

IBM® DB2 Universal Database™



Výukový program modulu Vizuální vysvětlení

Verze 8

IBM® DB2 Universal Database™



Výukový program modulu Vizuální vysvětlení

Verze 8

Než použijete tyto informace a odpovídající produkt, nezapomeňte si přečíst všeobecné informace uvedené v sekci *Upozornění*.

Tento dokument obsahuje informace, které jsou vlastnictvím společnosti IBM. Je poskytován na základě licenčního ujednání a je chráněn autorským zákonem. Informace obsažené v této publikaci neobsahují žádné záruky týkající se produktu a žádný výrok uvedený v této příručce nelze v tomto smyslu interpretovat.

Příručky vydávané společností IBM si můžete objednat v síti Internet nebo prostřednictvím místního zastoupení společnosti IBM.

- Chcete-li si příručky objednat v síti Internet, přejděte na stránky střediska IBM Publications Center na adrese www.ibm.com/shop/publications/order
- Chcete-li zjistit, kde najdete místní zastoupení společnosti IBM, přejděte na stránky IBM Directory of Worldwide Contacts na adrese www.ibm.com/planetwide

Chcete-li si objednat příručky o systému DB2 u oddělení DB2 Marketing and Sales v USA nebo v Kanadě, volejte číslo 1-800-IBM-4YOU (426-4968).

Pokud odešlete informace společnosti IBM, udělujete tím společnosti IBM nevýhradní právo použít nebo distribuovat tyto informace libovolným způsobem, který společnost považuje za odpovídající, bez vyžádání vašeho svolení.

© Copyright International Business Machines Corporation 2000 - 2002. Všechna práva vyhrazena.

Obsah

Informace o tomto výukovém programu . . . v	
Informace specifické pro prostředí	vi
Lekce 1. Vytváření snímků Explain	1
Vytváření tabulek Explain	1
Použití snímků Explain	1
Vytváření snímků Explain pro dynamické příkazy SQL	3
Vytváření snímků Explain pro statické příkazy SQL	4
Další postup	4
Lekce 2. Zobrazení a použití grafu přístupového plánu	5
Zobrazení grafu přístupového plánu výběrem ze seznamu dříve vysvětlených příkazů SQL	5
Čtení symbolů v grafu přístupového plánu.	5
Použití posuvného ovladače zvětšení ke zvětšování částí grafu	6
Získání dalších podrobností o objektech v grafu	7
Získání statistiky pro tabulky, indexy a tabulkové funkce	7
Získání podrobností o operátorech v grafu	8
Získání statistiky pro funkce	8
Získání statistiky pro tabulkové prostory	8
Získání statistiky pro sloupce v příkazu SQL	9
Získání informací o konfiguračních parametrech a volbách vazby	9
Změna vzhledu grafu	9
Další postup	9
Lekce 3. Zlepšení přístupového plánu v databázovém prostředí s jednou oblastí	11
Práce s grafy přístupového plánu	11
Spuštění dotazu bez indexů a statistiky	12
Shromáždění aktuální statistiky pro tabulky a indexy pomocí příkazu runstats	15
Vytvoření indexů pro sloupce použité ke spojení tabulek v dotazu	19
Vytvoření dalších indexů pro sloupce tabulky	24
Další postup	28
Lekce 4. Zlepšení přístupového plánu v databázovém prostředí děleném na oblasti	29
Práce s grafy přístupových plánů	29
Spuštění dotazu bez indexů a statistiky	30
Shromáždění aktuální statistiky pro tabulky a indexy pomocí příkazu runstats	32
Vytvoření indexů pro sloupce použité ke spojení tabulek v dotazu	36
Vytvoření dalších indexů pro sloupce tabulky	40
Další postup	44
Dodatek A. Koncepte modulu Vizualní vysvětlení	45
Přístupový plán	45
Graf přístupového plánu	45
Uzel grafu přístupového plánu	46
Dělení do klastrů	47
Kontejner	47
Náklady	47
Blokování kurzoru	48
Tabulkový prostor prostoru spravovaného databází (DMS)	48
Dynamické příkazy SQL.	48
Snímek Explain	49
Vysvětlitelné příkazy	49
Vysvětlený příkaz.	50
Operand	50
Operátor	50
CMPEXP	51
DELETE	51
EISCAN	52
FETCH	52
FILTER.	53
GENROW	53
GRPBY.	53
HSJOIN	54
INSERT	54
IXAND.	55
IXSCAN	55
MSJOIN	56
NLJOIN	57
PIPE	57
RETURN	57
RIDSCN	58

RPD	58	INSERT	72
SHIP	58	IXAND	72
SORT	59	IXSCAN	73
TBSCAN	60	MSJOIN	73
TEMP	61	NLJOIN	74
TQUEUE	61	PIPE	75
UNION	61	RETURN	75
UNIQUE	62	RIDSCN	75
UPDATE	62	RPD	76
Optimalizátor	62	SHIP	76
Balík	62	SORT	76
Predikát	62	TBSCAN	77
Třída optimalizace dotazu	63	TEMP	78
Selektivita predikátů	64	TQUEUE	78
Hvězdicové spojení	65	UNION	79
Statické příkazy SQL	66	UNIQUE	79
Tabulkové prostory prostoru spravovaného systémem (SMS)	66	UPDATE	79
Tabulkový prostor	66	Dodatek C. Koncepce DB2	81
Vizuální vysvětlení	67	Databáze	81
Dodatek B. Abecední seznam operátorů modulu Vizualní vysvětlení	69	Schémata	81
CMPEXP	69	Tabulky	81
DELETE	69	Dodatek D. Poznámky	83
EISCAN	69	Ochranné známky	86
FETCH	70	Rejstřík	89
FILTER	70	Kontaktování společnosti IBM	91
GENROW	70	Informace o produktu	91
GRPBY	71		
HSJOIN	71		

Informace o tomto výukovém programu

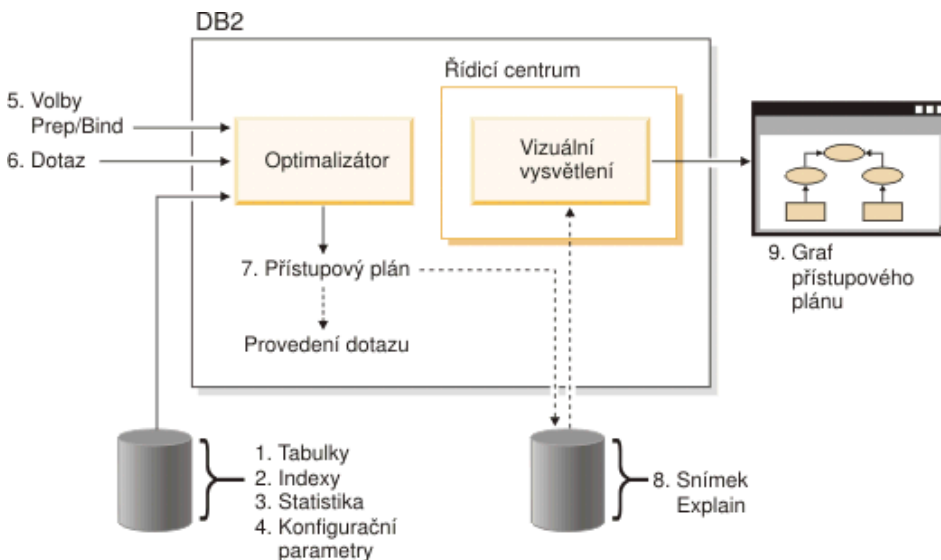
Tento výukový program představuje příručku pro funkce modulu Vizualní vysvětlení produktu DB2. Provedením lekcí v tomto výukovém programu se dozvíte, jak modul Vizualní vysvětlení umožňuje zobrazit přístupový plán pro vysvětlené příkazy SQL pomocí grafu. Dozvíte se rovněž, jak využít informace získané z takového grafu k ladění příkazů SQL, abyste dosáhli lepšího výkonu.

Pomocí svého optimalizátoru prověří produkt DB2 vaše příkazy SQL a určí nejlepší přístup k datům. Tato cesta k datům se nazývá přístupový plán. Produkt DB2 umožňuje prohlížet akce provedené optimalizátorem tím, že vám dovoluje nahlédnout do přístupového plánu vybraného k provedení konkrétního příkazu SQL. Pomocí modulu Vizualní vysvětlení můžete přístupový plán zobrazit jako graf. Graf představuje vizuální znázornění databázových objektů zahrnutých v dotazu (například tabulek a indexů). Zahrnuje také operace provedené na těchto objektech (například procházení a řazení) a zobrazuje tok dat.

Přístup dotazu k datům můžete zlepšit provedením některé nebo všech následujících akcí vyladění:

1. Vyladění návrhu tabulky a reorganizace dat tabulky
2. Vytvoření vhodných indexů
3. Poskytnutí aktuální statistiky pro optimalizátor pomocí příkazu **runstats**
4. Výběr vhodných konfiguračních parametrů
5. Výběr vhodných voleb vazby
6. Návrh dotazů pro načtení pouze požadovaných dat
7. Práce s přístupovým plánem
8. Vytvoření snímků Explain
9. Vylepšení přístupového plánu pomocí grafu přístupového plánu

Tyto akce související s výkonem odpovídají akcím zobrazeným na následující ilustraci. (Přerušovanou čarou jsou označeny povinné akce pro modul Vizuelní vysvětlení.)



Tento výukový program obsahuje lekce k těmto tématům:

- Vytvoření snímků Explain. Ty jsou potřebné pro zobrazení grafů přístupových plánů.
- Zobrazení a manipulace s grafem přístupového plánu.
- Provedení akcí vyladění a zjištění, jak ovlivňují přístupový plán.

Poznámka: Vyladění výkonu se dělí na lekci pro databázová prostředí s jednou oblastí a lekci pro databázová prostředí dělená na oblasti.

Při práci na lekcích budete používat databázi SAMPLE dodanou s produktem DB2. Pokud jste databázi SAMPLE dosud nevytvořili, najdete příslušné informace v příručce *Administration Guide*.

Informace specifické pro prostředí



Informace označené touto ikonou se týkají pouze prostředí databáze s jednou oblastí.



Informace označené touto ikonou se týkají pouze prostředí dělené databáze.

Lekce 1. Vytváření snímků Explain

V této lekci vytvoříte snímky Explain. Prostředek SQL Explain slouží k zachycování informací o prostředí, ve kterém je statický nebo dynamický příkaz SQL kompilován. Zachycené informace umožňují pochopení struktury příkazů SQL a jejich potenciálního výkonu při provádění. Snímek Explain představuje komprimované informace shromážděné při vysvětlení příkazu SQL. Je uložen jako objekt BLOB (binary large object) v tabulce EXPLAIN_STATEMENT a obsahuje následující informace:

- interní reprezentaci přístupového plánu, včetně jeho operátorů a přístupovaných tabulek a indexů,
- kritéria pro rozhodování používaná optimalizátorem, včetně statistických údajů pro databázové objekty a kumulativních nákladů pro každou operaci.

Pro zobrazení grafu přístupového plánu vyžaduje modul Vizualní vysvětlení informace obsažené ve snímku Explain.

Vytváření tabulek Explain

Chcete-li vytvořit snímky Explain, je třeba zajistit, aby pro vaše jméno uživatele existovaly následující tabulky Explain:

- EXPLAIN_INSTANCE
- EXPLAIN_STATEMENT

Chcete-li zkontrolovat, zda existují, použijte příkaz **DB2 list tables**. Pokud tyto tabulky neexistují, je třeba je vytvořit podle následujících instrukcí:

1. Pokud ještě nebyl produkt DB2 spuštěn, zadejte příkaz **db2start**.
2. Z příkazového procesoru DB2 se připojte k databázi, kterou chcete použít. V tomto výukovém programu se připojte k databázi SAMPLE pomocí příkazu **connect to sample**.
3. Vytvořte tabulky Explain, a to pomocí ukázkového příkazového souboru, který je součástí souboru EXPLAIN.DDL. Tento soubor je uložen v adresáři sqllib/misc. Chcete-li spustit příkazový soubor, přejděte do tohoto adresáře a zadejte příkaz **db2 -tf EXPLAIN.DDL**. Tento příkazový soubor vytvoří tabulky Explain, pro které se jako předpona používá jméno připojeného uživatele. Toto jméno uživatele musí mít v databázi oprávnění CREATETAB, anebo SYSADM či DBADM.

Použití snímků Explain

Pro snazší seznámení s modulem Vizualní vysvětlení jsou pro vás k dispozici čtyři ukázkové snímky. Informace o vytváření vlastních snímků naleznete v následujících oddílech; není však nutné vytvářet vlastní snímky pro práci s tímto výukovým programem.

- Vytváření snímků Explain pro dynamické příkazy SQL
- Vytváření snímků Explain pro statické příkazy SQL

Dotaz použitý pro ukázkové snímky uvádí jméno, oddělení a příjmy všech zaměstnanců, kteří nejsou v manažerských funkcích a kteří vydělávají více než 90 % nejvyššího manažerského platu.

```
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
```

Dotaz má dvě části:

1. Poddotaz (v uvozovkách) vytváří řádky dat, které se skládají z 90% podílů platů jednotlivých manažerů. Protože je poddotaz kvalifikován prvkem ALL, bude z této tabulky načtena pouze nejvyšší hodnota.
2. Hlavní dotaz spojí všechny řádky v tabulkách ORG a STAFF, ve kterých se shodují čísla oddělení, prvek JOB se nerovná 'Mgr' a součet platu a provize je vyšší než hodnota vrácená z poddotazu.

Hlavní dotaz obsahuje následující tři predikáty (porovnání):

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')

Tyto predikáty představují (v uvedeném pořadí):

1. Predikát spojení, který spojuje tabulky ORG a STAFF tam, kde se shodují čísla oddělení.
2. Lokální predikát ve sloupci JOB tabulky STAFF.
3. Lokální predikát ve sloupcích SALARY a COMM tabulky STAFF, která používá výsledek poddotazu.


Chcete-li načíst ukázkové snímky, postupujte takto:

1. Pokud ještě nebyl produkt DB2 spuštěn, zadejte příkaz **db2start**.
2. Ověřte, zda ve vaší databázi existují tabulky Explain. Postupujte přitom podle instrukcí v oddíle Vytváření tabulek Explain.
3. Připojte se k databázi, kterou chcete použít. V tomto výukovém programu se připojíte k databázi SAMPLE. Pro připojení k databázi SAMPLE zadejte v příkazovém procesoru DB2 příkaz **connect to sample**.

Pokud není databáze SAMPLE dosud vytvořena, vyhledejte informace v oddíle týkajícím se její instalace v příručce *Administration Guide*.

4. Předdefinované snímky nainportujete spuštěním příkazového souboru produktu DB2 VESAMPL.DDL.

-  - Tento soubor je uložen v adresáři sqllib\samples\ve.

-  - Tento soubor je uložen v adresáři sqllib\samples\ve\inter.

Chcete-li spustit příkazový soubor, přejděte do tohoto adresáře a zadejte příkaz **db2 -tf vesampl.ddl**.

- Tento příkazový soubor je nutné spustit pomocí stejného jména uživatele, které bylo použito k vytvoření tabulek Explain.
- Tento příkazový soubor pouze importuje předdefinované snímky. Nevytváří tabulky nebo data. V tabulkách a datech v databázi SAMPLE se spustí akce vyladení, které budou popsány později (například CREATE INDEX a runstats).

Nyní jste připraveni k zobrazení a použití grafů přístupového plánu.

Vytváření snímků Explain pro dynamické příkazy SQL

Poznámka: Informace o vytváření snímků Explain v tomto oddíle jsou jen pro referenci. Protože jsou vám k dispozici ukázkové snímky Explain, není nutné provádět tuto úlohu, abyste mohli pokračovat ve výukovém programu.

Podle těchto kroků vytvoříte snímek Explain pro dynamický příkaz SQL:

1. Pokud ještě nebyl produkt DB2 spuštěn, zadejte příkaz **db2start**.
2. Ověřte, zda ve vaší databázi existují tabulky Explain. Postupujte přitom podle instrukcí v oddíle Vytváření tabulek Explain.
3. Z příkazového procesoru DB2 se připojte k databázi, kterou chcete použít. Chcete-li se například připojit k databázi SAMPLE, zadejte příkaz **connect to sample**.

Chcete-li databázi SAMPLE vytvořit, vyhledejte informace v oddíle týkajícím se její instalace v příručce *Administration Guide*.

4. Snímek Explain pro dynamický příkaz SQL vytvoříte pomocí jednoho z následujících příkazů z příkazového procesoru DB2:
 - Chcete-li vytvořit snímek Explain bez provedení příkazu SQL, zadejte příkaz **set current explain snapshot=explain**.
 - Chcete-li vytvořit snímek Explain a provést příkaz SQL, zadejte příkaz **set current explain snapshot=yes**.

Tento příkaz nastaví speciální registr Explain. Po jeho nastavení budou ovlivněny všechny následující příkazy SQL. Další informace naleznete v oddílech týkajících se aktuálních snímků Explain v příručce *SQL Reference*.

5. Spusťte příkazy SQL z příkazového procesoru DB2.
6. Chcete-li pro snímek zobrazit graf přístupového plánu, obnovte okno Historie vysvětlených příkazů (dostupné z modulu Řídicí centrum) a klepněte dvakrát na snímek.
7. Volitelné: Chcete-li vypnout funkce snímku, zadejte po spuštění příkazů SQL příkaz **set current explain snapshot=no**.

Vytváření snímků Explain pro statické příkazy SQL

Poznámka: Informace o vytváření snímků Explain v tomto oddíle jsou jen pro referenci. Protože jsou vám k dispozici ukázkové snímky Explain, není nutné provádět tuto úlohu, abyste mohli pokračovat ve výukovém programu.

Podle těchto kroků vytvoříte snímek Explain pro statický příkaz SQL:

1. Pokud ještě nebyl produkt DB2 spuštěn, zadejte příkaz **db2start**.
2. Ověřte, zda ve vaší databázi existují tabulky Explain. Postupujte přitom podle instrukcí v oddíle Vytváření tabulek Explain.
3. Z příkazového procesoru DB2 se připojte k databázi, kterou chcete použít. Chcete-li se například připojit k databázi SAMPLE, zadejte příkaz **connect to sample**.
4. Snímek Explain pro statický příkaz SQL vytvořte pomocí volby EXPLSNAP při vázání nebo přípravě aplikace. Zadejte například příkaz **bind váš_soubor explsnap yes**.
5. Volitelné: Chcete-li pro snímek zobrazit graf přístupového plánu, obnovte okno Historie vysvětlených příkazů (dostupné z modulu Řídicí centrum) a klepněte dvakrát na snímek.

Informace o použití volby EXPLSNAP pro odpovídající rozhraní API naleznete v příslušných oddílech v příručce *Application Development Guide*.

Další postup

V oddíle “Lekce 2. Zobrazení a použití grafu přístupového plánu” na stránce 5 se naučíte zobrazovat graf přístupového plánu a porozumíte jeho obsahu.

Lekce 2. Zobrazení a použití grafu přístupového plánu

V této lekci použijete okno Graf přístupového plánu k zobrazení grafu přístupového plánu a k práci s ním. Graf přístupového plánu je grafická reprezentace přístupového plánu. Můžete v něm zobrazit podrobnosti pro:

- tabulky (a jejich asociované sloupce) a indexy,
- operátory (jako například procházení tabulky, řazení či spojení),
- tabulkové prostory a funkce.

Graf přístupového plánu můžete zobrazit:

- výběrem ze seznamu dříve vysvětlených příkazů,
- výběrem ze seznamu vysvětlitelných příkazů v balíku,
- dynamickým vysvětlením příkazu SQL.

Protože s grafy přístupového plánu budete pracovat na ukázkových snímcích Explain, které jste načeti v Lekci 1, provedete výběr ze seznamu dříve vysvětlených příkazů. Informace o dalších metodách zobrazení grafů přístupového plánu naleznete v nápovědě k modulu Vizualní vysvětlení.

Zobrazení grafu přístupového plánu výběrem ze seznamu dříve vysvětlených příkazů SQL

Chcete-li zobrazit graf přístupového plánu výběrem ze seznamu dříve vysvětlených příkazů, postupujte takto:

1. V modulu Řídicí centrum rozbalte strom objektů, dokud nenaleznete databázi SAMPLE.
2. Klepněte pravým tlačítkem myši na databázi a vyberte příkaz **Zobrazit historii vysvětlených příkazů** z rozevírací nabídky. Otevře se okno Historie vysvětlených příkazů.
3. Graf přístupového plánu lze zobrazit pouze pro příkaz se snímkem Explain. Příkazy, které splňují tuto podmínku, budou mít položku ANO ve sloupci **Snímek Explain**. Klepněte dvakrát na položku označenou jako Číslo dotazu 1 (je možné, že bude nutné přejít doprava, aby se zobrazil sloupec **Číslo dotazu**). Pro příkaz se otevře okno Graf přístupového plánu.

Poznámka: Graf se čte odspodu směrem nahoru. První krok dotazu je uveden na dolním konci grafu a poslední krok je uveden nahoře.

Čtení symbolů v grafu přístupového plánu

Graf přístupového plánu zobrazuje strukturu přístupového plánu jako strom. *Uzly* stromu představují:

- tabulky, zobrazené jako obdélníky,

- indexy, zobrazené jako kosočtverce,
- operátory, zobrazené jako osmiúhelníky, operátory TQUEUE, zobrazené jako rovnoběžníky,
- tabulkové funkce, zobrazené jako šestiúhelníky.

U operátorů je číslo v závorce vpravo od typu operátoru jedinečným identifikátorem každého uzlu. Číslo pod typem operátoru představuje kumulativní náklady.

Použití posuvného ovladače zvětšení ke zvětšování částí grafu

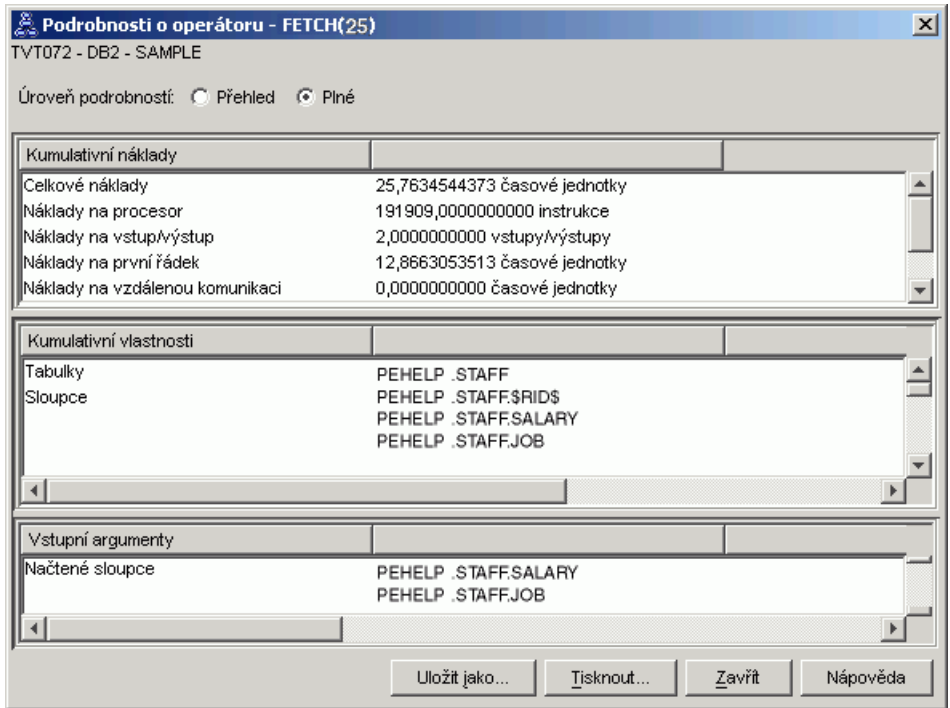
Po zobrazení grafu přístupového plánu se zobrazí celý graf. Je možné, že nebudou vidět podrobnosti odlišující jednotlivé uzly.

Ke zvětšení částí grafu slouží **posuvný ovladač zvětšení** v okně Graf přístupového plánu:

1. Umístěte ukazatel myši na malého jezdce v pruhu posuvného ovladače zvětšení na levé straně grafu.
2. Klepněte levým tlačítkem myši a přetáhněte posuvný ovladač, dokud graf nedosáhne požadovanou úroveň zvětšení.

K zobrazení jiných částí grafu použijte posouvací pruh.

K zobrazení velkého a komplikovaného grafu přístupového plánu použijte okno Přehled grafů. V tomto okně lze snadno zjistit, která část grafu je zobrazena, a lze zde provádět operace zvětšení či posuvu. Zobrazená část přístupového plánu je znázorněna v okénku lupy.



Chcete-li posouvat zobrazenou část grafu, přemístěte ukazatel myši na zvýrazněnou oblast v okně Přehled grafu, stiskněte a podržte tlačítko myši 1 a poté myši vyhledejte požadovanou část přístupového plánu.

Získání dalších podrobností o objektech v grafu

Máte přístup k dalším informacím o objektech v grafu přístupového plánu. Můžete zobrazit:

- statistiku systémového katalogu pro objekty, jako například:
 - tabulky, indexy nebo tabulkové funkce,
 - informace o operátorech, například jejich náklady, vlastnosti a vstupní argumenty,
 - vestavěné funkce nebo funkce definované uživatelem,
 - tabulkové prostory,
 - sloupce odkazované v příkazu SQL,
- informace o konfiguračních parametrech a volbách vazby (parametry optimalizace).

Získání statistiky pro tabulky, indexy a tabulkové funkce

Chcete-li zobrazit statistiku katalogu pro jednu tabulku (obdélník), index (kosočtverec) nebo tabulkovou funkci (šestiúhelník) v grafu, dvakrát klepněte na uzel daného prvku.

Otevře se okno Statistika pro vybrané objekty, zobrazující informace o statistice, které platily v okamžiku vytvoření snímku, i informace, které jsou aktuálně obsaženy v tabulkách systémového katalogu.

Chcete-li zobrazit statistiku katalogu pro *několik* tabulek, indexů nebo tabulkových funkcí v grafu, vyberte každý z těchto prvků klepnutím (zvýrazní se); potom vyberte volbu **Uzel→Zobrazit statistiku**. Pro každý z vybraných objektů se otevře okno Statistika. (Okna se mohou překrývat, takže se může stát, že k získání přístupu ke všem z nich bude nutné opakovaně provést přetažení.)

