

CS-A1150 – Tietokannat Projektityö

Shahram Barai 729983
Viktoria Malikova 730286

Palautettu 6.04.2022

Sisällysluettelo

| | |
|---|----|
| 1 ESIVALMISTELU | 2 |
| 2 UML-KAAVIO | 4 |
| 3 RELAATIOT | 5 |
| 4 SELOSTUS..... | 7 |
| 5 VASTAUKSET KYSYMYKSIIN | 9 |
| 6 TIETOKANNAN MUUNTAMINEN SQL-MUOTOON | 10 |
| 7 TOIMINTOJEN TOTEUTUS..... | 14 |
| 8 KÄYTTÖTAPAUKSIA..... | 19 |
| KORJAUKSIA EDELLISESSÄ PALAUTUKSESSA | 21 |

1 Esivalmistelu

Tähän osioon on kopioitu tehtävänannosta tietokannalle olennaisia tietoja, ja käsitelty muotoon mikä helpottaa UML kaavion suunnittelua.

- **Autokorjaamo** tekee sekä autojen **määräaikaishuoltoja** että **yllättäen ilmaantuneiden vikojen korjauksia**. Jatkossa sana **huolto** tarkoittaa sekä **määräaikaishuoltoa** että **korjausta**.
- **Määräaikaishuolto** sisältää joukon **valmistajan** määäämiä **toimenpiteitä** (esim. jakohihnan kunnon tarkistus tai polttoainesuodattimen vaihto), jotka riippuvat siitä, mistä **määräaikaishuollosta** on kysymys (paljonko **autolla** on **ajettu**). Harjoitustyön yksinkertaistamiseksi oletetaan, että kaikki **määräaikaishuoltoon** kuuluvat **toimenpiteet** ja niiden **kesto** ovat etukäteen tiedossa.
- Myös **korjaus** koostuu **toimenpiteistä**. Harjoitustyön yksinkertaistamiseksi oletetaan, että myös kaikki tiettyyn **korjaukseen** kuuluvat **toimenpiteet** ja niiden **kesto** ovat etukäteen tiedossa. (Tämä ei ole realistinen oletus, mutta se tekee harjoitustyön selvästi yksinkertaisemmaksi.)
- **Huollon** suorittaa sama **työntekijä** yhteen menoon alusta loppuun.
- Osa **toimenpiteistä** vaatii **laitteita**, joita **korjaamolla** on käytössä vain **rajoitettu määrä** (esimerkiksi nosturi). Samanlaista **laitetta** voi kuitenkin olla **useampi kappale** (esimerkiksi kolme samaa mallia olevaa nosturia).
- **Korjaamo** huoltaa vain **autoja**. **Autot** tunnistetaan **rekisterinumeron** avulla. **Korjaamolla** on tiedossa auton huoltoon tuonut **asiakas** ja sen **omistaja(t)**. **Omistajia** voi olla useita (esimerkiksi pariskunta voi omistaa auton puoliksi). Auton huoltoon tuonut **asiakas** ei ole välttämättä sama kuin auton **omistaja**.
- **Asiakas** voi varata **huoltoajan** etukäteen. Tällöin **aika** annetaan niin, että korjaamolla on sovittuun **aikaan** käytössä huollossa tarvittavat **laitteet ja työntekijä** huollon tekemiseen. **Työntekijä** varataan koko **huollon ajaksi**, mutta **laite** vain sen **toimenpiteen ajaksi**, jossa laitetta oikeasti tarvitaan.
- **Huollossa** voidaan tarvita **varaosia**.
- **Huollosta** kirjoitetaan **lasku**, jonka suuruuteen vaikuttaa se, kuinka monta **tuntia työntekijä** on tehnyt töitä **huollon** eteen ja mitä **varaosia** huollossa on tarvittu.
- Jos **asiakas** ei maksa **laskua** ajoissa, asiakkaalle lähetetään **maksumuistutus**. **Maksumuistutuksia** voidaan lähettää tarvittaessa **useita**. Jokaisesta lähetetystä muistutuksesta voi tulla lisämaksu alkuperäiseen laskuun.

