

OHJ-3300 Johdatus tietokantoihin, tentti 28.01.2008

Opno: _____ Nimi: _____
 (Nimi ja opiskelijanumero kaikkiin palautettaviin papereihin!)

Tehtävä 1. (yksi piste kustakin kohdasta)

Valitse **(ympyröi)** kussakin kohdassa **yksi** (mielestäsi kaikkein oikein) vaihtoehto. Väärästä vastauksesta ei rikota miinuspisteillä.

1.1 Näkymiä voidaan käyttää relaatiotietokannoissa:

- parantamaan suorituskykyä
- esittämään sama tieto eri käyttäjille eri muodossa
- aina ainakin tietojen päivittämiseen
- Kaikki edellisistä vaihtoehdoista
- Ei mikään edellisistä

1.2 Henkilötietolain mukaan:

- Lakia sovelletaan henkilötietojen automaattiseen käsittelyyn
- Henkilön terveydentilaa saa käsitellä henkilökisterissa
- Rekisterinpitäjän ei tarvitse perustella henkilötietojen käsittelyä
- Kaikki edellisistä vaihtoehdoista
- Ei mikään edellisistä

1.3 Relaatiotietokannoissa relaation vierasavain:

- ei voi olla moniosainen
- ei voi olla tyhjä
- on sama kuin toisen (tai saman) relaation pääavain tai tyhjä
- Kaikki edellisistä
- Ei mikään edellisistä

1.4 Tietokannan hallintajärjestelmä (TKHJ):

- sallii yleensä tietokannan käsittelyn myös ilman tietokannan hallintajärjestelmää
- ei turvaa tietokantaa luvattomalta käytöltä ja vahingoilta
- ei mahdollista useiden käyttäjien yhtäaikaista tiedonkäsittelyä
- Kaikki edellisistä vaihtoehdoista
- Ei mikään edellisistä

1.5 Kyselykieli on relationaalisesti täydellinen jos:

- sillä pystytään esittämään kaikki samat kyselyt kuin relaatiokalkyyllilla
- relaatioalgebralla pystytään esittämään kaikki samat kyselyt kuin sillä
- relaatiokalkyyllilla pystytään esittämään kaikki samat kyselyt kuin sillä
- Kaikki edellisistä vaihtoehdoista
- Ei mikään edellisistä

Tehtävä 2. (yksi piste kustakin kohdasta)

Vastaa kysymyksiin niille varattuun tilaan. Rivien ulkopuolella olevat vastausten osat jätetään huomioimatta. Voit selventää vastauksia esimerkeillä, jos riveillä on tilaa.

2.1 Mitkä ovat olio- ja oliorelaatiotietokantojen olennaisimmat erot?

2.2 Mitä eroa on upotettua SQL:iä ja ODBC:tä käyttäen toteutetulla ohjelmalla (suunnittelijan/ohjelmoijan näkökulmasta)?

2.3 Jos sinulla on jotakin tapahtumaa varten luotu hakemisto, voitko lisätä sen loppuun sarakkeita, jotta se sopisi jollekin toiselle tapahtumalle? Miksi/miksi ei?

2.4 Mitkä ovat nk. tapahtumankäsittelyjärjestelmien ja tietovarastojen (data warehouse) olennaisimmat erot?

2.5 Mitkä ovat TRC ja DRC relaatiokalkyylien olennaisimmat erot?

Tehtävä 3.

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

Tehtävässä käytetään seuraavia relaatioita:

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

Suorite voi olla tyyppiään "luento", "tentti", "välikoe", "harjoitus", "vkharjoitus" (viikkoharjoitus) tai "kokarvosana" (kokonaisarvosana). Vuosi tarkoittaa lukuvuotta. Opiskelijanumero on vähintään 5-numeroinen. Viikko on väliltä 1-53 ja Osa väliltä 1-16. Päävain on merkitty @päävain ja vierasvain Vierasvain->Relatio.

