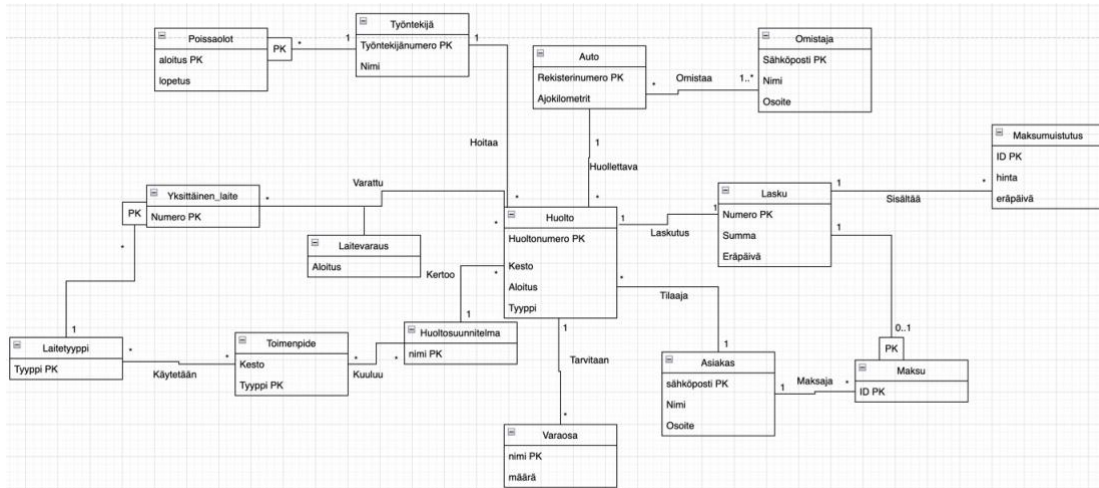


CS-A1150 Tietokannat

Harjoitustyö, osa 1 & 2

7.5.2022



Relaatiot:

Huolto(huoltoNo, aloitusPäivä, aloitusAika, lopetusAika, tyyppi, rekisteriNo, laskuNo, asiakasSposti, huoltoNimi, työntekijäNo)

Auto(rekisteriNo, kilometrit)

Omistaa(rekisteriNo, heTu)

Omistaja(hetu, nimi, osoite)

Lasku(no, summa, eräpäivä)

Maksumuistutus(id, hinta, eräpäivä, laskuNo)

Maksu(id, laskuNo, asiakasSposti)

Asiakas(sposti, nimi, osoite)

Varaosa(nimi)

Tarvitaan(nimi, huoltoNo, määrä)

Huoltosuunnitelma(nimi)

Kuuluu(huoltoNimi, toimenpideTyyppi)

Toimenpide(tyyppi, kesto)

Käytetään(toimenpideTyyppi, laitetyyppi)

Laitetyyppi(tyyppi)

Yksittäinen_laite(laiteNo, laitetyyppi)

Laittevaraus(aloitusPäivä, aloitusAika, lopetusAika, huoltoNo, laiteNo, laitetyyppi)

Työntekijä(työntekijäNo, nimi)

Poissaolot(työntekijäNo, aloitus, lopetus)

Epätriviaalit funktionaaliset riippuvuudet (relaatioittain):

Huolto: huoltoNo -> aloitusPäivä aloitusAika lopetusAika tyyppi rekisteriNo asiakasSposti

huoltoNimi työntekijäNo Auto: rekisteriNo -> kilometrit

Omistaja: hetu -> nimi osoite

Lasku: no -> summa eräpäivä

Maksumuistutus: id -> hinta eräpäivä laskuNo

Maksu: id laskuNo -> asiakasSposti

Asiakas: sposti -> nimi osoite

Tarvitaan: nimi huoltoNo -> määrä

Toimenpide: tyyppi -> kesto

Työntekijä: työntekijäNo -> nimi

Poissaolot: työntekijäNo aloitus -> lopetus

Kaikissa funktionaalisissa riippuvuuksissa vasemmalla puolella on pelkästään avain tai avaimet, jotka määrittävät kaikki muut kyseisen relaation attribuutit. Tietokantaa läpikäydessä ei löydy yhtäkään muuta funktionaalista riippuvuutta lukuunottamatta yllä mainittuja riippuvuuksia relaation avaimista muihin attribuutteihin.

