

IBM[®] DB2 Universal Database[™]



Tutorial do Visual Explain

Versão 8

IBM[®] DB2 Universal Database[™]



Tutorial do Visual Explain

Versão 8

Antes de utilizar estas informações e o produto a que elas se referem, certifique-se de ter lido as informações gerais na seção *Avisos*.

Este documento contém informações de propriedade da IBM. Ele é fornecido sob um acordo de licença e é protegido pela lei de copyright. As informações contidas nesta publicação não incluem garantias de produto, e nenhuma declaração feita neste manual deve ser interpretada como tal.

Você pode solicitar publicações da IBM online ou através do representante IBM local.

- Para solicitar publicações online, acesse o IBM Publications Center em www.ibm.com/shop/publications/order
- Para localizar o representante IBM local, acesse o IBM Directory of Worldwide Contacts em www.ibm.com/planetwide

Quando o Cliente envia seus comentários para a IBM, concede direitos não-exclusivos à IBM para usá-los ou distribuí-los da maneira que achar conveniente, sem que isso implique em qualquer compromisso ou obrigação para com o Cliente.

Índice

Sobre este Tutorial	v
Informações de Ambiente Específico	vi
Lição 1. Criando Instantâneos de Explicação	1
Criando as Tabelas de Explicação	1
Utilizando Instantâneos de Explicação.	2
Criando Instantâneos de Explicação para Instruções SQL Dinâmicas.	3
Criando Instantâneos de Explicação para Instruções SQL Estáticas	4
O que Vem a Seguir	5
Lição 2. Exibindo e Utilizando um Gráfico do Plano de Acesso	7
Exibindo um Gráfico do Plano de Acesso Escolhendo em uma Lista de Instruções SQL Explicadas Anteriormente	7
Lendo os Símbolos em um Gráfico do Plano de Acesso	8
Utilizando a Régua de Controle de Zoom para Ampliar Partes de um Gráfico	8
Obtendo mais Detalhes sobre os Objetos em um Gráfico.	9
Obtendo Estatísticas para Tabelas, Índices e Funções de Tabela	9
Obtendo Detalhes sobre Operadores em um Gráfico	10
Obtendo Estatísticas para Funções	10
Obtendo Estatísticas para Áreas de Tabela	10
Obtendo Estatísticas para Colunas em uma Instrução SQL	11
Obtendo Informações sobre Parâmetros de Configuração e Opções de União	11
Alterando a Aparência de um Gráfico	11
O que Vem a Seguir	12
Lição 3. Melhorando um Plano de Acesso em um Ambiente de Banco de Dados de Partição Única.	13
Trabalhando com Gráficos do Plano de Acesso	13
Executando uma Consulta sem Índices e Sem Estatísticas	14
Coletando Estatísticas Atuais para as Tabelas e Índices Utilizando runstats.	17
Criando Índices em Colunas Utilizadas para Unir Tabelas em uma Consulta	21
Criando Índices Adicionais em Colunas de Tabela	27
O que Vem a Seguir	29
Lição 4. Melhorando o Plano de Acesso em um Ambiente de Banco de Dados Particionado	31
Trabalhando com Gráficos do Plano de Acesso	31
Executando uma Consulta sem Índices e Sem Estatísticas	32
Coletando Estatísticas Atuais para as Tabelas e Índices Utilizando runstats.	35
Criando Índices em Colunas Utilizadas para Unir Tabelas em uma Consulta	39
Criando Índices Adicionais em Colunas de Tabela	44
O que Vem a Seguir	46
Apêndice A. Conceitos do Visual Explain	47
Plano de Acesso.	47
Gráfico do Plano de Acesso	47
Nó do Gráfico do Plano de Acesso	49
Armazenamento em Cluster.	49
Contêiner	49
Custo	49
Bloqueio de Cursor	50
Área de Tabela DMS (database-managed space)	50
SQL Dinâmica	51
Instantâneo de Explicação	51
Instrução Explicável	52
Instrução Explicada	52
Operando	52
Operador	52
CMPEXP	54
DELETE	54
EISCAN	54
FETCH.	55
FILTER.	55
GENROW.	56

GRPBY	56	FILTER.	74
HSJOIN	57	GENROW.	75
INSERT	57	GRPBY.	75
IXAND	58	HSJOIN	75
IXSCAN	58	INSERT	76
MSJOIN	59	IXAND	76
NLJOIN	60	IXSCAN	77
PIPE	61	MSJOIN	78
RETURN	61	NLJOIN	79
RIDSCN	61	PIPE	79
RQUERY	62	RETURN	80
SORT	62	RIDSCN	80
TBSCAN	63	RQUERY	80
TEMP	64	SORT	81
TQUEUE	64	TBSCAN	82
UNION	65	TEMP	83
UNIQUE	65	TQUEUE	83
UPDATE	65	UNION	83
Otimizador	66	UNIQUE	84
Pacote	66	UPDATE	84
Predicado	66		
Classe de Otimização de Consulta	67	Apêndice C. Conceitos do DB2	85
Seletividade de Predicados	68	Banco de Dados.	85
Unões Estrela	69	Esquemas.	85
SQL Estática	70	Tabelas.	86
Áreas de Tabela SMS (System-managed space)	70		
Área de Tabela	70	Apêndice D. Avisos	87
Visual Explain	71	Marcas.	90
Apêndice B. Lista em Ordem Alfabética		Índice Remissivo	93
dos Operadores do Visual Explain.	73		
CMPEXP	73	Entrando em Contato com a IBM	95
DELETE	73	Informações sobre o Produto	95
EISCAN	73		
FETCH.	74		

Sobre este Tutorial

Este tutorial fornece um guia para os recursos do DB2 Visual Explain. Ao completar as lições neste tutorial, você irá aprender como o Visual Explain permite que você exiba o plano de acesso para as instruções SQL explicadas como um gráfico. Você também irá descobrir como utilizar as informações disponíveis de tal gráfico para ajustar suas consultas SQL para obter um melhor desempenho.

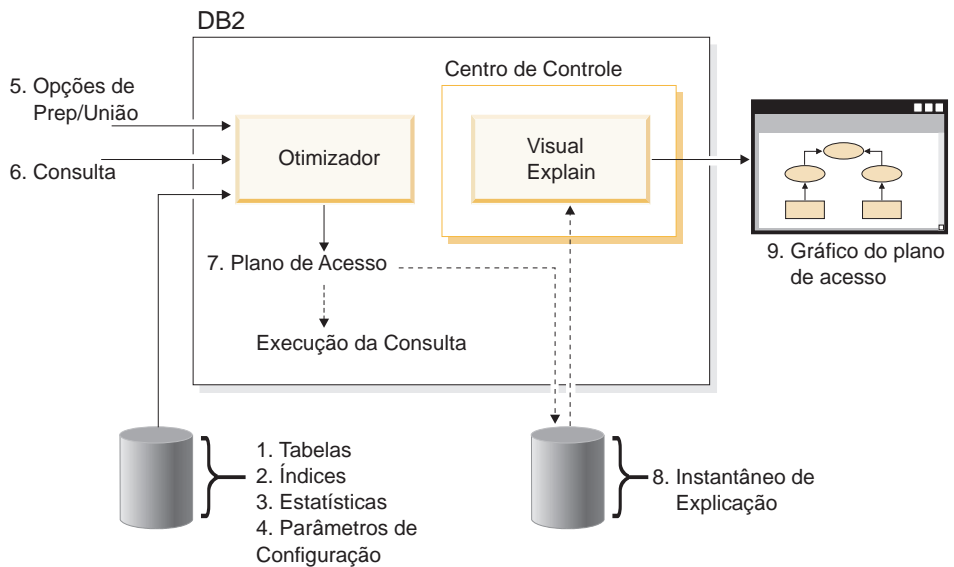
Utilizando seu otimizador, o DB2 examina suas consultas SQL e determina a melhor maneira de acessar seus dados. Este caminho para os dados é chamado de plano de acesso. O DB2 permite exibir o que o otimizador fez permitindo que você observe o plano de acesso selecionado para executar uma consulta SQL específica. Você poderá utilizar o Visual Explain para exibir o plano de acesso como gráfico. O gráfico é uma apresentação visual dos objetos do banco de dados envolvidos em uma consulta (por exemplo, tabelas e índices). Ele também inclui as operações executadas nesses objetos (por exemplo, varreduras e ordenações) e mostra o fluxo dos dados.

Você poderá melhorar o acesso a dados de uma consulta executando alguma ou todas as atividades de ajuste a seguir:

1. Ajuste seu projeto de tabela e a reorganização dos dados de tabela.
2. Crie índices apropriados.
3. Utilize o comando **runstats** para fornecer ao otimizador as estatísticas atuais.
4. Selecione os parâmetros de configuração apropriados.
5. Selecione as opções de união apropriadas.
6. Projete consultas para recuperar apenas dados requeridos.
7. Trabalhe com um plano de acesso.
8. Crie instantâneos de explicação.
9. Utilize um gráfico do plano de acesso para melhorar o plano de acesso.

Essas atividades relacionadas a desempenho correspondem àquelas mostradas na seguinte ilustração. (Linhas quebradas indicam ações requeridas para o

Visual Explain).



Este tutorial contém lições:

- Criando instantâneos de explicação. Esses são requerimentos para exibir gráficos do plano de acesso.
- Exibindo e manipulando um gráfico do plano de acesso.
- Executando atividades de ajuste e examinando como melhoram seu plano de acesso.

Nota: O ajuste de desempenho é dividido em uma lição para ambientes de banco de dados de partição única e uma lição para ambientes de bancos de dados particionados.

Você irá utilizar o banco de dados SAMPLE fornecido pelo DB2 para concluir as lições. Consulte a publicação *Administration Guide* caso ainda não tenha criado o banco de dados SAMPLE.

Informações de Ambiente Específico



Informações marcadas com esse ícone pertencem somente a ambientes de banco de dados de partição única.



Informações marcadas com esse ícone pertencem somente a ambientes de banco de dados particionado.

Lição 1. Criando Instantâneos de Explicação

Nesta lição, você irá criar instantâneos de explicação. O recurso de explicação SQL é utilizado para capturar informações sobre o ambiente no qual uma instrução SQL estática ou dinâmica é compilada. As informações capturadas permitem que você entenda a estrutura e desempenho de execução em potencial de suas instruções SQL. Um instantâneo de explicação são informações compactadas coletadas quando uma instrução SQL é explicada. Ela é armazenada como um BLOB (binary large object) na tabela EXPLAIN_STATEMENT e contém as seguintes informações:

- A representação interna do plano de acesso, incluindo seus operadores e tabelas, além dos índices acessados.
- Os critérios de decisão utilizados pelo otimizador, incluindo estatísticas para objetos de banco de dados e o custo cumulativo de cada operação.

Para exibir um gráfico do plano de acesso, o Visual Explain requer as informações contidas em um instantâneo de explicação.

Criando as Tabelas de Explicação

Para criar instantâneos de explicação, você deve assegurar que as seguintes tabelas de explicação existam para seu ID de usuário:

- EXPLAIN_INSTANCE
- EXPLAIN_STATEMENT

Para verificar se elas existem, utilize o comando **DB2 list tables**. Se essas tabelas não existirem, será necessário criá-las utilizando as seguintes instruções:

1. Se o DB2 ainda não tiver sido iniciado, execute o comando **db2start**.
2. A partir do prompt do processador de linha de comandos (CLP) do DB2, conecte-se ao banco de dados que deseja utilizar. Para este tutorial, conecte-se ao banco de dados SAMPLE utilizando o comando **connect to sample**.
3. Crie as tabelas de explicação, utilizando o arquivo de comando de exemplo fornecido no arquivo EXPLAIN.DDL. Este arquivo está localizado no diretório sqllib\misc. Para executar o arquivo de comando, acesse este diretório e execute o comando **db2 -tf EXPLAIN.DDL**. Este arquivo de comando cria tabelas de explicação pré-fixadas com o ID de usuário conectado. Este ID de usuário deve possuir privilégio CREATETAB no banco de dados, ou autoridade SYSADM ou DBADM.

Utilizando Instantâneos de Explicação

São fornecidos quatro instantâneos de exemplo para ajudá-lo a descobrir mais sobre o Visual Explain. Informações sobre criação de seus próprios instantâneos são fornecidas nas seções a seguir, mas não será necessário criar seus próprios instantâneos para trabalhar com este tutorial.

- Criando instantâneos de explicação para instruções SQL dinâmicas
- Criando instantâneos de explicação para instruções SQL estáticas

A consulta utilizada para os instantâneos de exemplo listam o nome, departamento e salário de todos os funcionários que não fazem parte da gerência que ganham mais de 90% do salário do gerente com o maior salário.

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

A consulta tem duas partes:

1. A subconsulta (entre parênteses) produz linhas de dados constituídos de 90% do salário de cada gerente. Como a subconsulta está qualificada por ALL, apenas o maior valor desta tabela será recuperado.
2. A consulta principal une todas as linhas nas tabelas ORG e STAFF em que os números de departamento são iguais, JOB não é igual a 'Mgr', e salário mais comissão é maior que o valor retornado da subconsulta.

A consulta principal contém os três seguintes predicados (comparações):

1. O.DEPTNUMB = S.DEPT
2. S.JOB <> 'Mgr'
3. S.SALARY+S.COMM > ALL (SELECT ST.SALARY*.9
 FROM STAFF ST
 WHERE ST.JOB='Mgr')

Esses predicados representam, respectivamente:



1. Um predicado de junção, que une as tabelas ORG e STAFF em que os número de departamento são iguais
2. Um predicado local na coluna JOB da tabela STAFF
3. Um predicado local nas colunas SALARY e COMM da tabela STAFF que utilizam o resultado da subconsulta

Para carregar os instantâneos de exemplo:

1. Se o DB2 ainda não tiver sido iniciado, execute o comando **db2start**.

2. Assegure que as tabelas de explicação existam em seu banco de dados. Para fazer isto, siga as instruções em Criando tabelas de explicação.
3. Conecte-se ao banco de dados que deseja utilizar. Para este tutorial, você poderá conectar-se ao banco de dados SAMPLE. Para conectar-se ao banco de dados SAMPLE, a partir do processador de linha de comandos (CLP) do DB2, execute o comando **connect to sample**.

Se ainda não tiver criado, consulte a seção sobre instalação do banco de dados na publicação *Administration Guide*.

4. Para importar os instantâneos pré-definidos, execute o arquivo de comando do DB2 chamado VESAMPL.DDL.
 -  Este arquivo está localizado no diretório sqllib\samples\ve.
 -  Este arquivo está localizado no diretório sqllib\samples\ve\inter.

Para executar o arquivo de comando, acesse este diretório e execute o comando **db2 -tf vesAMPL.ddl**.

- Este arquivo de comando deve ser executado utilizando o mesmo ID de usuário utilizado para criar as tabelas de explicação.
- Este arquivo de comando importa os instantâneos pré-definidos. Ele não cria tabelas ou dados. As atividades de ajuste descritas posteriormente (por exemplo, CREATE INDEX e runstats), serão executadas em tabelas e dados no banco de dados SAMPLE.

Você agora está pronto para exibir e utilizar os gráficos do plano de acesso.

Criando Instantâneos de Explicação para Instruções SQL Dinâmicas

Nota: As informações de criação do instantâneo de explicação são fornecidas para sua referência. Como são fornecidos instantâneos de explicação de exemplo, não será necessário concluir esta tarefa para trabalhar com o tutorial.

Siga estas etapas para criar um instantâneo de explicação para uma instrução SQL dinâmica:

1. Se o DB2 ainda não tiver sido iniciado, execute o comando **db2start**.
2. Assegure que as tabelas de explicação existam em seu banco de dados. Para fazer isto, siga as instruções em Criando tabelas de explicação.
3. A partir do prompt do processador de linha de comandos do DB2, conecte-se ao banco de dados que deseja utilizar. Por exemplo, para conectar-se ao banco de dados SAMPLE execute o comando **connect to sample**.

Para criar o banco de dados SAMPLE, consulte a seção sobre instalação do banco de dados SAMPLE na publicação *Administration Guide*.

4. Crie um instantâneo de explicação para uma instrução SQL dinâmica, utilizando os seguintes comandos a partir do prompt do processador de linha de comandos do DB2:
 - Para criar um instantâneo de explicação sem executar a instrução SQL, execute o comando **set current explain snapshot=explain**.
 - Para criar um instantâneo de explicação e executar a instrução SQL, execute o comando **set current explain snapshot=yes**.

Este comando define o registro especial de explicação. Assim que estiver definida, as instruções SQL serão afetadas. Para obter mais informações, consulte as seções sobre os instantâneos de explicação atuais na publicação *SQL Reference*.

5. Envie suas instruções SQL no prompt do processador de linha de comandos do DB2.
6. Para exibir um gráfico do plano de acesso para o instantâneo, atualize a janela do Histórico das Instruções Explicadas (disponível a partir do Centro de Controle), e dê um clique duplo no instantâneo.
7. Opcional. Para desligar o recurso de instantâneo, execute o comando **set current explain snapshot=no** após ter enviado suas instruções SQL.

Criando Instantâneos de Explicação para Instruções SQL Estáticas

Nota: As informações de criação do instantâneo de explicação são fornecidas para sua referência. Como são fornecidos instantâneos de explicação de exemplo, não será necessário concluir esta tarefa para trabalhar com o tutorial.

Siga estas etapas para criar um instantâneo de explicação para uma instrução SQL estática:

1. Se o DB2 ainda não tiver sido iniciado, execute o comando **db2start**.
2. Assegure que as tabelas de explicação existam em seu banco de dados. Para fazer isto, siga as instruções em Criando tabelas de explicação.
3. A partir do prompt do processador de linha de comandos do DB2, conecte-se ao banco de dados que deseja utilizar. Por exemplo, para conectar-se ao banco de dados SAMPLE execute o comando **connect to sample**.
4. Crie um instantâneo de explicação para uma instrução SQL estática utilizando a opção EXPLSNAP ao unir ou preparar seu aplicativo. Por exemplo, execute o comando **bind seu arquivo explsnap yes**.
5. Opcional. Para exibir um gráfico do plano de acesso para o instantâneo, atualize a janela do Histórico das Instruções Explicadas (disponível a partir do Centro de Controle), e dê um clique duplo no instantâneo.

Para obter as informações sobre a utilização da opção EXPLSNAP para as APIs equivalentes, consulte as seções para cada uma dessas na publicação *Application Development Guide*.

O que Vem a Seguir

Na “Lição 2. Exibindo e Utilizando um Gráfico do Plano de Acesso” na página 7, você irá aprender como exibir um gráfico do plano de acesso e compreender seu conteúdo.

Lição 2. Exibindo e Utilizando um Gráfico do Plano de Acesso

Neste lição, você irá utilizar a janela Gráfico do Plano de Acesso para exibir e utilizar um gráfico do plano de acesso. Um gráfico do plano de acesso é uma representação gráfica de um plano de acesso. A partir dele, você poderá exibir detalhes para:

- Tabelas (e suas colunas associadas) e índices
- Operadores (como varreduras, ordenações e uniões de tabelas)
- Áreas de tabelas e funções.

Você poderá exibir um gráfico do plano de acesso:

- Selecionando a partir de uma lista de instruções explicadas anteriormente.
- Selecionando a partir de uma lista de instruções explicáveis em um pacote.
- Explicando dinamicamente uma instrução SQL.

Como você estará trabalhando com os gráficos do plano de acesso para os instantâneos de explicação de exemplo carregados na Lição 1, você irá escolher entre uma lista de instruções explicadas anteriormente. Para obter informações sobre os outros métodos de exibição de gráficos do plano de acesso, consulte a Ajuda do Visual Explain.

Exibindo um Gráfico do Plano de Acesso Escolhendo em uma Lista de Instruções SQL Explicadas Anteriormente

Para exibir um gráfico do plano de acesso selecionando em uma lista de instruções explicadas anteriormente:

1. No Centro de Controle, expanda a árvore de objeto até encontrar um banco de dados de exemplo.
2. Dê um clique duplo no banco de dados e selecione **Mostrar histórico de instruções explicadas** a partir do menu popup. A janela Histórico de Instruções Explicadas será aberta.
3. Você poderá exibir apenas um gráfico do plano de acesso para uma instrução que possua um instantâneo de explicação. Instruções que se qualificam terão uma entrada YES na coluna **Instantâneo de Explicação**. Dê um clique duplo na entrada identificada como Consulta Número 1 (pode ser necessário deslocar para a direita para localizar a coluna **Número da Consulta**). A janela gráfico do plano de acesso da instrução será aberta.

Nota: O gráfico será lido de baixo para cima. A primeira etapa da consulta está listada na parte inferior do gráfico e a última etapa está listada na parte superior.

Lendo os Símbolos em um Gráfico do Plano de Acesso

O gráfico do plano de acesso mostra a estrutura de um plano de acesso como árvore. Os *nós* da árvore representam:

- Tabelas, mostradas como retângulos
- Índices, mostrados como losangos
- Operadores, mostrados como octágonos. Operadores TQUEUE, mostrados como paralelogramos
- Funções de tabela, mostrados como hexágonos.

Para operadores, o número entre colchetes à direita do tipo de operador, é um identificador exclusivo para cada nó. O número abaixo do tipo de operador é o custo cumulativo.

Utilizando a Régua de Controle de Zoom para Ampliar Partes de um Gráfico

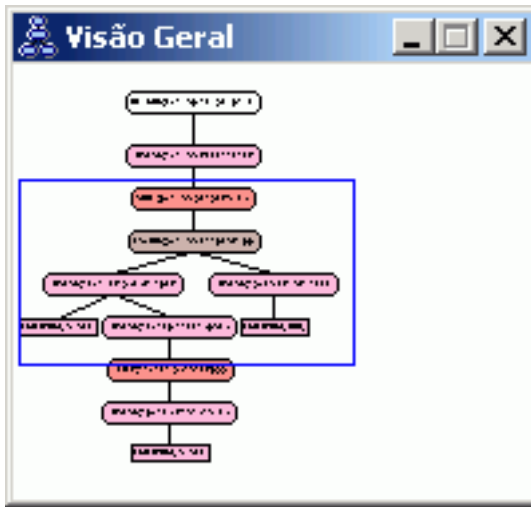
Quando exibir um gráfico do plano de acesso, todo o gráfico será mostrado, e você não poderá ver os detalhes que distinguem cada nó.

A partir da janela Gráfico do Plano de Acesso, utilize a **régua de controle de zoom** para ampliar partes de um gráfico:

1. Posicione o indicador do mouse sobre a caixa pequena de deslocamento na barra da régua de controle de Zoom à esquerda do gráfico.
2. Clique com o botão esquerdo e arraste a régua de controle até que o gráfico esteja no nível de ampliação desejado.

Para exibir partes diferentes do gráfico, utilize a barra de deslocamento.

Para exibir um gráfico do plano de acesso grande e complexo, utilize a janela Visão Geral do Gráfico. Você poderá utilizar esta janela para ver qual parte do gráfico você está exibindo, e para ampliar ou deslocar-se pelo gráfico. A seção na caixa de ampliação será mostrada no plano de acesso.



Para deslocar-se pelo gráfico, posicione o indicador do mouse sobre a área destacada na janela Visão Geral do gráfico, pressione e mantenha pressionado o botão 1 do mouse, e em seguida mova o mouse até ver a parte desejada do gráfico do plano de acesso.

Obtendo mais Detalhes sobre os Objetos em um Gráfico

Você poderá acessar mais informações sobre os objetos em um gráfico do plano de acesso. Você poderá exibir:

- Estatísticas de catálogo de sistema para objetos como:
 - Tabelas, índices ou funções de tabela
 - Informações sobre operadores, como seu custo, propriedades e argumentos de entrada
 - Funções integradas ou funções definidas pelo usuário
 - Áreas de tabela
 - Colunas referenciadas em uma instrução SQL
- Informações sobre parâmetros de configurações e opções de união (parâmetros de otimização).

Obtendo Estatísticas para Tabelas, Índices e Funções de Tabela

Para exibir estatísticas de catálogo para uma única tabela (retângulo), índice (losango) ou função de tabela (hexágono) em um gráfico, dê um clique duplo em seu nó. Uma janela de Estatísticas será aberta para os objetos selecionados, exibindo informações sobre as estatísticas em efeito no momento que o instantâneo foi criado, além daquela que existem atualmente nas tabelas de catálogo de sistema.

