

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

SEMESTRÁLNÍ PRÁCE NA
ARCHITEKTURY DATABÁZOVÝCH
SYSTÉMŮ

MATĚJ TRAKAL
Poslední úprava: 8. listopadu 2010

Obsah

1	Zadání	2
2	Vypracování	2
2.1	Dotaz 1: Zjistěte seznam pošt v okrese Pardubice	2
2.1.1	Postup vykonávání dotazů	2
2.1.2	Zrychlení dotazu	3
2.1.3	Postup vykonávání dotazů po optimalizaci	3
2.2	Dotaz 2: Najděte PSČ všech částí obcí v obci Hradec Králové . . .	4
2.2.1	Postup vykonávání dotazů	4
2.2.2	Zrychlení dotazu	4
2.2.3	Postup vykonávání dotazů po optimalizaci	5
2.3	Dotaz 3: Zjistěte, kolik procent domácností v okrese Kolín si nepřeje vkládat reklamní materiály	5
2.3.1	Postup vykonávání dotazů	6
2.3.2	Zrychlení dotazu	7
2.3.3	Postup vykonávání dotazů po optimalizaci	7

1 Zadání

Vybrané exekuční plány 3 dotazů zkopírujte jako obrázky do pdf souboru a odevzdejte do 9.11.2010 včetně přes STAG – odevzdávání prací – blok Exekuční plány – téma Skupina (den v týdnu, týden a hodina, kdy chodíte na cvičení), označení souboru `Prijmeni_Jmeno_dotazy_Posty.pdf`

Všechny 3 exekuční plány popište, aby z popisu bylo zřejmé:

- jaké druhy spojení a přístupů jsou použity (česky),
- jak jdou operace za sebou,
- jaké indexy se používají a k čemu.

Minimálně jeden popisovaný exekuční plán musí být nad dotazem, který spojuje minimálně 3 tabulky.

2 Vypracování

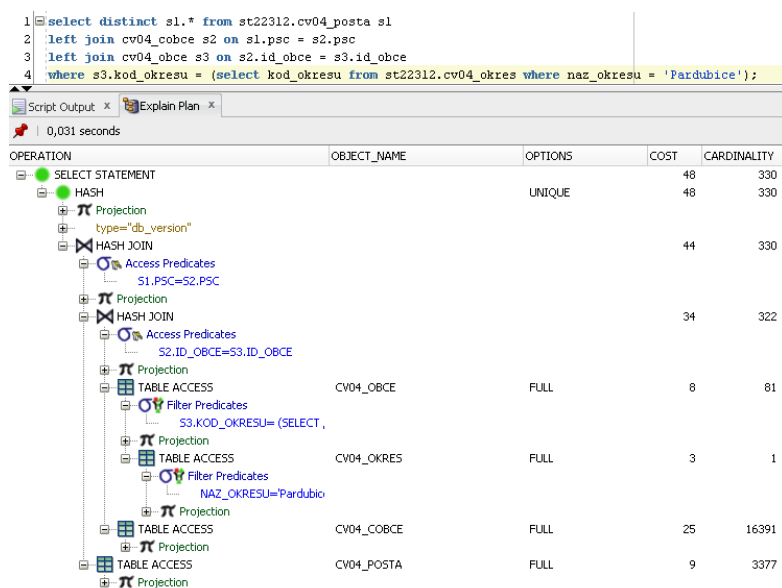
U popisků vynechávám v názvech tabulek a případných indexech nápis CV04, pro zjednodušení a přehlednost práce.

2.1 Dotaz 1: Zjistěte seznam pošt v okrese Pardubice

```
select distinct s1.* from st22312.cv04_posta s1
  left join cv04_cobce s2 on s1.psc = s2.psc
  left join cv04_obce s3 on s2.id_obce = s3.id_obce
where s3.kod_okresu = (
select kod_okresu from st22312.cv04_okres where naz_okresu = 'Pardubice'
);
```

2.1.1 Postup vykonávání dotazů

1. Celé prohledání tabulky `posta`
2. Prohledání celé tabulky `cobce`
3. Celé prohledání tabulky `okres` s filtrem na název okresu Pardubice
4. Hash spojení všech tabulek s ohledem na primární klíče



Obrázek 1: První dotaz před úpravou exekučního plánu

2.1.2 Zrychlení dotazu

Z obrázku 1 je patrné, že pro vykonávání `naz_okresu = 'Pardubice'` je použito plné prohledání tabulky. Vytvořením indexu nad sloupečkem `naz_okresu` se nemusí chodit pro výsledek do tabulky a lze využít indexu, což urychlí vykonání dotazu, viz obrázek 2.