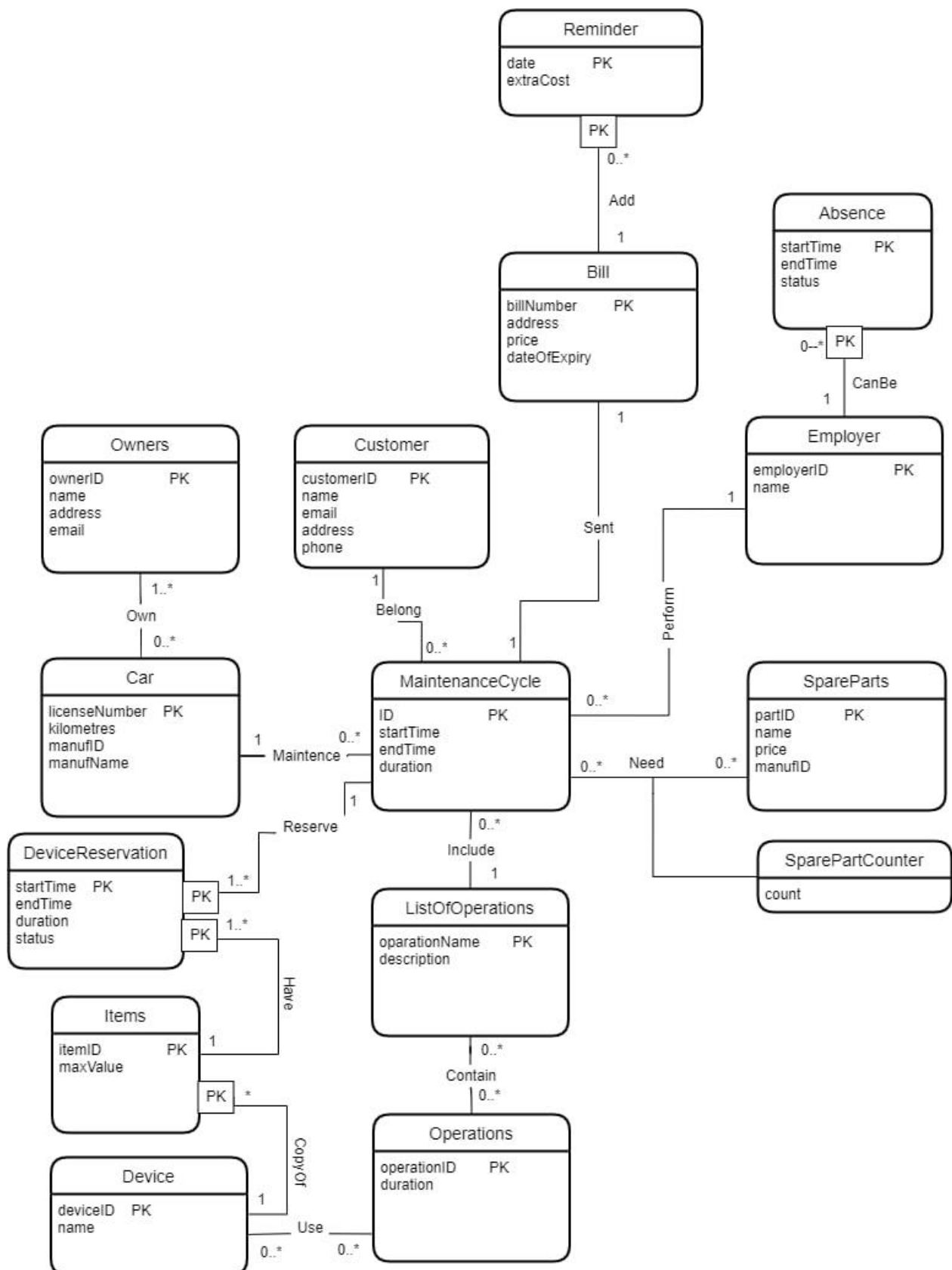
Tietokannassa pitää olla mahdollisuus esimerkiksi seuraaviin toimenpiteisiin (lisäksi järjestelmä saa tukea muitakin toimenpiteitä). Näitä toimenpiteitä ei vielä varsinaisesti esitetä UML-kaaviossa ja relaatiomallissa, mutta ne pitää suunnitella siten, että halutut toiminpiteet ovat mahdollisia:

- Tallentaa tieto siitä, mitä **toimenpiteitä** kuuluu **määrättyyn määräaikaishuoltoon** tai **korjaukseen** ja mitä **laitteita** eri **toimenpiteissä** tarvitaan. Laitteiden osalta tieto koskee **laitetyyppejä**. Jos esimerkiksi korjaamolla on kolme samanlaista nosturia, tietokantaan pitää olla mahdollisuus merkitä, että toimenpiteessä tarvitaan tällainen nosturi yksilöimättä sitä, mitä samanlaisista nostureista toimenpiteessä pitää käyttää.

- Tallentaa tieto siitä, kuka **työntekijä** on varattu määrättyä **huoltokertaa** varten ja mitä **laitteita** on varattu siinä tarvittaviin **toimenpiteisiin** miksikin **aikaa**. Tässä yksilöidään se, mikä samanlaisista **laitteista** varataan. **Huoltokerrasta** tallennetaan sen suunniteltu **alkamis- ja loppumisaika**.
- Etsiä **ajankohta**, jolloin joku **työntekijä** on vapaana tarpeeksi pitkän ajan yhteen menoon haluttua **huoltoa** varten ja varata ko. **työntekijä** tätä **huoltoa** varten.
- Etsiä **ajankohta**, jolloin haluttu **laite** on vapaana johonkin huoltoon kuuluvaa toimenpidettä varten ja varata ko. **laite** tätä **huoltoa** varten. (Laite varataan vain toimenpiteen, ei koko huollon ajaksi.) Tässä tarkoitetaan vain laitteita, joita korjaamolla on käytössä rajoitettu määrä. Yksinkertaisia perustyökaluja oletetaan olevan käytössä niin paljon, että niitä ei tarvitse erikseen varata.
- Etsiä ne **työntekijät**, jotka ovat vapaana halutulla välillä.
- Etsiä **aikoja**, jolloin jokin haluttu **laite** on vapaana. (Mikä tahansa samanlaisista halutuista laitteista.)
- Tallentaa tieto työntekijöiden ennakolta suunnitelluista **poissaoloista**, esimerkiksi lomista. Työntekijää ei voi varata huoltokertaa varten silloin, kun hän on poissa töistä.
- Tallentaa tietoja korjaamoon tuoduista **autoista**, niiden **omistajista** ja auton korjaamoon tuoneesta **asiakkaasta**.
- Selvittää auton huoltoon tuoneen **asiakkaan yhteystiedot**.
- Selvittää, mitä **huoltoja** tietylle **autolle** on tehty tässä samassa korjaamossa aikaisemmin.
- Tallentaa tieto **huoltokerran laskuun** vaikuttavista asioista eli käytetyistä **varaosista** ja niiden **määristä** sekä siitä, kuka **työntekijä** on tehnyt **huollon** ja kuinka kauan hän on käyttänyt siihen **aikaa**.
- Tallentaa ja selvittää **laskujen lähetys- ja maksutiedot**. Tallentaa ja selvittää laskuun liittyvien mahdollisten **maksumuistutusten** lähetys- ja maksutiedot. Järjestelmä ei saa rajoittaa sitä, kuinka monta maksumuistutusta yhdestä laskusta voidaan lähettää.
- Etsiä **asiakkaat**, joilla on **maksamattomia maksuja**.
- Etsiä **asiakkaat**, joille on lähetetty määrätty määrä **maksumuistutuksia**, mutta jotka eivät ole silti vielä maksaneet laskua.