Para exibir estatísticas de catálogo para *múltiplas* tabelas, índices ou funções de tabela em um gráfico, selecione cada um clicando nele (ele estará destacado); em seguida selecione **Nó**→**Exibir estatísticas**. Será aberta uma janela de Estatísticas para cada objeto selecionado. (As janelas poderão ficar empilhadas e pode ser necessário arrastar e soltar para poder acessar todas).

Se a entrada para **STATS_TIME** na coluna **Explicado** contiver a entrada **Estatísticas não atualizadas**, então nenhuma estatística existiu quando o otimizador criou o plano de acesso. Conseqüentemente, se o otimizador for requerido em certas estatísticas para criar um plano de acesso, ela utilizou seus padrões. Se o padrão foi utilizado pelo otimizador, elas foram identificadas como (**padrão**) na coluna Explicado.

Obtendo Detalhes sobre Operadores em um Gráfico

Para exibir estatísticas de catálogo para um único operador (octágono), dê um clique duplo em seu nó. Uma janela Detalhes do operador será aberta para o operador selecionado, exibindo informações como:

- O custo cumulativo estimado (E/S, instruções da CPU e custo total)
- A cardinalidade (isto é, o número estimado de linhas pesquisadas) até o momento
- Tabelas que foram acessadas e unidas no plano até o momento
- Colunas dessas tabela que foram acessadas até o momento
- Predicados que foram aplicadas até o momento, incluindo sua seletividade estimada
- Os argumentos de entrada para cada operador.

Para exibir detalhes para *múltiplos operadores*, selecione cada um clicando nele (eles estarão destacados); em seguida selecione **Nó**→**Mostrar Detalhes**. Será aberta uma janela de Estatísticas para cada objeto selecionado. (As janelas poderão ficar empilhadas e pode ser necessário arrastar e soltar para ser possível acessar todas).

Obtendo Estatísticas para Funções

Para exibir estatísticas de catálogo para funções integradas e funções definidas pelo usuário, selecione **Instrução**→**Exibir estatísticas**→**Funções**. Selecione uma ou mais entradas da lista exibida na janela do Funções e clique em **OK**. Uma janela Estatísticas da Função será aberta para cada uma das funções selecionadas.

Obtendo Estatísticas para Áreas de Tabela

Para exibir estatísticas de catálogo para áreas de tabela, selecione **Instrução**→**Exibir estatísticas**→**Áreas de Tabelas**. Selecione uma ou mais entradas da lista exibida na janela Áreas de Tabela e clique em **OK**. Uma janela da Estatísticas da Área de Tabela será aberta para cada uma das áreas de tabela selecionadas.

Obtendo Estatísticas para Colunas em uma Instrução SQL

Para obter estatísticas para as colunas de referência em uma instrução SQL:

1. Dê um clique duplo em uma tabela no gráfico do plano de acesso. A janela Estatísticas da Tabela é aberta.
2. Clique no botão **Colunas de Referência**. A janela Colunas de Referência será aberta, listando as colunas na tabela.
3. Selecione uma ou mais colunas na lista, e clique em **OK**. Uma janela Estatísticas da Coluna de Referência será aberta para cada coluna selecionada.

Obtendo Informações sobre Parâmetros de Configuração e Opções de União

Para exibir informações sobre parâmetros de configuração e opções de união (parâmetros de otimização), selecione **Instrução→Exibir parâmetros de otimização** a partir da janela Gráfico do Plano de Acesso. A janela Parâmetros de Otimização será aberta, exibindo informações sobre os valores de parâmetro em efetivos no momento em que o instantâneo foi criado, além dos valores atuais.

Alterando a Aparência de um Gráfico

Para alterar várias características de como um gráfico é exibido:

1. A partir da janela Gráfico do Plano de Acesso, selecione **Exibir→Definições**. O bloco de notas Definições do Gráfico do Plano de Acesso será aberto.
2. Para alterar a cor do plano de fundo, selecione a guia Gráfico.
3. Para alterar a cor dos vários operadores, utilize as guias Básico, Estender, Atualizar e Vários.
4. Para alterar a cor da tabela, índice ou nós de função da tabela, selecione a guia Operando.
5. Para especificar qual tipo de informação será mostrada nos nós de operadores (tipo de custo ou cardinalidade, que é o número estimado de linha das retornadas até o momento), selecione a guia Operador.
6. Para especificar se os nomes dos esquemas ou IDs de usuários são mostrados nos nós de tabela, selecione a guia Operando.
7. Para especificar se os nós são mostrados de maneira bidimensional ou tridimensional, selecione a guia Nó.
8. Para atualizar o gráfico com as opções selecionadas e salvar as definições, clique em **Aplicar**.

O que Vem a Seguir

Se estiver trabalhando em um ambiente de banco de dados de partição única, siga para “Lição 3. Melhorando um Plano de Acesso em um Ambiente de Banco de Dados de Partição Única” na página 13, onde você irá descobrir como diferentes atividades de ajuste podem alterar e melhorar um plano de acesso.

Se estiver trabalhando em um ambiente de banco de dados particionado, siga para “Executando uma Consulta sem Índices e Sem Estatísticas” na página 14, onde você irá aprender como diferentes atividades de ajuste podem alterar e melhorar um plano de acesso.

Lição 3. Melhorando um Plano de Acesso em um Ambiente de Banco de Dados de Partição Única

Nesta lição, você irá descobrir como o plano de acesso e janelas relacionadas para a consulta básica são alterados quando você executa várias atividades de ajuste. Utilizando uma série de exemplos, acompanhados por ilustrações, você irá descobrir como o custo total estimado do plano de acesso, até mesmo de uma consulta simples, pode ser melhorado utilizando o comando **runstats** e incluindo índices apropriados.

Conforme você ganha experiência com o Visual Explain, será necessário descobrir outras maneiras de ajustar as consultas.

Trabalhando com Gráficos do Plano de Acesso

Utilizando os quatro instantâneos de explicação de exemplo, você irá descobrir como ajustes são uma parte importante do desempenho do banco de dados.

As consultas associadas aos instantâneos de explicação estão numeradas de 1 a 4. Cada consulta utiliza a mesma instrução SQL (descrita na Lição 1):

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Mas cada iteração da consulta utiliza mais técnicas de ajuste do que a execução anterior. Por exemplo, Consulta 1 não possui ajuste de desempenho, enquanto a Consulta 4 possui o número maior. As diferenças nas consultas estão descritas abaixo:

Consulta 1

Executando uma consulta sem índices e sem estatísticas

Consulta 2

Coletando estatísticas atuais para as tabelas e índices em uma consulta

Consulta 3

Criando índices em colunas utilizadas para unir tabelas em um consulta

Consulta 4

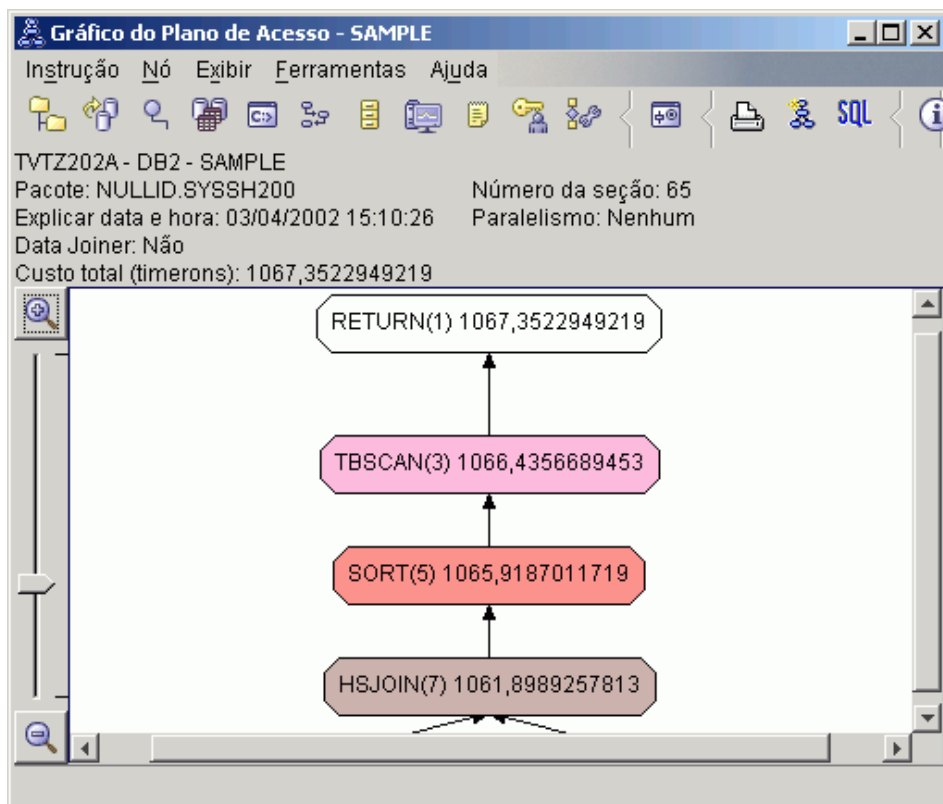
Criando índices adicionais em colunas de tabela

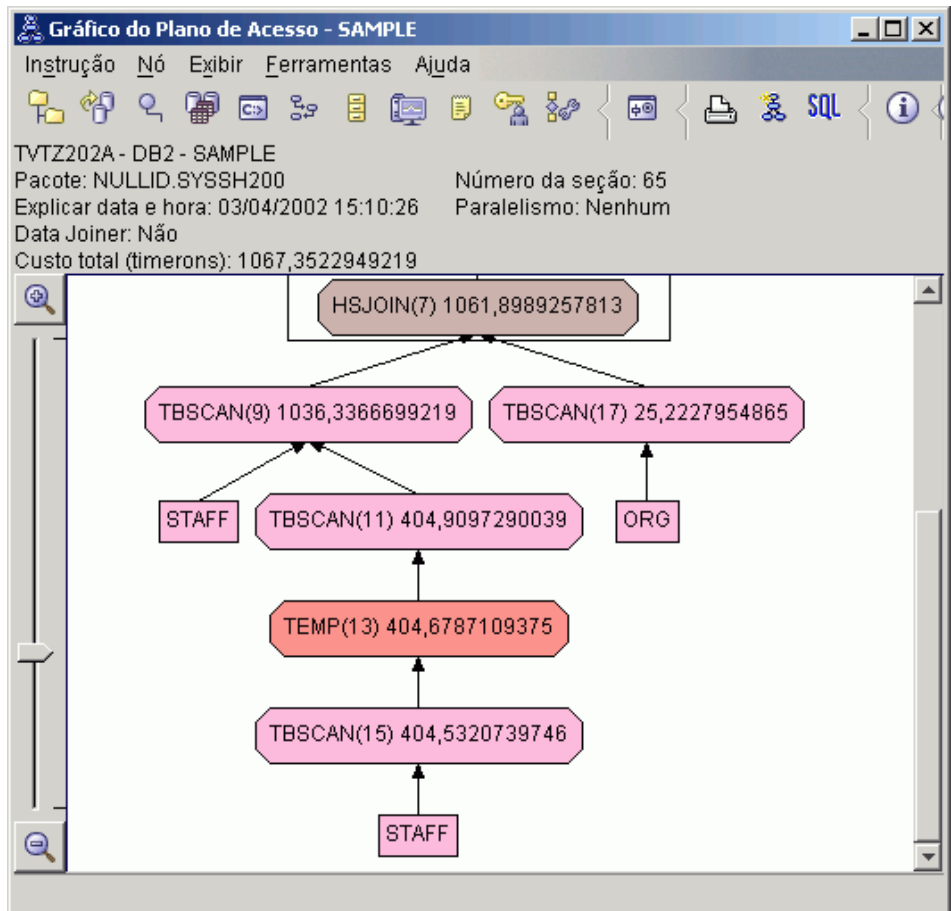
Executando uma Consulta sem Índices e Sem Estatísticas

Neste exemplo, o plano de acesso foi criado para a consulta SQL sem índices e sem estatísticas.

Para exibir o gráfico do plano de acesso para esta consulta (Consulta 1):

1. No Centro de Controle, expanda a árvore de objeto até encontrar o banco de dados SAMPLE.
2. Dê um clique duplo no banco de dados e selecione **Mostrar histórico de instruções explicadas** a partir do menu popup. A janela Histórico de Instruções Explicadas será aberta.
3. Dê um clique duplo na entrada identificada como Consulta Número 1 (pode ser necessário deslocar para a direita para localizar a coluna **Número da Consulta**). A janela Gráfico do Plano de Acesso da instrução será aberta.





Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

1. As estatísticas atuais existem para cada tabela na consulta?

Para verificar se as estatísticas atuais existem para cada tabela na consulta, dê um clique duplo em cada nó de tabela no gráfico do plano de acesso. Na janela Estatísticas da Tabela aberta, a linha **STATS_TIME** sob a coluna **Explicado** estão as palavras "Estatísticas não atualizadas" se nenhuma estatística foi coletada no momento em que o instantâneo foi criado.

Se as estatísticas atuais não existirem, o otimizador utiliza estatísticas padrão, que poderão ser diferentes das estatísticas atuais. As estatísticas padrão são identificadas pela palavra "padrão" sob a coluna **Explicado** na janela Estatísticas da Tabela.

De acordo com as informações na janela Estatísticas da Tabela para a tabela ORG, o otimizador utilizou estatísticas padrão (conforme indicado

próximo aos valores explicados). As estatísticas padrão foram utilizadas pois as estatísticas atuais não estavam disponíveis quando o instantâneo foi criado (conforme indicado na linha **STATS_TIME**).

Estatísticas	Explicado	Atual
CREATE_TIME	03/04/2002 15:05:03	
STATS_TIME	Estatísticas não atualizadas	Estatísticas não atualizadas
CARD	55(padão)	
NPAGES	1(padão)	
FPAGES	1(padão)	
COLCOUNT	5(padão)	
OVERFLOW	0(padão)	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLÁTIL	Não(padão)	Não

- Este plano de acesso utiliza os métodos mais eficientes de acesso de dados?

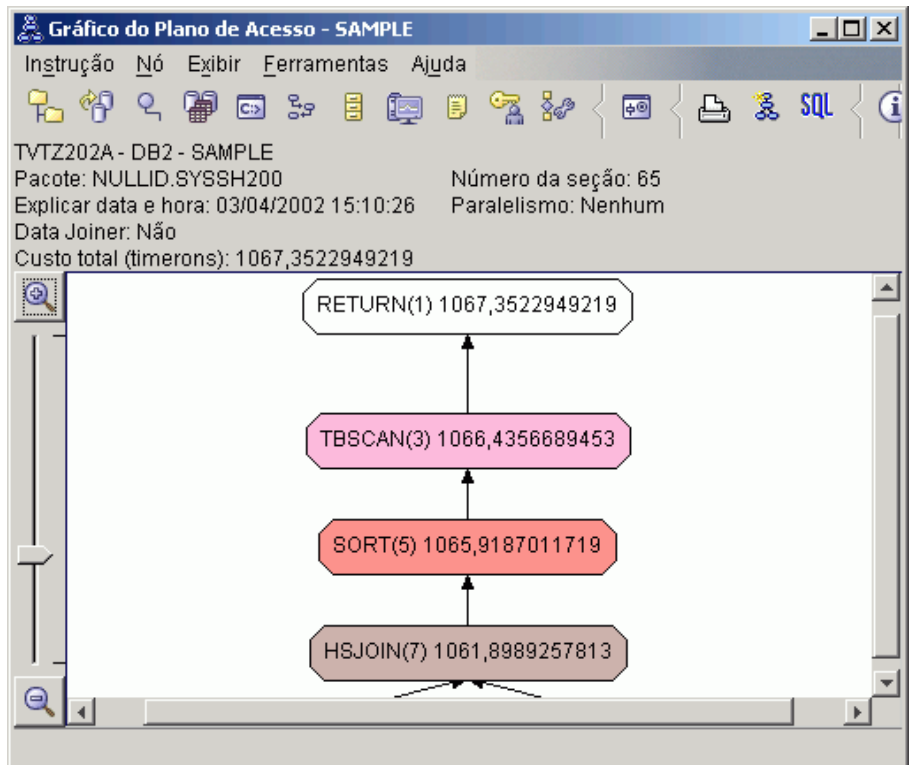
Este plano de acesso contém as varreduras de tabela e não varreduras de índice. Varreduras de tabela são mostradas como octágonos e são rotuladas como TBSCAN. Caso as varreduras de índice tivessem sido utilizadas, elas apareceriam como losangos e seriam rotuladas como IXSCAN. O uso de um índice criado para uma tabela tem um melhor custo do que uma varredura de tabela caso pequenas quantidades de dados estejam sendo extraídas.

- Quão efetivo é este plano de acesso?

Você poderá determinar a eficiência de um plano de acesso caso ele seja baseado em estatísticas atuais. Como o otimizador utilizou estatísticas padrão no plano de acesso, você não poderá determinar o quão efetivo é o plano.

Em geral, você deve tomar nota do custo total estimado para o plano de acesso com planos de acesso revisados. O custo listado em cada nó é cumulativo, desde a primeira etapa de sua consulta até (e incluindo) o nó.

Na janela Gráfico do Plano de Acesso o custo total é aproximadamente 1,067 timerons, mostrados em **RETURN (1)** na parte superior do gráfico. O custo total estimado também é mostrado na parte superior da janela.



4. O que vem a seguir?

A Consulta 2 aponta para um plano de acesso para a consulta básica após **runstats** ter sido executado. Utilizar o comando **runstats** fornece ao otimizador as estatísticas atuais em todas as tabelas acessadas pela consulta.

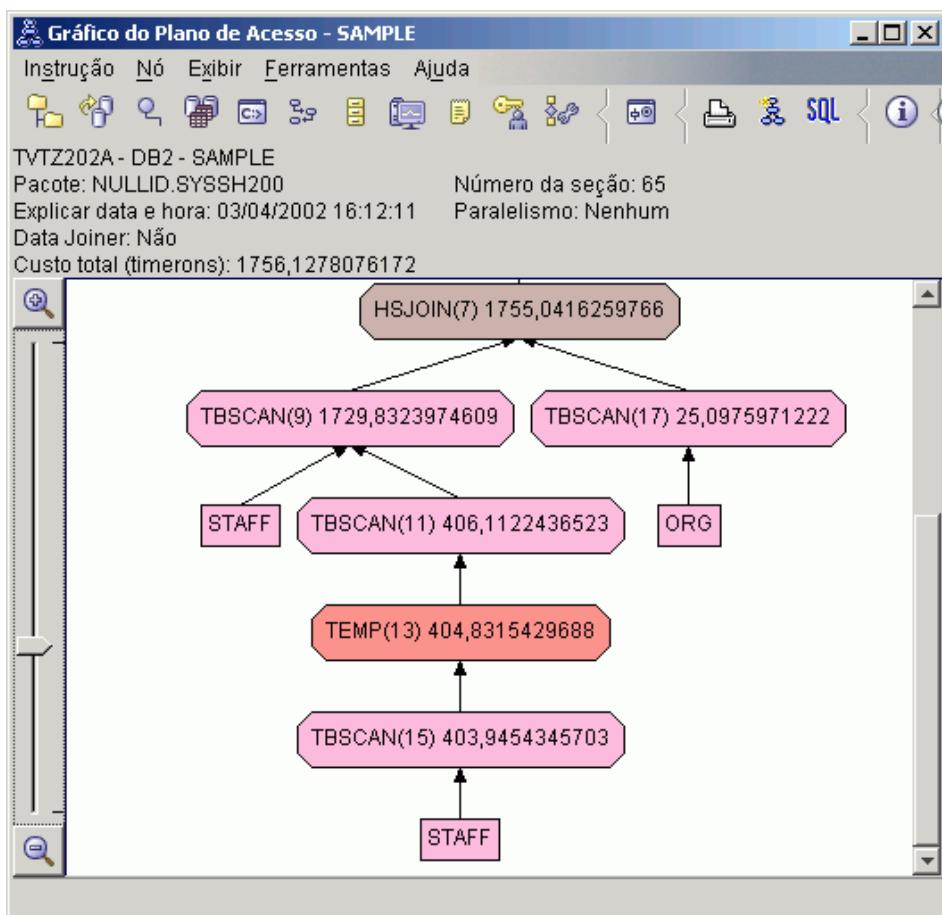
Coletando Estatísticas Atuais para as Tabelas e Índices Utilizando **runstats**

Este exemplo é baseado no plano de acesso descrito na Consulta 1 coletando as estatísticas atuais com o comando **runstats**.

É altamente recomendado que você utilize o comando **runstats** para coletar as estatísticas atuais em tabelas e índices, especialmente se atividades de atualizações significativas ocorrerem ou se novos índices foram criados desde a última vez em que o comando **runstats** foi executado. Isto fornece ao otimizador as informações mais precisas com as quais você poderá determinar o melhor plano de acesso. Se as estatísticas atuais não estiverem disponíveis, o otimizador poderá escolher um plano de acesso ineficiente baseado em estatísticas padrão imprecisas.

Certifique-se de utilizar o **runstats** após fazer suas atualizações de tabela; caso contrário, a tabela poderá aparentar estar vazia para o otimizador. Este problema é evidente se a cardinalidade na janela Detalhes do operador for igual a zero. Neste caso, conclua suas atualizações de tabela, execute o comando **runstats** novamente e recrie os instantâneos para tabelas afetadas.

Para exibir o gráfico do plano de acesso para esta consulta (Consulta 2): na janela Histórico de Instruções Explicadas, dê um clique duplo na entrada identificada como Consulta Número 2. A janela Gráfico do Plano de Acesso para esta execução da instrução será aberta.



Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

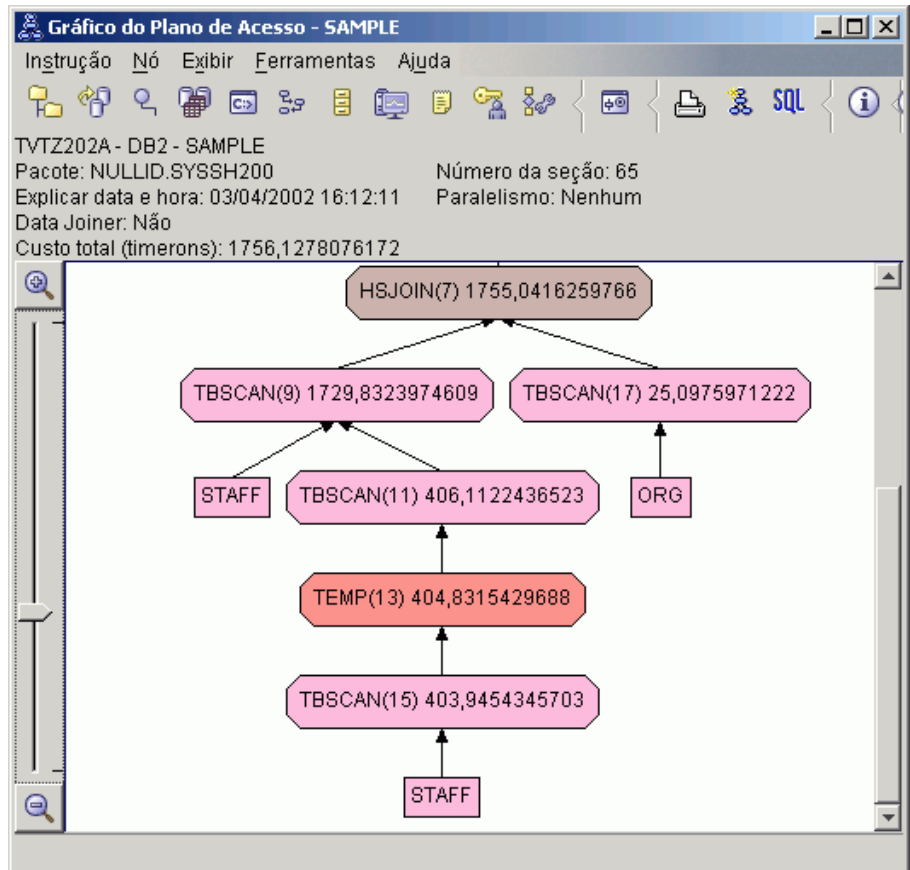
1. As estatísticas atuais existem para cada tabela na consulta?

