

Tiedonhallintajärjestelmät T-76.1143

Exam 13.5.2009

(1/4)

- Manufacturers have name, which we may assume unique, an address, and a phone number.
- Products have a model number and a type (e.g., "television set"). Each product is made by one manufacturer, and different manufacturers may have different products with the same model number. However, you may assume that no manufacturer would have two products with the same model number.
- Customers are identified by their unique social security number. They have email addresses, physical addresses. Several customers may live at the same (physical) address, but we assume that no two customers have the same email.
- An order has a unique order number, and a date. An order is placed by one customer. For each order, there are one or more products ordered, and there is a quantity for each product on the order.

Do the following:

- a) Draw an entity/relationship diagram that represents the above information. Indicate keys by underlying.
- b) Give a relational database schema (just the relations and attributes – not an SQL schema declaration) for the same information. Indicate keys by underlying.
- c) Suppose we had a relation Customer(ssNo, email, addr) as part of our relational database schema. What functional dependencies would you expect to hold?

1. Ensimmäinen tehtävä perustuu seuraavaan informaatioon.

- Kullakin tuottajalla on nimi, jonka voimme olettaa olevan yksilöivä, osoite ja puhelinnumero.
- Kullakin tuotteella on mallinumero ja tyyppi (esim. televisiovastaanotin). Jokaisen tuotteen on tehnyt yksi tuottaja, ja eri tuottajilla voi olla eri tuote samalla mallinumerolla. Voimme kuitenkin olettaa, ettei kenelläkään tuottajalla ole kahta tuotetta samalla mallinumerolla.
- Asiakkaat tunnustetaan yksilöivien henkilötunnustensa perusteella. Asiakkailla on myös sähköpostiosoite ja osoite. Useita asiakkaita voi asua samassa osoitteessa, mutta oletamme kuitenkin ettei kahdella asiakkaalla ole samaa sähköpostiosoitetta.
- Tilauksilla on yksilöivä tilausnumero, ja päiväys. Kukin tilaus liittyy yhteen asiakkaaseen. Kuhunkin tilaukseen liittyy yksi tai useampi tuote, ja jokaiseen tilattuun tuotteeseen liittyy kappalemäärä.

Tee seuraavat tehtävät:

- a) Esitä ER-kaavio, joka sisältää edellä mainitut asiat. Muista alleviivata avaimet.
- b) Esitä relaatiomallin mukainen tietokantakaava (pelkästään relaatiot ja niiden attribuutit – ei SQL kaaviomäärittelyä).
- c) Olettaen, että yksi tietokantakaavion relaatio on Asiakas(hetu, email, osoite). Mitä funktionaalisia riippuvuuksia siinä voi olettaa olevan voimassa?

2. A bank has the following relations, whose meaning should be apparent. Note, however, that an account can have several owners (i.e., joint accounts are allowed).

Accounts (accNo, type, balance)

Owens (acctNo, custNo)

Customers (custNo, name, addr)

Write the following in **relational algebra**. You may write complete expressions, or break your answer into several steps by assigning intermediate relations to temporaries.

- a) Find the types of accounts held by customer "Lee".
- b) Find all the "dangling accounts", that is, account numbers that appear in the Accounts relation, but either:
 1. Do not appear in the Owens relation, or
 2. Appears in the Owens relation, but every customer with which they are associated fails to appear in the Customers relation.

2. Pankkitietokanta sisältää seuraavan relaation, jonka merkitys ilmenee sen attribuuteista. Huomaa kuitenkin, että yhdellä tilillä voi olla monta omistajaa eli yhteistilit ovat sallittuja.

Tilit (tiliNo, tyyppi, saldo)

Omistaa (tiliNo, asNo)

Asiakkaat (asNo, nimi, osoite)

Esitä seuraavat kyselyt **relaatioalgebralla**. Voit esittää kyselyt relaatioalgebran lausekkeina tai jakamalla kyselyn osakyselyiksi ja yhdistämällä niiden tulokset lopullisessa kyselyssä.

- a) Mitä eri tyyppisiä tilejä on Lee-nimisellä asiakkaalla ?
- b) Etsi kaikki "isännättömät" tilit, eli tilinumerot, jotka esiintyvät Tilit relaatiossa, mutta jotka
 1. eivät esiinny Omistaa relaatiossa tai
 2. esiintyvät Omistaa relaatiossa, mutta asiakkaat, johon ne on kytketty eivät esiinny Asiakkaat relaatiossa.

3. Suppose we have a relation $R(a, b, c, d, e)$ with the following functional dependencies:

$ab \rightarrow c$, $cd \rightarrow e$, $c \rightarrow a$, and $e \rightarrow d$.

