이스가 단일 데이터베이스 파티션에 있어야 합니다.

JFS2 파일 시스템에 기본 EFS를 사용하여 데이터베이스 테이블의 데이터가 포함된 운영 체제 파일을 암호화할 수 있습니다.

암호화를 설정하려면 다음 단계를 따르십시오.

1. 시스템에서 EFS를 사용 가능으로 설정하십시오.
2. DB2 데이터베이스 디먼이 실행되는 사용자 어카운트의 키스토어를 로드하십시오.
3. 데이터베이스 파일 시스템에서 EFS를 사용 가능으로 설정하십시오.
4. 암호화할 운영 체제 파일을 판별하십시오.
5. EFS 보호가 필요한 데이터베이스 테이블이 포함된 파일을 암호화하십시오.

시스템에서 EFS를 사용 가능으로 설정

EFS를 사용 가능으로 설정하려면 `clic.rte` 파일 세트가 설치되어 있어야 합니다. 확장 팩 CD에서 `clic.rte` 설치 이미지를 볼 수 있습니다.

다음 명령을 루트로 실행하여 시스템에서 EFS를 사용 가능으로 설정하십시오.

```
% efsenable –a
```

efsenable 명령은 한 번만 실행해야 합니다.

키스토어 로드

다음 구성 예제에서는 데이터베이스 디먼이 실행되는 DB2 사용자 어카운트를 abst라고 합니다. 사용자 abst에는 키스토어가 있어야 하며 abst가 구성원으로 속해 있는 그룹에도 키스토어가 있어야 합니다.

1. DB2 디먼을 시작하기 전에 모든 키스토어가 abst 프로세스에 연결되어 있어야 합니다.

   다음 예제에서와 같이, efskeymgr –V 명령을 사용하여 모든 키스토어가 abst 프로세스에 연결되어 있는지 확인할 수 있습니다.

   ```
   # lsuser abst
   abst id=203 pgrp=abstgp groups=abstgp,staff ...

   # efskeymgr -V
   List of keys loaded in the current process:
     Key #0:
                       Kind ..................... User key
                       Id (uid / gid) ......... 203
                       Type ..................... Private
   key
                       Algorithm ................ RSA_1024
                       Validity ................. Key is
   valid
   ```

```
                    Fingerprint ..............
24c88df2:d91cb6a2:c3e11b6a:4c13f8b4:666fabd8

  Key #1:
                    Kind ..................... Group
key
                    Id (uid / gid) ......... 1
                    Type ..................... Private
key
                    Algorithm ................ RSA_1024
                    Validity ................. Key is
valid
                    Fingerprint ..............
03fead42:57e7646e:a1715626:cfa56c8e:8abed1c1

  Key #2:
                    Kind ..................... Group
key
                    Id (uid / gid) ......... 212
                    Type ..................... Private
key
                    Algorithm ................ RSA_1024
                    Validity ................. Key is
valid
                    Fingerprint ..............
339dfb19:bc850f4c:5551c975:7fe4961b:2dddf3bc
```

2. abst 프로세스와 연결된 키스토어가 없으면 다음 명령을 사용하여 키스토어를 로드하십시오. `% efskeymgr –o ksh`

   이 명령을 사용하려면 초기에 로그인 암호로 설정된 키스토어 암호를 입력해야 합니다.

3. 다음 명령을 사용하여 사용자 키와 그룹 키가 로드되었는지 확인하십시오. `% efskeymgr –V`

   사용자 키와 그룹 키가 나열되어야 합니다. 그래도 그룹 키스토어가 나열되지 않으면 단계 4를 계속 진행하십시오.

4. 그룹이 작성된 방법에 따라 그룹 키스토어가 없을 수 있습니다. efskeymgr –V 명령을 사용해도 사용자의 그룹 키스토어가 나열되지 않으면 그룹 키스토어를 작성해야 합니다.

   루트 또는 RBAC 역할 aix.efs_admin으로 그룹 키스토어를 작성합니다.

   `% efskeymgr -C group_name`

5. 적합한 각 사용자에게 그룹 키스토어 액세스 권한을 지정하십시오.

   `% efskeymgr -k group /group_name -s user/user_name`

   사용자가 이미 로그인되어 있으면, 그룹 키스토어에 대한 액세스 권한이 즉시 부여되지 않으므로 efskeymgr –o ksh 명령을 사용하거나 다시 로그인하여 키스토어를 다시 로드해야 합니다.

데이터베이스 파일 시스템에서 EFS를 사용 가능으로 설정

EFS는 JFS2 파일 시스템에서만 실행되며 명확하게 사용 가능으로 설정해야 합니다.

데이터베이스가 기존 파일 시스템에 상주하는 경우, `% chfs –a efs=yes` *`filesystem`* 명령을 실행하여 EFS를 사용 가능으로 설정하십시오. 예를 들면 다음과 같습니다.

```
% chfs –a efs=yes /foo
```

새 파일 시스템을 작성하는 경우, smit 명령 또는 crfs 명령과 함께 `-a efs=yes`옵션을 사용하여 EFS를 사용 가능으로 설정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
% crfs -v jfs2 -a efs=yes -m mount_point -d devide -A yes
```

이제 파일 시스템에서 EFS를 사용할 수 있지만 켜져 있지는 않습니다. 암호화된 데이터가 필요한 특정 데이터베이스 테이블에만 EFS를 켜십시오(efsmgr 명령 및 상속에 대한 자세한 정보는 AIX EFS 문서 참조).

암호화할 파일 판별

EFS 암호화를 통해 보호하려는 특정 데이터베이스 테이블이 포함된 파일을 판별하려면 예제에서와 같이 EMPLOYEE 테이블을 사용하여 다음 단계를 수행하십시오.

1. 다음 예제에서와 같이 쿼리를 사용하여 테이블의 TBSPACEID를 찾으십시오.

   ```
   SELECT TABNAME, TBSPACEID FROM syscat.tables WHERE tabname='EMPLOYEE'
   ```

   이 쿼리의 결과는 다음과 같습니다.

TABNAME	TBSPACEID
EMPLOYEE	2

2. 다음 예제에서와 같이 쿼리를 사용하여 TBSPACEID의 테이블 스페이스를 찾으십시오.

   ```
   LIST TABLESPACE CONTAINERS FOR 2
   ```

   이 쿼리의 결과는 다음과 같습니다.

컨테이너 ID	이름	유형
0	/foo/abst/NODE0000/BAR/T0000002/C0000000.LRG	파일

   이제 이 테이블 스페이스가 `/foo/abst/NODE0000/BAR/T0000002/C0000000.LRG`라고 하는 운영 체제 파일에 포함되어 있음을 알 수 있습니다. 이 파일이 암호화해야 할 파일입니다.

파일 암호화

데이터 또는 데이터베이스를 변경하기 전에 먼저 데이터베이스를 백업하십시오.

다음 단계에 따라 파일을 암호화하십시오.

1. 파일을 나열하십시오. 예:

   ```
   # ls -U /foo/abst/NODE0000/BAR/T0000002/C0000000.LRG

   -rw-------- 1 abst abstgp 33554432 Jul 30 18:01
   /foo/abst/NODE0000/BAR/T0000002/C0000000.LRG
   ```

2. efsmgr 명령을 사용하여 파일을 암호화하십시오. 예:

   ```
   # efsmgr -e /foo/abst/NODE0000/BAR/T0000002/C0000000.LRG
   ```

   파일을 다시 나열하면 파일이 암호화되어 있음을 나타내는 허용 문자열 끝에『e』가 표시됩니다. 예를 들어, 다음과 같습니다.

   ```
   # ls -U /foo/abst/NODE0000/BAR/T0000002/C0000000.LRG

   -rw-------e 1 abst abstgp 33554432 Jul 30 18:03
   /foo/abst/NODE0000/BAR/T0000002/C0000000.LRG
   ```

3. DB2 데이터베이스 관리 프로그램을 시작하여 평소처럼 사용하십시오. EMPLOYEE 테이블 및 암호화된 이 테이블 스페이스에 추가된 모든 데이터는 기본 파일 시스템에서 EFS에 의해 암호화됩니다. 데이터를 검색할 때마다, DB2 데이터베이스 관리 프로그램을 통해 암호가 해독되고 정상적으로 표시됩니다.

DB2 활동 감사

DB2 감사 기능 개요

다양한 인증 및 액세스 제어 메커니즘을 사용하여 허용되는 데이터 액세스에 대한 규칙과 제어사항을 설정함으로써 sensitive 데이터에 대한 액세스를 관리할 수 있습니다. 그러나 DB2 감사 기능을 사용하여 데이터 액세스를 모니터하면 알 수 없는 동작이나 허용할 수 없는 동작으로부터 보호할 수 있습니다.

원하지 않는 데이터 액세스 및 후속 분석을 성공적으로 모니터링하면 데이터 액세스 제어가 향상되며 궁극적으로 데이터에 대해 악의적이거나 부주의한 권한이 없는 액세스를 방지할 수 있습니다. 시스템 관리 조치를 포함하여 응용프로그램 및 각 사용자 액세스의 모니터링은 사용자 데이터베이스 시스템에서의 활동의 실행기록 레코드를 제공할 수 있습니다.

DB2 감사 기능은 일련의 사전 정의된 데이터베이스 이벤트의 감사 추적을 생성하고 유지보수할 수 있게 해줍니다. 이 기능에서 생성된 레코드는 감사 로그 파일에 보존됩니

다. 이러한 레코드 분석을 통해 시스템 오용을 식별하는 사용 패턴을 알 수 있습니다. 일단 시스템 오용이 식별되면, 이러한 시스템 오용을 감소시키거나 줄이기 위한 조치가 취해질 수 있습니다.

감사 기능은 모든 인스턴스 및 데이터베이스 레벨 활동을 각각 별도의 로그에 기록할지에 관계없이, 인스턴스 및 개별 데이터베이스 레벨 모두에서 감사하는 기능을 제공합니다. 시스템 관리자(SYSADM 권한 보유)는 db2audit 도구를 사용하여 인스턴스 레벨에서 감사를 구성하고 이러한 감사 정보를 수집하는 시기를 제어할 수 있습니다. 시스템 관리자가 db2audit 도구를 사용하여 인스턴스 및 데이터베이스 감사 로그를 둘 다 아카이브하고 어느 한 유형의 아카이브된 로그에서 감사 데이터를 추출할 수 있습니다.

보안 관리자(데이터베이스 내에서 SECADM 권한 보유)는 감사 규정을 AUDIT SQL 문과 함께 사용하여 개별 데이터베이스에 대한 감사 요구사항을 구성 및 제어할 수 있습니다. 또한 다음 감사 루틴을 사용하여 지정된 태스크를 수행할 수 있습니다.

- SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저 아카이브 감사 로그
- The SYSPROC.AUDIT_LIST_LOGS 테이블 함수를 사용하면 원하는 로그를 찾을 수 있습니다.
- SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저는 분석할 데이터를 구분된 파일로 추출합니다.

보안 관리자는 이러한 루틴에 대한 EXECUTE 특권을 다른 사용자에게 부여하여 원하는 경우 이러한 태스크를 위임할 수 있습니다.

파티션된 데이터베이스 환경에서 많은 감사 이벤트는 사용자가 연결된 데이터베이스 파티션(코디네이터 파티션) 또는 카탈로그 파티션(동일한 데이터베이스 파티션이 아닌 경우)에서 발생합니다. 즉 이는 둘 이상의 데이터베이스 파티션이 감사 레코드를 생성할 수 있다는 것을 의미합니다. 각 감사 레코드 파트에는 코디네이터 파티션과 원래 파티션(감사 레코드가 비롯된 파티션)을 식별하는 정보가 들어 있습니다.

인스턴스 레벨에서는 감사 기능을 중지한 다음 db2audit start 및 db2audit stop 명령을 사용하여 명시적으로 시작해야 합니다. 인스턴스 레벨 감사가 시작되면 기존 감사 구성 정보가 사용됩니다. 감사 기능은 DB2 데이터베이스 서버와 무관하므로 인스턴스가 중지되는 경우에도 계속 활성 상태입니다. 실제로 인스턴스가 중지되면, 감사 레코드가 감사 로그에 생성될 수 있습니다. 데이터베이스 레벨에서 감사를 시작하려면 먼저 감사 규정을 작성한 다음 이 감사 규정을 모니터하려는 오브젝트(예: 권한 부여 ID, 데이터베이스 권한, 트러스트된 컨텍스트 또는 특정 테이블)와 연관시켜야 합니다.

감사 레코드의 범주

생성될 수 있는 여러 가지 범주의 감사 레코드가 있습니다. 감사에 사용 가능한 이벤트 범주에 대한 아래 설명에서 각 범주의 이름 뒤에는 범주 유형을 식별하는 데 사용되는 한 단어로 된 키워드가 옵니다. 감사에 사용 가능한 이벤트의 범주는 다음과 같습니다.

- 감사(AUDIT). 감사 설정이 변경될 때 또는 감사 로그에 액세스할 때 레코드를 생성합니다.
- 권한 부여 검사(CHECKING). DB2 데이터베이스 오브젝트 또는 기능을 액세스 또는 조작하려는 시도의 권한 부여 검사 중에 레코드를 생성합니다.
- 오브젝트 유지보수(OBJMAINT). 데이터 오브젝트를 작성 또는 삭제할 때 또는 특정 오브젝트를 변경하는 경우 레코드를 생성합니다.
- 보안 유지보수(SECMAINT). 다음과 같은 경우에 레코드를 생성합니다.
  - 오브젝트 특권 또는 데이터베이스 권한을 부여하거나 취소하는 경우
  - 보안 레이블 또는 면제 권한을 부여하거나 취소하는 경우
  - 그룹 권한 부여, 역할 권한 부여 또는 LBAC 보안 규정의 겹쳐쓰기 또는 제한 속성을 변경하는 경우
  - SETSESSIONUSER 특권을 부여하거나 취소하는 경우
  - SYSADM_GROUP, SYSCTRL_GROUP, SYSMAINT_GROUP 또는 SYSMON_GROUP 구성 매개변수를 수정하는 경우
- 시스템 관리(SYSADMIN). SYSADM, SYSMAINT 또는 SYSCTRL 권한이 필요한 조작이 수행될 때 레코드를 생성합니다.
- 사용자 유효성 확인(VALIDATE). 사용자를 인증하거나 시스템 보안 정보를 검색할 때 레코드를 생성합니다.
- 조작 컨텍스트(CONTEXT). 데이터베이스 조작이 수행될 때 조작 컨텍스트를 표시하기 위해 레코드를 생성합니다. 이 범주는 감사 로그 파일의 더 나은 해석을 허용합니다. 로그의 이벤트 상관자 필드를 사용할 때, 이벤트 그룹은 다시 하나의 데이터베이스 조작에 연관될 수 있습니다. 예를 들어, 동적 쿼리에 대한 쿼리 명령문, 정적 쿼리를 위한 패키지 ID 또는 CONNECT와 같이 수행되는 조작 유형의 표시기는 감사 결과를 분석할 때 필요한 컨텍스트를 제공할 수 있습니다.

  주: 조작 컨텍스트를 제공하는 SQL 또는 XQuery문은 매우 길 수 있으며 CONTEXT 레코드에서 완전히 표시됩니다. 이것은 CONTEXT 레코드를 매우 크게 만들 수 있습니다.
- 실행(EXECUTE). SQL문 실행 중 레코드를 생성합니다.

위 범주에 대해 실패 또는 성공 중 하나를 감사하거나 둘 다 감사할 수 있습니다.

데이터베이스 서버에 대한 조작을 수행하면 여러 개의 레코드가 생성될 수 있습니다. 감사 로그에 생성되는 실제 레코드 수는 감사 기능 구성에 지정된 기록할 이벤트 범주 수에 따라 달라집니다. 또한 성공, 실패 또는 둘 다 감사되는지 여부에 따라 다릅니다. 이 때문에 감사할 이벤트를 선택할 수 있는 것이 중요합니다.

감사 규정

보안 관리자는 감사 규정을 사용하여 필요한 데이터와 오브젝트에 대한 정보만 수집하도록 감사 기능을 구성할 수 있습니다.

보안 관리자는 감사 규정을 작성하여 개별 데이터베이스 내에서 감사되는 대상을 제어할 수 있습니다. 연관된 감사 규정이 있는 오브젝트는 다음과 같습니다.

- 전체 데이터베이스

  데이터베이스 내에서 발생하는 모든 감사 가능한 이벤트가 감사 규정에 따라 감사를 받습니다.

- 테이블

  모든 데이터 처리 언어(DML) 및 테이블(유형이 지정되지 않음), 구체화된 쿼리 테이블(MQT) 또는 별칭에 대한 XQUERY 액세스가 감사 대상입니다. 규정에 다른 범주를 감사해야 된다고 표시되어 있어도, 테이블에 액세스할 때 데이터가 있거나 없는 EXECUTE 범주 감사 이벤트만 생성됩니다.

- 트러스트된 컨텍스트

  특정 트러스트된 컨텍스트에 의해 정의된 트러스트된 연결 내에서 발생하는 모든 감사 가능한 이벤트가 감사 규정에 따라 감사를 받습니다.

- 사용자, 그룹 또는 역할을 나타내는 권한 부여 ID

  지정된 사용자에 의해 시작된 모든 감사 가능한 이벤트가 감사 규정에 따라 감사를 받습니다.

  그룹 또는 역할의 구성원인 사용자에 의해 시작된 모든 감사 가능한 이벤트가 감사 규정에 따라 감사를 받습니다. 다른 역할 또는 그룹을 통해 사용되는 간접적 역할 멤버쉽도 포함됩니다.

  워크로드 관리 이벤트 모니터를 사용하거나 그룹에 대한 워크로드를 정의한 다음 활동 세부사항을 캡처하는 방식으로 유사한 데이터를 캡처할 수 있습니다. 워크로드에 대한 맵핑에는 권한 부여 ID뿐만 아니라 속성도 포함될 수 있는데 이로 인해 원하는 자세한 감사 결과를 얻지 못할 수 있으며, 다른 속성이 수정된 경우 연결이 다른 워크로드(모니터되지 않을 수 있음)에 맵핑될 수 있습니다. 감사 솔루션은 사용자, 그룹 또는 역할을 감사할 수 있도록 보장합니다.

- 권한(SYSADM, SECADM, DBADM, SQLADM, WLMADM, ACCESSCTRL, DATAACCESS, SYSCTRL, SYSMAINT, SYSMON)

  지정된 권한이 이벤트에 필요하지 않은 경우에도 해당 권한을 보유하는 사용자가 시작한 모든 감사 가능한 이벤트가 감사 규정에 따라 감사를 받습니다.

보안 관리자는 감사 규정을 여러 개 작성할 수 있습니다. 예를 들어, sensitive 데이터를 감사하는 규정이나 DBADM 권한을 보유한 사용자의 활동을 감사하는 규정이 회사에 필요할 수 있습니다. 한 명령문에 여러 개의 감사 규정이 적용되는 경우, 개별 감사 규정을 통해 감사되어야 하는 모든 이벤트가 감사됩니다(한 번만 감사됨). 예를 들어, 데이터베이스 감사 규정에 따라 특정 테이블의 성공한 EXECUTE 이벤트를 감사해야 하고, 사용자 감사 규정에 따라 동일한 테이블의 실패한 EXECUTE 이벤트를 감사해야 하는 경우, 해당 테이블에 액세스할 때 성공 및 실패한 시도가 모두 감사됩니다.

특정 오브젝트의 경우 한 개의 감사 규정만 적용될 수 있습니다. 예를 들어, 동일한 테이블과 연관된 감사 규정이 동시에 여러 개 있을 수는 없습니다.

감사 규정은 뷰 또는 유형이 지정된 테이블과 연관될 수 없습니다. 연관된 감사 규정이 있는 테이블에 액세스하는 뷰는 기본 테이블의 규정에 따라 감사를 받습니다.

테이블에 적용되는 감사 규정은 해당 테이블을 기반으로 하는 MQT에 자동으로 적용되지 않습니다. 감사 규정을 테이블과 연관시키는 경우 해당 테이블을 기반으로 하는 MQT와 동일한 규정을 연관시키십시오.

트랜잭션 중에는 감사 규정 및 트랜잭션 시작 시의 감사 규정 연관을 기반으로 감사가 수행됩니다. 예를 들어 보안 관리자가 감사 규정을 사용자와 연관시키는데 그 당시에 해당 사용자가 트랜잭션 내에 있을 경우, 해당 트랜잭션 내에서 수행되는 나머지 명령문에는 감사 규정이 적용되지 않습니다. 또한 감사 규정에 대한 변경사항이 커미트될 때까지는 이러한 변경사항이 적용되지 않습니다. 보안 관리자가 ALTER AUDIT POLICY 문을 발행할 경우 해당 명령문이 커미트될 때까지 적용되지 않습니다.

보안 관리자는 CREATE AUDIT POLICY문을 사용하여 감사 규정을 작성하고, ALTER AUDIT POLICY문을 사용하여 감사 규정을 수정합니다. 이러한 명령문에는 다음 사항을 지정할 수 있습니다.

- 감사할 이벤트의 상태 값: 없음, 성공, 실패, 또는 둘 다

  지정된 상태 값과 일치하는 감사 가능 이벤트만 감사를 받습니다.
- 감사 중 오류가 발생할 경우의 서버 동작

보안 관리자는 AUDIT문을 사용하여 현재 데이터베이스 또는 현재 서버에 있는 데이터베이스 오브젝트와 감사 규정을 연관시킵니다. 오브젝트가 사용 중인 경우에는 항상 이 감사 규정에 따라 감사를 받습니다.

보안 관리자는 DROP문을 사용하여 감사 규정을 삭제할 수 있습니다. 감사 규정이 오브젝트와 연관되어 있는 경우에는 감사 규정을 삭제할 수 없습니다. 오브젝트와 연관된 나머지 연관을 제거하려면 AUDIT REMOVE문을 사용하십시오. 보안 관리자는 COMMENT문을 사용하여 감사 규정에 메타데이터를 추가할 수 있습니다.

전체 연결이 설정되기 전에 생성되는 이벤트

연결 및 사용자 전환 조작 중에 생성되는 일부 이벤트의 경우, 유일하게 사용 가능한 규정 정보는 데이터베이스와 연관된 규정입니다. 이러한 이벤트는 다음 표와 같습니다.

표 4. 연결 이벤트

이벤트	감사 범주	설명
CONNECT	CONTEXT	
CONNECT_RESET	CONTEXT	
AUTHENTICATION	VALIDATE	트러스트된 연결 내에서 연결 및 사용자 전환 중에 수행되는 인증이 포함됩니다.
CHECKING_FUNC	CHECKING	시도된 액세스는 SWITCH_USER입니다.

이러한 이벤트는 다른 오브젝트(예: 사용자, 그룹 또는 권한)와 연관된 감사 규정이 아니라, 데이터베이스와 연관된 감사 규정에 따라 감사를 받습니다. 연결 중에 발생하는 CONNECT 및 AUTHENTICATION 이벤트의 경우, 데이터베이스가 활성화될 때까지 인스턴스 레벨 감사 설정이 사용됩니다. 데이터베이스는 처음 연결할 때 또는 ACTIVATE DATABASE 명령이 발행될 때 활성화됩니다.

사용자 전환 결과

트러스트된 연결 내에서 한 사용자가 전환되면 원래 사용자는 아무도 남아 있지 않게 됩니다. 이 경우 원래 사용자와 연관된 감사 규정이 더 이상 고려되지 않으므로 적용 가능한 감사 규정이 새로운 사용자에 따라 재평가됩니다. 트러스트된 연결과 연관된 감사 규정은 여전히 유효합니다.

SET SESSION USER문을 사용한 경우 세션 권한 부여 ID만 전환됩니다. 원래 사용자 권한 부여 ID(시스템 권한 부여 ID)의 감사 규정은 계속 유효하며 새로운 사용자의 감사 규정도 사용됩니다. 하나의 세션 내에서 여러 개의 SET SESSION USER문이 발행된 경우 원래 사용자(시스템 권한 부여 ID)와 현재 사용자(세션 권한 부여 ID)에 연관된 감사 규정만 고려됩니다.

DDL 제한사항

다음 데이터 정의 언어(DDL)문은 AUDIT 독점 SQL문이라고 합니다.

- AUDIT
- CREATE AUDIT POLICY, ALTER AUDIT POLICY 및 DROP AUDIT POLICY
- DROP ROLE 및 DROP TRUSTED CONTEXT(삭제하려는 역할 또는 트러스트된 컨텍스트가 감사 규정과 연관된 경우)

AUDIT 독점 SQL 문 사용과 관련된 몇 가지 제한사항은 다음과 같습니다.

- 각 명령문 다음에 COMMIT 또는 ROLLBACK이 있어야 합니다.

• 이러한 명령문은 XA 트랜잭션과 같은 전역 트랜잭션 내에서 발행할 수 없습니다.

모든 파티션에서 한 번에 단 하나의 커미트되지 않은 AUDIT 독점 DDL문만 허용됩니다. 커미트되지 않은 AUDIT 독점 DDL문이 실행 중인 경우, 이후 AUDIT 독점 DDL문은 현재 AUDIT 독점 DDL문이 커미트되거나 롤백될 때까지 기다립니다.

주: 변경사항은 시스템 카탈로그에 기록되지만, 명령문을 발행하는 연결일지라도 COMMIT이 실행될 때까지는 적용되지 않습니다.

특정 테이블에 대한 액세스 감사 예

회사의 EMPLOYEE 테이블에 매우 중요한 정보가 들어 있어 이 테이블에 있는 데이터에 대한 모든 SQL 액세스를 감사하려고 합니다. EXECUTE 범주를 사용하여 테이블에 대한 모든 액세스를 추적할 수 있습니다. 이 범주는 SQL문을 감사하고 해당 명령문 실행 시에 제공된 입력 데이터 값을 선택적으로 감사합니다.

테이블에 대한 활동을 추적하는 단계는 두 단계로 구성됩니다. 먼저 보안 관리자가 EXECUTE 범주를 지정하는 감사 규정을 작성한 다음 이 규정을 테이블과 연관시킵니다.

```
CREATE AUDIT POLICY SENSITIVEDATAPOLICY
    CATEGORIES EXECUTE STATUS BOTH ERROR TYPE AUDIT
COMMIT

AUDIT TABLE EMPLOYEE USING POLICY SENSITIVEDATAPOLICY
COMMIT
```

SYSADM 또는 DBADM에 의한 조치 감사 예

보안 준수 인증을 완료하려면 회사에서 시스템 관리(SYSADM) 또는 데이터베이스 관리(DBADM) 권한을 보유한 사용자가 수행하는 데이터베이스 내의 모든 활동을 모니터할 수 있다는 것을 표시해야 합니다.

데이터베이스 내의 모든 조치를 캡처하려면 EXECUTE 및 SYSADMIN 범주를 둘 다 감사해야 합니다. 보안 관리자는 이러한 두 범주를 감사하는 감사 규정을 작성합니다. AUDIT문을 사용하여 이 감사 규정을 SYSADM 및 DBADM 권한과 연관시킬 수 있습니다. 그러면 SYSADM 또는 DBADM 권한을 보유한 사용자가 감사 가능한 이벤트를 로깅할 수 있습니다. 다음 예는 이러한 감사 규정을 작성한 다음 SYSADM 및 DBADM 권한과 연관시키는 방법을 보여줍니다.

```
CREATE AUDIT POLICY ADMINSPOLICY CATEGORIES EXECUTE STATUS BOTH,
   SYSADMIN STATUS BOTH ERROR TYPE AUDIT
COMMIT
AUDIT SYSADM, DBADM USING POLICY ADMINSPOLICY
COMMIT
```

특정 역할에 의한 액세스 감사 예

회사에서 사내 데이터베이스에 대한 웹 응용 프로그램 액세스를 허용했습니다. 웹 응용 프로그램을 사용하는 개인은 정확하게 파악할 수 없습니다. 사용된 역할만 알 수 있으므로 해당 역할을 데이터베이스 권한 부여를 관리하는 데 사용합니다. 회사에서는 해당 역할의 구성원인 사용자의 조치를 모니터하여 이들이 데이터베이스에 제출한 요청을 검사하고 웹 응용 프로그램을 통해서만 데이터베이스에 액세스할 수 있게 하려고 합니다.

EXECUTE 범주에는 이러한 상황에 대한 사용자의 활동을 추적하는 데 필요한 수준의 감사가 포함되어 있습니다. 적절한 감사 규정을 작성한 다음 웹 응용 프로그램에서 사용하는 역할(이 예의 경우 TELLER 및 CLERK)을 해당 규정과 연관시키십시오.

```
CREATE AUDIT POLICY WEBAPPPOLICY CATEGORIES EXECUTE WITH DATA
   STATUS BOTH ERROR TYPE AUDIT
COMMIT
AUDIT ROLE TELLER, ROLE CLERK USING POLICY WEBAPPPOLICY
COMMIT
```

감사 로그의 분석 및 스토리지

감사 로그를 아카이브하면 서버가 새로운 활성 감사 로그에 쓰기 시작하는 동안 활성 감사 로그가 아카이브 디렉토리로 이동됩니다. 나중에 아카이브된 로그의 데이터를 구분 파일에 추출한 다음 이러한 파일의 데이터를 DB2 데이터베이스 테이블에 로드하여 분석할 수 있습니다.

감사 로그의 위치를 구성하면 데이터베이스 파티셔닝 기능(DPF) 설치의 각 노드에 별도의 디스크를 사용할 수 있는 옵션과 함께 크기가 큰 고속 디스크에 감사 로그를 배치할 수 있습니다. DPF 환경에서, 아카이브 감사 로그의 경로는 각 노드에 고유한 디렉토리가 될 수 있습니다. 각 노드에 고유한 디렉토리를 사용하면, 각 노드가 다른 디스크에 쓰기 때문에 파일 경쟁을 피할 수 있습니다.

Windows 운영 체제에서 감사 로그의 기본 경로는 *instance*₩security₩auditdata이고, Linux 및 UNIX 운영 체제에서는 *instance*/security/auditdata입니다. 기본 위치를 사용하지 않고 다른 디렉토리를 선택할 수 있습니다(다른 디렉토리가 없는 경우 시스템에 새 디렉토리를 작성하여 대체 위치로 사용할 수 있음). 아카이브 감사 로그 위치와 아카이브된 감사 로그 위치의 경로를 설정하려면 다음 예제에서와 같이 datapath 및 archivepath 매개변수와 함께 db2audit configure 명령을 사용하십시오.

```
db2audit configure datapath /auditlog archivepath /auditarchive
```

db2audit를 사용하여 설정한 감사 로그 스토리지 위치는 해당 인스턴스의 모든 데이터베이스에 적용됩니다.

주: 서버에 여러 개의 인스턴스가 있는 경우, 각 인스턴스에는 별도의 데이터와 아카이브 경로가 있어야 합니다.

DPF 환경에서 아카이브 감사 로그의 경로(datapath)

DPF 환경에서, 각 파티션에 동일한 활성 감사 로그 위치(**datapath** 매개변수에 의해 설정됨)를 사용해야 합니다. 두 가지 방법으로 이를 수행할 수 있습니다.

1. **datapath** 매개변수를 지정할 때 데이터베이스 파티션 표현식을 사용하십시오. 데이터베이스 파티션 표현식을 사용하면 감사 로그 파일의 경로에 파티션 번호를 포함할 수 있으므로 각 데이터베이스 파티션에 다른 경로를 사용할 수 있습니다.
2. 모든 노드에서 동일한 공유 드라이브를 사용하십시오.

datapath 매개변수에 지정한 값 내에서 데이터베이스 파티션 표현식을 사용할 수 있습니다. 예를 들어, 데이터베이스 파티션 번호가 10인 3-노드 시스템에서 다음 명령은

```
db2audit configure datapath '/pathForNode $N'
```

다음과 같은 파일을 작성합니다.

- `/pathForNode10`
- `/pathForNode20`
- `/pathForNode30`

주: 데이터베이스 파티션 표현식을 사용하여 아카이브 로그 파일 경로(**archivepath** 매개변수)를 지정할 수 없습니다.

활성 감사 로그 아카이브

시스템 관리자가 db2audit 도구를 사용하여 인스턴스 및 데이터베이스 감사 로그를 둘 다 아카이브하고 어느 한 유형의 아카이브된 로그에서 감사 데이터를 추출할 수 있습니다.

보안 관리자 또는 보안 관리자가 감사 루틴에 EXECUTE 특권을 부여한 사용자는 SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저를 실행하여 활성 감사 로그를 아카이브할 수 있습니다. SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저를 사용하여 로그에서 데이터를 추출하고 이를 구분 파일에 로드할 수 있습니다.

다음은 감사 루틴을 사용하여 감사 로그를 아카이브하고 추출하는 단계입니다.

1. SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저를 사용하여 활성 감사 로그의 일반 아카이브를 수행하도록 응용프로그램의 스케줄을 설정하십시오.
2. 원하는 아카이브된 로그 파일을 판별하십시오. SYSPROC.AUDIT_LIST_LOGS 테이블 함수를 사용하여 아카이브된 감사 로그를 모두 나열하십시오.
3. 파일 이름을 SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저에 매개변수로 전달하여 로그에서 데이터를 추출한 다음 이를 구분 파일에 로드하십시오.
4. 분석을 위해 감사 데이터를 DB2 데이터베이스 테이블에 로드하십시오.

아카이브된 로그 파일을 분석하기 위해 테이블에 즉시 로드할 필요는 없습니다. 저장한 후 나중에 분석해도 됩니다. 예를 들어, 기업 데이터 감사가 수행될 때 아카이브된 로그 파일을 검색하기만 하면 됩니다.

아카이브하는 동안 문제점이 발생하면(예: 아카이브 경로에 디스크 공간 부족 또는 아카이브 경로가 존재하지 않는 경우), 아카이브 프로세스에 실패하며 파일 확장자가 .bk인 중간 로그 파일이 감사 로그 데이터 경로에 생성됩니다(예: db2audit.instance.log.0.20070508172043640941.bk). 문제점이 해결되면(예: 아카이브 경로에 디스크 공간을 충분히 할당하거나 아카이브 경로 작성) 이 중간 로그를 아카이브 경로에 이동해야 합니다. 그러면 이 로그를 성공적으로 아카이브된 로그와 동일한 방법으로 사용할 수 있습니다.

DPF 환경에서 활성 감사 로그 아카이브

DPF 환경에서, 인스턴스가 실행되는 중에 아카이브 명령이 발생되면 아카이브 프로세스가 모든 노드에서 자동으로 실행됩니다. 모든 노드에서 아카이브 로그 파일 이름에 동일한 시간소인이 사용됩니다. 예를 들어, 데이터베이스 파티션 번호가 10인 3-노드 시스템에서 다음 명령은

```
db2audit archive to /auditarchive
```

다음과 같은 파일을 작성합니다.

- /auditarchive/db2audit.log.10.*timestamp*
- /auditarchive/db2audit.log.20.*timestamp*
- /auditarchive/db2audit.log.30.*timestamp*

인스턴스가 실행되지 않는 동안 아카이브 명령이 발생되면, 다음 방법 중 하나로 아카이브를 실행할 노드를 제어할 수 있습니다.

- 현재 노드에서만 아카이브를 수행하려면 db2audit 명령과 함께 node 옵션을 사용하십시오.
- 모든 노드에서 아카이브를 실행하려면 db2_all 명령을 사용하십시오.

  예를 들어, 다음과 같습니다.

  ```
  db2_all db2audit archive node to /auditarchive
  ```

  이 명령을 사용하면 명령이 호출되는 노드를 나타내도록 DB2NODE 환경 변수가 설정됩니다.

또는 각 노드에 별도로 개별 아카이브 명령을 발행할 수 있습니다. 예를 들어, 다음과 같습니다.

- 노드 10에서:

  ```
  db2audit archive node 10 to /auditarchive
  ```

- 노드 20에서:

  ```
  db2audit archive node 20 to /auditarchive
  ```

- 노드 30에서:

  ```
  db2audit archive node 30 to /auditarchive
  ```

주: 인스턴스가 실행되고 있지 않을 때, 아카이브된 감사 로그 파일 이름의 시간소인은 각 노드에서 동일하지 않습니다.

주: 모든 노드에서 아카이브 경로를 공유하는 것이 좋지만 필수는 아닙니다.

주: AUDIT_DELIM_EXTRACT 스토어드 프로시저 및 AUDIT_LIST_LOGS 테이블 함수는 현재(코디네이터) 노드에 표시되는 아카이브된 로그 파일에만 액세스할 수 있습니다.

로그 아카이브 및 테이블에 데이터 추출의 예

회사에서 감사 데이터를 캡처하여 나중에 사용하기 위해 저장하려면 6시간 마다 새 감사 로그를 작성하여 현재 감사 로그를 WORM 드라이브에 아카이브해야 합니다. 회사에서 보안 관리자 또는 보안 관리자가 AUDIT_ARCHIVE 스토어드 프로시저에서 EXECUTE 특권을 부여한 사용자가 6시간 마다 다음 SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저 호출을 발행하도록 스케줄을 설정합니다. 아카이브된 로그의 경로는 기본 아카이브 경로인 /auditarchive이며, 모든 노드에서 아카이브가 실행됩니다.

```
CALL SYSPROC.AUDIT_ARCHIVE( '/auditarchive', -2 )
```

회사는 보안 프로시저의 일부로서, 의심스러운 많은 동작을 식별하여 정의했거나 감사 데이터에서 감시가 필요한 활동을 허용하지 않았습니다. 여러 감사 로그의 모든 데이터를 추출하여 관련 테이블에 배치한 다음 SQL 쿼리를 사용하여 이러한 활동을 찾으려고 합니다. 회사는 감사할 적절한 범주를 결정하고 필요한 감사 규정을 데이터베이스 또는 기타 데이터베이스 오브젝트에 연관시켰습니다.

예를 들어, SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저를 호출하여 기본 분리문자를 사용하여 2006년 4월 시간소인으로 작성된 모든 노드에서 모든 범주의 아카이브된 감사 로그를 추출할 수 있습니다.

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT(
     '', '', '/auditarchive',  'db2audit.%.200604%', '' )
```

다른 예를 들어, SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저를 호출하여 원하는 시간소인을 사용하여 파일에서 EXECUTE 범주의 성공 이벤트와 CHECKING 범주의 실패 이벤트가 있는 아카이브된 감사 레코드를 추출할 수 있습니다.

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT( '', '', '/auditarchive',
    'db2audit.%.20060419034937', 'categories
     execute status success, checking status failure );
```

감사 로그 파일 이름:

감사 로그 파일에는 인스턴스 레벨인지 또는 데이터베이스 레벨 로그인지 구분하고 데이터베이스 파티셔닝 기능(DPF) 환경에서 비롯되는 파티션을 구분하는 이름이 있습니다. 아카이브된 감사 로그에는 파일 이름에 추가되어 아카이브 명령이 실행된 시기를 나타내는 시간소인이 있습니다.

활성 감사 로그 파일 이름

DPF 환경에서, 활성 감사 로그의 경로는 각 파티션에 고유한 디렉토리이므로 각 파티션이 개별 파일에 쓸 수 있습니다. 감사 레코드의 원점을 정확하게 추적할 수 있도록 파티션 번호가 감사 로그 파일 이름의 일부로 포함되어 있습니다. 예를 들어, 파티션 20에서 인스턴스 레벨 감사 로그 파일 이름은 db2audit.instance.log.20입니다. 이 인스턴스에서 testdb라고 하는 데이터베이스의 경우 감사 로그 파일은 db2audit.db.testdb.log.20입니다.

비DPF 환경에서 파티션 번호는 0(영)으로 간주됩니다. 이 경우, 인스턴스 레벨 감사 로그 파일 이름은 db2audit.instance.log.0입니다. 이 인스턴스에서 testdb라고 하는 데이터베이스의 경우 감사 로그 파일은 db2audit.db.testdb.log.0입니다.

아카이브된 감사 로그 파일 이름

활성 감사 로그가 아카이브되면 *YYYYMMDDHHMMSS* 형식으로 현재 시간소인이 파일 이름에 추가됩니다. 여기서 YYYY는 년, MM은 월, DD는 일, HH는 시, MM은 분, SS는 초입니다.

아카이브 감사 로그의 파일 이름 형식은 감사 로그의 레벨에 따라 다릅니다.

인스턴스 레벨 아카이브된 감사 로그

인스턴스 레벨 아카이브된 감사 로그의 파일 이름은 db2audit.instance.log.*partition*.*YYYYMMDDHHMMSS*입니다.

데이터베이스 레벨 아카이브된 감사 로그

데이터베이스 레벨 아카이브된 감사 로그의 파일 이름은 db2audit.db*database*.log.*partition*.*YYYYMMDDHHMMSS*입니다.

비DPF 환경에서 *partition*의 값은 0(영)입니다.

시간소인은 아카이브 명령이 실행된 시간을 나타내므로 로그의 마지막 레코드 시간이 항상 정확하게 반영되지는 않습니다. 다음과 같은 이유로 아카이브된 감사 로그 파일에는 로그 파일 이름의 시간소인보다 몇 초 늦은 시간 소인이 있는 레코드가 있을 수 있습니다.

- 아카이브 명령이 발행될 때, 감사 기능은 아카이브된 로그 파일을 작성하기 전에 진행 중인 레코드 쓰기가 완료되기를 기다립니다.

- 여러 시스템 환경에서는 리모트 시스템의 시스템 시간이 아카이브 명령이 발행되는 시스템과 동기화되지 않을 수 있습니다.

DPF 환경에서, 아카이브가 실행될 때 서버가 실행되고 있으면 파티션 전체에서 시간소인이 일치하고 아카이브가 수행된 파티션에서 생성된 시간소인을 반영합니다.

DB2 감사 데이터를 보유할 테이블 작성:

데이터베이스 테이블의 감사 데이터를 사용하려면 데이터를 보유할 테이블을 작성해야 합니다. 테이블의 데이터를 권한이 없는 사용자로부터 격리하려면 별도의 스키마에 이들 테이블을 작성해야 합니다.

- 스키마를 작성하는 데 필요한 권한 및 특권에 대해서는 CREATE SCHEMA문을 참조하십시오.
- 테이블을 작성하는 데 필요한 권한 및 특권에 대해서는 CREATE TABLE문을 참조하십시오.
- 테이블을 보유하는 데 사용할 테이블 스페이스를 결정하십시오. (이 주제에서는 테이블 스페이스를 작성하는 방법을 설명하지 않습니다.)

주: 감사 데이터 보관을 위해 작성해야 하는 테이블의 형식은 릴리스별로 다를 수 있습니다. 새 컬럼이 추가되거나 기존 컬럼의 크기가 변경될 수 있습니다. db2audit.ddl 스크립트는 감사 레코드를 포함할 수 있는 올바른 형식의 테이블을 작성합니다.

다음 예는 구분 파일의 레코드를 보유할 테이블을 작성하는 방법을 보여줍니다. 원하는 경우, 이들 테이블을 포함할 별도의 스키마를 작성할 수 있습니다.

파일에 포함된 일부 데이터를 사용하지 않으려는 경우, 필요에 따라 테이블 정의에서 컬럼을 생략하거나 특정 테이블 작성을 생략할 수 있습니다. 테이블 정의에서 컬럼을 생략하는 경우, 이들 테이블에 데이터를 로드하는 데 사용하는 명령을 수정해야 합니다.

1. DB2 명령 창을 열려면 db2 명령을 발행하십시오.
2. 선택사항. 테이블을 보유할 스키마를 작성하십시오. 다음 예에서 스키마는 AUDIT입니다.

   ```
   CREATE SCHEMA AUDIT
   ```

3. 선택사항. AUDIT 스키마를 작성한 경우, 테이블을 작성하기 전에 스키마로 전환하십시오.

   ```
   SET CURRENT SCHEMA = 'AUDIT'
   ```

4. db2audit.ddl 스크립트를 실행하여 감사 레코드를 포함하는 테이블을 작성하십시오.

   db2audit.ddl 스크립트는 sqllib/misc 디렉토리에 있습니다(Windows에서 sqllib₩misc). 이 스크립트에서는 데이터베이스와 연결되어 있고 8K의 테이블 스페이스를 사용할 수 있다고 가정합니다. 스크립트를 실행할 명령은 db2 +o -tf

sqllib/misc/db2audit.ddl입니다. 스크립트가 작성하는 테이블은 AUDIT, CHECKING, OBJMAINT, SECMAINT, SYSADMIN, VALIDATE, CONTEXT 및 EXECUTE 입니다.

5. 테이블을 작성하면, 보안 관리자가 SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저를 사용하거나 시스템 관리자가 db2audit extract 명령을 사용하여 아카이브된 감사 로그 파일에서 감사 레코드를 구분 파일에 추출할 수 있습니다. 구분 파일에서 감사 데이터를 방금 작성한 데이터베이스 테이블에 로드할 수 있습니다.

DB2 감사 데이터를 테이블에 로드:

감사 로그 파일을 구분 파일에 아카이브하고 추출한 후 감사 데이터를 보유할 데이터베이스 테이블을 작성하면, 구분 파일에서 감사 데이터를 데이터베이스 테이블에 로드하여 분석할 수 있습니다.

로드 유틸리티를 사용하여 감사 데이터를 테이블에 로드할 수 있습니다. 각각의 테이블에 대하여 별도의 로드 명령을 발행하십시오. 테이블 정의에서 하나 이상의 컬럼을 생략한 경우, 데이터를 성공적으로 로드하는 데 사용되는 LOAD 명령의 버전을 수정해야 합니다. 또한 감사 데이터를 추출할 때 디폴트가 아닌 구분 문자를 지정한 경우, 사용하는 LOAD 명령의 버전도 수정해야 합니다.

1. DB2 명령 창을 열려면 db2 명령을 발행하십시오.
2. AUDIT 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM audit.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.AUDIT
   ```

   주: 2진 데이터를 올바르게 구문 분석할 수 있도록 DELPRIORITYCHAR 수정자를 지정하십시오.

   주: LOAD 명령의 LOBSINFILE 옵션을 지정하십시오(*크기*가 큰 오브젝트의 인라인 데이터를 32K로 제한해야 하는 제한사항으로 인해). LOBS FROM 옵션을 사용해야 하는 경우도 있습니다.

   주: 파일 이름 지정 시 완전한 경로 이름을 사용하십시오. 예를 들어, Windows 기반 컴퓨터의 C: 드라이브에 DB2 데이터베이스 시스템을 설치한 경우 `audit.del` 파일의 완전한 파일 이름으로 `C:¥Program Files¥IBM¥SQLLIB¥`*instance*`¥security¥audit.del`을 지정해야 합니다.
3. CHECKING 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM checking.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.CHECKING
   ```
4. OBJMAINT 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM objmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.OBJMAINT
   ```

5. SECMAINT 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM secmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.SECMAINT
   ```

6. SYSADMIN 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM sysadmin.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.SYSADMIN
   ```

7. VALIDATE 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM validate.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.VALIDATE
   ```

8. CONTEXT 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM context.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.CONTEXT
   ```

9. EXECUTE 테이블을 로드하려면, 다음 명령을 발행하십시오.

   ```
   LOAD FROM execute.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.EXECUTE
   ```

10. 테이블에 데이터 로드를 완료한 후, sqllib 디렉토리의 security/auditdata 서브디렉토리에서 .del 파일을 삭제하십시오.
11. 테이블에 감사 데이터를 로드한 후, 이러한 테이블에서 데이터를 선택하여 분석할 수 있습니다.

테이블을 이미 채웠으며 다시 이를 수행하려는 경우, INSERT 옵션을 사용하여 새 테이블 데이터를 기존 테이블 데이터에 추가하십시오. 이전 db2audit extract 조작의 레코드를 테이블에서 제거하려는 경우, REPLACE 옵션을 사용하여 테이블을 다시 로드하십시오.

아카이브 감사 및 스토어드 프로시저 추출:

보안 관리자는 SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저 및 테이블 함수, SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저 및 SYSPROC.AUDIT_LIST_LOGS 테이블 함수를 사용하여 감사 로그를 아카이브하고 데이터를 구분된 파일로 추출할 수 있습니다.

보안 관리자는 이러한 루틴에 대한 EXECUTE 특권을 사용자에게 부여하여 해당 루틴의 사용을 다른 사용자에게 위임할 수 있습니다. 보안 관리자만 이러한 루틴에 대한 EXECUTE 특권을 부여할 수 있습니다. EXECUTE 특권의 WITH GRANT OPTION은 이러한 루틴에 대해 부여할 수 없습니다(SQLSTATE 42501).

이러한 스토어드 프로시저와 테이블 함수를 사용하여 데이터베이스의 감사 로그를 아카이브 또는 나열하려면 데이터베이스에 연결된 상태여야 합니다.

아카이브된 파일을 다른 데이터베이스 시스템으로 복사하고, 스토어드 프로시저와 테이블 함수를 사용하여 이러한 파일에 액세스하려는 경우, 데이터베이스 이름이 같은지 확인하고 그렇지 않을 경우 같은 데이터베이스 이름을 포함하도록 파일 이름을 바꾸십시오.

이러한 스토어드 프로시저와 테이블 함수는 인스턴스 레벨 감사 로그를 아카이브 또는 나열하지 않습니다. 시스템 관리자가 db2audit 명령을 사용하여 인스턴스 레벨 감사 로그를 아카이브하고 추출해야 합니다.

스토어드 프로시저와 테이블 함수를 사용하여 수행할 수 있는 조작은 다음과 같습니다.

표 5. 시스템 스토어드 프로시저 및 테이블 함수 감사

스토어드 프로시저 및 테이블 함수	조작	설명
AUDIT_ARCHIVE	현재 감사 로그를 아카이브합니다.	아카이브 경로가 입력으로 사용됩니다. 아카이브 경로를 제공하지 않을 경우, 이 스토어드 프로시저는 감사 구성 파일의 아카이브 경로를 사용합니다. 아카이브는 각 노드에서 실행되며 동기화된 시간소인이 감사 로그 파일의 이름에 추가됩니다.
AUDIT_LIST_LOGS	현재 데이터베이스에 대해 지정된 경로에서 아카이브된 감사 로그 목록을 리턴합니다.	

표 5. 시스템 스토어드 프로시저 및 테이블 함수 감사 (계속)

스토어드 프로시저 및 테이블 함수	조작	설명
AUDIT_ DELIM_EXTRACT	2진 아카이브 로그에서 데이터를 추출하여 구분된 파일로 로드합니다.	추출된 감사 레코드가 DB2 데이터베이스 테이블로 로드하는 데 적합한 구분된 형식으로 배치됩니다. 출력은 각 범주에 대해 하나씩, 별도 파일에 배치됩니다. 또한 감사 데이터에 포함되는 모든 대형 오브젝트(LOB)를 포함할 수 있도록 **auditlobs** 파일도 작성됩니다. 파일 이름은 다음과 같습니다. • **audit.del** • **checking.del** • **objmaint.del** • **secmaint.del** • **sysadmin.del** • **validate.del** • **context.del** • **execute.del** • **auditlobs** 파일이 이미 존재하면 출력이 파일에 추가됩니다. CONTEXT 또는 EXECUTE 범주가 추출된 경우 **auditlobs** 파일이 작성됩니다. 현재 데이터베이스에 대한 아카이브된 감사 로그만 추출할 수 있습니다. 코디네이터 노드에 표시되는 파일만 추출할 수 있습니다. 인스턴스 소유자만 아카이브된 감사 로그를 삭제할 수 있습니다.

SQL문 감사를 위한 EXECUTE 범주

EXECUTE 범주를 사용하여 사용자가 발행하는 SQL문을 정확히 추적할 수 있습니다. 버전 9.5에서는 이 정보를 찾는 데 CONTEXT 범주가 사용되었습니다.

이 EXECUTE 범주는 SQL문 텍스트, 컴파일 환경 및 나중에 명령문을 재생하는 데 필요한 기타 값을 캡처합니다. 예를 들어, 명령문을 재생하면 SELECT문이 리턴한 행을 정확하게 알 수 있습니다. 명령을 다시 실행하려면 먼저 데이터베이스 테이블을 명령문이 발행된 당시의 상태로 리스토어해야 합니다.

EXECUTE 범주를 사용하여 감사할 경우, 정적 및 동적 SQL에 대한 명령문 텍스트가 입력 매개변수 표시문자 및 호스트 변수로 기록됩니다. EXECUTE 범주는 입력 값을 사용하거나 사용하지 않고 감사하도록 구성할 수 있습니다.

주: 전역 변수는 감사하지 않습니다.

EXECUTE 이벤트의 감사는 이벤트가 완료될 때(SELECT문의 경우 커서가 닫힐 때) 수행됩니다. 이벤트 완료 상태도 저장됩니다. EXECUTE 이벤트는 완료 시에 감사되므로 오래 실행 중인 쿼리는 감사 로그에 즉시 나타나지 않습니다.

주: 명령문 준비는 실행에 포함되지 않습니다. 대부분의 권한 부여 검사는 준비 시 수행됩니다(예: SELECT 특권). 따라서 권한 부여 오류로 인해 준비 중에 실패한 명령문에서는 EXECUTE 이벤트가 생성되지 않습니다.

지정된 실행 레코드마다 명령문 값 인덱스, 명령문 값 유형 및 명령문 값 데이터 필드가 반복될 수 있습니다. 추출을 통해 생성된 보고서 형식의 경우, 각 레코드에 여러 개의 값이 나열됩니다. 구분된 파일 형식의 경우, 여러 개의 행이 사용됩니다. 첫 번째 행은 STATEMENT 이벤트 유형을 포함하여 값이 없습니다. 다음 행은 DATA 이벤트 유형을 포함하며, 각 행은 SQL문과 연관된 데이터 값을 포함합니다. 이벤트 상관자 및 응용프로그램 ID 필드에서 STATEMENT 및 DATA 행을 함께 링크할 수 있습니다. 명령문 텍스트, 명령문 분리 레벨 및 컴파일 환경 설명 컬럼은 DATA 이벤트에 표시되지 않습니다.

감사되는 명령문 텍스트 및 입력 데이터 값은 디스크에 저장될 때 데이터베이스 코드 페이지로 변환됩니다. 참고로, 감사되는 모든 필드는 데이터베이스 코드 페이지에 저장됩니다. 입력 데이터의 코드 페이지가 데이터베이스 코드 페이지와 호환되지 않더라도 오류가 리턴되지 않습니다. 대신 변환되지 않은 데이터가 로그됩니다. 각 데이터베이스는 고유한 감사 로그를 사용하므로 데이터베이스의 코드 페이지가 서로 달라도 문제가 발생하지 않습니다.

ROLLBACK 및 COMMIT은 응용 프로그램에 의해 실행되거나 다른 명령(예: BIND)의 일부로 내재적으로 발행된 경우에 감사됩니다.

감사되는 테이블에 대한 액세스로 인해 EXECUTE 이벤트가 감사된 후에는 작업 단위(UOW) 내에서 다른 명령문의 실행에 영향을 주는 모든 명령문이 감사됩니다. 이러한 명령문은 COMMIT, ROLLBACK, ROLLBACK TO SAVEPOINT 및 SAVEPOINT입니다.

세이브포인트 ID 필드

세이브포인트 ID 필드를 사용하여 ROLLBACK TO SAVEPOINT문의 영향을 받는 명령문을 추적할 수 있습니다. 일반 DML문(예: SELECT, INSERT 등)에서는 현재 세이브포인트 ID가 감사되지 않습니다. 그러나 ROLLBACK TO SAVEPOINT 문의 경

우 롤백되는 세이브포인트 ID가 대신 감사됩니다. 따라서 다음 예에서와 같이 세이브포인트 ID가 해당 ID보다 크거나 같은 모든 명령문이 롤백됩니다. 다음 표는 명령문 실행 순서를 보여줍니다. 세이브포인트 ID가 2보다 크거나 같은 모든 이벤트가 롤백됩니다. 값 3(첫 번째 INSERT문)만 T1 테이블에 삽입됩니다.

표 6. ROLLBACK TO SAVEPOINT문의 결과를 보여주기 위한 명령문의 순서

명령문	세이브포인트 ID
INSERT INTO T1 VALUES (3)	1
SAVEPOINT A	2
INSERT INTO T1 VALUES (5)	2
SAVEPOINT B	3
INSERT INTO T1 VALUES (6)	3
ROLLBACK TO SAVEPOINT A	2
COMMIT	

WITH DATA 옵션

WITH DATA 옵션을 지정하면 입력 값 중 일부만 감사됩니다. LOB, LONG, XML 및 구조화된 입력 매개변수는 NULL로 표시됩니다.

날짜, 시간 및 시간소인 필드는 ISO 형식으로 기록됩니다.

한 규정에는 WITH DATA가 지정되어 있고, SQL문 실행에 포함된 오브젝트와 연관된 다른 규정에는 WITHOUT DATA가 지정된 경우, WITH DATA가 우선적으로 적용되어 해당 특정 명령문에 대한 데이터가 감사됩니다. 예를 들어, 사용자와 연관된 감사 규정에는 WITHOUT DATA가 지정되어 있고 테이블과 연관된 규정에는 WITH DATA가 지정되어 있는 경우, 해당 사용자가 해당 테이블에 액세스하면 명령문에 사용된 입력이 감사됩니다.

위치가 지정된 UPDATE 또는 위치가 지정된 DELETE문에서 수정된 행은 판별할 수 없습니다. 기본 SELECT문 실행만 로그되며 개별 FETCH는 로그되지 않습니다. EXECUTE 레코드를 통해 명령문 발행 시 커서가 있던 행을 판별할 수 없습니다. 나중에 명령문을 재생할 때 SELECT문을 발행하여 영향을 받은 행의 범위만 알 수 있습니다.

이전 활동 재생 예

이 예에서는 종합적인 보안 규정의 일환으로, 회사에서 최대 7년 전으로 되돌아가서 데이터베이스의 특정 테이블에 대한 특정 요청의 결과를 분석할 수 있는 권한을 갖기를 원한다고 가정합니다. 이를 위해 회사에서는 선택한 기간 동안의 데이터베이스를 재구성할 수 있도록 주간 백업 및 연관된 로그 파일을 아카이브하는 규정을 시행합니다. 이때 데이터베이스 감사가 데이터베이스에 대해 이루어진 모든 요청에 대한 충분한 정보

를 캡처하여 관련된 리스토어 데이터베이스에 대한 모든 요청을 재생 및 분석할 수 있도록 해야 합니다. 이를 위해서는 정적 및 동적 SQL문이 모두 필요합니다.

다음 예는 SQL문 발행 시 시행되어야 하는 감사 규정 및 감사 로그를 아카이브하고 나중에 감사 로그를 추출 및 재생하는 단계를 보여줍니다.

1. EXECUTE 범주를 감사하는 감사 규정을 작성한 다음 이 규정을 데이터베이스에 적용합니다.

   ```
   CREATE AUDIT POLICY STATEMENTS CATEGORIES EXECUTE WITH DATA
      STATUS BOTH ERROR TYPE AUDIT
   COMMIT

   AUDIT DATABASE USING POLICY STATEMENTS
   COMMIT
   ```

2. 감사 로그를 정기적으로 아카이브하여 아카이브 사본을 작성합니다.

   SYSPROC.AUDIT_ARCHIVE 스토어드 프로시저에 대한 EXECUTE 특권이 부여된 사용자 또는 보안 관리자는 다음 명령문을 정기적(예: 로그된 데이터의 양에 따라 일주일에 한 번 또는 하루에 한 번)으로 실행해야 합니다. 이러한 아카이브된 파일은 필요한 기간 동안 보관할 수 있습니다. AUDIT_ARCHIVE 프로시저는 두 개의 입력 매개변수 즉, 아카이브 디렉토리에 대한 경로 및 -2(아카이브를 모든 노드에서 실행해야 함을 나타냄)를 사용하여 호출됩니다.

   ```
   CALL SYSPROC.AUDIT_ARCHIVE( '/auditarchive', -2 )
   ```

3. SYSPROC.AUDIT_LIST_LOGS 테이블 함수에 대한 EXECUTE 특권이 부여된 사용자 또는 보안 관리자는 AUDIT_LIST_LOGS를 통해 2006년 4월부터 사용 가능한 모든 감사 로그를 확인하여 필요한 데이터를 포함하고 있는 로그를 판별할 수 있습니다.

   ```
   SELECT FILE FROM TABLE(SYSPROC.AUDIT_LIST_LOGS('/auditarchive'))
      AS T WHERE FILE LIKE 'db2audit.dbname.log.0.200604%'
   FILE----------------------------------------
   ...
   db2audit.dbname.log.0.20060418235612
   db2audit.dbname.log.0.20060419234937
   db2audit.dbname.log.0.20060420235128
   ```

4. 이 출력을 통해 보안 관리자는 필요한 로그가 db2audit.dbname.log.20060419234937라는 한 파일에 있어야 한다는 것을 알 수 있습니다. 시간소인은 감사자가 확인하려고 하는 날짜의 마지막 날에 이 파일이 아카이브되었음을 나타냅니다.

   SYSPROC.AUDIT_DELIM_EXTRACT 스토어드 프로시저에 대한 EXECUTE 특권이 부여된 사용자 또는 보안 관리자는 이 파일 이름을 AUDIT_DELIM_EXTRACT에 대한 입력으로 사용하여 감사 데이터를 분리된 파일로 추출합니다. 이 파일의 감사 데이터는 DB2 데이터베이스 테이블에 로드할 수 있으며, 이 테이블을 분석하여 감사자가 원하는 특정 명령문을 찾을 수 있습니다.

감사자가 단일 SQL문에만 관심이 있다 하더라도 작업 단위(UOW)의 여러 명령문이 관심 있는 명령문에 영향을 준다면 이러한 명령문도 검사해야 합니다.

5. 명령문을 재생하려면 보안 관리자가 다음 조치를 취해야 합니다.
   - 감사 레코드에서 발행될 정확한 명령문을 판별합니다.
   - 감사 레코드에서 명령문을 발행한 사용자를 판별합니다.
   - 사용자가 명령문을 발행할 당시 사용자의 정확한 사용 권한(LBAC 보호 포함)을 재작성합니다.
   - 감사 레코드의 컴파일 환경 컬럼을 SET COMPILATION ENVIRONMENT문과 함께 사용하여 컴파일 환경을 재생합니다.
   - 명령문 발행 당시의 정확한 상태로 데이터베이스를 리스토어합니다.

   프로덕션 시스템을 방해하지 않으려면 다른 데이터베이스 시스템에서 데이터베이스 리스토어 및 명령문 재생을 수행해야 합니다. 명령문을 발행한 사용자로 실행 중인 보안 관리자는 명령문 값 데이터 요소에 제공된 입력 변수를 사용하여 명령문 텍스트에 있는 대로 명령문을 다시 발행할 수 있습니다.

감사 기능 관리

감사 기능 동작

이 주제는 로그에 감사 레코드를 기록하는 타이밍이 데이터베이스에 미치는 영향, 감사 기능 내에서 발생하는 오류를 관리하는 방법, 여러 상황에서 감사 레코드가 생성되는 방식에 대한 사용자의 이해를 돕기 위한 배경 정보를 제공합니다.

감사 레코드를 사용 중인 로그에 기록하는 타이밍 제어

활성 로그에 감사 레코드를 작성하는 것은 감사 레코드를 생성하는 이벤트의 발생과 함께 동기적으로 또는 비동기적으로 수행될 수 있습니다. *audit_buf_sz* 데이터베이스 관리 프로그램 구성 매개변수의 값은 감사 레코드가 수행되는 시기를 판별합니다.

*audit_buf_sz*의 값이 0일 경우 동기적으로 기록되며, 감사 레코드를 생성하는 이벤트는 레코드가 디스크에 기록될 때까지 대기합니다. 각 레코드와 연관된 대기는 DB2 데이터베이스의 성능을 저하시킵니다.

*audit_buf_sz*의 값이 0보다 크면, 레코드 작성은 비동기적으로 완료됩니다. *audit_buf_sz*의 값이 0보다 커지면 내부 버퍼를 작성하기 위해 몇 개의 4KB 페이지가 사용됩니다. 감사 레코드의 그룹을 디스크에 작성하기 전에 몇 개의 감사 레코드를 보존하기 위해 내부 버퍼가 사용됩니다. 감사 레코드를 감사 이벤트의 결과로서 생성하는 명령문은 레코드가 디스크에 작성될 때까지 기다리지 않고 조작을 계속할 수 있습니다.

비동기 경우에, 감사 레코드가 얼마 동안 채워지지 않은 버퍼에 남아 있을 수 있습니다. 이 상황이 확장된 기간 동안 발생하지 않게 하기 위해 데이터베이스 관리 프로그램

은 감사 레코드를 정기적으로 기록합니다. 감사 기능의 권한이 부여된 사용자는 명시적 요청으로 감사 버퍼를 비울 수 있습니다. 또한 아카이브 조작 중에는 버퍼가 자동으로 비워집니다.

동기 레코드 작성이 있는지 비동기 레코드 작성이 있는지 여부에 따라 오류 발생 시기가 다릅니다. 비동기 모드의 경우, 감사 레코드가 디스크에 기록되기 전에 버퍼 처리되기 때문에 몇몇 레코드가 유실될 수 있습니다. 동기 모드에서는 오류가 발생해도 많아야 하나의 감사 레코드만 쓰기 금지되기 때문에 하나의 레코드가 유실될 수 있습니다.

감사 기능 오류 관리

ERRORTYPE 감사 기능 매개변수 설정은 DB2 데이터베이스와 감사 기능 간에 오류가 관리되는 방식을 제어합니다. 감사 기능을 사용 중이고 ERRORTYPE 감사 기능 매개변수 설정이 AUDIT이면, 감사 기능은 DB2 데이터베이스의 또다른 파트와 동일한 방법으로 처리됩니다. 성공적이라고 간주되는 명령문과 연관된 감사 이벤트의 감사 레코드가 작성되어야 합니다(동기 모드에서 디스크로 또는 비동기 모드에서 감사 버퍼로). 이 모드를 실행할 때 오류가 발생할 때마다, 감사 레코드를 생성하는 명령문에 대해 음의 SQLCODE가 응용프로그램에 리턴됩니다.

오류 유형이 NORMAL로 설정되면, db2audit의 모든 오류는 무시되며 조작의 SQLCODE가 리턴됩니다.

다른 상황에서 생성되는 감사 레코드

API 또는 쿼리 명령문과 감사 설정에 따라, 하나 또는 여러 감사 레코드가 특정 이벤트에 대해 생성되거나 전혀 생성되지 않을 수 있습니다. 예를 들어, SELECT 서브쿼리가 있는 SQL UPDATE문은 테이블에서 UPDATE 특권에 대한 권한 부여 검사의 결과가 들어 있는 하나의 감사 레코드와 테이블에서 SELECT 특권에 대한 권한 부여 점검의 결과가 들어 있는 또다른 레코드의 결과를 가져올 수 있습니다.

동적 DML(Data Manipulation Language)문의 경우, 명령문이 준비되면 모든 권한 부여 검사에 대해 감사 레코드가 생성됩니다. 이때는 어떠한 감사 검사도 발생하지 않으므로 동일한 사용자에 의한 이들 명령문의 재사용은 다시 감사되지 않습니다. 그러나 특권 정보가 들어 있는 카탈로그 테이블 중 하나가 변경되면, 다음 작업 단위(UOW)에서 캐시된 동적 SQL 또는 XQuery문에 대한 명령문 특권이 다시 검사되며 하나 이상의 새로운 감사 레코드가 작성됩니다.

정적 DML문만 들어 있는 패키지의 경우, 감사 레코드를 생성할 수 있는 유일한 감사 가능한 이벤트는 사용자가 해당 패키지를 실행할 특권을 가지고 있는지 여부를 보기 위한 권한 부여 검사입니다. 패키지에서 정적 SQL 또는 XQuery문에 필요한 가능한 감사 레코드 작성과 권한 부여 검사는 패키지가 프리컴파일되거나 바인드될 때 수행됩니다. 패키지 내에서 실행되는 정적 SQL 또는 XQuery문은 EXECUTE 범주를 사용하

여 감사할 수 있습니다. 패키지가 사용자에 의해 명시적으로 또는 시스템에 의해 내재적으로 다시 바운드될 때, 정적 SQL 또는 XQuery문에 의해 필요한 권한 부여 검사에 대한 감사 레코드가 생성됩니다.

권한 부여 검사가 명령문 실행 시간에서 수행되는 명령문의 경우(예: DDL, GRANT 및 REVOKE문), 이들 명령문이 사용될 때마다 감사 레코드가 생성됩니다.

주: DDL을 실행할 때, 감사 레코드의 모든 이벤트(컨텍스트 이벤트 제외)에 기록되는 섹션 번호는 명령문의 실제 섹션 번호와 상관없이 0이 됩니다.

감사 기능 추가 정보 및 기술

감사 관리에 대한 우수 사례에는 감사 로그를 정기적으로 아카이브하고, 감사 규정을 작성할 때 오류 유형 AUDIT를 사용하는 방법 등이 있으며 기타 추가 정보는 아래에 설명되어 있습니다.

감사 로그 아카이브

감사 로그는 정기적으로 아카이브해야 합니다. 감사 로그를 아카이브하면 현재 감사 로그가 아카이브 디렉토리로 이동하며 서버는 새로운 활성 감사 로그에 정보를 기록합니다. 개별 아카이브 로그 파일의 이름에는 시간소인이 포함되어 있어 나중에 분석할 때 원하는 로그 파일을 쉽게 식별할 수 있습니다.

장기 보관을 위해 수신 파일 그룹을 압축해야 할 수도 있습니다.

아카이브된 감사 로그가 더 이상 필요하지 않을 경우 인스턴스 소유자가 운영 체제에서 해당 파일을 간단히 삭제할 수 있습니다.

오류 조절

감사 규정을 작성하는 경우 오류 유형 AUDIT를 사용해야 합니다. 단, 테스트 감사 규정을 작성하는 경우는 예외입니다. 예를 들어, 오류 유형이 AUDIT로 설정되어 있는 상태에서 디스크 공간 부족과 같은 오류가 발생하면 오류가 리턴됩니다. 먼저 오류 조건을 수정해야 감사 가능한 조치를 계속 수행할 수 있습니다. 그러나 오류 유형이 NORMAL로 설정된 경우에는 단순히 로깅만 실패하고 사용자에게 오류가 리턴되지는 않습니다. 따라서 오류가 발생하지 않은 것처럼 조작이 계속됩니다.

아카이브하는 동안 문제점이 발생하면(예: 아카이브 경로에 디스크 공간 부족 또는 아카이브 경로가 존재하지 않는 경우), 아카이브 프로세스에 실패하며 파일 확장자가 `.bk`인 중간 로그 파일이 감사 로그 데이터 경로에 생성됩니다(예: `db2audit.instance.log.0.20070508172043640941.bk`). 문제점이 해결되면(예: 아카이브 경로에 디스크 공간을 충분히 할당하거나 아카이브 경로 작성) 이 중간 로그를 아카이브 경로에 이동해야 합니다. 그러면 이 로그를 성공적으로 아카이브된 로그와 동일한 방법으로 사용할 수 있습니다.

DDL문 제한사항

AUDIT 독점 SQL문이라고 하는 일부 데이터 정의 언어(DDL)문은 다음 작업 단위(UOW) 때까지 적용되지 않습니다. 그러므로 이러한 명령문을 실행한 후에는 COMMIT 문을 즉시 사용하는 것이 좋습니다.

AUDIT 독점 SQL문은 다음과 같습니다.

- AUDIT
- CREATE AUDIT POLICY, ALTER AUDIT POLICY 및 DROP AUDIT POLICY
- DROP ROLE 및 DROP TRUSTED CONTEXT(삭제하려는 역할 또는 트러스트된 컨텍스트가 감사 규정과 연관된 경우)

아카이브된 데이터를 보관하기 위한 테이블의 형식은 변경될 수 있음

보안 관리자의 경우 SYSPROC.AUDIT_DEL_EXTRACT 스토어드 프로시저를 사용하여, 시스템 관리자의 경우 db2audit extract 명령을 사용하여 아카이브된 감사 로그 파일의 감사 레코드를 구분된 파일로 추출할 수 있습니다. 분석을 위해 구분된 파일의 감사 데이터를 DB2 데이터베이스 테이블로 로드할 수 있습니다. 감사 데이터 보관을 위해 작성해야 하는 테이블의 형식은 릴리스별로 다를 수 있습니다.