A janela Estatísticas da Tabela para a tabela ORG mostra que o otimizador utilizou estatísticas atuais (o valor **STATS_TIME** é o tempo real em que as estatísticas foram coletadas). A precisão das estatísticas depende da existência de alterações significativas para o conteúdo das tabelas desde a execução do comando **runstats**.

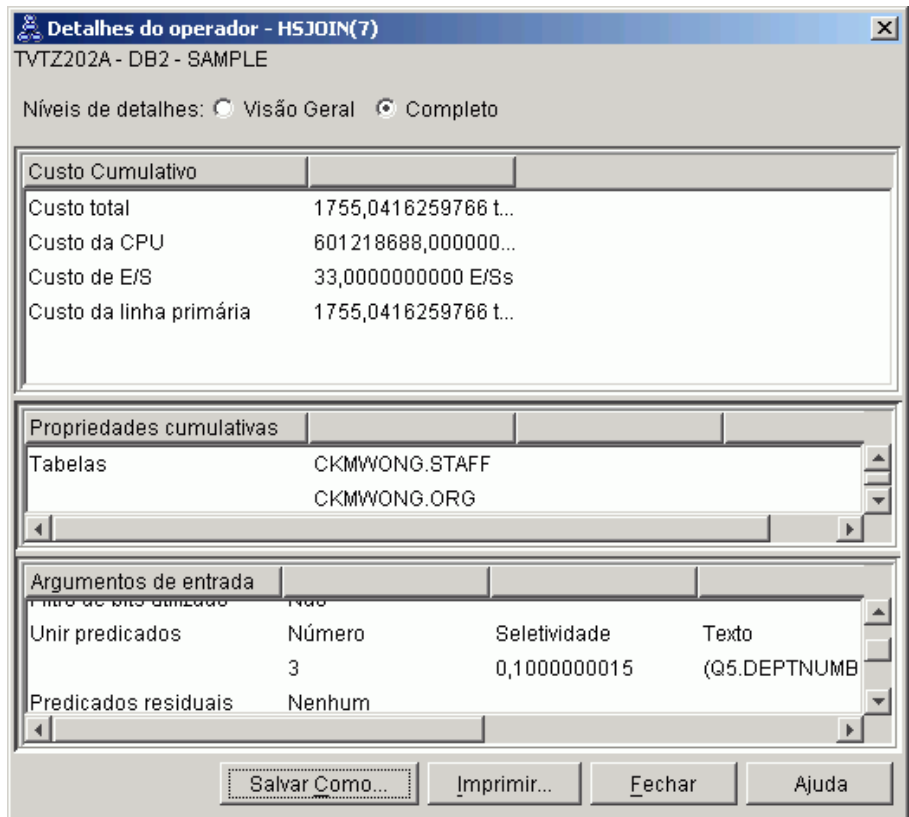
Estatísticas	Explicado	Atual
CREATE_TIME	12/07/2002 22:09:14	12/07/2002 22:09:14
STATS_TIME	13/07/2002 19:56:29	13/07/2002 19:56:29
CARD	8	8
NPAGES	1	1
FPAGES	1	1
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLÁTIL	Não	Não

- Este plano de acesso utiliza os métodos mais eficientes de acesso de dados?

Como na Consulta 1, o plano de acesso na Consulta 2 utiliza varreduras de tabela (TBSCAN) e não varreduras de tabela (IXSCAN). Embora existam estatísticas atuais, uma varredura de índice não foi feita porque não houveram índices nas colunas utilizadas pela consulta. Uma maneira de melhorar a consulta seria fornecer ao otimizador os índices e colunas que são utilizados para unir tabelas (isto é, em colunas utilizadas para unir predicados). Neste exemplo, esta é a primeira junção de varredura de combinação: HSJOIN (7).



Na janela Detalhes do operador para o operador HSJOIN (7), procure na seção **Unir predicados** em **Argumentos de Entrada**. As colunas utilizadas nessa operação de união são listadas sob a coluna **Texto**. Neste exemplo, essas colunas são DEPTNUMB e DEPT.



3. Quão efetivo é este plano de acesso?

Planos de acesso baseados em estatísticas atualizadas sempre produzem um custo estimado realista (medido em timerons). Como o custo estimado na Consulta 1 foi baseado em estatísticas padrão, não é possível comparar o custo de dois gráficos do plano de acesso para determinar qual é mais efetivo. Se o custo é maior ou menor não é relevante. Será necessário comparar o custo dos planos de acesso baseados em estatísticas atuais para obter uma medida válida de eficiência.

4. O que vem a seguir?

A Consulta 3 observa os efeitos da inclusão de índices nas colunas DEPTNUMB e DEPT. A inclusão de índices nas colunas utilizadas em Unir Predicados podem melhorar o desempenho.

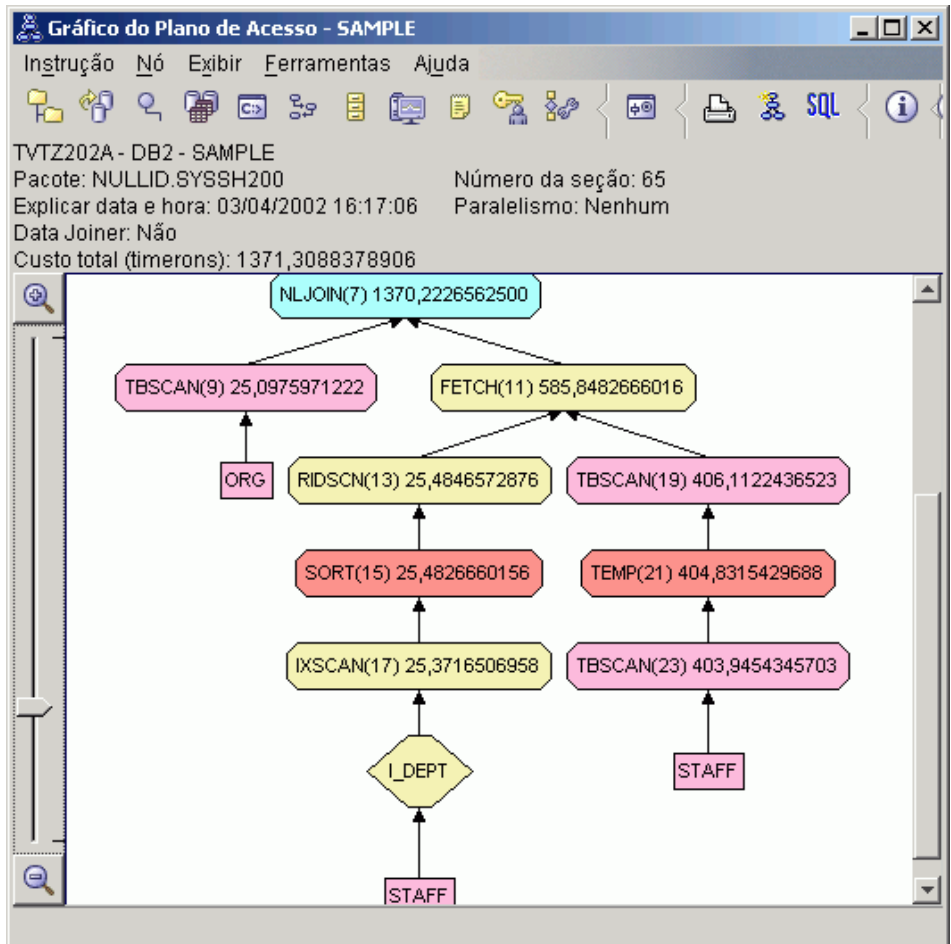
Criando Índices em Colunas Utilizadas para Unir Tabelas em uma Consulta

Este exemplo é baseado no plano de acesso descrito na Consulta 2 ao criar índices na coluna DEPT da tabela STAFF e na coluna DEPTNUMB da tabela ORG.

Nota: Na Versão 8, os índices recomendados podem ser criados utilizando o assistente Desempenho de Carga de Trabalho.

Para exibir o gráfico do plano de acesso para esta consulta (Consulta 3): na janela Histórico de Instruções Explicadas, dê um clique duplo na entrada identificada como Consulta Número 3. A janela Gráfico do Plano de Acesso para esta execução da instrução será aberta.

Nota: Embora um índice tenha sido criado para DEPTNUM, o otimizador não a utilizou.

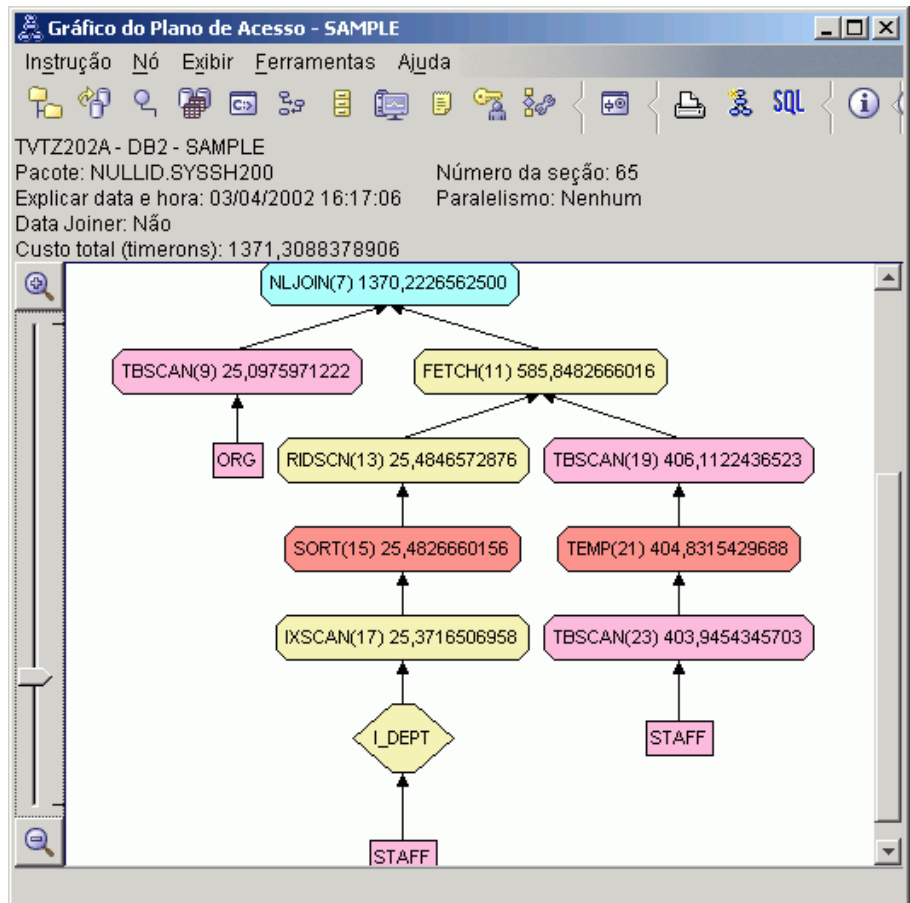


Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

1. O que mudou no plano de acesso com índices?

Uma união de loop aninhada, NLJOIN (7), substituiu a união de varredura de combinação HSJOIN (7) utilizada na Consulta 2. A utilização de uma união de loop aninhada resultou em um custo estimado inferior do que uma união de varredura de combinação, pois este tipo de união não requer nenhuma ordenação ou tabelas temporárias.

Um nó com formato de losango, **I_DEPT**, foi incluído logo acima da tabela STAFF. Este nó representa o índice criado em DEPT, e mostra que o otimizador utilizou uma varredura de índice em vez de uma varredura de tabela para determinar quais linhas devem ser recuperadas.



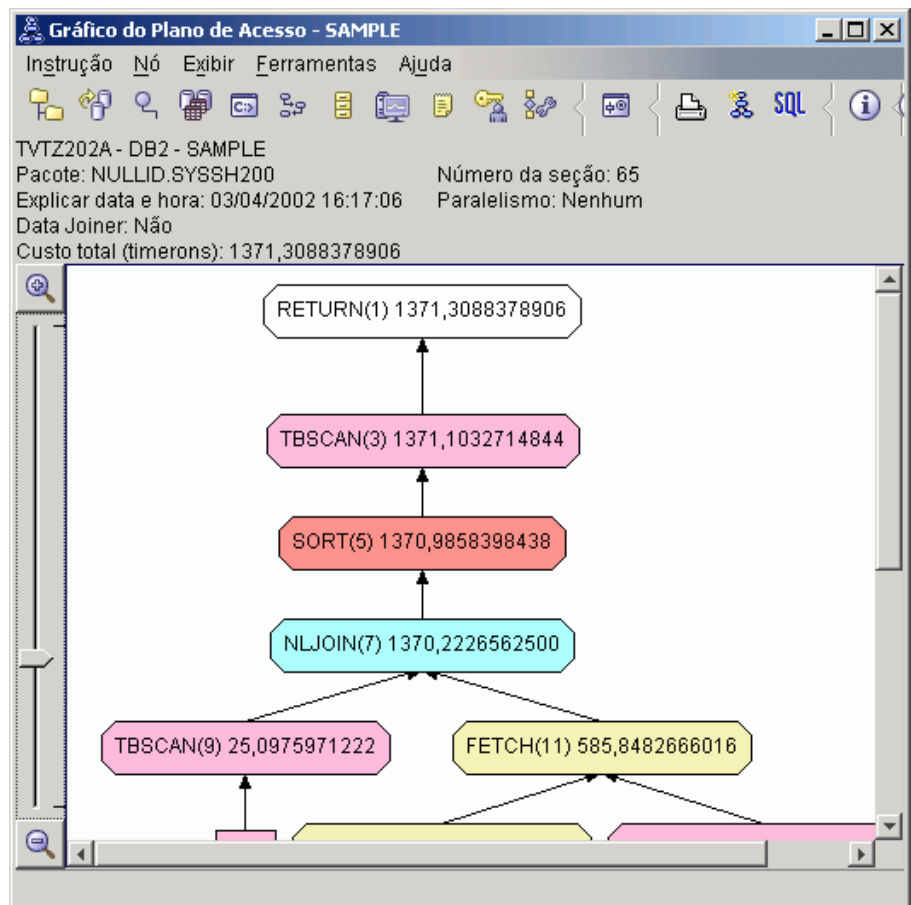
Nesta parte do gráfico do plano de acesso, observe que um novo índice (**I_DEPT**) foi criado na coluna DEPT e IXSCAN (17) foi utilizado para acessar a tabela STAFF. Na Consulta 2, uma varredura de tabela foi utilizada para acessar a tabela STAFF.

2. Este plano de acesso utiliza os métodos mais eficientes de acesso de dados?

Como resultado da inclusão de índices, um nó IXSCAN IXSCAN (17), foi utilizado para acessar a tabela STAFF. A Consulta 2 não tinha um índice; conseqüentemente, uma varredura de tabela foi utilizada nesse exemplo.

O nó FETCH, FETCH (11), mostra que além de utilizar uma varredura de índice para recuperar a coluna DEPT, o otimizador recuperou colunas adicionais da tabela STAFF, utilizando índice como um indicador. Neste caso, a combinação de uma varredura e busca de índice são calculadas para ter um custo menor do que a varredura completa de tabela utilizada em planos de acesso anteriores.

Nota: O nó da tabela STAFF aparece duas vezes, para mostrar seu relacionamento com o índice de DEPT e com a operação FETCH.

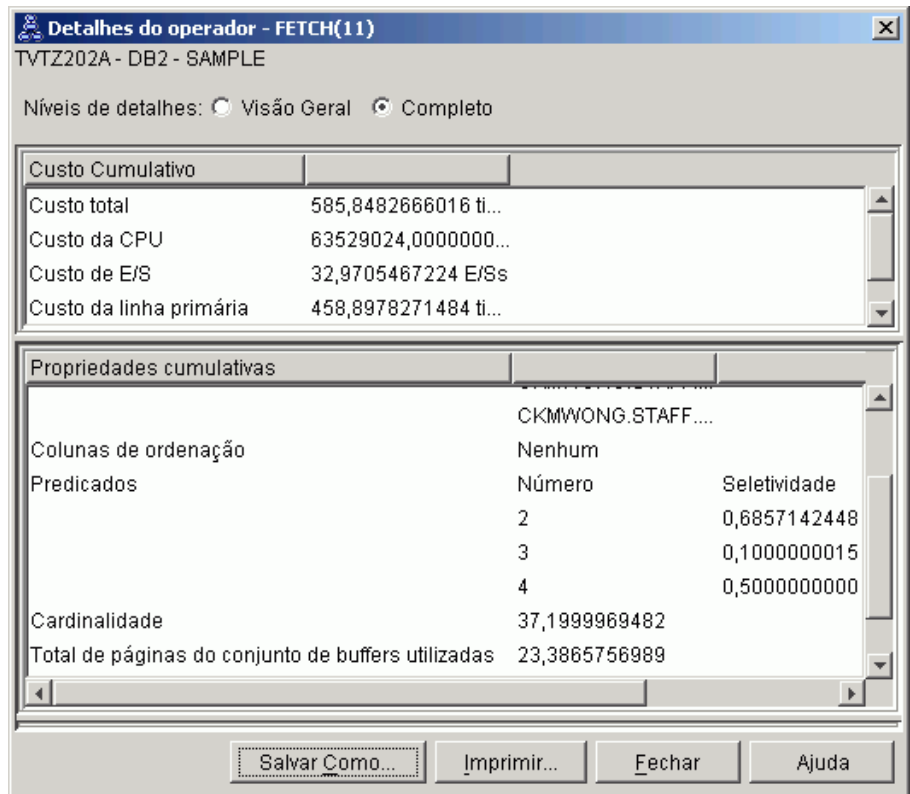


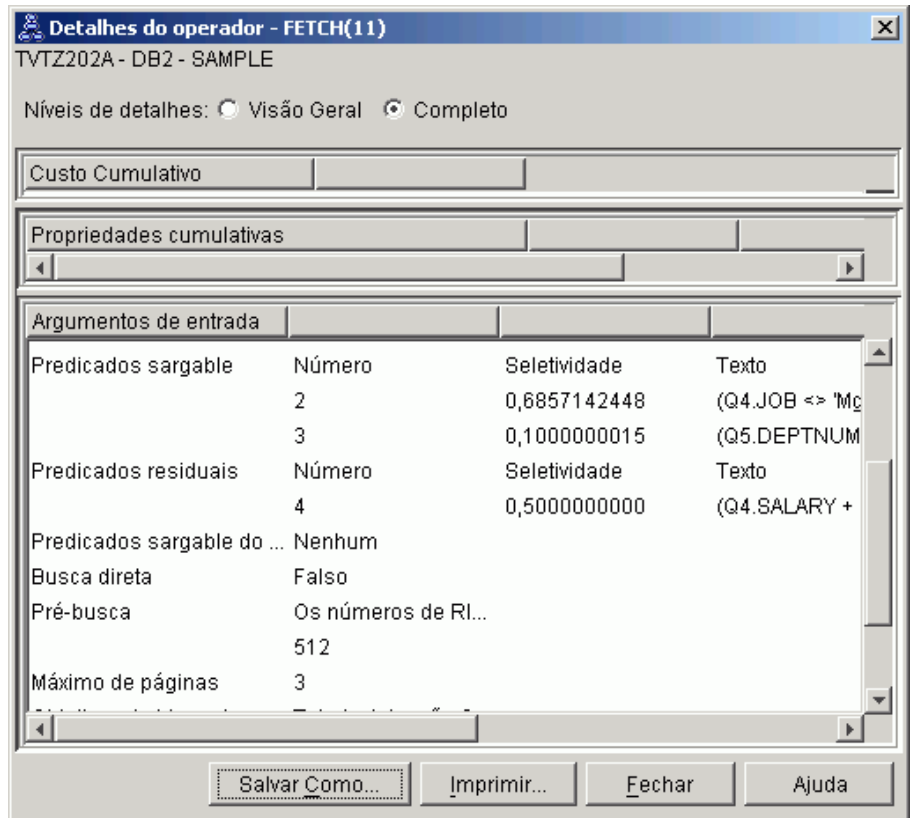
O plano de acesso para esta consulta mostra o efeito de criação de índices em colunas envolvidas em Unir Predicados. Índices também aumentam a aplicação dos predicados locais. Vamos olhar os predicados locais para

cada tabela nesta consulta para ver como a inclusão de índices em colunas referenciadas em predicados locais pode afetar o plano de acesso.

Na janela Detalhes do operador do operador FETCH (11), observe as colunas sob **Propriedades cumulativas**. A coluna utilizada no predicado para esta operação de busca é JOB, conforme mostrado na seção Predicados.

Nota: A seletividade deste predicado é .69. Isto significa que com este predicado, 69% das linhas serão selecionadas para processamento posterior.





A janela Detalhes do operador do operador FETCH (11) mostra as colunas sendo utilizadas nesta operação. Você poderá ver que DEPTNAME está listado na primeira linha ao lado de **Colunas recuperadas** sob **Argumentos de Entrada**.

3. Quão efetivo é este plano de acesso?

Este plano de acesso tem um melhor custo do que o do exemplo anterior. O custo cumulativo foi reduzido de aproximadamente 1.755 timerons na Consulta 2 para aproximadamente 959 timerons na Consulta 3.

No entanto, o plano de acesso da Consulta 3 mostra uma varredura de índice IXSCAN (17) e FETCH (11) para a tabela STAFF. Enquanto uma varredura de índice combinada com uma operação de busca custa menos do que uma varredura de tabela completa, isto significa que para cada linha recuperada, a tabela e o índice serão acessados uma vez. Vamos tentar reduzir este acesso duplo na tabela STAFF.

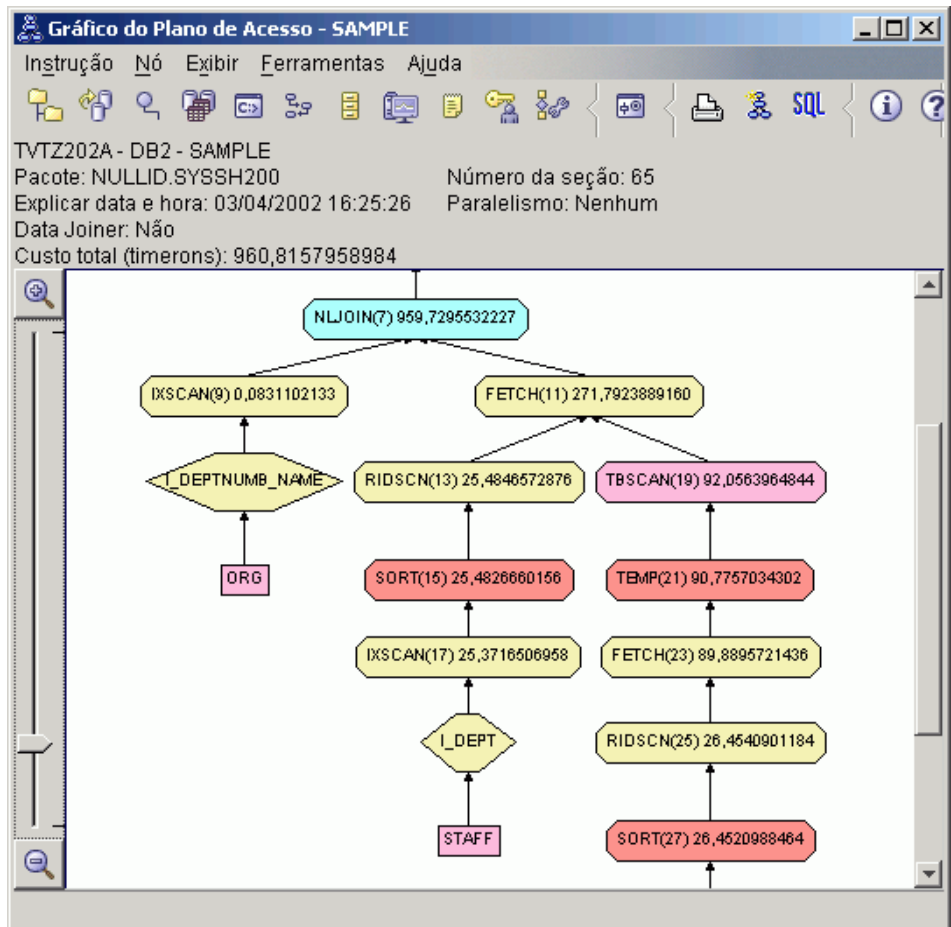
4. O que vem a seguir?

