

**Univerzita Pardubice**  
**Fakulta ekonomicko-správní**

**Hinty Oracle a jejich vliv na výkon dotazu**

**Bc. Josef Pírk**

**Diplomová práce**  
**2010**

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Josef PIRKL**  
Osobní číslo: **E08437**  
Studijní program: **N6209 Systémové inženýrství a informatika**  
Studijní obor: **Informatika ve veřejné správě**  
Název tématu: **Hinty Oracle a jejich vliv na výkon dotazu**  
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

Práce se zabývá vlivem použití hintů (sql hints) optimalizátoru Oracle na výkon dotazu. Testuje srovnání výkonu klasických SQL dotazů oproti SQL dotazům, které používají Oracle hinty. Je rovněž zkoumáno, jakým způsobem je při použití hintu ovlivněna výkonnost přítomností indexů, jak se hinty chovají při spojování více tabulek v dotazu a jak hinty ovlivňuje navolený režim optimalizátoru Oracle. Základním kritériem pro posuzování výkonnosti je CPU trvání zpracování dotazu (CPU time).

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

**ALAPATI, Sam.** *Expert Oracle Database 10g Administration.* Berkeley: APress, 2005. 1305 s. ISBN 1-59059-451-7.

**MELOUN, Milan, MILITKÝ, Jan.** *Statistické zpracování experimentálních dat.* Praha: Plus, s.r.o., 1994. 840 s. ISBN 80-85297-56-6.

**MCCULLOUGH-DIETER, Carol** *Mistrovství v Oracle 8.* Praha: Cpress, 1999. 952 s. ISBN 80-7226-204-1.

Vedoucí diplomové práce:



**Ing. Jan Panuš, Ph.D.**

Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: **5. října 2009**

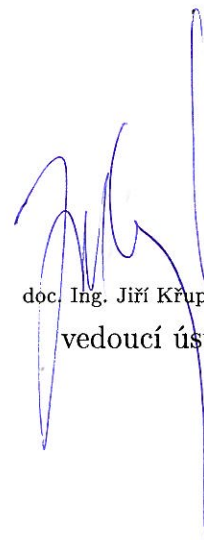
Termín odevzdání diplomové práce: **30. dubna 2010**



doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.



doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 5. října 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 25. 04. 2010

Josef Pirkl

Poděkování

Děkuji Ing. Janu Panušovi, PhD. za hodnotné rady a odborné vedení během mé práce.

## **SOUHRN**

Práce se zabývá vlivem použití hintů (sql hints) optimalizátoru Oracle na výkon dotazu. Testuje srovnání výkonu klasických SQL dotazů oproti SQL dotazům, které používají Oracle hinty. Je rovněž zkoumáno, jakým způsobem je při použití hintu ovlivněna výkonnost přítomností indexů, jak se hinty chovají při spojování více tabulek v dotazu a jak hinty ovlivňuje navolený režim optimalizátoru Oracle. Základním kritériem pro posuzování výkonnosti je CPU trvání zpracování dotazu (CPU time).

## **KLÍČOVÁ SLOVA**

hint, sql hint, Oracle, optimalizátor Oracle, náklad dotazu, SQL, rychlost odezvy

## **TITLE**

Oracle hints and their influence on query execution

## **ABSTRACT**

This thesis examines the influence of the Oracle Optimizer hints (sql hints) usage on the query execution. It tests the comparison of the classic SQL queries execution towards SQL queries which use Oracle hints. It also examines how the efficiency is influenced by the presence of indices when using hints, how the hints behave when connecting several tables in a query and how the hints are influenced by the chosen Oracle optimizer mode. The CPU time is the basic criterion for the assessment of the efficiency.

## **KEYWORDS**

hint, sql hint, Oracle, query optimizer, cost, SQL, response time

# Obsah

<b>ÚVOD</b> .....	<b>6</b>
<b>1. DATABÁZE ORACLE</b> .....	<b>7</b>
1.1 Zpracování SQL dotazu .....	7
1.2 Činnost optimalizátoru Oracle .....	8
1.3 Módy optimalizátoru .....	8
1.4 Přístupové metody k tabulkám a indexům .....	9
1.4.1 Přístup přes celou tabulku .....	9
1.4.2 Přístup přes index .....	10
1.5 Metody spojování více tabulek .....	10
1.5.1 Hašované spojení .....	10
1.5.2 Vnořené spojení .....	11
1.5.3 Tříděné slučovací spojení .....	12
1.6 Statistiky .....	13
<b>2. HINTY ORACLE</b> .....	<b>15</b>
2.1 Důležité hinty a jejich popis .....	16
<b>3. TESTOVÁNÍ</b> .....	<b>19</b>
3.1 Změna přístupové metody - měření nákladů.....	19
3.1.1 Ovlivnění prováděcího plánu statistikami .....	19
3.1.2 Neexistence statistik nad tabulkou - změna index -> full.....	21
3.1.3 Existence statistik nad tabulkou - změna full -> index.....	25
3.1.4 Změna módu optimalizátoru pomocí hintu .....	29
3.2 Aplikace hintů na spojení více tabulek .....	32
3.2.1 Model 1:1 - výběr všech vět .....	33
3.2.2 Model 1:1 - výběr přes operátor LIKE .....	40
3.2.3 Model 1:N - výběr všech vět .....	47
3.2.4 Model 1:N - výběr přes operátor LIKE .....	54
<b>ZÁVĚR</b> .....	<b>62</b>

<b>SEZNAM POUŽITÝCH ZKRATEK.....</b>	<b>63</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>64</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>66</b>
<b>SEZNAM TABULEK.....</b>	<b>67</b>
<b>SEZNAM GRAFŮ .....</b>	<b>69</b>
<b>SEZNAM DOTAZŮ .....</b>	<b>71</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>73</b>



# Úvod

Hinty Oracle jsou speciální komentáře vložené do SQL dotazu, kterými je možné významnou měrou ovlivnit způsob zpracování dotazu. Optimalizátor při vytváření plánu dotazu nemusí z různých důvodů určit skutečně optimální plán pro právě vytvářený dotaz. V takovém případě lze pomocí hintu pozměnit prováděcí plán takovým způsobem, aby více odpovídal vývojářem očekávaným podmínkám. Značnou výhodou hintů je to, že s jejich pomocí dochází ke změně pouze na úrovni dotazu, kde došlo k jejich použití - u ostatních dotazů zůstává chování implicitní.

Tato práce je rozdělena do dvou částí. První část ukazuje teoretický základ použití hintu v Oracle, podobu činnosti optimalizátoru, metody spojování tabulek. Ukazuje způsob použití hintů a jejich rozdělení do skupin společně s popisem nejdůležitějších dokumentovaných hintů.

V druhé části dochází k testování, při kterém jsou porovnávány dotazy většinou provedené optimalizátorem implicitní cestou proti dotazům, kde byl uplatněn hint Oracle. Kritéria, dle kterých je hodnoceno, jsou náklady dotazu, jak je stanoví prováděcí plán dotazu a CPU trvání dotazu.

Je třeba říci, že způsob, jakým hint ovlivní databázový dotaz, může být závislý na mnoha faktorech: hint by měl být použitý pouze v tom případě, kdy vývojář přesně ví, jaká data může na daném místě očekávat. Stejně jako lze v případě správného použití od hintu očekávat zlepšení výkonu SQL dotazu; v případě neuvážené aplikace může být výsledek o mnoho horší oproti navrženému implicitnímu plánu.

Cílem práce je popsat základní strukturu a použití hintů Oracle a jejich roli jak ovlivňují činnost optimalizátoru. V rámci testování je pomocí technik explorační analýzy dat sledováno, zda optimalizátor za různých podmínek implicitně navrhuje prováděcí plány v dané situaci optimálně, či lze-li podobu dotazu upravit pomocí hintů tak, aby v rámci sledovaných kritérií vracely SQL dotazy výhodnější výsledky.

# 1. Databáze Oracle

Oracle je jedním z nejpokročilejších databázových systémů s podporou objektově-relačního uložení dat (ORDBMS). Návrh relačního modelu pochází z roku 1970, kdy jeho podobu publikoval E.F.Codd v článku *A Relational Model of Data for Large Shared Data Banks* uveřejněným v časopise *Communication of the ACM* [1].

Pro vlastní přístup a práci s daty používá Oracle jazyk SQL (*Structured Query Language*), nástupce jazyka SEQUEL (*Structured English Query Language*), který byl vyvinut firmou IBM dle Coddova relačního návrhu. SQL byl poprvé komerčně použit v roce 1979 v Oracle verze 2 [1].

V okamžiku, kdy uživatel provede SQL dotaz, který vrací specifickou množinu dat, je provedeno několik kroků zpracování dotazu (angl. query processing). Jedná se o transformaci SQL dotazu do podoby interního prováděcího plánu, jehož výsledkem jsou požadovaná data [1]. Při zpracování dotazu je do činnosti zapojen optimalizátor databáze, který je zodpovědný za určení prováděcího plánu pro přístup k datům.

Činnost optimalizátoru Oracle je popsána v následující části práce.

## 1.1 Zpracování SQL dotazu

Úkolem optimalizátoru Oracle je optimalizace SQL dotazu tak, aby cílová výstupní množina dat byla uživatelem získána co možná nejdříve [12].

V závislosti na zvoleném módu optimalizátoru Oracle jsou za tímto účelem využívány tyto metody [12]:

- statistiky,
- histogramy,
- hinty (uživatelsky).

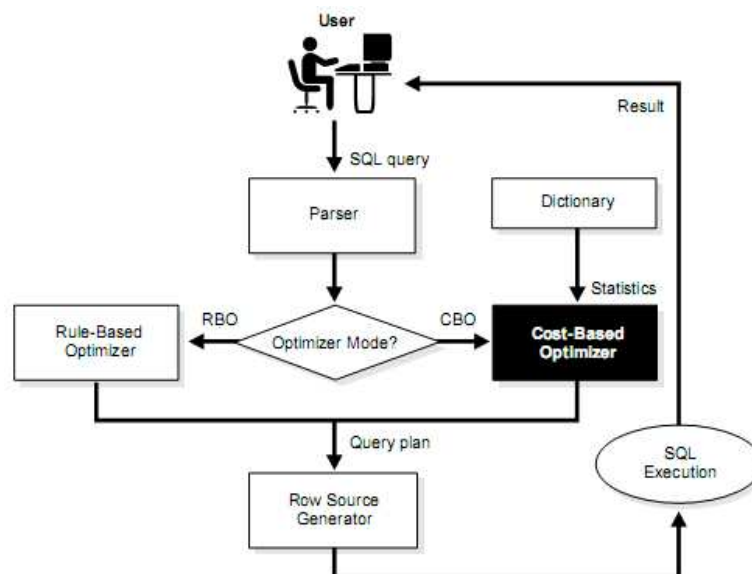
Dřívější verze Oracle (před 10g) poskytovaly pravidlový mód optimalizátoru (angl. rule-based, RBO). Ve verzi 10g sice existuje možnost, jak se k tomuto módu vrátit, přesto jsou nynější módy optimalizátoru postaveny na bázi nákladového určení (angl. cost-based, CBO), při němž je hledán prováděcí plán dotazu s nejmenšími náklady.

Vlastní zpracování SQL dotazu Oracle popisuje obrázek 1. Po zadání dotazu dochází v rámci *parseru* k jeho syntaktické a sémantické kontrole. Optimalizátor od verze Oracle 10g volí vždy metody postavené na nákladech<sup>1</sup>, k čemuž mu mohou být velmi

---

<sup>1</sup> Pravidlový RBO mód optimalizátoru si lze vynutit pomocí hintu RULE.

užitečné existující aktuální statistiky o objektech. Na základě existujících informací je sestaven prováděcí plán dotazu, který reprezentuje stromová struktura popisující jednotlivé fáze, kde jsou definovány přístupové metody k tabulkám a metody i pořadí jejich spojování. Pro provedení dotazu je následně vrácena výsledná množina dat.



Obrázek 1: Oracle zpracování SQL dotazu - převzato z [2]

## 1.2 Činnost optimalizátoru Oracle

Výstupem z činnosti optimalizátoru je *prováděcí plán* (angl. execution plan), který mapuje cestu k datům. K tomuto účelů optimalizátor navrhuje interně více cest, ze kterých pak volí podle daných kritérií tu nejvýhodnější [8].

Při existenci dotazu, který obsahuje jednotlivé výběrové podmínky v sekci WHERE, jsou tyto podmínky oceněny, přičemž se bere v úvahu přítomnost a využitelnost indexů, dostupnost statistik, které podávají bližší informace o uložených datech ve sloupcích tabulek. Optimalizátor pak navrhuje takový prováděcí plán, v němž je na prvním místě podmínka s nejnižší vyhodnocenou cenou [8].

## 1.3 Módy optimalizátoru

Nastavení módu optimalizátoru určuje Oracle parametr OPTIMIZER\_MODE. Dostupné možnosti (10g) jsou následující:

### ALL\_ROWS

Při tomto módu optimalizátoru dochází k optimalizaci výkonnosti dotazu. Přístup je založený na nákladech dotazu [4]. Tento přístup je používán zejména u OLAP systémů, kde jsou data zpracovávána procedurálně v dávkách a není nutné mít co

nejrychleji navracený první řádek. Optimalizace výstupu dotazu je v tomto případě taková, aby získání posledního řádku dotazu bylo co nejrychlejší [11].

### **FIRST\_ROWS\_n**

Takto nastavený mód optimalizátoru využívá statistik a nákladů tak, aby došlo optimalizování navracení prvních  $n$  výstupních řádek dotazu [4]. Tento způsob optimalizace je využitelný v interaktivních prostředích OLTP systémů, kde uživatel prochází vybranými řádky - důležité je v tomto případě co nejrychlejší zobrazení první řádky [11].

### **RULE**

Mód optimalizátoru je nastaven pomocí pravidel - nevyužívá se odhad nákladů dotazu. Tento mód je zachován pouze z důvodu zpětné kompatibility.

Mód optimalizátoru je možné změnit pro každou spuštěnou instanci Oracle, vedle toho však existuje možnost změnit mód optimalizátoru u každého jednotlivého dotazu pomocí příslušného hintu [4].

## **1.4 Přístupové metody k tabulkám a indexům**

Optimalizátor Oracle volí při přístupu k tabulkám z těchto různých podporovaných způsobů:

- Přístup přes celou tabulku (angl. full table scan),
- přístup přes index (angl. index scan),
- další varianty přístupu (angl. hash scan, cluster scan, sample table scan).

Implicitní přístupové metody, které zvolí optimalizátor, mohou být následně s použitím hintů změněny.

### **1.4.1 Přístup přes celou tabulku**

Při hledání odpovídajících řádků, které vyhovují SQL dotazu, dochází při této variantě k průchodu přes všechny řádky tabulky. Tento způsob může být zvolen i tehdy, usoudí-li optimalizátor, že není zapotřebí využít index (existuje-li). Uvádí se, že optimalizátor použije *přístup přes celou tabulku*, pokud je předpoklad, že ve výsledném dotazu bude alespoň 5-25% řádků z tabulky [12]. V takovém případě nemusí dojít k použití indexové založeného přístupu a může být použit *přístup přes celou tabulku*.

*Přístup přes celou tabulku* může optimalizátor zvolit i v tom případě, pokud se mu podaří načíst většinu bloků tabulky na jedno fyzické načtení. Počet načítaných bloků

určuje parametr databáze Oracle `DB_FILE_MULTIBLOCK_READ_COUNT`. Velikost samotného bloku pak udává parametr `DB_BLOCK_SIZE` [12].

### 1.4.2 Přístup přes index

Při tomto přístupu k tabulce je využitý již existující index, kdy dochází k vyhledání hodnoty v rámci indexu a následně k získání ROWID<sup>2</sup>, který určuje jednoznačné umístění hledané informace v tabulce [12]. K využití tohoto přístupu se optimalizátor implicitně rozhodne tehdy, jedná-li se o čtení výběrových částí tabulky. Pokud je předpoklad, že výsledku bude vyhovovat více řádků, vzniklá režie spojená s načítáním indexu může způsobit, že optimalizátor raději volí použití *přístupu přes celou tabulku*.

## 1.5 Metody spojování více tabulek

Oracle poskytuje následující metody pro spojování více tabulek:

- Hašované spojení (angl. hash join),
- vnořené spojení (angl. nested loops join),
- tříděné slučovací spojení (angl. sort merge join),
- další metody, například pro kartézská spojení.

Způsob spojování tabulek je možné výrazně ovlivnit k tomu určenými příslušnými hinty.

### 1.5.1 Hašované spojení

Hašované spojení je využíváno pro spojení rozsáhlých celků dat. K aplikaci tohoto spojení pomáhá *hash* hodnota [4]. Pro procházení tabulek je použitý *přístup přes celou tabulku* [2].

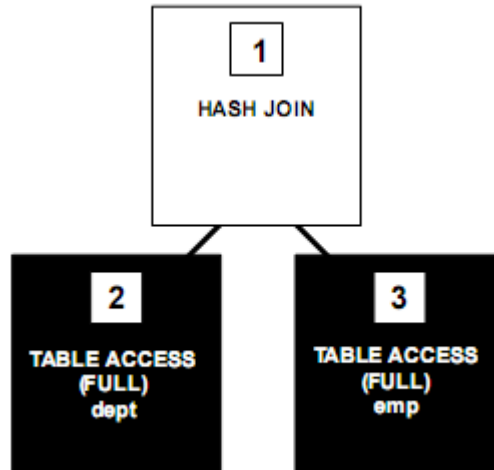
Jednotlivé kroky provedené při uskutečnění hašovaného spojení :

1. V paměti (či při nedostatku místa na disku) je vytvořena tzv. *hash tabulka*, a to na základě menší z obou tabulek spojení. Pro každý řádek je vypočítán hash klíč dle spojovacího klíče [8].
2. Pro každý řádek ve druhé (rozsáhlejší) tabulce dochází rovněž k výpočtu hash hodnoty dle spojovacího klíče - v případě, že odpovídá hodnotě v *hash tabulce*, je dvojice navracena [8].

Graficky zobrazuje hašované spojení obrázek 2.

---

<sup>2</sup> ROWID je interní logický ukazatel na řádek v tabulce [12]. Může se však změnit při reorganizaci tabulky, například při exportu/importu.



Obrázek 2: Jednotlivé kroky hašovaného spojení - převzato z [2]

Efektivitu hašovaného spojení lze ovlivnit nastavením inicializačních parametrů Oracle. Parametr `HASH_AREA_SIZE` ovlivňuje velikost paměti používané pro hash operace [4].

Pro vynucení hašovaného spojení lze použít hint `USE_HASH`. Hašované spojení lze zakázat hintem `NO_USE_HASH` [7].

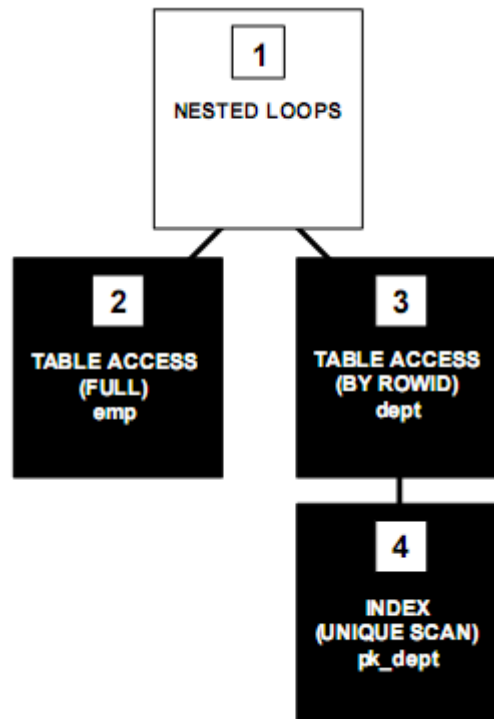
### 1.5.2 Vnořené spojení

Vnořené spojení je charakteristické tím, že pro každý řádek vnější (angl. outer) vyhledává odpovídající řádky v tabulce vnitřní (angl. inner). Metoda je vhodná pro uskutečnění spojení z menší na velmi velkou datovou množinu [4]; je rovněž velmi vhodné, aby v případě použití vnořného spojení byl nad vnitřní řízenou tabulkou definován vhodný index [8].

Jednotlivé kroky provedené při uskutečnění vnořného spojení [4]:

1. Optimalizátor zvolí vnější (řídící) tabulku.
2. Druhá tabulka je nazývána vnitřní (řízená).
3. Pro každý řádek vnější tabulky jsou vyhledány všechny odpovídající řádky v tabulce vnitřní (dle použité podmínky spojení).

Graficky zobrazuje vnořné spojení obrázek 3.



Obrázek 3: Jednotlivé kroky vnořeného spojení - převzato z [2]

Pro vynucení vnořeného spojení lze použít hint `USE_NL` nebo `USE_NL_WITH_INDEX`. Vnořené spojení lze zakázat hintem `NO_USE_NL`[7].

Typickou oblastí, kde může být vnořené spojení dobře využito, je model hlavička objednávky - řádky objednávky. Jde o typ spojení, kdy vnitřní tabulka obsahuje výrazně méně řádků, nežli tříděná tabulka vnější.

### 1.5.3 Tříděné slučovací spojení

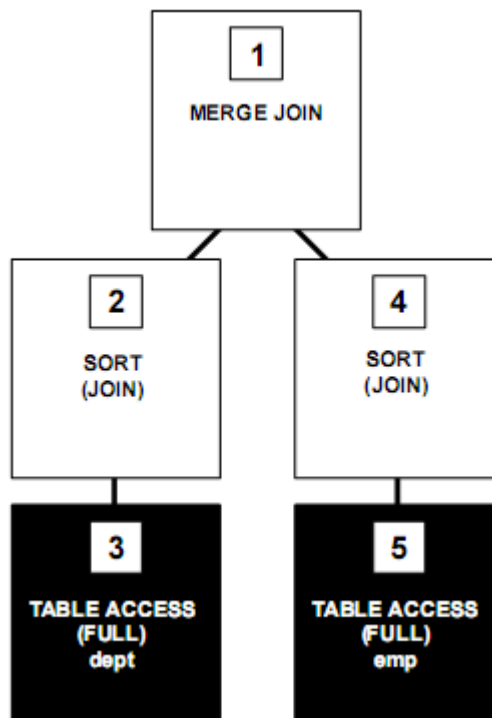
Slučovací spojení poskytuje v případě větších datových zdrojů z hlediska výkonnosti převážně lepší výsledky než vnořené spojení, ne však takové jako hašované spojení [8]. Výjimkou může být situace, kdy jsou datové zdroje již setříděny, čímž dochází k ušetření režie týkající se třídění dat (což je jeden z důležitých kroků při uskutečňování tohoto druhu spojení) [4].

Jednotlivé kroky provedené při uskutečnění tříděného slučovacího spojení [4]:

1. Oba datové zdroje jsou setříděné dle spojovacích polí.
2. S využitím takto setříděných zdrojů dochází ke spojení tabulek.

Výhoda oproti vnořenému spojení je v tom, že vnitřní tabulka není procházena tolikrát, kolik řádek má tabulka vnější, ale pouze jednou. Z tohoto důvodu vyplývá pak lepší efektivita při práci s většími objemy dat [8].

Graficky zobrazuje tříděné slučovací spojení obrázek 4.



