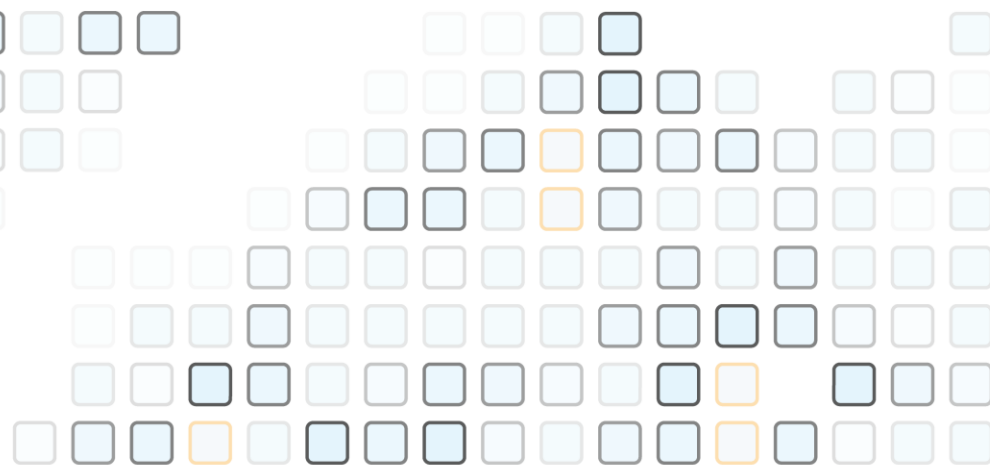


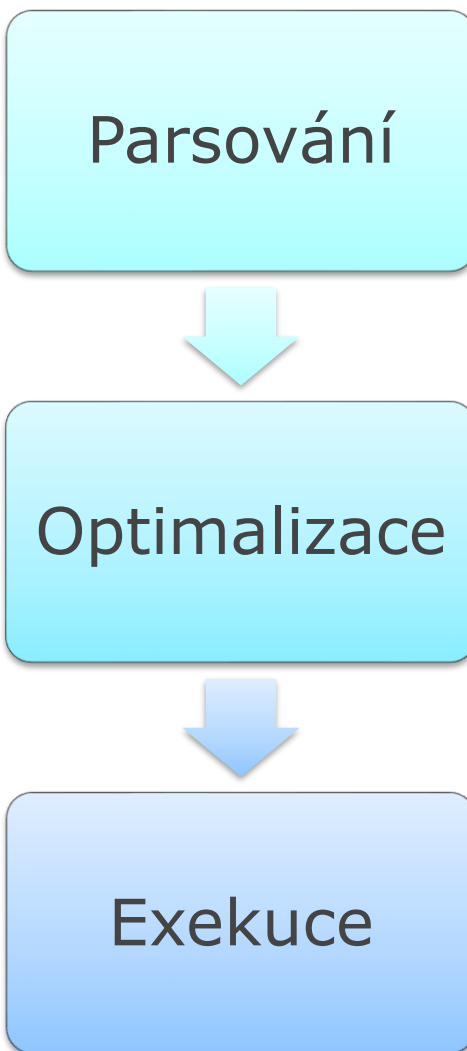
IS/STAG
is-stag.zcu.cz



Jak vlastně to SQL funguje

Petr Jiroušek

Jak to zhruba funguje



Parsování

Načtení do
PGA



Syntaktická
analýza



Sémantická
analýza

Optimalizace



Exekuce



„All in one“



Příklady - použité tabulky:

STUDENTI: osobidno, os_cislo, ...

OSOBY: prijmeni, jmeno, rod_cislo, osobidno, statidno_obca, stav ...

PRERUSENI: os_cislo, ...

STUDENTI_VYJEZDY: os_cislo, ...

PRED_VARIANTY: prac_zkr, zkr_predm, rok_varianty, ...

UCITELE_PREDMETU: prac_zkr, zkr_predm, rok_varianty, ...

- Náš vzorový select:

```
Select o.prijmeni,o.jmeno,s.os_cislo  
from osoby o, studenti s  
where o.statidno_obca<>203  
and o.osobidno=s.osobidno  
order by o.prijmeni
```


Sémantická analýza

- Budeme hledat v tzv. data dictionary (row cache):
 - Existuje tabulka osoby?
 - Má na tabulku osoby uživatel právo select?
 - Existuje tabulka studenti?
 - Má na tabulku studenti uživatel právo select?
 - Existuje v tabulce osoby sloupec prijmeni?
 - Existuje v tabulce osoby sloupec jmeno?
 - Existuje v tabulce osoby sloupec statidno_obca?
 - Existuje v tabulce studenti sloupec os_cislo?
 - Jakého datového typu je statidno_obca?