A Consulta 4 reduz a busca e varredura de índice para uma única varredura de índice sem uma busca. A criação de índices adicionais pode reduzir o custo estimado para o plano de acesso.

Criando Índices Adicionais em Colunas de Tabela

Este exemplo é baseado no plano de acesso descrito na Consulta 3, criando um índice na coluna JOB da tabela STAFF, e incluindo DEPTNAME no índice existente na tabela ORG. (Incluir um índice separado pode causar um acesso adicional).

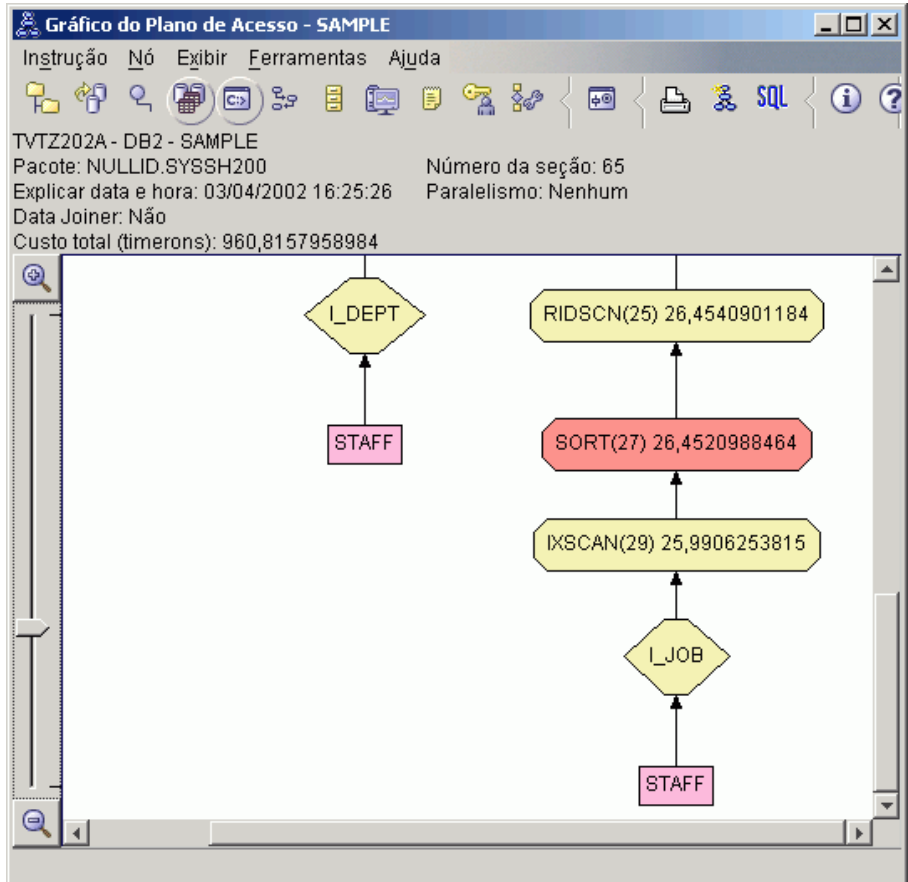
Para exibir o plano de acesso para esta consulta (Consulta 4): na janela Histórico de Instruções Explicadas, dê um clique duplo na entrada identificada como Consulta Número 4. A janela Gráfico do Plano de Acesso para esta execução da instrução será aberta.



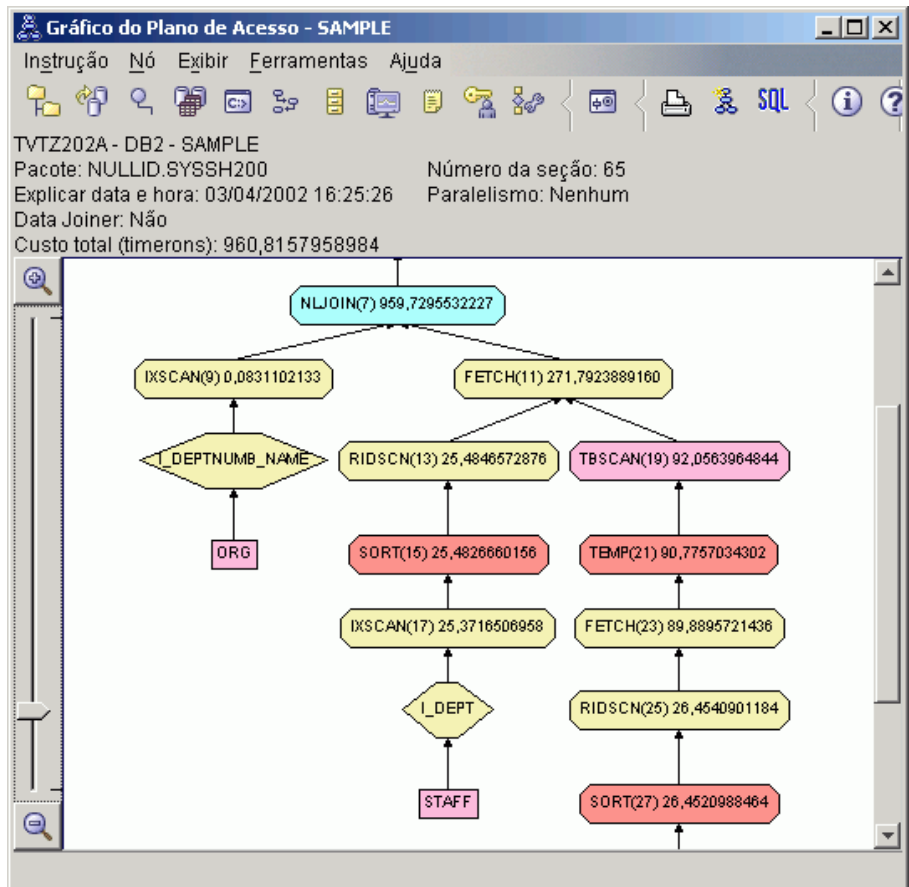
Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

1. O que mudou neste plano de acesso como resultado da criação de índices adicionais?

O otimizador utilizou o índice criado na coluna JOB na tabela STAFF (representado por um losango rotulado como I_JOB) para refinar posteriormente este plano de acesso.



Na parte intermediária do gráfico do plano de acesso, observe que para a tabela ORG, a varredura de índice anterior e busca foram alteradas para apenas um IXSCAN (9) de varredura de índice. A Inclusão da coluna DEPTNAME no índice na parte superior da tabela ORG permitiu que o otimizador elimine o acesso adicional envolvendo a busca.



2. Quão efetivo é este plano de acesso?

Este plano de acesso tem um melhor custo do que o do exemplo anterior. O custo cumulativo foi reduzido de aproximadamente 1.370 timerons na Consulta 3 para aproximadamente 959 timerons na Consulta 4.

O que Vem a Seguir

Consulte o *Administration Guide* para encontrar informações detalhadas sobre procedimentos adicionais que você poderá realizar para melhorar o desempenho. Você então poderá retornar ao Visual Explain para avaliar o impacto de suas ações.

Lição 4. Melhorando o Plano de Acesso em um Ambiente de Banco de Dados Particionado

Nesta lição, você irá descobrir como o plano de acesso e janelas relacionadas para a consulta básica são alterados quando você executa várias atividades de ajuste. Utilizando uma série de exemplos, acompanhados por ilustrações, você irá descobrir como o custo total estimado do plano de acesso, até mesmo uma consulta simples, pode ser melhorado utilizando o comando **runstats** e incluindo índices apropriados.

Conforme você ganha experiência com o Visual Explain, você descobrirá outras maneiras de ajustar as consultas.

Trabalhando com Gráficos do Plano de Acesso

Utilizando os quatro instantâneos de explicação de exemplo, você irá descobrir como os ajustes são uma parte importante do desempenho do banco de dados.

As consultas associadas aos instantâneos de explicação estão numeradas de 1 a 4. Cada consulta utiliza a mesma instrução SQL (descrita na Lição 1):

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Mas cada iteração da consulta utiliza mais técnicas de ajuste do que a execução anterior. Por exemplo, Consulta 1 não possui ajuste de desempenho, enquanto a Consulta 4 possui o número maior. As diferenças nas consultas estão descritas abaixo:

Consulta 1

Executando uma consulta sem índices e sem estatísticas

Consulta 2

Coletando estatísticas atuais para as tabelas e índices em uma consulta

Consulta 3

Criando índices em colunas utilizadas para unir tabelas em um consulta

Consulta 4

Criando índices adicionais em colunas de tabela

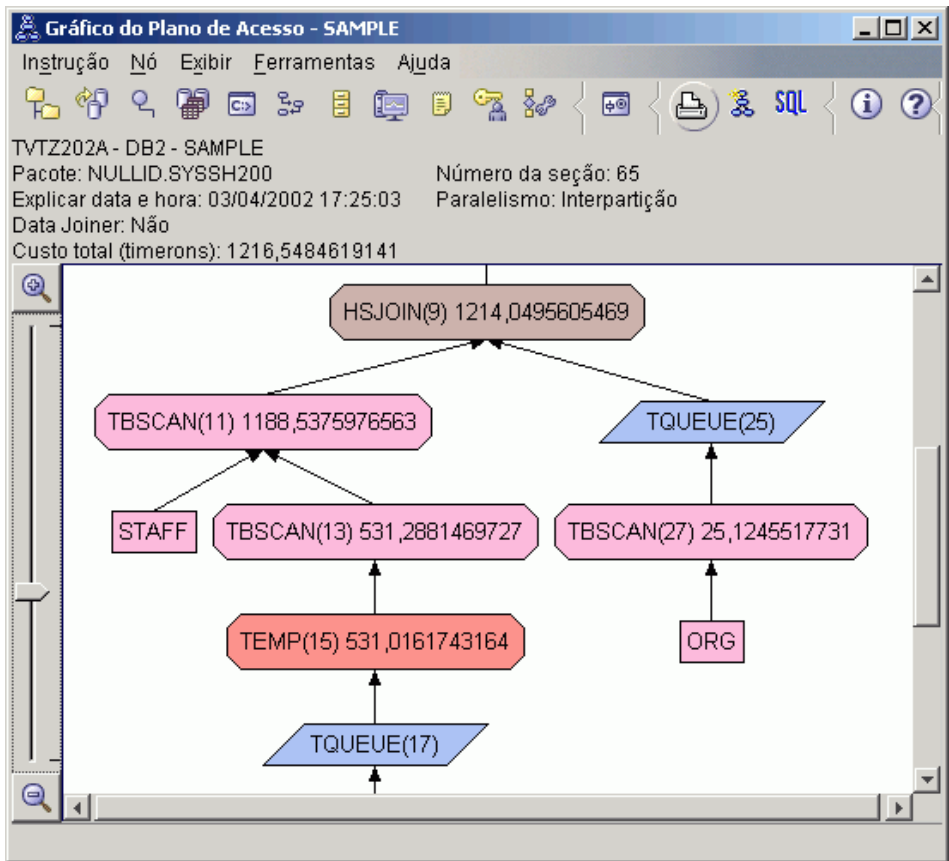
Estes exemplos foram produzidos em uma máquina RS/6000 SP com 7 nós físicos utilizando paralelismo entre partições.

Executando uma Consulta sem Índices e Sem Estatísticas

Neste exemplo, o plano de acesso foi criado para a consulta SQL sem índices e sem estatísticas.

Para exibir o gráfico do plano de acesso para esta consulta (Consulta 1):

1. No Centro de Controle, expanda a árvore de objeto até encontrar o banco de dados SAMPLE.
2. Clique com o botão direito no banco de dados e selecione **Mostrar histórico de instruções explicadas** no menu popup. A janela Histórico de Instruções Explicadas será aberta.
3. Dê um clique duplo nas entradas identificadas como Consulta Número 1 (pode ser necessário deslocar para a direita para localizar a coluna **Número da Consulta**). A janela Gráfico do Plano de Acesso da instrução será aberto.



Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

1. As estatísticas atuais existem para cada tabela na consulta?

Para verificar se as estatísticas atuais existem para cada tabela na consulta, dê um clique duplo em cada nó de tabela no gráfico do plano de acesso. Na janela Estatísticas da Tabela correspondente que será aberta, a linha **STATS_TIME** sob a coluna **Explicado** existem as palavras "Estatísticas não atualizadas", indicando que nenhuma estatística foi coletada no momento em que o instantâneo foi criado.

Se as estatísticas atuais não existirem, o otimizador utiliza estatísticas padrão, que poderão ser diferentes das estatísticas atuais. As estatísticas padrão são identificadas pela palavra "padrão" sob a coluna **Explicado** na janela Estatísticas da Tabela.

De acordo com as informações na janela Estatísticas da Tabela para a tabela ORG, o otimizador utilizou estatísticas padrão (conforme indicado próximo aos valores explicados). As estatísticas padrão foram utilizadas

pois as estatísticas atuais não estavam disponíveis quando o instantâneo foi criado (conforme indicado na linha **STATS_TIME**).

Estatísticas	Explicado	Atual
CREATE_TIME	26/03/2002 13:35:42	
STATS_TIME	Estatísticas não atualizadas	Estatísticas não atualizadas
CARD	55(padrão)	
NPAGES	1(padrão)	
FPAGES	1(padrão)	
COLCOUNT	5(padrão)	
OVERFLOW	0(padrão)	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLÁTIL	Não(padrão)	Não

2. Este plano de acesso utiliza os métodos mais eficientes de acesso de dados?

Este plano de acesso contém as varreduras de tabela e não varreduras de índice. Varreduras de tabela são mostradas como octágonos e são rotuladas como TBSCAN. Se as varreduras de índice foram utilizadas, elas seriam exibidas como losangos e seriam rotuladas como IXSCAN. O uso de um índice criado para uma tabela tem um melhor custo do que uma varredura de tabela caso pequenas quantidades de dados estejam sendo extraídas.

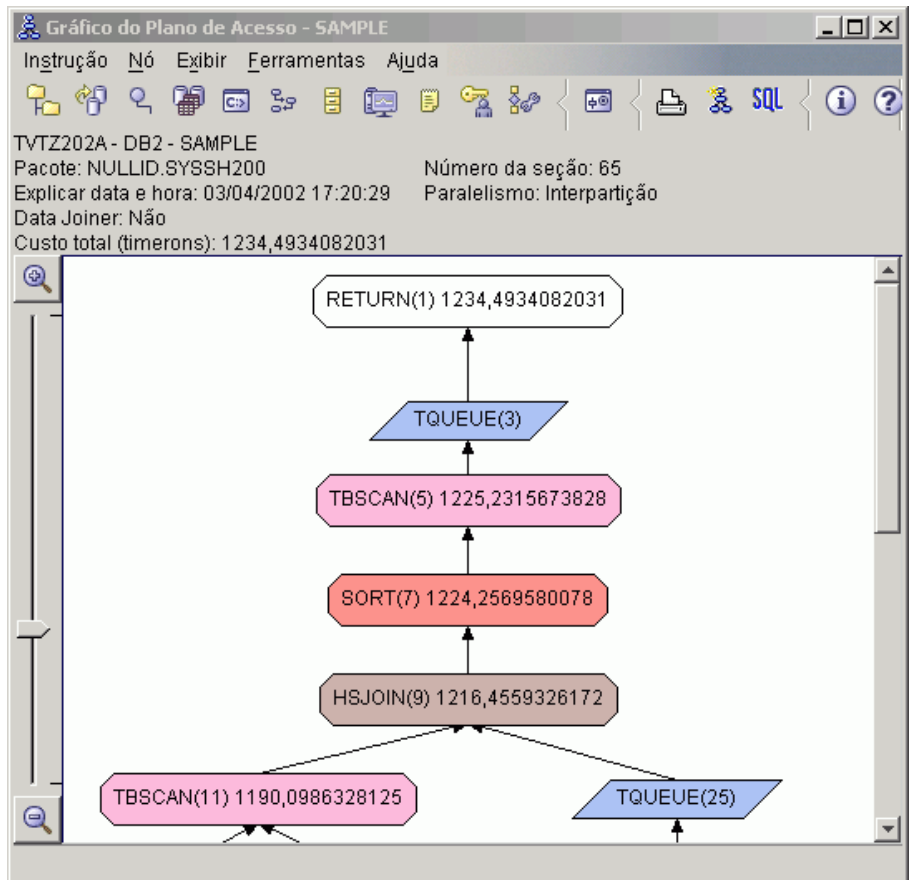
3. Quão efetivo é este plano de acesso?

Você poderá determinar a eficiência de um plano de acesso caso ele seja baseado em estatísticas atuais. Como o otimizador utilizou estatísticas padrão no plano de acesso, você não poderá determinar o quão efetivo é o plano.

Em geral, você deve tomar nota do custo total estimado para o plano de acesso com planos de acesso revisados. O custo listado em cada nó é cumulativo, desde a primeira etapa de sua consulta até (e incluindo) o nó.

Nota: Para bancos de dados particionados, este é o custo cumulativo para o nó que utiliza o maior número de recursos.

Na janela Gráfico do Plano de Acesso, o custo total é aproximadamente 1.234 timerons, mostrados em **RETURN (1)** na parte superior do gráfico. O custo total estimado também é mostrado na parte superior da janela.



4. O que vem a seguir?

A Consulta 2 aponta para um plano de acesso para a consulta básica após **runstats** ter sido executado. Utilizar o comando **runstats** fornece ao otimizador as estatísticas atuais em todas as tabelas acessadas pela consulta.

Coletando Estatísticas Atuais para as Tabelas e Índices Utilizando **runstats**

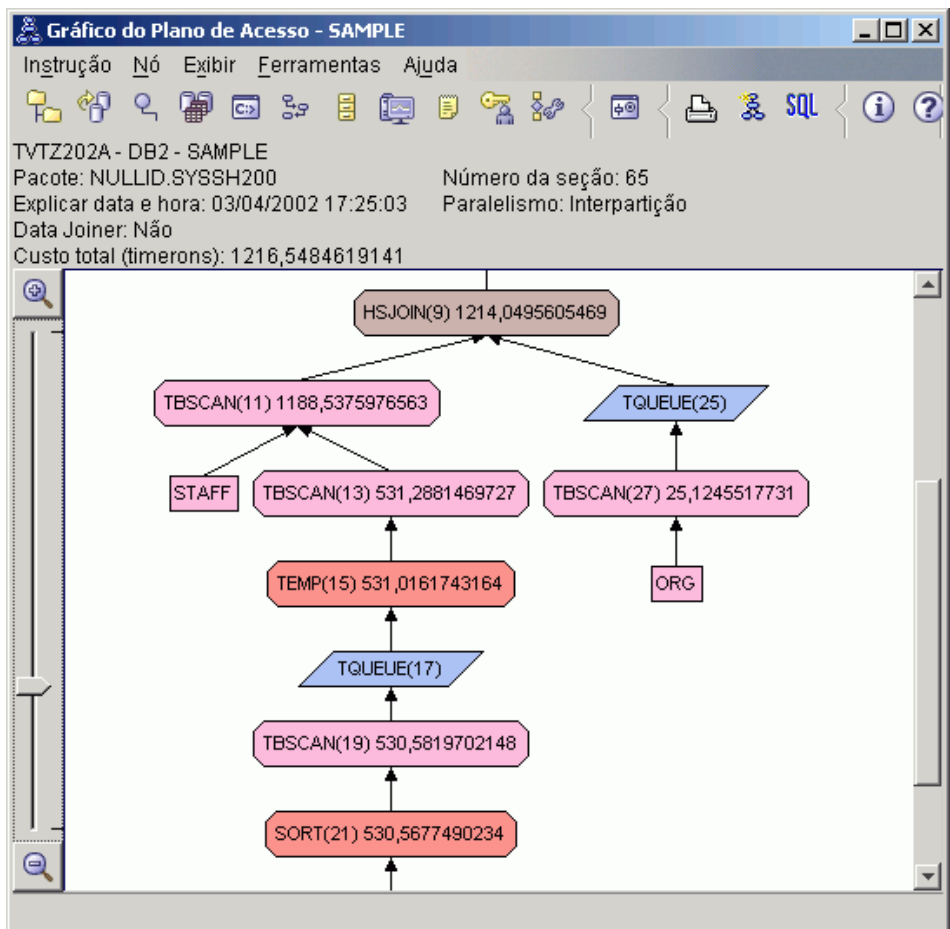
Este exemplo é baseado no plano de acesso descrito na Consulta 1 coletando as estatísticas atuais com o comando **runstats**.

É altamente recomendado que você utilize o comando **runstats** para coletar as estatísticas atuais em tabelas e índices, especialmente se atividades de atualizações significativas ocorrerem ou se novos índices foram criados desde a última vez em que o comando **runstats** foi executado. Isto fornece ao otimizador as informações mais precisas com as quais você poderá determinar

o melhor plano de acesso. Se as estatísticas atuais não estiverem disponíveis, o otimizador poderá escolher um plano de acesso ineficiente baseado em estatísticas padrão imprecisas.

Certifique-se de utilizar o **runstats** após fazer suas atualizações de tabela; caso contrário, a tabela poderá aparentar estar vazia para o otimizador. Este problema é evidente se a cardinalidade na janela Detalhes do operador for igual a zero. Neste caso, conclua suas atualizações de tabela, execute o comando **runstats** e recrie os instantâneos para tabelas afetadas.

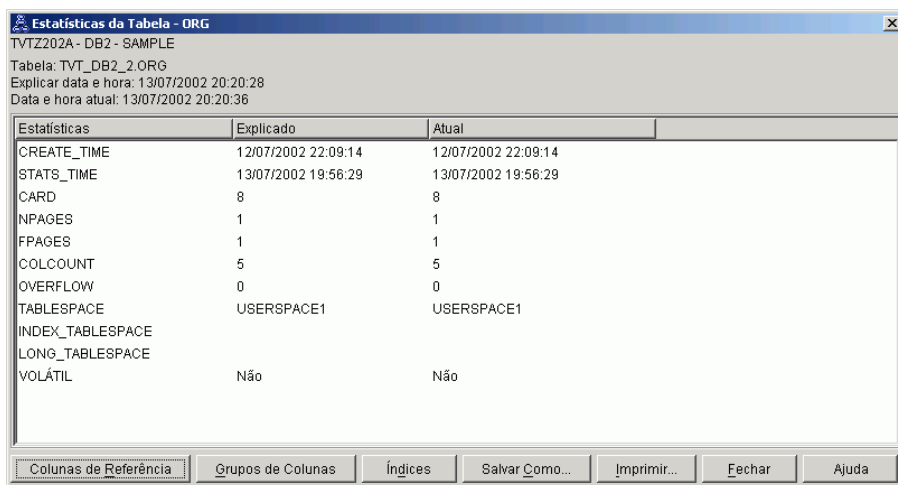
Para exibir o plano de acesso gráfico para esta consulta (Consulta 2): na janela Histórico de Instruções Explicadas, dê um clique duplo na entrada identificada como Consulta Número 2. A janela Gráfico do Plano de Acesso para esta execução da instrução será aberta.



Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

1. As estatísticas atuais existem para cada tabela na consulta?

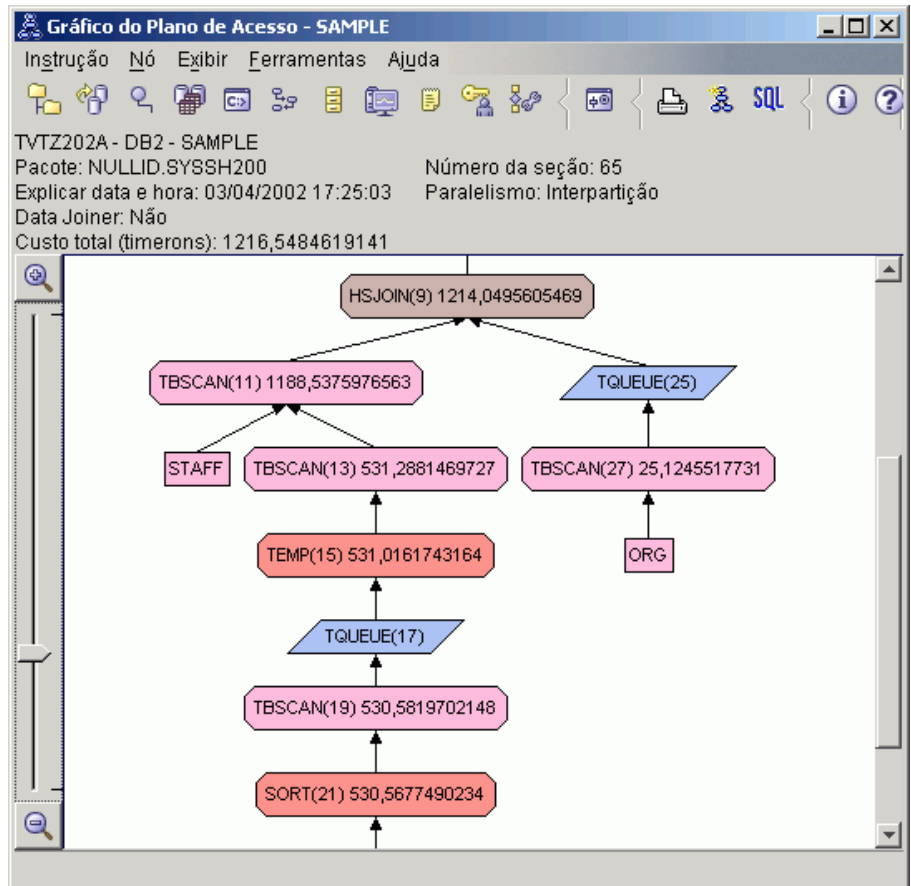
A janela Estatísticas da Tabela da tabela ORG mostra que o otimizador utilizou estatísticas atuais (o valor **STATS_TIME** é o tempo real em que as estatísticas foram coletadas). A precisão das estatísticas depende da existência de alterações significativas para o conteúdo das tabelas desde a execução do comando **runstats**.



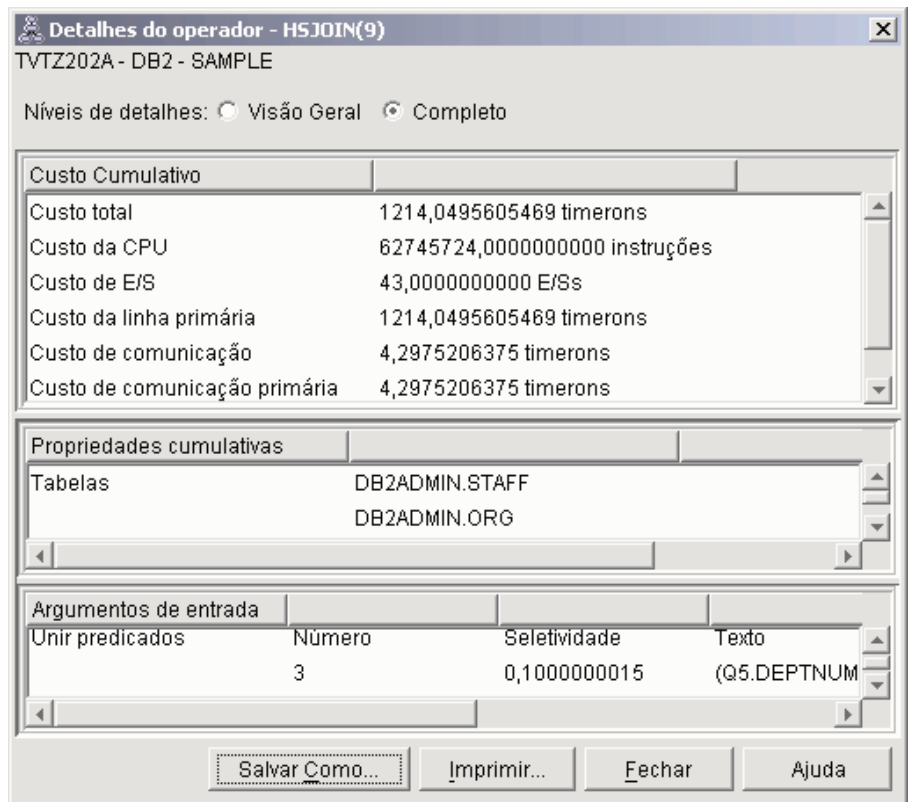
Estatísticas	Explicado	Atual
CREATE_TIME	12/07/2002 22:09:14	12/07/2002 22:09:14
STATS_TIME	13/07/2002 19:56:29	13/07/2002 19:56:29
CARD	8	8
NPAGES	1	1
FPAGES	1	1
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLÁTIL	Não	Não

2. Este plano de acesso utiliza os métodos mais eficientes de acesso de dados?

Como na Consulta 1, o plano de acesso na Consulta 2 utiliza varreduras de tabela (TBSCAN) e não varreduras de tabela (IXSCAN). Embora existam estatísticas atuais, uma varredura de índice não foi feita porque não houveram índices nas colunas utilizadas pela consulta. Uma maneira de melhorar a consulta seria fornecer ao otimizador os índices e colunas utilizados para unir tabelas (isto é, em colunas utilizadas para unir predicados). Neste exemplo, esta é a primeira união de varredura de combinação: HSJOIN (9).



Na janela Detalhes do operador do operador HSJOIN (9), consulte a seção **Unir Predicados em Argumentos de entrada**. As colunas utilizadas nessa operação de união são listadas sob a coluna **Texto**. Neste exemplo, essas colunas são DEPTNUMB e DEPT.



3. Quão efetivo é este plano de acesso?

Planos de acesso baseados em estatísticas atualizadas sempre produzem um custo estimado realista (medido em timerons). Como o custo estimado na Consulta 1 foi baseado em estatísticas padrão, não é possível comparar o custo de dois gráficos do plano de acesso para determinar qual é mais efetivo. Se o custo é maior ou menor não é relevante. Será necessário comparar o custo dos planos de acesso baseados em estatísticas atuais para obter uma medida válida de eficiência.

4. O que vem a seguir?

A Consulta 3 observa os efeitos da inclusão de índices nas colunas DEPTNUMB e DEPT. A inclusão de índice nas colunas utilizadas em Unir Predicados podem melhorar o desempenho.

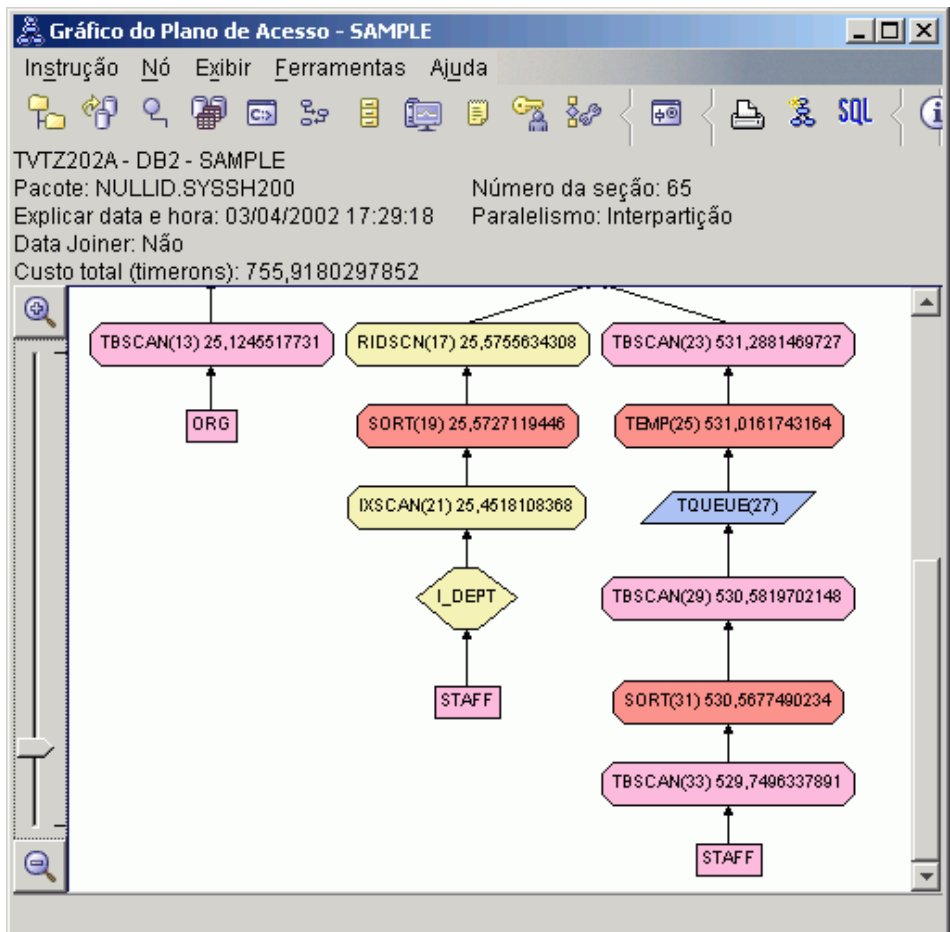
Criando Índices em Colunas Utilizadas para Unir Tabelas em uma Consulta

Este exemplo é baseado no plano de acesso descrito na Consulta 2 ao criar índices na coluna DEPT da tabela STAFF e na coluna DEPTNUMB da tabela ORG.

Nota: Na Versão 8, os índices recomendados podem ser criados utilizando o assistente Desempenho de Carga de Trabalho.

Para exibir o gráfico do plano de acesso para esta consulta (Consulta 3): na janela Histórico de Instruções Explicadas, dê um clique duplo na entrada identificada como Consulta Número 3. A janela Gráfico do Plano de Acesso para esta execução da instrução será aberta.

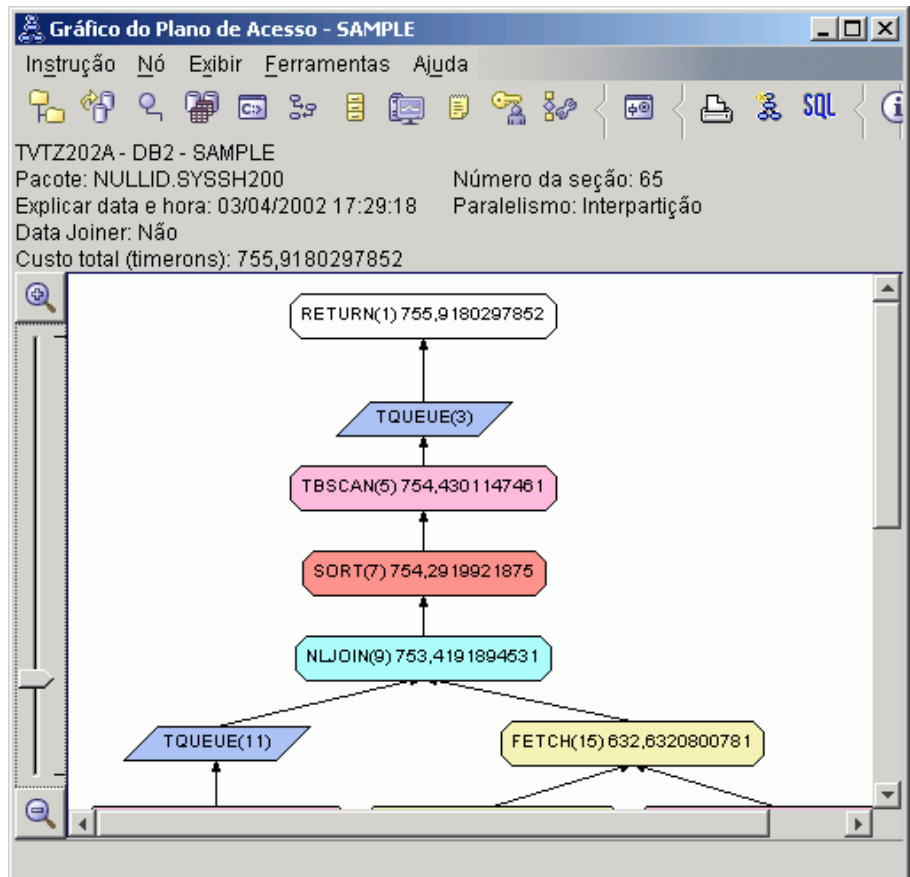
Nota: Embora um índice tenha sido criado para DEPTNUM, o otimizador não a utilizou.



Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

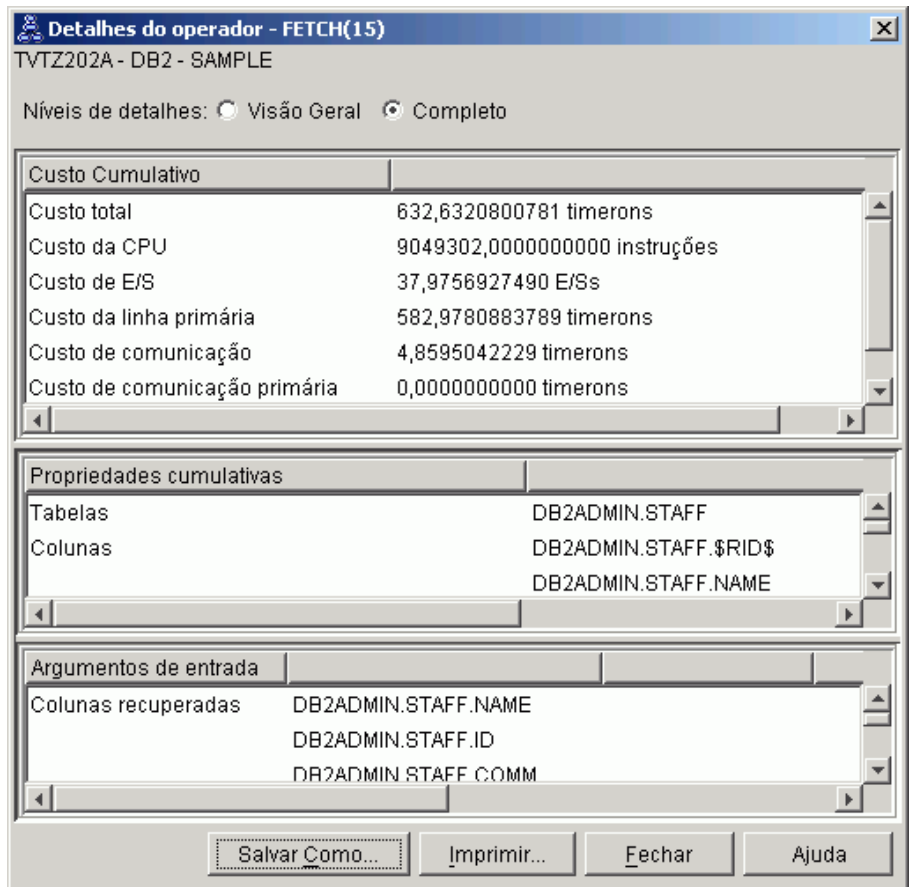
1. O que mudou no plano de acesso com índices?

Um nó com formato de losango, **I_DEPT**, foi incluído logo acima da tabela STAFF. Este nó representa o índice criado em DEPT, e mostra que o otimizador utilizou uma varredura de índice em vez de uma varredura de tabela para determinar quais linhas devem ser recuperadas.



2. Este plano de acesso utiliza os métodos mais eficientes de acesso de dados?

O plano de acesso para esta consulta mostra o efeito da criação de índices na coluna DEPTNUMB da tabela ORG, resultando em FETCH (15) e IXSCAN (21) na coluna DEPT da tabela STAFF. A Consulta 2 não possuía este índice; conseqüentemente, uma varredura de tabela foi utilizada nesse exemplo.



A combinação do índice e busca são calculados para ter um custo menor do que a varredura de tabela completa utilizada nos planos de acesso anteriores.

3. Quão efetivo é este plano de acesso?

Este plano de acesso tem um melhor custo do que o do exemplo anterior. O custo cumulativo foi reduzido de aproximadamente 1.214 timerons na Consulta 2 para aproximadamente 755 timerons na Consulta 3.

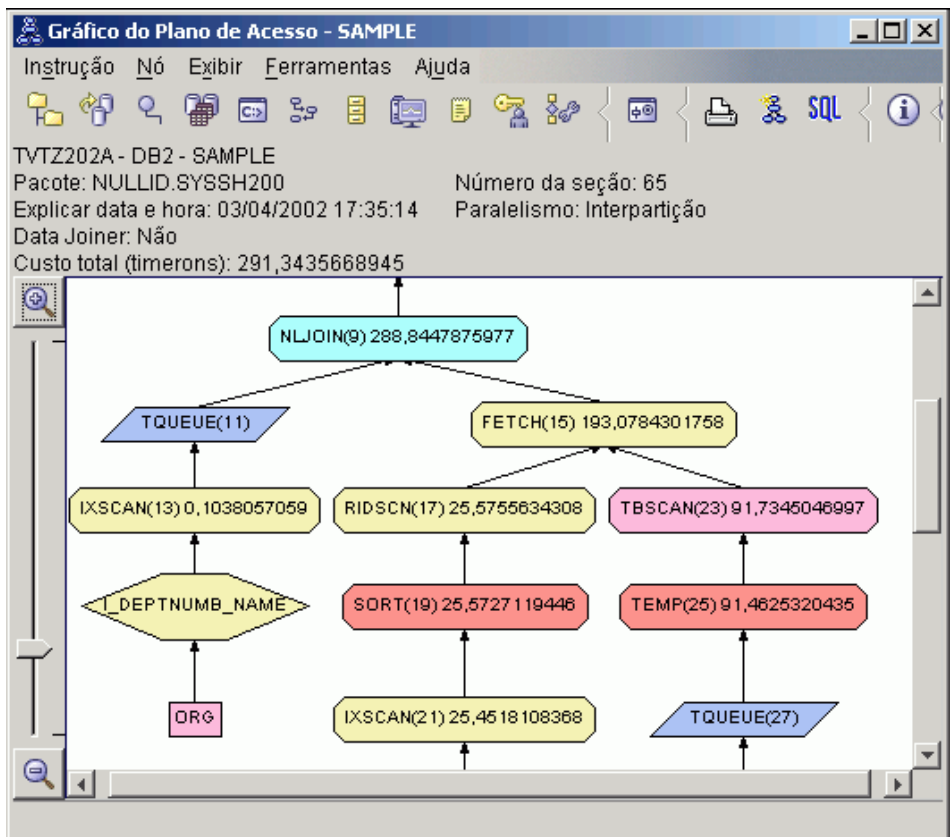
4. O que vem a seguir?

A Consulta 4 reduz a busca e varredura de índice para uma única varredura de índice sem uma busca. A criação de índices adicionais pode reduzir o custo estimado para o plano de acesso.

Criando Índices Adicionais em Colunas de Tabela

Este exemplo é baseado no plano de acesso descrito na Consulta 3, criando um índice na coluna JOB da tabela STAFF, e incluindo DEPTNAME no índice existente na tabela ORG. (Incluir um índice separado pode causar um acesso adicional).

Para exibir o plano de acesso para esta consulta (Consulta 4): na janela Histórico de Instruções Explicadas, dê um clique duplo na entrada identificada como Consulta Número 4. A janela Gráfico do Plano de Acesso para esta execução da instrução será aberta.

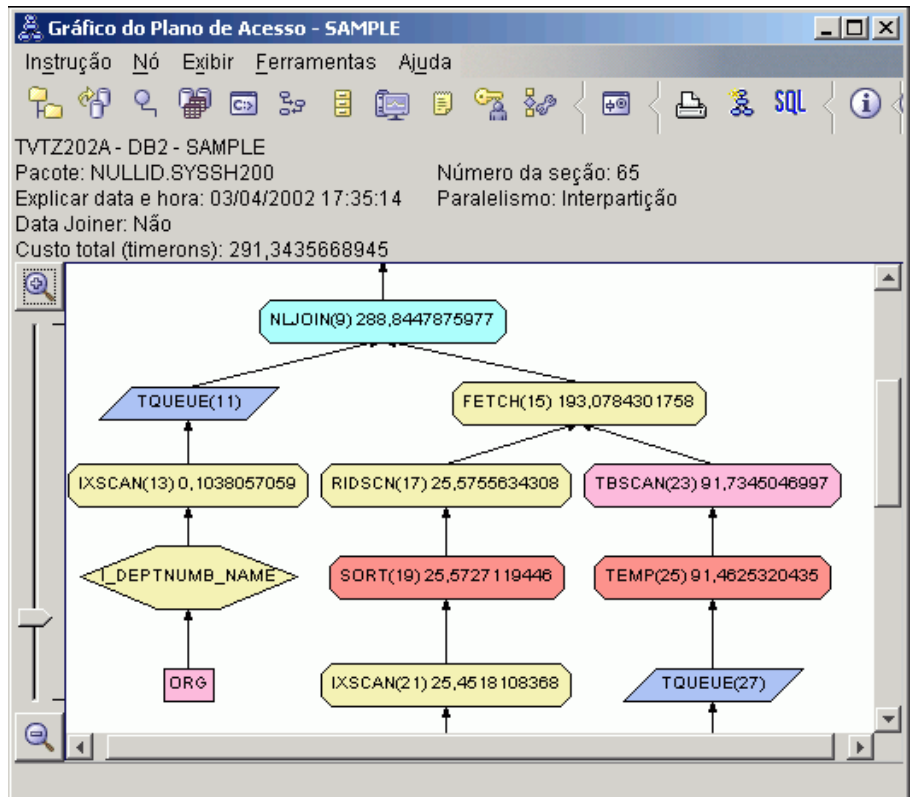


Responder as seguintes perguntas irá ajudá-lo a entender como melhorar a consulta.

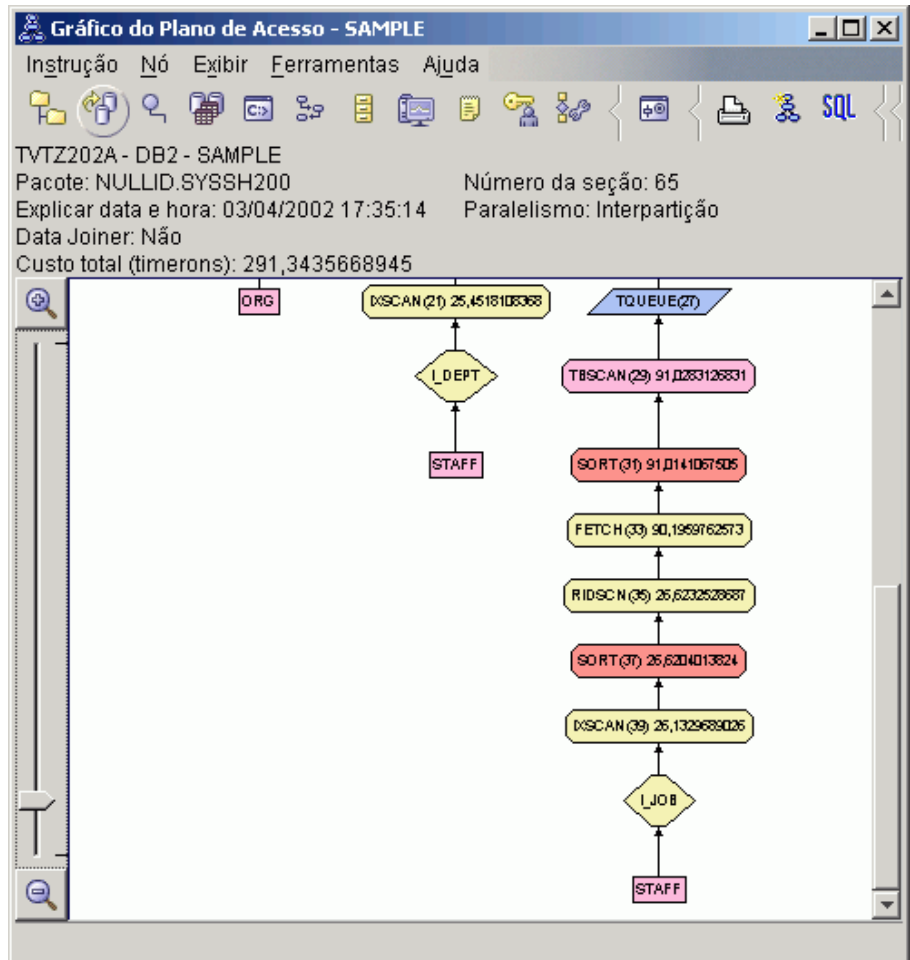
1. O que mudou neste plano de acesso como resultado da criação de índices adicionais?