Obrázek 4: Jednotlivé kroky tříděného slučovacího spojení - převzato z [2]

Pro vynucení tříděného slučovacího spojení lze použít hint `USE_MERGE`. Vnořené spojení lze zakázat hintem `NO_USE_MERGE` [7].

Tento druh spojení může být výhodný v případech, kdy existuje předpoklad, že obě množiny dat jsou setříděné - typickým případem jsou například chronologická data, která jsou jako setříděná již do tabulky vkládána.

## 1.6 Statistiky

Významnou součástí přístupů postavených na nákladech jsou *statistiky*. Statistiky jsou předdefinované informace o existujících objektech databáze. Tyto údaje jsou využity při volbě prováděcího plánu, dle kterého dochází k získání vlastní výstupní množiny dat ze zadaného SQL dotazu. Statistiky významným způsobem ovlivňují provedení dotazu právě při použití některého nákladového optimalizátoru [12].

Statistiky obsahují [4]:

- údaje o tabulkách  
(počet řádků v tabulce, počet bloků použitých tabulkou, průměrná délka řádku),
- údaje o sloupcích tabulek  
(počet jednoznačných hodnot ve sloupci, počet *null* hodnot ve sloupci, histogram hodnot sloupce),



- údaje o indexech  
(počet *leaf* bloků, úrovně),
- systémové statistiky  
(využití I/O operací, využití CPU).

Statistiky používá nákladově orientovaná optimalizace [12]. V případě, že statistiky existují, optimalizátor je schopen kvalitnějšího odhadu nákladů pro jednotlivé případně použité přístupové metody k tabulkám nebo při spojování tabulek.

V případě, že nejsou statistiky přítomné, optimalizátor Oracle může v některých situacích použít *defaultní statistické hodnoty*, a to například na základě alokovaného prostoru tabulky [2].

Výpočet statistiky nad tabulkou je možné vyvolat pomocí příkazu ANALYZE TABLE, jak ukazuje dotaz 1, přičemž varianta COMPUTE provádí výpočet statistik nad celou databázovou tabulkou, pro větší tabulky je možné užití varianty ESTIMATE, která provádí odhad.

```
analyze table ba.account_i1 compute statistics;
```

Dotaz 1: Výpočet a uložení statistiky nad tabulkou ba.account\_i1

Další možností pro práci se statistikami je využití balíčku (angl. package) DBMS\_STATS, který obsahuje rozpracovanější procedury a funkce pro podporu práce s nimi.

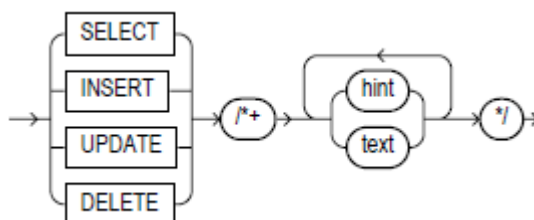
Ve verzi 10g je správa statistik již implicitně automatizovaná, podrobná nastavení lze administrátorsky provést v nástroji *Oracle Enterprise Manager 10g* v sekci *Administration\Statistics Management* [3].

## 2. Hinty Oracle

Hint Oracle je speciální forma komentáře, pomocí kterého lze explicitně ovlivnit činnost optimalizátoru a způsob zpracování zadaného SQL dotazu. Oracle disponuje ve své verzi 10g zhruba 60 různými dokumentovanými hinty, kterými lze ovlivnit činnosti optimalizátoru. Vedle těchto hintů existuje i skupina hintů, které nejsou firmou Oracle oficiálně dokumentované. Hinty lze rozčlenit do následujících funkčních kategorií:

- hinty pro určení cíle optimalizace,
- hinty pro volbu přístupové metody,
- hinty pro pořadí spojování tabulek,
- hinty pro operace spojení,
- hinty pro podporu paralelního provádění,
- hinty pro transformaci dotazu,
- jiné hinty [7].

Hinty musí být uvedeny hned za prvním klíčovým slovem příkazu (nejčastěji SELECT) [7]. Uzavírají mezi znaky `/*+` a `*/`, případně je možné využít počáteční znaky `/*--`, což pak ale předpokládá, že pro hint bude věnován celý řádek. Obecnou syntaxi použití hintů ukazuje obrázek 5.



Obrázek 5: Syntaxe použití hintů v Oracle - převzato z [2]

Praktické použití hintu v dotazu, kdy je zakázáno využívání jakýchkoli indexů nad tabulkou, ukazuje dotaz 2.

```
select /*+ NO_INDEX(a) */ * from ba.account_i a
where
a.syn = '611'
```

Dotaz 2: Ukázka použití hintů v dotazu - zákaz využívání indexů

V případě, že je v rámci hintu použita špatná syntaxe, Oracle to nedá žádným způsobem najevo [7]. Optimalizátor navíc může i správně napsaný hint ignorovat<sup>3</sup>. To

---

<sup>3</sup> Během práce s hinty nebylo něco takového sledováno.

se může stát v situacích, kdy je například hintem specifikovaný index v rozporu s použitou spojovací metodou [2].

## 2.1 Důležité hinty a jejich popis

### ALL\_ROWS

Provede přepnutí optimalizátoru (parametr OPTIMIZER\_MODE) do režimu ALL\_ROWS pro nákladově nejvýkonnější vrácení celého výsledku dotazu [7].

### APPEND

Pro příkaz INSERT nastavuje tento hint tzv. APPEND mód, kdy nová data jsou přidána do tabulky, nejsou využity případné existující volné bloky alokované v rámci tabulky [7].

### DRIVING\_SITE

V případě distribuovaného zpracování dochází ke zpracování dotazu na specifikovaném umístění, kde se nachází tabulka, uvedená v parametru tohoto hintu. Řádky případných dalších tabulek dotazu jsou přesunuty ke zpracování do tohoto umístění [7].

### FIRST\_ROWS

Provede přepnutí optimalizátoru (parametr OPTIMIZER\_MODE) do režimu FIRST\_ROWS pro co nejrychlejší zobrazení prvních  $n$  řádků ( $n$  je parametrem tohoto hintu). Tento hint je ignorován v situacích, kdy je nutné pro vykonání dotazu načíst všechny řádky – příkladem může použití GROUP BY klauzule nebo operátoru DISTINCT [7].

### FULL

Tento hint umožňuje přikázat optimalizátoru, aby z tabulky četl formou *přístupu přes celou tabulku* [7]. Při této variantě dochází při získávání dat k průchodu přes celou tabulku, případné výběrové podmínky se pak testují v rámci každého řádku.

Značnou nevýhodou *přístupu přes celou tabulku* může být v případě Oracle to, že pokud tabulka obsahuje značné množství smazaných vět, které byly smazané přes příkaz DELETE TABLE, dochází i k zahrnutí těchto bloků do procházení řádků<sup>4</sup> [12].

### INDEX

Použití tohoto hintu říká optimalizátoru, aby při přístupu k datům tabulky použil v rámci *přístupu přes index* (angl. index scan) daný konkrétní index, který je uveden jako parametr tohoto hintu [7]. V případě uvedení více indexů volí optimalizátor nákladově nejvýhodnější z nich. Stejným způsobem je postupováno, pokud v rámci tohoto hintu není uvedený žádný konkrétní index [7].

---

<sup>4</sup> Řešením tohoto problému je smazání vět přes TRUNCATE TABLE.

## **LEADING**

Tento hint umožňuje explicitně určit pořadí spojování tabulek, skládá-li se dotaz z více tabulek [7].

## **NOAPPEND**

Ruší tzv. APPEND mód u příkazu INSERT [7].

## **NO\_EXPAND**

Inverzní varianta k hintu USE\_CONCAT, která zabraňuje využití zřetězení OR podmínek [7]. Viz USE\_CONCAT.

## **NO\_USE\_HASH<sup>5</sup>**

Pro vytvoření spojení mezi tabulkami nebude použito hašovaného spojení [7].

## **NO\_USE\_MERGE**

Pro vytvoření spojení mezi tabulkami nebude použito tříděné slučovací spojení [7].

## **NO\_USE\_NL**

Pro vytvoření spojení mezi tabulkami nebude použito vnořené spojení [7].

## **NOINDEX**

Hint zakazuje použití konkrétního či jakéhokoli indexu nad tabulkou (tabulka je určena parametrem, zakázané indexy lze volitelně vyjmenovat, nejsou-li uvedeny, jsou zakázány všechny indexy) [7].

## **NOPARALLEL**

Bezparametrický hint, který deaktivuje využití paralelního zpracování při přístupu k tabulce, je-li toto zpracování nastaveno nad tabulkou například přes příkaz ALTER TABLE nebo přímo v rámci vytvoření tabulky příkazem CREATE TABLE [7].

## **ORDERED**

Tabulky budou spojovány v tom pořadí, jak jsou uvedeny v sekci FROM [7].

## **PARALLEL**

Umožňuje přistupovat k tabulce pomocí paralelního zpracování za podpory více procesorů. Tímto způsobem je možné rozložit zpracování dotazu na více procesorů, přičemž jako parametr tohoto hintu lze nastavit *stupeň paralelního zpracování* (angl. *degree of parallelism*=počet procesorů [8]) [12]. Paralelní zpracování musí být verzí Oracle podporováno, ve verzi 10g je paralelní zpracování aktivní pouze ve variantě *EE (Enterprise Edition)*, přičemž musí být nastavený i inicializační parametr PARALLEL\_SERVER (více procesorů nebo více procesorových jader) [5].

## **RULE**

Hint přepíná optimalizátor do plavidlového režimu (*RBO*) [7]. Při provádění dotazu nedochází k využití nákladových informací ani statistik.

---

<sup>5</sup> Při pokusu o potlačení všech typů spojení pomocí hintu (NO\_USE\_HASH, NO\_USE\_MERGE, NO\_USE\_NL) mezi dvěma tabulkami, volil optimalizátor původní implicitní spojení.

## **USE\_CONCAT**

V případě OR podmínek v rámci dotazu umožňuje tento hint zpracování každé části podmínky jednotlivě v každém samostatném dotazu a následné spojení těchto výsledků do jedné datové množiny použitím operátoru UNION ALL. Tento proces se nazývá zřetězení (angl. concatenation) [7]. Hint USE\_CONCAT může přinést zlepšení výkonu dotazu v případě velmi rozsáhlých datových zdrojů [12].<sup>6</sup>

## **USE\_HASH**

K zajištění spojení mezi tabulkami bude použité hašované spojení [7].

## **USE\_MERGE**

K zajištění spojení mezi tabulkami bude použité tříděné slučovací spojení [7].

## **USE\_NL**

K zajištění spojení mezi tabulkami bude použité vnořené spojení [7].

---

<sup>6</sup> Při testování hintu USE\_CONCAT se nepodařilo vynutit využití zřetězení, přičemž inverzní varianta NO\_EXPAND, pokud optimalizátor sám zvolil použití zřetězení, pracovala.

## 3. Testování

Tato část je zaměřena na testování jednotlivých přístupových metod k tabulkám a na efekt, který způsobí aplikace hintu pro ovlivnění tohoto přístupu.

Sledovanými testovacími kritérii v této části jsou:

- náklad dotazu,
- CPU trvání dotazu.

Náklad dotazu (angl. cost) je analyzován dle navrácené hodnoty přes prováděcí plán dotazu (angl. explain plan). Náklad dotazu je hodnota vyjadřující odhad očekávaných zdrojů, které budou potřebné pro vykonání dotazu s každým jednotlivým plánem vykonávaného dotazu. Do této hodnoty jsou zahrnuty prováděné vstupně-výstupní operace, CPU cykly a operace s pamětí. Čím je hodnota nákladu dotazu vyšší, tím jsou vyšší i potřebné očekávané zdroje [4]. Pomocí kritéria nákladu dotazu lze relativně porovnávat mezi sebou dva dotazy, vracející stejnou výstupní množinu dat.

CPU trvání dotazu je sledovaným kritériem pro testování délky trvání provedení dotazu. Tento časový interval je v rámci tohoto testu vyjádřen ve vteřinách tzv. CPU času. Takto vyjádřený čas není ovlivněn případným čekáním na obsazené systémové prostředky. V případě vzniku existence velkého rozdílu mezi CPU časem a skutečným uplynulým (elapsed) časem, nasvědčuje toto o častém čekání na přidělení systémových zdrojů [6]. Pro měření CPU trvání dotazu je v rámci této práce používán vlastní SQL skript, který zaznamenává CPU trvání pro 1000 volání příslušného dotazu (příloha D).

### 3.1 Změna přístupové metody - měření nákladů

#### 3.1.1 Ovlivnění prováděcího plánu statistikami

Při neexistenci statistik nad tabulkou s 1 000 000 záznamů volí optimalizátor při dotazu, při kterém je prováděn dotaz nad sloupcem, nad kterým existuje index, *přístup přes index*, pokud tento dotaz vrací zhruba 52% všech řádků tabulky (dotaz 3, tabulka 1).

```
select * from ba.account_i1
where
value <= 520000
```

Dotaz 3: Požadavek na zobrazení 52 % procent dat z tabulky (bez statistik)

Tabulka 1: Prováděcí plán pro dotaz 3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		368K	25M	1502 (1)	00:00:19
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	368K	25M	1502 (1)	00:00:19
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	368K		1092 (1)	00:00:14

V případě, že je výsledkem takového dotazu zhruba 53 -100% ze všech řádků tabulky, optimalizátor rozhodne pro použití *přístupu přes celou tabulku* (dotaz 4, tabulka 2).

```
select * from ba.account_i1
where
value <= 530000
```

Dotaz 4: Požadavek na zobrazení 53 % procent dat z tabulky (bez statistik)

Tabulka 2: Prováděcí plán pro dotaz 4

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		385K	27M	1570 (4)	00:00:19
* 1	TABLE ACCESS FULL	ACCOUNT_I1	385K	27M	1570 (4)	00:00:19

Použitou zvolenou metodu přístupu k tabulce významně ovlivňuje existence statistik. V případě vytvoření počítaných statistik (angl. *compute statistics*) optimalizátor volí u tohoto dotazu *přístup přes index* v případě, že je požadováno zhruba do 15% řádků z celkového počtu řádků tabulky (dotaz 5, tabulka 3).

```
select * from ba.account_i1
where
value <= 150000
```

Dotaz 5: Požadavek na zobrazení 15 % procent dat z tabulky (statistiky)

Tabulka 3: Prováděcí plán pro dotaz 5

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		150K	7470K	1570 (1)	00:00:19
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	150K	7470K	1570 (1)	00:00:19
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	150K		320 (2)	00:00:04

*Přístup přes celou tabulku* je pak ve stejné situaci zvolen při požadavku na zobrazení 16-100% ze všech řádků z tabulky (dotaz 6, tabulka 4).

```
select * from ba.account_i1
where
value <= 160000
```

Dotaz 6: Požadavek na zobrazení 16 % procent dat z tabulky (statistiky)

Tabulka 4: Prováděcí plán pro dotaz 6

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		160K	7968K	1588 (5)	00:00:20
* 1	TABLE ACCESS FULL	ACCOUNT_I1	160K	7968K	1588 (5)	00:00:20

Optimalizátor tedy v případě přítomnosti validních statistik nad tabulkou volí *přístup přes index* zhruba do 15 % všech řádků tabulky, je-li požadavkem zobrazení více jak 15 %, je zvolen *přístup přes celou tabulku*.

### 3.1.2 Neexistence statistik nad tabulkou - změna index -> full

Následující měření nákladů sleduje situaci, kdy optimalizátor volí implicitně *přístup přes index* a pomocí hintu je následně vynucen *přístup přes celou tabulku*. Náklady dotazu jsou pak porovnány proti sobě. Měření probíhá v situaci, kdy nad tabulkou neexistují statistiky. Sledování efektu změny je nad tabulkou měřeno celkem třikrát, a to v situacích, kdy dotazu s podmínkou nad indexovaným polem vyhovuje 10%, 30%, 50% všech řádků tabulky.

#### Varianta - vyhovuje 10% všech řádků tabulky

Při tomto dotazu s výběrem nad sloupcem, který je součástí indexu, vyhovuje výběrové podmínce 10% ze všech řádků tabulky. Optimalizátor v tomto případě (statistiky neexistují) implicitně volí *přístup dle indexu* (dotaz 7, tabulka 5).

```
select * from ba.account_i1
where
value <= 100000
```

Dotaz 7: Požadavek na zobrazení 10 % procent dat z tabulky

Tabulka 5: Prováděcí plán pro dotaz 7

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		83606	6041K	342 (0)	00:00:05
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	83606	6041K	342 (0)	00:00:05
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	83606		249 (0)	00:00:03

Vynucený *přístup přes celou tabulku* je aplikován pomocí hintu FULL (dotaz 8). Získaný plán dotazu pak ukazuje tabulka 6.



```
select /*+ FULL(account_i1) */ * from ba.account_i1
where
value <= 100000
```

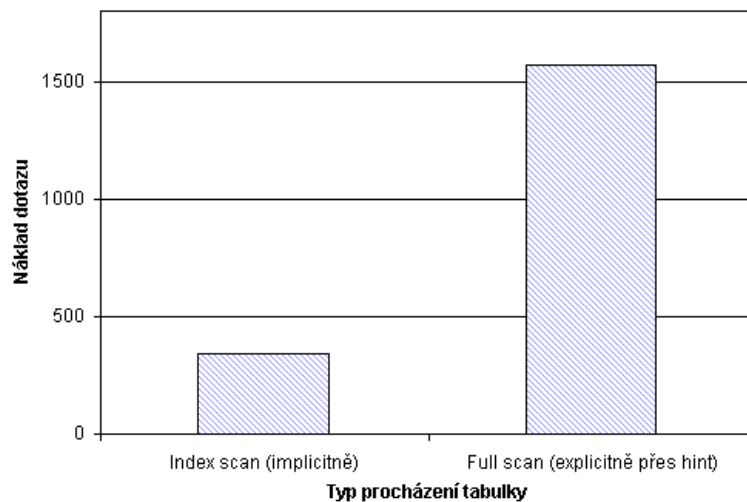
Dotaz 8: Požadavek na zobrazení 10 % procent dat z tabulky - aplikace hintu FULL

Tabulka 6: Prováděcí plán pro dotaz 8

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		83606	6041K	1570 (4)	00:00:19
* 1	TABLE ACCESS FULL	ACCOUNT_I1	83606	6041K	1570 (4)	00:00:19

Srovnáním obou plánů dotazu lze konstatovat, že vyžádání plného přístupu nad tabulkou, bez existence validních statistik, kde je vyhledáváno nad sloupcem, který je součástí indexu a vyhovuje 10% řádků tabulky, zvýšilo náklady dotazu z implicitní hodnoty 342 na 1570. Vynucená změna pomocí hintu FULL zvýšila tedy náklady dotazu více jak 4x.

Porovnání nákladů implicitní a explicitní varianty s hintem ukazuje graf 1.



Graf 1: Porovnání nákladů pro změnu index->full, 10%, bez statistik

### Varianta - vyhovuje 30% všech řádků tabulky

Stejným způsobem jako v předchozím případě je nyní testováno 30% vyhovujících řádků indexovaného vyhledávaného pole. Podmínky jsou jinak stejné jako u varianty 10%.

Optimalizátor volí implicitně *přístup přes index*, plán dotazu ukazuje tabulka 7.

Tabulka 7: Prováděcí plán, 30% vyhovujících řádků, bez statistik

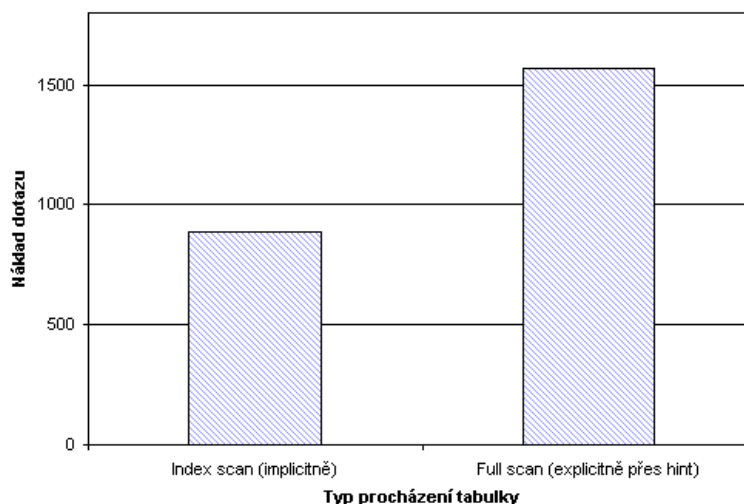
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		217K	15M	889 (1)	00:00:11
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	217K	15M	889 (1)	00:00:11
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	217K		647 (1)	00:00:08

Změnu plánu dotazu po vynucené změně na přístup přes celou tabulku pomocí hintu FULL ukazuje tabulka 8.

Tabulka 8: Prováděcí plán, 30% vyhovujících řádků, bez statistik, vynucené FULL

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		217K	15M	1570 (4)	00:00:19
* 1	TABLE ACCESS FULL	ACCOUNT_I1	217K	15M	1570 (4)	00:00:19

Srovnáním obou plánů dotazu lze říci, že vyžádání plného přístupu nad tabulkou, bez existence validních statistik, kde je vyhledáváno v poli, nad kterým je index a vyhovuje 30% řádků tabulky, zvýšilo náklady dotazu z implicitní hodnoty 889 na 1570. Jak je vidět, náklady dotazu při vynuceném přístupu přes celou tabulku pomocí hintu FULL jsou stejné, jako při testování 10% vyhovujících řádků. Porovnání nákladů implicitní a explicitní varianty s hintem ukazuje graf 2.



Graf 2: Porovnání nákladů pro změnu index->full, 30%, bez statistik

### Varianta - vyhovuje 50% všech řádků tabulky

V případě neexistence statistik, kdy výsledku dotazu vyhovuje 50% všech řádků tabulky, volí optimalizátor *přístup přes index*. Výsledný plán dotazu ukazuje tabulka 9.

Tabulka 9: Prováděcí plán, 50% vyhovujících řádků, bez statistik

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		351K	24M	1433 (1)	00:00:18
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	351K	24M	1433 (1)	00:00:18
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	351K		1042 (1)	00:00:13

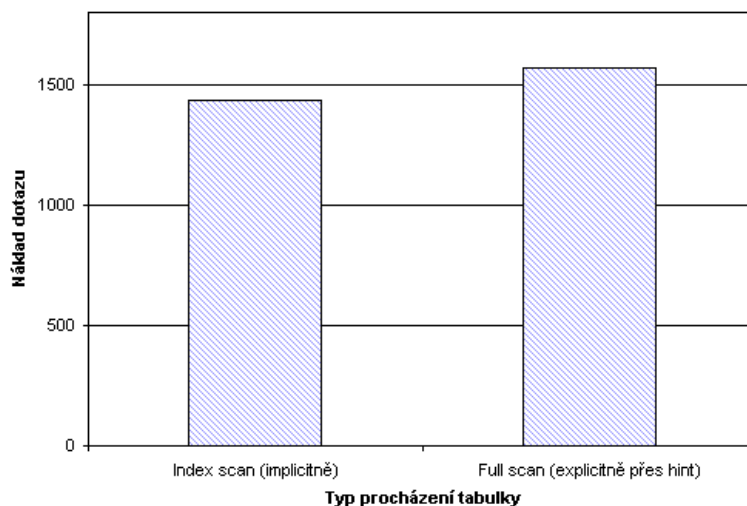
Vzhledem k variantě z 30% vyhovujících řádků došlo ke zvýšení nákladů z hodnoty 889 na 1433.