Suunnitelmassamme tietokannassa ei ole anomalioita. Monesta yhteen assosiaatioissa olemme tallentaneet monesta puolelle toisen relaation avaimen, ja monesta moneen assosiaatioista olemme tehneet assosiaatiorelaation. Tämän pakollisen toisteisuuden lisäksi tietokannassa ei ole toisteisuutta. Päivitysanomaliaa tietokannassa ei ole, sillä ylimääräistä toisteisuuttakaan ei ole. Poistoanomaliaakaan ei tietokannassa ole. Poistaessa monikoita ylimääräistä tietoa ei katoa tietokannasta, sillä erilliset tiedot ovat tallennettu erillisiin relaatioihin.

Tietokanta on Boyce-Codd-normaali muodossa, sillä jokaisen riippuvuuden sulkeumat sisältävät kaikki kyseisen relaation attribuutit. (Jokainen funktionaalinen riippuvuus on relaation avaimista relaation muihin attribuutteihin, jolloin sulkeuma sisältää kaikki relaation attribuutit.) Emme siis osita tietokantaa enää edelleen, vaan pidämme yllä mainitun tietokannan muodon.

Tietokannan oleellisimpana relaationa on Huolto-relaatio, joka on monen assosiaation toinen osapuoli. Jokainen luokka ja relaatio liittyy jollain tavalla johonkin tiettyyn huoltomonikkoon. Luokkien attribuuteiksi on poimittu tehtävänannosta kaikista oleellimmat, sekä lisätty joitain yksittäisiä attribuutteja halutun toiminnallisuuden saavuttamiseksi. Yksi esimerkki tällaisesta

attribuutista on huoltoluokan attribuutti huoltonumero, jolla yksilöidään huoltomonikot toisistaan. Muut attributit huoltoluokassa kuvaavat huollon kestoa, aloitusajankohtaa ja huollon tyyppiä, eli onko huolto määräaikaishuolto vai korjaus.

Kaaviossa on nähtävissä kaksi silmukkaa, jotka muodostavat yhdessä suurimman ja tärkeimmän osan tietokannan toiminnallisuudesta. Toinen silmukoista keskittyy huollossa tarvittaviin toimenpiteisiin ja niissä käytettäviin laitteisiin, kun taas toinen silmukka keskittyy laskutukseen.

Kaavion muoto ja siitä seuraavat relaatiot mahdollistavat tehtävänannon 3 kohdassa esitettyjä toimintoja. Kaavion vasemmanpuoleisesta, toimenpiteisiin ja laitteisiin keskittyvä silmukasta, voidaan suoraan nähdä mitä toimenpiteitä kuuluu määräaikaishuoltoon tai korjaukseen (Huoltosuunnitelma-relaatio) ja mitä laitetyppejä toimenpiteessä käytetään (laitetyypin ja toimenpiteen välinen assosiaatio), kuitenkin yksilöimättä laitetta, jota varten kaaviossa on oma relaatio "Yksittäinen_laitte".

Huoltoluokan ja työntekijäluokan välisen assosiaation avulla voidaan selvittää, kuka työntekijä on varattu määrättyä huoltokertaa varten. Huollon ja yksittäisen laitteen välissä olevan **Laittevaraus**-relaation avulla voidaan määrittää, mitä yksittäisiä laitteita käytetään kyseisen huollon aikana ja kuinka kauan ne ovat varattuna. Laittevarauksen kesto saadaan toimenpide luokan kesto attribuutin avulla.

Työntekijä-relaatio ja Poissaolot-relaatio välisen assosiaation, ja huoltoluokan ja työntekijäluokan välisen assosiaation, sekä huollon aloitusajankohdan ja huollon keston avulla voidaan selvittää, milloin yksittäinen työntekijä on vapaa huoltoa varten. Samalla tavoin huollon ja yksittäisen laitteen välisen assosiaation avulla voidaan määrittää, milloin yksittäinen laite on käytössä tai vapaana.

