

IN2090 – Repetisjonsoppgaver

SQL

I disse oppgavene skal vi bruke skjemaet under som beskriver et online match-basert spill-nettside. Nettsiden har mange spill, og hver bruker kan spille de ulike spillene mot andre brukere. Hver spiller får også en poengsum for å vinne i et spill, men hvor mange poeng kan variere fra spill til spill. Skjemaet set slik ut:

```
Spiller(brukernavn, mail)
Spill(sid, navn, vinner_poeng)
Match(mid, sid, spiller1, spiller2, spiller1_poeng, spiller2_poeng)
```

Understrekede kolonner er primærnøkklene. Tabellen **Spiller** beskriver spillerne, hvor **brukernavn** er en streng uten mellomrom som beskriver brukernavnet til spilleren og **mail** er en streng som inneholder spillerens mailadresse og kan ikke være NULL.

Tabellen **Spill** inneholder informasjon om ulike spill som kan spilles. I vårt lille eksempel inneholder denne tabellen kun navnet på spillet (**navn**), samt antall poeng spilleren som vinner får. Tanken er at hver spiller kan opparbeide seg en poengsum som kan brukes til f.eks. en "High-scores"-tabell, el.

Tabellen **Match** inneholder informasjon om et spill mellom to spillere. Kolonnen **sid** peker på **Spill(sid)** og sier hvilket type spill spillet er av. Kolonnene **spiller1** og **spiller2** refererer til **Spiller(brukernavn)** og sier hvilke spillere som deltok i spillet. Kolonnene **spiller1_poeng** og **spiller2_poeng** inneholder hvor mange poeng hhv. **spiller1** og **spiller2** fikk i dette spillet.

Generelt vil vi alltid si at spilleren som vinner en match er den som fikk flest poeng i den matchen. Vi antar at det alltid er en vinner av hvert spill, altså kan poengsummen til to spillere i en match aldri være like.

For eksempel vil spillet *football* ha 3 poeng for den som vinner (kolonnen **Spill(vinner_poeng)**). Poengene som da puttes i de to kolonnene **Spill(spiller1_poeng)** og **Spill(spiller2_poeng)** vil da være antall mål hvert lag fikk i den matchen.

*(Om du ønsker en ekstra-oppgave kan du skrive SQL-scriptet som lager skjemaet over, løsningen kommer på neste side. Legg alle tabellene i et eget skjema med navn **sp**.)*

Vi antar at tabellene er laget med følgende SQL-script:

```
CREATE SCHEMA sp;

CREATE TABLE sp.spiller(
  brukernavn text PRIMARY KEY CHECK (NOT brukernavn LIKE '% %'),
  mail text NOT NULL
);
CREATE TABLE sp.spill(
  sid int PRIMARY KEY,
  navn text,
  vinner_poeng int
);
CREATE TABLE sp.match(
  mid int PRIMARY KEY,
  sid int REFERENCES sp.spill(sid),
  spiller1 text REFERENCES sp.spiller(brukernavn),
  spiller2 text REFERENCES sp.spiller(brukernavn),
  spiller1_poeng int,
  spiller2_poeng int CHECK (spiller2_poeng != spiller1_poeng)
);
```

Oppgave 1

Skriv en spørring som finner alle unike mail til spillere som har spilt mot spilleren med brukernavn 'dr1337' som spiller1.

Løsningsforslag

```
SELECT DISTINCT s.mail
FROM sp.match AS m
INNER JOIN sp.spiller AS s
ON (m.spiller2 = s.brukernavn)
WHERE m.spiller1 = 'dr1337';
```

Oppgave 2

a) Skriv en spørring som finner antall matcher som er gjort av hvert spill. Skriv ut sid for hvert spill, i tillegg til antallet matcher. (Ikke ta med spill som er spilt 0 ganger)

b) Skriv en spørring som finner det samme som i a), men som også inkluderer spill som er spilt 0 ganger.

