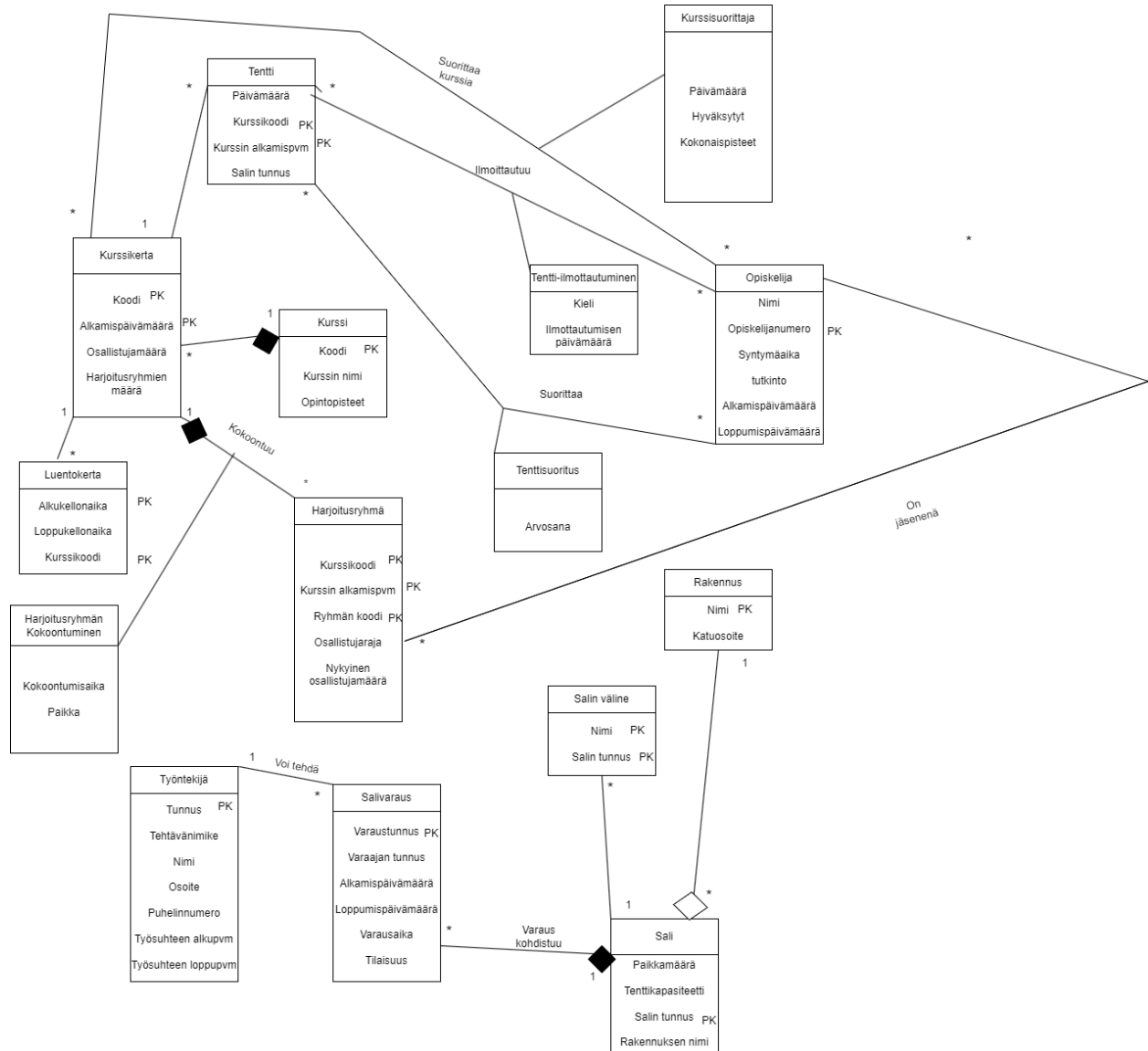


Topias Salokorpi – topias.salokorpi@aalto.fi

Paroj Kadel – paroj.kadel@aalto.fi

Granit Zogjani – granit.zogjani@aalto.fi

1. UML-kaavio



2. Relatiokaavio

Opiskelija (Nimi, Opiskelijanumero, Syntymäaika, Tutkinto, Alkamispäivämäärä, Loppumispäivämäärä)

Tentti (Päivämäärä, Kurssikoodi, Kurssin alkamispäivämäärä, Salin tunnus)

Tentti-ilmoittautuminen (Opiskelijanumero, Kurssikoodi, Kurssin alkamispäivämäärä, Kieli, Ilmoittautumisen päivämäärä)

Tenttisuoritus (Opiskelijanumero, Kurssikoodi, Suorituspäivämäärä, Arvosana)

Kurssikerta (Koodi, Alkamispäivämäärä, Osallistujamäärä, Harjoitusryhmien määrä)

Kurssi (Koodi, Kurssin Nimi, Opintopisteet)

Kurssisuorittaja (Kurssikoodi, Opiskelijanumero, Kurssin alkupäivämäärä, Hyväksytyt tentit kurssilla, Kokonaispisteet)

Luentokerta (Päivämäärä, Alkukellonaika, Loppukellonaika, Kurssikoodi)

Harjoitusryhmän Kokoontuminen (Kurssikoodi, Ryhmän koodi, Kokoontumisaika, Paikka)

Harjoitusryhmän jäsen (Kurssikoodi, Ryhmän koodi, Opiskelijanumero)

Harjoitusryhmä (Kurssikoodi, Kurssin alkamispäivämäärä, Ryhmän koodi, Osallistujaraja, Nykyinen Osallistujamäärä)

Työntekijä (Tunnus, Tehtävänimike, Nimi, Osoite, Puhelinnumero, Työsuhteen alkupäivämäärä, Työsuhteen loppupäivämäärä)

Salivaraus (Varaustunnus, Salin tunnus, Varaajan tunnus, Alkamispäivämäärä, Loppumispäivämäärä, Varausaika, Tilaisuus)

Salin väline (Nimi, Salin tunnus)

Sali (Paikkamäärä, Tenttikapasiteetti, Salin tunnus, Rakennuksen nimi)

Rakennus (Nimi, Katuosoite)

3. Selostus ratkaisusta:

Tietokantaan on koottu kaikki tehtävänannossa ilmoitetut luokat ja niiden assosiaatioluokat. Tietokannallamme voidaan vastata esimerkiksi seuraaviin tarpeisiin:

Tietokantaan täytyy olla mahdollista tallentaa tieto kursseista ja kurssikerroista, opiskelijoista, kurssin luennoista, harjoitusryhmistä ja niiden ajoista, tenteistä ja opiskelijoiden ilmoittautumisista, tenttisuorituksista ja harjoitustehtävapististä.