2 UML-kaavio

Tässä luvussa esitetään UML-kaavio.



3 Relaatiot

Luokka-relaatiot

1. MaintenanceCycle(ID, startTime, endTime, duration, operationName, licenseNumber, employerID, customerID, billNumber)
 - ID → startTime endTime duration operationName licenseNumber employerID customerID
 - {ID}⁺ = {ID, startTime, endTime, duration, operationName, licenseNumber, employerID, customerID}
2. ListOfOperations(operationName, description)
 - operationName → description
 - {operationName}⁺ = {operationName, description}
3. Operations(operationID, duration)
 - operationID → duration
 - {operationID}⁺ = {operationID, duration}
4. Device(deviceID, name)
 - deviceID → name
 - {deviceID}⁺ = {deviceID, name}
5. Items(deviceID, itemID, maxValve)
 - deviceID itemID → maxValve
 - {deviceID, itemID}⁺ = {deviceID, itemID, maxValve}
6. DeviceReservation(ID, itemID, deviceID, startTime, endTime, duration, status)
 - ID itemID deviceID startTime → endTime duration status
 - {ID, itemID, deviceID, startTime}⁺ = {ID, itemID, deviceID, startTime, endTime, duration, status}
7. Customer(customerID, name, email, address, phone)
 - customerID → name email address phone
 - {customerID}⁺ = {customerID, name, email, address, phone}
8. Car(licenseNumber, kilometres, manufID, manufName)
 - licenseNumber → kilometers manufID manufName
 - {licenseNumber}⁺ = {licenseNumber, kilometres, manufID, manufName}
9. Owners(ownerID, name, address, email)
 - ownerID → name address email
 - {ownerID}⁺ = {ownerID, name, address, email}
10. Employer(emploerID, name)
 - emploerID → name
 - {emploerID}⁺ = {emploerID, name}
11. Absence(employerID, startTime, endTime, status)
 - employerID startTime → endTime status
 - {employerID, startTime}⁺ = {employerID, startTime, endTime, status}
12. Bill(billNumber, address, price, dateOfExpiry)
 - billNumber → address price dateOfExpiry
 - {billNumber}⁺ = {billNumber, address, price, dateOfExpiry}

13. Reminder(billNumber, date, extraCost)
 - billNumber date → date extraCost
 - {billNumber, date}⁺ = {billNumber, date, extraCost}
14. SparePart(partID, name, price, manufID)
 - partID → name price manufID
 - {partID}⁺ = {partID, name, price, manufID}

Assosiaatio-relaatiot

Assosiaatio-relaation on ainoastaan yhdessä kohdassa UML-kaaviota:

1. SparePartCounter(ID, partID, count)
 - ID partID → count
 - {ID, partID}⁺ = {ID, partID, count}
2. Owns(licenseNumber, ownerID)
 - {licenseNumber, ownerID}⁺ = {licenseNumber, ownerID}
3. Contain(operationID, operationName)
 - {operationID, operationName}⁺ = {operationID, operationName}
4. Use(operationID, deviceID)
 - {operationID, deviceID}⁺ = {operationID, deviceID}

4 Selostus

Tietokannan kuvaus

Autokorjaamo tekee sekä autojen määräaikaishuoltoja että yllättäen ilmaantuneiden vikojen korjauksia eli huolto-operaatiot (MaintenanceCycle), joilla jokaisella on niille uniikki tunniste (ID) sekä sen suunniteltu alkamis- ja loppumisaika (startTime, endTime) ja kesto (duration). Autokorjaamolla on olemassa huolto-operaatioiden lista (ListOfOperations), jossa on toimipiteiden nimet (operationName) sekä kuvaus (description). Huolto-operaatioiden lista sisältää useita kappaleita operaatiota (Operation), joilla on uniikki tunniste (operationID) sekä kesto (duration).

Osa toimenpiteistä vaatii laitteita (Device). Jokaisella laitteilla on oma uniikki tunniste (deviceId) sekä nimi (name). Samanlaista laitetta autokorjaamolla on kuitenkin useampi kappale (Items) (esimerkiksi kolme samaa mallia olevaa nosturia), joita tunnustetaan uniikki tunnisteella (itemID). Laitteita korjaamolla on käytössä vain rajoitettu määrä (maxValue) (esimerkiksi nosturi).

Huollossa voidaan tarvita varaosia (SparePart). Varaosilla on uniikki tunniste (partID), nimi (name), hinta (price) sekä valmistajan uniikki tunniste (manufID).

Korjaamo huoltaa vain autoja (Car). Autot tunnustetaan rekisterinumeron avulla (licenseNumber). Autojen määräaikaishuolto sisältää joukon valmistajan (manufID) määräämiä toimenpiteitä (esim. jakohihnan kunnon tarkistus tai polttoainesuodattimen vaihto), jotka riippuvat siitä, mistä määräaikaishuollosta on kysymys (paljonko autolla on ajettu). Sen vuoksi autokorjaamon täytyy varmistaa ajetun kilometrimäärän (kilometres). Lisäksi jokaisella auton valmistajalla on nimi (manufName).