Pokud položka **STATS_TIME** ve sloupci **Vysvětleno** obsahuje vstup **Statistika nebyla aktualizována**, neexistovala při vytvoření přístupového plánu optimalizátorem žádná statistika. Pokud tedy optimalizátor vyžadoval pro vytvoření přístupového plánu určitou statistiku, použil výchozí hodnoty. Pokud optimalizátor použil výchozí statistiku, bude označena hodnotou (**výchozí**) ve sloupci **Vysvětleno**.

Získání podrobností o operátorech v grafu

Chcete-li zobrazit statistiku katalogu pro jeden operátor (osmiúhelník), klepněte dvakrát na jeho uzel. Pro vybraný operátor se otevře okno Podrobnosti o operátoru, zobrazující informace jako například:

- odhadované kumulativní náklady (vstup/výstup, instrukce procesoru a celkové náklady),
- dosavadní kardinalita (tzn. odhadovaný počet vyhledaných řádků),
- tabulky, ke kterým byl dosud v rámci plánu získán přístup a které byly připojeny,
- sloupce těchto tabulek, ke kterým byl dosud získán přístup,
- dosud použité predikáty, včetně jejich odhadované selektivity,
- vstupní argumenty pro každý operátor.

Chcete-li zobrazit podrobnosti pro *několik* operátorů, vyberte každý z nich klepnutím (zvýrazní se); potom vyberte volbu **Uzel→Zobrazit podrobnosti**. Otevře se okno Statistika pro každý z vybraných objektů. (Okna se mohou překrývat, takže se může stát, že k získání přístupu ke všem z nich bude nutné opakovaně provést přetažení.)

Získání statistiky pro funkce

Chcete-li zobrazit statistiku katalogu pro vestavěné funkce a uživatelské funkce, vyberte volbu **Příkaz→Zobrazit statistiku→Funkce**. Vyberte jednu nebo více položek ze seznamu zobrazeného v okně **Funkce** a klepněte na tlačítko **OK**. Pro každou z vybraných funkcí se otevře okno Statistika funkcí.

Získání statistiky pro tabulkové prostory

Chcete-li zobrazit statistiku katalogu pro tabulkové prostory, vyberte volbu **Příkaz→Zobrazit statistiku→Tabulkové prostory**. Vyberte jednu nebo více položek ze seznamu zobrazeného v okně **Tabulkové prostory** a klepněte na tlačítko **OK**. Pro každý z vybraných tabulkových prostorů se otevře okno Statistika tabulkových prostorů.

Získání statistiky pro sloupce v příkazu SQL

Chcete-li získat statistiku pro sloupce odkazované v příkazu SQL, postupujte takto:

1. Klepněte dvakrát na tabulku v grafu přístupového plánu. Otevře se okno Statistika tabulek.
2. Klepněte na tlačítko **Odkazované sloupce**. Otevře se okno Odkazované sloupce se seznamem sloupců v tabulce.
3. Vyberte jeden nebo více sloupců ze seznamu a klepněte na tlačítko **OK**. Otevře se okno Statistika odkazovaných sloupců pro každý z vybraných sloupců.

Získání informací o konfiguračních parametrech a volbách vazby

Chcete-li zobrazit informace o konfiguračních parametrech a volbách vazby (parametry optimalizace), vyberte volbu **Příkaz→Zobrazit parametry optimalizace** v okně Graf přístupového plánu. Otevře se okno Parametry optimalizace zobrazující informace o hodnotách parametrů, které platily v okamžiku vytvoření snímku, i informace o aktuálních hodnotách.

Změna vzhledu grafu

Chcete-li změnit různé charakteristiky způsobu zobrazení grafu, postupujte takto:

1. V okně Graf přístupového plánu vyberte volbu **Zobrazit→Nastavení**. Otevře se zápisník Nastavení grafu přístupového plánu.
2. Chcete-li změnit barvu pozadí zvolte kartu Graf.
3. Chcete-li změnit barvu různých operátorů, použijte karty Základní, Rozšířené, Aktualizovat a Různé.
4. Chcete-li změnit barvu uzlů tabulek, indexů nebo tabulkových funkcí, vyberte kartu Operand.
5. Chcete-li zadat, který typ informací se zobrazí v uzlech operátorů (typ nákladů nebo kardinality, což je odhadovaný počet dosud vrácených řádků), zvolte kartu Operátor.
6. Chcete-li zadat, zda se v uzlech tabulek mají zobrazovat jména schémat nebo jména uživatelů, vyberte kartu Operand.
7. Chcete-li určit, zda se uzly budou zobrazovat dvojrozměrně nebo trojrozměrně, vyberte kartu Uzel.
8. Chcete-li graf aktualizovat na základě vybraných voleb a nastavení uložit, klepněte na tlačítko **Použít**.

Další postup

Pracujete-li v databázovém prostředí s jednou oblastí, přejděte k oddílu “Lekce 3. Zlepšení přístupového plánu v databázovém prostředí s jednou oblastí” na stránce 11, kde se dozvíte, jak různé akce vyladění mohou změnit a zlepšit přístupový plán.

Pracujete-li v prostředí dělené databáze, přejděte k oddílu “Spuštění dotazu bez indexů a statistiky” na stránce 12, kde se dozvíte, jak různé akce vyladění mohou změnit a zlepšit přístupový plán.

Lekce 3. Zlepšení přístupového plánu v databázovém prostředí s jednou oblastí

V této lekci se dozvíte, jak se přístupový plán a související okna pro základní dotaz mění při provádění různých akcí vyladění. Prostřednictvím řady příkladů doprovázených ilustracemi se dozvíte, jak lze zlepšit odhadované celkové náklady pro přístupový plán (i v případě jednoduchého dotazu) pomocí příkazu **runstats** a přidáním příslušných indexů.

Až získáte s modulem Vizualní vysvětlení zkušenosti, objevíte další způsoby vyladění dotazů.

Práce s grafy přístupového plánu

Čtyři ukázkové snímky Explain nám poslouží jako příklady pro vysvětlení, proč je pro výkon databáze důležité vyladění.