Sémantická analýza

- Existence objektů
 - ORA-942: Tabulka nebo pohled neexistuje
- Práva na select
 - ORA-1031: Nedostatečná oprávnění
- Sloupečky
 - ORA-904: Neplatný identifikátor
- Datové typy
 - Zapnutí automatické konverze

- Cílem je:
 - Minimalizovat počet dotazů do row cache
 - Vyhnout se automatickým konverzím
- Ideální by bylo i nepoužívat „Select * from abc“, protože přibýde dotaz do row cache: *„jaké sloupce má tabulka abc“* ... ale kdo tam má pokaždý ty sloupečky vyjmenovávat, že ☺

Sémantická analýza

**Select o.prijmeni,o.jmeno,s.os_cislo
from osoby o, studenti s
where o.statidno_obca<>203 and
o.osobidno=s.osobidno
order by o.prijmeni**

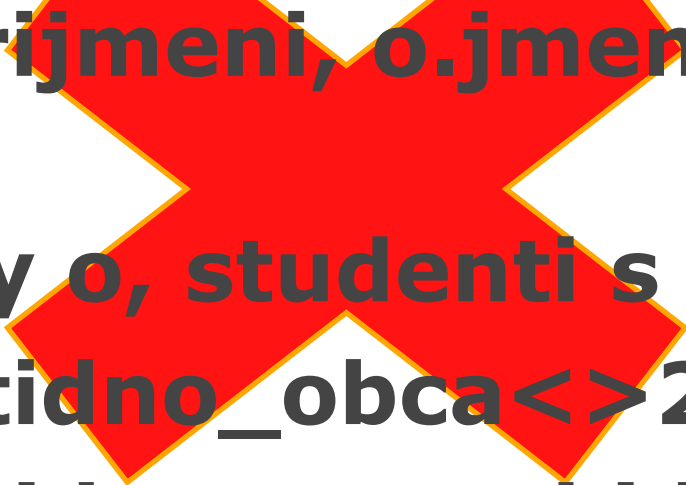
OK

```
Select osoby.prijmeni, osoby.jmeno,  
studenti.os_cislo  
from osoby, studenti  
where osoby.statidno_obca<>203  
and  
osoby.osobidno=studenti.osobidno  
order by osoby.prijmeni
```

OK

**Select prijmeni, jmeno, os_cislo
from osoby, studenti
where statidno_obca<>203
and
osoby.osobidno=studenti.osobidno
order by prijmeni**

**Select o.prijmeni, o.jmeno,
s.os_cislo
from osoby o, studenti s
where statidno_obca<>203
and s.osobidno=o.osobidno
order by prijmeni**



Prohledání library cache

- Databáze „si pamatuje“ předchozí dotazy a pokud již daný dotaz nalezne v library cache, odpadá vytváření plánu exekuce
- Vytváření plánu exekuce je zdaleka nejnáročnější operace při parsování
- Dotazy se hledají dle tzv. *hashe*

Prohledání library cache

- Při syntaktické analýze se jednotlivé části selectu „seřadí“, tj. následující dvojice jsou pro hashování shodné
 - *Select jmeno, prijmeni from osoby*
 - *Select prijmeni, jmeno from osoby*
- *Nebo*
 - *Select * from osoby where prijmeni like 'A%' and stav=3*
 - *Select * from osoby where stav=3 and prijmeni like 'A%'*

Prohledání library cache

- *Stejně ale nejsou:*

*Select * from osoby where
upper(jmeno)=upper('Adam')*

*Select * from osoby where
upper(jmeno)=upper('adam')*

*Select * from osoby where upper(jmeno)='ADAM'*

*Select * from osoby where upper(jmeno)='Adam'*

- *Vždy se vytváří nový exekuční plán*
- *Místo literátů je nutné používat parametry!!!*

Vytvoření exekučního plánu

- *Opakování matka moudrosti - vytváření exekučního plánu je nejnáročnější operace při parsování!*
- *Důležitým atributem plánu je tzv. cena – měla by říkat, jak je operace drahá (není to ale vždy 100% je to jen odhad ...)*
- *Oracle se samozřejmě snaží vymyslet co nejlepší plán exekuce a my mu v tom můžeme (a měli bychom) pomoci*
- *K tomu existuje několik zásad a nespočet „triků“ ☺*

Vytvoření exekučního plánu

- *Zásady (spíš pro správce databáze):*
 - *Každá „větší“ tabulka musí mít statistiky ... a teď co je to „větší“? (možná je dnes už lepší každá)*
 - *Sloupce, podle kterých často vyhledáváme by měly být oindexované*
 - *Oindexované by měly být i všechny sloupce, které tvoří vazby mezi tabulkami*
 - *Správně používat bitmapový index: omezená množina hodnot, minimální insert, delete a update klíče*