Na parte intermediária do gráfico do plano de acesso, observe que para a tabela ORG, a varredura de tabela anterior foi alterada para uma

varredura de índice, IXSCAN (7). A inclusão da coluna DEPTNAME no índice na tabela ORG permitiu que o otimizador refine o acesso envolvendo a varredura de tabela.



Na parte inferior do gráfico do plano de acesso, observe que para a tabela STAFF a varredura e busca de índice anterior foram alteradas para apenas um IXSCAN (39) de varredura de índice. Criar o índice JOB na tabela STAFF permitiu ao otimizador eliminar o acesso adicional envolvendo a busca.



2. Quão efetivo é este plano de acesso?

Este plano de acesso tem um melhor custo do que o do exemplo anterior. O custo cumulativo foi reduzido de aproximadamente 753 timerons na Consulta 3 para aproximadamente 288 timerons na Consulta 4.

O que Vem a Seguir

Consulte o *Administration Guide* para encontrar informações detalhadas sobre procedimentos adicionais que você poderá realizar para melhorar o desempenho. Você então poderá retornar ao Visual Explain para avaliar o impacto de suas ações.

Apêndice A. Conceitos do Visual Explain

Plano de Acesso

Determinados dados são necessários para resolver uma instrução SQL explicável. Um *plano de acesso* especifica uma ordem de operações para acessar estes dados. Ele permite que você veja estatísticas de tabelas selecionadas, índices ou colunas, propriedades de operadores, informações globais como uma área de tabela e estatísticas de funções e parâmetros de configuração relevantes para otimização. Com o Visual Explain, você pode visualizar o plano de acesso para uma instrução SQL no formato de um gráfico.

O otimizador produz um plano de acesso sempre que uma instrução SQL explicável for compilada. Isto acontece no momento de prep/bind de instruções estáticas e no momento da execução de instrução dinâmicas.

É importante compreender que um plano de acesso é uma *estimativa* baseada nas informações que estão disponíveis. O otimizador baseia suas estimativas em informações como as seguintes:

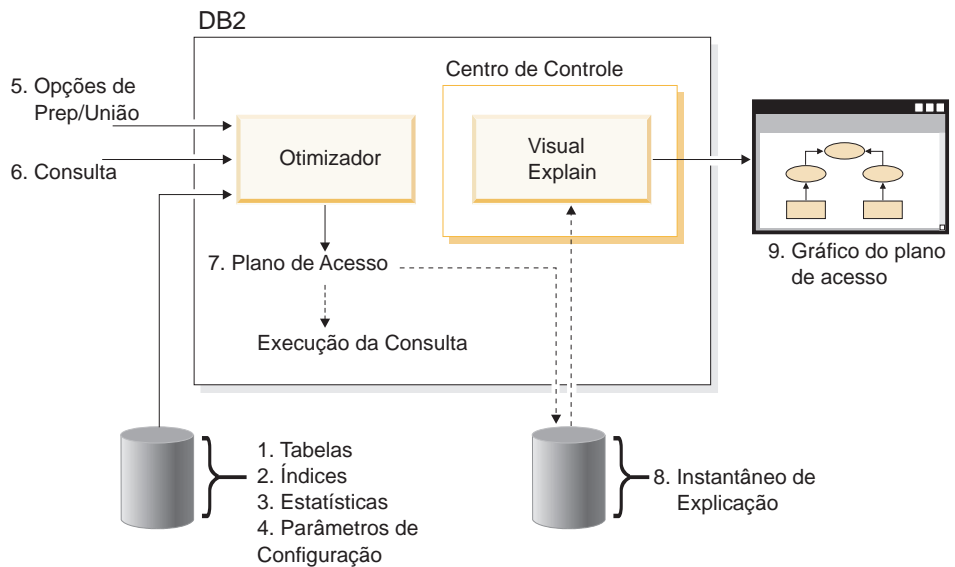
- Estatísticas em tabelas de catálogo do sistema (se as estatísticas não são atuais, atualize-as utilizando o comando runstats.)
- Parâmetros de configuração
- Opções de união
- A classe de otimização de consulta

Informações de custo associadas a um plano de acesso é a *melhor estimativa* do otimizador de uso do recurso para uma consulta. O tempo decorrido real para uma consulta pode variar dependendo de fatores externos ao escopo do DB2 (por exemplo, a quantidade de outros aplicativos sendo executados ao mesmo tempo). O tempo decorrido real pode ser medido enquanto a consulta é executada, utilizando monitoração de desempenho.

Gráfico do Plano de Acesso

O Visual Explain utiliza informações de várias origens para produzir um gráfico do plano de acesso, conforme mostrado na ilustração abaixo. Com base em várias entradas, o otimizador escolhe um plano de acesso e Visual Explain mostra-o em um *gráfico do plano de acesso*. Os nós no gráfico representam tabelas e índices e cada operação feita neles. As ligações entre os

nós representam o fluxo de dados.



A lista de tarefas a seguir corresponde às tarefas mostradas na ilustração acima. (Linhas pontilhadas indicam etapas que são necessárias para o Visual Explain).

1. Ajuste o design de sua tabela e reorganize os dados da tabela .
2. Crie índices apropriados.
3. Utilize o comando `runstats` para fornecer ao otimizador estatísticas atuais .
4. Escolha parâmetros de configuração apropriados.
5. Escolha opções de união apropriadas.
6. Crie consultas para recuperar apenas dados necessários .
7. Crie um plano de acesso.
8. Crie instantâneos de explicação.
9. Visualize e utilize um gráfico do plano de acesso.

Por exemplo, para utilizar o Visual Explain, primeiro atualize estatísticas atuais utilizando o comando **runstats** nas tabelas e índices utilizados pela instrução. Estas estatísticas, os parâmetros de configuração, opções de união e a própria consulta são utilizados para criar um plano de acesso e um instantâneo de explicação quando o pacote estiver unido. O Visual Explain utiliza o instantâneo de explicação resultante para exibir o gráfico do plano de acesso para a instrução.

Nó do Gráfico do Plano de Acesso

O gráfico do plano de acesso é constituído de uma árvore exibindo *nós*. Estes nós representam:

- Tabelas, mostradas como retângulos
- Índices, mostrados como losangos
- Operadores, mostrados como octógonos (8 lados). Operadores TQUEUE, mostrados como paralelogramos
- Funções de tabelas, mostradas como hexágonos (6 lados).

Armazenamento em Cluster

Com o tempo, as atualizações podem fazer com que linhas em páginas de dados tenham sua localização alterada, diminuindo o grau de *cluster* que existe entre um índice e as páginas de dados. Reorganizar uma tabela respeitando um índice selecionado, refaz o cluster dos dados. Um índice com cluster é mais útil para colunas que possuam predicados de intervalo, porque permite um melhor acesso seqüencial de dados na tabela base. O resultado é uma menor busca de páginas, já que valores semelhantes estão na mesma página de dados.

Em geral, apenas um dos índices em uma tabela pode ter um alto grau de cluster.

Para verificar o grau de cluster de um índice, dê um clique duplo em seu nó para exibir a janela Estatísticas de Índice. Os valores de proporção de cluster ou de fator de cluster são mostrados nesta janela. Se um valor estiver baixo, considere reorganizar os dados da tabela.

Para obter mais informações, consulte a seção sobre reorganização de dados de tabela no *Administration Guide*.

Contêiner

Um *contêiner* é uma localização de armazenamento físico dos dados. Ele está associado a uma área de tabela e pode ser um arquivo, um diretório ou um dispositivo.

Custo

Custo, no contexto do Visual Explain, é o uso total estimado do recurso necessário para executar o plano de acesso para uma instrução (ou os elementos de uma instrução). O custo origina-se de uma combinação de custo da CPU (em número de instruções) e E/S (em números de buscas e transferências de página).

A unidade do custo é o *timeron*. Um timeron não se equipara diretamente a nenhum tempo decorrido real, mas dá uma estimativa relativa aproximada dos recursos (custo) requeridos pelo gerenciador de banco de dados para executar dois planos para a mesma consulta.

O custo mostrado em cada nó de operador de um gráfico do plano de acesso é o custo cumulativo, do início da execução do plano de acesso até a (e incluindo) execução deste operador específico. Ele não reflete fatores como a carga de trabalho no sistema ou o custo de linhas de dados retornadas para o usuário.

Bloqueio de Cursor

Bloqueio do cursor é uma técnica que reduz o overhead fazendo com que o gerenciador de banco de dados recupere um *bloco* de linhas em uma única operação. Estas linhas são armazenadas em cache enquanto são processadas. A cache é alocada quando um aplicativo emite um pedido OPEN CURSOR e é desalocada quando o cursor é fechado. Quando todas as linhas tiverem sido processadas, outro bloco de linhas é recuperado.

Utilize a opção BLOCKING nos comando **PREP** ou **BIND** junto com os seguintes parâmetros para especificar o tipo de bloqueio de cursor:

UNAMBIG

Apenas cursores não ambíguos são bloqueados (o padrão).

ALL Cursores ambíguos e não ambíguos são bloqueados.

NO Cursores não são bloqueados.

Para obter mais informações consulte a seção sobre bloqueio de cursor no *Administration Guide*.

Área de Tabela DMS (database-managed space)

Há dois tipos de áreas de tabela que podem existir em um banco de dados: área gerenciada pelo banco de dados (DMS) e área gerenciada pelo sistema (SMS).

Áreas de tabela DMS são gerenciadas pelo gerenciador de banco de dados e são projetadas e ajustadas para atender aos seus requisitos.

A definição de área de tabela DMS inclui uma lista de arquivos (ou dispositivos) nos quais os dados do banco de dados são armazenados em seu formato de área de tabela DMS.

Você pode incluir arquivos (ou dispositivos) pré-alocados em uma área de tabela DMS existente para aumentar sua capacidade de armazenamento. O gerenciador de banco de dados regula automaticamente dados existentes em todos os contêineres que pertencem a esta área de tabela.

Áreas de tabela DMS e SMS podem coexistir no mesmo banco de dados.

SQL Dinâmica

Instruções SQL dinâmicas são instruções SQL que são preparadas e executadas dentro de um programa aplicativo enquanto o programa é executado. Em SQL dinâmica:

- Você emite a instrução SQL interativamente, utilizando CLI ou CLP
- A origem SQL é contida em variáveis de linguagem de host que são embutidas em um programa aplicativo.

Quando o DB2 executa uma instrução SQL dinâmica, ele cria um plano de acesso que é baseado em estatísticas e parâmetros de configuração de catálogos. Este plano de acesso pode mudar de uma execução do programa aplicativo de instruções para outra.

A alternativa para SQL dinâmica é SQL estática.

Instantâneo de Explicação

Com o Visual Explain você pode examinar o conteúdo de um instantâneo de explicação.

Um *instantâneo de explicação* é formado por informações compactadas que são coletadas quando uma instrução SQL é explicada. Ele é armazenado como um BLOB (binary large object) na tabela EXPLAIN_STATEMENT e contém as seguintes informações:

- A representação interna do plano de acesso, incluindo seus operadores e as tabelas e índices acessados.
- Os critérios de decisão utilizados pelo otimizador, incluindo estatísticas para objetos de bancos de dados e o custo cumulativo para cada operação.

Um instantâneo de explicação é exigido se você quiser exibir a representação gráfica do plano de acesso de uma instrução SQL. Para assegurar que um instantâneo de explicação seja criado:

1. Tabelas de explicação devem existir no gerenciador de banco de dados para armazenar os instantâneos de explicação. Para obter informações sobre como criar estas tabelas, consulte Criando tabelas de explicação na ajuda online.

2. Para obter uma pacote contendo instruções SQL estáticas, defina a opção EXPLSNAP para ALL ou YES quando executar bind ou prep no pacote. Você receberá um instantâneo de explicação para cada instrução SQL explicável no pacote. Para obter mais informações sobre os comandos **BIND** e **PREP**, consulte a *Command Reference*.
3. Para instruções SQL dinâmicas, defina a opção EXPLSNAP para ALL quando efetuar a união do aplicativo que as emite, ou defina o registro especial CURRENT EXPLAIN SNAPSHOT para YES ou EXPLAIN antes de emití-las interativamente. Para obter mais informações, consulte a seção sobre instantâneos de explicação atuais na *SQL Reference*.

Instrução Explicável

Uma *instrução explicável* é uma instrução SQL para a qual uma operação de explicação pode ser realizada.

Instruções SQL explicáveis são:

- SELECT
- INSERT
- UPDATE
- DELETE
- VALUES

Instrução Explicada

Uma *instrução explicada* é uma instrução SQL para a qual uma operação de explicação foi realizada. Instruções explicadas são mostradas na janela Histórico de Instruções Explicadas.

Operando

Um operando é uma entidade na qual uma operação é realizada. Por exemplo, uma tabela ou um índice é um operando de vários operados como TBSCAN e IXSCAN.

Operador

Um *operador* é uma ação que deve ser realizada em dados ou a saída de uma tabela ou de um índice, quando o plano de acesso para uma instrução SQL é executado.

Os seguintes operadores podem ser exibidos no gráfico do plano de acesso:

DELETE

Exclui linhas de uma tabela.

EISCAN

Varre um índice definido pelo usuário para produzir um fluxo reduzido de linhas.

FETCH

Busca colunas de uma tabela utilizando um identificador de registro específico.

FILTER

Filtra dados aplicando um ou mais predicados neles.

GRPBY

Agrupar linhas por valores de coluna ou funções designadas em comum e avalia funções definidas.

HSJOIN

Representa uma união de hash, onde duas ou mais tabelas tem hash executado nas colunas de união.

INSERT

Insere linhas em uma tabela.

IXAND

Executa ANDs nos identificadores de linha (RIDs) de duas ou mais varreduras de índice.

IXSCAN

Varre um índice de uma tabela com condições iniciar/parar opcionais, produzindo um fluxo ordenado de linhas.

MSJOIN

Representa uma união de combinação, onde tanto a tabela externa quanto a interna devem estar na ordem união-predicado.

NLJOIN

Representa uma união de loop aninhado que acessa uma tabela interna uma vez para cada linha da tabela externa.

RETURN

Representa o retorno dos dados da consulta para o usuário.

RIDSCN

Varre uma lista de identificadores de linha (RIDs) obtidos de um ou mais índices.

SHIP

Recupera dados de uma origem de dados remota. Utilizado no sistema federado.

SORT

Ordena linhas na ordem das colunas especificadas e opcionalmente elimina entradas duplicadas.

TBSCAN

Recupera linhas lendo todos os dados requeridos diretamente das páginas de dados.

TEMP Armazena dados em uma tabela temporária para serem lidos posteriormente (possivelmente várias vezes).

TQUEUE

Transfere dados de tabela entre agentes de banco de dados.

UNION

Concatena fluxos de linhas de várias tabelas.

UNIQUE

Elimina linhas com valores duplicados, para colunas especificadas.

UPDATE

Atualiza linhas em uma tabela.

CMPEXP

Nome do operador: CMPEXP

Representa: A computação de expressões exigidas para resultados intermediários ou finais.

(Este operador é apenas para o modo de depuração).

DELETE

Nome do operador: DELETE

Representa: A exclusão de linhas de uma tabela.

Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre-se em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas a serem excluídas.

Sugestão de Desempenho:

- Se você estiver excluindo todas as linhas de uma tabela, considere a instrução **DROP TABLE** ou o comando **LOAD REPLACE**.

EISCAN

Nome do operador: EISCAN

Representa: Este operador varre um índice definido pelo usuário para produzir um fluxo reduzido de linhas. A varredura utiliza as várias condições iniciar/parar da função criadora de intervalo fornecida pelo usuário.

Esta operação é realizada para diminuir o conjunto de linhas qualificadas antes de acessar a tabela base (com base em predicados).

Sugestão de Desempenho:

- Com o tempo, atualizações de banco de dados podem fazer com que um índice torne-se fragmentado, resultando em mais páginas de índice que o necessário. Isto pode ser corrigido eliminando e recriando o índice, ou reorganizando-o.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando `runstats`

FETCH

Nome do operador: FETCH

Representa: A busca colunas de uma tabela utilizando um identificador de linha (RID) específico.

Sugestões de desempenho:

- Expanda as chaves de índice para incluir as colunas buscadas para que as páginas de dados não tenham que ser acessadas.
- Encontre um índice relacionado à busca e dê um clique duplo em seu nó para exibir sua janela de estatísticas. Assegure que o grau de cluster seja alto para o índice.
- Aumente o tamanho do buffer se o valor de entrada/saída (E/S) criado pela busca for maior que o número de páginas na tabela.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando `runstats`.

O quantil e as estatísticas de valor freqüentes oferecem informações sobre a seletividade de predicados, o que determina quando varreduras de índice são escolhida em varreduras de tabela. Para atualizar estas estatísticas, utilize o comando **runstats** em uma tabela com a cláusula `WITH DISTRIBUTION`.

FILTER

Nome do operador: FILTER

Representa: A aplicação de predicados residuais para que dados sejam filtrados com base nos critérios fornecidos pelos predicados.

Sugestões de desempenho:

- Assegure que você tenha utilizado predicados que recuperam apenas os dados necessários. Por exemplo, assegure que o valor de seletividade dos predicados represente a parte da tabela que você quer que seja retornada.
- Assegure que a classe de otimização seja pelo menos 3 para que o otimizador utilize uma união ao invés de uma subconsulta. Se isto não for possível, tente escrever novamente a consulta SQL à mão para eliminar a subconsulta. Para obter um exemplo, consulte a seção sobre regravações de consultas pelo compilador SQL no *Administration Guide*.

GENROW

Nome do operador: GENROW

Representa: Uma função interna que gera uma tabela de linhas, não utilizando entrada de tabelas, índices ou operadores.

GENROW pode ser utilizado pelo otimizador para gerar linhas de dados (por exemplo, para uma instrução INSERT ou para algumas listas IN que são transformadas em uniões).

Para exibir as estatísticas estimadas para as tabelas geradas pela função GENROW, dê um clique duplo em seu nó.

GRPBY

Nome do operador: GRPBY

Representa: O agrupamento de linhas de acordo com valores comuns de colunas ou funções designadas. Esta operação é necessária para produzir um grupo de valores ou para avaliar funções definidas.

Se nenhuma coluna GROUP BY for especificada, o operador GRPBY ainda pode ser utilizado se houver funções de agregação na lista SELECT, indicando que a tabela inteira foi tratada como um único grupo quando esta agregação foi feita.

Sugestões de desempenho:

- Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas para serem agrupadas.

- Para melhorar o desempenho de uma instrução SELECT que contenha uma única função de agregação mas nenhuma cláusula GROUP BY, tente o seguinte:
 - Para uma função de agregação MIN(C), crie um índice ascendente em C.
 - Para uma função de agregação MAX(C), crie um índice descendente em C.

HSJOIN

Nome do operador: HSJOIN

Representa: Uma união hash para a qual as linhas qualificadas de tabelas tem hash produzido para permitir união direta, sem pré-ordenação do conteúdo das tabelas.

Uma união é necessária sempre que houver mais que uma tabela com referência em uma cláusula FROM. Uma união de hash é possível sempre que houver um predicado de união que iguale colunas de duas tabelas diferentes. Os predicados de união precisam ter exatamente o mesmo tipo de dados. Uniões de hash também surgem de uma subconsulta escrita novamente, como é o caso de NLJOIN.

Uma união de hash não requer que as tabelas de entrada sejam ordenadas. A união é realizada varrendo-se a tabela interna da união de hash e gerando-se uma tabela de consulta produzindo hash dos valores de coluna de união. Ele lê a tabela externa, produzindo hash dos valores de coluna de união e verificando na tabela de consulta gerada para a tabela interna.

Para obter mais informações, consulte a seção sobre conceitos de união no *Administration Guide*.

Sugestões de desempenho:

- Utilize predicados locais (ou seja, predicados que façam referência a uma tabela) para reduzir o número de linhas a serem unidas.
- Aumente o tamanho do heap de ordenação para torná-lo grande o suficiente para conter a tabela de consulta de hash na memória.
- Se as estatísticas não estiverem atualizadas, atualize-as, utilizando o comando runstats.

INSERT

Nome do operador: INSERT

Representa: A inserção de linhas em uma tabela.

Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas para serem inseridas.

IXAND

Nome do operador: IXAND

Representa: A execução de AND dos resultados de várias varreduras de índice utilizando técnicas de Bitmap Dinâmico. O operador permite que predicados com AND sejam aplicados a vários índices, para reduzir acessos de tabela básicos ao mínimo.

Este operador é executado para:

- Reduzir o conjunto de linhas antes de acessar a tabela base.
- Executar AND de predicados aplicados a vários índices
- Executar AND nos resultados de semi-uniões, utilizados em uma união estrela.

Sugestões de desempenho:

- Com o tempo, atualizações de banco de dados podem fazer com que um índice torne-se fragmentado, resultando em mais página de índice que o necessário. Isto pode ser corrigido eliminando e recriando o índice, ou reorganizando-o.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando `runstats`.
- Em geral, varreduras de índices são mais eficientes quando apenas algumas linhas são qualificadas. Para estimar o número de linhas qualificadas, o otimizador utiliza as estatísticas que estão disponíveis para as colunas com referência em predicados. Se alguns valores ocorrerem mais freqüentemente que outros, é importante solicitar estatísticas de distribuição utilizando a cláusula `WITH DISTRIBUTION` para o comando `runstats`. Utilizando as estatísticas de distribuição não uniformes, o otimizador pode distinguir entre valores que ocorrem freqüentemente e não freqüentemente.
- IXAND pode explorar melhor índices de coluna únicos, já que chaves de início e parada são cruciais no uso de IXAND.
- Para uniões estrela, crie índices de coluna única para cada uma das colunas mais seletivas na tabela de fato e nas tabelas de dimensões relacionadas.

IXSCAN

Nome do operador: IXSCAN

Representa: A varredura de um índice definido pelo usuário para produzir um fluxo reduzido de linhas. A varredura pode utilizar condições de início/parada otimizados ou pode aplicar-se a predicados indexáveis que fazem referência a colunas do índice.

Esta operação é realizada para diminuir o conjunto de linhas qualificadas antes de acessar a tabela base (com base em predicados).

Para obter mais informações, consulte a seção sobre varreduras de índice no *Administration Guide*.

Sugestões de desempenho:

- Com o tempo, atualizações de banco de dados podem fazer com que um índice torne-se fragmentado, resultando em mais páginas de índice que o necessário. Isto pode ser corrigido eliminando e recriando o índice, ou reorganizando-o.
- Quando duas ou mais tabelas estão sendo acessadas, o acesso à tabela interna através de um índice pode ser mais eficiente fornecendo um índice na coluna de união da tabela externa.

Para obter mais instruções sobre índices, consulte a ajuda online do Visual Explain.

- Se as estatísticas não estiverem atualizadas, atualize-as utilizando o comando `runstats`.
- Em geral, varreduras de índices são mais eficientes quando apenas algumas linhas são qualificadas. Para estimar o número de linhas qualificadas, o otimizador utiliza as estatísticas que estão disponíveis para as colunas com referência em predicados. Se alguns valores ocorrerem mais frequentemente que outros, é importante solicitar estatísticas de distribuição utilizando a cláusula `WITH DISTRIBUTION` para o comando **runstats**. Utilizando as estatísticas de distribuição não uniformes, o otimizador pode distinguir entre valores que ocorrem frequentemente e não frequentemente.

MSJOIN

Nome do operador: MSJOIN

Representa: Uma união de combinação para a qual as linhas qualificadas das tabelas externa e interna devem estar na ordem união-predicado. Uma combinação também é denominada uma *união de varredura de combinação* ou *união de combinação ordenada*.

Uma união é necessária sempre que houver mais que uma tabela com referência em uma cláusula `FROM`. Uma união de combinação é possível

sempre que houver um predicado de união que iguale colunas de duas tabelas diferentes. Também pode ser criada a partir de uma subconsulta escrita novamente.

Uma união de combinação requer colunas de entrada e saída ordenadas, já que as tabelas são geralmente varridas apenas uma vez. Esta entrada ordenada é obtida acessando-se um índice ou uma tabela ordenada.

Para obter mais informações, consulte a seção sobre conceitos de união no *Administration Guide*.

Sugestões de desempenho:

- Utilize predicados locais (ou seja, predicados que façam referência a uma tabela) para reduzir o número de linhas a serem unidas.

Para obter instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

NLJOIN

Nome do operador: NLJOIN

Representa: Uma união de loop aninhado que varre (geralmente com uma varredura de índice) a tabela interna uma vez para cada linha da tabela externa.