Dotazy spojené se snímky Explain jsou číslovány od 1 do 4. Každý dotaz používá stejný příkaz SQL (popsaný v Lekci 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
```

Ale každá iterace dotazu používá více metod vyladění než jeho předchozí provedení. Například u Dotazu 1 nebylo provedeno žádné vyladění výkonu, zatímco u Dotazu 4 jich bylo provedeno nejvíce. Rozdíly v dotazech jsou popsány níže:

Dotaz 1

Spuštění dotazu bez indexů a statistiky

Dotaz 2

Shromáždění aktuální statistiky pro tabulky a indexy v dotazu

Dotaz 3

Vytvoření indexů ve sloupcích použitých ke spojení tabulek v dotazu

Dotaz 4

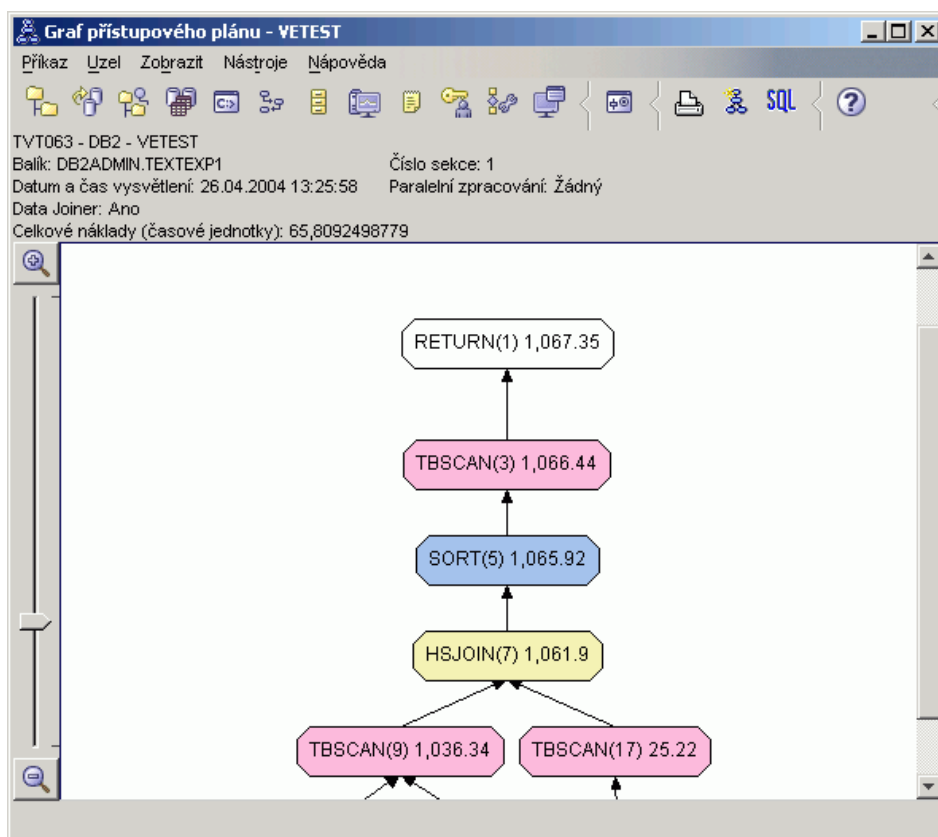
Vytvoření dalších indexů pro sloupce tabulky

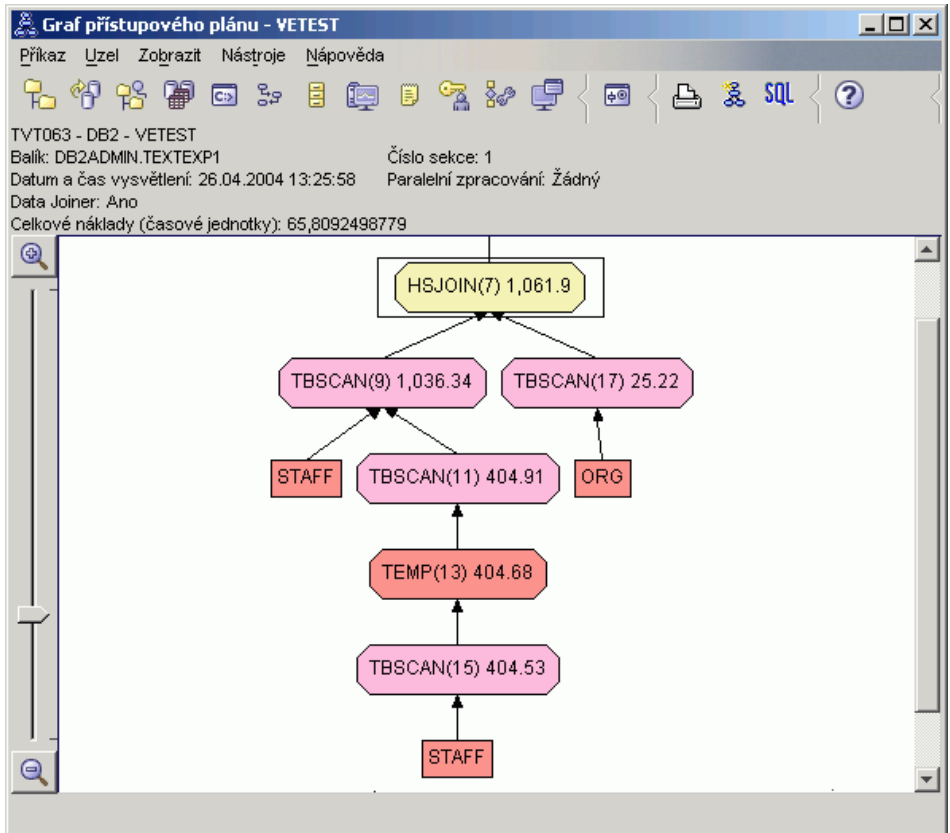
Spuštění dotazu bez indexů a statistiky

V tomto příkladu byl přístupový plán vytvořen pro dotaz SQL bez indexů a statistiky.

Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 1), postupujte takto:

1. V modulu Řídicí centrum rozbalte strom objektů, dokud nenaleznete databázi SAMPLE.
2. Klepněte pravým tlačítkem myši na databázi a vyberte příkaz **Zobrazit historii vysvětlených příkazů** z rozevírací nabídky. Otevře se okno Historie vysvětlených příkazů.
3. Klepněte dvakrát na položku označenou jako Číslo dotazu 1 (je možné, že bude nutné přejít doprava, aby se zobrazil sloupec **Číslo dotazu**). Pro příkaz se otevře okno Graf přístupového plánu.





Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

1. Existuje aktuální statistika pro každou tabulku v dotazu?

Chcete-li ověřit, zda existuje aktuální statistika pro každou tabulku v dotazu, dvakrát klepněte na každý uzel tabulky v grafu přístupového plánu. V okně Statistika tabulek, které se otevře, obsahuje řádek **STATS_TIME** ve sloupci **Vysvětleno** slova "Statistika nebyla aktualizována" v případě, že v okamžiku vytvoření snímku nebyla shromážděna žádná statistika.

Pokud aktuální statistika neexistuje, použijte optimalizátor výchozí statistiku, která se může lišit od skutečné statistiky. Výchozí statistika je označena slovem "výchozí" ve sloupci **Vysvětleno** v okně Statistika tabulek.

Podle informací v okně Statistika tabulek v tabulce **ORG** použil optimalizátor výchozí statistiku (jak je označeno vedle vysvětlených hodnot). Výchozí statistika byla použita, protože skutečná statistika nebyla při vytvoření snímku k dispozici (jak je označeno v řádku **STATS_TIME**).

Statistika tabulek - ORG
 TVT072 - DB2 - SAMPLE
 Tabulka: CKMWDNG.DRG
 Datum a čas vysvětlení: 30.04.2004 19:00:02
 Aktuální datum a čas: 30.04.2004 19:00:03

Statistika	Vysvětleno	Aktuální
CREATE_TIME	28.04.2004 11:02:33	28.04.2004 11:02:33
STATS_TIME	Statistika nebyla aktualizována	28.04.2004 11:02:38
CARD	60(výchozí)	8
NPAGES	1(výchozí)	1
FPAGES	1(výchozí)	1
COLCOUNT	5(výchozí)	5
OVERFLOW	0(výchozí)	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
NESTALÉ	Ne(výchozí)	Ne

Odkazované sloupce Skupiny sloupců Indexy Uložit jako... Tisknout... Zavřít Nápověda

2. Používá tento přístupový plán neefektivnější metody přístupu k datům?

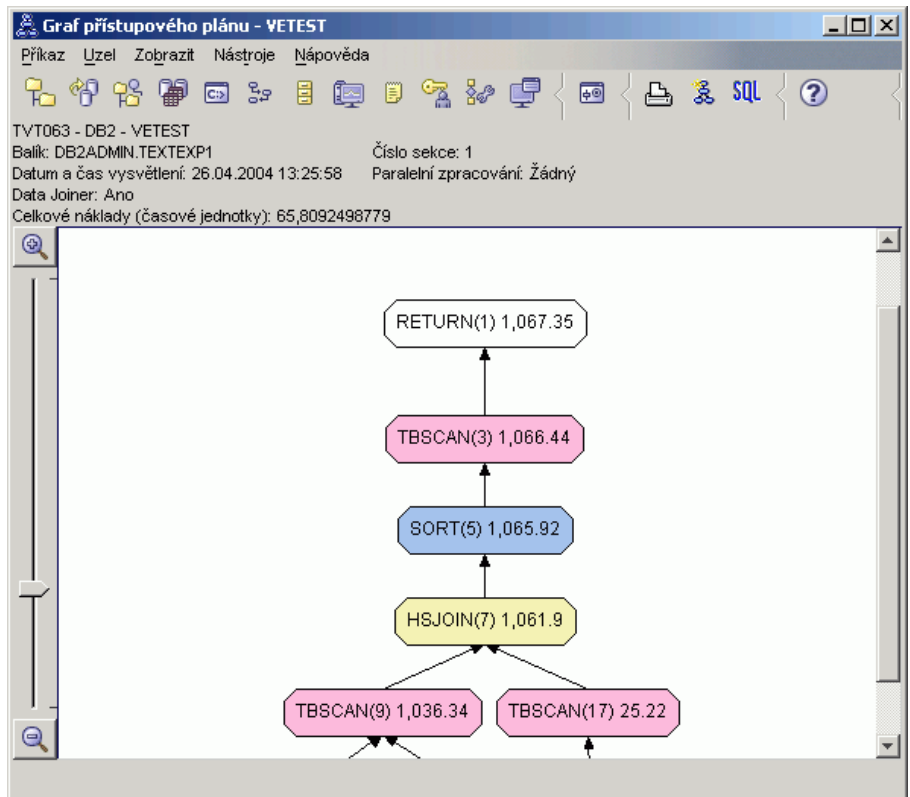
Tento přístupový plán obsahuje procházení tabulky, nikoli procházení indexu. Procházení tabulky jsou zobrazena jako osmiúhelníky a označena hodnotou TBSCAN. Pokud by byla použita procházení indexu, zobrazila by se jako kosočtverce a byla by označena hodnotou IXSCAN. Použití indexu vytvořeného pro tabulku je nákladově efektivnější než procházení tabulky, pokud se extrahují malé objemy dat.

3. Jak efektivní je tento přístupový plán?

Efektivitu přístupového plánu můžete určit, pouze pokud je založena na skutečné statistice. Protože optimalizátor použil v přístupovém plánu výchozí statistiku, nelze zjistit, jak je plán efektivní.

Obecně řečeno, je vhodné si poznačit celkové odhadované náklady na přístupový plán pro pozdější porovnání s revidovanými přístupovými plány. Náklady uvedené v jednotlivých uzlech jsou kumulativní, od prvních kroků dotazu až po daný uzel a včetně něho.

V okně Graf přístupového plánu jsou celkové náklady přibližně 1067 časových jednotek, což je zobrazeno u uzlu **RETURN (1)** v horní části grafu. Celkové odhadované náklady se rovněž zobrazují v horní oblasti okna.



4. Další postup

Dotaz 2 zkoumá přístupový plán pro základní dotaz po spuštění příkazu **runstats**. Použití příkazu **runstats** poskytne optimalizátoru aktuální statistiku ke všem tabulkám, ke kterým dotaz přistupuje.

Shromáždění aktuální statistiky pro tabulky a indexy pomocí příkazu **runstats**

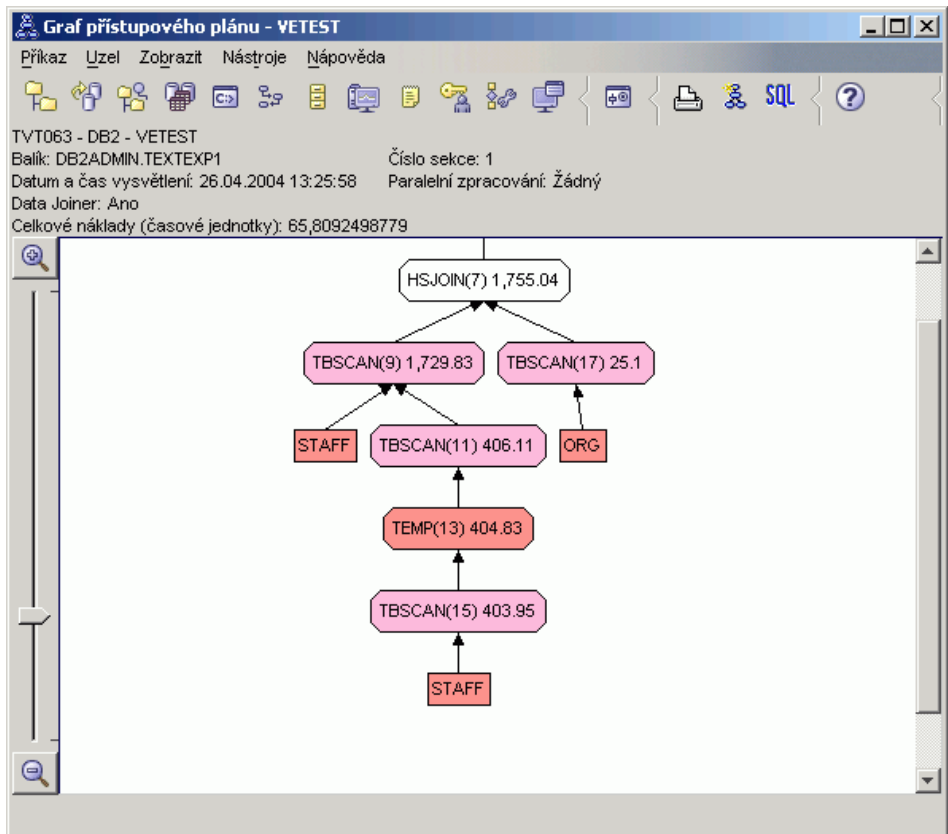
Tento příklad navazuje na přístupový plán popsáný v Dotazu 1. Nyní pro něho bude shromážděna aktuální statistika pomocí příkazu **runstats**.

Důrazně se doporučuje pomocí příkazu **runstats** shromáždít aktuální statistiku tabulek a indexů, zejména pokud od posledního spuštění příkazu **runstats** došlo ke značnému množství aktualizací nebo k vytvoření nových indexů. To optimalizátoru zajišťuje nejpresnější informace, pomocí kterých pak lze určit nejvhodnější přístupový plán. Pokud aktuální statistika není k dispozici, může optimalizátor na základě nepřesné výchozí statistiky zvolit neefektivní přístupový plán.

Příkaz **runstats** je třeba spustit *po* aktualizaci tabulek; v opačném případě může optimalizátor považovat tabulku za prázdnou. Tento problém je evidentní, pokud je

kardinalita v okně Podrobnosti o operátoru rovna nule. V takovém případě dokončete aktualizace tabulky, spusíte znovu příkaz **runstats** a znovu vytvoříte pro příslušné tabulky snímky Explain.

Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 2), postupujte takto: V okně Historie vysvětlených příkazů dvakrát klepněte na položku označenou jako Číslo dotazu 2. Otevře se okno Graf přístupového plánu pro toto provedení příkazu.



Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

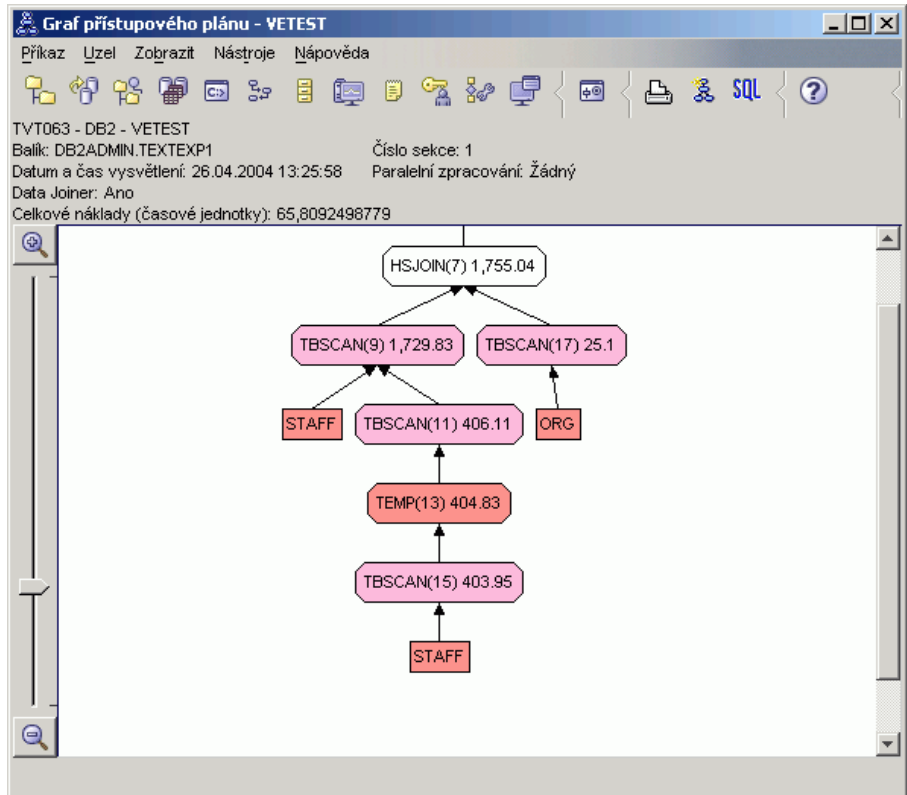
1. Existuje aktuální statistika pro každou tabulku v dotazu?

Okno Statistika tabulek pro tabulku ORG ukazuje, že optimalizátor použil skutečnou statistiku (hodnota **STATS_TIME** je skutečný čas shromáždění statistiky). Přesnost statistiky závisí na tom, zda došlo k zásadním změnám obsahu tabulek od spuštění příkazu **runstats**.

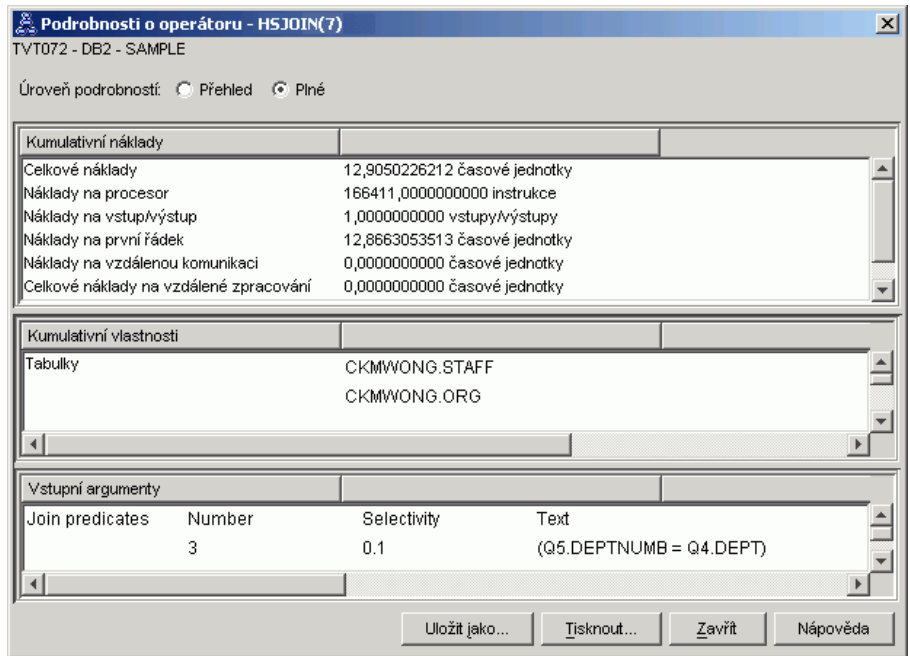
Statistika	Vysvětleno	Aktuální
CREATE_TIME	28.04.2004 11:02:33	28.04.2004 11:02:33
STATS_TIME	28.04.2004 11:02:38	28.04.2004 11:02:38
CARD	8	8
NPAGES	1	1
FPAGES	1	1
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
NESTÁLÉ	Ne(výchozí)	Ne

2. Používá tento přístupový plán nejefektivnější metody přístupu k datům?

Jako v Dotazu 1 používá přístupový plán v Dotazu 2 procházení tabulky (TBSCAN), nikoli procházení indexu (IXSCAN). Ačkoli existuje aktuální statistika, nebylo procházení indexu provedeno, protože ve sloupcích použitých dotazem nejsou žádné indexy. Jedním ze způsobů zlepšení dotazu by bylo zadání indexů pro sloupce použité ke spojení tabulek (tzn. pro sloupce použité v predikátech spojení) do optimalizátoru. V tomto příkladu se jedná o první spojení pro sloučené procházení: HSJOIN (7).



V okně Podrobnosti o operátoru pro operátor HSJOIN (7) si všimněte sekce **Predikáty spojení** v části **Vstupní argumenty**. Sloupce použité v této operaci spojení jsou uvedeny ve sloupci **Text**. V tomto příkladu jsou těmito sloupci DEPTNUMB a DEPT.



3. Jak efektivní je tento přístupový plán?

Přístupové plány založené na aktuální statistice vždy vytvářejí realistický odhad nákladů (měřený v časových jednotkách). Protože odhadované náklady v Dotazu 1 byly založeny na výchozí statistice, nemůžeme porovnat náklady těchto dvou grafů přístupového plánu, abychom určili, který je efektivnější. To, zda jsou náklady nižší nebo vyšší, není relevantní. Abychom mohli efektivitu změřit platným způsobem, je třeba porovnat náklady na přístupové plány založené na skutečné statistice.

4. Další postup

Dotaz 3 zkoumá dopady přidání indexů pro sloupce DEPTNUMB a DEPT. Přidání indexů pro sloupce použité v predikátech spojení může zlepšit výkon.

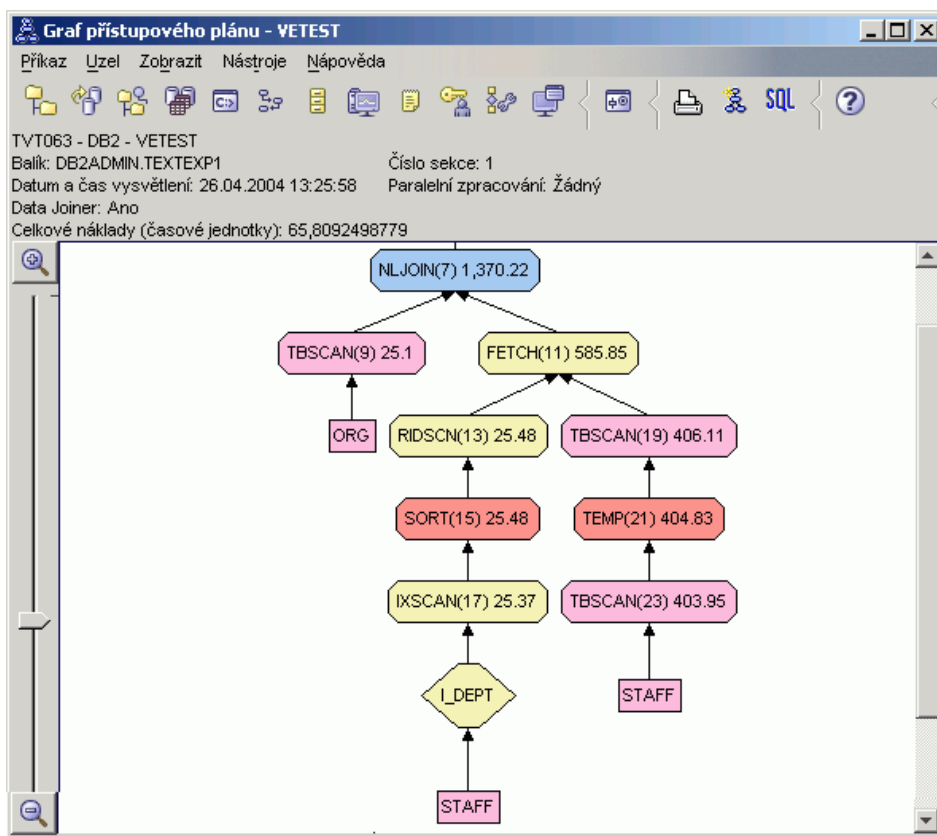
Vytvoření indexů pro sloupce použité ke spojení tabulek v dotazu

Tento příklad navazuje na přístupový plán popsáný v Dotazu 2. Nyní budou vytvořeny indexy pro sloupec DEPT v tabulce STAFF a pro sloupec DEPTNUMB v tabulce ORG.

Poznámka: Ve verzi 8 lze doporučené indexy vytvořit pomocí průvodce Výkon zátěže.

Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 3), postupujte takto: V okně Historie vysvětlených příkazů dvakrát klepněte na položku označenou jako Číslo dotazu 3. Otevře se okno Graf přístupového plánu pro toto provedení příkazu.

Poznámka: Ačkoli byl vytvořen index pro sloupec DEPTNUMB, optimalizátor jej nepoužil.

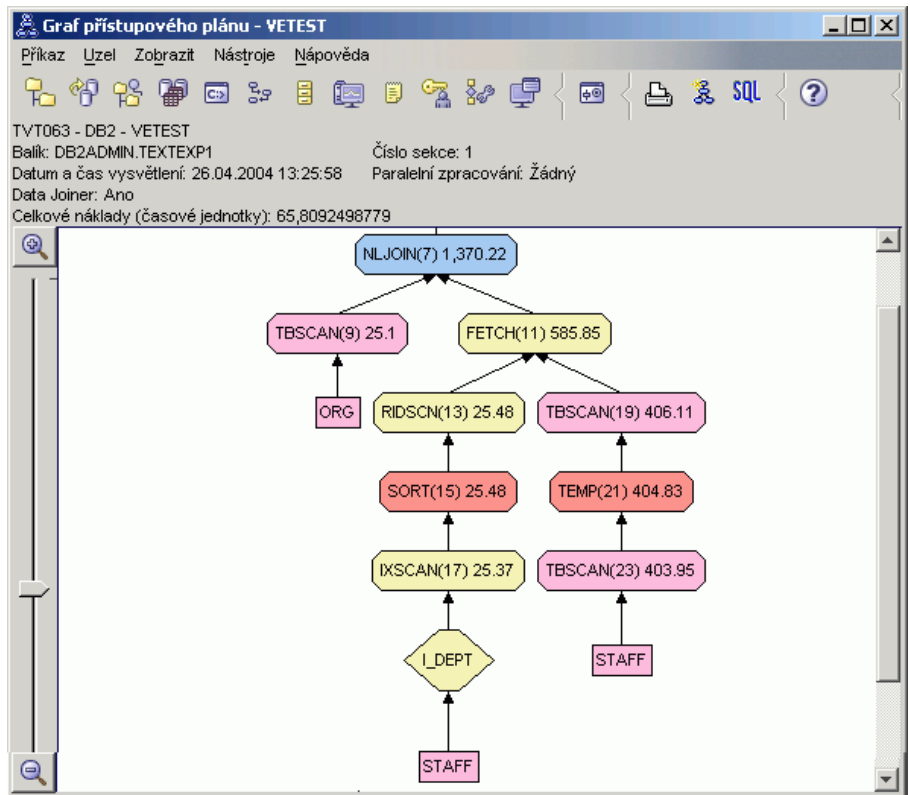


Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

1. Co se v přístupovém plánu změnilo díky indexům?

Vnořené spojení v podobě smyčky NLJOIN (7) nahradilo spojení pro sloučené procházení HSJOIN (7) použité v Dotazu 2. Výsledkem použití vnořného spojení v podobě smyčky byly nižší odhadované náklady než u spojení pro sloučené procházení, protože tento typ spojení nevyžaduje žádný druh dočasných tabulek.

Přímo nad tabulku STAFF byl přidán nový uzel ve tvaru kosočtverce, **I_DEPT**. Tento uzel představuje index vytvořený pro sloupec DEPT a ukazuje, že optimalizátor použil k určení řádků, které se mají načíst, procházení indexu namísto procházení tabulky.



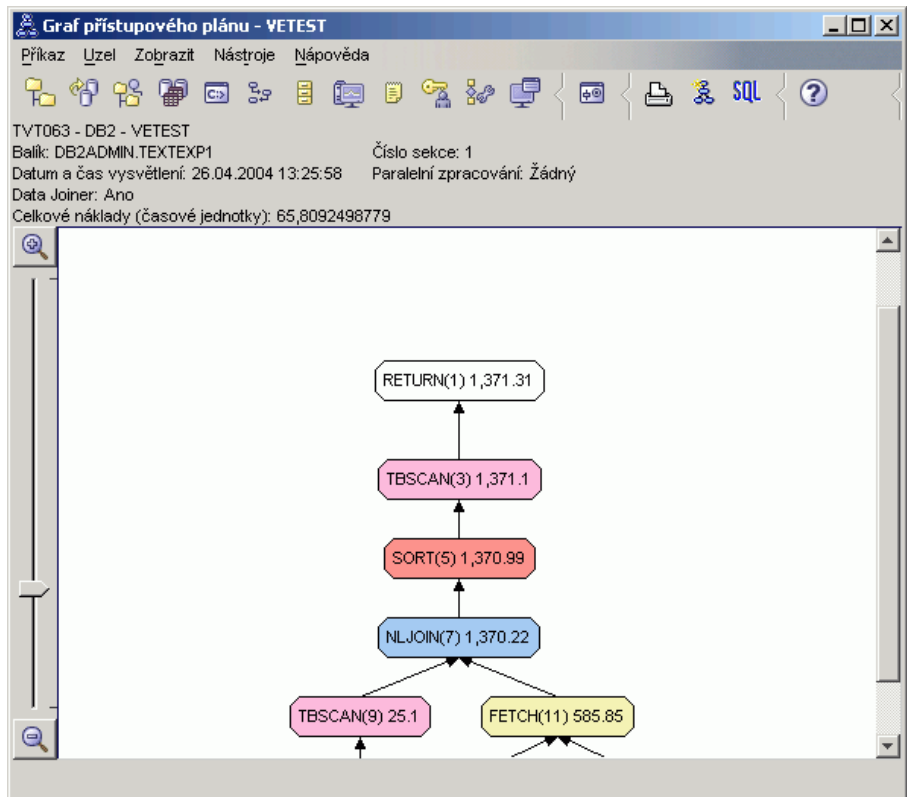
V této části grafu přístupového plánu si všimněte, že byl vytvořen nový index (I_DEPT) ve sloupci DEPT a uzel IXSCAN (17) byl použit pro přístup k tabulce STAFF. V Dotazu 2 bylo pro přístup k tabulce STAFF použito procházení tabulky.

2. Používá tento přístupový plán nejefektivnější metody přístupu k datům?

Výsledkem přidání indexů bylo, že k přístupu k tabulce STAFF byl použit uzel IXSCAN, IXSCAN (17). Dotaz 2 neobsahoval index; proto bylo použito procházení tabulky.

Uzel FETCH, FETCH (11), naznačuje, že kromě použití procházení indexu k načtení sloupce DEPT načtl optimalizátor další sloupce z tabulky STAFF a přitom využil index jako ukazatel. V tomto případě je kombinace procházení indexu a načtení vypočtena jako méně nákladná než úplné procházení tabulky použité v předchozích přístupových plánech.

Poznámka: Uzel tabulky STAFF se zobrazuje dvakrát, aby byl vyjádřen její vztah k indexu pro sloupec DEPT i k operaci FETCH.



Přístupový plán pro tento dotaz ukazuje dopad vytvoření indexů na sloupce zahrnuté do predikátů spojení. Indexy mohou rovněž urychlit aplikaci lokálních predikátů. Podívejme se na lokální predikáty pro každou tabulku v tomto dotazu, abychom zjistili, jak přidání indexů do sloupců odkazovaných v lokálních predikátech může ovlivnit přístupový plán.

V okně Podrobnosti o operátoru pro operátor FETCH (11) si všimněte sloupců v části **Kumulativní vlastnosti**. Sloupec použitý v predikátu pro tuto operaci načtení je JOB, jak je zobrazeno v sekci Predikáty.

Poznámka: Selektivita tohoto predikátu je 0,69. To znamená, že s tímto predikátem bude pro další zpracování vybráno 69 % řádků.

Podrobnosti o operátoru - FETCH(11)
 TVT072 - DB2 - SAMPLE

Úroveň podrobností: Přehled Plně

Kumulativní náklady	
Celkové náklady	25,7634544373 časové jednotky
Náklady na procesor	191909,0000000000 instrukce
Náklady na vstup/výstup	2,0000000000 vstupy/výstupy
Náklady na první řádek	12,8663053513 časové jednotky

Kumulativní vlastnosti	
Tables	CKMWONG.STAFF
Columns	CKMWONG.STAFF.NAME CKMWONG.STAFF.ID CKMWONG.STAFF.COMM CKMWONG.STAFF.SALARY
Order columns	None
Predicates	Number Selectivity Text
	2 0.69 (Q4.JOB <=> Mgr)
	3 0.1 (Q5.DEPTNUMB = Q4.DEPT)
	4 0.5 (Q4.SALARY + Q4.COMM) > ...
Cardinality	37.1
Total buffer pool pages used	23.39

Uložit jako... Tisknout... Zavřít Nápověda

Podrobnosti o operátoru - FETCH(11)
 TVT072 - DB2 - SAMPLE

Úroveň podrobností: Přehled Plně

Kumulativní náklady	

Kumulativní vlastnosti	

Vstupní argumenty	
Sargable predicates	Number Selectivity Text
	2 0.69 (Q4.JOB <=> 'Mgr)
	3 0.1 (Q5.DEPTNUMB = Q4.DEPT)
Residual predicates	Number Selectivity Text
	4 0.5 (Q4.SALARY + Q4.COMM) > ...
Block sargable predicates	None
Direct fetch	False
Prefetch	Number of RIDs for prefetch are 512
Maximum pages	3
Lock intents	Table: Intent Share Row: Next Key Share Block: None

Uložit jako... Tisknout... Zavřít Nápověda

Okno Podrobnosti o operátoru pro operátor FETCH (11) zobrazuje sloupce používané v této operaci. Vidíte, že na prvním řádku vedle položky **Načtené sloupce** v části **Vstupní argumenty** je uvedena položka DEPTNAME.

3. Jak efektivní je tento přístupový plán?

Tento přístupový plán je nákladově efektivnější než přístupový plán z předchozího příkladu. Kumulativní náklady byly sníženy z přibližně 1755 časových jednotek v Dotazu 2 na přibližně 959 časových jednotek v Dotazu 3.

Přístupový plán pro Dotaz 3 však zobrazuje procházení indexu IXSCAN (17) a operátor FETCH (11) pro tabulku STAFF. Ačkoli je procházení indexu kombinované s operací načtení méně nákladné než úplné procházení tabulky, znamená, že pro každý načtený řádek dojde k jednomu přístupu k tabulce a jednomu přístupu k indexu. Nyní vyloučíme tento dvojí přístup k tabulce STAFF.

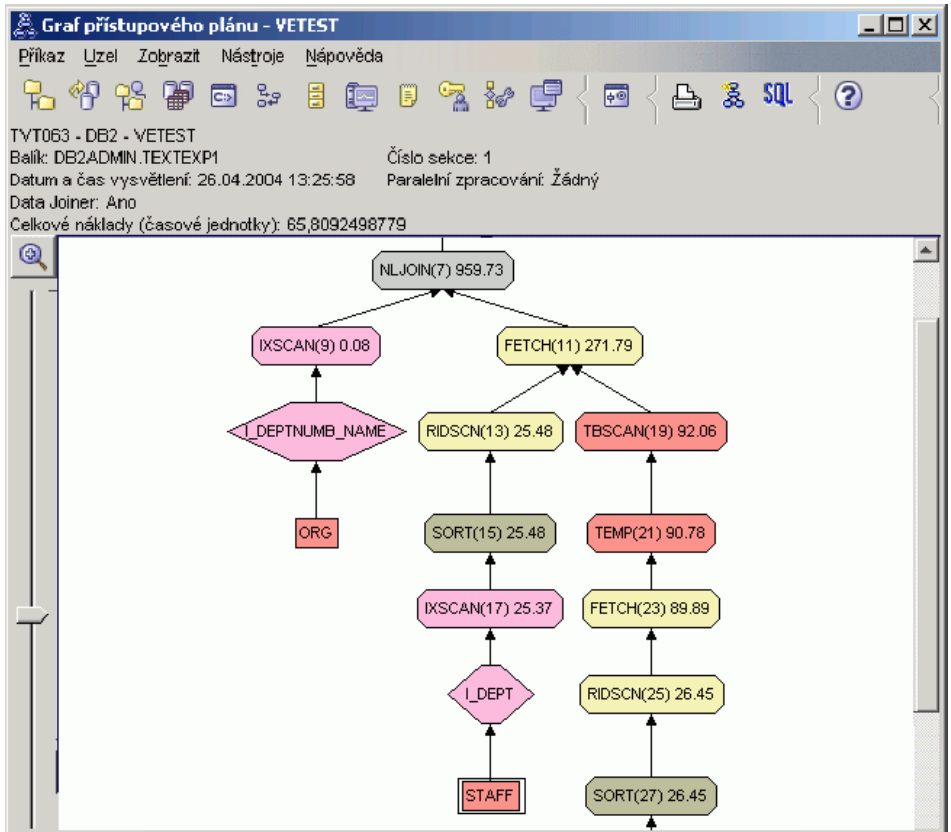
4. Další postup

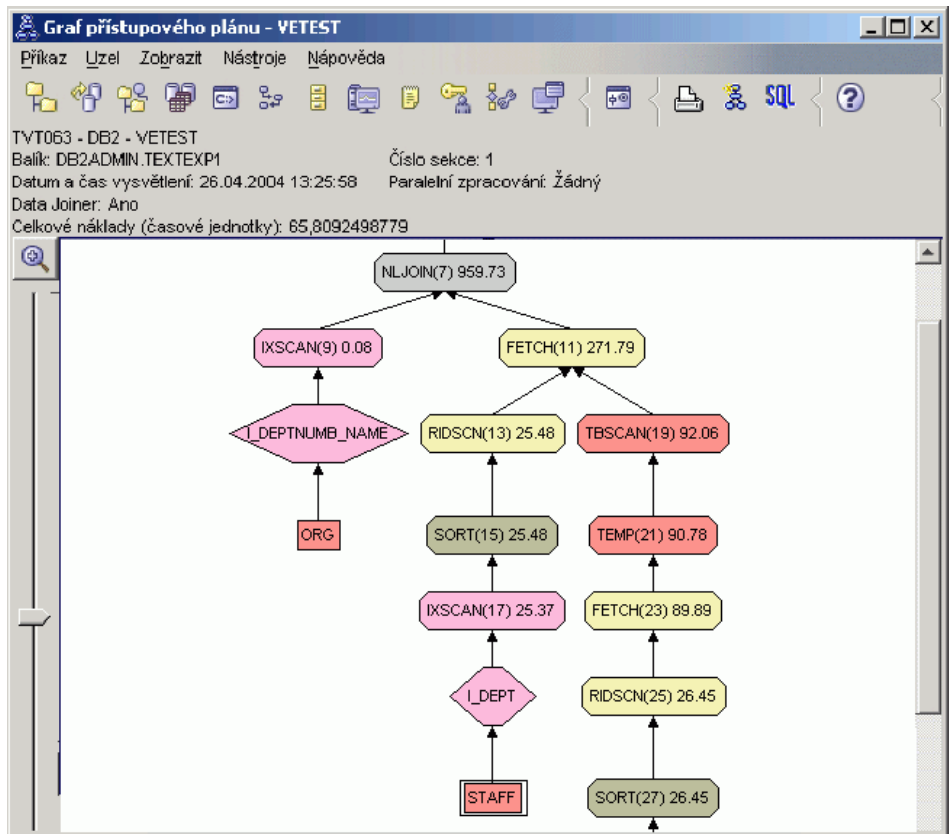
Dotaz 4 redukuje načtení a procházení indexu do jednoho procházení indexu bez načtení. Vytvoření dalších indexů může snížit odhadované náklady přístupového plánu.

Vytvoření dalších indexů pro sloupce tabulky

Tento příklad navazuje na přístupový plán popsáný v Dotazu 3. Nyní je k němu vytvořen index pro sloupec JOB v tabulce STAFF a přidána položka DEPTNAME do existujícího indexu v tabulce ORG. (Přidání samostatného indexu by mohlo způsobit další přístup.)

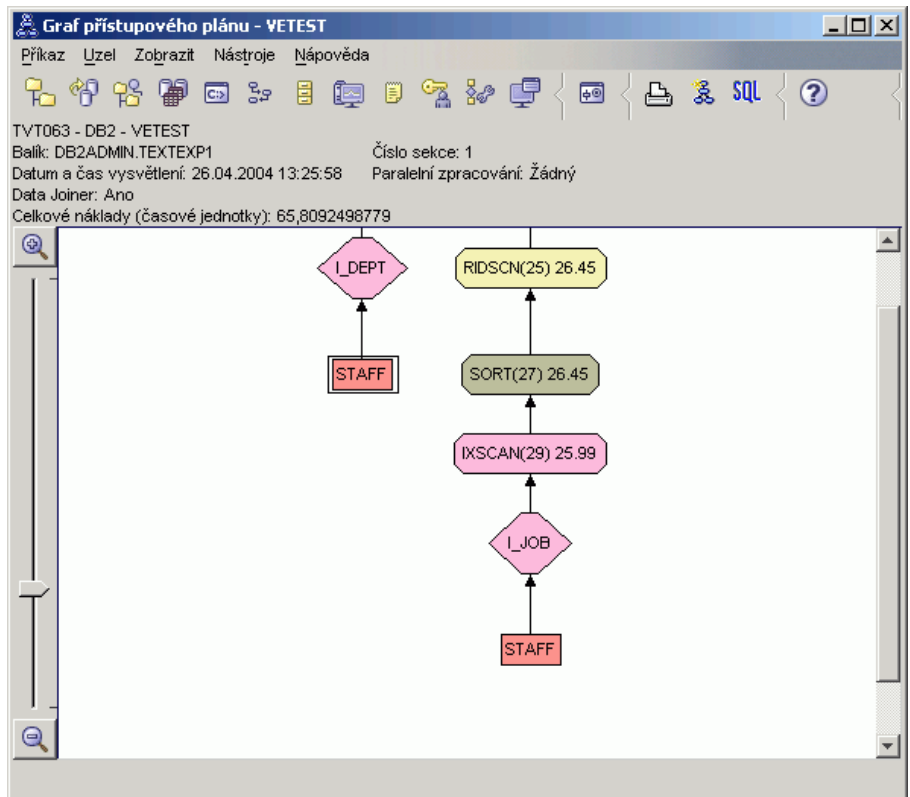
Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 4), postupujte takto: V okně Historie vysvětlených příkazů dvakrát klepněte na položku označenou jako Číslo dotazu 4. Otevře se okno Graf přístupového plánu pro toto provedení příkazu.



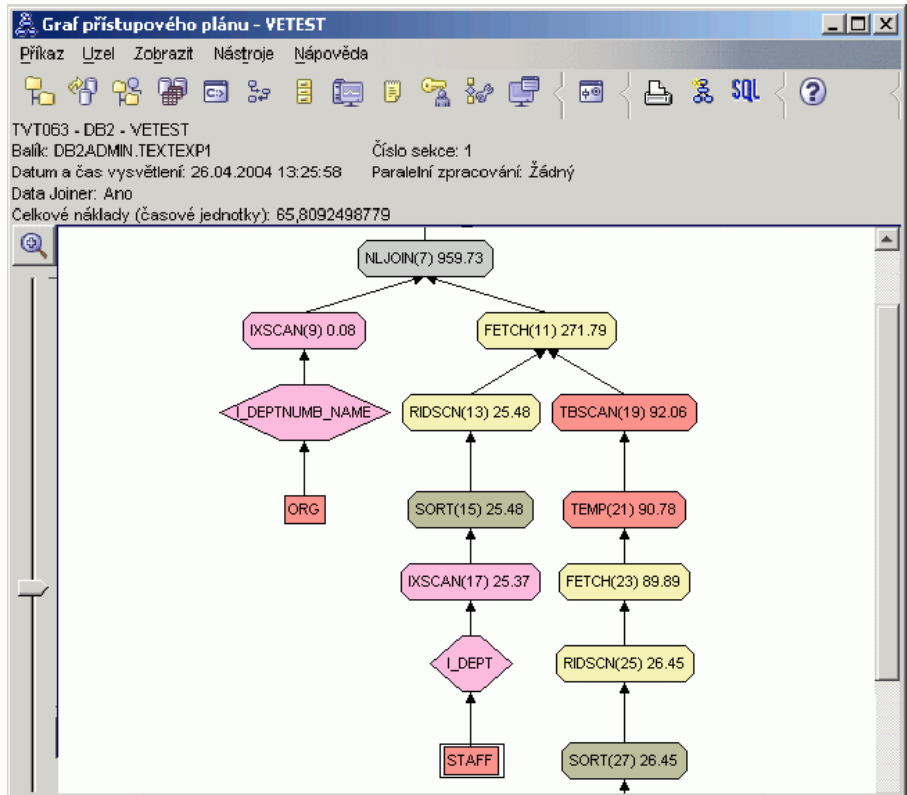


Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

1. Co se v tomto přístupovém plánu změnilo jako výsledek vytvoření dalších indexů?
 Optimalizátor využil index vytvořený ve sloupci JOB v tabulce STAFF (reprezentovaný kosočtvercem označeným položkou **I_JOB**) pro další upřesnění tohoto přístupového plánu.



Ve střední části grafu přístupového plánu si všimněte, že u tabulky ORG bylo předchozí procházení indexu a načtení změněno na pouhé procházení indexu IXSCAN (9). Přidání sloupce DEPTNAME do indexu v tabulce ORG umožnilo optimalizátoru vyloučit nadbytečný přístup zahrnující načtení.



2. Jak efektivní je tento přístupový plán?

Tento přístupový plán je nákladově efektivnější než přístupový plán z předchozího příkladu. Kumulativní náklady byly sníženy z přibližně 1370 časových jednotek v Dotazu 3 na přibližně 959 časových jednotek v Dotazu 4.

Další postup

V příručce *Administration Guide* naleznete podrobné informace o dalších krocích, které lze provést ke zlepšení výkonu. Můžete se pak vrátit do modulu Vizualní vysvětlení a zhodnotit dopad provedených akcí.

Lekce 4. Zlepšení přístupového plánu v databázovém prostředí děleném na oblasti

V této lekci se dozvíte, jak se přístupový plán a související okna pro základní dotaz mění při provádění různých akcí vyladění. Prostřednictvím řady příkladů doprovázených ilustracemi se dozvíte, jak lze zlepšit odhadované celkové náklady pro přístupový plán (i v případě jednoduchého dotazu) pomocí příkazu **runstats** a přidáním příslušných indexů.

Až získáte s modulem Vizualní vysvětlení zkušenosti, objevíte další způsoby vyladění dotazů.

Práce s grafy přístupových plánů

Čtyři ukázkové snímky Explain nám poslouží jako příklady pro vysvětlení, proč je pro výkon databáze důležité vyladění.

Dotazy spojené se snímky Explain jsou číslovány od 1 do 4. Každý dotaz používá stejný příkaz SQL (popsaný v Lekci 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
```