Při vyžádané změně pro použití *přístupu přes celou tabulku* pomocí hintu FULL je náklad dotazu roven hodnotě 1570, stejně jako v předchozích případech. Plán dotazu ukazuje tabulka 10.

Tabulka 10: Prováděcí plán, 50% vyhovujících řádků, bez statistik, vynucené FULL

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		351K	24M	1570 (4)	00:00:19
* 1	TABLE ACCESS FULL	ACCOUNT_I1	351K	24M	1570 (4)	00:00:19

Porovnání nákladů implicitní a explicitní varianty s hintem ukazuje Graf 3.



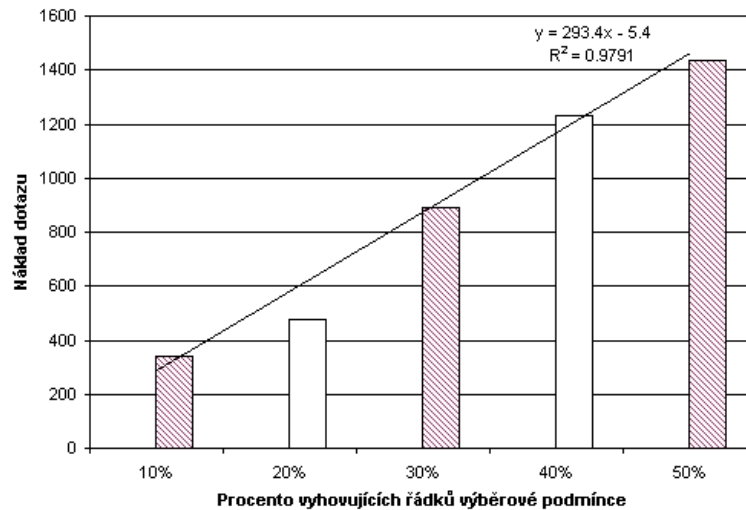
Graf 3: Porovnání nákladů pro změnu index->full, 50%, bez statistik

### Shrnutí dílčích závěrů

Při testování a vyžádané změně nad tabulkou s 1 000 000 záznamů při vyhledávání nad polem, nad kterým existuje index, volí Oracle pro zobrazení 10%, 30%, 50% vyhovujících záznamů z tabulky vždy *přístup přes index*. Explicitní změna požadavkem na *přístup přes celou tabulku* pomocí hintu FULL nezlepšila ani v jednom případě

hodnotu celkových nákladů - z hlediska nákladů byl *přístup přes index* v každém testovaném případě výhodnější.

Z podoby vrácených nákladů pro implicitní *přístup přes index*, který ukazuje graf 4, lze vyčíst, že výše nákladů stoupá lineárně s koeficientem determinace 0.9791 (bílé sloupce jsou další doměřené hodnoty pro přesnější odhad koeficientu determinace).



Graf 4: Porovnání nákladů u implicitního přístupu přes index

### 3.1.3 Existence statistik nad tabulkou - změna full -> index

Při existenci statistik dochází v případě tabulky *ACCOUNT\_I1* ke zvolení *přístupu přes celou tabulku* už zhruba od 16 % řádků, které vyhovují při dotazu nad indexovaným sloupcem *VALUE*. Následující testování sleduje ovlivnění nákladů při explicitní změně pomocí hintu *INDEX*, který mění implicitní *přístup přes celou tabulku* na *přístup přes index*.

#### Varianta A - vyhovuje 25% všech řádků tabulky

Při tomto dotazu s výběrem nad sloupcem, který je indexován, vyhovuje 25% ze všech řádků tabulky. Optimalizátor v tomto případě (statistiky existují) implicitně volí *přístup přes celou tabulku* (dotaz 9).

```
select * from ba.account_i1
where
value <= 250000
```

Dotaz 9: Požadavek na zobrazení 25 % procent dat z tabulky

Plán dotazu ukazuje tabulka 11.

Tabulka 11: Prováděcí plán pro dotaz 9

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		250K	12M	1588 (5)	00:00:20
* 1	TABLE ACCESS FULL	ACCOUNT_I1	250K	12M	1588 (5)	00:00:20

Na tento dotaz je nyní použitý hint INDEX, který explicitně vyžaduje použití existujícího indexu nad sloupcem *VALUE*, jak ukazuje dotaz 10. Získaný plán dotazu popisuje tabulka 12.

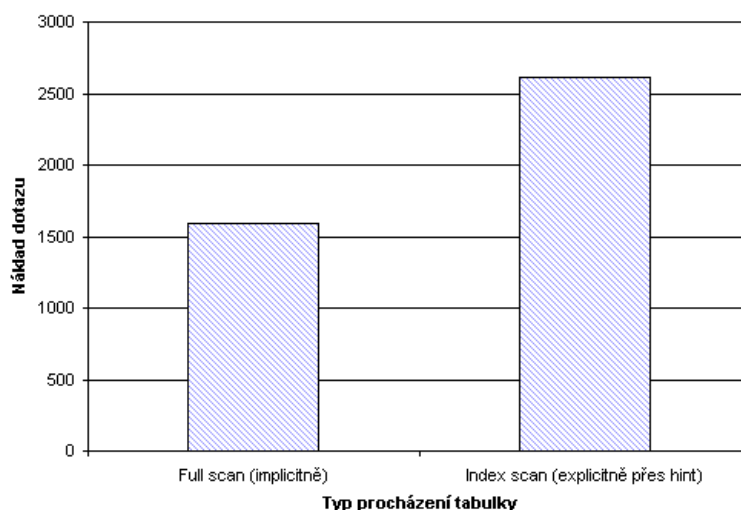
```
select /*+ INDEX(a ACCOUNT_I1_VALUE) */ * from ba.account_i1 a
where
a.value <= 250000
```

Dotaz 10: Požadavek na zobrazení 25 % procent dat z tabulky, hint INDEX

Tabulka 12: Prováděcí plán pro dotaz 10

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		250K	12M	2616 (1)	00:00:32
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	250K	12M	2616 (1)	00:00:32
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	250K		533 (2)	00:00:07

Při použití indexu dochází v tomto případě, kdy dotazu vyhovuje 25 % všech řádek tabulky, k navýšení nákladu dotazu z hodnoty 1588 na 2616. Rozdíl mezi získanými náklady u implicitní a hintem vynucené explicitní verze ukazuje graf 5.



Graf 5: Porovnání nákladů pro změnu full->index, 25%, existence statistik

### Varianta B - vyhovuje 50% všech řádků tabulky

U obdoby předchozího dotazu vyhovuje výběrové podmínce 50% ze všech řádků tabulky. Optimalizátor volí implicitně (statistiky existují) *přístup přes celou tabulku* (dotaz 11).

```
select * from ba.account_i1
where
value <= 500000
```

Dotaz 11: Požadavek na zobrazení 50 % procent dat z tabulky

Získaný plán dotazu ukazuje tabulka 13.

Tabulka 13: Prováděcí plán pro dotaz 11

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		500K	24M	1588 (5)	00:00:20
* 1	TABLE ACCESS FULL	ACCOUNT_I1	500K	24M	1588 (5)	00:00:20

Při vynucené aplikaci použití indexu přes hint INDEX (dotaz 12) je získán prováděcí plán dotazu, který popisuje Tabulka 14.

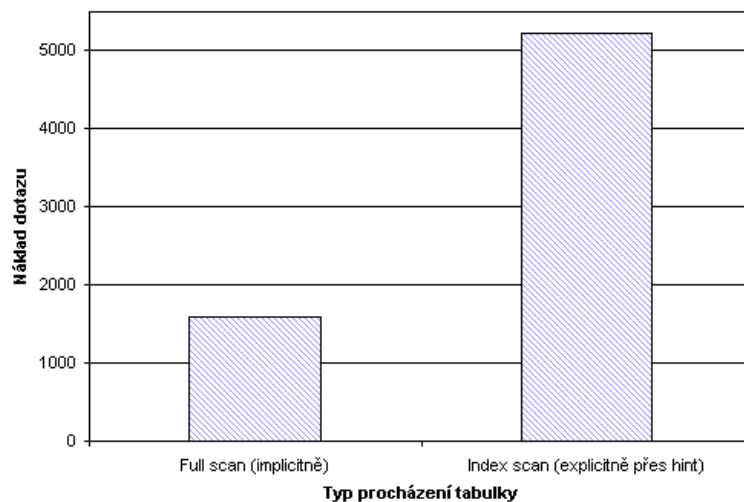
```
select /*+ INDEX(a ACCOUNT_I1_VALUE) */ * from ba.account_i1 a
where
a.value <= 500000
```

Dotaz 12: Požadavek na zobrazení 50 % procent dat z tabulky, hint INDEX

Tabulka 14: Prováděcí plán pro dotaz 12

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		500K	24M	5228 (1)	00:01:03
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	500K	24M	5228 (1)	00:01:03
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	500K		1063 (2)	00:00:13

Optimalizátorem zvolený implicitní *přístup přes celou tabulku* je v tomto případě stejně nákladný jako u varianty A. Získané náklady implicitní a explicitní varianty přes hint ukazuje graf 6.



Graf 6: Porovnání nákladů pro změnu full->index, 50%, existence statistik

### Varianta C - vyhovuje 75% všech řádků tabulky

Při požadavku na zobrazení 75% všech řádků tabulky výběrem nad sloupcem, u něhož existuje index, volí optimalizátor za přítomnosti validních vypočítaných statistik nad tabulkou *přístup přes celou tabulku* (dotaz 13). Zvolený plán provádění dotazu ukazuje tabulka 15.

```
select * from ba.account_i1
where
value <= 750000
```

Dotaz 13: Požadavek na zobrazení 75 % procent dat z tabulky

Tabulka 15: Prováděcí plán pro dotaz 13

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		750K	36M	1588 (5)	00:00:20
* 1	TABLE ACCESS FULL	ACCOUNT_I1	750K	36M	1588 (5)	00:00:20

Po změnu *přístupu přes celou tabulku* na přístup s využitím indexu nad sloupcem VALUE ukazuje dotaz 14. Prováděcí plán dotazu ukazuje tabulka 16.

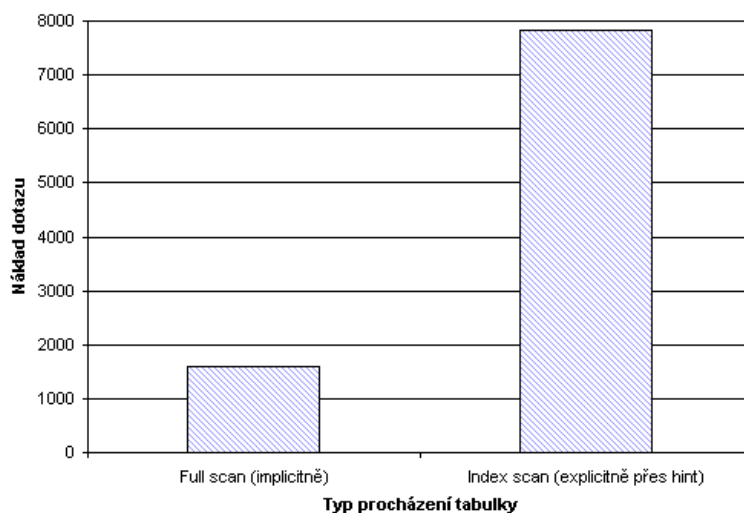
```
select /*+ INDEX(a ACCOUNT_I1_VALUE) */ * from ba.account_i1 a
where
a.value <= 750000
```

Dotaz 14: Požadavek na zobrazení 75 % procent dat z tabulky, hint INDEX

Tabulka 16: Prováděcí plán pro dotaz 14

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		750K	36M	7840 (1)	00:01:35
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I1	750K	36M	7840 (1)	00:01:35
* 2	INDEX RANGE SCAN	ACCOUNT_I1_VALUE	750K		1593 (2)	00:00:20

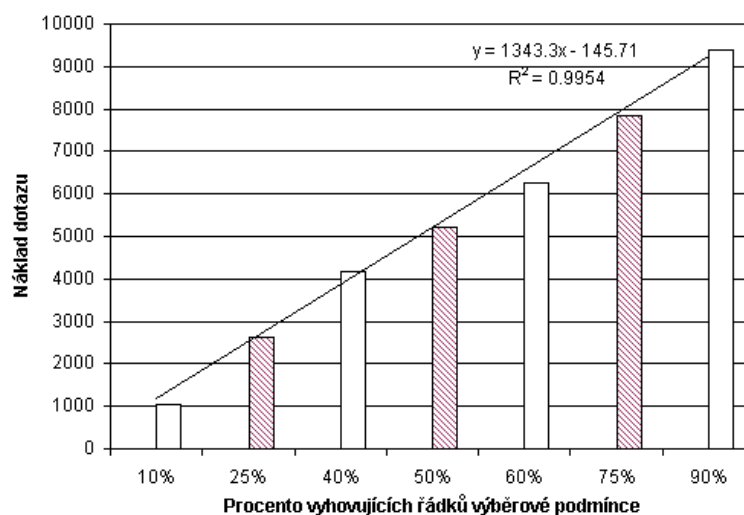
Jak je vidět z obou prováděcích plánů: *přístup přes celou tabulku* vykazuje v tomto případě opět konečné náklady ve výši 1588, zatímco *přístup přes index* náklady ve výši 7840. Vlastní získání příslušných ROWID řádků z indexu přes INDEX RANGE SCAN má náklad 1593 - největší nárůst nákladů je možné vysledovat ve fázi přístupu k řádkům tabulky přes tato ROWID - zde je náklad již 7840 - čili u vlastního vyhledávání řádků přes ROWID činí v tomto případě výše nákladů 6247 (od celkových nákladů byly odečteny náklady INDEX RANGE SCAN). Grafické porovnání obou způsobů přístupů k tabulce ukazuje graf 7.



Graf 7: Porovnání nákladů pro změnu full->index, 75%, existence statistik

### Shrnutí dílčích závěrů

Implicitní použití *přístupu přes celou tabulku*, který v tomto testovaném případě volí optimalizátor, zabezpečuje v případě tohoto výběru skutečně nejnižší náklady. Vynucené přepnutí přístupu přes aktivní index, který je postaven nad vyhledávaným sloupcem, v situaci, kdy existují validní statistiky nad tabulkou, nezlepší nákladově výkon dotazu - na bázi provedených měření lze odhadovat, že náklad stoupá lineárně, jak ukazuje graf 8, s koeficientem determinace rovnajícím se hodnotě 0.9954 (bílé sloupce jsou další doměřené hodnoty pro přesnější odhad koeficientu determinace).



Graf 8: Porovnání nákladů při explicitním přístupu přes index (hint)

### 3.1.4 Změna módu optimalizátoru pomocí hintu

Za pomoci hintu je možné v rámci každého SQL dotazu změnit použitý mód optimalizátoru. Následující testování sleduje, jak se projeví změna mezi použitým módem ALL\_ROWS a FIRST\_ROWS u dotazu nad tabulkou s 1 000 000 řádků, při



výběrové podmínce nad sloupcem ve dvou variantách A a B (výběrové podmínce vyhovuje 330 365 řádků).

Varianta A pracuje s tabulkou, nad kterou není definovaný index nad vyhledávaným polem; varianta B pak tento index definovaný má. V každé variantě dochází k testování změny módu ALL\_ROWS a FIRST\_ROWS.

V obou těchto případech existují aktualizované statistiky nad použitou tabulkou.

### Varianta A - tabulka bez indexu nad vyhledávaným polem

U této varianty je testovací dotaz položen nad tabulkou, která neobsahuje index u sloupce, nad nímž se provádí vyhledávání.

Nejprve je položen dotaz nad implicitní hodnotu OPTIMIZER\_MODE, která je ALL\_ROWS (dotaz 15) . Získaný prováděcí plán dotazu ukazuje tabulka 17.

```
select /*+ ALL_ROWS */ * from ba.account a
where
a.syn = '611'
```

Dotaz 15: Dotaz s výběrem z neindexované tabulky, mód ALL\_ROWS

Tabulka 17: Prováděcí plán pro dotaz 15

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		71429	3557K	1632 (4)	00:00:20
* 1	TABLE ACCESS FULL	ACCOUNT	71429	3557K	1632 (4)	00:00:20

Odhad nákladu tohoto dotazu je dle optimalizátoru 1632.

Dotaz 16 je obdobou předchozího dotazu, přepíná ale mód optimalizátoru do režimu FIRST\_ROWS, kdy jako parametr je použita hodnota 100 - jedná se tedy o požadavek na co nejrychlejší zobrazení prvních 100 vět. Získaný plán dotazu ukazuje tabulka 18.

```
select /*+ FIRST_ROWS(100) */ * from ba.account a
where
a.syn = '611'
```

Dotaz 16: Dotaz s výběrem z neindexované tabulky, mód FIRST\_ROWS, 100

Tabulka 18: Prováděcí plán pro dotaz 16

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		100	5100	4 (0)	00:00:01
* 1	TABLE ACCESS FULL	ACCOUNT	100	5100	4 (0)	00:00:01

Z plánu dotazu je viditelné opravdu výrazné snížení nákladu, dojde-li k vynucení FIRST\_ROWS módu optimalizátoru. ALL\_ROWS náklad 1632 byl snížen na FIRST\_ROWS výši 4 (100 prvních načítaných řádek).

### Varianta B - tabulka s indexem nad vyhledávaným polem

U této varianty probíhá dotaz za stejných podmínek jako u varianty A, v tomto případě ale nad tabulkou, u které existuje index nad prohledávaným sloupcem.

U položeného dotazu v rámci této varianty, ve které je pomocí hintu vynucen mód optimalizátoru ALL\_ROWS (dotaz 17), nedochází k použití indexu z důvodu velkého množství řádků volí optimalizátor *přístup přes celou tabulku* (tabulka 19). Získané náklady jsou tak stejné jako v případě použití tabulky bez indexu.

```
select /*+ ALL_ROWS */ * from ba.account_i a
where
a.syn = '611'
```

Dotaz 17: Dotaz s výběrem z indexované tabulky, mód ALL\_ROWS

Tabulka 19: Prováděcí plán pro dotaz 17

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		71429	3557K	1632 (4)	00:00:20
* 1	TABLE ACCESS FULL	ACCOUNT_I	71429	3557K	1632 (4)	00:00:20

Při dotazu v případě použití režimu optimalizátoru FIRST\_ROWS se 100 požadovanými řádky (dotaz 18) dochází v této situaci k použití indexu, protože se jedná o významně menší rozsah požadovaných vět. Prováděcí plán dotazu ukazuje tabulka 20.

```
select /*+ FIRST_ROWS(100) */ * from ba.account_i a
where
a.syn = '611'
```

Dotaz 18: Dotaz s výběrem z indexované tabulky, mód FIRST\_ROWS, 100

Tabulka 20: Prováděcí plán pro dotaz 18

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		100	5100	7 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID	ACCOUNT_I	100	5100	7 (0)	00:00:01
* 2	INDEX RANGE SCAN	ACCOUNT_I_SYN			3 (0)	00:00:01

## Shrnutí dílčích závěrů

U obou variant (A,B) se ukázalo, že použití režimu FIRST\_ROWS (100 vět) vrací nákladově výhodnější hodnoty, nežli procházení všech vět u režimu optimalizátoru ALL\_ROWS.

Určitou zajímavostí je to, že v případě varianty B optimalizátor zvolil pro přístup k indexované tabulce v režimu FIRST\_ROWS(100) použití indexu (dotaz 18, tabulka 20) - i když tímto způsobem zvýšil náklady z hodnoty 4 (dotaz 16, tabulka 18) na hodnotu 7. V tomto případě by bylo výhodnější zakázat pomocí hintu NO\_INDEX použití indexu (nebo vynutit *přístup přes celou tabulku* hintem FULL), čímž klesne náklad tohoto dotazu na hodnotu 4 (dotaz 19, tabulka 21).

```
select /*+ FIRST_ROWS(100) NO_INDEX(a) */ * from ba.account_i a
where
a.syn = '611'
```

Dotaz 19: Návrh snížení nákladů u výběru z indexované tabulky, FIRST\_ROWS, 100

Tabulka 21: Prováděcí plán pro dotaz 19

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		100	5100	4 (0)	00:00:01
* 1	TABLE ACCESS FULL	ACCOUNT_I	100	5100	4 (0)	00:00:01

## 3.2 Aplikace hintů na spojení více tabulek

V rámci této části dochází k testování změny nákladů a změny CPU rychlosti dotazu pro jednotlivé varianty dotazu, kde dochází ke spojování dvou tabulek.

Předpoklady u následujících měření:

- obě tabulky mají index nad spojovacím polem,
- neexistence statistik nad tabulkami,
- použitím hintu LEADING(a,b) nebo LEADING(b,a) může být ovlivněno pořadí spojování tabulek - pro testování bude zvoleno takové pořadí, které vrací nižší náklady dotazu.

Pro spojení jsou testovány modely 1:1 a 1:N. To v praxi znamená, že u modelu 1:1 dochází ke spojení s jednou větou ve druhé tabulce; u modelu 1:N pak ke spojení k žádné, jedné nebo více větám.

V rámci každého tohoto modelu je dále proveden:

- výběr všech vět,
- výběr LIKE vracející několik vět.

Pro každý model a specifikovaný výběr je proveden dotaz, jsou měřeny jeho náklady a CPU délka trvání dotazu pro všechny tři typy spojení mezi více tabulkami, typ spojení je vynucen použitím hintu. Získané hodnoty jsou pak porovnány s implicitními hodnotami dle prováděcího plánu dotazu, jak jej výchozím způsobem určí optimalizátor Oracle.

### 3.2.1 Model 1:1 - výběr všech vět

V modelu 1:1 (všechny vět) je pro každý jeden řádek jednotlivých prodejů dohledáván jeden konkrétní řádek s názvem položky prodaného zboží.

#### Varianta A1\_VSE - implicitní prováděcí plán

Pro dotaz 20 volí optimalizátor Oracle za definovaných podmínek hašované spojení (tabulka 22).

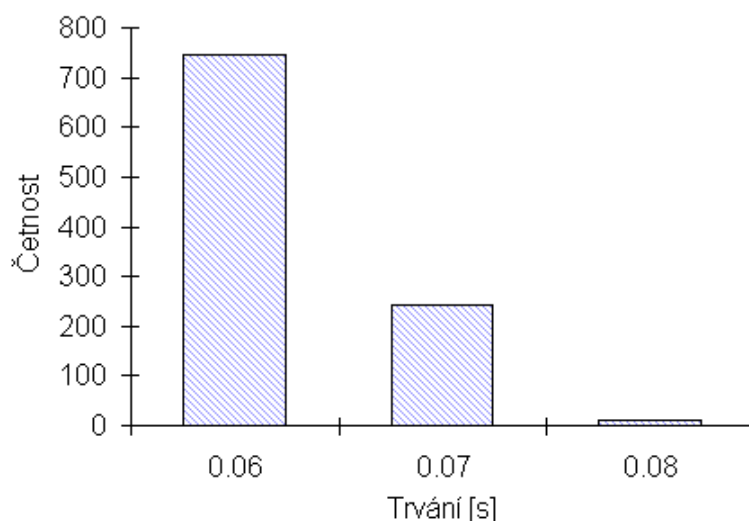
```
select a.item, b.item_name
from
ba.sale_i a, ba.item_i b
where
b.item = a.item
```

Dotaz 20: Model 1:1, všechny vět, implicitní plán

Tabulka 22: Prováděcí plán pro dotaz 20

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		39219	2527K		204 (2)	00:00:03
* 1	HASH JOIN		39219	2527K	1112K	204 (2)	00:00:03
2	INDEX FAST FULL SCAN	SALE_I_IDX_ITEM	39218	651K		34 (0)	00:00:01
3	TABLE ACCESS FULL	ITEM_I	25784	1233K		38 (3)	00:00:01

Náklady implicitního prováděcího plánu jsou 204. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 9.