Tieto kursseista voidaan tallentaa relaatioon Kurssi. Kyseiseen relaatioon tallennetaan lisäksi kyseisen kurssin koodia vastaava nimi ja opintopisteet. Lisäksi yksittäiset kurssikerrat on mahdollistaa tallentaa omineen tarpeellisin tietoineen. Tämän lisäksi tietokannasta löytyy relaatio Opiskelija vastaamaan tietoihin opiskelijoista, Kurssin luento vastaamaan tietoa kurssin luennoista sekä Harjoitusryhmä vastaamaan tietyn kurssikerran harjoitusryhmiä. Harjoitusryhmä voidaan yhdistää kurssikertaan kurssikoodin ja kyseisen kurssikerran alkamispäivän avulla. Harjoitusryhmästä voidaan selvittää sen kokoontumisajat relaation Harjoitusryhmän kokoontuminen avulla.

Opiskelijoista on perustiedot relaatioissa Opiskelija. Lisäksi opiskelijan tenttisuoritukset voidaan selvittää relaation Tenttisuoritus avulla. Opiskelija ja hänen tenttisuorituksensa voidaan yhdistää uniikin opiskelijanumeron avulla. Lisäksi voidaan selvittää opiskelijan ilmoittautumiset relaation Tentti-ilmoittautuminen avulla. Opiskelijan yksittäistä kurssisuoritustilastoa vastaa relaatio Kurssisuorittaja. Tästä voidaan selvittää opiskelijan harjoitustehtävapistet ja tenttisuoritusten määrä kyseisellä kurssilla.

Tietokantaan täytyy olla mahdollista tallentaa tieto yliopiston rakennuksista, saleista ja salivarauksista. Varauksista pitää olla tallessa myös historiatieto eli pitää pystyä selvittämään esimerkiksi, minkä kurssin millekään harjoitusryhmälle sali oli varattu viime vuonna määrättyyn aikaan

Relaatioihin Rakennus, Sali ja Salivaraus voidaan tallentaa nimensä mukaisesti tiedot rakennuksista, saleista ja saleihin kohdistuvista varauksista. Oletuksena on, että yliopistosta ei löydy kahta samannimistä rakennusta. Sali voidaan yhdistää rakennukseen tämän uniikin nimen avulla. Oletuksena on myös, että kahdesta eri rakennuksesta ei löydy kahta saman tunnuksista salia. Tällöin salivaraus voidaan yhdistää saliin uniikin tunnuksen avulla. Salivarauksiin kirjataan, mitä tilaisuutta varten ne on tehty, joten tästä voidaan selvittää, mitkä salivaraukset ovat ryhmälle H02 kurssilla CS-1234. Tämä voidaan seuloa merkkijonomuotoisesta tilaisuusattribuutista.

Tietokannassa täytyy olla mahdollisuus tallentaa tieto yliopistojen työntekijöistä ja näiden tekemistä salivarauksista

Tietokannan relaatiosta Työntekijä löytyy perustiedot työntekijästä. Työntekijä voidaan yhdistää varaukseen uniikin tunnuksensa kautta.

Tietokannassa täytyy olla mahdollisuus selvittää, mitä kurseja järjestetään haluttuna ajankohtana

Yksittäiset kurssikerrat löytyvät tietokannasta alkamis- ja loppumisaikoihin. Käyttäjä voi rajata haluamansa ajankohdan osuvan tähän välille

Tietokannassa täytyy olla mahdollisuus selvittää, mitä tenttejä halutulla kurssilla on halutulla aikavälillä.

Tentti-relaatiossa on tieto tentin ajankohdasta ja siitä, mitä kurssia se vastaa. Käyttäjä voi rajata tietyn aikavälin ja kurssikoodin ja tätä kautta määrätä haluamansa tentit.

Tietokannassa täytyy olla mahdollisuus selvittää, koska haluttu kurssi on järjestetty tai koska se tullaan järjestämään

Tähän ongelmaan vastaa relaatio Kurssikerta. Kurssikerrasta voidaan määrittää sen alkamispäivämäärä ja tämän avulla voidaan selvittää halutut kurssit rajaamalla ne tietyllä kurssikoodilla.

Tietokannassa täytyy olla mahdollisuus selvittää, mitä luentoja etsittyyn kurssikertaan liittyy.

Relaatio Luentokerta voidaan yhdistää tiettyyn kurssikertaan kurssikoodin ja sen alkamispäivämäärän avulla. Tämän kautta voidaan selvittää haluttu kysely.

Tietokannassa täytyy olla mahdollisuus selvittää, mitä harjoitusryhmiä etsittyyn kurssikertaan liittyy ja koska ja missä haluttu harjoitusryhmä kokoontuu.

Luentojen tavoin relaatio Harjoitusryhmä voidaan yhdistää kurssikertaan kurssikoodin ja alkamispäivämäärän avulla. Harjoitusryhmän kokoontumispaikka ja -aika voidaan sen sijaan selvittää relaation Harjoitusryhmän kokoontuminen avulla, joka voidaan yhdistää kurssikertaan tai harjoitusryhmään kurssikoodin ja kurssin alkamispäivämäärän avulla.

