

## OHJ-3300 Johdatus tietokantoihin, tentti 21.04.2008

Opazo: \_\_\_\_\_ Nimi: \_\_\_\_\_  
 (Nimi ja opiskelijanumero kaikkiin palautettaviin papereihin!)

Vastaa erilliselle konseptille. Muista laittaa myös konseptiin nimi ja opiskelijanumero.

## Tehtävä 1. (kaksi pistettä kustakin kohdasta)

- 1.1 Millainen on SQL '99:n (SQL2003) mukainen olio-relaatio -malli?  
 1.2 Mitä eroa on näkyvillä ja perusrelaatioilla?  
 1.3 Mitkä ovat tietokannan hallintajärjestelmän tehtävät?  
 1.4 Selitä eheyden käsite relaatiotietokannoissa? Miten sitä hallitaan?  
 1.5 Mitä on tietokantojen yhteydessä metatieto ja mihin sitä tarvitaan?

## Tehtävä 2.

Tehtävässä käytetään seuraavia relaatioita:

```
KURSSI (@Kurssinum, Nimi)
OPINTOJAKSO (@Opjaksoid, Kurssinum -> KURSSI, Vuosi, Periodi)
HTRYHMA (@Ryhmanro, @Opjaksoid -> OPINTOJAKSO, Vtpaiva, Aika, Paikka)
OPISKELIJA (@Opnum, Etunimi, Sukunimi, Email)
OPISKELU (@Opnum -> OPISKELIJA, @Opjaksoid -> OPINTOJAKSO)
HARJOITUS (@Opnum -> OPISKELIJA, (@Ryhmanro, @Opjaksoid) -> HTRYHMA)
SUORITE (@SuoriteID, Pvm, Tyyppi, Aika, Osa, Viikko, Opjaksoid -> OPINTOJAKSO)
SUORITUS (@Suorite -> SUORITE(SuoriteID), @Opiskelija -> OPISKELIJA(Opnum), Pisteet)
```

Suorite voi olla tyyppitään "luento", "tentti", "valikoe", "harjoitus", "viharjoitus" (viikkoharjoitus) tai "kokarvosana" (kokonaisarvosana). Vuosi tarkoittaa lukuvuotta. Opiskelijanumero on vähintään 5-numeroinen. Viikko on väliltä 1-53 ja Osa väliltä 1-16. Pääsävain on merkitty @pääsävain ja vierosävain Vierosävain->Relaatio.

Oleta, että muissa tauluissa on tietoa, mutta taulut Suorite ja Suoritus ovat tyhjiä. Tämän jälkeen tietokannalle annetaan seuraavalla sivulla olevat SQL-komennot.

```
INSERT INTO Kurssi VALUES (1111, 'snobailun peruskurssi');
COMMIT;
INSERT INTO Opintojakso VALUES (1, 1111, 2004, 3);
INSERT INTO Opintojakso VALUES (2, 1111, 2005, 3);
INSERT INTO Opintojakso VALUES (3, 1111, 2006, 3);
INSERT INTO Opintojakso VALUES (4, 1111, 2007, 3);
COMMIT;
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (1, 'tentti', 3);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (2, 'viharjoitus', 3);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (3, 'kokarvosana', 1);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (4, 'kokarvosana', 2);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (5, 'harjoitus', 3);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (6, 'kokarvosana', 3);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (7, 'kokarvosana', 4);
COMMIT;
INSERT INTO Suoritus VALUES (321, 123456, 4);
INSERT INTO Suoritus VALUES (321, 123456, 4);
INSERT INTO Suoritus VALUES (123456, 321, 4);
ROLLBACK;
INSERT INTO Suoritus VALUES(1, 123456, 12);
INSERT INTO Suoritus VALUES (2, 123456, 0);
INSERT INTO Suoritus VALUES (3, 123456, 1);
INSERT INTO Suoritus VALUES (5, 123456, 4);
INSERT INTO Suoritus VALUES (6, 123456, 2);
COMMIT;
UPDATE Suoritus SET Pisteet = 15
WHERE Opiskelija =123456 AND SuoriteID = 1;
UPDATE Suoritus SET Pisteet = 3
WHERE Opiskelija =123456 AND SuoriteID = 6;
COMMIT;
INSERT INTO Suoritus VALUES(1, 180180, 9);
INSERT INTO Suoritus VALUES(2, 180180, 0);
INSERT INTO Suoritus VALUES(5, 180180, 1);
INSERT INTO Suoritus VALUES(6, 180180);
COMMIT;
```

## 2.1 (yksi piste)

Selitä mitä seuraava SQL lause tekee ja näytä lopputulos kun kannan sisältö on yllä annettun mukainen.

```
SELECT Opiskelija, AVG(Pisteet)
FROM Suoritus, Suorite
WHERE Suoritus.SuoriteID = Suorite.SuoriteID AND Suorite.tyyppi = ' kokarvosana'
GROUP BY Opiskelija HAVING MIN(Pisteet) > 1 ORDER BY 2 desc;
```

## 2.2 (kaksi pistettä)

Tee kantaa vasten SQL-lause, joilla saat poistettua kaikki ne opiskelijat, joilla ei ole yhtään suoritusta. Perustele vastauksesi.