Työntekijöiden suunnitellut poissaolot voidaan tallentaa tietokantaan Poissaolot-relaation avulla, johon tallennetaan työntekijä, poissaolon aloitusaika ja lopetusaika. Näin voidaan jo etukäteen suunnitella, ketkä työntekijät ovat saatavilla tietyinä ajankohtana. Myös nopeammalla aikataululla vastaan tulevat poissa-olot voidaan tallentaa tietokantaan tämän saman relaation avulla. Relaation avulla voidaan hakea myös työntekijöiden jo menneitä poissaoloja, jos se olisi tarpeen.

Tietokannassa on omat relaatiot autoja, niiden omistajia ja asiakkaita varten. Huollolla on assosiaatio auton ja asiakkaan kanssa, ja autolla on assosiaatio omistajan kanssa. Asiakas yhdistetään autoon huolto-relaation kautta. Näiden relaatioiden avulla voidaan selvittää auton tuoneen asiakkaan yhteystiedot, omistajan yhteystiedot tai autolle aiemmin tehdyt huollot.

Huollon laskun suuruus saadaan myös laskettua tietokannan avulla. Huollolla on assosiaatio käytettyihin varaosiin sekä lasku-relaatioon. Lasku-relaatiolla on assosiaatio maksumuistutusrelaatioon, minkä avulla maksumuistutuksien määrä ei ole rajattu. Näiden tietojen avulla saadaan laskettua laskun summa. Laskujen lähetys- ja maksutiedot saadaan asiakas-, lasku- ja maksu-relaation avulla. Lasku on maksettu, jos maksu-relaatiolle on sellainen maksu-monikko, jolla on laskua vastaava laskunumero. Tämän avulla voidaan etsiä asiakkaita, jotka eivät ole vielä maksaneet laskujaan, tai asiakkaita, joille on kertynyt tietty määrä maksumuistutuksia.

Tietokannan Huolto-, Toimenpide-, Varattu- ja Poissaolo-relaatiot ovat sidottuna aikaan, joten näiden avulla voidaan hakea tietokannasta erilaisia vapaana oloon ja keston liittyviä asioita. Nämä tiedot riittävät tietokannalle, sillä muut varaustoiminnot hoituvat tehtävänannon mukaan sovellusohjelman kautta.

Harjoitustyö, osa 2

Harjoitustyön ensimmäisen osan muutokset on merkattu ylemmäksi punaisella. Tiedoston alussa olevaa UML-kaaviota ei ole korjattu. Lyhyesti kerrotuna muutokset ovat seuraavat. Huollon ja yksittäisen laitteen välille on lisätty Laitevaraus-relaatio, joka kertoo, milloin tietty laite on varattuna tietylle huollolle. Vastaavasti huollon ja vara-osan välille on lisätty Tarvitaan-relaatio, joka kertoo, mitä vara-osia huolto tarvitsee ja kuinka paljon. Huolto- ja Laitevaraus-

relaatioiden kestot on muutettu lopetusajoiksi. Lisäksi tietokannasta on poistettu tieto omistajan sähköpostista ja korvattu henkilötunnuksella.

Tietokannan taulujen luontikomennnot ja niiden perustelut ovat alla. Jokaisessa taulussa on siihen suunnitellut attribuutit ja attribuuteille sopivat tyypit.