Korjaamolla on tiedossa auton huoltoon tuonut asiakas (Customer) ja sen omistaja(t) (Owners). Omistajia voi olla useita (esimerkiksi pariskunta voi omistaa auton puoliksi). Auton huoltoon tuonut asiakas ei ole välttämättä sama kuin auton omistaja. Asiakkailta on uniikki tunniste (customerID), nimi (name), sähköpostiosoite (email), kotiosoite (address) sekä puhelinnumero (phone). Omistajalla on uniikki tunniste (ownerID), nimi (name), kotiosoite (address) ja sähköpostiosoite (email).

Asiakas voi varata huoltoajan etukäteen. Tällöin aika annetaan niin, että korjaamolla on sovittuun aikaan käytössä huollossa tarvittavat laitteet ja työntekijä huollon tekemiseen. Laitte varataan (DeviceReservation) vain sen toimenpiteen ajaksi, jossa laitetta oikeasti tarvitaan, mutta työntekijä varataan koko huollon ajaksi. Laittevaraukseen kuuluu sen käytön alkamis- (startTime) ja lopetusaika (endTime) sekä laitteen saatavuus (status), jotta tiedetään mitä laitteita on varattu tarvittaviin toimenpiteisiin miksikin aikaa.

Huollon suorittaa sama työntekijä (Employer) yhteen menoon alusta loppuun. Työntekijä on varattu määrättyä huoltokertaa varten. Hänellä on uniikki tunniste (employeeID) sekä nimi (name). Työntekijää ei voi varata huoltokertaa varten silloin, kun hän on poissa töistä eli poissaoleva (Absence). Poissaolevalla työntekijällä on uniikki tunniste (employeeID), alkamis- (startTime) ja lopetusaika (endTime) sekä status (status).

Huollosta kirjoitetaan lasku (Bill), jonka suuruuteen vaikuttaa se, kuinka monta tuntia työntekijä on tehnyt töitä huollon eteen ja mitä varaosia huollossa on tarvittu. Laskulla on laskunumero (billNumber), asiakkaan osoite (address), autohuollon hinta (price) sekä eräpäivä (dateOfExpiry).

Jos asiakas ei maksa laskua ajoissa, asiakkaalle lähetetään maksumuistutus (Reminder). Maksumuistutuksia voidaan lähettää tarvittaessa useita. Maksumuistutuksilla on numero (billNumber) ja päivämäärä (date). Jokaisesta lähetetystä muistutuksesta voi tulla lisämaksu alkuperäiseen laskuun (extraCost).

5 Vastaukset kysymyksiin

Funktionaaliset riippuvuudet, Boyce-Codd normaalimuotoisuus sekä anomaliat

Tietokannan relaatioiden yhteydessä on esitetty funktionaaliset riippuvuudet sekä niiden sulkeumat (mikäli se on mahdollista). Jokainen sulkeuma sisältää kyseisen relaation kaikki attribuutit, joten tietokanta on Boyce-Codd normaalimuodossa. Täten relaatioiden osittamista ei tarvitse tehdä.

- Tiedon toisteisuus anomalioita (redundancy anomalies) ei ole tässä tietokannassa, sillä ainoat attribuutit, jotka toistuvat eri relaatioissa ovat itse avaimet.
- Päivitysanomalioita (update anomalies) ei ole myöskään tässä tietokannassa, sillä ainoat attribuutit, jotka toistuvat eri relaatioissa ovat itse avaimet.
- Poistoanomaliat (deletion anomalies) ei ole myöskään tässä tietokannassa, sillä ainoat attribuutit, jotka toistuvat eri relaatioissa ovat itse avaimet.

Tästä voidaan todeta, että tietokannan eheys on korkealaatuista.

6 Tietokannan muuntaminen SQL-muotoon

Taulujen teko

Käytimme taulukoiden teossa ainoastaan SQLite hyväksymiä tyyppejä kuten:

- INTEGER (kokonaisluku)
- REAL (desimaaliluku)
- TEXT (merkkijono)
- BLOB (sellaisenaan tallennettava binääridata)

```
CREATE TABLE Bill(  
    billNumber TEXT,  
    address TEXT,  
    price REAL CHECK (price > 0.0),  
    dateOfExpiry TEXT,  
    PRIMARY KEY (billNumber)  
);
```

Attribuutille price on valittu tyyppi REAL CHECK, sillä hinta ei voi olla negatiivinen.

```
CREATE TABLE Reminder(  
    billNumber TEXT REFERENCES Bill(billNumber),  
    date TEXT,  
    extraCost REAL NOT NULL,  
    PRIMARY KEY (billNumber, date)  
);
```

Attribuutille extraCost on valittu tyyppi REAL NOT NULL, sillä lisämaksu pitäisi olla joku desimaaliluku siis attribuutti ei voi olla NULL.

```
CREATE TABLE Employer(  
    employerID TEXT,  
    name TEXT,  
    PRIMARY KEY (employerID)  
);
```

```
CREATE TABLE Absence(  
    employerID TEXT REFERENCES Employer(employerID),  
    startTime TEXT NOT NULL,  
    endTime TEXT NOT NULL,  
    status TEXT CHECK (status IN ('available', 'not available', 'sick leave', 'vacation')),  
    PRIMARY KEY (employerID, startTime)  
);
```