2.1.3 Postup vykonávání dotazů po optimalizaci

1. Celé prohledání tabulky `posta`,
2. rychlé prohledání indexu `cobce`,
3. prohledání indexu s unikátními klíči `idx_naz_okresu`,
4. prohledání tabulky `okres` dle indexů řádků,
5. prohledání celé tabulky `obce`,
6. hash spojení všech tabulek.

```

1 select distinct s1.* from st22312.cv04_posta s1
2 left join cv04_cobce s2 on s1.psc = s2.psc
3 left join cv04_obce s3 on s2.id_obce = s3.id_obce
4 where s3.kod_okresu = (select kod_okresu from st22312.cv04_okres where naz_okresu = 'Pardubice');

```

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			33	330
HASH		UNIQUE	33	330
Projection				
Projection				
HASH JOIN			31	330
Access Predicates				
Projection				
HASH JOIN			21	322
Access Predicates				
Projection				
TABLE ACCESS	CV04_OBCE	FULL	8	81
Filter Predicates				
Projection				
TABLE ACCESS	CV04_OKRES	BY INDEX ROWID	1	1
Projection				
INDEX	CV04_IDX_NAZ_OKRESU	UNIQUE SCAN	0	1
Access Predicates				
Projection				
INDEX	CV04_IDX_COBCE	FAST FULL SCAN	12	16391
Projection				
TABLE ACCESS	CV04_POSTA	FULL	9	3377
Projection				

Obrázek 2: První dotaz po úpravě exekučního plánu

2.2 Dotaz 2: Najděte PSČ všech částí obcí v obci Hradec Králové

```

select s3.psc
from cv04_posta s3, cv04_cobce s2, cv04_obce s1
where s1.naz_obce = 'Hradec Králové'
and s1.id_obce = s2.id_obce
and s2.psc = s3.psc
order by s3.psc;

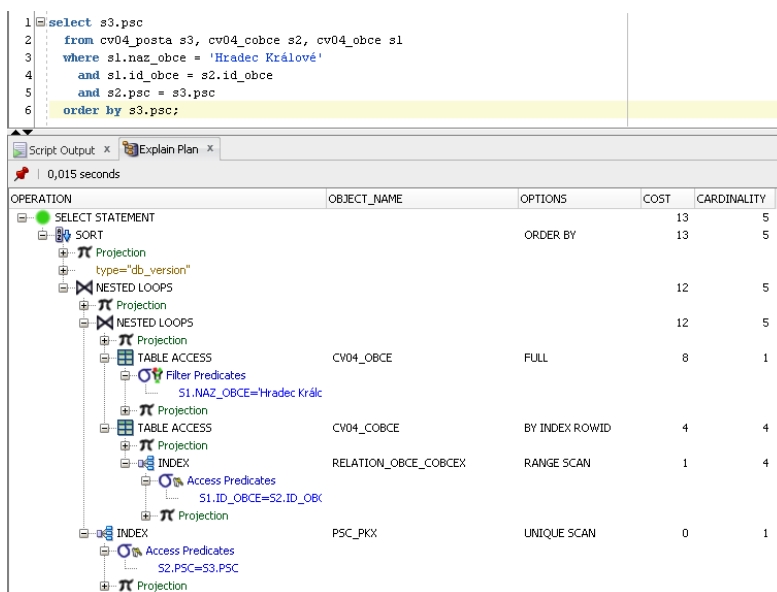
```

2.2.1 Postup vykonávání dotazů

1. Prohledání unikátního indexu `psc_pkx`,
2. prohledání indexu metodou range scan,
3. přístup do tabulky `cobce` dle indexu řádků,
4. prohledání celé tabulky `obce` s restrikcí na název obce Hradec Králové,
5. seřazení tabulky.

2.2.2 Zrychlení dotazu

Z obrázku 4 je patrné, že přidáním indexů nad sloupce, kde nastával samotný přístup do tabulky a její úplné prohledání, se výsledky zlepšily.



Obrázek 3: Druhý dotaz před úpravou exekučního plánu

2.2.3 Postup vykonávání dotazů po optimalizaci

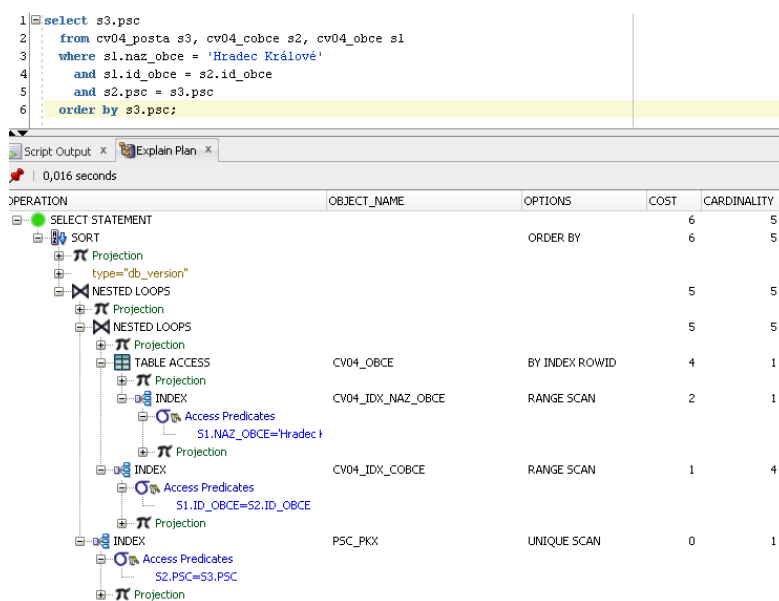
1. Prohledání unikátního indexu psc_pkx,
2. range scan prohledání indexu idx_cobce,
3. range scan nad indexem idx_naz_okresu,
4. prohledání tabulky obce dle indexů řádků,
5. seřazení tabulky.