Graf 9: Histogram naměřených CPU trvání pro dotaz 20 (1000 měření)<sup>7</sup>

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 23).

Tabulka 23: Varianta A1\_VSE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{A1\_VSE}$ (průměr)	0,0626 [s]
$s_{A1\_VSE}$ (směrodatná odchylka)	0,0046 [s]

### Varianta B1\_VSE - hintem vynucené vnořené spojení

V této variantě je původně použitý dotaz (dotaz 20) upravený pomocí hintu tak, že pro spojení tabulek je použito vynucené vnořené spojení (dotaz 21).

```
select /*+ USE_NL(a,b) */ a.item, b.item_name
from
ba.sale_i a, ba.item_i b
where
b.item = a.item
```

Dotaz 21: Model 1:1, všechny věty, vynucené vnořené spojení

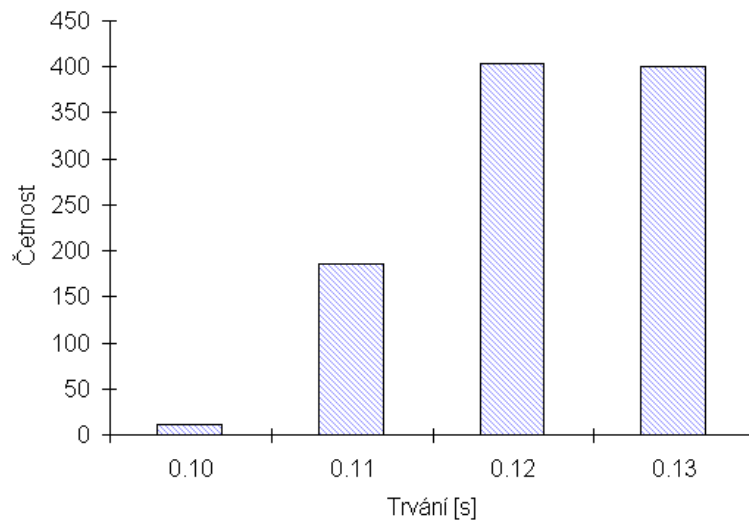
Získaný prováděcí plán dotazu ukazuje tabulka 24.

Tabulka 24: Prováděcí plán pro dotaz 21

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		39219	2527K	25858 (1)	00:05:11
1	NESTED LOOPS		39219	2527K	25858 (1)	00:05:11
2	TABLE ACCESS FULL	ITEM_I	25784	1233K	38 (3)	00:00:01
* 3	INDEX RANGE SCAN	SALE_I_IDX_ITEM	2	34	1 (0)	00:00:01

<sup>7</sup> Zdroj: Dávkové měření CPU trvání přes vlastní SQL skript.

Náklady dotazu jsou odhadnuty optimalizátorem na 25 858. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 10.



Graf 10: Histogram naměřených CPU trvání pro dotaz 21 (1000 měření)<sup>8</sup>

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 25).

Tabulka 25: Varianta B1\_VSE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{B1\_VSE}$ (průměr)	0,1219 [s]
$s_{B1\_VSE}$ (směrodatná odchylka)	0,0076 [s]

### Varianta C1\_VSE - hintem vynucené tříděné slučovací spojení

Varianta C1\_VSE upravuje původně použitý dotaz (dotaz 20) pomocí hintu tak, že pro spojení je použito tříděné slučovací spojení (dotaz 22).

```
select /*+ USE_MERGE(a,b) */ a.item, b.item_name
from
ba.sale_i a, ba.item_i b
where
b.item = a.item
```

Dotaz 22: Model 1:1, všechny věty, vynucené tříděné slučovací spojení

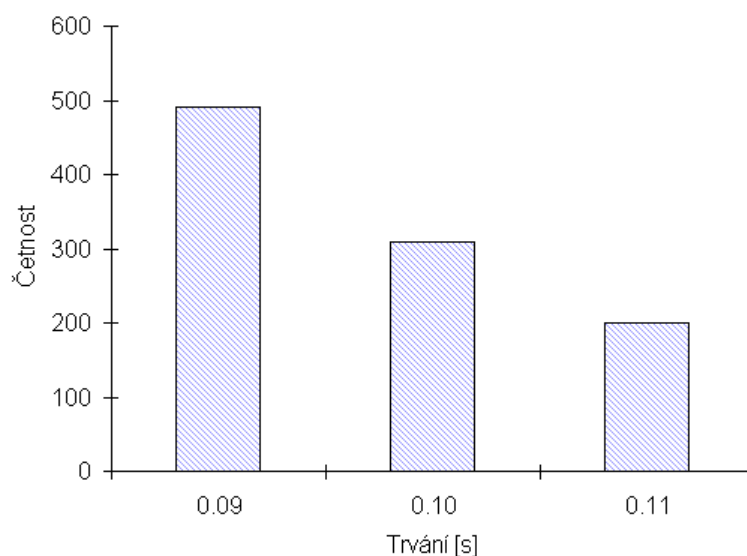
Získaný prováděcí plán dotazu ukazuje tabulka 26.

<sup>8</sup> Zdroj: Dávkové měření CPU trvání přes vlastní SQL skript.

Tabulka 26: Prováděcí plán pro dotaz 22

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		39219	2527K		545 (2)	00:00:07
1	MERGE JOIN		39219	2527K		545 (2)	00:00:07
2	INDEX FULL SCAN	SALE_I_IDX_ITEM	39218	651K		187 (0)	00:00:03
* 3	SORT JOIN		25784	1233K	3048K	358 (2)	00:00:05
4	TABLE ACCESS FULL	ITEM_I	25784	1233K		38 (3)	00:00:01

Náklady dotazu jsou odhadnuty optimalizátorem na 545. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 11.



Graf 11: Histogram naměřených CPU trvání pro dotaz 22 (1000 měření)<sup>9</sup>

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 27).

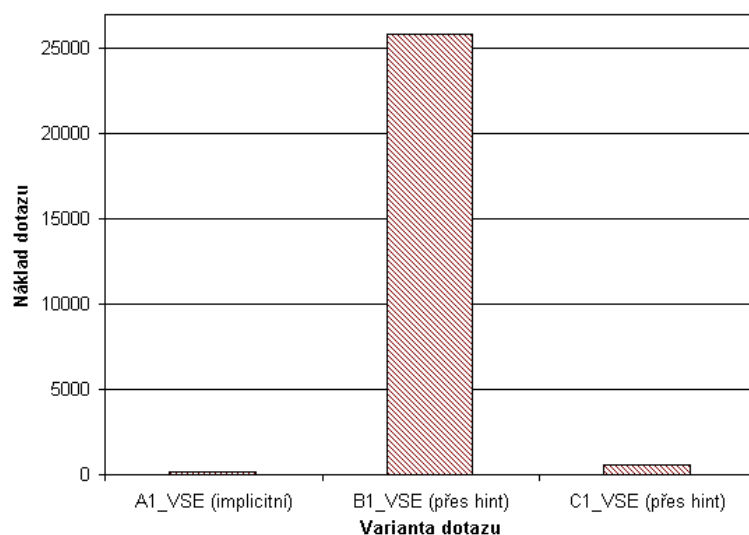
Tabulka 27: Varianta C1\_VSE - charakteristiky ze vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{C1\_VSE}$ (průměr)	0,0971 [s]
$s_{C1\_VSE}$ (směrodatná odchylka)	0,0077 [s]

### Porovnání nákladů dotazu

Při použití testovacího kritéria, kterým jsou náklady dotazu, došlo u testovaných variant (A1\_VSE, B1\_VSE, C1\_VSE) ke značnému rozdílu v odhadu potřebných nákladů na dotaz. Vizuálně ukazuje porovnání jednotlivých získaných odhadů nákladů dotazu graf 12.

<sup>9</sup> Zdroj: Dávkové měření CPU trvání přes vlastní SQL skript.



Graf 12: Porovnání odhadů nákladů dotazu pro varianty A1\_VSE,B1\_VSE,C1\_VSE

Ze získaných nákladů lze konstatovat, že v dané situaci optimalizátor správně implicitně použil hašované spojení (varianta A1\_VSE), ostatní obě varianty, které vynucují pomocí hintu jiný druh spojení (B1\_VSE, C1\_VSE), vykazují vyšší náklady, u varianty B1\_VSE jsou tyto náklady výrazně vyšší.

#### Porovnání CPU trvání dotazu

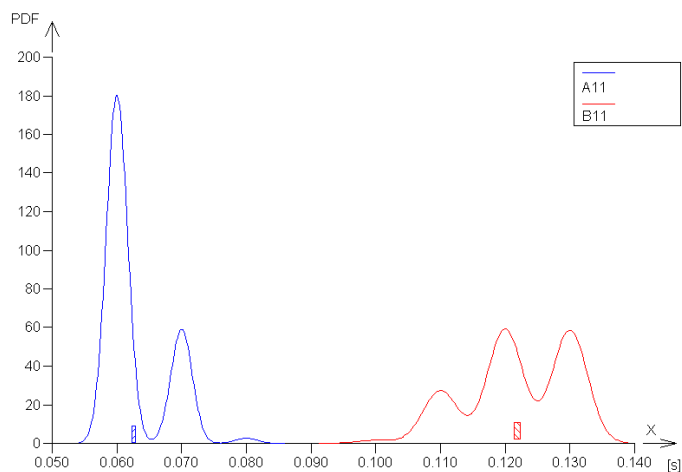
Porovnání CPU trvání dotazu je založeno na 1000 měřeních pro dotaz každé testované varianty (A1\_VSE, B1\_VSE, C1\_VSE). K porovnání vždy dvou získaných výběrů proti sobě byl použit statistický nástroj QC Expert<sup>10</sup>, kdy byla vždy pro oba tyto výběry (A1\_VSE-B1\_VSE, A1\_VSE-C1\_VSE, B1\_VSE-C1\_VSE) provedena funkce *Porovnání 2 výběrů*. Výsledky tohoto porovnání jednoznačně dospěly k závěru, že vzorky A1\_VSE-B1\_VSE a A1\_VSE-C1\_VSE jsou rozdílné jak z hlediska shody rozptylů, tak z hlediska shody průměrů (protokol porovnání viz příloha C), přičemž při porovnání bylo užito i robustních testů [8] a testování shody K-S.

U testovaných variant B1\_VSE-C1\_VSE byla rozdělení pomocí K-S označena také jako rozdílná, i když testování shody rozptylů označilo rozptyly výběrů jako shodné s pravděpodobností 0.34 a 0.36 (příloha C).

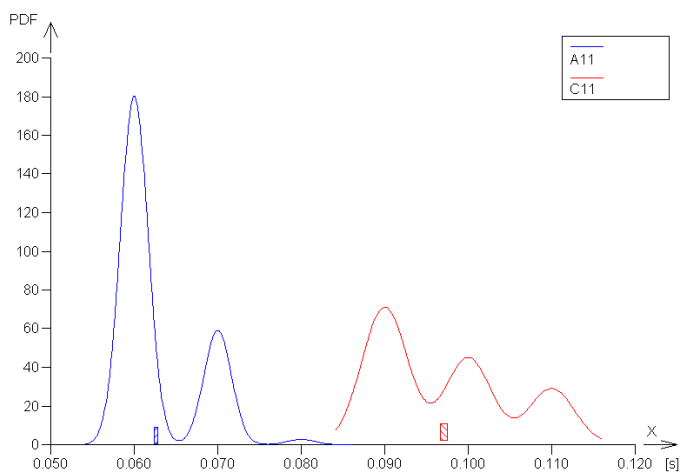
Diference mezi porovnávanými výběry jsou vizuálně znázorněné v grafu jádrových odhadů pravděpodobnosti pro vždy pro dva porovnávané výběry (graf 13, graf 14, graf 15).

<sup>10</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

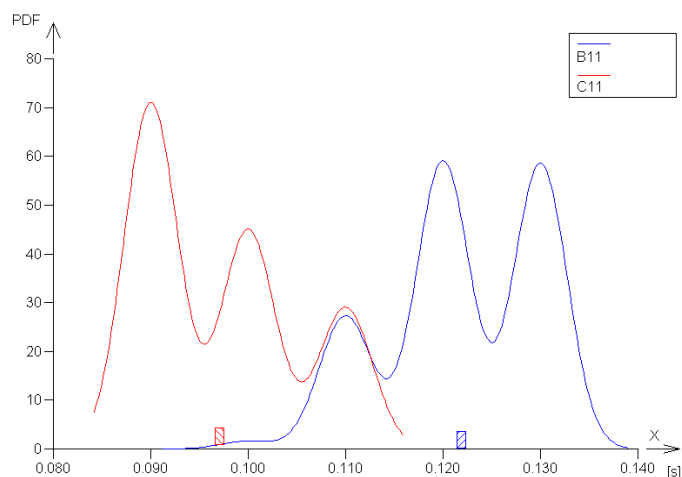




Graf 13: Jádřové odhady hustoty pravděpodobnosti pro výběry A1\_VSE-B1\_VSE<sup>11</sup>



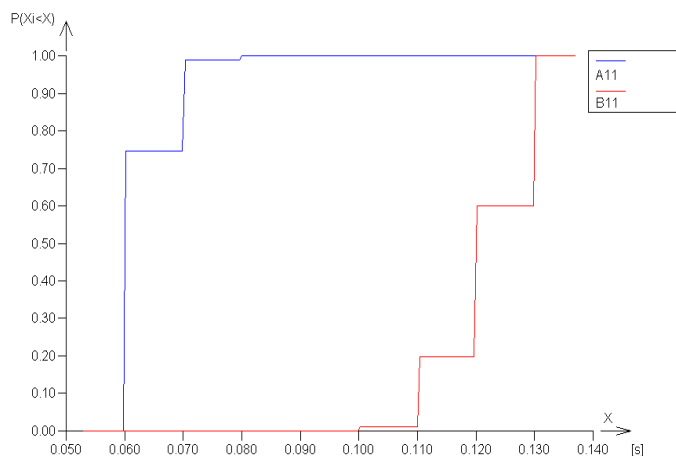
Graf 14: Jádřové odhady hustoty pravděpodobnosti pro výběry A1\_VSE-C1\_VSE<sup>11</sup>



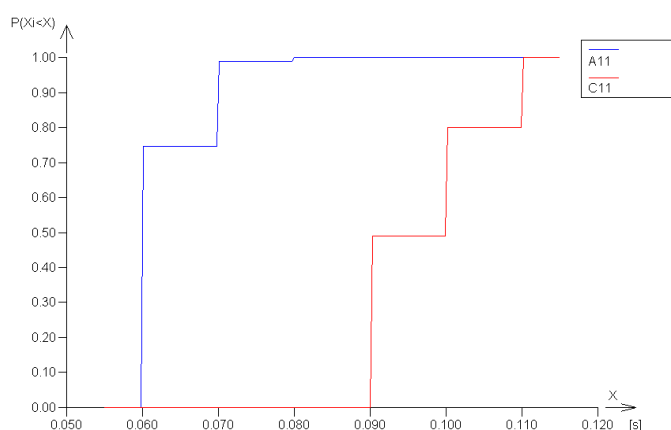
Graf 15: Jádřové odhady hustoty pravděpodobnosti pro výběry B1\_VSE-C1\_VSE<sup>11</sup>

Porovnání příslušných distribučních funkcí vždy pro dva porovnávané výběry ukazují graf 16, graf 17, graf 18 .

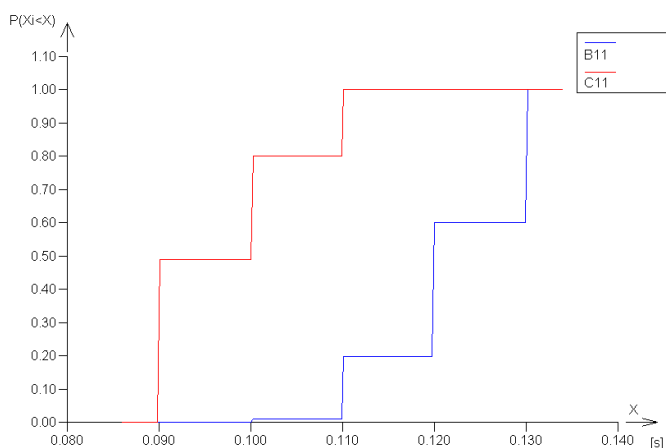
<sup>11</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.



Graf 16: Empirické distribuční funkce pro porovnávané výběry A1\_VSE-B1\_VSE<sup>12</sup>



Graf 17: Empirické distribuční funkce pro porovnávané výběry A1\_VSE-C1\_VSE<sup>12</sup>

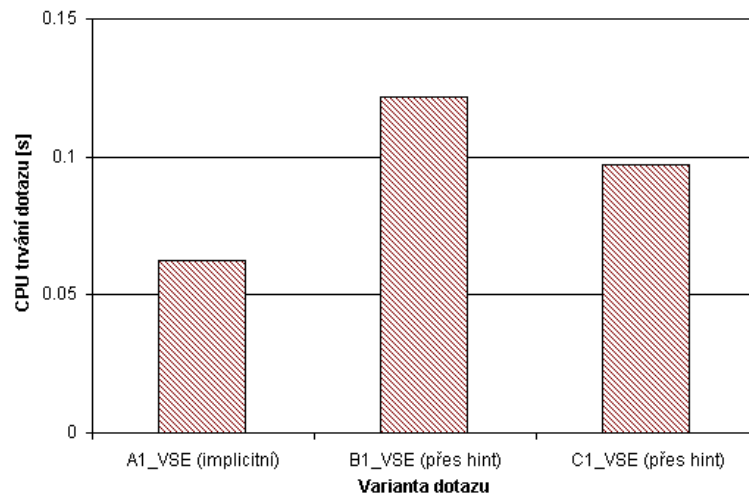


Graf 18: Empirické distribuční funkce pro porovnávané výběry (B1\_VSE-C1\_VSE)<sup>12</sup>

Lze říci, že u sledovaných variant v tomto modelu 1:1 byla z hlediska průměrné CPU rychlosti dotazu nejrychlejší varianta A1\_VSE, kterou také optimalizátor Oracle použil

<sup>12</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

jako implicitní (graf 19). I zde platí (jako u nákladů dotazu), že varianta B1\_VSE podává v rámci testovaného modelu z hlediska průměrných hodnot nejhorší výsledky.



Graf 19: Průměry CPU trvání dotazu A1\_VSE,B1\_VSE,C1\_VSE (1000 měření)

### Shrnutí dílčích závěrů

Pomocí hintu lze významným způsobem ovlivnit způsob spojování více tabulek dotazu volbou použité metody spojování. V rámci testovaného modelu 1:1 jak z hlediska testování nákladu, tak z hlediska CPU rychlosti dotazu, podává varianta A1\_VSE, kterou implicitně použil optimalizátor, nejlepší výkon.

### 3.2.2 Model 1:1 - výběr přes operátor LIKE

V modelu 1:1 (LIKE) je přes operátor LIKE proveden výběr specifického počtu řádek jednotlivých prodejů (6179 řádků) a ke každému z nich je dohledán konkrétní jeden řádek s názvem položky prodaného zboží.

#### Varianta A1\_LIKE - Implicitní prováděcí plán

Pro dotaz 23 volí optimalizátor Oracle za definovaných podmínek hašované spojení (tabulka 28).

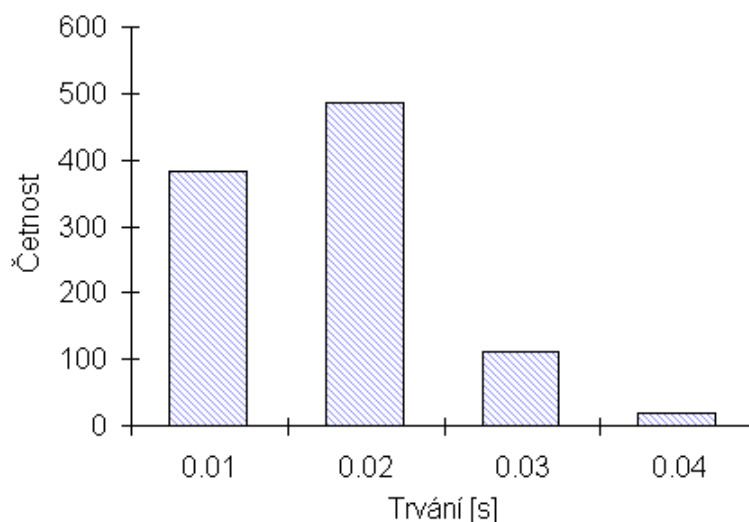
```
select a.item, b.item_name
from
ba.sale_i a, ba.item_i b
where
a.item like 'S-%' and
b.item = a.item
```

Dotaz 23: Model 1:1, výběr LIKE, implicitní plán

Tabulka 28: Prováděcí plán pro dotaz 23

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6435	414K	71 (3)	00:00:01
* 1	HASH JOIN		6435	414K	71 (3)	00:00:01
* 2	INDEX RANGE SCAN	SALE_I_IDX_ITEM	6436	106K	32 (0)	00:00:01
* 3	TABLE ACCESS FULL	ITEM_I	5339	255K	38 (3)	00:00:01

Náklady implicitního prováděcího plánu jsou 71. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 20.



Graf 20: Histogram naměřených CPU trvání pro dotaz 23 (1000 měření)

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 29).

Tabulka 29: Varianta A1\_LIKE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{A1\_LIKE}$ (průměr)	0,0176 [s]
$s_{A1\_LIKE}$ (směrodatná odchylka)	0,0072 [s]

### Varianta B1\_LIKE - hintem vynucené vnořené spojení

V této variantě je původně použitý dotaz (dotaz 23) upraven pomocí hintu tak, že pro spojení tabulek je použito vynucené vnořené spojení (dotaz 24).

```
select /*+ USE_NL(a,b) */ a.item, b.item_name
from
ba.sale_i a, ba.item_i b
where
a.item like 'S-%' and
b.item = a.item
```

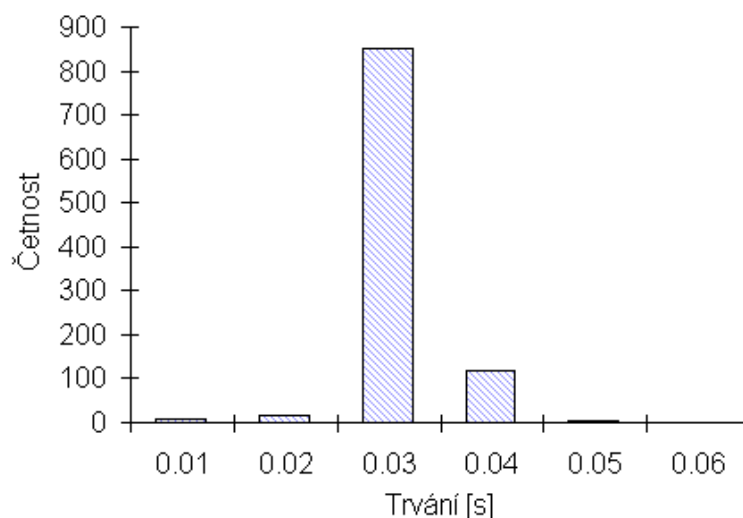
Dotaz 24: Model 1:1, výběr LIKE, vynucené vnořené spojení

Získaný prováděcí plán dotazu ukazuje tabulka 30.