Huollolla on tyyppi, joka tarkistetaan, onko se määräaikaishuolto vai korjaus. Lisäksi sillä on viiteavaimia muihin tauluihin.

```
CREATE TABLE Huolto (    huoltoNo INTEGER PRIMARY
KEY,    aloitusPäivä DATE,    aloitusAika TIME,    lopetusAika
TIME,    tyyppi TEXT CHECK (tyyppi IN ('määräaikaishuolto',
'korjaus')),    rekisteriNo REFERENCES Auto(rekisteriNo),
laskuNo REFERENCES Lasku(nro),    asiakasSposti
REFERENCES Asiakas(sposti),    huoltoNimi REFERENCES
Huoltosuunnitelma(nimi),    työntekijäNo REFERENCES
Työntekijä(nro)
);
```

Autolta tarkastetaan, että sen rekisterinumero on oikeassa muodossa.

```
CREATE TABLE Auto(    rekisteriNo TEXT PRIMARY KEY CHECK
(rekisteriNo LIKE '____-____'),    kilometrit INTEGER
);
```

Omistajasta tarkistetaan, että hänen henkilötunnus on muotoiltu oikein.

```
CREATE TABLE Omistaja ( heTu TEXT PRIMARY KEY  
CHECK (heTu LIKE '_____'), nimi TEXT, osoite TEXT  
);
```

```
CREATE TABLE Omistaa ( rekisteriNo  
REFERENCES Auto(rekisteriNo), heTu  
TEXT REFERENCES Omistaja(heTu),  
PRIMARY KEY (rekisteriNo, heTu)  
);
```

```
CREATE TABLE Lasku ( nro  
INTEGER PRIMARY KEY,  
summa REAL, eräpäivä DATE  
);
```

```
CREATE TABLE Maksumuistutus (  
id INTEGER PRIMARY KEY,  
hinta REAL, eräpäivä DATE,  
laskuNo REFERENCES Lasku(nro)  
);
```

```
CREATE TABLE Maksu ( id INTEGER,  
asiakasSposti REFERENCES Asiakas(sposti),  
laskuNo REFERENCES Lasku(nro), PRIMARY  
KEY (id, laskuNo) Asiakkaasta tarkistetaan,  
voiko hänen sähköpostinsa olla edes luultavasti  
oikea. CREATE TABLE Asiakas ( sposti
```

```
);
```

```
TEXT PRIMARY KEY CHECK (sposti LIKE  
'% @%'), nimi TEXT, osoite TEXT  
);
```

```
CREATE TABLE Varaosa (  
nimi TEXT PRIMARY KEY  
);
```

```
CREATE TABLE Tarvitaan ( määrä  
INTEGER, nimi REFERENCES  
Varaosa(nimi), huoltoNo REFERENCES  
Huolto(huoltoNo),  
PRIMARY KEY (nimi, huoltoNo)  
);
```

```
CREATE TABLE Huoltosuunnitelma (  
nimi TEXT PRIMARY KEY  
);
```

```
CREATE TABLE Kuuluu ( huoltoNimi  
REFERENCES Huoltosuunnitelma(nimi),  
toimenpideTyyppi REFERENCES Toimenpide(tyyppi),  
PRIMARY KEY (huoltoNimi, toimenpideTyyppi)  
);
```

```
CREATE TABLE Toimenpide ( tyyppi  
TEXT PRIMARY KEY, kesto  
INTEGER
```

```
);
```



```
CREATE TABLE Käytetään (    toimenpidetyyppi
REFERENCES Toimenpide(tyyppi),    laitetyyppi
REFERENCES Laitetyyppi(tyyppi),    PRIMARY KEY
(toimenpidetyyppi, laitetyyppi)
);
```

```
CREATE TABLE Laitetyyppi (
tyyppi TEXT PRIMARY KEY
);
```

```
CREATE TABLE Yksittäinen_laite (
laiteNo INTEGER,
    laitetyyppi REFERENCES Laitetyyppi(tyyppi),
    PRIMARY KEY (laiteNo, laitetyyppi)
);
```

```
CREATE TABLE Laitevaraus (    aloitusPäivä
DATE,    aloitusAika TIME,    lopetusAika
TIME,    huoltoNo REFERENCES
Huolto(huoltoNo),    laiteNo INTEGER,
laitetyyppi TEXT,
    FOREIGN KEY (laiteNo, laitetyyppi) REFERENCES Yksittäinen_laite(laiteNo,
laitetyyppi),
    PRIMARY KEY (aloitusPäivä, aloitusAika, huoltoNo, laiteNo, laitetyyppi)
);
```

```
CREATE TABLE Työntekijä ( nro
INTEGER PRIMARY KEY,
nimi TEXT

);
```