Vytvoření exekučního plánu

- *Triky*
 - *Union x union all*
 - *Select **
 - *Nepoužívat Or*
 - *Outer join x subquery*
 - *Konverze*
 - *Nepoužívat na indexované sloupce žádnou fci, výpočet apod.*
 - *Nepoužívat order a distinct pokud to není nutné*

Vytvoření exekučního plánu

- *Union x union all*

```
select os_cislo from preruseni  
union  
select os_cislo from studenti_vyjezdy
```

Cena 22

```
select os_cislo from preruseni  
union all  
select os_cislo from studenti_vyjezdy
```

Cena 20

```
select distinct os_cislo from  
(select os_cislo from preruseni  
union all  
select os_cislo from studenti_vyjezdy)
```

Cena 21

Vytvoření exekučního plánu

- *Select * from ...*
 - *Není vždy nejlepší řešení*

```
select * from pred_varianty  
where prac_zkr='KIV'
```

CENA 477

```
select zkr_predm  
from pred_varianty  
where prac_zkr='KIV'
```

CENA 4

Vytvoření exekučního plánu

- *Vyhnout se OR*
 - *OR je ze všech základních operátorů nejnáročnější pro vytváření exekučního plánu*
 - *Pokud to logika věci dovolí, je lepší se mu vyhnout*

Vytvoření exekučního plánu

- Nepoužívat na indexované sloupce žádnou funkci, výpočet apod.*

```
select * from cis_pracovist where  
zkr='KIV'
```

CENA 1 (použije se index)

```
select * from cis_pracovist where  
upper(zkr)='KIV'
```

CENA 4 (FULL SCAN)

Vytvoření exekučního plánu

- *Nepoužívat automatické konverze*

```
select * from osoby where  
rod_cislo='1234567890'
```

CENA 1 (použije se index)

```
select * from osoby where  
rod_cislo=1234567890
```

CENA 1423 (FULL SCAN)

Vytvoření exekučního plánu

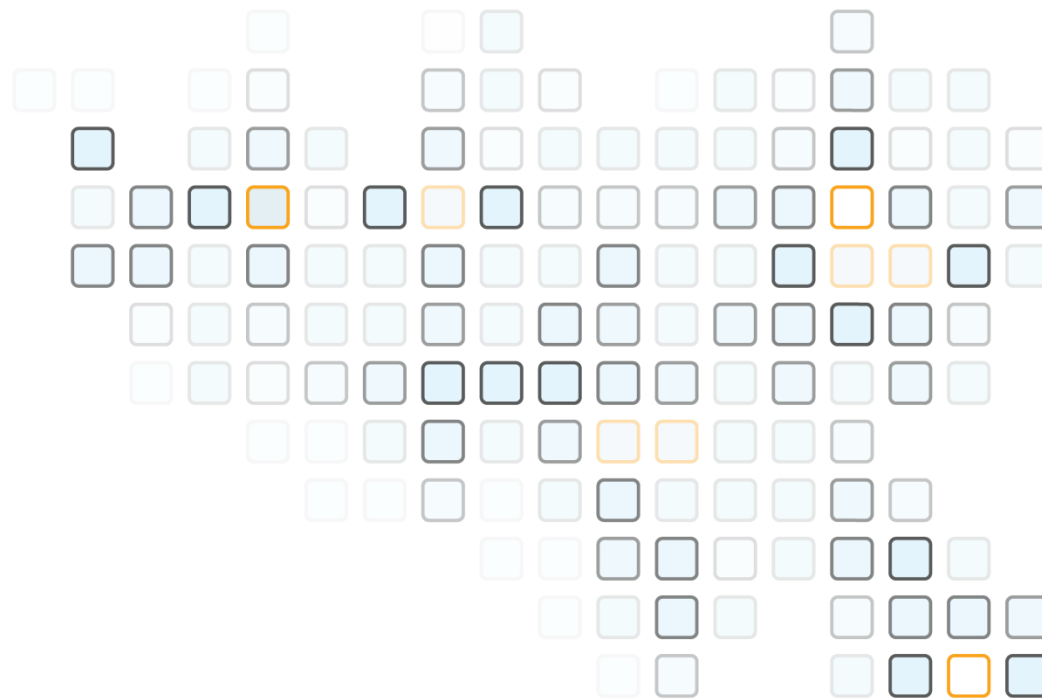
- *Nepoužívat order by pokud to není nutné*

```
select o.rod_cislo,s.os_cislo  
from osoby o, studenti s  
where s.osobidno=o.osobidno
```

CENA 1601

```
select o. rod_cislo,s.os_cislo  
from osoby o, studenti s  
where s.osobidno=o.osobidno  
order by o. rod_cislo
```

CENA 3014



A TO JE VŠE ...

Dotazy? Připonínky?