중요사항: db2audit.ddl 스크립트는 감사 레코드를 포함할 수 있는 올바른 형식의 테이블을 작성합니다. 따라서 릴리스에 따라 컬럼이 추가되거나 기존 컬럼의 크기가 변경될 수 있으므로 db2audit.ddl은 릴리스별로 실행해야 합니다.

CHECKING 이벤트 사용

CHECKING 이벤트로 작업할 때 대부분의 경우, 감사 레코드의 오브젝트 유형 필드는 필수 특권 또는 권한이 오브젝트에 액세스하려는 사용자 ID에 의해 보유되는지 여부를 알아보기 위해 검사되는 오브젝트입니다. 예를 들어, 사용자가 컬럼을 추가하여 테이블을 변경하려고 하는 경우, CHECKING 이벤트 감사 레코드는 시도된 액세스가 『ALTER』이고 검사 중인 오브젝트 유형이 『TABLE』(검사 대상이 테이블 특권이므로 컬럼이 아님)임을 나타냅니다.

그러나 사용자 ID가 오브젝트를 작성 또는 바인드하거나, 오브젝트를 삭제할 수 있도록 하는 데이터베이스 권한이 있는지 여부를 확인하는 것이 검사에 포함되면, 데이터베이스에 대한 검사가 있다고 하더라도, 오브젝트 유형 필드는 (데이터베이스 자체보다는) 작성, 바인드 또는 삭제되는 오브젝트를 지정합니다.

테이블에 인덱스를 작성하는 경우 인덱스 작성 특권이 필요하므로, CHECKING 이벤트 감사 레코드는 『작성』이 아닌 『index』 액세스 시도 유형을 갖습니다.

패키지 바인드용으로 작성된 감사 레코드

기존의 패키지를 바인딩할 때, OBJMAINT 이벤트 감사 레코드가 패키지의 DROP용으로 작성된 다음, 또 다른 OBJMAINT 이벤트 감사 레코드는 패키지의 새 사본의 CREATE용으로 작성됩니다.

ROLLBACK 후 CONTEXT 이벤트 정보 사용

데이터 정의 언어(DDL)는 성공적인 것으로 로그되는 OBJMAINT 또는 SECMAINT 이벤트를 생성할 수 있습니다. 그러나 이벤트 로깅 이후에 뒤따르는 오류가 ROLLBACK을 유발할 수 있습니다. 이것은 오브젝트를 작성하지 않은 채로 두거나, GRANT 또는 REVOKE 조치를 완료하지 않은 채로 둡니다. 이 경우, CONTEXT 이벤트의 사용이 중요합니다. 이러한 CONTEXT 이벤트 감사 레코드, 특히 이벤트를 종료하는 명령문은 시도된 조작의 완료 특성을 나타냅니다.

로드 분리문자

DB2 데이터베이스 관계형 테이블로 로드하는 데 적합한 분리된 형식으로 감사 레코드를 추출하는 경우, 명령문 텍스트 필드에서 사용되는 분리문자에 대해 명확히 알고 있어야 합니다. 이는 다음을 사용하여 분리된 파일을 추출하는 경우에 필요합니다.

```
db2audit extract delasc delimiter <load delimiter>
```

*load delimiter*는 단일 문자(예: ″)이거나 16진수 값(예: 『0xff』)을 나타내는 4바이트 문자열일 수 있습니다. 유효한 명령의 예는 다음과 같습니다.

```
   db2audit extract delasc
   db2audit extract delasc delimiter !
db2audit extract delasc delimiter 0xff
```

추출할 때 디폴트 로드 분리문자가 아닌 다른 문자를 분리문자로 사용한 경우, LOAD 명령에 MODIFIED BY 옵션을 사용해야 합니다. 분리문자로서 『0xff』가 사용된 LOAD 명령의 부분 예는 다음과 같습니다.

```
db2 load from context.del of del modified by chardel0xff replace into ...
```

이 예에서는 디폴트 로드 문자열 분리문자인 ″(큰따옴표)를 겹쳐씁니다.

제 2 장 역할

역할은 그룹과 동일한 기능을 제공하지만 동일한 제한사항이 적용되지 않으므로 특권 관리가 간편해집니다.

역할은 하나 이상의 특권이 그룹화된 데이터베이스 오브젝트로, GRANT문을 사용하여 사용자, 그룹, PUBLIC 또는 다른 역할에 지정하거나, CREATE TRUSTED CONTEXT 또는 ALTER TRUSTED CONTEXT문을 사용하여 트러스트된 컨텍스트에 지정할 수 있습니다. 워크로드 정의의 SESSION_USER ROLE 연결 속성에 역할을 지정할 수도 있습니다.

역할은 데이터베이스 시스템의 특권을 쉽게 관리할 수 있도록 해 주는 다음과 같은 몇 가지 이점을 제공합니다.

- 보안 관리자는 조직의 구조를 반영하는 방식으로 데이터베이스에 대한 액세스를 제어할 수 있습니다. (보안 관리자는 조직의 업무 기능에 직접 맵핑되는 역할을 데이터베이스에 작성할 수 있습니다.)
- 사용자에게는 자신의 업무 내용을 반영하는 역할의 멤버쉽이 부여되었습니다. 따라서 업무 내용이 변경되면 이에 따라 해당 역할의 멤버쉽을 손쉽게 부여 및 취소할 수 있습니다.
- 특권 지정이 간소화되었습니다. 동일한 특권 세트를 특정 업무 기능의 개별 사용자 각각에게 부여하는 대신, 관리자가 해당 업무 기능을 나타내는 역할에 이 특권 세트를 부여한 다음 해당 업무 기능의 각 사용자에게 해당 역할을 부여할 수 있습니다.
- 역할의 특권은 갱신할 수 있으며 해당 역할을 부여 받은 모든 사용자가 갱신사항을 수신할 수 있습니다. 관리자가 모든 사용자에 대한 특권을 개별적으로 갱신할 필요가 없습니다.
- 뷰, 트리거, 구체화된 쿼리 테이블(MQT), 정적 SQL 및 SQL 루틴을 작성할 때 역할에 부여된 특권과 권한은 항상 사용되는 반면, 그룹에 (직접 또는 간접적으로) 부여된 특권과 권한은 사용되지 않습니다.

  이는 그룹이 써드 파티 소프트웨어(예: 운영 체제 도는 LDAP 서버)에서 관리되므로 DB2 데이터베이스 시스템에서는 그룹의 멤버쉽이 변경되는 시기를 판별할 수 없기 때문입니다. 반면 역할은 데이터베이스 내에서 관리되므로 DB2 데이터베이스 시스템에서 권한 부여가 변경되는 시기를 판별하여 이에 따라 조치를 취할 수 있습니다. 같은 이유로 그룹에 부여된 역할은 고려되지 않습니다.
- 사용자가 연결을 설정하면 해당 사용자에게 지정된 모든 역할을 사용할 수 있으므로, 사용자 연결 시 역할에 부여된 모든 특권과 권한이 고려됩니다. 역할은 명시적으로 사용 또는 사용 안함으로 설정할 수 없습니다.
- 보안 관리자가 역할 관리를 다른 사용자에게 위임할 수 있습니다.

데이터베이스 내에서 부여할 수 있는 DB2 특권 및 권한은 모두 역할에 부여할 수 있습니다. 예를 들어, 역할을 다음 권한 및 특권 중 하나를 부여할 수 있습니다.

- DBADM, SECADM, DATAACCESS, ACCESSCTRL, SQLADM, WLMADM, LOAD 및 IMPLICIT_SCHEMA 데이터베이스 권한
- CONNECT, CREATETAB, CREATE_NOT_FENCED, BINDADD, CREATE_EXTERNAL_ROUTINE, 또는 QUIESCE_CONNECT 데이터베이스 권한
- 데이터베이스 오브젝트 특권(CONTROL 포함)

사용자가 데이터베이스에 연결되면 사용자의 역할이 자동으로 사용 가능해지며 권한 부여 대상으로 고려됩니다. SET ROLE문을 사용하여 역할을 활성화할 필요가 없습니다. 예를 들어, 뷰, 구체화된 쿼리 테이블(MQT), 트리거, 패키지 또는 SQL 루틴을 작성하는 경우 역할을 통해 부여 받은 특권이 적용됩니다. 그러나 사용자가 구성원으로 있는 그룹에 부여된 역할을 통해 부여 받은 특권은 적용되지 않습니다.

역할은 소유자가 없습니다. 따라서 다른 사용자가 역할 멤버쉽을 제어할 수 있도록 보안 관리자가 GRANT문의 WITH ADMIN OPTION절을 사용하여 역할 관리를 다른 사용자에게 위임할 수 있습니다.

제한사항

역할 사용과 관련하여 몇 가지 제한사항이 있습니다.

- 역할은 데이터베이스 오브젝트를 소유할 수 없습니다.
- 다음 데이터베이스 오브젝트를 작성하는 경우 그룹에 부여된 권한과 역할이 고려되지 않습니다.
  - 정적 SQL을 포함하는 패키지
  - 뷰
  - 구체화된 쿼리 테이블(MQT)
  - 트리거
  - SQL 루틴

  이러한 오브젝트를 작성하는 경우 해당 오브젝트를 작성하는 사용자 또는 PUBLIC에 직접 또는 간접적(예: 역할 계층을 통해)으로 부여된 역할만 고려됩니다.

역할 멤버쉽 작성 및 부여

보안 관리자는 역할을 작성, 삭제, 부여 및 취소하고 역할에 대한 주석을 표시할 수 있는 권한을 보유하고 있습니다. 역할 멤버쉽을 권한 부여 ID에 부여할 때는 GRANT(Role)문을 사용하고, 권한 부여 ID로부터 역할 멤버쉽을 취소할 때는 REVOKE(Role)문을 사용합니다.

보안 관리자는 역할 멤버쉽 관리를 권한 부여 ID에 위임할 수 있는데, WITH ADMIN OPTION을 사용하여 권한 부여 ID에 역할 멤버쉽을 부여하면 됩니다. GRANT(Role) 문의 WITH ADMIN OPTION절이 다른 사용자에게 부여하는 권한은 다음과 같습니다.

- 다른 사용자에게 역할 부여
- 다른 사용자의 역할 취소
- 역할에 대한 주석 표시

WITH ADMIN OPTION절이 제공하지 않는 권한은 다음과 같습니다.

- 역할 삭제
- 역할에 대한 WITH ADMIN OPTION을 다른 권한 부여 ID로부터 취소
- 다른 사용자에게 WITH ADMIN OPTION 부여(SECADM 권한을 보유하지 않은 경우)

보안 관리자가 역할을 작성한 후에는 데이터베이스 관리자가 GRANT문을 사용하여 권한 및 특권을 역할에 지정할 수 있습니다. 데이터베이스 내에서 부여할 수 있는 DB2 특권 및 권한은 모두 역할에 부여할 수 있습니다. 인스턴스 레벨 권한(예: SYSADM 권한)은 역할에 지정할 수 없습니다.

보안 관리자가 WITH ADMIN OPTION을 사용하여 역할 멤버쉽을 부여한 사용자 또는 보안 관리자는 GRANT(Role)문을 사용하여 해당 역할 멤버쉽을 다른 사용자, 그룹, PUBLIC 또는 역할에 부여할 수 있습니다. PUBLIC, 그룹 또는 역할을 통해 직접 또는 간접적으로 WITH ADMIN OPTION으로 역할 멤버쉽이 사용자에게 부여되었을 수 있습니다.

사용자가 세션을 설정하면 해당 사용자에게 지정된 모든 역할을 사용할 수 있습니다. DB2 데이터베이스 시스템에서 권한 부여를 검사하는 경우 사용자의 역할과 연관된 모든 특권과 권한이 고려됩니다. 일부 데이터베이스 시스템에서는 SET ROLE문을 사용하여 특정 역할을 활성화합니다. DB2 데이터베이스 시스템에서는 SET ROLE문을 사용하는 다른 제품과의 호환성을 위해 SET ROLE을 지원합니다. DB2 데이터베이스 시스템에서 SET ROLE문은 세션 사용자가 역할의 구성원인지 확인한 다음 아닐 경우 오류를 리턴합니다.

역할에 대한 WITH ADMIN OPTION 권한을 보유한 사용자 또는 보안 관리자는 REVOKE (Role)문을 사용하여 역할 멤버쉽을 사용자로부터 취소할 수 있습니다.

예

역할은 특정 특권 세트를 보유하고 있으며, 이 역할 멤버쉽이 부여된 사용자는 이 특권을 상속합니다. 이러한 특권 상속으로 인해 한 사용자의 특권을 다른 사용자에게 재지

정할 때 특권을 개별적으로 관리할 필요가 없어집니다. 역할 사용 시 유일하게 필요한 조작은 역할 멤버쉽을 한 사용자로부터 취소한 다음 해당 역할 멤버쉽을 다른 사용자에게 부여하는 것입니다.

예를 들어, DEV 부서에서 근무하는 BOB과 ALICE라는 직원은 SERVER, CLIENT 및 TOOLS 테이블에 대한 SELECT 특권을 보유하고 있습니다. 어느 날 경영진에서 이들을 새로운 QA 부서로 이전하기로 결정하여 데이터베이스 관리자가 SERVER, CLIENT 및 TOOLS 테이블에 대한 SELECT 특권을 취소해야 합니다.이후에 DEV 부서에서 TOM이라는 새로운 직원을 채용하여 데이터베이스 관리자는 SERVER, CLIENT 및 TOOLS 테이블에 대한 SELECT 특권을 TOM에게 부여해야 합니다.

역할 사용 시 발생하는 단계는 다음과 같습니다.

1. 보안 관리자가 DEVELOPER 역할을 작성합니다.

   ```
   CREATE ROLE DEVELOPER
   ```

2. 데이터베이스 관리자(DBADM 권한 보유)가 SERVER, CLIENT 및 TOOLS 테이블에 대한 SELECT 특권을 DEVELOPER 역할에 부여합니다.

   ```
   GRANT  SELECT ON TABLE SERVER TO ROLE DEVELOPER
   GRANT  SELECT ON TABLE CLIENT TO ROLE DEVELOPER
   GRANT  SELECT ON TABLE TOOLS TO ROLE DEVELOPER
   ```

3. 보안 관리자가 DEVELOPER 역할을 DEV 부서의 사용자인 BOB과 ALICE에게 부여합니다.

   ```
   GRANT ROLE DEVELOPER TO USER BOB, USER ALICE
   ```

4. BOB과 ALICE과 DEV 부서를 떠나면 보안 관리자는 사용자 BOB과 ALICE의 DEVELOPER 역할을 취소합니다.

   ```
   REVOKE ROLE DEVELOPER FROM USER BOB, USER ALICE
   ```

5. TOM이 DEV 부서에 채용된 경우 보안 관리자는 DEVELOPER 역할을 사용자 TOM에게 부여합니다.

   ```
   GRANT ROLE DEVELOPER TO USER TOM
   ```

역할 계층 구조

한 역할이 다른 역할의 멤버쉽에 부여되는 경우 역할 계층 구조가 생성됩니다.

첫 번째 역할에 다른 역할이 부여되는 경우 첫 번째 역할에 다른 역할이 *포함*된 상태가 됩니다. 따라서 다른 역할은 첫 번째 역할의 모든 특권을 상속하게 됩니다. 예를 들어, SURGEON 역할에 DOCTOR 역할이 부여된 경우 SURGEON이 DOCTOR을 포함하게 됩니다. 따라서 SURGEON은 DOCTOR 역할의 모든 특권을 상속하게 됩니다.

역할 계층 구조에서 순환은 허용되지 않습니다. 한 역할이 다른 역할에 부여되고 해당 다른 역할이 다시 원래 역할에 부여되는 순환 방식으로 역할이 부여된 경우에 *순환*이 발생합니다. 예를 들어, SURGEON 역할에 DOCTOR 역할이 부여된 다음 SURGEON

역할이 다시 DOCTOR 역할에 부여되는 경우가 이에 해당합니다. 역할 계층 구조에서 순환이 발생할 경우 오류가 리턴됩니다(SQLSTATE 428GF).

역할 계층 구조 작성 예

다음 예는 병원의 의사 레벨을 나타내는 역할 계층 구조를 작성하는 방식을 보여줍니다.

DOCTOR, SPECIALIST 및 SURGEON 역할이 있다고 가정합니다. 한 역할에 다른 역할을 부여하되, 순환은 발생하지 않는 방식으로 역할 계층 구조가 작성됩니다. SPECIALIST 역할에는 DOCTOR 역할이 부여되고, SURGEON 역할에는 SPECIALIST 역할이 부여됩니다.

SURGEON 역할을 DOCTOR 역할에 부여할 경우 순환이 발생하므로 이 방식은 허용되지 않습니다.

보안 관리자는 다음 SQL문을 실행하여 역할 계층 구조를 작성할 수 있습니다.

```
CREATE ROLE DOCTOR
CREATE ROLE SPECIALIST
CREATE ROLE SURGEON

GRANT ROLE DOCTOR TO ROLE SPECIALIST

GRANT ROLE SPECIALIST TO ROLE SURGEON
```

역할로부터 특권 취소 결과

특권을 취소하면 경우에 따라 뷰, 패키지 또는 트리거와 같은 종속 데이터베이스 오브젝트가 유효하지 않거나 작동하지 않을 수 있습니다.

다음 예제는 권한 부여 ID로부터 일부 특권은 취소하고 일부 특권은 역할 또는 다른 방법을 통해 보유하고 있는 경우, 데이터베이스 오브젝트에 대해 어떤 결과가 발생하는지를 보여줍니다.

역할로부터 특권 취소 예

1. 보안 관리자가 DEVELOPER 역할을 작성한 다음 이 역할의 멤버쉽을 사용자 BOB에게 부여합니다.

   ```
   CREATE ROLE DEVELOPER
   GRANT ROLE DEVELOPER TO USER BOB
   ```

2. 사용자 ALICE가 WORKITEM 테이블을 작성합니다.

   ```
   CREATE TABLE WORKITEM (x int)
   ```

3. 데이터베이스 관리자가 테이블에 대한 SELECT 및 INSERT 특권을 PUBLIC과 DEVELOPER 역할에 부여합니다.

```
GRANT SELECT ON TABLE ALICE.WORKITEM TO PUBLIC
GRANT INSERT ON TABLE ALICE.WORKITEM TO PUBLIC
GRANT SELECT ON TABLE ALICE.WORKITEM TO ROLE DEVELOPER
GRANT INSERT ON TABLE ALICE.WORKITEM TO ROLE DEVELOPER
```

4. 사용자 BOB이 WORKITEM 테이블을 사용하는 PROJECT 뷰와 WORKITEM 테이블에 종속된 PKG1 패키지를 작성합니다.

```
CREATE VIEW PROJECT AS SELECT * FROM ALICE.WORKITEM
PREP emb001.sqc BINDFILE PACKAGE USING PKG1 VERSION 1
```

5. 데이터베이스 관리자가 ALICE.WORKITEM 테이블에 대한 SELECT 특권을 PUBLIC으로부터 취소하더라도, 뷰 정의자 BOB이 여전히 DEVELOPER 역할의 멤버쉽을 통해 필요한 특권을 보유하고 있기 때문에 BOB.PROJECT 뷰는 작동 상태로 유지되고 PKG1 패키지도 유효한 상태로 유지됩니다.

```
REVOKE SELECT ON TABLE ALICE.WORKITEM FROM PUBLIC
```

6. 데이터베이스 관리자가 ALICE.WORKITEM 테이블에 대한 SELECT 특권을 DEVELOPER 역할로부터 취소할 경우, 뷰 정의자 BOB이 다른 방법을 통해 필요한 특권을 보유하고 있지 않기 때문에 BOB.PROJECT 뷰는 작동하지 않으며 PKG1 패키지도 유효하지 않은 상태로 유지됩니다.

```
REVOKE SELECT ON TABLE ALICE.WORKITEM FROM ROLE DEVELOPER
```

DBADM 권한 취소 예

이 예에서 DEVELOPER 역할은 DBADM 권한을 보유하고 있으며 사용자 BOB에게 부여되었습니다.

1. 보안 관리자가 DEVELOPER 역할을 작성합니다.

```
CREATE ROLE DEVELOPER
```

2. 시스템 관리자가 DBADM 권한을 DEVELOPER 역할에 부여합니다.

```
GRANT DBADM ON DATABASE TO ROLE DEVELOPER
```

3. 보안 관리자가 이 역할의 멤버쉽을 사용자 BOB에게 부여합니다.

```
GRANT ROLE DEVELOPER TO USER BOB
```

4. 사용자 ALICE가 WORKITEM 테이블을 작성합니다.

```
CREATE TABLE WORKITEM (x int)
```

5. 사용자 BOB이 WORKITEM 테이블을 사용하는 PROJECT 뷰, WORKITEM 테이블에 종속된 TRG1 패키지 및 WORKITEM 테이블에 종속된 TRG1 트리거를 작성합니다.

```
CREATE VIEW PROJECT AS SELECT * FROM ALICE.WORKITEM
PREP emb001.sqc BINDFILE PACKAGE USING PKG1 VERSION 1
CREATE TRIGGER TRG1 AFTER DELETE ON ALICE.WORKITEM
              FOR EACH STATEMENT MODE DB2SQL
       INSERT INTO ALICE.WORKITEM VALUES (1)
```

6. 보안 관리자가 사용자 BOB의 DEVELOPER 역할을 취소합니다.

```
REVOKE ROLE DEVELOPER FROM USER BOB
```

사용자 BOB의 DEVELOPER 역할을 취소하면 DBADM 권한을 보유하고 있던 역할이 취소되기 때문에 DBADM 권한이 손실됩니다. 뷰, 패키지 및 트리거도 영향을 받습니다.

- BOB.PROJECT 뷰는 여전히 유효합니다.
- PKG1 패키지는 유효하지 않습니다.
- BOB.TRG1 트리거는 여전히 유효합니다.

PKG1 패키지는 사용할 수 없지만, BOB.PROJECT 뷰 및 PKG1 패키지는 사용 가능합니다. DBADM 권한을 보유한 권한 부여 ID에 의해 작성된 뷰 및 트리거 오브젝트는 DBADM 권한이 손실되어도 영향을 받지 않습니다.

WITH ADMIN OPTION절을 사용하여 역할 유지보수 위임

보안 관리자는 GRANT (Role) SQL문의 WITH ADMIN OPTION절을 사용하여 역할 멤버쉽 관리 및 제어를 다른 사용자에게 위임할 수 있습니다.

WITH ADMIN OPTION절은 역할 멤버쉽을 다른 사용자에게 부여하고, 역할 멤버쉽을 역할의 다른 구성원으로부터 취소하며, 역할을 삭제하는 대신 역할에 대한 주석을 표시하는 권한을 다른 사용자에게 제공합니다.

그러나 WITH ADMIN OPTION절은 역할에 대한 WITH ADMIN OPTION을 다른 사용자에게 부여할 수 있는 권한을 다른 사용자에게 제공하지는 않습니다. 또한 역할에 대한 WITH ADMIN OPTION을 다른 권한 부여 ID로부터 취소할 수 있는 권한도 제공하지 않습니다.

WITH ADMIN OPTION절 사용 예

1. 보안 관리자가 DEVELOPER 역할을 작성한 다음 WITH ADMIN OPTION절을 사용하여 사용자 BOB에게 새 역할을 부여합니다.

   ```
   CREATE ROLE DEVELOPER
   GRANT ROLE DEVELOPER TO USER BOB WITH ADMIN OPTION
   ```

2. 사용자 BOB은 역할 멤버쉽을 다른 사용자(예: ALICE)에게 부여하거나 다른 사용자로부터 취소할 수 있습니다.

   ```
   GRANT ROLE DEVELOPER TO USER ALICE
   REVOKE ROLE DEVELOPER FROM USER ALICE
   ```

3. 사용자 BOB은 역할을 삭제하거나 WITH ADMIN OPTION을 다른 사용자에게 부여할 수 없습니다. 보안 관리자만 이러한 두 가지 조작을 수행할 수 있습니다. BOB이 발행하는 다음 명령은 실패합니다.

   ```
   DROP ROLE DEVELOPER  - FAILURE!
                        - only a security administrator is allowed to drop the role
   GRANT ROLE DEVELOPER TO USER ALICE WITH ADMIN OPTION - FAILURE!
                        - only a security administrator can grant WITH ADMIN OPTION
   ```

4. 사용자 BOB은 보안 관리자(SECADM) 권한을 보유하고 있지 않기 때문에 역할 관리 특권(WITH ADMIN OPTION에 의해 부여됨)을 DEVELOPER 역할의 사용자로부터 취소할 수 없습니다. 따라서 BOB이 다음 명령을 발행할 경우 실패합니다.

   ```
   REVOKE ADMIN OPTION FOR ROLE DEVELOPER FROM USER SANJAY - FAILURE!
   ```

5. 보안 관리자는 DEVELOPER 역할의 역할 관리 특권(WITH ADMIN OPTION에 의해 부여됨)을 사용자 BOB으로부터 취소할 수 있으며, 사용자 BOB에게는 아직 DEVELOPER 역할이 부여되어 있습니다.

   ```
   REVOKE ADMIN OPTION FOR ROLE DEVELOPER FROM USER BOB
   ```

   또는 보안 관리자가 사용자 BOB의 DEVELOPER 역할을 취소한 경우, BOB은 DEVELOPER 역할의 구성원이 부여한 특권과 WITH ADMIN OPTION절을 통해 받은 역할에 대한 권한을 모두 잃게 됩니다.

   ```
   REVOKE ROLE DEVELOPER FROM USER BOB
   ```

그룹 대 역할

뷰, 구체화된 쿼리 테이블(MQT), SQL 루틴, 트리거 및 정적 SQL을 포함하는 패키지를 작성하는 경우, 그룹에 부여된 특권과 권한은 고려되지 않습니다. 그룹 대신 역할을 사용하면 이 제한사항을 방지할 수 있습니다.

역할을 사용하면 사용자가 역할을 통해 획득한 특권을 사용하여 데이터베이스 오브젝트를 작성할 수 있습니다. 역할은 DB2 데이터베이스 시스템에서 제어하며, 그룹과 사용자는 DB2 데이터베이스 시스템 외부(예: 운영 체제 또는 LDAP 서버)에서 제어합니다.

그룹 대신 역할을 사용하는 예

이 예는 그룹 대신 역할을 사용하는 방법을 보여줍니다.

DEVELOPER_G, TESTER_G, SALES_G라는 세 개의 그룹이 있다고 가정합니다. 사용자 BOB, ALICE 및 TOM은 다음 표에서와 같이 이러한 그룹의 구성원입니다.

표 7. 사용자 및 그룹 예

그룹	이 그룹에 속한 사용자
DEVELOPER_G	BOB
TESTER_G	ALICE, TOM
SALES_G	ALICE, BOB

1. 보안 관리자가 그룹 대신 사용할 DEVELOPER, TESTER 및 SALES 역할을 작성합니다.

```
CREATE ROLE DEVELOPER
CREATE ROLE TESTER
CREATE ROLE SALES
```

2. 보안 관리자가 이 역할의 멤버쉽을 사용자에게 부여합니다. 그룹에 속한 사용자의 멤버쉽은 시스템 관리자가 설정할 수 있습니다.

```
GRANT  ROLE DEVELOPER TO USER BOB
GRANT  ROLE TESTER TO USER ALICE, USER TOM
GRANT  ROLE SALES TO USER BOB, USER ALICE
```

3. 데이터베이스 관리자가 그룹이 보유했던 것과 유사한 특권 또는 권한을 역할에 부여할 수 있습니다. 예를 들면 다음과 같습니다.

```
GRANT <privilege> ON <object> TO ROLE DEVELOPER
```

그런 다음 데이터베이스 관리자는 이러한 특권을 그룹에서 취소할 수 있으며, 시스템에서 해당 그룹을 제거해 줄 것을 시스템 관리자에게 요청할 수 있습니다.

역할을 통해 획득한 특권을 사용하여 트리거를 작성하는 예

이 예는 사용자 BOB이 DEVELOPER 역할을 통해 필요한 특권을 보유한 경우, 이 사용자가 트리거를 성공적으로 작성할 수 있음을 보여줍니다.

1. 먼저 사용자 ALICE가 WORKITEM 테이블을 작성합니다.

```
CREATE TABLE WORKITEM (x int)
```

2. 그런 다음 데이터베이스 관리자가 ALICE의 테이블을 변경할 수 있는 특권을 DEVELOPER 역할에 부여합니다.

```
GRANT ALTER ON ALICE.WORKITEM TO ROLE DEVELOPER
```

3. 사용자 BOB은 DEVELOPER 역할의 구성원이므로 트리거를 성공적으로 작성합니다.

```
CREATE TRIGGER TRG1 AFTER DELETE ON ALICE.WORKITEM
          FOR EACH STATEMENT MODE DB2SQL INSERT INTO ALICE.WORKITEM VALUES (1)
```

IBM Informix Dynamic Server에서 이주 후 역할 사용

IBM Informix® Dynamic Server에서 DB2 데이터베이스 시스템으로 이주한 후 역할 사용과 관련하여 유념해야 할 몇 가지 사항이 있습니다.

IDS(Informix Dynamic Server) SQL문인 GRANT ROLE은 WITH GRANT OPTION 절을 제공합니다. DB2 데이터베이스 시스템 GRANT ROLE문은 이와 동일한 기능의 WITH ADMIN OPTION절(SQL 표준을 준수함)을 제공합니다. IDS를 DB2 데이터베이스 시스템으로 이주하는 중 dbschema 도구에서 CREATE ROLE 및 GRANT ROLE문을 생성하면 발생한 WITH GRANT OPTION이 WITH ADMIN OPTION으로 대체됩니다.

IDS 데이터베이스 시스템에서 SET ROLE문은 특정 역할을 활성화합니다. DB2 데이터베이스 시스템에서는 해당 SQL문을 사용하는 다른 제품과의 호환성을 위해 SQL ROLE문을 지원합니다. SET ROLE문은 세션 사용자가 역할의 구성원인지 확인한 다음 아닐 경우 오류를 리턴합니다.

dbschema 출력 예

IDS 데이터베이스에 DEVELOPER, TESTER 및 SALES 역할이 포함되어 있다고 가정합니다. 사용자 BOB, ALICE 및 TOM에게는 서로 다른 역할이 부여되어 있습니다. 즉, DEVELOPER 역할은 BOB에게, TESTER 역할은 ALICE에게, TESTER 및 SALES 역할은 TOM에게 부여되어 있습니다. DB2 데이터베이스 시스템으로 이주하려면 dbschema 도구를 사용하여 데이터베이스에 대해 CREATE ROLE 및 GRANT ROLE문을 생성하십시오.

```
CREATE ROLE DEVELOPER
CREATE ROLE TESTER
CREATE ROLE SALES

GRANT  DEVELOPER TO BOB
GRANT  TESTER TO ALICE, TOM
GRANT  SALES TO TOM
```

DB2 데이터베이스 시스템에 데이터베이스를 작성한 다음 해당 데이터베이스에서 위의 명령문을 실행하여 역할 및 역할 지정을 재작성해야 합니다.

제 3 장 트러스트된 컨텍스트 및 트러스트된 연결 사용

DB2 데이터베이스와 연결이 설정되면 응용프로그램 내에서 요청을 수행하여 명시적으로 트러스트된 연결을 설정할 수 있습니다. 보안 관리자는 사용자가 설정하려는 연결의 속성과 일치하는 속성이 있는 CREATE TRUSTED CONTEXT문을 사용하여 트러스트된 컨텍스트를 미리 정의해야 합니다(후반부의 단계 1 참조).

연결을 설정할 때 명시적으로 트러스트된 연결을 요청하는 데 사용하는 API는 현재 사용하고 있는 응용프로그램의 유형에 따라 다릅니다(단계 2의 테이블 참조).

명시적으로 트러스트된 연결을 설정하면, 응용프로그램이 응용프로그램 유형에 맞는 API를 사용하여 해당 연결의 사용자 ID를 다른 사용자 ID로 전환할 수 있습니다(단계 3의 테이블 참조).

1. 보안 관리자는 CREATE TRUSTED CONTEXT문을 사용하여 서버에 트러스트된 컨텍스트를 정의합니다. 예를 들어, 다음과 같습니다.

```
CREATE TRUSTED CONTEXT MYTCX
       BASED UPON CONNECTION USING SYSTEM AUTHID NEWTON
       ATTRIBUTES (ADDRESS   '192.0.2.1')
       WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
       ENABLE
```

2. 트러스트된 연결을 설정하려면 응용프로그램에 다음 API 중 하나를 사용하십시오.

옵션	설명
응용프로그램	**API**
CLI/ODBC	SQLConnect, SQLSetConnectAttr
XA CLI/ODBC	Xa_open
JAVA	getDB2TrustedPooledConnection, getDB2TrustedXAConnection

3. 인증하거나 인증하지 않고 다른 사용자로 전환하려면 응용프로그램에 다음 API 중 하나를 사용하십시오.

옵션	설명
응용프로그램	**API**
CLI/ODBC	SQLSetConnectAttr
XA CLI/ODBC	SQLSetConnectAttr
JAVA	getDB2Connection, reuseDB2Connection
.NET	DB2Connection.ConnectionString 키워드: TrustedContextSystemUserID 및 TrustedContextSystemPassword

명시적으로 트러스트된 연결과 연관된 트러스트된 컨텍스트 오브젝트의 정의에 따라 새 사용자 ID를 인증하거나 인증하지 않고 전환을 수행할 수 있습니다. 예를 들어, 보안 관리자가 다음과 같은 트러스트된 컨텍스트 오브젝트를 작성한다고 가정해 봅시다.

```
CREATE TRUSTED CONTEXT CTX1
 BASED UPON CONNECTION USING SYSTEM AUTHID USER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR USER2 WITH AUTHENTICATION,
               USER3 WITHOUT AUTHENTICATION
  ENABLE
```

또한 명시적으로 트러스트된 연결이 설정되어 있다고 가정해 봅시다. 인증 정보를 제공하지 않고 트러스트된 연결에서 사용자 ID를 USER3으로 전환하려는 요청이 허용됩니다. USER3은 인증이 필요하지 않은 트러스트된 컨텍스트 CTX1의 사용자로 정의되어 있기 때문입니다. 하지만 인증 정보를 제공하지 않고 트러스트된 연결에서 사용자 ID를 USER2로 전환하려는 요청은 실패합니다. USER2는 인증 정보를 제공해야 하는 트러스트된 컨텍스트 CTX1의 사용자로 정의되어 있기 때문입니다.

명시적으로 트러스트된 연결 설정 및 사용자 ID 전환의 예

다음 예에서, 중간 티어 서버는 일반 사용자를 대신해서 몇 가지 데이터베이스 요청을 발행해야 하지만 일반 사용자 대신 데이터베이스 연결을 설정하기 위한 일반 사용자의 증명서에 액세스할 수 있는 권한이 없습니다.

데이터베이스 서버에 트러스트된 컨텍스트 오브젝트를 작성하여 중간 티어 서버가 데이터베이스에 명시적으로 트러스트된 연결을 설정하도록 할 수 있습니다. 명시적으로 트러스트된 연결이 설정되면, 중간 티어 서버는 데이터베이스 서버에서 새 사용자 ID를 인증하지 않고도 해당 연결의 현재 사용자 ID를 새 사용자 ID로 전환할 수 있습니다. 다음 CLI 코드 조각은 단계 1에 정의된 트러스트된 컨텍스트 MYTCX를 사용하여 트러스트된 연결을 설정하는 방법과, 트러스트된 연결에서 인증하지 않고 사용자를 전환하는 방법을 보여줍니다.

```
int main(int argc, char *argv[])
{
 SQLHANDLE henv;         /* environment handle */
 SQLHANDLE hdbc1;       /* connection handle */
 char origUserid[10] = "newton";
  char password[10] = "test";
   char switchUserid[10] = "zurbie";
 char dbName[10] = "testdb";

// Allocate the handles
SQLAllocHandle( SQL_HANDLE_ENV, &henv );
SQLAllocHandle( SQL_HANDLE_DBC, &hdbc1 );

// Set the trusted connection attribute
SQLSetConnectAttr( hdbc1, SQL_ATTR_USE_TRUSTED_CONTEXT,
```

```
SQL_TRUE, SQL_IS_INTEGER );

// Establish a trusted connection
SQLConnect( hdbc1, dbName, SQL_NTS, origUserid, SQL_NTS,
password, SQL_NTS );

//Perform some work under user ID "newton"
. . . . . .

 // Commit the work
SQLEndTran(SQL_HANDLE_DBC, hdbc1, SQL_COMMIT);


// Switch the user ID on the trusted connection
SQLSetConnectAttr( hdbc1,
SQL_ATTR_TRUSTED_CONTEXT_USERID, switchUserid,
SQL_IS_POINTER
);

//Perform new work using user ID "zurbie"
. . . . . .

//Commit the work
SQLEndTranSQL_HANDLE_DBC, hdbc1, SQL_COMMIT);

// Disconnect from database
SQLDisconnect( hdbc1 );

return 0;

}  /* end of main */
```

사용자 ID가 실제로 전환되는 시기

트러스트된 연결에서 사용자를 전환하는 명령이 발행되면, 다음 명령문이 서버에 전달된 후에 사용자 전환 요청이 수행됩니다. 아래 예에서는 다음 명령문이 발행될 때까지 list applications 명령에 원래 사용자 ID가 표시되어 있음을 알 수 있습니다.

1. USERID1을 사용하여 명시적으로 트러스트된 연결을 설정하십시오.
2. 사용자 전환 명령(예: USERID2에 getDB2Connection)을 발행하십시오.
3. `db2 list applications`를 실행하십시오. USERID1이 아직 연결되어 있다고 표시됩니다.
4. 트러스트된 연결에서 명령문(예: `executeQuery("values current sqlid")`)을 발행하여 서버에서 사용자 전환 요청을 수행하십시오.
5. `db2 list applications`를 다시 실행하십시오. 이제 USERID2가 연결되어 있다고 표시됩니다.

트러스트된 컨텍스트 및 트러스트된 연결

트러스트된 컨텍스트는 데이터베이스와 외부 엔티티(예: 응용프로그램 서버) 간 연결에 대한 트러스트 관계를 정의하는 데이터베이스 오브젝트입니다.

트러스트 관계는 다음과 같은 속성 세트를 기반으로 합니다.

- 시스템 권한 부여 ID: 데이터베이스 연결을 설정하는 사용자를 나타냅니다.
- IP 주소(또는 도메인 이름): 데이터베이스 연결이 설정되는 호스트를 나타냅니다.
- 데이터 스트림 암호화: 데이터베이스 서버와 데이터베이스 클라이언트 간의 데이터 통신에 대한 암호화 설정(있는 경우)을 나타냅니다.

사용자가 데이터베이스 연결을 설정하면 DB2 데이터베이스 시스템에서 연결이 데이터베이스에 있는 트러스트된 컨텍스트 오브젝트의 정의와 일치하는 확인합니다. 일치하는 경우 데이터베이스 연결이 트러스트된 연결로 간주됩니다.

이 트러스트된 연결의 개시자는 트러스트된 연결을 통해 트러스트된 연결 밖에서는 사용할 수 없는 추가 기능을 사용할 수 있습니다. 이러한 추가 기능은 트러스트된 연결이 명시적인지 아니면 내재적인지 여부에 따라 달라집니다.

명시적으로 트러스트된 연결의 개시자는 다음과 같은 기능을 수행할 수 있습니다.

- 인증을 사용하거나 사용하지 않고 연결의 현재 사용자 ID를 다른 사용자 ID로 전환
- 트러스트된 컨텍스트의 역할 상속 기능을 통해 추가 특권 획득

내재된 트러스트된 연결은 명시적으로 요청되지 않은 트러스트된 연결로서, 명시적으로 트러스트된 연결 요청이 아닌 일반 연결 요청으로 인해 발생합니다. 내재된 연결을 위해 응용프로그램 코드를 변경할 필요는 없습니다. 또한 내재된 트러스트된 연결의 설정 여부는 연결 리턴 코드에 영향을 주지 않습니다. 명시적으로 트러스트된 연결을 요청한 경우에는 연결 리턴 코드가 요청 성공 여부를 나타냅니다. 내재된 트러스트된 연결의 개시자는 트러스트된 컨텍스트의 역할 상속 기능을 통해서만 추가 기능을 얻을 수 있으며, 사용자 ID를 전환할 수는 없습니다.

트러스트된 컨텍스트를 통해 보안을 강화하는 방법

3계층 응용프로그램 모델은 2계층으로 구성된 표준 클라이언트 및 서버 모델을 확장한 것으로, 클라이언트 응용프로그램과 데이터베이스 서버 사이에 중간 계층이 배치되어 있습니다. 이 모델은 웹 기반 기술과 J2EE(Java 2 Enterprise Edition) 플랫폼이 부상하면서 최근에 특히 큰 인기를 얻고 있습니다. 3계층 응용프로그램 모델을 지원하는 소프트웨어 제품의 예로는 IBM WAS(WebSphere® Application Server)가 있습니다.

3계층 응용프로그램 모델에서 중간 계층은 클라이언트 응용프로그램을 실행하는 사용자를 인증하고, 데이터베이스 서버와의 상호 작용을 관리하는 역할을 담당합니다. 기본적으로 데이터베이스 서버와의 모든 상호 작용은 중간 계층과 데이터베이스 서버를 식별하는 증명서 및 사용자 ID 조합을 사용하여 중간 계층에서 설정하는 데이터베이스 연

결을 통해 발생합니다. 즉, 데이터베이스 서버는 모든 권한 부여 검사 및 감사를 수행할 때 중간 계층의 사용자 ID와 연관된 데이터베이스 특권을 사용합니다. 이때 권한 부여 검사 및 감사는 사용자 대신 중간 계층에서 수행하는 액세스를 비롯한 데이터베이스 액세스에 대해 발생해야 합니다.

3계층 응용프로그램 모델에는 여러 가지 이점이 있지만, 데이터베이스 서버와의 상호 작용(예: 사용자 요청)이 모두 중간 계층의 권한 부여 ID 하에서 발생하도록 할 경우 몇 가지 보안 위험이 발생할 수 있습니다. 이러한 위험은 다음과 같습니다.

- 사용자 ID 손실

  일부 엔터프라이즈에서는 액세스 제어를 위해 데이터베이스에 액세스하는 실제 사용자의 ID를 알고 싶어합니다.

- 사용자의 책임 감소

  감사를 통한 책임은 데이터베이스 보안의 기본 원칙입니다. 사용자 ID를 알지 못할 경우, 중간 계층에서 자체적인 목적을 위해 수행하는 트랜잭션과 사용자를 대신해서 수행하는 트랜잭션을 구분하기 힘듭니다.

- 중간 계층의 권한 부여 ID에 과도한 특권 부여

  중간 계층의 권한 부여 ID는 모든 사용자의 모든 요청을 실행하는 데 필요한 모든 특권을 가지고 있어야 합니다. 하지만 이로 인해 특정 정보에 액세스할 필요가 없는 사용자가 액세스 권한을 얻게 되는 보안 문제가 발생합니다.

- 보안 약화

  앞에서 지적한 특권 문제 이외에도, 현재 방법을 사용할 경우 사용자 요청에 의해 액세스될 수 있는 모든 자원에 대한 특권을 중간 계층에서 연결 시 사용하는 권한 부여 ID에 부여해야 합니다. 따라서 이러한 중간 계층의 권한 부여 ID가 침해될 경우 해당 자원이 모두 위험에 노출됩니다.

- 동일한 연결을 사용하는 사용자 과잉

  이전 사용자가 변경한 내용이 현재 사용자에게 영향을 미칠 수 있습니다.

따라서 사용자를 대신하여 중간 계층에서 수행하는 데이터베이스 요청에 대해 실제 사용자의 ID와 데이터베이스 특권을 사용하는 메커니즘이 필요합니다. 이를 충족할 수 있는 가장 확실한 방법은 중간 계층에서 사용자의 ID와 암호를 사용하여 새 연결을 설정한 다음 이 연결을 통해 사용자의 요청을 전달하는 것입니다. 간단하기는 하지만 이 방법에는 다음과 같은 몇 가지 단점이 있습니다.

- 특정 중간 계층에는 적용할 수 없을 가능성. 대부분의 중간 계층 서버는 연결에 필요한 사용자 인증 증명서를 가지고 있지 않습니다.
- 성능 오버헤드. 실제 연결을 새로 작성하고 데이터베이스 서버에서 재인증하는 것과 관련하여 명백한 성능 오버헤드가 발생합니다.

- 유지보수 오버헤드. 중앙 집중식 보안이 설정되어 있지 않거나 단일 로그온을 사용하지 않는 경우, 두 개의 사용자 정의(하나는 중간 계층에 다른 하나는 서버에 생김)가 생기므로 유지보수 오버헤드가 발생합니다. 이 경우 다른 위치에서 암호를 변경해야 합니다.

트러스트된 컨텍스트 기능은 이 문제점을 해결합니다. 보안 관리자는 데이터베이스와 중간 계층 간의 트러스트 관계를 정의하는 트러스트된 컨텍스트 오브젝트를 데이터베이스에 작성할 수 있습니다. 그러면 중간 계층에서는 데이터베이스에 대한 명시적으로 트러스트된 연결을 설정할 수 있습니다. 이 경우 중간 계층에서는 인증을 사용하거나 사용하지 않고 연결의 현재 사용자 ID를 다른 사용자 ID로 전환할 수 있습니다. 트러스트된 컨텍스트는 일반 사용자의 ID 확인 문제점을 해결할 뿐 아니라 다른 이점도 제공합니다. 데이터베이스 사용자가 특정 특권을 사용할 수 있게 되는 시기를 제어할 수 있다는 것이 이러한 이점에 해당합니다. 사용자가 특권을 사용할 수 있게 되는 시기를 제어할 수 없는 경우 전반적인 보안이 약화됩니다. 예를 들어, 원래 의도된 목적이 아닌 다른 목적에 특권이 사용될 수 있습니다. 보안 관리자는 하나 이상의 특권을 한 역할에 지정한 다음 해당 역할을 트러스트된 컨텍스트 오브젝트에 지정할 수 있습니다. 따라서 트러스트된 컨텍스트의 정의와 일치하는 트러스트된 데이터베이스 연결(명시적 또는 내재적)만 해당 역할과 연관된 특권을 사용할 수 있습니다.

성능 향상

트러스트된 연결을 사용할 경우 다음과 같은 이점을 통해 성능이 극대화될 수 있습니다.

- 연결의 현재 사용자 ID가 전환될 때 새로운 연결이 설정되지 않습니다.
- 트러스트된 컨텍스트 정의에서 전환할 사용자 ID의 인증을 요구하지 않는 경우, 데이터베이스 서버에서 새로운 사용자를 인증하는 것과 관련된 오버헤드가 발생하지 않습니다.

트러스트된 컨텍스트 작성 예

보안 관리자가 다음과 같은 트러스트된 컨텍스트 오브젝트를 작성한다고 가정합니다.

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID USER2
  ATTRIBUTES (ADDRESS   '192.0.2.1')
  DEFAULT ROLE managerRole
   ENABLE
```

사용자 *user1*이 IP 주소 192.0.2.1에서 트러스트된 연결을 요청하는 경우, DB2 데이터베이스 시스템에서는 트러스트된 연결을 설정할 수 없으며 사용자 *user1*은 트러스트되지 않은 연결만 얻을 수 있음을 나타내는 경고(SQLSTATE 01679, SQLCODE +20360)를 리턴합니다. 그러나 사용자 *user2*가 IP 주소 192.0.2.1에서 트러스트된 연결을 요청하는 경우 연결 속성이 트러스트된 컨텍스트 CTX1에 의해 충족되므로 요청이 수행됩니다. 이제 사용자 *user2*는 트러스트된 연결을 설정했으므로 트러스트된 컨텍

스트 역할 managerRole과 연관된 특권과 권한을 모두 얻을 수 있습니다. 그러나 이 트러스트된 연결 범위밖에서는 사용자 *user2*가 이러한 특권과 권한을 사용하지 못할 수 있습니다.

트러스트된 컨텍스트를 통한 역할 멤버쉽 상속

관련 트러스트된 컨텍스트 정의의 일부로 보안 관리자에 의해 지정된 경우, 트러스트된 연결의 현재 사용자는 트러스트된 컨텍스트를 통한 역할 자동 상속을 통해 추가 특권을 얻을 수 있습니다.

디폴트로 트러스트된 연결의 모든 사용자는 역할을 상속할 수 있습니다. 보안 관리자는 또한 트러스트된 컨텍스트 정의를 사용하여 특정 사용자가 상속할 역할을 지정할 수 있습니다.

트러스트된 연결 중 세션 권한 부여 ID가 보유할 수 있는 활성 역할은 다음과 같습니다.

- 세션 권한 부여 ID가 일반적으로 구성원으로 간주되는 역할
- 트러스트된 컨텍스트 디폴트 역할 또는 트러스트된 컨텍스트 사용자별 역할(정의된 경우)

주:

- 연결 성공 시 사용자 정의 보안 플러그인에서 생성하는 시스템 권한 부여 ID 및 세션 권한 부여 ID가 서로 다르도록 작성된 사용자 정의 보안 플러그인을 사용하여 사용자 인증을 구성하는 경우, 트러스트된 연결이라 하더라도 해당 연결을 통해 트러스트된 컨텍스트 역할을 상속할 수 없습니다.
- 역할을 통해 얻은 트러스트된 컨텍스트 특권은 동적 DML 조작에 대해서만 유효하며, 다음에 대해서는 유효하지 않습니다.
  - DDL 조작
  - 정적 SQL(BIND, REBIND, 내재된 리바인드, 증분식 바인드 등 정적 SQL을 포함하는 조작)

트러스트된 컨텍스트 사용별 특권 획득

보안 관리자는 트러스트된 컨텍스트 정의를 사용하여 다음 작업이 가능하도록 역할을 트러스트된 컨텍스트와 연관시킬 수 있습니다.

- 트러스트된 연결의 모든 사용자가 디폴트로 지정된 역할을 상속할 수 있음
- 트러스트된 연결의 특정 사용자가 지정된 역할을 상속할 수 있음

트러스트된 연결의 사용자가 새 권한 부여 ID로 전환되었으며 이 새 권한 부여 ID에 대한 트러스트된 컨텍스트 사용자별 역할이 있는 경우, 아래 예에서와 같이 사용자별 역할이 트러스트된 컨텍스트 디폴트 역할을 대체합니다(있는 경우).

디폴트 역할 및 사용자별 역할을 지정하는 트러스트된 컨텍스트 작성 예

보안 관리자가 다음과 같은 트러스트된 컨텍스트 오브젝트를 작성한다고 가정합니다.

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID USER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR USER2 WITH AUTHENTICATION,
               USER3 WITHOUT AUTHENTICATION
  DEFAULT ROLE AUDITOR
  ENABLE
```

USER1이 트러스트된 연결을 설정하면 이 권한 부여 ID가 AUDITOR 역할에 부여된 특권을 상속합니다. 마찬가지로, 트러스트된 연결의 현재 권한 부여 ID가 자신의 사용자 ID로 전환된 경우 이와 동일한 특권을 USER3이 상속합니다. 특정 지점에서 연결의 사용자 ID가 USER2로 전환된 경우, USER2도 트러스트된 컨텍스트 디폴트 역할인 AUDITOR를 상속합니다. 보안 관리자는 USER3가 트러스트된 컨텍스트 디폴트 역할이 아닌 다른 역할을 상속하도록 선택할 수도 있습니다. 또한 다음과 같이 특정 역할을 이 사용자에게 지정하는 방식으로 다른 역할을 상속하도록 할 수 있습니다.

```
CREATE TRUSTED CONTEXT CTX1
 BASED UPON CONNECTION USING SYSTEM AUTHID USER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR USER2 WITH AUTHENTICATION,
               USER3 WITHOUT AUTHENTICATION ROLE OTHER_ROLE
  DEFAULT ROLE AUDITOR
  ENABLE
```

트러스트된 연결의 현재 사용자 ID가 USER3로 전환되면 이 사용자는 더 이상 트러스트된 컨텍스트 디폴트 역할을 상속하지 않습니다. 대신 보안 관리자가 자신에게 지정한 특정 역할인 OTHER_ROLE을 상속합니다.

명시적으로 트러스트된 연결의 사용자 ID 전환을 위한 규칙

명시적으로 트러스트된 연결에서는 연결의 사용자 ID를 다른 사용자 ID로 전환할 수 있습니다. 이때 특정한 규칙이 적용됩니다.

1. 전환 요청이 명시적으로 트러스트된 연결에서 이루어지지 않았는데 전환 요청이 처리를 위해 서버로 전송된 경우, 연결이 종료되고 오류 메시지가 리턴됩니다 (SQLSTATE 08001, SQLCODE -30082, 이유 코드 41).
2. 전환 요청이 트랜잭션 경계에서 이루어지지 않았는데 트랜잭션이 롤백되고 전환 요청이 처리를 위해 서버로 전송된 경우, 연결이 미연결 상태로 전환되면서 오류 메시지가 리턴됩니다(SQLSTATE 58009, SQLCODE -30020).
3. 전환 요청이 스토어드 프로시저 내에서 이루어진 경우, 이 환경에서 부적합한 조작임을 나타내는 오류 메시지가 리턴됩니다(SQLCODE -30090, 이유 코드 29). 연결 상태는 유지되므로 연결이 미연결 상태로 전환되지는 않습니다. 이후 요청은 처리될 수 있습니다.

4. 전환 요청이 (데이터베이스 연결이 아닌) 인스턴스 접속의 서버로 전달된 경우, 접속이 종료되고 오류 메시지가 리턴됩니다(SQLCODE -30005).
5. 트러스트된 연결에서 허용되지 않는 권한 부여 ID를 사용하여 전환 요청이 이루어진 경우, 오류(SQLSTATE 42517, SQLCODE -20361)가 리턴되면서 연결이 미연결 상태로 전환됩니다.
6. 트러스트된 연결 WITH AUTHENTICATION에서 허용되는 권한 부여 ID를 사용하여 전환 요청이 이루어졌으나, 적합한 인증 토큰이 제공되지 않은 경우, 오류(SQLSTATE 42517, SQLCODE -20361)가 리턴되면서 연결이 미연결 상태로 전환됩니다.
7. 트러스트된 연결과 연관된 트러스트된 컨텍스트 오브젝트를 사용할 수 없는데 해당 트러스트된 연결에 대한 전환 요청이 이루어진 경우, 오류(SQLSTATE 42517, SQLCODE -20361)가 리턴되면서 연결이 미연결 상태로 전환됩니다.

   이 경우, 트러스트된 연결을 설정한 사용자 ID 또는 NULL 사용자 ID가 지정된 사용자 전환 요청만 허용됩니다. 트러스트된 연결을 설정한 사용자 ID에 대한 전환이 이루어진 경우, 이 사용자 ID는 어떠한 트러스트된 컨텍스트 역할도 상속하지 않습니다. 트러스트된 컨텍스트 디폴트 역할과 트러스트된 컨텍스트 사용자별 역할도 상속하지 않습니다.
8. 트러스트된 연결과 연관된 트러스트된 컨텍스트 오브젝트의 시스템 권한 부여 ID 속성을 사용할 수 없는데 해당 트러스트된 연결에 대한 전환 요청이 이루어진 경우, 오류(SQLSTATE 42517, SQLCODE -20361)가 리턴되면서 연결이 미연결 상태로 전환됩니다.

   이 경우, 트러스트된 연결을 설정한 사용자 ID 또는 NULL 사용자 ID가 지정된 사용자 전환 요청만 허용됩니다. 트러스트된 연결을 설정한 사용자 ID에 대한 전환이 이루어진 경우, 이 사용자 ID는 어떠한 트러스트된 컨텍스트 역할도 상속하지 않습니다. 트러스트된 컨텍스트 디폴트 역할과 트러스트된 컨텍스트 사용자별 역할도 상속하지 않습니다.
9. 트러스트된 연결과 연관된 트러스트된 컨텍스트 오브젝트가 삭제되었는데 해당 트러스트된 연결에 대한 전환 요청이 이루어진 경우, 오류(SQLSTATE 42517, SQLCODE -20361)가 리턴되면서 연결이 미연결 상태로 전환됩니다.

   이 경우, 트러스트된 연결을 설정한 사용자 ID 또는 NULL 사용자 ID가 지정된 사용자 전환 요청만 허용됩니다. 트러스트된 연결을 설정한 사용자 ID에 대한 전환이 이루어진 경우, 이 사용자 ID는 어떠한 트러스트된 컨텍스트 역할도 상속하지 않습니다. 트러스트된 컨텍스트 디폴트 역할과 트러스트된 컨텍스트 사용자별 역할도 상속하지 않습니다.

10. 트러스트된 연결에서 허용되는 사용자 ID를 사용하여 전환 요청이 이루어졌는데 해당 사용자 ID가 데이터베이스에 대한 CONNECT 특권을 보유하고 있지 않은 경우, 연결이 미연결 상태로 전환되면서 오류 메시지가 리턴됩니다(SQLSTATE 08004, SQLCODE -1060).
11. 트러스트된 컨텍스트 시스템 권한 부여 ID가 WITH USE FOR절에 표시되는 경우, DB2 데이터베이스 시스템에서는 사용자 전환 요청 시 시스템 권한 부여 ID에 대한 인증 설정을 그대로 적용하여 시스템 권한 부여 ID로 다시 전환합니다. 트러스트된 컨텍스트 시스템 권한 부여 ID가 WITH USE FOR절에 표시되지 않는 경우, 인증을 사용하지 않아도 시스템 권한 부여 ID로 다시 전환하기 위한 사용자 전환 요청이 항상 허용됩니다.

주: 연결이 미연결 상태로 전환된 경우, "응용프로그램 상태에 오류가 있습니다. 데이터베이스 연결이 없습니다."라는 오류(SQLCODE -900)가 리턴되지 않는 허용되는 요청은 다음과 같습니다.

- 사용자 전환 요청
- COMMIT 또는 ROLLBACK문
- DISCONNECT, CONNECT RESET 또는 CONNECT 요청

주: 트러스트된 연결의 사용자 ID가 새로운 사용자 ID로 전환될 경우, 이전 사용자가 사용했던 연결 환경에 대한 모든 추적사항이 제거됩니다. 즉, 사용자 ID를 전환하면 새 연결 환경과 동일한 환경이 제공됩니다. 예를 들어, 연결의 이전 사용자 ID가 임시 테이블 또는 WITH HOLD 커서를 열어 둔 경우, 해당 연결의 사용자 ID가 새로운 사용자 ID로 전환될 때 이러한 오브젝트가 완전히 손실됩니다.

트러스트된 컨텍스트 문제점 판별

명시적으로 트러스트된 연결은 구체적이고 명시적인 트러스트된 연결 요청에 의해 설정되는 연결입니다. 명시적으로 트러스트된 연결을 요청할 권한이 없는 사용자가 이를 요청하면 일반 연결이 설정되며 경고(+20360)가 표시됩니다.

사용자가 트러스트된 연결을 설정할 수 없는 이유를 판별하려면, 보안 관리자가 시스템 카탈로그와 연결 속성에서 트러스트된 컨텍스트 정의를 찾아야 합니다. 특히, 연결이 설정되는 IP 주소, 데이터 스트림 또는 네트워크의 암호화 레벨, 연결을 설정하는 시스템 권한 부여 ID를 찾아야 합니다. db2pd 유틸리티의 -application 옵션은 이 정보와 함께 다음과 같은 추가적인 정보를 리턴합니다.

- 연결 트러스트 유형: 연결이 트러스트되어 있는지 여부를 나타냅니다. 연결이 트러스트된 경우, 이 유형은 명시적으로 트러스트된 연결인지 내재된 트러스트된 연결인지 나타냅니다.
- 트러스트된 컨텍스트 이름: 트러스트된 연결과 연관된 트러스트된 컨텍스트의 이름입니다.

- 상속된 역할: 트러스트된 연결을 통해 상속된 역할입니다.

다음은 명시적으로 트러스트된 연결 설정에 실패하는 가장 일반적인 원인입니다.

- 클라이언트 응용프로그램이 DB2 서버와 통신하는 데 TCP/IP를 사용하고 있지 않습니다. TCP/IP는 클라이언트 응용프로그램이 트러스트된 연결(명시적 또는 내재적)을 설정하는 데 사용할 수 있는 DB2 서버와 통신할 때 지원되는 유일한 프로토콜입니다.
- 데이터베이스 서버 인증 유형이 CLIENT로 설정되어 있지 않습니다.
- 데이터베이스 서버에서 트러스트된 컨텍스트 오브젝트를 사용할 수 없습니다. 트러스트된 컨텍스트가 수신 연결의 속성과 일치하다고 간주되려면 트러스트된 컨텍스트 오브젝트의 정의가 명시적으로 ENABLE을 나타내야 합니다.
- 데이터베이스 서버의 트러스트된 컨텍스트 오브젝트가 표시된 트러스트 속성과 일치하지 않습니다. 예를 들어, 다음 상황 중 하나가 적용될 수 있습니다.
  - 연결의 시스템 권한 부여 ID가 트러스트된 컨텍스트 오브젝트 시스템 권한 부여 ID와 일치하지 않습니다.
  - 연결이 시작된 IP 주소가 연결에 고려되는 트러스트된 컨텍스트 오브젝트의 IP 주소와 일치하지 않습니다.
  - 연결에 사용된 데이터 스트림 암호화가 연결에 고려되는 트러스트된 컨텍스트 오브젝트의 ENCRYPTION 속성 값과 일치하지 않습니다.

  db2pd 도구를 사용하여 연결이 설정되는 IP 주소, 연결에 사용된 데이터 스트림 또는 네트워크의 암호화 레벨, 연결을 설정하는 시스템 권한 부여 ID를 찾을 수 있습니다. SYSCAT.CONTEXTS 및 SYSCAT.CONTEXTATTRIBUTES 카탈로그 뷰를 참조하여 특정 트러스트된 컨텍스트 오브젝트의 정의(예: 시스템 권한 부여 ID, 허용되는 IP 주소 세트, ENCRYPTION 속성 값)를 찾을 수 있습니다.

다음은 사용자 전환에 실패하는 가장 일반적인 원인입니다.

- 전환할 사용자 ID가 데이터베이스에서 CONNECT 특권을 가지고 있지 않습니다. 이 경우, SQL1060N이 리턴됩니다.
- 명시적으로 트러스트된 연결과 연관된 트러스트된 컨텍스트 오브젝트의 WITH USE FOR 절에 전환할 사용자 ID 또는 PUBLIC이 정의되어 있지 않습니다.
- 인증이 있어야 사용자를 전환할 수 있는데, 해당 사용자에게 증명서가 없거나 잘못된 증명서가 있습니다.
- 트랜잭션 경계에서 사용자 전환이 요청되지 않았습니다.
- 트러스트된 연결과 연관된 트러스트된 컨텍스트가 사용 불가능, 삭제 또는 변경되었습니다. 이 경우, 트러스트된 연결을 설정한 사용자 ID만 전환할 수 있습니다.

제 4 장 레이블 기반 액세스 제어(LBAC)

레이블 기반 액세스 제어(LBAC)를 사용하면 데이터에 액세스할 수 있는 사용자에 대한 제어 능력이 대폭 증가됩니다. LBAC를 사용하면 쓰기 액세스 권한을 갖고 있는 사용자와 개별 행 및 개별 컬럼에 대해 읽기 액세스 권한을 갖고 있는 사용자를 정확히 판별할 수 있습니다.

LBAC가 수행하는 작업

LBAC 기능은 쉽게 구성할 수 있으며 특정 보안 환경에 알맞게 조정할 수 있습니다. 모든 LBAC 구성은 *보안 관리자*에 의해 수행되는데, 보안 관리자는 SECADM 권한이 부여된 사용자입니다.

보안 관리자는 보안 레이블 구성요소를 작성하여 LBAC 시스템을 구성합니다. *보안 레이블 구성요소*란 사용자가 데이터에 액세스해야 하는지 판별하는 데 사용할 기준을 나타내는 데이터베이스 오브젝트입니다. 예를 들어, 사용자가 특정 부서에 소속되어 있는지 여부 또는 사용자가 특정 프로젝트를 수행하고 있는지 여부가 기준이 될 수 있습니다. *보안 규정*은 어떤 데이터에 대해 액세스 권한을 갖고 있는 사용자를 판별하는 데 사용되는 기준에 대해 기술합니다. 보안 규정은 하나 이상의 보안 레이블 구성요소를 포함합니다. 임의의 한 테이블을 보호하는 데는 하나의 보안 테이블만 사용할 수 있으나 다른 테이블은 다른 보안 규정을 사용하여 보호할 수 있습니다.

보안 규정을 작성한 후, 보안 관리자는 해당 규정의 일부인 *보안 레이블*이라는 오브젝트를 작성합니다. 보안 레이블은 보안 레이블 구성요소를 포함합니다. 보안 레이블을 구성하는 정확한 내용은 보안 규정에 의해 결정되며 조직에서 특정 데이터 항목에 대한 액세스 권한이 있어야 한다고 판단하는 데 사용되는 기준을 표시하도록 구성될 수 있습니다. 예를 들어, 회사에서 직원의 직위와 그들이 참여한 프로젝트를 찾아 해당 직원이 보아야 하는 데이터를 결정하기로 결정한 경우, 각 레이블에 해당 정보가 포함되도록 보안 레이블을 구성할 수 있습니다. LBAC는 매우 복잡한 기준에서 매우 간단한 시스템(각 레벨은 "높은" 또는 "낮음" 신뢰도를 표시함)에 이르기까지 모든 것을 설정할 수 있을 정도로 매우 유연합니다.

작성되면, 보안 레벨을 개별 테이블의 행 및 컬럼에 연관시켜 여기에 데이터가 보류되는 것을 방지할 수 있습니다. 보안 레이블에 의해 보호되는 데이터를 *보호 데이터*라고 합니다. 보안 관리자는 사용자에게 보안 레이블을 부여함으로써 사용자가 보호 데이터에 액세스하게 할 수 있습니다. 사용자가 보호 데이터에 액세스를 시도하면 사용자의 보안 레이블을 데이터를 보호하는 보안 레이블과 비교합니다. 보호 레이블은 일부 보안 레이블은 차단하고 일부 레이블은 차단하지 않습니다.

사용자, 역할 또는 그룹은 한 번에 여러 보안 규정에 대한 보안 레이블을 보유할 수 있습니다. 그러나 주어진 보안 규정에 대해서는 읽기 액세스와 쓰기 액세스당 하나의 레이블만 보유할 수 있습니다.

보안 관리자는 사용자에게 면제 권한도 부여할 수 있습니다. *면제* 권한을 통해 사용자의 보안 레이블에서 액세스를 허용하지 않는 보호 데이터에 액세스할 수 있습니다. 보안 레이블과 면제를 모두 *LBAC 증명서*라고 합니다.

LBAC 증명서에서 액세스를 허용하지 않는 보호 컬럼에 액세스를 시도하면 액세스에 실패하며 오류 메시지를 수신합니다.

LBAC 증명서에서 읽기를 허용하지 않는 보호 행을 읽으려고 시도하면 DB2는 마치 행이 존재하지 않는 것처럼 작동합니다. 이러한 행은 SELECT, UPDATE 또는 DELETE를 비롯하여 실행 중인 모든 SQL문의 일부로 선택할 수 없습니다. 집계 함수에서도 LBAC에서 읽기를 허용하지 않는 행은 무시합니다. 예를 들어, COUNT(*) 함수는 읽기 액세스 권한이 있는 행에 대한 계수만 리턴합니다.

뷰 및 LBAC

보호되지 않은 테이블에 대해 뷰를 정의할 때와 동일한 방법으로 보호 테이블에 대해서도 뷰를 정의할 수 있습니다. 이러한 뷰에 액세스할 때 기본 테이블에 대한 LBAC 보호를 강제실행합니다. 사용되는 LBAC 증명서는 세션 권한 부여 ID입니다. 동일한 뷰에 액세스하는 두 사용자는 LBAC 증명서에 따라 다른 행을 볼 수 있습니다.

참조 무결성 제한조건 및 LBAC

다음 규칙은 참조 무결성 제한조건이 적용될 경우 LBAC 규칙을 강제실행하는 방법에 대해 설명합니다.

- **규칙 1**: LBAC 읽기 액세스 규칙은 내부적으로 생성된 하위 테이블 스캔에는 적용되지 않습니다. 이는 상위 항목이 삭제된 하위가 없도록 하기 위해서 입니다.
- **규칙 2**: LBAC 읽기 액세스 규칙은 내부적으로 생성된 상위 테이블 스캔에는 적용되지 않습니다.
- **규칙 3**: LBAC 쓰기 규칙은 하위 테이블에서 CASCADE 조작을 수행할 경우에만 적용됩니다. 예를 들어, 사용자가 상위를 삭제하나 LBAC 쓰기 규칙 위반으로 인해 하위 중 하나를 삭제할 수 없는 경우, 삭제는 롤백되고 오류가 발생합니다.

LBAC 사용 시 스토리지 오버헤드

LBAC를 사용하여 행 레벨에서 테이블을 보호하는 경우 행 보안 레이블 컬럼의 비용이 추가 스토리지 비용입니다. 이 비용은 선택한 보안 레이블의 유형에 따라 달라집니다. 예를 들어, 두 개의 구성요소를 포함하는 보안 규정을 작성할 경우 해당 보안 규정의 보안 레이블은 16바이트(각 구성요소당 8바이트)를 차지합니다. 행 보안 레이블 컬럼은 널(NULL) 입력이 가능하지 않은 컬럼으로 간주되므로, 이 경우 총 비용은 행당

20바이트입니다. 일반적으로 행당 총 비용은 *N**8 + 4바이트입니다. 여기서 *N*은 테이블을 보호하는 보안 규정에 포함된 구성요소 수입니다.

LBAC를 사용하여 컬럼 레벨에서 테이블을 보호하는 경우 컬럼 보안 레이블이 메타데이터입니다. 즉, 이 레이블은 SYSCOLUMNS 카탈로그 테이블에 컬럼의 메타데이터와 함께 저장됩니다. 이 메타데이터는 단순히 컬럼을 보호하는 보안 레이블의 ID입니다. 이 경우 사용자 테이블에서는 스토리지 오버헤드가 발생하지 않습니다.

LBAC가 수행하지 않는 작업

- LBAC는 임의 액세스 제어(Discretionary Access Control)에 의해 숨겨진 데이터에는 액세스할 수 없습니다.

  예: 테이블을 읽을 사용 권한이 없으면, LBAC가 액세스를 허용한 행 및 컬럼일지라도 테이블에 있는 데이터를 읽을 수 없습니다.
- LBAC 증명서는 보호 데이터에 대한 액세스만 제한합니다. 보호되지 않는 데이터에 대한 액세스에는 영향을 미치지 않습니다.
- 테이블 또는 데이터베이스에 보호 데이터가 포함되어 있더라도 테이블 또는 데이터베이스를 삭제할 때 LBAC 증명서를 점검하지 않습니다.
- 데이터를 백업할 때 LBAC 증명서를 점검하지 않습니다. 테이블에 대해 백업을 실행할 수 있는 경우, 백업되는 행은 데이터에 대한 LBAC 보호에 의해 제한을 받지 않습니다. 또한, 백업 미디어의 데이터도 LBAC에 의해 보호되지 않습니다. 데이터베이스 내의 데이터만이 보호됩니다.
- 다음 유형의 테이블을 보호하기 위해 LBAC를 사용할 수 없습니다.
  - 구체화된 쿼리 테이블
  - 구체화된 쿼리 테이블이 종속하는 테이블
  - 스테이징 테이블
  - 스테이징 테이블이 종속하는 테이블
  - 유형이 지정된 테이블
- 별칭에는 LBAC 보호를 적용할 수 없습니다.

LBAC 자습서

LBAC 사용의 기본적인 사항을 안내하는 자습서는 http://www.ibm.com/developerworks/db2에서 온라인으로 사용 가능한 DB2 레이블 기반 액세스 제어, 실습 안내서입니다.

LBAC 보안 규정

보안 관리자는 보안 규정을 사용하여 테이블의 개별 행과 개별 컬럼에 대해 쓰기 액세스 권한을 갖고 있는 사용자와 읽기 액세스 권한을 갖고 있는 사용자를 판별하는 기준을 정의할 수 있습니다.

보안 규정에는 다음 정보가 포함됩니다.

