



Kustannuslaskentaoptimointi

- Laskenta edellyttää tilastoaineistoa
- Tilastoaineisto pitää uudistaa kun tietokantaan on tehty merkittävästi muunnoksia.
- Tilastoaineistoon kuuluvat esim.
 - taulujen ja indeksien koot
 - sivujen lukumäärä, täyttösuhde
 - sarakeen pienin ja suurin arvo, mahdollinen arvojen jakautumatieta

1



Kustannuslaskentaoptimointi

- Esimerkiksi Oraclessa on komento Analyze, jolla kustannuslaskennassa tarvittavan tilastotiedon saa tuotettua
- Analyze table opiskelija estimate statistics;
 - laskee taulukohtaisia tilastotietoja oletuskokoinen otoksen perusteella (otoskoon voi myös antaa)
- Analyze table opiskelija compute statistics for columns aloitusvuosi size 10;
 - laskee sarakkeelle tilastotietoja käymällä läpi koko taulun, tuottaa myös jakautumatieta (arvoväli jaettu 10 osaan)

2



Kustannuslaskentaoptimointi

- Oraclessa optimoijan tekemään suunnitelman saa selville lauseella EXPLAIN PLAN
- esim:


```
EXPLAIN PLAN
SET STATEMENT_ID = 'Raise in Chicago'
INTO plan_table FOR
UPDATE emp SET sal = sal * 1.10
WHERE deptno =
(SELECT deptno FROM dept WHERE loc = 'CHICAGO');
```
- Yllä suunnitelma kirjoitetaan tässä plan_table nimiseen tauluun. Taulu täytyy olla luotu ennen lauseen käyttöä sen rakenne on seuraava:

3



Kustannuslaskentaoptimointi

```
CREATE TABLE plan_table (
statement_id VARCHAR2(30),
timestamp DATE,
remarks VARCHAR2(80),
operation VARCHAR2(30),
options VARCHAR2(30),
object_node VARCHAR2(128),
object_owner VARCHAR2(30),
object_name VARCHAR2(30),
object_instance NUMERIC,
object_type VARCHAR2(30),
optimizer VARCHAR2(255),
search_columns NUMERIC,
id NUMERIC,
parent_id NUMERIC,
position NUMERIC,
cost NUMERIC,
cardinality NUMERIC,
bytes NUMERIC,
other_tag VARCHAR2(255),
other LONG);
```

tieto taulussa puumaisena rakenteena, parent_id kertoo mikä operaatio suoritetaan rivillä kuvattavan jälkeen

4



Kustannuslaskentaoptimointi

- esim hierarkkinen kysely


```
SELECT LPAD(' ',2*(LEVEL-1))||operation operation, options, object_name, position
FROM plan_table
START WITH id = 0 AND statement_id = 'Raise in Chicago'
CONNECT BY PRIOR id = parent_id AND statement_id = 'Raise in Chicago';
```

voisi tuottaa tuloksen

OPERATION	OPTIONS	OBJECT_NAME	POSITION
UPDATE STATEMENT			1
FILTER			0
TABLE ACCESS	FULL	EMP	1
TABLE ACCESS	FULL	DEPT	2

5



Kustannuslaskentaoptimointi

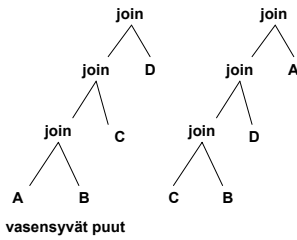
- Kustannuslaskentaoptimoinnissa tuotetaan vaihtoehtoisia toteutus suunnitelmia ja lasketaan niille kustannus.
- Se, jonka kustannus on pienin valitaan.
- Kaikkia vaihtoehtoja ei lasketa,
 - Esimerkiksi useiden liitosten järjestyksen määrittämiseksi käytetään ns. vasensyvää puuta (left deep tree). Tässä liitosrakenne on muotoa ((A join B) join C) join D ...
 - eli valmiiseen liitostulokseen liitetään taulu kerrallaan
 - lokaalia optimointia: mikä liitos kannattaa tehdä ensin, mikä kannattaa liittää, sitten tämän tulokseen, jne
 - liitosrakenteita muotoa (A join B) join (C join D) ei arvioida

6

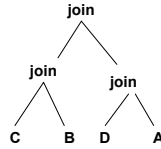


Kustannuslaskentaoptimointi

mahdollisia liitosjärjestyksiä
tauluille A,B,C,D



ei mahdollinen liitosjärjestys
tauluille A,B,C,D



7



Kustannuslaskentaoptimointi

- Vaikka optimoijalla onkin tiedossa tilastoaineistoa se voi silti päätyä ratkaisuun, joka ei ole välttämättä paras kyseiseen tilanteeseen
- Monissa tkhj:ssä käyttäjä pystyy kyselyyn upotettujen vihjeiden avulla vaikuttamaan optimoijan toimintaan, alla Oracle vihje:

```
SELECT /*+ ORDERED USE_NL(customers) */ accounts.balance,  
        customers.last_name, customers.first_name  
FROM accounts, customers  
WHERE accounts.custno = customers.custno;
```

Käytä sisäkkäisiä silmukoita, customers sisempänä tauluna

8