Tabulka 30: Prováděcí plán pro dotaz 24

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6435	414K	6029 (11)	00:01:13
1	NESTED LOOPS		6435	414K	6029 (11)	00:01:13
* 2	TABLE ACCESS FULL	ITEM_I	5339	255K	38 (3)	00:00:01
* 3	INDEX RANGE SCAN	SALE_I_IDX_ITEM	1	17	1 (0)	00:00:01

Náklady dotazu jsou odhadnuty optimalizátorem na 6029. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 21.



Graf 21: Histogram naměřených CPU trvání pro dotaz 24 (1000 měření)

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 31).

Tabulka 31: Varianta B1\_LIKE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{B1\_LIKE}$ (průměr)	0,0309 [s]
$s_{B1\_LIKE}$ (směrodatná odchylka)	0,0042 [s]

### Varianta C1\_LIKE - hintem vynucené tříděné slučovací spojení

Varianta C1\_LIKE upravuje původně použitý dotaz (dotaz 23) pomocí hintu tak, že pro spojení je použité tříděné slučovací spojení (dotaz 25).

```
select /*+ USE_MERGE(a,b) */ a.item, b.item_name
from
ba.sale_i a, ba.item_i b
where
a.item like 'S-%' and
b.item = a.item
```

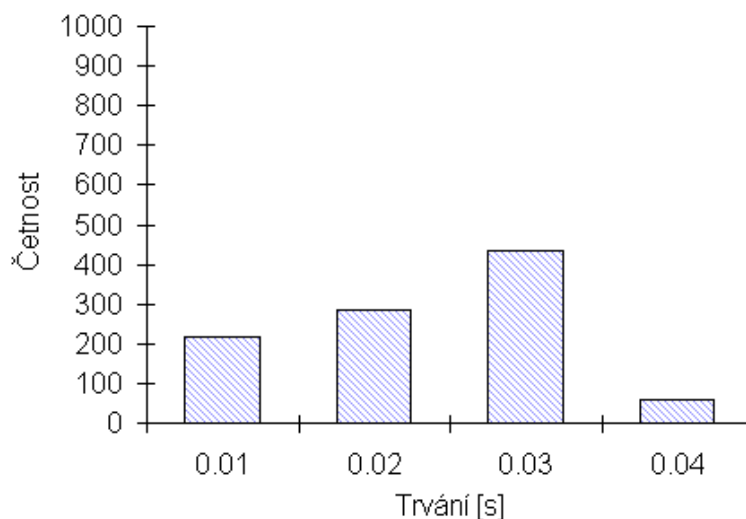
Dotaz 25: Model 1:1, výběr LIKE, vynucené tříděné slučovací spojení

Získaný prováděcí plán dotazu ukazuje tabulka 32.

Tabulka 32: Prováděcí plán pro dotaz 25

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		6435	414K		138 (3)	00:00:02
1	MERGE JOIN		6435	414K		138 (3)	00:00:02
* 2	INDEX RANGE SCAN	SALE_I_IDX_ITEM	6436	106K		32 (0)	00:00:01
* 3	SORT JOIN		5339	255K	648K	106 (3)	00:00:02
* 4	TABLE ACCESS FULL	ITEM_I	5339	255K		38 (3)	00:00:01

Náklady dotazu jsou optimalizátorem odhadnuty na 138. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 22.



Graf 22: Histogram naměřených CPU trvání pro dotaz 22 (1000 měření)

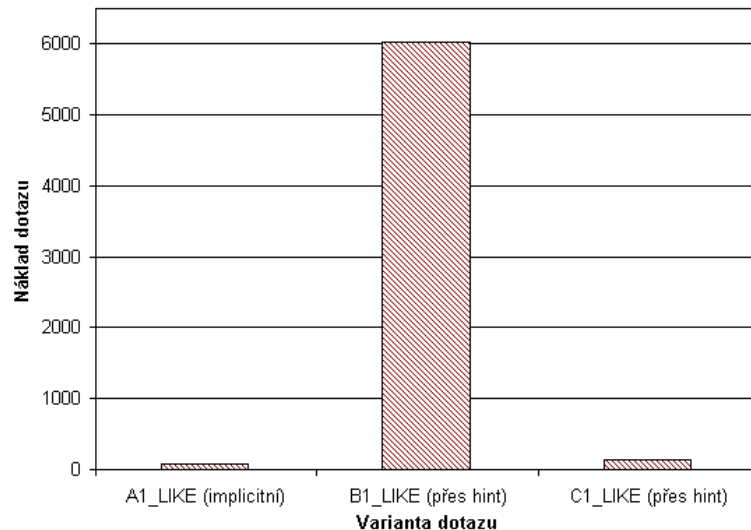
Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 33).

Tabulka 33: Varianta C1\_LIKE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{C1\_LIKE}$ (průměr)	0,0233 [s]
$s_{C1\_LIKE}$ (směrodatná odchylka)	0,0088 [s]

### Porovnání nákladů dotazu

U varianty 1:1 LIKE je srovnáním nákladů dotazu viditelný rozdíl ve výrazně vysokých nákladech varianty B1\_LIKE. Graficky ukazuje odhad použitých nákladů graf 23.



Graf 23: Porovnání odhadů nákladů dotazu pro varianty A1\_LIKE, B1\_LIKE, C1\_LIKE

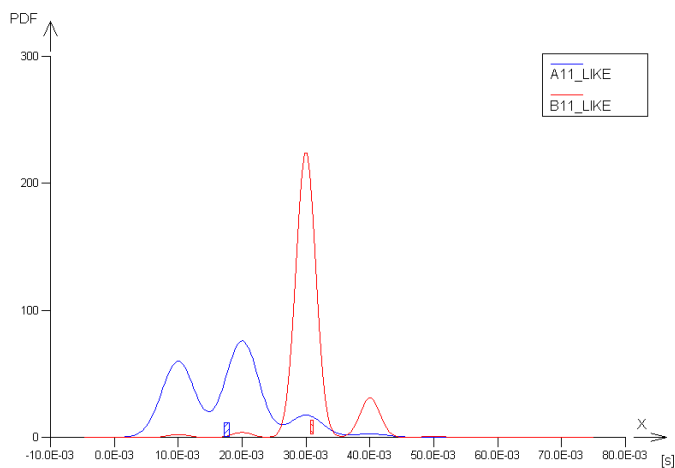
Optimalizátor správně zvolil použití plánu A1\_LIKE, který z hlediska kritéria nákladu dotazu vykazuje také nejnižší průměrný náklad. Varianta B1\_LIKE je v této popsané situaci nákladově nevyhovující. C1\_LIKE je použitelná, má o něco vyšší náklady než varianta A1\_LIKE.

#### Porovnání CPU trvání dotazu

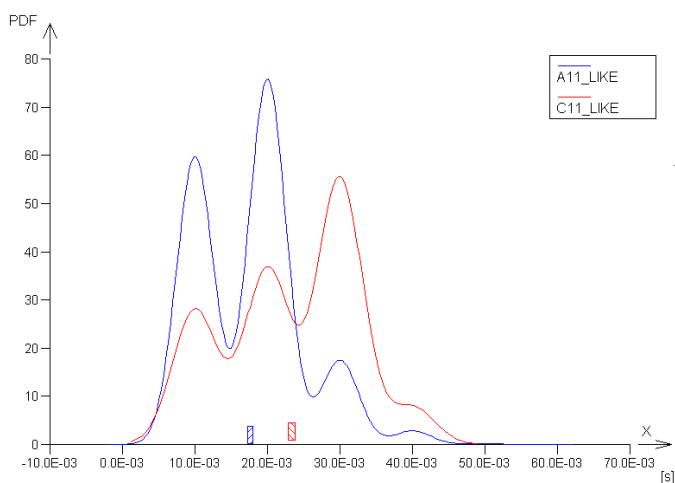
Pro porovnání CPU trvání dotazu bylo provedeno 1000 měření pro dotaz každé testované varianty (A1\_LIKE, B1\_LIKE, C1\_LIKE). Porovnáním jednotlivých testovaných variant mezi sebou (A1\_LIKE-B1\_LIKE, A1\_LIKE-C1\_LIKE, B1\_LIKE-C1\_LIKE) ve statistickém nástroji QC Expert<sup>13</sup> bylo zjištěno, že všechny tři porovnávané varianty jsou rozdílné (příloha C).

Diference mezi porovnávanými výběry jsou vizuálně znázorněné v grafu jádrových odhadů pravděpodobnosti pro vždy pro dva porovnávané výběry (graf 24, graf 25, graf 26).

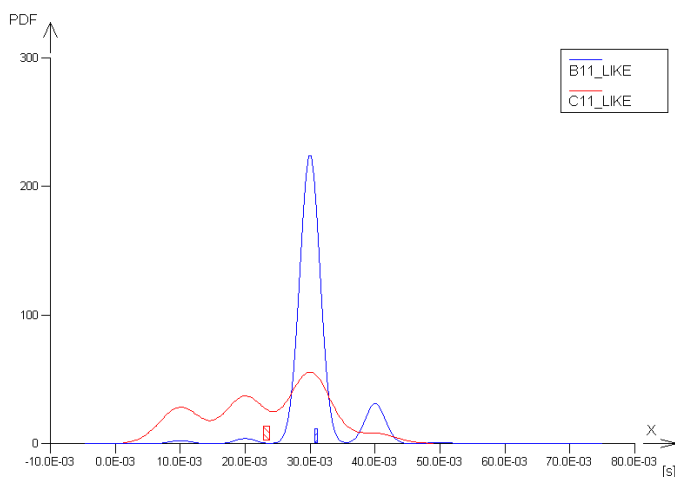
<sup>13</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.



Graf 24: Jádřové odhady hustoty pravděpodobnosti pro výběry A1\_LIKE-B1\_LIKE<sup>14</sup>



Graf 25: Jádřové odhady hustoty pravděpodobnosti pro výběry A1\_LIKE-C1\_LIKE<sup>14</sup>

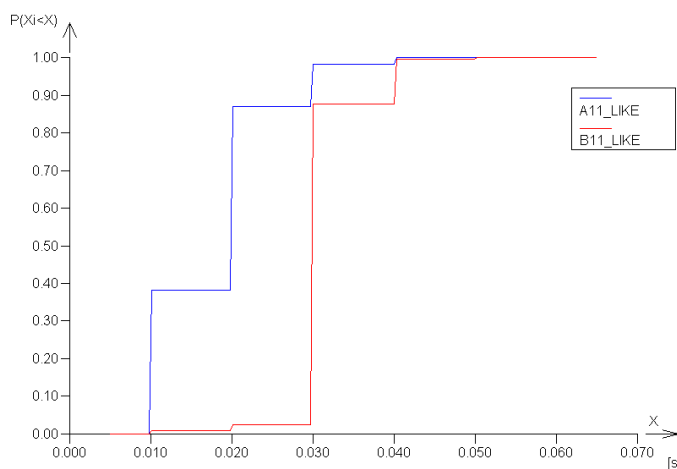


Graf 26: Jádřové odhady hustoty pravděpodobnosti pro výběry B1\_LIKE-C1\_LIKE<sup>14</sup>

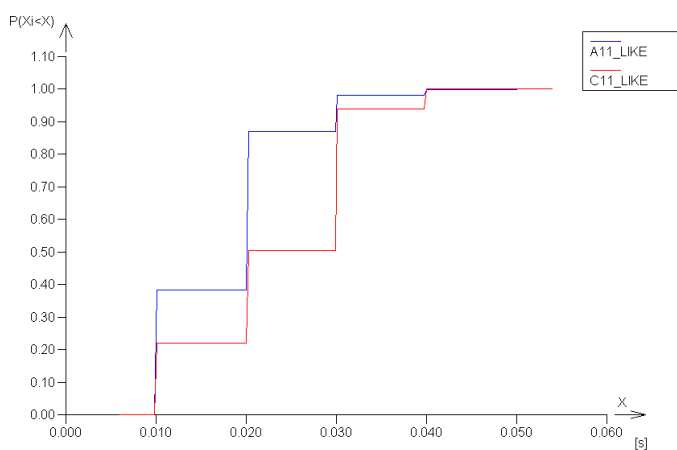
<sup>14</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.



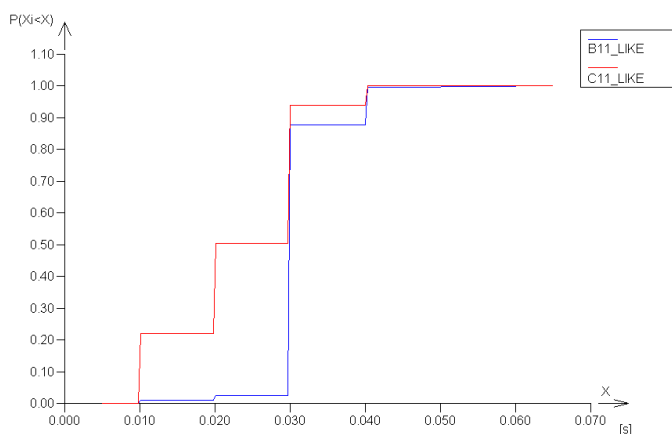
Porovnání příslušných distribučních funkcí vždy pro dva porovnávané výběry ukazují graf 27, graf 28, graf 29.



Graf 27: Empirické distribuční funkce pro porovnávané výběry A1\_LIKE-B1\_LIKE<sup>15</sup>



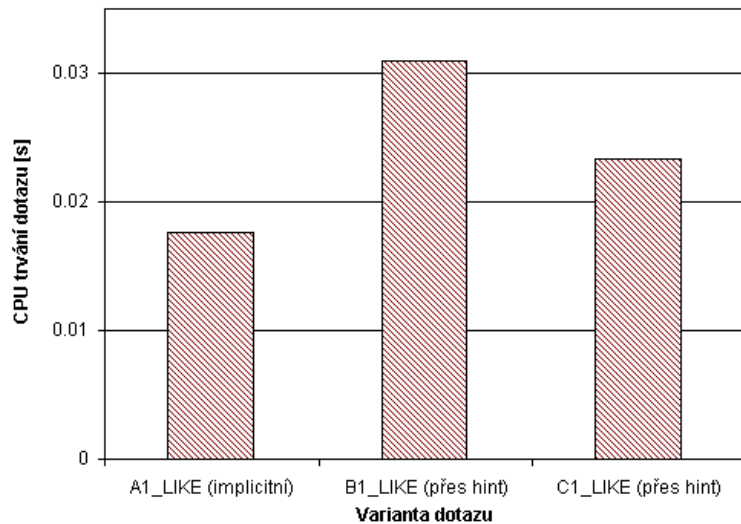
Graf 28: Empirické distribuční funkce pro porovnávané výběry A1\_LIKE-C1\_LIKE<sup>15</sup>



Graf 29: Empirické distribuční funkce pro porovnávané výběry B1\_LIKE-C1\_LIKE<sup>15</sup>

<sup>15</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

V případě tohoto testovaného modelu 1:1 LIKE vykazují jednotlivé porovnávané varianty odlišné výsledky, všechny porovnávané výběry byly označeny jako rozdílné. Z hlediska průměrné CPU rychlosti zpracování dotazu se jeví jako nejrychlejší varianta A1\_LIKE, kterou rovněž optimalizátor použil jako implicitní (graf 30).



Graf 30: Průměry CPU trvání dotazu A1\_LIKE,B1\_LIKE,C1\_LIKE (1000 měření)

### Shrnutí dílčích závěrů

V rámci modelu 1:1 s výběrovým dotazem přes operátor LIKE použil optimalizátor správně hašované spojení (varianta A1\_LIKE). Z hlediska obou posuzovaných kritérií se tato varianta jeví na základě provedených testování jako nejvýhodnější.

### 3.2.3 Model 1:N - výběr všech vět

V modelu 1:N (VŠE) jsou pro všechny řádky jednotlivých položek dohledány odpovídající prodeje. Vazba je 1:N, přičemž kardinalita spojení se pohybuje v rozmezí od 0 do 650 vět, průměrná hodnota kardinality ve vztahu mezi použitými tabulkami je 1.59 vět.

#### Varianta A2\_VSE - Implicitní prováděcí plán

Pro dotaz 26 volí optimalizátor Oracle za výše definovaných podmínek vnořené spojení (tabulka 34).

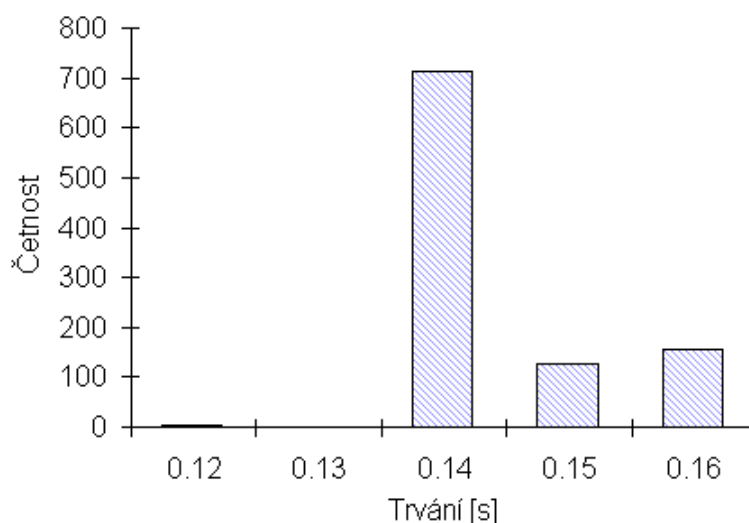
```
select a.item, b.period, b.date_, b.price
from ba.item_i a, ba.sale_i b
where
b.item = a.item
```

Dotaz 26: Model 1:N, všechny věty, implicitní plán (vnořené spojení)

Tabulka 34: Prováděcí plán pro dotaz 26

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		39219	2336K	81 (19)	00:00:01
1	NESTED LOOPS		39219	2336K	81 (19)	00:00:01
2	TABLE ACCESS FULL	SALE_I	39218	1685K	68 (3)	00:00:01
* 3	INDEX UNIQUE SCAN	ITEM_I_IDX1	1	17	0 (0)	00:00:01

Náklady tohoto implicitního prováděcího plánu, kdy optimalizátor použil vnořené spojení mezi tabulkami, jsou 81. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 31.



Graf 31: Histogram naměřených CPU trvání pro dotaz 26 (1000 měření)

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 35).

Tabulka 35: Varianta A2\_VSE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{A2\_VSE}$ (průměr)	0,1443 [s]
$s_{A2\_VSE}$ (směrodatná odchylka)	0,0075 [s]

### Varianta B2\_VSE - hintem vynucené hašované spojení

U varianty B2\_VSE je na spojení dvou tabulek použito hintem vynucené hašované spojení (dotaz 27). Získaný prováděcí plán dotazu popisuje tabulka 36.

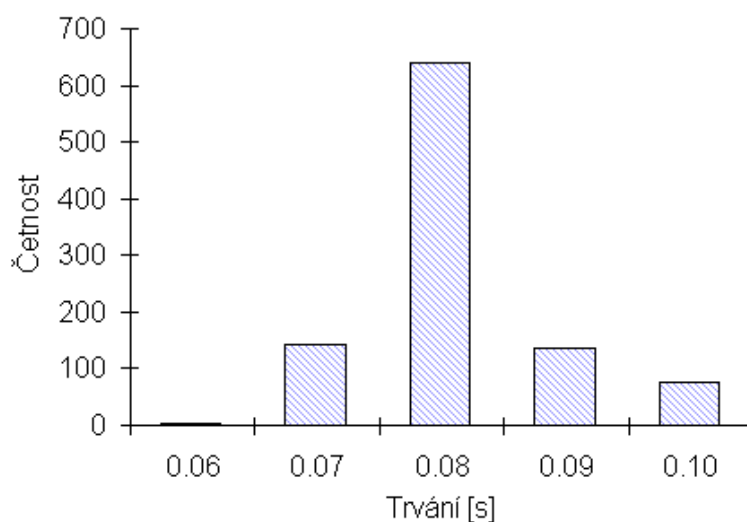
```
select /** USE_HASH(a,b) */ a.item, b.period, b.date_, b.price
from ba.item_i a, ba.sale_i b
where
b.item = a.item
```

Dotaz 27: Model 1:N, všechny věty, vynucené hašované spojení

Tabulka 36: Prováděcí plán pro dotaz 27

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		39219	2336K	91 (5)	00:00:02
* 1	HASH JOIN		39219	2336K	91 (5)	00:00:02
2	INDEX FAST FULL SCAN	ITEM_I_IDX1	25784	428K	21 (0)	00:00:01
3	TABLE ACCESS FULL	SALE_I	39218	1685K	68 (3)	00:00:01

Náklady vynuceného hašovaného spojení jsou v rámci prováděcího plánu odhadnuty na 91. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 32.



Graf 32: Histogram naměřených CPU trvání pro dotaz 27 (1000 měření)

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 37).

Tabulka 37: Varianta B2\_VSE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{B2\_VSE}$ (průměr)	0,0814 [s]
$s_{B2\_VSE}$ (směrodatná odchylka)	0,0076 [s]

### Varianta C2\_VSE - hintem vynucené tříděné slučovací spojení

Varianta C2\_VSE upravuje původně použitý dotaz (dotaz 26) pomocí hintu tak, že pro spojení je použité tříděné slučovací spojení (dotaz 28). Získaný prováděcí plán dotazu ukazuje tabulka 38.

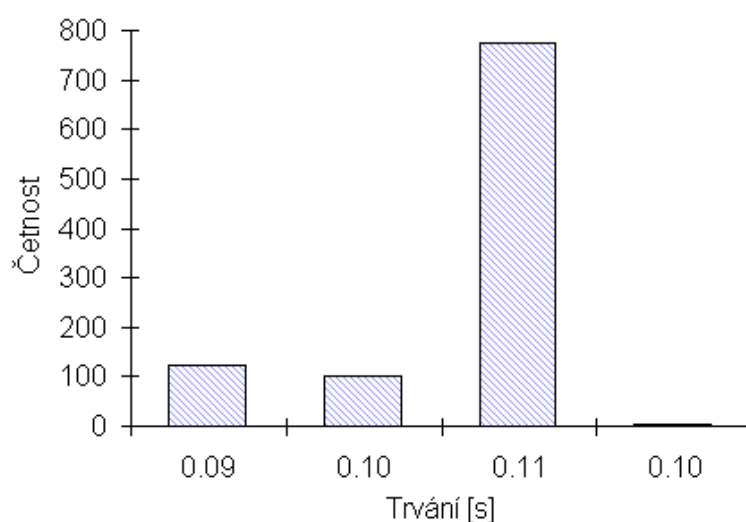
```
select /** USE_MERGE(a,b) */ a.item, b.period, b.date_, b.price
from ba.item_i a, ba.sale_i b
where
b.item = a.item
```

Dotaz 28: Model 1:N, všechny věty, vynucené tříděné slučovací spojení

Tabulka 38: Prováděcí plán pro dotaz 28

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		39219	2336K		628 (2)	00:00:08
1	MERGE JOIN		39219	2336K		628 (2)	00:00:08
2	INDEX FULL SCAN	ITEM_I_IDX1	25784	428K		113 (0)	00:00:02
* 3	SORT JOIN		39218	1685K	4344K	515 (2)	00:00:07
4	TABLE ACCESS FULL	SALE_I	39218	1685K		68 (3)	00:00:01

Náklady vynuceného tříděného slučovacího spojení jsou v rámci příslušného získaného prováděcího plánu odhadnuty na 628. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 33.