- 규정에 포함된 보안 레이블에 사용할 보안 레이블 구성요소
- 이러한 보안 레이블 구성요소를 비교할 때 사용할 규칙
- 규정으로 보호되는 데이터에 액세스할 때 사용할 특정 선택적 동작
- 보안 규정으로 보호되는 데이터에 강제로 액세스할 때 추가로 고려할 보안 레이블 및 면제. 예를 들어, 역할 및 그룹에 부여된 보안 레이블을 고려할지 여부에 대한 옵션은 보안 규정을 통해 제어됩니다.

모든 보호 테이블에는 연관된 하나의 보안 규정만 있어야 합니다. 해당 테이블의 행 및 컬럼은 해당 보안 규정의 일부인 보안 레이블을 사용하여 보호될 수 있으며 보호 데이터에 대한 모든 액세스는 해당 규정의 규칙을 따릅니다. 단일 데이터베이스에 복수의 보안 규정이 있을 수는 있으나 주어진 테이블을 보호하는 둘 이상의 보안 규정이 있을 수는 없습니다.

보안 규정 작성

보안 규정을 작성하려면 보안 관리자여야 합니다. CREATE SECURITY POLICY SQL 문을 사용하여 보안 규정을 작성합니다. CREATE SECURITY POLICY문을 실행하기 전에 보안 규정에 나열된 보안 레이블 구성요소를 작성해야 합니다. 보안 규정을 작성할 때 구성요소를 나열하는 순서는 우선순위 또는 구성요소 간의 기타 관계를 나타내지는 않으나 SECLABEL과 같은 내장 함수를 사용하여 보안 레이블을 작성할 경우에는 순서를 알아야 합니다.

작성한 보안 규정을 통해 데이터를 보호할 보안 레이블을 작성할 수 있습니다.

보안 규정 변경

보안 관리자는 ALTER SECURITY POLICY문을 사용하여 보안 규정을 수정할 수 있습니다.

보안 규정 삭제

보안 규정을 삭제하려면 보안 관리자여야 합니다. DROP SQL문을 사용하여 보안 규정을 삭제할 수 있습니다.

보안 규정이 테이블과 연관(테이블에 추가)되어 있으면 이를 삭제할 수 없습니다.

LBAC 보안 레이블 구성요소 개요

*보안 레이블 구성요소*는 LBAC(Lbel-Based Access Control)의 일부인 데이터베이스 오브젝트입니다. 보안 레이블 구성요소를 사용하여 조직의 보안 구조를 모델화할 수 있습니다.

보안 레이블 구성요소는 사용자가 제공된 데이터에 대한 액세스 권한이 있는지를 판별하기 위해 사용하는 임의의 기준을 나타낼 수 있습니다. 이러한 기준의 일반적인 예는 다음과 같습니다.

- 사용자의 신뢰도
- 사용자가 소속된 부서
- 사용자가 특정 프로젝트에 참여하는지 여부

예: 사용자가 소속된 부서에 따라 사용자가 액세스할 수 있는 데이터가 다르게 하기 위해, DEPT라는 구성요소를 작성한 후 해당 구성요소에 회사 내 여러 부서의 이름을 지정하는 요소를 정의할 수 있습니다. 그런 다음 보안 규정에 DEPT라는 구성요소를 포함시킵니다.

보안 레이블 구성요소의 *요소*는 해당 구성요소에 허용되는 하나의 특정 "설정"입니다.

예: 신뢰도를 나타내는 보안 레이블 구성요소는 일급 비밀, 비밀, 분류 또는 분류되지 않음의 네 가지 요소를 가질 수 있습니다.

보안 레이블 구성요소 작성

보안 레이블 구성요소를 작성하려면 보안 관리자여야 합니다. CREATE SECURITY LABEL COMPONENT SQL문을 사용하여 보안 레이블 구성요소를 작성합니다.

보안 레이블 구성요소를 작성할 때 다음 정보를 제공해야 합니다.

- 구성요소의 이름
- 구성요소의 유형(ARRAY, TREE 또는 SET)
- 허용되는 요소의 전체 목록
- ARRAY 및 TREE 유형의 경우, 각 요소를 구성요소 구조에 맞게 조정하는 방법을 기술해야 합니다.

보안 레이블 구성요소를 작성한 후에는 이 구성요소를 기준으로 보안 규정을 작성할 수 있습니다. 이 보안 규정을 통해 데이터를 보호할 보안 레이블을 작성할 수 있습니다.

구성요소 유형

보안 레이블 구성요소 유형에는 다음 세 가지가 있습니다.

- TREE: 각 요소가 트리 구조의 노드를 나타냄

- ARRAY: 각 요소가 선형 스케일의 한 지점을 나타냄
- SET: 각 요소가 세트의 한 구성원을 나타냄

유형은 요소가 서로 관련될 수 있는 다른 방법을 모델화하는 데 사용됩니다. 예를 들어, 회사 내의 하나 이상의 부서를 기술하기 위한 구성요소를 작성 중인 경우, 대부분의 비즈니스 구조가 트리 형태이기 때문에 TREE 유형의 구성요소를 사용할 수 있습니다. 개인의 신뢰도를 나타내기 위한 구성요소를 작성 중인 경우, 두 가지 신뢰도 중 하나가 항상 다른 하나보다 높기 때문에 ARRAY 유형의 구성요소를 사용할 수 있습니다.

요소가 서로에 대해 가질 수 있는 관계에 대한 자세한 설명을 비롯하여 각 유형에 대한 세부사항은 자체 절에 설명되어 있습니다.

보안 레이블 구성요소 변경

보안 관리자는 ALTER SECURITY LABEL COMPONENT문을 사용하여 보안 레이블 구성요소를 수정할 수 있습니다.

보안 레이블 구성요소 삭제

보안 레이블 구성요소를 삭제하려면 보안 관리자여야 합니다. SQL문 DROP을 사용하여 보안 레이블 구성요소를 삭제할 수 있습니다.

LBAC 보안 레이블 구성요소 유형: SET

SET은 레이블 기반 액세스 제어(LBAC) 보안 규정에 사용될 수 있는 보안 레이블 구성요소의 한 유형입니다.

SET 유형의 구성요소는 순서가 지정되지 않은 요소 목록입니다. 이러한 구성요소 유형을 가진 요소에 대해서는 제공된 요소가 목록에 있는지 여부만 비교할 수 있습니다.

LBAC 보안 레이블 구성요소 유형: ARRAY

ARRAY는 보안 레이블 구성요소의 한 유형입니다.

이 ARRAY 유형의 구성요소에서 구성요소를 작성할 때 요소를 나열하는 순서는 첫 번째 나열된 요소가 최고값이 되고 마지막에 나열된 요소가 최저값이 되는 스케일을 정의합니다.

예: 다음과 같은 방법으로 mycomp 구성요소를 정의한다고 가정합니다.

```
CREATE SECURITY LABEL COMPONENT mycomp
  ARRAY [ '일급 비밀', '비밀', '직원', '공용' ]
```

이 경우, 요소는 다음과 같은 구조로 구성된다고 간주됩니다.

ARRAY 유형의 구성요소에서, 요소는 서로에 대해 다음과 같은 유형의 관계를 가질 수 있습니다.

보다 높음

ARRAY절에서 요소 A가 요소 B의 앞에 있으면 요소 A가 요소 B보다 높습니다.

보다 낮음

ARRAY절에서 요소 A가 요소 B의 뒤에 있으면 요소 A가 요소 B보다 낮습니다.

LBAC 보안 레이블 구성요소 유형: TREE

TREE는 레이블 기반 액세스 제어(LBAC) 보안 규정에 사용될 수 있는 보안 레이블 구성요소의 한 유형입니다.

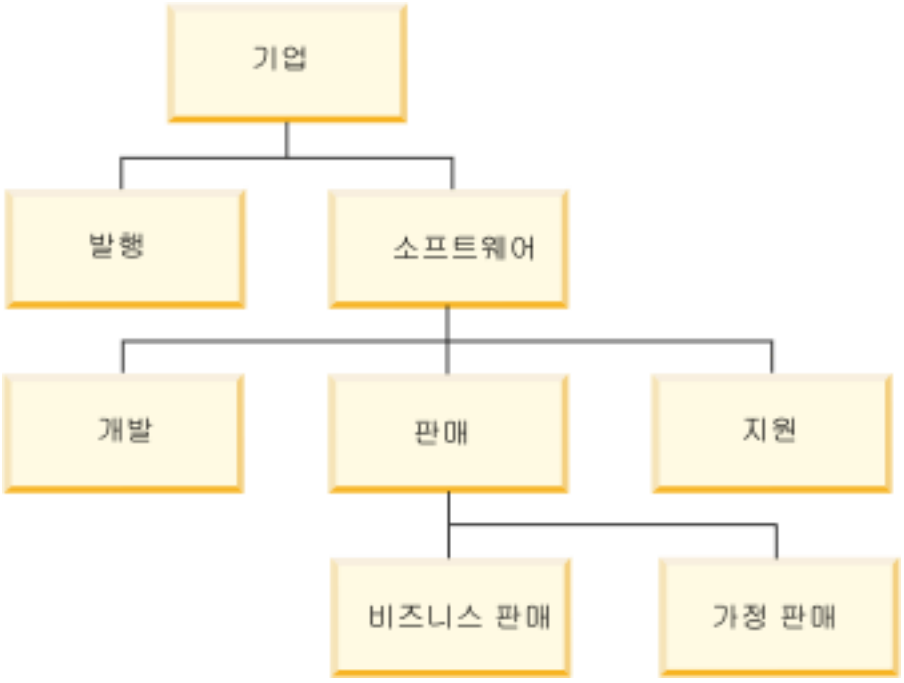
이 유형의 구성요소에서는 요소가 트리 구조로 배열되어 있다고 간주됩니다. TREE 유형을 가진 구성요소의 일부인 요소를 지정할 경우 요소가 속해 있는 다른 요소도 지정해야 합니다. 한 가지 예외는 첫 번째 요소로서, 트리의 ROOT가 되도록 지정해야 합니다. 이 요소를 통해 트리 구조에 요소를 조직할 수 있습니다.

예: 다음과 같은 방법으로 mycomp 구성요소를 정의한다고 가정합니다.

```
CREATE SECURITY LABEL COMPONENT mycomp
TREE (
        '기업'          ROOT,
        '발행'          UNDER '기업',
        '소프트웨어'    UNDER '기업',
        '개발'          UNDER '소프트웨어',
```

```
        '판매'           UNDER '소프트웨어',
        '지원'           UNDER '소프트웨어'
        '비즈니스 판매'  UNDER '판매'
        '가정 판매'        UNDER '판매'
)
```

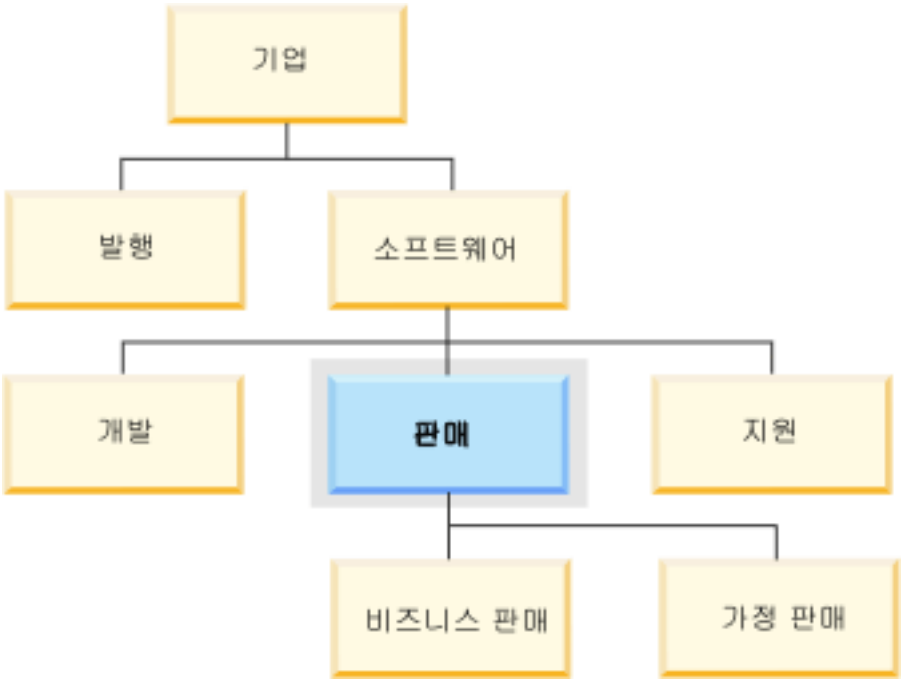
이 경우, 요소는 다음과 같은 트리 구조로 구성된다고 간주됩니다.



TREE 유형의 구성요소에서, 요소는 서로에 대해 다음과 같은 유형의 관계를 가질 수 있습니다.

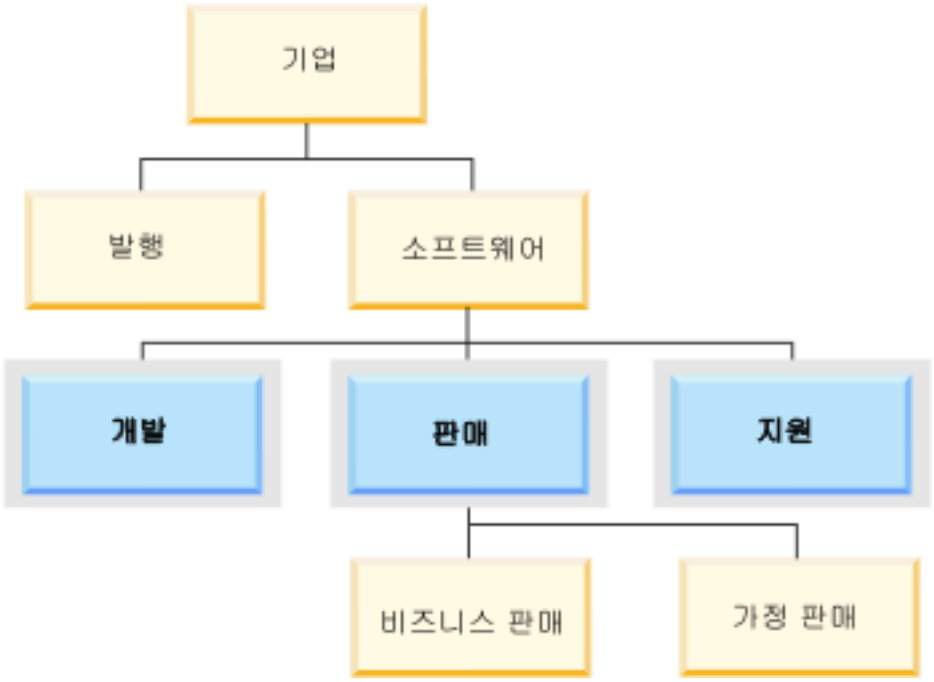
상위 요소 B가 A의 아래에 있으면 요소 A가 B의 상위입니다.

예: 이 다이어그램은 비즈니스 판매 요소의 상위를 표시합니다.



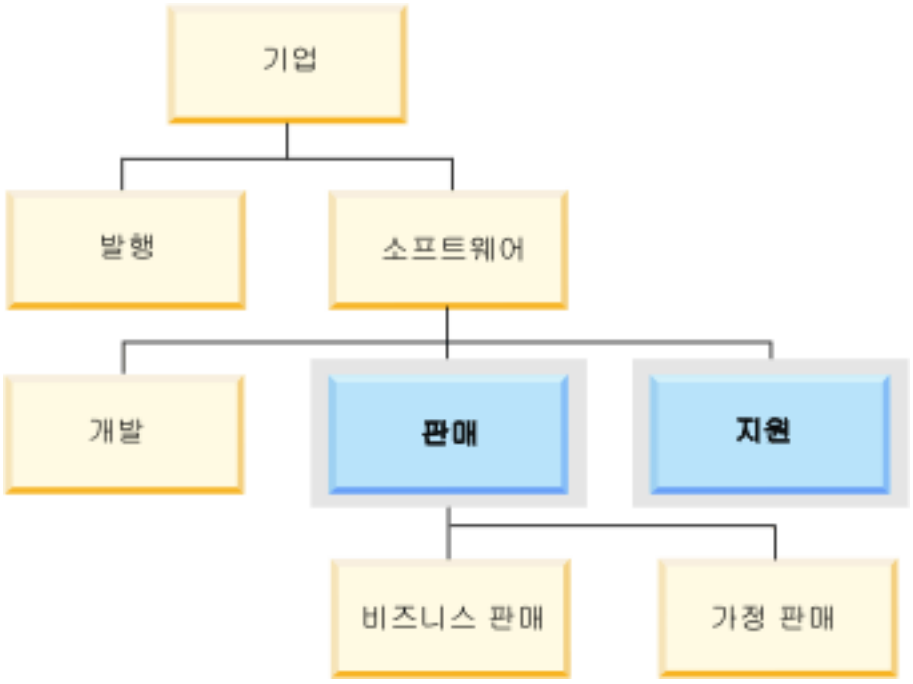
하위 요소 A가 B의 아래에 있으면 요소 A가 B의 하위입니다.

예: 이 다이어그램은 소프트웨어 요소의 하위를 표시합니다.



동위 두 요소가 동일한 상위를 가질 경우 두 요소가 동일합니다.

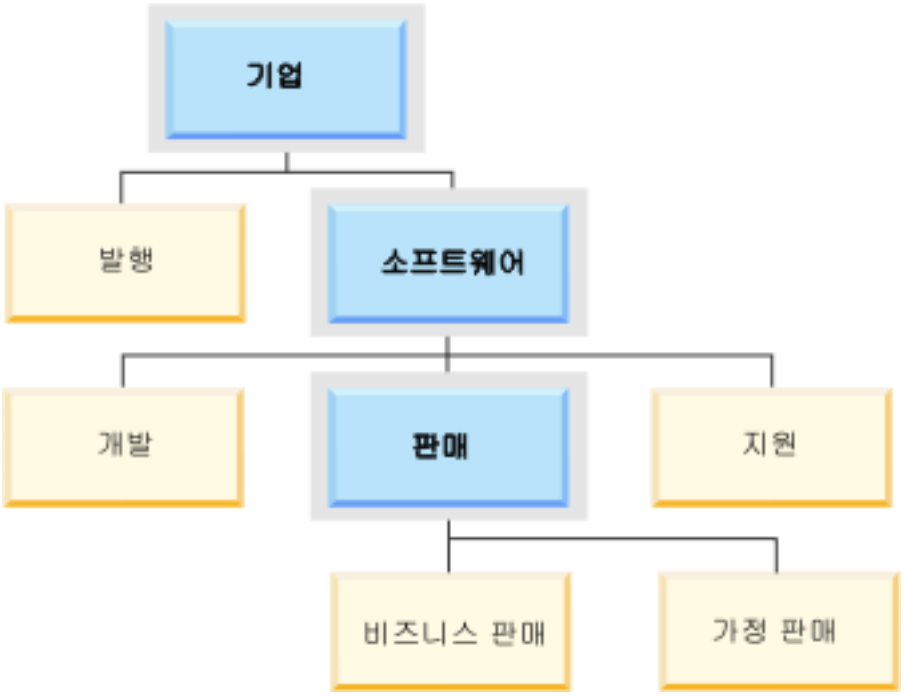
예: 이 다이어그램은 개발 요소의 동위를 표시합니다.



상위 구성원

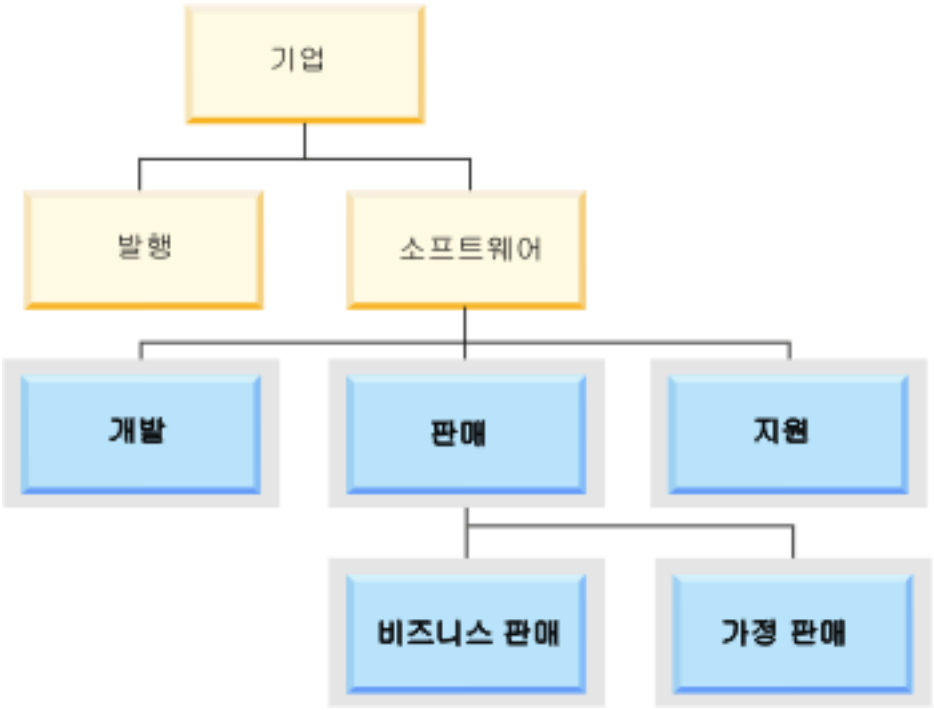
요소 A가 B의 상위이거나 B의 상위의 상위일 경우 요소 A는 요소 B의 상위 구성원입니다. 루트 요소는 트리 내의 다른 모든 요소의 상위 구성원입니다.

예: 이 다이어그램은 가정 판매 요소의 상위 구성원입니다.



하위 요소 A가 B의 하위이거나 B의 하위의 하위일 경우 요소 A는 요소 B의 하위입니다.

예: 이 다이어그램은 소프트웨어 요소의 하위를 표시합니다.



LBAC 보안 레이블

레이블 기반 액세스 제어(LBAC)에서 *보안 레이블*은 특정 세트의 보안 기준에 대해 기술하는 데이터베이스 오브젝트입니다. 데이터를 보호하기 위해 데이터에 보안 레이블을 적용합니다. 사용자가 보호 데이터에 액세스할 수 있도록 하기 위해 보안 레이블을 부여합니다.

사용자가 보호 데이터에 액세스하면 이 보안 레이블을 데이터를 보호하는 보안 레이블과 비교합니다. 보호 보안 레이블은 일부 보안 레이블은 차단하고 일부 레이블은 차단하지 않습니다. 사용자의 보안 레이블이 차단되면 사용자가 데이터에 액세스할 수 없습니다.

각 보안 레이블은 단 하나의 보안 규정의 일부이며 해당 보안 규정의 각 구성요소에 대해 하나의 값을 포함합니다. 보안 레이블 구성요소의 컨텍스트에 있는 값은 해당 구성요소에 허용되는 0개 이상의 요소로 이루어진 목록입니다. ARRAY 유형을 가진 구성요소 값은 0개 또는 하나의 요소를 포함할 수 있으며 다른 유형을 가진 구성요소의 값은 0개 이상의 요소를 가질 수 있습니다. 요소가 하나도 포함되지 않는 값을 *공백 값*이라고 합니다.

예: TREE 유형 구성요소에 세 가지 요소(인적 자원, 영업 및 운송)가 있을 경우, 다음은 해당 구성요소에 대해 올바른 값 중 일부입니다.

- 인적 자원(또는 요소 자체 중 하나)

- 인적 자원, 운송(또는 요소의 임의의 조합 - 요소가 한 번 이상 포함되지 않는 경우)
- *공백 값*

특정 보안 레이블이 다른 보안 레이블을 차단할지 여부는 레이블에 있는 각 구성요소의 값 및 테이블의 보안 규정에 지정된 LBAC 규칙 세트에 의해 결정됩니다. 비교 방법에 대한 세부사항은 LBAC 보안 레이블을 비교하는 방법에 대한 주제에 나와 있습니다.

보안 레이블을 텍스트 문자열로 변환할 경우, 보안 레이블은 보안 레이블 값의 형식에 대한 주제에 설명된 형식을 사용합니다.

보안 레이블 작성

보안 레이블을 작성하려면 보안 관리자여야 합니다. CREATE SECURITY LABEL SQL문을 사용하여 보안 레이블을 작성합니다. 보안 레이블을 작성할 때 다음 정보를 제공해야 합니다.

- 레이블의 이름
- 레이블이 속한 보안 규정
- 보안 규정에 포함된 하나 이상의 구성요소의 값

값이 지정되지 않은 모든 구성요소는 값이 공백이라고 가정합니다. 보안 레이블은 최소한 하나의 공백이 아닌 값을 가져야 합니다.

보안 레이블 변경

보안 레이블은 변경할 수 없습니다. 보안 레이블을 변경하는 유일한 방법은 해당 레이블을 삭제한 후 재작성하는 것입니다.그러나 보안 레이블의 *구성요소*는 보안 관리자가 ALTER SECURITY LABEL COMPONENT문을 사용하여 수정할 수 있습니다.

보안 레이블 삭제

보안 레이블을 삭제하려면 보안 관리자여야 합니다. DROP SQL문을 사용하여 보안 레이블을 삭제할 수 있습니다. 데이터베이스에서 데이터를 보호하는 데 사용 중이거나 현재 하나 이상의 사용자가 보유하고 있는 보안 레이블은 삭제할 수 없습니다.

보안 레이블 권한 부여

사용자, 그룹 또는 역할에 보안 레이블을 부여하려면 보안 관리자여야 합니다. GRANT SECURITY LABEL SQL문을 사용하여 보안 레이블을 부여하십시오. 보안 레이블을 부여할 때, 읽기 액세스, 쓰기 액세스 또는 읽기 및 쓰기 액세스 모두에 대해 보안 레이블을 부여할 수 있습니다. 한 사용자, 그룹 또는 역할은 동일한 유형의 액세스에 대해 동일한 보안 규정의 보안 레이블을 두 개 이상 보유할 수 없습니다.

보안 레이블 취소

사용자, 그룹 또는 역할에서 보안 레이블을 취소하려면 보안 관리자여야 합니다. 보안 레이블을 취소하려면 REVOKE SECURITY LABEL SQL문을 사용하십시오.

보안 레이블과 호환 가능한 데이터 유형

보안 레이블은 SYSPROC.DB2SECURITYLABEL 유형의 데이터를 갖고 있습니다. SYSPROC.DB2SECURITYLABEL 및 VARCHAR(128) FOR BIT DATA 간의 데이터 변환이 지원됩니다.

사용자가 보유한 보안 레이블 판별

다음 쿼리를 사용하여 사용자가 보유한 보안 레이블을 판별할 수 있습니다.

```
SELECT A.grantee, B.secpolicyname, c.seclabelname
FROM syscat.securitylabelaccess A, syscat.securitypolicies B, syscat.securitylabels C
WHERE A.seclabelid = C.seclabelid and B.secpolicyid = C.secpolicyid
```

보안 레이블 값의 형식

종종 보안 레이블의 값은 문자열 형식으로 표시됩니다(예: 내장 함수인 SECLABEL을 사용할 경우).

보안 레이블의 값이 문자열로 표시되면 다음과 같은 형식이 사용됩니다.

- 구성요소의 값은 구성요소가 보안 규정의 CREATE SECURITY POLICY문에 나열된 순서와 동일한 순서로 나열됩니다.
- 요소는 해당 요소의 이름으로 표시됩니다.
- 다른 구성요소의 요소는 콜론(:)으로 분리됩니다.
- 동일한 구성요소에 대해 둘 이상의 요소가 제공된 경우 요소는 괄호(()) 안에 표시되며 쉼표(,)로 구분됩니다.
- 공백 값은 공백 괄호(()) 세트로 표시됩니다.

예: 보안 레이블은 구성요소가 세 개(레벨, 부서 및 프로젝트 순)인 보안 규정의 일부입니다. 보안 레이블은 다음과 같은 값을 가집니다.

표 8.

구성요소	값
레벨	비밀
부서	*공백 값*
프로젝트	• Epsilon 37 • Megaphone • Cloverleaf

이 보안 레이블 값은 다음과 같은 문자열로 표시됩니다.

```
'Secret:():(Epsilon 37,Megaphone,Cloverleaf)'
```

LBAC 보안 레이블을 비교하는 방법

레이블 기반 액세스 제어(LBAC)로 보호되는 데이터에 액세스할 경우 액세스가 차단되었는지 여부를 판별하기 위해 LBAC 증명서를 하나 이상의 보안 레이블과 비교합니다. LBAC 증명서는 사용자가 보유하는 모든 보안 레이블과 면제입니다.

수행할 수 있는 비교 유형은 두 가지뿐입니다. 읽기 액세스의 경우에는 단일 보안 레이블을 LBAC 증명서와 비교하고 쓰기 액세스의 경우에는 LBAC 증명서를 단일 보안 레이블과 비교할 수 있습니다. 갱신 및 삭제는 읽기 및 쓰기 조작으로 이루어집니다. 조작에 복수의 비교가 필요한 경우 각 비교는 별도로 수행됩니다.

사용되는 보안 레이블

복수의 보안 레이블을 갖고 있더라도 하나의 보안 레이블만이 보호 보안 레이블과 비교됩니다. 사용되는 레이블은 다음 기준을 만족하는 레이블입니다.

- 액세스되는 테이블을 보호하는 보안 규정의 일부입니다.
- 액세스 유형(읽기 또는 쓰기)에 부여되었습니다.

이러한 기준을 만족하는 보안 레이블이 없으면 모든 구성요소에 대해 공백 값을 갖고 있는 디폴트 보안 레이블을 사용한다고 가정합니다.

비교 수행 방법

보안 레이블은 구성요소 단위로 구성요소와 비교됩니다. 보안 레이블에 구성요소 중 하나에 대한 값이 없으면 공백 값을 갖고 있다고 가정합니다. 각 구성요소는 검토되기 때문에, 해당 구성요소의 값에 있는 요소가 보호 레이블의 동일한 구성요소의 값에 있는 요소에 의해 차단되는지 여부를 결정하기 위해 LBAC 규칙 세트의 적절한 규칙을 사용합니다. 값 중 하나가 차단될 경우 LBAC 증명서는 보호 보안 레벨에 의해 차단됩니다.

비교에 사용되는 LBAC 규칙 세트는 보안 규정에 지정됩니다. 어떤 규칙이 있는지와 각 규칙의 용도를 찾으려면 해당 규칙 세트에 대한 설명을 참조하십시오.

면제가 비교에 미치는 영향

두 값을 비교하는 데 사용되는 규칙에 대한 면제를 갖고 있으면 비교가 수행되지 않으며 보호 값은 보안 레이블에 있는 값을 차단하지 않는다고 가정합니다.

예: LBAC 규칙 세트는 DB2LBACRULES이며 보안 규정에는 두 개의 구성요소가 있습니다. 한 구성요소는 ARRAY 유형을 갖고 다른 구성요소는 TREE 유형을 갖습니다. TREE 유형의 구성요소 값을 비교할 때, 사용자에게 DB2LBACREADTREE 규칙에 대한 면제 권한이 부여됩니다. DB2LBACREADTREE 규칙은 읽기 액세스에 사

용되는 규칙입니다. 사용자가 보호 데이터를 읽으려 하면, 규칙이 사용되지 않기 때문에 TREE 구성요소에 대해 갖고 있는 모든 값(공백 값도 포함)은 액세스를 차단하지 않습니다. 사용자가 데이터를 읽을 수 있는지 여부는 전적으로 레이블의 ARRAY 구성요소의 값에 달려 있습니다.

LBAC 규칙 세트 개요

LBAC 규칙 세트는 보안 레이블을 비교할 때 사용되는 사전 정의된 규칙 세트입니다. 두 보안 레이블의 값을 비교할 때, 규칙 세트에 있는 하나 이상의 규칙을 사용하여 하나의 값이 다른 값을 차단하는지 여부를 판별합니다.

각 LBAC 규칙 세트는 고유한 이름으로 식별됩니다. 보안 규정을 작성할 때 해당 규정에 사용할 LBAC 규칙을 지정해야 합니다. 해당 규정의 일부인 보안 레이블을 비교할 때 해당 LBAC 규칙 세트를 사용합니다.

규칙 세트의 각 규칙은 고유한 이름으로 식별됩니다. 해당 규칙에 대한 면제를 부여할 때 규칙 이름을 사용합니다.

세트에 있는 규칙의 수와 각 규칙의 용도는 규칙 세트마다 다릅니다.

현재는 하나의 LBAC 규칙 세트만이 지원됩니다. 해당 규칙 세트의 이름은 DB2LBACRULES입니다.

LBAC 규칙 세트: DB2LBACRULES

DB2LBACRULES LBAC 규칙 세트는 보안 레이블 구성요소의 값을 비교하기 위한 일반적인 규칙 세트를 제공합니다. LBAC 규칙 세트는 write-up 및 write-down 모두를 보호합니다.

write-up 및 write down의 개념

Write-up 및 write-down은 ARRAY 유형의 구성요소와 쓰기 액세스에만 적용됩니다. Write up은 사용자가 쓰려고 하는 데이터 보호 값이 사용자의 값보다 높은 경우에 발생합니다. Write-down은 데이터 보호 값이 사용자의 값보다 낮은 경우에 발생합니다. 디폴트로 write-up 및 write-down 모두 허용되지 않습니다. 이는 사용자의 값과 동일한 값에 의해 보호되는 데이터만 쓸 수 있음을 의미합니다.

동일한 구성요소에 대한 두 값을 비교할 경우, 사용되는 규칙은 구성요소의 유형(ARRAY, SET 또는 TREE)과 시도하는 액세스 유형에 따라 다릅니다. 이 테이블에는 규칙이 나열되고, 규칙의 용도를 알려주며, 액세스가 차단될 경우 규칙을 판별하는 방법에 대해 설명합니다.

표 9. DB2LBACRULES 규칙에 대한 요약

규칙 이름	구성요소의 값을 비교할 때 사용되는 유형	액세스할 때 사용되는 유형	액세스가 차단되는 조건
DB2LBACREADARRAY	ARRAY	읽기	사용자의 값이 보호 값보다 낮음
DB2LBACREADSET	SET	읽기	사용자가 보유하지 않는 하나 이상의 보호 값이 있습니다.
DB2LBACREADTREE	TREE	읽기	사용자의 값 모두가 보호 값과 동일하지 않거나 보호 값 중 하나의 상위 구성원과 동일하지 않음
DB2LBACWRITEARRAY	ARRAY	쓰기	사용자의 값이 보호 값보다 높거나 보호 값보다 낮습니다. [1]
DB2LBACWRITESET	SET	쓰기	사용자가 보유하지 않는 하나 이상의 보호 값이 있습니다.
DB2LBACWRITETREE	TREE	쓰기	사용자의 값 모두가 보호 값과 동일하지 않거나 보호 값 중 하나의 상위 구성원과 동일하지 않음

주:

1. DB2LBACWRITEARRAY 규칙은 두 개의 다른 규칙이 결합된 것이라고 생각하면 됩니다. 한 규칙은 사용자의 레벨보다 높은 데이터를 쓰지 못하게 하고(write-up) 다른 규칙은 사용자의 레벨보다 낮은 데이터를 쓰지 못하게 합니다(write-down). 이 규칙에 대해 면제 권한을 부여하면 이러한 규칙 또는 모든 규칙으로부터 사용자를 제외시킬 수 있습니다.

규칙이 공백 값을 처리하는 방법

모든 규칙은 공백 값을 동일한 방법으로 처리합니다. 공백 값은 어떤 값도 차단하지 않으며 공백이 아닌 값에 의해 차단됩니다.

DB2LBACREADSET 및 DB2LBACWRITESET 예

이 예는 보호 데이터를 읽거나 쓰려고 하는 사용자에게 적용됩니다. 이 예에서는 값이 SET 유형(요소가 1, 2, 3, 4임)을 가진 구성요소용이라고 가정합니다.

표 10. DB2LBACREADSET 및 DB2LBACWRITESET 규칙을 적용한 예

사용자의 값	보호 값	액세스 차단 여부
'1'	'1'	차단되지 않습니다. 값이 동일합니다.
'(1,2,3)'	'1'	차단되지 않습니다. 사용자 값에 'one' 요소가 포함됩니다.
'(1,2)'	'(1,2,3)'	차단됩니다. 요소 '4'는 보호 값에는 있으나 사용자의 값에는 없습니다.
'()'	'1'	차단됩니다. 공백 값은 공백이 아닌 값에 의해 차단됩니다.
'1'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.
'()'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.

DB2LBACREADTREE 및 DB2LBACWRITETREE

이 예들은 읽기 액세스 및 쓰기 액세스 모두에 대해 유효합니다. 이 예에서는 값이 다음과 같은 배열에 다음과 같은 방법이 정의된 TREE 유형을 가진 구성요소용이라고 가정합니다.

```
CREATE SECURITY LABEL COMPONENT mycomp
TREE (
        '기업'           ROOT,
        '발행'           UNDER '기업',
        '소프트웨어'     UNDER '기업',
        '개발'           UNDER '소프트웨어',
        '판매'           UNDER '소프트웨어',
        '지원'           UNDER '소프트웨어'
        '비즈니스 판매'  UNDER '판매'
        '가정 판매'        UNDER '판매'
)
```

여기서 요소는 다음과 같은 배열을 갖습니다.

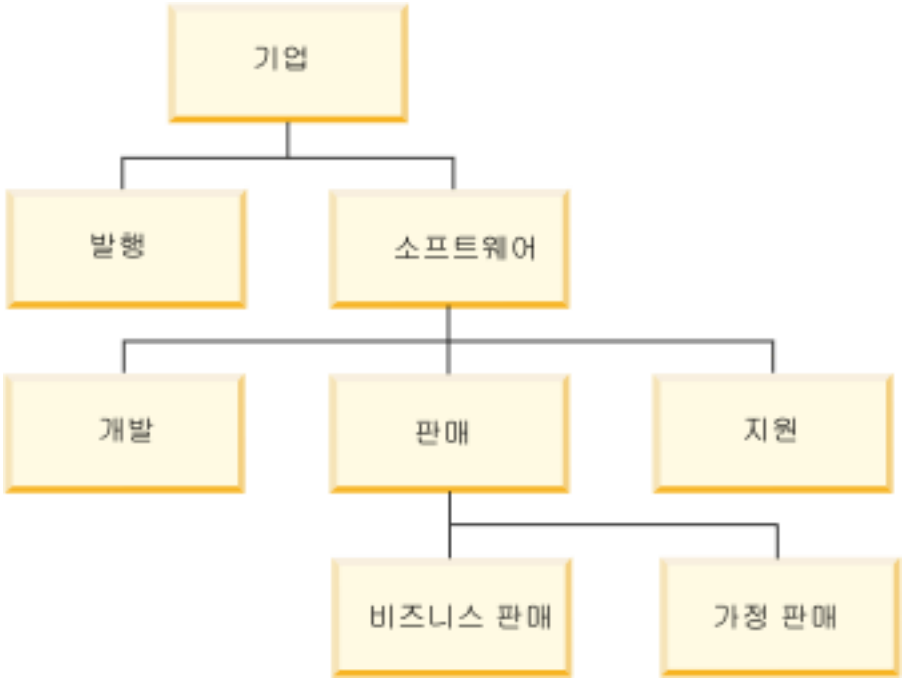


표 11. DB2LBACREADTREE 및 DB2LBACWRITETREE 규칙을 적용한 예

사용자의 값	보호 값	액세스 차단 여부
'(지원, 판매)'	'개발'	차단됩니다. '개발' 요소는 사용자 값 중 하나가 아니며 '지원' 및 '판매' 모두 '개발'의 상위 구성원이 아닙니다.
'(개발, 소프트웨어)'	'(비즈니스 판매, 발행)'	차단되지 않습니다. '소프트웨어' 요소는 '비즈니스 판매'의 상위 구성원이 아닙니다.
'(발행, 판매)'	'(발행, 지원)'	차단되지 않습니다. '발행'은 두 개의 값 세트에 속합니다.

표 11. DB2LBACREADTREE 및 DB2LBACWRITETREE 규칙을 적용한 예 (계속)

사용자의 값	보호 값	액세스 차단 여부
'기업'	'개발'	차단되지 않습니다. 루트 값은 다른 모든 값의 상위 구성원에 있습니다.
'()'	'판매'	차단됩니다. 공백 값은 공백이 아닌 값에 의해 차단됩니다.
'가정 판매'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.
'()'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.

DB2LBACREADARRAY 예

이 예는 읽기 액세스에만 적용됩니다. 이 예에서는 값이 다음과 같은 배열에 다음과 같은 요소를 포함하는 ARRAY 유형을 갖는 구성요소용이라고 가정합니다.

표 12. DB2LBACREADARRAY 규칙을 적용한 예

사용자의 값	보호 값	읽기 액세스 차단 여부
'비밀'	'직원'	차단되지 않습니다. '비밀' 요소가 '직원' 요소에 비해 높습니다.
'비밀'	'비밀'	차단되지 않습니다. 값이 동일합니다.
'비밀'	'일급 비밀'	차단됩니다. '일급 비밀' 요소가 '비밀' 요소보다 높습니다.
'()'	'공용'	차단됩니다. 공백 값은 공백이 아닌 값에 의해 차단됩니다.

표 12. DB2LBACREADARRAY 규칙을 적용한 예 (계속)

사용자의 값	보호 값	읽기 액세스 차단 여부
'공용'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.
'()'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.

DB2LBACWRITEARRAY 예

이 예는 쓰기 액세스에만 적용됩니다. 이 예에서는 값이 다음과 같은 배열에 다음과 같은 요소를 포함하는 ARRAY 유형을 갖는 구성요소용이라고 가정합니다.

표 13. DB2LBACWRITEARRAY 규칙을 적용한 예

사용자의 값	보호 값	쓰기 액세스 차단 여부
'비밀'	'직원'	차단됩니다. '직원' 요소가 '비밀' 요소보다 낮습니다.
'비밀'	'비밀'	차단되지 않습니다. 값이 동일합니다.
'비밀'	'일급 비밀'	차단됩니다. '일급 비밀' 요소가 '비밀' 요소보다 높습니다.
'()'	'공용'	차단됩니다. 공백 값은 공백이 아닌 값에 의해 차단됩니다.
'공용'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.
'()'	'()'	차단되지 않습니다. 공백 값에 의해서는 값이 차단되지 않습니다.

LBAC 규칙 면제

특정 보안 규정의 특정 규칙에 대한 LBAC 규칙 면제를 보유하고 있을 경우, 해당 보안 규정으로 보호되는 데이터에 액세스할 때 해당 규칙이 적용되지 않습니다.

면제가 부여된 것과 다른 보안 규정의 보안 레이블을 비교할 경우 면제는 영향을 미치지 않습니다.

예:

두 개의 테이블 T1과 T2가 있다고 가정합니다. T1은 보안 규정 P1에 의해 보호되고 T2는 보안 규정 P2에 의해 보호됩니다. 두 보안 규정에는 하나의 구성요소가 있습니다. 각 구성요소의 유형은 ARRAY입니다. T1 및 T2 모두 하나의 데이터 행만 갖고 있습니다. 읽기 액세스에 대해 보안 규정 P1이 적용된 보안 레이블에서는 T1의 행에 대한 액세스를 허용하지 않습니다. 읽기 액세스에 대해 보안 규정 P2가 적용된 보안 레이블에서는 T2의 행에 대한 액세스를 허용하지 않습니다.

이제 P1 규정의 DB2LBACREADARRAY에 대해 면제 권한이 부여되었습니다. 따라서 T1의 행은 읽을 수 있으나 T2의 행은 읽을 수 없습니다. 이유는 T2는 다른 보안 규정에 의해 보호되므로 해당 규정의 DB2LBACREADARRAY 규칙에 대한 면제를 보유하지 않기 때문입니다.

복수의 면제를 보유할 수 있습니다. 보안 규정에 사용되는 모든 규칙에 대한 면제를 보유할 경우 해당 보안 규정에 의해 보호되는 모든 데이터에 대해 전체 액세스 권한을 갖게 됩니다.

LBAC 규칙 면제 권한 부여

LBAC 규칙 면제 권한을 부여하려면 보안 관리자여야 합니다. LBAC 규칙 면제 권한을 부여하려면 GRANT EXEMPTION ON RULE SQL문을 사용하십시오.

LBAC 규칙 면제 권한을 부여할 때 다음 정보를 제공해야 합니다.

- 면제할 규칙
- 면제할 보안 규정
- 면제 권한을 부여하려는 사용자, 그룹 또는 역할

중요: LBAC 규칙 면제는 가장 강력한 액세스 권한을 제공합니다. 그러므로 면제 권한을 제공할 때에는 신중을 기하도록 하십시오.

LBAC 규칙 면제 권한 취소

LBAC 규칙 면제 권한을 취소하려면 보안 관리자여야 합니다. LBAC 규칙 면제 권한을 취소하려면 REVOKE EXEMPTION ON RULE SQL문을 사용하십시오.

사용자가 보유한 규칙 면제 판별

다음 쿼리를 사용하여 사용자가 보유한 규칙 면제를 판별할 수 있습니다.

```
SELECT A.grantee, A.accessrulename, B.secpolicyname
FROM syscat.securitypolicyexemptions A, syscat.securitypolicies B
WHERE A.secpolicyid = B.secpolicyid
```

LBAC 보안 레이블을 관리하기 위한 내장 함수

내장 함수 SECLABEL, SECLABEL_BY_NAME 및 SECLABEL_TO_CHAR은 레이블 기반 액세스 제어(LBAC) 보안 레이블을 관리하기 위해 제공됩니다.

각 함수는 이 절에서 간략히 설명되며 *SQL 참조서*에 자세히 설명되어 있습니다.

SECLABEL

이 내장 함수는 레이블 내의 각 구성요소에 대한 보안 규정 및 값을 지정함으로써 보안 레이블을 빌드하는 데 사용됩니다. 리턴되는 값은 DB2SECURITYLABEL 유형의 데이터를 가지며 표시된 보안 규정의 일부인 보안 레이블이고 구성요소에 대해 표시된 값을 가집니다. 표시된 값을 가진 보안 레벨이 이미 존재해야 하는 것은 아닙니다.

예: 테이블 T1에는 두 개의 컬럼이 있습니다. 첫 번째 컬럼의 데이터 유형은 DB2SECURITYLABEL이고 두 번째 컬럼의 데이터 유형은 INTEGER입니다. T1은 보안 규정 P1에 의해 보호됩니다. 보안 규정 P1에는 레벨, 부서 및 그룹의 세 가지 보안 레벨 구성요소가 있습니다. UNCLASSIFIED이 구성요소 레벨의 요소이고 ALPHA 및 SIGMA는 모두 부서 구성요소의 요소이며 G2는 그룹 구성요소의 요소인 경우, 다음과 같이 보안 레이블을 삽입할 수 있습니다.

```
INSERT INTO T1 VALUES
   ( SECLABEL( 'P1', 'UNCLASSIFIED:(ALPHA,SIGMA):G2' ), 22 )
```

SECLABEL_BY_NAME

이 내장 함수에는 보안 규정의 이름과 보안 규정의 일부인 보안 레이블의 이름을 사용할 수 있습니다. 그런 다음 표시된 보안 레이블을 DB2SECURITYLABEL로 리턴합니다. 기존의 보안 레이블을 데이터 유형이 DB2SECURITYLABEL인 컬럼에 삽입할 때 이 함수를 사용해야 합니다.

예: 테이블 T1에는 두 개의 컬럼이 있습니다. 첫 번째 컬럼의 데이터 유형은 DB2SECURITYLABEL이고 두 번째 컬럼의 데이터 유형은 INTEGER입니다. L1이라는 보안 레이블은 보안 규정 P1의 일부입니다. 다음 SQL은 보안 레이블을 삽입합니다.

```
INSERT INTO T1 VALUES ( SECLABEL_BY_NAME( 'P1', 'L1' ), 22 )
```

이 SQL문은 작동하지 않습니다.

```
INSERT INTO T1 VALUES ( P1.L1, 22 )     // Syntax Error!
```

SECLABEL_TO_CHAR

이 내장 함수는 보안 레이블을 구성하는 값의 문자열 표현을 리턴합니다.

예: 테이블 T1은 컬럼 C1의 데이터 유형은 DB2SECURITYLABEL입니다. T1은 보안 규정 P1에 의해 보호됩니다. 보안 규정 P1에는 레벨, 부서 및 그룹의 세 가지 보안 레벨 구성요소가 있습니다. T1에는 하나의 행이 있고 C1 컬럼에는 각 구성요소에 대해 다음과 같은 요소를 갖고 있는 값이 있습니다.

구성요소	요소
level	SECRET
departments	DELTA 및 SIGMA
groups	G3

행 읽기가 허용되는 LBAC 증명서를 갖고 있는 사용자가 다음 SQL문을 실행합니다.

```
SELECT SECLABEL_TO_CHAR( 'P1', C1 ) AS C1 FROM T1
```

다음과 같이 출력됩니다.

```
   C1

'SECRET:(DELTA,SIGMA):G3'
```

LBAC를 사용하여 데이터 보호

레이블 기반 액세스 제어(LBAC)를 사용하여 데이터의 행, 데이터의 컬럼 또는 둘 다를 보호할 수 있습니다. 테이블 내의 행은 테이블을 보호하는 보안 규정의 일부인 보안 레이블에 의해 보호됩니다. 테이블을 작성하거나 나중에 테이블을 변경할 때 보안 규정 추가를 비롯한 데이터 보호가 이루어집니다.

동일한 CREATE TABLE 또는 ALTER TABLE문의 일부로서, 보안 규정을 테이블에 추가하고 해당 테이블의 데이터를 보호할 수 있습니다.

일반적으로, 현재 LBAC 증명서에서는 해당 데이터에 대한 쓰기를 허용하지 않으므로 이러한 방법으로는 데이터를 보호할 수 없습니다.

보안 규정을 테이블에 추가

CREATE TABLE문의 SECURITY POLICY절을 사용하여 테이블을 작성할 때 테이블에 보안 규정을 추가할 수 있습니다. ALTER TABLE문의 ADD SECURITY POLICY절을 사용하여 기존 테이블에 보안 규정을 추가할 수 있습니다. 테이블에 보안 규정을 추가할 경우에는 SECADM 권한 또는 LBAC 증명서가 없어도 됩니다.

LBAC를 사용하여 보호할 수 없는 테이블 유형에는 보안 규정을 추가할 수 없습니다. LBAC를 사용하여 보호할 수 없는 테이블 유형 목록에 대한 정보는 LBAC 개요를 참조하십시오.

테이블에는 보안 규정을 두 개 이상 추가할 수 없습니다.

행 보호

테이블을 작성할 때 데이터 유형이 DB2SECURITYLABEL인 컬럼을 포함시켜 새 테이블의 행을 보호할 수 있습니다. CREATE TABLE문에서도 테이블에 보안 규정을 추가해야 합니다. 이러한 테이블을 작성할 경우에는 SECADM 권한 또는 LBAC 증명서가 없어도 됩니다.

데이터 유형이 DB2SECURITYLABEL인 컬럼을 추가할 때 기존 테이블의 행을 보호할 수 있습니다. 이러한 컬럼을 추가하려면, 테이블이 이미 보안 규정에 의해 보호되거나 컬럼을 추가하는 ALTER TABLE문에서도 테이블에 보안 규정을 추가해야 합니다. 컬럼을 추가하면, 쓰기 액세스에 대해 사용자가 보유하는 보안 레이블은 모든 기존의 행을 보호하는 데 사용됩니다. 쓰기 액세스에 필요한 보안 레이블(테이블을 보호하는 보안 규정의 일부)을 보유하지 않을 경우, 데이터 유형이 DB2SECURITYLABEL인 컬럼을 추가할 수 없습니다.

테이블에 DB2SECURITYLABEL 유형의 컬럼이 있는 경우, 해당 컬럼에 보안 레이블을 저장하여 새 데이터 행을 보호할 수 있습니다. 보호 작동 방법에 대한 세부사항은 LBAC 보호 데이터의 삽입 및 갱신에 대한 주제에 설명되어 있습니다. 컬럼 유형이 DB2SECURITYLABEL인 테이블에 행을 삽입하려면 LBAC 증명서가 있어야 합니다.

데이터 유형이 DB2SECURITYLABEL인 컬럼은 삭제하거나 다른 데이터 유형으로 변경할 수 없습니다.

컬럼 보호

CREATE TABLE문의 SECURED 컬럼 옵션을 사용하여 테이블을 적상할 때 컬럼을 보호할 수 있습니다. ALTER TABLE문의 SECURED WITH 옵션을 사용하여 기존 컬럼에 보호를 추가할 수 있습니다.

특정 보안 레이블을 사용하여 컬럼을 보호할 경우, 해당 보안 레이블에 의해 보호되는 데이터를 쓸 수 있는 LBAC 증명서가 있어야 합니다. SECADM 권한은 필요하지 않습니다.

테이블을 보호하는 보안 규정의 일부인 보안 레이블을 사용하여 컬럼을 보호할 수 있습니다. 보안 규정이 없는 테이블의 컬럼은 보호할 수 없습니다. 보안 규정을 사용하여 테이블을 보호하고 동일한 명령문으로 하나 이상의 컬럼을 보호할 수 있습니다.

숫자에 관계없이 테이블의 행을 보호할 수 있으나 둘 이상의 보안 레이블을 사용하여 컬럼을 보호할 수는 없습니다.

LBAC 보호 데이터 읽기

레이블 기반 액세스 제어(LBAC)를 사용하여 보호되는 데이터를 읽을 경우 읽기용 LBAC 증명서를 데이터를 보호하는 보안 레이블과 비교합니다. 보호 레이블이 증명서를 차단하지 않을 경우 데이터를 읽을 수 있습니다.

보호 컬럼의 경우, 보호 보안 레이블은 테이블 스키마에 정의되어 있습니다. 해당 컬럼에 대한 보호 보안 레이블은 테이블의 모든 행에 대해 동일합니다. 보호 행의 경우, 보호 보안 레이블은 행의 DB2SECURITYLABEL 유형을 가진 컬럼에 저장됩니다. 보호 보안 레이블은 테이블의 행마다 다를 수 있습니다.

LBAC 증명서를 보안 레이블과 비교하는 방법에 대한 세부사항은 LBAC 보안 레이블을 비교하는 방법에 대한 주제에 나와 있습니다.

보호 컬럼 읽기

보호 컬럼을 읽을 경우, LBAC 증명서를 컬럼을 보호하는 보안 레이블과 비교합니다. 비교 결과를 토대로 액세스가 차단되거나 승인됩니다. 액세스가 차단되면 오류가 리턴되고 명령문이 실패합니다. 차단되지 않으면 명령문이 정상적으로 처리됩니다.

LBAC 증명서에서 읽기가 허용되지 않는 컬럼을 읽으면 명령문 전체가 실패합니다.

예:

T1 테이블에는 두 개의 보호 행이 있습니다. 컬럼 C1은 보안 레벨 L1에 의해 보호됩니다. 컬럼 C2는 보안 레벨 L2를 사용하여 보호됩니다.

사용자 Jyoti는 보안 레이블 L1에는 액세스할 수 있으나 L2에는 액세스할 수 없는 읽기용 LBAC 증명서를 갖고 있습니다. Jyoti가 다음 SQL문을 발행할 경우 명령문이 실패합니다.

```
SELECT * FROM T1
```

컬럼 C2가 와일드 카드(*)의 일부로서 SELECT절에 포함되어 있기 때문에 명령문이 실패합니다.

Jyoti가 다음 SQL문을 발행할 경우 명령문이 정상적으로 실행됩니다.

```
SELECT C1 FROM T1
```

SELECT절에서 보호 컬럼은 C1뿐이므로 Jyoti의 LBAC 증명서에서는 해당 컬럼의 읽기를 허용합니다.

보호 행 읽기

사용자에게 행을 읽을 수 있는 LBAC 증명서가 없으면 해당 행이 없는 것처럼 보입니다.

보호 행을 읽을 경우, LBAC 증명서에 읽기 액세스 권한이 있는 행만 리턴됩니다. 이는 DB2SECURITYLABEL 유형을 가진 컬럼이 SELECT절의 일부가 아닌 경우에도 해당됩니다.

해당 LBAC 증명서에 따라, 각 사용자마다 보호 행이 있는 테이블에서 볼 수 있는 행이 각기 다릅니다. 예를 들어, T1에 보호 행이 있고 사용자가 다른 LBAC 증명서를 갖고 있으면 `SELECT COUNT(*) FROM T1` 명령문을 실행하는 두 사용자가 각기 다른 결과를 얻을 수 있습니다.

LBAC 증명서는 SELECT문은 물론 UPDATE 및 DELETE와 같은 SQL문에도 영향을 미칩니다. 행 읽기를 허용하는 LBAC 증명서가 없는 경우 해당 행에 영향을 미칠 수 없습니다.

예:

T1 테이블에 다음과 같은 행 및 컬럼이 있습니다. ROWSECURITYLABEL 컬럼의 데이터 유형은 DB2SECURITYLABEL입니다.

표 14.

LASTNAME	DEPTNO	ROWSECURITYLABEL
Rjaibi	55	L2
Miller	77	L1
Fielding	11	L3
Bird	55	L2

사용자 Dan이 보안 레이블 L1에 의해 보호되는 데이터는 읽을 수 있으나 L2 또는 L3에 의해 보호되지 않는 데이터는 읽을 수 없는 LBAC 증명서를 갖고 있다고 가정합니다.

Dan이 다음과 같은 SQL문을 발행합니다.

```
SELECT * FROM T1
```

SELECT문에서 Miller에 대한 행만 리턴됩니다. 오류 또는 경고 메시지는 리턴되지 않습니다.

T1 테이블에 대한 Dan의 뷰는 다음과 같습니다.

표 15.

LASTNAME	DEPTNO	ROWSECURITYLABEL
Miller	77	L1

보안 레이블에 의해 읽기 액세스가 차단되기 때문에 Rjaibi, Fielding 및 Bird에 대한 행은 리턴되지 않습니다. Dan은 이러한 행을 삭제 또는 갱신할 수 없습니다. 이 행들은 모든 집계 함수에도 포함되지 않습니다. Dan에게는 이러한 행이 존재하지 않은 것처럼 보입니다.

Dan이 다음과 같은 SQL문을 발행합니다.

```
SELECT COUNT(*) FROM T1
```

Dan은 MIller에 대한 행만 읽을 수 있기 때문에 명령문은 값 1을 리턴합니다.

보호 컬럼을 포함하는 보호 행 읽기

행에 액세스하기 전에 컬럼 액세스를 점검합니다. 읽기 액세스용 LBAC 증명서가 사용자가 선택한 컬럼 중 하나를 보호하는 보안 레이블에 의해 차단될 경우, 명령문 전체가 실패합니다. 차단되지 않을 경우, 명령문은 계속 처리되며 LBAC 증명서가 읽기 액세스를 허용하는 보안 레이블에 의해 보호되는 행만이 리턴됩니다.

예

T1 테이블의 LASTNAME 컬럼은 보안 레이블 L1에 의해 보호됩니다. DEPTNO 컬럼은 보안 레이블 L2를 사용하여 보호됩니다. ROWSECURITYLABEL 컬럼의 데이터 유형은 DB2SECURITYLABEL입니다. T1의 데이터는 다음과 같습니다.

표 16.

LASTNAME *보호 레이블 L1*	**DEPTNO** *보호 레이블 L2*	**ROWSECURITYLABEL**
Rjaibi	55	L2
Miller	77	L1
Fielding	11	L3

사용자 Sakari가 보안 레이블 L1에 의해 보호되는 데이터는 읽을 수 없으나 L2 또는 L3에 의해 보호되지 않는 데이터는 읽을 수 없는 LBAC를 갖고 있다고 가정합니다.

Sakari는 다음과 같은 SQL문을 발행합니다.

```
SELECT * FROM T1
```

SELECT절에서 DEPTNO 컬럼을 포함하는 와일드 카드(*)를 사용하기 때문에 명령문이 실패합니다. DEPTNO 컬럼은 보안 레이블 L2에 의해 보호되기 때문에 Sakari의 LBAC 증명서에서는 해당 컬럼의 읽기를 허용하지 않습니다.

Sakari가 이후에 다음과 같은 SQL문을 발행합니다.

```
SELECT LASTNAME, ROWSECURITYLABEL FROM T1
```

Select절에는 Sakari가 읽을 수 있는 컬럼만 포함되므로 명령문은 계속 처리됩니다. 그러나 다른 행이 보안 레이블 2 또는 3에 의해 보호되므로 한 행만 리턴됩니다.

표 17.

LASTNAME	ROWSECURITYLABEL
Miller	L1

LBAC 보호 데이터 삽입

보호된 컬럼에 데이터를 삽입하거나 보호된 행이 있는 테이블에 새 행을 삽입하려는 경우, LBAC 증명서가 해당 INSERT문이 처리되는 방식을 결정합니다.

보호 컬럼에 삽입

보호된 컬럼에 데이터를 삽입할 때 쓰기용 LBAC 증명서를 해당 컬럼을 보호하는 보안 레이블과 비교하여 비교 결과를 토대로 액세스가 차단되거나 승인됩니다.

두 보안 레이블을 비교하는 방법에 대한 세부사항은 LBAC 보안 레이블을 비교하는 방법에 대한 주제에 나와 있습니다.

액세스할 수 있는 경우, 명령문은 정상적으로 처리됩니다. 액세스가 차단될 경우, 삽입에 실패하며 오류가 리턴됩니다.

행을 삽입할 때 보호 컬럼에 대한 값을 제공하지 않으면 사용 가능한 경우 디폴트값이 삽입됩니다. 이는 LBAC 증명서가 해당 컬럼에 대한 쓰기 액세스를 허용하지 않을 경우에도 발생합니다. 디폴트값은 다음과 같은 경우에 사용 가능합니다.

- 컬럼이 WITH DEFAULT 옵션을 사용하여 선언됨
- 컬럼에서 컬럼을 생성함
- 컬럼이 사전 트리거를 통해 제공된 디폴트값을 갖고 있음
- 컬럼의 데이터 유형은 DB2SECURITYLABEL입니다. 이 경우 쓰기 액세스에 대해 갖고 있는 보안 레이블이 디폴트값입니다.

보호된 행에 삽입

보호된 행이 있는 테이블에 새 행을 삽입하는 경우 유형이 DB2SECURITYLABEL인 컬럼의 값은 제공하지 않아도 됩니다. 해당 컬럼의 값을 제공하지 않아도 컬럼이 자동으로 쓰기 액세스용으로 부여된 보안 레이블로 채워집니다. 쓰기 액세스용 레이블이 부여되지 않으면 오류가 리턴되고 삽입에 실패합니다.

SECLABEL과 같은 내장 함수를 사용하여 명시적으로 유형이 DB2SECURITYLABEL인 컬럼에 삽입할 보안 레이블을 제공할 수 있습니다. LBAC 증명서에서 삽입할 보안 레이블에 의해 보호되는 데이터에 대한 쓰기를 허용할 경우, 제공된 보안 레이블만이 사용됩니다.

쓸 수 없는 보안 레이블을 제공할 경우, 테이블을 보호하는 보안 규정에 따라 삽입이 실패할 수도 있고 성공할 수도 있습니다. 보안 규정에 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션이 있는 경우 삽입이 실패하고 오류가 리턴됩니다. 보안 규정에 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션이 없거나 대신 OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL 옵션이 있는 경우, 사용자가 제공하는 보안 레이블이 무시되며, 쓰기 액세스용 보안 레이블을 보유한 경우에는 해당 레이블이 대신 사용됩니다. 쓰기 액세스용 보안 레이블을 보유하지 않은 경우 오류가 리턴됩니다.

예

T1 테이블은 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션 없이 작성된 보안 규정 P1에 의해 보호됩니다. T1 테이블은 두 개의 열을 가지며 행은 없습니다. 열은 LASTNAME 및 LABEL입니다. LABEL 컬럼의 데이터 유형은 DB2SECURITYLABEL입니다.

사용자 Joe는 쓰기 액세스용 보안 레벨 L2를 보유합니다. 보안 레이블 L2를 통해 보안 레이블 L2를 사용하여 보호되는 데이터는 쓸 수 있으나 레이블 L1 또는 L3에 의해 보호되는 데이터는 쓸 수 없다고 가정합니다.

Joe가 다음과 같은 SQL문을 발행합니다.

```
INSERT INTO T1 (LASTNAME, DEPTNO) VALUES ('Rjaibi', 11)
```

INSERT문에 보안 레이블이 포함되어 있지 않기 때문에 Joe의 쓰기 액세스용 보안 레이블이 LABEL 행에 삽입됩니다.

T1 테이블은 다음과 같습니다.

표 18.

LASTNAME	LABEL
Rjaibi	L2

Joe가 다음과 같은 SQL문을 발행할 경우 Joe는 LABEL 컬럼에 삽입할 보안 레이블을 제공합니다.

```
INSERT INTO T1 VALUES ('Miller', SECLABEL_BY_NAME('P1', 'L1') )
```

명령문의 SECLABEL_BY_NAME 함수는 보안 규정 P1의 일부로서 L1이라는 보안 레이블을 리턴합니다. Joe는 L1에 의해 보호되는 데이터를 쓸 수 없으므로 L1을 LABEL 컬럼에 삽입할 수 없습니다.

T1을 보호하는 보안 규정은 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션 없이 작성되므로 Joe가 쓰기용으로 보유하는 보안 레이블이 삽입됩니다. 오류 메시지는 리턴되지 않습니다.

테이블은 다음과 같습니다.

표 19.

LASTNAME	LABEL
Rjaibi	L2
Miller	L2

테이블을 보호하는 보안 규정이 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션을 사용하여 작성된 경우, 삽입 조작은 실패하며 오류가 리턴됩니다.

다음에 Joe에게 LBAC 규칙 중 하나에 대한 면제가 부여됩니다. 새 LBAC 증명서에서 보안 레이블 L1 및 L2에 의해 보호되는 데이터에 대한 쓰기를 허용한다고 가정합니다. 쓰기 액세스용으로 Joe에게 부여된 보안 레이블은 변경되지 않고 L2를 유지합니다.

Joe가 다음과 같은 SQL문을 발행합니다.

```
INSERT INTO T1 VALUES ('Bird', SECLABEL_BY_NAME('P1', 'L1') )
```

Joe의 새 LBAC 증명서를 사용하면 보안 레이블 L1을 통해 보호되는 데이터를 쓸 수 있습니다. 따라서 L1의 삽입도 허용됩니다. 테이블은 다음과 같습니다.

표 20.

LASTNAME	LABEL
Rjaibi	L2
Miller	L2
Bird	L1

LBAC 보호 데이터 갱신

데이터를 갱신하려면 먼저 LBAC 증명서를 통해 데이터에 대한 쓰기 액세스 권한을 부여받아야 합니다. 보호된 행을 갱신할 경우, LBAC 증명서를 통해 행에 대한 읽기 액세스 권한을 부여받아야 합니다.

보호 컬럼 갱신

보호 컬럼을 갱신할 경우, LBAC 증명서를 컬럼을 보호하는 보안 레이블과 비교해야 합니다. 쓰기 액세스에 필요한 비교를 수행합니다. 쓰기 액세스가 차단될 경우 오류가 리턴되고 명령문이 실패하며, 차단되지 않을 경우 갱신이 처리됩니다.

LBAC 증명서를 보안 레이블과 비교하는 방법에 대한 세부사항은 LBAC 보안 레이블을 비교하는 방법에 대한 주제에 나와 있습니다.

예:

DEPTNO 컬럼은 보안 레이블 L2에 의해 보호되고 PAYSCALE 컬럼은 보안 레이블 L3에 의해 보호되는 T1 테이블이 있다고 가정하십시오. T1의 데이터는 다음과 같습니다.

표 21. T1 테이블

EMPNO	**LASTNAME**	**DEPTNO** *보호 레이블* **L2**	**PAYSCALE** *보호 레이블* **L3**
1	Rjaibi	11	4
2	Miller	11	7
3	Bird	11	9

사용자 Lhakpa에게는 LBAC 증명서가 없습니다. Lhakpa가 다음과 같은 SQL문을 발행합니다.

```
UPDATE T1 SET EMPNO = 4
   WHERE LASTNAME = "Bird"
```

이 명령문은 보호 컬럼을 갱신하지 않기 때문에 오류 없이 실행됩니다. T1은 이제 다음과 같습니다.

표 22. 갱신 후 T1 테이블

EMPNO	**LASTNAME**	**DEPTNO** *보호 레이블* **L2**	**PAYSCALE** *보호 레이블* **L3**
1	Rjaibi	11	4
2	Miller	11	7
4	Bird	11	9

Lhakpa는 이후에 다음과 같은 SQL문을 발행합니다.

```
UPDATE T1 SET DEPTNO = 55
   WHERE LASTNAME = "Miller"
```

DEPTNO가 보호되고 Lhakpa에게 LBAC 증명서가 없기 때문에 이 명령문은 실패하며 오류가 리턴됩니다.

Lhakpa에게 LBAC 증명서가 부여되었고 다음 표에 요약되어 있는 액세스가 허용된다고 가정합니다. 이 예에서는 이 증명서에 대한 정보와 보안 레이블에 있는 요소에 대한 세부사항은 중요하지 않습니다.

데이터를 보호하는 보안 레이블	읽기 가능 여부	쓰기 가능 여부
L2	아니오	예
L3	아니오	아니오

Lhakpa가 다음과 같은 SQL문을 다시 발행합니다.

```
UPDATE T1 SET DEPTNO = 55
   WHERE LASTNAME = "Miller"
```

이 경우, Lhakpa는 LBAC 증명서를 통해 DEPTNO 컬럼을 보호하는 보안 레이블에 의해 보호되는 데이터에 대한 쓰기가 허용되기 때문에 명령문은 오류 없이 실행됩니다. 동일한 컬럼을 읽을 수 있는지 여부는 문제되지 않습니다. T1의 데이터는 다음과 같습니다.

표 23. 두 번째 갱신 후 T1 테이블

EMPNO	**LASTNAME**	**DEPTNO** *보호 레이블* **L2**	**PAYSCALE** *보호 레이블* **L3**
1	Rjaibi	11	4
2	Miller	55	7
4	Bird	11	9

이후에 Lhakpa는 다음과 같은 SQL문을 발행합니다.

```
UPDATE T1 SET DEPTNO = 55, PAYSCALE = 4
   WHERE LASTNAME = "Bird"
```

PAYSCALE 컬럼이 보안 레이블 L3에 의해 보호되므로 Lhakpa의 LBAC 증명서에서는 데이터에 대한 쓰기를 허용하지 않습니다. Lhakpa는 컬럼에 데이터를 쓸 수 없기 때문에 갱신에 실패하며 데이터는 변경되지 않습니다.

보호 행 갱신

LBAC 증명서에서 행 읽기를 허용하지 않으면 행이 없는 것처럼 보이므로 해당 행을 갱신할 방법이 없습니다. 읽을 수 있는 행의 경우, 행을 갱신하려면 행에 값을 쓸 수도 있어야 합니다.

행을 갱신할 경우, 쓰기용 LBAC 증명서를 행을 보호하는 보안 레이블과 비교해야 합니다. 쓰기 액세스가 차단된 경우, 갱신에 실패하고 오류가 리턴됩니다. 쓰기 액세스가 차단되지 않은 경우 갱신이 계속됩니다.

수행되는 갱신은 보호되지 않는 행을 갱신하는 것과 동일한 방법으로 수행됩니다. 단, 데이터 유형이 DB2SECURITYLABEL인 컬럼은 다르게 처리됩니다. 해당 컬럼의 값을 명시적으로 설정하지 않으면 쓰기 액세스용으로 보유하고 있는 보안 레이블로 자동 설정됩니다. 쓰기 액세스에 대한 보안 레이블이 없으면 오류가 리턴되고 명령문이 실패합니다.

갱신 시 명시적으로 데이터 유형이 DB2SECURITYLABEL인 컬럼이 설정되면 LBAC 증명서를 다시 점검합니다. 수행하려는 갱신이 현재 LBAC 증명서에서 쓰기를 허용하

지 않는 행을 작성할 경우, 테이블을 보호하는 보안 규정에 따라 발생하는 결과가 달라집니다. 보안 규정에 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션이 있는 경우 갱신이 실패하고 오류가 리턴됩니다. 보안 규정에 RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL 옵션이 없거나 대신 OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL 옵션이 있는 경우, 사용자가 제공하는 보안 레이블이 무시되며, 쓰기 액세스용 보안 레이블을 보유한 경우에는 해당 레이블이 대신 사용됩니다. 쓰기 액세스용 보안 레이블을 보유하지 않은 경우 오류가 리턴됩니다.