Uma união é necessária sempre que houver mais que uma tabela com referência em uma cláusula FROM. Uma união de loop aninhado não requer um predicado de união, mas geralmente tem melhor desempenho com um.

Uma união de loop aninhado é realizada:

- Varrendo-se a tabela interna para cada linha acessada da tabela externa.
- Realizando-se uma consulta de índice na tabela interna para cada linha acessada da tabela externa.

Para obter mais informações, consulte a seção sobre conceitos de união no *Administration Guide*.

Sugestões de desempenho:

- Uma união de loop aninhado provavelmente será mais eficiente se houver um índice nas colunas união-predicado da tabela interna (a tabela exibida à direita do operador NLJOIN). Verifique se a tabela interna é uma TBSCAN ao invés de IXSCAN. Se for, considere incluir um índice em suas colunas de união.

Outra maneira (menos importante) de tornar a união mais eficiente é criar um índice nas colunas de união da tabela externa para que ela seja ordenada.

Para obter mais instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

Informações relacionadas:

- Uniões Estrela.

PIPE

Nome do operador: PIPE

Representa: A transferência de linhas para outros operadores sem alterações às linhas.

(Este operador é apenas para o modo de depuração).

RETURN

Nome do operador: RETURN

Representa: O retorno de dados de uma consulta para o usuário. Este é o operador final no gráfico do plano de acesso e mostra os valores e custos totais acumulados do plano de acesso.

Este operador representa uma operação necessária.

Sugestão de Desempenho:

- Assegure que você tenha utilizado predicados que recuperem apenas os dados necessários. Por exemplo, assegure que o valor de seletividade dos predicados represente a parte da tabela que você quer que seja retornada.

RIDSCN

Nome do operador: RIDSCN

Representa: A varredura de uma lista de identificadores de linha (RIDs) obtidos de um ou mais índices.

Este operador é considerado pelo otimizador quando:

- Predicados são conectados por palavras-chave OR ou existe um predicado IN. Pode-se utilizar uma técnica denominada ORing de índice, que combina resultados de vários acessos de índices na mesma tabela.
- É proveitoso utilizar pré-busca de lista para um único acesso de índice, já que ordenar os identificadores de linha antes de acessar as linhas base tornam a E/S mais eficiente.

RQUERY

Nome do operador: SHIP

Representa: Um operador utilizado no sistema federado para recuperar dados de uma origem de dados remota. Este operador é considerado pelo otimizador quando: Um operador SHIP envia uma instrução SQL SELECT para uma origem de dados remota para recuperar o resultado da consulta. A instrução SELECT é gerada utilizando o dialeto SQL suportado pela origem de dados e pode conter qualquer consulta válida, conforme permitido pela origem de dados.

Sugestão de Desempenho: Consulte o Capítulo 4 da publicação *Administration Guide Vol 2, Federated Database Query and Network Tuning Information*.

SORT

Nome do operador: SORT

Representa: A ordenação das linhas em uma tabela na ordem de uma ou mais de suas colunas, eliminando opcionalmente entradas duplicadas.

A ordenação é necessária quando não existem índices que satisfaçam os pedidos solicitados ou quando a ordenação pode ser menos cara que uma varredura de índice. A ordenação geralmente é realizada como uma operação final quando as linhas requeridas são buscadas, ou para ordenar dados anteriores para uma união ou para um agrupamento.

Se o número de linhas for alto ou se os dados ordenados não puderem ter canal processado, a operação requer a geração de tabelas temporárias de alto custo.

Para obter mais informações sobre ordenações, consulte o *Administration Guide*.

Sugestões de desempenho:

- Considere incluir um índice nas colunas de ordenação.

Para obter instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Assegure que você tenha utilizado predicados que recuperem apenas os dados necessários. Por exemplo, assegure que o valor de seletividade dos predicados represente a parte da tabela que você quer que seja retornada.
- Verifique se o tamanho da pré-busca da área de tabela temporária do sistema é adequado, ou seja, não esteja ligado a E/S. (Para verificar isto, selecione **Instrução->Exibir estatísticas->Áreas de Tabelas**).
- Se ordenações grandes forem freqüentemente necessárias, considere aumentar os valores dos seguintes parâmetros de configuração:
 - Tamanho do heap de ordenação (sortheap). Para alterar este parâmetro, clique com o botão direito no banco de dados no Centro de Controle e selecione *Configurar* neste menu popup. Selecione a guia **Desempenho** no bloco de notas que é exibido.
 - Limite de heap de ordenação (sheapthres). Para alterar este parâmetro, clique com o botão direito na instância de banco de dados no Centro de Controle e selecione *Configurar* neste menu popup. Selecione a guia **Desempenho** no bloco de notas que é exibido.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

TBSCAN

Nome do operador: TBSCAN

Representa: Uma varredura de tabela (varredura relacional) que recupera linhas lendo todos os dados necessários diretamente das páginas de dados.

Este tipo de varredura é escolhida pelo otimizador em uma varredura de índice quando:

- O intervalo de valores varridos ocorre com freqüência (ou seja, a maioria das tabelas devem ser acessada)
- A tabela for pequena
- O armazenamento em cluster for baixo
- Um índice não existir

Para obter mais informações sobre varreduras de tabelas e de índices, consulte o *Administration Guide*.

Sugestões de desempenho:

- Uma varredura de índice pode ser mais eficiente que uma varredura de tabela se a tabela for grande e a maioria de suas linhas não forem acessadas. Para aumentar a possibilidade da varredura de índice ser

utilizada pelo otimizador para esta situação, considere incluir índices em colunas para as quais há predicados seletivos.

Para obter mais instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Se um índice já existir mas não tiver sido utilizado, verifique se existem predicados seletivos em cada uma de suas colunas iniciais. Se estes predicados existirem, verifique se o grau de armazenamento em cluster é alto para o índice. (Para ver estas estatísticas, abra a janela Estatísticas de Tabela da tabela abaixo da ordenação e selecione o botão *Índices* para exibir a janela Estatísticas de Índice).
- Verifique se o tamanho da pré-busca da área de tabela é adequado, ou seja, não esteja ligado a E/S. (Para verificar isto, selecione **Instrução->Exibir estatísticas->Áreas de Tabelas**).

Para obter mais informações, consulte a seção sobre pré-busca de dados no conjunto de buffers no *Administration Guide*.

- Se as estatísticas não estiverem atualizadas, atualize-as utilizando o comando `runstats`.

O quantil e as estatísticas de valor freqüentes oferecem informações sobre a seletividade de predicados. Por exemplo, estas estatísticas podem ser utilizadas para determinar quando varreduras de índice são escolhidas em varreduras de tabelas. Para atualizar estes valores, utilize o comando `runstats` em uma tabela com a cláusula `WITH DISTRIBUTION`.

TEMP

Nome do operador: TEMP

Representa: A ação de ordenar dados em uma tabela temporária, para serem lidos posteriormente por outro operador (possivelmente várias vezes). A tabela é removida depois que a instrução SQL é processada, se não tiver sido removida antes.

Este operador é necessário para avaliar subconsultas ou para armazenar resultados intermediários. Em algumas situações (como quando a instrução pode ser atualizada), ela pode ser obrigatória.

TQUEUE

Nome do operador: TQUEUE

Representa: Uma fila de tabela que é utilizada para passar dados de tabela de um agente de banco de dados para outro quando há vários agentes de banco de dados processando uma consulta. Vários agentes de banco de dados são utilizados para processar uma consulta quando paralelismo está envolvido.

Os tipos de filas de tabela são:

- **Local:** A fila de tabela é utilizada para passar dados entre agentes de banco de dados dentro de um único nó. Uma fila de tabela local é utilizada para paralelismo intrapartição.
- **Não Local:** A fila de tabela é utilizada para passar dados entre agentes de banco de dados em nós diferentes.

UNION

Nome do operador: UNION

Representa: A concatenação de fluxos de linhas de várias tabelas.

Este operador representa uma operação necessária. Para melhorar os custos de planos de acesso, concentre-se em outros operadores (como varreduras e uniões) que definam o conjunto de linhas a serem concatenadas.

UNIQUE

Nome do operador: UNIQUE

Representa: A eliminação de linhas tendo valores duplicados para colunas especificadas.

Sugestão de Desempenho:

- Este operador não é necessário apenas se um índice exclusivo existir em colunas apropriadas.

Para obter instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

UPDATE

Nome do operador: UPDATE

Representa: A atualização de dados nas linhas de uma tabela.

Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre-se em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas para serem atualizadas.

Otimizador

O *otimizador* é o componente do compilador SQL que escolhe um plano de acesso para uma instrução SQL DML (data manipulation language). Ele faz isto modelando o custo de execução de vários planos de acesso alternativos e selecionando aquele com o menor custo estimado.

Pacote

Um *pacote* é um objeto armazenado no banco de dados que inclui as informações necessárias para processar as instruções SQL associadas a um arquivo de origem de um programa aplicativo. Ele é gerado através de:

- Pré-compilação de um arquivo de origem com o comando **PREP**.
- Ligação de um arquivo de união que foi gerado pelo pré-compilador com o comando **BIND**.

Predicado

Um *predicado* é um elemento de uma condição de pesquisa que expressa ou implica uma operação de comparação. Predicados são incluídos em cláusulas começando com WHERE ou HAVING.

Por exemplo, na instrução SQL a seguir:

```
SELECT * FROM SAMPLE
  WHERE NAME = 'SMITH' AND
  DEPT = 895 AND YEARS > 5
```

Os seguintes itens são predicados: NAME = 'SMITH', DEPT = 895; e YEARS > 5.

Predicados pertencem a uma das seguintes categorias, ordenadas da mais eficiente à menos eficiente:

1. Parêntese de condições de início e de parada (reduzir) uma varredura de índice. (Estas condições também são denominadas predicados de delimitação de intervalo).
2. Predicados de página de índice (também denominados sargable de índice) podem ser avaliados a partir de um índice porque as colunas envolvidas no predicado são parte da chave de índice.
3. Predicados de página de dados (também denominados sargable de dados) não podem ser avaliados a partir de um índice, mas podem ser avaliados enquanto linhas permanecem no buffer.
4. Predicados residuais geralmente requerem E/S além do simples acesso de uma tabela base e devem ser aplicados depois dos dados serem copiados

da página de buffer. Ele incluem predicados que contêm subconsultas, ou aqueles que lêem dados LONG VARCHAR ou LOB armazenados em arquivos separados da tabela.

Ao projetar predicados, você deve ter como objetivo a maior seletividade possível para que o mínimo de linhas seja retornado.

Os seguintes tipos de predicados são os mais eficientes e utilizados mais freqüentemente:

- Um *predicado de união de igualdade simples* é exigido para uma união de combinação. Ele tem o formato `tabela1.coluna = tabela2.coluna` e permite que colunas em duas tabelas diferentes sejam igualadas para que as tabelas sejam unidas.
- Um *predicado local* é aplicado apenas a uma tabela.

Para obter mais informações, consulte as seções sobre conceitos e otimização de acesso de dados no *Administration Guide*.

Classe de Otimização de Consulta

Uma *classe de otimização de consulta* é um conjunto de regras de regravação de consulta e técnicas de otimização para compilação de consultas.

As classes de otimização de consulta principais são:

- 1 Otimização restrita. Útil quando recursos de memória e processamento estão gravemente restringidos. Aproximadamente equivalente à otimização oferecida pela Versão 1.
- 2 Otimização leve. Especifica um nível para otimização superior ao da Versão 1, mas com custo de otimização significativamente menor que dos níveis 3 e superiores, especialmente para consultas muito complexas.
- 3 Otimização moderada. Chega o mais próximo de corresponder às características da otimização de consulta do DB2 para MVS/ESA.
- 5 Otimização normal. Recomendada para um ambiente misto utilizando transações simples e consultas complexas.
- 7 Otimização normal. Igual à otimização de consulta 5, exceto por não reduzir a quantidade de otimização de consulta para consultas SQL dinâmicas complexas.

Outras classes de otimização de consulta, para serem utilizadas apenas sob circunstâncias especiais, são:

- 0 Otimização mínima. Utilize apenas quando for necessária otimização mínima ou nula (ou seja, para consultas muito simples em tabelas bem indexadas).
- 9 Otimização máxima. Utiliza recursos de memória e de processamento substanciais. Utilize apenas se a classe 5 for insuficiente (ou seja, para consultas muito complexas e de longo prazo que não têm bom desempenho na classe 5).

No geral, utilize uma classe de otimização mais alta para consultas estáticas e para consultas que você prevê que levarão bastante tempo para serem executadas e uma classe de otimização mais baixa para consultas simples que sejam emitidas dinamicamente ou que sejam executadas apenas algumas vezes.

Para definir a otimização de consulta para instruções de SQL dinâmica, digite o seguinte comando no Processador de Linha de Comandos:

```
SET CURRENT QUERY OPTIMIZATION = n;
```

em que 'n' é a classe de otimização de consulta desejada.

Para definir a otimização de consulta para instruções SQL estáticas, utilize a opção **QUERYOPT** nos comandos **BIND** ou **PREP**.

Para obter mais informações, consulte a seção sobre ajuste da classe de otimização no *Administration Guide*.

Seletividade de Predicados

Seletividade refere-se à probabilidade de qualquer linha satisfazer um predicado (ou seja, ser verdadeira).

Por exemplo, uma seletividade de 0,01 (1%) para um predicado sendo operado em uma tabela com 1.000.000 linhas significa que este predicado retorna uma estimativa de 10.000 linhas (1% de 1.000.000) e descarta uma estimativa de 990.000 linhas.

Um predicado altamente seletivo (um com uma seletividade de 0,10 ou mais) é proveitoso. Estes predicados retornam menos linhas para serem trabalhadas por operadores futuros, exigindo assim menos CPU e E/S para satisfazer a consulta.

Exemplo

Suponha que você tenha uma tabela de 1.000.000 linhas e que a consulta original contenha uma cláusula 'ORDER BY' exigindo uma etapa de

ordenação adicional. Com um predicado que tenha seletividade menor que 0,01, a ordenação teria que ser feita em uma estimativa de 10.000 linhas. Entretanto, com um predicado menos seletivo de 0,50, a ordenação teria que ser feita em uma estimativa de 500.000 linhas, exigindo mais tempo de CPU e E/S.

Uniões Estrela

Um conjunto de uniões são considerados uma união estrela quando uma tabela de fato (tabela central maior) é unida a duas ou mais tabelas de dimensão (tabelas menores contendo descrições dos valores de coluna na tabela de fato).

Uma união Estrela é formada por 3 partes principais:

- Semi-uniões
- Index ANDing dos resultados das Semi-uniões
- Conclusão das semi-uniões.

Ela aparece como duas ou mais uniões alimentando um operador IXAND.

Uma Semi-união é uma forma especial de união na qual o resultado da união é apenas o RID (Identificador de Linha) da tabela interna, ao invés da união das colunas da tabela interna e externa.

Uniões estrela utilizam Semi-uniões para fornecer Identificadores de Linha para um operador Index ANDing. O operador Index ANDing acumula o efeito de filtragem de várias uniões. A saída do operador Index ANDing é alimentada em um operador Index ORing, que ordena os Identificadores de Linha e elimina todas as linhas duplicadas que podem ter resultado das uniões alimentando o operador Index ANDing. As linhas da tabela de fato são buscadas utilizando-se um operador Fetch. Finalmente, a tabela de fato reduzida é unida a todas as tabelas de dimensão, para concluir as uniões.

Sugestões de desempenho:

- Crie índices na tabela de fato para cada uma das uniões de tabela de dimensão.
- Assegure que o limite de heap de ordenação seja alto o suficiente para permitir alocação do filtro de bit do operador Index ANDing. Para uniões estrela, isto pode requerer até 12MB, ou 3000 páginas de 4K. Para paralelismo de intrapartição, o filtro de bit é alocado a partir do mesmo segmento de memória compartilhada do dbheap, mas ele é ligado pelo sortheap (e sheapthres na instância). A memória compartilhada é, portanto, controlada por sortheap e sheapthres e pode exigir mais de 12MB.
- Aplique predicados de filtragem nas tabelas de dimensão. Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

SQL Estática

Uma instrução *SQL estática* é embutida no programa aplicativo. Todas estas instruções embutidas devem ser pré-compiladas e ligadas em um *pacote* para que o aplicativo possa ser executado.

Quando o DB2 compila estas instruções, ele cria um plano de acesso para cada uma que for baseada nas estatísticas de catálogo e nos parâmetros de configuração no momento que as instruções foram pré-compiladas e ligadas.

Estes planos de acesso são sempre utilizados quando o aplicativo é executado; eles não mudam até que o pacote seja ligado novamente.

A alternativa para SQL estática é SQL dinâmica.

Áreas de Tabela SMS (System-managed space)

Há dois tipos de áreas de tabela que podem existir em um banco de dados: área gerenciada pelo sistema (SMS) e área gerenciada pelo banco de dados (DMS).

Uma área de tabela SMS é gerenciada pelo sistema operacional, que armazena os dados do banco de dados em um espaço que é atribuído quando uma área de tabela é criada. A definição de table space inclui uma lista de um ou mais caminhos de diretório onde esses dados são armazenados.

O sistema de arquivos gerencia a alocação e o gerenciamento do armazenamento de mídia.

Áreas de tabela SMS e DMS podem coexistir no mesmo banco de dados.

Área de Tabela

É mais fácil gerenciar bancos de dados muito grandes se ele for particionado em partes gerenciadas separadamente denominadas *áreas de tabela*.

Uma área de tabela permite que você atribua a localização de dados a dispositivos lógicos particulares ou a partes deles. Por exemplo, quando uma tabela é criada, você pode especificar que seus índices ou suas colunas longas com dados de objeto grande (LOB) sejam mantidas longe do resto dos dados da tabela.

Uma área de tabela pode ser espalhada em um ou mais dispositivos de armazenamento físico (contêineres) para melhorar o desempenho. Entretanto, recomendamos que todos os dispositivos ou contêineres dentro de uma área de tabela tenha características de desempenho semelhantes.

Uma área de tabela pode ser gerenciada de duas maneiras diferentes: como uma área gerenciada pelo sistema (SMS) ou como uma área gerenciada pelo banco de dados (DMS).

Visual Explain

Nota: Como na Versão 6, o Visual Explain não pode mais ser iniciado a partir da linha de comandos. Entretanto, ele ainda pode ser iniciado a partir de vários objetos de banco de dados no Centro de Controle. Para esta versão, a documentação continua a utilizar o nome Visual Explain.

O Visual Explain permite que você visualize o plano de acesso das instruções SQL explicadas, como um gráfico. Você pode utilizar as informações disponíveis no gráfico para ajustar suas consultas SQL para melhorar o desempenho.

Um gráfico do plano de acesso mostra detalhes de:

- Tabelas (e suas colunas associadas) e índices
- Operadores (como varreduras, ordenações e uniões de tabela)
- Áreas de tabelas e funções.

Você também pode utilizar o Visual Explain para:

- Exibir as características que foram utilizadas no momento da otimização. Você pode comparar estas estatísticas às estatísticas do catálogo atual para ajudá-lo a determinar se a nova união do pacote pode melhorar o desempenho.
- Determinar se um índice foi ou não utilizado para acessar uma tabela. Se um índice não tiver sido utilizado, o Visual Explain pode ajudá-lo a determinar quais colunas podem ser beneficiadas se forem indexadas.
- Exibir os efeitos de se realizar várias técnicas de ajuste comparando as versões anteriores e posteriores do gráfico do plano de acesso para uma consulta.
- Obter informações sobre cada operação no plano de acesso, incluindo o custo estimado total e o número de linhas recuperadas (cardinalidade).

Apêndice B. Lista em Ordem Alfabética dos Operadores do Visual Explain

CMPEXP

Nome do operador: CMPEXP

Representa: A computação de expressões exigidas para resultados intermediários ou finais.

(Este operador é apenas para o modo de depuração).

DELETE

Nome do operador: DELETE

Representa: A exclusão de linhas de uma tabela.

Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre-se em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas a serem excluídas.

Sugestão de Desempenho:

- Se você estiver excluindo todas as linhas de uma tabela, considere a instrução `DROP TABLE` ou o comando **LOAD REPLACE**.

EISCAN

Nome do operador: EISCAN

Representa: Este operador varre um índice definido pelo usuário para produzir um fluxo reduzido de linhas. A varredura utiliza as várias condições iniciar/parar da função criadora de intervalo fornecida pelo usuário.

Esta operação é realizada para diminuir o conjunto de linhas qualificadas antes de acessar a tabela base (com base em predicados).

Sugestão de Desempenho:

- Com o tempo, atualizações de banco de dados podem fazer com que um índice torne-se fragmentado, resultando em mais páginas de índice que o necessário. Isto pode ser corrigido eliminando e recriando o índice, ou reorganizando-o.

- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando `runstats`

FETCH

Nome do operador: FETCH

Representa: A busca colunas de uma tabela utilizando um identificador de linha (RID) específico.

Sugestões de desempenho:

- Expanda as chaves de índice para incluir as colunas buscadas para que as páginas de dados não tenham que ser acessadas.
- Encontre um índice relacionado à busca e dê um clique duplo em seu nó para exibir sua janela de estatísticas. Assegure que o grau de cluster seja alto para o índice.
- Aumente o tamanho do buffer se o valor de entrada/saída (E/S) criado pela busca for maior que o número de páginas na tabela.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando `runstats`.

O quantil e as estatísticas de valor freqüentes oferecem informações sobre a seletividade de predicados, o que determina quando varreduras de índice são escolhida em varreduras de tabela. Para atualizar estas estatísticas, utilize o comando **runstats** em uma tabela com a cláusula WITH DISTRIBUTION.

FILTER

Nome do operador: FILTER

Representa: A aplicação de predicados residuais para que dados sejam filtrados com base nos critérios fornecidos pelos predicados.

Sugestões de desempenho:

- Assegure que você tenha utilizado predicados que recuperam apenas os dados necessários. Por exemplo, assegure que o valor de seletividade dos predicados represente a parte da tabela que você quer que seja retornada.
- Assegure que a classe de otimização seja pelo menos 3 para que o otimizador utilize uma união ao invés de uma subconsulta. Se isto não for possível, tente escrever novamente a consulta SQL à mão para eliminar a subconsulta. Para obter um exemplo, consulte a seção sobre regravações de consultas pelo compilador SQL no *Administration Guide*.

GENROW

Nome do operador: GENROW

Representa: Uma função interna que gera uma tabela de linhas, não utilizando entrada de tabelas, índices ou operadores.

GENROW pode ser utilizado pelo otimizador para gerar linhas de dados (por exemplo, para uma instrução INSERT ou para algumas listas IN que são transformadas em uniões).

Para exibir as estatísticas estimadas para as tabelas geradas pela função GENROW, dê um clique duplo em seu nó.

GRPBY

Nome do operador: GRPBY

Representa: O agrupamento de linhas de acordo com valores comuns de colunas ou funções designadas. Esta operação é necessária para produzir um grupo de valores ou para avaliar funções definidas.

Se nenhuma coluna GROUP BY for especificada, o operador GRPBY ainda pode ser utilizado se houver funções de agregação na lista SELECT, indicando que a tabela inteira foi tratada como um único grupo quando esta agregação foi feita.

Sugestões de desempenho:

- Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas para serem agrupadas.
- Para melhorar o desempenho de uma instrução SELECT que contenha uma única função de agregação mas nenhuma cláusula GROUP BY, tente o seguinte:
 - Para uma função de agregação MIN(C), crie um índice ascendente em C.
 - Para uma função de agregação MAX(C), crie um índice descendente em C.

HSJOIN

Nome do operador: HSJOIN

Representa: Uma união hash para a qual as linhas qualificadas de tabelas tem hash produzido para permitir união direta, sem pré-ordenação do conteúdo das tabelas.

Uma união é necessária sempre que houver mais que uma tabela com referência em uma cláusula FROM. Uma união de hash é possível sempre que houver um predicado de união que iguale colunas de duas tabelas diferentes. Os predicados de união precisam ter exatamente o mesmo tipo de dados. Uniões de hash também surgem de uma subconsulta escrita novamente, como é o caso de NLJOIN.