Attribuutti startTime ja endTime ei voi olla NULL, sen vuoksi on valittu tyyppi TEXT NOT NULL.
Attribuutti status tarkistaa onko työntekijä saattavilla tai ei tai hän on sairaslomalla tai lomalla.

```
CREATE TABLE SpareParts(  
    partID TEXT,  
    name TEXT NOT NULL,  
    price REAL CHECK (price > 0.0),  
    manufID TEXT,  
    PRIMARY KEY (partID)  
);
```

```
CREATE TABLE ListOfOperations(  
    operationName TEXT,  
    description TEXT,  
    PRIMARY KEY (operationName)  
);
```

```
CREATE TABLE Operations(  
    operationID TEXT,  
    duration REAL CHECK (duration > 0.0),  
    PRIMARY KEY (operationID)  
);
```

```
CREATE TABLE Customer(  
    customerID TEXT,  
    name TEXT NOT NULL,  
    email TEXT,  
    address TEXT NOT NULL,  
    phone INTEGER NOT NULL,  
    PRIMARY KEY (customerID)  
);
```

```
CREATE TABLE Owners(  
    ownerID TEXT,  
    name TEXT NOT NULL,  
    address TEXT NOT NULL,  
    email TEXT,  
    PRIMARY KEY (ownerID)  
);
```

```
CREATE TABLE Car(  
    licenseNumber TEXT,  
    kilometres INTEGER CHECK (kilometres > 0),  
    manufID TEXT,  
    manufName TEXT,  
    PRIMARY KEY (licenseNumber)  
);
```

Attribuutille kilometers täytyy olla suurempi kuin nolla, jolloin valittu tyyppi on INTEGER CHECK.

```
CREATE TABLE Device(  
    deviceID TEXT,  
    name TEXT,  
    PRIMARY KEY (deviceID)  
);
```

```
CREATE TABLE Items(  
    deviceID TEXT REFERENCES Device(deviceID),  
    itemID TEXT,  
    maxValve INTEGER CHECK (maxValue >= 0),  
    PRIMARY KEY (deviceID, itemID)  
);
```

Laiteita korjaamolla on käytössä vain rajoitettu määrä, jolloin attribuutille maxValve on valittu INTEGER CHECK -tyyppi.

```
CREATE TABLE MaintenanceCycle(  
    ID TEXT,  
    startTime TEXT NOT NULL,  
    endTime TEXT NOT NULL,  
    duration REAL CHECK (duration > 0.0),  
    operationName TEXT REFERENCES ListOfOperations(operationName),  
    licenseNumber TEXT REFERENCES Car(licenseNumber),  
    employerID TEXT REFERENCES Employer(employerID),  
    customerID TEXT REFERENCES Customer(customerID),  
    billNumber TEXT REFERENCES Bill(billNumber),  
    PRIMARY KEY (ID)  
);
```

Kesto voi olla vain ja ainoastaan positiivinen desimaaliluku, sen vuoksi attribuutille duration on valittu tyyppi REAL CHECK.

```
CREATE TABLE DeviceReservation(  
    ID TEXT,  
    itemID TEXT,  
    deviceID TEXT,  
    startTime TEXT,  
    endTime TEXT,  
    duration REAL CHECK (duration > 0.0),  
    status TEXT CHECK (status IN ('available', 'not available')),  
    FOREIGN KEY (ID) REFERENCES MaintenanceCycle(ID),  
    FOREIGN KEY (itemID, deviceID) REFERENCES Items(itemID, deviceID),  
    PRIMARY KEY (ID, itemID, deviceID, startTime)  
);
```

```

CREATE TABLE SparePartCounter(
  ID TEXT REFERENCES MaintenanceCycle(ID),
  partID TEXT REFERENCES SpareParts(partID),
  count INTEGER CHECK (count >= 0),
  PRIMARY KEY (ID, partID)
);

CREATE TABLE Owns(
  licenseNumber TEXT REFERENCES Car(licenseNumber),
  ownerID TEXT REFERENCES Owners(ownerID),
  PRIMARY KEY (licenseNumber, ownerID)
);

CREATE TABLE Contain(
  operationID TEXT REFERENCES Operations(operationID),
  operationName TEXT REFERENCES ListOfOperations(operationName),
  PRIMARY KEY (operationID, operationName)
);

CREATE TABLE Use(
  operationID TEXT REFERENCES Operations(operationID),
  deviceID TEXT REFERENCES Device(deviceID),
  PRIMARY KEY (operationID, deviceID)
);