Graf 33: Histogram naměřených CPU trvání pro dotaz 28 (1000 měření)

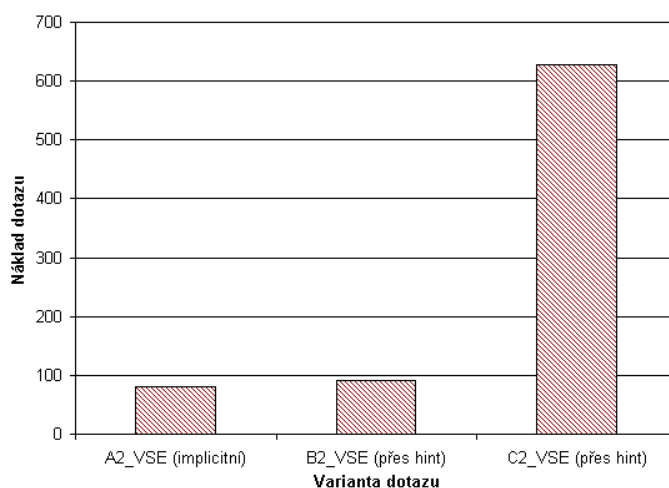
Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 39).

Tabulka 39: Varianta C2\_VSE - charakteristiky získané ze vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{C2\_VSE}$ (průměr)	0,1065 [s]
$s_{C2\_VSE}$ (směrodatná odchylka)	0,0069 [s]

### Porovnání nákladů dotazu

U varianty 1:N VSE je srovnáním nákladů dotazu viditelný rozdíl zejména ve výrazně vysokých nákladech varianty C2\_VSE. Graficky ukazuje odhad použitých nákladů graf 34.



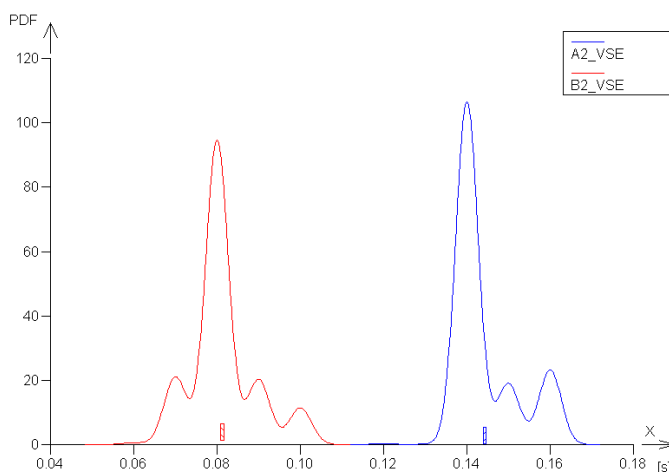
Graf 34: Porovnání odhadů nákladů dotazu pro varianty A2\_VSE,B2\_VSE,C2\_VSE

Optimalizátor správně zvolil použití plánu A2\_VSE, který z hlediska kritéria nákladu dotazu vykazuje o něco nižší hodnotu nežli varianta B2\_VSE. Varianta C2\_VSE je z hlediska tohoto kritéria v této popsané situaci nákladově nevyhovující.

### Porovnání CPU trvání dotazu

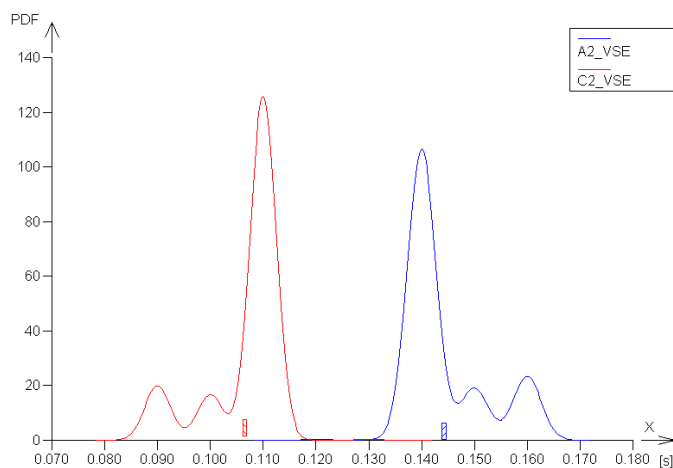
Pro porovnání CPU trvání dotazu bylo provedeno 1000 měření pro dotaz každé testované varianty (A2\_VSE, B2\_VSE, C2\_VSE). Porovnáním jednotlivých testovaných variant mezi sebou (A2\_VSE-B2\_VSE, A2\_VSE-C2\_VSE, B2\_VSE-C2\_VSE) ve statistickém nástroji QC Expert<sup>16</sup> bylo zjištěno, že všechny tři porovnávané varianty jsou rozdílné (příloha C).

Diference mezi porovnávanými výběry jsou vizuálně znázorněné v grafu jádrových odhadů pravděpodobnosti pro vždy pro dva porovnávané výběry (graf 35, graf 36, graf 37).

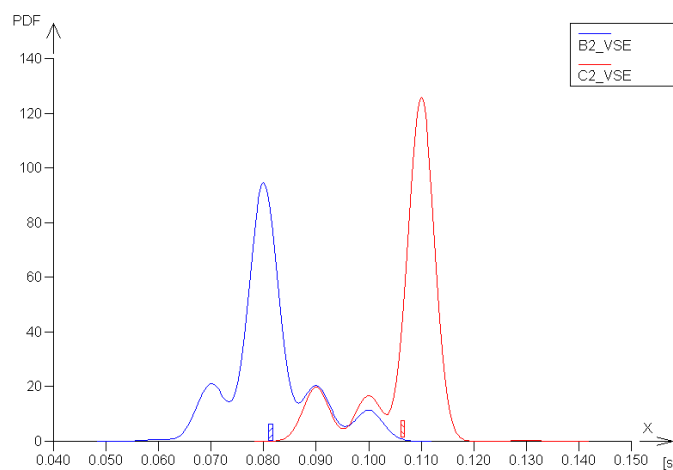


Graf 35: Jádrové odhady hustoty pravděpodobnosti pro výběry A2\_VSE-B2\_VSE<sup>16</sup>

<sup>16</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

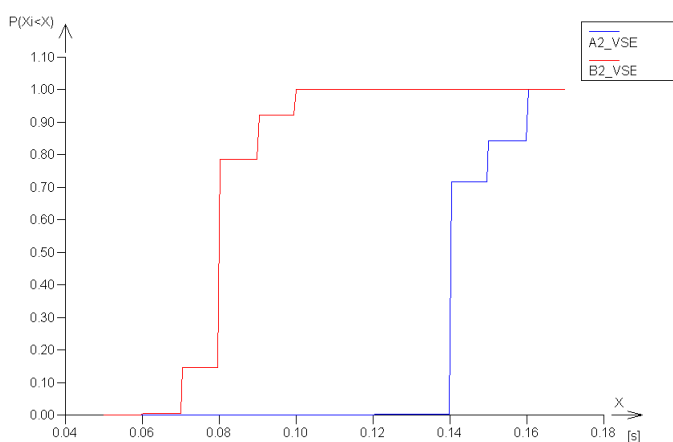


Graf 36: Jádřové odhady hustoty pravděpodobnosti pro výběry A2\_VSE-C2\_VSE<sup>17</sup>



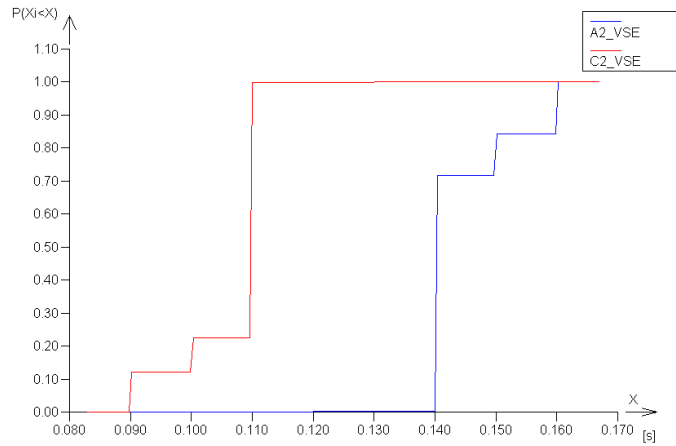
Graf 37: Jádřové odhady hustoty pravděpodobnosti pro výběry B2\_VSE-C2\_VSE<sup>17</sup>

Porovnání příslušných distribučních funkcí vždy pro dva porovnávané výběry ukazují graf 38, graf 39, graf 40.

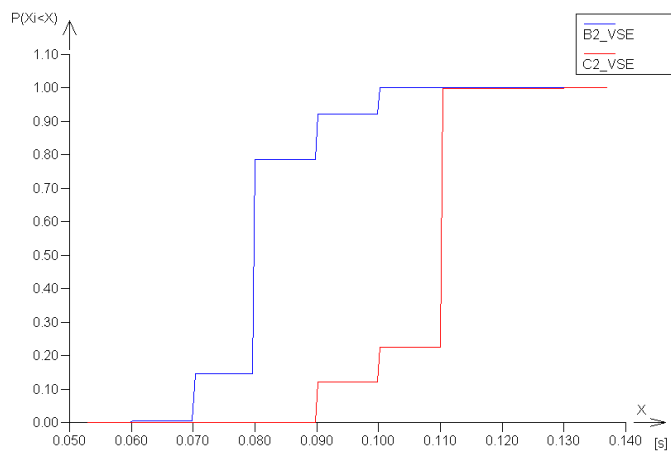


Graf 38: Empirické distribuční funkce pro porovnávané výběry A2\_VSE-B2\_VSE<sup>17</sup>

<sup>17</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

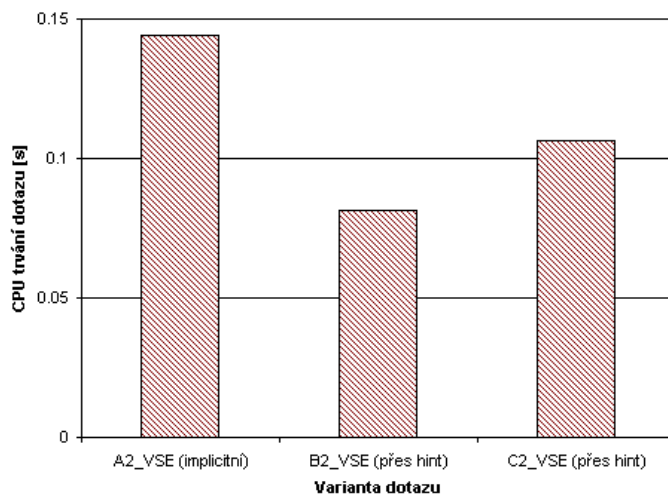


Graf 39: Empirické distribuční funkce pro porovnávané výběry A2\_VSE-C2\_VSE<sup>18</sup>



Graf 40: Empirické distribuční funkce pro porovnávané výběry B2\_VSE-C2\_VSE<sup>18</sup>

Z hlediska průměrné CPU rychlosti zpracování dotazu se jeví jako nejrychlejší varianta B2\_VSE, kde je hodnota přístupu nejmenší (graf 41).



Graf 41: Průměry CPU trvání dotazu A2\_VSE, B2\_VSE, C2\_VSE (1000 měření)

<sup>18</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.



## Shrnutí dílčích závěrů

V rámci modelu 1:N bez výběrového dotazu nad základní tabulkou (VSE) použil kompilátor z hlediska nákladů dotazu správně implicitně variantu A2\_VSE, kde je výše nákladů nejnižší. Co se týče kritéria průměrné CPU rychlosti zpracování dotazu, varianta A2\_VSE neposkytuje nejnižší naměřenou průměrnou hodnotu - tu vrací varianta B2\_VSE.

### 3.2.4 Model 1:N - výběr přes operátor LIKE

Pro testování modelu 1:N (LIKE) byla použita stejná pravidla jako při testování modelu 1:N (VŠE), pouze dotaz rodičovské tabulky je omezen přes operátor LIKE na vybraný počet řádek jednotlivých prodejů (6179 řádků).

#### Varianta A2\_LIKE - implicitní prováděcí plán

Pro dotaz 29 volí optimalizátor Oracle za definovaných podmínek při existenci výběru přes operátor LIKE vnořené spojení (tabulka 40).

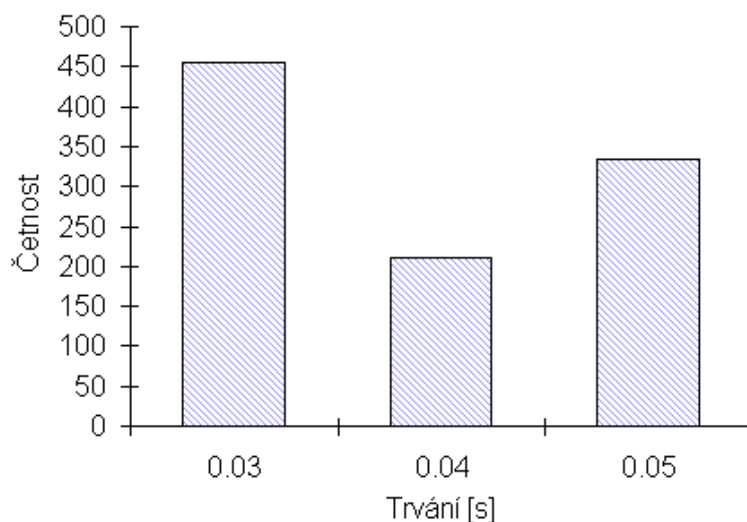
```
select a.item, b.period, b.date_, b.price
from ba.item_i a, ba.sale_i b
where
a.item like 'S%' and
b.item = a.item
```

Dotaz 29: Model 1:N, výběr LIKE, implicitní plán (vnořené spojení)

Tabulka 40: Prováděcí plán pro dotaz 29

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7862	468K	71 (8)	00:00:01
1	NESTED LOOPS		7862	468K	71 (8)	00:00:01
* 2	TABLE ACCESS FULL	SALE_I	7863	337K	68 (3)	00:00:01
* 3	INDEX UNIQUE SCAN	ITEM_I_IDX1	1	17	0 (0)	00:00:01

Náklady tohoto implicitního prováděcího plánu, kde optimalizátor použil vnořené spojení mezi tabulkami, jsou 71. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 42.



Graf 42: Histogram naměřených CPU trvání pro dotaz 29 (1000 měření)

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 41).

Tabulka 41: Varianta A2\_LIKE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{A2\_LIKE}$ (průměr)	0,0387 [s]
$s_{A2\_LIKE}$ (směrodatná odchylka)	0,0088 [s]

### Varianta B2\_LIKE - hintem vynucené hašované spojení

V této variantě je pro spojení tabulek použito hintem vynuceného hašovaného spojení (dotaz 30). Získaný prováděcí plán dotazu popisuje tabulka 42.

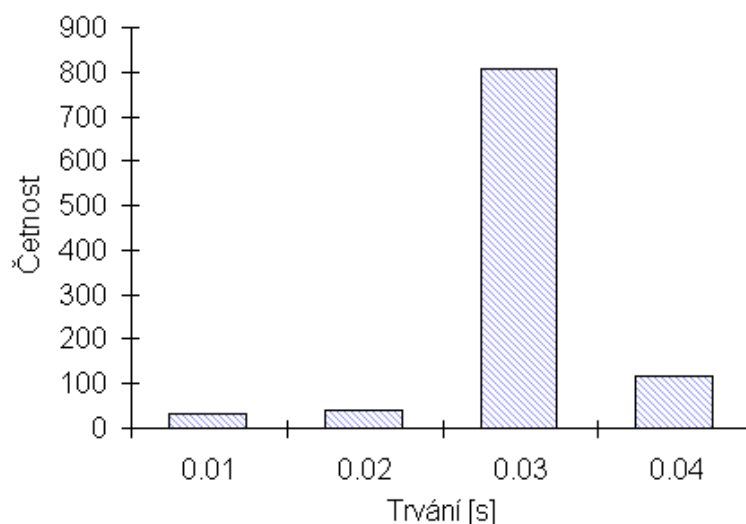
```
select /**+ USE_HASH(a,b) */ a.item, b.period, b.date_, b.price
from ba.item_i a, ba.sale_i b
where
a.item like 'S%' and
b.item = a.item
```

Dotaz 30: Model 1:N, výběr LIKE, vynucené hašované spojení

Tabulka 42: Prováděcí plán pro dotaz 30

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7862	468K	90 (4)	00:00:02
* 1	HASH JOIN		7862	468K	90 (4)	00:00:02
* 2	INDEX FAST FULL SCAN	ITEM_I_IDX1	6055	100K	21 (0)	00:00:01
* 3	TABLE ACCESS FULL	SALE_I	7863	337K	68 (3)	00:00:01

Náklady vynuceného hašovaného spojení jsou v rámci prováděcího plánu odhadnuty na 90. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 43.



Graf 43: Histogram naměřených CPU trvání pro dotaz 30 (1000 měření)

Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 43).

Tabulka 43: Varianta B2\_LIKE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{B2\_LIKE}$ (průměr)	0,0301 [s]
$s_{B2\_LIKE}$ (směrodatná odchylka)	0,0053 [s]

### Varianta C2\_LIKE - hintem vynucené tříděné slučovací spojení

Varianta C2\_LIKE upravuje původně použitý dotaz (dotaz 29) pomocí hintu tak, že pro spojení je použito tříděné slučovací spojení (dotaz 31). Získaný prováděcí plán dotazu ukazuje tabulka 44.

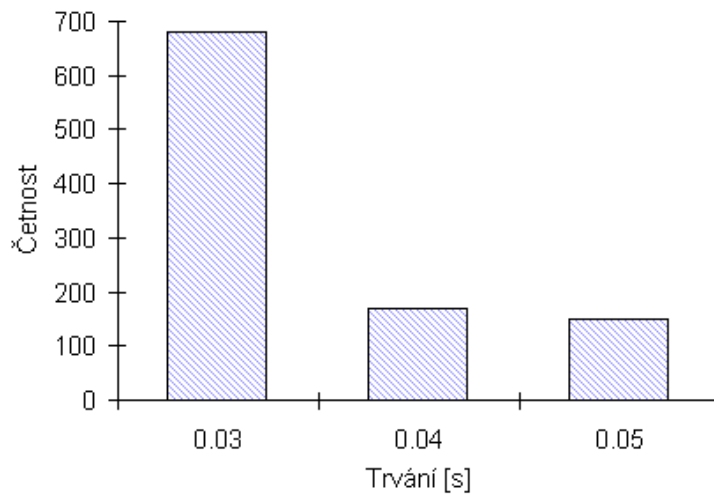
```
select /*+ USE_MERGE(a,b) */ a.item, b.period, b.date_, b.price
from ba.item_i a, ba.sale_i b
where
a.item like 'S%' and
b.item = a.item
```

Dotaz 31: Model 1:N, výběr LIKE, vynucené tříděné slučovací spojení

Tabulka 44: Prováděcí plán pro dotaz 31

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		7862	468K		182 (4)	00:00:03
1	MERGE JOIN		7862	468K		182 (4)	00:00:03
2	SORT JOIN		6055	100K		23 (9)	00:00:01
* 3	INDEX FAST FULL SCAN	ITEM_I_IDX1	6055	100K		21 (0)	00:00:01
* 4	SORT JOIN		7863	337K	888K	159 (3)	00:00:02
* 5	TABLE ACCESS FULL	SALE_I	7863	337K		68 (3)	00:00:01

Náklady vynuceného tříděného slučovacího spojení jsou v rámci příslušného získaného prováděcího plánu odhadnuty na 182. Získané CPU trvání dotazu v rámci 1000 provedených měření ukazuje graf 44.



Graf 44: Histogram naměřených CPU trvání pro dotaz 31 (1000 měření)

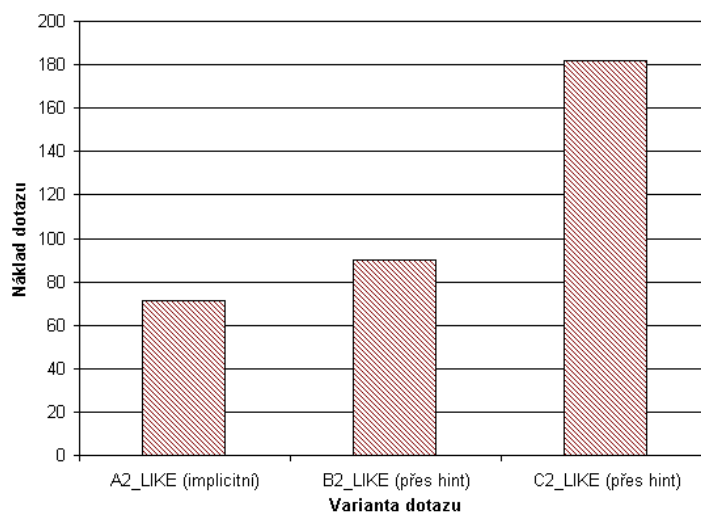
Ze získaných měření byly vypočítány tyto charakteristiky vzorku dat (tabulka 45).

Tabulka 45: Varianta C2\_LIKE - charakteristiky vzorku dat (1000 měření)

Charakteristika	Hodnota
$\bar{t}_{C2\_LIKE}$ (průměr)	0,0347 [s]
$s_{C2\_LIKE}$ (směrodatná odchylka)	0,0074 [s]

### Porovnání nákladů dotazu

U varianty 1:N LIKE je srovnáním nákladů dotazu zjevný rozdíl v relativně vysokých nákladech varianty C2\_LIKE. Graficky ukazuje odhad použitých nákladů graf 45.



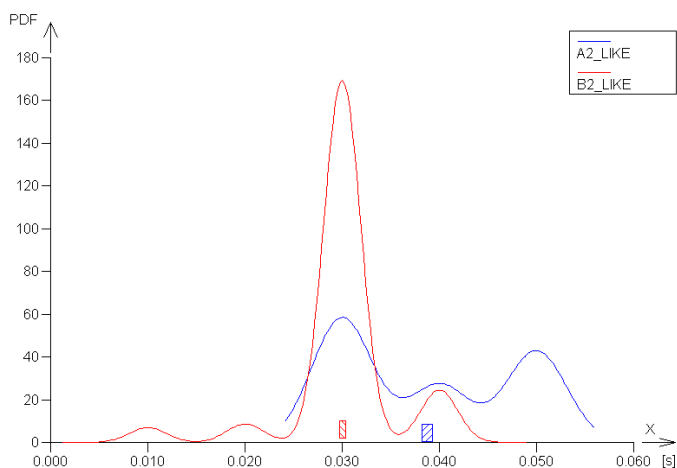
Graf 45: Porovnání odhadů nákladů dotazu pro A2\_LIKE, B2\_LIKE, C2\_LIKE

Optimalizátor správně implicitně zvolil použití plánu A2\_LIKE, který z hlediska kritéria nákladu dotazu vykazuje také nejnižší odhad nákladu. Varianta C2\_LIKE je v této popsané situaci nákladově nevyhovující.