Ale každá iterace dotazu používá více metod vyladění než jeho předchozí provedení. Například u Dotazu 1 nebylo provedeno žádné vyladění výkonu, zatímco u Dotazu 4 jich bylo provedeno nejvíce. Rozdíly v dotazech jsou popsány níže:

Dotaz 1

Spuštění dotazu bez indexů a statistiky

Dotaz 2

Shromáždění aktuální statistiky pro tabulky a indexy v dotazu

Dotaz 3

Vytvoření indexů ve sloupcích použitých ke spojení tabulek v dotazu

Dotaz 4

Vytvoření dalších indexů pro sloupec tabulky

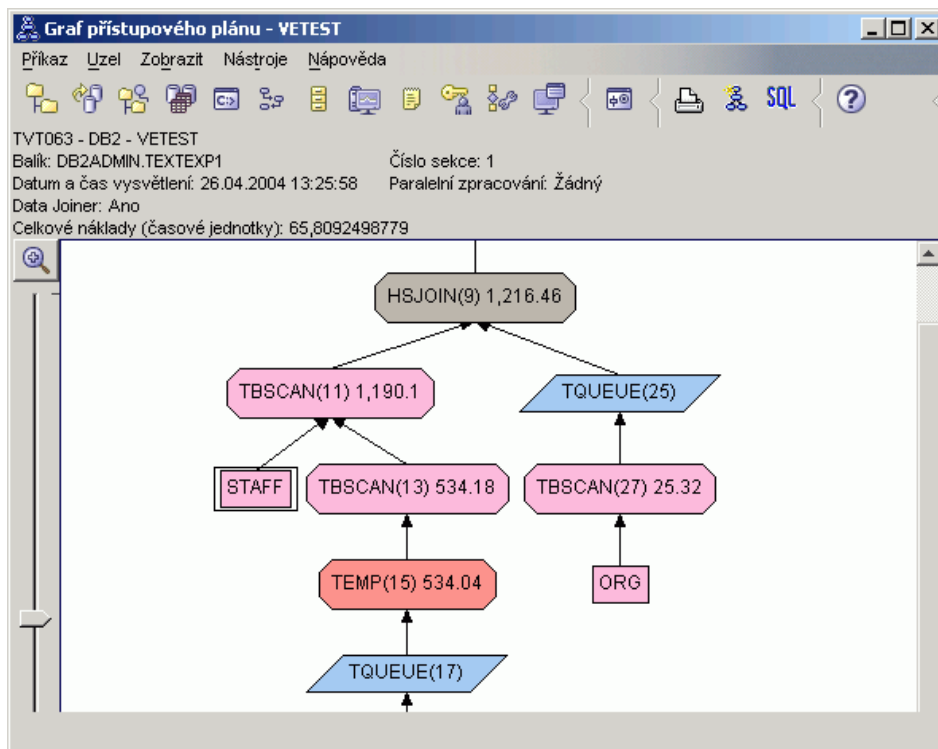
Tyto příklady byly vytvořeny na počítači RS/6000 SP se 7 fyzickými uzly využívajícími paralelní zpracování mezi oblastmi.

Spuštění dotazu bez indexů a statistiky

V tomto příkladu byl přístupový plán vytvořen pro dotaz SQL bez indexů a statistiky.

Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 1), postupujte takto:

1. V modulu Řídicí centrum rozbalte strom objektů, dokud nenaleznete databázi SAMPLE.
2. Klepněte pravým tlačítkem myši na databázi a vyberte příkaz **Zobrazit historii vysvětlených příkazů** z rozevírací nabídky. Otevře se okno Historie vysvětlených příkazů.
3. Klepněte dvakrát na položku označenou jako Číslo dotazu 1 (je možné, že bude nutné přejít doprava, aby se zobrazil sloupec **Číslo dotazu**). Pro příkaz se otevře okno Graf přístupového plánu.



Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

1. Existuje aktuální statistika pro každou tabulku v dotazu?

Chcete-li ověřit, zda existuje aktuální statistika pro každou tabulku v dotazu, dvakrát klepněte na každý uzel tabulky v grafu přístupového plánu.

V odpovídajícím okně Statistika tabulek, které se otevře, obsahuje řádek **STATS_TIME** ve sloupci **Vysvětleno** slova "Statistika nebyla aktualizována", což znamená, že nebyla v okamžiku vytvoření snímku shromážděna žádná statistika.

Pokud aktuální statistika neexistuje, použijte optimalizátor výchozí statistiku, která se může lišit od skutečné statistiky. Výchozí statistika je označena slovem "výchozí" ve sloupci **Vysvětleno** v okně Statistika tabulek.

Podle informací v okně Statistika tabulek v tabulce ORG použil optimalizátor výchozí statistiku (jak je označeno vedle vysvětlených hodnot). Výchozí statistika byla použita, protože skutečná statistika nebyla při vytvoření snímku k dispozici (jak je označeno v řádku **STATS_TIME**).

Statistika	Vysvětleno	Aktuální
CREATE_TIME	28.04.2004 11:02:33	28.04.2004 11:02:33
STATS_TIME	Statistika nebyla aktualizována	Statistika nebyla aktualizována
CARD	60(výchozí)	-1
NPAGES	1(výchozí)	-1
FPAGES	1(výchozí)	-1
COLCOUNT	5(výchozí)	5
OVERFLOW	0(výchozí)	-1
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
NESTALÉ	Ne(výchozí)	Ne

2. Používá tento přístupový plán nejefektivnější metody přístupu k datům?

Tento přístupový plán obsahuje procházení tabulky, nikoli procházení indexu. Procházení tabulky jsou zobrazena jako osmiúhelníky a označena hodnotou TBSCAN. Pokud by byla použita procházení indexu, zobrazila by se jako kosočtverce a byla by označena hodnotou IXSCAN. Použití indexu vytvořeného pro tabulku je nákladově efektivnější než procházení tabulky, pokud se extrahují malé objemy dat.

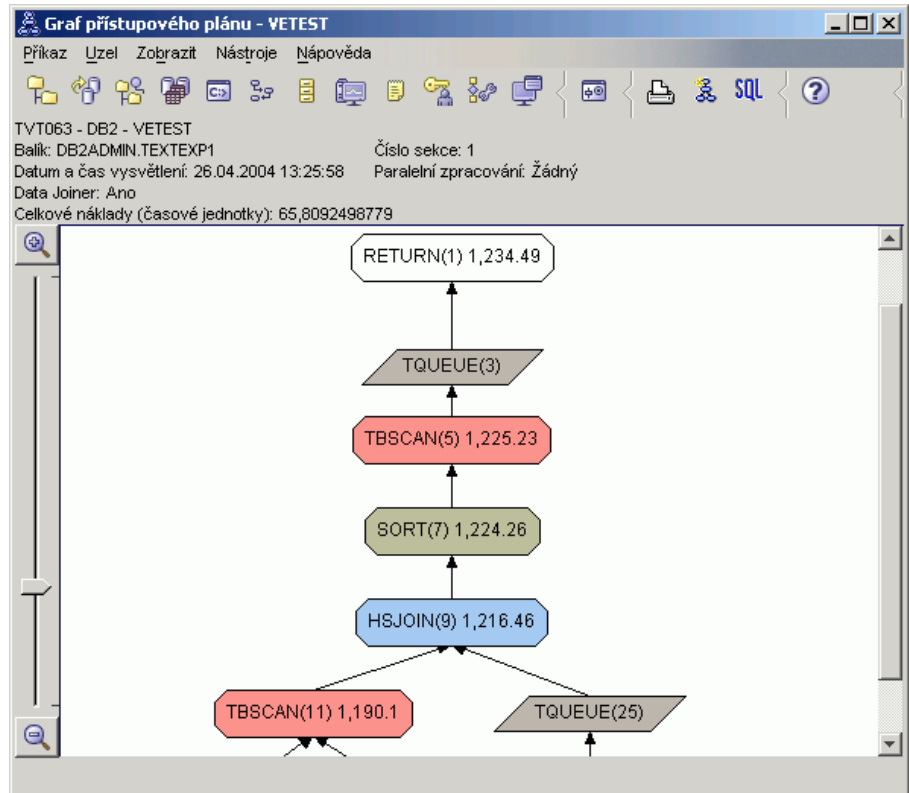
3. Jak efektivní je tento přístupový plán?

Efektivitu přístupového plánu můžete určit, pouze pokud je založena na skutečné statistice. Protože optimalizátor použil v přístupovém plánu výchozí statistiku, nelze zjistit, jak je plán efektivní.

Obecně řečeno, je vhodné si poznačit celkové odhadované náklady na přístupový plán pro pozdější porovnání s revidovanými přístupovými plány. Náklady uvedené v jednotlivých uzlech jsou kumulativní, od prvních kroků dotazu až po daný uzel a včetně něho.

Poznámka: U dělených databází se jedná o kumulativní náklady pro uzel, který užívá nejvíce prostředků.

V okně Graf přístupového plánu jsou celkové náklady přibližně 1234 časových jednotek, což je zobrazeno v uzlu **RETURN (1)** v horní části grafu. Celkové odhadované náklady se rovněž zobrazují v horní oblasti okna.



4. Další postup

Dotaz 2 zkoumá přístupový plán pro základní dotaz po spuštění příkazu **runstats**. Použití příkazu **runstats** poskytne optimalizátoru aktuální statistiku ke všem tabulkám, ke kterým dotaz přistupuje.

Shromáždění aktuální statistiky pro tabulky a indexy pomocí příkazu **runstats**

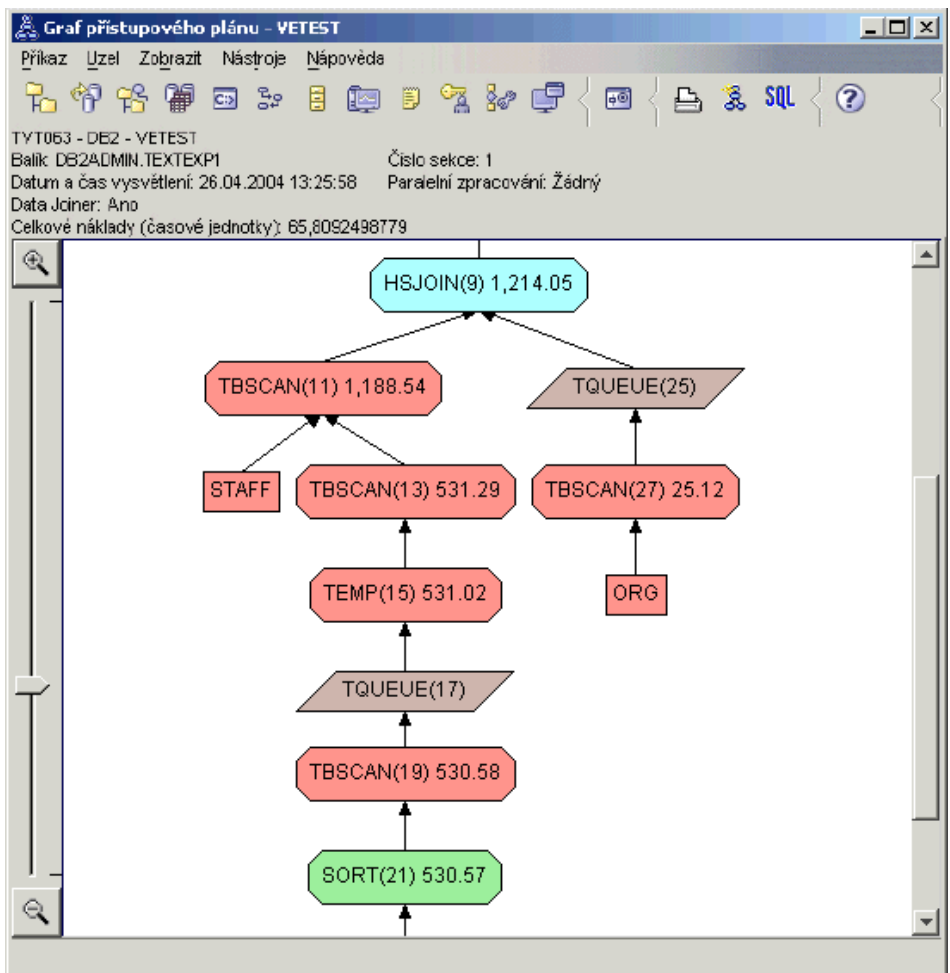
Tento příklad navazuje na přístupový plán popsany v Dotazu 1. Nyní pro něho bude shromážděna aktuální statistika pomocí příkazu **runstats**.

Důrazně se doporučuje pomocí příkazu **runstats** shromáždít aktuální statistiku tabulek a indexů, zejména pokud od posledního spuštění příkazu **runstats** došlo ke značnému množství aktualizací nebo k vytvoření nových indexů. To optimalizátoru zajišťuje nej přesnější informace, pomocí kterých pak lze určit nejvhodnější přístupový plán.

Pokud aktuální statistika není k dispozici, může optimalizátor na základě nepřesné výchozí statistiky zvolit neefektivní přístupový plán.

Příkaz **runstats** je třeba spustit *po* aktualizaci tabulek; v opačném případě může optimalizátor považovat tabulku za prázdnou. Tento problém je evidentní, pokud je kardinalita v okně Podrobnosti o operátoru rovna nule. V takovém případě dokončete aktualizace tabulky, spusťte znovu příkaz **runstats** a znovu vytvořte pro příslušné tabulky snímky Explain.

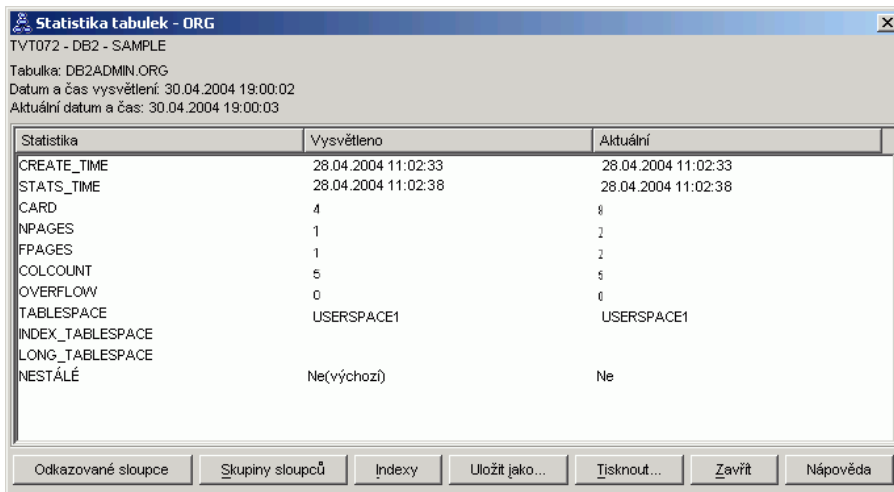
Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 2), postupujte takto: V okně Historie vysvětlených příkazů dvakrát klepněte na položku označenou jako Číslo dotazu 2. Otevře se okno Graf přístupového plánu pro toto provedení příkazu.



Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

1. Existuje aktuální statistika pro každou tabulku v dotazu?

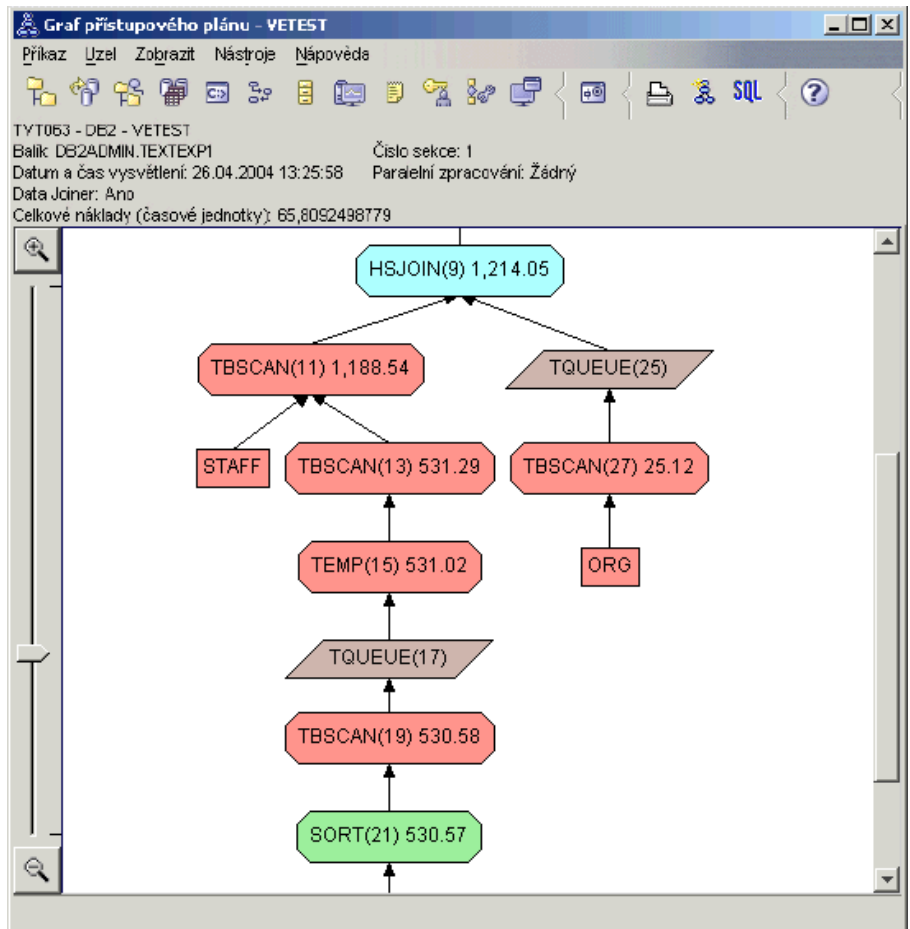
Okno Statistika tabulek pro tabulku ORG ukazuje, že optimalizátor použil skutečnou statistiku (hodnota **STATS_TIME** je skutečný čas shromáždění statistiky). Přesnost statistiky závisí na tom, zda došlo k zásadním změnám obsahu tabulek od spuštění příkazu **runstats**.



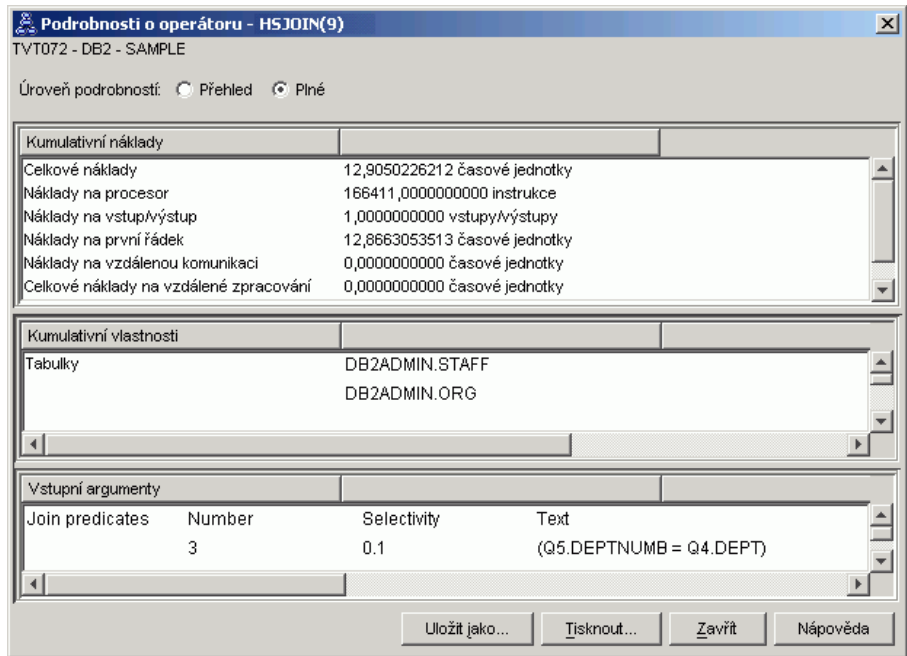
Statistika	Vysvětleno	Aktuální
CREATE_TIME	28.04.2004 11:02:33	28.04.2004 11:02:33
STATS_TIME	28.04.2004 11:02:38	28.04.2004 11:02:38
CARD	4	8
NPAGES	1	2
FPAGES	1	2
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
NESTÁLÉ	Ne(výchozí)	Ne

2. Používá tento přístupový plán nejefektivnější metody přístupu k datům?

Jako v Dotazu 1 používá přístupový plán v Dotazu 2 procházení tabulky (TBSCAN), nikoli procházení indexu (IXSCAN). Ačkoli existuje aktuální statistika, nebylo procházení indexu provedeno, protože ve sloupcích použitých dotazem nejsou žádné indexy. Jedním ze způsobů zlepšení dotazu by bylo zadání indexů pro sloupce použité ke spojení tabulek (tzn. pro sloupce použité v predikátech spojení) do optimalizátoru. V tomto příkladu se jedná o první spojení pro sloučené procházení: HSJOIN (9).



V okně Podrobnosti o operátoru pro operátor HSJOIN (9) si všimněte sekce **Predikáty spojení** v části **Vstupní argumenty**. Sloupce použité v této operaci spojení jsou uvedeny ve sloupci **Text**. V tomto příkladu jsou těmito sloupci DEPTNUMB a DEPT.



3. Jak efektivní je tento přístupový plán?

Přístupové plány založené na aktuální statistice vždy vytvářejí realistický odhad nákladů (měřený v časových jednotkách). Protože odhadované náklady v Dotazu 1 byly založeny na výchozí statistice, nemůžeme porovnat náklady těchto dvou grafů přístupového plánu, abychom určili, který je efektivnější. To, zda jsou náklady nižší nebo vyšší, není relevantní. Abychom mohli efektivitu změřit platným způsobem, je třeba porovnat náklady na přístupové plány založené na skutečné statistice.

4. Další postup

Dotaz 3 zkoumá dopady přidání indexů pro sloupce DEPTNUMB a DEPT. Přidání indexů pro sloupce použité v predikátech spojení může zlepšit výkon.

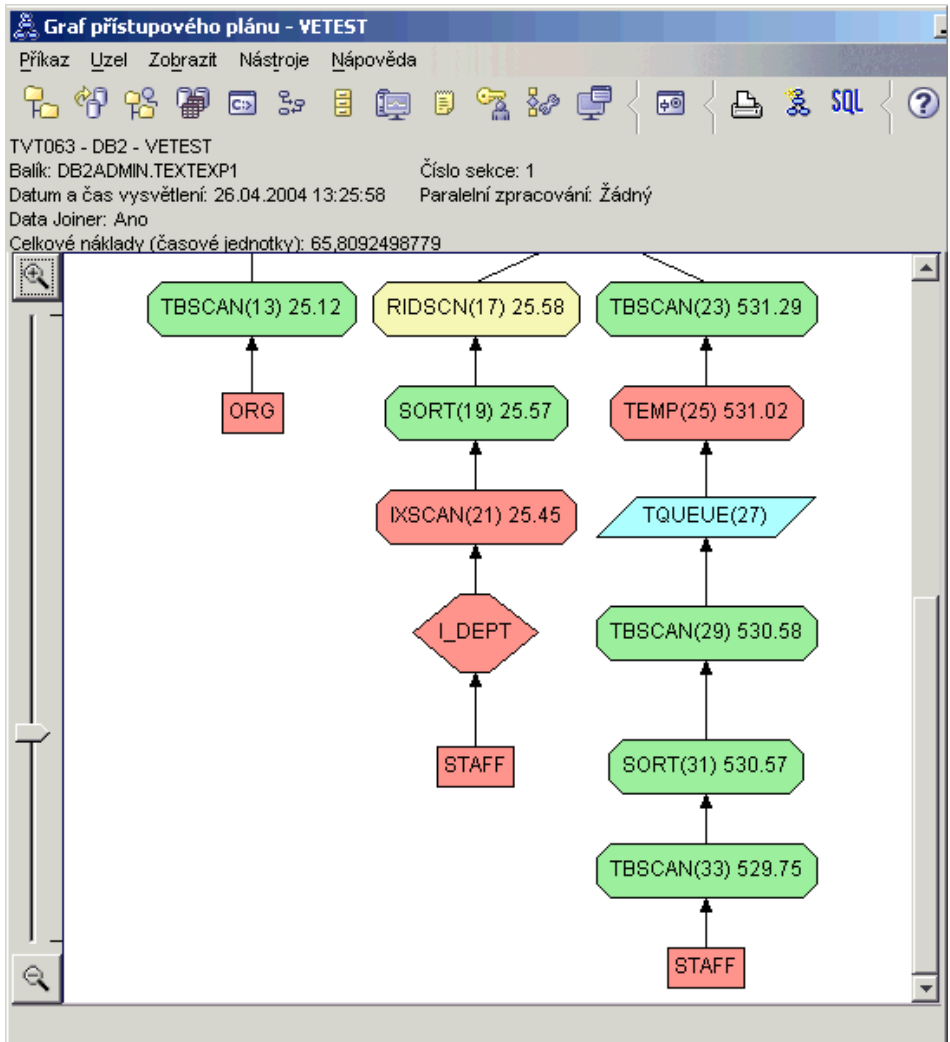
Vytvoření indexů pro sloupce použité ke spojení tabulek v dotazu

Tento příklad navazuje na přístupový plán popsany v Dotazu 2. Nyní budou vytvořeny indexy pro sloupec DEPT v tabulce STAFF a pro sloupec DEPTNUMB v tabulce ORG.