예:

T1 테이블이 P1이라는 보안 규정에 의해 보호되고 데이터 유형이 DB2SECURITYLABEL인 LABEL이라는 컬럼을 갖고 있다고 가정하십시오.

T1의 데이터는 다음과 같습니다.

표 24. T1 테이블

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	11	L1
2	Miller	11	L2
3	Bird	11	L3

사용자 Jenni에게 보안 레이블 L0 및 L1에 의해 보호되는 데이터의 읽기 및 쓰기는 허용하나 다른 보안 레이블에 의해 보호되는 데이터의 읽기 및 쓰기는 허용하지 않는 LBAC 증명서가 있다고 가정합니다. Jenni가 읽기 및 쓰기 모두에 대해 갖고 있는 보안 레이블은 L0입니다. 이 예에서는 전체 증명서와 레이블에 있는 요소에 대한 세부사항은 중요하지 않습니다.

Jenni는 다음과 같은 SQL문을 발행합니다.

```
SELECT * FROM T1
```

테이블에는 한 행만 표시됩니다.

표 25. Jenni의 SELECT 쿼리 결과

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	11	L1

Jenni의 LBAC를 통해서는 이러한 행을 읽을 수 없기 때문에 레이블 L2 및 L3에 의해 보호되는 행은 결과 세트에 포함되어 있지 않습니다. Jenni에게는 이러한 행이 존재하지 않는 것처럼 보입니다.

Jenni는 다음과 같은 SQL문을 발행합니다.

```
UPDATE T1 SET DEPTNO = 44 WHERE DEPTNO = 11;
SELECT * FROM T1;
```

쿼리에서 리턴되는 결과 세트는 다음과 같습니다.

표 26. Jenni의 UPDATE & SELECT 쿼리 결과

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L0

테이블의 실제 데이터는 다음과 같습니다.

표 27. T1 테이블

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L0
2	Miller	11	L2
3	Bird	11	L3

명령문은 오류 없이 실행되나 첫 번째 행에만 영향을 받습니다. 두 번째 및 세 번째 행은 Jenni가 읽을 수 없으므로 WHERE절의 조건을 만족하더라도 명령문에 의해 갱신 대상으로 선택되지 않습니다.

UPDATE문에 해당 컬럼이 명시적으로 설정되어 있지 않아도 갱신된 행의 LABEL 컬럼의 값이 변경되었음을 주의하십시오. 컬럼은 Jenni가 쓰기에 대해 갖고 있는 보안 레이블로 설정됩니다.

지금 Jenni에게는 다른 보안 레이블에 의해 보호되는 데이터를 읽을 수 있는 LBAC 증명서가 부여되었습니다. 쓰기에 대한 LBAC 증명서는 변경되지 않았습니다. 현재 L0 및 L1에 의해 보호되는 데이터만 쓸 수 있습니다.

Jenni는 다시 다음과 같은 SQL문을 발행합니다.

```
UPDATE T1 SET DEPTNO = 44 WHERE DEPTNO = 11
```

이 경우 두 번째 및 세 번째 행으로 인해 갱신에 실패합니다. Jenni는 이 행을 읽을 수 있으므로 명령문에 의해 갱신 대상으로 선택됩니다. 그러나 이 행은 보안 레이블 L2 및 L3에 의해 보호되기 때문에 값을 쓸 수는 없습니다. 갱신은 발생하지 않으며 오류가 리턴됩니다.

Jenni는 이제 이 SQL문을 발행합니다.

```
UPDATE T1
 SET DEPTNO = 55, LABEL = SECLABEL_BY_NAME( 'P1', 'L2' )
 WHERE LASTNAME = "Rjaibi"
```

이 명령문의 SECLABEL_BY_NAME 함수는 L2라는 보안 레이블을 리턴합니다. Jenni가 명시적으로 첫 번째 행을 보호하는 보안 레이블을 설정합니다. Jenni의 LBAC 증명서에서는 첫 번째 행의 읽기를 허용하므로 갱신 대상으로 선택됩니다. Jenni의 LBAC 증명서에서는 보안 레이블 L0에 의해 보호되는 행에 대한 쓰기를 허용하므로 행을 갱신할 수 있습니다. 그러나 Jenni의 LBAC 증명서에서는 보안 레이블 L2에 의해 보호

되는 행에 대한 쓰기를 허용하지 않으므로 LABEL 컬럼을 해당 값으로 설정할 수 없습니다. 명령문은 실패하며 오류가 발생합니다. 행의 컬럼은 갱신되지 않습니다.

Jenni는 이제 이 SQL문을 발행합니다.

```
UPDATE T1 SET LABEL = SECLABEL_BY_NAME( 'P1', 'L1' ) WHERE LASTNAME = "Rjaibi"
```

Jenni가 보안 레이블 L1에 의해 보호되는 행을 쓸 수 있기 때문에 명령문이 완료됩니다.

T1은 이제 다음과 같습니다.

표 28. T1 테이블

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L1
2	Miller	11	L2
3	Bird	11	L3

보호 컬럼을 포함하는 보호 행 갱신

보호 행을 사용하여 테이블의 보호 컬럼을 갱신할 경우, LBAC 증명서에서는 갱신의 영향을 받는 모든 보호 컬럼에 대한 쓰기가 허용되어야 합니다. 그렇지 않으면 갱신에 실패하며 오류가 리턴됩니다. 이에 대한 내용은 앞의 ″보호 컬럼 갱신″ 절에 설명되어 있습니다. 갱신의 영향을 받는 모든 보호 컬럼을 갱신할 수 있는 경우, LBAC 증명서에서 읽기 및 쓰기가 허용된 행만 갱신할 수 있습니다. 이에 대한 내용은 앞의 ″보호 행 갱신″ 절에 설명되어 있습니다. 데이터 유형이 DB2SECURITYLABEL인 컬럼은 갱신이 보호 컬럼에 영향을 미치는지 여부에 따라 동일하게 처리됩니다.

데이터 유형이 DB2SECURITYLABEL인 컬럼 자체가 보호 컬럼인 경우 LBAC 증명서에서는 해당 컬럼에 대한 쓰기를 허용합니다. 그렇지 않으면 테이블의 행을 갱신할 수 없습니다.

LBAC 보호 데이터 삭제(delete 또는 drop)

LBAC에서 보호하는 테이블의 데이터를 삭제할 수 있는지 여부는 LBAC 증명서에 따라 달라집니다.

보호 행 삭제

LBAC 증명서가 행을 읽도록 허용하지 않는 경우 행이 존재하지 않는 것과 마찬가지이므로 해당 행을 삭제할 수 없습니다. 읽을 수 있는 행을 삭제하려면 LBAC 증명서도 행에 값을 쓰도록 허용해야 합니다. 보호 컬럼이 있는 테이블에서 임의의 행을 삭제하려면 테이블의 모든 보호 컬럼에 쓸 수 있는 LBAC 증명서가 있어야 합니다.

행을 삭제할 경우 쓰기용 LBAC 증명서가 행을 보호하는 보안 레이블과 비교됩니다. 보호 보안 레이블이 LBAC 증명서에 의해 쓰기 액세스를 블록하는 경우 DELETE문이 실패하고 오류가 리턴되며 행은 삭제되지 않습니다.

예

보호 테이블 T1에 다음과 같은 행이 있습니다.

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2
Fielding	77	L3

사용자 Pat가 다음 표에 요약된 대로 액세스 권한을 갖는 LBAC 증명서를 갖는다고 가정합니다.

보안 레이블	읽기 액세스	쓰기 액세스
L1	예	예
L2	예	아니오
L3	아니오	아니오

LBAC 증명서 및 보안 레이블의 정확한 세부사항은 이 예에서 중요하지 않습니다.

Pat가 다음 SQL문을 발행합니다.

```
SELECT * FROM T1 WHERE DEPTNO != 999
```

명령문이 실행되고 다음 결과 세트를 리턴합니다.

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2

T1의 마지막 행은 Pat가 해당 행에 대한 읽기 액세스 권한이 없기 때문에 결과에 포함되지 않습니다. 해당 행은 Pat에게 존재하지 않는 것과 마찬가지입니다.

Pat가 다음 SQL문을 발행합니다.

```
DELETE FROM T1 WHERE DEPTNO != 999
```

Pat는 둘 다 L2에 의해 보호되는 첫 번째 또는 세 번째 행에 대한 쓰기 액세스 권한이 없습니다. 따라서 해당 행을 읽을 수 있어도 삭제할 수는 없습니다. DELETE문은 실패하고 어떤 행도 삭제되지 않습니다.

Pat가 다음 SQL문을 발행합니다.

```
DELETE FROM T1 WHERE DEPTNO = 77;
```

이 명령문은 Pat가 LASTNAME 컬럼에서 Miller를 갖는 행에 쓸 수 있기 때문에 성공합니다. 그 행이 명령문에 의해 선택되는 유일한 행입니다. LASTNAME 컬럼에 Fielding을 갖는 행은 Pat의 LBAC 증명서가 해당 행을 읽도록 허용하지 않기 때문에 선택되지 않습니다. 이 행은 삭제에 대해 전혀 고려되지 않으므로 오류가 발생하지 않습니다.

이제 테이블의 실제 행은 다음과 비슷하게 보입니다.

LASTNAME	**DEPTNO**	**LABEL**
Rjaibi	55	L2
Bird	55	L2
Fielding	77	L3

보호 컬럼을 갖는 행 삭제

보호 컬럼이 있는 테이블에서 임의의 행을 삭제하려면 테이블의 모든 보호 컬럼에 쓸 수 있는 LBAC 증명서가 있어야 합니다. 테이블에 LBAC 증명서에서 쓰도록 허용하지 않는 임의의 행이 있는 경우 삭제는 실패하며 오류가 리턴됩니다.

테이블에 보호 컬럼과 보호 열이 둘 다 있고 특정 행을 삭제하려는 경우 테이블의 모든 보호 컬럼에 쓸 수 있으며 삭제하려는 행에서 읽고 행에 쓸 수 있는 LBAC 증명서가 있어야 합니다.

예

보호 테이블 T1에서 컬럼 DEPTNO가 보안 레이블 L2에 의해 보호됩니다. T1은 다음 행을 포함하고 있습니다.

LASTNAME	**DEPTNO** **L2에** *의해 보호됨*	**LABEL**
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2
Fielding	77	L3

사용자 Benny가 이 테이블에 요약되는 액세스를 허용하는 LBAC 증명서를 갖고 있다고 가정하십시오.

보안 레이블	읽기 액세스	쓰기 액세스
L1	예	예

보안 레이블	읽기 액세스	쓰기 액세스
L2	예	아니오
L3	아니오	아니오

LBAC 증명서 및 보안 레이블의 정확한 세부사항은 이 예에서 중요하지 않습니다.

Benny가 다음 SQL문을 발행합니다.

```
DELETE FROM T1 WHERE DEPTNO = 77
```

Benny가 컬럼 DEPTNO에 대한 쓰기 액세스가 없기 때문에 명령문은 실패합니다.

이제 Benny가 이 테이블에 요약된 대로 액세스할 수 있도록 그의 LBAC 증명서가 변경됩니다.

보안 레이블	읽기 액세스	쓰기 액세스
L1	예	예
L2	예	예
L3	예	아니오

Benny가 다음 SQL문을 다시 발행합니다.

```
DELETE FROM T1 WHERE DEPTNO = 77
```

이번에는 Benny가 컬럼 DEPTNO에 대한 쓰기 액세스를 가지므로 삭제가 계속됩니다. 삭제 명령문은 LASTNAME 컬럼에 값 Miller를 갖는 행만을 선택합니다. LASTNAME 컬럼에 값 Fielding을 갖는 행은 Benny의 LBAC 증명서가 해당 행을 읽도록 허용하지 않기 때문에 선택되지 않습니다. 행이 명령에 의해 삭제되도록 선택되지 않기 때문에 Benny가 행에 쓸 수 없다는 것은 문제가 되지 않습니다.

선택된 하나의 행이 보안 레이블 L1에 의해 보호됩니다. Benny의 LBAC 증명서는 L1에 의해 보호되는 데이터에 쓸 수 있도록 허용하므로 삭제가 성공합니다.

테이블 T1의 실제 행은 이제 다음과 비슷합니다.

LASTNAME	**DEPTNO** **L2에** *의해 보호됨*	**LABEL**
Rjaibi	55	L2
Bird	55	L2
Fielding	77	L3

보호 데이터 삭제

사용자의 LBAC 증명서가 보안 레이블에 의해 보호되는 컬럼에 쓰도록 허용하지 않으면 해당 컬럼을 삭제할 수 없습니다.

DB2SECURITYLABEL의 데이터 유형을 갖는 컬럼은 테이블에서 삭제할 수 없습니다. 제거하려면 먼저 테이블에서 보안 규정을 삭제해야 합니다. 보안 규정을 삭제할 때 테이블은 더 이상 LBAC로 보호되지 않으며 컬럼의 데이터 유형이 자동으로 DB2SECURITYLABEL에서 VARCHAR(128) FOR BIT DATA로 변경됩니다. 그런 다음 컬럼을 삭제할 수 있습니다.

LBAC 증명서는 보호 데이터가 들어 있는 전체 테이블 또는 데이터베이스를 삭제하지 못하게 막지 않습니다. 정상적으로 테이블 또는 데이터베이스를 삭제할 사용 권한이 있는 경우, 데이터베이스에 보호되는 데이터가 있는 경우에도 삭제하는 데 LBAC 증명서가 필요하지 않습니다.

데이터에서 LBAC 보호 제거

테이블의 보안 규정을 제거하려면 SECADM 권한이 있어야 합니다. 테이블에서 보안 규정을 제거하기 위해 ALTER TABLE문의 DROP SECURITY POLICY절을 사용하십시오. 그러면 테이블의 모든 행과 컬럼의 보호가 자동으로 제거됩니다.

행에서 보호 제거

보호 행이 있는 테이블의 모든 행은 보안 레이블에 의해 보호되어야 합니다. 개별 행에서 LBAC 보호를 제거할 방법을 없습니다.

테이블에서 보안 규정을 제거하지 않고는 DB2SECURITYLABEL 유형의 컬럼을 변경 또는 제거할 수 없습니다.

컬럼에서 보호 제거

ALTER TABLE SQL문의 DROP COLUMN SECURITY절을 사용하여 컬럼 보호를 제거할 수 있습니다. 컬럼에서 보호를 제거하려면 테이블 변경에 필요한 일반 특권 및 권한을 비롯하여 해당 컬럼을 읽고 쓸 수 있는 LBAC 권한이 있어야 합니다.

제 5 장 보안 정보에 시스템 카탈로그 사용

각 데이터베이스에 대한 정보는 자동으로 시스템 카탈로그라고 하는 뷰 세트에 유지보수됩니다. 이 시스템 카탈로그는 데이터베이스가 작성될 때 작성됩니다. 시스템 카탈로그에서는 테이블, 컬럼, 인덱스, 프로그램, 특권 및 기타 오브젝트에 대해 설명합니다.

다음 보기 및 테이블 함수에는 사용자 보유 특권, 특권 부여 사용자 ID, 오브젝트 소유권에 대한 정보가 있습니다.

SYSCAT.COLAUTH
컬럼 특권 나열

SYSCAT.DBAUTH
데이터베이스 특권 나열

SYSCAT.INDEXAUTH
인덱스 특권을 나열합니다.

SYSCAT.MODULEAUTH
모듈 특권을 나열합니다.

SYSCAT.PACKAGEAUTH
패키지 특권을 나열합니다.

SYSCAT.PASSTHRUAUTH
서버 특권을 나열합니다.

SYSCAT.ROLEAUTH
역할 특권을 나열합니다.

SYSCAT.ROUTINEAUTH
루틴(함수, 메소드 및 스토어드 프로시저) 특권을 나열합니다.

SYSCAT.SCHEMAAUTH
스키마 특권을 나열합니다.

SYSCAT.SEQUENCEAUTH
시퀀스 특권을 나열합니다.

SYSCAT.SURROGATEAUTHIDS
다른 권한 부여 ID가 대리할 수 있는 권한 부여 ID 나열

SYSCAT.TABAUTH
테이블 및 뷰 특권 나열

SYSCAT.TBSPACEAUTH
테이블 스페이스 특권을 나열합니다.

SYSCAT.VARIABLEAUTH
변수 특권을 나열합니다.

SYSCAT.WORKLOADAUTH
워크로드 특권을 나열합니다.

SYSCAT.XSROBJECTAUTH
XSR 오브젝트 특권을 나열합니다.

시스템이 사용자에게 권한 부여한 특권은 SYSIBM을 권한 준 사용자로 갖게 됩니다. SYSADM, SYSMAINT SYSCTRL 및 SYSMON은 시스템 카탈로그에 나열되지 않습니다.

CREATE 및 GRANT문은 시스템 카탈로그에 특권을 위치시킵니다. ACCESSCTRL 및 SECADM 권한을 보유한 사용자는 시스템 카탈로그 뷰에 대한 SELECT 특권을 부여하고 취소할 수 있습니다.

권한 부여된 특권을 사용하여 권한 부여 이름 검색

PRIVILEGES 및 기타 관리 뷰를 사용하여 데이터베이스에 특권의 권한을 부여한 권한 부여 이름에 대한 정보를 검색할 수 있습니다.

예를 들어, 다음 쿼리는 모든 명시적 특권 및 권한 부여된 권한 부여 ID를 검색하고 추가로 PRIVILEGES 관리 뷰의 기타 정보까지 검색합니다.

```
SELECT AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE FROM SYSIBMADM.PRIVILEGES
```

다음 쿼리는 AUTHORIZATIONIDS 관리 뷰를 사용하여 특권 또는 권한 부여된 권한 부여 ID를 모두 찾아 해당 유형을 표시합니다.

```
SELECT AUTHID, AUTHIDTYPE FROM SYSIBMADM.AUTHORIZATIONIDS
```

또한 SYSIBMADM.OBJECTOWNERS 관리 뷰 및 SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID 테이블 함수를 사용하여 보안 관련 정보를 찾을 수 있습니다.

버전 9.1 이전에는 단일 시스템 카탈로그 뷰에 모든 특권에 관한 정보가 포함되지 않았습니다. 버전 9.1 이전 릴리스의 경우, 다음 명령문에서는 특권을 가진 모든 권한 부여 이름을 검색합니다.

```
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'DATABASE' FROM SYSCAT.DBAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'TABLE   ' FROM SYSCAT.TABAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'PACKAGE ' FROM SYSCAT.PACKAGEAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'INDEX   ' FROM SYSCAT.INDEXAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'COLUMN  ' FROM SYSCAT.COLAUTH
UNION
```

```
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SCHEMA  ' FROM SYSCAT.SCHEMAAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SERVER  ' FROM SYSCAT.PASSTHRUAUTH
ORDER BY GRANTEE, GRANTEETYPE, 3
```

정기적으로, 이 명령문에 의해 검색되는 목록은 시스템 보안 기능에 정의된 사용자 및 그룹 이름의 목록과 비교되어야 합니다. 그러면 더 이상 유효하지 않은 해당 권한 부여 이름을 식별할 수 있습니다.

주: 리모트 데이터베이스 클라이언트를 지원하고 있는 경우, 리모트 클라이언트에만 권한 부여 이름이 정의되고 데이터베이스 서버 머신에는 정의되지 않도록 할 수 있습니다.

DBADM 권한이 있는 모든 이름 검색

다음 명령문은 직접적으로 DBADM 권한이 권한 부여된 모든 권한 부여 이름을 검색합니다.

```
SELECT DISTINCT GRANTEE, GRANTEETYPE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'
```

테이블에 액세스할 수 있는 권한이 부여된 이름 검색

PRIVILEGES 및 기타 관리 뷰를 사용하여 데이터베이스에 특권의 권한을 부여한 권한 부여 이름에 대한 정보를 검색할 수 있습니다.

다음 명령문에서는 JAMES 규정자로 테이블 EMPLOYEE에 액세스하도록 직접 권한이 부여된 모든 권한 부여 이름(및 해당 유형)을 검색합니다.

```
SELECT DISTINCT AUTHID, AUTHIDTYPE FROM SYSIBMADM.PRIVILEGES WHERE OBJECTNAME =
'EMPLOYEE' AND OBJECTSCHEMA = 'JAMES'
```

버전 9.1 이전 릴리스의 경우, 다음 쿼리가 동일한 정보를 검색합니다.

```
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
   WHERE TABNAME = 'EMPLOYEE'
     AND TABSCHEMA = 'JAMES'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
   WHERE TABNAME = 'EMPLOYEE'
     AND TABSCHEMA = 'JAMES'
```

규정자 JAMES로 테이블 EMPLOYEE를 갱신할 수 있는 사용자를 알아내려면, 다음 명령문을 발행하십시오.

```
 SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
   WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
      (CONTROLAUTH  = 'Y' OR
       UPDATEAUTH IN ('G','Y'))
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.DBAUTH
   WHERE DBADMAUTH = 'Y'
```

```
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
   WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
   PRIVTYPE = 'U'
```

이는 CONTROL 또는 UPDATE 특권이 직접 권한 부여된 이름뿐만 아니라, DBADM 권한을 갖는 모든 권한 부여 이름도 검색합니다.

권한 부여 이름 중 일부는 개별 사용자가 아닌 그룹 ID일 수도 있음을 기억하십시오.

사용자에게 권한 부여된 모든 특권 검색

시스템 카탈로그 뷰에 대해 쿼리함으로써, 사용자가 보유하고 있는 특권의 목록과 다른 사용자에게 권한 부여한 특권의 목록을 검색할 수 있습니다.

PRIVILEGES 및 기타 관리 뷰를 사용하여 데이터베이스에 특권의 권한을 부여한 권한 부여 이름에 대한 정보를 검색할 수 있습니다.예를 들어, 다음 쿼리는 현재 세션 권한 부여 ID에 권한 부여된 모든 특권을 검색합니다.

```
SELECT * FROM SYSIBMADM.PRIVILEGES
 WHERE AUTHID = SESSION_USER AND AUTHIDTYPE = 'U'
```

이 명령문에서 SESSION_USER 키워드는 특수 레지스터로 현재 사용자의 권한 부여 이름 값과 동일합니다.

다음 예에서는 버전 9.1 이전의 릴리스에 대한 유사한 정보를 제공합니다. 예를 들어, 다음 명령문에서는 개별 권한 부여 이름(JAMES)에 직접 권한 부여된 데이터베이스 특권의 목록을 검색합니다.

```
SELECT * FROM SYSCAT.DBAUTH
 WHERE GRANTEE = 'JAMES' AND GRANTEETYPE = 'U'
```

다음 명령문에서는 사용자(JAMES)에 의해 직접 권한 부여된 테이블 특권의 목록을 검색합니다.

```
SELECT * FROM SYSCAT.TABAUTH
 WHERE GRANTOR  = 'JAMES'
```

다음 명령문에서는 사용자(JAMES)에 의해 직접 권한 부여된 개별 컬럼 특권의 목록을 검색합니다.

```
SELECT * FROM SYSCAT.COLAUTH
   WHERE GRANTOR  = 'JAMES'
```

시스템 카탈로그 뷰 보안

시스템 카탈로그 뷰는 데이터베이스의 모든 오브젝트를 설명하기 때문에, 중요한 데이터가 있는 경우 시스템 카탈로그 뷰의 액세스를 제한할 수 있습니다.

다음 권한은 모든 카탈로그 테이블에 SELECT 특권을 가집니다.

- ACCESSCTRL
- DATAACCESS
- DBADM
- SECADM
- SQLADM

또한 다음 인스턴스 레벨 권한을 통해 SYSCAT.BUFFERPOOLS, SYSCAT.DBPARTITIONGROUPS, SYSCAT.DBPARTITIONGROUPDEF, SYSCAT.PACKAGES 및 SYSCAT.TABLES 중에서 선택할 수 있습니다.

- SYSADM
- SYSCTRL
- SYSMAINT
- SYSMON

CREATE DATABASE ... RESTRICTIVE 명령을 사용하여 PUBLIC에 특권이 자동으로 권한 부여되지 않는 데이터베이스를 작성할 수 있습니다. 이 경우 다음 일반 디폴트 권한 부여 조치가 발생하지 않습니다.

- CREATETAB
- BINDADD
- CONNECT
- IMPLICIT_SCHEMA
- 스키마 SQLJ에 있는 모든 프로시저에서 GRANT로 EXECUTE
- SYSPROC 스키마의 모든 함수 및 프로시저에 대한 EXECUTE with GRANT
- NULLID 스키마에 작성된 모든 패키지에 대한 BIND
- NULLID 스키마에 작성된 모든 패키지에서 EXECUTE
- SQLJ 스키마에 대한 CREATEIN
- NULLID 스키마에 대한 CREATEIN
- 테이블 스페이스 USERSPACE1에서의 USE
- SYSIBM 카탈로그 테이블에 대한 SELECT 액세스
- SYSCAT 카탈로그 뷰에 대한 SELECT 액세스
- SYSIBMADM 관리 뷰에 대한 SELECT 액세스

- SYSSTAT 카탈로그 뷰에 대한 SELECT
- SYSSTAT 카탈로그 뷰에 대한 UPDATE 액세스

RESTRICTIVE 옵션을 사용하여 데이터베이스를 작성했고 PUBLIC에 권한 부여된 사용 권한이 제한되었는지를 점검하려는 경우, 다음 쿼리를 발행하여 어떤 스키마에 PUBLIC이 액세스할 수 있는지 확인하십시오.

```
SELECT DISTINCT OBJECTSCHEMA FROM SYSIBMADM.PRIVILEGES WHERE AUTHID='PUBLIC'

OBJECTSCHEMA
------------
SYSFUN
SYSIBM
SYSPROC
```

무슨 액세스가 여전히 SYSIBM에 대해 PUBLIC에 있는지를 보려면 다음 쿼리를 발행하여 SYSIBM에 무슨 특권이 권한 부여되었는지 점검할 수 있습니다. 특정 프로시저 및 함수에서 EXECUTE만 권한 부여되었음을 결과가 표시합니다.

```
SELECT * FROM SYSIBMADM.PRIVILEGES WHERE OBJECTSCHEMA = 'SYSIBM'

AUTHID       AUTHIDTYPE  PRIVILEGE   GRANTABLE  OBJECTNAME         OBJECTSCHEMA    OBJECTTYPE
---------... ----------  ----------  ---------  ---------------... ------------... ----------
PUBLIC       G           EXECUTE     N          SQL060207192129400 SYSPROC         FUNCTION
PUBLIC       G           EXECUTE     N          SQL060207192129700 SYSPROC         FUNCTION
PUBLIC       G           EXECUTE     N          SQL060207192129701 SYSPROC
...
PUBLIC       G           EXECUTE     Y          TABLES             SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          TABLEPRIVILEGES    SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          STATISTICS         SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SPECIALCOLUMNS     SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PROCEDURES         SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PROCEDURECOLS      SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PRIMARYKEYS        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          FOREIGNKEYS        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          COLUMNS            SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          COLPRIVILEGES      SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          UDTS               SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          GETTYPEINFO        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SQLCAMESSAGE       SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SQLCAMESSAGECCSID  SYSIBM          PROCEDURE
```

주: SYSIBMADM.PRIVILEGES 관리 뷰는 버전 9.1의 DB2 데이터베이스 관리 프로그램부터 사용할 수 있습니다.

버전 9.1 이전 릴리스의 DB2 데이터베이스 관리 프로그램의 경우, 데이터베이스가 작성되는 동안 시스템 카탈로그 뷰의 SELECT 특권이 PUBLIC에 권한 부여됩니다. 대부분의 경우, 보안 문제점이 나타나지 않습니다. 그러나 매우 중요한 데이터의 경우에 이들 테이블이 데이터베이스에 있는 모든 오브젝트를 서술하기 때문에, 적절하지 않을 수도 있습니다. 이 경우, PUBLIC으로부터 SELECT 특권을 권한 취소하고, 특정 사용자에게 필요한 SELECT 특권을 권한 부여하는 방법을 고려해 보십시오. 시스템 카탈

로그 뷰에 SELECT 권한 부여 및 권한 취소 방법은 모든 뷰에서 동일하지만, 이를 수행할 수 있는 ACCESSCTRL 또는 SECADM 권한을 가지고 있어야 합니다.

다른 사용자가 액세스한 오브젝트가 무엇인지 어떤 사용자도 알 수 없도록 하려면, 다음 카탈로그 및 관리 뷰에 대한 액세스를 제한을 고려하십시오.

- SYSCAT.COLAUTH
- SYSCAT.DBAUTH
- SYSCAT.INDEXAUTH
- SYSCAT.PACKAGEAUTH
- SYSCAT.PASSTHRUAUTH
- SYSCAT.ROUTINEAUTH
- SYSCAT.SCHEMAAUTH
- SYSCAT.SECURITYLABELACCESS
- SYSCAT.SECURITYPOLICYEXEMPTIONS
- SYSCAT.SEQUENCEAUTH
- SYSCAT.SURROGATEAUTHIDS
- SYSCAT.TABAUTH
- SYSCAT.TBSPACEAUTH
- SYSCAT.XSROBJECTAUTH
- SYSIBMADM.AUTHORIZATIONIDS
- SYSIBMADM.OBJECTOWNERS
- SYSIBMADM.PRIVILEGES

이렇게 하면 데이터베이스에 액세스하는 모든 사용자가 사용자 특권에 대한 정보를 사용할 수 없게 됩니다.

또한 통계가 수집되는 컬럼도 조사해야 합니다. 시스템 카탈로그에 기록된 일부 통계에는 사용자의 환경에서 중요한 데이터가 될 수 있는 데이터 값이 들어 있습니다. 이러한 통계에 관련 데이터가 들어 있으면 SYSCAT.COLUMNS 및 SYSCAT.COLDIST 카탈로그 뷰에 대해 PUBLIC으로부터 SELECT 특권을 취소할 수도 있습니다.

시스템 카탈로그 뷰로의 액세스를 제한하려는 경우, 뷰를 정의하여 각 권한 부여 이름이 자체 특권 정보를 검색하도록 할 수 있습니다.

예를 들어, 다음 뷰 MYSELECTS에는 사용자의 권한 부여 이름이 SELECT 특권에 직접 권한 부여된 모든 테이블의 이름 및 소유자가 있습니다.

```
CREATE VIEW MYSELECTS AS
   SELECT TABSCHEMA, TABNAME FROM SYSCAT.TABAUTH
   WHERE GRANTEETYPE = 'U'
     AND GRANTEE = USER
     AND SELECTAUTH = 'Y'
```

이 명령문에서 키워드 USER는 현재 세션 권한 부여 이름 값과 같습니다.

다음 명령문에서는 모든 권한 부여 이름에 사용 가능한 뷰를 작성합니다.

```
GRANT SELECT ON TABLE MYSELECTS TO PUBLIC
```

그리고 마지막으로, 뷰 및 기본 테이블에 대한 SELECT 특권을 권한 취소해야 합니다.

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC

REVOKE SELECT ON TABLE SYSIBM.SYSTABAUTH FROM PUBLIC
```

제 6 장 방화벽 지원

*방화벽*은 시스템 또는 네트워크에 대한 무단 액세스를 방지하는 데 사용되는 관련 프로그램 세트로 네트워크 게이트웨이에 위치합니다.

방화벽에는 다음의 네 가지 유형이 있습니다.

1. 네트워크 레벨, 패킷 필터 또는 스크리닝 라우터 방화벽
2. 클래식 응용프로그램 레벨 프록시 방화벽
3. 회선 레벨 또는 투명한 프록시 방화벽
4. SMLI(Stateful Multi-Layer Inspection) 방화벽

위에 나열된 방화벽 유형 중 하나가 통합된 기존의 방화벽 제품들이 있습니다. 위 유형의 일부가 조합되어 통합된 다른 방화벽 제품들도 많이 있습니다.

스크리닝 라우터 방화벽

스크리닝 라우터 방화벽은 네트워크 레벨 또는 패킷 필터 방화벽이라고도 합니다. 이러한 방화벽은 들어오는 패킷을 프로토콜 속성으로 스크리닝함으로써 작동합니다. 스크리닝되는 프로토콜 속성에는 소스 또는 목적지 주소, 프로토콜 유형, 소스 또는 목적지 포트, 또는 그 외의 몇 가지 프로토콜 특정 속성들이 포함될 수 있습니다.

모든 방화벽 솔루션(SOCKS 제외)의 경우 DB2 데이터베이스에서 사용하는 모든 포트가 수신 및 전송 패킷에 대해 열려 있는지 확인해야 합니다. DB2 데이터베이스에서는 DB2 데이터베이스 도구에서 사용하는 DAS(DB2 Administration Server)에 대해 포트 523을 사용합니다. 서비스 파일을 사용하여 서버 데이터베이스 관리 프로그램 구성 파일에 있는 서비스 이름을 해당 포트 번호로 맵핑함으로써 모든 서버 인스턴스에서 사용하는 포트를 결정하십시오.

응용프로그램 프록시 방화벽

프록시 또는 프록시 서버는 웹 클라이언트와 웹 서버 사이에서 중개자 역할을 하는 기술입니다. 프록시 방화벽은 클라이언트로부터의 요청에 대한 하나의 게이트웨이로서 동작합니다.

방화벽이 클라이언트 요청을 수신하면 프록시 소프트웨어가 최종 서버 목적지 주소를 판별합니다. 응용프로그램 프록시가 주소를 변환하고 필요한 경우 추가 액세스 제어 점검 및 로그를 수행한 다음 클라이언트를 대신하여 서버에 연결합니다.

방화벽 머신의 DB2 Connect 제품은 목적지 서버에 대한 프록시의 역할을 할 수 있습니다. 또한, 최종 목적지 서버에 대한 홉 서버의 역할을 하는 방화벽의 DB2 데이터베이스 서버는 응용프로그램 프록시처럼 작동합니다.

회선 레벨 방화벽

회선 레벨 방화벽은 투명한 프록시 방화벽이라고도 합니다.

투명한 프록시 방화벽은 요청을 수정하지 않으며 프록시 인증 및 식별에 필요한 것 이외에는 응답하지 않습니다. 투명한 프록시 방화벽의 한 예가 SOCKS입니다.

DB2 데이터베이스 시스템에서는 SOCKS 버전 4를 지원합니다.

SMLI(Stateful Multi-Layer Inspection) 방화벽

SMLI(Stateful Multi-Layer Inspection)은 OSI(Open System Interconnection) 모델의 일곱 계층 모두를 검사하는 정교한 양식의 패킷 필터링을 사용합니다.

각 패킷은 잘 알려진 상태에서 패킷 친화적으로 실험 및 비교되었습니다. 스크리닝 라우터 방화벽이 패킷 헤더만을 조사하는 반면, SMLI 방화벽은 데이터를 비롯하여 전체 패킷을 조사합니다.

제 7 장 보안 플러그인

DB2 데이터베이스 시스템에 대한 인증은 *보안 플러그인*을 사용하여 수행됩니다. 보안 플러그인은 인증 보안 서비스를 제공하는 동적으로 로드 가능한 라이브러리입니다.

DB2 데이터베이스 시스템에서 제공하는 플러그인의 유형은 다음과 같습니다.

- 그룹 검색 플러그인: 지정된 사용자에 대한 그룹 멤버쉽 정보를 검색합니다.
- 클라이언트 인증 플러그인: DB2 클라이언트에 대한 인증을 관리합니다.
- 서버 인증 플러그인: DB2 서버에 대한 인증을 관리합니다.

DB2 데이터베이스 관리 프로그램이 플러그인 인증에 대해 지원하는 두 가지 메커니즘은 다음과 같습니다.

사용자 ID/암호 인증

이 인증에는 사용자 ID와 암호를 사용하는 인증이 포함됩니다. 사용자 ID/암호 인증 플러그인을 사용하여 구현되는 인증 유형은 다음과 같습니다.

- CLIENT
- SERVER
- SERVER_ENCRYPT
- DATA_ENCRYPT
- DATA_ENCRYPT_CMP

이러한 인증 유형에 따라 사용자 인증이 수행되는 방식과 위치가 결정됩니다. 사용되는 인증 유형은 *authentication* 데이터베이스 관리 프로그램 구성 매개변수에 지정된 인증 유형에 따라 다릅니다. SRVCON_AUTH 매개변수가 지정된 경우, 연결 또는 접속 조작을 처리할 때 이 매개변수가 AUTHENTICATION보다 우선적으로 처리됩니다.

GSS-API 인증

GSS-API의 공식적인 이름은 *Generic Security Service Application Program Interface, Version 2*(IETF RFC2743) 및 *Generic Security Service API Version 2: C-Bindings*(IETF RFC2744)입니다. GSS-API를 사용하면 Kerberos 인증도 구현됩니다. GSS-API 인증 플러그인을 사용하여 구현되는 인증 유형은 다음과 같습니다.

- KERBEROS
- GSSPLUGIN
- KRB_SERVER_ENCRYPT
- GSS_SERVER_ENCRYPT

KRB_SERVER_ENCRYPT 및 GSS_SERVER_ENCRYPT는 GSS-API 인증과 사용자 ID/암호 인증을 모두 지원하지만, GSS-API 인증이 선호하는 인증 유형입니다.

주: 인증 유형에 따라 사용자 인증이 수행되는 방식과 위치가 결정됩니다. 특정 인증 유형을 사용하려면 인증 데이터베이스 관리 프로그램 구성 매개변수를 갱신하십시오.

각 플러그인을 개별적으로 사용하거나 하나 이상의 다른 플러그인과 함께 사용할 수 있습니다. 예를 들어, 서버 인증만 사용하고 클라이언트 그룹 인증에 대해서는 DB2 디폴트 값을 사용할 수 있습니다. 또는 그룹 또는 클라이언트 인증 플러그인만 사용할 수도 있습니다. GSS-API 인증 플러그인의 경우에만 클라이언트 및 서버 플러그인이 둘 다 필요합니다.

디폴트 동작은 인증에 대해 운영 체제 레벨 메커니즘을 구현하는 사용자 ID/암호 플러그인을 사용하는 것입니다. 이전 릴리스에서는 플러그인 구현 없이 운영 체제 레벨 인증을 직접 사용했습니다. 클라이언트 측 Kerberos는 AIX, Windows 및 Linux 운영 체제에서 지원됩니다. Windows 플랫폼의 경우 Kerberos가 디폴트로 지원됩니다.

DB2 데이터베이스 시스템에는 그룹 검색, 사용자 ID/암호 인증 및 Kerberos 인증에 사용되는 플러그인 세트가 있습니다. 보안 플러그인 인증을 사용할 경우, 사용자 고유의 플러그인을 개발하거나 써드 파티 플러그인을 구매하는 방식으로 DB2 클라이언트 및 서버 인증을 사용자 정의할 수 있습니다.

DB2 클라이언트에서 보안 플러그인 전개

DB2 클라이언트는 하나의 그룹 플러그인과 하나의 사용자 ID/암호 인증 플러그인을 지원할 수 있으며, 특정 GSS-API 플러그인에 대해서는 DB2 서버와 협상합니다. 이 협상은 클라이언트에 구현된 인증 플러그인과 처음으로 일치하는 인증 플러그인 이름의 구현된 DB2 서버의 GSS-API 플러그인 목록을 클라이언트에서 스캔하는 과정으로 이루어집니다. 서버의 플러그인 목록은 *srvcon_gssplugin_list* 데이터베이스 관리 프로그램 구성 매개변수 값에 지정되어 있으며, 이 값은 서버에 구현된 모든 플러그인의 이름이 포함되어 있습니다. 다음 그림은 DB2 클라이언트의 보안 플러그인 인프라스트럭처를 나타냅니다.

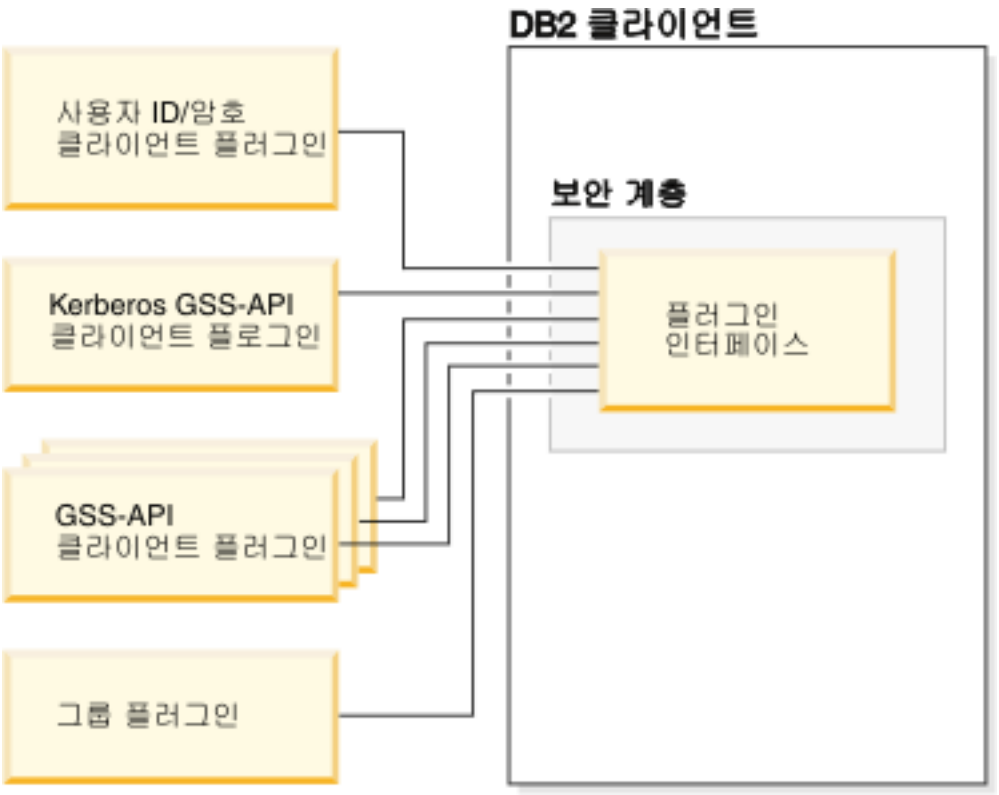


그림 5. DB2 클라이언트에서 보안 플러그인 전개

DB2 서버에서 보안 플러그인 전개

DB2 서버는 하나의 그룹 플러그인, 하나의 사용자 ID/암호 인증 플러그인 및 여러 개의 GSS-API 플러그인을 지원합니다. 여러 개의 GSS-API 플러그인은 *srvcon_gssplugin_list* 관리 프로그램 구성 매개변수 값에 목록으로 지정되어 있으며, 목록에 있는 한 개의 GSS-API 플러그인만 Kerberos 플러그인이 될 수 있습니다.

서버 측 보안 이외에도 클라이언트 권한 부여 플러그인을 데이터베이스 서버에 전개해야 할 수도 있습니다. db2start 및 db2trc와 같은 인스턴스 레벨 조작을 실행하는 경우, DB2 데이터베이스 관리 프로그램은 클라이언트 인증 플러그인을 사용하여 이러한 조작에 대한 권한 부여 검사를 수행합니다. 따라서 *authentication* 데이터베이스 관리 프로그램 구성 매개변수에 지정된 서버 플러그인에 해당하는 클라이언트 인증 플러그인을 설치해야 합니다. *authentication*과 *srvcon_auth* 간에는 큰 차이가 있습니다. 특히 이들 매개변수는 데이터베이스 연결 인증과 로컬 권한 부여에 각기 다른 메커니즘이 사용될 수 있도록 서로 다른 값으로 설정할 수 있습니다. 가장 일반적인 경우는 *srvcon_auth*는 GSSPLUGIN으로 설정하고 *authentication*는 SERVER로 설정하는 입니다. 데이터베이스 서버에서 클라이언트 인증 플러그인을 사용하지 않을 경우, 인스턴스 레벨 조작(예: db2start)이 실패하게 됩니다. 예를 들어, 인증 유형이 SERVER이고 사용자가 제공하는 클라이언트 플러그인을 사용하지 않는 경우, DB2 데이터베이스 시스템은 IBM에서 제공하는 디폴트 클라이언트 운영 체제 플러그인을 사용합니다. 다음

그림은 DB2 서버의 보안 플러그인 인프라스트럭처를 나타냅니다.

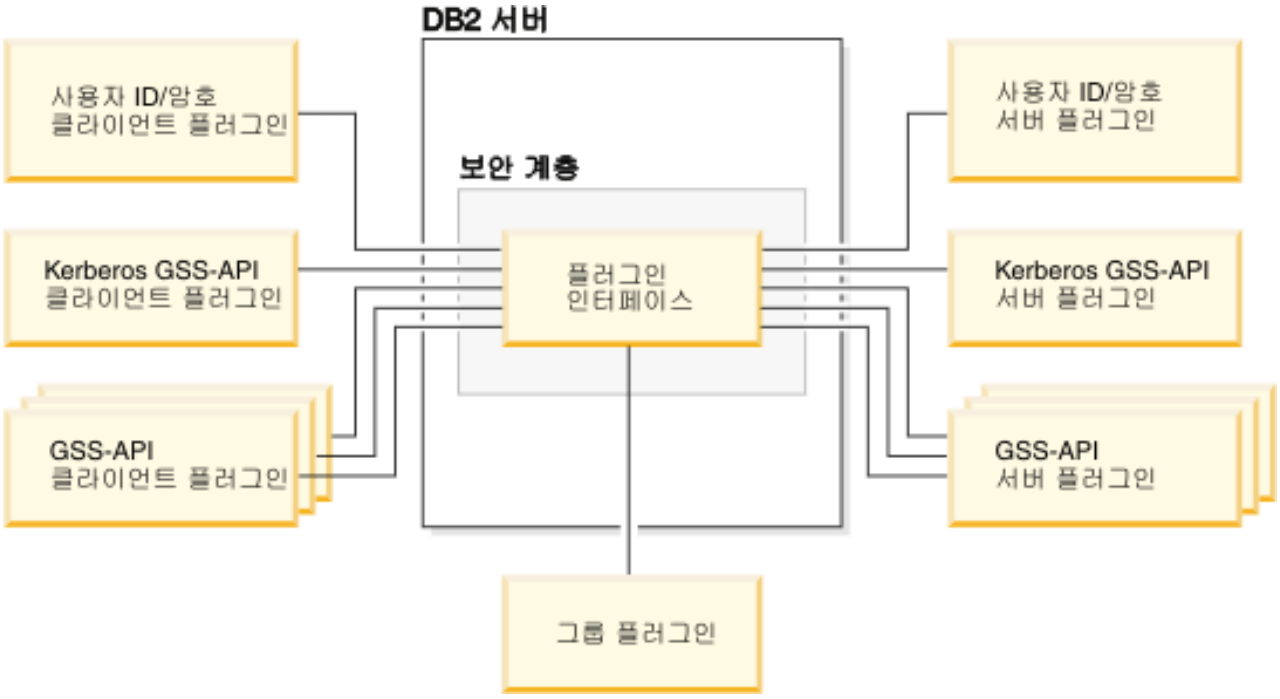


그림 6. DB2 서버에서 보안 플러그인 전개

주: 보안 플러그인의 전개를 적절하게 코딩, 검토 및 테스트하지 않은 경우 DB2 데이터베이스 시스템 설치의 무결성이 손상될 수 있습니다. DB2 데이터베이스 시스템은 일반적으로 발생하는 여러 가지 오류 유형에 대한 예방 조치를 취하고는 있지만, 이러한 예방 조치가 사용자가 작성한 보안 플러그인 배치 시의 무결성을 완전하게 보장하지는 못합니다.

보안 플러그인 사용

시스템 관리자는 특정 플러그인 관련 데이터베이스 관리 프로그램 구성 매개변수를 갱신하여 각 인증 메커니즘에 사용할 플러그인의 이름을 지정할 수 있습니다. 이러한 매개변수가 널(NULL)일 경우, 그룹 검색, 사용자 ID/암호 관리 또는 Kerberos(서버에서 인증이 Kerberos로 설정된 경우)에 대해 DB2 제공 플러그인이 디플트로 제공됩니다. DB2에서는 디폴트 GSS-API 플러그인을 제공하지 않습니다. 따라서 시스템 관리자가 *authentication* 매개변수에 인증 유형을 GSSPLUGIN으로 지정한 경우, *srvcon_gssplugin_list*에 GSS-API 인증 플러그인도 지정해야 합니다.

DB2의 보안 플러그인 로드 방식

데이터베이스 관리 프로그램이 시작되면 데이터베이스 관리 프로그램 구성 매개변수에 지정된 지원되는 플러그인이 모두 로드됩니다.

DB2 클라이언트는 연결 또는 접속 조작 중 서버와 협상한 보안 메커니즘에 적합한 플러그인을 로드합니다. 클라이언트 응용프로그램은 여러 개의 보안 플러그인을 동시에 로드하여 사용할 수 있습니다. 예를 들어, 서로 다른 인스턴스의 서로 다른 데이터베이스에 동시에 연결되어 있는 스레드 프로그램이 이와 같은 경우에 해당합니다.

연결 또는 접속 이외의 조치를 수행하려면 권한 부여(예: 데이터베이스 관리 프로그램 구성 갱신, 데이터베이스 관리 프로그램 시작 및 중지, DB2 추적 설정/해제)도 필요합니다. 이러한 조치의 경우 DB2 클라이언트 프로그램은 다른 데이터베이스 관리 프로그램 구성 매개변수에 지정된 플러그인을 로드합니다. *authentication*이 GSSPLUGIN으로 설정된 경우, DB2 데이터베이스 관리 프로그램은 *local_gssplugin*에 지정된 플러그인을 사용합니다. *authentication*이 KERBEROS로 설정된 경우, DB2 데이터베이스 관리 프로그램은 *clnt_krb_plugin*에 지정된 플러그인을 사용합니다. 또는 DB2 데이터베이스 관리 프로그램은 *clnt_pw_plugin*에 지정된 플러그인을 사용합니다.

보안 플러그인 API는 IPv4 플랫폼 또는 IPv6 플랫폼에서 호출할 수 있습니다. IPv4 주소는 a.b.c.d로 구성된 읽기 가능한 32비트 주소입니다. 여기서 a-d는 0-255의 10진수를 나타냅니다. IPv6 주소는 a:b:c:d:e:f:g:h로 구성된 128비트 주소입니다. 여기서 a-h 각각은 4개의 16진수를 나타냅니다.

보안 플러그인 개발

보안 플러그인을 개발 중인 경우, DB2 데이터베이스 관리 프로그램에서 사용할 표준 인증 함수를 구현해야 합니다. 사용자 정의된 보안 플러그인을 사용 중인 경우, CLP를 통해 발행된 연결 명령문 또는 동적 SQL문에 최대 255자의 사용자 ID를 사용할 수 있습니다. 사용 가능한 유형의 플러그인에 대해 구현해야 할 기능은 다음과 같습니다.

그룹 검색

사용자가 속해 있는 그룹의 목록을 가져옵니다.

사용자 ID/암호 인증

- 디폴트 보안 컨텍스트를 식별합니다(클라이언트만 해당).
- 암호의 유효성을 확인하고 선택적으로 암호를 변경합니다.
- 제공된 문자열이 유효한 사용자인지 판별합니다(서버만 해당).
- 서버로 보내기 전에 클라이언트에 제공된 사용자 ID 또는 암호를 수정합니다(클라이언트만 해당).
- 지정된 사용자와 연관된 DB2 권한 부여 ID를 리턴합니다.

GSS-API 인증

- 필수 GSS-API 함수를 구현합니다.
- 디폴트 보안 컨텍스트를 식별합니다(클라이언트만 해당).
- 사용자 ID 및 암호를 기준으로 초기 증명서를 생성하고 선택적으로 암호를 변경합니다(클라이언트만 해당).

- 보안 티켓을 작성 및 승인합니다.
- 지정된 GSS-API 보안 컨텍스트와 연관된 DB2 권한 부여 ID를 리턴합니다.

보안 플러그인 라이브러리 위치

보안 플러그인을 자체 개발하거나 써드 파티로부터 구매하는 방법으로 보안 플러그인을 얻은 후에는 이를 데이터베이스 서버의 특정 위치로 복사하십시오.

DB2 클라이언트가 클라이언트 측 사용자 인증 플러그인을 찾는 디렉토리는 다음과 같습니다.

- UNIX 32비트: `$DB2PATH/security32/plugin/client`
- UNIX 64비트: `$DB2PATH/security64/plugin/client`
- WINDOWS 32비트 및 64비트: `$DB2PATH¥security¥plugin¥`*instance name*`¥client`

주: Windows 기반 플랫폼의 경우, *instance name* 및 *client* 서브디렉토리는 자동으로 작성되지 않습니다. 인스턴스 소유자가 수동으로 작성해야 합니다.

DB2 데이터베이스 관리 프로그램이 서버 측 사용자 인증 플러그인을 찾는 디렉토리는 다음과 같습니다.

- UNIX 32비트: `$DB2PATH/security32/plugin/server`
- UNIX 64비트: `$DB2PATH/security64/plugin/server`
- WINDOWS 32비트 및 64비트: `$DB2PATH¥security¥plugin¥`*instance name*`¥server`

주: Windows 기반 플랫폼의 경우, *instance name* 및 *server* 서브디렉토리는 자동으로 작성되지 않습니다. 인스턴스 소유자가 수동으로 작성해야 합니다.

DB2 데이터베이스 관리 프로그램이 그룹 플러그인을 찾는 디렉토리는 다음과 같습니다.

- UNIX 32비트: `$DB2PATH/security32/plugin/group`
- UNIX 64비트: `$DB2PATH/security64/plugin/group`
- WINDOWS 32비트 및 64비트: `$DB2PATH¥security¥plugin¥`*instance name*`¥group`

주: Windows 기반 플랫폼의 경우, *instance name* 및 *group* 서브디렉토리는 자동으로 작성되지 않습니다. 인스턴스 소유자가 수동으로 작성해야 합니다.

보안 플러그인 이름 지정 규칙

보안 플러그인 라이브러리의 파일 이름 확장자는 플랫폼별로 달라야 합니다. C 또는 C++에서 작성된 보안 플러그인 라이브러리의 파일 이름 확장자도 플랫폼별로 달라야 합니다.

- Windows: `.dll`
- AIX: `.a` 또는 `.so`, 두 확장자가 모두 있는 경우 `.a` 확장자가 사용됩니다.
- Linux, HP IPF 및 Solaris: `.so`
- HPUX/PA-RISC: `.sl` 또는 `.so`, 두 확장자가 모두 있는 경우 `.sl` 확장자가 사용됩니다.

주: 사용자가 DB2 Universal JDBC 드라이버를 사용하는 보안 플러그인을 개발할 수도 있습니다.
예를 들어, `MyPlugin`이라는 보안 플러그인 라이브러리가 있다고 가정할 경우, 지원되는 운영 체제 각각에 적합한 라이브러리 파일 이름은 다음과 같습니다.

- Windows 32비트: MyPlugin.dll
- Windows 64비트: MyPlugin64.dll
- AIX 32 또는 64비트: MyPlugin.a 또는 MyPlugin.so
- SUN 32 또는 64비트, Linux 32 또는 64비트, HP 32 또는 64비트/IPF: MyPlugin.so
- HP-UX 32 또는 64비트/PA-RISC: MyPlugin.sl 또는 MyPlugin.so

주: 접미부 ″64″는 64비트 Windows 보안 플러그인에 대한 라이브러리 이름에만 필요합니다.

데이터베이스 관리 프로그램 구성을 보안 플러그인의 이름으로 갱신하는 경우, 접미부 ″64″없이 전체 라이브러리 이름을 사용하고 이름의 완전한 경로 부분과 파일 확장자는 생략하십시오. 보안 플러그인 라이브러리 `MyPlugin`은 운영 체제에 관계없이 다음과 같이 등록됩니다.

```
UPDATE DBM CFG USING CLNT_PW_PLUGIN MyPlugin
```

보안 플러그인 이름은 대소문자를 구분하므로 라이브러리 이름과 정확하게 일치해야 합니다. DB2 데이터베이스 시스템은 관련 데이터베이스 관리 프로그램 구성 매개변수의 값을 사용하여 라이브러리 경로를 어셈블한 다음 해당 라이브러리 경로를 사용하여 보안 플러그인 라이브러리를 로드합니다.

보안 플러그인의 이름이 충돌하는 것을 방지하려면 사용된 인증 메소드 및 플러그인을 작성한 회사의 식별 기호를 사용하여 플러그인 이름을 지정해야 합니다. 예를 들어, `Foo, Inc.`라는 회사가 인증 메소드 `FOOsomemethod`를 구현하는 플러그인을 작성했다면 플러그인 이름은 `FOOsomemethod.dll`입니다.

플러그인 이름의 최대 길이(파일 확장자 및 접미부 ″64″ 제외)는 32바이트로 제한됩니다. 데이터베이스 서버에서 지원하는 최대 플러그인 수는 없지만, 데이터베이스 관리 프로그램 구성에 있는 플러그인의 쉼표로 구분된 목록의 최대 길이는 255바이트입니다. 내장 파일 sqlenv.h에 있는 두 개의 define으로 이러한 두 제한을 식별할 수 있습니다.

```
#define SQL_PLUGIN_NAME_SZ     32     /* plug-in name */
#define SQL_SRVCON_GSSPLUGIN_LIST_SZ 255 /* GSS API plug-in list */
```

보안 플러그인 라이브러리 파일은 다음과 같은 파일 권한을 가지고 있어야 합니다.

- 인스턴스 소유자에 의해 소유됨
- 시스템의 모든 사용자가 읽을 수 있음
- 시스템의 모든 사용자가 실행할 수 있음

두 파트 사용자 ID에 대한 보안 플러그인 지원

Windows에 설치된 DB2 데이터베이스 관리 프로그램에서는 두 파트 사용자 ID를 사용할 수 있으며, 두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑할 수 있습니다.

예를 들어, 도메인과 사용자 ID로 구성된 Windows 운영 체제의 두 파트 사용자 ID(예: MEDWAY₩pieter)가 있다고 가정합니다. 이 예에서 MEDWAY는 도메인이고 pieter는 사용자 이름입니다. DB2 데이터베이스 시스템에서는 이 두 파트 사용자 ID를 한 파트 권한 부여 ID 또는 두 파트 권한 부여 ID에 맵핑할지 여부를 지정할 수 있습니다.

두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑할 수는 있지만, 디폴트 동작은 아닙니다. 디폴트로 한 파트 사용자 ID와 두 파트 사용자 ID는 한 파트 권한 부여 ID에 맵핑됩니다. 두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑할 수는 있지만, 디폴트 동작은 아닙니다.

두 파트 사용자 ID를 한 파트 사용자 ID에 맵핑하는 디폴트 동작을 사용하는 경우 사용자가 다음을 사용하여 데이터베이스에 연결할 수 있습니다.

```
db2 connect to db user MEDWAY₩pieter using pw
```

이때 디폴트 동작을 사용할 경우, 사용자 ID MEDWAY₩pieter는 권한 부여 ID PIETER로 분석됩니다. 두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑할 수 있는 경우, 권한 부여 ID는 MEDWAY₩PIETER입니다.

DB2에서는 두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑할 수 있도록 하기 위해 두 가지 인증 플러그인 세트를 제공합니다.

- 한 세트는 배타적으로 한 파트 사용자 ID를 한 파트 권한 부여 ID에 맵핑하고 두 파트 사용자 ID를 한 파트 권한 부여 ID에 맵핑합니다.
- 다른 세트는 한 파트 사용자 ID와 두 파트 사용자 ID를 모두 두 파트 권한 부여 ID에 맵핑합니다.

작업 환경의 사용자 이름을 다른 위치에 정의된 여러 어카운트(예: 로컬 어카운트, 도메인 어카운트, 트러스트된 도메인 어카운트)에 맵핑할 수 있는 경우, 두 파트 권한 부여 ID 맵핑이 가능한 플러그인을 지정할 수 있습니다.

도메인과 사용자 ID로 구성되는 두 파트 권한 부여 ID(예: MEDWAY₩pieter)와 한 파트 권한 부여 ID(예: PIETER)는 기능적으로 구별되는 권한 부여 ID라는 점을 유념하십시오. 이러한 권한 부여 ID 중 하나와 연관된 특권 세트는 다른 권한 부여 ID와 연관된 특권 세트와 완전히 다를 수 있습니다. 따라서 한 파트 권한 부여 ID와 두 파트 권한 부여 ID 관련 작업을 수행할 때는 주의해야 합니다.

다음 표는 DB2 데이터베이스 시스템에서 제공하는 플러그인의 종류 및 특정 인증 구현에 대한 플러그인 이름을 보여줍니다.

표 29. DB2 보안 플러그인

인증 유형	한 파트 사용자 ID 플러그인의 이름	두 파트 사용자 ID 플러그인의 이름
사용자 ID/암호(클라이언트)	IBMOSauthclient	IBMOSauthclientTwoPart
사용자 ID/암호(서버)	IBMOSauthserver	IBMOSauthserverTwoPart
Kerberos	IBMkrb5	IBMkrb5TwoPart

주: Windows 64비트 플랫폼의 경우 여기에 나열된 플러그인 이름에 ″64″가 추가됩니다.

사용자 ID/암호 또는 Kerberos 플러그인이 필요한 인증 유형을 지정한 경우, 위의 표에 있는 ″한 파트 사용자 ID 플러그인의 이름″ 컬럼에 나열된 플러그인이 디폴트로 사용됩니다.

두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑하려면 두 파트 플러그인(디폴트 플러그인이 아님)을 사용하도록 지정해야 합니다. 보안 관련 데이터베이스 관리 프로그램 구성 매개변수를 다음과 같이 설정하면 보안 플러그인이 인스턴스 레벨에서 지정됩니다.

두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑하는 서버 인증의 경우, 다음과 같이 설정해야 합니다.

- srvcon_pw_plugin을 IBMOSauthserverTwoPart로 설정
- clnt_pw_plugin을 IBMOSauthclientTwoPart로 설정

두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑하는 클라이언트 인증의 경우, 다음과 같이 설정해야 합니다.

- srvcon_pw_plugin을 IBMOSauthserverTwoPart로 설정
- clnt_pw_plugin을 IBMOSauthclientTwoPart로 설정

두 파트 사용자 ID를 두 파트 권한 부여 ID에 맵핑하는 Kerberos 인증의 경우, 다음과 같이 설정해야 합니다.

- srvcon_gssplugin_list를 IBMOSkrb5TwoPart로 설정
- clnt_krb_plugin을 IBMkrb5TwoPart로 설정

보안 플러그인 라이브러리는 Microsoft Windows Security Account Manager 호환 형식으로 지정된 두 파트 사용자 ID를 허용합니다(예: *domain₩user ID* 형식). 도메인 및 사용자 ID 정보는 연결 시 DB2 인증 및 권한 부여 프로세스에 사용됩니다.

기존 데이터베이스의 한 파트 권한 부여 ID와 충돌하지 않도록 새 데이터베이스를 작성하는 경우에는 두 파트 플러그인을 구현해야 합니다. 두 파트 권한 부여 ID를 사용하는 새 데이터베이스는 한 파트 권한 부여 ID를 사용하는 데이터베이스와는 다른 인스턴스에 작성해야 합니다.

보안 플러그인 API 버전

DB2 데이터베이스 시스템에서는 보안 플러그인 API의 버전 번호를 지원합니다. DB2 UDB 버전 8.2의 경우 이 버전 번호는 1로 시작하는 정수입니다.

DB2가 보안 플러그인 API에 전달하는 버전 번호는 DB2에서 지원할 수 있는 가장 높은 버전 번호로서, 구조 버전 번호에 해당합니다. 더 높은 API 버전을 지원하는 플러그인의 경우, DB2에서 요청한 버전에 대한 함수 포인터를 리턴해야 합니다. 더 낮은 API 버전을 지원하는 플러그인의 경우, 낮은 버전에 대한 함수 포인터를 채워야 합니다. 어떤 경우든지, 보안 플러그인 API는 함수 구조의 버전 필드에서 지원되는 API의 버전 번호를 리턴해야 합니다.

DB2에서는 API의 매개변수가 변경된 경우와 같이 필요한 경우에만 보안 플러그인의 버전 번호가 변경됩니다. 버전 번호는 DB2 릴리스 번호와 함께 자동으로 변경되지 않습니다.

보안 플러그인에 대한 32비트 및 64비트 고려사항

일반적으로 32비트 DB2 인스턴스는 32비트 보안 플러그인을 사용하고, 64비트 DB2 인스턴스는 64비트 보안 플러그인을 사용합니다. 그러나 64비트 인스턴스에서 DB2는 32비트 플러그인 라이브러리가 필요한 32비트 응용프로그램을 지원합니다.

32비트 응용프로그램과 64비트 응용프로그램을 모두 실행할 수 있는 데이터베이스 인스턴스를 하이브리드 인스턴스라고 합니다. 하이브리드 인스턴스에서 32비트 응용프로그램을 실행하려는 경우, 32비트 플러그인 디렉토리에서 32비트 보안 플러그인을 사용할 수 있는지 확인하십시오. Linux 및 UNIX 운영 체제(Linux IPF 제외)에 있는 DB2 인스턴스의 경우, security32 및 security64 디렉토리가 표시됩니다. Windows x64F 또는 IPF에 있는 64비트 DB2 인스턴스의 경우, 32비트 및 64비트 보안 플러그인이 같은 디렉토리에 있지만 64비트 플러그인 이름의 끝에는 ″64″가 붙습니다.

32비트 인스턴스에서 64비트 인스턴스로 업그레이드하려는 경우에는 64비트용으로 재컴파일할 보안 플러그인의 버전을 얻어야 합니다.

64비트 플러그인 라이브러리를 제공하지 않는 벤더로부터 보안 플러그인을 얻은 경우에는 32비트 응용프로그램을 실행하는 64비트 스텁을 구현할 수 있습니다. 이 경우 보안 플러그인은 라이브러리가 아닌 외부 프로그램입니다.

보안 플러그인 문제점 판별

보안 플러그인 관련 문제점은 SQL 오류 및 관리 통지 로그를 통해 보고됩니다.

보안 플러그인과 관련된 SQLCODE 값은 다음과 같습니다.

- db2start 또는 db2stop 중 플러그인 오류가 발생하면 SQLCODE -1365가 리턴됩니다.
- 로컬 권한 부여 문제점이 발생할 때마다 SQLCODE -1366이 리턴됩니다.
- 모든 연결 관련 플러그인 오류에 대해서는 SQLCODE -30082가 리턴됩니다.

관리 통지 로그를 통해 보안 플러그인을 디버깅하고 관리할 수 있습니다. UNIX에서 관리 통지 로그 파일을 확인하려면 sqllib/db2dump/*instance name*.*N*.nfy를 확인하십시오. Windows 운영 체제에서 관리 통지 로그를 확인하려면 이벤트 표시기 도구를 사용하십시오. 이벤트 표시기는 Windows 운영 체제의 "시작" 단추를 누른 다음 설정 -> 제어판 -> 관리 도구 -> 이벤트 표시기로 이동하면 찾을 수 있습니다. 보안 플러그인과 관련된 관리 통지 로그 값은 다음과 같습니다.

- 13000은 GSS-API 보안 플러그인 API에 대한 호출이 오류로 인해 실패했으며 선택적으로 오류 메시지가 리턴되었음을 나타냅니다.

  ```
  SQLT_ADMIN_GSS_API_ERROR (13000)
  Plug-in "plug-in name" received error code "error code" from
  GSS API  "gss api name" with the error message "error message"
  ```

- 13001은 DB2 보안 플러그인 API에 대한 호출이 오류로 인해 실패했으며 선택적으로 오류 메시지가 리턴되었음을 나타냅니다.

  ```
  SQLT_ADMIN_PLUGIN_API_ERROR(13001)
  Plug-in "plug-in name" received error code "error code" from DB2
  security plug-in API "gss api name" with the error message
  "error message"
  ```

- 13002는 DB2에서 플러그인이 언로드되지 않았음을 나타냅니다.

  ```
  SQLT_ADMIN_PLUGIN_UNLOAD_ERROR (13002)
  Unable to unload plug-in "plug-in name". No further action required.
  ```

- 13003은 핵심부 이름이 잘못되었음을 나타냅니다.

  ```
  SQLT_ADMIN_INVALID_PRIN_NAME (13003)
  The principal name "principal name" used for "plug-in name"
  is invalid. Fix the principal name.
  ```

- 13004는 플러그인 이름이 유효하지 않음을 나타냅니다. 플러그인 이름에는 경로 구분자(UNIX의 경우 "/", Windows의 경우 "₩")를 사용할 수 없습니다.

  ```
  SQLT_ADMIN_INVALID_PLGN_NAME (13004)
  The plug-in name "plug-in name" is invalid. Fix the plug-in name.
  ```

- 13005는 보안 플러그인이 로드되지 않았음을 나타냅니다. 플러그인이 올바른 디렉토리에 있으며, 적합한 데이터베이스 관리 프로그램 구성 매개변수가 갱신되었는지 확인하십시오.

  ```
  SQLT_ADMIN_PLUGIN_LOAD_ERROR (13005)
  Unable to load plug-in "plug-in name". Verify the plug-in existence and
  directory where it is located is correct.
  ```

- 13006은 보안 플러그인에서 예기치 않은 오류가 발생했음을 나타냅니다. 모든 db2support 정보를 수집하고, 가능한 경우 db2trc를 캡처한 다음 IBM Support에 추가 지원을 요청하십시오.