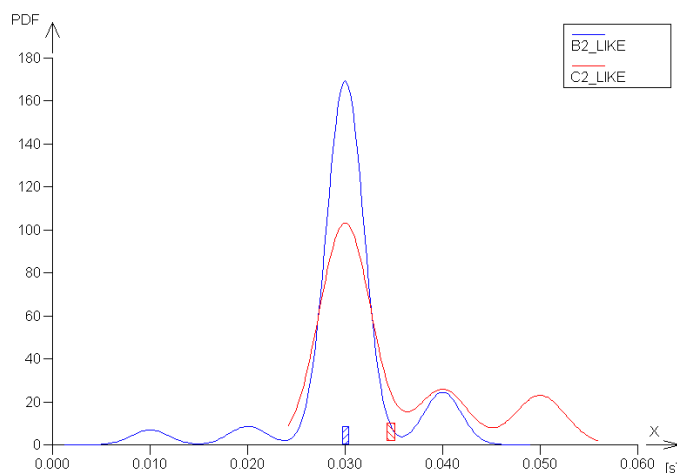
### Porovnání CPU trvání dotazu

Pro porovnání CPU trvání dotazu bylo provedeno 1000 měření pro dotaz každé testované varianty (A2\_LIKE, B2\_LIKE, C2\_LIKE). Porovnáním jednotlivých testovaných variant mezi sebou (A2\_LIKE-B2\_LIKE, A2\_LIKE-C2\_LIKE, B2\_LIKE-C2\_LIKE) ve statistickém nástroji QC Expert<sup>19</sup> bylo zjištěno, že všechny tři porovnávané varianty jsou rozdílné (příloha C).

Diference mezi porovnávanými výběry jsou znázorněné v grafu jádrových odhadů pravděpodobnosti pro vždy pro dva porovnávané výběry (graf 46, graf 47, graf 48).

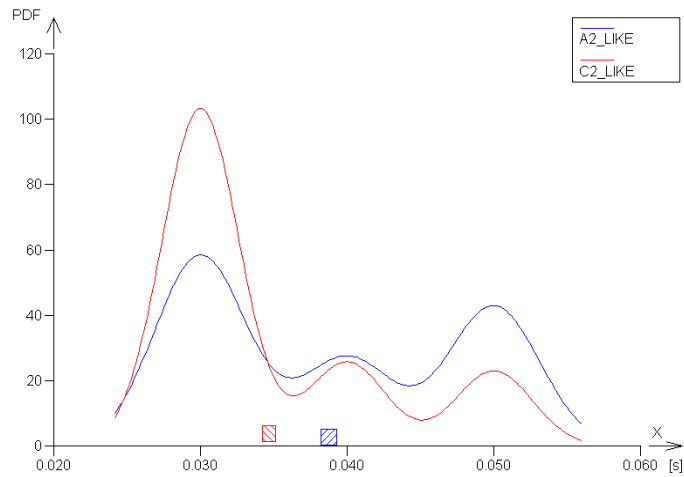


Graf 46: Jádrové odhady hustoty pravděpodobnosti pro výběry A2\_LIKE-B2\_LIKE<sup>19</sup>



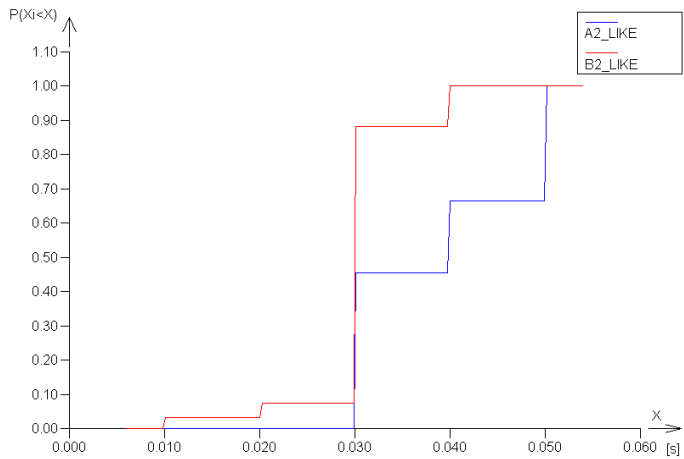
Graf 47: Jádrové odhady hustoty pravděpodobnosti pro výběry B2\_LIKE-C2\_LIKE<sup>19</sup>

<sup>19</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

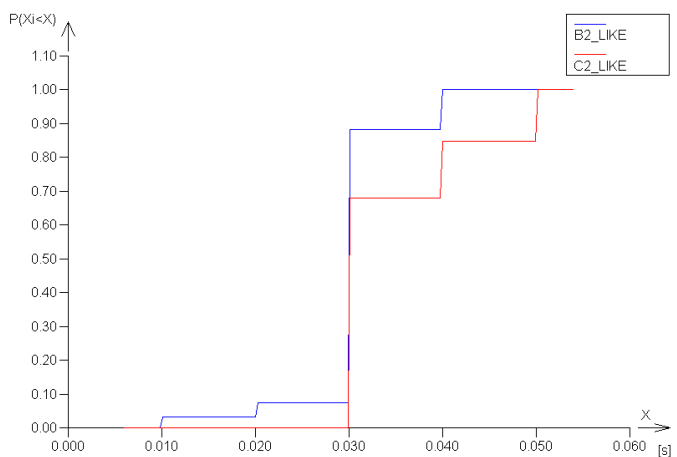


Graf 48: Jádrové odhady hustoty pravděpodobnosti pro výběry A2\_LIKE-C2\_LIKE<sup>20</sup>

Porovnání příslušných distribučních funkcí vždy pro dva porovnávané výběry ukazují graf 49, graf 50, graf 51.

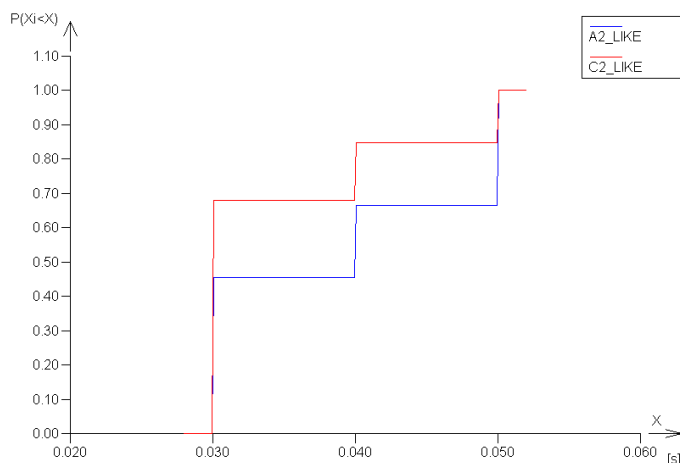


Graf 49: Empirické distribuční funkce pro porovnávané výběry A2\_LIKE-B2\_LIKE<sup>20</sup>



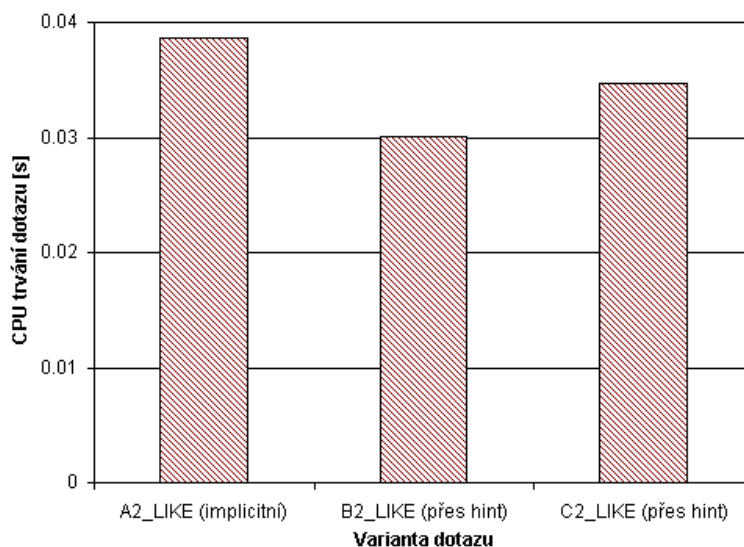
Graf 50: Empirické distribuční funkce pro porovnávané výběry B2\_LIKE-C2\_LIKE<sup>20</sup>

<sup>20</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.



Graf 51: Empirické distribuční funkce pro porovnávané výběry A2\_LIKE-C2\_LIKE<sup>21</sup>

U sledovaných variant v tomto modelu 1:N byla z hlediska průměrné CPU rychlosti dotazu nejrychlejší varianta B2\_LIKE (graf 52). V této variantě bylo hintem vynuceno hašované spojení. Rozdíly v průměrném čase jsou však mezi všemi testovanými variantami (LIKE) minimální.



Graf 52: Průměry CPU trvání dotazu A2\_VSE, B2\_VSE, C2\_VSE (1000 měření)

### Shrnutí dílčích závěrů

V rámci testovaného modelu 1:N při výběru LIKE optimalizátor z hlediska kritéria nákladu dotazu správně zvolil vnořené spojení pro propojení obou tabulek (A2\_LIKE). Varianta B2\_LIKE je nákladově pouze o něco méně výhodnější nežli varianta A2\_LIKE. U varianty C2\_LIKE s hintem vynuceným tříděným slučovací spojováním odhadl optimalizátor přibližně dvojnásobně vysoké náklady vzhledem k implicitně použité A2\_LIKE.

<sup>21</sup> QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE.

Z hlediska průměrné CPU rychlosti dotazu jsou rozdíly mezi variantami zanedbatelné, z hlediska průměrného trvání se nepatrně jeví výhodnější varianta B2\_VSE, kde je pro spojení tabulek užito hašované spojení.



## Závěr

Pomocí hintů lze výrazně ovlivnit podobu optimalizátorem navrženého prováděcího plánu SQL dotazu. Oracle poskytuje několik skupin hintů, s nimiž lze velmi dobře dynamicky pracovat zejména se zvoleným módem optimalizátoru, volbou přístupové metody k tabulce, rovněž lze významně ovlivňovat pořadí a způsob spojování více tabulek.

Z prováděných měření lze říci, že optimalizátor Oracle v naprosté většině případů volil velmi dobře implicitně navržený prováděcí plán, který se jevil optimální v porovnání s ostatními variantami (které byly vynuceny pomocí hintů), a to jak ze sledovaného kritéria výše nákladů dotazu, tak i z kritéria CPU trvání dotazu.

V rámci testování byla pozorována jediná významnější výjimka z tohoto tvrzení: u testování změny módu optimalizátoru z ALL\_ROWS na FIRST\_ROWS nad indexovanou tabulkou zvolil optimalizátor použití *přístupu přes index*, i když tímto způsobem došlo ke zvýšení nákladů dotazu. Pomocí hintu bylo možné dotaz upravit tak, aby celkové odhadnuté náklady dotazu byly nižší.

Použití hintů se ukázalo nákladově velmi výhodné při změně módu optimalizátoru na FIRST\_ROWS (oproti ALL\_ROWS), kdy se jedná o požadavek na co nejrychlejší blokové načtení daného počtu řádků. Lze si velmi dobře představit využití tohoto způsobu zpracování dotazu zejména u interaktivních aplikací OLTP systémů.

Značnou výhodou použití hintů je to, že ovlivňují pouze dotaz, ve kterém jsou použity - toho lze využít právě v situaci, kdy je prováděný dotaz či jeho data něčím extrémně specifický, což odpovídá i situaci, pro kterou jsou hinty určeny.

Cíl práce bylo splněn: základní struktura a použití hintů byla popsána. Rovněž bylo provedeno porovnání implicitního chování optimalizátoru a obdobných dotazů, které byly upraveny pomocí hintu. Lze konstatovat, že optimalizátor během testů z hlediska sledovaných kritérií volil velmi dobře prováděcí plán dotazu. Použití hintů lze doporučit ve výjimečných a velmi specifických situacích. Určitou nevýhodu lze vidět v tom, že pro aplikaci hintů v SQL dotazu musí být na straně vývoje dotazu určitá zkušenost, je třeba si pečlivě uvědomit, zda tvar dat bude i do budoucna odpovídat současné podobě - zda se nebude s přibývajícím časem výrazným způsobem měnit.

## Seznam použitých zkratk

CBO	Optimalizátor dle nákladů dotazu (angl. Cost-Based Optimizer)
CPU	Centrální procesorová jednotka (angl. Central processor unit)
OLAP	Analytické zpracování dat (angl. Online analytical processing)
OLTP	Online transakční zpracování dat (angl. Online transaction processing)
ORDBMS	Objektově-relační databázový systém (angl. Object-relational database management system)
SEQUEL	Strukturovaný anglický dotazovací jazyk (angl. Structured english query language)
SQL	Strukturovaný dotazovací jazyk (angl. Structured query language)

## Seznam použité literatury

- [1] ALAPATI, S. *Expert Oracle Database 10g Administration*, APress, Berkeley, 2005. 1305 s. ISBN 1-59059-451-7.
- [2] CYRAN, M. *Oracle 8i - Designing and Tuning for Performance*, Release 2 (8.1.6), Redwood City, Oracle Corporation, 1999. 646 s. A76992-01.
- [3] FOOT, CH. *10g Optimizer Statistics Gathering* [online]. DBAZine.com, 2006. [cit. 2010-03-20]. Dostupné z:  
<http://www.dbazine.com/blogs/blog-cf/chrisfoot/blogentry.2006-02-11.0939245208>
- [4] CHAN, I. *Oracle Database Performance Tuning Guide*, 10g Release 2 (10.2). 1th ed., Redwood City, Oracle Corporation, 2005. 474 s. B14211-01.
- [5] CHEEVERS, S. *Oracle Database 10g Product Family* [online]. [cit. 2009-10-05]. Oracle, 2006. Dostupné z:  
[http://www.oracle.com/technology/products/database/oracle10g/pdf/twp\\_general\\_10gdb\\_product\\_family.pdf](http://www.oracle.com/technology/products/database/oracle10g/pdf/twp_general_10gdb_product_family.pdf)
- [6] KYTE, T. *Expert One-on-One Oracle*. Berkeley: Apress, 2003. 1544 s. ISBN 1590592433.
- [7] LORENTZ, D. *Oracle Database SQL Reference*, 10g Release 2 (10.2), Oracle, 2005. 1428 s. B14200-02.
- [8] MCCULLOUGH-DIETER, C. *Mistrovství v Oracle 8*. 952 s. 1. vyd. Computer Press, 1999. ISBN 80-7226-204-1.
- [9] MELOUN, M., MILITKÝ, J. *Statistické zpracování experimentálních dat*. 1. vyd. Praha: PLUS spol. s.r.o., 1994. ISBN 80-85297-56-6.
- [10] *Nested loops, Hash join and Sort Merge joins – difference ?* [online]. [cit. 2009-10-21]. Dostupné z:  
<http://oracle-online-help.blogspot.com/2007/03/nested-loops-hash-join-and-sort-merge.html>

- [11] *Optimizer\_mode – ALL\_ROWS or FIRST\_ROWS ?* [online]. [cit. 2009-10-01].  
Dostupné z:  
<http://oracle-online-help.blogspot.com/2007/03/optimizermode-allrows-or-firstrows.html>
- [12] POWELL, G. *Oracle High Performance Tuning for 9i and 10g*, Digital Press, 2004. ISBN 1555583059.

## Seznam obrázků

Obrázek 1: Oracle zpracování SQL dotazu - převzato z [2] .....	8
Obrázek 2: Jednotlivé kroky hašovaného spojení - převzato z [2] .....	11
Obrázek 3: Jednotlivé kroky vnořeného spojení - převzato z [2] .....	12
Obrázek 4: Jednotlivé kroky tříděného slučovacího spojení - převzato z [2] .....	13
Obrázek 5: Syntaxe použití hintů v Oracle - převzato z [2] .....	15

## Seznam tabulek

Tabulka 1: Prováděcí plán pro dotaz 3 .....	20
Tabulka 2: Prováděcí plán pro dotaz 4 .....	20
Tabulka 3: Prováděcí plán pro dotaz 5 .....	20
Tabulka 4: Prováděcí plán pro dotaz 6 .....	21
Tabulka 5: Prováděcí plán pro dotaz 7 .....	21
Tabulka 6: Prováděcí plán pro dotaz 8 .....	22
Tabulka 7: Prováděcí plán, 30% vyhovujících řádků, bez statistik .....	23
Tabulka 8: Prováděcí plán, 30% vyhovujících řádků, bez statistik, vynucené FULL....	23
Tabulka 9: Prováděcí plán, 50% vyhovujících řádků, bez statistik .....	24
Tabulka 10: Prováděcí plán, 50% vyhovujících řádků, bez statistik, vynucené FULL..	24
Tabulka 11: Prováděcí plán pro dotaz 9 .....	26
Tabulka 12: Prováděcí plán pro dotaz 10 .....	26
Tabulka 13: Prováděcí plán pro dotaz 11 .....	27
Tabulka 14: Prováděcí plán pro dotaz 12 .....	27
Tabulka 15: Prováděcí plán pro dotaz 13 .....	28
Tabulka 16: Prováděcí plán pro dotaz 14 .....	28
Tabulka 17: Prováděcí plán pro dotaz 15 .....	30
Tabulka 18: Prováděcí plán pro dotaz 16 .....	30
Tabulka 19: Prováděcí plán pro dotaz 17 .....	31
Tabulka 20: Prováděcí plán pro dotaz 18 .....	31
Tabulka 21: Prováděcí plán pro dotaz 19 .....	32
Tabulka 22: Prováděcí plán pro dotaz 20 .....	33
Tabulka 23: Varianta A1_VSE - charakteristiky vzorku dat (1000 měření).....	34
Tabulka 24: Prováděcí plán pro dotaz 21 .....	34
Tabulka 25: Varianta B1_VSE - charakteristiky vzorku dat (1000 měření).....	35
Tabulka 26: Prováděcí plán pro dotaz 22 .....	36
Tabulka 27: Varianta C1_VSE - charakteristiky ze vzorku dat (1000 měření).....	36
Tabulka 28: Prováděcí plán pro dotaz 23 .....	41
Tabulka 29: Varianta A1_LIKE - charakteristiky vzorku dat (1000 měření) .....	41
Tabulka 30: Prováděcí plán pro dotaz 24 .....	42
Tabulka 31: Varianta B1_LIKE - charakteristiky vzorku dat (1000 měření) .....	42
Tabulka 32: Prováděcí plán pro dotaz 25 .....	43
Tabulka 33: Varianta C1_LIKE - charakteristiky vzorku dat (1000 měření) .....	43
Tabulka 34: Prováděcí plán pro dotaz 26 .....	48

Tabulka 35: Varianta A2_VSE - charakteristiky vzorku dat (1000 měření).....	48
Tabulka 36: Prováděcí plán pro dotaz 27 .....	49
Tabulka 37: Varianta B2_VSE - charakteristiky vzorku dat (1000 měření).....	49
Tabulka 38: Prováděcí plán pro dotaz 28 .....	50
Tabulka 39: Varianta C2_VSE - charakteristiky získané ze vzorku dat (1000 měření)	50
Tabulka 40: Prováděcí plán pro dotaz 29 .....	54
Tabulka 41: Varianta A2_LIKE - charakteristiky vzorku dat (1000 měření) .....	55
Tabulka 42: Prováděcí plán pro dotaz 30 .....	55
Tabulka 43: Varianta B2_LIKE - charakteristiky vzorku dat (1000 měření) .....	56
Tabulka 44: Prováděcí plán pro dotaz 31 .....	56
Tabulka 45: Varianta C2_LIKE - charakteristiky vzorku dat (1000 měření) .....	57

## Seznam grafů

Graf 1: Porovnání nákladů pro změnu index->full, 10%, bez statistik .....	22
Graf 2: Porovnání nákladů pro změnu index->full, 30%, bez statistik .....	23
Graf 3: Porovnání nákladů pro změnu index->full, 50%, bez statistik .....	24
Graf 4: Porovnání nákladů u implicitního přístupu přes index .....	25
Graf 5: Porovnání nákladů pro změnu full->index, 25%, existence statistik .....	26
Graf 6: Porovnání nákladů pro změnu full->index, 50%, existence statistik .....	27
Graf 7: Porovnání nákladů pro změnu full->index, 75%, existence statistik .....	29
Graf 8: Porovnání nákladů při explicitním přístupu přes index (hint) .....	29
Graf 9: Histogram naměřených CPU trvání pro dotaz 20 (1000 měření).....	34
Graf 10: Histogram naměřených CPU trvání pro dotaz 21 (1000 měření).....	35
Graf 11: Histogram naměřených CPU trvání pro dotaz 22 (1000 měření).....	36
Graf 12: Porovnání odhadů nákladů dotazu pro varianty A1_VSE,B1_VSE,C1_VSE.	37
Graf 13: Jádrové odhady hustoty pravděpodobnosti pro výběry A1_VSE-B1_VSE ....	38
Graf 14: Jádrové odhady hustoty pravděpodobnosti pro výběry A1_VSE-C1_VSE ....	38
Graf 15: Jádrové odhady hustoty pravděpodobnosti pro výběry B1_VSE-C1_VSE ....	38
Graf 16: Empirické distribuční funkce pro porovnávané výběry A1_VSE-B1_VSE .....	39
Graf 17: Empirické distribuční funkce pro porovnávané výběry A1_VSE-C1_VSE .....	39
Graf 18: Empirické distribuční funkce pro porovnávané výběry (B1_VSE-C1_VSE) ...	39
Graf 19: Průměry CPU trvání dotazu A1_VSE,B1_VSE,C1_VSE (1000 měření).....	40
Graf 20: Histogram naměřených CPU trvání pro dotaz 23 (1000 měření).....	41
Graf 21: Histogram naměřených CPU trvání pro dotaz 24 (1000 měření).....	42
Graf 22: Histogram naměřených CPU trvání pro dotaz 22 (1000 měření).....	43
Graf 23: Porovnání odhadů nákladů dotazu pro varianty A1_LIKE,B1_LIKE,C1_LIKE	44
Graf 24: Jádrové odhady hustoty pravděpodobnosti pro výběry A1_LIKE-B1_LIKE ...	45
Graf 25: Jádrové odhady hustoty pravděpodobnosti pro výběry A1_LIKE-C1_LIKE ...	45
Graf 26: Jádrové odhady hustoty pravděpodobnosti pro výběry B1_LIKE-C1_LIKE ...	45
Graf 27: Empirické distribuční funkce pro porovnávané výběry A1_LIKE-B1_LIKE ....	46
Graf 28: Empirické distribuční funkce pro porovnávané výběry A1_LIKE-C1_LIKE ....	46
Graf 29: Empirické distribuční funkce pro porovnávané výběry B1_LIKE-C1_LIKE ....	46
Graf 30: Průměry CPU trvání dotazu A1_LIKE,B1_LIKE,C1_LIKE (1000 měření).....	47
Graf 31: Histogram naměřených CPU trvání pro dotaz 26 (1000 měření).....	48
Graf 32: Histogram naměřených CPU trvání pro dotaz 27 (1000 měření).....	49
Graf 33: Histogram naměřených CPU trvání pro dotaz 28 (1000 měření).....	50
Graf 34: Porovnání odhadů nákladů dotazu pro varianty A2_VSE,B2_VSE,C2_VSE.	51