Poznámka: Ve verzi 8 lze doporučené indexy vytvořit pomocí průvodce Výkon zátěže.

Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 3), postupujte takto: V okně Historie vysvětlených příkazů dvakrát klepněte na položku označenou jako Číslo dotazu 3. Otevře se okno Graf přístupového plánu pro toto provedení příkazu.

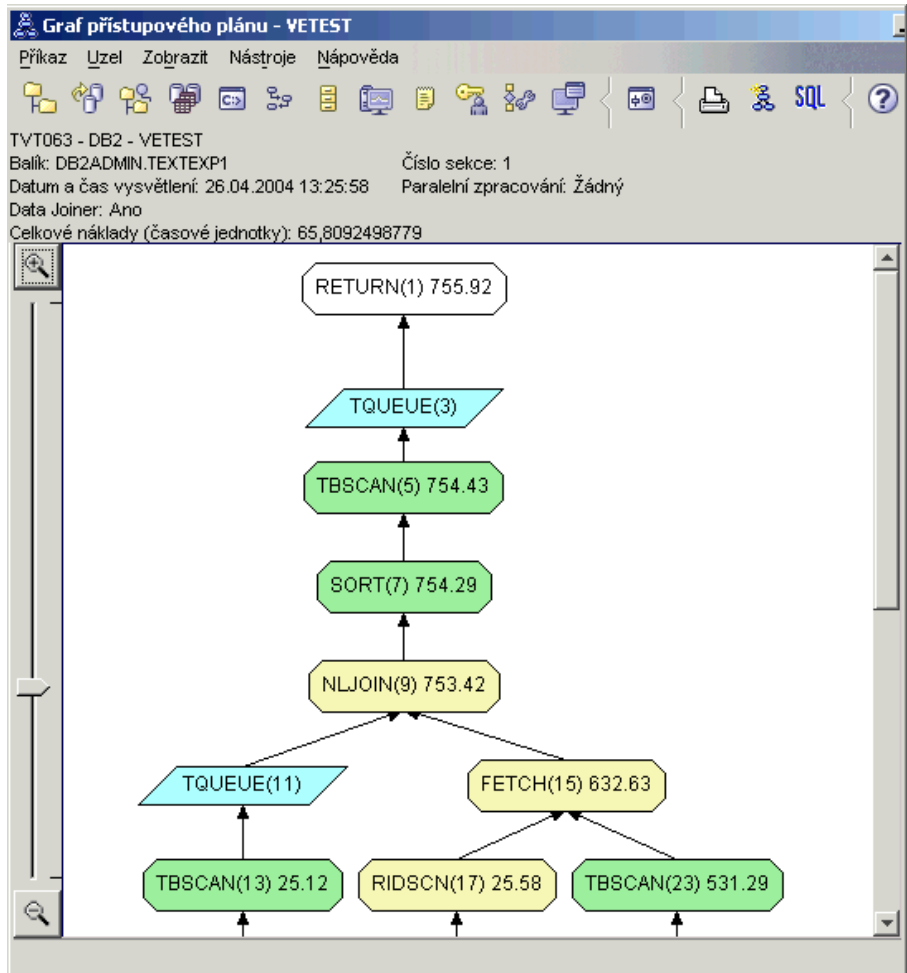
Poznámka: Ačkoli byl vytvořen index pro sloupec DEPTNUM, optimalizátor jej nepoužil.



Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

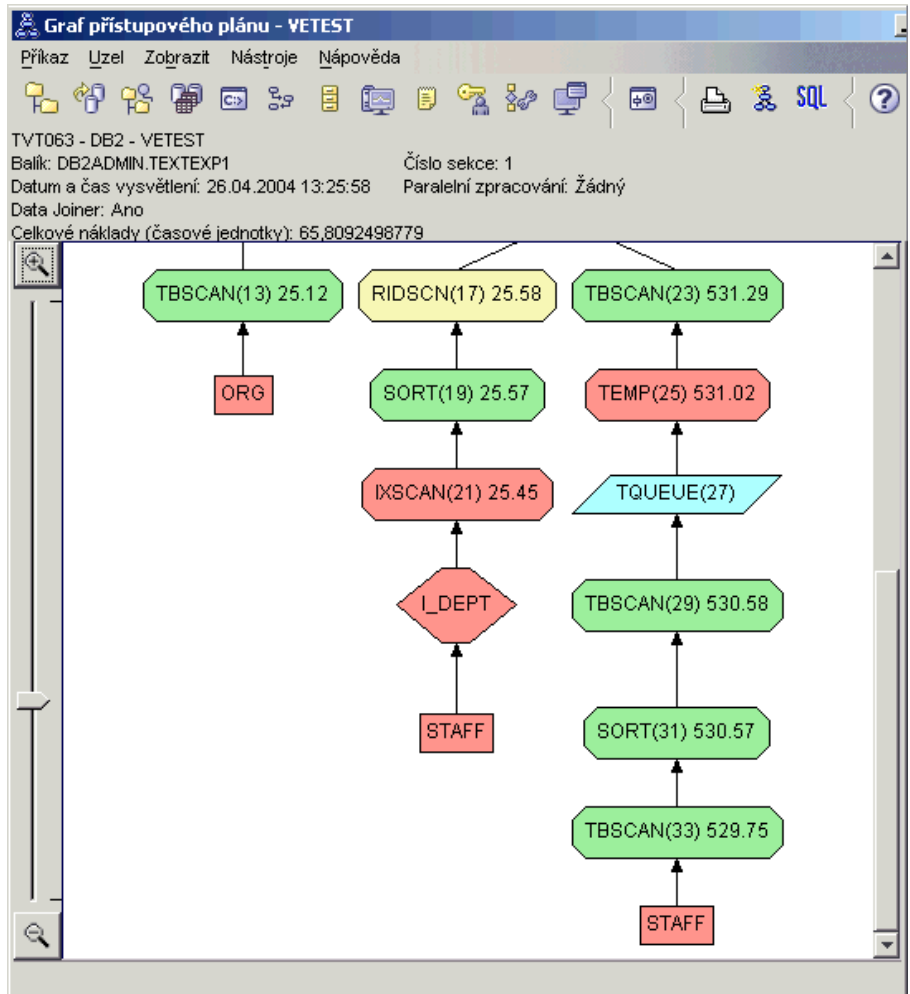
1. Co se v přístupovém plánu změnilo díky indexům?

Přímo nad tabulku STAFF byl přidán nový uzel ve tvaru kosočtverce, **L_DEPT**. Tento uzel představuje index vytvořený pro sloupec DEPT a ukazuje, že optimalizátor použil k určení řádků, které se mají načíst, procházení indexu namísto procházení tabulky.

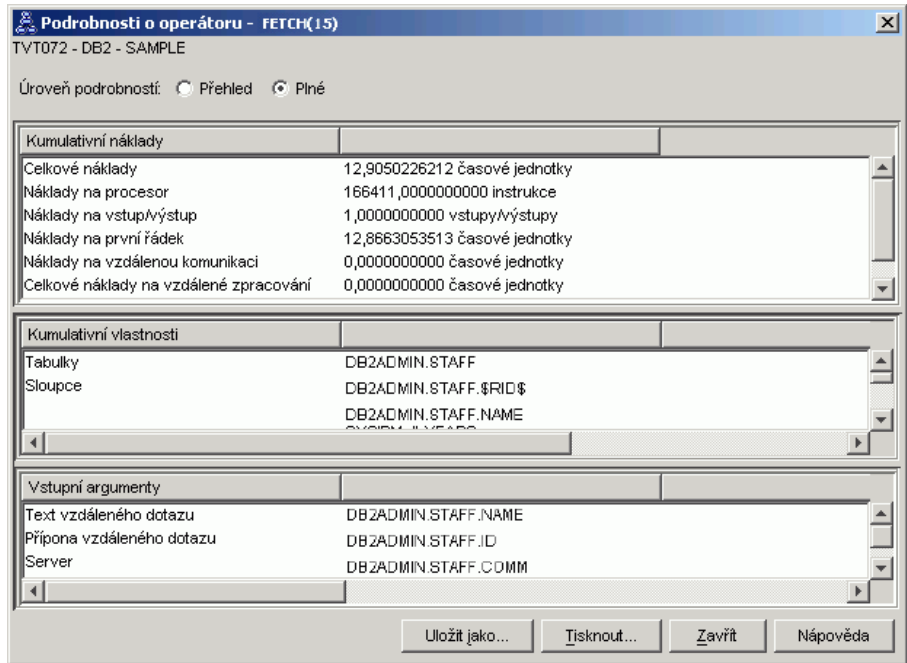


2. Používá tento přístupový plán neefektivnější metody přístupu k datům?

Přístupový plán pro tento dotaz ukazuje dopad vytvoření indexů pro sloupec DEPTNUMB tabulky ORG, jehož výsledkem je uzel FETCH (15) a IXSCAN (21), a pro sloupec DEPT tabulky STAFF. Dotaz 2 neobsahoval tento index; proto tam bylo použito procházení tabulky.



Okno Podrobnosti o operátoru pro operátor FETCH (15) zobrazuje sloupce používané v této operaci.



Kombinace indexu a načtení je vypočtena jako méně nákladná než úplné procházení tabulky použité v předchozích přístupových plánech.

3. Jak efektivní je tento přístupový plán?

Tento přístupový plán je nákladově efektivnější než přístupový plán z předchozího příkladu. Kumulativní náklady byly sníženy z přibližně 1214 časových jednotek v Dotazu 2 na přibližně 755 časových jednotek v Dotazu 3.

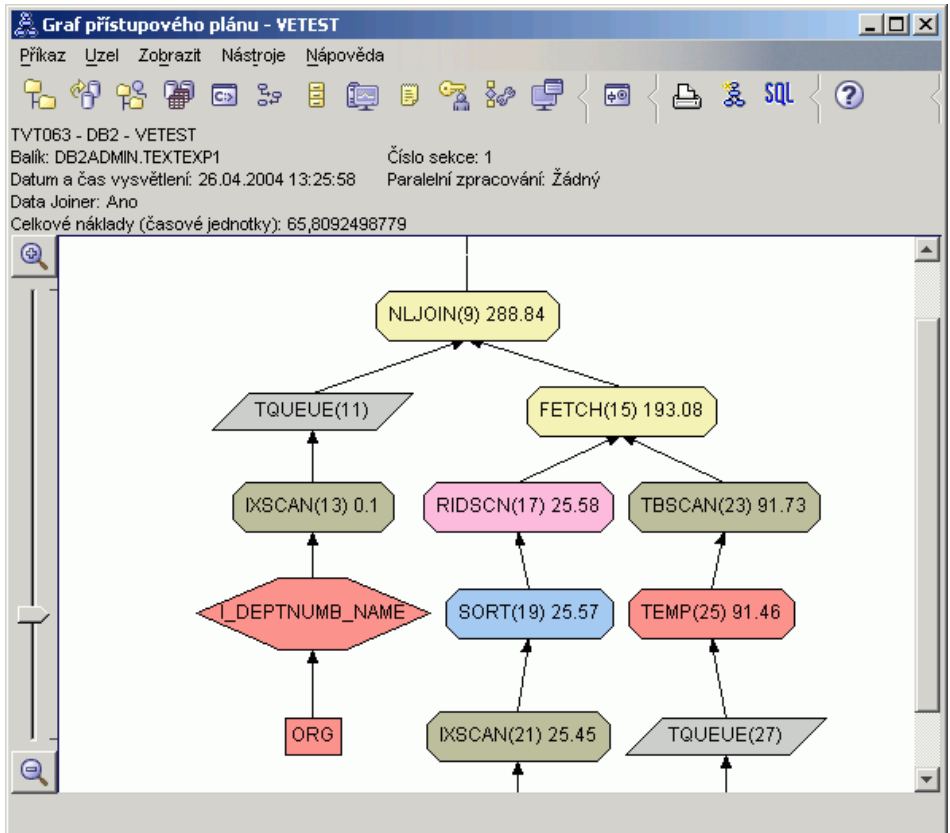
4. Další postup

Dotaz 4 redukuje načtení a procházení indexu do jednoho procházení indexu bez načtení. Vytvoření dalších indexů může snížit odhadované náklady přístupového plánu.

Vytvoření dalších indexů pro sloupce tabulky

Tento příklad navazuje na přístupový plán popsany v Dotazu 3. Nyní je k němu vytvořen index pro sloupec JOB v tabulce STAFF a přidána položka DEPTNAME do existujícího indexu v tabulce ORG. (Přidání samostatného indexu by mohlo způsobit další přístup.)

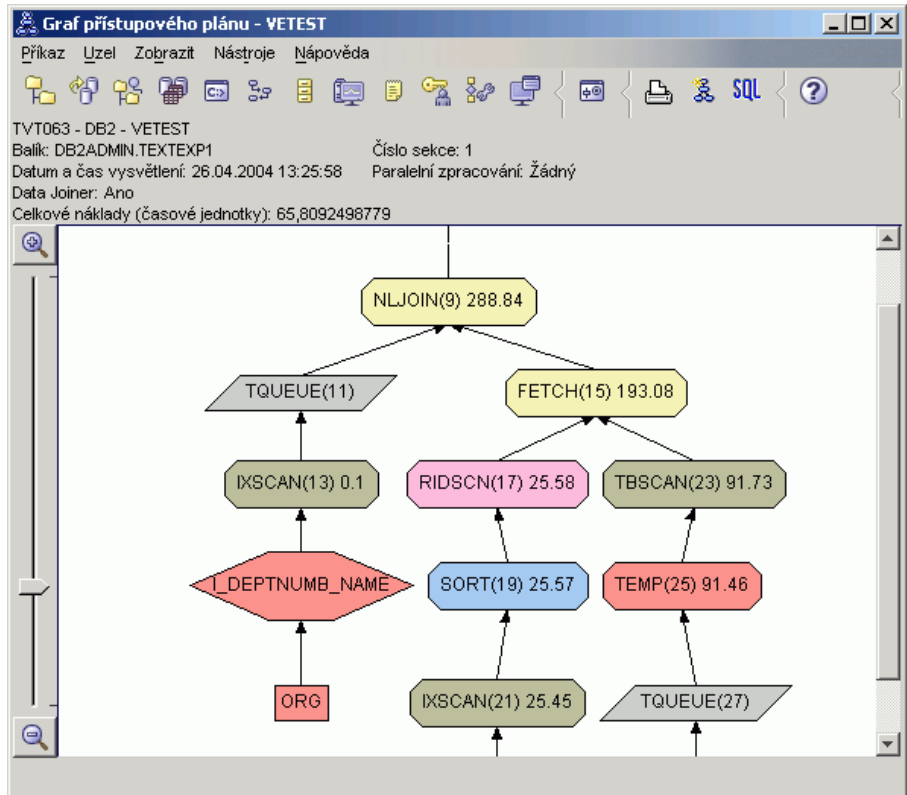
Chcete-li zobrazit graf přístupového plánu pro tento dotaz (Dotaz 4), postupujte takto: V okně Historie vysvětlených příkazů dvakrát klepněte na položku označenou jako Číslo dotazu 4. Otevře se okno Graf přístupového plánu pro toto provedení příkazu.



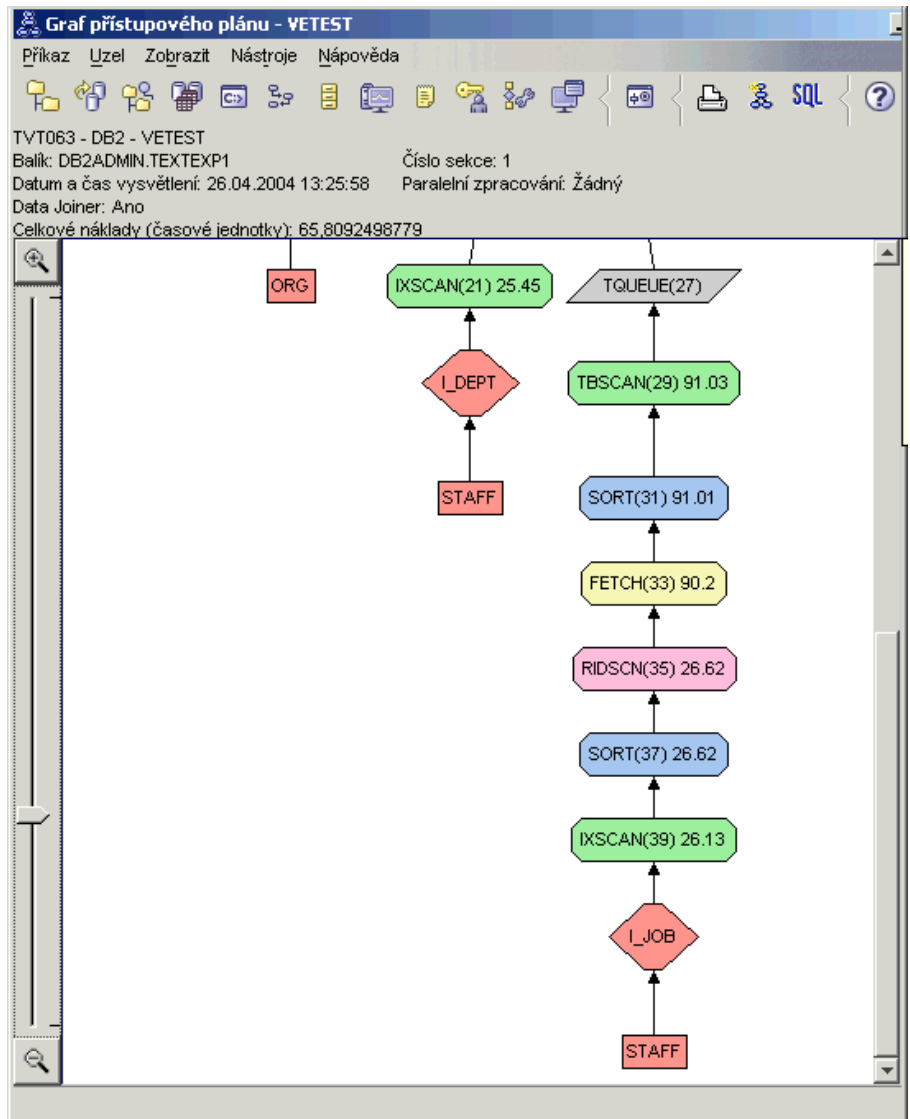
Odpovědi na následující otázky vám pomohou porozumět tomu, jak lze dotaz zlepšit.

1. Co se v tomto přístupovém plánu změnilo jako výsledek vytvoření dalších indexů?

Ve střední části grafu přístupového plánu si všimněte, že u tabulky ORG bylo předchozí procházení tabulky změněno na procházení indexu, IXSCAN (7). Přidání sloupce DEPTNAME do indexu v tabulce ORG umožnilo optimalizátoru zúžit přístup zahrnující procházení tabulky.



V dolní části grafu přístupového plánu si všimněte, že u tabulky STAFF byla předchozí procházení indexu a načtení změněna na pouhé procházení indexu IXSCAN (39). Vytvoření indexu JOB v tabulce STAFF umožnilo optimalizátoru vyloučit nadbytečný přístup zahrnující načtení.



2. Jak efektivní je tento přístupový plán?

Tento přístupový plán je nákladově efektivnější než přístupový plán z předchozího příkladu. Kumulační náklady byly sníženy z přibližně 753 časových jednotek v Dotazu 3 na přibližně 288 časových jednotek v Dotazu 4.

Další postup

V příručce *Administration Guide* naleznete podrobné informace o dalších krocích, které lze provést ke zlepšení výkonu. Můžete se pak vrátit do modulu Vizuelní vysvětlení a zhodnotit dopad provedených akcí.

Dodatek A. Koncepce modulu Vizuelní vysvětlení

Přístupový plán

Před převodem vysvětlitelného příkazu SQL je třeba načíst určité údaje. Pořadí operací pro přístup k těmto datům určuje *přístupový plán*. Tento plán umožňuje zobrazit statistiku pro vybrané tabulky, indexy a sloupce, dále vlastnosti pro operátory či globální informace, jako jsou statistiky funkcí a tabulkových prostorů, a konečně také konfigurační parametry relevantní pro optimalizaci. V modulu Vizuelní vysvětlení můžete v grafické podobě zobrazit přístupové plány pro příkazy SQL.

Optimalizátor vytvoří přístupový plán při každé kompilaci vysvětlitelného příkazu SQL. U statických příkazů je tato operace prováděna během předběžné kompilace/vazby (příkazy prep/bind) a u dynamických příkazů během provádění.

Je důležité pochopit, že přístupový plán je *odhadem* založeným na dostupných informacích. Optimalizátor zakládá své odhady například na následujících informacích:

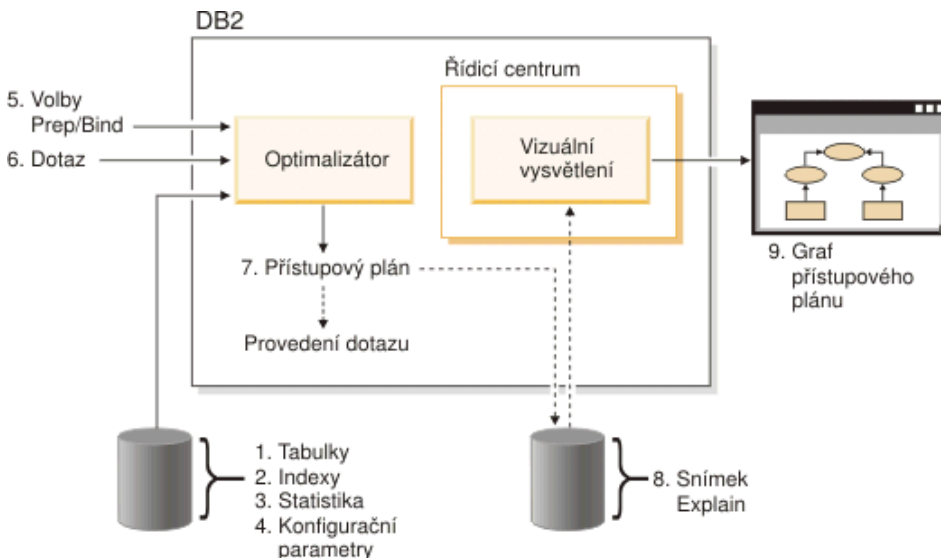
- Statistiky v tabulkách systémového katalogu (nejsou-li statistiky aktuální, aktualizujte je pomocí příkazu runstats.)
- Konfigurační parametry
- Volby pro vazbu
- Třída optimalizace dotazu

Údaje o nákladech spojených s přístupovým plánem představují *nejlepší odhad* optimalizátoru týkající se využití prostředků pro určitý dotaz. Skutečně uplynulá doba se pro jednotlivé dotazy může lišit v závislosti na vnějších faktorech mimo rámec produktu DB2 (například v závislosti na počtu dalších souběžně spuštěných aplikacích). Skutečně uplynulou dobu lze změřit při provádění dotazu pomocí služby monitorování výkonu.

Graf přístupového plánu

Modul Vizuelní vysvětlení používá při vytváření grafu přístupového plánu informace z většího počtu zdrojů, jak je znázorněno dále na ilustraci. Na základě různých vstupních údajů vybere optimalizátor přístupový plán a modul Vizuelní vysvětlení jej zobrazí v *grafu přístupového plánu*. Uzly v grafu reprezentují tabulky a indexy se

všemi operacemi. Propojení mezi uzly reprezentují tok dat.



Prvkům na uvedené ilustraci odpovídá následující seznam úloh. (Přerušovanou čarou jsou označeny povinné kroky pro modul Vizuální vysvětlení.)

1. Vyladění návrhu tabulky a reorganizace dat v tabulce.
2. Vytvoření vhodných indexů.
3. Vytvoření aktuální statistiky pro optimalizátor pomocí příkazu `runstats`.
4. Výběr vhodných konfiguračních parametrů.
5. Výběr vhodných voleb vazby.
6. Návrh dotazů pro načtení pouze požadovaných dat .
7. Vytvoření přístupového plánu.
8. Vytvoření snímků Explain.
9. Zobrazení a využití grafu přístupového plánu.

Příklad: Chcete-li použít modul Vizuální vysvětlení, aktualizujte nejprve stávající statistiku pomocí příkazu `runstats` pro tabulky a indexy používané příkazem. Tyto statistické údaje, konfigurační parametry, volby vazby a samotný dotaz jsou optimalizátorem použity při vytvoření přístupového plánu a snímku Explain při vázání balíku. Modul Vizuální vysvětlení na základě výsledného snímku Explain zobrazí graf přístupového plánu pro příkaz.

Uzel grafu přístupového plánu

Graf přístupového plánu je tvořen stromem se zobrazenými *uzly*. Tyto uzly reprezentují následující objekty:

- Tabulky, zobrazené jako obdélníky
- Indexy, zobrazené jako kosočtverce
- Operátory, zobrazené jako osmiúhelníky Operátory TQUEUE, zobrazené jako rovnoběžníky

- Tabulkové funkce, zobrazené jako šestiúhelníky

Dělení do klastrů

Operace aktualizací mohou postupem času vést k tomu, že se změní umístění řádků na datových stránkách, a tím se sníží stupeň *dělení do klastrů* existujícího mezi indexem a datovými stránkami. Reorganizace tabulky s ohledem na vybraný index umožní nové rozdělení dat do klastrů. Sdružený index je nejvíce užitečný pro sloupce, které mají predikáty rozsahu, protože umožňuje lepší sekvenční přístup k datům v základní tabulce. To vede k nižšímu počtu načtených stránek, protože podobné hodnoty se nacházejí na stejné stránce.

Obecně řečeno může mít pouze jeden z indexů v tabulce vysoký stupeň dělení do klastrů.