  ```
  SQLT_ADMIN_PLUGIN_UNEXP_ERROR (13006)
  Plug-in encountered unexpected error. Contact IBM Support for further assistance.
  ```

주: Windows 64비트 데이터베이스 서버에서 보안 플러그인을 사용하는데 보안 플러그인에 대한 로드 오류가 표시되는 경우, 32비트 및 64비트 고려사항 및 보안 플러그인 이름 지정 규칙에 대한 주제를 참조하십시오. 64비트 플러그인 라이브러리의 경우 라이브러리 이름에 "64"라는 접미부가 표시되어 하지만, 보안 플러그인 데이터베이스 관리 프로그램 구성 매개변수의 항목에는 이 접미부가 표시되어서는 안됩니다.

플러그인 사용

그룹 검색 플러그인 전개

DB2 보안 시스템의 그룹 검색 동작을 사용자 정의하기 위해, 해당 그룹 검색 플러그인을 개발하거나 써드파티로부터 구입할 수 있습니다.

데이터베이스 관리 시스템에 적합한 그룹 검색 플러그인을 확보한 후 이를 전개할 수 있습니다.

- 데이터베이스 서버에 그룹 검색 플러그인을 전개하려면 다음 단계를 수행하십시오.
  1. 그룹 검색 플러그인 라이브러리를 서버의 그룹 플러그인 디렉토리에 복사하십시오.
  2. 데이터베이스 관리 프로그램 구성 매개변수 *group_plugin*을 해당 플러그인의 이름으로 갱신하십시오.
- 데이터베이스 클라이언트에 그룹 검색 플러그인을 전개하려면 다음 단계를 수행하십시오.
  1. 그룹 검색 플러그인 라이브러리를 클라이언트의 그룹 플러그인 디렉토리에 복사하십시오.

2. 데이터베이스 클라이언트에서, 데이터베이스 관리 프로그램 구성 매개변수 *group_plugin*을 해당 플러그인의 이름으로 갱신하십시오.

사용자 ID/암호 플러그인 전개

DB2 보안 시스템의 사용자 ID/암호 인증 동작을 사용자 정의하기 위해, 해당 사용자 ID/암호 인증 플러그인을 개발하거나 써드파티에서 구입할 수 있습니다.

원하는 용도에 따라, 모든 사용자 ID/암호 기반 인증 플러그인을 클라이언트 플러그인 디렉토리 또는 서버 플러그인 디렉토리에 배치해야 합니다. 플러그인을 클라이언트 플러그인 디렉토리에 배치하는 경우, 클라이언트가 서버에 연결을 시도할 때 로컬 인증 검사 및 클라이언트 유효성 확인에 사용됩니다. 플러그인을 서버 플러그인 디렉토리에 배치하는 경우, 서버의 수신 연결을 처리하고 권한 부여 ID가 존재하는지 여부를 검사하는 데 사용됩니다. 또한 USER 또는 GROUP 키워드를 지정하지 않고 GRANT문을 발행할 때마다 유효합니다. 대부분의 경우, 사용자 ID/암호 인증에는 서버 측 플러그인만 필요합니다. 클라이언트 사용자 ID/암호 플러그인만 사용할 수는 있지만 대개 효율성이 떨어집니다. 일반적으로 클라이언트와 서버에서 사용자 ID/암호 플러그인이 일치해야 합니다.

주: *기존* 플러그인의 *새* 버전을 전개하기 전에 플러그인을 사용하는 모든 응용프로그램 또는 DB2 서버를 중지해야 합니다. 프로세스에서 동일한 이름의 새 버전이 복사될 때 프로세스에서 여전히 플러그인을 사용하고 있으면 트랩을 비롯한 정의되지 않은 동작이 발생합니다. 이 제한사항은 처음으로 플러그인을 전개하거나 플러그인을 사용하고 있지 않을 때는 적용되지 않습니다.

데이터베이스 관리 시스템에 적합한 사용자 ID/암호 인증 플러그인을 확보한 후 이를 전개할 수 있습니다.

- 데이터베이스 서버에 사용자 ID/암호 인증 플러그인을 전개하려면 데이터베이스 서버에서 다음 단계를 수행하십시오.
  1. 사용자 ID/암호 인증 플러그인 라이브러리를 서버 플러그인 디렉토리에 복사하십시오.
  2. 데이터베이스 관리 프로그램 구성 매개변수 *srvcon_pw_plugin*을 서버 플러그인의 이름으로 갱신하십시오. 이 플러그인은 CONNECT 및 ATTACH 요청이 처리될 때 서버에 사용됩니다.
  3. 또는 다음과 같습니다.
     - 데이터베이스 관리 프로그램 구성 매개변수 *srvcon_auth*를 CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP 인증 유형으로 설정하십시오. 또는:

– 데이터베이스 관리 프로그램 구성 매개변수 *srvcon_auth*를 NOT_SPECIFIED로 설정하고 *authentication*을 CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP 인증 유형으로 설정하십시오.

- 데이터베이스 클라이언트에 사용자 ID/암호 인증 플러그인을 전개하려면 각 클라이언트에서 다음 단계를 수행하십시오.
  1. 사용자 ID/암호 인증 플러그인 라이브러리를 클라이언트 플러그인 디렉토리에 복사하십시오.
  2. 데이터베이스 관리 프로그램 구성 매개변수 *clnt_pw_plugin*을 클라이언트 플러그인의 이름으로 갱신하십시오. 이 플러그인은 데이터베이스 관리 프로그램 구성 매개변수 *authentication*이 CLIENT로 설정되어 있을 때뿐만 아니라 인증이 수행되는 위치와 관계없이 로드되고 호출됩니다.
- 사용자 ID/암호 인증 플러그인을 사용하여 클라이언트, 서버 또는 게이트웨이에 로컬 인증을 수행하려면 각 클라이언트, 서버 또는 게이트웨이에서 다음 단계를 수행하십시오.
  1. 사용자 ID/암호 인증 플러그인 라이브러리를 클라이언트, 서버 또는 게이트웨이의 클라이언트 플러그인 디렉토리에 복사하십시오.
  2. 데이터베이스 관리 프로그램 구성 매개변수 *clnt_pw_plugin*을 해당 플러그인의 이름으로 갱신하십시오.
  3. 데이터베이스 관리 프로그램 구성 매개변수 *authentication*을 CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP로 설정하십시오.

GSS-API 플러그인 전개

DB2 보안 시스템의 인증 동작을 사용자 정의하기 위해, GSS-API를 사용하여 해당 인증 플러그인을 개발하거나 써드파티에서 구입할 수 있습니다.

플러그인 유형이 Kerberos가 아닌 경우, 클라이언트 및 서버의 플러그인 이름이 일치하고 플러그인 유형이 동일해야 합니다. 클라이언트 및 서버의 플러그인을 동일한 벤더가 제공할 필요는 없지만, 호환 가능한 GSS-API 토큰을 생성하고 사용해야 합니다. Kerberos 플러그인은 표준화되어 있으므로 클라이언트 및 서버에 전개된 모든 Kerberos 플러그인을 조합할 수 있습니다. 하지만 표준화되지 않은 다른 GSS-API를 구현하는 경우(예: *x.509* 증명서), DB2 데이터베이스 시스템과 부분적으로만 호환할 수 있습니다. 원하는 용도에 따라, 모든 GSS-API 인증 플러그인을 클라이언트 플러그인 디렉토리 또는 서버 플러그인 디렉토리에 배치해야 합니다. 플러그인을 클라이언트 플러그인 디렉토리에 배치하는 경우, 클라이언트가 서버에 연결을 시도할 때 로컬 인증 검사 및 클라이언트 유효성 확인에 사용됩니다. 플러그인을 서버 플러그인 디렉토리에 배치하는 경

우, 서버의 수신 연결을 처리하고 권한 부여 ID가 존재하는지 여부를 검사하는 데 사용됩니다. 또한 USER 또는 GROUP 키워드를 지정하지 않고 GRANT문을 발행할 때마다 유효합니다.

주: *기존* 플러그인의 *새* 버전을 전개하기 전에 플러그인을 사용하는 모든 응용프로그램 또는 DB2 서버를 중지해야 합니다. 프로세스에서 동일한 이름의 새 버전이 복사될 때 프로세스에서 여전히 플러그인을 사용하고 있으면 트랩을 비롯한 정의되지 않은 동작이 발생합니다. 이 제한사항은 처음으로 플러그인을 전개하거나 플러그인을 사용하고 있지 않을 때는 적용되지 않습니다.

데이터베이스 관리 시스템에 적합한 GSS-API 인증 플러그인을 확보한 후 이를 전개할 수 있습니다.

- 데이터베이스 서버에 GSS-API 인증 플러그인을 전개하려면 서버에서 다음 단계를 수행하십시오.
  1. GSS-API 인증 플러그인 라이브러리를 서버 플러그인 디렉토리에 복사하십시오. 이 디렉토리에 다수의 GSS-API 플러그인을 복사할 수 있습니다.
  2. 데이터베이스 관리 프로그램 구성 매개변수 *srvcon_gssplugin_list*를 GSS-API 플러그인 디렉토리에 설치된 플러그인의 순서 지정되어 있고 쉼표로 구분된 이름 목록으로 갱신하십시오.
  3. 또는 다음과 같습니다.
     - 데이터베이스 관리 프로그램 구성 매개변수 *srvcon_auth*를 GSSPLUGIN 또는 GSS_SERVER_ENCRYPT로 설정하면 서버가 GSSAPI PLUGIN 인증 방법을 사용할 수 있습니다. 또는:
     - 데이터베이스 관리 프로그램 구성 매개변수 *srvcon_auth*를 NOT_SPECIFIED으로 설정하고 *authentication*을 GSSPLUGIN 또는 GSS_SERVER_ENCRYPT로 설정하면 서버가 GSSAPI PLUGIN 인증 방법을 사용할 수 있습니다.
- 데이터베이스 클라이언트에 GSS-API 인증 플러그인을 전개하려면 각 클라이언트에서 다음 단계를 수행하십시오.
  1. GSS-API 인증 플러그인 라이브러리를 클라이언트 플러그인 디렉토리에 복사하십시오. 이 디렉토리에 다수의 GSS-API 플러그인을 복사할 수 있습니다.클라이언트는 클라이언트에서 사용할 수 있는 서버의 플러그인에 포함된 첫 번째 GSS-API 플러그인을 선택하여 CONNECT 또는 ATTACH 조작 중에 인증에 사용할 GSS-API 플러그인을 선택합니다.
  2. 선택사항: 클라이언트가 액세스하는 데이터베이스를 카탈로그화하여 클라이언트가 GSS-API 인증 플러그인만 인증 메커니즘으로 허용함을 나타낼 수 있습니다. 예를 들어, 다음과 같습니다.

     ```
     CATALOG DB testdb AT NODE testnode AUTHENTICATION GSSPLUGIN
     ```

- GSS-API 인증 플러그인을 사용하여 클라이언트, 서버 또는 게이트웨이에 로컬 인증을 수행하려면 다음 단계를 수행하십시오.
  1. GSS-API 인증 플러그인 라이브러리를 클라이언트, 서버 또는 게이트웨이의 클라이언트 플러그인 디렉토리에 복사하십시오.
  2. 데이터베이스 관리 프로그램 구성 매개변수 *local_gssplugin*을 해당 플러그인의 이름으로 갱신하십시오.
  3. 데이터베이스 관리 프로그램 구성 매개변수 *authentication*을 GSSPLUGIN 또는 GSS_SERVER_ENCRYPT로 설정하십시오.

Kerberos 플러그인 전개

DB2 보안 시스템의 Kerberos 인증 동작을 사용자 정의하기 위해, 해당 Kerberos 인증 플러그인을 개발하거나 써드파티에서 구입할 수 있습니다. Kerberos 보안 플러그인은 IPv6를 지원하지 않습니다.

주: *기존* 플러그인의 *새* 버전을 전개하기 전에 플러그인을 사용하는 모든 응용프로그램 또는 DB2 서버를 중지해야 합니다. 프로세스에서 동일한 이름의 새 버전이 복사될 때 프로세스에서 여전히 플러그인을 사용하고 있으면 트랩을 비롯한 정의되지 않은 동작이 발생합니다. 이 제한사항은 처음으로 플러그인을 전개하거나 플러그인을 사용하고 있지 않을 때는 적용되지 않습니다.

데이터베이스 관리 시스템에 적합한 Kerberos 인증 플러그인을 확보한 후 이를 전개할 수 있습니다.

- 데이터베이스 서버에 Kerberos 인증 플러그인을 전개하려면 서버에서 다음 단계를 수행하십시오.
  1. Kerberos 인증 플러그인 라이브러리를 서버 플러그인 디렉토리에 복사하십시오.
  2. 순서 지정되어 있고 쉼표로 구분된 목록으로 제공되는 데이터베이스 관리 프로그램 구성 매개변수 **srvcon_gssplugin_list**를 갱신하여 Kerberos 서버 플러그인 이름을 포함하십시오. 이 목록에 있는 플러그인 하나만 Kerberos 플러그인이 될 수 있습니다. 이 목록이 비어 있고 **authentication**이 KERBEROS 또는 KRB_SVR_ENCRYPT로 설정되어 있으면, 디폴트 DB2 Kerberos 플러그인 IBMkrb5가 사용됩니다.
  3. 필요한 경우, 현재 인증 유형을 겹쳐쓰도록 **srvcon_auth** 데이터베이스 관리 프로그램 구성 매개변수를 설정하십시오. **srvcon_auth** 데이터베이스 관리 프로그램 구성 매개변수가 설정되어 있지 않으면, DB2 데이터베이스 관리 프로그램이 **authentication** 구성 매개변수 값을 사용합니다. **authentication** 구성 매개변수가 현재 다음 인증 유형 중 하나로 설정되어 있으면 Kerberos 플러그인을 전개하여 사용할 수 있습니다.
     - KERBEROS
     - KRB_SERVER_ENCRYPT

– GSSPLUGIN

– GSS_SERVER_ENCRYPT

현재 인증 유형을 겹쳐써야 하는 경우, **srvcon_auth** 구성 매개변수를 다음 인증 유형 중 하나로 설정하십시오.

– KERBEROS

– KRB_SERVER_ENCRYPT

– GSSPLUGIN

– GSS_SERVER_ENCRYPT

- 데이터베이스 클라이언트에 Kerberos 인증 플러그인을 전개하려면 각 클라이언트에서 다음 단계를 수행하십시오.
  1. Kerberos 인증 플러그인 라이브러리를 클라이언트 플러그인 디렉토리에 복사하십시오.
  2. 데이터베이스 관리 프로그램 구성 매개변수 **clnt_krb_plugin**을 Kerberos 플러그인의 이름으로 갱신하십시오. **clnt_krb_plugin**이 비어 있으면 DB2는 클라이언트가 Kerberos 인증을 사용할 수 없다고 가정합니다. 이 설정은 서버가 플러그인을 지원할 수 없을 때만 적합합니다. 서버 및 클라이언트 둘 다 보안 플러그인을 지원하는 경우, 클라이언트 값 **clnt_krb_plugin**보다 우선하여 디폴트 서버 플러그인 *IBMkrb5*가 사용됩니다. Kerberos 인증 플러그인을 사용하여 클라이언트, 서버 또는 게이트웨이에 로컬 인증을 수행하려면 다음 단계를 수행하십시오.
     a. Kerberos 인증 플러그인 라이브러리를 클라이언트, 서버 또는 게이트웨이의 클라이언트 플러그인 디렉토리에 복사하십시오.
     b. 데이터베이스 관리 프로그램 구성 매개변수 **clnt_krb_plugin**을 해당 플러그인의 이름으로 갱신하십시오.
     c. 데이터베이스 관리 프로그램 구성 매개변수 **authentication**을 KERBEROS 또는 KRB_SERVER_ENCRYPT로 설정하십시오.
  3. 선택사항: 클라이언트가 액세스하는 데이터베이스를 카탈로그화하여 클라이언트가 Kerberos 인증 플러그인만 사용함을 나타낼 수 있습니다. 예를 들어, 다음과 같습니다.

     ```
     CATALOG DB testdb AT NODE testnode AUTHENTICATION KERBEROS
             TARGET PRINCIPAL service/host@REALM
     ```

주: Kerberos를 지원하는 플랫폼의 경우, IBMkrb5 라이브러리는 클라이언트 플러그인 디렉토리에 제공됩니다. Kerberos 플러그인은 GSS-API 플러그인을 사용하여 구현되므로 DB2 데이터베이스 관리 프로그램은 이 라이브러리를 유효한 GSS-API 플러그인으로 인식합니다.

LDAP 기반 인증 및 그룹 찾아보기 지원

DB2 데이터베이스 관리 프로그램 및 DB2 Connect는 LDAP 보안 플러그인 모듈을 사용하거나 투명한 LDAP을 통해 LDAP 기반 인증 및 그룹 찾아보기 기능을 지원합니다.

AIX 운영 체제에서 LDAP 기반 인증 지원이 향상되었습니다. 이제 LDAP을 통해 투명한 LDAP 인증을 사용하여 사용자 인증 및 그룹 멤버쉽을 중앙에서 관리할 수 있습니다. 운영 체제에서 사용자를 인증하고 사용자 그룹을 획득할 수 있도록 DB2 인스턴스를 구성할 수 있습니다. AIX 운영 체제는 LDAP 서버를 통해 인증을 수행합니다. 투명한 LDAP 인증이 가능하게 하려면 **DB2AUTH** 기타 레지스트리 변수를 OSAUTHDB로 설정하십시오.

LDAP 보안 플러그인을 사용하여 LDAP 기반 인증을 구현할 수도 있습니다. LDAP 보안 플러그인 모듈을 사용하면 DB2 데이터베이스 관리 프로그램이 LDAP 디렉토리에 정의된 사용자를 인증하므로, 사용자와 그룹을 운영 체제에 정의할 필요가 없습니다. 지원되는 운영 체제는 다음과 같습니다.

- AIX
- Itanium 기반 HP Integrity Series 시스템의 HP-UX(IA-64)
- IA32, x64 또는 zSeries 하드웨어에 있는 Linux
- Solaris
- Windows

보안 플러그인 모듈과 함께 사용할 수 있도록 지원되는 LDAP 서버는 다음과 같습니다.

- IBM Lotus® Domino® LDAP Server 버전 7.0 이상
- IBM Tivoli® Directory Server(ITDS) 버전 5.2, 6.0 이상
- Microsoft Active Directory(MSAD) 버전 2000, 2003 이상
- Novell eDirectory 버전 8.7 이상
- OpenLDAP 서버 버전 2.3.32 이상
- Sun Java System Directory Server Enterprise Edition 버전 5.2 이상
- z/OS Integrated Security Services LDAP Server 버전 V1R6 이상

주: LDAP 플러그인 모듈을 사용하는 경우 데이터베이스와 연관된 모든 사용자를 LDAP 서버에 정의해야 합니다. 이러한 사용자에는 DB2 인스턴스 소유자 ID와 분리 사용자가 포함됩니다. 이러한 사용자는 일반적으로 운영 체제에 정의되어 있지만 LDAP에도 정의해야 합니다. 마찬가지로 LDAP 그룹 플러그인 모듈을 사용하는 경우 권한 부여에

필요한 그룹을 LDAP 서버에 정의해야 합니다. 이러한 그룹에는 데이터베이스 관리 프로그램 구성에 정의된 SYSADM, SYSMAINT, SYSCTRL 및 SYSMON 그룹이 포함됩니다.

DB2 보안 플러그인 모듈은 아래에 설명된 서버 측 인증, 클라이언트 측 인증 및 그룹 찾아보기에 사용할 수 있습니다. 특정 환경에 따라 하나, 둘 또는 세 개 모두의 플러그인 유형을 사용해야 합니다.

DB2 보안 플러그인 모듈을 사용하려면 다음 단계를 수행하십시오.

1. 서버, 클라이언트 또는 그룹 플러그인 모듈 또는 이러한 모듈 조합이 필요한지 여부를 결정하십시오.
2. IBM LDAP 보안 플러그인 구성 파일(디폴트 이름: `IBMLDAPSecurity.ini`)에서 값을 설정하여 플러그인 모듈을 구성하십시오. 적합한 값을 판별하려면 LDAP 관리자에게 문의해야 합니다.
3. 플러그인 모듈을 사용 가능하게 설정하십시오.
4. 여러 LDAP 사용자 ID와의 연결을 테스트하십시오.

서버 인증 플러그인

서버 인증 플러그인 모듈은 CONNECT 및 ATTACH문에 클라이언트가 제공한 사용자 ID와 암호에 대해 서버 유효성 확인을 수행합니다. 또한 필요한 경우 LDAP 사용자 ID를 DB2 a 권한 부여 ID에 맵핑하는 방법도 제공합니다. 서버 플러그인 모듈은 일반적으로 사용자가 자신의 LDAP 사용자 ID 및 암호를 사용하여 DB2 데이터베이스 관리 프로그램을 인증하려는 경우에 필요합니다.

클라이언트 인증 플러그인

클라이언트 인증 플러그인 모듈은 클라이언트 시스템에서 사용자 ID 및 암호에 대한 유효성 확인이 발행한 경우 즉, 클라이언트의 SRVCON_AUTH 또는 AUTHENTICATION 설정을 사용하여 DB2 서버가 구성된 경우에 사용됩니다. 클라이언트에서는 CONNECT 또는 ATTACH문에 제공된 사용자 ID와 암호의 유효성을 확인한 다음 사용자 ID를 DB2 서버로 전송합니다. 클라이언트 인증은 보안 유지가 어려우므로 일반적으로 권장되지 않습니다.

클라이언트 인증 플러그인 모듈은 데이터베이스 서버에 있는 로컬 운영 체제 사용자 ID가 해당 사용자와 연관된 DB2 권한 부여 ID와 다른 경우에도 필요합니다. 데이터베이스 서버에서 로컬 명령(예: `db2start`에 대한 권한 부여 검사를 수행하기 전에 클라이언트 측 플러그인을 사용하여 로컬 운영 체제 사용자 ID를 DB2 권한 부여 ID에 맵핑할 수 있습니다.

그룹 찾아보기 플러그인

그룹 찾아보기 플러그인 모듈은 LDAP 서버에서 특정 사용자에 대한 그룹 멤버쉽 정보를 검색하며, LDAP을 사용하여 그룹 정의를 저장하려는 경우에 필요합니다. 가장 일반적으로 필요한 경우는 다음과 같습니다.

- 모든 사용자 및 그룹이 LDAP 서버에 정의된 경우
- 데이터베이스 서버에 로컬로 저장된 사용자가 동일한 사용자 ID를 사용하여 LDAP 서버에도 정의된 경우(인스턴스 소유자 및 분리 사용자 포함)
- DB2 서버에서 암호 유효성 확인이 발생한 경우(즉, 서버 DBM 구성 파일에 SERVER, SERVER_ENCRYPT 또는 DATA_ENCRYPT의 AUTHENTICATION 또는 SRVCON_AUTH 값이 설정되어 있는 경우)

일반적으로 서버 인증 플러그인 모듈과 그룹 찾아보기 플러그인 모듈만 서버에 설치해도 충분합니다. DB2 클라이언트에는 보통 LDAP 플러그인 모듈을 설치할 필요가 없습니다.

LDAP 그룹 찾아보기 플러그인 모듈은 다른 형태의 인증 플러그인(예: Kerberos)과 함께 사용해야 합니다. 이 경우 LDAP 그룹 찾아보기 플러그인 모듈은 사용자와 연관된 DB2 권한 부여 ID에 제공됩니다. 플러그인 모듈은 LDAP 디렉토리에서 AUTHID_ATTRIBUTE와 일치하는 사용자를 검색한 다음 해당 사용자 오브젝트와 연관된 그룹을 검색합니다.

인증 및 그룹 찾아보기용 DB2 LDAP 플러그인 모듈을 SSL 옵션과 함께 사용(Linux, HP 및 Solaris)

다음 정보는 Linux, HP 및 Solaris 운영 체제에서 SSL 옵션과 함께 인증 및 그룹 찾아보기용 DB2 LDAP 플러그인 모듈을 사용 중일 때만 적용됩니다.

여기서 참조된 SSL 옵션은 `IBMLDAPSEcurity.ini` 구성 파일에서 ENABLE_SSL 속성 설정이 참입니다. 이것은 DB2 서버 및 DB2 클라이언트 간에 데이터 통신 암호화용 SSL을 구성하는 다른 프로시저입니다.

이 섹션이 사용자의 상황에 적용되는 경우, 인증 및 그룹 찾아보기용 DB2 LDAP 플러그인 모듈에 SSL 옵션을 사용하기 위해서는 아래 단계를 수행해야 합니다.

Linux on x64, Linux for IBM System z 64, Linux PPC 64, HPUX IA 64, Solaris SPARC 64 및 Solaris x64 플랫폼의 경우, *DB2 install path*`/sqllib/lib64`에 위치한 12개의 GSKit 버전 8 라이브러리가 있습니다.

- `libgsk8acmeidup_64.so`
- `libgsk8cms_64.so`
- `libgsk8dbfl_64.so`
- `libgsk8drld_64.so`

- libgsk8iccs_64.so
- libgsk8kicc_64.so
- libgsk8km_64.so
- libgsk8ldap_64.so
- libgsk8p11_64.so
- libgsk8ssl_64.so
- libgsk8sys_64.so
- libgsk8valn_64.so

운영 체제의 해당 디렉토리(HPUX IA 64의 /usr/lib, Solaris SPARC 64 및 Solaris x64의 /usr/lib/64 또는 Linux on x64, Linux for IBM system z 64 및 Linux PPC 64의 /usr/lib64)에서 루트 권한이 있는 사용자로서 ln 명령을 발행하여 위의 각 라이브러리에 대한 기호 링크를 작성하십시오. 예를 들어, 다음과 같습니다.

```
ln -s DB2 install path/sqllib/lib64/libgsk8acmeidup_64.so .
```

Linux on x86의 경우, *DB2 install path*/sqllib/lib에 위치한 12개의 GSKit 버전 8 라이브러리가 있습니다.

- libgsk8acmeidup.so
- libgsk8cms.so
- libgsk8dbfl.so
- libgsk8drld.so
- libgsk8iccs.so
- libgsk8kicc.so
- libgsk8km.so
- libgsk8ldap.so
- libgsk8p11.so
- libgsk8ssl.so
- libgsk8sys.so
- libgsk8valn.so

/usr/lib 디렉토리에서, 루트 권한이 있는 사용자로서 ln 명령을 발행하여 위의 각 라이브러리에 대한 기호 링크를 작성하십시오. 예를 들어, 다음과 같습니다.

```
ln -s DB2 install path/sqllib/lib32/libgsk8acmeidup.so .
```

LDAP 플러그인 모듈 구성

LDAP 플러그인 모듈을 구성하려면 사용 중인 환경에 적합하도록 IBM LDAP 보안 플러그인 구성 파일을 갱신해야 합니다. 대부분의 경우 LDAP 관리자에게 문의해야 적합한 구성 값을 판별할 수 있습니다.

IBM LDAP 보안 플러그인 구성 파일의 디폴트 이름과 위치는 다음과 같습니다.

- UNIX: `INSTHOME/sqllib/cfg/IBMLDAPSecurity.ini`
- Windows: `%DB2PATH%₩cfg₩IBMLDAPSecurity.ini`

선택적으로 DB2LDAPSecurityConfig 환경 변수를 사용하여 이 파일의 위치를 지정할 수 있습니다. Windows의 경우, DB2 서비스에서 선택할 수 있도록 전역 시스템 환경에서 DB2LDAPSecurityConfig를 설정해야 합니다.

다음 표는 적합한 구성 값을 판별하는 데 유용한 정보를 제공합니다.

표 30. 서버 관련 값

매개변수	설명
LDAP_HOST	LDAP 서버의 이름입니다. LDAP 서버 호스트 이름 또는 IP 주소가 공백으로 구분된 목록이며, 각각에 대한 포트 번호가 선택적으로 포함될 수 있습니다. 예: `host1[:port] [host2:[port2] ... ]` 디폴트 포트 번호는 389 또는 636(SSL이 사용 가능한 경우)입니다.
ENABLE_SSL	SSL 지원이 가능하게 하려면 ENABLE_SSL을 TRUE로 설정하십시오(GSKit이 설치되어 있어야 함). 이 매개변수는 선택적 매개변수이므로 디폴트 값이 FALSE(SSL 지원 안됨)로 설정됩니다.
SSL_KEYFILE	SSL 키 링에 대한 경로입니다. LDAP 서버가 GSKit 설치에 의해 자동으로 트러스트되지 않는 인증서를 사용하고 있는 경우에만 키 파일이 필요합니다. 예: `SSL_KEYFILE = /home/db2inst1/IBMLDAPSecurity.kdb`
SSL_PW	SSL 키 링 암호입니다. 예: `SSL_PW = keyfile-password`

표 31. 사용자 관련 값

매개변수	설명
USER_ OBJECTCLASS	사용자에 대해 사용되는 LDAP 오브젝트 클래스입니다. 일반적으로 USER_OBJECTCLASS를 `inetOrgPerson`(Microsoft Active Directory용 사용자)으로 설정하십시오. 예: `USER_OBJECTCLASS = inetOrgPerson`
USER_BASEDN	사용자를 검색할 때 사용할 LDAP 기본 DN입니다. 지정되지 않은 경우 LDAP 디렉토리의 루트부터 사용자를 검색합니다. 일부 LDAP 서버에서는 이 매개변수에 대한 값을 지정해야 합니다. 예: `USER_BASEDN = o=ibm`

표 31. 사용자 관련 값 (계속)

매개변수	설명

USERID_ ATTRIBUTE	사용자 ID를 나타내는 LDAP 사용자 속성입니다.
	USERID_ATTRIBUTE 속성은 USER_OBJECTCLASS 및 USER_BASEDN (지정된 경우)과
	함께 사용되어 사용자가 규정되지 않은 사용자 ID로
	DB2 CONNECT문을 발행하는 경우
	LDAP 검색 필터를 생성합니다.
	예를 들어, `USERID_ATTRIBUTE = uid`일 경우 다음 명령문을 발행하면
	`db2 connect to MYDB user bob using bobpass`
	다음 검색 필터가 생성됩니다.
	`&(objectClass=inetOrgPerson)(uid=bob)`

AUTHID_ ATTRIBUTE	DB2 권한 부여 ID를 나타내는 LDAP 사용자 속성입니다. 이 매개변수는 USERID_ATTRIBUTE와 동일합니다. 예: `AUTHID_ATTRIBUTE = uid`

표 32. 그룹 관련 값

매개변수	설명
GROUP_ OBJECTCLASS	그룹에 대해 사용되는 LDAP 오브젝트 클래스입니다. 일반적으로 `groupOfNames` 또는 `groupOfUniqueNames`입니다(Microsoft Active Directory의 경우 `group`). 예: `GROUP_OBJECTCLASS = groupOfNames`
GROUP_BASEDN	그룹을 검색할 때 사용할 LDAP 기본 DN입니다. 지정되지 않은 경우 LDAP 디렉토리의 루트부터 그룹을 검색합니다. 일부 LDAP 서버에서는 이 매개변수에 대한 값을 지정해야 합니다. 예: `GROUP_BASEDN = o=ibm`
GROUPNAME_ ATTRIBUTE	그룹 이름을 나타내는 LDAP 그룹 속성입니다. 예: `GROUPNAME_ATTRIBUTE = cn`
GROUP_LOOKUP_ METHOD	사용자에 대한 그룹 멤버쉽을 찾는 데 사용되는 메소드를 판별합니다. 가능한 값은 다음과 같습니다. • SEARCH_BY_DN - 사용자를 구성원으로 나열하는 그룹에 대한 검색을 나타냅니다. 멤버쉽은 그룹 속성에 GROUP_LOOKUP_ATTRIBUTE으로 정의된 그룹 속성으로 표시됩니다(일반적으로 `member` 또는 `uniqueMember`). • USER_ATTRIBUTE - 이 경우 사용자 그룹이 사용자 오브젝트 자체의 속성으로 나열됩니다. 이 설정은 GROUP_LOOKUP_ATTRIBUTE로 정의된 사용자 속성을 검색하여 사용자 그룹을 가져오도록 지정합니다(일반적으로 Microsoft Active Directory의 경우 `memberOf` 또는 IBM Tivoli Directory Server의 경우 `ibm-allGroups`). 예: `GROUP_LOOKUP_METHOD = SEARCH_BY_DN` `GROUP_LOOKUP_METHOD = USER_ATTRIBUTE`

GROUP_LOOKUP_ ATTRIBUTE	GROUP_LOOKUP_METHOD에 설명된 것과 같이 그룹 멤버쉽을 판별하는 데 사용되는 속성 이름입니다.
	
	예를 들어, 다음과 같습니다.
	`GROUP_LOOKUP_ATTRIBUTE = member`
	`GROUP_LOOKUP_ATTRIBUTE = ibm-allGroups`

표 32. 그룹 관련 값 (계속)

매개변수	설명
NESTED_GROUPS	NESTED_GROUPS이 TRUE일 경우, DB2 데이터베이스 관리 프로그램이 발견된 모든 그룹의 그룹 멤버쉽을 찾는 방식으로 그룹 멤버쉽을 반복적으로 검색합니다. 순환(예: A는 B에 속하고, B는 A에 속함)이 올바르게 처리되었습니다. 이 매개변수는 선택적 매개변수이므로 디폴트 값이 FALSE로 설정됩니다.

표 33. 기타 값

매개변수	설명

SEARCH_DN, SEARCH_PW	LDAP 서버에서 익명 액세스를 지원하지 않거나 사용자 또는 그룹을 검색할 때 익명 액세스로는 충분하지 않은 경우, 선택적으로 검색을 수행하는 데 사용할 DN과 암호를 정의할 수 있습니다.
	
	예를 들어, 다음과 같습니다.
	`SEARCH_DN = cn=root`
	`SEARCH_PW = rootpassword`

DEBUG	DEBUG를 TRUE로 설정하면 LDAP 관련 문제를 디버깅하는 데 유용한 추가 정보가 db2diag 로그 파일에 기록됩니다. 대부분의 추가 정보는 DIAGLEVEL 4(INFO)에서 로그됩니다. DEBUG는 디폴트로 false로 설정됩니다.

LDAP 플러그인 모듈 사용

컴파일된 2진 LDAP 플러그인 모듈은 DB2 인스턴스 디렉토리에 있습니다.

다음 표는 DB2 인스턴스에서 LDAP 플러그인 모듈의 위치를 보여줍니다.

표 34. 64비트 UNIX 및 Linux 시스템의 경우

플러그인 모듈 유형	위치
서버	`/sqllib/security64/plugin/IBM/server`
클라이언트	`/sqllib/security64/plugin/IBM/client`
그룹	`/sqllib/security64/plugin/IBM/group`

표 35. 32비트 UNIX 및 Linux 시스템의 경우

플러그인 모듈 유형	위치
서버	`/sqllib/security32/plugin/IBM/server`
클라이언트	`/sqllib/security32/plugin/IBM/client`
그룹	`/sqllib/security32/plugin/IBM/group`

표 36. Windows 시스템(64비트 및 32비트)

플러그인 모듈 유형	위치
서버	%DB2PATH%₩security₩plugin₩IBM₩*instance-name*₩server
클라이언트	%DB2PATH%₩security₩plugin₩IBM₩*instance-name*₩client
그룹	%DB2PATH%₩security₩plugin₩IBM₩*instance-name*₩group

주: 64비트 Windows 플러그인 모듈의 파일 이름에는 숫자 64가 포함됩니다.

DB2 명령행 처리기를 사용하여 필요한 플러그인 모듈을 사용할 수 있도록 데이터베이스 관리 프로그램 구성을 갱신하십시오.

- 서버 플러그인 모듈:

  ```
  UPDATE DBM CFG USING SRVCON_PW_PLUGIN IBMLDAPauthserver
  ```

- 클라이언트 플러그인 모듈:

  ```
  UPDATE DBM CFG USING CLNT_PW_PLUGIN IBMLDAPauthclient
  ```

- 그룹 플러그인 모듈:

  ```
  UPDATE DBM CFG USING GROUP_PLUGIN IBMLDAPgroups
  ```

db2 terminate 명령을 사용하여 실행 중인 DB2 명령행 처리기 백엔드 프로세스를 모두 종료한 다음 db2stop 및 db2start 명령을 사용하여 인스턴스를 중지했다가 재시작하십시오.

LDAP 사용자 ID와 연결

DB2 인스턴스에서 LDAP 보안 플러그인이 구성된 후에는 사용자가 다양한 사용자 문자열을 사용하여 데이터베이스에 연결할 수 있습니다.

LDAP 디렉토리 내에서의 오브젝트 위치는 식별 이름(DN)에 의해 정의됩니다. DN은 여러 종류의 계층을 반영하는 다중 파트 이름입니다. 예를 들면 다음과 같습니다.

```
cn=John Smith, ou=Sales, o=WidgetCorp
```

사용자의 *사용자 ID*는 사용자 오브젝트와 연관된 속성(보통 uid 속성)에 의해 정의되며, 단순 문자열(예: jsmith)이거나, 조직 계층 구조의 파트를 반영하는 전자 우편 주소(예: jsmith@sales.widgetcorp.com)일 수 있습니다.

사용자의 DB2 *권한 부여 ID*는 DB2 데이터베이스 내에서 해당 사용자와 연관된 이름입니다.

과거에는 사용자가 서버의 호스트 운영 체제에 정의되어 사용자 ID와 권한 부여 ID가 동일했습니다. 단, 권한 부여 ID는 보통 대문자로 표시되었습니다. DB2 LDAP 플러그인 모듈을 사용하면 LDAP 사용자 오브젝트의 다른 속성을 사용자 ID 및 권한 부여 ID와 연관시킬 수 있습니다. 대부분의 경우 사용자 ID와 권한 부여 ID는 동일한

문자열이므로, USERID_ATTRIBUTE 및 AUTHID_ATTRIBUTE에 동일한 속성 이름을 사용할 수 있습니다. 그러나 사용 중인 환경에서 사용자 ID 속성이 권한 부여 ID에 포함시키고 싶지 않은 추가 정보가 포함되어 있는 경우, 플러그인 초기화 파일에서 다른 AUTHID_ATTRIBUTE를 구성할 수 있습니다. AUTHID_ATTRIBUTE 속성의 값은 서버에서 검색되어 사용자에 대한 내부 DB2 표현으로 사용됩니다.

예를 들어, LDAP 사용자 ID가 전자 우편 주소(예: jsmith@sales.widgetcorp.com)지만 사용자 부분(jsmith)만 DB2 권한 부여 ID로 사용하려는 경우, 다음과 같이 할 수 있습니다.

1. 단축 이름을 포함하는 새 속성을 LDAP에 있는 모든 사용자 오브젝트와 연관시키십시오.
2. 이 새 속성을 사용하여 AUTHID_ATTRIBUTE를 구성하십시오.

그러면 사용자가 자신의 LDAP 사용자 ID와 암호를 지정하여 DB2 데이터베이스에 연결할 수 있습니다.

```
db2 connect to MYDB user 'jsmith@sales.widgetcorp.com' using 'pswd'
```

그러나 내부적으로 DB2 데이터베이스 관리 프로그램은 AUTHID_ATTRIBUTE(이 경우 jsmith)를 사용하여 검색된 단축 이름의 사용자를 참조합니다.

LDAP 플러그인 모듈이 사용 가능해지고 구성된 후에는 사용자가 다음과 같은 다양한 문자열을 사용하여 DB2 데이터베이스에 연결할 수 있습니다.

- 전체 DN. 예를 들어, 다음과 같습니다.

  ```
  connect to MYDB user 'cn=John Smith, ou=Sales, o=WidgetCorp'
  ```

- 부분 DN - 부분 DN 및 해당 검색 기본 DN(정의된 경우)을 사용하는 LDAP 디렉토리 검색이 정확히 하나의 일치 항목을 리턴하는 경우. 예를 들어, 다음과 같습니다.

  ```
  connect to MYDB user 'cn=John Smith' connect to MYDB user uid=jsmith
  ```

- 단순 문자열(등호가 포함되지 않음). 문자열이 USERID_ATTRIBUTE로 규정되며 부분 DN으로 처리됩니다. 예를 들어, 다음과 같습니다.

  ```
  connect to MYDB user jsmith
  ```

주: CONNECT문 또는 ATTACH 명령에 제공되는 문자열이 공백이나 특수 문자를 포함하는 경우, 작은따옴표로 구분해야 합니다.

그룹 찾아보기에 대한 고려사항

그룹 멤버쉽 정보는 LDAP 서버에 사용자 오브젝트의 속성 또는 그룹 오브젝트의 속성으로 표시됩니다.

- 사용자 오브젝트의 속성으로 표시되는 경우

각 사용자 오브젝트에는 GROUP_LOOKUP_ATTRIBUTE라는 속성이 있는데, 이 속성을 쿼리하면 해당 사용자에 대한 모든 그룹 멤버쉽을 검색할 수 있습니다.

- 그룹 오브젝트의 속성으로 표시되는 경우

  각 그룹 오브젝트에는 GROUP_LOOKUP_ATTRIBUTE이라는 속성이 있는데, 이 속성을 사용하면 그룹의 구성원인 모든 사용자 오브젝트를 나열할 수 있습니다. 사용자 오브젝트가 구성원으로 나열되는 모든 그룹을 검색하여 특정 사용자에 대한 그룹을 나열할 수 있습니다.

대부분의 LDAP 서버는 이러한 방식 중 하나로 구성할 수 있지만 일부의 경우 동시에 두 메소드를 모두 지원합니다. LDAP 서버가 구성되는 방식을 판별하려면 LDAP 관리자에게 문의하십시오.

LDAP 플러그인 모듈을 구성하는 경우 GROUP_LOOKUP_METHOD 매개변수를 사용하여 그룹 찾아보기가 수행되는 방식을 지정할 수 있습니다.

- 사용자 오브젝트의 GROUP_LOOKUP_ATTRIBUTE 속성을 사용하여 그룹 멤버쉽을 찾아야 하는 경우, GROUP_LOOKUP_METHOD를 USER_ATTRIBUTE로 설정하십시오.
- 그룹 오브젝트의 GROUP_LOOKUP_ATTRIBUTE 속성을 사용하여 그룹 멤버쉽을 찾아야 하는 경우, GROUP_LOOKUP_METHOD를 SEARCH_BY_DN으로 설정하십시오.

대부분의 LDAP 서버는 그룹 오브젝트의 GROUP_LOOKUP_ATTRIBUTE 속성을 사용하여 멤버쉽을 판별하며, 다음 예와 같이 구성할 수 있습니다.

```
GROUP_LOOKUP_METHOD = SEARCH_BY_DN
GROUP_LOOKUP_ATTRIBUTE = groupOfNames
```

Microsoft Active Directory는 일반적으로 그룹 멤버쉽을 사용자 속성으로 저장하므로 이 예에 표시된 것과 같이 구성할 수 있습니다.

```
GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ATTRIBUTE = memberOf
```

IBM Tivoli Directory Server는 동시에 두 메소드를 모두 지원합니다. 사용자에 대한 그룹 멤버쉽을 쿼리하려면 이 예에 표시된 것과 같이 특수 사용자 속성 ibm-allGroups를 사용할 수 있습니다.

```
GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ATTRIBUTE = ibm-allGroups
```

기타 LDAP 서버는 그룹 멤버쉽을 검색하는 데 도움이 되는 유사한 특수 속성을 제공합니다. 일반적으로 사용자 속성을 통해 멤버쉽을 검색하는 것이 사용자가 그룹으로 나열되는 그룹을 검색하는 것보다 빠릅니다.

LDAP 사용자 인증 또는 그룹 검색 문제점 해결

LDAP 사용자 인증 또는 그룹 검색 중 문제점이 발생한 경우 db2diag 로그 파일과 관리 로그를 확인하면 문제점 해결에 도움되는 정보를 찾을 수 있습니다.

LDAP 플러그인 모듈은 일반적으로 오류가 발생할 경우 LDAP 리턴 코드, 검색 필터 및 기타 유용한 데이터를 로그합니다. LDAP 플러그인 구성 파일에서 DEBUG 옵션을 사용할 수 있는 경우, 플러그인 모듈이 db2diag 로그 파일에 더 많은 정보를 로그합니다. 이 점이 문제점 해결에 도움이 될 수는 있지만, 모든 추가 데이터를 단일 파일에 기록하는 관계로 발생하는 오버헤드로 인해 프로덕션 시스템의 확장된 사용에는 사용하지 않는 것이 좋습니다.

LDAP 플러그인 모듈의 모든 메시지를 캡처할 수 있도록 데이터베이스 관리 프로그램의 DIAGLEVEL 구성 매개변수가 4로 설정되어 있는지 확인하십시오.

보안 플러그인 쓰기

DB2의 보안 플러그인 로드 방식

DB2 데이터베이스 시스템이 보안 플러그인 함수를 호출하는 데 필요한 정보를 포함하게 하려면 보안 플러그인에 초기화 함수가 올바르게 설정되어 있어야 합니다.

각 플러그인 라이브러리는 플러그인 유형으로 판별되는 특정 이름이 포함된 초기화 함수를 포함해야 합니다.

- 서버 측 인증 플러그인: `db2secServerAuthPluginInit()`
- 클라이언트 측 인증 플러그인: `db2secClientAuthPluginInit()`
- 그룹 플러그인: `db2secGroupPluginInit()`

이 함수를 플러그인 초기화 함수라고 합니다. 플러그인 초기화 함수는 지정된 플러그인을 초기화하며 플러그인 함수를 호출하는 데 필요한 정보를 DB2에 제공합니다. 플러그인 초기화 함수에서 사용할 수 있는 매개변수는 다음과 같습니다.

- 플러그인을 호출하는 DB2 인스턴스가 지원할 수 있는 함수 포인터 구조의 가장 높은 버전 번호
- 구현이 필요한 모든 API에 대한 포인터를 포함하는 구조에 대한 포인터
- db2diag 로그 파일에 로그 메시지를 추가하는 함수에 대한 포인터
- 오류 메시지 문자열에 대한 포인터
- 오류 메시지의 길이

다음은 그룹 검색 플러그인의 초기화 함수에 대한 함수 서명입니다.

```
SQL_API_RC SQL_API_FN db2secGroupPluginInit(
  db2int32 version,
  void *group_fns,
  db2secLogMessage *logMessage_fn,
  char **errormsg,
  db2int32  *errormsglen);
```

주: 플러그인 라이브러리가 C++로 컴파일된 경우 `extern "C"`로 모든 함수를 선언해야 합니다. DB2에서는 기본 운영 체제 플러그인 동적 로더를 사용하여 C++ 사용자 작성 플러그인 라이브러리 내에 사용된 C++ 컨스트럭터와 디스트럭터를 처리합니다.

초기화 함수는 플러그인 라이브러리에서 유일하게 규정된 함수 이름을 사용하는 함수입니다. 다른 플러그인 함수는 초기화 함수에서 리턴된 함수 포인트를 통해 참조됩니다. 서버 플러그인은 DB2 서버가 시작될 때 로드되고, 클라이언트 플러그인은 클라이언트에서 필요할 때 로드됩니다. DB2에서는 플러그인 라이브러리를 로드하는 즉시 초기화 함수의 위치를 분석하여 이를 호출합니다. 이 함수의 구체적인 태스크는 다음과 같습니다.

- 함수 포인터를 해당 함수 구조에 대한 포인터로 캐스트합니다.
- 라이브러리에 다른 함수에 대한 포인터를 채웁니다.
- 리턴되는 함수 포인터 구조의 버전 번호를 채웁니다.

DB2는 플러그인 초기화 함수를 여러 번 호출할 수도 있습니다. 이러한 상황은 응용프로그램이 DB2 클라이언트 라이브러리를 동적으로 로드, 언로드 및 재로드했는데 재로드하기 이전과 이후에 플러그인의 인증 함수를 수행하는 경우에 발생할 수 있습니다. 이 경우 플러그인 라이브러리가 언로드된 다음 재로드되지 않을 수 있습니다. 그러나 이 동작은 운영 체제에 따라 다릅니다.

플러그인 초기화 함수를 여러 번 호출하는 DB2의 또 다른 경우는 스토어드 프로시저 또는 페더레이티드 시스템 호출 실행 중에 발생합니다. 이때 데이터베이스 서버 자체는 클라이언트로 사용될 수 있습니다. 데이터베이스 서버에서 클라이언트 및 서버 플러그인이 같은 파일에 있는 경우, DB2에서 플러그인 초기화 함수를 두 번 호출할 수 있습니다.

플러그인에서 `db2secGroupPluginInit`가 여러 번 호출되었음을 발견한 경우, 이 이벤트를 종료한 다음 플러그인 라이브러리를 다시 초기화합니다. 이와 같이 플러그인 초기화 함수는 함수 포인터 세트를 다시 리턴하기 전에 `db2secPluginTerm`에 대한 호출이 수행하는 전체 정리 태스크를 수행해야 합니다.

UNIX 또는 Linux 기반 운영 체제에서 실행 중인 DB2 서버에서 DB2는 여러 프로세스에서 플러그인 라이브러리를 여러 번 로드하고 초기화할 수도 있습니다.

보안 플러그인 라이브러리 개발의 제한사항

플러그인 라이브러리 개발 방법에 영향을 주는 특정 제한사항이 있습니다.

다음은 플러그인 라이브러리 개발의 제한사항입니다.

C-연계 플러그인 라이브러리는 C-연계와 연결되어야 합니다. 프로토타입을 제공하는 헤더 파일, 플러그인을 구현하는 데 필요한 데이터 구조, 오류 코드 정의는 C/C++에만 제공됩니다. DB2가 로드 시간에 해결하는 함수는 외부 "C"로 선언되어야 합니다(플러그인 라이브러리가 C++로 컴파일되는 경우).

.NET 일반 언어 런타임이 지원되지 않음

플러그인 라이브러리의 소스 코드 컴파일 및 연결에 .NET 일반 언어 런타임(CLR)이 지원되지 않습니다.

신호 핸들러

플러그인 라이브러리가 신호 핸들러를 설치하거나 신호 마스크를 변경하지 않아야 합니다. DB2의 신호 핸들러를 방해하기 때문입니다. DB2 신호 핸들러를 방해하면 플러그인 코드 자체의 트랩을 포함하여 오류를 보고하고 복구하는 DB2의 기능에 심각한 방해가 될 수 있습니다. DB2의 오류 처리에 방해가 될 수 있으므로 플러그인 라이브러리에서도 C++ 예외가 발생하지 않아야 합니다.

스레드 안전

플러그인 라이브러리는 스레드 안전해야 하며 재진입 가능해야 합니다. 플러그인 초기화 기능은 다시 입력할 필요가 없는 유일한 API입니다. 플러그인 초기화 기능은 다른 프로세스에서 여러 번 호출될 수 있습니다. 이 경우, 플러그인에서 사용된 모든 자원이 정리되고 플러그인 자체가 다시 초기화됩니다.

Exit 핸들러 및 표준 C 라이브러리 및 운영 체제 호출 겹쳐쓰기

플러그인 라이브러리는 표준 C 라이브러리 또는 운영 체제 호출을 겹쳐쓰지 않아야 합니다. 또한 플러그인 라이브러리는 exit 핸들러 또는 `pthread_atfork` 핸들러를 설치하지 않아야 합니다. 프로그램을 종료하기 전에는 exit 핸들러를 언로드할 수 없으므로 이 핸들러를 사용하지 않는 것이 좋습니다.

라이브러리 종속성

Linux 또는 UNIX에서, 플러그인 라이브러리를 로드하는 프로세스는 `setuid` 또는 `setgid` 중 하나입니다. 즉 `$LD_LIBRARY_PATH`, `$SHLIB_PATH` 또는 `$LIBPATH` 환경 변수를 사용하여 종속 라이브러리를 찾을 수 없음을 의미합니다. 따라서 다음과 같은 다른 방법을 통해 종속 라이브러리에 액세스할 수 없는 경우를 제외하고 플러그인 라이브러리는 추가 라이브러리를 사용할 수 없습니다.

- `/lib` 또는 `/usr/lib`에 배치하여
- 지정되는 OS에 상주하는 디렉토리 사용하여 (예: Linux의 `ld.so.conf` 파일)

- 플러그인 라이브러리 자체의 RPATH에 지정하여

이 제한사항은 Windows 운영 체제에는 적용되지 않습니다.

기호 충돌

가능하면, 기호 충돌 가능성을 줄일 수 있는 옵션(예: 바인딩되지 않은 외부 기호 참조를 줄이는 옵션)을 사용하여 플러그인 라이브러리를 컴파일하고 연결해야 합니다. 예를 들어, HP, Solaris 및 Linux에서 ″-Bsymbolic″ 링커 옵션을 사용하면 기호 충돌과 관련된 문제점을 방지할 수 있습니다. 하지만 AIX에 작성된 플러그인에는 "-brtl" 링커 옵션을 명시적 또는 내재적으로 사용하지 마십시오.

32비트 및 64비트 응용프로그램

32비트 응용프로그램은 32비트 플러그인을 사용해야 하며, 64비트 응용프로그램은 64비트 플러그인을 사용해야 합니다. 자세한 정보는 32비트 및 62비트 고려사항에 대한 주제를 참조하십시오.

텍스트 문자열

입력 텍스트 문자열이 널(null) 종료된다고 보장되지 않으며, 출력 문자열은 널(null) 종료될 필요가 없습니다. 대신, 모든 입력 문자열에 정수 길이가 지정되고 리턴될 길이에 정수의 포인터가 지정됩니다.

권한 부여 ID 매개변수 전달

DB2가 플러그인에 전달하는 권한 부여 ID(권한 ID) 매개변수(입력 권한 ID 매개변수)에 채워진 공백이 제거된 상태의 대문자 권한 ID가 포함됩니다. 플러그인이 DB2에 리턴하는 권한 ID 매개변수(출력 권한 ID 매개변수)는 특수 처리가 필요하지 않지만, DB2가 내부 DB2 표준에 따라 권한 ID를 대문자로 포함시키고 공백으로 채웁니다.

매개변수 *크기* 한계

플러그인 API에서 사용하는 매개변수의 길이 한계는 다음과 같습니다.

```
#define DB2SEC_MAX_AUTHID_LENGTH 255
#define DB2SEC_MAX_USERID_LENGTH 255
#define DB2SEC_MAX_USERNAMESPACE_LENGTH 255
#define DB2SEC_MAX_PASSWORD_LENGTH 255
#define DB2SEC_MAX_DBNAME_LENGTH 128
```

특정 플러그인을 구현하려면 권한 부여 ID, 사용자 ID 및 암호의 최대 길이가 작아야 하거나 작게 설정해야 합니다. 특히, DB2 데이터베이스 시스템에 제공되는 운영 체제 인증 플러그인은 운영 체제 한계가 위에 설명한 한계보다 작은 경우 운영 체제에서 강제 설정된 최대 사용, 그룹 및 이름 스페이스 길이 한계로 제한됩니다.

AIX의 보안 플러그인 라이브러리 확장자

AIX 시스템에서, 보안 플러그인 라이브러리에 확장자가 *.a* 또는 *.so*인 파일 이

름을 사용할 수 있습니다. 플러그인 라이브러리를 로드하는 데 사용되는 메커니즘은 사용되는 확장자에 따라 다릅니다.

- 확장자가 *.a*인 파일 이름이 있는 플러그인 라이브러리는 공유 오브젝트 구성원이 포함된 아카이브로 간주됩니다. 이러한 구성원의 이름은 *shr.o*(32비트) 또는 *shr64.o*(64비트)로 지정해야 합니다. 단일 아카이브는 32비트와 64비트 구성원을 모두 포함할 수 있으므로 두 유형의 플랫폼에 전개할 수 있습니다.

  예를 들어, 32비트 아카이브 스타일 플러그인 라이브러리를 빌드하려면 다음과 같이 구성하십시오.

  ```
  xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
  ar rv MyPlugin.a shr.o
  ```

- 확장자가 *.so*인 파일 이름이 있는 플러그인 라이브러리는 동적으로 로드할 수 있는 공유 오브젝트로 간주됩니다. 이러한 오브젝트는 빌드 시 사용된 컴파일러 또는 링커 옵션에 따라 32비트 또는 64비트입니다. 예를 들어, 32비트 플러그인 라이브러리를 빌드하려면 다음과 같이 구성하십시오.

  ```
  xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
  ```

AIX를 제외한 모든 플랫폼에서, 보안 플러그인 라이브러리는 항상 동적으로 로드할 수 있는 공유 오브젝트로 간주됩니다.

보안 플러그인에 대한 제한사항

보안 플러그인 사용과 관련하여 몇 가지 제한사항이 있습니다.

DB2 데이터베이스 계열 지원 제한사항

GSS-API 플러그인을 사용하여 Linux, UNIX 및 Windows에 설치된 DB2 클라이언트와 다른 DB2 계열 서버(예: z/OS용 DB2) 간의 연결을 인증할 수 없습니다. 또한 클라이언트로 사용되는 다른 DB2 데이터베이스 계열 제품과 Linux, UNIX 또는 Windows에 설치된 DB2 서버 간의 연결도 인증할 수 없습니다.

Linux, UNIX 또는 Windows에서 DB2 클라이언트를 사용하여 다른 DB2 데이터베이스 계열 서버에 연결하는 경우, 클라이언트 측 사용자 ID/암호 플러그인(예: IBM에서 제공하는 운영 체제 인증 플러그인)을 사용하거나 고유한 사용자 ID/암호 플러그인을 작성할 수 있습니다. 내장 Kerberos 플러그인을 사용하거나 고유한 플러그인을 구현할 수도 있습니다.

Linux, UNIX 또는 Windows에서 DB2 클라이언트를 사용할 경우에는 GSSPLUGIN 인증 유형을 사용하여 데이터베이스를 카탈로그해서는 안됩니다.

AUTHID 식별자에 대한 제한사항 DB2 데이터베이스 시스템의 버전 9.5 이상에서는 128바이트의 권한 부여 ID를 사용할 수 있지만, 권한 부여 ID가 운영 체제 사용자 ID 또는 그룹 이름으로 해석되는 경우 운영 체제 이름 지정 제한사항이 적용됩니다(예: 8

또는 30자 사용자 ID 및 30자 그룹 이름 제한). 따라서 128바이트의 권한 부여 ID를 부여할 수는 있지만 해당 권한 부여 ID를 사용하는 사용자로 연결할 수는 없습니다. 사용자 고유의 보안 플러그인을 작성하는 경우에는 확장된 크기의 권한 부여 ID를 사용할 수 있습니다. 예를 들어, 보안 플러그인에 30바이트 사용자 ID를 제공할 수 있으며, 이 보안 플러그인은 연결 시 사용할 수 있는 128바이트 권한 부여 ID를 인증 중에 리턴할 수 있습니다.

InfoSphere™ Federation Server 지원 제한사항

DB2 II에서는 GSS_API 플러그인의 위임된 증명서를 사용하여 데이터 소스에 대한 아웃바운드 연결을 설정할 수 없습니다. 데이터 소스에 대한 연결은 계속 CREATE USER MAPPING 명령을 사용해야 합니다.

Database Administration Server 지원 제한사항

DAS(DB2 Administration Server)에서는 보안 플러그인이 지원되지 않습니다. DAS는 운영 체제 인증 메커니즘만 지원합니다.

DB2 클라이언트(Windows용) 보안 플러그인 문제점 및 제한사항

Windows 운영 체제의 DB2 클라이언트에 배치될 보안 플러그인을 개발하는 경우 플러그인 종료 함수에서 보조 라이브러리를 언로드하지 마십시오. 이 제한사항은 그룹, 사용자 ID 및 암호, Kerberos, GSS-API 플러그인 등 모든 유형의 클라이언트 보안 플러그인에 적용됩니다. 이러한 종료 API(예: db2secPluginTerm, db2secClientAuthPluginTerm 및 db2secServerAuthPluginTerm)는 Windows 플랫폼에서 호출되지 않으므로, 적절히 자원을 정리해야 합니다.

이 제한사항은 Windows에서 DLL 언로드와 연관된 정리 문제와 관련이 있습니다.

AIX에서 확장자가 .a 또는 .so인 플러그인 라이브러리 로드

AIX에서 보안 플러그인 라이브러리의 파일 이름 확장자는 `.a` 또는 `.so`입니다. 플러그인 라이브러리를 로드하는 데 사용되는 메커니즘은 사용되는 확장자에 따라 다릅니다.

- 파일 이름 확장자가 `.a`인 플러그인 라이브러리

  파일 이름 확장자가 .a인 플러그인 라이브러리는 공유 오브젝트 구성원을 포함하는 아카이브로 간주됩니다. 이러한 구성원의 이름은 `shr.o`(32비트) 또는 `shr64.o`(64비트)여야 합니다. 단일 아카이브는 32비트와 64비트 구성원을 모두 포함할 수 있으므로 두 유형의 플랫폼에 전개할 수 있습니다.

  예를 들어, 32비트 아카이브 스타일 플러그인 라이브러리를 빌드하려면 다음과 같이 구성하십시오.

```
xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
    ar rv MyPlugin.a shr.o
```

- 파일 이름 확장자가 .so인 플러그인 라이브러리

  파일 이름 확장자가 .so인 플러그인 라이브러리는 동적으로 로드 가능한 공유 오브젝트로 간주됩니다. 이러한 오브젝트는 빌드 시 사용된 컴파일러 또는 링커 옵션에 따라 32비트 또는 64비트입니다. 예를 들어, 32비트 플러그인 라이브러리를 빌드하려면 다음과 같이 구성하십시오.

```
xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
```

AIX를 제외한 모든 플랫폼에서, 보안 플러그인 라이브러리는 항상 동적으로 로드할 수 있는 공유 오브젝트로 간주됩니다.

GSS-API 보안 플러그인은 메시지 암호화 및 서명을 지원하지 않음

GSS-API 보안 플러그인에서는 메시지 암호화 및 서명을 사용할 수 없습니다.

보안 플러그인의 리턴 코드

모든 보안 플러그인 API는 API 실행의 성공 또는 실패를 나타내는 정수 값을 리턴해야 합니다. 리턴 코드 값 0은 API 실행에 성공했음을 나타냅니다. -3, -4 및 -5를 제외한 모든 음수 리턴 코드는 API에 오류가 발생했음을 나타냅니다.

리턴 코드 -3, -4 또는 -5를 제외하고 보안 플러그인 API에서 리턴된 모든 음수 리턴 코드는 SQLCODE -1365, SQLCODE -1366 또는 SQLCODE -30082에 맵핑됩니다. -3, -4 및 -5 값은 권한 부여 ID가 유효한 사용자 또는 그룹을 나타내는지 여부를 표시하는 데 사용됩니다.

모든 보안 플러그인 API 리턴 코드는 SQLLIB/include 디렉토리를 포함하여 DB2에서 볼 수 있는 db2secPlugin.h에 정의됩니다.

다음 표에서 모든 보안 플러그인 리턴 코드에 대한 세부사항을 볼 수 있습니다.

표 37. 보안 플러그인 리턴 코드

리턴 코드	값 정의	의미	적용 가능한 API
0	DB2SEC_PLUGIN_OK	플러그인 API 실행에 성공했습니다.	모두
-1	DB2SEC_PLUGIN_UNKNOWNERROR	플러그인 API에 예기치 않은 오류가 발생했습니다.	모두
-2	DB2SEC_PLUGIN_BADUSER	입력하여 제공된 사용자 ID가 정의되지 않았습니다.	db2secGenerateInitialCred db2secValidatePassword db2secRemapUserid db2secGetGroupsForUser

표 37. 보안 플러그인 리턴 코드 (계속)

리턴 코드	값 정의	의미	적용 가능한 API
-3	DB2SEC_PLUGIN_INVALIDUSERORGROUP	해당 사용자 또는 그룹이 없습니다.	db2secDoesAuthIDExist db2secDoesGroupExist
-4	DB2SEC_PLUGIN_USERSTATUSNOTKNOWN	알 수 없는 사용자 상태입니다. 이는 DB2에서 오류로 취급되지 않습니다. GRANT문에 사용되어 권한 ID가 사용자 또는 운영 체제 그룹을 나타내는지 판별하는 데 사용됩니다.	db2secDoesAuthIDExist
-5	DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN	알 수 없는 그룹 상태입니다. 이는 DB2에서 오류로 취급되지 않습니다. GRANT문에 사용되어 권한 ID가 사용자 또는 운영 체제 그룹을 나타내는지 판별하는 데 사용됩니다.	db2secDoesGroupExist
-6	DB2SEC_PLUGIN_UID_EXPIRED	만기된 사용자 ID입니다.	db2secValidatePassword db2GetGroupsForUser db2secGenerateInitialCred
-7	DB2SEC_PLUGIN_PWD_EXPIRED	만기된 암호입니다.	db2secValidatePassword db2GetGroupsForUser db2secGenerateInitialCred
-8	DB2SEC_PLUGIN_USER_REVOKED	권한 취소된 사용자입니다.	db2secValidatePassword db2GetGroupsForUser
-9	DB2SEC_PLUGIN_USER_SUSPENDED	일시중단된 사용자입니다.	db2secValidatePassword db2GetGroupsForUser
-10	DB2SEC_PLUGIN_BADPWD	잘못된 암호입니다.	db2secValidatePassword db2secRemapUserid db2secGenerateInitialCred
-11	DB2SEC_PLUGIN_BAD_NEWPASSWORD	잘못된 새 암호입니다.	db2secValidatePassword db2secRemapUserid
-12	DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED	암호 변경이 지원되지 않습니다.	db2secValidatePassword db2secRemapUserid db2secGenerateInitialCred
-13	DB2SEC_PLUGIN_NOMEM	메모리가 부족하여 플러그인의 메모리 할당 시도에 실패했습니다.	모두
-14	DB2SEC_PLUGIN_DISKERROR	플러그인에 디스크 오류가 발생했습니다.	모두
-15	DB2SEC_PLUGIN_NOPERM	파일에 잘못된 권한이 있기 때문에 플러그인의 파일 액세스 시도에 실패했습니다.	모두

표 37. 보안 플러그인 리턴 코드 (계속)

리턴 코드	값 정의	의미	적용 가능한 API
-16	DB2SEC_PLUGIN_NETWORKERROR	플러그인에 네트워크 오류가 발생했습니다.	모두
-17	DB2SEC_PLUGIN _CANTLOADLIBRARY	플러그인이 필수 라이브러리를 로드할 수 없습니다.	db2secGroupPluginInit db2secClientAuthPluginInit db2secServerAuthPluginInit
-18	DB2SEC_PLUGIN_CANT _OPEN_FILE	파일 누락 또는 파일 권한 부족 이외의 이유로 인해 플러그인이 파일을 열고 읽을 수 없습니다.	모두
-19	DB2SEC_PLUGIN_FILENOTFOUND	파일 시스템에 파일이 누락되어 플러그인이 파일을 열고 읽을 수 없습니다.	모두
-20	DB2SEC_PLUGIN _CONNECTION_DISALLOWED	연결이 허용되는 데이터베이스 또는 TCP/IP 주소가 특정 데이터베이스에 연결할 수 없는 제한사항으로 인해 플러그인이 연결을 거부합니다.	모든 서버 측 플러그인 API
-21	DB2SEC_PLUGIN_NO_CRED	GSS API 플러그인에만 해당: 초기 클라이언트 증명서가 누락되었습니다.	db2secGetDefaultLoginContext db2secServerAuthPluginInit
-22	DB2SEC_PLUGIN_CRED_EXPIRED	GSS API 플러그인에만 해당: 클라이언트 증명서가 만기되었습니다.	db2secGetDefaultLoginContext db2secServerAuthPluginInit
-23	DB2SEC_PLUGIN _BAD_PRINCIPAL_NAME	GSS API 플러그인에만 해당: 핵심부 이름이 유효하지 않습니다.	db2secProcessServer PrincipalName
-24	DB2SEC_PLUGIN _NO_CON_DETAILS	이 리턴 코드는 db2secGetConDetails 콜백에서 리턴되어 (예: DB2에서 플러그인으로) DB2가 클라이언트의 TCP/IP 주소를 판별할 수 없음을 나타냅니다.	db2secGetConDetails
-25	DB2SEC_PLUGIN _BAD_INPUT_PARAMETERS	플러그인 API가 호출될 때 일부 매개변수가 유효하지 않거나 누락되었습니다.	모두
-26	DB2SEC_PLUGIN _INCOMPATIBLE_VER	플러그인에서 보고한 API 버전을 DB2에 사용할 수 없습니다.	db2secGroupPluginInit db2secClientAuthPluginInit db2secServerAuthPluginInit
-27	DB2SEC_PLUGIN_PROCESS_LIMIT	플러그인에 충분하지 않은 자원을 사용하여 새 프로세스를 작성할 수 있습니다.	모두
-28	DB2SEC_PLUGIN_NO_LICENSES	플러그인에 사용자 허가 문제점이 발생했습니다. 기본 메커니즘 라이센스가 한계에 도달했을 가능성이 있습니다.	모두
-29	DB2SEC_PLUGIN_ROOT_NEEDED	플러그인이 루트 특권이 필요한 응용프로그램을 실행하려고 시도합니다.	모두

표 37. 보안 플러그인 리턴 코드 (계속)

리턴 코드	값 정의	의미	적용 가능한 API
-30	DB2SEC_PLUGIN_UNEXPECTED_SYSTEM_ERROR	플러그인에 예기치 않은 시스템 오류가 발생했습니다. 현재 시스템 구성이 지원되지 않을 가능성이 있습니다.	모두

보안 플러그인의 오류 메시지 조절

보안 플러그인 API에 오류가 발생하면, API가 errormsg 필드의 ASCII 텍스트 문자열을 리턴하여 리턴 코드보다 많은 특정 문제점 설명을 제공할 수 있습니다.

예를 들어, errormsg 문자열에는 "File /home/db2inst1/mypasswd.txt does not exist"가 있을 수 있습니다. DB2가 이 문자열 전체를 DB2 관리 통지 로드에 쓰며 일부 SQL 메시지에 잘린 버전을 토큰으로 포함할 수도 있습니다. SQL 메시지의 토큰은 제한된 길이로만 포함할 수 있으므로, 이러한 메시지는 짧아야 하며 메시지의 중요 변수 부분이 문자열 앞에 표시되어야 합니다. 디버깅을 위해 오류 메시지에 보안 플러그인 이름을 추가하는 것이 좋습니다.

암호 만기 오류 등과 같은 긴급하지 않은 오류의 경우, errormsg 문자열은 DIAGLEVEL 데이터베이스 관리 프로그램 구성 매개변수가 4로 설정된 경우에만 덤프됩니다.

이러한 오류 메시지의 메모리는 보안 플러그인이 할당해야 합니다. 따라서 플러그인은 API를 제공하여 db2secFreeErrormsg에 대한 메모리를 제거해야 합니다.

API가 0이 아닌 값을 리턴하는 경우에만 DB2에서 errormsg 필드가 선택됩니다. 따라서 오류가 없는 경우에는 플러그인은 리턴된 이 오류 메시지에 메모리를 할당하지 않아야 합니다.

초기화할 때, 메시지 로깅 함수 포인터인 logMessage_fn이 그룹, 클라이언트 및 서버 플러그인에 전달됩니다. 플러그인은 이 함수를 사용하여 db2diag 로그 파일에 디버깅 정보를 로그할 수 있습니다. 예를 들어, 다음과 같습니다.