Løsningsforslag

a)

```
SELECT s.sid, count(*) AS antall
FROM sp.spill AS s
     INNER JOIN sp.match AS m USING (sid)
GROUP BY s.sid;
```

b)

```
SELECT s.sid, count(m.mid) AS antall
FROM sp.spill AS s
     LEFT OUTER JOIN sp.match AS m USING (sid)
GROUP BY s.sid;
```

Oppgave 3

Skriv en spørring som for hver match skriver ut brukernavnet på den spilleren som vant, samt sid-verdien. Husk at en spiller kan vinne det samme spillet flere ganger.

Hint: Her kan man bruke en variant av `UNION`.

Løsningsforslag

```
(SELECT sid, spiller1 AS spiller
FROM sp.match
WHERE spiller1_poeng > spiller2_poeng
)
UNION ALL
(SELECT sid, spiller2 AS spiller
FROM sp.match
WHERE spiller2_poeng > spiller1_poeng
);
```

Oppgave 4

Skriv en spørring som finner antall poeng hver spiller har. Skriv ut brukernavnet på hver spiller, sammen med spillerens poeng.

Hint: Du kan bruke spørringen fra forrige oppgave som delspørring. Om du ikke klarte å løse den oppgaven, anta at tabellen `vinnere` er et view over spørringen i den oppgaven.

Løsningsforslag

```
WITH
  vinnere AS (
    (SELECT sid, spiller1 AS spiller
     FROM sp.match
     WHERE spiller1_poeng > spiller2_poeng
    )
    UNION ALL
    (SELECT sid, spiller2 AS spiller
     FROM sp.match
     WHERE spiller2_poeng > spiller1_poeng
    )
  ),
  spill_poeng AS (
    SELECT v.spiller, s.vinner_poeng
    FROM vinnere AS v INNER JOIN sp.spill AS s
         USING (sid)
  )
SELECT p.spiller, sum(p.vinner_poeng)
FROM spill_poeng AS p
GROUP BY p.spiller;
```

eller, litt kortete:

```
WITH
  vinnere AS (
    (SELECT sid, spiller1 AS spiller
     FROM sp.match
     WHERE spiller1_poeng > spiller2_poeng
    )
    UNION ALL
    (SELECT sid, spiller2 AS spiller
     FROM sp.match
     WHERE spiller2_poeng > spiller1_poeng
    )
  )
SELECT v.spiller, sum(s.vinner_poeng)
FROM vinnere AS v INNER JOIN sp.spill AS s
     USING (sid)
GROUP BY v.spiller;
```

Oppgave 5

Skriv en spørring som finner ut hvor mange ganger hver spiller har vunnet hvert spill.

Hint: Igjen kan det være lurt å bruke spørringen fra oppgave 3 som delspørring. Om du ikke klarte å løse den oppgaven, anta at tabellen `vinnere` er et view over spørringen i den oppgaven.

Løsningsforslag

```
WITH
  vinnere AS (
    (SELECT sid, spiller1 AS spiller
     FROM sp.match
     WHERE spiller1_poeng > spiller2_poeng
    )
    UNION ALL
    (SELECT sid, spiller2 AS spiller
     FROM sp.match
     WHERE spiller2_poeng > spiller1_poeng
    )
  ),
  spill_vinninger AS (
    SELECT sid, spiller, count(*) AS antall
    FROM vinnere
    GROUP BY spiller, sid
  )
SELECT sv.spiller, s.navn, sv.antall
FROM spill_vinninger AS sv INNER JOIN sp.spill AS s
  USING (sid);
```

eller, litt kortete:

```
WITH
  vinnere AS (
    (SELECT sid, spiller1 AS spiller
     FROM sp.match
     WHERE spiller1_poeng > spiller2_poeng
    )
    UNION ALL
    (SELECT sid, spiller2 AS spiller
     FROM sp.match
     WHERE spiller2_poeng > spiller1_poeng
    )
  )
```

```
SELECT v.spiller, s.navn, count(*) AS antall
FROM vinnere AS v INNER JOIN sp.spill AS s
      USING (sid)
GROUP BY v.spiller, s.sid, s.navn;
```

Oppgave 6

Skriv en SQL-kommando som senker `vinner_poeng`-verdien med 1 for alle spill med en `vinner_poeng`-verdi høyere enn (eller lik) 5.

Løsningsforslag

```
UPDATE sp.spill
SET vinner_poeng = vinner_poeng - 1
WHERE vinner_poeng >= 5;
```