```

7 Toimintojen toteutus

```
-- Bill(billNumber, address, price, dateOfExpiry)
INSERT INTO Bill VALUES ('48849', 'Otaranta 8', 50.0, '2022-06-01 11:00');
INSERT INTO Bill VALUES ('35990', 'Otaranta 8', 50.0, '2023-06-01 11:00');
INSERT INTO Bill VALUES ('48953', 'Otaranta 8', 50.0, '2024-06-01 11:00');
INSERT INTO Bill VALUES ('83830', 'Servinkuja 3', 250.0, '2022-06-01 14:00');
INSERT INTO Bill VALUES ('13048', 'Servinkuja 3', 250.0, '2023-06-01 14:00');
INSERT INTO Bill VALUES ('83802', 'Servinkuja 3', 250.0, '2024-06-01 14:00');
INSERT INTO Bill VALUES ('49870', 'Aaltokatu 4', 250.0, '2022-06-02 14:00');
INSERT INTO Bill VALUES ('82904', 'Aaltokatu 4', 250.0, '2022-06-03 14:00');
INSERT INTO Bill VALUES ('20394', 'Aaltokatu 4', 250.0, '2022-06-04 14:00');

-- Reminder(billNumber, date, extraCost)
INSERT INTO Reminder VALUES('48849', '2022-10-12', 5.00);
INSERT INTO Reminder VALUES('83830', '2022-09-17', 5.00);
INSERT INTO Reminder VALUES('49870', '2022-03-06', 5.00);

-- Employer(employerID, name)
INSERT INTO Employer VALUES('573hdik', 'Jukka Pekkanen');
INSERT INTO Employer VALUES('487jrfl', 'Teemu Teekkari');
INSERT INTO Employer VALUES('836hrtl', 'Tiia Viljanen');
INSERT INTO Employer VALUES('650uthg', 'Ilma Aalto');

-- Statuts: 'available', 'not available', 'sick leave', 'vacation'
-- Absence(employerID, startTime, endTime, status)
INSERT INTO Absence VALUES('573hdik', '2022-01-01 8:00', '2022-07-31 16:00', 'available');
INSERT INTO Absence VALUES('487jrfl', '2022-06-01 00:00', '2022-06-30 00:00', 'not available');
INSERT INTO Absence VALUES('836hrtl', '2022-05-01 00:00', '2022-05-14 00:00', 'sick leave');
INSERT INTO Absence VALUES('650uthg', '2022-08-01 00:00', '2022-08-21 00:00', 'vacation');

-- SpareParts(partID, name, price, manufID)
INSERT INTO SpareParts VALUES('58250', 'tiiviste', 19.90, '1-hk0');
INSERT INTO SpareParts VALUES('84375', 'jarrulevy', 47.90, '4-7n6');
INSERT INTO SpareParts VALUES('24368', 'suodatin', 24.90, '2-jt8');
INSERT INTO SpareParts VALUES('97741', 'akku', 139.00, '9-4ns');
INSERT INTO SpareParts VALUES('84692', 'öljy', 19.90, '3-jfn');
```

```

-- ListOfOperations(operationName, description)
INSERT INTO ListOfOperations VALUES('jarrujen tarkastus', 'tarkastetaan jarruja');
INSERT INTO ListOfOperations VALUES('öljyn vaihto', 'vaihdetaan öljy');
INSERT INTO ListOfOperations VALUES('ilmansuodattimen vaihto', 'vaihdetaan ilmansuodatin');
INSERT INTO ListOfOperations VALUES('rikkinaisien osien vaihto', 'osat vaihdetaan');

-- Operations(operationID, duration)
INSERT INTO Operations VALUES('5hieu59', 1.50);
INSERT INTO Operations VALUES('60riker', 0.50);
INSERT INTO Operations VALUES('49jrfhe', 0.70);
INSERT INTO Operations VALUES('tiksje8', 1.00);

-- Customer(customerID, name, email, address, phone)
INSERT INTO Customer VALUES('112233', 'Teemu Teekkari', 'tteekkari@gmail.com', 'Servinkuja 3',
453828);
INSERT INTO Customer VALUES('554422', 'Riina Raksalainen', 'riinar@yahoo.com', 'Otaranta 8',
765439);
INSERT INTO Customer VALUES('37856', 'Antti Virta', 'antti@hotmail.com', 'Aaltokatu 4', 958694);
INSERT INTO Customer VALUES('77233', 'Nina Teekkari', 'nite@hotmail.com', 'Servinkuja 3',
846923);
INSERT INTO Customer VALUES('11112A', 'Oskari Teekkari', 'oskari@gmail.com', 'Servinkuja 3',
887328);

-- Owners(ownerID, name, address, email)
INSERT INTO Owners VALUES('74382A', 'Teemu Teekkari', 'Servinkuja 3', 'tteekkari@gmail.com');
INSERT INTO Owners VALUES('84929B', 'Riina Raksalainen', 'Otaranta 8', 'riinar@yahoo.com');
INSERT INTO Owners VALUES('38495C', 'Tiina Saksalainen', 'Aaltokatu 9', 'tiinas@yahoo.com');
INSERT INTO Owners VALUES('76463D', 'Pekka Pekkanen', 'Metsäkatu 4', 'pekkap@hotmail.com');
INSERT INTO Owners VALUES('11111A', 'Oskari Teekkari', 'Servinkuja 3', 'oskari@gmail.com');
INSERT INTO Owners VALUES('11111B', 'Tia Raksalainen', 'Otaranta 8', 'tia@yahoo.com');
INSERT INTO Owners VALUES('11111C', 'Viivi Saksalainen', 'Aaltokatu 9', 'viivi@yahoo.com');
INSERT INTO Owners VALUES('11111D', 'Martin Pekkanen', 'Metsäkatu 4', 'martin@hotmail.com');

-- Car(licenseNumber, kilometres, manufID, manufName)
INSERT INTO Car VALUES('110-aaa', 183000, '850jh', 'bmw');
INSERT INTO Car VALUES('111-aaa', 92112, '850jh', 'bmw');
INSERT INTO Car VALUES('112-aaa', 222412, '850jh', 'bmw');
INSERT INTO Car VALUES('113-aaa', 313643, '850jh', 'bmw');
INSERT INTO Car VALUES('114-aaa', 7658, '850jh', 'bmw');

INSERT INTO Car VALUES('220-aaa', 158000, '574rb', 'volvo');
INSERT INTO Car VALUES('221-aaa', 5643, '574rb', 'volvo');
INSERT INTO Car VALUES('222-aaa', 287432, '574rb', 'volvo');