Graf 35: Jádrové odhady hustoty pravděpodobnosti pro výběry A2_VSE-B2_VSE ....	51
Graf 36: Jádrové odhady hustoty pravděpodobnosti pro výběry A2_VSE-C2_VSE ....	52
Graf 37: Jádrové odhady hustoty pravděpodobnosti pro výběry B2_VSE-C2_VSE ....	52
Graf 38: Empirické distribuční funkce pro porovnávané výběry A2_VSE-B2_VSE .....	52
Graf 39: Empirické distribuční funkce pro porovnávané výběry A2_VSE-C2_VSE .....	53
Graf 40: Empirické distribuční funkce pro porovnávané výběry B2_VSE-C2_VSE .....	53
Graf 41: Průměry CPU trvání dotazu A2_VSE,B2_VSE,C2_VSE (1000 měření).....	53
Graf 42: Histogram naměřených CPU trvání pro dotaz 29 (1000 měření).....	55
Graf 43: Histogram naměřených CPU trvání pro dotaz 30 (1000 měření).....	56
Graf 44: Histogram naměřených CPU trvání pro dotaz 31 (1000 měření).....	57
Graf 45: Porovnání odhadů nákladů dotazu pro A2_LIKE, B2_LIKE, C2_LIKE .....	57
Graf 46: Jádrové odhady hustoty pravděpodobnosti pro výběry A2_LIKE-B2_LIKE ...	58
Graf 47: Jádrové odhady hustoty pravděpodobnosti pro výběry B2_LIKE-C2_LIKE ...	58
Graf 48: Jádrové odhady hustoty pravděpodobnosti pro výběry A2_LIKE-C2_LIKE ...	59
Graf 49: Empirické distribuční funkce pro porovnávané výběry A2_LIKE-B2_LIKE ....	59
Graf 50: Empirické distribuční funkce pro porovnávané výběry B2_LIKE-C2_LIKE ....	59
Graf 51: Empirické distribuční funkce pro porovnávané výběry A2_LIKE-C2_LIKE ....	60
Graf 52: Průměry CPU trvání dotazu A2_VSE,B2_VSE,C2_VSE (1000 měření).....	60

## Seznam dotazů

Dotaz 1: Výpočet a uložení statistiky nad tabulkou ba.account_i1 .....	14
Dotaz 2: Ukázka použití hintů v dotazu - zákaz využívání indexů .....	15
Dotaz 3: Požadavek na zobrazení 52 % procent dat z tabulky (bez statistik) .....	19
Dotaz 4: Požadavek na zobrazení 53 % procent dat z tabulky (bez statistik) .....	20
Dotaz 5: Požadavek na zobrazení 15 % procent dat z tabulky (statistiky) .....	20
Dotaz 6: Požadavek na zobrazení 16 % procent dat z tabulky (statistiky) .....	21
Dotaz 7: Požadavek na zobrazení 10 % procent dat z tabulky .....	21
Dotaz 8: Požadavek na zobrazení 10 % procent dat z tabulky - aplikace hintu FULL ..	22
Dotaz 9: Požadavek na zobrazení 25 % procent dat z tabulky .....	25
Dotaz 10: Požadavek na zobrazení 25 % procent dat z tabulky, hint INDEX .....	26
Dotaz 11: Požadavek na zobrazení 50 % procent dat z tabulky .....	27
Dotaz 12: Požadavek na zobrazení 50 % procent dat z tabulky, hint INDEX .....	27
Dotaz 13: Požadavek na zobrazení 75 % procent dat z tabulky .....	28
Dotaz 14: Požadavek na zobrazení 75 % procent dat z tabulky, hint INDEX .....	28
Dotaz 15: Dotaz s výběrem z neindexované tabulky, mód ALL_ROWS .....	30
Dotaz 16: Dotaz s výběrem z neindexované tabulky, mód FIRST_ROWS, 100 .....	30
Dotaz 17: Dotaz s výběrem z indexované tabulky, mód ALL_ROWS .....	31
Dotaz 18: Dotaz s výběrem z indexované tabulky, mód FIRST_ROWS, 100 .....	31
Dotaz 19: Návrh snížení nákladů u výběru z indexované tabulky, FIRST_ROWS, 100 ..	32
Dotaz 20: Model 1:1, všechny věty, implicitní plán .....	33
Dotaz 21: Model 1:1, všechny věty, vynucené vnořené spojení .....	34
Dotaz 22: Model 1:1, všechny věty, vynucené tříděné slučovací spojení .....	35
Dotaz 23: Model 1:1, výběr LIKE, implicitní plán .....	40
Dotaz 24: Model 1:1, výběr LIKE, vynucené vnořené spojení .....	41
Dotaz 25: Model 1:1, výběr LIKE, vynucené tříděné slučovací spojení .....	42
Dotaz 26: Model 1:N, všechny věty, implicitní plán (vnořené spojení) .....	47
Dotaz 27: Model 1:N, všechny věty, vynucené hašované spojení .....	48
Dotaz 28: Model 1:N, všechny věty, vynucené tříděné slučovací spojení .....	49
Dotaz 29: Model 1:N, výběr LIKE, implicitní plán (vnořené spojení) .....	54
Dotaz 30: Model 1:N, výběr LIKE, vynucené hašované spojení .....	55
Dotaz 31: Model 1:N, výběr LIKE, vynucené tříděné slučovací spojení .....	56



## Seznam příloh

- A. Použité softwarové nástroje
- B. Testovací data
- C. Výsledky porovnání dvou výběru z nástroje QC.Expert
- D. Obsah DVD s elektronickou formou dat

## Příloha A

### Použité softwarové nástroje

Testování probíhalo na stanici HP COMPAQ, Intel Core 2 CPU, 1.86 GHz, 3 GB RAM, 250 GB HD, Windows XP, Service Pack 2.

Jako databáze byla použita:

- Oracle Database 10g Express Edition Release 10.2.0.1.0 (dostupná volně na webu na adrese <http://www.oracle.com/technology/software/products/database/xe/htdocs/102xewinsoft.html>)

K pokládání dotazů, zobrazování jejich výsledků a k rozborům nákladů každého z nich byly použity nástroje:

- pro Oracle klient Toad verze 9.5 společnosti Quest Software.

K získání 1000 měření CPU trvání jednotlivých dotazů byly použity vlastní SQL skripty, které jsou součástí přiloženého CD (příloha D).

Pro statistická porovnání vzorků, výpočty jejich průměrů a směrodatných odchylek a výstupy statistických grafů byl použit softwarový nástroj QC.Expert (ADSTAT) firmy TriloByte STATISTICAL SOFTWARE ([www.trilobyte.cz](http://www.trilobyte.cz)).

K vyjádření trasovacích charakteristik dotazů byl použit nástroj Oracle TKPROF.

Popis datových tabulek, na kterých probíhalo testování analytických funkcí, je uveden v příloze B.

## **Důležité parametry nastaveného prostředí Oracle**

<b>Název parametru</b>	<b>Hodnota parametru</b>
DB_BLOCK_SIZE	8192
DB_FILE_MULTIBLOCK_READ_COUNT	128
COMPATIBLE	10.2.0.1.0
OPEN_CURSORS	300
SGA_TARGET	805306368
PGA_AGGREGATE_TARGET	268435456
PROCESSES	40
SESSIONS	49

## Příloha B

### Testovací data

Pro testování byla použita tato testovací data:

#### Tabulka ACCOUNT, ACCOUNT\_I1

Tabulka ACCOUNT představuje řádky prodejů za jednotlivé období. ACCOUNT\_I1 je indexovaná verze tabulky ACCOUNT.

Tabulka: Struktura databázové tabulky ACCOUNT (ACCOUNT\_I1)

Název sloupce	Typ	Popis
PERIOD	VARCHAR2(6)	Období ve tvaru YYYYMM.
DATE_	DATE	Datum prodeje.
TASK	VARCHAR2(3)	Číslo účetní úlohy.
SYN	VARCHAR2(3)	Syntetický účet.
ANL	VARCHAR2(6)	Analytický účet.
MD_D	VARCHAR2(2)	Účetní strana (MD/D)
TRANSACTION	VARCHAR2(2)	Druh pohybu.
ITEM	VARCHAR2(30)	Číslo položky.
PRICE_TOTAL	NUMBER	Celková cena.
VALUE	NUMBER	Pomocná číselná hodnota.

Tabulka: Index tabulky ACCOUNT\_I

Název indexu	Typ	Sloupce tvořící index
ACCOUNT_I1_VALUE	neunikátní	VALUE

Počet řádků tabulky: 1 000 000.

#### Tabulka ITEM\_I

Tabulka ITEM\_I představuje katalog skladových položek.

Tabulka: Struktura databázové tabulky ITEM\_I

Název sloupce	Typ	Popis
ITEM	VARCHAR2(30)	Číslo položky.
ITEM_NAME	VARCHAR2(60)	Označení položky.

Tabulka: Indexy tabulky ITEM\_I

Název indexu	Typ	Sloupce tvořící index
ITEM_I_IDX1	Unikátní	ITEM

Počet řádků tabulky: 24 361.

## Tabulka SALE\_I

Tabulka SALE\_I představuje řádky prodejů za jednotlivé období.

Tabulka: Struktura databázové tabulky SALE\_I

Název sloupce	Typ	Popis
PERIOD	VARCHAR2(6)	Období ve tvaru YYYYMM.
ITEM	VARCHAR2(30)	Číslo položky.
SBJ_ID	NUMBER(10)	ID klíč odběratele.
SEGMENT	VARCHAR2(1)	Segment prodeje (A,B,C).
DATE_	DATE	Datum prodeje.
QUANTITY	NUMBER(9,3)	Množství, které bylo prodáno.
UNIT	VARCHAR2(2)	Měrná jednotka (KS).
PRICE	FLOAT(0)	Cena za měrnou jednotku.
PRICE_TOTAL	FLOAT(0)	Celková cena.

Tabulka: Indexy tabulky SALE\_I

Název indexu	Typ	Sloupce tvořící index
SALE_I_IDX_ITEM	neunikátní	ITEM
SALE_I_IDX_1	neunikátní	PERIOD
SALE_I_IDX_2	neunikátní	PERIOD, SEGMENT

Počet řádků tabulky: 38 850.



## Příloha C

### QC.Expert - model 1:1 - porovnání dvou výběrů "A1\_VSE - B1\_VSE"

Porovnání dvou výběrů

Hladina významnosti :	0,05	
Porovnávané sloupce :	A	B
Počet dat :	1000	1000
Průměr :	0,06266	0,12192
Směr. odchylka :	0,004663276332	0,007705385972
Rozptyl :	2,174614615E-005	5,937297297E-005

Test shody rozptylů

Poměr rozptylů :	2,730275635	
Počet stupňů volnosti :	999	999
Kritická hodnota :	1,10696448	
Závěr :	<b>Rozptyly jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	6,963020856E-055	

Robustní test shody rozptylů

Poměr rozptylů :	2,730275635	
Redukované stupně volnosti :	405	405
Kritická hodnota :	1,17317294	
Závěr :	<b>Rozptyly jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	1,929369173E-023	

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika :	208,0654807	
Počet stupňů volnosti :	1998	
Kritická hodnota :	1,961152015	
Závěr :	<b>Průměry jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	0	

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika :	208,0654807	
Redukované stupně volnosti :	1644	
Kritická hodnota :	1,961408017	
Závěr :	<b>Průměry jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	0	

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF :	1	
Kritická hodnota :	0,06073614619	
Závěr :	<b>Rozdělení jsou ROZDÍLNÁ</b>	

## QC.Expert - model 1:1 - porovnání dvou výběrů "A1\_VSE - C1\_VSE"

Porovnání dvou výběrů

Hladina významnosti :	0,05	
Porovnávané sloupce :	A	B
Počet dat :	1000	1000
Průměr :	0,06266	0,0971
Směr. odchylka :	0,004663276332	0,007787852763
Rozptyl :	2,174614615E-005	6,065065065E-005

Test shody rozptylů

Poměr rozptylů :	2,789029847	
Počet stupňů volnosti :	999	999
Kritická hodnota :	1,10696448	
Závěr :	<b>Rozptyly jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	4,715708486E-057	

Robustní test shody rozptylů

Poměr rozptylů :	2,789029847	
Redukované stupně volnosti :	494	494
Kritická hodnota :	1,15555491	
Závěr :	<b>Rozptyly jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	3,050957779E-029	

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika :	119,9797571	
Počet stupňů volnosti :	1998	
Kritická hodnota :	1,961152015	
Závěr :	<b>Průměry jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	0	

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika :	119,9797571	
Redukované stupně volnosti :	1634	
Kritická hodnota :	1,961416861	
Závěr :	<b>Průměry jsou ROZDÍLNÉ</b>	
Pravděpodobnost :	0	

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF :	1	
Kritická hodnota :	0,06073614619	
Závěr :	<b>Rozdělení jsou ROZDÍLNÁ</b>	

## QC.Expert - model 1:1 - porovnání dvou výběrů "B1\_VSE - C1\_VSE"

Porovnání dvou výběrů

Hladina významnosti :	0,05	
Porovnávané sloupce :	A	B
Počet dat :	1000	1000
Průměr :	0,12192	0,0971
Směr. odchylka :	0,007705385972	0,007787852763
Rozptyl :	5,937297297E-005	6,065065065E-005

Test shody rozptylů

Poměr rozptylů :	1,021519517	
Počet stupňů volnosti :	999	999
Kritická hodnota :	1,10696448	
Závěr :	<b>Rozptyly jsou SHODNÉ</b>	
Pravděpodobnost :	0,3396129341	

Robustní test shody rozptylů

Poměr rozptylů :	1,021519517	
Redukované stupně volnosti :	640	640
Kritická hodnota :	1,135413125	
Závěr :	<b>Rozptyly jsou SHODNÉ</b>	
Pravděpodobnost :	0,3632159257	

Test shody průměrů

pro SHODNÉ rozptyly	
t-statistika :	71,64211689
Počet stupňů volnosti :	1998
Kritická hodnota :	1,961152015
Závěr :	<b>Průměry jsou ROZDÍLNÉ</b>
Pravděpodobnost :	0

Test shody průměrů

pro ROZDÍLNÉ rozptyly	
t-statistika :	71,64211689
Redukované stupně volnosti :	1998
Kritická hodnota :	1,961152015
Závěr :	<b>Průměry jsou ROZDÍLNÉ</b>
Pravděpodobnost :	0

Test dobré shody rozdělení

dvouvýběrový K-S test	
Diference DF :	0,803
Kritická hodnota :	0,06073614619
Závěr :	<b>Rozdělení jsou ROZDÍLNÁ</b>

## QC.Expert - model 1:1 - porovnání dvou výběrů "A1\_LIKE - B1\_LIKE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,01768 0,03095  
Směr. odchylka : 0,007229790444 0,004267987314  
Rozptyl : 5,226986987E-005 1,821571572E-005

Test shody rozptylů

Poměr rozptylů : 2,869493062  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 5,206732729E-060

Robustní test shody rozptylů

Poměr rozptylů : 2,869493062  
Redukované stupně volnosti : 205 205  
Kritická hodnota : 1,251993435  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 7,685737915E-014

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika : 49,98282134  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika : 49,98282134  
Redukované stupně volnosti : 1620  
Kritická hodnota : 1,961429426  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF : 0,845  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:1 - porovnání dvou výběrů "A1\_LIKE - C1\_LIKE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,01768 0,02337  
Směr. odchylka : 0,007229790444 0,008855592669  
Rozptyl : 5,226986987E-005 7,842152152E-005

Test shody rozptylů

Poměr rozptylů : 1,500319816  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 7,761542862E-011

Robustní test shody rozptylů

Poměr rozptylů : 1,500319816  
Redukované stupně volnosti : 479 479  
Kritická hodnota : 1,158159198  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 4,509498295E-006

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika : 15,73942185  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 1,040794655E-052

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika : 15,73942185  
Redukované stupně volnosti : 1921  
Kritická hodnota : 1,961199664  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 1,356059502E-052

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF : 0,364  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:1 - porovnání dvou výběrů "B1\_LIKE - C1\_LIKE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,03095 0,02337  
Směr. odchylka : 0,004267987314 0,008855592669  
Rozptyl : 1,821571572E-005 7,842152152E-005

Test shody rozptylů

Poměr rozptylů : 4,305157302  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Robustní test shody rozptylů

Poměr rozptylů : 4,305157302  
Redukované stupně volnosti : 331 331  
Kritická hodnota : 1,193285663  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 1,584614135E-037

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika : 24,38355102  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika : 24,38355102  
Redukované stupně volnosti : 1439  
Kritická hodnota : 1,961613904  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF : 0,481  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:N - porovnání dvou výběrů "A2\_VSE - B2\_VSE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,14434 0,08141  
Směr. odchylka : 0,007564497447 0,007640043866  
Rozptyl : 5,722162162E-005 5,837027027E-005

Test shody rozptylů

Poměr rozptylů : 1,020073682  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou SHODNÉ**  
Pravděpodobnost : 0,347421028

Robustní test shody rozptylů

Poměr rozptylů : 1,020073682  
Redukované stupně volnosti : 327 327  
Kritická hodnota : 1,194575787  
Závěr : **Rozptyly jsou SHODNÉ**  
Pravděpodobnost : 0,3953649044

Test shody průměrů

pro SHODNÉ rozptyly  
t-statistika : 185,0947266  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test shody průměrů

pro ROZDÍLNÉ rozptyly  
t-statistika : 185,0947266  
Redukované stupně volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test dobré shody rozdělení

dvouvýběrový K-S test  
Diference DF : 1  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:N - porovnání dvou výběrů "A2\_VSE - C2\_VSE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,14434 0,10658  
Směr. odchylka : 0,007564497447 0,006939146327  
Rozptyl : 5,722162162E-005 4,815175175E-005

Test shody rozptylů

Poměr rozptylů : 1,188360123  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 0,002965602557

Robustní test shody rozptylů

Poměr rozptylů : 1,188360123  
Redukované stupně volnosti : 310 310  
Kritická hodnota : 1,200352459  
Závěr : **Rozptyly jsou SHODNÉ**  
Pravděpodobnost : 0,05959172953

Test shody průměrů

pro SHODNÉ rozptyly  
t-statistika : 116,3232543  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test shody průměrů

pro ROZDÍLNÉ rozptyly  
t-statistika : 116,3232543  
Redukované stupně volnosti : 1983  
Kritická hodnota : 1,961161007  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test dobré shody rozdělení

dvouvýběrový K-S test  
Diference DF : 0,998  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**



## QC.Expert - model 1:N - porovnání dvou výběrů "B2\_VSE - C2\_VSE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,08141 0,10658  
Směr. odchylka : 0,007640043866 0,006939146327  
Rozptyl : 5,837027027E-005 4,815175175E-005

Test shody rozptylů

Poměr rozptylů : 1,212214886  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 0,001097087389

Robustní test shody rozptylů

Poměr rozptylů : 1,212214886  
Redukované stupně volnosti : 276 276  
Kritická hodnota : 1,213577311  
Závěr : **Rozptyly jsou SHODNÉ**  
Pravděpodobnost : 0,05095043439

Test shody průměrů

pro SHODNÉ rozptyly  
t-statistika : 77,11937739  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test shody průměrů

pro ROZDÍLNÉ rozptyly  
t-statistika : 77,11937739  
Redukované stupně volnosti : 1980  
Kritická hodnota : 1,961162822  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test dobré shody rozdělení

dvouvýběrový K-S test  
Diference DF : 0,801  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:N - porovnání dvou výběrů "A2\_LIKE - B2\_LIKE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,03879 0,0301  
Směr. odchylka : 0,008804170229 0,005386930389  
Rozptyl : 7,751341341E-005 2,901901902E-005

Test shody rozptylů

Poměr rozptylů : 2,671124526  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 1,082253617E-052

Robustní test shody rozptylů

Poměr rozptylů : 2,671124526  
Redukované stupně volnosti : 468 468  
Kritická hodnota : 1,160151733  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 9,357367776E-026

Test shody průměrů

pro SHODNÉ rozptyly  
t-statistika : 26,62434025  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test shody průměrů

pro ROZDÍLNÉ rozptyly  
t-statistika : 26,62434025  
Redukované stupně volnosti : 1655  
Kritická hodnota : 1,961398412  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 0

Test dobré shody rozdělení

dvouvýběrový K-S test  
Diference DF : 0,428  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:N - porovnání dvou výběrů "B2\_LIKE - C2\_LIKE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,0301 0,03471  
Směr. odchylka : 0,005386930389 0,007427723142  
Rozptyl : 2,901901902E-005 5,517107107E-005

Test shody rozptylů

Poměr rozptylů : 1,901203863  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 3,611802648E-024

Robustní test shody rozptylů

Poměr rozptylů : 1,901203863  
Redukované stupně volnosti : 254 254  
Kritická hodnota : 1,223616601  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 1,816989483E-007

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika : 15,88804416  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 1,278214147E-053

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika : 15,88804416  
Redukované stupně volnosti : 1822  
Kritická hodnota : 1,96126685  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 2,466426551E-053

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF : 0,203  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## QC.Expert - model 1:N - porovnání dvou výběrů "A2\_LIKE - C2\_LIKE"

Porovnání dvou výběrů

Hladina významnosti : 0,05  
Porovnávané sloupce : A B

Počet dat : 1000 1000  
Průměr : 0,03879 0,03471  
Směr. odchylka : 0,008804170229 0,007427723142  
Rozptyl : 7,751341341E-005 5,517107107E-005

Test shody rozptylů

Poměr rozptylů : 1,404964811  
Počet stupňů volnosti : 999 999  
Kritická hodnota : 1,10696448  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 3,854542167E-008

Robustní test shody rozptylů

Poměr rozptylů : 1,404964811  
Redukované stupně volnosti : 712 712  
Kritická hodnota : 1,127942812  
Závěr : **Rozptyly jsou ROZDÍLNÉ**  
Pravděpodobnost : 2,799310359E-006

Test shody průměrů

pro SHODNÉ rozptyly

t-statistika : 11,20082707  
Počet stupňů volnosti : 1998  
Kritická hodnota : 1,961152015  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 2,758142417E-028

Test shody průměrů

pro ROZDÍLNÉ rozptyly

t-statistika : 11,20082707  
Redukované stupně volnosti : 1943  
Kritická hodnota : 1,961185665  
Závěr : **Průměry jsou ROZDÍLNÉ**  
Pravděpodobnost : 2,905839518E-028

Test dobré shody rozdělení

dvouvýběrový K-S test

Diference DF : 0,225  
Kritická hodnota : 0,06073614619  
Závěr : **Rozdělení jsou ROZDÍLNÁ**

## **Příloha D**

### **Obsah DVD s elektronickou formou dat**

1. Kompletní použitá databáze Oracle 10g Express (\data\oracle).
2. Získané naměřené hodnoty CPU trvání dotazů (\statistic\source\_data.xls).
3. SQL skripty použité k získání 1000 měření CPU trvání dotazů (\sql).
4. Výsledky porovnání dvou výběrů přes nástroj QC.Expert (\statistic\qcexpert\comparing).