- a) What are all the keys for R ?
- b) Give an example of a functional dependency that is a BCNF violation for R . Into what two relations does this violation tell us to decompose R ?

3. Oletetaan, että meillä on relaatio $R(a, b, c, d, e)$ ja sen funktionaaliset riippuvuudet ovat:

$ab \rightarrow c$, $cd \rightarrow e$, $c \rightarrow a$, ja $e \rightarrow d$.

- a) Anna kaikki relaation R avaimet.
- b) Etsi sellainen funktionaalinen riippuvuus, joka rikkoisi relaatiota R olemasta BCNF:ssä. Millaisen osituksen se tuottaisi relaatiolle R ?

4. Consider the movie database with three relations, whose meaning should be apparent:

MovieDirector(title, director, year)
 MovieCast (title, actor, salary)
 MovieReview (title, reviewer, score)

All attributes are string except year, salary, and score, which are integers.

Write your SQL-queries so that the final answers will never include duplicates, but add the keyword DISTINCT only in cases where duplicates would otherwise be produced.

- Using subqueries in WHERE, find the directors who have also acted. Sort their names alphabetically.
- Without using subqueries in WHERE, find the actors who have never directed.
- Find the titles of all movies in the 1960's starring both Dustin Hoffman and Anne Bancroft.
- For each reviewer, find the average scores of all his/her reviews.
- Find the titles of the movies with exactly one review.
- Find the highest salary.
- Find the sum of salaries for each movie.

4. Tarkastellaan seuraavaa kolmesta relaatiosta muodostuvaa elokuvatietokantaa. Tietokannan sisällön merkityksen voi päätellä relaatioiden attribuuteista.

ElokuvaOhjaaja (elokuvanNimi, ohjaaja, vuosi)
 ElokuvaNäyttelijä (elokuvanNimi, näyttelijä, palkkio)
 ElokuvaArvio (elokuvanNimi, arvioija, pistemäärä)

Kaikki attribuutit ovat tyyppiä string paitsi attribuutit vuosi, palkkio, pistemäärä, jotka ovat tyyppiä integer. Esitä SQL-kyselysi niin, ettei kyselyn tulos sisällä duplikaatteja (toisteisia rivejä), mutta käytä määrettä DISTINCT ainoastaan niissä tapauksissa, jolloin se on välttämätöntä duplikaattien poistamiseksi.

- Käyttämällä alikyselyä WHERE-osassa, etsi ne ohjaajat, jotka ovat myös näyttelleet. Järjestä tulosjoukko aakkoselliseen järjestykseen.
- Käyttämättä alikyselyä, etsi ne näyttelijät, jotka eivät ole ohjanneet elokuvia.
- Etsi kaikki ne 1960-luvulla tuotettujen elokuvien nimet, joissa näyttelevät sekä Dustin Hoffman että Anne Bancroft.
- Etsi jokaisen arvioijan antamien pistemäärien keskiarvo.
- Etsi niiden elokuvien nimet, jotka on arvioinut vain yksi arvioija.
- Etsi korkein maksettu näyttelijän palkkio.
- Etsi elokuvakohtaisesti maksettujen palkkioiden summa.

5. Based on the schema of the Figure 1 give the following queries in OQL

- a) The length of the movie "Gone With The Wind".
- b) The names of the stars in the movie "Casablanca".
- c) The names of the stars of Disney studio movies. Eliminate duplicate names.

```

Interface Movie {
    attribute string title;
    attribute integer year;
    attribute integer length;
    relationship Set<Stars> stars
        inverse Stars::starredIn;
    relationship Studio ownedBy
        inverse Studio::owns;
};

Interface Star {
    attribute string name;
    attribute Struct Addr
        {string street, string city} address;
    relationship Set<Movie> starredIn
        inverse Movie::stars;
};

Interface Studio {
    attribute string name;
    attribute string address;
    relationship Set<Movie> owns
        inverse Movie::ownedBy;
};

```

Figure 1. ODL classes and their relationships

5. Laadi edellä esitetyn ODL-kuvauksen (Figure 1) pohjalta seuraavat OQL-kyselyt.

- a) Elokuvan Gone With the Wind pituus.
- b) Niiden näyttelijöiden nimet, jotka esiintyvät elokuvassa Casablanca.
- c) Disney studion tuottamien elokuvien näyttelijöiden nimet. Poista tuloksesta toisteiset nimet.

Maximum credits:

task 1 max 16
 task 2 max 8
 task 3 max 10
 task 4 max 14
 task 5 max 12