```



```
INSERT INTO Car VALUES('223-aaa', 187432, '574rb', 'volvo');
INSERT INTO Car VALUES('224-aaa', 387432, '574rb', 'volvo');
```

```
INSERT INTO Car VALUES('330-aaa', 11000, '375sv', 'kia');
INSERT INTO Car VALUES('331-aaa', 143010, '375sv', 'kia');
INSERT INTO Car VALUES('440-aaa', 135000, '259hv', 'audi');
INSERT INTO Car VALUES('441-aaa', 152540, '259hv', 'audi');
INSERT INTO Car VALUES('442-aaa', 212540, '259hv', 'audi');
```

```
-- Owns(licenseNumbeer, ownerID)
```

```
INSERT INTO Owns VALUES('110-aaa', '11111A');
INSERT INTO Owns VALUES('111-aaa', '11111B');
INSERT INTO Owns VALUES('112-aaa', '74382A');
INSERT INTO Owns VALUES('113-aaa', '84929B');
INSERT INTO Owns VALUES('114-aaa', '38495C');
```

```
INSERT INTO Owns VALUES('220-aaa', '11111A');
INSERT INTO Owns VALUES('221-aaa', '11111B');
INSERT INTO Owns VALUES('222-aaa', '11111C');
INSERT INTO Owns VALUES('223-aaa', '74382A');
INSERT INTO Owns VALUES('224-aaa', '76463D');
```

```
INSERT INTO Owns VALUES('330-aaa', '38495C');
INSERT INTO Owns VALUES('331-aaa', '74382A');
```

```
INSERT INTO Owns VALUES('440-aaa', '11111A');
INSERT INTO Owns VALUES('441-aaa', '11111B');
INSERT INTO Owns VALUES('442-aaa', '76463D');
```

```
-- Device(deviceID, name)
```

```
INSERT INTO Device VALUES('874fhk', 'nosturi');
INSERT INTO Device VALUES('936kjd', 'kompressor');
INSERT INTO Device VALUES('648grb', 'hitsauskone');
INSERT INTO Device VALUES('385ihr', 'akselipukki');
```

```
-- Items(deviceID, itemID, maxValue)
```

```
INSERT INTO Items VALUES('874fhk', '1-kjdhf', 4);
INSERT INTO Items VALUES('936kjd', '2-kjdhf', 3);
INSERT INTO Items VALUES('648grb', '3-kjdhf', 2);
INSERT INTO Items VALUES('385ihr', '4-kjdhf', 4);
```

```

-- MaintenanceCycle(ID, startTime, endTime, duration, operationName, licenseNumber,
employerID, customerID, billNumber)
INSERT INTO MaintenanceCycle VALUES('88888A', '2022-06-01 9:00', '2022-06-01 11:00', 2,
'jarrujen tarkastus', '110-aaa', '573hdik', '11112A', '48849');
INSERT INTO MaintenanceCycle VALUES('88888B', '2023-06-01 9:00', '2023-06-01 11:00', 2,
'jarrujen tarkastus', '110-aaa', '573hdik', '11112A', '35990');
INSERT INTO MaintenanceCycle VALUES('88888C', '2024-06-01 9:00', '2024-06-01 11:00', 2,
'jarrujen tarkastus', '110-aaa', '573hdik', '11112A', '48953');
INSERT INTO MaintenanceCycle VALUES('88888D', '2022-06-01 12:00', '2022-06-01 14:00', 2, 'öljyn
vaihto', '220-aaa', '573hdik', '11112A', '83830');
INSERT INTO MaintenanceCycle VALUES('88888E', '2023-06-01 12:00', '2023-06-01 14:00', 2, 'öljyn
vaihto', '220-aaa', '573hdik', '11112A', '13048');
INSERT INTO MaintenanceCycle VALUES('88888F', '2024-06-01 12:00', '2024-06-01 14:00', 2, 'öljyn
vaihto', '220-aaa', '573hdik', '11112A', '83802');
INSERT INTO MaintenanceCycle VALUES('77777A', '2022-06-02 12:00', '2022-06-02 14:00', 2, 'öljyn
vaihto', '220-aaa', '573hdik', '112233', '49870');
INSERT INTO MaintenanceCycle VALUES('77777B', '2022-06-03 12:00', '2022-06-03 14:00', 2, 'öljyn
vaihto', '220-aaa', '573hdik', '554422', '82904');
INSERT INTO MaintenanceCycle VALUES('77777C', '2022-06-04 12:00', '2022-06-04 14:00', 2, 'öljyn
vaihto', '220-aaa', '573hdik', '37856', '20394');

```

```

-- DeviceReservation(ID, itemID, deviceID, startTime, endTime, duration, status)
INSERT INTO DeviceReservation VALUES('88888A', '1-kjdhf', '874fhk', '2022-06-01 9:00', '2022-06-
01 11:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('88888B', '1-kjdhf', '874fhk', '2023-06-01 9:00', '2023-06-
01 11:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('88888C', '1-kjdhf', '874fhk', '2024-06-01 9:00', '2024-06-
01 11:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('88888D', '1-kjdhf', '874fhk', '2022-06-01 12:00', '2022-
06-01 14:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('88888E', '1-kjdhf', '874fhk', '2023-06-01 12:00', '2023-06-
01 14:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('88888F', '1-kjdhf', '874fhk', '2024-06-01 12:00', '2024-06-
01 14:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('77777A', '1-kjdhf', '874fhk', '2022-06-02 12:00', '2022-06-
02 14:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('77777B', '1-kjdhf', '874fhk', '2022-06-03 12:00', '2022-06-
03 14:00', 2, 'available');
INSERT INTO DeviceReservation VALUES('77777C', '1-kjdhf', '874fhk', '2022-06-04 12:00', '2022-06-
04 14:00', 2, 'available');

