

Laskinten tai lisämateriaalin käyttö tentissä ei ole sallittua

1. a) (8 p) Tee seuraavan kuvauksen pohjalta UML-kaavio kirjaston tietokantaa varten. Käytä kurssilla opetettua notaatiota ja merkitse avainattribuutit kurssilla opetetulla tavalla. Tietokannassa säilytetään tietoa kirjaston asiakkaista, kirjoista, lainauksista ja varauksista. Tehtävän yksinkertaistamiseksi oletetaan, että kirjastosta voi lainata vain kirjoja ja että kirjastolla on vain yksi toimipiste.

Kirjastossa voi olla samaa kirjaa useita kappaleita, joita kutsutaan seuraavassa niteiksi. Jokaisesta kirjasta tiedetään sen ISBN-numero (yksikäsitteinen), nimi, ilmestymisvuosi, kustantajan nimi ja sivumäärä. Lisäksi tiedetään kirjan tekijä tai tekijät, joita siis voi olla useita. Kirjan tekijällä on yksikäsitteinen tunnusnumero ja lisäksi nimi sekä syntymävuosi. Kirjan niteellä on myös tunnusnumero, mutta se on yksikäsitteinen vain saman kirjan eri niteillä. Jokaiselle niteelle on määritelty myös maksimilainausaika, esimerkiksi 7 vuorokautta tai neljä viikkoa. Saman kirjan eri niteillä voi olla eri maksimilainausaika.

Kirjaston asiakkaalla on yksikäsitteinen asiakasnumero, nimi, syntymävuosi ja osoite. Lisäksi jokaisesta asiakkaasta tiedetään, mitä niteitä hänellä on lainassa ja mikä on lainojen lainauspäivä. Tietokannassa säilytetään myös tietoja jo palautetuista lainoista usean kuukauden ajalta. Näistä lainoista tiedetään lainassa olleen niteen lisäksi ainakin asiakas, lainauspäivä ja palautuspäivä. Sama asiakas voi lainata saman niteen useaan kertaan (jos hän välillä palauttaa sen), mutta ei samana päivänä.

Asiakkaasta tiedetään myös, mitä varauksia hänellä on. Varaus kohdistuu aina tiettyyn kirjaan, ei niteeseen, koska asiakkaalle ei ole väliä sillä, minkä kappaleen hän saa samanlaisista niteistä. Varauksista on tiedossa varauspäivä ja varauksen tila (esim. jonossa, odottaa noutoa tai varaus rauennut). Asiakas voi varata saman kirjan useaan kertaan, mutta ei samana päivänä.

b) (2 p) Muuta a-kohdan UML-kaavio relaatiomalliin. Esitä relaatiokaaviot ja alleviivaa niistä avainattribuutit (eli PRIMARY KEY -määreellä merkittävät attribuutit).

2. Tehtävä liittyy seuraavaan tietokantakaavioon, joka kuvaa erään kauppaketjun myymälöitä, asiakkaita, ruokatilauksia toimittavia robotteja ja toimitettavia tilauksia. Tehtävän yksinkertaistamiseksi tässä esitetään tietokannasta vain osa, josta puuttuvat esim. tiedot robottien reiteistä ja tilauksiin kuuluvista tuotteista.

Tietokantakaavio:

Customers(CID, name, email, phone, streetAddress)

Stores(SID, name, address)

Robots(RID, storeID, maintained, status)

Orders(OID, custID, storeID, robotID, orderTime, price, status)



Tauluun `Customers` on tallennettu asiakkaiden tiedot. Mukana voi olla myös sellaisia asiakkaita, jotka eivät ole vielä tehneet yhtään tilausta, mutta ovat vasta rekisteröityneet kauppaketjun järjestelmään. Jokaisesta asiakkaasta on tallennettu asiakastunnus, nimi, sähköpostiosoite, puhelinnumero ja katuosoite. Kauppaketjun myymälöiden tiedot on tallennettu tauluun `Stores`. Jokaisesta myymälästä on tallennettu myymälätunnus, nimi ja osoite.

Robottien tiedot on tallennettu tauluun `Robots`. Kunkin robotti toimittaa vain yhden myymälän tilauksia. Tämän myymälän tunnus on attribuutti `storeID`. Lisäksi roboteista on tallennettu robotin tunnus, viimeisen huollon päivämäärä sekä robotin tämänhetkinen tila, joka voi olla joko *available* (robotti on kaupassa valmiina odottamassa toimitettavaa tilausta), *packing* (robottiin ollaan pakkaamassa tilausta), *deliveringOK* (robotti on viemästä tilausta asiakkaalle, ja kaikki näyttää olevan kunnossa), *deliveringSTUCK* (robotti on viemässä tilausta asiakkaalle, mutta on jäänyt jumiin esim. painonapeilla toimivaan liikennevaloon), *returning* (robotti on palaamassa asiakkaan luota takaisin myymälään) tai *outOfOrder* (robotti on epäkunnossa).