Chcete-li zkontrolovat stupeň dělení do klastrů pro index, klepněte dvakrát na příslušný uzel. Otevře se okno Statistika indexů. V tomto okně jsou uvedeny hodnoty poměru klastrů a faktoru klastrů. Je-li tato hodnota nízká, zvažte reorganizaci dat v tabulce.

Další informace najdete v sekci týkající se reorganizace dat tabulky v příručce *Administration Guide*.

Kontejner

Kontejner je fyzické úložiště pro umístění dat. Je asociován s tabulkovým prostorem a může se jednat o soubor, o adresář nebo o zařízení.

Náklady

Termínem *náklady* je v kontextu modulu Vizualní vysvětlení označováno celkové odhadované využití prostředků potřebné k provedení přístupového plánu pro určitý příkaz (nebo pro prvky příkazu). Faktor nákladů je odvozen z kombinace nákladů na činnost procesoru (v počtu instrukcí) a nákladů na akce I/O (v počtu operací hledání a přenosů stránek).

Jednotkou nákladů je časová jednotka *timeron*. Jednotka timeron není přímo ekvivalentní žádnému skutečně uplynulému časovému úseku, avšak umožňuje vytvořit hrubý relativní odhad vytížení prostředků (nákladů) požadovaných správcem databází k provedení dvou různých plánů pro stejný dotaz.

Náklady zobrazené pro každý uzel operátoru v grafu přístupového plánu představují kumulativní náklady zahrnující úsek od zahájení provádění přístupového plánu až po zpracování příslušného operátoru. Tato hodnota nezahrnuje takové faktory, jako je zátěž systému nebo náklady na navrácení datových řádků uživateli.

Blokování kurzoru

Blokování kurzoru je technika pro omezení režie, jejímž principem je načtení *bloku* řádků správcem databázi v jediné operaci. Tyto řádky jsou během zpracování uloženy do mezipaměti. Mezipaměť je alokována v okamžiku, kdy aplikace vydá požadavek OPEN CURSOR a je dealokována při uzavření kurzoru. Jakmile jsou zpracovány všechny řádky, je načten další blok řádků.

Typ blokování kurzoru lze určit pomocí volby BLOCKING příkazu **PREP** nebo **BIND** spolu s následujícími parametry:

UNAMBIG

Blokovány jsou pouze jednoznačné kurzory (výchozí nastavení).

ALL Blokovány jsou jednoznačné i nejednoznačné kurzory.

NO Kurzory nejsou blokovány.

Další informace najdete v sekci týkající se blokování kurzorů v příručce *Administration Guide*.

Tabulkový prostor prostoru spravovaného databázi (DMS)

V databázi mohou existovat dva typy tabulkových prostorů: DMS (prostor spravovaný databázi) a SMS (prostor spravovaný systémem).

Tabulkové prostory DMS jsou spravovány správcem databáze a jsou navrženy a vyladěny s ohledem na jeho požadavky.

Definice tabulkového prostoru DMS obsahuje seznam souborů (nebo zařízení), do kterých jsou ukládána data databáze v příslušném formátu tabulkového prostoru DMS.

Chcete-li zvýšit prostorovou kapacitu existujícího tabulkového prostoru DMS, můžete do něj přidat předem alokované soubory (nebo zařízení). Správce databázi automaticky provede nové vyvážení (operace rebalance) existujících dat ve všech kontejnerech náležících k danému tabulkovému prostoru.

Tabulkové prostory DMS a SMS mohou existovat souběžně v jedné databázi.

Dynamické příkazy SQL

Dynamické příkazy SQL jsou příkazy SQL připravené a provedené při běhu aplikačního programu. Pro dynamické příkazy SQL platí některé z následujících tvrzení:

- Příkaz SQL je zadáván interaktivně prostřednictvím rozhraní CLI nebo příkazového procesoru.
- Zdrojová data SQL jsou obsažena v proměnných jazyka hostitele vložených v aplikačním programu.

Při spuštění dynamického příkazu SQL program DB2 vytvoří přístupový plán založený na aktuální statistice katalogu a na konfiguračních parametrech. Mezi jednotlivými spuštěními příkazů aplikačním programem může být tento přístupový plán změněn.

Alternativou k dynamickému příkazu SQL je statický příkaz SQL.

Snímek Explain

Pomocí modulu Vizuální vysvětlení lze prohlížet obsah snímku Explain.

Snímek Explain obsahuje komprimované informace shromážděné při vysvětlení příkazu SQL. Je uložen jako objekt BLOB v tabulce EXPLAIN_STATEMENT a obsahuje následující informace:

- interní reprezentaci přístupového plánu, včetně příslušných operátorů a tabulek a indexů s přístupem,
- kritéria pro rozhodování používaná optimalizátorem, včetně statistických údajů pro databázové objekty a kumulativních nákladů pro každou operaci.

Snímek Explain je povinný, pokud chcete zobrazit grafickou reprezentaci přístupového plánu příkazu SQL. Podmínky pro vytvoření snímku Explain:

1. Ve správci databází musí existovat tabulky Explain pro uložení snímků Explain. Informace týkající se vytvoření těchto tabulek najdete v oddílu Vytvoření tabulek Explain v kontextové nápovědě.
2. Při předběžné kompilaci (prep) nebo vazbě (bind) balíku obsahujícího statické příkazy SQL nastavte volbu EXPLSNAP na hodnotu ALL nebo YES. Pro každý vysvětlitelný příkaz SQL v balíku tak získáte snímek Explain. Další informace týkající se příkazů **BIND** a **PREP** najdete v příručce *Command Reference*.
3. Při svázání aplikace, která zadává dynamické příkazy SQL, nastavte volbu EXPLSNAP na hodnotu ALL. Pokud budete příkazy zadávat interaktivně, nastavte speciální registr CURRENT EXPLAIN SNAPSHOT na hodnotu YES nebo EXPLAIN. Další informace najdete v sekci týkající se aktuálních snímků Explain v příručce *SQL Reference*.

Vysvětlitelné příkazy

Vysvětlitelný příkaz je příkaz SQL, pro který lze provést operaci vysvětlení (Explain).

Vysvětlitelné příkazy SQL jsou následující:

- SELECT
- INSERT
- UPDATE
- DELETE
- VALUES

Vysvětlený příkaz

Vysvětlený příkaz je příkaz SQL, pro který byla provedena operace vysvětlení (Explain). Vysvětlené příkazy jsou zobrazeny v okně Historie vysvětlených příkazů.

Operand

Operand je entita, na které je prováděna operace. Operandem pro různé operátory (například TBSCAN a IXSCAN) může být například tabulka nebo index.

Operátor

Operátor je akce, kterou je třeba provést na datech, nebo výstup z tabulky nebo indexu při provedení přístupového plánu pro příkaz SQL.

V grafu přístupového plánu mohou být uvedeny následující operátory:

DELETE

Odstraní řádky z tabulky.

EISCAN

Prochází uživatelský index s cílem vytvořit zúžený proud řádků.

FETCH

Načte sloupce z tabulky s použitím specifického identifikátoru záznamu.

FILTER

Filtruje data s použitím jednoho nebo více predikátů.

GRPBY

Seskupí řádky podle společných hodnot určených sloupců nebo funkcí a vyhodnotí nastavené funkce.

HSJOIN

Reprezentuje hašované spojení, kdy dvě nebo více tabulek je hašováno na základě sloupců spojení.

INSERT

Vloží řádky do tabulky.

IXAND

Provede operaci AND pro identifikátory řádků (RID) ze dvou nebo více procházení indexu.

IXSCAN

Prochází index tabulky s volitelnými podmínkami spuštění/ukončení a vytvoří uspořádaný proud řádků.

MSJOIN

Reprezentuje spojení se sloučením, kde vnější a vnitřní tabulky musí být v pořadí predikátů operace JOIN.

NLJOIN

Reprezentuje vnořené cyklické spojení s jedním přístupem k vnitřní tabulce pro každý řádek vnější tabulky.

RETURN

Reprezentuje navrácení dat z dotazu uživateli.

RIDSCN

Prochází seznam identifikátorů řádků (RID) získaných z jednoho nebo více indexů.

RPD (Remote PushDown)

Operátor pro vzdálené plány. Je velice podobný operátoru SHIP ve verzi 8 (operátor RQUERY v předchozích verzích), avšak neobsahuje příkaz SQL.

SHIP Načte data ze zdroje vzdálené databáze. Pro použití ve federovaném systému.

SORT Uspořádá řádky podle pořadí specifikovaných sloupců a volitelně vyloučí duplicitní položky.

TBSCAN

Vyplní řádky načtením všech požadovaných dat přímo z datových stránek.

TEMP Uloží data do dočasné tabulky pro zpětné načtení (případně vícenásobné).

TQUEUE

Provede přenos dat tabulky mezi databázovými agenty.

UNION

Zřetězí proudy řádků z více tabulek.

UNIQUE

Pro určené sloupce vyloučí řádky s duplicitními hodnotami.

UPDATE

Aktualizuje řádky v tabulce.

CMPEXP

Jméno operátoru: CMPEXP

Reprezentuje: Výpočet výrazů požadovaných pro přechodné nebo konečné výsledky.

(Tento operátor je pouze pro režim ladění.)

DELETE

Jméno operátoru: DELETE

Reprezentuje: Odstranění řádků z tabulky.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady na přístupový plán, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být odstraněny.

Doporučení z hlediska výkonu:

- Pokud odstraňujete všechny řádky z tabulky, zvažte použití příkazu DROP TABLE nebo **LOAD REPLACE**.

EISCAN

Jméno operátoru: EISCAN

Reprezentuje: Tento operátor prochází uživatelský index, aby zredukoval proud řádků. Pro procházení se používá několik podmínek počátku/konce z funkce vytváření rozsahu zadané uživatelem.

Tato operace se provádí za účelem zmenšení množiny kvalifikovaných řádků před přístupem k základní tabulce (na základě predikátů).

Doporučení z hlediska výkonu:

- Při aktualizacích databáze může průběžně docházet k fragmentaci indexu, což má za následek více stránek indexu, než je nezbytně nutné. To lze napravit odstraněním a opětovným vytvořením indexu nebo jeho reorganizací.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats.

FETCH

Jméno operátoru: FETCH

Reprezentuje: Načtení sloupců z tabulky pomocí identifikátoru specifického řádku (RID).

Doporučení z hlediska výkonu:

- Rozšířením klíčů indexu zahrňte načítané sloupce, aby nebylo třeba přistupovat k datovým stránkám.
- Vyhledejte index vztahující se na načtená data a dvojným klepnutím na jeho uzel zobrazte okno statistiky. Zkontrolujte, že je stupeň dělení do klastrů pro index vysoký.
- Pokud je počet vstupně-výstupních operací (I/O) při načítání dat vyšší než počet stránek v tabulce, zvýšte velikost vyrovnávací paměti.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats..

Statistika kvantilů a častých hodnot poskytuje informace o selektivitě predikátů, která určuje, kdy se dává přednost procházení indexu před procházením tabulky. K aktualizaci této statistiky použijte u tabulky s klauzulí WITH DISTRIBUTION příkaz **runstats**.

FILTER

Jméno operátoru: FILTER

Reprezentuje: Aplikace zbytkových predikátů k tomu, aby byla data filtrována na základě kritérií zadaných predikátů.

Doporučení z hlediska výkonu:

- Zkontrolujte, že jste použili predikáty, které načítají pouze potřebná data. Přesvědčte se například, zda hodnota selektivity pro predikáty představuje tu část tabulky, kterou chcete vrátit.
- Přesvědčte se, že je hodnota třídy optimalizace alespoň 3, aby optimalizátor místo poddotazů používal spojení (JOIN). Pokud to není možné, pokuste se přepsat dotaz SQL ručně tak, aby byl poddotaz eliminován. Příklad je uveden v příručce *Administration Guide* v části týkající se přepisování dotazů kompilátorem SQL.

GENROW

Jméno operátoru: GENROW

Reprezentuje: Vestavěná funkce, která generuje tabulku řádků bez vstupu z tabulek, indexů či operátorů.

Operátor GENROW lze používat optimalizátorem ke generování řádků dat (například pro příkaz INSERT nebo pro některé seznamy IN transformované do spojení).

Chcete-li zobrazit odhadovanou statistiku pro tabulky generované funkcí GENROW, klepněte dvakrát na příslušný uzel.

GRPBY

Jméno operátoru: GRPBY

Reprezentuje: Seskupení řádků podle společných hodnot určených sloupců nebo funkcí. Tato operace je nutná k vytvoření skupiny hodnot nebo k vyhodnocení množinových funkcí.

Operátor GRPBY lze použít i v případě, že nejsou zadány žádné sloupce GROUP BY, a to pokud seznam SELECT obsahuje agregační funkce, což označuje, že se celá tabulka při této agregaci považuje za jedinou skupinu.

Doporučení z hlediska výkonu:

- Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být seskupeny.

- Pokud chcete zvýšit výkon příkazu SELECT, který obsahuje jednu agregační funkci, ale žádnou klauzuli GROUP BY, zkuste provést následující postup:
 - V případě agregační funkce MIN(C) vytvořte vzestupný index pro sloupec C.
 - V případě agregační funkce MAX(C) vytvořte sestupný index pro sloupec C.

HSJOIN

Jméno operátoru: HSJOIN

Reprezentuje: Hašované spojení, pro které jsou příslušné řádky z tabulek hašovány za účelem přímého spojení bez předběžného uspořádání obsahu tabulek.

Spojení je nutné v případě, že klauzule FROM obsahuje odkaz na více tabulek. Hašované spojení je možné používat vždy v případě existence predikátu spojení srovnávajícího sloupce ze dvou různých tabulek. Predikáty spojení musí mít přesně stejný datový typ. Hašovaná spojení mohou být také výsledkem přeepsaného poddotazu, jako v případě operátoru NLJOIN.

Hašované spojení nevyžaduje, aby byly vstupní tabulky seřazeny. Spojení probíhá procházením vnitřní tabulky hašovaného spojení a generováním vyhledávací tabulky hašováním hodnot spojovaného sloupce. Následně je čtena vnější tabulka, je prováděno hašování hodnot spojovaného sloupce a dohledání ve vyhledávací tabulky generované pro vnitřní tabulku.

Další informace naleznete v příručce *Administration Guide* v části týkající se koncepcí spojování (JOIN).

Doporučení z hlediska výkonu:

- Pro snížení počtu spojovaných řádků používejte lokální predikáty (tj. predikáty odkazující na jednu tabulku).
- Zvyšte velikost haldy řazení, aby byla dostatečně velká a umožňovala udržovat hašovací vyhledávací tabulku v paměti.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats..

INSERT

Jméno operátoru: INSERT

Reprezentuje: Vložení řádků do tabulky.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být vloženy.

IXAND

Jméno operátoru: IXAND

Reprezentuje: Průnik (AND) výsledků několika procházení indexů pomocí dynamických bitových map. Tento operátor umožňuje aplikovat predikáty spojené operací AND na několik indexů, a snížit tak na minimum přístup k odpovídajícím tabulkám.

Tento operátor se používá k následujícím akcím:

- Zmenšení množiny odpovídajících řádků před přístupem k základní tabulce.
- Průnik (AND) predikátů aplikovaných na několik indexů.
- Průnik (AND) výsledků polospojení (SEMIJOIN) používaných ve hvězdicových spojeních.

Doporučení z hlediska výkonu:

- Při aktualizacích databáze může průběžně docházet k fragmentaci indexu, což má za následek více stránek indexu, než je nezbytně nutné. To lze napravit odstraněním a opětovným vytvořením indexu nebo jeho reorganizací.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.
- Obecně platí, že je procházení indexu nejefektivnější, pokud dojde ke kvalifikování pouze malého počtu řádků. Pro odhad počtu kvalifikovaných řádků používá optimalizátor statistiky, které jsou k dispozici pro sloupce, na něž se odkazuje v predikátech. Pokud se některé hodnoty vyskytují častěji než jiné, je důležité pro příkaz `runstats` požadovat statistiku distribuce pomocí klauzule `WITH DISTRIBUTION`. Pomocí statistiky nerovnoměrné distribuce dokáže optimalizátor rozlišovat časté a méně časté hodnoty.
- Operátor IXAND lze nejlépe využít u indexů s jedním sloupcem, protože při používání operátoru IXAND jsou velmi důležité počáteční a koncové klíče.
- Pro hvězdicová spojení (STAR JOIN) vytvořte jednosloupcové indexy pro všechny nejselektivnější sloupce v tabulce faktů a souvisejících tabulkách dimenzí.

IXSCAN

Jméno operátoru: IXSCAN

Reprezentuje: Procházení indexu, aby se zredukoval proud řádků. Procházení může používat volitelné podmínky začátku/konce nebo se může vztahovat na indexovatelné predikáty odkazující na sloupce indexu.

Tato operace se provádí za účelem zmenšení množiny kvalifikovaných řádků před přístupem k základní tabulce (na základě predikátů).

Další informace naleznete v příručce *Administration Guide* v části týkající se procházení indexů.

Doporučení z hlediska výkonu:

- Při aktualizacích databáze může průběžně docházet k fragmentaci indexu, což má za následek více stránek indexu, než je nezbytně nutné. To lze napravit odstraněním a opětovným vytvořením indexu nebo jeho reorganizací.
- Při přístupu ke dvěma či více tabulkám lze přístup k vnitřní tabulce pomocí indexu zefektivnit poskytnutím indexu pro spojovaný sloupec vnější tabulky.
Další informace o používání indexů získáte v nápovědě online k modulu Vizuální vysvětlení.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.
- Obecně platí, že je procházení indexu neefektivnější, pokud dojde ke kvalifikování pouze malého počtu řádků. Pro odhad počtu kvalifikovaných řádků používá optimalizátor statistiky, které jsou k dispozici pro sloupce, na něž se odkazuje v predikátech. Pokud se některé hodnoty vyskytují častěji než jiné, je důležité pro příkaz `runstats` požadovat statistiku distribuce pomocí klauzule `WITH DISTRIBUTION`. Pomocí statistiky nerovnoměrné distribuce dokáže optimalizátor rozlišovat časté a méně časté hodnoty.

MSJOIN

Jméno operátoru: MSJOIN

Reprezentuje: Sloučené spojení (MERGE JOIN), pro které musí být kvalifikované řádky z vnější i vnitřní tabulky v pořadí predikátů spojení. Spojení MERGE JOIN se také říká *MERGE SCAN JOIN* neboli *spojení se seřazeným sloučením*.

Spojení je nutné v případě, že klauzule FROM obsahuje odkaz na více tabulek. Sloučené spojení (MERGE JOIN) je možné používat vždy v případě existence predikátu spojení srovnávajícího sloupce ze dvou různých tabulek. Může být také výsledkem přepsaného poddotazu.

Sloučené spojení (MERGE JOIN) vyžaduje také seřazený vstup spojovaných sloupců, protože tabulky jsou obvykle procházeny pouze jednou. Seřazení vstupu je docíleno přístupem k indexu nebo setříděné tabulce.

Další informace naleznete v příručce *Administration Guide* v části týkající se koncepcí spojování (JOIN).

Doporučení z hlediska výkonu:

- Pro snížení počtu spojovaných řádků použijte lokální predikáty (tj. predikáty odkazující na jednu tabulku).
Informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizuální vysvětlení.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.

NLJOIN

Jméno operátoru: NLJOIN

Reprezentuje: Spojení NESTED LOOP JOIN, které prochází (obvykle procházením indexu) vnitřní tabulku pro každý řádek vnější tabulky.

Spojení je nutné v případě, že klauzule FROM obsahuje odkaz na více tabulek. Spojení NESTED LOOP JOIN nevyžaduje predikát spojení. Obvykle však při jeho použití lépe pracuje.

Spojení NESTED LOOP JOIN se provádí jedním z následujících způsobů:

- Procházením vnitřní tabulky pro každý použitý řádek vnější tabulky.
- Prohledáváním indexu vnitřní tabulky pro každý použitý řádek vnější tabulky.

Další informace naleznete v příručce *Administration Guide* v části týkající se koncepcí spojování (JOIN).

Doporučení z hlediska výkonu:

- Spojení NESTED LOOP JOIN je obvykle efektivnější v případě používání indexu u sloupců predikátů spojení vnitřní tabulky (tabulka zobrazená napravo od operátoru NLJOIN). Zkontrolujte, že vnitřní tabulka je typu TBSCAN (nikoli IXSCAN). Pokud tomu tak je, zvažte přidání indexu k jejím sloupcům spojení.

Dalším (méně důležitým) způsobem zefektivnění spojování (JOIN) je vytvoření indexu pro sloupce spojení vnější tabulky, aby byla vnější tabulka seřazena.

Další informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.

- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.

Související informace:

- Hvězdicová spojení.

PIPE

Jméno operátoru: PIPE

Reprezentuje: Předání řádků dalším operátorům bez provedení jakýchkoli změn v řádcích.

(Tento operátor je pouze pro režim ladění.)

RETURN

Jméno operátoru: RETURN

Reprezentuje: Vrácení dat z dotazu uživateli. Jedná se o konečný operátor v grafu přístupového plánu, který ukazuje celkové akumulované hodnoty a náklady pro přístupový plán.

Tento operátor představuje nezbytnou operaci.

Doporučení z hlediska výkonu:

- Zkontrolujte, že jste použili predikáty, které načítají pouze potřebná data. Přesvědčte se například, zda hodnota selektivity pro predikáty představuje tu část tabulky, kterou chcete vrátit.

RIDSCN

Jméno operátoru: RIDSCN

Reprezentuje: Procházení seznamu identifikátorů řádku (RID) získaných z jednoho či více indexů.

Tento operátor bere optimalizátor do úvahy v případě, že:

- Predikáty jsou spojeny klíčovými slovy OR nebo existuje predikát IN. Lze použít metodu zvanou "index ORing", která kombinuje výsledky z několika přístupů pomocí indexů ke stejné tabulce.
- Pro přístup pomocí jednoho indexu je vhodné používat seznam načtení v předstihu, protože seřazení identifikátorů řádků před přístupem k základní tabulce zefektivňuje vstupně-výstupní operace.

RPD

Jméno operátoru: RPD

Reprezentuje: Operátor používaný ve federovaném systému k načtení dat ze vzdáleného zdroje dat pomocí nerelačního modulu wrapper.

Tento operátor bere optimalizátor do úvahy v případě, že obsahuje vzdálený plán, který nebude optimalizátorem kontrolován. Operátor RPD odešle požadavek vzdálenému nerelačnímu zdroji dat, aby získal výsledky dotazu. Požadavek je generován nerelačním modulem wrapper prostřednictvím rozhraní API podporovaného zdrojem dat.

Doporučení z hlediska výkonu: Viz Kapitola 4, Federated Database Query and Network Tuning Information (Dotazy na federované databáze a informace o ladění sítě), v příručce Administration Guide 2.

SHIP

Jméno operátoru: SHIP

Reprezentuje: Operátor používaný ve federovaném systému k načtení dat ze vzdáleného zdroje dat. Tento operátor bere optimalizátor do úvahy v případě, že obsahuje vzdálený plán, který nebude optimalizátorem kontrolován. Operátor SHIP odešle příkaz SQL SELECT vzdálenému zdroji dat, aby získal výsledky dotazu. Příkaz SELECT je generován pomocí dialektu jazyka SQL podporovaného zdrojem dat a může obsahovat libovolný platný dotaz povolený zdrojem dat.

Doporučení z hlediska výkonu: Viz Kapitola 4, Federated Database Query and Network Tuning Information (Dotazy na federované databáze a informace o ladění sítě), v příručce Administration Guide 2.

SORT

Jméno operátoru: SORT

Reprezentuje: Řazení řádků v tabulce podle jednoho či více sloupců tabulky s volitelným vyloučením duplicitních položek.

Řazení se vyžaduje v případě, že neexistují žádné indexy, které by odpovídaly požadovanému řazení, nebo v případě, že je řazení méně náročné než procházení indexu. Řazení se obvykle provádí jako poslední operace po načtení požadovaných řádků nebo se provádí za účelem řazení dat před spojením (JOIN) nebo seskupením (GROUP BY).

V případě velkého počtu řádků nebo v případě, že seřazená data nelze propojit, vyžaduje operace velmi náročné generování dočasných tabulek.

Další informace o řazení získáte v příručce *Administration Guide*.

Doporučení z hlediska výkonu:

- Zvažte přidání indexu ke sloupcům pro řazení.
Informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.
- Zkontrolujte, že jste použili predikáty, které načítají pouze potřebná data. Přesvědčte se například, zda hodnota selektivity pro predikáty představuje tu část tabulky, kterou chcete vrátit.
- Zkontrolujte, zda je velikost pro načítání v předstihu u systémového dočasného tabulkového prostoru adekvátní, tj. zda nepředstavuje omezení z hlediska vstupně-výstupních operací. (Chcete-li tuto skutečnost zkontrolovat, vyberte volbu **Příkaz->Zobrazit statistiku->Tabulkové prostory**.)
- Pokud se často vyžaduje rozsáhlé řazení, zvažte zvýšení hodnot následujících konfiguračních parametrů:

- Velikost haldy pro řazení (sortheap). Chcete-li tento parametr změnit, klepněte pravým tlačítkem myši na databázi v modulu Řídicí centrum a z rozevírací nabídky vyberte volbu *Konfigurovat*. V zobrazeném zápisníku klepněte na kartu **Výkon**.
- Práh haldy pro řazení (sheapthres). Chcete-li tento parametr změnit, klepněte pravým tlačítkem myši na instanci databáze v modulu Řídicí centrum a z rozevírací nabídky vyberte volbu *Konfigurovat*. V zobrazeném zápisníku klepněte na kartu **Výkon**.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats..

TBSCAN

Jméno operátoru: TBSCAN

Reprezentuje: Procházení tabulky (procházení relace), které načítá řádky čtením všech požadovaných dat přímo z datových stránek.

Tento typ procházení zvolí optimalizátor místo procházení indexu v těchto případech:

- Daný rozsah hodnot je procházen často (tj. je třeba přistupovat k většině dat tabulky).
- Tabulka je malá.
- Dělení indexu do klastrů je na nízké úrovni.
- Index neexistuje.

Další informace o procházení tabulek a indexů získáte v příručce *Administration Guide*.

