

# CSE-A1200 Tietokannat, kevät 2019

## Harjoitustyö, osa 1 (UML-mallinnus ja relaatiomalli)

### 1. Johdanto

Harjoitustyön tavoitteena on harjoitella relaatiotietokantojen suunnittelua ja toteutusta käytännössä. Harjoitustyö koostuu kahdesta palautettavasta osasta. Tässä osassa suunnitellaan tietokanta UML-kaavion avulla ja määritellään sitä vastaavat relaatiot. Toisessa osassa toteutetaan ensimmäisessä osassa suunniteltu tietokanta SQL-kielellä.

Harjoitustyö tehdään 2-3 hengen ryhmissä. Ryhmää voi etsiä esimerkiksi luennoilla, harjoituksissa tai kurssin Piazza-ryhmässä. **Ryhmätyö on pakollinen: työtä ei voi tehdä yksin.** Molemmat osat tehdään saman ryhmän kanssa.

Jos tässä tehtävänannossa on jotain epäselvää, voit kysyä tarkennuksia kurssin Piazza-keskusteluryhmässä. Tehtävänannossa ei ole kuitenkaan tarkoituksella määritelty kaikkia yksityiskohtia, koska osa harjoitustyön tekemistä on juuri yksityiskohtien suunnittelu.

### 2. Tehtävänanto

Harjoitustehtävänä on suunnitella tietokanta pääkaupunkiseudun kirjastoja varten. Toimintaympäristöä on kuitenkin yksinkertaistettu todellisesta.

#### Tehtävät

- Suunnittele UML-kaavio annetun informaation pohjalta. Käytä kurssilla opetettua notaatiota.
- Muunna kyseinen UML-kaavio relaatiomalliin (relaatiokaavioiksi). Esitä relaatiokaaviot ja alleviivaa niissä avainattribuutit.
- **Vastaa seuraaviin kysymyksiin ja perustele vastauksesi:** Mitä funktionaalisia riippuvuuksia tietokannassa vallitsee? Onko suunnitellussa tietokannassa anomaliaita? Onko tietokanta Boyce-Codd-normaalimuodossa? Jos ei ole, osita relaatiot Boyce-Codd-normaalimuotoon (palauta sekä ositettu että osittamaton versio).

### 3. Yksityiskohtaisempi kuvaus tietojärjestelmästä

#### - Toimintaympäristö

- Kirjastoon kuuluu useita eri toimipisteitä, joilla on yksikäsitteinen nimi. Lisäksi toimipisteistä tallennetaan muuta järjestelmän kannalta relevanttia tietoa (esim. osoite).
- Kirjastosta voi lainata kirjoja, lehtiä, CD-levyjä ja DVD-levyjä. (Tehtävän yksinkertaistamiseksi oletetaan, että muunlaisia lainattavia tyypejä ei ole.) Samaa teosta (kirja / lehti / CD / DVD) voi olla kirjastossa (samassa ja / tai eri toimipisteissä) useita kappaleita. Teoksesta tallennetaan olennaiset tiedot. (CD-levystä ei tarvitse tallentaa levyn sisältämiä yksittäisiä kappaleita, vaan levyn yleistiedot riittävät.)
- Kirjastolla on joukko asiakkaita, joilla on oikeus käyttää kirjaston palveluja. Asiakkaista tallennetaan tietokannan kannalta tarpeelliset tiedot.
- Asiakas voi myös varata teoksia. Jos teoksen kaikki kappaleet ovat lainassa, palautettuja

kappaleita toimitaan niitä varanneille asiakkaille varausjärjestyksessä. Teoksesta voi olla myös kappaleita, jotka eivät ole lainkaan varattavissa, vaan joita lainataan toimipisteestä ilman varausta.

- Jokaisella kappaleella on oma kotitoimipiste. Jos kappale ei ole lainattu tai varattu jollekin käyttäjälle, kappale palautetaan sen kotitoimipisteeseen.
- Teosten kappaleita kuitenkin kuljetetaan tarvittaessa toimipisteestä toiseen. Asiakas voi palauttaa lainan eri toimipisteeseen kuin mistä se on lainattu. Asiakas voi myös varata haluamansa teoksen, jolloin jokin sen kappale kuljetetaan asiakkaan haluamaan toimipisteeseen, josta asiakas voi noutaa sen.
- Eri kappaleilla voi olla erilaisia maksimilainausaikoja. Kirjaston tietokannassa voi myös olla kappaleita, joita ei voi lainata ollenkaan.
- Kirjasto voi kerätä maksuja asiakkailta, esimerkiksi myöhästymismaksuja ja varausmaksuja.