Uma união de hash não requer que as tabelas de entrada sejam ordenadas. A união é realizada varrendo-se a tabela interna da união de hash e gerando-se uma tabela de consulta produzindo hash dos valores de coluna de união. Ele lê a tabela externa, produzindo hash dos valores de coluna de união e verificando na tabela de consulta gerada para a tabela interna.

Para obter mais informações, consulte a seção sobre conceitos de união no *Administration Guide*.

Sugestões de desempenho:

- Utilize predicados locais (ou seja, predicados que façam referência a uma tabela) para reduzir o número de linhas a serem unidas.
- Aumente o tamanho do heap de ordenação para torná-lo grande o suficiente para conter a tabela de consulta de hash na memória.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

INSERT

Nome do operador: INSERT

Representa: A inserção de linhas em uma tabela.

Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas para serem inseridas.

IXAND

Nome do operador: IXAND

Representa: A execução de AND dos resultados de várias varreduras de índice utilizando técnicas de Bitmap Dinâmico. O operador permite que predicados com AND sejam aplicados a vários índices, para reduzir acessos de tabela básicos ao mínimo.

Este operador é executado para:

- Reduzir o conjunto de linhas antes de acessar a tabela base.

- Executar AND de predicados aplicados a vários índices
- Executar AND nos resultados de semi-uniões, utilizados em uma união estrela.

Sugestões de desempenho:

- Com o tempo, atualizações de banco de dados podem fazer com que um índice torne-se fragmentado, resultando em mais página de índice que o necessário. Isto pode ser corrigido eliminando e recriando o índice, ou reorganizando-o.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando `runstats`.
- Em geral, varreduras de índices são mais eficientes quando apenas algumas linhas são qualificadas. Para estimar o número de linhas qualificadas, o otimizador utiliza as estatísticas que estão disponíveis para as colunas com referência em predicados. Se alguns valores ocorrerem mais freqüentemente que outros, é importante solicitar estatísticas de distribuição utilizando a cláusula `WITH DISTRIBUTION` para o comando `runstats`. Utilizando as estatísticas de distribuição não uniformes, o otimizador pode distinguir entre valores que ocorrem freqüentemente e não freqüentemente.
- `IXAND` pode explorar melhor índices de coluna únicos, já que chaves de início e parada são cruciais no uso de `IXAND`.
- Para uniões estrela, crie índices de coluna única para cada uma das colunas mais seletivas na tabela de fato e nas tabelas de dimensões relacionadas.

IXSCAN

Nome do operador: IXSCAN

Representa: A varredura de um índice definido pelo usuário para produzir um fluxo reduzido de linhas. A varredura pode utilizar condições de início/parada otimizados ou pode aplicar-se a predicados indexáveis que fazem referência a colunas do índice.

Esta operação é realizada para diminuir o conjunto de linhas qualificadas antes de acessar a tabela base (com base em predicados).

Para obter mais informações, consulte a seção sobre varreduras de índice no *Administration Guide*.

Sugestões de desempenho:

- Com o tempo, atualizações de banco de dados podem fazer com que um índice torne-se fragmentado, resultando em mais páginas de índice que o necessário. Isto pode ser corrigido eliminando e recriando o índice, ou reorganizando-o.

- Quando duas ou mais tabelas estão sendo acessadas, o acesso à tabela interna através de um índice pode ser mais eficiente fornecendo um índice na coluna de união da tabela externa.

Para obter mais instruções sobre índices, consulte a ajuda online do Visual Explain.

- Se as estatísticas não estiverem atualizadas, atualize-as utilizando o comando `runstats`.
- Em geral, varreduras de índices são mais eficientes quando apenas algumas linhas são qualificadas. Para estimar o número de linhas qualificadas, o otimizador utiliza as estatísticas que estão disponíveis para as colunas com referência em predicados. Se alguns valores ocorrerem mais freqüentemente que outros, é importante solicitar estatísticas de distribuição utilizando a cláusula `WITH DISTRIBUTION` para o comando `runstats`. Utilizando as estatísticas de distribuição não uniformes, o otimizador pode distinguir entre valores que ocorrem freqüentemente e não freqüentemente.

MSJOIN

Nome do operador: MSJOIN

Representa: Uma união de combinação para a qual as linhas qualificadas das tabelas externa e interna devem estar na ordem união-predicado. Uma combinação também é denominada uma *união de varredura de combinação* ou *união de combinação ordenada*.

Uma união é necessária sempre que houver mais que uma tabela com referência em uma cláusula `FROM`. Uma união de combinação é possível sempre que houver um predicado de união que iguale colunas de duas tabelas diferentes. Também pode ser criada a partir de uma subconsulta escrita novamente.

Uma união de combinação requer colunas de entrada e saída ordenadas, já que as tabelas são geralmente varridas apenas uma vez. Esta entrada ordenada é obtida acessando-se um índice ou uma tabela ordenada.

Para obter mais informações, consulte a seção sobre conceitos de união no *Administration Guide*.

Sugestões de desempenho:

- Utilize predicados locais (ou seja, predicados que façam referência a uma tabela) para reduzir o número de linhas a serem unidas.

Para obter instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

NLJOIN

Nome do operador: NLJOIN

Representa: Uma união de loop aninhado que varre (geralmente com uma varredura de índice) a tabela interna uma vez para cada linha da tabela externa.

Uma união é necessária sempre que houver mais que uma tabela com referência em uma cláusula FROM. Uma união de loop aninhado não requer um predicado de união, mas geralmente tem melhor desempenho com um.

Uma união de loop aninhado é realizada:

- Varrendo-se a tabela interna para cada linha acessada da tabela externa.
- Realizando-se uma consulta de índice na tabela interna para cada linha acessada da tabela externa.

Para obter mais informações, consulte a seção sobre conceitos de união no *Administration Guide*.

Sugestões de desempenho:

- Uma união de loop aninhado provavelmente será mais eficiente se houver um índice nas colunas união-predicado da tabela interna (a tabela exibida à direita do operador NLJOIN). Verifique se a tabela interna é uma TBSCAN ao invés de IXSCAN. Se for, considere incluir um índice em suas colunas de união.

Outra maneira (menos importante) de tornar a união mais eficiente é criar um índice nas colunas de união da tabela externa para que ela seja ordenada.

Para obter mais instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

Informações relacionadas:

- Uniões Estrela.

PIPE

Nome do operador: PIPE

Representa: A transferência de linhas para outros operadores sem alterações às linhas.

(Este operador é apenas para o modo de depuração).

RETURN

Nome do operador: RETURN

Representa: O retorno de dados de uma consulta para o usuário. Este é o operador final no gráfico do plano de acesso e mostra os valores e custos totais acumulados do plano de acesso.

Este operador representa uma operação necessária.

Sugestão de Desempenho:

- Assegure que você tenha utilizado predicados que recuperem apenas os dados necessários. Por exemplo, assegure que o valor de seletividade dos predicados represente a parte da tabela que você quer que seja retornada.

RIDSCN

Nome do operador: RIDSCN

Representa: A varredura de uma lista de identificadores de linha (RIDs) obtidos de um ou mais índices.

Este operador é considerado pelo otimizador quando:

- Predicados são conectados por palavras-chave OR ou existe um predicado IN. Pode-se utilizar uma técnica denominada ORing de índice, que combina resultados de vários acessos de índices na mesma tabela.
- É proveitoso utilizar pré-busca de lista para um único acesso de índice, já que ordenar os identificadores de linha antes de acessar as linhas base tornam a E/S mais eficiente.

RQUERY

Nome do operador: SHIP

Representa: Um operador utilizado no sistema federado para recuperar dados de uma origem de dados remota. Este operador é considerado pelo otimizador quando: Um operador SHIP envia uma instrução SQL SELECT para uma origem de dados remota para recuperar o resultado da consulta. A

instrução SELECT é gerada utilizando o dialeto SQL suportado pela origem de dados e pode conter qualquer consulta válida, conforme permitido pela origem de dados.

Sugestão de Desempenho: Consulte o Capítulo 4 da publicação *Administration Guide Vol 2, Federated Database Query and Network Tuning Information*.

SORT

Nome do operador: SORT

Representa: A ordenação das linhas em uma tabela na ordem de uma ou mais de suas colunas, eliminando opcionalmente entradas duplicadas.

A ordenação é necessária quando não existem índices que satisfaçam os pedidos solicitados ou quando a ordenação pode ser menos cara que uma varredura de índice. A ordenação geralmente é realizada como uma operação final quando as linhas requeridas são buscadas, ou para ordenar dados anteriores para uma união ou para um agrupamento.

Se o número de linhas for alto ou se os dados ordenados não puderem ter canal processado, a operação requer a geração de tabelas temporárias de alto custo.

Para obter mais informações sobre ordenações, consulte o *Administration Guide*.

Sugestões de desempenho:

- Considere incluir um índice nas colunas de ordenação.
Para obter instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.
- Assegure que você tenha utilizado predicados que recuperem apenas os dados necessários. Por exemplo, assegure que o valor de seletividade dos predicados represente a parte da tabela que você quer que seja retornada.
- Verifique se o tamanho da pré-busca da área de tabela temporária do sistema é adequado, ou seja, não esteja ligado a E/S. (Para verificar isto, selecione **Instrução->Exibir estatísticas->Áreas de Tabelas**).
- Se ordenações grandes forem freqüentemente necessárias, considere aumentar os valores dos seguintes parâmetros de configuração:
 - Tamanho do heap de ordenação (sortheap). Para alterar este parâmetro, clique com o botão direito no banco de dados no Centro de Controle e selecione *Configurar* neste menu popup. Selecione a guia **Desempenho** no bloco de notas que é exibido.

- Limite de heap de ordenação (sheaphres). Para alterar este parâmetro, clique com o botão direito na instância de banco de dados no Centro de Controle e selecione *Configurar* neste menu popup. Selecione a guia **Desempenho** no bloco de notas que é exibido.
- Se as estatísticas não estiverem atualizadas, atualize-as. utilizando o comando runstats.

TBSCAN

Nome do operador: TBSCAN

Representa: Uma varredura de tabela (varredura relacional) que recupera linhas lendo todos os dados necessários diretamente das páginas de dados.

Este tipo de varredura é escolhida pelo otimizador em uma varredura de índice quando:

- O intervalo de valores varridos ocorre com freqüência (ou seja, a maioria das tabelas devem ser acessada)
- A tabela for pequena
- O armazenamento em cluster for baixo
- Um índice não existir

Para obter mais informações sobre varreduras de tabelas e de índices, consulte o *Administration Guide*.

Sugestões de desempenho:

- Uma varredura de índice pode ser mais eficiente que uma varredura de tabela se a tabela for grande e a maioria de suas linhas não forem acessadas. Para aumentar a possibilidade da varredura de índice ser utilizada pelo otimizador para esta situação, considere incluir índices em colunas para as quais há predicados seletivos.

Para obter mais instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

- Se um índice já existir mas não tiver sido utilizado, verifique se existem predicados seletivos em cada uma de suas colunas iniciais. Se estes predicados existirem, verifique se o grau de armazenamento em cluster é alto para o índice. (Para ver estas estatísticas, abra a janela Estatísticas de Tabela da tabela abaixo da ordenação e selecione o botão *Índices* para exibir a janela Estatísticas de Índice).
- Verifique se o tamanho da pré-busca da área de tabela é adequado, ou seja, não esteja ligado a E/S. (Para verificar isto, selecione **Instrução->Exibir estatísticas->Áreas de Tabelas**).

Para obter mais informações, consulte a seção sobre pré-busca de dados no conjunto de buffers no *Administration Guide*.

- Se as estatísticas não estiverem atualizadas, atualize-as utilizando o comando `runstats`.

O quantil e as estatísticas de valor freqüentes oferecem informações sobre a seletividade de predicados. Por exemplo, estas estatísticas podem ser utilizadas para determinar quando varreduras de índice são escolhidas em varreduras de tabelas. Para atualizar estes valores, utilize o comando `runstats` em uma tabela com a cláusula `WITH DISTRIBUTION`.

TEMP

Nome do operador: TEMP

Representa: A ação de ordenar dados em uma tabela temporária, para serem lidos posteriormente por outro operador (possivelmente várias vezes). A tabela é removida depois que a instrução SQL é processada, se não tiver sido removida antes.

Este operador é necessário para avaliar subconsultas ou para armazenar resultados intermediários. Em algumas situações (como quando a instrução pode ser atualizada), ela pode ser obrigatória.

TQUEUE

Nome do operador: TQUEUE

Representa: Uma fila de tabela que é utilizada para passar dados de tabela de um agente de banco de dados para outro quando há vários agentes de banco de dados processando uma consulta. Vários agentes de banco de dados são utilizados para processar uma consulta quando paralelismo está envolvido.

Os tipos de filas de tabela são:

- **Local:** A fila de tabela é utilizada para passar dados entre agentes de banco de dados dentro de um único nó. Uma fila de tabela local é utilizada para paralelismo intrapartição.
- **Não Local:** A fila de tabela é utilizada para passar dados entre agentes de banco de dados em nós diferentes.

UNION

Nome do operador: UNION

Representa: A concatenação de fluxos de linhas de várias tabelas.

Este operador representa uma operação necessária. Para melhorar os custos de planos de acesso, concentre-se em outros operadores (como varreduras e uniões) que definam o conjunto de linhas a serem concatenadas.

UNIQUE

Nome do operador: UNIQUE

Representa: A eliminação de linhas tendo valores duplicados para colunas especificadas.

Sugestão de Desempenho:

- Este operador não é necessário apenas se um índice exclusivo existir em colunas apropriadas.

Para obter instruções sobre índices, consulte Criando índices apropriados na ajuda online do Visual Explain.

UPDATE

Nome do operador: UPDATE

Representa: A atualização de dados nas linhas de uma tabela.

Este operador representa uma operação necessária. Para melhorar os custos do plano de acesso, concentre-se em outros operadores (como varreduras ou uniões) que definem o conjunto de linhas para serem atualizadas.

Apêndice C. Conceitos do DB2

Banco de Dados

Um banco de dados relacional apresenta dados como um conjunto de tabelas. Uma tabela é constituída de um conjunto definido de colunas e qualquer número de linhas. Os dados em cada tabela são relacionados logicamente, e os relacionamentos podem ser definidos entre as tabelas. Os dados podem ser exibidos e manipulados com base em princípios e operações matemáticas chamados relacionamentos (como INSERT, SELECT e UPDATE).

Um banco de dados é autodescritivo pois ele contém, além dos dados, uma descrição de sua própria estrutura. Ele inclui um conjunto de tabelas de catálogo, que descrevem a estrutura lógica e física dos dados; um arquivo de configuração, que contém os valores de parâmetro associados ao banco de dados; e um log de recuperação, que registra transações em andamento e transações que podem ser arquivadas.

Bancos de dados podem ser locais ou remotos. Um banco de dados local está localizado fisicamente na estação de trabalho em uso, enquanto um banco de dados em outra máquina é considerado remoto.

Esquemas

Um esquema é um identificador exclusivo utilizado para agrupar um conjunto de objetos de banco de dados (como tabelas, exibições, índices e aliases). Isto é, se você estava criando uma tabela denominada PAYROLL, seria tedioso ter de pesquisar todos os banco de dados para descobrir se algum outro usuário já criou uma tabela com o mesmo nome. O nome de cada objeto precisa ser exclusivo apenas dentro de seu próprio esquema.

A maioria dos objetos de banco de dados possuem um nome de objeto em duas partes, com a primeira parte sendo o nome de esquema e a segunda parte o nome do objeto. Quando um objeto é criado, você poderá atribuí-lo para um esquema específico. Se não especificar um esquema, ele será atribuído ao esquema padrão, que geralmente é o ID de usuário da pessoa que criou o objeto. Por exemplo, um usuário chamado Silva poderá ter uma tabela chamada SILVA.PAYROLL.

O esquema também irá tornar-se um objeto no banco de dados. Ele será criado quando o primeiro objeto no esquema for criado. Um esquema pode ser de propriedade de um indivíduo, e o proprietário poderá controlar o acesso aos dados e aos objetos dentro dele.

Tabelas

Um banco de dados relacional apresenta dados como um conjunto de tabelas. Uma tabela é constituída de dados organizados logicamente em colunas e linhas (mais conhecidos como registros).

Cada tabela possui um nome, e dentro de uma tabela, cada coluna possui um nome. Nenhuma ordenação em específica é mantida entre as linhas de uma tabela, mas as linhas podem ser recuperadas em uma ordem determinada pelos valores em suas colunas. Os dados em uma tabela estão relacionados logicamente. Todos os dados do banco de dados e da tabela são atribuídos a áreas de tabela.

Apêndice D. Avisos

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente que não infrinja os direitos de propriedade intelectual da IBM ou outros direitos legalmente protegidos, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138/146
Botafogo
Rio de Janeiro - RJ
CEP: 22290-240

Para pedidos de licença relacionados a informações de byte duplo (DBCS), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO “NO ESTADO EM QUE SE ENCONTRA” SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE NÃO-VIOLAÇÃO, MERCADO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações, portanto, esta disposição pode não aplicar-se ao Cliente.

Estas informações podem incluir imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a sites não-IBM na Web são fornecidas apenas por conveniência e não representam de forma alguma um endosso a estes sites na Web. Os materiais contidos nestes sites da Web não fazem parte dos materiais deste produto IBM e a utilização desses sites da Web é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este), e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138/146
Botafogo
Rio de Janeiro, RJ
CEP: 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença do Programa Internacional IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas de nível de desenvolvimento e não há garantia de que tais medidas serão iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para o seu ambiente específico.

As informações relativas a produtos não-IBM foram obtidas junto aos fornecedores dos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão do desempenho, compatibilidade nem qualquer outra

reivindicação relacionada a produtos não-IBM. Dúvidas sobre os recursos de produtos não-IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as instruções referentes à direção futura da IBM ou intenções estão sujeitas a alterações ou remoções sem qualquer aviso e representam apenas metas e objetivos.

Estas informações podem conter exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-lo da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos os nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações podem conter programas aplicativos, na linguagem fonte, que ilustram as técnicas de programação em diversas plataformas operacionais. Você pode copiar, modificar e distribuir estes exemplos de programas sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação do aplicativo para a plataforma operacional para a qual os programas de exemplo são criados. Estes exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Cada cópia ou parte deste exemplo de programa ou qualquer trabalho derivado deve incluir um aviso de copyright com os dizeres:

© (*nome da sua empresa*) (*ano*). Partes deste código são derivadas de Exemplo de Programa da IBM Corp. © Copyright IBM Corp. *_digite o ano ou anos_*.
Todos os direitos reservados.

Marcas

Os termos a seguir são marcas da International Business Machines Corporation nos Estados Unidos e/ou em outros países e foram utilizados em pelo menos um dos documentos da biblioteca de documentação do DB2 UDB.

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

Os termos a seguir são marcas ou marcas registradas de outras empresas e foram utilizados em pelo menos um dos documentos da biblioteca de documentação do DB2 UDB:

Microsoft, Windows, Windows NT e o logotipo Windows são marcas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Intel e Pentium são marcas da Intel Corporation nos Estados Unidos e/ou em outros países.

Java e todas as marcas baseadas em Java são marcas da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Outros nomes de empresas, produtos ou serviços podem ser marcas ou marcas de serviço de terceiros.

Índice Remissivo

A

- áreas de tabela
 - definição 70
- DMS
 - definição 50
- áreas de tabela, obtendo
 - estatísticas 10
- áreas de tabela DMS
 - definição 50
- áreas de tabela gerenciadas pelo sistema
 - definição 70
- arquivo/comando
 - EXPLAIN.DDL 1
- arquivos, EXPLAIN.DDL 1

B

- bloqueio de linha
 - bloqueio do cursor 50
- bloqueio do cursor
 - definição 50

C

- classes de otimização de consulta
 - definição 67
- coletando estatísticas atuais para tabelas e índices 17, 35
- colunas em instruções SQL, obtendo estatísticas 11
- comando LIST TABLES 1
- comando SET CURRENT EXPLAIN SNAPSHOT 4
- comando VESAMPL.DDL 2
- comandos, EXPLAIN.DDL 1
- comandos, LIST TABLES 1
- comandos, opção EXPLSNAP em BIND 4
- comandos, SET CURRENT EXPLAIN SNAPSHOT 4
- comandos, VESAMPL.DDL 2
- consultas sem índices e sem estatísticas 14
- contêineres
 - definição 49
- criando índices adicionais em colunas de tabela 27, 44
- criando índices em colunas utilizadas para unir tabelas em uma consulta 21, 39

- custo
 - definição 49

E

- estatísticas para tabelas, índices e funções de tabelas 9
- executando uma consulta sem índices e estatísticas 32
- explique os instantâneos para instruções SQL estáticas, criando 4

F

- função GENROW
 - definição 56, 75
- funções, obtendo estatísticas 10
- funções definidas pelo usuário, obtendo estatísticas 10
- funções integradas, obtendo estatísticas 10

G

- gráfico do plano de acesso, alterando a aparência do 11
- gráfico do plano de acesso, detalhes de objeto 9
- gráfico do plano de acesso, exibindo e utilizando 7
- gráfico do plano de acesso, lendo os símbolos 8
- gráficos do plano de acesso
 - criação
 - definição 47
 - lista de operadores utilizados 52
 - nós
 - definição 49

I

- índices
 - armazenamento em cluster
 - definição 49
 - instantâneos, para Visual Explain 2
 - instantâneos de explicação
 - definição 51
 - instantâneos de explicação, criando 1
 - instantâneos de explicação, exemplo para Visual Explain 2

- instantâneos de explicação para instrução SQL dinâmicas, criando 3
- instruções explicadas
 - definição 52
- instruções explicáveis
 - definição 52
- instruções SQL dinâmicas, criando instantâneos de explicação para 3
- instruções SQL estáticas, criando instantâneos de explicação para 4
- instruções SQL explicadas, selecionando 7

O

- opção EXPLSNAP (no comando BIND) 4
- opções de união, obtendo informações sobre 11
- operador CMPEXP
 - definição 54, 73
- operador DELETE
 - definição 54, 73
- operador EISCAN
 - definição 54, 73
- operador FETCH
 - definição 55, 74
- operador FILTER
 - definição 55, 74
- operador GRPBY
 - definição 56, 75
- operador HSJOIN
 - definição 57, 75
- operador INSERT
 - definição 57, 76
- operador IXAND
 - definição 58, 76
- operador IXSCAN
 - definição 58, 77
- operador MSJOIN
 - definição 59, 78
- operador NLJOIN
 - definição 60, 79
- operador PIPE
 - definição 61, 79
- operador RETURN
 - definição 61, 80
- operador RIDSCN
 - definição 61, 80

- operador SHIP
 - definição 62, 80
- operador SORT
 - definição 62, 81
- operador TBSCAN
 - definição 63, 82
- operador TEMP
 - definição 64, 83
- operador TQUEUE
 - definição 64, 83
- operador UNION
 - definição 65, 83
- operador UNIQUE
 - definição 65, 84
- operador UPDATE
 - definição 65, 84
- operadores
 - definição 52
 - listar 52
- operadores, obtendo detalhes de 10
- operandos
 - definição 52
- otimizador
 - definição 66

P

- pacotes
 - definição 66
- parâmetros de configuração, obtendo informações sobre 11
- plano de acesso, melhorando 13, 31
- planos de acesso
 - definição 47
- predicados
 - definição 66

R

- régua de controle de zoom, para ampliar gráficos do plano de acesso 8

S

- seletividade de predicados
 - definição 68
- SQL dinâmica
 - definição 51
- SQL estática
 - definição 70

T

- tabelas de explicação, criando 1

U

- uniões estrela
 - definição 69

V

- Visual Explain
 - descrição 71

Entrando em Contato com a IBM

Para localizar um escritório da IBM em seu país ou região, acesse o Directory of Worldwide Contacts da IBM no endereço www.ibm.com/planetwide

Informações sobre o Produto

As informações relacionadas aos produtos DB2 Universal Database estão disponíveis por telefone ou através da World Wide Web no endereço www.ibm.com/software/data/db2/udb

Este site contém as informações mais recentes sobre a biblioteca técnica, solicitação de manuais, downloads para Clientes, newsgroups, FixPacks, notícias e links para recursos da Web.

Para obter informações sobre como entrar em contato com a IBM fora dos Estados Unidos, acesse a página Worldwide da IBM no endereço www.ibm.com/planetwide



Impresso em Brazil