

## Tiedonhallintajärjestelmät T-76.1143

Exam 8.3.2006

1. We wish to design a database consistent with the following facts. Trains are either local trains or express trains, but never both. A train has a unique number and an engineer. Stations are either express stops or local stops, but never both. A station has a name (assumed unique) and an address. All local trains stop at all stations. Express trains stop only at express stations. For each train and each station the train stops at, there is a time.
  - a) Draw an ER diagram that captures this information.
  - b) Using SQL, create relations to hold the same information as the ER model.
  
1. Suunnittele tietokanta, joka sisältää seuraavat asiat. Junat ovat joko pikajunia tai paikallisjunia, mutta eivät koskaan molempia. Junalla ja veturilla on yksikäsitteinen tunniste. Asemat ovat joko pikajunien asemia tai paikallisjunien asemia, mutta eivät koskaan molempia. Jokaisella asemalla on yksikäsitteinen nimi ja osoite. Paikallisjunat pysähtyvät kaikilla asemilla, mutta pikajunat pysähtyvät ainoastaan pikajunien asemilla. Jokaiselle pikajunan ja paikallisjunan pysähdykselle asemalla on määritetty aika.
  - a) Laadi ER-kaavio, joka sisältää yllämainitut asiat.
  - b) Luo SQL:n avulla ER-kaaviosta taulut, jotka sisältävät ylläesitetyn informaation.
  
2. Consider a relation R with five attributes ABCDE. You are given the following dependencies:  $A \rightarrow B$ ,  $BC \rightarrow E$ , ja  $ED \rightarrow A$ .
  - a) List all keys for R.
  - b) Is R in BCNF? If not, decompose R into relations that are in BCNF.
  
2. Tarkastellaan viisiattribuuttista relaatiota  $R(A,B,C,D,E)$ , jossa ovat voimassa seuraavat funktionaaliset riippuvuudet:  $A \rightarrow B$ ,  $BC \rightarrow E$  ja  $ED \rightarrow A$ .
  - a) Anna kaikki R:n avaimet.
  - b) Onko R BCNF:ssä? Jos se ei ole, niin osita se BCNF:ssä oleviksi relaatioiksi.

3. The following relations keep track of airline flight information:
- Flights (fno, from, to, distance, departs, arrives, price)
  - Aircraft (aid, aname, cruisingrange)
  - Certified (eid, aid)
  - Employees(eid, ename, salary)
- (keys are underlined)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly.

- a) Using relational algebra write the query "The names of the pilots that are certified to fly an aircraft having cruisingdistance more than 3 000".
- b) Using SQL find the names of pilots whose salary is less than the price of the cheapest flight from Toronto to Honolulu.
- c) Using SQL find the number of the pilots.
- d) Using SQL print the name and salary of every non pilot whose salary is more than the average salary for pilots.

3. Seuraavat relaatiot sisältävät informaatiota lentoyhtiön lennoista.
- Lennot (lnum, mistä, minne, etäisyys, lähtöaika, tuloaika, hinta)
  - Konetyyppi (ktunniste, tyyppinimi, lentosäde)
  - Lentolupakirjat (ttunnus, ktunniste)
  - Työntekijä (ttunnus, tnimi, palkka)
- (avaimet on alleviivattu)

Huom. Työntekijätaulu ei kuvaa ainoastaan lentäjiä vaan kaikkia työntekijöitä. Jokaisella lentäjällä on lentolupakirja yhteen tai useampaan konetyyppiin.

- a) Esitä relaatioalgebralla kysely "Niiden lentäjien nimet, jotka ovat oikeutettu lentämään jotakin konetyyppiä, jonka lentosäde on yli 3000".
- b) Esitä SQL-kielellä kysely "Niiden lentäjien nimet, joiden palkka on pienempi kuin halvin lento Torontosta Honoluluun".
- c) Esitä SQL-kielen avulla kysely, joka tulostaa lentäjien lukumäärän.
- d) Esitä SQL-kielellä kysely "Niiden työntekijöiden nimet ja palkat, jotka eivät ole lentäjiä ja joiden palkka on enemmän kuin lentäjien keskipalkka".

**Interface Movie**

```

(extent Movies
key (title, year))
{attribute string title;
attribute integer year;
attribute integer length,
relationship Set<Star> stars
                inverse Star::starredIn;
relationship Studio ownedBy
                inverse Studio::owns;};

```

**Interface Star**

```

(extent Stars
key name)
{attribute string name;
relationship Set<Movie> starredIn;
                inverse Movie::stars;};

```

**Interface Studio**

```

(extent Studios
key name)
{attribute string name;
attribute string address;
relationship Set<Movie> owns
                inverse Movie::ownedBy;};

```

Figure 1 ( Kuvio 1).

4. Figure 1 represents a database in ODL. Using OQL find the names of the stars of Disney movies.
4. Kuviossa 1 on erään tietokannan ODL-kuvaus. Esitä OQL-kielellä kysely "Disneyn elokuvissa esiintyvien tähtien nimet".

**Maximum credits: task 1 (max20), task 2 (max 12), task 3 (max 20), task 4 (max 8).**