Kursus/workshop: SQL

Tips & Tricks til god SQL…

Frank Jensen

Indledning

- På Bankdata er ca. 80% af tidsforbruget ved programkørsel i DB2.
  - Så sig ikke at SQL ikke er et vigtigt programmeringssprog…
- Påstand: 30% af det samlede CPU forbrug er pga. ineffektiv SQL
  - Bankdata har gennemført 2 store optimeringsprojekter for at minimere CPU forbruget. Først sparrede vi 20% og igen senere 15% af det samlede CPU forbrug.
  - Begge projekter blev gennemført med få ressourcer og indenfor 6 måneder, og der blev IKKE ændret på forretningslogik, kun SQL statements!

Det vigtigste punkt

Skriv nu kønt SQL...

Designet af tabellen

- Selvom normaliseringen siger 1 tabel, så overvej om dataene hænger forretningsmæssigt sammen eller skal ligge i flere tabeller.
  - F.eks. BRYGGERI og AARLIGT_SALG
  - Det betyder noget for hvordan dataene gemmes og hvordan man har mulighed for at lave opdateringer/styring af låse på dataene
- Overvej hvilke access type der er vigtigst. Er det at kunne oprette eller læse?
  - Hvis det er at læse skal det helst være en specifik nøgle
  - Men hvis det er at oprette skal det være en fortløbende nøgle
- Felter der opdateres skal ligge til sidst pga. loggen
- Sidste felt er altid SENEST_OPDATERING hvis der kan opdateres