Tilausten tiedot on tallennettu tauluun `Orders`. Taulussa on tiedot myös jo aikaisemmin toimitetuista tilauksista esim. tilastointia ja mahdollisten valitusten käsittelyä varten. Tilauksesta on tallennettu tilausnumero, tilauksen tehneen asiakkaan ja tilauksen myymälän tunnukset, tilauksen toimittavan robotin tunnus, tilausaika, tilauksen hinta (tilaukseen kuuluvien tuotteiden yhteenlaskettu hinta + toimitusmaksu) sekä tilauksen tila, joka voi olla joko *waiting* (tilaus odottaa käsittelyä), *started* (työntekijä on hakemassa tai pakkaamassa tilauksen tuotteita), *delivering* (robotti on toimittamassa tilausta), *delivered* (tilaus on toimitettu) tai *cancelled* (tilaus on peruutettu).

### Kirjoita seuraavat kyselyt SQL-kielellä:

a) (2 p)

Niiden asiakkaiden tunnukset, nimet ja katuosoitteet, joiden tilausta ollaan tällä hetkellä toimittamassa (tilauksen tila on *delivering*).

b) (2 p)

Niiden asiakkaiden tunnukset ja nimet, jotka ovat tehneet Halpis-Otaniemi-nimisestä kaupasta tilauksen, joka on jossain vaiheessa peruutettu (tilauksen tila on tietokannassa *cancelled*).

c) (2 p)

Niiden robottien tunnukset ja myymälätunnukset, jotka eivät ole koskaan toimittaneet yli 30 euron arvoista tilausta.

d) (2 p)

Haetaan robotit, joiden tila on *available* ja jotka ovat toimittaneet yli 300 tilausta (otetaan mukaan tilaukset, joiden tila on *delivered*). Kyselyn pitää tuottaa näistä roboteista robotin tunnus, sen myymälän tunnus ja tämän robotin toimittamien tilausten yhteishinta (kaikkien tämän robotin toimittamien tilausten hintojen summa - tähänkin otetaan mukaan ne tilaukset, joiden tila on *delivered*).

**JATKUU**



**Kirjoita seuraavat kyselyt relaatioalgebran lausekkeina:**

e) (2 p)

Kaikkien niiden asiakkaiden nimet ja katuosoitteet, joiden tilausta ollaan tällä hetkellä toimittamassa (tilauksen tila on *delivering*), mutta tilausta toimittava robotti on jäänyt jumiin (robotin tila on *deliveringSTUCK*).

f) (2 p)

Niiden asiakkaiden tunnukset ja nimet, jotka ovat tehneet robotilla toimitettavia tilauksia vähintään kahdesta eri myymälästä.

3. (4 p)

Oletetaan, että tehtävän 2 taulussa *Orders* on miljoonia rivejä eli miljoonien eri tilausten tiedot.

Taulun *Orders* rivit on järjestetty tilausajan eli attribuutin *orderTime* mukaan. Oletetaan edelleen, että myymäläketjulla on 10 eri myymälää ja tilauksia on tehnyt yli 200 000 eri asiakasta. Tietokannassa tehdään hyvin usein kyselyitä, joissa haetaan joko tietyn myymälän tai tietyn asiakkaan tilauksia eli kyselyitä, jotka ovat muotoa

```
SELECT *
FROM Orders
WHERE storeID = 'XXX';
```

tai

```
SELECT *
FROM Orders
WHERE custID = 'YYY':
```

Myymälöiden ja asiakkaiden tunnukset vaihtelevat eri kyselykerroilla eivätkä ole välttämättä samat kuin yllä olevissa esimerkeissä.

a) Kannattaako taululle *Orders* tehdä hakemisto (engl. *index*) attribuutin *storeID* mukaan? Perustele vastauksesi muutamalla lauseella.

b) Kannattaako taululle *Orders* tehdä hakemisto (engl. *index*) attribuutin *custID* mukaan? Perustele vastauksesi muutamalla lauseella.

4. Tarkastellaan relaatiota  $R(A, B, C, D, E)$ , jossa on voimassa funktionaaliset riippuvuudet  $A \rightarrow B$ ,  $B \rightarrow C$  ja  $D E \rightarrow B$ .

a) (1 p) Perustele, miksi relaatio ei ole Boyce-Codd-normaali muodossa (BCNF:ssä).

b) (6 p) Osita relaatio Boyce-Codd-normaali muotoon käyttämällä kurssilla (ja oppikirjassa) esitettyä algoritmia. Perustele lyhyesti jokainen muodostamasi uusi relaatio. Jatka osittamista niin pitkälle, että jäljellä on vain BCNF:ssä olevia relaatioita. Perustele, miksi lopulliset relaatiot ovat BCNF:ssä.



5.

- a) (2 p) Onko väite "Sarjallistuvuus (engl. isolation, serialization) tarkoittaa sitä, että transaktiot (engl. transactions) pitää aina suorittaa yksi kerrallaan alusta loppuun eikä kahta eri transaktiota voi suorittaa vuorotellen käsky kerrallaan." tosi vai ei? Perustele vastauksesi lyhyesti.
- b) (2 p) Tarkastellaan tehtävän kaksi kauppaketjun tietokantaa. Anna esimerkki kolmesta erityyppisestä virhetilanteesta, joka voi sattua, jos tietokannan hallintajärjestelmä ei toteuta transaktioilta vaadittavaa eheys-ominaisuutta (engl. consistency).
- c) (3 p) Selitä muutamalla lauseella, mikä ero on transaktioiden eristyvyystasoilla (engl. isolation level) READ COMMITTED, REPEATABLE READ ja SERIALIZABLE.