Tietokannassa täytyy olla mahdollisuus etsiä sali, jossa on vähintään haluttu määrä paikkoja ja joka on vapaa haluttuna aikana

Halutut paikat voidaan selvittää tekemällä kysely relaatioon sali tietyin paikkamääräehdoin. Karteesisen tulon avulla voidaan yhdistää salit salivarauksiin. Yhdistäminen tapahtuu uniikin Salin tunnus – attribuutin avulla.

Tietokannassa täytyy olla mahdollisuus hakea tieto siitä, mihin tarkoitukseen määrätty sali on varattu haluttuna aikana tai kuka varauksen on tehnyt.

Salin varauksen syy voidaan selvittää relaation Salivaraus attribuutista tilaisuus. Haluttu aika voidaan seuloa kyseisen relaation attribuutin varausaika mukaan. Varaus voidaan yhdistää työntekijään. Yhdistäminen tapahtuu salivarauksen attribuutin varaajan tunnus ja työntekijän tunnuksen yhdistämällä.

Tietokannassa täytyy olla mahdollisuus ilmoittaa opiskelija johonkin tenttiin tai harjoitusryhmään. Tätä varten pitää pystyä selvittämään, että opiskelijalla ei ole jo kahta hyväksyttyä tenttisuoritusta tästä kurssista.

Opiskelija voidaan ilmoittaa tenttiin tai harjoitusryhmään relaatioiden Tentti-ilmoittautuminen ja Harjoitusryhmän osallistuja avulla. Kurssisuoritusten määrä voidaan selvittää tekemällä kysely relaatioon Kurssisuorittaja ja rajaamalla halutun opiskelijanumeron. Jos löytyy kurssisuorittaja, jolla on 2 hyväksyttyä tenttisuoritusta, tulee ilmoittautumista rajoittaa.

Tietokannassa täytyy olla mahdollisuus hakea kaikki halutulle kurssikerralle, haluttuun harjoitusryhmään tai haluttuun tenttiin ilmoittautuneet opiskelijat

Relaatioon Kurssisuorittaja lisätään opiskelija siinä vaiheessa, kun hän ilmoittautuu kurssille. Tässä samalla opiskelija liitetään johonkin kurssin harjoitusryhmään. Valitsemalla kaikki nämä 3 relaatiota ja valitsemalla vain opiskelijanumero attribuutiksi ja yhdistämällä tämä opiskelijaluokkaan, saadaan haluttu tulos.

Tietokannassa täytyy olla mahdollisuus etsiä halutulta kurssikerralta ne harjoitusryhmät, joissa on vielä tilaa

Tämä saadaan rajaamalla harjoitusryhmät aluksi haluttuun kurssikertaan oikealla koodilla ja kurssin alkamispäivämäärällä. Tämän jälkeen rajataan ne harjoitusryhmät, jossa attribuutti Nykyinen osallistujamäärä on pienempi kuin attribuutti Osallistujaraja.

Tietokannassa täytyy olla mahdollisuus merkitä opiskelijalle kurssin (tentti)suoritus ja selvittää, mitä suorituksia opiskelijalla on

Tämä on mahdollista relaation Tenttisuoritus avulla

Tietokannassa täytyy olla mahdollisuus Merkitä opiskelijalle harjoitustehtävapistettä jollekin kurssikerralle ja selvittää, kuinka monta harjoitustehtävapistettä opiskelija on saanut määrättyllä kurssikerralla.

Relaatioissa kurssisuorittaja on attribuutti harjoitustehtävapistettä varten. Näitä muuttamalla ja tutkimalla saadaan haluttu tulos

4. BCNF-tarkistus

Relaatioissa Opiskelija pätee funktionaalinen riippuvuus

Opiskelijanumero -> Nimi, Syntymäaika, Tutkinto, Alkamispäivämäärä, Loppumispäivämäärä

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Opiskelijanumero}\}^+ = \{\text{Opiskelijanumero, Nimi, Syntymäaika, Tutkinto, Alkamispäivämäärä, Loppumispäivämäärä}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatioissa Tentti pätee funktionaalinen riippuvuus

Päivämäärä Kurssikoodi -> Kurssin alkamispäivämäärä Salin tunnus

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Päivämäärä, Kurssikoodi}\}^+ = \{\text{Päivämäärä, Kurssikoodi, Kurssin alkamispäivämäärä, Salin tunnus}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatioissa Tentti-Ilmoittautuminen pätee funktionaalinen riippuvuus

Opiskelijanumero Kurssikoodi Kurssin alkamispäivä -> Kieli Ilmoittautumisen päivämäärä

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Opiskelijanumero, Kurssikoodi, Kurssin alkamispäivä}\}^+ = \{\text{Opiskelijanumero, Kurssikoodi, Kurssin alkamispäivä, Kieli, Ilmoittautumisen päivämäärä}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatioissa Tenttisuoritus pätee funktionaalinen riippuvuus

Opiskelijanumero Kurssikoodi Suorituspäivämäärä -> Arvosana

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Opiskelijanumero, Kurssikoodi, Suorituspäivämäärä}\}^+ = \{\text{Opiskelijanumero Kurssikoodi Suorituspäivämäärä, Arvosana}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatioissa Kurssikerta pätee funktionaalinen riippuvuus

Koodi Alkamispäivämäärä -> Osallistujamäärä Harjoitusryhmien määrä

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Koodi, Alkamispäivämäärä}\}^+ = \{\text{Koodi, Alkamispäivämäärä, Osallistujamäärä, Harjoitusryhmien määrä}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatiossa Kurssi pätee funktionaaliset riippuvuudet

Koodi -> Kurssin nimi Opintopisteet

Kurssin nimi -> Opintopisteet (Oletuksena on, että samanimiset, mutta eri koodilla olevat kurssit ovat saman kurssin eri versioita = yhtä paljon opintopisteitä)

Näiden vasempien puolten sulkeumiksi saadaan:

$\{Koodi\}^+ = \{Koodi, Kurssin\ nimi, Opintopisteet\}$

$\{Kurssin\ nimi\}^+ = \{Kurssin\ nimi, Opintopisteet\}$

Havaitaan, että kurssin nimen sulkeuma ei sisällä kaikki attribuutteja ja täten se ei ole yliavain. Ositetaan relaatio riippuvuuden Kurssin nimi -> Opintopisteet suhteen. Toiseksi saadaan:

$R1(Kurssin\ nimi, Opintopisteet)$ ja toiseksi $R2(Koodi, Kurssin\ nimi)$

Nyt voimassa on riippuvuudet

Koodi -> Kurssin nimi

Kurssin nimi -> Opintopisteet

Sulkeumat:

$\{koodi\}^+ = \{Koodi, Kurssin\ nimi$

$\{Kurssin\ nimi\}^+ = \{Kurssin\ nimi, Opintopisteet\}$

Nyt sulkeumat sisältävät kaikki attribuutit, joten relaatio on BCNF-muodossa.