- Järjestelmässä pitää olla mahdollisuus esimerkiksi seuraaviin toimenpiteisiin (näitä toimenpiteitä ei vielä varsinaisesti esitetä UML-kaaviossa ja relaatiomallissa, mutta ne pitää suunnitella siten, että halutut toiminpiteet ovat mahdollisia)

- Lisätä uusia toimipisteitä, asiakkaita, teoksia ja niiden kappaleita.
- Tallentaa kappaleiden lainauksia ja palautuksia.
- Tallentaa tieto varauksista.
- Selvittää, missä toimipisteissä on vapaana halutun teoksen kappale.
- Selvittää, mitä kappaleita käyttäjällä on lainassa ja mitkä ovat lainojen viimeiset palautuspäivät.
- Selvittää, kenellä on aikaisemmin ollut lainassa palautettu kappale (jos sen esimerkiksi havaitaan rikkoutuneen). Aikaisemmat lainaajat pitää pystyä selvittämään vähintään viimeisen vuoden ajalta.
- Hakea teoksia sopivien hakuehtojen mukaan (esimerkiksi annetun kirjailijan teokset).
- Selvittää, mitä palautetulle kappaleelle pitäisi seuraavaksi tehdä (jättää palautustoimipisteeseen / palauttaa kotitoimipisteeseen / kuljettaa toiseen toimipisteeseen teoksen vuorossa olevalle varaajalle). Jos teoksen varaaja ei nouda hänelle varattua kappaletta määräajan kuluessa, kuljetetaan kappale seuraavalle varaajalle tai takaisin sen kotikirjastoon (jos teokseen ei kohdistu varauksia.)
- Tallentaa tieto tehtävistä kuljetuksista ja selvittää, mihin kuljetukseen jokin kappale pitäisi lisätä, jotta se menisi oikeaan toimipisteeseen.
- Selvittää, mitkä lainat ovat myöhässä ja selvittää lainan tehneet asiakkaat tarpeellisia jatkotoimenpiteitä (sakon määrääminen tai muistutuksen lähettäminen) varten.
- Tallentaa tieto asiakkaan tekemistä maksuista.
- Selvittää asiakkaan maksamatta olevat maksut ja mistä ne johtuvat (myöhästymissakko / varausmaksu). Maksusta ei tarvitse pystyä selvittämään, mihin yksittäiseen lainaan tai varaukseen se liittyy. Riittää, että maksun yleinen syy on tiedossa.

- Järjestelmän ei tarvitse pystyä esimerkiksi seuraaviin asioihin

- Kirjaston henkilökunnan tietojen käsittely.
- Toimipisteiden aukioloaikojen tms. käsittely.
- Kirjastosta poistettujen tai kadonneiden kappaleiden käsittely.
- Kuljetusten optimointi. Järjestelmän ei tarvitse optimoida kappaleiden kuljetuksia millään tavalla. Riittää, että se löytää jonkin kuljetuksen, jolla kappale saadaan siirrettyä haluttuun

toimipisteeseen. Mieti kuitenkin itse, millaisia kuljetusreittejä kannattaa järjestää ja miten tieto niistä tallennetaan, jotta sopiva kuljetus voitaisiin selvittää järkevästi.

Sinun on siis itse mietittävä, mitä tietoa toimipisteistä, teoksista, kappaleista, asiakkaista, kuljetuksista jne. tarvitaan, jotta kirjaston tietokanta voisi toimia halutulla tavalla.

#### 4. Arvosteluperusteet

Kaikki ryhmän jäsenet saavat työstä saman pistemäärän. Koko kevään harjoitustyösuoritus muodostuu kahdesta osasta, jotka molemmat ovat 20 pisteen arvoisia.

- Harjoitustyön hyväksytty suoritus vaatii yhteensä vähintään 20 pistettä sekä molempien osien palauttamisen.
- Mikäli saat vähintään 30 pistettä, saat yhden arvosanan korotuksen kurssiarvosanaan. (Tentistä tulee kuitenkin saada hyväksytty arvosana!)

Hyväksytty harjoitustyösuoritus ja siitä mahdollisesti saatu korotus on voimassa vielä kevätlukukauden 2020 ensimmäisessä tentissä.

Tämän harjoituskierroksen arvostelun maksimipistemäärä on 20 pistettä. Se jakautuu eri osiin seuraavasti:

- UML-kaavion oikeaoppisuus ja virheettömyys tehtävänantoon nähden 8 p.
- UML-kaavion muuntaminen relaatiomalliin 4 p.
- Funktionaaliset riippuvuudet ja normaalimuotoisuus 4 p.
- Dokumentoinnin laatu 4 p.

#### 5. Palautus

Harjoitustyö palautetaan A+-järjestelmään kohtaan Project Part I 3.4.2019 klo 20:00 mennessä. Korkeintaan 7 vuorokautta myöhässä palautetut työt arvostellaan, mutta myöhästymisestä vähennetään 3 pistettä, jos työ on alle 3 vuorokautta myöhässä ja 5 pistettä, jos työ on 3-7 vuorokautta myöhässä. Tätä myöhemmin palautettuja töitä ei arvostella, jos myöhästymiseen ei ole todella pätevää syytä (esim. pitkäaikainen sairaus, josta on lääkärintodistus).

Raportin tulee olla PDF-muodossa. Raportin voi kirjoittaa suomeksi, ruotsiksi tai englanniksi. Raportin rakenteen on oltava seuraava:

- Kansilehti, joka sisältää opiskelijoiden nimet ja sähköpostiosoitteet.
- UML-kaavio.
- UML-kaaviota vastaava relaatiokaavio.
- Selostus ratkaisusta (suositeltu pituus 3-4 sivua). Tässä kohdassa voit esimerkiksi perustella tekemiäsi ratkaisuja ja antaa esimerkkejä siitä, miten järjestelmässä pystytään toteuttamaan kohdassa 3 esitetyjä toimintoja.
- Perustellut vastaukset kohdassa 2 annettuihin kysymyksiin normaalimuodosta, funktionaalisista riippuvuuksista, anomaliaista sekä tarvittaessa relaatioiden osittaminen Boyce-Codd-normaalimuotoon.