Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

08. Juli 2013

Datenbanken II B: DBMS-Implementierung — Hausaufgabe 12A —

Legen Sie in Ihrem kleinen DBMS eine Relation `R` an, mit zwei Attributen:

- `A` vom Typ `int`,
- `B` vom Typ `str` mit der Maximallänge 40.
- `C` vom Typ `int`.

Fügen Sie in die Relation 10 000 Tupel ein, wobei `A` von `1` bis `10000` läuft, `C` in umgekehrter Richtung von `99999` bis `0`, und `B` immer den festen Wert `ABCDEFGHIJKLMNOPQRSTUVWXYZ` hat (die Länge sind 26 Zeichen, so dass Systeme mit festem Tupelformat ein wenig benachteiligt werden — sie haben dafür andere Vorteile).

a) Wie lange dauert es, die Relation anzulegen und mit Daten zu füllen?

b) Wie viele Blöcke hat die Relation (d.h. wie viele Blöcke sind wirklich belegt und müßten bei einem Full Table Scan durchlaufen werden)? Wie viele Blöcke muß die Datei mindestens haben, um die Relation speichern zu können?

c) Verwenden Sie eine Puffergröße, die es erlaubt, die ganze Relation im Hauptspeicher zu halten (1000 Blöcke sollten sicher reichen). Wie viel Zeit braucht Ihr Programm, um den "Full Table Scan" 500 Mal auszuführen? Es soll jeweils der Wert der Attribute `A` und `C` abgefragt werden, und geprüft werden, dass die Summe immer 10 000 ist. Starten Sie in einem Zustand, in dem der Puffer leer ist und, die Datei noch nicht geöffnet ist.

— Hausaufgabe 12B —

Gegeben sei folgende Anfrage an die EMP-DEPT-Datenbank:

```
SELECT EMPNO, ENAME, JOB, DEPTNO
FROM   EMP E
WHERE  EXISTS(SELECT EMPNO
              FROM   EMP U
              WHERE  U.MGR = E.EMPNO)
```

Es gibt einen unique Index `EMP_EMPNO_PK` über `EMP(EMPNO)`. Falls man den Optimizer Mode `all_rows` wählt, erzeugt Oracle folgenden Auswertungsplan:


```
0   SELECT STATEMENT Optimizer=ALL_ROWS (Cost=6 Card=6 Bytes=150)
1 0   MERGE JOIN (SEMI) (Cost=6 Card=6 Bytes=150)
2 1     TABLE ACCESS (BY INDEX ROWID) OF EMP (TABLE) (Cost=2 Card=14 Bytes=294)
3 2       INDEX (FULL SCAN) OF EMP_EMPNO_PK (INDEX (UNIQUE)) (Cost=1 Card=14)
4 1     SORT (UNIQUE) (Cost=4 Card=13 Bytes=52)
5 4       TABLE ACCESS (FULL) OF EMP (TABLE) (Cost=3 Card=13 Bytes=52)
```

Wählt man dagegen den Optimizer Mode `first_rows`, so erhält man folgenden Auswertungsplan:

```
0   SELECT STATEMENT Optimizer=FIRST_ROWS (Cost=10 Card=6 Bytes=150)
1 0   NESTED LOOPS
2 1     NESTED LOOPS (Cost=10 Card=6 Bytes=150)
3 2       SORT (UNIQUE) (Cost=3 Card=13 Bytes=52)
4 3         TABLE ACCESS (FULL) OF EMP (TABLE) (Cost=3 Card=13 Bytes=52)
5 2       INDEX (UNIQUE SCAN) OF EMP_EMPNO_PK (INDEX (UNIQUE)) (Cost=0 Card=1)
6 1     TABLE ACCESS (BY INDEX ROWID) OF EMP (TABLE) (Cost=1 Card=1 Bytes=21)
```

a) Erläutern Sie die Auswertungspläne: Wie wird die Anfrage jeweils ausgewertet?

b) Erklären Sie, warum Oracle für die unterschiedlichen Optimierungsziele diese unterschiedlichen Zugriffspläne erzeugt.

— Hausaufgabe 12C —

Es sei folgende SQL-Anfrage an eine Relation `R(A,B,C,D)` gegeben: Der Typ der Attribute `A`, `B`, `C` sei `NUMERIC(3)`, der Typ von `D` sei `VARCHAR(1000)`. Schlüssel ist `A`.

```
SELECT A, D
FROM   R
WHERE  B = 5
AND    C BETWEEN 30 AND 100
```

Es gebe eine Index `I1` über `R(A)`, `I2` über `R(B)` und `I3` über `R(C)`. Geben Sie folgende drei Zugriffspläne mit den in Oracle QEPs üblichen Operatoren an und erklären Sie, wie man mit den im Oracle Data Dictionary vorhandenen Daten die Anzahl Blockzugriffe schätzen kann:

a) Full Table Scan

b) Zugriff über `I2` (Bedingung `B = 5`)

c) Zugriff über `I3` (Bedingung `C BETWEEN 30 AND 100`)

Falls Daten im Oracle Data Dictionary fehlen, erklären Sie, welche Informationen noch nützlich wären.