Poissaolosta tarkistetaan, onko aloituspäivä ennen lopetuspäivää.

```
CREATE TABLE Poissaolot (    aloitus DATE,    työntekijäNo  
REFERENCES Työntekijä(nro),    lopetus DATE,  
    PRIMARY KEY (aloitus, työntekijäNo),  
    CHECK (aloitus < lopetus)  
);
```

Tietokannassamme ei ole muita hakemistoja kuin SQLiten valmiiksi luomat hakemistot. SQLite luo tietokantaan hakemiston jokaisen taulun avaimille. Tämän lisäksi tähän tietokantaan ei ole kannattavaa luoda muita hakemistoja, sillä ne vain hidastaisivat tietokannan toimintaa lisäämällä levyhakujen määrää. Muut attribuutit kuin taulujen avaimet esiintyvät tietokannassa useita kertoja eikä tauluja ole klusteroitu mikään attribuutin mukaan, minkä takia levyhakuja tulisi aina useita. Esimerkiksi hakemistoa huollon päivämäärän mukaan ei kannata luoda, sillä samana päivänä on useita huoltoja, mikä tarkoittaa useita levyhakuja.

Olemme luoneet yhden valmiin näkymän asiakkaille, joilla on maksamattomia laskuja, sillä vastaavaa kyselyä tarvitaan toistuvasti tietokantaa käytettäessä.

```
CREATE VIEW Maksamattomat AS  
    SELECT Asiakas.nimi  
    FROM Lasku, Huolto, Asiakas  
    WHERE Huolto.laskuNo=Lasku.numero AND Asiakas.sposti=Huolto.asiakasSposti  
    AND Lasku.numero NOT IN (  
        SELECT laskuNo  
        FROM Maksu  
    );
```

SQL kyselyt:

Etsii ne työntekijät, jotka ovat vapaana 07.05.2022 klo 13-16:

```

SELECT nimi
FROM Työntekijä
WHERE nro NOT IN (
    SELECT Huolto.työntekijäNo
    FROM Huolto
    WHERE ((Huolto.aloitusAika BETWEEN '13:00:00' AND '15:59:59') OR
(Huolto.lopetusAika BETWEEN '13:00:01' AND '15:59:59')) AND
Huolto.aloitusPäivä='2022-05-07'
UNION
SELECT T.nro
FROM Työntekijä AS T, Poissaolot AS P
WHERE P.työntekijäNo = T.nro AND '2022-05-07' BETWEEN P.aloitus AND P.lopetus
);

```

Etsii asiakkaan yhteystiedot auton rekisterinumeron ('ZIN-602') perusteella:

```

SELECT nimi, posti, osoite
FROM Huolto AS H, Auto AS A, Asiakas
WHERE posti = asiakasPosti AND H.rekisteriNo = A.rekisteriNo AND A.rekisteriNo =
'ZIN-602';

```

Listaa asiakkaat ja heidän laskujensa summan:

```

SELECT Asiakas.nimi, SUM(Lasku.summa)
FROM Lasku, Huolto, Asiakas
WHERE Asiakas.posti=Huolto.asiakasPosti AND Huolto.laskuNo=Lasku.nro
GROUP BY Asiakas.posti;

```

Listaa asiakkaat ja heidän maksumuistutuksiensa summan:

```

SELECT Asiakas.nimi, SUM(Maksumuistutus.hinta)
FROM Maksumuistutus, Asiakas, Lasku, Huolto
WHERE Maksumuistutus.laskuNo=Lasku.nro AND Asiakas.posti=Huolto.asiakasPosti
AND Huolto.laskuNo=Lasku.nro
GROUP BY Asiakas.posti;

```

Etsii kaikki huollot, mitä autolle ('ZIN-602') on tehty ja kertoo ne sekä niiden päiväykset:

```
SELECT huoltoNimi, aloitusPäivä
FROM Auto AS A, Huolto AS H
WHERE A.rekisteriNo = H.rekisteriNo AND A.rekisteriNo = 'ZIN-602';
```