**2.3 (yksi piste)**

Tee SQL kysely, joka tuottaa tsmallleen saman lopputuloksen kuin seuraava TRC-kysely (Tuple relational Calculus). Perustelee vastauksesi sanallisesti:

```
{ S.Opnum, S.Pisteet | SUORITUS(S) AND SUORITE(SUO)
AND S.SuoriteID = SUO.SuoriteID
AND SUO.tyyppi = 'tentti' AND S.Pisteet > 12 }
```

**2.4 (kaksi pistettä)**

Selitä sanallisesti mitä seuraava tallennettu proseduuri tekee ja miten se toimii.

```
*CREATE PROCEDURE HaePisteet(Opnum INTEGER, Vuosi INTEGER, Periodi INTEGER)
RETURNS (Viikko INTEGER, Pisteet INTEGER)
BEGIN
```

```
EXEC SQL whenever sqlerror rollback, abort;
```

```
→ EXEC SQL PREPARE sel_cursor
SELECT Viikko, AVG(Pisteet)
FROM SUORITUS, SUORITE, OPINTOJAKSO
WHERE SUORITUS.Suorite = SUORITE.SuoriteID AND
SUORITUS.Opiskelija = ? AND
SUORITE.Tyyppi = 'vkharjoitus' AND
suorite.opjaksoid = opintojako.opjaksoid AND
kurssinum = 'OHJ-3300' AND vuosi=? AND periodi=?
GROUP BY Viikko;
```

```
→ EXEC SQL EXECUTE sel_cursor USING (Opnum, Vuosi, Periodi)
INTO (Viikko, Pisteet);
EXEC SQL FETCH sel_cursor;
```

```
WHILE SQLSUCCESS LOOP
RETURN row;
EXEC SQL FETCH sel_cursor;
END LOOP;
```

```
END*;
```

**Tehtävä 3.**

Tehtävässä käytetään samoja relaatioita kuin tehtävässä kaksi, mutta kaikissa tauluissa oletetaan olevan paljon tietoa (useilta vuosilta, kursseilta, opiskelijoilta, suoritteista, suorituksista, jne.)

**3.1 (kaksi pistettä)**

Laadi SQLää käyttäen päivitettävä näkymä, missä näkyvät vähintään kymmenen viikkoharjoitusporkkanapistettä opintojaksolla (vuosi 2005, periodi 4, kurssinumero OHJ-3320) saaneet opiskelijat. Kurssilla on viisi (5) harjoituskertaa, joista jokaisesta voi saada "porkkanaita" tehdyistä tehtävistä. Tietokannassa on 'vkharjoitus' suoritteita vastaavan suorituksen kohdalla tallennettuna nimenomaan näitä porkkanapisteitä.

**3.2 (kaksi pistettä)**

Laadi SQLää käyttäen hakemisto (index), joka tehostaisi kohdassa 4.1. tehtävän kyselyn suoritusta. Perustelee sanallisesti, miksi hakemisto tehostaisi kyselyn suoritusta.

**3.3 (kaksi pistettä)**

Anna SQLää käyttäen Teukalle (käyttäjänimi) oikeudet päivittää kaikkien opintojaksolle (vuosi 2005, periodi 3, kurssinumero OHJ-3300) osallistuneiden opiskelijoiden tietoja. Keijo ei saa nähdä eikä pystyä muuttamaan muiden opintojaksojen opiskelijoiden (opiskelija-taulussa olevia) tietoja.

**3.4 (kaksi pistettä)**

Laadi SQL-lause(et) (voit käyttää kaikkia oppimiasi mekanismeja), jolla lasket ja syötät kokonaisarvosanat niille opintojaksolle (vuosi 2005, periodi 4, kurssinumero OHJ-3320) osallistuneista opiskelijoista, jotka ovat saaneet vähintään yhden lopullisen viikkoharjoituspisteen kyseisellä opintojaksolla.

Lopulliset viikkoharjoituspisteet lasketaan porkkanapisteiden perusteella seuraavan kaavan mukaan:  $FLOOR((SUM(porkkanapisteet)+4)/8)$ . Porkkanapisteillä on vaikutusta vain niille opiskelijoille, jotka ovat saaneet vähintään yhden lopullisen viikkoharjoituspisteen opintojaksolla.

Kokonaisarvosana lasketaan kaavalla  $FLOOR((tenttipisteet + lopulliset viikkoharjoituspisteet+9)/5)-1$ . Kokonaisarvosanat sijoitetaan SUORITUS-taulun kenttään PISTEET. (Kokonaisarvosanat ovat siis opiskelijan suorituksia käsiteltävän opintojakson suoritteeseen tyyppiä 'kokarvosana'. Tehtävässä voi olettaa, että SUORITE-taulussa on ko.opintojaksolla valmiiksi olemassa 'kokarvosana' tyyppinen suorite, sekä että kyseisillä opiskelijoilla ei ole vielä kokonaissuoritusta ko. opintojaksolta).