```

```

-- SparePartCounter(ID, partID, count)
INSERT INTO SparePartCounter VALUES('88888A', '84375', 4);
INSERT INTO SparePartCounter VALUES('88888D', '24368', 1);
INSERT INTO SparePartCounter VALUES('88888E', '24368', 1);
INSERT INTO SparePartCounter VALUES('88888F', '24368', 1);
INSERT INTO SparePartCounter VALUES('88888D', '84692', 2);
INSERT INTO SparePartCounter VALUES('88888E', '84692', 2);
INSERT INTO SparePartCounter VALUES('88888F', '84692', 2);
INSERT INTO SparePartCounter VALUES('77777A', '24368', 1);
INSERT INTO SparePartCounter VALUES('77777B', '24368', 1);
INSERT INTO SparePartCounter VALUES('77777C', '24368', 1);
INSERT INTO SparePartCounter VALUES('77777A', '84692', 1);
INSERT INTO SparePartCounter VALUES('77777B', '84692', 1);
INSERT INTO SparePartCounter VALUES('77777C', '84692', 1);

-- Contain(operationID, operationName)
INSERT INTO Contain VALUES('5hieu59', 'jarrujen tarkastus');
INSERT INTO Contain VALUES('5hieu59', 'öljyn vaihto');
INSERT INTO Contain VALUES('5hieu59', 'ilmansuodattimen vaihto');
INSERT INTO Contain VALUES('5hieu59', 'rikkinaisien osien vaihto');
INSERT INTO Contain VALUES('60riker', 'jarrujen tarkastus');
INSERT INTO Contain VALUES('60riker', 'öljyn vaihto');
INSERT INTO Contain VALUES('60riker', 'ilmansuodattimen vaihto');
INSERT INTO Contain VALUES('60riker', 'rikkinaisien osien vaihto');
INSERT INTO Contain VALUES('49jrfhe', 'ilmansuodattimen vaihto');
INSERT INTO Contain VALUES('49jrfhe', 'rikkinaisien osien vaihto');
INSERT INTO Contain VALUES('tiksje8', 'öljyn vaihto');
INSERT INTO Contain VALUES('tiksje8', 'ilmansuodattimen vaihto');
INSERT INTO Contain VALUES('tiksje8', 'rikkinaisien osien vaihto');

-- Use(operationID, deviceID)
INSERT INTO Use VALUES('5hieu59', '874fhk');
INSERT INTO Use VALUES('5hieu59', '936kjd');
INSERT INTO Use VALUES('5hieu59', '648grb');
INSERT INTO Use VALUES('5hieu59', '385ihr');
INSERT INTO Use VALUES('60riker', '874fhk');
INSERT INTO Use VALUES('60riker', '936kjd');
INSERT INTO Use VALUES('60riker', '648grb');
INSERT INTO Use VALUES('60riker', '385ihr');
INSERT INTO Use VALUES('49jrfhe', '874fhk');
INSERT INTO Use VALUES('49jrfhe', '936kjd');
INSERT INTO Use VALUES('tiksje8', '874fhk');
INSERT INTO Use VALUES('tiksje8', '385ihr');

```

8 Käyttötapauksia

1) Etsi kaikki autokorjaamon autoja

```
SELECT *  
FROM Car;
```

2) Tulostaa kaikki autokorjaamon operaatiot

```
SELECT operationName  
FROM ListOfOperations;
```

3) Tulostaa kaikkien osien nimet ja hinnat

```
SELECT name, price  
FROM SpareParts;
```

4) Laskee keskiarvoinen operaatioiden kesto

```
SELECT AVG(duration)  
FROM Operations;
```

5) Tulostaa asiakkaan Riina Raksalainen puhelinnumero

```
SELECT phone  
FROM Customer  
WHERE name = 'Riina Raksalainen';
```

6) Tulostaa työntekijöiden ID sekä nimet, joiden status on "saattavilla"

```
SELECT Employer.employerID, name  
FROM Absence, Employer  
WHERE Absence.employerID = Employer.employerID AND status = 'available';
```

7) Tulostaa autojen rekisterinumerot, joiden ajettu kilometrinmäärä on pienempi kuin 100000

```
SELECT licenseNumber  
FROM Car  
WHERE kilometres < 100000;
```

8) Tulostaa kaikkien työntekijöiden- sekä laiteiden nimet, joiden status on 'saattavilla'

```
SELECT DISTINCT Employer.name, Device.name  
FROM DeviceReservation, Device, Absence, Employer  
WHERE DeviceReservation.status = 'available' AND Absence.status = 'available' AND  
DeviceReservation.deviceID = Device.deviceID AND Absence.employerID = Employer.employerID
```

9) Tulostaa montako kappaletta laitteita autokorjaamolla on

```
SELECT name, maxValue  
FROM Device, Items  
WHERE Device.deviceID = Items.deviceID
```

10) Tulostaa kaikki asiakkaiden tiedot, jotka tuo autoja omassa omistuksessa autokorjaamoon

```
SELECT *  
FROM Customer, Owners  
WHERE Customer.name = Owners.name;
```

11) Tulostaa asiakkaiden nimet, jotka omistavat autoja, mutta eivät tuoneet niitä vielä autokorjaamoon

```
SELECT name  
FROM Owns, Owners  
WHERE Owns.ownerID = owners.ownerID  
EXCEPT  
SELECT Customer.name  
FROM Customer, Owners  
WHERE Customer.name = Owners.name;
```

12) Tulostaa kaikki laitteiden nimi sekä määrä, joiden määrä on enemmän kuin kaksi

```
SELECT name, maxValue  
FROM Items, Device  
WHERE Device.deviceID = Items.deviceID AND maxValue > 2
```

Korjauksia edellisessä palautuksessa

- 1) Assosiaatio-relaatio – osion lisättiin Owns-, Contain- ja Use-relaatiot.
- 2) MaintenanceCycle-relaatiolle lisättiin billNumber-attribuutti.