Doporučení z hlediska výkonu:

- Procházení indexu je efektivnější než procházení tabulky, pokud je tabulka rozsáhlá a přistupuje se pouze k malému počtu řádků této tabulky. Ke zvýšení pravděpodobnosti, že bude optimalizátor v těchto situacích používat procházení indexu, zvažte přidání indexů ke sloupcům, ke kterým existují výběrové predikáty. Další informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.
- Pokud již index existuje, avšak nebyl použit, zkontrolujte, zda u všech jeho úvodních sloupců existují výběrové predikáty. Jestliže tyto predikáty existují, je třeba dále zkontrolovat, zda je stupeň dělení do klastrů pro index vysoký. (Chcete-li zobrazit tuto statistiku, otevřete okno Statistika tabulek pro tabulku pod řazením a vyberte její tlačítko *Indexy*. Tím zobrazíte okno Statistika indexů.)
- Zkontrolujte, zda je velikost pro načítání v předstihu u tabulkového prostoru adekvátní, tj. zda nedochází k omezení z hlediska vstupně-výstupních operací. (Chcete-li tuto skutečnost zkontrolovat, vyberte volbu **Příkaz->Zobrazit statistiku->Tabulkové prostory**.)

Další informace naleznete v části týkající se načítání dat v předstihu do fondu vyrovnávacích pamětí v příručce *Administration Guide*.

- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.
Statistika kvantilů a častých hodnot poskytuje informace o selektivitě predikátů. Určuje například, kdy se dává přednost procházení indexu před procházením tabulky. K aktualizaci těchto hodnot použijte u tabulky s klauzulí `WITH DISTRIBUTION` příkaz `runstats`.

TEMP

Jméno operátoru: TEMP

Reprezentuje: Akce ukládání dat do dočasné tabulky, která bude čtena jiným operátorem (a to i vícekrát). Po zpracování příkazu SQL je tabulka odebrána (v některých případech i dříve).

Tento operátor se vyžaduje k vyhodnocení dílčích dotazů nebo k uložení dočasných výsledků. Je možné, že v některých situacích (například pokud lze příkaz aktualizovat), bude povinný.

TQUEUE

Jméno operátoru: TQUEUE

Reprezentuje: Fronta tabulek, která se používá k předávání dat tabulek mezi databázovými agenty (pokud dotaz zpracovává více databázových agentů). Více databázových agentů se používá v případě paralelního zpracování dotazu.

Typy front tabulek:

- **Lokální:** Tato fronta tabulek se používá k předávání dat mezi databázovými agenty v rámci jednoho uzlu. Lokální fronta tabulek se používá pro paralelní zpracování v rámci oblasti.
- **Nelokální:** Tato fronta tabulek se používá k předávání dat mezi databázovými agenty na rozdílných uzlech.

UNION

Jméno operátoru: UNION

Reprezentuje: Zřetězení proudu řádků z několika tabulek.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být zřetězeny.

UNIQUE

Jméno operátoru: UNIQUE

Reprezentuje: Vyločení řádků s duplicitními hodnotami pro zadané sloupce.

Doporučení z hlediska výkonu:

- Tento operátor není nutné použít pouze v případě, že pro příslušné sloupce existuje jedinečný index.

Informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.

UPDATE

Jméno operátoru: UPDATE

Reprezentuje: Aktualizace dat v řádcích tabulky.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být aktualizovány.

Optimalizátor

Optimalizátor je komponentou kompilátoru SQL, která provádí výběr přístupového plánu pro příkaz SQL jazyka DML (data manipulation language). Při této operaci modeluje náklady na provedení mnoha alternativních přístupových plánů a poté vybere plán s minimálními odhadovanými náklady.

Balík

Balík je objekt uložený v databázi obsahující informace potřebné pro zpracování příkazů SQL asociovaných s jedním zdrojovým souborem aplikačního programu. Je generován při některé z následujících operací:

- předběžná kompilace zdrojového souboru příkazem **PREP**,
- svázání vázaného souboru generovaného prekompilátorem příkazem **BIND**.

Predikát

Predikát je prvek podmínky pro vyhledávání, který vyjadřuje či zahrnuje operaci porovnání. Predikáty jsou obsaženy v klauzulích začínajících klíčovými slovy WHERE nebo HAVING.

Příklad: V následujícím příkazu SQL:

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

Jsou uvedeny následující predikáty: NAME = 'SMITH'; DEPT = 895;
a YEARS > 5.

Predikáty lze zařadit do následujících kategorií, seřazených od neúčinnější k nejméně účinné:

1. Počáteční a koncové podmínky omezují (zdužují) procházení indexu. (Tyto podmínky jsou označovány také jako predikáty pro omezení rozsahu.)
2. Predikáty typu indexová stránka (s možností vyhledání v indexu) lze vyhodnotit prostřednictvím indexu, protože sloupce obsažené v predikátu jsou součástí klíče indexu.
3. Predikáty typu datová stránka (s možností vyhledání v datech) nelze vyhodnotit prostřednictvím indexu, avšak lze je vyhodnotit při uložení řádků ve vyrovnávací paměti.
4. Zbytkové predikáty obvykle vyžadují jinou operaci I/O než jednoduchý přístup do základní tabulky a lze je použít teprve po zkopírování dat ze stránky vyrovnávací paměti. Patří mezi ně predikáty obsahující dílčí dotazy a predikáty s načtením dat typu LONG VARCHAR nebo LOB uložených v souborech odděleně od tabulky.

Při návrhu predikátů je třeba mít na zřeteli co nejvyšší možnou selektivitu, aby bylo navrženo co nejméně řádků.

Neúčinnější a nepoužívanější jsou následující typy predikátů:

- Pro spojení se sloučením je vyžadován *predikát spojení jednoduché rovnosti*. Tento predikát má tvar tabulka1.sloupec = tabulka2.sloupec a umožňuje porovnat sloupce ve dvou různých tabulkách, aby bylo možné tabulky spojit.
- *Lokální predikát* je používán pouze pro jednu tabulku.

Další informace najdete v oddílech týkajících se koncepce a optimalizace přístupu k datům v příručce *Administration Guide*.

Třída optimalizace dotazu

Třída optimalizace dotazů je sada pravidel pro přepis dotazů a optimalizačních postupů pro kompilaci dotazů.

Primární třídy optimalizace dotazů jsou následující:

- 1 Omezená optimalizace. Tento typ je užitečný při silně omezených paměťových prostředcích a prostředcích pro zpracování. Zhruba ekvivalentní optimalizaci poskytované ve verzi 1.

- 2 Mírná optimalizace. Určuje stupeň optimalizace vyšší než ve verzi 1, avšak při podstatně nižších optimalizačních nákladech než u úrovně 3 a vyšší, zvláště pro velmi složité dotazy.
- 3 Střední optimalizace. Nejvíce se přibližuje charakteristice optimalizace dotazů DB2 pro /ESA.
- 5 Běžná optimalizace. Doporučená pro smíšená prostředí se zpracováním jednoduchých transakcí i složitých dotazů.
- 7 Běžná optimalizace. Shodná s optimalizací dotazů 5 kromě toho, že neomezuje míru optimalizace dotazů pro složité dynamické dotazy SQL.

Další třídy optimalizace dotazů pro použití pouze ve speciálních případech jsou následující:

- 0 Minimální optimalizace. Určena pouze pro případy, kdy je vyžadována pouze malá či nulová míra optimalizace (tzn. pro velmi jednoduché dotazy na tabulkách s dobře zpracovanými indexy).
- 9 Maximální optimalizace. Využívá ve velké míře paměťové prostředky a prostředky pro zpracování. Použijte pouze při nedostatečnosti třídy 5 (tzn. pro velmi složité dotazy a dotazy s dlouhým zpracováním, které nelze dobře zpracovat na úrovni třídy 5).

Obecně řečeno, vyšší třídy optimalizace používejte pro statické dotazy a pro dotazy, u nichž předpokládáte dlouhou dobu zpracování, zatímco nižší třídy optimalizace používejte pro jednoduché dotazy spouštěné dynamicky nebo pro dotazy spouštěné pouze s malým počtem opakování.

Chcete-li nastavit optimalizaci dotazů pro dynamické příkazy SQL, zadejte v příkazovém procesoru následující příkaz:

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

kde 'n' označuje požadovanou třídu optimalizace dotazů.

Chcete-li nastavit optimalizaci dotazů pro statické příkazy SQL, použijte u příkazu **BIND** nebo **PREP** volbu **QUERYOPT**.

Další informace najdete v sekci týkající se přizpůsobení třídy optimalizace v příručce *Administration Guide*.

Selektivita predikátů

Selektivita je ekvivalentní s pravděpodobností, že všechny řádky budou splňovat podmínku predikátu (tzn. že podmínka bude pravdivá).

Příklad: Selektivita s hodnotou 0,01 (1%) pro určitý predikát uplatněný na tabulku s 1.000.000 řádky znamená, že predikát odhadem navrátí 10.000 řádků (1% z 1.000.000) a že vyřadí zbývajících 990.000 řádků.

Žádoucí jsou predikáty s vysokou selektivitou (s hodnotou selektivity 0,10 nebo nižší). Takové predikáty navracejí méně řádků pro zpracování dalšími operátory, a díky tomu vyžadují pro zpracování dotazu méně operací procesoru a méně operací I/O.

Příklad

Předpokládejme, že máte tabulku s 1.000.000 řádků a že původní dotaz obsahuje klauzuli 'ORDER BY' vyžadující další krok řazení. S predikátem s hodnotou selektivity 0,01 bude třeba provést řazení přibližně pro 10.000 řádků. Avšak s predikátem s nižší mírou selektivity (hodnota 0,5) bude třeba provést řazení přibližně pro 500.000 řádků, což si vyžádá mnohem více času pro operace procesoru a operace I/O.

Hvězdicové spojení

Sada spojení je označována jako hvězdicové spojení, pokud tabulka faktů (rozměrná centrální tabulka) je spojena se dvěma nebo třemi tabulkami dimenzí (menší tabulky obsahující popisy hodnot sloupců v tabulce faktů).

Hvězdicové spojení sestává ze tří hlavních částí:

- Částečná spojení
- Logický součin (AND) indexů výsledků částečných spojení
- Kompletace částečných spojení

Jeví se jako dvě nebo více spojení pro vyplnění operátoru IXAND.

Částečné spojení je speciální formou spojení, u níž je výsledkem spojení namísto spojení sloupců vnitřní a vnější tabulky pouze identifikátor RID (identifikátor řádku) vnitřní tabulky.

U hvězdicových spojení dodávají částečná spojení identifikátory řádků pro operátor logického součinu (AND) indexů. Operátor logického součinu indexů akumuluje vliv filtrování různých spojení. Výstup z operátoru logického součinu indexů je předán do operátoru logického součtu (OR), který seřadí identifikátory řádků a vyloučí všechny duplicitní řádky, které mohou být výsledkem spojení poskytujících data operátoru logického součinu indexů. Poté jsou pomocí operátoru Fetch načteny řádky z tabulky faktů. Nakonec je redukováná tabulka faktů spojena se všemi tabulkami dimenzí, a tím jsou spojení dokončena.

Návrhy pro výkon:

- Vytvořte indexy na tabulce faktů pro všechna spojení tabulek dimenzí.

- Zkontrolujte, zda je prahová hodnota haldy řazení dostatečně vysoká, aby bylo možné alokovat bitový filtr operátoru logického součinu indexů. Pro hvězdicová spojení může být potřeba až 12 MB, neboli 3000 stránek o velikosti 4 kB. Pro paralelní zpracování v rámci oblasti je bitový filtr alokován ze stejného segmentu paměti jako halda dbheap, avšak je omezen hodnotou sortheap (a v rámci instance hodnotou sheapthres). Sdílená paměť je proto řízena hodnotami sortheap a sheapthres a může vyžadovat mnohem více prostoru než 12 MB.
- Použijte predikáty filtrování na tabulky dimenzí. Pokud nejsou statistické údaje aktuální, aktualizujte je pomocí příkazu runstats.

Statické příkazy SQL

Statický příkaz SQL je vložen v aplikačním programu. Před provedením aplikace musí být všechny vložené příkazy předběžně zkompileovány a svázány do *balíku*.

Pokud program DB2 tyto příkazy zkompileje, vytvoří pro každý z nich přístupový plán založený na statistice katalogu a na konfiguračních parametrech aktuálních při předběžné kompilaci a svázání.

Tyto přístupové plány jsou použity vždy při spuštění aplikace. Jsou změněny až při novém svázání balíku.

Alternativou ke statickému příkazu SQL je dynamický příkaz SQL.

Tabulkové prostory prostoru spravovaného systémem (SMS)

V databázi mohou existovat dva typy tabulkových prostorů: prostor spravovaný systémem (SMS) a prostor spravovaný databází (DMS).

Tabulkový prostor SMS je spravován operačním systémem, který ukládá data databáze do prostoru alokovaného v okamžiku vytvoření tabulkového prostoru. Definice tabulkového prostoru obsahuje seznam jedné nebo více cest k adresářům, kde jsou tato data uložena.

Alokaci a správu média s úložným prostorem provádí souborový systém.

Tabulkové prostory SMS a DMS mohou existovat souběžně ve stejné databázi.

Tabulkový prostor

Správu velmi rozsáhlých databází lze usnadnit jejich rozdělením do samostatně spravovaných částí nazývaných *tabulkové prostory*.

Tabulkový prostor umožňuje asociovat umístění dat s určitými logickými zařízeními nebo s jejich částmi. Při vytváření tabulky můžete například určit, že její indexy nebo její dlouhé sloupce s dlouhými nebo rozsáhlými datovými objekty (LOB) mají být uloženy odděleně od ostatních dat tabulky.

Tabulkový prostor může být za účelem zvýšení výkonu rozdělen do více fyzických úložných zařízení (kontejnerů). Doporučuje se však zajistit, aby všechna zařízení a kontejnery v rámci tabulkového prostoru měly podobnou charakteristiku výkonu.

Tabulkový prostor lze spravovat dvěma různými způsoby: jako prostor spravovaný systémem (SMS) nebo jako prostor spravovaný databází (DMS).

Vizuální vysvětlení

Poznámka: Ve verzi 6 již nelze modul Vizuální vysvětlení volat z příkazového řádku. Lze jej však stále volat z různých databázových objektů v okně Řídicí centrum. V této verzi je v dokumentaci stále používán název Vizuální vysvětlení.

Modul Vizuální vysvětlení umožňuje zobrazit přístupový plán pro vysvětlené příkazy SQL pomocí grafu. Údaje dostupné v grafu můžete použít k vyladění dotazů SQL s cílem lepšího výkonu.

V přístupovém plánu jsou zobrazeny podrobné údaje o následujících objektech:

- tabulky (a jejich asociované sloupce) a indexy,
- operátory (jako je například procházení tabulky, řazení či spojení),
- tabulkové prostory a funkce.

V modulu Vizuální vysvětlení lze dále provádět následující operace:

- Zobrazení statistických údajů použitých při optimalizaci. Poté lze tyto statistické údaje porovnat s aktuálními statistickými údaji katalogu a na základě porovnání určit, zda může nové svázání balíku vést ke zlepšení výkonu.
- Určení, zda byl pro přístup k tabulce použit index, či nikoli. Pokud index nebyl použit, modul Vizuální vysvětlení umožní určit, které sloupce by bylo výhodné opatřit indexem.
- Zobrazení účinků různých postupů pro vyladění porovnáním verzí grafu přístupového plánu pro dotaz před laděním a po něm.
- Získání informací o každé operaci v přístupovém plánu, včetně celkových odhadovaných nákladů a počtu načtených řádků (kardinalita).

Dodatek B. Abecední seznam operátorů modulu Vizuální vysvětlení

CMPEXP

Jméno operátoru: CMPEXP

Reprezentuje: Výpočet výrazů požadovaných pro přechodné nebo konečné výsledky.

(Tento operátor je pouze pro režim ladění.)

DELETE

Jméno operátoru: DELETE

Reprezentuje: Odstranění řádků z tabulky.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady na přístupový plán, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být odstraněny.

Doporučení z hlediska výkonu:

- Pokud odstraňujete všechny řádky z tabulky, zvažte použití příkazu DROP TABLE nebo **LOAD REPLACE**.

EISCAN

Jméno operátoru: EISCAN

Reprezentuje: Tento operátor prochází uživatelský index, aby zredukoval proud řádků. Pro procházení se používá několik podmínek počátku/konce z funkce vytváření rozsahu zadané uživatelem.

Tato operace se provádí za účelem zmenšení množiny kvalifikovaných řádků před přístupem k základní tabulce (na základě predikátů).

Doporučení z hlediska výkonu:

- Při aktualizacích databáze může průběžně docházet k fragmentaci indexu, což má za následek více stránek indexu, než je nezbytně nutné. To lze napravit odstraněním a opětovným vytvořením indexu nebo jeho reorganizací.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats.

FETCH

Jméno operátoru: FETCH

Reprezentuje: Načtení sloupců z tabulky pomocí identifikátoru specifického řádku (RID).

Doporučení z hlediska výkonu:

- Rozšířením klíčů indexu zahrňte načítané sloupce, aby nebylo třeba přistupovat k datovým stránkám.
- Vyhledejte index vztahující se na načtená data a dvojným klepnutím na jeho uzel zobrazte okno statistiky. Zkontrolujte, že je stupeň dělení do klastrů pro index vysoký.
- Pokud je počet vstupně-výstupních operací (I/O) při načítání dat vyšší než počet stránek v tabulce, zvyšte velikost vyrovnávací paměti.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.

Statistika kvantilů a častých hodnot poskytuje informace o selektivě predikátů, která určuje, kdy se dává přednost procházení indexu před procházením tabulky. K aktualizaci této statistiky použijte u tabulky s klauzulí `WITH DISTRIBUTION` příkaz `runstats`.

FILTER

Jméno operátoru: FILTER

Reprezentuje: Aplikace zbytkových predikátů k tomu, aby byla data filtrována na základě kritérií zadaných predikáty.

Doporučení z hlediska výkonu:

- Zkontrolujte, že jste použili predikáty, které načítají pouze potřebná data. Přesvědčte se například, zda hodnota selektivity pro predikáty představuje tu část tabulky, kterou chcete vrátit.
- Přesvědčte se, že je hodnota třídy optimalizace alespoň 3, aby optimalizátor místo poddotazů používal spojení (JOIN). Pokud to není možné, pokuste se přepsat dotaz SQL ručně tak, aby byl poddotaz eliminován. Příklad je uveden v příručce *Administration Guide* v části týkající se přepisování dotazů kompilátorem SQL.

GENROW

Jméno operátoru: GENROW

Reprezentuje: Vestavěná funkce, která generuje tabulku řádků bez vstupu z tabulek, indexů či operátorů.

Operátor GENROW lze používat optimalizátorem ke generování řádků dat (například pro příkaz `INSERT` nebo pro některé seznamy `IN` transformované do spojení).

Chcete-li zobrazit odhadovanou statistiku pro tabulky generované funkcí GENROW, klepněte dvakrát na příslušný uzel.

GRPBY

Jméno operátoru: GRPBY

Reprezentuje: Seskupení řádků podle společných hodnot určených sloupců nebo funkcí. Tato operace je nutná k vytvoření skupiny hodnot nebo k vyhodnocení množinových funkcí.

Operátor GRPBY lze použít i v případě, že nejsou zadány žádné sloupce GROUP BY, a to pokud seznam SELECT obsahuje agregační funkce, což označuje, že se celá tabulka při této agregaci považuje za jedinou skupinu.

Doporučení z hlediska výkonu:

- Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být seskupeny.
- Pokud chcete zvýšit výkon příkazu SELECT, který obsahuje jednu agregační funkci, ale žádnou klauzuli GROUP BY, zkuste provést následující postup:
 - V případě agregační funkce MIN(C) vytvořte vzestupný index pro sloupec C.
 - V případě agregační funkce MAX(C) vytvořte sestupný index pro sloupec C.

HSJOIN

Jméno operátoru: HSJOIN

Reprezentuje: Hašované spojení, pro které jsou příslušné řádky z tabulek hašovány za účelem přímého spojení bez předběžného uspořádání obsahu tabulek.

Spojení je nutné v případě, že klauzule FROM obsahuje odkaz na více tabulek. Hašované spojení je možné používat vždy v případě existence predikátu spojení srovnávajícího sloupce ze dvou různých tabulek. Predikáty spojení musí mít přesně stejný datový typ. Hašovaná spojení mohou být také výsledkem přešpaného poddotazu, jako v případě operátoru NLJOIN.

Hašované spojení nevyžaduje, aby byly vstupní tabulky seřazeny. Spojení probíhá procházením vnitřní tabulky hašovaného spojení a generováním vyhledávací tabulky hašováním hodnot spojovaného sloupce. Následně je čtena vnější tabulka, je prováděno hašování hodnot spojovaného sloupce a dohledání ve vyhledávací tabulky generované pro vnitřní tabulku.

Další informace naleznete v příručce *Administration Guide* v části týkající se koncepcí spojování (JOIN).

Doporučení z hlediska výkonu:

- Pro snížení počtu spojovaných řádků používejte lokální predikáty (tj. predikáty odkazující na jednu tabulku).
- Zvyšte velikost haldy řazení, aby byla dostatečně velká a umožňovala udržovat hašovací vyhledávací tabulku v paměti.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`..

INSERT

Jméno operátoru: INSERT

Reprezentuje: Vložení řádků do tabulky.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být vloženy.

IXAND

Jméno operátoru: IXAND

Reprezentuje: Průnik (AND) výsledků několika procházení indexů pomocí dynamických bitových map. Tento operátor umožňuje aplikovat predikáty spojené operací AND na několik indexů, a snížit tak na minimum přístup k odpovídajícím tabulkám.

Tento operátor se používá k následujícím akcím:

- Zmenšení množiny odpovídajících řádků před přístupem k základní tabulce.
- Průnik (AND) predikátů aplikovaných na několik indexů.
- Průnik (AND) výsledků polospojení (SEMIJOIN) používaných ve hvězdicových spojeních.

Doporučení z hlediska výkonu:

- Při aktualizacích databáze může průběžně docházet k fragmentaci indexu, což má za následek více stránek indexu, než je nezbytně nutné. To lze napravit odstraněním a opětovným vytvořením indexu nebo jeho reorganizací.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`..
- Obecně platí, že je procházení indexu neefektivnější, pokud dojde ke kvalifikování pouze malého počtu řádků. Pro odhad počtu kvalifikovaných řádků používá optimalizátor statistiky, které jsou k dispozici pro sloupce, na něž se odkazuje v predikátech. Pokud se některé hodnoty vyskytují častěji než jiné, je důležité pro příkaz `runstats` požadovat statistiku distribuce pomocí klauzule `WITH DISTRIBUTION`. Pomocí statistiky nerovnoměrné distribuce dokáže optimalizátor rozlišovat časté a méně časté hodnoty.

- Operátor IXAND lze nejlépe využít u indexů s jedním sloupcem, protože při používání operátoru IXAND jsou velmi důležité počáteční a koncové klíče.
- Pro hvězdicová spojení (STAR JOIN) vytvořte jednosloupcové indexy pro všechny nejselektivnější sloupce v tabulce faktů a souvisejících tabulkách dimenzí.

IXSCAN

Jméno operátoru: IXSCAN

Reprezentuje: Procházení indexu, aby se zredukoval proud řádků. Procházení může používat volitelné podmínky začátku/konce nebo se může vztahovat na indexovatelné predikáty odkazující na sloupce indexu.

Tato operace se provádí za účelem zmenšení množiny kvalifikovaných řádků před přístupem k základní tabulce (na základě predikátů).

Další informace naleznete v příručce *Administration Guide* v části týkající se procházení indexů.

Doporučení z hlediska výkonu:

- Při aktualizacích databáze může průběžně docházet k fragmentaci indexu, což má za následek více stránek indexu, než je nezbytně nutné. To lze napravit odstraněním a opětovným vytvořením indexu nebo jeho reorganizací.
- Při přístupu ke dvěma či více tabulkám lze přístup k vnitřní tabulce pomocí indexu zefektivnit poskytnutím indexu pro spojovaný sloupec vnější tabulky.

Další informace o používání indexů získáte v nápovědě online k modulu Vizualní vysvětlení.

- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.
- Obecně platí, že je procházení indexu nejefektivnější, pokud dojde ke kvalifikování pouze malého počtu řádků. Pro odhad počtu kvalifikovaných řádků používá optimalizátor statistiky, které jsou k dispozici pro sloupce, na něž se odkazuje v predikátech. Pokud se některé hodnoty vyskytují častěji než jiné, je důležité pro příkaz `runstats` požadovat statistiku distribuce pomocí klauzule `WITH DISTRIBUTION`. Pomocí statistiky nerovnoměrné distribuce dokáže optimalizátor rozlišovat časté a méně časté hodnoty.

MSJOIN

Jméno operátoru: MSJOIN

Reprezentuje: Sloučené spojení (MERGE JOIN), pro které musí být kvalifikované řádky z vnější i vnitřní tabulky v pořadí predikátů spojení. Spojení MERGE JOIN se také říká *MERGE SCAN JOIN* neboli *spojení se seřazeným sloučením*.

Spojení je nutné v případě, že klauzule FROM obsahuje odkaz na více tabulek. Sloučené spojení (MERGE JOIN) je možné používat vždy v případě existence predikátu spojení srovnávajícího sloupce ze dvou různých tabulek. Může být také výsledkem přeepsaného poddotazu.

Sloučené spojení (MERGE JOIN) vyžaduje také seřazený vstup spojovaných sloupců, protože tabulky jsou obvykle procházeny pouze jednou. Seřazení vstupu je docíleno přístupem k indexu nebo setříděné tabulce.

Další informace naleznete v příručce *Administration Guide* v části týkající se koncepcí spojování (JOIN).

Doporučení z hlediska výkonu:

- Pro snížení počtu spojovaných řádků použijte lokální predikáty (tj. predikáty odkazující na jednu tabulku).

Informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.

- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats..

NLJOIN

Jméno operátoru: NLJOIN

Reprezentuje: Spojení NESTED LOOP JOIN, které prochází (obvykle procházením indexu) vnitřní tabulku pro každý řádek vnější tabulky.

Spojení je nutné v případě, že klauzule FROM obsahuje odkaz na více tabulek. Spojení NESTED LOOP JOIN nevyžaduje predikát spojení. Obvykle však při jeho použití lépe pracuje.