```
// Log an message indicate init successful
(*(logMessage_fn))(DB2SEC_LOG_CRITICAL,
                   "db2secGroupPluginInit successful",
                    strlen("db2secGroupPluginInit successful"));
```

db2secLogMessage 함수의 각 매개변수에 대한 세부사항은 각 플러그인 유형의 API 초기화를 참조하십시오.

보안 플러그인 API의 호출 시퀀스

DB2 데이터베이스 관리 프로그램이 보안 플러그인 API를 호출하는 시퀀스는 보안 플러그인 API가 호출되는 시나리오에 따라 다릅니다.

다음은 DB2 데이터베이스 관리 프로그램이 보안 플러그인 API를 호출하는 기본 시나리오입니다.

- 데이터베이스 연결을 위해 클라이언트에서(내재적 및 명시적)
  - CLIENT
  - 서버 기반(SERVER, SERVER_ENCRYPT, DATA_ENCRYPT)
  - GSSAPI 및 Kerberos
- 로컬 권한 부여를 위해 클라이언트, 서버 또는 게이트웨이에서
- 데이터베이스 연결을 위해 서버에서
- GRANT문을 위해 서버에서
- 권한 부여 ID가 속하는 그룹의 목록을 가져오기 위해 서버에서

주: DB2 데이터베이스 서버는 로컬 권한 부여가 필요한 데이터베이스 조치(예: db2start, db2stop 및 db2trc)를 클라이언트 응용프로그램처럼 취급합니다.

이러한 각 조작에 대해, DB2 데이터베이스 관리 프로그램이 보안 플러그인 API를 호출하는 시퀀스는 다릅니다. 다음은 각 시나리오에서 DB2 데이터베이스 관리 프로그램이 API를 호출하는 시퀀스입니다.

CLIENT - 내재적

사용자가 구성한 인증 유형이 CLIENT이면, DB2 클라이언트 응용프로그램은 다음과 같은 보안 플러그인 API를 호출합니다.

- `db2secGetDefaultLoginContext();`
- `db2secValidatePassword();`
- `db2secFreetoken();`

내재적 인증의 경우, 특정 사용자 ID 또는 암호를 지정하지 않고 연결할 때 사용자 ID/암호 플러그인을 사용하면 `db2secValidatePassword` API가 호출됩니다. 필요한 경우 플러그인 개발자는 이 API를 사용하여 내재적 인증을 금지할 수 있습니다.

CLIENT - 명시적

명시적 인증 즉, 사용자 ID와 암호 둘 다 지정되어 있는 데이터베이스에 연결하려고 할 때 *authentication* 데이터베이스 관리 프로그램 구성 매개변수가 CLIENT로 설정되어 있으면 DB2 클라이언트 응용프로그램은 구현에 필요할 경우 다음과 같은 보안 플러그인 API를 여러 번 호출합니다.

- `db2secRemapUserid();`
- `db2secValidatePassword();`
- `db2secFreeToken();`

Server 기반(SERVER, SERVER_ENCRYPT, DATA_ENCRYPT) - 내재적

내재적 인증에서, 클라이언트 및 서버에 협상된 사용자 ID/암호 인증이 있는 경

우(예: 서버에서 *srvcon_auth* 매개변수가 SERVER; SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP로 설정되어 있는 경우), 클라이언트 응용프로그램이 다음과 같은 보안 플러그인 API를 호출합니다.

- `db2secGetDefaultLoginContext();`
- `db2secFreeToken();`

Server 기반(SERVER, SERVER_ENCRYPT, DATA_ENCRYPT) - 명시적

명시적 인증에서, 클라이언트 및 서버에 협상된 사용자 ID/암호 인증이 있는 경우(예: 서버에서 srvcon_auth 매개변수가 SERVER; SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP로 설정되어 있는 경우), 클라이언트 응용프로그램이 1 다음과 같은 보안 플러그인 API를 호출합니다.

- `db2secRemapUserid();`

GSSAPI 및 Kerberos - 내재적

내재적 인증에서, 클라이언트 및 서버에 협상된 GSS-API 또는 Kerberos 인증이 있는 경우(예: 서버에서 *srvcon_auth* 매개변수가 KERBEROS; KRB_SERVER_ENCRYPT, GSSPLUGIN 또는 GSS_SERVER_ENCRYPT로 설정되어 있는 경우), 클라이언트 응용프로그램이 다음과 같은 보안 플러그인 API를 호출합니다. (gss_init_sec_context() 호출에서는 GSS_C_NO_CREDENTIAL을 입력 증명서로 사용합니다.)

- `db2secGetDefaultLoginContext();`
- `db2secProcessServerPrincipalName();`
- `gss_init_sec_context();`
- `gss_release_buffer();`
- `gss_release_name();`
- `gss_delete_sec_context();`
- `db2secFreeToken();`

멀티플로우 GSS-API 지원 기능을 사용하면, 구현에 필요할 경우 `gss_init_sec_context()`를 여러 번 호출할 수 있습니다.

GSSAPI 및 Kerberos - 명시적

협상된 인증 유형이 GSS-API 또는 Kerberos일 경우, 클라이언트 응용프로그램은 다음과 같은 순서로 GSS-API 플러그인에 사용할 보안 플러그인 API를 호출합니다. 이러한 API는 별도의 설명이 없는 한 내재적 인증 및 명시적 인증 둘 다에 사용됩니다.

- `db2secProcessServerPrincipalName();`
- `db2secGenerateInitialCred();`(명시적 인증에만 해당)
- `gss_init_sec_context();`
- `gss_release_buffer ();`

- gss_release_name();
- gss_release_cred();
- db2secFreeInitInfo();
- gss_delete_sec_context();
- db2secFreeToken();

상호 인증 토큰이 서버에서 리턴되고 구현에서 필요할 경우 API gss_init_sec_context()를 여러 번 호출할 수 있습니다.

로컬 권한 부여를 위해 클라이언트, 서버 또는 게이트웨이에서

로컬 권한 부여의 경우, 사용되는 DB2 명령이 다음과 같은 보안 플러그인 API를 호출합니다

- db2secGetDefaultLoginContext();
- db2secGetGroupsForUser();
- db2secFreeToken();
- db2secFreeGroupList();

이러한 API는 사용자 ID/암호 및 GSS-API 인증 메커니즘에 호출됩니다.

데이터베이스 연결을 위해 서버에서

데이터베이스 서버의 데이터베이스 연결의 경우, DB2 에이전트 프로세스 또는 스레드가 사용자 ID/암호 인증 메커니즘에 다음과 같은 보안 플러그인 API를 호출합니다.

- db2secValidatePassword(); *authentication* 데이터베이스 구성 매개변수가 CLIENT가 아닐 경우에만
- db2secGetAuthIDs();
- db2secGetGroupsForUser();
- db2secFreeToken();
- db2secFreeGroupList();

데이터베이스의 CONNECT의 경우, DB2 에이전트 프로세스 또는 스레드가 GSS-API 인증 메커니즘에 다음과 같은 보안 플러그인 API를 호출합니다.

- gss_accept_sec_context();
- gss_release_buffer();
- db2secGetAuthIDs();
- db2secGetGroupsForUser();
- gss_delete_sec_context();
- db2secFreeGroupListMemory();

GRANT문을 위해 서버에서

USER 또는 GROUP 키워드를 지정하지 않는 GRANT문의 경우 (예: "GRANT CONNECT ON DATABASE TO user1"), DB2 에이전트 프로세스 또는 스레드는 user1이 사용자, 그룹 또는 둘 다인지 판별할 수 있어야 합니다. 따라서 DB2 에이전트 프로세스 또는 스레드가 다음과 같은 보안 플러그인 API를 호출합니다.

- db2secDoesGroupExist();
- db2secDoesAuthIDExist();

권한 ID가 속하는 그룹의 목록을 가져오기 위해 서버에서

데이터베이스 서버에서 권한 부여 ID가 속하는 그룹의 목록을 가져와야 하는 경우, DB2 에이전트 프로세스 또는 스레드가 권한 부여 ID만을 입력으로 사용하여 다음과 같은 보안 플러그인 API를 호출합니다.

- db2secGetGroupsForUser();

다른 보안 플러그인의 토큰은 없습니다.

제 8 장 보안 플러그인 API

DB2 데이터베이스 시스템 인증 및 그룹 멤버쉽 찾아보기 동작을 사용자 정의할 수 있도록 하기 위해, DB2 데이터베이스 시스템에서는 기존 플러그인 모듈을 수정하거나 보안 플러그인 모듈을 새로 작성하는 데 사용할 수 있는 API를 제공합니다.

보안 플러그인 모듈을 개발 중인 경우, DB2 데이터베이스 관리 프로그램에서 호출할 표준 인증 또는 그룹 멤버쉽 찾아보기 함수를 구현해야 합니다. 사용 가능한 세 가지 유형의 플러그인 모듈에 대해 구현해야 할 기능은 다음과 같습니다.

그룹 검색
: 제공된 사용자에 대한 그룹 멤버쉽 정보를 검색하고 제공된 문자열이 유효한 사용자인지 판별합니다.

사용자 ID/암호 인증
: 디폴트 보안 컨텍스트를 식별하고(클라이언트만 해당), 암호를 검증하고 선택적으로 변경하며, 제공된 문자열이 유효한 사용자인지 판별하고(서버만 해당), 클라이언트에 제공된 사용자 ID나 암호를 서버로 보내기 전에 수정하며(클라이언트만 해당), 제공된 사용자와 연관된 DB2 권한 부여 ID를 리턴하는 인증입니다.

GSS-API 인증
: 필수 GSS-API 함수를 구현하고, 디폴트 보안 컨텍스트를 식별하며(클라이언트만 해당), 사용자 ID와 암호를 기준으로 초기 증명서를 생성하고, 선택적으로 암호를 변경하며(클라이언트만 해당), 보안 티켓을 작성 및 허용하고, 제공된 GSS-API 보안 컨텍스트와 연관된 DB2 권한 부여 ID를 리턴하는 인증입니다.

다음은 플러그인 API 설명에 사용된 용어에 대한 정의입니다.

플러그인
: 사용자가 작성한 인증 또는 그룹 멤버쉽 찾아보기 함수에 액세스하기 위해 DB2에서 로드하는 동적으로 로드 가능한 라이브러리입니다.

내재된 인증
: 사용자 ID나 암호를 지정하지 않고 데이터베이스에 연결하는 것입니다.

명시적 인증
: 사용자 ID와 암호를 지정하여 데이터베이스에 연결하는 것입니다.

권한 ID
: 데이터베이스 내에 있는 권한과 특권이 부여되는 개인 또는 그룹을 나타내는 내부 ID입니다. 내부적으로 DB2 권한 ID는 대문자로 변환되며 최소 8자(8자까

지 공백으로 채워짐)입니다. 현재 DB2에서는 권한 ID, 사용자 ID, 암호, 그룹 이름, 이름 스페이스 및 도메인 이름을 7비트 ASCII로 나타낼 수 있어야 합니다.

로컬 권한 부여

사용자가 데이터베이스 관리 프로그램 시작 및 중지, DB2 추적 설정 및 해제 또는 데이터베이스 관리 프로그램 구성 갱신 등의 조치(데이터베이스에 연결하는 작업 아님)를 수행할 수 있는 권한이 있는지 확인하는 서버 또는 클라이언트로 국한되는 권한 부여입니다.

이름 스페이스

고유한 개별 사용자 ID를 포함해야 하는 사용자 콜렉션 또는 그룹입니다. 일반적인 예로는 Windows 도메인 및 Kerberos 범주가 있습니다. 예를 들어, Windows 도메인n ″usa.company.com″의 모든 사용자 이름은 고유해야 합니다(예: ″user1@usa.company.com″). 그러나 ″user1@canada.company.com″의 경우와 같이 다른 도메인에 있는 동일한 사용자 ID는 다른 사람을 나타냅니다. 완전한 사용자 ID에는 사용자 ID와 이름 스페이스 쌍(예: ″user@domain.name″ 또는 ″domain₩user″)이 포함됩니다.

입력 DB2가 보안 플러그인 API 매개변수에 대한 값을 채우는 것을 나타냅니다.

출력 보안 플러그인 API가 API 매개변수의 값을 채우는 것을 나타냅니다.

그룹 검색 플러그인용 API

그룹 검색 플러그인 모듈의 경우, 다음과 같은 API를 구현해야 합니다.

- db2secGroupPluginInit

  주: db2secGroupPluginInit API는 다음과 같은 프로토타입을 사용하여 API에 포인터 *logMessage_fn을 입력으로 사용합니다.

```
SQL_API_RC (SQL_API_FN db2secLogMessage)
(
db2int32 level,
void    *data,
db2int32 length
);
```

  플러그인은 db2secLogMessage API를 사용하여 디버깅 또는 정보용으로 db2diag 로그 파일에 메시지를 로그할 수 있습니다. 이 API는 DB2 데이터베이스 시스템이 제공하므로 구현할 필요는 없습니다.

- db2secPluginTerm
- db2secGetGroupsForUser
- db2secDoesGroupExist
- db2secFreeGroupListMemory

- db2secFreeErrormsg
- 외부적으로 해결해야 하는 유일한 API는 db2secGroupPluginInit입니다. 이 API는 void * 매개변수를 사용하므로 해당 유형으로 캐스트해야 합니다.

```
typedef struct db2secGroupFunctions_1
{
db2int32 version;
db2int32 plugintype;
SQL_API_RC (SQL_API_FN * db2secGetGroupsForUser)
(
const char *authid,
db2int32    authidlen,
const char *userid,
db2int32    useridlen,
const char *usernamespace,
db2int32    usernamespacelen,
db2int32    usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
const void *token,
db2int32    tokentype,
db2int32    location,
const char *authpluginname,
db2int32    authpluginnamelen,
void      **grouplist,
db2int32   *numgroups,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secDoesGroupExist)
(
const char *groupname,
db2int32    groupnamelen,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeGroupListMemory)
(
void     *ptr,
char    **errormsg,
db2int32 *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *msgtobefree
);

SQL_API_RC (SQL_API_FN * db2secPluginTerm)
(
char    **errormsg,
db2int32 *errormsglen
);

} db2secGroupFunctions_1;
```

db2secGroupPluginInit API는 외부적으로 사용할 수 있는 나머지 함수에 주소를 지정합니다.

주: _1은 API 버전 1에 해당하는 구조임을 나타냅니다. 이후 인터페이스 버전에서의 확장명은 _2, _3 등이 사용됩니다.

db2secDoesGroupExist API - 그룹이 존재하는지 여부 확인

authid가 그룹을 나타내는지 여부를 판별합니다.

그룹 이름이 존재하면, API는 DB2SEC_PLUGIN_OK 값을 리턴하여 성공을 나타내야 합니다. 또한 그룹 이름이 유효하지 않으면 API가 DB2SEC_PLUGIN_INVALIDUSERORGROUP 값을 리턴할 수 있어야 합니다. 입력이 유효한 그룹인지 판별할 수 없는 경우 API가 DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN 값을 리턴할 수 있습니다. 유효하지 않은 그룹(DB2SEC_PLUGIN_INVALIDUSERORGROUP) 또는 알 수 없는 그룹(DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN) 값이 리턴되면, DB2는 USER 및 GROUP 키워드를 사용하지 않고 GRANT문을 발급할 때 authid가 그룹인지, 사용자인지 판별하지 못할 수 있습니다. 이로 인해 SQLCODE -569, SQLSTATE 56092 오류가 사용자에게 리턴됩니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secDoesGroupExist)
                       ( const char *groupname,
                         db2int32 groupnamelen,
                         char      **errormsg,
                         db2int32 *errormsglen );
```

db2secDoesGroupExist API 매개변수

groupname

입력. 뒤 공백이 없는 대문자 authid.

groupnamelen

입력. groupname 매개변수 값의 길이(바이트).

errormsg

출력. db2secDoesGroupExist API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secFreeErrormsg API - 오류 메시지 메모리 제거

이전 API 호출의 오류 메시지를 보유하는 데 사용된 메모리를 제거합니다. 오류 메시지를 리턴하지 않는 유일한 API입니다. 이 API가 오류를 리턴하면 DB2가 이 오류를 로그하고 계속 진행합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secFreeErrormsg)
                        ( char *errormsg );
```

db2secFreeErrormsg API parameters

msgtofree

입력. 이전 API 호출에서 할당된 메모리의 포인터.

db2secFreeGroupListMemory API - 그룹 목록 메모리 제거

이전 db2secGetGroupsForUser API 호출의 그룹 목록을 보유하는 데 사용된 메모리를 제거합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secFreeGroupListMemory)
                        ( void *ptr,
                          char      **errormsg,
                          db2int32 *errormsglen );
```

db2secFreeGroupListMemory API 매개변수

ptr 입력. 사용 가능해지는 메모리의 포인터.

errormsg

출력. db2secFreeGroupListMemory API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secGetGroupsForUser API - 사용자의 그룹 목록 가져오기

사용자가 속해 있는 그룹의 목록을 리턴합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secGetGroupsForUser)
                        ( const char *authid,
                          db2int32 authidlen,
                          const char *userid,
                          db2int32 useridlen,
                          const char *usernamespace,
```

```
db2int32 usernamespacelen,
db2int32 usernamespacetype,
const char *dbname,
db2int32 dbnamelen,
void *token,
db2int32 tokentype,
db2int32 location,
const char *authpluginname,
db2int32 authpluginnamelen,
void      **grouplist,
db2int32 *numgroups,
char      **errormsg,
db2int32 *errormsglen );
```

db2secGetGroupsForUser API 매개변수

authid

입력. 이 매개변수 값은 SQL 권한 ID입니다. 즉, DB2가 이 값을 뒤에 공백이 없는 대문자 문자열로 변환합니다. DB2는 authid 매개변수에 항상 널(NULL)이 아닌 값을 제공합니다. API는 다른 입력 매개변수와 관계없이 권한 ID가 속하는 그룹 목록을 리턴해야 합니다. 판별할 수 없는 경우 간단한 목록 또는 빈 목록을 리턴할 수 있습니다.

사용자가 존재하지 않는 경우 API는 리턴 코드 DB2SEC_PLUGIN_BADUSER을 리턴해야 합니다. 권한 ID에 연관된 그룹이 없을 수 있으므로 DB2에서는 사용자가 존재하지 않는 상황을 오류로 취급하지 않습니다. 예를 들어, db2secGetAuthids API는 운영 체제에 존재하지 않는 권한 ID를 리턴할 수 있습니다. 권한 ID가 그룹에 연관되어 있지 않지만 직접 특권을 부여할 수 있기 때문입니다.

API가 권한 ID만 사용하여 그룹의 전체 목록을 리턴할 수 없으면, 그룹 지원과 관련된 특정 SQL 함수에 몇 가지 제한사항이 있습니다. 가능한 문제점 시나리오 목록을 보려면 이 주제에 있는 사용법 참고 사항 절을 참조하십시오.

authidlen

입력. authid 매개변수 값의 길이(바이트). DB2 데이터베이스 관리 프로그램은 authidlen 매개변수에 항상 0이 아닌 값을 제공합니다.

userid 입력. 권한 ID에 해당하는 사용자 ID. 이 API가 비연결 시나리오의 서버에서 호출되면, DB2가 이 매개변수를 입력하지 않습니다.

useridlen

입력. userid 매개변수 값의 길이(바이트).

usernamespace

입력. 사용자 ID를 가져온 이름 스페이스.사용자 ID를 사용할 수 없으면, DB2 데이터베이스 관리 프로그램이 이 매개변수를 입력하지 않습니다.

usernamespacelen

입력. usernamespace 매개변수 값의 길이(바이트).

usernamespacetype

입력. 이름 스페이스 유형. db2secPlugin.h에 정의된 usernamespacetype 매개변수에 유효한 값은 다음과 같습니다.

- DB2SEC_NAMESPACE_SAM_COMPATIBLE: 사용자 이름 스타일에 해당 (예: domain₩myname)
- DB2SEC_NAMESPACE_USER_PRINCIPAL: 사용자 이름 스타일에 해당 (예: myname@domain.ibm.com)

현재 DB2 데이터베이스 시스템은
DB2SEC_NAMESPACE_SAM_COMPATIBLE 값만 지원합니다. 사용자 ID를 사용할 수 없는 경우, usernamespacetype 매개변수가 db2secPlugin.h에 정의된 DB2SEC_USER_NAMESPACE_UNDEFINED 값으로 설정됩니다.

dbname

입력. 연결할 데이터베이스의 이름. 비연결 시나리오에서 이 매개변수는 NULL이 될 수 있습니다.

dbnamelen

입력. dbname 매개변수 값의 길이(바이트). 비연결 시나리오에서 dbname 매개변수가 NULL이면 이 매개변수가 0으로 설정됩니다.

token 입력. 인증 플러그인이 제공하는 데이터의 포인터. DB2에는 사용되지 않습니다. 플러그인 기록기에 사용자 및 그룹 정보를 조정할 수 있는 기능을 제공합니다. 이 매개변수가 제공되지 않는 경우도 있습니다(예: 비연결 시나리오). 이 경우 값은 NULL이 됩니다. 사용되는 인증 플러그인이 GSS-API를 기반으로 하는 경우, 토큰이 GSS-API 컨텍스트 핸들(gss_ctx_id_t)로 설정됩니다.

tokentype

입력. 인증 플러그인이 제공하는 데이터의 유형을 나타냅니다. 사용되는 인증 플러그인이 GSS-API를 기반으로 하는 경우, 토큰이 GSS-API 컨텍스트 핸들(gss_ctx_id_t)로 설정됩니다. 사용되는 인증 플러그인이 사용자 ID/암호를 기반으로 하는 경우, 일반 유형이 됩니다. db2secPlugin.h에 정의된 tokentype 매개변수에 유효한 값은 다음과 같습니다.

- DB2SEC_GENERIC: 토큰이 사용자 ID/암호 기반 플러그인에서 제공됨을 나타냅니다.
- DB2SEC_GSSAPI_CTX_HANDLE: 토큰이 GSS-API(Kerberos 포함) 기반 플러그인에서 제공됨을 나타냅니다.

location

입력. DB2가 이 API를 클라이언트 측 또는 서버 측에서 호출하는지를 나타냅니다. db2secPlugin.h에 정의된 location 매개변수에 유효한 값은 다음과 같습니다.

- DB2SEC_SERVER_SIDE: API가 데이터베이스 서버에 호출됩니다.

- DB2SEC_CLIENT_SIDE: API가 클라이언트에서 호출됩니다.

authpluginname

입력. 토큰의 데이터를 제공한 인증 플러그인의 이름. db2secGetGroupsForUser API이 이 정보를 사용하여 올바른 그룹 멤버쉽을 판별해야 합니다. 권한 ID가 인증되지 않으면(예: 권한 ID가 현재 연결된 사용자와 일치하지 않는 경우) DB2가 이 매개변수를 입력하지 않습니다.

authpluginnamelen

입력. authpluginname 매개변수 값의 길이(바이트).

grouplist

출력. 사용자가 속해 있는 그룹의 목록. 그룹 목록은 병합된 varchar가 포함된 플러그인에서 할당한 메모리 섹션에 포인터로 리턴되어야 합니다(varchar는 문자 배열로서 첫 번째 바이트가 뒤에 오는 바이트 수를 나타냄). 길이는 서명되지 않은 char(1바이트)이며 그룹 이름을 최대 255자로 제한합니다(예: ″₩006GROUP1₩007MYGROUP₩008MYGROUP3″). 각 그룹 이름에는 유효한 DB2 권한 ID가 있어야 합니다. 이 배열에 사용되는 메모리는 플러그인이 할당해야 합니다. 따라서 플러그인은 API(예: DB2가 메모리를 제거하기 위해 호출하는 db2secFreeGroupListMemory API)를 제공해야 합니다.

numgroups

출력. grouplist 매개변수에 포함된 그룹의 수.

errormsg

출력. db2secGetGroupsForUser API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

사용 시 참고사항

다음은 이 API가 불완전한 그룹 목록을 DB2에 전달할 때 문제점이 발생하는 시나리오 목록입니다.

- 대체 권한 부여가 CREATE SCHEMA문에 제공됨. CREATE SCHEMA문에 중첩된 CREATE문이 있으면 AUTHORIZATION NAME 매개변수에 대해 그룹 찾아보기가 수행됩니다.
- MPP 환경에서 jar 파일 처리. MPP 환경에서, 세션 권한 ID가 있는 코디네이터 노드에서 jar 처리 요청이 전달됩니다. 카탈로그 노드는 이 요청을 받아 세션 권한 ID의 특권을 기반으로 jar 파일을 처리합니다(jar 처리 요청을 실행하는 사용자).

- jar 파일 설치. 세션 권한 ID에는 DBADM 또는 CREATEIN 권한(jar 스키마에서 내재적 또는 명시적으로 부여) 중 하나가 있어야 합니다. 위의 권한이 세션 권한 ID가 있는 그룹에 부여되지만 세션 권한 ID에 명시적으로 부여되지 않으면 조작에 실패합니다.
- jar 파일 제거. 세션 권한 ID에는 DBADM 또는 DROPIN 권한(jar 스키마에서 내재적 또는 명시적으로 부여) 중 하나가 있거나 jar 파일의 정의자여야 합니다. 위의 권한이 세션 권한 ID가 있는 그룹에 부여되지만 세션 권한 ID에 명시적으로 부여되지 않고, 세션 권한 ID가 jar 파일의 정의자가 아니면 조작에 실패합니다.
- jar 파일 교체. jar 파일 제거와 동일하며 교체 후 jar 파일을 설치합니다. 위의 두 사항이 모두 적용됩니다.

• SET SESSION_USER문이 발행될 때. 이 명령문에 의해 지정된 권한 ID의 컨텍스트에 따라 후속 DB2 조작이 실행됩니다. SESSION_USER의 그룹 중 하나가 소유한 필수 특권이 SESSION_USER 권한 ID에 명시적으로 부여되지 않으면 이러한 조작에 실패합니다.

db2secGroupPluginInit API - 그룹 플러그인 초기화

플러그인 로딩 직후 DB2 관리 프로그램이 호출하는 그룹 검색 플러그인용 초기화 API.

API 및 데이터 구조 구문

```
SQL_API_RC SQL_API_FN db2secGroupPluginInit
                              ( db2int32 version,
                                void *group_fns,
                                db2secLogMessage *logMessage_fn,
                                char     **errormsg,
                                db2int32 *errormsglen );
```

db2secGroupPluginInit API 매개변수

version

입력. 플러그인을 로딩하는 인스턴스에서 지원되는 가장 높은 버전의 API. db2secPlugin.h에 정의된 DB2SEC_API_VERSION 값에는 DB2 데이터베이스 관리 프로그램이 현재 지원하는 최신 버전의 API가 있습니다.

group_fns

출력. db2secGroupFunctions_<version_number> (group_functions_<version_number>이라고도 함) 구조의 포인터. db2secGroupFunctions_<version_number> 구조에는 그룹 검색 플러그인에 구현된 API의 포인터가 있습니다. 향후에는 다른 버전의 API(예: db2secGroupFunctions_<version_number>)가 포함될 수 있으므로, group_fns 매개변수가 플러그인이 구현한 버전에 해당하는 db2secGroupFunctions_<version_number> 구조의 포인터로 캐스트됩니다. group_functions_<version_number> 구조의 첫 번째 매개변수는 플러그인이 구

현한 API 버전을 DB2에 알려줍니다. 참고: 캐스팅은 DB2 버전이 플러그인이 구현한 API 버전과 같거나 높을 경우에만 수행됩니다. 버전 번호는 플러그인이 구현한 API의 버전을 나타내며 pluginType이 DB2SEC_PLUGIN_TYPE_GROUP으로 설정되어야 합니다.

logMessage_fn

입력. DB2 데이터베이스 시스템이 구현하는 db2secLogMessage API의 포인터. db2secGroupPluginInit API는 db2secLogMessage API를 호출하여 디버깅 또는 정보용으로 db2diag 로그 파일에 메시지를 로그할 수 있습니다. db2secLogMessage API의 첫 번째 매개변수(level)는 db2diag 로그 파일에 기록되는 진단 오류의 유형을 지정하고, 마지막 두 매개변수는 각각 메시지 문자열과 길이입니다. db2secPlugin.h에 정의된 dbesecLogMessage API의 첫 번째 매개변수의 올바른 값은 다음과 같습니다.

- DB2SEC_LOG_NONE: (0) 로깅 없음
- DB2SEC_LOG_CRITICAL: (1) 서버 오류 발견
- DB2SEC_LOG_ERROR: (2) 오류 발견
- DB2SEC_LOG_WARNING: (3) 경고
- DB2SEC_LOG_INFO: (4) 정보용

db2secLogMessage API의 'level' 매개변수 값이 diaglevel 데이터베이스 관리 프로그램 구성 매개변수 이하일 경우에만 diag.log 로그 파일에 메시지 텍스트가 표시됩니다. 예를 들어, DB2SEC_LOG_INFO 값을 사용하는 경우 diaglevel 데이터베이스 관리 프로그램 구성 매개변수가 4로 설정된 경우에만 db2diag 로그 파일에 메시지 텍스트가 표시됩니다.

errormsg

출력. db2secGroupPluginInit API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secPluginTerm - 그룹 플러그인 자원 정리

그룹 검색 플러그인에 사용된 자원을 제거합니다.

이 API는 DB2 데이터베이스 관리 프로그램이 그룹 검색 플러그인을 언로드하기 직전에 호출합니다. 이 API는 플러그인 라이브러리가 보유하는 자원을 적절하게 정리하는 방법으로 구현해야 합니다. 예를 들어, 플러그인에서 할당한 메모리를 제거하고, 열려 있는 파일을 닫고, 네트워크 연결을 닫습니다. 플러그인은 자원 제거를 위해 자원을 추적해야 합니다. 이 API는 Windows 플랫폼에서는 호출되지 않습니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secPluginTerm)
                      ( char      **errormsg,
                        db2int32 *errormsglen );
```

db2secPluginTerm API 매개변수

errormsg

출력. db2secPluginTerm API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

사용자 ID/암호 인증 플러그인용 API

사용자 ID/암호 인증 플러그인 모듈에서는 다음과 같은 클라이언트 측 API를 구현해야 합니다.

- db2secClientAuthPluginInit

  주: db2secClientAuthPluginInit API는 다음과 같은 프로토타입을 사용하여 API에 포인터 *logMessage_fn을 입력으로 사용합니다.

  ```
  SQL_API_RC (SQL_API_FN db2secLogMessage)
  (
  db2int32 level,
  void    *data,
  db2int32 length
  );
  ```

  플러그인은 db2secLogMessage API를 사용하여 디버깅 또는 정보용으로 db2diag 로그 파일에 메시지를 로그할 수 있습니다. 이 API는 DB2 데이터베이스 시스템이 제공하므로 구현할 필요는 없습니다.
- db2secClientAuthPluginTerm
- db2secGenerateInitialCred(gssapi에만 사용됨)
- db2secRemapUserid(선택적)
- db2secGetDefaultLoginContext
- db2secValidatePassword
- db2secProcessServerPrincipalName(GSS-API에만 사용됨)
- db2secFreeToken(DLL에서 보유한 메모리를 제거하는 함수)
- db2secFreeErrormsg
- db2secFreeInitInfo

- 외부적으로 해결해야 하는 유일한 API는 db2secClientAuthPluginInit입니다. 이 API는 void * 매개변수를 사용하므로 다음 중 하나로 캐스트해야 합니다.

```
typedef struct db2secUseridPasswordClientAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

SQL_API_RC (SQL_API_FN * db2secGetDefaultLoginContext)
(
char        authid[DB2SEC_MAX_AUTHID_LENGTH],
db2int32   *authidlen,
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
db2int32    useridtype,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
void      **token,
char      **errormsg,
db2int32   *errormsglen
);
/* Optional */
SQL_API_RC (SQL_API_FN * db2secRemapUserid)
(
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
char        password[DB2SEC_MAX_PASSWORD_LENGTH],
db2int32   *passwordlen,
char        newpassword[DB2SEC_MAX_PASSWORD_LENGTH],
db2int32   *newpasswordlen,
const char *dbname,
db2int32    dbnamelen,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secValidatePassword)
(
const char *userid,
db2int32    useridlen,
const char *usernamespace,
db2int32    usernamespacelen,
db2int32    usernamespacetype,
const char *password,
db2int32    passwordlen,
const char *newpassword,
db2int32    newpasswordlen,
const char *dbname,
db2int32    dbnamelen,
db2Uint32   connection_details,
void      **token,
char      **errormsg,
```

```
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeToken)
(
void     **token,
char     **errormsg,
db2int32  *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *errormsg
);

SQL_API_RC (SQL_API_FN * db2secClientAuthPluginTerm)
(
char     **errormsg,
db2int32  *errormsglen
);
}
```

또는

```
typedef struct db2secGssapiClientAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

SQL_API_RC (SQL_API_FN * db2secGetDefaultLoginContext)
(
char        authid[DB2SEC_MAX_AUTHID_LENGTH],
db2int32   *authidlen,
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
db2int32    useridtype,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
void      **token,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secProcessServerPrincipalName)
(
const void *data,
gss_name_t *gssName,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secGenerateInitialCred)
(
const char    *userid,
db2int32       useridlen,
```

```
const char    *usernamespace,
db2int32       usernamespacelen,
db2int32       usernamespacetype,
const char    *password,
db2int32       passwordlen,
const char    *newpassword,
db2int32       newpasswordlen,
const char    *dbname,
db2int32       dbnamelen,
gss_cred_id_t *pGSSCredHandle,
void         **initInfo,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeToken)
(
void      *token,
char     **errormsg,
db2int32  *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *errormsg
);


SQL_API_RC (SQL_API_FN * db2secFreeInitInfo)
(
void      *initInfo,
char     **errormsg,
db2int32  *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secClientAuthPluginTerm)
(
char     **errormsg,
db2int32  *errormsglen
);

/* GSS-API specific functions -- refer to db2secPlugin.h
   for parameter list*/

  OM_uint32 (SQL_API_FN * gss_init_sec_context )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_delete_sec_context )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_display_status )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_release_buffer )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_release_cred )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_release_name )(<parameter list>);
  }
```

사용자 ID/암호 플러그인을 작성하는 경우에는 db2secUseridPasswordClientAuthFunctions_1 구조를 사용해야 합니다. GSS-API(Kerberos 포함) 플러그인을 작성하는 경우에는 db2secGssapiClientAuthFunctions_1 구조를 사용해야 합니다.

사용자 ID/암호 플러그인 라이브러리의 경우, 다음과 같은 서버 측 API를 구현해야 합니다.

- db2secServerAuthPluginInit

  db2secServerAuthPluginInit API는 다음과 같은 프로토타입을 사용하여 db2secLogMessage API에 포인터 *logMessage_fn을, db2secGetConDetails API에 포인터 *getConDetails_fn을 입력으로 사용합니다.

```
SQL_API_RC (SQL_API_FN db2secLogMessage)
(
db2int32 level,
void    *data,
db2int32 length
);

SQL_API_RC (SQL_API_FN db2secGetConDetails)
(
db2int32    conDetailsVersion,
const void *pConDetails
);
```

  플러그인은 db2secLogMessage API를 사용하여 디버깅 또는 정보용으로 db2diag 로그 파일에 메시지를 로그할 수 있습니다. 플러그인은 db2secGetConDetails API를 사용하여 데이터베이스 연결을 시도하는 클라이언트에 대한 세부사항을 가져올 수 있습니다. db2secLogMessage API 및 db2secGetConDetails API 둘 다 DB2 데이터베이스 시스템이 제공하므로, 이러한 API를 구현할 필요는 없습니다. 그런 다음 db2secGetConDetails API는 두 번째 매개변수로 다음과 같은 구조 중 하나의 포인터인 pConDetails를 사용합니다.

  db2sec_con_details_1:

```
typedef struct db2sec_con_details_1
{
  db2int32  clientProtocol;
  db2Uint32 clientIPAddress;
  db2Uint32 connect_info_bitmap;
  db2int32  dbnameLen;
  char      dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
} db2sec_con_details_1;
```

  db2sec_con_details_2:

```
typedef struct db2sec_con_details_2
{
  db2int32  clientProtocol;     /* See SQL_PROTOCOL_ in sqlenv.h */
  db2Uint32 clientIPAddress;    /* Set if protocol is TCPIP4     */
  db2Uint32 connect_info_bitmap;
  db2int32  dbnameLen;
  char dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
  db2Uint32 clientIP6Address[4];/* Set if protocol is TCPIP6     */
} db2sec_con_details_2;
```

  db2sec_con_details_3:

```
typedef struct db2sec_con_details_3
{
  db2int32 clientProtocol; /* See SQL_PROTOCOL_ in sqlenv.h */
  db2Uint32 clientIPAddress; /* Set if protocol is TCPIP4 */
  db2Uint32 connect_info_bitmap;
  db2int32 dbnameLen;
  char dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
  db2Uint32 clientIP6Address[4];/* Set if protocol is TCPIP6 */
  db2Uint32 clientPlatform; /* SQLM_PLATFORM_* from sqlmon.h */
  db2Uint32 _reserved[16];
} db2sec_con_details_3;
```

conDetailsVersion에 사용 가능한 값은 DB2SEC_CON_DETAILS_VERSION_1, DB2SEC_CON_DETAILS_VERSION_2 및 DB2SEC_CON_DETAILS_VERSION_3로서 API의 버전을 나타냅니다.

주: db2sec_con_details_1, db2sec_con_details_2 또는 db2sec_con_details_3를 사용할 때 다음 사항을 고려하십시오.

- db2sec_con_details_1 구조와 DB2SEC_CON_DETAILS_VERSION_1 값을 사용하는 기존 플러그인은 버전 8.2에서 작동되었으므로 db2GetConDetails API를 호출할 때 계속 작동됩니다. 이 API가 IPv4 플랫폼에서 호출되면, 클라이언트 IP 주소가 해당 구조의 clientIPAddress 필드에 리턴됩니다. 이 API가 IPv6 플랫폼에서 호출되면, clientIPAddress 필드에 0 값이 리턴됩니다. IPv6 플랫폼에서 클라이언트 IP 주소를 검색하려면, db2sec_con_details_2 구조와 DB2SEC_CON_DETAILS_VERSION_2 값 또는 db2sec_con_details_3 구조와 DB2SEC_CON_DETAILS_VERSION_3 값을 사용하도록 보안 플러그인 코드를 변경해야 합니다 .
- 새 플러그인은 db2sec_con_details_3 구조와 DB2SEC_CON_DETAILS_VERSION_3 값을 사용해야 합니다. db2secGetConDetails API가 IPv4 플랫폼에서 호출되면 클라이언트 IP 주소가 db2sec_con_details_3 구조의 clientIPAddress 필드에 리턴되고, API가 IPv6 플랫폼에서 호출되면 클라이언트 IP 주소가 db2sec_con_details_3 구조의 clientIP6Address 필드에 리턴됩니다. 연결 세부사항 구조의 *clientProtocol* 필드가 SQL_PROTOCOL_TCPIP(IPv4, v1의 구조 사용), SQL_PROTOCOL_TCPIP4(IPv4, v2의 구조 사용) 또는 SQL_PROTOCOL_TCPIP6(IPv6, v2 또는 v3의 구조 사용) 중 하나로 설정됩니다.
- db2sec_con_details_3 구조는 sqlmon.h에 정의된 플랫폼 유형 상수(예: SQLM_PLATFORM_AIX)를 사용하여 클라이언트 플랫폼 유형(통신 계층이 보고함)을 식별하는 추가 필드(*clientPlatform*)가 포함된 것을 제외하고 db2sec_con_details_2 구조와 동일합니다.

- db2secServerAuthPluginTerm
- db2secValidatePassword

- db2secGetAuthIDs
- db2secDoesAuthIDExist
- db2secFreeToken
- db2secFreeErrormsg
- 외부적으로 해결해야 하는 유일한 API는 db2secServerAuthPluginInit입니다. 이 API는 void * 매개변수를 사용하므로 다음 중 하나로 캐스트해야 합니다.

```
typedef struct db2secUseridPasswordServerAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

  /* parameter lists left blank for readability
    see above for parameters */
SQL_API_RC (SQL_API_FN * db2secValidatePassword)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secGetAuthIDs)(<parameter list);
SQL_API_RC (SQL_API_FN * db2secDoesAuthIDExist)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secFreeToken)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secServerAuthPluginTerm)();
} userid_password_server_auth_functions;
```

또는

```
typedef struct db2secGssapiServerAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;
gss_buffer_desc serverPrincipalName;
gss_cred_id_t ServerCredHandle;
SQL_API_RC (SQL_API_FN * db2secGetAuthIDs)(<parameter list);
SQL_API_RC (SQL_API_FN * db2secDoesAuthIDExist)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secServerAuthPluginTerm)();

/* GSS-API specific functions
refer to db2secPlugin.h for parameter list*/
OM_uint32 (SQL_API_FN * gss_accept_sec_context )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_display_name )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_delete_sec_context )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_display_status )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_release_buffer )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_release_cred )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_release_name )(<parameter list>);

} gssapi_server_auth_functions;
```

사용자 ID/암호 플러그인을 작성하는 경우에는 db2secUseridPasswordServerAuthFunctions_1 구조를 사용해야 합니다. GSS-API(Kerberos 포함) 플러그인을 작성하는 경우에는 db2secGssapiServerAuthFunctions_1 구조를 사용해야 합니다.

db2secClientAuthPluginInit API - 클라이언트 인증 플러그인 초기화

플러그인 로딩 직후 DB2 데이터베이스 관리 프로그램이 호출하는 클라이언트 인증 플러그인용 초기화 API.

API 및 데이터 구조 구문

```
SQL_API_RC SQL_API_FN db2secClientAuthPluginInit
                              ( db2int32 version,
                                void *client_fns,
                                db2secLogMessage *logMessage_fn,
                                char    **errormsg,
                                db2int32 *errormsglen );
```

db2secClientAuthPluginInit API 매개변수

version

입력. 현재 DB2 데이터베이스 관리 프로그램을 지원하는 가장 높은 버전의 API. db2secPlugin.h에 정의된 DB2SEC_API_VERSION 값에는 현재 DB2를 지원하는 최신 버전의 API가 있습니다.

client_fns

출력. GSS-API 인증이 사용되는 경우 db2secGssapiClientAuthFunctions_<version_number> 구조(gssapi_client_auth_functions_<version_number>라고도 함), 사용자 ID/암호 인증이 사용되는 경우 db2secUseridPasswordClientAuthFunctions_<version_number> 구조(userid_password_client_auth_functions_<version_number>라고도 함)용으로 DB2 데이터베이스 관리 프로그램에서 제공하는 메모리의 포인터. db2secGssapiClientAuthFunctions_<version_number> 구조 및 db2secUseridPasswordClientAuthFunctions_<version_number> 구조에는 각각 GSS-API 인증 플러그인 및 사용자 ID/암호 인증 플러그인에 구현된 API의 포인터가 있습니다. 향후 DB2 버전에는 다른 버전의 API가 포함되므로, client_fns 매개변수는 플러그인이 구현한 버전에 해당하는 gssapi_client_auth_functions_<version_number> 구조의 포인터로 캐스트됩니다.

gssapi_client_auth_functions_<version_number> 구조 또는 userid_password_client_auth_functions_<version_number> 구조의 첫 번째 매개변수는 플러그인이 구현한 API 버전을 DB2 데이터베이스 관리 프로그램에 알려줍니다.

주: 캐스팅은 DB2 버전이 플러그인이 구현한 API 버전과 같거나 높을 경우에만 수행됩니다.

gssapi_server_auth_functions_<version_number> 또는 userid_password_server_auth_functions_<version_number> 구조 내에서

plugintype 매개변수는 DB2SEC_PLUGIN_TYPE_USERID_PASSWORD, DB2SEC_PLUGIN_TYPE_GSSAPI 또는 DB2SEC_PLUGIN_TYPE_KERBEROS로 설정되어야 합니다. 향후 버전의 API에서는 다른 값이 정의될 수 있습니다.

logMessage_fn

입력. DB2 데이터베이스 관리 프로그램이 구현하는 db2secLogMessage API의 포인터. db2secClientAuthPluginInit API는 db2secLogMessage API를 호출하여 디버깅 또는 정보용으로 db2diag 로그 파일에 메시지를 로그할 수 있습니다. db2secLogMessage API의 첫 번째 매개변수(level)는 db2diag 로그 파일에 기록되는 진단 오류의 유형을 지정하고, 마지막 두 매개변수는 각각 메시지 문자열과 길이입니다. db2secPlugin.h에 정의된 dbesecLogMessage API의 첫 번째 매개변수의 올바른 값은 다음과 같습니다.

- DB2SEC_LOG_NONE (0) 로깅 없음
- DB2SEC_LOG_CRITICAL (1) 서버 오류 발견
- DB2SEC_LOG_ERROR (2) 오류 발견
- DB2SEC_LOG_WARNING (3) 경고
- DB2SEC_LOG_INFO (4) 정보용

db2secLogMessage API의 'level' 매개변수 값이 diaglevel 데이터베이스 관리 프로그램 구성 매개변수 이하일 경우에만 db2diag 로그 파일에 메시지 텍스트가 표시됩니다. 예를 들어, DB2SEC_LOG_INFO 값을 사용하는 경우 diaglevel 데이터베이스 관리 프로그램 구성 매개변수가 4로 설정된 경우에만 db2diag 로그 파일에 메시지 텍스트가 표시됩니다.

errormsg

출력. db2secClientAuthPluginInit API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secClientAuthPluginTerm API - 클라이언트 인증 플러그인 자원 정리

클라이언트 인증 플러그인에 사용된 자원을 제거합니다.

이 API는 DB2 데이터베이스 관리 프로그램이 클라이언트 인증 플러그인을 언로드하기 직전에 호출합니다. 이 API는 플러그인 라이브러리가 보유하는 자원을 적절하게 정리하는 방법으로 구현해야 합니다. 예를 들어, 플러그인에서 할당한 메모리를 제거하고, 열려 있는 파일을 닫고, 네트워크 연결을 닫습니다. 플러그인은 자원 제거를 위해 자원을 추적해야 합니다. 이 API는 Windows 플랫폼에서는 호출되지 않습니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secClientAuthPluginTerm)
                      ( char     **errormsg,
                        db2int32 *errormsglen);
```

db2secClientAuthPluginTerm API 매개변수

errormsg

출력. db2secClientAuthPluginTerm API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secDoesAuthIDExist - 인증 ID가 존재하는지 여부 확인

권한 ID가 개별 사용자를 나타내는지 여부를 판별합니다(예: API가 권한 ID를 외부 사용자 ID에 맵핑할 수 있는지 여부).

API는 권한 ID가 유효한 경우(성공) DB2SEC_PLUGIN_OK, 유효하지 않은 경우 DB2SEC_PLUGIN_INVALID_USERORGROUP, 권한 ID 존재를 판별할 수 없는 경우 DB2SEC_PLUGIN_USERSTATUSNOTKNOWN을 리턴합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secDoesAuthIDExist)
                      ( const char *authid,
                        db2int32 authidlen,
                        char       **errormsg,
                        db2int32 *errormsglen );
```

db2secDoesAuthIDExist API 매개변수

authid

입력. 유효성을 확인할 권한 ID. 대문자이며 뒤 공백이 없습니다.

authidlen

입력. authid 매개변수 값의 길이(바이트).

errormsg

출력. db2secDoesAuthIDExist API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이를 나타내는 정수의 포인터.

db2secFreeInitInfo API - db2secGenerateInitialCred에서 보유한 자원 정리

db2secGenerateInitialCred API에서 할당한 자원을 제거합니다. 예를 들어, 여기에는 GSS-API 증명서 캐시용으로 작성된 증명서 캐시 또는 기본 메커니즘 컨텍스트의 핸들이 포함됩니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secFreeInitInfo)
                        ( void *initinfo,
                          char     **errormsg,
                          db2int32 *errormsglen);
```

db2secFreeInitInfo API 매개변수

initinfo

입력. DB2 데이터베이스 관리 프로그램에 알려지지 않은 데이터의 포인터. 플러그인은 이 메모리를 사용하여 증명서 핸들 작성 프로세스에 할당되는 자원 목록을 유지할 수 있습니다. 이러한 자원은 이 API를 호출하면 제거됩니다.

errormsg

출력. db2secFreeInitInfo API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secFreeToken API - 토큰에서 보유한 메모리 제거

토큰에서 보유한 메모리를 제거합니다. 이 API는 DB2 데이터베이스 관리 프로그램이 토큰 매개변수가 보유한 메모리가 더 이상 필요하지 않을 때 호출합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secFreeToken)
                        ( void *token,
                          char       **errormsg,
                          db2int32 *errormsglen );
```

db2secFreeToken API 매개변수

token 입력. 사용 가능해지는 메모리의 포인터.

errormsg

출력. db2secFreeToken API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secGenerateInitialCred API - 초기 증명서 생성

db2secGenerateInitialCred API는 GSS-API 전달되는 사용자 ID 및 암호를 기준으로 초기 증명서를 가져옵니다.

Kerberos의 경우, TGT(Ticket-Granting Ticket)입니다. pGSSCredHandle 매개변수에 리턴되는 증명서 핸들은 gss_init_sec_context API에 사용되는 핸들로서 INITIATE 또는 BOTH 증명서 중 하나여야 합니다. db2secGenerateInitialCred API는 사용자 ID 및 암호가 제공될 때만 호출됩니다. 그렇지 않으면, DB2 데이터베이스 관리 프로그램은 gss_init_sec_context API를 호출할 때 GSS_C_NO_CREDENTIAL 값을 지정하여 현재 로그인 컨텍스트에서 가져온 디폴트 증명서가 사용됨을 나타냅니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secGenerateInitialCred)
                      ( const char *userid,
                        db2int32 useridlen,
                        const char *usernamespace,
                        db2int32 usernamespacelen,
                        db2int32 usernamespacetype,
                        const char *password,
                        db2int32 passwordlen,
                        const char *newpassword,
                        db2int32 newpasswordlen,
                        const char *dbname,
                        db2int32 dbnamelen,
                        gss_cred_id_t *pGSSCredHandle,
                        void          **InitInfo,
                        char          **errormsg,
                        db2int32 *errormsglen );
```

db2secGenerateInitialCred API 매개변수

userid 입력. 데이터베이스 서버에서 암호의 유효성을 확인할 사용자 ID.

useridlen

입력. userid 매개변수 값의 길이(바이트).

usernamespace

입력. 사용자 ID를 가져온 이름 스페이스.

usernamespacelen

입력. usernamespace 매개변수 값의 길이(바이트).

usernamespacetype

입력. 이름 스페이스 유형.

password

입력. 유효성을 확인할 암호.

passwordlen

입력. password 매개변수 값의 길이(바이트).

newpassword

입력. 새 암호(암호가 변경되는 경우). 암호 변경 요청이 없는 경우 newpassword 매개변수는 NULL로 설정됩니다. 이 매개변수가 NULL로 설정되지 않으면, 암호를 새 값으로 설정하기 전에 API가 이전 암호의 유효성을 확인합니다. API가 암호 변경 요청을 이행할 필요는 없지만, 이행하지 않을 경우 이전 암호의 유효성을 확인하지 않고 리턴 값 DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED를 즉시 리턴해야 합니다.

newpasswordlen

입력. newpassword 매개변수 값의 길이(바이트).

dbname

입력. 연결할 데이터베이스의 이름. API가 이 매개변수를 무시하거나, 유효한 암호를 가진 사용자의 특정 데이터베이스에 대한 액세스를 제한하는 규정이 있는 경우 리턴 값 DB2SEC_PLUGIN_CONNECTION_DISALLOWED를 리턴할 수 있습니다.

dbnamelen

입력. dbname 매개변수 값의 길이(바이트).

pGSSCredHandle

출력. GSS-API 증명서 핸들의 포인터.

InitInfo

출력. DB2에 알려지지 않은 데이터의 포인터. 플러그인은 이 메모리를 사용하여 증명서 핸들 작성 프로세스에 할당되는 자원 목록을 유지할 수 있습니다. DB2 데이터베이스 관리 프로그램은 이러한 자원을 사용할 수 있는 인증 프로세스 마지막 부분에서 db2secFreeInitInfo API를 호출합니다. db2secGenerateInitialCred API가 이 목록을 유지할 필요가 없으면 NULL을 리턴합니다.

errormsg

출력. db2secGenerateInitialCred API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

주: 이 API의 경우, 리턴 값이 잘못된 사용자 ID 또는 암호를 나타내는 경우에는 오류 메시지가 작성되지 않습니다. API가 올바르게 완료되지 못하도록 방해하는 내부 오류가 있을 경우에만 오류 메시지가 리턴됩니다.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secGetAuthIDs API - 인증 ID 가져오기

인증된 사용자에게 SQL 권한 ID를 리턴합니다. 이 API는 데이터베이스에 연결되어 있는 동안 사용자 ID/암호 및 GSS-API 인증 방법에 호출됩니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secGetAuthIDs)
                         ( const char *userid,
                           db2int32 useridlen,
                           const char *usernamespace,
                           db2int32 usernamespacelen,
                           db2int32 usernamespacetype,
                           const char *dbname,
                           db2int32 dbnamelen,
                           void      **token,
                           char SystemAuthID[DB2SEC_MAX_AUTHID_LENGTH],
                           db2int32 *SystemAuthIDlen,
                           char InitialSessionAuthID[DB2SEC_MAX_AUTHID_LENGTH],
                           db2int32 *InitialSessionAuthIDlen,
                           char username[DB2SEC_MAX_USERID_LENGTH],
                           db2int32 *usernamelen,
                           db2int32 *initsessionidtype,
                           char      **errormsg,
                           db2int32 *errormsglen );
```

db2secGetAuthIDs API 매개변수

userid 입력. 인증된 사용자. 인증 없이 사용자 조작 전환이 허용되도록 트러스트된 컨텍스트가 정의되지 않은 경우에는 일반적으로 GSS-API 인증에는 사용되지 않습니다. 이 경우, 사용자 전환 요청에 제공된 사용자 이름이 이 매개변수에 전달됩니다.

useridlen

입력. userid 매개변수 값의 길이(바이트).

usernamespace

입력. 사용자 ID를 가져온 이름 스페이스.

usernamespacelen

입력. usernamespace 매개변수 값의 길이(바이트).

usernamespacetype

입력. Namespacetype 값. 현재 지원되는 유일한 이름 스페이스 유형 값은 DB2SEC_NAMESPACE_SAM_COMPATIBLE입니다 (사용자 이름 스타일에 해당, 예: domain₩myname).

dbname

입력. 연결할 데이터베이스의 이름. API는 이 값을 무시하거나, 동일한 사용자가 다른 데이터베이스에 연결할 때 다른 권한 ID를 리턴할 수 있습니다. 이 매개변수는 NULL이 될 수 있습니다.

dbnamelen

입력. dbname 매개변수 값의 길이(바이트). dbname 매개변수가 NULL이면 이 매개변수가 0으로 설정됩니다.

token 입력 또는 출력. 플러그인이 db2secGetGroupsForUser API에 전달하는 데이터. GSS-API의 경우, 컨텍스트 핸들(gss_ctx_id_t)입니다. 일반적으로 token은 입력 전용 매개변수이며 이 값은 db2secValidatePassword API에서 비롯됩니다. 또한 클라이언트에서 인증이 수행되면 출력 매개변수도 될 수 있으며 이 경우 db2secValidatePassword API가 호출되지 않습니다. 트러스트된 컨텍스트가 정의된 환경(인증 없이 사용자 전환 조작이 허용됨)에서, db2secGetAuthIDs API는 이 token 매개변수의 NULL 값을 수신할 수 있어야 하며, 위의 userid 및 useridlen 매개변수를 기반으로 시스템 권한 부여 ID를 파생할 수 있어야 합니다.

SystemAuthID

출력. 인증된 사용자의 ID에 해당하는 시스템 권한 부여 ID. 크기는 255바이트이지만 DB2 데이터베이스 관리 프로그램은 현재 최대 30바이트까지 지원합니다.

SystemAuthIDlen

출력. SystemAuthID 매개변수 값의 길이(바이트).

InitialSessionAuthID

출력. 이 연결 세션에 사용된 권한 ID. 이 매개변수는 일반적으로 SystemAuthID 매개변수와 동일하지만, 다른 경우도 있습니다(예: SET SESSION AUTHORIZATION문 발행). 크기는 255바이트이지만 DB2 데이터베이스 관리 프로그램은 현재 최대 30바이트까지 지원합니다.

InitialSessionAuthIDlen

출력. InitialSessionAuthID 매개변수 값의 길이(바이트).

username

출력. 인증된 사용자 및 권한 ID에 해당하는 사용자 이름. 이 매개변수는 감사에만 사용되며 CONNECT문의 감사 레코드에 있는 ″사용자 ID″ 필드에 로그됩니다. 이 API가 username 매개변수에 채워지지 않으면 DB2 데이터베이스 관리 프로그램이 userid에서 이 매개변수를 복사합니다.

usernamelen

출력. username 매개변수 값의 길이(바이트).

initsessionidtype

출력. InitialSessionAuthid 매개변수가 역할 또는 권한 ID인지 여부를 나타내는 세션 권한 ID 유형. API는 db2secPlugin.h에 정의된 다음 값 중 하나를 리턴합니다.

- DB2SEC_ID_TYPE_AUTHID (0)
- DB2SEC_ID_TYPE_ROLE (1)

errormsg

출력. db2secGetAuthIDs API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secGetDefaultLoginContext API - 디폴트 로그인 컨텍스트 가져오기

디폴트 로그인 컨텍스트와 연관된 사용자를 판별합니다. 즉, 사용자 ID를 명시적으로 지정하지 않고(데이터베이스에 대한 내재적 인증 또는 로컬 인증) DB2 명령을 호출하는 사용자의 DB2 권한 ID를 판별합니다. 이 API는 권한 ID와 사용자 ID 둘 다 리턴해야 합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secGetDefaultLoginContext)
                      ( char authid[DB2SEC_MAX_AUTHID_LENGTH],
                        db2int32 *authidlen,
                        char userid[DB2SEC_MAX_USERID_LENGTH],
                        db2int32 *useridlen,
                        db2int32 useridtype,
                        char usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
                        db2int32 *usernamespacelen,
                        db2int32 *usernamespacetype,
                        const char *dbname,
                        db2int32 dbnamelen,
                        void      **token,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

db2secGetDefaultLoginContext API 매개변수

authid

출력. 권한 ID를 리턴하는 매개변수. 리턴된 값은 DB2 권한 ID 이름 지정 규칙을 따라야 합니다. 그렇지 않으면 사용자는 요청된 조치를 수행할 권한을 부여 받지 못합니다.

authidlen

출력. authid 매개변수 값의 길이(바이트).

userid 출력. 디폴트 로그인 컨텍스트와 연관된 사용자 ID가 리턴되는 매개변수.

useridlen

출력. userid 매개변수 값의 길이(바이트).

useridtype

입력. 프로세스의 실 사용자 ID 또는 유효 사용자 ID가 지정되는지를 나타냅니다. Windows에서는 실 사용자 ID만 존재합니다. UNIX 및 Linux에서, 응용프로그램의 `uid` 사용자 ID가 프로세스를 실행하는 사용자의 ID와 다를 경우 실 사용자 ID와 유효 ID가 다를 수 있습니다. db2secPlugin.h에 정의된 userid 매개변수에 유효한 값은 다음과 같습니다.

DB2SEC_PLUGIN_REAL_USER_NAME

실 사용자 ID가 지정됨을 나타냅니다.

DB2SEC_PLUGIN_EFFECTIVE_USER_NAME

유효 사용자 ID가 지정됨을 나타냅니다.

주: 일부 플러그인 구현에서는 실 사용자 ID와 유효 사용자 ID를 구분하지 않을 수 있습니다. 특히, 사용자의 UNIX 또는 Linux ID를 사용하여 DB2 권한 부여 ID를 설정하지 않는 플러그인에서는 이 두 사용자 ID의 차이가 무시됩니다.

usernamespace

출력. 사용자 ID의 이름 스페이스.

usernamespacelen

출력. usernamespace 매개변수 값의 길이(바이트). usernamespacetype 매개변수를 db2secPlugin.h에 정의된
DB2SEC_NAMESPACE_SAM_COMPATIBLE 값으로 설정해야 한다는 제한사항에 따라 현재 지원되는 최대 길이는 15바이트입니다.

usernamespacetype

출력. Namespacetype 값. 현재 지원되는 유일한 이름 스페이스 유형은 DB2SEC_NAMESPACE_SAM_COMPATIBLE입니다(사용자 이름 스타일에 해당, 예: domain₩myname).

dbname

입력. 이 호출이 데이터베이스 연결의 컨텍스트에 사용되는 경우, 연결할 데이터베이스의 이름이 포함됩니다. 로컬 인증 조치 또는 인스턴스 첨부의 경우, 이 매개변수는 NULL로 설정됩니다.

dbnamelen

입력. dbname 매개변수 값의 길이(바이트).

token 출력. 플러그인이 할당하는 데이터의 포인터로서 플러그인에 있는 후속 인증 호출 또는 그룹 검색 플러그인에 전달할 수 있습니다. 이 데이터의 구조는 플러그인 기록기에 의해 판별됩니다.

errormsg

출력. db2secGetDefaultLoginContext API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secProcessServerPrincipalName API - 서버에서 리턴된 서비스 핵심부 이름 처리

db2secProcessServerPrincipalName API는 서버에서 리턴된 서비스 핵심부 이름을 처리하며 gss_init_sec_context API에 사용되는 gss_name_t 내부 형식으로 핵심부 이름을 리턴합니다.

또한 db2secProcessServerPrincipalName API는 Kerberos 인증이 사용될 때 데이터베이스 디렉토리에 카탈로그된 서비스 핵심부 이름을 처리합니다. 일반적으로 이 변환에서는 gss_import_name API를 사용합니다. 컨텍스트가 설정된 후 gss_name_t 오브젝트는 gss_release_name API가 호출되면 제거됩니다. gssName 매개변수가 유효한 GSS 이름을 나타내면 db2secProcessServerPrincipalName API는 DB2SEC_PLUGIN_OK 값을 리턴하고 핵심부 이름이 유효하지 않으면 DB2SEC_PLUGIN_BAD_PRINCIPAL_NAME 오류 코드를 리턴합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secProcessServerPrincipalName)
                      ( const char *name,
                        db2int32 namelen,
                        gss_name_t *gssName,
                        char       **errormsg,
                        db2int32 *errormsglen );
```

db2secProcessServerPrincipalName API 매개변수

name 입력. GSS_C_NT_USER_NAME 형식의 서비스 핵심부 이름(예: service/host@REALM).

namelen

입력. name 매개변수 값의 길이(바이트).

gssName

출력. GSS-API 내부 형식의 출력 서비스 핵심부 이름의 포인터.

errormsg

출력. db2secProcessServerPrincipalName API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secRemapUserid API - 사용자 ID 및 암호 다시 맵핑

이 API는 DB2 데이터베이스 관리 프로그램이 클라이언트 측에서 호출하여 지정된 사용자 ID 및 암호(새 암호 및 사용자 이름 스페이스 포함)를 연결 시에 지정된 값과 다른 값에 다시 맵핑합니다.

DB2 데이터베이스 관리 프로그램은 연결 시에 사용자 ID 및 암호가 제공된 경우에만 이 API를 호출합니다. 이렇게 하면 플러그인에서 사용자 ID가 자동으로 사용자 ID/암호 쌍에 다시 맵핑되지 않습니다. 이 API는 선택적이며 보안 플러그인에서 제공하거나 구현하지 않으면 호출되지 않습니다.

API 및 데이터 구조 구문

```
  SQL_API_RC ( SQL_API_FN *db2secRemapUserid)
( char userid[DB2SEC_MAX_USERID_LENGTH],
  db2int32 *useridlen,
  char usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
  db2int32 *usernamespacelen,
  db2int32 *usernamespacetype,
  char password[DB2SEC_MAX_PASSWORD_LENGTH],
  db2int32 *passwordlen,
  char newpasswd[DB2SEC_MAX_PASSWORD_LENGTH],
  db2int32 *newpasswdlen,
  const char *dbname,
  db2int32 dbnamelen,
  char      **errormsg,
  db2int32 *errormsglen);
```

db2secRemapUserid API 매개변수

userid 입력 또는 출력. 다시 맵핑되는 사용자 ID. 입력 사용자 ID 값이 없으면, API가 입력 사용자 ID 값과 동일하거나 다른 출력 사용자 ID 값을 제공해야 합니다. 입력 사용자 ID 값이 없으면, API가 출력 사용자 ID 값을 리턴하지 않습니다.

useridlen

입력 또는 출력. userid 매개변수 값의 길이(바이트).

usernamespace

입력 또는 출력. 사용자 ID의 이름 스페이스. 이 값은 선택적으로 다시 맵핑할 수 있습니다. 입력 매개변수 값이 지정되지 않았지만 출력 값이 리턴되는 경우, DB2 데이터베이스 관리 프로그램이 CLIENT 유형 인증에 usernamespace만 사용하며 다른 인증 유형에서는 무시됩니다.

usernamespacelen

입력 또는 출력. usernamespace 매개변수 값의 길이(바이트). usernamespacetype 매개변수를 db2secPlugin.h에 정의된

DB2SEC_NAMESPACE_SAM_COMPATIBLE 값으로 설정해야 한다는 제한사항에 따라 현재 지원되는 최대 길이는 15바이트입니다.

usernamespacetype

입력 또는 출력. 이전 및 새 namespacetype 값. 현재 지원되는 유일한 이름 스페이스 유형 값은 DB2SEC_NAMESPACE_SAM_COMPATIBLE입니다 (사용자 이름 스타일에 해당, 예: domain₩myname).

password

입력 또는 출력. 입력으로서는 다시 맵핑되는 암호이며, 출력으로서는 다시 맵핑된 암호입니다. 이 매개변수에 출력 값이 지정되면, API는 입력 값과 다른 출력 값을 리턴해야 합니다. 출력 값이 지정되지 않으면, API는 출력 암호 값을 리턴하지 않아야 합니다.

passwordlen

입력 또는 출력. password 매개변수 값의 길이(바이트).

newpasswd

입력 또는 출력. 입력으로서는 설정되는 새 암호이며, 출력으로서는 확인된 새 암호입니다.

주: DB2 데이터베이스 관리 프로그램이 클라이언트 또는 서버에서(인증 데이터베이스 관리 프로그램 구성 매개변수의 값에 따라) db2secValidatePassword API의 newpassword 매개변수에 전달하는 새 암호입니다. 새 암호가 입력으로 전달되면, API가 출력 값을 리턴해야 하며 다른 새 암호가 될 수 있습니다. 입력으로 전달된 새 암호가 없으면, API가 출력 새 암호를 리턴하지 않아야 합니다.

newpasswdlen

입력 또는 출력. newpasswd 매개변수 값의 길이(바이트).

dbname

입력. 클라이언트가 연결되는 데이터베이스의 이름.

dbnamelen

입력. dbname 매개변수 값의 길이(바이트).

errormsg

출력. db2secRemapUserid API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secServerAuthPluginInit - 서버 인증 플러그인 초기화

db2secServerAuthPluginInit API는 플러그인 로딩 직후 DB2 데이터베이스 관리 프로그램이 호출하는 서버 인증 플러그인용 초기화 API입니다.

GSS-API의 경우, 플러그인은 초기화할 때 gssapi_server_auth_functions 구조 내에서 serverPrincipalName 매개변수에 서버의 핵심부 이름을 입력하고, the gssapi_server_auth_functions 구조 내에서 serverCredHandle 매개변수에 서버의 증명서 핸들을 제공해야 합니다. 핵심부 이름과 증명서 핸들을 보유하기 위해 할당된 메모리 제거 작업은 gss_release_name 및 gss_release_cred API를 호출하여 db2secServerAuthPluginTerm API가 수행해야 합니다.

API 및 데이터 구조 구문

```
SQL_API_RC SQL_API_FN db2secServerAuthPluginInit
                              ( db2int32 version,
                                void *server_fns,
                                db2secGetConDetails *getConDetails_fn,
                                db2secLogMessage *logMessage_fn,
                                char               **errormsg,
                                db2int32 *errormsglen );
```

db2secServerAuthPluginInit API 매개변수

version

입력. 현재 DB2 데이터베이스 관리 프로그램을 지원하는 가장 높은 버전의 API. db2secPlugin.h에 정의된 DB2SEC_API_VERSION 값에는 DB2 데이터베이스 관리 프로그램이 현재 지원하는 최신 버전의 API가 있습니다.

server_fns

출력. GSS-API 인증이 사용되는 경우 db2secGssapiServerAuthFunctions_<version_number> 구조(gssapi_server_auth_functions_<version_number>라고도 함), 사용자 ID/암호 인증이 사용되는 경우 db2secUseridPasswordServerAuthFunctions_<version_number> 구조(userid_password_server_auth_functions_<version_number>라고도 함)용으로 DB2 데이터베이스 관리 프로그램에서 제공하는 메모리의 포인터. db2secGssapiServerAuthFunctions_<version_number> 구조 및 db2secUseridPasswordServerAuthFunctions_<version_number> 구조에는 각각 GSS-API 인증 플러그인 및 사용자 ID/암호 인증 플러그인에 구현된 API의 포인터가 있습니다.

server_fns 매개변수는 플러그인이 구현한 버전에 해당하는 gssapi_server_auth_functions_<version_number> 구조의 포인터로 캐스트됩니다. gssapi_server_auth_functions_<version_number> 구조 또는

userid_password_server_auth_functions_<version_number> 구조의 첫 번째 매개변수는 플러그인이 구현한 API 버전을 DB2 데이터베이스 관리 프로그램에 알려줍니다.

주: 캐스팅은 DB2 버전이 플러그인이 구현한 API 버전과 같거나 높을 경우에만 수행됩니다.

gssapi_server_auth_functions_<version_number> 또는 userid_password_server_auth_functions_<version_number> 구조 내에서 plugintype 매개변수는 DB2SEC_PLUGIN_TYPE_USERID_PASSWORD, DB2SEC_PLUGIN_TYPE_GSSAPI 또는 DB2SEC_PLUGIN_TYPE_KERBEROS로 설정되어야 합니다. 향후 버전의 API에서는 다른 값이 정의될 수 있습니다.

getConDetails_fn

입력. DB2 가 구현하는 db2secGetConDetails API의 포인터. db2secServerAuthPluginInit API는 다른 인증 API 중 하나에 db2secGetConDetails API 호출하여 데이터베이스 연결과 관련된 세부사항을 가져올 수 있습니다. 이러한 세부사항에는 연결과 관련된 통신 메커니즘(예: TCP/IP의 경우, IP 주소)에 대한 정보가 포함됩니다. 이 정보는 플러그인 기록기가 인증을 결정할 때 참조해야 하는 정보일 수 있습니다. 예를 들어, 플러그인은 사용자가 특정 IP 주소에서 연결하지 않으면 이 사용자의 연결을 허용하지 않을 수 있습니다. db2secGetConDetails API는 선택적으로 사용할 수 있습니다.

db2secGetConDetails API가 데이터베이스 연결과 관련되지 않은 상황에서 호출되면 DB2SEC_PLUGIN_NO_CON_DETAILS 값을 리턴하고, 그렇지 않으면 성공을 나타내는 0을 리턴합니다.

db2secGetConDetails API는 두 개의 입력 매개변수 즉, db2sec_con_details_<version_number> 구조의 포인터인 pConDetails와 db2sec_con_details 구조가 사용할 버전 번호를 나타내는 conDetailsVersion을 사용합니다. db2sec_con_details1이 사용되면 DB2SEC_CON_DETAILS_VERSION_1, db2sec_con_details2가 사용되면 DB2SEC_CON_DETAILS_VERSION_2 값을 사용할 수 있습니다. 버전 번호 DB2SEC_CON_DETAILS_VERSION_2를 사용하는 것이 좋습니다.

리턴에 성공하면, db2sec_con_details 구조(db2sec_con_details1 또는 db2sec_con_details2)에 다음 정보가 포함됩니다.

- 서버에 연결하는 데 사용된 프로토콜. 프로토콜 정의 목록은 sqlenv.h 파일(포함 디렉토리에 있음)에 있습니다(SQL_PROTOCOL_*). 이 정보는 clientProtocol 매개변수에 채워집니다.

- clientProtocol이 SQL_PROTOCOL_TCPIP or SQL_PROTOCOL_TCPIP4 일 경우 서버와 인바운드 연결의 TCP/IP 주소. 이 정보는 clientIPAddress 매개변수에 채워집니다.
- 클라이언트가 연결을 시도하는 데이터베이스 이름. 이 정보는 인스턴스 첨부에 설정되지 않습니다. 이 정보는 dbname 및 dbnameLen 매개변수에 채워집니다.
- db2secValidatePassword API의 connection_details 매개변수에 설명된 것과 동일한 세부사항이 있는 연결 정보 비트맵. 이 정보는 connect_info_bitmap 매개변수에 채워집니다.
- clientProtocol이 SQL_PROTOCOL_TCPIP6일 경우 서버와 인바운드 연결의 TCP/IP 주소. 이 정보는 clientIP6Address 매개변수에 채워지며, DB2SEC_CON_DETAILS_VERSION_2가 db2secGetConDetails API 호출에 사용되는 경우에만 사용할 수 있습니다.

logMessage_fn

입력. DB2 데이터베이스 관리 프로그램이 구현하는 db2secLogMessage API의 포인터. db2secClientAuthPluginInit API는 db2secLogMessage API를 호출하여 디버깅 또는 정보용으로 db2diag 로그 파일에 메시지를 로그할 수 있습니다. db2secLogMessage API의 첫 번째 매개변수(level)는 db2diag 로그 파일에 기록되는 진단 오류의 유형을 지정하고, 마지막 두 매개변수는 각각 메시지 문자열과 길이입니다. `db2secPlugin.h`에 정의된 dbesecLogMessage API의 첫 번째 매개변수의 올바른 값은 다음과 같습니다.

DB2SEC_LOG_NONE (0)
로깅 없음

DB2SEC_LOG_CRITICAL (1)
서버 오류 발견

DB2SEC_LOG_ERROR (2)
오류 발견

DB2SEC_LOG_WARNING (3)
경고

DB2SEC_LOG_INFO (4)
정보용

db2secLogMessage API의 'level' 매개변수 값이 diaglevel 데이터베이스 관리 프로그램 구성 매개변수 이하일 경우에만 db2diag 로그 파일에 메시지 텍스트가 표시됩니다.

예를 들어, DB2SEC_LOG_INFO 값을 사용하는 경우 diaglevel 데이터베이스 관리 프로그램 구성 매개변수가 4로 설정된 경우에만 db2diag 로그 파일에 메시지 텍스트가 표시됩니다.

errormsg

출력. db2secServerAuthPluginInit API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secServerAuthPluginTerm API - 서버 인증 플러그인 자원 정리

db2secServerAuthPluginTerm API는 서버 인증 플러그인에 사용된 자원을 제거합니다.

이 API는 DB2 데이터베이스 관리 프로그램이 서버 인증 플러그인을 언로드하기 직전에 호출합니다. 이 API는 플러그인 라이브러리가 보유하는 자원을 적절하게 정리하는 방법으로 구현해야 합니다. 예를 들어, 플러그인에서 할당한 메모리를 제거하고, 열려 있는 파일을 닫고, 네트워크 연결을 닫습니다. 플러그인은 자원 제거를 위해 자원을 추적해야 합니다. 이 API는 Windows 플랫폼에서는 호출되지 않습니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secServerAuthPluginTerm)
                       ( char     **errormsg,
                         db2int32 *errormsglen );
```

db2secServerAuthPluginTerm API 매개변수

errormsg

출력. db2secServerAuthPluginTerm API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

db2secValidatePassword API - 암호 유효성 확인

데이터베이스 연결을 조작하는 중에 사용자 ID 및 암호 스타일 인증을 수행할 수 있는 방법을 제공합니다.

주: 클라이언트 측에서 API가 실행될 때, API 코드는 CONNECT문을 실행하는 사용자의 특권으로 실행됩니다. 이 API는 인증 구성 매개변수가 CLIENT로 설정된 경우에만 클라이언트 측에서 호출됩니다.