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

```
INSERT INTO Kurssi VALUES (1111, 'snobbailun peruskurssi');
COMMIT;
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, 'vkharjoitus', 3);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (3, 'kokarvosana', 3);
INSERT INTO Suorite (SuoriteID, tyyppi, Opjaksoid) VALUES (7, 'kokarvosana', 4);
COMMIT;
INSERT INTO Suoritus VALUES (123456, 321, 4);
ROLLBACK;
INSERT INTO Suoritus(SuoriteID, Opiskelija) VALUES(1, 123456,12);
INSERT INTO Suoritus VALUES (2, 123456, 0);
INSERT INTO Suoritus VALUES (3, 123456, 1);
INSERT INTO Suoritus VALUES (7, 123456, 4);
COMMIT;
UPDATE Suoritus SET Pisteet = 5
WHERE Opiskelija = 123456 AND SuoriteID = 1;
INSERT INTO Suoritus VALUES(1, 180180, 9);
INSERT INTO Suoritus VALUES(2, 180180, 0);
INSERT INTO Suoritus VALUES(3, 180180);
COMMIT;
```

Kysymykset seuraavilla sivuilla.

3.1 (yksi piste)

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

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

3.2 (kaksi pistettä)

Tee kantaa vasten SQL-lause, jolla saat poistettua kaikki ne opiskelijat, jotka eivät opiskele millään kurssilla. Perustelee vastauksesi.

3.3 (kaksi pistettä)

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

```
{ O | OPISKELIJA(O) AND OPISKELU(OP)
AND O.Opnum = OP.Opnum AND O.Opnum > 170000 }
```

3.4 (kaksi pistettä)

Selitä sanallisesti mitä seuraava tallennettu proseduri tekee ja miten se toimii:

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

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

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

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

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

```
END";
```

Opno: _____ Nimi: _____
 (Nimi ja opiskelijanumero kaikkiin palautettaviin papereihin!)

Tehtävä 4.

Vastaa erilliselle konseptille. Muista laittaa myös konseptiin nimi ja opiskelijanumero. Tehtävässä käytetään samoja relaatioita kuin tehtävässä kolme, mutta kaikissa tauluissa oletetaan olevan paljon tietoa (useilta vuosilta, kursseilta, opiskelijoilta, suoritteista, suorituksista, jne.)

4.1 (kaksi pistettä)

Laadi SQLää käyttäen päivitettävä näkymä, jossa näkyvät ne opintojaksolle (vuosi 2005, periodi 4, kurssinumero OHJ-3320) osallistuneet opiskelijat, joilla on hyväksytty kokonaissuoritus (kokarvosana ei tyhjä ja > 0) opintojaksolta (vuosi 2005, periodi 3, kurssinumero OHJ-3300). Perustele näkymän päivitettävyyttä.

4.2 (kaksi pistettä)

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

4.3 (yksi piste)

Anna SQLää käyttäen Keijolle (käyttäjänimi) oikeudet selata kaikkien niiden opintojaksolle (vuosi 2005, periodi 4, kurssinumero OHJ-3320) osallistuneiden opiskelijoiden tietoja, joilla on hyväksytty kokonaissuoritus opintojaksolta (vuosi 2005, periodi 3, kurssinumero OHJ-3300). Keijo pystyy myös muuttamaan ko. opiskelijoiden (opiskelija-taulussa olevia) tietoja.

4.4 (kolme pistettä)

Laadi SQL-lause(et) (voit käyttää kaikkia oppimiasi mekanismeja), joilla lasket ja syötät kokonaisarvosanat niille opintojaksolle (vuosi 2005, periodi 4, kurssinumero OHJ-3320) osallistuneista opiskelijoista, joilla on hyväksytty kokonaissuoritus opintojaksolta (vuosi 2005, periodi 3, kurssinumero OHJ-3300) ja jotka ovat saaneet vähintään yhden lopullisen viikkoharjoituspisteen opintojaksolla (vuosi 2005, periodi 4, kurssinumero OHJ-3320).

Kurssilla on viisi (5) harjoituskertaa, joista jokaisesta voi saada "porkkanorta" tehdyistä tehtävistä. Tietokannassa on 'vkharjoitus' suoritetta vastaavan suorituksen kohdalla tallennettuna nimenomaan näitä porkkanapisteitä, joista lasketaan lopulliset harjoituspisteet 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-tauluun 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).