Relaatiossa Kurssisuorittaja pätee riippuvuudet

Kurssikoodi Opiskelijanumero Kurssin alkupäivämäärä -> Hyväksytyt kurssin tentit Kokonaispisteet

Vasemman puolen sulkeumaksi saadaan:

$\{Kurssikoodi, Opiskelijanumero, Kurssin\ alkupäivämäärä\}^+ = \{Kurssikoodi, Opiskelijanumero, Kurssin\ alkupäivämäärä, Hyväksytyt\ kurssin\ tentit, Kokonaispisteet\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatiossa Luentokerta pätee riippuvuudet

Päivämäärä Kurssin koodi -> Alkukellonaika Loppukellonaika

Vasemman puolen sulkeumaksi saadaan:

$\{Päivämäärä, Kurssin\ koodi\}^+ = \{Päivämäärä, Kurssin\ koodi, Alkukellonaika, Loppukellonaika\}$

Koska vasemman puolenn sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatiossa Harjoitusryhmän kokoontuminen pätee riippuvuudet

Kurssikoodi Kokoontumisaika Ryhmän koodi -> Paikka

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Kurssikoodi, Kokoontumisaika, Ryhmän koodi}\}^+ = \{\text{Kurssikoodi, Kokoontumisaika, Ryhmän koodi, Paikka}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatiossa Harjoitusryhmän jäsen

Ei ole funktionaalisia riippuvuuksia, joten se on BCNF-muodossa

Relaatiossa Harjoitusryhmä pätee riippuvuudet

Kurssikoodi Kurssin alkamispäivämäärä Ryhmän koodi -> Osallistujaraja Nykyinen Osallistujamäärä

Vasemman puolen sulkeumaksi saadaan:

$\{\text{Kurssikoodi, Kurssin alkamispäivämäärä, Ryhmän koodi}\}^+ = \{\text{Kurssikoodi, Kurssin alkamispäivämäärä, Ryhmän koodi, Osallistujaraja, Nykyinen Osallistujamäärä}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaatiossa Työntekijä pätee riippuvuudet

Tunnus -> Tehtävänimike Nimi Osoite Puhelinnumero Työsuhteen alkupäivämäärä Työsuhteen loppupäivämäärä

Puhelinnumero -> Nimi Osoite

Lasketaan sulkeumat riippuvuuksien vasemmille puolille

$\{\text{Tunnus}\}^+ = \{\text{Tunnus, Tehtävänimike, Nimi, Osoite, Puhelinnumero, Työsuhteen alkupäivämäärä, Työsuhteen loppupäivämäärä}\}$

$\{\text{Puhelinnumero}\}^+ = \{\text{Nimi, Osoite}\}$

Havaitaan, että riippuvuus Puhelinnumero -> Nimi Osoite rikkoo BCNF-ehtoa. Ositetaan tämän suhteen. Tällöin uudet relaatiota ovat

R1(Puhelinnumero, Nimi, Osoite) R2(Tunnus, Puhelinnumero, Tehtävänimike, Työsuhteen alkupäivämäärä, Työsuhteen loppupäivämäärä)

Riippuvuuksien

Puhelinnumero -> Nimi Osoite

Tunnus -> Tehtävänimike Puhelinnumero Työsuhteen alkupäivämäärä Työsuhteen loppupäivämäärä

Vasempien puolten sulkeumiksi saadaan

$\{\text{Puhelinnumero}\}^+ = \{\text{Puhelinnumero}, \text{Nimi}, \text{Osoite}\}$

$\{\text{Tunnus}\}^+ = \{\text{Tunnus}, \text{Puhelinnumero}, \text{Tehtävänimike}, \text{Työsuhteen alkupäivämäärä}, \text{Työsuhteen loppupäivämäärä}\}$

Nyt riippuvuuksien vasenten puolten sulkeumat sisältävät kaikki attribuutit, joten relaatiot ovat BCNF-muodossa

Relaation Salivaraus riippuvuudet

Varaustunnus \rightarrow Varaajan tunnus Alkamispäivämäärä Loppumispäivämäärä Varausaika Tilaisuus

Vasemman puolen sulkeumaksi saadaan

$\{\text{Varaustunnus}\}^+ = \{\text{Varaustunnus}, \text{Varaajan tunnus}, \text{Alkamispäivämäärä}, \text{Loppumispäivämäärä}, \text{Varausaika}, \text{Tilaisuus}\}$

Riippuvuuden vasemman puolen sulkeuma sisältää kaikki attribuutit eli se on yliavain, joten relaatio on BCNF-muodossa.

Relaation Salin väline riippuvuus

Kyseisellä relaatioilla ei ole epätriviaaleja riippuvuuksia, joten se on BCNF-muodossa

Relaation Sali riippuvuus

Salin tunnus \rightarrow Rakennuksen nimi Paikkamäärä Tenttikapasiteetti

Vasemman puolen sulkeumaksi saadaan

$\{\text{Salin tunnus}\}^+ = \{\text{Salin tunnus}, \text{Rakennuksen nimi}, \text{Paikkamäärä}, \text{Tenttikapasiteetti}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

Relaation Rakennus riippuvuus

Nimi \rightarrow Katuosoite

Vasemman puolen sulkeumaksi saadaan

$\{\text{Nimi}\}^+ = \{\text{Nimi}, \text{Katuosoite}\}$

Koska vasemman puolen sulkeuma sisältää kaikki attribuutit, on se relaation yliavain ja täten relaatio on BCNF-muodossa.