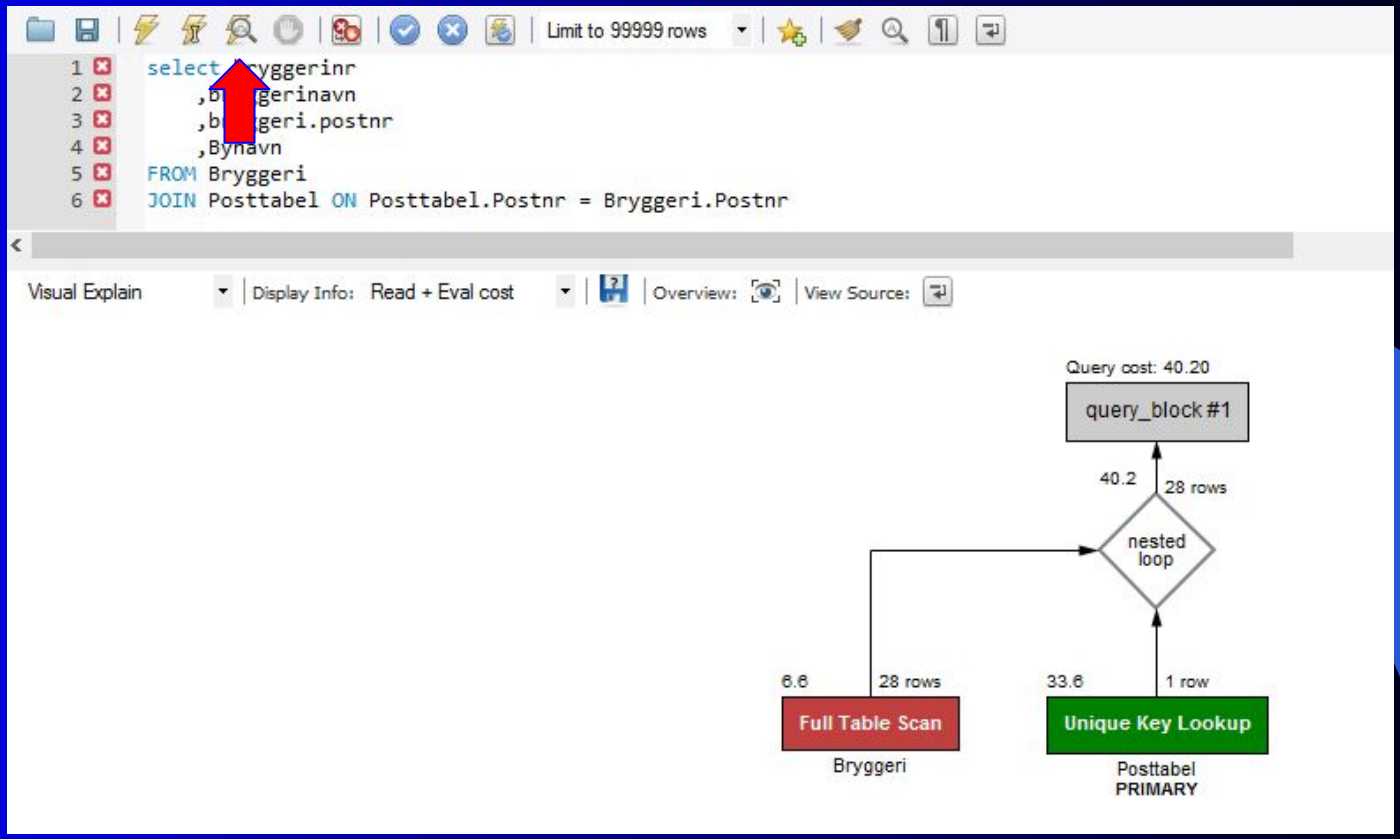
Indeksering…

Et indeks er en genvej til at finde dine rækker når du søger…

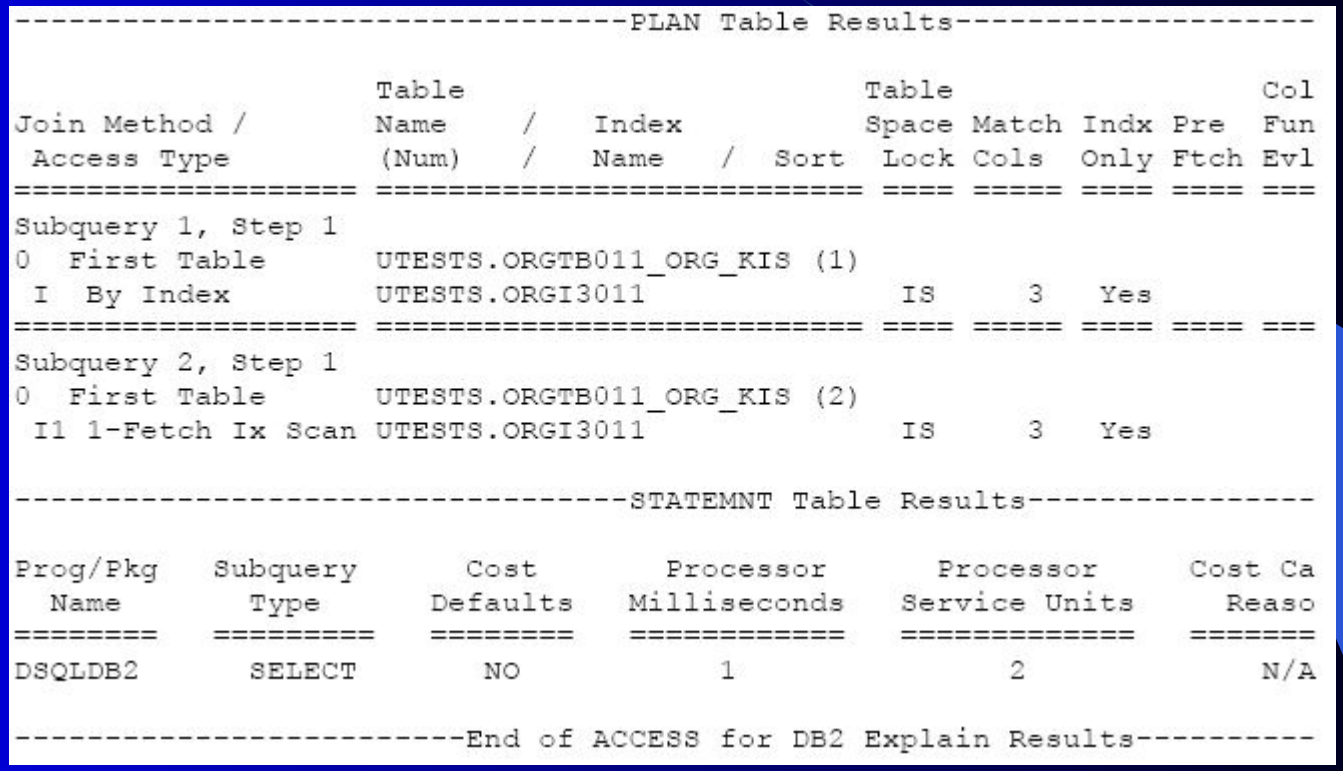
Men

- Tænk på et indeks som en ny tabel.
- Hvis du har en tabel med 2 indeks, betyder det at for hver INSERT du laver, sker der automatisk 2 inserts mere, et i hvert indeks!
- Hvis man opdaterer felter i et indeks, mister det en del effektivitet
  - Og bliver fragmenteret og kræver derfor services
- Der skal være en rigtig god grund til at oprette over 4 indeks udover clusteret(primært) index

Analysere dit SQL (Explain)



Analysere dit SQL (Explain)



```
----------------------------------PLAN Table Results--------------------

                      Table                          Table                 Col
Join Method /         Name     /   Index             Space Match Indx Pre  Fun
 Access Type          (Num)    /   Name    /  Sort    Lock  Cols  Only Ftch Evl
==================== ============================ ==== ===== ==== ==== ===
Subquery 1, Step 1
0  First Table       UTESTS.ORGTB011_ORG_KIS (1)
 I  By Index         UTESTS.ORGI3011                IS     3   Yes
==================== ============================ ==== ===== ==== ==== ===
Subquery 2, Step 1
0  First Table       UTESTS.ORGTB011_ORG_KIS (2)
 I1 1-Fetch Ix Scan UTESTS.ORGI3011                IS     3   Yes

----------------------------------STATEMNT Table Results----------------

Prog/Pkg    Subquery      Cost       Processor      Processor      Cost Ca
  Name        Type      Defaults   Milliseconds   Service Units     Reaso
========    =========   ========   ============   =============   =======
DSQLDB2       SELECT       NO            1               2            N/A

------------------------End of ACCESS for DB2 Explain Results----------
```

