

Kap9.Delspørring

Subqueries

- Delspørring.
- Delspørring med flere kolonner.
- Sammenholdt delspørring.

Delspørring

WHERE-betingelsen i en SELECT-setning kan inneholde en annen SELECT-setning. Dette kalles for en DELSPØRRING.

EKSEMPEL:

```
SELECT *
FROM VARE_LEV
WHERE PRIS < (SELECT PRIS
              FROM VARE_LEV
              WHERE LEVNR = 3010
              AND VARENr = 1001)
```

Delspørringen utføres først og returnerer en verdi. Deretter testes det mot denne verdien.

I dette eksemplet går delspørringen mot samme tabell som hoved-spørringen. Man kan også ha en delspørring mot andre tabeller.

ANY og ALL

I forrige eksempel returnerte delspørringen en enkelt verdi, som det er greit å teste mot. Men en SELECT- setning kan jo returnere flere rader. Hvordan skal da en hovedspørringen vite hvilken den skal teste mot?

Det løser vi ved å ta med ANY eller ALL.

EKSEMPEL:

```
SELECT * FROM
VARE_LEV
WHERE PRIS < ALL (SELECT PRIS
                  FROM VARE_LEV
                  WHERE VARENr = 1001)
```

Bedre

*where pris < (select
max (...))*

☞ I stedet for = ANY kan vi si IN

☞ I stedet for !=ALL kan vi si NOT IN

Det betyr det samme.

Delspørring med flere kolonner

Hittil har vi sett delspørringen som returnerer en kolonne. Det er også mulig at delspørringen returnerer flere kolonner. Da må vi teste mot alle, og sette dem i parantes i WHERE-betingelsen.

```
SELECT ....  
FROM ....  
WHERE (kolonne,...,kolonne) =  
      (SELECT kolonne,...,kolonne  
       FROM ...)
```

Flere delspøringer

Vi kan ha flere delspøringer

```
SELECT ...  
FROM ...  
WHERE ...      (SELECT ...)  
AND            (SELECT ...)
```

og flere nivåer av delspøringer

```
SELECT ...  
FROM .....  
WHERE .....  (SELECT ...  
              FROM ...  
              WHERE ..(SELECT ...  
                       FROM.....  
                       WHERE .....))
```

correlated subqueries

Sammenholdt delspørring

Det kan tenkes at vi ikke kan la delspørringen utføres før utførelsen av hovedspørringen starter. F.eks. hvis WHERE-betingelsen i delspørringen skal teste mot en kolonne i hovedspørringen

Da har vi en sammenholdt delspørring.

Hvis begge går mot samme tabell, må vi bruke et midlertidig merke på tabellnavnet i delspørringen eller hovedspørringen, for at delspørringen skal vite hva det er vi henviser til.

EKSEMPEL:

```
SELECT VL1.*  
FROM VARE_LEV VL1  
WHERE VL1.PRIS > (SELECT AVG(VL2.PRIS)  
                  FROM VARE_LEV VL2  
                  WHERE VL2.VARENr = VL1.VARENr)
```

EXISTS - operatoren

EXISTS gir verdiene TRUE hvis delspørringen returnerer minst en rad, FALSE hvis den ikke returnerer noen rader.

EKSEMPEL:

```
SELECT *  
FROM VARE V  
WHERE EXISTS (SELECT *  
              FROM VARE_LEV VL  
              WHERE VL.VARENK = V.VARENK)
```

/ Lager først de fem brukerne: */*

```
Create user PER identified by PER_pw;  
Create user KARI identified by KARI_pw;  
Create user KNUT identified by KNUT_pw;  
Create user KJELL identified by KJELL_pw;  
Create user KÅRE identified by KÅRE_pw;
```

/ Gir brukerne rettighet til å logge på ved å tildele */*

/ den innebygde rollen CONNECT: */*

```
Grant CONNECT to PER, KARI, KNUT, KJELL, KÅRE;
```

/ Lager to roller: */*

```
Create role A;  
Create role B;
```

/ Gir aksessrettigheter til rolle A: */*

```
Grant SELECT, UPDATE, INSERT, DELETE on VARE to A;  
Grant SELECT, UPDATE, INSERT, DELETE on LEV to A;  
Grant SELECT, UPDATE, INSERT, DELETE on VARE_LEV to A;
```

```
Grant DELETE on BESTILLING to A;
```

```
Grant DELETE on BESTLINJE to A;
```

/ Gir aksessrettigheter til rolle B: */*

```
Grant SELECT on VARE to B;  
Grant SELECT on LEV to B;  
Grant SELECT on VARE_LEV to B;
```

```
Grant SELECT, UPDATE, INSERT on BESTILLING to B;
```

```
Grant SELECT, UPDATE, INSERT on BESTLINJE to B;
```

/ Tildeler roller til de ansatte: */*

```
Grant A to PER, KARI;
```

```
Grant B to PER, KARI, KNUT, KJELL, KÅRE
```

/ kan erstattes med Grant B to PUBLIC */*