5. Muita anomaliaita:

Kurssisuorittajia jää historiaan kaikilta kursseilta, jolloin myös vanhat hyväksytyt kurssin tentit jäävät voimaan. Siis vanhentunutta tietoa jää voimaan. Tämä ei mitä luultavimmin ole tarkoituksenmukaista

TOINEN OSA ALKAA TÄSTÄ

6. Muutokset ensimmäiseen osaan

- Relaatiomallissa harjoitusryhmällä ei ole kurssin alkupäivämäärää. Tämä lisätään.
- Harjoitusryhmän kokoontuminen ei ole enää assosiaatioluokka, vaan oma luokkansa
- Salivarauksen ja työntekijän välille assosiaatioluokka TekiVarauksen(varaajan tunnus, varaustunnus). Tämän seurauksena luokasta salivaraus on poistettu attribuutti varaajan tunnus.
- Salin ja rakennuksen välille assosiaatioluokka SaliSijaitsee(Salin tunnus, rakennuksen nimi). Tämän seurauksena luokasta sali poistettu attribuutti rakennuksen nimi.
- Luokkien ja attribuuttien nimien rakennetta muutettu SQL:ään sopivaksi (esim. Välilyönnit pois)

7. Tietokannan luonti SQL-komennoilla

- Löytyy tiedostosta create-table-commands.sql

Perustelut käytetyille tietotyypeille

CREATE TABLE –komentoja tutkiessa voidaan havaita, että käytössä on vain kahta tietotyyppiä: TEXT ja INTEGER. Tämä johtuu siitä, että SQLitessä on vain neljä tietotyyppiä, joista REAL:ia ja BLOB:ia ei tarvita, sillä minkään arvon ei tarvitse olla desimaali- tai binääriluku tietokannassa.

Tietotyyppiä TEXT on käytetty silloin, kun data tulee suurella todennäköisyydellä sisältämään kirjaimia. Lisäksi kyseistä tietotyyppiä käytetään joissain erikoistapauksissa, joissa ei esiinny kirjaimia, kuten päivämäärät ja opiskelijanumerot. Tämä johtuu siitä, että vaikka kyseessä ovat numerosarjoja, ei niillä suoriteta matemaattista laskentaa, mistä johtuen on koherentimpaa pitää ne TEXT-tietotyyppinä. INTEGER-tietotyyppiä on käytetty esimerkiksi pisteiden ja arvosanojen esittämisessä. Tämä johtuu siitä, että kyseessä on matemaattiseen käyttötarkoitukseen tarkoitettuja arvoja. (Esim. Pisteitä tulee voida lisätä).

8. Hakemistot ja näkymät

- Hakemistot Löytyvät tiedostosta create-table-commands.sql tietokannan luontikomentojen jälkeen

Perustelut luoduille hakemistoille

Relaatiolle Opiskelija on luotu hakemistot nimen ja opiskelijanumeron perusteella. Tämä johtuu siitä, että opiskelijanumero on yksilöllinen sekä avainattribuutti, joten ei ole järkeä selata koko levyä läpi etsien yhtä tiettyä opiskelijaa. Toisaalta vaikka opiskelijoita olisi runsaasti, löytyisi tietokannasta enintään muutama samanniminen opiskelija, jolloin myös nimen perusteella on järkevää luoda hakemisto. Nämä ovat haut, joita pääsääntöisesti kohdistetaan kyseiseen relaatioon. Sen sijaan esimerkiksi opiskelijan syntymävuoteen perustuvat haut eivät juurikaan ole kiinnostavia ja aloitus- ja lopetusvuoteen liittyviin hakuihin liittyy sen verran suuri määrä opiskelijoita, että hakemiston luominen ei ole kannattavaa.

Relaatiolla Kurssi on hakemistot nimelle sekä kurssikoodille. Kurssikoodi on relaation monikkoja yksilöivä avainattribuutti, mistä johtuen siitä on järkevää tehdä hakemisto. Toisaalta nimen perusteella voidaan tehdä hakuja varsinkin silloin, kun opiskelija on kiinnostunut tietyn aihepiirin kursseista. Samannimisiä kursseja ei kuitenkaan löydy useimmissa tapauksissa kovinkaan paljon, mistä johtuen olisi tehotonta selata koko levy läpi etsien muutamaa kurssia. Opintopisteiden perusteella ei ole järkevää tehdä hakemistoa, sillä vaikka tämän attribuutin perusteella saattaakin kohdistua kyselyitä, sijoittuvat arvot

lähes kaikissa tapauksissa välille 1 ja 10, mistä johtuen hakemiston luominen ei todennäköisesti lisäisi tehokkuutta.

Relaatioon Rakennus on luotu hakemisto vain rakennuksen nimen perusteella. Kyseessä on relaation monikoita yksilöivä avain, joten hakemisto on järkevä. Toisaalta rakennuksen nimi on hyvä keino etsiä haluttu rakennus. Katuosoitteen perusteella ei ole tehty hakemistoa, sillä rakennusten katuosoitteita ei mitä luultavimmin ole niin paljon, että hakemisto nopeuttaisi kyselyä, vaikka joissain tapauksissa voisi olla kiinnostavaa selvittää, millä kadulla sijaitsee mitäkin rakennuksia.

Relaatioon Sali on luotu hakemistot salin tunnuksen ja paikkamäärän perusteella. Salin tunnus on relaation monikoita yksilöivä avainattribuutti, mistä johtuen hakemiston luominen tämän attribuutin perusteella on järkevää. Lisäksi paikkamäärän perusteella on luotu hakemisto, sillä usein voi olla kiinnostavaa selvittää, mihin saleihin mahtuu haluttu määrä ihmisiä. Paikkamäärä ei teoriassa ole rajattu, joten kyseinen attribuutti voi saada hyvinkin laajalta skaalalta arvoja.

