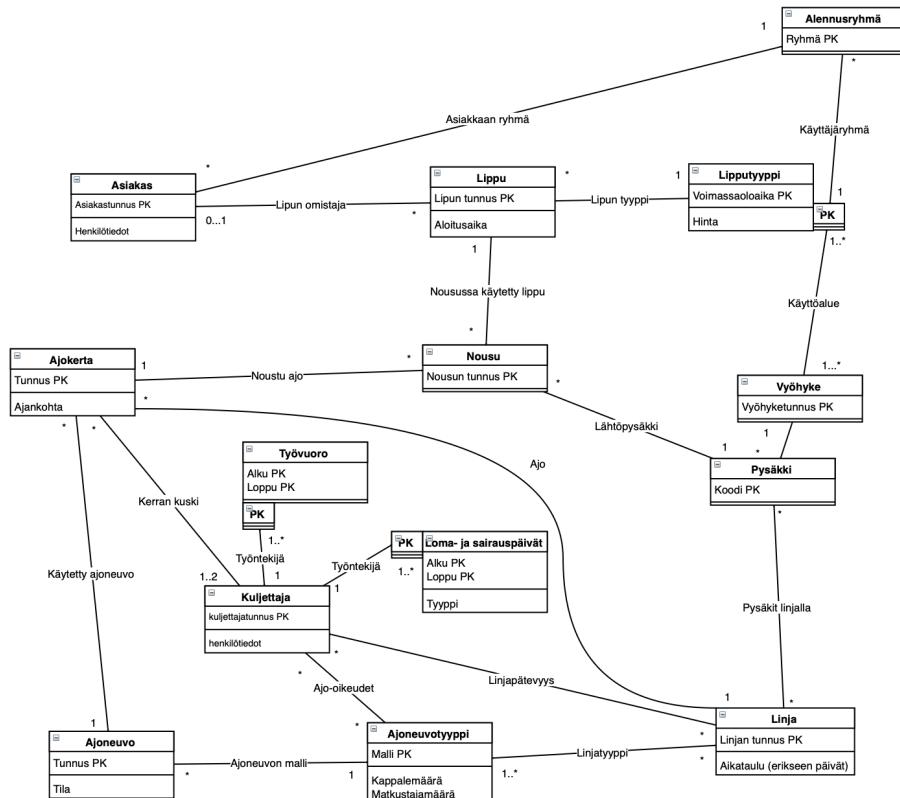


Aalto University
CS-A1150

Tietokannat-projekti Palautus 2

1 UML-kaavio



Kuva 1: UML-kaavio

2 Riippuvuudet

Kuvassa 1 on esitetty UML-kaavio suunnitellusta joukkolinjaoperaattorin tietokannasta. Tietokannassa on voimassa seuraavat funktioalaiset riippuvuudet.

asiakastunnus → henkilötiedot, asiakkaan alennusryhmä

lipun tunnus → lipun aloitusaika, lipun omistaja, lipun tyyppi

lipputyyppin voimassaoloaika, lipputyyppin käyttäjäryhmä, lipputyyppin vyöhyke-ryhmä → lipun hinta

nousun tunnus → nousussa käytetty lippu, noustu ajo, nousupysäkki

ajokerran tunnus → ajokerran ajankohta, ajokerran kuljettaja, käytetty ajo-neuvo

kuljettajatunnus → kuljettajan ajo-oikeudet, -||- linjapätevytiedot, -||- henkilötiedot

työntekijä, vapaan alku, vapaan loppu → vapaan tyyppi

linjan tunnus → linjan aikataulu, linjan ajoneuvon tyyppi, linjan pysäkit, linjan ajokerrat

ajoneuvon malli → ajoneuvotyyppi, ajoneuvotyyppin kappalemäärä, ajoneuvotyyppin matkustajamäärä

ajoneuvon tunnus → ajoneuvon tila, ajoneuvon tyyppi

vyöhyketunnus → pysäkit

pysäkkitunnus → vyöhyke

Funktioaalista riippuvuuksista kaikki, joilla on vasemmalla puolella jokin tunnus ovat käypiä, sillä jokaiselle monikolle tunnus on unikki ja näin ovat siihen liittyvät tiedot aina yksilölliset. Lipun hinta puolestaan riippuu lipun kestosta ja laajuudesta, tähän annetaan myös alennus riippuen käyttäjäryhmästä ja näin saatiin kolmas riippuvuuksista. Jokaisella työntekijällä voi olla kerrallaan vain yksi vapaa joten työntekijä, vapaan alku ja vapaan loppu määrittelevät näin yksikäsitteisesti sen tilan luoden kuudennessa riippuvuuden. Yhdeksänessä riippuvuudessa kuvataan miten ajoneuvon malli kertoo minkä tyyppinen ajonevo on (bussi,juna,,,), sen sillä matkustavien henkilöiden sallitun määrä ja koska käsittelemme vain yhtä operaattoria, on mallin ajoneuvojen määrä yksikäsiteinen.