Den gode programmør analyserer ALTID sit SQL

Generelle T&T ved select

- Select kun felter du skal bruge, aldrig SELECT *
- Navngiv felter ved bruger af funktioner, f.eks. Count(*) AS Antal

- Benyt aldrig COUNT(*) som eksistenstjek. Benyt SELECT 'X' og FETCH FIRST 1 ROW ONLY

- Prisen ved CURSOR er at åbne den (findes rækker) og hente den første række.

- Skriv din FROM sætning som du gerne vil have læst tabellerne (Det er ingen garanti, men kun en hjælp til Optimizeren)

Generelle T&T ved WHERE

- Du behøver ikke selecte felter fordi de bruges i ORDER BY
  og behøver heller ikke selecte felter du allerede kender
  F.eks. Order by 1,2 desc,3 er muligt - undtagen at læse
- Brug IN-lister eller UNION frem for OR
- Brug UNION ALL hvis der alligevel ikke er dubletter
- Hardcode værdier i SQL er bedre end input-variabler
- Hvis du har en GROUP BY og kun er interesseret i data der er et vist antal af: HAVING COUNT(*) > 1

CASE i stedet for UNION

Hvis du f.eks. skal lave en sum på to tal i samme tabel afhængig af fortegn?

```
Select '+', count(*) from konto where saldo > 1
union all
Select '-', count(*) from konto where saldo < 1
```

CASE i stedet for UNION

```
EXEC SQL
     DECLARE CSR-SUMANT CURSOR FOR
     SELECT COALEACE(SUM(
                CASE WHEN T2.RGN_BEL <= 0
                     THEN 1
                     ELSE 0 END ), 0) AS ANTAL_MINUS
          , COALEACE(SUM(
                CASE WHEN T2.RGN_BEL > 0
                     THEN 1
                     ELSE 0 END ), 0) AS ANTAL_PLUS
          , COALESCE(SUM(
                CASE WHEN T2.RGN_BEL <= 0
                     THEN RGN_BEL
                     ELSE 0 END ), 0) AS RGN_BEL_MINUS
          , COALESCE(SUM(
                CASE WHEN T2.RGN_BEL >  0
                     THEN RGN_BEL
                     ELSE 0 END ), 0) AS RGN_BEL_PLUS
       FROM PSTTB204_POST_IX4 T1 INNER JOIN
            PSTTB002_POSTERING T2 ON T1.POST_ID = T2.POST_ID
      WHERE T1.BANKNR          = :PSTTB204.BANKNR
        AND T1.TERM_ID         = :PSTTB204.TERM-ID
        AND T1.BOGF_DATO     BETWEEN :WS-SQL-FRA-DATO
        AND                          :WS-SQL-TIL-DATO
     FOR FETCH ONLY
     OPTIMIZE FOR 1 ROW
END-EXEC
```

Korrelerede kontra non-korrelerede subselects

- Korrelerede/Sammen hængde

```
SELECT   T1.ORG_ENHEDNR
INTO     :ORGTB011.ORG-ENHEDNR
FROM     ORGTB011_ORG_KIS T1
WHERE    T1.BANKNR    = :ORGTB011.BANKNR
AND      T1.HIRAKINR = :ORGTB011.HIRAKINR
AND      T1.REFNR     = :ORGTB011.REFNR
AND      T1.START_TID = (
           SELECT   MAX(START_TID)
           FROM     ORGTB011_ORG_KIS T2
           WHERE    T2.BANKNR    = T1.BANKNR
           AND      T2.HIRAKINR = T1.HIRAKINR
           AND      T2.REFNR     = T1.REFNR )
```

- Non-korrelrede

```
SELECT   T1.ORG_ENHEDNR
INTO     :ORGTB011.ORG-ENHEDNR
FROM     ORGTB011_ORG_KIS T1
WHERE    T1.BANKNR    = :ORGTB011.BANKNR
AND      T1.HIRAKINR = :ORGTB011.HIRAKINR
AND      T1.REFNR     = :ORGTB011.REFNR
AND      T1.START_TID = (
           SELECT   MAX(START_TID)
           FROM     ORGTB011_ORG_KIS T2
           WHERE    T2.BANKNR    = :ORGTB011.BANKNR
           AND      T2.HIRAKINR = :ORGTB011.HIRAKINR
           AND      T2.REFNR     = :ORGTB011.REFNR )
```

	Elaps	CPU	Wait
korreleret	0,015969	0,015353	0,000297
non-korreleret	0,007477	0,007145	0,000352