Spojení NESTED LOOP JOIN se provádí jedním z následujících způsobů:

- Procházením vnitřní tabulky pro každý použitý řádek vnější tabulky.
- Prohledáváním indexu vnitřní tabulky pro každý použitý řádek vnější tabulky.

Další informace naleznete v příručce *Administration Guide* v části týkající se koncepcí spojování (JOIN).

Doporučení z hlediska výkonu:

- Spojení NESTED LOOP JOIN je obvykle efektivnější v případě používání indexu u sloupců predikátů spojení vnitřní tabulky (tabulka zobrazená napravo od operátoru NLJOIN). Zkontrolujte, že vnitřní tabulka je typu TBSCAN (nikoli IXSCAN). Pokud tomu tak je, zvažte přidání indexu k jejím sloupcům spojení.

Dalším (méně důležitým) způsobem zefektivnění spojování (JOIN) je vytvoření indexu pro sloupce spojení vnější tabulky, aby byla vnější tabulka seřazena.

Další informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.

- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.

Související informace:

- Hvězdicová spojení.

PIPE

Jméno operátoru: PIPE

Reprezentuje: Předání řádků dalším operátorům bez provedení jakýchkoli změn v řádcích.

(Tento operátor je pouze pro režim ladění.)

RETURN

Jméno operátoru: RETURN

Reprezentuje: Vrácení dat z dotazu uživateli. Jedná se o konečný operátor v grafu přístupového plánu, který ukazuje celkové akumulované hodnoty a náklady pro přístupový plán.

Tento operátor představuje nezbytnou operaci.

Doporučení z hlediska výkonu:

- Zkontrolujte, že jste použili predikáty, které načítají pouze potřebná data. Přesvědčte se například, zda hodnota selektivity pro predikáty představuje tu část tabulky, kterou chcete vrátit.

RIDSCN

Jméno operátoru: RIDSCN

Reprezentuje: Procházení seznamu identifikátorů řádku (RID) získaných z jednoho či více indexů.

Tento operátor bere optimalizátor do úvahy v případě, že:

- Predikáty jsou spojeny klíčovými slovy OR nebo existuje predikát IN. Lze použít metodu zvanou "index ORing", která kombinuje výsledky z několika přístupů pomocí indexů ke stejné tabulce.
- Pro přístup pomocí jednoho indexu je vhodné používat seznam načtení v předstihu, protože seřazení identifikátorů řádků před přístupem k základní tabulce zefektivňuje vstupně-výstupní operace.

RPD

Jméno operátoru: RPD

Reprezentuje: Operátor používaný ve federovaném systému k načtení dat ze vzdáleného zdroje dat pomocí nerelačního modulu wrapper.

Tento operátor bere optimalizátor do úvahy v případě, že obsahuje vzdálený plán, který nebude optimalizátorem kontrolován. Operátor RPD odešle požadavek vzdálenému nerelačnímu zdroji dat, aby získal výsledky dotazu. Požadavek je generován nerelačním modulem wrapper prostřednictvím rozhraní API podporovaného zdrojem dat.

Doporučení z hlediska výkonu: Viz Kapitola 4, Federated Database Query and Network Tuning Information (Dotazy na federované databáze a informace o ladění sítě), v příručce Administration Guide 2.

SHIP

Jméno operátoru: SHIP

Reprezentuje: Operátor používaný ve federovaném systému k načtení dat ze vzdáleného zdroje dat. Tento operátor bere optimalizátor do úvahy v případě, že obsahuje vzdálený plán, který nebude optimalizátorem kontrolován. Operátor SHIP odešle příkaz SQL SELECT vzdálenému zdroji dat, aby získal výsledky dotazu. Příkaz SELECT je generován pomocí dialektu jazyka SQL podporovaného zdrojem dat a může obsahovat libovolný platný dotaz povolený zdrojem dat.

Doporučení z hlediska výkonu: Viz Kapitola 4, Federated Database Query and Network Tuning Information (Dotazy na federované databáze a informace o ladění sítě), v příručce Administration Guide 2.

SORT

Jméno operátoru: SORT

Reprezentuje: Řazení řádků v tabulce podle jednoho či více sloupců tabulky s volitelným vyloučením duplicitních položek.

Řazení se vyžaduje v případě, že neexistují žádné indexy, které by odpovídaly požadovanému řazení, nebo v případě, že je řazení méně náročné než procházení indexu. Řazení se obvykle provádí jako poslední operace po načtení požadovaných řádků nebo se provádí za účelem řazení dat před spojením (JOIN) nebo seskupením (GROUP BY).

V případě velkého počtu řádků nebo v případě, že seřazená data nelze propojit, vyžaduje operace velmi náročné generování dočasných tabulek.

Další informace o řazení získáte v příručce *Administration Guide*.

Doporučení z hlediska výkonu:

- Zvažte přidání indexu ke sloupcům pro řazení.
Informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizuální vysvětlení.
- Zkontrolujte, že jste použili predikáty, které načítají pouze potřebná data. Přesvědčte se například, zda hodnota selektivity pro predikáty představuje tu část tabulky, kterou chcete vrátit.
- Zkontrolujte, zda je velikost pro načítání v předstihu u systémového dočasného tabulkového prostoru adekvátní, tj. zda nepředstavuje omezení z hlediska vstupně-výstupních operací. (Chcete-li tuto skutečnost zkontrolovat, vyberte volbu **Příkaz->Zobrazit statistiku->Tabulkové prostory**.)
- Pokud se často vyžaduje rozsáhlé řazení, zvažte zvýšení hodnot následujících konfiguračních parametrů:
 - Velikost haldy pro řazení (sortheap). Chcete-li tento parametr změnit, klepněte pravým tlačítkem myši na databázi v modulu Řídicí centrum a z rozevírací nabídky vyberte volbu *Konfigurovat*. V zobrazeném zápisníku klepněte na kartu **Výkon**.
 - Práh haldy pro řazení (sheapthres). Chcete-li tento parametr změnit, klepněte pravým tlačítkem myši na instanci databáze v modulu Řídicí centrum a z rozevírací nabídky vyberte volbu *Konfigurovat*. V zobrazeném zápisníku klepněte na kartu **Výkon**.
- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu runstats..

TBSCAN

Jméno operátoru: TBSCAN

Reprezentuje: Procházení tabulky (procházení relace), které načítá řádky čtením všech požadovaných dat přímo z datových stránek.

Tento typ procházení zvolí optimalizátor místo procházení indexu v těchto případech:

- Daný rozsah hodnot je procházen často (tj. je třeba přistupovat k většině dat tabulky).
- Tabulka je malá.
- Dělení indexu do klastrů je na nízké úrovni.
- Index neexistuje.

Další informace o procházení tabulek a indexů získáte v příručce *Administration Guide*.

Doporučení z hlediska výkonu:

- Procházení indexu je efektivnější než procházení tabulky, pokud je tabulka rozsáhlá a přistupuje se pouze k malému počtu řádků této tabulky. Ke zvýšení

pravděpodobnosti, že bude optimalizátor v těchto situacích používat procházení indexu, zvažte přidání indexů ke sloupcům, ke kterým existují výběrové predikáty. Další informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.

- Pokud již index existuje, avšak nebyl použit, zkontrolujte, zda u všech jeho úvodních sloupců existují výběrové predikáty. Jestliže tyto predikáty existují, je třeba dále zkontrolovat, zda je stupeň dělení do klastrů pro index vysoký. (Chcete-li zobrazit tuto statistiku, otevřete okno Statistika tabulek pro tabulku pod řazením a vyberte její tlačítko *Indexy*. Tím zobrazíte okno Statistika indexů.)
- Zkontrolujte, zda je velikost pro načítání v předstihu u tabulkového prostoru adekvátní, tj. zda nedochází k omezení z hlediska vstupně-výstupních operací. (Chcete-li tuto skutečnost zkontrolovat, vyberte volbu **Příkaz->Zobrazit statistiku->Tabulkové prostory**.)

Další informace naleznete v části týkající se načítání dat v předstihu do fondu vyrovnávacích pamětí v příručce *Administration Guide*.

- Pokud není statistika aktuální, aktualizujte ji pomocí příkazu `runstats`.

Statistika kvantilů a častých hodnot poskytuje informace o selektivitě predikátů. Určuje například, kdy se dává přednost procházení indexu před procházením tabulky. K aktualizaci těchto hodnot použijte u tabulky s klauzulí `WITH DISTRIBUTION` příkaz `runstats`.

TEMP

Jméno operátoru: TEMP

Reprezentuje: Akce ukládání dat do dočasné tabulky, která bude čtena jiným operátorem (a to i vícekrát). Po zpracování příkazu SQL je tabulka odebrána (v některých případech i dříve).

Tento operátor se vyžaduje k vyhodnocení dílčích dotazů nebo k uložení dočasných výsledků. Je možné, že v některých situacích (například pokud lze příkaz aktualizovat), bude povinný.

TQUEUE

Jméno operátoru: TQUEUE

Reprezentuje: Fronta tabulek, která se používá k předávání dat tabulek mezi databázovými agenty (pokud dotaz zpracovává více databázových agentů). Více databázových agentů se používá v případě paralelního zpracování dotazu.

Typy front tabulek:

- **Lokální:** Tato fronta tabulek se používá k předávání dat mezi databázovými agenty v rámci jednoho uzlu. Lokální fronta tabulek se používá pro paralelní zpracování v rámci oblasti.
- **Nelokální:** Tato fronta tabulek se používá k předávání dat mezi databázovými agenty na rozdílných uzlech.

UNION

Jméno operátoru: UNION

Reprezentuje: Zřetězení proudu řádků z několika tabulek.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být zřetězeny.

UNIQUE

Jméno operátoru: UNIQUE

Reprezentuje: Vyložení řádků s duplicitními hodnotami pro zadané sloupce.

Doporučení z hlediska výkonu:

- Tento operátor není nutné použít pouze v případě, že pro příslušné sloupce existuje jedinečný index.

Informace o vytváření indexů získáte v tématu Vytváření vhodných indexů v nápovědě online k modulu Vizualní vysvětlení.

UPDATE

Jméno operátoru: UPDATE

Reprezentuje: Aktualizace dat v řádcích tabulky.

Tento operátor představuje nezbytnou operaci. Chcete-li zlepšit náklady přístupového plánu, zaměřte se na další operátory (například procházení a spojení) definující sadu řádků, které mají být aktualizovány.

Dodatek C. Koncepce DB2

Databáze

Relační databáze reprezentuje data jako kolekci tabulek. Tabulka sestává z definované sady sloupců a libovolného počtu řádků. Data v každé tabulce spolu logicky souvisejí a mezi tabulkami mohou být definovány relace. Data lze zobrazovat a manipulovat s nimi na základě matematických principů a operací označovaných výrazem relace (například INSERT, SELECT a UPDATE).

Databáze je samovysvětlující v tom smyslu, že kromě dat obsahuje popis vlastní struktury. Zahrnuje sadu systémových tabulek katalogů popisujících logickou a fyzickou strukturu dat, konfigurační soubor obsahující hodnoty parametrů asociované s databází a žurnál zotavení, ve kterém jsou zaznamenány probíhající transakce a transakce, které lze archivovat.

Databáze mohou být lokální nebo vzdálené. Lokální databáze je fyzicky umístěna na používané pracovní stanici, zatímco databáze umístěná v jiném počítači je považována za vzdálenou.

Schémata

Schéma je jedinečný identifikátor používaný pro seskupení sady databázových objektů (například tabulek, pohledů, indexů a aliasů). To znamená, že pokud vytváříte tabulku PAYROLL, je velmi únavné prohledávat databáze a zjišťovat, zda již jiný uživatel nevytvořil tabulku se stejným jménem. Jméno každého objektu musí být jedinečné pouze v rámci příslušného schématu.

Jména většiny databázových objektů mají dvě části: první část je tvořena jménem schématu, druhou částí je jméno objektu. Při vytváření můžete objekt přiřadit ke specifickému schématu. Pokud schéma neurčíte, bude objekt přiřazen k výchozímu schématu, kterým je obvykle jméno uživatele, který objekt vytvořil. Uživatel Smith například mohl vytvořit tabulku se jménem SMITH.PAYROLL.

Schéma se také stává objektem v databázi. Je vytvořeno při vytvoření prvního objektu ve schématu. Vlastníkem schématu může být jednotlivce. Vlastník může řídit přístup k datům a objektům v rámci schématu.

Tabulky

Relační databáze reprezentuje data jako kolekci tabulek. Tabulka sestává z dat, která jsou logicky uspořádána do sloupců a řádků (obecně označovaných výrazem záznamy).

Každá tabulka má jméno a každý sloupec v tabulce má jméno. V případě řádků tabulky není udržováno žádné konkrétní řazení, lze je však načítat v pořadí určeném hodnotami ve sloupcích. Data v tabulce spolu logicky souvisí. Veškerá data databází a tabulek jsou přiřazena k tabulkovým prostorům.

Dodatek D. Poznámky

Společnost IBM nemusí produkty, služby nebo funkce uvedené v tomto dokumentu nabízet ve všech zemích. Informace o produktech a službách, které jsou ve vaší oblasti aktuálně dostupné, získáte od místního zástupce společnosti IBM. Odkazy na produkty, programy nebo služby společnosti IBM v této publikaci nejsou míněny jako vyjádření nutnosti použití pouze uvedených produktů, programů či služeb společnosti IBM. Místo produktu, programu nebo služby společnosti IBM lze použít libovolný funkčně ekvivalentní produkt, program nebo službu, která neporušuje intelektuální vlastnická práva společnosti IBM. Ověření funkčnosti produktu, programu nebo služby pocházející od jiného výrobce je však povinností uživatele.

K jednotlivým subjektům popisovaným v tomto dokumentu se mohou vztahovat patenty nebo nevyřízené patentové přihlášky společnosti IBM. Vlastnictví tohoto dokumentu uživateli neposkytuje žádná licenční práva k těmto patentům. Dotazy týkající se licencí můžete posílat písemně na adresu:

IBM Director of
Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Odpovědi na dotazy týkající se licencí pro dvoubajtové znakové sady (DBCS) získáte od oddělení IBM Intellectual Property Department ve vaší zemi, nebo tyto dotazy můžete zasílat písemně na adresu:

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

Následující odstavec se netýká Spojeného království ani jiných zemí, ve kterých je takovéto vyjádření v rozporu s místními zákony: SPOLEČNOST INTERNATIONAL BUSINESS MACHINES CORPORATION TUTO PUBLIKACI POSKYTUJE "TAK, JAK JE" BEZ JAKÉKOLI ZÁRUKY, AŽ UŽ PŘÍMÉ ČI ODVOZENÉ, VČETNĚ, ALE NE VÝHRADNĚ, ODVOZENÝCH ZÁRUK TÝKAJÍCÍCH SE PORUŠOVÁNÍ ZÁKONŮ, PRODEJNOSTI ČI VHODNOSTI K URČITÉMU ÚČELU. V některých státech nejsou prohlášení týkající se přímých či odvozených záruk v určitých případech dovolena, a proto se vás toto prohlášení nemusí týkat.

Uvedené údaje mohou obsahovat technické nepřesnosti nebo typografické chyby. Údaje zde uvedené jsou pravidelně upravovány a tyto změny budou zahrnuty v nových vydáních této publikace. Společnost IBM může kdykoli bez upozornění provádět vylepšení nebo změny v produktech či programech popsáných v této publikaci.

Veškeré uvedené odkazy na stránky WWW, které nespravuje společnost IBM, jsou uváděny pouze pro referenci a v žádném případě neslouží jako záruka funkčnosti těchto stránek. Materiály uvedené na těchto stránkách WWW nejsou součástí materiálů pro tento produkt IBM a použití uvedených stránek je pouze na vlastní nebezpečí.

Společnost IBM může použít nebo distribuovat jakékoli informace, které jí sdělíte, libovolným způsobem, který společnost považuje za odpovídající, bez vyžádání vašeho svolení.

Vlastníci licence k tomuto programu, kteří chtějí získat informace o možnostech (i) výměny informací s nezávisle vytvořenými programy a jinými programy (včetně tohoto) a (ii) oboustranného využití vyměňovaných informací, mohou kontaktovat informační středisko na adrese:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Poskytnutí takových informací může být podmíněno dodržením určitých podmínek a požadavků zahrnujících v některých případech uhrazení stanoveného poplatku.

Licencovaný program popsáný v tomto dokumentu a veškerý licencovaný materiál k němu dostupný jsou společností IBM poskytovány na základě podmínek uvedených ve smlouvách IBM Customer Agreement, IBM International Program License Agreement nebo v jiné ekvivalentní smlouvě.

Jakékoli údaje o výkonnosti obsažené v této publikaci byly zjištěny v řízeném prostředí. Výsledky získané v jakémkoli jiném operačním prostředí se proto mohou výrazně lišit. Některá měření mohla být prováděna na vývojových verzích systémů a není zaručeno, že tato měření budou stejná i na běžně dostupných systémech. Některé údaje mohly být navíc zjištěny pomocí extrapolace. Skutečné výsledky mohou být jiné. Čtenáři tohoto dokumentu by měli zjistit použitelné údaje pro své specifické prostředí.

Informace týkající se produktů jiných výrobců pocházejí od dodavatelů těchto produktů, z jejich veřejných oznámení nebo z jiných veřejně dostupných zdrojů. Společnost IBM tyto produkty netestovala a nemůže potvrdit jejich správnou výkonnost, kompatibilitu ani žádné jiné výroky týkající se produktů jiných výrobců než IBM. Otázky týkající se kompatibility produktů jiných výrobců by měly být směřovány dodavatelům těchto produktů.

Veškerá tvrzení týkající se budoucího směru vývoje nebo záměrů společnosti IBM se mohou bez upozornění změnit nebo mohou být zrušena a reprezentují pouze cíle a plány společnosti.

Tyto údaje mohou obsahovat příklady dat a sestav používaných v běžných obchodních operacích. Aby byla představa úplná, používají se v příkladech jména osob, společností, značek a produktů. Všechna tato jména jsou fiktivní a jejich podobnost se jmény a adresami používanými ve skutečnosti je zcela náhodná.

LICENČNÍ INFORMACE:

Tyto informace mohou obsahovat ukázkové aplikační programy ve zdrojovém jazyce ilustrující programovací techniky na různých operačních platformách. Tyto ukázkové programy můžete bez závazků vůči společnosti IBM jakýmkoli způsobem kopírovat, měnit a distribuovat za účelem vývoje, používání, odbytu či distribuce aplikačních programů odpovídajících rozhraní API pro operační platformu, pro kterou byly ukázkové programy napsány. Tyto příklady nebyly plně testovány za všech podmínek. Společnost IBM proto nemůže zaručit spolehlivost, upotřebitelnost nebo funkčnost těchto programů.

Každá kopie nebo část těchto ukázkových programů nebo jakákoli práce z nich odvozená musí obsahovat následující copyrightovou doložku:

© *(název vaší společnosti) (rok)*. Části tohoto kódu jsou odvozeny z ukázkových programů společnosti IBM. © Copyright IBM Corp. *_zadejte rok nebo roky_*. Všechna práva vyhrazena.

Ochranné známky

Následující termíny jsou ochrannými známkami společnosti International Business Machines Corporation v USA anebo dalších zemích, a byly použity nejméně v jednom dokumentu z knihovny s dokumentací 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 Query Patroller	System/390
DB2 Universal Database	SystemView
Distributed Relational Database Architecture	Tivoli
DRDA	VisualAge
eServer	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
IBM	WebSphere
IMS	WIN-OS/2
IMS/ESA	z/OS
iSeries	zSeries

Následující termíny jsou ochrannými známkami nebo registrovanými ochrannými známkami jiných společností, a byly použity nejméně v jednom dokumentu z knihovny s dokumentací DB2 UDB:

Microsoft, Windows, Windows NT a logo Windows jsou ochranné známky společnosti Microsoft Corporation v USA anebo dalších zemích.

Intel a Pentium jsou ochranné známky společnosti Intel Corporation v USA anebo dalších zemích.

Java a všechny ochranné známky založené na termínu Java jsou ochrannými známkami společnosti Sun Microsystems, Inc. v USA anebo dalších zemích.

UNIX je registrovaná ochranná známka společnosti The Open Group v USA nebo dalších zemích.

Jména dalších společností, produktů nebo služeb mohou být ochrannými známkami nebo značkami služeb ostatních společností.

Rejstřík

B

- balíky
 - definice 62
- blokování kurzoru
 - definice 48
- blokování řádku
 - blokování kurzoru 48

C

- CMPEXP, operátor
 - definice 51, 69

D

- DELETE, operátor
 - definice 51, 69
- dotaz bez indexů a statistiky 12
- dynamické příkazy SQL
 - definice 48
- dynamické příkazy SQL, vytváření snímků Explain 3

E

- EISCAN, operátor
 - definice 52, 69
- EXPLAIN.DDL, soubor/příkaz 1
- EXPLSNAP, volba (v příkazu BIND) 4

F

- FETCH, operátor
 - definice 52, 70
- FILTER, operátor
 - definice 53, 70
- funkce, získání statistiky 8

G

- GENROW, funkce
 - definice 53, 70
- graf přístupového plánu, čtení symbolů 5
- graf přístupového plánu, podrobnosti o objektech 7
- graf přístupového plánu, změna vzhledu 9
- graf přístupového plánu, zobrazení a použití 5
- grafy přístupových plánů
 - seznam použitých operátorů 50
 - uzly
 - definice 46

- grafy přístupových plánů (*pokračování*)
 - vytvoření
 - definice 45
- GRPBY, operátor
 - definice 53, 71

H

- HSJOIN, operátor
 - definice 54, 71
- hvězdicová spojení
 - definice 65

I

- indexy
 - dělení do klastrů
 - definice 47
- INSERT, operátor
 - definice 54, 72
- IXAND, operátor
 - definice 55, 72
- IXSCAN, operátor
 - definice 55, 73

K

- konfigurační parametry, získání informací 9
- kontejnery
 - definice 47

L

- LIST TABLES, příkaz 1

M

- MSJOIN, operátor
 - definice 56, 73

N

- náklady
 - definice 47
- NLJOIN, operátor
 - definice 57, 74

O

- operandy
 - definice 50
- operátory
 - definice 50
 - seznam 50
- operátory, získání podrobností 8

- optimalizátor
 - definice 62

P

- PIPE, operátor
 - definice 57, 75
- posuvný ovladač zvětšení, pro zvětšení grafů přístupových plánů 6
- predikáty
 - definice 62
- příkaz VESAMPL.DDL 1
- příkazy, EXPLAIN.DDL 1
- příkazy, LIST TABLES 1
- příkazy, SET CURRENT EXPLAIN SNAPSHOT 3
- příkazy, VESAMPL.DDL 1
- příkazy, volba EXPLSNAP v příkazu BIND 4
- přístupové plány
 - definice 45
- přístupový plán, zlepšení 11, 29

R

- RETURN, operátor
 - definice 57, 75
- RIDSCN, operátor
 - definice 58, 75
- RPD, operátor
 - definice 58, 76

S

- selektivita predikátů
 - definice 64
- SET CURRENT EXPLAIN SNAPSHOT, příkaz 3
- SHIP, operátor
 - definice 58, 76
- shromáždění aktuální statistiky pro tabulky a indexy 15, 32
- sloupce v příkazech SQL, získání statistiky 9
- snímky Explain
 - definice 49
- snímky Explain pro dynamické příkazy SQL, vytváření 3
- snímky Explain pro statické příkazy SQL, vytváření 4
- snímky Explain, ukázky pro modul Vizualní vysvětlení 1
- snímky Explain, vytváření 1

snímky, ukázky pro modul Vizualní
vysvětlení 1
SORT, operátor
definice 59, 76
soubory, EXPLAIN.DDL 1
spuštění dotazu bez indexů
a statistiky 30
statické příkazy SQL
definice 66
statické příkazy SQL, vytváření snímků
Explain 4
statistika, pro tabulky, indexy, tabulkové
funkce 7

T

tabulkové prostory
definice 66
DMS
definice 48
tabulkové prostory DMS
definice 48
tabulkové prostory spravované systémem
definice 66
tabulkové prostory, získání statistiky 8
tabulky Explain, vytváření 1
TBSCAN, operátor
definice 60, 77
TEMP, operátor
definice 61, 78
TQUEUE, operátor
definice 61, 78
třídy optimalizace dotazů
definice 63

U

UNION, operátor
definice 61, 79
UNIQUE, operátor
definice 62, 79
UPDATE, operátor
definice 62, 79
uživatelské funkce, získání statistiky 8

V

vestavěné funkce, získání statistiky 8
Vizualní vysvětlení
popis 67
volby vazby, získání informací 9
vysvětlené příkazy
definice 50
vysvětlené příkazy SQL, výběr 5
vysvětlitelné příkazy
definice 49
vytvoření dalších indexů pro sloupce
tabulky 24, 40

Kontaktování společnosti IBM

V USA se můžete obrátit na zástupce společnosti IBM prostřednictvím následujících telefonních čísel:

- 1-800-IBM-SERV (1-800-426-7378) - služby zákazníkům
- 1-888-426-4343 - informace o možnostech dostupných služeb
- 1-800-IBM-4YOU (426-4968) - marketing a prodej produktů DB2

V Kanadě se můžete obrátit na zástupce společnosti IBM prostřednictvím následujících telefonních čísel:

- 1-800-IBM-SERV (1-800-426-7378) - služby zákazníkům
- 1-800-465-9600 - informace o možnostech dostupných služeb
- 1-800-IBM-4YOU (1-800-426-4968) - marketing a prodej produktů DB2

Pokud hledáte zastoupení společnosti IBM ve vašem státě nebo regionu, pomůže vám webová stránka IBM's Directory of Worldwide Contacts na adrese www.ibm.com/planetwide

Informace o produktu

Informace o produktech řady DB2 Universal Database jsou k dispozici telefonicky nebo na webové adrese www.ibm.com/software/data/db2/udb

Tento server obsahuje nejnovější informace o technické knihovně, objednávání příruček, stahování klientů, diskusních skupinách, sadách FixPak, novinkách a odkazech na webové zdroje informací.

Jestliže žijete v USA, telefonujte na jedno z následujících čísel:

- 1-800-IBM-CALL (1-800-426-2255) - objednávání produktů a obecné informace
- 1-800-879-2755 - objednávání příruček

Informace o možnostech komunikace se společností IBM mimo území USA najdete na webových stránkách IBM Worldwide na adrese www.ibm.com/planetwide.



Vytištěno v Dánsku společností IBM Danmark A/S.