3 Relaatiokaavio

Ensimmäiseksi muutetaan UML-kaavio relaatiokaavioksi normaalien käytäntöjen mukaan luoden välirelaatiot monesta moneen luokkien välille.

Alennusryhmä(ryhmä)

Asiakas(asiakastunnus, henkiötiedot, alennusryhmä)

Lippu(lipun tunnus, aloitusaika, omistaja, lipuntyyppi)

Lipputyyppi(voimassaoloaika, alennusryhmä, käyttöalue)

Käyttöalue(ryhmän nimi, vyöhyketunnus)

Vyöhyke(vyöhyketunnus)

Pysäkki(pysäkkitunnus, vyöhyke)

PysäkitLinjalla(linjatunnus, pysäkkitunnus)
Linja(linjatunnus, ajoneuvotyyppi, aikatauluArki, aikatauluLauantai, aikatauluPyhä)
Ajoneuvotyyppi(malli, ajoneuvotyyppi, kappalemäärä, matkustajamäärä)
Ajoneuvo(rekisteritunnus, ajoneuvonmalli, tila)
Ajokerta(ajotunnus, käytettyAjoneuvo, kuski, ajankohta)
Kuljettaja(henkilötiedot)
Linjapätevyydet(kuski, linja)
Ajoneuvopätevyydet(kuski, ajoneuvotyyppi)
Työvuoro(kuljettaja, alku, loppu)
LomaSairausPäivä(kuljettaja, alku, loppu, tyyppi)
Nousu(nousuntunnus, noustuAjokerta, käytettyLippu, käytettyPysäkki)

Mallista voi huomata jokaisen avaimen olevan relaationsa yliavain, sillä kaikki relaatioiden muut ominaisuudet riippuvat / ovat yksikäsitteisesti määritelty sen avaimen mukaan. Tämä tarkoittaa relatiokaavion olevan jo valmiiksi BCNF:ssä ja siihen ei tarvitse tehdä muutoksia.

4 Mallin kuvaus

4.1 Selitys

Käyttöalue on joukko vyöhykkeitä, joilla lipputyyppi on kelvollinen. Pysäkit linjalla on joukko pysäkkejä, jotka kuuluvat tietyn linjan reitille. Kullakin ajoneuvolla on tiedossa tarkka malli, josta riippuu sen matkustajakapasiteetti sekä operaattorin tästä mallia omistamien ajoneuvojen kappalemäärä. Ajoneuvotyyppi taas kertoo, millainen kulkuväline on kyseessä, esim. metro.

Kullakin kuljettajalla on oikeudet ajaa vain tiettyjä ajoneuvotyypejä, kuten esimerkiksi D-luokan ajokortti bussin ajamiseen. Lisäksi näistä tyypeistä kuljettajalla on erikseen pätevyydet kullekin linjalle, joita kuljettaja voi saada ajamalla linjan muutaman kerran kokeneemman kuljettajan kanssa. Tällaisista harjoituskerroista tallennetaan tietokantaan ajokerta, jossa on kaksi kuljettajaa.

Operaattori myy lippuja myös esimerkiksi lippuautomaateilla, jolloin lipun omistajasta ei tallennu tietoa. Tästä syystä tietokannassa voi olla lippu, jolla ei ole omistajaa. Tällaiseen lippuun ei kuitenkaan ole saatavilla alennuksia ja siksi alennetun lipun voi ostaa vain alennusryhmään kuuluva, ja siten tietokantaan tallennettu asiakas. Yksi alennusryhmä on lisäksi "ei alennusryhmää", joten kaikille rekisteröidyllille asiakkaille on määritetty alennusryhmä.

4.2 Anomalia

Lipputyyppissä voi esiintyä anomalioita, sillä tällaisessa mallissa jokaisen alennusryhmää käyttöalue-yhdistelmän hinta on erikseen tallennettu tietokantaan. Vaihtoehtoista jokaiselle alennusryhmälle voisi lisätä alennusprosentti-attribuutin, poistaa alennusryhmän PK-yhteyden lipputyyppiin ja määrittää kullekin lipputyyppille vain yhden hinnan. Tämä vähentäisi lipputyyppin hinta-attribuutin toisteisuitta, mutta samalla myös vähentäisi lipputyyppien hintojen yksilötävyyttä kullekin alennusryhmälle.