서버 측에서 API가 실행될 때, API 코드는 인스턴스 소유자의 특권으로 실행됩니다.

인증에 특수 특권(예: UNIX에서 루트 수준 시스템 액세스 권한)이 필요할 경우 플러그인 기록기는 이러한 사항을 고려해야 합니다.

이 API는 암호가 유효하면 DB2SEC_PLUGIN_OK(성공)를, 암호가 유효하지 않으면 오류 코드(예: DB2SEC_PLUGIN_BADPWD)를 리턴해야 합니다.

API 및 데이터 구조 구문

```
SQL_API_RC ( SQL_API_FN *db2secValidatePassword)
                       ( const char *userid,
                         db2int32 useridlen,
                         const char *usernamespace,
                         db2int32 usernamespacelen,
                         db2int32 usernamespacetype,
                         const char *password,
                         db2int32 passwordlen,
                         const char *newpasswd,
                         db2int32 newpasswdlen,
                         const char *dbname,
                         db2int32 dbnamelen,
                         db2Uint32 connection_details,
                         void      **token,
                         char      **errormsg,
                         db2int32 *errormsglen );
```

db2secValidatePassword API 매개변수

userid 입력. 암호의 유효성을 확인할 사용자 ID.

useridlen

입력. userid 매개변수 값의 길이(바이트).

usernamespace

입력. 사용자 ID를 가져온 이름 스페이스.

usernamespacelen

입력. usernamespace 매개변수 값의 길이(바이트).

usernamespacetype

입력. 이름 스페이스 유형. db2secPlugin.h에 정의된 usernamespacetype 매개변수에 유효한 값은 다음과 같습니다.

- DB2SEC_NAMESPACE_SAM_COMPATIBLE 사용자 이름 스타일에 해당(예: domain₩myname)
- DB2SEC_NAMESPACE_USER_PRINCIPAL 사용자 이름 스타일에 해당(예: myname@domain.ibm.com)

현재 DB2 데이터베이스 시스템은
DB2SEC_NAMESPACE_SAM_COMPATIBLE 값만 지원합니다. 사용자 ID를 사용할 수 없는 경우, usernamespacetype 매개변수가 db2secPlugin.h에 정의된 DB2SEC_USER_NAMESPACE_UNDEFINED로 설정됩니다.

password

입력. 유효성을 확인할 암호.

passwordlen

입력. password 매개변수 값의 길이(바이트).

newpasswd

입력. 새 암호(암호가 변경되는 경우). 암호 변경 요청이 없는 경우 이 매개변수는 NULL로 설정됩니다. 이 매개변수가 NULL로 설정되지 않으면, 새 암호로 변경하기 전에 API가 이전 암호의 유효성을 확인합니다. API가 암호 변경 요청을 이행할 필요는 없지만, 이행하지 않을 경우 이전 암호의 유효성을 확인하지 않고 리턴 값
DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED를 즉시 리턴해야 합니다.

newpasswdlen

입력. newpasswd 매개변수 값의 길이(바이트).

dbname

입력. 연결할 데이터베이스의 이름. API가 dbname 매개변수를 무시하거나, 유효한 암호를 가진 사용자의 특정 데이터베이스에 대한 액세스를 제한하는 규정이 있는 경우 리턴 값 DB2SEC_PLUGIN_CONNECTIONREFUSED를 리턴할 수 있습니다. 이 매개변수는 NULL이 될 수 있습니다.

dbnamelen

입력. dbname 매개변수 값의 길이(바이트). dbname 매개변수가 NULL이면 이 매개변수가 0으로 설정됩니다.

connection_details

입력. 다음 정보를 저장하는 데 현재 3비트가 사용되는 32비트 매개변수.

- 맨 오른쪽 비트는 사용자 ID 소스가 db2secGetDefaultLoginContext API의 디폴트인지, 연결 중에 명시적으로 제공된 것인지를 나타냅니다.
- 오른쪽에서 두 번째 비트는 연결이 로컬(파티션된 데이터베이스 환경에서 db2nodes.cfg의 노드 중 하나로부터의 연결 또는 IPC(Inter Process Communication) 사용) 또는 리모트(네트워크 또는 루프백을 통해)인지를 나타냅니다. 이를 통해 API는 동일한 시스템에 있는 클라이언트가 암호를 사용하지 않고 DB2 서버에 연결할 수 있는지 결정할 수 있습니다. 디폴트 운

영 체제 기반 사용자 ID/암호 플러그인으로 인해, 동일한 시스템에 있는 클라이언트가 제공하는 암호 없이 로컬 연결이 허용됩니다(사용자가 연결 특권을 가지고 있다고 가정).

- 오른쪽에서 세 번째 비트는 DB2 데이터베이스 관리 프로그램이 API를 서버 측 또는 클라이언트 측에서 호출하는지를 나타냅니다.

비트 값은 db2secPlugin.h에 정의됩니다.

- DB2SEC_USERID_FROM_OS(0x00000001): OS로부터 사용자 ID를 가져오며 연결 명령문에 명시적으로 제공되지 않음을 나타냅니다.
- DB2SEC_CONNECTION_ISLOCAL(0x00000002): 로컬 연결을 나타냅니다.
- DB2SEC_VALIDATING_ON_SERVER_SIDE(0x0000004): DB2 데이터베이스 관리 프로그램이 서버 측 또는 클라이언트 측에서 호출하여 암호의 유효성을 확인하는지를 나타냅니다. 이 비트 값이 설정되면 DB2 데이터베이스 관리 프로그램이 서버 측에서 호출합니다. 그렇지 않으면 클라이언트 측에서 호출합니다.

내재적 인증에 대한 DB2 데이터베이스 시스템 디폴트 동작은 암호의 유효성을 확인하지 않고 연결을 허용하는 것입니다. 하지만 플러그인 개발자가 DB2SEC_PLUGIN_BADPASSWORD 오류를 실행하여 내재적 인증을 허용하지 않을 수 있습니다.

token 입력. 현재 연결 중에 연속 API 호출에 매개변수로 전달할 수 있는 데이터의 포인터. 호출할 수 있는 API로는 db2secGetAuthIDs API 및 db2secGetGroupsForUser API가 있습니다.

errormsg

출력. db2secValidatePassword API 실행이 성공적이지 않을 경우 이 매개변수에 리턴될 수 있는 플러그인이 할당한 ASCII 오류 메시지 문자열 주소의 포인터.

errormsglen

출력. errormsg 매개변수에 있는 오류 메시지 문자열의 길이(바이트 단위)를 나타내는 정수의 포인터.

GSS-API 인증 플러그인의 필수 API 및 정의

다음은 DB2 보안 플러그인 인터페이스에 필요한 GSS-API의 전체 목록입니다.

해당 스펙을 따르는 지원되는 API: *Generic Security Service Application Program Interface, 버전 2*(IETF RFC2743) 및 *Generic Security Service API 버전 2: C-바인딩*(IETF RFC2744). GSS-API 기반 플러그인을 구현하기 전에, 이러한 스펙을 완전히 이해해야 합니다.

표 38. GSS-API 인증 플러그인의 필수 API 및 정의

이름		설명
클라이언트 측 API	gss_init_sec_context	피어 응용프로그램을 사용하여 보안 컨텍스트를 시작합니다.
서버 측 API	gss_accept_sec_context	피어 응용프로그램을 사용하여 시작된 보안 컨텍스트를 승인합니다.
서버 측 API	gss_display_name	내부 형식 이름을 텍스트로 변환합니다.
일반 API	gss_delete_sec_context	설정된 보안 컨텍스트를 삭제합니다.
일반 API	gss_display_status	GSS-API 상태 코드와 연관된 텍스트 오류 메시지를 가져옵니다.
일반 API	gss_release_buffer	버퍼를 삭제합니다.
일반 API	gss_release_cred	GSS-API 증명서와 연관된 로컬 데이터 구조를 릴리스합니다.
일반 API	gss_release_name	내부 형식 이름을 삭제합니다.
필수 정의	GSS_C_DELEG_FLAG	삭제를 요청합니다.
필수 정의	GSS_C_EMPTY_BUFFER	gss_buffer_desc에 데이터가 포함되어 있지 않음을 나타냅니다.
필수 정의	GSS_C_GSS_CODE	GSS 주 상태 코드를 나타냅니다.
필수 정의	GSS_C_INDEFINITE	메커니즘이 컨텍스트 만기를 지원하지 않음을 나타냅니다.
필수 정의	GSS_C_MECH_CODE	GSS 부 상태 코드를 나타냅니다.
필수 정의	GSS_C_MUTUAL_FLAG	상호 인증이 요청되었습니다.
필수 정의	GSS_C_NO_BUFFER	gss_buffer_t 변수가 유효한 gss_buffer_desc 구조를 가리키지 않음을 나타냅니다.
필수 정의	GSS_C_NO_CHANNEL_BINDINGS	통신 채널 바인딩이 없습니다.
필수 정의	GSS_C_NO_CONTEXT	gss_ctx_id_t 변수가 유효한 컨텍스트를 가리키지 않음을 나타냅니다.
필수 정의	GSS_C_NO_CREDENTIAL	gss_cred_id_t 변수가 유효한 증명서 핸들을 가리키지 않음을 나타냅니다.
필수 정의	GSS_C_NO_NAME	gss_name_t 변수가 유효한 내부 이름을 가리키지 않음을 나타냅니다.
필수 정의	GSS_C_NO_OID	디폴트 인증 메커니즘을 사용합니다.
필수 정의	GSS_C_NULL_OID_SET	디폴트 메커니즘을 사용합니다.
필수 정의	GSS_S_COMPLETE	API가 완료되었습니다.
필수 정의	GSS_S_CONTINUE_NEEDED	처리가 완료되지 않았으며 피어에서 수신한 회신 토큰을 사용하여 API를 다시 호출해야 합니다.

GSS-API 인증 플러그인의 제한사항

다음은 GSS-API 인증 플러그인의 제한사항 목록입니다.

- 디폴트 보안 메커니즘이 있는 것으로 항상 간주되므로 OID 고려사항은 없습니다.
- gss_init_sec_context()에 요청되는 유일한 GSS 서비스는 상호 인증 및 위임입니다. DB2 데이터베이스 관리 프로그램은 항상 위임을 위한 티켓을 요청하지만 이 티켓을 사용하여 새 티켓을 생성하지는 않습니다.
- 디폴트 컨텍스트 시간만 요청됩니다.

- gss_delete_sec_context()의 컨텍스트 토큰은 클라이언트에서 서버로, 또는 서버에서 클라이언트로 전송되지 않습니다.
- 익명이 지원되지 않습니다.
- 채널 바인딩이 지원되지 않습니다.
- 처음 증명서가 만기되면 DB2 데이터베이스 관리 프로그램은 이 증명서를 자동으로 갱신하지 않습니다.
- GSS-API 스펙은 gss_init_sec_context() 또는 gss_accept_sec_context()가 실패해도 두 함수 중 하나가 토큰을 리턴하여 피어에 전송하도록 합니다. 하지만 DRDA 제한사항으로 인해, DB2 데이터베이스 관리 프로그램은 gss_init_sec_context()가 실패하는 경우에만 토큰을 전송하고 첫 번째 호출에서 토큰을 생성합니다.

제 9 장 감사 기능 레코드 레이아웃

감사 로그에서 감사 레코드가 추출되면 각 레코드는 다음 표에 표시된 형식 중 하나를 갖습니다. 각 테이블 앞에는 샘플 레코드가 있습니다.

레코드의 각 항목에 대한 설명은 해당 테이블에서 한 번에 하나의 행에 표시됩니다. 각 항목은 추출 조작 후 구분된 파일에 출력된 것과 동일한 순서로 테이블에 표시됩니다.

주:

1. 샘플 레코드의 모든 필드가 값을 갖지는 않습니다.
2. 『액세스가 시도됨』과 같은 일부 필드는 컬럼 식별자가 있는 ASCII 형식의 비트맵으로 저장됩니다. 그러나 이 플랫 보고서 파일에서는 이러한 필드가 비트맵 값을 나타내는 문자열 세트로 표시됩니다.

감사 레코드 오브젝트 유형

다음 표는 각 감사 레코드 오브젝트 유형에 CHECKING, OBJMAINT 및 SECMAINT 이벤트를 생성할 수 있는지 여부를 보여줍니다.

표 39. 감사 이벤트를 기초로 한 감사 레코드 오브젝트 유형

오브젝트 유형	CHECKING 이벤트	OBJMAINT 이벤트	SECMAINT 이벤트
ACCESS_RULE			X
ALIAS	X	X	
ALL	X		
AUDIT_POLICY	X	X	
BUFFERPOOL	X	X	
CHECK_CONSTRAINT		X	
DATABASE	X		X
DATA TYPE		X	
EVENT_MONITOR	X	X	
FOREIGN_KEY		X	
FUNCTION	X	X	X
FUNCTION MAPPING	X	X	
GLOBAL_VARIABLE	X	X	X
HISTOGRAM TEMPLATE	X	X	
INDEX	X	X	X
INDEX EXTENSION		X	
INSTANCE	X		
JAR_FILE		X	
METHOD_BODY	X	X	X

표 39. 감사 이벤트를 기초로 한 감사 레코드 오브젝트 유형 (계속)

오브젝트 유형	CHECKING 이벤트	OBJMAINT 이벤트	SECMAINT 이벤트
MODULE	X	X	X
NICKNAME	X	X	X
NODEGROUP	X	X	
NONE	X	X	X
OPTIMIZATION PROFILE	X		
PACKAGE	X	X	X
PACKAGE CACHE	X		
PRIMARY_KEY		X	
REOPT_VALUES	X		
ROLE	X	X	X
SCHEMA	X	X	X
SECURITY LABEL		X	X
SECURITY LABEL COMPONENT		X	
SECURITY POLICY		X	X
SEQUENCE	X	X	
SERVER	X	X	X
SERVER OPTION	X	X	
SERVICE CLASS	X	X	
STORED_PROCEDURE	X	X	X
SUMMARY TABLES	X	X	X
TABLE	X	X	X
TABLESPACE	X	X	X
THRESHOLD	X	X	
TRIGGER		X	
TRUSTED CONTEXT	X	X	X
TYPE MAPPING	X	X	
TYPE&TRANSFORM	X	X	
UNIQUE_CONSTRAINT		X	
USER MAPPING	X	X	
VIEW	X	X	X
WORK ACTION SET	X	X	
WORK CLASS SET	X	X	
WORKLOAD	X	X	X
WRAPPER	X	X	
XSR 오브젝트	X	X	X

AUDIT 이벤트에 대한 레코드 레이아웃 감사

다음 표는 AUDIT 이벤트에 사용되는 감사 레코드의 레이아웃을 보여줍니다.

샘플 감사 기록:

```
timestamp=2007-04-10-08.29.52.000001;
category=AUDIT;
audit event=START;
event correlator=0;
event status=0;
userid=newton;
authid=NEWTON;
application id=*LOCAL_APPLICATION;
application name=db2audit.exe;
```

표 40. AUDIT 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. AUDIT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 AUDIT 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이 이름이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
Application Name	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름
패키지 섹션	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
패키지 버전	VARCHAR(64)	감사 이벤트 발생 시 사용 중인 패키지 버전
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조로서 트랜잭션 로그의 일부입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값

표 40. AUDIT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할
규정 이름	VARCHAR(128)	감사 규정의 이름
규정 연관 오브젝트 유형	CHAR(1)	감사 규정이 연결되어 있는 오브젝트의 유형. 가능한 값은 다음과 같습니다. • N = 별칭 • S = MQT • T = 테이블(유형이 지정되지 않음) • i = 권한 부여 ID • g= 권한 • x = 트러스트된 컨텍스트 • 공백 = 데이터베이스
규정 연관 서브오브젝트 유형	CHAR(1)	감사 규정이 연결되어 있는 서브오브젝트의 유형. 오브젝트 유형이 ?(권한 부여 ID)인 경우, 가능한 값은 다음과 같습니다. • U = 사용자 • G = 그룹 • R = 역할
규정 연관 오브젝트 이름	VARCHAR(128)	감사 규정이 연결되어 있는 오브젝트의 이름.
규정 연관 오브젝트 스키마	VARCHAR(128)	감사 규정이 연결되어 있는 오브젝트의 스키마 이름. 규정 연관 오브젝트 유형이 스키마가 적용되지 않은 오브젝트를 식별하는 경우 이 이름은 NULL입니다.
감사 상태	CHAR(1)	감사 규정에서 AUDIT 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공
확인 상태	CHAR(1)	감사 규정에서 CHECKING 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공

표 40. AUDIT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
컨텍스트 상태	CHAR(1)	감사 규정에서 CONTEXT 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공
실행 상태	CHAR(1)	감사 규정에서 EXECUTE 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공
데이터를 포함한 실행	CHAR(1)	감사 규정에서 EXECUTE 범주의 WITH DATA 옵션. 가능한 값은 다음과 같습니다. • Y - 데이터 포함 • N - 데이터 포함 안함
Objmaint 상태	CHAR(1)	감사 규정에서 OBJMAINT 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공
Secmaint 상태	CHAR(1)	감사 규정에서 SECMAINT 범주의 상태. 가능한 값은 감사 상태 필드를 참조하십시오.
Sysadmin 상태	CHAR(1)	감사 규정에서 SYSADMIN 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공
유효성 확인 상태	CHAR(1)	감사 규정에서 VALIDATE 범주의 상태. 가능한 값은 다음과 같습니다. • B - 둘 다 • F - 실패 • N - 없음 • S - 성공
오류 유형	CHAR(8)	감사 규정의 오류 유형. 가능한 값은 AUDIT 및 NORMAL입니다.
데이터 경로	VARCHAR(1024)	db2audit configure 명령에 지정된 활성 감사 로그의 경로
아카이브 경로	VARCHAR(1024)	db2audit configure 명령에 지정된 아카이브된 감사 로그의 경로

CHECKING 이벤트의 감사 레코드 레이아웃

다음 표는 CHECKING 이벤트의 감사 레코드 형식을 보여줍니다.

샘플 감사 기록:

```
timestamp=1998-06-24-08.42.11.622984;
category=CHECKING;
audit event=CHECKING_OBJECT;
event correlator=2;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SYSSH200;
package section=0;
object schema=GSTAGER;
object name=NONE;
object type=REOPT_VALUES;
access approval reason=DBADM;
access attempted=STORE;
```

표 41. CHECKING 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. CHECKING
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 CHECKING 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름

표 41. CHECKING 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
패키지 섹션 번호	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
오브젝트 스키마	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 스키마
오브젝트 이름	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 이름
오브젝트 유형	VARCHAR(32)	감사 이벤트가 생성된 오브젝트의 유형. 가능한 값은 『감사 레코드 오브젝트 유형』이라는 제목의 주제에 표시됩니다.
액세스 승인 이유	CHAR(18)	액세스가 이 감사 이벤트에 승인된 이유를 나타냅니다. 가능한 값은 『가능한 CHECKING 액세스 승인 이유 목록』이라는 제목의 주제에 표시됩니다.
시도된 액세스	CHAR(18)	시도된 액세스 유형을 나타냅니다. 가능한 값은 『가능한 CHECKING 액세스가 시도된 유형 목록』이라는 주제에 표시됩니다.
패키지 버전	VARCHAR (64)	감사 이벤트 발생 시 사용 중인 패키지 버전
점검된 권한 부여 ID	VARCHAR(128)	권한 부여 ID는 감사 이벤트 발생 시 권한 부여 ID와 다를 때 점검됩니다. 예를 들어, 이 ID는 TRANSFER OWNERSHIP문의 목표 소유자가 될 수 있습니다. 감사 이벤트가 SWITCH_USER인 경우, 이 필드는 전환되는 권한 부여 ID를 나타냅니다.
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조의 데이터 필드로서 트랜잭션 로그의 일부입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할

CHECKING 액세스 승인 이유

다음 목록은 가능한 CHECKING 액세스 승인 이유를 보여줍니다.

감사 레코드 하나에 여러 개의 감사 승인 이유가 포함될 수 있습니다(예: `access approval reason=DATAACCESS,ACCESSCTRL;`). 여러 개의 액세스 승인 이유가 있는 경우, 사용자가 시도된 액세스에 대해 권한 부여 점검을 전달하기 위해서는 명시된 모든 권한 및 특권이 있어야 합니다.

0x0000000000000001 ACCESS DENIED

액세스가 승인되지 않고 거부되었습니다.

0x0000000000000002 SYSADM

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SYSADM 권한을 가집니다.

0x0000000000000004 SYSCTRL

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SYSCTRL 권한을 가집니다.

0x0000000000000008 SYSMAINT

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SYSMAINT 권한을 가집니다.

0x0000000000000010 DBADM

액세스가 승인되었으며, 응용프로그램 또는 사용자가 DBADM 권한을 가집니다.

0x0000000000000020 DATABASE

액세스가 승인되었으며, 응용프로그램 또는 사용자가 데이터베이스에서 명시적 특권을 가집니다.

0x0000000000000040 OBJECT

액세스가 승인되었으며, 응용프로그램 또는 사용자가 오브젝트 또는 기능에 특권을 가집니다.

0x0000000000000080 DEFINER

액세스가 승인되었으며, 응용프로그램 또는 사용자가 오브젝트 또는 기능의 정의자입니다.

0x0000000000000100 OWNER

액세스가 승인되었으며, 응용프로그램 또는 사용자가 오브젝트 또는 기능의 소유자입니다.

0x0000000000000200 CONTROL

액세스가 승인되었으며, 응용프로그램 또는 사용자가 오브젝트 또는 기능에서 CONTROL 특권을 가집니다.

0x0000000000000400 BIND

액세스가 승인되었으며, 응용프로그램 또는 사용자가 패키지에서 바인드 특권을 가집니다.

0x0000000000000800 SYSQUIESCE

액세스가 승인되었으며, 인스턴스 또는 데이터베이스가 Quiesce 모드에 있으면 응용프로그램 또는 사용자가 연결 또는 접속할 수도 있습니다.

0x0000000000001000 SYSMON

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SYSMON 권한을 가집니다.

0x0000000000002000 SECADM

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SECADM 권한을 가집니다.

0x0000000000004000 SETSESSIONUSER

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SETSESSIONUSER 권한을 가집니다.

0x0000000000008000 TRUSTED_CONTEXT_MATCH

DB2 서버에서 정의된 고유한 트러스트된 컨텍스트의 속성과 연결 속성이 일치합니다.

0x0000000000010000 TRUSTED_CONTEXT_USE

트러스트된 컨텍스트를 사용하도록 액세스가 승인되었습니다.

0x0000000000020000 SQLADM

액세스가 승인되었으며, 응용프로그램 또는 사용자가 SQLADM 권한을 가집니다.

0x0000000000040000 WLMADM

액세스가 승인되었으며, 응용프로그램 또는 사용자가 WLMADM 권한을 가집니다.

0x0000000000080000 EXPLAIN

액세스가 승인되었으며, 응용프로그램 또는 사용자가 EXPLAIN 권한을 가집니다.

0x0000000000100000 DATAACCESS

액세스가 승인되었으며, 응용프로그램 또는 사용자가 DATAACCESS 권한을 가집니다.

0x0000000000200000 ACCESSCTRL

액세스가 승인되었으며, 응용프로그램 또는 사용자가 ACCESSSCTRL 권한을 가집니다.

CHECKING 액세스 시도 유형

다음 목록은 가능한 CHECKING 액세스가 시도된 유형을 보여줍니다.

감사 이벤트가 CHECKING_TRANSFER이면 감사 항목은 특권이 있는지 또는 없는지를 반영합니다.

0x0000000000000001 CONTROL

CONTROL 특권이 있는지 확인 시도

0x0000000000000002 ALTER

오브젝트 변경 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 ALTER 특권이 있는지 확인 시도

0x0000000000000004 DELETE

오브젝트 삭제 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 DELETE 특권이 있는지 확인 시도

0x0000000000000008 INDEX

인덱스 사용을 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 INDEX 특권이 있는지 확인 시도

0x0000000000000010 INSERT

오브젝트에 삽입 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 INSERT 특권이 있는지 확인 시도

0x0000000000000020 SELECT

테이블 쿼리 또는 보기 시도 또는 감사 이벤트가 CHECKING_TRANSFER 인 경우 SELECT 특권이 있는지 확인 시도

0x0000000000000040 UPDATE

데이터 갱신 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 UPDATE 특권이 있는지 확인 시도

0x0000000000000080 REFERENCE

오브젝트 사이에 참조 제한조건 설정 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 REFERENCE 특권이 있는지 확인 시도

0x0000000000000100 CREATE

오브젝트 작성 시도

0x0000000000000200 DROP

오브젝트 제거 시도

0x0000000000000400 CREATEIN

또 다른 스키마 내에서 오브젝트 작성 시도

0x0000000000000800 DROPIN

또 다른 스키마 내에서 발견된 오브젝트 제거 시도

0x0000000000001000 ALTERIN

또 다른 스키마 내에서 발견된 오브젝트 변경 또는 수정 시도

0x0000000000002000 EXECUTE

응용프로그램 실행 시도 또는 루틴 호출, 루틴(함수에만 적용됨)에서 함수 소스 생성 또는 DDL문에서 루틴 참조 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 EXECUTE 특권이 있는지 확인 시도

0x0000000000004000 BIND

응용프로그램 바인드 또는 준비 시도

0x0000000000008000 SET EVENT MONITOR

이벤트 모니터 스위치 설정 시도

0x0000000000010000 SET CONSTRAINTS

오브젝트에서 제한조건 설정 시도

0x0000000000020000 COMMENT ON

오브젝트에서 주석 작성 시도

0x0000000000040000 GRANT

다른 권한 부여 ID로 오브젝트에 특권 또는 역할을 부여하려고 시도합니다.

0x0000000000080000 REVOKE

권한 부여 ID에서 오브젝트의 특권 또는 역할을 취소하려고 시도합니다.

0x0000000000100000 LOCK

오브젝트 잠금 시도

0x0000000000200000 RENAME

오브젝트 이름 바꾸기 시도

0x0000000000400000 CONNECT

오브젝트 연결 시도

0x0000000000800000 Member of SYS Group

SYS 그룹의 구성원에 액세스하거나 사용하려고 시도

0x0000000001000000 Access All

보유된 오브젝트에서 모든 필수 특권으로 명령문 실행 시도(DBADM/SYSADM에만 사용)

0x0000000002000000 Drop All

다중 오브젝트 제거 시도

0x0000000004000000 LOAD

테이블 스페이스에서 테이블 로드 시도

0x0000000008000000 USE

테이블 스페이스에 테이블 작성 시도 또는 감사 이벤트가 CHECKING_TRANSFER인 경우 USE 특권이 있는지 확인 시도

0x0000000010000000 SET SESSION_USER

SET SESSION_USER문 실행 시도

0x0000000020000000 FLUSH

FLUSH문 실행 시도

0x0000000040000000 STORE

EXPLAIN_PREDICATE 테이블에서 다시 최적화된 명령문의 값을 보기 위한 시도

0x0000000400000000 TRANSFER

오브젝트 전송 시도

0x0000000800000000 ALTER_WITH_GRANT

ALTER with GRANT 특권이 있는지 확인 시도

0x0000001000000000 DELETE_WITH_GRANT

DELETE with GRANT 특권이 있는지 확인 시도

0x0000002000000000 INDEX_WITH_GRANT

INDEX with GRANT 특권이 있는지 확인 시도

0x0000004000000000 INSERT_WITH_GRANT

INSERT with GRANT 특권이 있는지 확인 시도

0x0000008000000000 SELECT_WITH_GRANT

SELECT with GRANT 특권이 있는지 확인 시도

0x0000010000000000 UPDATE_WITH_GRANT

UPDATE with GRANT 특권이 있는지 확인 시도

0x0000020000000000 REFERENCE_WITH_GRANT

REFERENCE with GRANT 특권이 있는지 확인 시도

0x0000040000000000 USAGE

시퀀스 또는 XSR 오브젝트 사용 시도 또는 점검 이벤트가 CHECKING_TRANSFER인 경우 USAGE 특권이 있는지 확인 시도

0x0000080000000000 SET ROLE

역할 설정 시도

0x0000100000000000 EXPLICIT_TRUSTED_CONNECTION

명시적으로 트러스트된 연결을 설정하려고 시도

0x0000200000000000 IMPLICIT_TRUSTED_CONNECTION

내재된 트러스트된 연결을 설정하려고 시도

0x0000400000000000 READ

전역 변수을 읽으려고 시도

0x0000800000000000 WRITE

전역 변수을 쓰려고 시도

0x0001000000000000 SWITCH_USER

명시적으로 트러스트된 연결에서 사용자 ID를 전환하려고 시도

0x0002000000000000 AUDIT_USING

감사 규정을 오브젝트와 연관시키려고 시도

0x0004000000000000 AUDIT_REPLACE

오브젝트와 감사 규정 연관을 바꾸려고 시도

0x0008000000000000 AUDIT_REMOVE

오브젝트와 감사 규정 연관을 제거하려고 시도

0x0010000000000000 AUDIT_ARCHIVE

감사 로그를 아카이브하려고 시도

0x0020000000000000 AUDIT_EXTRACT

감사 로그를 추출하려고 시도

0x0040000000000000 AUDIT_LIST_LOGS

감사 로그를 나열하려고 시도

0x0080000000000000 IGNORE_TRIGGERS

데이터베이스 오브젝트와 연관된 트리거를 무시하려고 시도

0x0100000000000000 PREPARE

SQL문을 준비하려고 시도하며 사용자에게 필요한 오브젝트 레벨 특권 또는 DATAACCESS 권한이 없습니다.

0x0200000000000000 DESCRIBE

명령문을 설명하려고 시도하며 사용자에게 필요한 오브젝트 레벨 특권 또는 DATAACCESS 권한이 없습니다.

OBJMAINT 이벤트의 감사 레코드 레이아웃

다음 표는 OBJMAINT 이벤트의 감사 레코드 형식을 보여줍니다.

샘플 감사 기록:

```
timestamp=1998-06-24-08.42.41.957524;
category=OBJMAINT;
audit event=CREATE_OBJECT;
  event correlator=3;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SQLC28A1;
package section=0;
object schema=BOSS;
object name=AUDIT;
object type=TABLE;
```

표 42. OBJMAINT 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. OBJMAINT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 OBJMAINT 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(256)	감사 이벤트 발생 시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
오브젝트 스키마	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 스키마
오브젝트 이름	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 이름
오브젝트 유형	VARCHAR(32)	감사 이벤트가 생성된 오브젝트의 유형. 가능한 값은 『감사 레코드 오브젝트 유형』이라는 제목의 주제에 표시됩니다.
패키지 버전	VARCHAR(64)	감사 이벤트 발생 시 사용 중인 패키지 버전
보안 규정 이름	VARCHAR(128)	오브젝트 유형이 TABLE이고 해당 테이블이 보안 규정과 연관되었을 경우 보안 규정의 이름

표 42. OBJMAINT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
변경 조치	VARCHAR(32)	특정 변경 조작 가능한 값은 다음과 같습니다. • ADD_PROTECTED_COLUMN • ADD_COLUMN_PROTECTION • DROP_COLUMN_PROTECTION • ADD_ROW_PROTECTION • ADD_SECURITY_POLICY • ADD_ELEMENT • ADD COMPONENT • USE GROUP AUTHORIZATIONS • IGNORE GROUP AUTHORIZATIONS • USE ROLE AUTHORIZATIONS • IGNORE ROLE AUTHORIZATIONS • OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL • RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL
보호된 컬럼 이름	VARCHAR(128)	변경 조치가 ADD_COLUMN_PROTECTION 또는 DROP_COLUMN_PROTECTION인 경우, 이 것이 영향을 받은 컬럼의 이름입니다.
컬럼 보안 레이블	VARCHAR(128)	필드 컬럼 이름에서 지정된 컬럼을 보호하는 보안 레이블
보안 레이블 컬럼 이름	VARCHAR(128)	행을 보호하는 보안 레이블을 포함한 컬럼의 이름
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조의 데이터 필드로서 트랜잭션 로그의 일부입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할
오브젝트 모듈	VARCHAR(128)	오브젝트가 속하는 모듈의 이름

SECMAINT 이벤트의 감사 레코드 레이아웃

다음 표는 SECMAINT 이벤트의 감사 레코드 형식을 보여줍니다.

샘플 감사 기록:

```
timestamp=1998-06-24-11.57.45.188101;
category=SECMAINT;
audit event=GRANT;
event correlator=4;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.boss.980624155728;
application name=db2bp;
package schema=NULLID;
package name=SQLC28A1;
package section=0;
object schema=BOSS;
object name=T1;
object type=TABLE;
grantor=BOSS;
grantee=WORKER;
grantee type=USER;
privilege=SELECT;
```

표 43. SECMAINT 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. SECMAINT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 SECMAINT 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름

표 43. SECMAINT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
오브젝트 스키마	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 스키마 오브젝트 유형 필드가 ACCESS_RULE인 경우 이 필드는 규칙과 연관된 보안 규정 이름을 포함합니다. 규칙의 이름은 오브젝트 이름 필드에 저장됩니다. 오브젝트 유형 필드가 SECURITY_LABEL인 경우 이 필드는 보안 레이블이 포함된 보안 규정의 이름을 가집니다. 보안 레이블의 이름은 오브젝트 이름 필드에 저장됩니다.
오브젝트 이름	VARCHAR(128)	감사 이벤트가 생성된 오브젝트의 이름 감사 이벤트가 다음 중 하나일 때 역할 이름을 나타냅니다. • ADD_DEFAULT_ROLE • DROP_DEFAULT_ROLE • ALTER_DEFAULT_ROLE • ADD_USER • DROP_USER • ALTER_USER_ADD_ROLE • ALTER_USER_DROP_ROLE • ALTER_USER_AUTHENTICATION 오브젝트 유형 필드가 ACCESS_RULE인 경우 이 필드는 규칙 이름을 포함합니다. 규칙과 연관된 규정 이름은 오브젝트 스키마 필드에 저장됩니다. 오브젝트 유형 필드가 SECURITY_LABEL인 경우 이 필드는 보안 레이블의 이름을 포함합니다. 보안 규정이 부분인 보안 규정의 이름은 오브젝트 스키마 필드에 저장됩니다.
오브젝트 유형	VARCHAR(32)	감사 이벤트가 생성된 오브젝트의 유형. 가능한 값은 『감사 레코드 오브젝트 유형』이라는 제목의 주제에 표시됩니다. 감사 이벤트가 다음 중 하나일 때 이 값은 ROLE입니다. • ADD_DEFAULT_ROLE • DROP_DEFAULT_ROLE • ALTER_DEFAULT_ROLE • ADD_USER • DROP_USER • ALTER_USER_ADD_ROLE • ALTER_USER_DROP_ROLE • ALTER_USER_AUTHENTICATION
권한 준 사용자	VARCHAR(128)	특권 또는 권한을 부여한 사용자 또는 취소한 사용자의 ID.

표 43. SECMAINT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
권한 받은 사용자	VARCHAR(128)	특권 또는 권한이 권한 부여되거나 권한 취소된 권한 받은 사용자 ID 감사 이벤트가 다음 중 하나일 때 트러스트된 컨텍스트 오브젝트를 나타냅니다. • ADD_DEFAULT_ROLE • DROP_DEFAULT_ROLE • ALTER_DEFAULT_ROLE • ADD_USER, DROP_USER • ALTER_USER_ADD_ROLE • ALTER_USER_DROP_ROLE • ALTER_USER_AUTHENTICATION
권한 받은 사용자 유형	VARCHAR(32)	권한 부여되거나 권한 취소된 권한 받은 사용자의 유형. 감사 이벤트가 다음 중 하나일 때 가능한 값은 USER, GROUP, ROLE, AMBIGUOUS 또는 TRUSTED_CONTEXT입니다. • ADD_DEFAULT_ROLE • DROP_DEFAULT_ROLE • ALTER_DEFAULT_ROLE • ADD_USER • DROP_USER • ALTER_USER_ADD_ROLE • ALTER_USER_DROP_ROLE • ALTER_USER_AUTHENTICATION
특권 또는 권한	CHAR(34)	특권 또는 권한 유형이 권한 부여되거나 권한 취소되었음을 나타냅니다. 가능한 값은 『가능한 SECMAINT 특권 또는 권한 목록』이라는 제목의 주제에 표시됩니다. 감사 이벤트가 다음 중 하나일 때 이 값은 ROLE MEMBERSHIP입니다. • ADD_DEFAULT_ROLE, DROP_DEFAULT_ROLE • ALTER_DEFAULT_ROLE • ADD_USER • DROP_USER • ALTER_USER_ADD_ROLE • ALTER_USER_DROP_ROLE • ALTER_USER_AUTHENTICATION
패키지 버전	VARCHAR(64)	감사 이벤트 발생 시 사용 중인 패키지 버전

표 43. SECMAINT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
액세스 유형	VARCHAR(32)	보안 레이블이 권한 부여되는 액세스 유형 가능한 값은 다음과 같습니다. • READ • WRITE • ALL 보안 규정이 변경되는 액세스 유형입니다. 가능한 값은 다음과 같습니다. • USE GROUP AUTHORIZATIONS • IGNORE GROUP AUTHORIZATIONS • USE ROLE AUTHORIZATIONS • IGNORE ROLE AUTHORIZATIONS • OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL • RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL
가정할 수 있는 권한 ID	VARCHAR(128)	권한 부여된 특권이 SETSESSIONUSER 특권인 경우 이는 권한 수령인이 세션 사용자로서 설정이 허용된 권한 ID입니다.
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조의 데이터 필드로서 트랜잭션 로그의 일부입니다.
권한 준 사용자 유형	VARCHAR(32)	권한 부여자의 유형. 가능한 값은 USER입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 사용자	VARCHAR(128)	감사 이벤트가 ADD_USER 또는 DROP_USER일 때 트러스트된 컨텍스트 사용자를 식별합니다.
트러스트된 컨텍스트 사용자 인증	INTEGER	감사 이벤트가 ADD_USER, DROP_USER 또는 ALTER_USER_AUTHENTICATION일 때 트러스트된 컨텍스트 사용자의 인증 설정을 지정합니다. 1 : 인증이 필요함 0 : 인증이 필요하지 않음
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할

SECMAINT 특권 또는 권한

다음 목록은 가능한 SECMAINT 특권 또는 권한을 보여줍니다.

0x00000000000000000000000000000001 Control Table
테이블 또는 뷰에서 권한 부여되거나 권한 취소된 Control 특권

0x00000000000000000000000000000002 ALTER
테이블 또는 시퀀스를 변경하기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000004 ALTER with GRANT
특권 권한 부여를 허용하여 테이블 또는 시퀀스를 변경하기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000008 DELETE TABLE
테이블 또는 뷰를 제거하기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000010 DELETE TABLE with GRANT
특권 권한 부여를 허용하여 테이블을 제거하기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000020 Table Index
인덱스에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000040 Table Index with GRANT
특권 권한 부여를 허용하여 인덱스에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000080 Table INSERT
테이블 또는 뷰의 삽입에서 부여되거나 취소된 특권

0x00000000000000000000000000000100 Table INSERT with GRANT
특권 권한 부여를 허용하여 테이블의 삽입 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000200 Table SELECT
테이블의 선택 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000400 Table SELECT with GRANT
특권 권한 부여를 허용하여 테이블의 선택 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000000800 Table UPDATE
테이블 또는 뷰의 갱신 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000001000 Table UPDATE with GRANT
특권 권한 부여를 허용하여 테이블 또는 뷰의 갱신 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000002000 Table REFERENCE
테이블의 참조 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000004000 Table REFERENCE with GRANT
특권 권한 부여를 허용하여 테이블의 참조 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000000020000 CREATEIN Schema
스키마에서 권한 부여되거나 권한 취소된 CREATEIN 특권

0x00000000000000000000000000040000 CREATEIN Schema with GRANT
특권 권한 부여를 허용하여 스키마에서 권한 부여되거나 권한 취소된 CREATEIN 특권

0x00000000000000000000000000080000 DROPIN Schema
스키마에서 권한 부여되거나 권한 취소된 DROPIN 특권

0x00000000000000000000000000100000 DROPIN Schema with GRANT
특권 권한 부여를 허용하여 스키마에서 권한 부여되거나 권한 취소된 DROPIN 특권

0x00000000000000000000000000200000 ALTERIN Schema
스키마에서 권한 부여되거나 권한 취소된 ALTERIN 특권

0x00000000000000000000000000400000 ALTERIN Schema with GRANT
특권 권한 부여를 허용하여 스키마에서 권한 부여되거나 권한 취소된 ALTERIN 특권

0x00000000000000000000000000800000 DBADM Authority
권한 부여되거나 권한 취소된 DBADM 권한

0x00000000000000000000000001000000 CREATETAB Authority
권한 부여되거나 권한 취소된 Createtab 권한

0x00000000000000000000000002000000 BINDADD Authority
권한 부여되거나 권한 취소된 Bindadd 권한

0x00000000000000000000000004000000 CONNECT Authority
권한 부여되거나 권한 취소된 CONNECT 권한

0x00000000000000000000000008000000 Create not fenced Authority
권한 부여되거나 권한 취소된 분리 권한을 작성하지 않음

0x00000000000000000000000010000000 Implicit Schema Authority
권한 부여되거나 권한 취소된 내재된 스키마 권한

0x00000000000000000000000020000000 Server PASSTHRU
이 서버에 대해 pass-through 기능을 사용하기 위해 권한 부여되거나 권한 취소된 특권(페더레이티드 데이터베이스 데이터 소스)

0x00000000000000000000000040000000 ESTABLISH TRUSTED CONNECTION
트러스트된 연결이 설정되었음

0x00000000000000000000000100000000 Table Space USE

테이블 스페이스에서 테이블을 작성하기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000200000000 Table Space USE with GRANT

특권이 허용된 테이블 스페이스에서 테이블을 작성하기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000400000000 Column UPDATE

하나 이상의 특정 테이블 컬럼의 갱신 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000000800000000 Column UPDATE with GRANT

특권 권한 부여를 허용하여 하나 이상의 특정 테이블 컬럼의 갱신 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000001000000000 Column REFERENCE

하나 이상의 특정 테이블 컬럼의 참조 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000002000000000 Column REFERENCE with GRANT

특권 권한 부여를 허용하여 하나 이상의 특정 테이블 컬럼의 참조 항목에서 권한 부여되거나 권한 취소된 특권

0x00000000000000000000004000000000 LOAD Authority

권한 부여되거나 권한 취소된 LOAD 권한

0x00000000000000000000008000000000 Package BIND

패키지에서 권한 부여되거나 권한 취소된 BIND 특권

0x00000000000000000000010000000000 Package BIND with GRANT

특권 권한 부여를 허용하여 패키지에서 권한 부여되거나 권한 취소된 BIND 특권

0x00000000000000000000020000000000 EXECUTE

패키지 또는 루틴에서 권한 부여되거나 권한 취소된 EXECUTE 특권

0x00000000000000000000040000000000 EXECUTE with GRANT

특권 권한 부여를 허용하여 패키지 또는 루틴에서 권한 부여되거나 권한 취소된 EXECUTE 특권

0x00000000000000000000080000000000 EXECUTE IN SCHEMA

스키마에 있는 모든 루틴에 대해 부여되거나 취소된 EXECUTE 특권

0x00000000000000000000100000000000 EXECUTE IN SCHEMA with GRANT

특권 부여가 허용된 상태에서 스키마에 있는 모든 루틴에 대해 부여되거나 취소된 EXECUTE 특권

0x00000000000000000000200000000000 EXECUTE IN TYPE
임의 유형의 모든 루틴에 대해 부여되거나 취소된 EXECUTE 특권

0x00000000000000000000400000000000 EXECUTE IN TYPE with GRANT
특권 부여가 허용된 상태에서 임의 유형의 모든 루틴에 대해 부여되거나 취소된 EXECUTE 특권

0x00000000000000000000800000000000 CREATE EXTERNAL ROUTINE
부여되거나 취소된 CREATE EXTERNAL ROUTINE 특권

0x00000000000000000001000000000000 QUIESCE_CONNECT
부여되거나 취소된 QUIESCE_CONNECT 특권

0x00000000000000000004000000000000 SECADM Authority
SECADM 권한 부여 또는 취소됨

0x00000000000000000008000000000000 USAGE Authority
시퀀스에서 권한 부여되거나 권한 취소된 USAGE 특권

0x00000000000000000010000000000000 USAGE with GRANT Authority
특권 권한 부여를 허용하여 시퀀스에서 권한 부여되거나 권한 취소된 USAGE 특권

0x00000000000000000020000000000000 WITH ADMIN Option
WITH ADMIN Option은 역할에서 권한 부여되거나 권한 취소됨

0x00000000000000000040000000000000 SETSESSIONUSER Privilege
SETSESSIONUSER 권한 부여 또는 취소됨

0x00000000000000000080000000000000 Exemption
예외 권한 부여 또는 취소됨

0x00000000000000000100000000000000 Security label
보안 레이블 권한 부여 또는 취소됨

0x00000000000000000200000000000000 WRITE with GRANT
특권 권한 부여를 허용하여 전역 변수를 쓰기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000000400000000000000 Role Membership
권한 부여되거나 권한 취소된 역할 멤버쉽

0x00000000000000000800000000000000 Role Membership with ADMIN Option
권한 부여되거나 권한 취소된 ADMIN 옵션이 포함된 역할 멤버쉽

0x00000000000000001000000000000000 READ
전역 변수를 읽기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000002000000000000000 READ with GRANT

특권 권한 부여를 허용하여 전역 변수를 읽기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000004000000000000000 WRITE

전역 변수를 쓰기 위해 권한 부여되거나 권한 취소된 특권

0x00000000000000010000000000000000 SQLADM

권한 부여되거나 권한 취소된 SQLADM 권한

0x00000000000000020000000000000000 WLMADM

권한 부여되거나 권한 취소된 WLMADM 권한

0x00000000000000040000000000000000 EXPLAIN

권한 부여되거나 권한 취소된 EXPLAIN 권한

0x00000000000000080000000000000000 DATAACCESS

권한 부여되거나 권한 취소된 DATAACCESS 권한

0x00000000000000100000000000000000 ACCESSCTRL

권한 부여되거나 권한 취소된 ACCESSCTRL 권한

SYSADMIN 이벤트의 감사 레코드 레이아웃

다음 표는 SYSADMIN 이벤트의 감사 레코드 레이아웃을 보여줍니다.

샘플 감사 기록:

```
timestamp=1998-06-24-11.54.04.129923;
category=SYSADMIN;
audit event=DB2AUDIT;
event correlator=1;
event status=0;
  userid=boss;authid=BOSS;
application id=*LOCAL.boss.980624155404;application name=db2audit;
```

표 44. SYSADMIN 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. SYSADMIN
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 SYSADMIN 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.

표 44. SYSADMIN 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
패키지 버전	VARCHAR(64)	감사 이벤트 발생 시 사용 중인 패키지 버전
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조의 데이터 필드로서 트랜잭션 로그의 일부입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할

VALIDATE 이벤트의 감사 레코드 레이아웃

다음 표는 VALIDATE 이벤트의 감사 레코드 형식을 보여줍니다.

샘플 감사 기록:

```
timestamp=2007-05-07-10.30.51.585626;
category=VALIDATE;
audit event=AUTHENTICATION;
event correlator=1;
event status=0;
userid=newton;
```

```
authid=NEWTON;
execution id=gstager;
application id=*LOCAL.gstager.070507143051;
application name=db2bp;
auth type=SERVER;
plugin name=IBMOSauthserver;
```

표 45. VALIDATE 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. VALIDATE
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값은 GET_GROUPS, GET_USERID, AUTHENTICATE_PASSWORD, VALIDATE_USER, AUTHENTICATION 및 GET_USERMAPPING_FROM_PLUGIN입니다.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID
실행 ID	VARCHAR(1024)	감사 이벤트 시 사용 중인 실행 ID
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
인증 유형	VARCHAR(32)	감사 이벤트 시 인증 유형
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
패키지 버전	VARCHAR(64)	감사 이벤트 발생 시 사용 중인 패키지 버전
플러그인 이름	VARCHAR(32)	감사 이벤트 발생 시 사용 중인 플러그인 이름
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조의 데이터 필드로서 트랜잭션 로그의 일부입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값

표 45. VALIDATE 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할의 이름

CONTEXT 이벤트의 감사 레코드 레이아웃

다음 표는 CONTEXT 이벤트에 사용되는 감사 레코드의 레이아웃을 보여줍니다.

샘플 감사 기록:

```
timestamp=1998-06-24-08.42.41.476840;
category=CONTEXT;
audit event=EXECUTE_IMMEDIATE;
  event correlator=3;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SQLC28A1;
package section=203;text=create table audit(c1 char(10), c2 integer);
```

표 46. CONTEXT 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 다음과 같습니다. CONTEXT
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 CONTEXT 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이것이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID 감사 이벤트가 SWITCH_USER인 경우, 이 필드는 전환되는 사용자 ID를 나타냅니다.

표 46. CONTEXT 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 권한 부여 ID 감사 이벤트가 SWITCH_USER인 경우, 이 필드는 전환되는 권한 부여 ID를 나타냅니다.
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름
패키지 섹션 번호	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
명령문 텍스트	CLOB(8M)	적용 가능한 경우, SQL 또는 XQuery문의 텍스트. SQL 또는 XQuery문 텍스트가 사용 가능하지 않은 경우 널(NULL)입니다.
패키지 버전	VARCHAR(64)	감사 이벤트 발생 시 사용 중인 패키지 버전
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조의 데이터 필드로서 트랜잭션 로그의 일부입니다.
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할

EXECUTE 이벤트의 감사 레코드 레이아웃

다음 표는 EXECUTE 범주의 일부로서 감사되는 모든 필드에 대해 설명합니다.

샘플 감사 기록:

주: 기타 감사 범주와는 달리, 감사 로그를 테이블 형식으로 표시할 경우 EXECUTE 범주에 하나의 이벤트를 설명하는 여러 개의 행이 표시될 수 있습니다. 첫 번째 레코드는 주 이벤트를 설명하며, 이벤트 컬럼에는 키워드 STATEMENT가 포함됩니다. 나머지 행은 매개변수 표시문자 또는 호스트 변수, 매개변수당 한 개 행을 설명하며 해당

이벤트 컬럼에는 키워드 DATA가 포함됩니다. 보고서 형식으로 감사 로그가 표시되면, 한 개 레코드가 있지만 여기에 있는 명령문 값에 여러 개의 항목이 있습니다. DATA 키워드는 테이블 형식으로만 표시됩니다.

```
timestamp=2006-04-10-13.20.51.029203;
  category=EXECUTE;
  audit event=STATEMENT;
  event correlator=1;
  event status=0;
  database=SAMPLE;
  userid=smith;
  authid=SMITH;
  session authid=SMITH;
  application id=*LOCAL.prodrig.060410172044;
  application name=myapp;
  package schema=NULLID;
  package name=SQLC2F0A;
  package section=201;
  uow id=2;
  activity id=3;
  statement invocation id=0;
  statement nesting level=0;
  statement text=SELECT * FROM DEPARTMENT WHERE DEPTNO = ? AND DEPTNAME = ?;
  statement isolation level=CS;
  compilation environment=
    isolation level=CS
    query optimization=5
    min_dec_div_3=NO
    degree=1
    sqlrules=DB2
    refresh age=+00000000000000.000000
    schema=SMITH
    maintained table type=SYSTEM
    resolution timestamp=2006-06-29-20.32.13.000000
    federated asynchrony=0;
  value index=0;
  value type=CHAR;
  value data=C01;
  value index=1;
  value type=VARCHAR;
  value index=INFORMATION CENTER;
```

표 47. EXECUTE 이벤트의 감사 레코드 레이아웃

이름	형식	설명
시간소인	CHAR(26)	감사 이벤트의 날짜 및 시간
범주	CHAR(8)	감사 이벤트의 범주. 가능한 값은 EXECUTE입니다.
감사 이벤트	VARCHAR(32)	특정 감사 이벤트 가능한 값 목록을 보려면 309 페이지의 『감사 이벤트』의 EXECUTE 범주 절을 참조하십시오.
이벤트 상관자	INTEGER	감사되는 조작에 대한 상관 ID. 단일 이벤트와 연관된 감사 레코드를 식별하기 위해 사용될 수 있습니다.

표 47. EXECUTE 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
이벤트 상태	INTEGER	SQLCODE가 표시하는 감사 이벤트의 상태 여기서, 성공 이벤트 > = 0 실패 이벤트 < 0
데이터베이스 이름	CHAR(8)	이벤트가 생성된 데이터베이스의 이름. 이 이름이 인스턴스 레벨 감사 이벤트인 경우 공백입니다.
사용자 ID	VARCHAR(1024)	감사 이벤트 시의 사용자 ID
권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 명령문 권한 부여 ID.
세션 권한 부여 ID	VARCHAR(128)	감사 이벤트 시의 세션 권한 부여 ID.
원래 노드 번호	SMALLINT	감사 이벤트가 발생한 노드 번호
코디네이터 노드 번호	SMALLINT	코디네이터 노드의 노드 번호
응용프로그램 ID	VARCHAR(255)	감사 이벤트 발생 시 사용 중인 응용프로그램 ID
응용프로그램 이름	VARCHAR(1024)	감사 이벤트 발생 시 사용 중인 응용프로그램 이름
클라이언트 사용자 ID	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT USERID 특수 레지스터의 값
클라이언트 어카운팅 문자열	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_ACCTNG 특수 레지스터의 값
클라이언트 워크스테이션 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_WRKSTNNAME 특수 레지스터의 값
클라이언트 응용프로그램 이름	VARCHAR(255)	감사 이벤트 발생 시 CURRENT CLIENT_APPLNAME 특수 레지스터의 값
트러스트된 컨텍스트 이름	VARCHAR(128)	트러스트된 연결과 연결된 트러스트된 컨텍스트의 이름.
연결 신뢰 유형	INTEGER	가능한 값은 다음과 같습니다. IMPLICIT_TRUSTED_ CONNECTION 및 EXPLICIT_TRUSTED_ CONNECTION
상속된 역할	VARCHAR(128)	트러스트된 연결을 통해 상속된 역할.
패키지 스키마	VARCHAR(128)	감사 이벤트 시 사용 중인 패키지의 스키마
패키지 이름	VARCHAR(128)	감사 이벤트 발생 시 사용 중인 패키지 이름

표 47. EXECUTE 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
패키지 섹션	SMALLINT	감사 이벤트 발생 시 사용 중인 패키지의 섹션 번호
패키지 버전	VARCHAR(164)	감사 이벤트 발생 시 사용 중인 패키지 버전
로컬 트랜잭션 ID	VARCHAR(10) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 로컬 트랜잭션 ID. 이 ID는 SQLU_TID 구조로서 트랜잭션 로그의 일부입니다.
전역 트랜잭션 ID	VARCHAR(30) FOR BIT DATA	감사 이벤트 발생 시 사용 중인 전역 트랜잭션 ID. 이 ID는 SQLP_GXID 구조로서 트랜잭션 로그의 일부입니다.
UOW ID	BIGINT	활동이 비롯되는 작업 단위(UOW) 식별자. 이 값은 응용프로그램 내의 각 작업 단위(UOW)에 대해 고유합니다.
활동 ID	BIGINT	작업 단위(UOW)내의 고유 활동 ID
명령문 호출 ID	BIGINT	SQL문이 실행된 루틴 호출의 식별자(ID). 이 값은 현재 중첩 레벨이 응용프로그램에서 활성 상태였을 때 발생한 루틴 호출의 번호를 나타냅니다. 이 요소를 명령문 중첩 레벨과 함께 사용하여 특정 SQL문의 호출을 고유하게 식별할 수 있습니다.
명령문 중첩 레벨	BIGINT	명령문이 실행되고 있었을 때 적용되는 중첩 또는 재귀 레벨. 각 중첩 레벨은 스토어드 프로시저 또는 사용자 정의 함수(UDF)의 중첩 또는 재귀 호출에 해당합니다.
활동 유형	VARCHAR(32)	활동의 유형. 가능한 값은 다음과 같습니다. • READ_DML • WRITE_DML • DDL • CALL • NONE
명령문 텍스트	CLOB(8M)	적용 가능한 경우, SQL 또는 XQuery문의 텍스트.

표 47. EXECUTE 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
명령문 분리 레벨	CHAR(8)	명령문이 실행되고 있었을 때 명령문에 적용되는 분리 값. 가능한 값은 다음과 같습니다. • NONE(분리가 지정되지 않음) • UR(언커미트 읽기) • CS(커서 안정성) • RS(읽기 안정성) • RR(반복 읽기)
컴파일 환경 설명	BLOB(8K)	SQL문을 컴파일할 때 사용된 컴파일 환경. 이 요소를 COMPILATION_ENV 테이블 함수 또는 SET COMPILATION ENVIRONMENT SQL문의 입력으로 제공할 수 있습니다.
수정된 행 수	INTEGER	다음 두 경우의 결과로서 삭제, 삽입, 갱신된 행의 총 수를 포함합니다. • 성공적인 삭제 조작 후 제한조건의 강제 시행 • 활성화된 트리거에서 트리거 SQL문의 처리 복합 SQL을 호출할 경우, 모든 부속 명령문 행 수의 누적을 포함합니다. 일부 경우에 필드는 내부 오류 포인터인 음수 값을 포함합니다. 이 값은 SQLCA의 sqlerrd(5) 필드에 해당합니다.
리턴된 행 수	BIGINT	명령문에서 리턴한 총 행 수입니다.
세이브포인트 ID	BIGINT	명령문이 실행되고 있었을 때 명령문에 적용되는 세이브포인트 ID. 감사 이벤트가 SAVEPOINT, RELEASE_SAVEPOINT 또는 ROLLBACK_SAVEPOINT일 경우 이 세이브포인트 ID는 각각 설정, 릴리스 또는 롤백되는 세이브포인트입니다.
명령문 값 색인	INTEGER	SQL문에 사용된 매개변수 표시 문자 또는 호스트 변수의 위치

표 47. EXECUTE 이벤트의 감사 레코드 레이아웃 (계속)

이름	형식	설명
명령문 값 유형	CHAR(16)	SQL문과 연관된 데이터 값 유형을 나타내는 문자열. 가능한 값은 INTEGER 또는 CHAR입니다.
명령문 값 데이터	CLOB(128K)	SQL문과 연관된 데이터 값을 나타내는 문자열. LOB, LONG, XML 및 구조화된 입력 매개변수는 제공되지 않습니다. 날짜, 시간 및 시간소인 필드는 ISO 형식으로 기록됩니다.

감사 이벤트

특정 유형의 이벤트가 각 감사 범주의 감사 레코드를 작성할 수 있습니다.

AUDIT 범주의 이벤트

- ALTER_AUDIT_POLICY
- ARCHIVE
- AUDIT_REMOVE
- AUDIT_REPLACE
- AUDIT_USING
- CONFIGURE
- CREATE_AUDIT_POLICY
- DB2AUD
- DROP_AUDIT_POLICY
- EXTRACT
- FLUSH
- LIST_LOGS
- PRUNE(버전 9.5 이상에서는 생성되지 않음)
- START
- STOP
- UPDATE_DBM_CFG

CHECKING 범주의 이벤트

- CHECKING_FUNCTION
- CHECKING_MEMBERSHIP_IN_ROLES
- CHECKING_OBJECT
- CHECKING_TRANSFER

CONTEXT 범주의 이벤트

표 48. CONTEXT 범주의 이벤트

CONNECT	SET_APPL_PRIORITY
CONNECT_RESET	RESET_DB_CFG
ATTACH	GET_DB_CFG
DETACH	GET_DFLT_CFG
DARI_START	UPDATE_DBM_CFG
DARI_STOP	SET_MONITOR
BACKUP_DB	GET_SNAPSHOT
RESTORE_DB	ESTIMATE_SNAPSHOT_SIZE
ROLLFORWARD_DB	RESET_MONITOR
OPEN_TABLESPACE_QUERY	OPEN_HISTORY_FILE
FETCH_TABLESPACE	CLOSE_HISTORY_FILE
CLOSE_TABLESPACE_QUERY	FETCH_HISTORY_FILE
OPEN_CONTAINER_QUERY	SET_RUNTIME_DEGREE
CLOSE_CONTAINER_QUERY	UPDATE_AUDIT
FETCH_CONTAINER_QUERY	DBM_CFG_OPERATION
SET_TABLESPACE_CONTAINERS	DISCOVER
GET_TABLESPACE_STATISTIC	OPEN_CURSOR
READ_ASYNC_LOG_RECORD	CLOSE_CURSOR
QUIESCE_TABLESPACE	FETCH_CURSOR
LOAD_TABLE	EXECUTE EXECUTE_IMMEDIATE
UNLOAD_TABLE	PREPARE DESCRIBE
UPDATE_RECOVERY_HISTORY	BIND
PRUNE_RECOVERY_HISTORY	REBIND
SINGLE_TABLESPACE_QUERY	RUNSTATS
LOAD_MSG_FILE	REORG
UNQUIESCE_TABLESPACE	REDISTRIBUTE
ENABLE_MULTIPAGE	COMMIT
DESCRIBE_DATABASE	ROLLBACK
DROP_DATABASE	REQUEST_ROLLBACK
CREATE_DATABASE	IMPLICIT_REBIND
ADD_NODE	EXTERNAL_CANCEL
FORCE_APPLICATION	SWITCH_USER

EXECUTE 범주의 이벤트

- COMMIT문의 COMMIT 실행
- 데이터베이스 연결의 CONNECT 설정
- 데이터베이스 연결의 CONNECT RESET 종료
- 명령문의 DATA A 호스트 변수 또는 매개변수 표시문자 데이터 값

이 이벤트는 명령문의 일부인 각 호스트 변수 또는 매개변수 표시문자에 대해 반복됩니다. 구분된 감사 로그 추출에만 제공됩니다.

- GLOBAL COMMIT: 전역 트랜잭션 내에서 COMMIT 실행
- GLOBAL ROLLBACK: 전역 트랜잭션 내에서 ROLLBACK 실행
- RELEASE SAVEPOINT: RELEASE SAVEPOINT문 실행
- ROLLBACK: ROLLBACK문 실행
- SAVEPOINT: SAVEPOINT문 실행
- STATEMENT: SQL문 실행
- SWITCH USER: 트러스트된 연결 내에서 사용자 전환

OBJMAINT 범주의 이벤트

- ALTER_OBJECT(보호되는 테이블 변경 또는 모듈 변경 시에 생성됨)
- CREATE_OBJECT
- DROP_OBJECT
- RENAME_OBJECT

SECMAINT 범주의 이벤트

- ADD_DEFAULT_ROLE
- ADD_USER
- ALTER_DEFAULT_ROLE
- ALTER SECURITY POLICY
- ALTER_USER_ADD_ROLE
- ALTER_USER_AUTHENTICATION
- ALTER_USER_DROP_ROLE
- DROP_DEFAULT_ROLE
- DROP_USER
- GRANT
- IMPLICIT_GRANT
- IMPLICIT_REVOKE
- REVOKE
- SET_SESSION_USER
- TRANSFER_OWNERSHIP
- UPDATE_DBM_CFG

SYSADMIN 범주의 이벤트

표 49. SYSADMIN 범주의 이벤트

START_DB2
STOP_DB2
CREATE_DATABASE
ALTER_DATABASE
DROP_DATABASE
UPDATE_DBM_CFG
UPDATE_DB_CFG
CREATE_TABLESPACE
DROP_TABLESPACE
ALTER_TABLESPACE
RENAME_TABLESPACE
CREATE_NODEGROUP
DROP_NODEGROUP
ALTER_NODEGROUP
CREATE_BUFFERPOOL
DROP_BUFFERPOOL
ALTER_BUFFERPOOL
CREATE_EVENT_MONITOR
DROP_EVENT_MONITOR
ENABLE_MULTIPAGE
MIGRATE_DB_DIR
DB2TRC
DB2SET
ACTIVATE_DB
ADD_NODE
BACKUP_DB
CATALOG_NODE
CATALOG_DB
CATALOG_DCS_DB
CHANGE_DB_COMMENT
DEACTIVATE_DB
DROP_NODE_VERIFY
FORCE_APPLICATION
GET_SNAPSHOT
LIST_DRDA_INDOUBT_TRANSACTIONS
MIGRATE_DB
RESET_ADMIN_CFG
RESET_DB_CFG
RESET_DBM_CFG
RESET_MONITOR
RESTORE_DB
ROLLFORWARD_DB
SET_RUNTIME_DEGREE
SET_TABLESPACE_CONTAINERS
UNCATALOG_DB
UNCATALOG_DCS_DB
UNCATALOG_NODE
UPDATE_ADMIN_CFG
UPDATE_MON_SWITCHES
LOAD_TABLE
DB2AUDIT
SET_APPL_PRIORITY
CREATE_DB_AT_NODE
KILLDBM
MIGRATE_SYSTEM_DIRECTORY
DB2REMOT
DB2AUD
MERGE_DBM_CONFIG_FILE
UPDATE_CLI_CONFIGURATION
OPEN_TABLESPACE_QUERY
SINGLE_TABLESPACE_QUERY
CLOSE_TABLESPACE_QUERY
FETCH_TABLESPACE
OPEN_CONTAINER_QUERY
FETCH_CONTAINER_QUERY
CLOSE_CONTAINER_QUERY
GET_TABLESPACE_STATISTICS
DESCRIBE_DATABASE
ESTIMATE_SNAPSHOT_SIZE
READ_ASYNC_LOG_RECORD
PRUNE_RECOVERY_HISTORY
UPDATE_RECOVERY_HISTORY
QUIESCE_TABLESPACE
UNLOAD_TABLE
UPDATE_DATABASE_VERSION
CREATE_INSTANCE
DELETE_INSTANCE
SET_EVENT_MONITOR
GRANT_DBADM(V97:더 이상 생성되지 않음)
REVOKE_DBADM(V97:더 이상 생성되지 않음)
GRANT_DB_AUTH(V97:더 이상 생성되지 않음)
REVOKE_DB_AUTH(V97:더 이상 생성되지 않음)
REDISTRIBUTE_NODEGROUP

VALIDATE 범주의 이벤트

- AUTHENTICATE
- CHECK_GROUP_MEMBERSHIP(버전 9.5 이상에서 생성되지 않음)
- GET_USERMAPPING_FROM_PLUGIN
- GET_GROUPS(버전 9.5 이상에서 생성되지 않음)
- GET_USERID(버전 9.5 이상에서 생성되지 않음)

제 10 장 운영 체제 보안 작업

운영 체제는 데이터베이스 설치에 보안을 지원하는 데 사용할 수 있는 보안 기능을 제공합니다.

DB2 및 Windows 보안

Windows 도메인은 특정 및 고유 이름으로 참조되는 클라이언트 및 서버 컴퓨터의 배열로서 SAM(Security Access Manager)이라고 하는 단일 사용자 어카운트 데이터베이스를 공유합니다. 도메인에 있는 컴퓨터 중 하나가 도메인 제어기입니다. 도메인 제어기는 사용자와 도메인 간 상호작용의 모든 측면을 관리합니다.

도메인 제어기는 도메인 사용자 어카운트 데이터베이스에 있는 정보를 사용하여 도메인 어카운트에 로그온하는 사용자를 인증합니다. 각 도메인에서 하나의 도메인 제어기가 기본 도메인 제어기(PDC)입니다. 도메인에는 기본 도메인 제어기가 없거나 기본 도메인 제어기가 사용 불가능한 경우에 사용자 어카운트를 인증하는 백업 도메인 제어기(BDC)가 있을 수도 있습니다. 백업 도메인 제어기에는 PDC에 있는 마스터 사본과 정기적으로 동기화되는 Windows SAM(Security Account Manager) 데이터베이스의 사본이 있습니다.

도메인 자원에 액세스하려면 기본 도메인 제어기에서 사용자 어카운트, 사용자 ID 및 암호를 정의하기만 하면 됩니다.

주: 두-파트 사용자 ID는 CONNECT문 및 ATTACH문으로 지원됩니다. SAM 호환 사용자 ID의 규정자는 최대 길이가 15자인 'Domain₩User' 스타일의 이름입니다.

Windows 서버가 설치되었을 때 설치 프로세스 동안 다음을 작성하도록 선택할 수 있습니다.

- 새 도메인에 기본 도메인 제어기 작성
- 알려진 도메인에 백업 도메인 제어기 작성
- 알려진 도메인에 독립형 서버 작성

새 도메인의 『제어기』를 선택하면 해당 서버가 기본 도메인 제어기가 됩니다.

사용자는 로컬 머신에 로그온할 수 있거나, 머신이 Windows 도메인에 설치된 경우 사용자는 도메인에 로그온할 수 있습니다. 사용자를 인증하기 위해 DB2는 먼저 로컬 머신을 검사한 후 현재 도메인에 대한 도메인 제어기를 검사한 후 마지막으로 도메인 제어기에 알려진 트러스트된 도메인을 검사합니다.

이 작업 방법을 보여주기 위해 DB2 인스턴스에 서버 인증이 필요하다고 가정합시다. 구성은 다음과 같습니다.

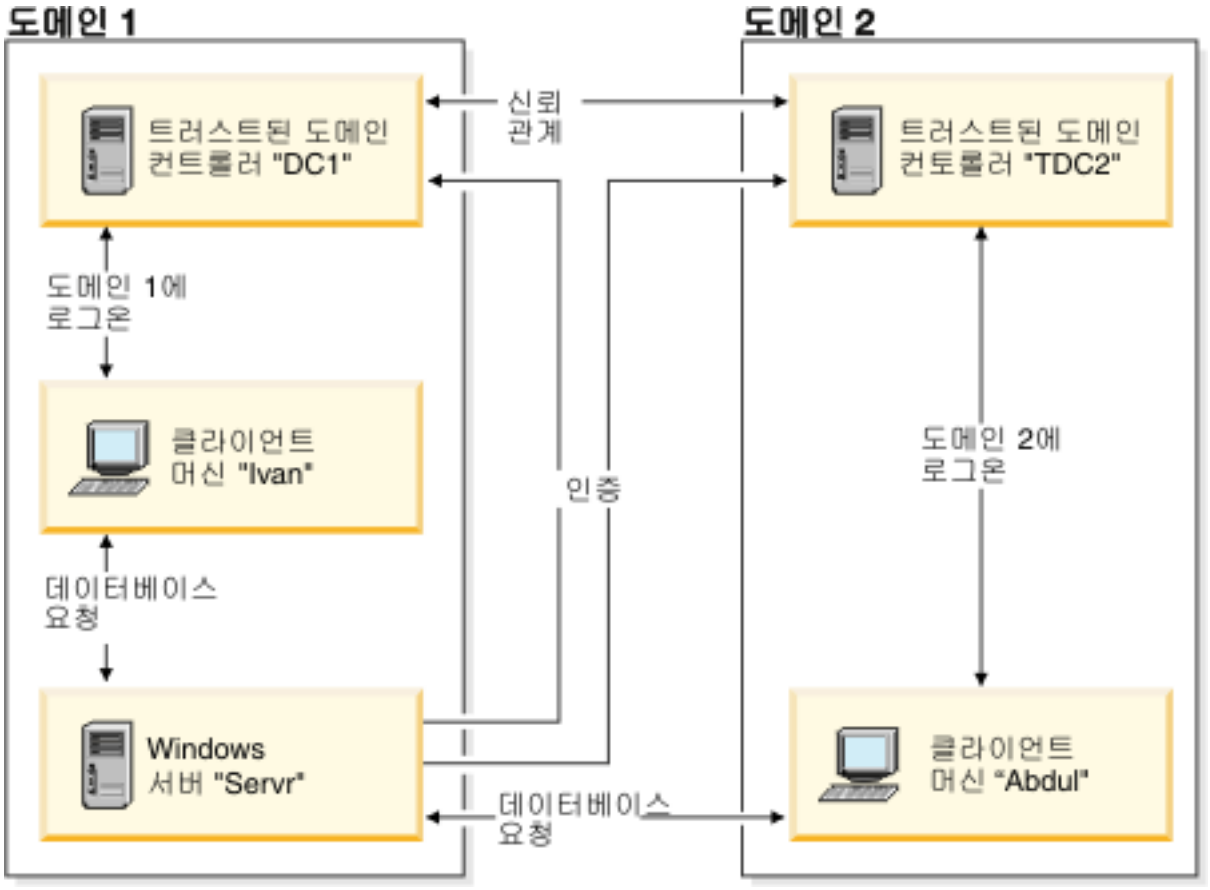


그림 7. Windows 도메인을 사용한 인증

각 머신에는 보안 데이터베이스인 SAM(Security Access Management)이 있습니다. DC1은 클라이언트 머신 Ivan과, DB2 서버 Servr이 등록되는 도메인 제어기입니다. TDC2는 DC1과 클라이언트 머신인 Abdul용 트러스트된 도메인으로, TDC2 도메인의 구성원입니다.