Relaatiolle Tentti on luotu hakemistot päivämäärän ja kurssikoodin perusteella. Relaatioon tentti on kiinnostavaa kohdistaa kyselyitä kurssikoodin perusteella, kun halutaan esimerkiksi tietää, milloin on jonkin tietyn kurssin tentit. Toisaalta työntekijää voi kiinnostaa omaa kurssia suunnitellessa, mitä tenttejä jonain päivänä on. Sen sijaan harvemmin ollaan kiinnostuneita, mitä tenttejä on jossain salissa tai siitä, milloin on tiettynä hetkenä alkaneen kurssikerran tentti. (Tämä sama asia voidaan selvittää kyselyllä kurssikoodin perusteella)

Relaatiolle Tenttillmottautuminen on luotu hakemistot kurssikoodin/kurssin alkamispäivämäärän sekä opiskelijanumeron perusteella. Kyseessä on assosiaatioluokka relaatioille tentti ja opiskelija. Pääasiallisesti kyselyissä ollaan kiinnostuneita rajaamaan, että ketkä opiskelijat ovat ilmoittautuneet tietyn kurssikerran tenttiin tai että mihin kurssiin tietty opiskelija on ilmoittautunut. Toisaalta kyseessä on avaimen muodostavat attribuutit. Näistä syistä näille on järkevää luoda hakemistot

Tenttisuoritukselle on luotu hakemistot kurssikoodille sekä opiskelijanumerolle. Kyseessä on muuten samat hakemistot/syyt hakemistoille paitsi kurssikoodilla ei ole parina kurssin alkamispäivämäärää. Tämä johtuu siitä, että tenttisuorituksia tutkiessa yleisesti kiinnostaa tietyn kurssin tenttisuoritukset eikä niinkään tietyn kurssikerran tenttisuoritukset.

Relaatiolle Kurssikerta on hakemisto attribuuttiparille kurssikoodi ja kurssin alkamispäivämäärä. Tämä johtuu siitä, että kyselyjä ilmenee pääasiassa kahdenlaisina: halutaan tietää, milloin jokin kurssi järjestetään, jolloin kysely tehdään pelkän kurssikoodin avulla. Toisaalta voidaan haluta tietoa tietyn kurssin osallistujien ja harjoitusryhmien määrästä, jolloin halutaan hakea kummankin avainattribuutin perusteella juuri haluttu kurssi. Kuitenkaan pelkän alkamispäivämäärän perusteella ei saada juurikaan kiinnostavaa tietoa, joten se voi olla hakemiston parina jälkimmäinen. (Hakemiston jälkimmäisen parin yksittäiseen kyselyyn ei ole hyötyä kyseisestä hakemistosta.). Kurssin osallistujien ja harjoitusryhmien määrän kyselyt eivät tuota juuri mitään käytännössä hyödyttävää informaatiota, joten ne eivät tarvitse hakemistoja.

Relaatioissa Kurssisuorittaja on hakemistot kaikille sen avaimille: opiskelijanumerolle, kurssikoodille sekä kurssin alkupäivämäärälle. Pelkkään opiskelijanumeroon voidaan kohdistaa kysely, kun ollaan kiinnostuneita, mitä kaikkia kurssikertoja opiskelija on ollut suorittamassa. Toisaalta voidaan haluta yhdistää kysely sekä opiskelijanumeroon, tiettyn kurssin alkamispäivämäärään sekä tiettyn

kurssikoodiin, jolloin saadaan opiskelijan kurssisuoritustiedot juuri tiettyyn kurssikertaan. Toisaalta voidaan olla kiinnostuneita pelkästään tietyn kurssin tai kurssikerran suorittajien tiedoista, jolloin myös sekä kurssikoodi että kurssin alkamispäivämäärä ansaitsevat omat hakemistot.

Relaatiolla Luentokerta on hakemisto sekä päivämäärälle, että kurssikoodille. Nämä hakemistot ovat olemassa sillä molemmat ovat avainattribuutteja. Lisäksi hakemistoja tarvitaan, koska voidaan olla kiinnostuneita hakemaan tiettyinä päivinä tapahtuvat luennot tai milloin jonkin tietyn kurssin luennot ovat. Sen sijaan alku- ja loppukellonaika antavat turhan spesifistä tietoa siihen, että niillä olisi kiinnostava tehdä kyselyitä, jolloin ne eivät tarvitse hakemistoja.

Relaatiolla harjoitusryhmanKokoontuminen on hakemistot kaikille sen attribuuteille paitsi paikalle, sillä ne ovat sen attribuutit ja toisaalta kaikkien näiden perusteella voidaan tehdä tarkoituksenmukaisia kyselyitä. Voidaan esimerkiksi tutkia, milloin jotain kurssikoodia vastaavat harjoitusryhmät kokoontuvat. Toisaalta voidaan etsiä ryhmän ja kurssikoodin yhdistelmän perusteella tarkka kokoontumisaika (muotoa "pvm-kellonaika") halutulle harjoitusryhmälle. Toisaalta voidaan määrittää esimerkiksi jokin kokoontumisaikojen väli ja tutkia mitkä ryhmän tietyllä kurssilla kokoontuvat tällä aikavälillä ja missä.

Relaatiossa harjoitusryhmanJasen on hakemisto opiskelijanumeron perusteella. Tämä johtuu siitä, että tässä relaatiossa ollaan pääasiassa kiinnostuneita tutkitamaan, missä kaikissa harjoitusryhmissä jokin opiskelija on osana. Lisäksi relaatiossa on hakemisto kurssikoodin, kurssin alkamispäivämäärän sekä ryhmän tunnuksen kolmikolle, sillä voidaan myös olla kiinnostuneita tutkimaan, kuka opiskelija löytyy mistäkin harjoitusryhmästä.

Relaatiossa Harjoitusryhma on luotu hakemisto kurssikoodin ja kurssin alkamispäivämäärän perusteella. Nämä attribuutit ovat osa relaation avainta. Kyseistä relaatiota voidaan esimerkiksi käyttää selvittämään, paljonko tilaa kussakin kurssin harjoitusryhmässä on.