2.3 Dotaz 3: Zjistěte, kolik procent domácností v okrese Kolín si nepřeje vkládat reklamní materiály

```

with ws1 as (
  select distinct s1.psc, s1.pocdom_rpm from cv04_posta s1
    left join cv04_cobce s2 on s1.psc = s2.psc
    left join cv04_obce s3 on s2.id_obce = s3.id_obce
  where s3.kod_okresu = (
    select kod_okresu from cv04_okres where naz_okresu = 'Kolín')
),
ws2 as (
  select count(*) celkem from cv04_schranky s1

```

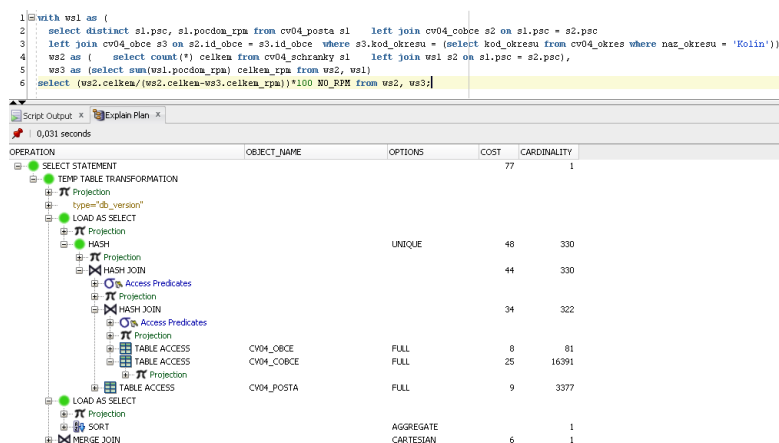


Obrázek 4: Druhý dotaz po úpravě exekučního plánu

```

left join ws1 s2 on s1.psc = s2.psc),
ws3 as (select sum(ws1.pocdom_rpm) celkem_rpm from ws2, ws1)
select (ws2.celkem / (ws2.celkem - ws3.celkem_rpm)) * 100 NO_RPM from ws2, ws3;

```



Obrázek 5: Třetí dotaz před úpravou exekučního plánu

2.3.1 Postup vykonávání dotazů

1. Úplné prohledání tabulky cobce,

2. úplné prohledání tabulky **obce** a jejich spojení,
3. úplné prohledání tabulky **posta**,
4. spojení těchto tabulek na unikátnost.
5. Druhou část dotazu pro zjednodušení neuvádím (vycházím z obrázků),
6. její seřazení.

2.3.2 Zrychlení dotazu

Z obrázku 6 je patrné, že přidáním indexu nad nejnáročnější událost (úplné procházení tabulky **cobce**) změnil její úplné procházení za rychlé procházení indexu, čímž klesne cena výpočtu.

2.3.3 Postup vykonávání dotazů po optimalizaci

1. Celé prohledání tabulky **posta**,
2. rychlé prohledání indexu **idx_cobce**,
3. úplné prohledání tabulky **obce** (což nechápu, jelikož je nad všemi sloupci index),
4. spojení dle unikátnosti záznamů.

```

1 with ws1 as ( select distinct s1.psc, s1.pocdom_rpa from cv04_posta s1 left join cv04_cobce s2 on s1.psc = s2.psc
2 left join cv04_obce s3 on s2.id_obce = s3.id_obce where s3.kod_okresu = (select kod_okresu from cv04_okres where naz_okresu = 'Kolin')),
3 ws2 as ( select count(*) ceiken from cv04_schranky s1 left join ws1 s2 on s1.psc = s2.psc),
4 ws3 as (select sum(ws1.pocdom_rpa) ceiken_rpa from ws2, ws1)
5 select (ws2.ceiken/(ws2.ceiken-ws3.ceiken_rpa))*100 NO_RPM from ws2, ws3;

```

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			63	1
TEMP TABLE TRANSFORMATION				
Projection				
LOAD AS SELECT				
HASH				
Projection				
HASH JOIN		UNIQUE	33	330
Projection			31	330
Access Predicates				
Projection				
HASH JOIN			21	322
Access Predicates				
Projection				
TABLE ACCESS	CV04_OBCE	FULL	8	81
INDEX	CV04_IDX_COBCE	FAST FULL SCAN	12	16391
TABLE ACCESS	CV04_POSTA	FULL	9	3377
LOAD AS SELECT				
MERGE JOIN		CARTESIAN	6	1

Obrázek 6: Třetí dotaz po úpravě exekučního plánu