인증 시나리오

서버 인증이 포함된 시나리오(Windows)

다음 예에서는 서버에 의한 사용자 인증을 보여줍니다.

1. Abdul은 TDC2 도메인에 로그온합니다(즉, TDC2 SAM 데이터베이스에 알려집니다).
2. 그런 다음, Abdul을 SRV3에 상주하기 위해 카탈로그에 등록되는 DB2 데이터베이스에 연결합니다.

```
db2 connect to remotedb user Abdul using fredpw
```

3. SRV3은 Abdul이 알려진 곳을 판별합니다. 이 정보를 찾기 위해 사용되는 API는 신뢰성 있는 도메인을 시도하기 전에 먼저 로컬 머신(SRV3)을 검색한 후 도메인 제어기(DC1)를 검색합니다. 사용자 이름 Abdul이 TDC2에서 발견됩니다. 이 검색 순서에서는 사용자 또는 그룹에 대해 하나의 이름 공간이 필요합니다.
4. 그런 다음, SRV3은 다음을 수행합니다.
   a. TDC2에 대해 사용자 이름과 암호의 유효성을 확인합니다.
   b. TDC2에 요청하여 Abdul이 관리자인지 찾습니다.
   c. TDC2에 요청하여 Abdul의 모든 그룹을 열거합니다.

클라이언트 인증 및 Windows 클라이언트 컴퓨터가 포함된 시나리오

다음 예에서는 클라이언트 컴퓨터에 의한 사용자 인증을 보여줍니다.

1. 관리자인 Dale은 SRV3에 로그온하여 데이터베이스 인스턴스에 대한 인증을 클라이언트로 변경합니다.

   ```
   db2 update dbm cfg using authentication client
   db2stop
   db2start
   ```

2. Windows 클라이언트 머신에 있는 Ivan이 DC1 도메인에 로그온합니다(즉 그는 DC1 SAM 데이터베이스에 알려져 있습니다).
3. 그런 다음, Ivan을 SRV3에 상주시키기 위해 카탈로그에 등록되는 DB2 데이터베이스에 연결합니다.

   ```
   DB2 CONNECT to remotedb user Ivan using johnpw
   ```

4. Ivan의 머신은 사용자 이름과 암호의 유효성을 확인합니다. 이 정보를 찾기 위해 사용되는 API는 신뢰성 있는 도메인을 시도하기 전에 먼저 로컬 머신(Ivan)을 검색한 후 도메인 제어기(DC1)를 검색합니다. 사용자 이름 Ivan이 DC1에서 발견됩니다.
5. 그러면 Ivan의 머신은 DC1에 대해 사용자 이름과 암호의 유효성을 확인합니다.
6. 그런 다음, SRV3은 다음을 수행합니다.
   a. Ivan이 알려진 곳을 판별합니다.
   b. DC1에 요청하여 Ivan이 관리자인지 찾습니다.
   c. DC1에 요청하여 Ivan의 모든 그룹을 나열합니다.

주: DB2 데이터베이스에 연결을 시도하기 전에, DB2 보안 서비스가 시작되었는지 확인하십시오. 보안 서비스는 Windows 설치의 일부로서 설치됩니다. 그런 다음 DB2가 설치되고 Windows 서비스로 『등록』되나, 자동으로 시작되지는 않습니다. DB2 보안 서비스를 시작하려면, `NET START DB2NTSECSERVER` 명령을 입력하십시오.

전역 그룹에 대한 지원(Windows)

DB2 데이터베이스 시스템에서는 전역 그룹을 지원합니다.

전역 그룹을 사용하려면 로컬 그룹 내에 전역 그룹이 포함되어 있어야 합니다. DB2 데이터베이스 관리 프로그램에서 사용자가 구성원으로 속해 있는 모든 그룹을 나열하면 사용자가 간접적으로 속해 있는 로컬 그룹도 나열됩니다(하나 이상의 로컬 그룹의 구성원인 전역 그룹의 효력으로).

전역 그룹은 두 가지 상황에서 사용됩니다.

- 로컬 그룹에 포함. 로컬 그룹에 사용 권한을 부여해야 합니다.
- 도메인 제어기에 포함. 전역 그룹에 사용 권한을 부여해야 합니다.

Windows의 DB2에 대한 사용자 인증 및 그룹 정보

사용자 이름 및 그룹 이름 제한사항(Windows)

Windows 환경과 관련하여 몇 가지 제한사항이 있습니다. 일반 DB2 오브젝트 이름 지정 규칙도 적용된다는 사실에 유의하십시오.

- Windows의 사용자 이름은 대소문자를 구분하지 않습니다. 그러나 암호는 대소문자를 구분합니다.
- 사용자 이름 및 그룹 이름은 대문자 및 소문자의 조합일 수 있습니다. 그러나 DB2 데이터베이스에서 사용될 때 보통 대문자로 변환됩니다. 예를 들어, 데이터베이스에 연결하고 schema1.table1 테이블을 작성하는 경우, 이 테이블은 데이터베이스에서 SCHEMA1.TABLE1로서 저장됩니다. 소문자 오브젝트 이름을 사용하려는 경우, 오브젝트 이름을 따옴표로 묶어 명령행 처리기에서 명령을 발행하거나, 써드 파티 ODBC 프론트엔드 도구를 사용하십시오.
- DB2 데이터베이스 관리 프로그램은 단일 이름 스페이스를 지원합니다. 즉 트러스트된 도메인 환경에서 실행 중일 때, 다중 도메인에 있거나 서버 머신의 로컬 SAM 및 다른 도메인에 있는 것과 동일한 이름의 사용자 어카운트를 가질 수 없습니다.
- 사용자 이름은 그룹 이름과 달라야 합니다.
- 로컬 그룹 이름은 도메인 레벨 그룹 이름과 달라야 합니다.

Windows에서의 그룹 및 사용자 인증

Windows에서는 『사용자 관리자』라는 Windows 관리 도구를 사용하여 사용자 어카운트를 작성함으로써 사용자를 정의합니다. 다른 어카운트(구성원이라고도 함)를 포함하는 어카운트가 그룹입니다.

그룹을 통해 Windows 관리자는 각 사용자를 개별적으로 유지보수하지 않고 그룹의 사용자에게 동시에 권한 및 사용 권한을 부여할 수 있게 됩니다. 사용자 어카운트와 마찬가지로, 그룹은 SAM(Security Access Manager) 데이터베이스에서 정의되고 유지보수됩니다.

그룹 유형에는 다음 두 가지가 있습니다.

- 로컬 그룹. 로컬 그룹에는 로컬 어카운트 데이터베이스에서 작성된 사용자 어카운트가 포함될 수 있습니다. 로컬 그룹이 도메인의 일부인 머신에 있으면, 로컬 그룹에 Windows 도메인의 도메인 어카운트 및 그룹도 포함될 수 있습니다. 워크스테이션에서 로컬 그룹을 작성하면, 이 로컬 그룹은 해당 워크스테이션에 고유합니다.
- 전역 그룹. 전역 그룹은 도메인 제어기에만 있으며 해당 도메인의 SAM 데이터베이스의 사용자 어카운트가 포함됩니다. 즉, 전역 그룹에는 해당 전역 그룹이 작성된 도메인의 사용자 어카운트만 포함될 수 있으며 다른 그룹은 구성원으로서 포함될 수 없습니다. 전역 그룹은 자체 도메인의 서버 및 워크스테이션에서 사용될 수 있습니다.

Windows 도메인 간의 신뢰 관계

신뢰 관계는 두 도메인 간의 관리 및 통신 링크입니다. 두 도메인 간의 신뢰 관계는 사용자 어카운트 및 전역 그룹이 정의된 도메인이 아닌 다른 도메인에서 그러한 어카운트를 사용할 수 있도록 합니다.

인증 과정 없이 트러스트된 도메인에 있는 사용자 어카운트 및 전역 그룹의 권한 및 허용의 유효성을 확인할 수 있도록 어카운트 정보가 공유됩니다. 신뢰 관계는 둘 이상의 도메인을 단일의 관리 단위로 결합함으로써 사용자 관리를 단순화합니다.

신뢰 관계에는 두 개의 도메인이 있습니다.

- 트러스트하는 도메인. 이 도메인은 다른 도메인이 그들을 대신하여 사용자를 인증할 것임을 신뢰합니다.
- 트러스트된 도메인. 이 도메인은 다른 도메인을 대신하여(신뢰 하에) 사용자를 인증합니다.

신뢰 관계는 이행적이지 않습니다. 이는 도메인 간에 각 방향으로 명시적 신뢰 관계를 수립해야 함을 의미합니다. 예를 들어, 트러스트하는 도메인은 트러스트된 도메인이 아닐 수도 있습니다.

그룹 및 도메인 보안 인증(Windows)

DB2 데이터베이스 시스템에서 특권을 부여하거나 권한 레벨을 정의할 때 로컬 그룹 또는 전역 그룹을 지정할 수 있습니다.

사용자 어카운트가 로컬 또는 전역 그룹에 명시적으로 정의되거나 로컬 그룹의 구성원으로 정의되는 전역 그룹의 구성원이 됨으로써 내재적으로 로컬 또는 전역 그룹에 정의될 경우, 해당 사용자는 그 그룹의 구성원으로 판별됩니다.

DB2 데이터베이스 관리 프로그램은 다음 유형의 그룹을 지원합니다.

- 로컬 그룹
- 전역 그룹
- 로컬 그룹 및 전역 그룹의 구성원

DB2 데이터베이스 관리 프로그램은 사용자가 발견된 보안 데이터베이스를 사용하여 해당 사용자가 구성원인 로컬 및 전역 그룹을 열거합니다. DB2 데이터베이스 시스템은 사용자 어카운트가 있는 위치에 관계없이 DB2 데이터베이스가 설치된 로컬 Windows 서버에 그룹을 강제로 열거하도록 하는 겹쳐쓰기를 제공합니다. 이 겹쳐쓰기는 다음과 같은 명령으로 수행됩니다.

– 전역 설정의 경우

```
db2set -g DB2_GRP_LOOKUP=local
```

– 예:

```
db2set -i <instance_name> DB2_GRP_LOOKUP=local
```

이 명령을 발행한 후에는 변경사항이 적용되도록 DB2 데이터베이스 인스턴스를 중지한 후 시작해야 합니다. 그런 다음 로컬 그룹을 작성하여 해당 로컬 그룹에 도메인 어카운트 또는 전역 그룹을 포함시키십시오.

설정된 모든 DB2 프로파일 레지스트리 변수를 보려면 다음을 입력하십시오.

```
db2set -all
```

DB2_GRP_LOOKUP 프로파일 레지스트리 변수가 로컬로 설정되면, DB2 데이터베이스 관리 프로그램은 로컬 머신에서만 사용자 그룹을 열거하려고 합니다. 사용자가 로컬 그룹 또는 로컬 그룹에 중첩된 전역 그룹의 구성원으로 정의되지 않은 경우 그룹 열거가 실패합니다. DB2 데이터베이스 관리 프로그램이 도메인 또는 도메인 제어기의 다른 머신에는 사용자 그룹을 열거하지 **않습니다**.

DB2 데이터베이스 관리 프로그램이 자원 도메인의 기본 또는 백업 도메인 제어기인 머신에서 실행 중인 경우, 모든 트러스트된 도메인에 있는 모든 도메인 제어기를 찾을 수 있습니다. 그 이유는 도메인 제어기에서 실행 중일 경우에만 트러스트된 도메인에 있는 백업 도메인 제어기의 도메인 이름을 알 수 있기 때문입니다.

액세스 토큰을 사용하여 사용자 그룹 정보 얻기(Windows)

액세스 토큰은 프로세스 또는 스레드의 보안 컨텍스트를 설명하는 오브젝트입니다. 액세스 토큰의 정보에는 프로세스 또는 스레드와 연관된 사용자 어카운트의 ID 및 특권이 포함됩니다.

로그온 시, 시스템은 암호를 보안 데이터베이스에 저장된 정보와 비교하여 검증합니다. 암호가 인증될 경우, 시스템은 액세스 토큰을 생성합니다. 사용자의 이름으로 실행되는 모든 프로세스는 이 액세스 토큰의 사본을 사용합니다.

또한 액세스 토큰은 캐시된 증명서를 기반으로 확보할 수 있습니다. 사용자가 시스템에서 인증되면 운영 체제에서 증명서를 캐시합니다. 도메인 제어기에 접속할 수 없는 때 마지막 로그온의 액세스 토큰을 캐시에서 참조할 수 있습니다.

액세스 토큰에는 사용자가 속하는 모든 그룹 즉, 로컬 그룹 및 다양한 도메인 그룹(전역 그룹, 도메인 로컬 그룹 및 범용 그룹)에 관한 정보가 포함됩니다.

주: 액세스 토큰 지원이 사용 가능한 경우에도 클라이언트 인증을 사용한 그룹 찾아보기가 리모트 연결에서 지원되지 않습니다.

액세스 토큰 지원을 사용 가능하게 하려면, db2set 명령을 사용하여 DB2_GRP_LOOKUP 레지스트리 변수를 갱신해야 합니다. DB2_GRP_LOOKUP에는 두 개의 매개변수를 쉼표로 구분하여 사용할 수 있습니다.

- 첫 번째 매개변수는 기본 그룹을 찾기 위한 것으로 " ", "LOCAL", "DOMAIN" 중 하나의 값을 사용할 수 있습니다.
- 두 번째 매개변수는 토큰 스타일 그룹을 찾기 위한 것으로 "TOKEN", "TOKENDOMAIN", "TOKENLOCAL" 중 하나의 값을 사용할 수 있습니다.

두 번째 매개변수(TOKEN, TOKENDOMAIN 또는 TOKENLOCAL)가 지정된 경우 이 매개변수가 기본 그룹 열거보다 우선합니다. DB2_GRP_LOOKUP의 첫 번째 매개변수가 지정된 경우 토큰 그룹 열거에 실패하면 기본 그룹 찾아보기가 수행됩니다.

TOKEN, TOKENDOMAIN 및 TOKENLOCA 값의 의미는 다음과 같습니다.

- TOKENLOCAL

  토큰을 사용하여 로컬 시스템에 있는 그룹을 열거합니다(기본 "LOCAL" 그룹 찾아보기와 같음).

- TOKENDOMAIN

  토큰을 사용하여 사용자가 정의되어 있는 위치(로컬 사용자의 경우 로컬 시스템, 도메인 사용자의 경우 도메인)에 그룹을 나열합니다. 기본 " " 또는 "DOMAIN" 그룹 찾아보기와 같습니다.

- TOKEN

  토큰을 사용하여 도메인과 로컬 시스템 둘 다에 있는 그룹을 열거합니다. 로컬 사용자의 경우 리턴된 그룹에 로컬 그룹이 포함되고, 도메인 사용자의 경우 리턴된 그룹에 도메인과 로컬 그룹이 둘 다 포함됩니다. 토큰 매개변수에 기본 그룹 찾아보기 매개변수 값 "LOCAL" 또는 "DOMAIN"에 해당하는 사항이 없습니다.

예를 들어, 다음 DB2_GRP_LOOKUP 설정을 사용하면 액세스 토큰이 로컬 그룹 열거를 지원합니다.

```
db2set DB2_GRP_LOOKUP=LOCAL,TOKENLOCAL
```

다음 예에서는 사용자 ID가 정의되어 있는 위치(도메인에서 어카운트가 정의된 경우)와 로컬 시스템에서 그룹 열거가 지원됩니다.

```
db2set DB2_GRP_LOOKUP=,TOKEN
```

마지막 예에서는 액세스 토큰이 사용자가 정의되어 있는 위치에서의 도메인 그룹 열거를 지원합니다.

```
db2set DB2_GRP_LOOKUP=DOMAIN,TOKENDOMAIN
```

주: 액세스 토큰 지원은 CLIENT 인증을 제외한 모든 인증 유형에서 사용할 수 있습니다.

DB2_GRP_LOOKUP 환경 변수 및 DB2 그룹 열거(Windows)

Windows에서, 한 명의 사용자가 도메인 레벨에서 정의된 그룹, 로컬 시스템에 정의된 그룹 또는 두 그룹 모두에 속할 수 있습니다.

DB2_GRP_LOOKUP 환경 변수는 로컬 시스템에 그룹을 열거할지, 또는 사용자(로컬 사용자일 경우 로컬 시스템에 또는 도메인 사용자일 경우 도메인 레벨에서)를 정의할지를 제어합니다. 따라서 보안 관리자가 권한 및 특권을 부여할 때, **DB2_GRP_LOOKUP**이 의도대로 설정되었고 의도된 인증이 올바른 사용자에게 부여되었는지 확인해야 합니다.

DB2_GRP_LOOKUP 프로파일 레지스트리 변수가 설정되지 않은 경우 다음이 수행됩니다.

1. DB2 데이터베이스 시스템은 먼저 동일한 머신에서 사용자를 찾으려고 합니다.
2. 사용자 이름이 로컬에 정의된 경우, 사용자는 로컬에서 인증됩니다.
3. 로컬로 사용자를 찾지 못하는 경우 DB2 데이터베이스 시스템은 도메인과 트러스트된 도메인에서 차례로 사용자 이름을 찾으려고 시도합니다.

예를 들어, **DB2_GRP_LOOKUP**가 설정되지 않은 다음과 같이 상황을 가정해 봅시다.

1. 도메인 사용자 DUSER1이 로컬 그룹 GROUP1의 구성원입니다.
2. SECADM 권한을 가지고 있는 보안 관리자가 GROUP1 그룹에 DBADM 권한을 부여합니다.

   ```
   GRANT DBADM ON database TO GROUP GROUP1
   ```

3. **DB2_GRP_LOOKUP**이 설정되어 있지 않으므로 사용자가 정의된 시스템에 그룹이 열거됩니다. 따라서 DUSER1의 그룹은 도메인 레벨에서 열거됩니다. DUSER1은 도메인 레벨에서 열거된 GROUP1 그룹에 속하지 않으므로, DUSER1은 DBADM 권한을 부여받지 않습니다.

계속해서 **DB2_GRP_LOOKUP**이 설정되어 있지 않고 UPGRADE DATABASE 명령과 관련된 보다 복잡한 시나리오를 가정해 봅시다.

1. 도메인 사용자 DUSER2이 로컬 관리자 그룹의 구성원입니다.
2. **sysadm_group** 구성 매개변수가 설정되어 있지 않으므로, 로컬 관리자 그룹의 구성원은 자동으로 SYSADM 권한을 갖게 됩니다.

3. DUSER2 사용자는 SYSADM 권한을 가지므로 UPGRADE DATABASE 명령을 발행할 수 있습니다. UPGRADE DATABASE 명령은 SYSADM 그룹(이 경우, 관리자 그룹)으로 업그레이드되는 데이터베이스에서 DBADM 권한을 부여합니다.
4. **DB2_GRP_LOOKUP**이 설정되어 있지 않으므로 사용자가 정의된 시스템에 그룹이 열거됩니다. 따라서 DUSER2의 그룹은 도메인 레벨에서 열거됩니다. DUSER2는 도메인 레벨에서 열거된 관리자 그룹에 속하지 않으므로, DUSER2는 DBADM 권한을 부여받지 않습니다.

이 시나리오에 사용 가능한 솔루션은 다음 중 하나로 변경하는 것입니다.

- **DB2_GRP_LOOKUP** = local로 설정하십시오.
- DBADM 권한을 가지고 있는 사용자를 도메인 제어기에 있는 관리자 또는 GROUP1 그룹에 추가하십시오.

다음 DUSER1의 예제에 표시된 대로
SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID 테이블 함수를 사용하여 사용자가 가지고 있는 권한을 확인할 수 있습니다.

```
SELECT AUTHORITY, D_USER, D_GROUP, D_PUBLIC, ROLE_USER, ROLE_GROUP, ROLE_PUBLIC, D_ROLE
   FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('DUSER1', 'U') ) AS T
   ORDER BY AUTHORITY
```

다음 DUSER1의 예제에 표시된 대로
SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID 테이블 함수를 사용하여 DB2 데이터베이스 관리 프로그램이 사용자가 속해 있다고 판별한 그룹을 확인할 수 있습니다.

```
SELECT * FROM TABLE (SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID ('DUSER1')) AS T
```

주: 도메인 레벨과 로컬 시스템에 동일한 그룹 이름을 사용하면 DB2 데이터베이스 관리 프로그램이 해당 그룹을 완벽하게 확인할 수 없기 때문에 혼동될 수 있습니다.

순서 지정된 도메인 목록을 사용하여 인증

트러스트된 도메인 포리스트(Forest)에 사용자 ID를 두 번 이상 정의할 수 있습니다. 트러스트된 도메인 포리스트(Forest)는 네트워크를 통해 상호 관련된 도메인 콜렉션입니다.

한 도메인의 사용자가 다른 도메인의 또다른 사용자와 동일한 사용자 ID를 사용할 수 있습니다. 이 경우 다음과 같은 작업을 수행하고자 할 때 문제점이 발생할 수도 있습니다.

- 동일한 사용자 ID를 갖지만 서로 다른 도메인에 위치하는 다중 사용자 인증
- 그룹을 기반으로 특권을 부여하거나 권한 취소하기 위한 그룹 찾아보기
- 암호 유효성 확인
- 네트워크 트래픽 제어

도메인 포리스트(forest)에서 다중 사용자가 동일한 사용자 ID를 가질 가능성이 발생하는 문제점을 방지하려면, db2set 및 레지스트리 변수 DB2DOMAINLIST를 사용하여 정의된 대로 순서 지정된 도메인 목록을 사용해야 합니다. 순서를 설정할 경우, 목록에 포함시킬 도메인을 쉼표로 분리해야 합니다. 사용자 인증 시 도메인 검색 순서에 관하여 신중한 의사결정을 수행해야 합니다.

도메인 목록에서 훨씬 아래쪽에 있는 도메인에 표시된 사용자 ID는 도메인 액세스를 위한 인증을 받고자 할 경우 이름을 변경해야 합니다.

액세스 제어는 도메인 목록을 통해 수행할 수 있습니다. 예를 들어, 사용자의 도메인이 목록에 없을 경우, 사용자는 연결을 수행할 수 없습니다.

주: DB2DOMAINLIST 레지스트리 변수는 CLIENT 인증이 데이터베이스 관리 프로그램 구성에서 설정된 경우에만 유효하며 Windows 도메인 환경에서 Windows 데스크탑으로부터의 단일 사인온이 필요한 경우 사용됩니다.DB2DOMAINLIST는 Windows 환경에 클라이언트나 서버가 없는 경우 강제 실행된 일부 버전의 DB2 서버에서 지원됩니다.

도메인 보안 지원(Windows)

다음 예는 DB2 데이터베이스 관리 시스템이 Windows 도메인 보안을 지원하는 방법을 보여줍니다. 사용자 이름 및 로컬 그룹이 동일한 도메인에 있으므로 연결이 작동합니다.

다음 시나리오에서는 사용자 이름 및 로컬 또는 전역 그룹이 동일한 도메인에 있으므로 연결이 작동합니다.

사용자 이름 및 로컬 또는 전역 그룹은 데이터베이스 서버가 실행 중이지만 서로 동일한 도메인에 있어야 하는 도메인에 정의될 필요가 없음을 기억하십시오.

표 50. 도메인 제어기를 사용한 성공적 연결

Domain1	**Domain2**
Domain2와의 신뢰 관계가 있습니다.	• Domain1과의 신뢰 관계가 있습니다. • 로컬 또는 전역 그룹 `grp2`가 정의됩니다. • 사용자 이름 `id2`가 정의됩니다. • 사용자 이름 `id2`는 `grp2`의 일부입니다.

DB2 서버는 이 도메인에서 실행합니다. 여기에서 다음 DB2 명령이 발행됩니다.
`REVOKE CONNECT ON db FROM public`
`GRANT CONNECT ON db TO GROUP grp2`
`CONNECT TO db USER id2`	

로컬 또는 전역 도메인이 스캔되지만 `id2`가 없습니다. 도메인 보안이 스캔됩니다.	

표 50. 도메인 제어기를 사용한 성공적 연결 (계속)

Domain1	Domain2
	사용자 이름 id2가 이 도메인에 있습니다. DB2는 이 사용자 이름에 대한 추가 정보를 얻습니다(즉, 이 사용자 이름은 grp2의 일부입니다).
사용자 이름 및 로컬 또는 전역 그룹은 동일한 도메인에 있으므로 연결이 작동합니다.	

SYSADM 권한을 보유하는 사용자 정의(Windows)

sysadm_group 데이터베이스 관리 프로그램 구성 매개변수가 설정되지 않은 경우(즉, NULL인 경우) 특정 사용자에게 SYSADM 권한이 있습니다.

이러한 사용자는 다음과 같습니다.

- 로컬 관리자 그룹의 구성원
- DB2 데이터베이스 관리 프로그램이 사용자가 정의된 위치에서 사용자의 그룹을 열거하도록 구성되어 있는 경우 도메인 제어기에서 관리자 그룹의 구성원 (**DB2_GRP_LOOKUP** 환경 변수를 사용하여 그룹 열거를 구성할 수 있음)
- Windows 확장 보안을 사용할 경우 DB2ADMNS 그룹의 구성원. DB2ADMNS 그룹의 위치는 설치 중 결정됩니다.
- LocalSystem 어카운트

위의 디폴트 동작이 권장되지 않는 경우가 있습니다. **sysadm_group** 데이터베이스 관리 프로그램 구성 매개변수를 사용하면 다음 중 한 방법으로 이 동작을 대체할 수 있습니다.

- DB2 서버 머신에서 로컬 그룹을 작성하고 SYSADM 권한을 가질 사용자(도메인 사용자 또는 로컬 사용자)에게 추가하십시오. 로컬 시스템에 있는 사용자의 그룹을 열거하도록 DB2 데이터베이스 관리 프로그램을 구성해야 합니다.
- 도메인 그룹을 작성하고 SYSADM 권한을 가질 사용자에게 추가하십시오. 사용자가 정의된 위치에서 사용자의 그룹을 열거하도록 DB2 데이터베이스 관리 프로그램을 구성해야 합니다.

다음 명령을 사용하여 **sysadm_group** 데이터베이스 관리 프로그램 구성 매개변수를 이 그룹으로 갱신하십시오.

```
db2stop
DB2 UPDATE DBM CFG USING SYSADM_GROUP group_name
db2start
```

Windows LocalSystem 어카운트 지원

Windows 플랫폼에서는 DB2 데이터베이스 시스템이 로컬 내재된 연결을 사용하는 LocalSystem 어카운트의 컨텍스트에서 실행되는 응용 프로그램을 지원합니다. LocalSystem 어카운트에 대한 권한 부여 ID는 SYSTEM입니다.

sysadm_group 데이터베이스 관리 프로그램 구성 매개변수가 NULL로 설정된 경우, LocalSystem 어카운트가 시스템 관리자(SYSADM 권한 보유)로 간주됩니다.

LocalSystem 어카운트의 컨텍스트에서 실행되는 응용프로그램이 SYSADM 범위에 해당하지 않는 데이터베이스 조치를 수행해야 하는 경우, 필요한 특권 또는 권한을 LocalSystem 어카운트에 부여해야 합니다. 예를 들어, 응용프로그램에 데이터베이스 관리자 기능이 필요할 경우 GRANT(데이터베이스 권한)문을 사용하여 LocalSystem 어카운트에 DBADM 권한을 부여하십시오.

이 어카운트에서 실행될 응용프로그램을 작성 중인 개발자는 DB2 데이터베이스 시스템에 『SYS』로 시작하는 스키마 이름을 사용하는 오브젝트에 대한 제한사항이 있다는 점에 주의해야 합니다. 따라서 응용프로그램에 DB2 데이터베이스 오브젝트를 작성하는 DDL문이 포함된 경우, 다음과 같이 작성해야 합니다.

- 정적 쿼리의 경우 디폴트값(SYSTEM)이 아닌 QUALIFIER 옵션에 대한 값으로 바운드되어야 합니다.
- 동적 쿼리의 경우, DB2 데이터베이스 관리 프로그램에서 지원하는 스키마 이름으로 작성될 오브젝트를 명시적으로 규정하거나, DB2데이터베이스 관리 프로그램에서 지원하는 스키마 이름으로 CURRENT SCHEMA 레지스터를 설정해야 합니다.

LocalSystem 어카운트에 대한 그룹 정보는 DB2 데이터베이스 인스턴스가 시작된 후 처음으로 그룹 찾아보기가 요청될 때 수집되며 인스턴스가 재시작될 때 새로 고쳐집니다.

DB2ADMNS 및 DB2USERS 그룹을 사용하는 확장된 Windows 보안

Windows 운영 체제에 설치된 모든 DB2 데이터베이스 제품에서는 디폴트로 확장된 보안을 사용할 수 있습니다. 단, IBM Data Server Runtime Client 및 DB2 Drivers는 예외입니다. IBM Data Server Runtime Client 및 DB2 Drivers는 Windows 플랫폼에서 확장된 보안을 지원하지 않습니다.

DB2 데이터베이스 제품을 설치하면 **DB2 오브젝트에 운영 체제 보안 사용** 패널에 **운영 체제 보안 사용** 선택란이 나타납니다. 이 옵션을 사용 안함으로 설정하지 않는 한, 설치 프로그램에서는 DB2ADMNS 및 DB2USERS라는 두 개의 새 그룹을 작성합니다. DB2ADMNS 및 DB2USERS는 디폴트 그룹 이름입니다. 선택적으로 설치 시 이 그룹에 대해 다른 이름을 지정할 수 있습니다. 자동 설치를 선택한 경우 이 이름을 설치 응답 파일에서 변경할 수 있습니다. 시스템에 이미 존재하는 그룹을 사용하려고 선택한 경우, 이 그룹의 특권이 수정됨을 주의하십시오. 필요에 따라 아래 테이블에 표시된 특권이 그룹에 주어집니다. 이 그룹은 *운영 체제* 레벨에서 보호를 위해 사용되고, SYSADM, SYSMAINT 및 SYSCTRL와 같은 DB2 권한 레벨과 연결될 수 없습니다. 그러나 디폴트 관리자의 그룹을 사용하는 대신 데이터베이스 관리자는 하나 또는 모든 DB2 권한 레벨에서 설치 프로그램 또는 관리자 프로그램에 관계없이 DB2ADMNS

그룹을 사용할 수 있습니다. SYSADM 그룹을 지정할 경우 DB2ADMNS 그룹을 사용할 것을 권장합니다. 이는 설치 시 또는 그 이후에 관리자가 지정할 수 있습니다.

주: DB2 관리자 그룹(DB2ADMNS 또는 설치 중 선택한 이름)과 DB2 사용자 그룹(DB2USERS 또는 설치 중 선택한 이름)을 로컬 그룹 또는 도메인 그룹으로 지정할 수 있습니다. 두 그룹은 동일한 유형이어야 하므로 둘 다 로컬이든지 둘 다 도메인이어야 합니다.

컴퓨터 이름을 변경하여 컴퓨터 그룹 DB2ADMNS 및 DB2USERS가 로컬 컴퓨터 그룹일 경우, DB2_ADMINGROUP and DB2_USERSGROUP 전역 레지스트리를 갱신해야 합니다. 컴퓨터 이름을 변경하고 컴퓨터를 재시작한 후 레지스트리 변수를 갱신하려면 다음 명령을 실행하십시오.

1. 명령 프롬프트를 여십시오.
2. db2extsec 명령을 실행하여 보안 설정을 갱신하십시오.

   ```
   db2extsec -a new computer name₩DB2ADMNS -u new computer name₩DB2USERS
   ```

주: Windows Vista에 설치된 DB2 데이터베이스 제품에서 확장된 보안을 사용할 수 있는 경우, DB2ADMNS 그룹에 속한 사용자만 그래픽 DB2 관리 도구를 실행할 수 있습니다. 또한 DB2ADMNS 그룹의 구성원은 전체 관리자 특권으로 도구를 시작해야 합니다. 단축키를 마우스 오른쪽 단추로 누른 다음 "관리자로 실행"을 선택하면 전체 관리자 특권으로 시작됩니다.

DB2ADMNS 및 DB2USERS 그룹을 통해 획득한 권한은

DB2ADMNS 및 DB2USERS 그룹은 구성원에게 다음 기능을 제공합니다.

- DB2ADMNS

  모든 DB2 오브젝트에 대한 전체 제어(아래의 보호 오브젝트 목록 참조)

- DB2USERS

  데이터베이스 시스템 디렉토리 아래의 설치 및 오브젝트에 위치한 모든 DB2 오브젝트에 대한 읽기 및 실행 액세스입니다. 그러나 데이터베이스 시스템 디렉토리 아래의 오브젝트에는 액세스하지 않으며 IPC 자원에는 제한된 액세스만 가능합니다.

  특정 오브젝트의 경우 쓰기 특권, 파일 추가 또는 갱신 특권 등과 같이 필요에 따라 추가 특권이 사용 가능할 수 있습니다. 이 그룹의 구성원은 데이터베이스 시스템 디렉토리 아래의 오브젝트에 대한 액세스가 없습니다.

  주: 액세스를 실행한다는 의미는 오브젝트 종속입니다. 예를 들어 **.dll** 또는 **.exe** 파일에 대해서는 액세스 실행이란 파일을 실행하는 권한이 있음을 의미하지만 디렉토리에 대해서는 디렉토리를 통과할 수 있는 권한이 있음을 의미합니다.

모든 DB2 관리자가 DB2ADMNS 그룹의 구성원이면서 동시에 로컬 관리자 그룹의 구성원인 것이 가장 이상적이나, 그러나 반드시 그래야 하는 것은 아닙니다. DB2 데이터베이스 시스템에 대한 액세스가 필요한 다른 모든 사용자는 DB2USERS 그룹의 *구성원이어야*합니다. 이들 그룹 중 하나에 사용자를 추가하려면 다음을 수행하십시오.

1. 사용자 및 암호 관리 도구를 실행하십시오.
2. 목록에서 추가할 사용자 이름을 선택하십시오.
3. 등록 정보를 누르십시오. 등록 정보 창에서 그룹 구성원 탭을 누르십시오.
4. 라디오 단추를 선택하십시오.
5. 드롭다운 목록에서 해당 그룹을 선택하십시오.

설치 후 확장된 보안 추가(db2extsec command)

확장된 보안을 사용할 수 없는 상태로 DB2 데이터베이스 시스템이 설치된 경우, **db2extsec** 명령을 실행하여 확장된 보안이 사용 가능하도록 설정할 수 있습니다. **db2extsec**명령을 실행하려면 로컬 관리자 그룹의 구성원이어야 보호된 오브젝트의 ACL를 수정할 수 있는 권한이 있습니다.

필요한 경우 **db2extsec** 명령을 여러 번 사용할 수 있으나 이 명령이 완료되면 확장된 보안을 사용 안할 수 없습니다. 이를 사용 불가능하게 설정하려면, **db2extsec**을 실행할 *때마다* 즉시 **db2extsec –r** 명령을 발행해야 합니다.

확장된 보안 제거

주의:
반드시 필요한 경우가 아니면, 확장된 보안을 사용 가능하게 설정한 후 확장된 보안을 제거하지 마십시오.

db2extsec -r 명령을 실행하여 확장된 보안을 제거할 수 있지만, 이는 확장된 보안을 사용 가능하게 한 후 다른 데이터베이스 조작(예: 데이터베이스 작성, 새 인스턴스 작성, 테이블 스페이스 추가 등)이 수행되지 않은 경우에만 가능합니다. 따라서 확장된 보안 옵션을 제거하는 가장 안전한 방법은 DB2 데이터베이스를 제거하여 관련된 모든 DB2 디렉토리(데이터베이스 디렉토리 포함)를 삭제한 다음 DB2 데이터베이스를 확장된 보안 사용없이 다시 설치합니다.

보호된 오브젝트

DB2ADMNS 및 DB2USERS 그룹을 사용하여 보호할 수 있는 *정적* 오브젝트

- 파일 시스템
  - 파일
  - 디렉토리
- 서비스

- 레지스트리 키

DB2ADMNS 및 DB2USERS 그룹을 사용하여 보호할 수 있는 동적 오브젝트는 다음과 같습니다.

- IPC 자원으로 다음을 포함합니다.
  - 파이프
  - 세마포어
  - 이벤트
- 공유 메모리

DB2ADMNS 및 DB2USERS 그룹이 소유한 특권

DB2ADMNS 및 DB2USERS 그룹에 지정된 특권을 다음 표에서 표시합니다.

표 51. DB2ADMNS 및 DB2USERS 그룹의 특권

특권	DB2ADMNS	DB2USERS	이유
토큰 오브젝트 작성 (SeCreateTokenPrivilege)	Y	N	토큰 조작(특정 토큰 조작에 필요하며 인증 및 권한 부여에 사용)
프로세스 레벨 토큰 교체 (SeAssignPrimaryTokenPrivilege)	Y	N	다른 사용자로 프로세스 작성
할당량 증가(SeIncreaseQuotaPrivilege)	Y	N	다른 사용자로 프로세스 작성
운영 체제의 일부로 작동(SeTcbPrivilege)	Y	N	LogonUser(Windows XP 이전에서 인증할 목적으로 LogonUser API를 실행하기 위해 필요)
보안 감사 생성(SeSecurityPrivilege)	Y	N	감사 및 보안 로그 조작
파일 또는 기타 오브젝트의 소유권 획득 (SeTakeOwnershipPrivilege)	Y	N	수정 오브젝트 ACL
스케줄 우선순위 증가 (SeIncreaseBasePriorityPrivilege)	Y	N	프로세스 작업 세트 수정
백업 파일 및 디렉토리 (SeBackupPrivilege)	Y	N	프로파일/레지스트리 조정(특정 사용 프로파일 및 레지스트리 조정 루틴: LoadUserProfile, RegSaveKey(Ex), RegRestoreKey, RegReplaceKey, RegLoadKey(Ex))
리스토어 파일 및 디렉토리 (SeRestorePrivilege)	Y	N	프로파일/레지스트리 조정(특정 사용 프로파일 및 레지스트리 조정 루틴: LoadUserProfile, RegSaveKey(Ex), RegRestoreKey, RegReplaceKey, RegLoadKey(Ex))
디버그 프로그램(SeDebugPrivilege)	Y	N	토큰 조작(특정 토큰 조작에 필요하며 인증 및 권한 부여에 사용)
감사 및 보안 로그 관리 (SeAuditPrivilege)	Y	N	감사 로그 항목 생성
서비스로 로그온(SeServiceLogonRight)	Y	N	서비스로 DB2 실행
네트워크로에서 이 컴퓨터에 액세스 (SeNetworkLogonRight)	Y	Y	네트워크 증명서 허용(DB2 데이터베이스 관리 프로그램이 LOGON32_LOGON_NETWORK 옵션을 사용하여 성능이 포함된 인증을 하도록 허용)

표 51. DB2ADMNS 및 DB2USERS 그룹의 특권 (계속)

특권	DB2ADMNS	DB2USERS	이유
인증 후 클라이언트 가장 (SeImpersonatePrivilege)	Y	N	클라이언트 가장(Windows에서 특정 API를 사용하여 DB2 클라이언트를 가장하는 데 필요함 :ImpersonateLoggedOnUser, ImpersonateSelf, RevertToSelf 등)
메모리의 잠금 페이지 (SeLockMemoryPrivilege)	Y	N	대형 페이지 지원
전역 오브젝트 작성 (SeCreateGlobalPrivilege)	Y	Y	터미널 서버 지원(Windows에 필수)

Vista에 대한 고려사항: 사용자 액세스 제어 기능

Windows Vista의 사용자 액세스 제어(UAC) 기능은 다음과 같은 방식으로 DB2 데이터베이스 시스템에 영향을 줍니다.

전체 관리 특권을 사용하여 응용프로그램 시작

Vista에서는 사용자가 로컬 관리자일지라도 디폴트로 표준 사용자 권한을 사용하여 응용프로그램이 시작됩니다. 추가 특권을 사용하여 응용프로그램을 시작하려면 전체 관리 특권으로 실행 중인 명령 창에서 명령을 실행해야 합니다. DB2 설치 프로세스를 실행하면 Vista 사용자를 위한 "명령 창 - 관리자"라는 단축키가 작성됩니다. 이 단축키는 관리 명령을 실행하려는 경우에 실행하는 것이 좋습니다.

전체 관리 특권을 보유하지 않은 사용자가 Windows Vista의 명령 프롬프트나 그래픽 도구에서 DB2 관리 태스크를 수행하려고 한 경우, 액세스가 거부되어 태스크를 성공적으로 완료할 수 없다는 내용의 여러 오류 메시지가 표시됩니다.

수행 중인 조치를 관리 태스크로 간주되는지 여부를 판별하려면 다음 사항 중 하나라도 해당하는지 확인하십시오.

- SYSADM, SYSCTRL 또는 SYSMAINT 권한이 필요합니다.
- 레지스트리의 HKLM 분기 아래에 있는 레지스트리 키를 수정합니다.
- `Program Files` 디렉토리의 하위 디렉토리에 기록합니다.

예를 들어, 다음 조치가 모두 관리 태스크로 간주됩니다.

- DB2 인스턴스 작성 및 삭제
- DB2 인스턴스 시작 및 정지
- 데이터베이스 작성
- 데이터베이스 관리 프로그램 구성 매개변수 또는 DAS(DB2 Administration Server) 구성 매개변수 갱신
- CLI 구성 매개변수 갱신 및 시스템 데이터 소스 이름(DSN) 구성
- DB2 추적 기능 시작

- db2pd 유틸리티 실행
- DB2 프로파일 레지스트리 변수 변경

문제점을 해결하려면 전체 관리자 특권으로 실행 중인 명령 프롬프트나 그래픽 도구에서 DB2 관리 태스크를 수행해야 합니다. 전체 관리자 특권으로 명령 프롬프트 또는 그래픽 도구를 실행하려면 단축키를 마우스 오른쪽 단추로 누른 다음 **관리자로 실행**을 선택하십시오.

주: 확장된 보안이 사용 가능한 경우, 사용자가 DB2ADMNS 그룹의 구성원이어야 그래픽 관리 도구(예: 명령 편집기 또는 제어 센터)를 실행할 수 있습니다.

사용자 데이터 위치

사용자 데이터(예: 인스턴스 디렉토리에 있는 파일)는 ProgramData₩IBM₩DB2₩*copy_name*에 저장됩니다. 여기서 *copy_name*은 DB2 사본의 이름입니다. 첫 번째로 설치된 사본의 이름은 디폴트로 DB2COPY1입니다. Vista 이외의 Windows 버전에서는 사용자 데이터가 Documents and Settings₩All Users₩Application Data₩IBM₩DB2₩*copy_name*에 저장됩니다.

DB2 및 UNIX 보안

미리 알아 두어야 하는 UNIX 플랫폼의 특정 보안 고려사항이 있습니다.

DB2 데이터베이스는 직접 데이터베이스 관리자의 역할을 하는 root를 지원하지 않습니다. su - <instance owner>를 데이터베이스 관리자로 사용해야 합니다.

보안 상의 이유로, 보통 인스턴스 이름을 분리 ID로 사용하지 마십시오. 그러나 분리 UDF 또는 스토어드 프로시저를 사용하려고 계획하지 않는 경우, 또 다른 사용자 ID를 작성하는 대신 인스턴스 이름에 분리 ID를 설정할 수 있습니다.

이 그룹과 연관하여 인식되는 사용자 ID를 작성하는 것이 좋습니다. 분리 UDF 및 스토어드 프로시저에 대한 사용자는 인스턴스 작성 스크립트의 매개변수(db2icrt... -u <FencedID>)로서 지정됩니다. DB2 Client 또는 DB2 Software Developer's Kit을 설치하는 경우에는 필수가 아닙니다.

DB2 및 Linux 보안

Linux 플랫폼과 관련하여 유의해야 할 몇 가지 보안 고려사항이 있습니다.

암호 변경 지원(Linux)

DB2 데이터베이스 제품은 Linux 운영 체제에서 암호 변경을 지원합니다.

이 지원은 IBMOSchgpwdclient.so 및 IBMOSchgpwdserver.so라는 보안 플러그인 라이브러리 사용을 통해 구현됩니다.

Linux에서 암호 변경 지원을 사용하려면 데이터베이스 관리 프로그램 구성 매개변수 **CLNT_PW_PLUGIN**을 IBMOSchgpwdclient로, **SRVCON_PW_PLUGIN**을 IBMOSchgpwdserver로 설정하십시오.

"db2"라는 PAM 구성 파일도 /etc/pam.d 디렉토리에 작성해야 합니다.

암호 변경 플러그인 전개(Linux)

Linux에서, DB2 데이터베이스 제품의 암호 변경을 지원하려면 보안 플러그인 IBMOSchgpwdclient 및 IBMOSchgpwdserver를 사용하도록 DB2 인스턴스를 구성해야 합니다.

플러그인 라이브러리는 다음 디렉토리에 있습니다.

- *INSTHOME*/sqllib/security*XX*/plugin/IBM/client/IBMOSchgpwdclient.so
- *INSTHOME*/sqllib/security*XX*/plugin/IBM/server/IBMOSchgpwdserver.so

여기서, *INSTHOME*은 인스턴스 소유자의 홈 디렉토리이며 *securityXX*는 인스턴스 비트 너비에 따라 security32 또는 security64입니다.

DB2 인스턴스에서 보안 플러그인을 전개하려면 다음 단계를 수행하십시오.

1. 루트 권한을 가진 사용자로 로그인하십시오.
2. PAM 구성 파일 /etc/pam.d/db2를 작성하십시오.

   파일에 시스템 관리자가 정의한 적절한 규칙 세트가 있어야 합니다. 예를 들어, SLES 9에서 다음과 같이 사용될 수 있습니다.

   ```
   auth     required pam_unix2.so    nullok
   account  required pam_unix2.so
   password required pam_pwcheck.so  nullok tries=1
   password required pam_unix2.so    nullok use_authtok use_first_pass
   session  required pam_unix2.so
   ```

   RHEL에서는 다음과 같이 사용될 수 있습니다.

   ```
   #%PAM-1.0
   auth     required   /lib/security/$ISA/pam_env.so
   auth     sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
   auth     required   /lib/security/$ISA/pam_deny.so

   account  required   /lib/security/$ISA/pam_unix.so
   account  sufficient /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
   account  required   /lib/security/$ISA/pam_permit.so

   password requisite  /lib/security/$ISA/pam_cracklib.so retry=3 dcredit=-1 ucredit=-1
   password sufficient /lib/security/$ISA/pam_unix.so nullok use_authtok md5 shadow remember=3
   password required   /lib/security/$ISA/pam_deny.so

   session  required   /lib/security/$ISA/pam_limits.so
   session  required   /lib/security/$ISA/pam_unix.so
   ```

3. DB2 인스턴스에서 보안 플러그인을 사용 가능으로 설정하십시오.

a. 데이터베이스 관리 프로그램 구성 매개변수 **SRVCON_PW_PLUGIN**을 IBMOSchgpwdserver 값으로 갱신하십시오.

```
db2 update dbm cfg using srvcon_pw_plugin IBMOSchgpwdserver
```

b. 데이터베이스 관리 프로그램 구성 매개변수 **CLNT_PW_PLUGIN**을 IBMOSchgpwdclient 값으로 갱신하십시오.

```
db2 update dbm cfg using CLNT_PW_PLUGIN IBMOSchgpwdclient
```

c. 데이터베이스 관리 프로그램 구성 매개변수 **SRVCON_AUTH**가 CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP로 설정되거나, 데이터베이스 관리 프로그램 구성 매개변수 **SRVCON_AUTH**가 NOT_SPECIFIED로 설정되어 있고, **AUTHENTICATION**이 CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT 또는 DATA_ENCRYPT_CMP로 설정되어야 합니다.

부록 A. DB2 기술 정보 개요

DB2 기술 정보는 다음 도구 및 메소드를 통해 사용할 수 있습니다.

- DB2 정보 센터
  - 주제 항목(태스크, 개념 및 참조 항목)
  - DB2 도구에 대한 도움말
  - 샘플 프로그램
  - 자습서
- DB2 서적
  - PDF 파일(다운로드)
  - PDF 파일(DB2 PDF DVD)
  - 인쇄된 서적
- 명령행 도움말
  - 명령 도움말
  - 메시지 도움말

주: DB2 정보 센터 주제는 PDF 또는 하드카피 서적보다 자주 갱신됩니다. 최신 정보를 보려면 사용 가능한 문서 갱신사항을 설치하거나 ibm.com에서 DB2 정보 센터를 참조하십시오.

ibm.com에서 추가 DB2 기술 정보(예: 기술 노트, 백서 및 IBM Redbooks® 서적)를 온라인으로 액세스할 수 있습니다. DB2 정보 관리 라이브러리 소프트웨어 사이트 http://www.ibm.com/software/data/sw-library/에 액세스하십시오.

문서 피드백

DB2 문서에 대한 피드백을 환영합니다. DB2 문서를 향상시키는 방법에 대해서 제안사항이 있는 경우 db2docs@ca.ibm.com으로 전자 우편을 보내십시오. DB2 문서 팀에서는 고객의 모든 피드백을 읽지만 직접 응답할 수는 없습니다. 고객의 문제를 더 잘 이해할 수 있도록 가능한 위치에 특정 예를 제공해주십시오. 특정 주제 또는 도움말 파일에 대한 피드백을 보내실 경우, 제목 및 URL을 알려주십시오.

DB2 고객 지원에 문의할 때 이 전자 우편 주소를 사용하지 마십시오. 문서에서 해결할 수 없는 DB2 기술 문제점이 있는 경우, 해당 지역의 IBM 서비스 센터에 도움을 요청하십시오.

DB2 기술 라이브러리(하드카피 또는 PDF 형식)

다음 표는 IBM Publications Center(www.ibm.com/shop/publications/order)에서 사용할 수 있는 DB2 라이브러리에 대해 설명합니다. PDF 형식의 영문 DB2 버전 9.7 매뉴얼 및 번역된 버전은 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947에서 다운로드할 수 있습니다.

표에 인쇄할 수 있는 책이 나와 있는 경우에도, 사용 국가 또는 지역에서 해당 책을 사용할 수 없을 수도 있습니다.

매뉴얼이 갱신될 때마다 문서 번호가 증가합니다. 다음 사항을 참조하여 읽고 있는 매뉴얼이 최신 버전인지 확인하십시오.

주: *DB2 정보 센터*는 PDF 또는 하드카피 서적보다 자주 갱신됩니다.

표 52. DB2 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
관리 API 참조서	SA30-3958-00	예	2009년 8월
관리 루틴 및 뷰	SA30-3955-00	아니오	2009년 8월
Call Level Interface Guide and Reference, Volume 1	SC27-2437-00	예	2009년 8월
Call Level Interface Guide and Reference, Volume 2	SC27-2438-00	예	2009년 8월
명령어 참조서	SA30-3959-00	예	2009년 8월
데이터 이동 유틸리티 안내서 및 참조서	SA30-3969-00	예	2009년 8월
데이터 복구 및 고가용성 안내서 및 참조서	SA30-3970-00	예	2009년 8월
데이터베이스 관리 개념 및 구성 참조서	SA30-3951-00	예	2009년 8월
데이터베이스 모니터링 안내서 및 참조서	SA30-3953-00	예	2009년 8월
데이터베이스 보안 안내서	SA30-3971-00	예	2009년 8월
DB2 Text Search Guide	SC27-2459-00	예	2009년 8월
Developing ADO.NET and OLE DB Applications	SC27-2444-00	예	2009년 8월
Developing Embedded SQL Applications	SC27-2445-00	예	2009년 8월
Developing Java Applications	SC27-2446-00	예	2009년 8월
Developing Perl, PHP, Python, and Ruby on Rails Applications	SC27-2447-00	아니오	2009년 8월

표 52. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
Developing User-defined Routines(SQL and External)	SC27-2448-00	예	2009년 8월
Getting Started with Database Application Development	GI11-9410-00	예	2009년 8월
Linux 및 Windows에서 DB2 설치 및 관리 시작하기	GA30-3960-00	예	2009년 8월
자국어 안내서	SA30-3972-00	예	2009년 8월
DB2 Server 설치	GA30-3962-00	예	2009년 8월
IBM Data Server Client 설치	GA30-3963-00	아니오	2009년 8월
Message Reference Volume 1	SC27-2450-00	아니오	2009년 8월
Message Reference Volume 2	SC27-2451-00	아니오	2009년 8월
Net Search Extender Administration and User's Guide	SC27-2469-00	아니오	2009년 8월
파티셔닝 및 클러스터링 안내서	SA30-3973-00	예	2009년 8월
pureXML Guide	SC27-2465-00	예	2009년 8월
Query Patroller 관리 및 사용자 안내서	SA30-3974-00	아니오	2009년 8월
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC27-2468-00	아니오	2009년 8월
SQL Procedural Languages: Application Enablement and Support	SC27-2470-00	예	2009년 8월
SQL 참조서, 볼륨 1	SA30-3956-00	예	2009년 8월
SQL 참조서, 볼륨 2	SA30-3957-00	예	2009년 8월
문제점 해결 및 데이터베이스 성능 조정	SA30-3952-00	예	2009년 8월
DB2 버전 9.7로 업그레이드	SA30-3961-00	예	2009년 8월
Visual Explain 자습서	SA30-3968-00	아니오	2009년 8월
DB2 버전 9.7의 새로운 내용	SA30-3967-00	예	2009년 8월
Workload Manager Guide and Reference	SC27-2464-00	예	2009년 8월

표 52. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
XQuery Reference	SC27-2466-00	아니오	2009년 8월

표 53. DB2 Connect 특정 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
DB2 Connect Personal Edition 설치 및 구성	SA30-3965-00	예	2009년 8월
DB2 Connect Server 설치 및 구성	SA30-3966-00	예	2009년 8월
DB2 Connect 사용자 안내서	SA30-3964-00	예	2009년 8월

표 54. Information Integration 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
Information Integration: Administration Guide for Federated Systems	SC19-1020-02	예	2009년 8월
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-04	예	2009년 8월
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-02	아니오	2009년 8월
Information Integration: SQL Replication Guide and Reference	SC19-1030-02	예	2009년 8월
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-02	예	2009년 8월

인쇄된 DB2 서적 주문

인쇄된 DB2 서적이 필요한 경우, 대부분 온라인으로 구매할 수 있으나 모든 국가 또는 지역에 해당되지는 않습니다. 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. *DB2 PDF 문서* DVD의 일부 소프트카피 서적은 인쇄할 수 없다는 점에 유의하십시오. 예를 들면, *DB2 메시지 참조서*의 어떤 볼륨도 인쇄된 서적으로 사용할 수 없습니다.

DB2 PDF 문서 DVD에서 사용할 수 있는 다수의 DB2 서적의 인쇄된 버전은 IBM에서 유료로 주문할 수 있습니다. 주문하는 위치에 따라 IBM Publications Center에서 온라인으로 서적을 주문할 수도 있습니다. 해당 국가 또는 지역에서 온라인 주문이

불가능하면, 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 모든 서적을 인쇄할 수는 없다는 점에 유의하십시오.

주: 가장 최신 및 완료된 DB2 문서는 http://publib.boulder.ibm.com/infocenter/db2luw/v9r7의 DB2 정보 센터에서 유지보수됩니다.

인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.

- 해당 국가 또는 지역에서 인쇄된 DB2 서적을 온라인으로 주문할 수 있는지 여부를 확인하려면 http://www.ibm.com/shop/publications/order의 IBM Publications Center를 확인하십시오. 서적 주문 정보에 액세스하려면 국가/지역/언어를 선택한 다음 해당 위치에서 주문 지시사항을 따르십시오.
- 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.
  1. 다음 웹 사이트 중 하나에서 해당 지역 담당자에 대한 문의처 정보를 찾으십시오.
     - www.ibm.com/planetwide에 있는 IBM 월드 와이드 문의처 디렉토리
     - http://www.ibm.com/shop/publications/order의 IBM Publications 웹 사이트. 사용 지역의 해당 서적 홈 페이지에 액세스하려면 해당 국가, 지역 또는 언어를 선택해야 합니다. 이 페이지에서 "이 제품의 정보" 링크를 수행하십시오.
  2. 전화로 주문할 경우, 주문할 DB2 서적을 지정하십시오.
  3. 담당자에게 주문하려는 서적의 제목 및 문서 번호를 제공하십시오. 서적의 제목 및 문서 번호는 336 페이지의 『DB2 기술 라이브러리(하드카피 또는 PDF 형식)』를 참조하십시오.

명령행 처리기에서 SQL 상태 도움말 표시

DB2 제품은 SQL문의 결과로 나타나는 상태에 대한 SQLSTATE 값을 리턴합니다. SQLSTATE 도움말은 SQL 상태 및 SQL 상태 클래스 코드의 의미를 설명합니다.

SQL 상태 도움말을 시작하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? sqlstate or ? class code
```

여기서, *sqlstate*는 유효한 5자리 숫자로 된 SQL 상태이고 *class code*는 SQL 상태의 처음 2자리 숫자를 나타냅니다.
예를 들어, `? 08003`은 08003 SQL 상태에 대한 도움말을 표시하고, `? 08`은 08 클래스 코드에 대한 도움말을 표시합니다.

DB2 정보 센터의 다른 버전에 액세스

DB2 버전 9.7 주제에 대한 DB2 정보 센터 URL은 http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/입니다.

DB2 버전 9.5 주제에 대한 DB2 정보 센터 URL은 http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/입니다.

DB2 버전 9 주제에 대한 DB2 정보 센터 URL은 http://publib.boulder.ibm.com/infocenter/db2luw/v9/입니다.

DB2 버전 8 주제에 대한 버전 8 정보 센터 URL은 http://publib.boulder.ibm.com/infocenter/db2luw/v8/입니다.

DB2 정보 센터에서 원하는 언어로 항목 표시

DB2 정보 센터는 브라우저 환경 설정에 지정된 언어로 주제 항목을 표시합니다. 주제가 원하는 언어로 변환되지 않은 경우, DB2 정보 센터는 영어로 주제 항목을 표시합니다.

- Internet Explorer 브라우저에서 원하는 언어로 항목을 표시하려면 다음을 수행하십시오.
  1. Internet Explorer에서 **도구 —> 인터넷 옵션 —> 언어...** 단추를 누르십시오. 언어 환경 설정 창이 열립니다.
  2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
     - 목록에 새 언어를 추가하려면 **추가...** 단추를 누르십시오.

       주: 언어를 추가했다고 컴퓨터에 원하는 언어로 항목을 표시하는 데 필요한 글꼴이 설치되는 것은 아닙니다.
     - 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 언어가 언어 목록의 첫 번째 항목이 될 때까지 **위로 이동** 단추를 누르십시오.
  3. 브라우저 캐시를 지운 후 페이지를 새로 고쳐 원하는 언어로 DB2 정보 센터를 표시하십시오.
- Firefox 또는 Mozilla 브라우저에서 원하는 언어로 주제 항목을 표시하려면 다음을 수행하십시오.
  1. **도구 —> 옵션 —> 고급** 대화 상자의 **언어** 섹션에서 단추를 선택하십시오. 환경 설정 창에 언어 패널이 표시됩니다.
  2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
     - 목록에 새 언어를 추가하려면 **추가...** 단추를 눌러 언어 추가 창에서 언어를 선택합니다.

– 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 언어가 언어 목록의 첫 번째 항목이 될 때까지 **위로 이동** 단추를 누르십시오.

3. 브라우저 캐시를 지운 후 페이지를 새로 고쳐 원하는 언어로 DB2 정보 센터를 표시하십시오.

일부 브라우저 및 운영 체제 조합에서 운영 체제의 국가별 설정을 선택한 로케일 및 언어로 변경해야 합니다.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신

로컬로 설치된 DB2 정보 센터는 주기적으로 갱신해야 합니다.

시작하기 전에

DB2 버전 9.7 정보 센터는 등록된 상태여야 합니다. 자세한 내용은 *DB2 Server 설치*의 『DB2 설치 마법사를 사용하여 DB2 정보 센터 설치』 주제를 참조하십시오. 정보 센터 설치에 적용되는 모든 전제조건 및 제한사항은 정보 센터 갱신에도 적용됩니다.

이 태스크에 대한 정보

기존의 DB2 정보 센터는 자동 또는 수동으로 갱신할 수 있습니다.

- 자동 갱신 - 기존 정보 센터 기능 및 언어를 갱신합니다. 자동 갱신의 또 다른 이점으로는 갱신 동안 정보 센터를 아주 잠시 동안만 사용 불가능하다는 점입니다. 또한 자동 갱신은 주기적으로 실행되는 기타 일괄처리 작업의 일부로 실행되도록 설정 가능합니다.
- 수동 갱신 - 갱신 프로세스 중에 기능이나 언어를 추가하려는 경우 사용하십시오. 예를 들어, 로컬 정보 센터는 기본적으로 영어와 프랑스어로 설치되어 있으며 기존 정보 센터의 기능 및 언어 갱신 외에도 수동 갱신으로 독어도 설치할 수 있습니다. 단, 수동 갱신을 수행하려면 정보 센터를 중지한 다음 갱신하고 재시작해야 합니다. 정보 센터는 갱신 프로세스 동안에는 사용할 수 없습니다.

프로시저

이 주제는 자동 갱신 프로세스에 대한 설명입니다. 수동 갱신에 대한 지시사항은 『컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신』 주제를 참조하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 자동으로 갱신하려면 다음을 수행하십시오.

1. Linux 운영 체제에서,
   a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 `/opt/ibm/db2ic/V9.7` 디렉토리에 디폴트로 설치됩니다.
   b. 설치 디렉토리에서 `doc/bin` 디렉토리까지 탐색하십시오.

c. 다음과 같이 ic-update 스크립트를 실행하십시오.

```
ic-update
```

2. Windows 운영 체제에서,
   a. 명령 창을 여십시오.
   b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>₩IBM₩DB2 Information Center₩Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
   c. 설치 디렉토리에서 doc₩bin 디렉토리까지 탐색하십시오.
   d. 다음과 같이 ic-update.bat 파일을 실행하십시오.

```
ic-update.bat
```

결과

DB2 정보 센터가 자동으로 재시작됩니다. 갱신사항이 사용 가능한 경우, 정보 센터에는 새로 갱신된 주제가 표시됩니다. 정보 센터 갱신을 사용할 수 없는 경우, 메시지가 로그에 추가됩니다. 로그 파일은 doc₩eclipse₩configuration 디렉토리에 있습니다. 이 로그 파일 이름은 임의로 생성된 번호입니다. (예: 1239053440785.log).

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신

DB2 정보 센터를 로컬로 설치한 경우, IBM으로부터 문서 갱신사항을 받아 설치할 수 있습니다.

로컬로 설치된 DB2 정보 센터를 수동으로 갱신하려면 다음을 수행하십시오.

1. 컴퓨터에서 DB2 정보 센터를 중지한 후 독립형 모드에서 다시 시작하십시오. 독립형 모드에서 정보 센터를 실행하면 사용자의 네트워크와 연결된 다른 사용자는 정보 센터에 액세스할 수 없으므로 갱신사항을 적용할 수 있습니다. DB2 정보 센터의 워크스테이션 버전은 항상 독립형 모드에서 실행됩니다. .
2. 어떤 갱신사항이 사용 가능한지 확인하려면 갱신 기능을 사용하십시오. 설치해야 할 갱신사항이 있는 경우, 갱신 기능을 사용하여 이를 가져온 후 설치할 수 있습니다.

   주: 인터넷에 연결되지 않은 머신에 DB2 정보 센터 갱신사항을 설치해야 할 경우, 인터넷에 연결된 머신을 사용하여 갱신 사이트를 로컬 파일 시스템에 미러해야 DB2 정보 센터가 설치됩니다. 네트워크 상에 문서 갱신사항을 설치하려는 사용자가 많을 경우에는 갱신 사이트를 로컬로 미러링하거나 갱신 사이트의 프록시를 작성하여 갱신을 수행하면 각 개인에게 필요한 시간을 줄일 수 있습니다.
   갱신 패키지가 사용 가능하면 갱신 기능을 사용하여 패키지를 가져오십시오. 그러나 갱신 기능은 독립형 모드에서만 사용할 수 있습니다.
3. 독립형 정보 센터를 중지한 후 컴퓨터에서 DB2 정보 센터를 재시작하십시오.

주: Windows 2008, Windows Vista 이상의 경우 이 섹션 다음에 나오는 명령은 관리자로 실행해야 합니다. 전체 관리자 권한으로 명령 프롬프트 또는 그래픽 도구를 열려면 단축키를 마우스 오른쪽 단추로 누른 후 **관리자로 실행**을 선택하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 갱신하려면 다음을 수행하십시오.

1. DB2 정보 센터를 중지하십시오.
   - Windows에서는 **시작 → 제어판 → 관리 도구 → 서비스**를 누르십시오. 그런 다음 **DB2 정보 센터** 서비스를 마우스 오른쪽 단추로 누른 후 **중지**를 선택하십시오.
   - Linux에서는 다음 명령을 입력하십시오.

     ```
     /etc/init.d/db2icdv97 stop
     ```
2. 독립형 모드에서 정보 센터를 시작하십시오.
   - Windows 사용자:
     a. 명령 창을 여십시오.
     b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>₩IBM₩DB2 Information Center₩Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
     c. 설치 디렉토리에서 doc₩bin 디렉토리까지 탐색하십시오.
     d. 다음과 같이 help_start.bat 파일을 실행하십시오.

        ```
        help_start.bat
        ```
   - Linux 사용자:
     a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 /opt/ibm/db2ic/V9.7 디렉토리에 디폴트로 설치됩니다.
     b. 설치 디렉토리에서 doc/bin 디렉토리까지 탐색하십시오.
     c. 다음과 같이 help_start 스크립트를 실행하십시오.

        ```
        help_start
        ```

   독립형 정보 센터를 표시하기 위해 시스템의 기본 웹 브라우저가 열립니다.
3. **갱신** 단추( )를 누르십시오. (JavaScript™가 브라우저에서 사용 가능해야 합니다.) 정보 센터의 오른쪽 패널에서 **갱신사항 찾기**를 누르십시오. 기존 문서의 갱신사항 목록이 표시됩니다.
4. 설치 프로세스를 시작하려면 설치할 선택란을 체크한 후 **갱신사항 설치**를 누르십시오.
5. 설치 프로세스가 완료되면 **완료**를 누르십시오.
6. 독립형 정보 센터를 중지하십시오.

- Windows에서 설치 디렉토리의 doc₩bin 디렉토리를 탐색한 후 다음과 같이 help_end.bat 파일을 실행하십시오.

  ```
  help_end.bat
  ```

  주: help_end 일괄처리 파일에는 help_start 일괄처리 파일로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start.bat를 중지하는 데 Ctrl-C 또는 다른 메소드를 사용하지 마십시오.
- Linux에서 설치 디렉토리의 doc/bin 디렉토리를 탐색한 후 다음과 같이 help_end 스크립트를 실행하십시오.

  ```
  help_end
  ```

  주: help_end 스크립트에는 help_start 스크립트로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start 스크립트를 중지하는 데 다른 메소드를 사용하지 마십시오.

7. DB2 정보 센터를 재시작하십시오.
   - Windows에서는 **시작 → 제어판 → 관리 도구 → 서비스**를 누르십시오. 그런 다음 **DB2 정보 센터** 서비스를 마우스 오른쪽 단추로 누른 후 **시작**을 선택하십시오.
   - Linux에서는 다음 명령을 입력하십시오.

     ```
     /etc/init.d/db2icdv97 start
     ```

갱신된 DB2 정보 센터에는 새로 갱신된 주제가 표시됩니다.

DB2 자습서

DB2 자습서는 DB2 제품의 다양한 측면에 대해 학습하는 데 유용합니다. 각 레슨은 단계별 지시사항을 제공합니다.

시작하기 전에

http://publib.boulder.ibm.com/infocenter/db2help/의 정보 센터에서 XHTML 버전의 자습서를 볼 수 있습니다.

일부 레슨에서는 샘플 데이터나 코드를 사용합니다. 특정 태스크에 대한 전제조건 설명은 자습서를 참조하십시오.

DB2 자습서

자습서를 보려면 제목을 누르십시오.

『**pureXML®**』(*pureXML Guide*)

XML 데이터를 저장하고 원시 XML 데이터 스토어로 기본 조작을 수행하려면 DB2 데이터베이스를 설정하십시오.

*Visual Explain 자습서*의 『**Visual Explain**』

Visual Explain을 사용하여 성능을 향상시킬 수 있도록 SQL문을 분석, 최적화 및 조정합니다.

DB2 문제점 해결 정보

DB2 데이터베이스 제품을 사용하는 데 도움이 되는 광범위한 문제를 해결하고 판별할 수 있는 정보가 있습니다.

DB2 문서

문제점 해결 정보는 *DB2 문제점 해결 안내서* 또는 *DB2 정보 센터*의 데이터베이스 기본 섹션을 참조하십시오. DB2 진단 도구 및 유틸리티를 사용하여 문제점을 찾아내고 식별하는 방법, 가장 일반적인 문제점에 대한 솔루션 및 DB2 데이터베이스 제품에서 발생할 수 있는 문제점을 해결하는 방법 등의 정보가 있습니다.

DB2 기술 지원 웹 사이트

문제점이 있는 경우 원인 및 솔루션을 찾으려면 DB2 기술 지원 웹 사이트를 참조하십시오. 기술 지원 사이트에는 최신 DB2 서적, 기술 노트, APAR(Authorized Program Analysis Report 또는 버그 수정), FixPack 및 기타 자원에 대한 링크가 있습니다. 이러한 기술 자료를 검색하여 문제에 대한 가능한 솔루션을 찾을 수 있습니다.

http://www.ibm.com/software/data/db2/support/db2_9/에서 DB2 기술 지원 웹 사이트에 액세스하십시오.

이용약관

다음 조건에 따라 이 책을 사용할 수 있습니다.

개인적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 본 문서를 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 문서 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

상업적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 본 문서를 귀하 사업장 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 문서의 2차적 저작물을 만들거나 본 문서 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이센스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 본 문서의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 본 문서의 내용에 대해 어떠한 보증도 제공하지 않습니다. 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

부록 B. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 비IBM 제품에 대한 정보는 이 책을 처음 발행할 때의 정보에 기초하고 있으며 변경될 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이센스까지 부여하는 것은 아닙니다. 라이센스에 대한 의문사항은 다음으로 문의하십시오.

135-700
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터
전화번호: 080-023-8080

2바이트 문자 세트(DBCS) 정보에 관한 라이센스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 『현상태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이센스 사용자는 다음 주소로 문의하십시오.

135-700
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠. 주식회사
고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등) 하에서 사용될 수 있습니다.

이 정보에 기술된 라이센스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이센스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이센스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 따라서 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이센스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 되는 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 어떠한 보증없이 ″있는 그대로″ 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 모든 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (*귀하의 회사명*) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. *_enter* 연도_. All rights reserved.

상표

IBM, IBM 로고 및 ibm.com®은 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 기타 회사의 상표입니다. 현재 IBM 상표 목록은 웹 “저작권 및 상표 정보”(www.ibm.com/legal/kr/copytrade.shtml)에 있습니다.

다음 표장은 기타 회사의 상표 또는 등록상표입니다.

- Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.
- Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.
- UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.
- Intel®, Intel 로고, Intel Inside®, Intel Inside 로고, Intel® Centrino®, Intel Centrino 로고, Celeron®, Intel® Xeon®, Intel SpeedStep®, Itanium® 및 Pentium®은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.
- Microsoft, Windows, Windows NT® 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

색인

[가]

[나]

[다]

[라]

[마]

[바]

[사]

[아]

[자]

[카]

[타]

[파]

[하]

A

B

C

D

E

G

I

K

L

N

O

P

Q

R

S

T

U

V

W

X

[특수 문자]

SA30-3971-00

Spine information:

Linux, UNIX 및 Windows용 IBM DB2 9.7

데이터베이스 보안 안내서

IBM