Työntekijä-relaatiossa on hakemisto sekä työntekijän tunnukselle että nimelle. Näiden perusteella voidaan identifioida työntekijä, joten myös kyselyt kohdistetaan jompaankumpaan näistä. Samannimisiä työntekijöitä on enintään muutama, joten olisi tehotonta käydä koko levy läpi etsiessä muutamaa nimeä. Tunnus sen sijaan on uniikki jokaisella työntekijällä.

Salivarauksella on hakemistot varaustunnus- ja salitunnusattribuuteille. Lisäksi löytyy hakemisto varauksen alkamisen ja loppumisen attribuuteille. Varaustunnuksen avulla voidaan hakea tietty salivaraus. Aina ei kuitenkaan tiedetä tarkkaa varaustunnusta, vaan ollaan esimerkiksi kiinnostuneita, onko tiettyyn saliin kohdistuvia varauksia tietyllä aikavälillä. Toisaalta voidaan selvittää, mitkä salit ovat varattuina minäkin aikana.

Relaatiolla Salinvaline on hakemistot sen kumpaankin attribuuttiin (jotka muodostavat relaation avaimen). Voidaan olla sekä kiinnostuneita tutkimaan, mitä välineitä tietyssä salissa on ja toisaalta tutkimaan esimerkiksi, että missä saleissa sijaitsevat yliopiston tulostimet.

Tekivarauksen ja Salisijaitsee –relaatioilla on hakemistot niiden molemmille avainattribuuteille. Kyseessä on assosiaatioluokat, jolloin liitokset muodostetaan näiden attribuuttien avulla. Tällöin avaimet yksilöivät monikot ja kyselyt kannattavat nopeuttaa niihin perustuvilla hakemistoilla

Luodut näkymät

-Näkymien määrittelyt löytyvät tiedostosta create-table-commands.sql hakemistojen luonnin jälkeen

Tietokannassa on kaksi näkymää. Näistä ensimmäinen, TIKopiskelijat, listaa kaikki opiskelijatiedot kaikista opiskelijoista, joiden tutkinto on tietotekniikka. Näin voidaan tutkia, ketkä kaikki suorittavat tietotekniikan tutkintoa ja kyselyä voidaan spesifioida näihin. Näkymä työntekijöidenVaraukset yhdistää kaikki salivaraukset ne tehneeseen työntekijään ja antaa attribuutteina varaajan nimen ja tunnuksen, salin tunnuksen sekä varauksen alkamis- ja loppumispäivämäärät. (Nämä pitävät sisällään myös kellonajan). Tällä tavalla voidaan tutkia, kuka on varannut minkäkin salin mihinkin aikaan ja näitä määreitä voidaan myös rajata.

9. Käyttötapausten SQL-kyselyt

-Kaikki käyttötapauksiin liittyvät kyselyt ja sql-komennot löytyvät tiedostosta database-usage.sql. Kyseisessä tiedostossa ennen kyselyä on aina ilmoitettu kommentissa, mistä kyselystä on kyse.

Käyttötapaus 1

Käyttötapauksessa 1 aluksi lisätään tietokantaan uusia opiskelijoita, minkä jälkeen haetaan kaikkien opiskelijoiden nimi ja tutkinto, jotka ovat aloittaneet vuonna 2020 tai sen jälkeen.

Ennen kyselyä tietokantaan on lisätty 2 henkilöä, Pasi ja Tero. Pasi aloitti vuonna 2015, kun taas Tero vuonna 2020. Odotetusti kyselyn tuloksena on Tero.

Käyttötapaus 2

Tero vaihtaa tutkintoansa tietotekniikasta teknilliseen fysiikkaan. Tämän vuoksi tietokannassa on nyt vanhentunutta tietoa, joka halutaan päivittää ajan tasalle. Suoritetaan siis Teron tutkinnon päivittäminen tietokannassa.

Käyttötapaus 3

Kolmannessa käyttötapauksessa lisätään tietokantaan yliopiston kurssivalikoimaan kuuluvia kursseja. Kyselyssä halutaan etsiä näistä kursseista nimet niistä, joista saatavat opintopisteet ovat 5 tai enemmän. Tällainen kysely voisi olla relevantti esimerkiksi opiskelijan etsiessä uusia suoritettavia kursseja.

Tuloksena tulee kurssi C#:n alkeet, kuten odotettiin, sillä sen opintopisteet ovat 15 ja toisaalta laaja suhteellisuusteoria jäi ulkopuolelle, sillä opintopisteitä on vaivaiset 3.

Käyttötapaus 4

Yliopistolle on tullut uusi rakennus, joka toi mukanaan uusia luentosaleja. Päivitetään tietokanta sisältämään nämä sekä päivitetään assosiaatioluokka SaliSijaitsee pitämään paikkansa. Kurssin pitäjä etsii näistä uusista saleista sopivan tenttiänsä varten. Hän haluaa tenttikapasiteetin olevan vähintään 30. Siis suoritetaan kysely, jossa rajataan relaatiota Sali niin, että tenttikapasiteetti on ainakin 30.

Tuloksena saadaan odotetut T1 ja T2

Käyttötapaus 5

Halutaan laskea, montako aktiivista opiskelijaa tietotekniikalla on. Toisin sanoen halutaan löytää opiskelijat, joiden lopetuspäivä on joko NULL tai myöhemmin kuin tämänhetkinen päivämäärä (6.5.2023). Voidaan hyödyntää näkymää "TIKopiskelijat". Lisätään tätä varten yksi tietotekniikan opiskelija tietokantaan, sillä Teron päivityksen jälkeen niitä ei ole yhtäkään.

Tuloksena saadaan odotetusti tämä uusi opiskelija, Sirpa.