Etsii mitä toimenpiteitä kuuluu yhteen huoltosuunnitelmaan ('Ratin vaihto'):

```
SELECT tyyppi
FROM Toimenpide, Huoltosuunnitelma, Kuuluu
WHERE tyyppi = toimenpideTyyppi AND nimi = huoltoNimi AND nimi = 'Ratin vaihto';
```

Listaa painepesurit, jotka ovat vapaana 01.05.2022 klo 15-17:

```
SELECT DISTINCT *
FROM Yksittäinen_laite
WHERE laiteTyyppi='Painepesuri' AND laiteNo NOT IN (
  SELECT laiteNo
  FROM Laitevaraus
  WHERE laitetyyppi='Painepesuri' AND aloitusPäivä = '2022-05-01' AND ((aloitusAika
  BETWEEN '15:00' AND '17:00') OR (lopetusAika BETWEEN '15:00' AND '17:00'))
);
```

Listaa kaikki poissaolot, ja järjestää ne aloitusajan perusteella:

```
SELECT aloitus, lopetus, nimi
FROM Työntekijä, Poissaolot
WHERE nro = työntekijäNo
ORDER BY aloitus;
```

Listaa huollot, joiden aloituspäivä on 01.05.2022 jälkeen ja listaa ne aloituspäivän perusteella:

```
SELECT huoltoNo, aloitusPäivä, rekisteriNo, asiakasSposti, huoltoNimi
FROM Huolto
WHERE aloitusPäivä > '2022-05-01'
ORDER BY aloitusPäivä;
```

Listaa työntekijät ja heidän huoltonsa järjestyksessä aloituspäivän mukaan:

```
SELECT T.nimi, T.nro, H.aloitusPäivä, H.tyyppi
FROM Työntekijä AS T, Huolto AS H
WHERE T.nro=H.työntekijäNo
ORDER BY H.aloitusPäivä;
```

Etsii ne asiakkaat, joille on lähetetty maksumuistutuksia, mutta eivät silti ole maksaneet laskua:

```
SELECT DISTINCT Asiakas.nimi, Lasku.nro
FROM Maksumuistutus, Lasku, Huolto, Asiakas
WHERE Asiakas.sposti=Huolto.asiakasSposti AND Maksumuistutus.laskuNo=Lasku.nro
AND Lasku.nro=Huolto.laskuNo AND Lasku.nro NOT IN (
    SELECT laskuNo
    FROM Maksu
);
```

Etsii mitä osia ja kuinka paljon tarvitaan tietyn auton tiettyä huoltosuunnitelmaa varten:

```
SELECT DISTINCT A.rekisteriNo, HS.nimi, V.nimi, T.määrä
FROM Huolto AS H, Tarvitaan AS T, Varaosa AS V, Huoltosuunnitelma AS HS, Auto AS A
WHERE T.nimi = V.nimi AND T.huoltoNo = H.huoltoNo AND HS.nimi = huoltoNimi AND
A.rekisteriNo = H.rekisteriNo;
```

Maksamattomien laskujen summa kaikilta asiakkailta:

```
SELECT SUM(summa)
FROM Lasku
WHERE nro NOT IN (
    SELECT laskuNo
    FROM Maksu
);
```

Niiden asiakkaiden yhteystiedot, joilla laskun eräpäivä on kesäkuussa:

```
SELECT DISTINCT A.nimi, sposti, osoite
FROM Asiakas AS A, Lasku AS L, Huolto AS H
WHERE sposti = asiakasSposti AND nro = laskuNo AND L.eräpäivä BETWEEN '2022-06-01' AND '2022-06-31';
```

Listaa omistajat ja heidän omistamiensa autojen määrän:

```
SELECT Omistaja.nimi, Omistaja.heTu, COUNT(Omistaa.rekisteriNo)
FROM Omistaa, Omistaja
WHERE Omistaa.heTu=Omistaja.heTu
GROUP BY Omistaa.heTu;
```