Käyttötapaus 6

Professori haluaa lisätä tietokantaan tiedon siitä, milloin laaja suhteellisuusteoria ja C# alkeet järjestetään. Täten hän lisää uusia rivejä relaatioon Kurssikerta. Opiskelija haluaa selvittää, milloin laajan suhteellisuusteorian kurssia järjestetään. Tämä selviää kyselyllä, jossa rajataan kurssin nimeksi tämä. Kyselyssä tarvitaan sekä relaatiota Kurssi että Kurssikerta.

Tuloksena saadaan kaikki halutut monikot.

Käyttötapaus 7

Professori haluaa lisätä harjoitusryhmät, niiden kokoontumiset sekä tentit molemmille laajan suhteellisuusteorian kursseille. Suoritetaan tietokantaan tarvittavat lisäykset.

Käyttötapaus 8

Opiskelija haluaa ilmoittautua laajan suhteellisuusteorian kurssille sekä sen harjoitusryhmään ja tenttiin. Lisätään opiskelija tarvittaviin relaatioihin ja päivitetään kurssin osallistujamäärä.

Käyttötapaus 9

Kurssinpitäjä haluaa selvittää toisen laajan suhteellisuusteorian kurssikerran harjoitusryhmien opiskelijamäärät. Hän haluaa tulokseen jokaisen harjoitusryhmän nimen ja näitä vastaavan opiskelijamäärä. Lisäksi tulosta selkeyttämään halutaan järjestää tulokset osallistujamäärän mukaan. Tulos on se, mikä pitikin.

Käyttötapaus 10

Laajan suhteellisuusteorian assistentti haluaa selvittää hänen ryhmänsä opiskelijoiden nimet ja opiskelijanumerot. Kyseinen assistentti pitää laajan suhteellisuusteorian ensimmäisen kurssikerran ryhmää H01. Suoritetaan tähän tarpeeseen vastaava kysely. Tuloksena saadaan Tero, kuten odotettiin.

Käyttötapaus 11

Työntekijä haluaa selvittää kaikkien luentosalien tunnuksat ja paikkamäärät, jotka sijaitsevat rakennuksessa "talo". Suoritetaan käyttötapausta vastaava kysely. Tuloksena saadaan salit T1, T2 ja T3, mikä oli odotettua.

Käyttötapaus 12

Yliopisto saa uusia tulostimia, jotka sijoitetaan luentosaleihin T1 ja T3. Lisätään nämä tietokantaan. Opiskelija haluaa selvittää, mihin saleihin nämä tulostimet laitettiin. Suoritetaan kysely. Tuloksena saadaan oikeat salit.

Käyttötapaus 13

Työntekijä Salla varaa salin T1 kahdelle eri ajankohdalle. Toinen työntekijä Matti varaa salin T2 yhdelle ajankohdalle. Kolmas henkilö haluaa selvittää, mitkä varaukset Matti on tehnyt. Lisätään työntekijät ja tarvittavat varaustiedot tietokantaan ja suoritetaan kysely. Tuloksena saadaan Matin tekemä varaus

Käyttötapaus 14

Laajan suhteellisuusteorian alkeiden ensimmäisen kurssikerran tentin järjestäjä haluaa selvittää, montako opiskelijaa on ilmoittautunut kutakin tenttikieltä kohden. Suoritetaan kysely. Tulos näyttää halutunlaiselta.

Käyttötapaus 15

Laajan suhteellisteorian kurssin pitäjä haluaa valita kurssiassistentiksi opiskelijoita, joilta löytyy kyseisen kurssin tentti suoritettuna arvosanalla 4. Hän haluaa listaukseen nimen ja opiskelijanumeron.

Käyttötapaus 16

Laajan suhteellisuusteorian professori haluaa tutkia tenttensä vaikeustasoa. Hän haluaa selvittää, mikä on tenttisuoritusten keskiarvo kutakin suorituspäivämäärää kohden. Keskiarvot vastaavat tässä kohdassa lisättyjä arvoja.

Käyttötapaus 17

Professorin lempisalissa ilmenee putkivika, joten se haisee, minkä vuoksi professori ei voi käyttää sitä. Hän haluaa etsiä kaikki luokat, jotka ovat vapaana 20.4.2023 kello 8–11 välisenä aikana rakennuksessa “talo” ja sieltä täytyy löytyä iso älytaulu. Tuloksena saadaan T1, joka täyttää ehdot.

Käyttötapaus 18

Uuteen työpaikkaan päässyt opiskelija on hyvin kiireinen, joten hänellä on hyvin tarkat ajat, jolloin hän kerkeää käydä yliopistolla. Tästä johtuen hän haluaa löytää kurssien nimet ja alkamispäivämäärät, joista löytyy harjoitusryhmä, jonka kokoontumisaika on joko kello 14 tai 16. Hän haluaa vain kurssit, jotka alkavat 1. tammikuuta 2023 jälkeen. Kyselyn suorittamalla saadaan haluttu tulos, laajan suhteellisuusteorian ensimmäinen kurssikerta.

Käyttötapaus 19

Professoria kiinnostaa jakauma eri tutkintoalojen välillä hänen laajan suhteellisuusteorian kurssillaan. Hän haluaa selvittää keskiarvon hänen kurssinsa suorittajan pisteille jokaista tutkintoalaa kohden. Tuloksena saadaan teknilliselle fysiikalle odotettu 22 ja tuotantotaloudelle 13.

Käyttötapaus 20

Laajan suhteellisuusteorian professori päättää uuden menetelmän tulevien kurssiassistenttien etsimiseen. Hän haluaa jokaiselta tutkintoalalta tenttinsä parhaan arvosanan saajan. Näistä hän haluaa vielä katsoa kokonaiskeskiarvoa. Tulokseen halutaan näiden opiskelijoiden nimi, opiskelijanumero sekä tenttisuoritusten keskiarvo. Mikäli parhaita opiskelijoita on useita, valitaan ne kaikki. Tuloksena saadaan odotetusti Pasi, Tero ja Sirpa oikeineen keskiarvoineen. Tässä käyttötapauksessa uusi opiskelija Salla ei sisälly tulokseen, sillä hän ei ole suorittanut kyseistä kurssia.