5 Ensimmäisen palautuksen jälkeen tehdyt muutokset

5.1 Muutokset relaatioihin

Korvataan Lippu-relaation lipuntyyppi-attribuutti Lipputyyppi-relaation pääavaimilla. Lisätään Lipputyyppi-relaatioon attribuuteiksi hinta ja lipun nimi, jotka riippuvat relaation pääavaimista. Hinnan lisääminen attribuutiksi korvaa tarvittavuuden selvittää yksittäiset alennusprosentit tietylle luokalle, sillä lipputyyppi sisältää kaikki mahdolliset ryhmät. Muutetaan Lipputyyppi-luokan "voimassaoloaika-attribuutin nimeksi "lipun kesto" kaavion selkeyttämiseksi.

Erotetaan ajoneuvotyyppi ja -malli toisistaan siten, että entisen Ajoneuvotyyppi-luokan nimeksi muutetaan Ajoneuvomalli ja luodaan uusi luokka Ajoneuvotyyppi, jonka ainoa attribuutti on sen pääavain "tyyppi".

Lisätään luokka "KäyttöalueVyöhyke", jonka pääavaimet ja ainoat attribuutit ovat aluetunnus ja vyöhyketunnus. Muutetaan Käyttöalue-luokkaa siten, että sen ainoa attribuutti on aluetunnus.

Luodaan uusi luokka "Aikataulu", jonka pääavaimiksi lisätään ajettu linja, lähtöaika sekä päivätunniste, joka kuuluu joukkoon {"arki", "la", "pyhä"}. Tämän myötä poistetaan linjan kaikki "aikataulu...-attribuutit Linja-luokasta. Lisätään Aikataulu-luokkaan sekä Ajokerta-luokkaan binääriattribuutti "ajosuunta", joka riippuu kummassakin luokassa sen luokan pääavaimista.

Lisätään Kuljettaja-luokkaan uusi pääavain "tunnus" ja muutetaan henkilötiedot perusattribuutiksi.

5.2 Muutokset assosiaatioihin

Assosiaatio Ajokerta-luokasta Ajoneuvo-luokkaan muutetaan tyypiksi *-0..1, sillä ajokerran ajoneuva ei välttämättä tiedetä. Lisäksi samasta syystä muutetaan

Ajokerta-luokan sekä Kuljettaja-luokan välinen assosiaatio tyypiksi *-0..2 ottaen huomioon tilanteen, jossa kuljettaja harjoittelee linjaa kokeneemman kuljettajan kanssa, jolloin kuljettajia on kaksi.

Uuteen Ajoneuvotyyppi-luokkaan siirtyvät assosiaatiot "Linjatyyppi" sekä "Ajo-oikeudet" samoilla assosiaatiotyypeillä. Ajoneuvotyyppi- sekä Ajoneuvomalli-luokkien välille luodaan assosiaatio "Mallin tyyppi", joka on tyypia 1*.

Siirretään "Käyttöalue-assosiaatio Käyttöalue-luokan ja Lipputyyppi-luokan väliksi ja lisätään Käyttöalue- ja Vyöhyke-luokkien välille assosiaatio "Alueen vyöhykkeet", joka on tyypia 1...*-1...*. Lisätään Aikataulu- ja Linja-luokkien välille assosiaatio "aikataulun linja", joka on muotoa *-1.

5.3 Muutokset riippuvuksiin

Poistetaan seuraavat virheelliset riippuvuudet:

pysäkkitunnus → vyöhyke

kuljettajatunnus → kuljettajan ajo-oikeudet, kuljettajan linjapätevydet

Lisätään seuraavat edellä mainitut uudet riippuvuudet:

lipputyyppin voimassaoloaika, lipputyyppin käyttäjäryhmä, lipputyyppin vyöhyke-ryhmä → lipun nimi

lähtöaika, ajettu linja, arkipäivätunniste → ajosuunta

kuljettajatunnus → kuljettajan henkilötiedot

ajotunnus → suunta, linja

Todetaan, että relatiokaaviossa ei ole muita riippuvuuksia, joten kaavio pysyy BCNF:ssä

6 Käytännölliset haut

Tässä kappaleessa näytämme käytännöllisiä hakuja tietokannassamme. Nämä komennot ovat suoritettu dokumentin lopussa tehdylle tietokannalle ja muukaan on liitetty haun tuottama printti.

6.1 Pysäkin aikataulu

Voidaan hakea minkä tahansa pysäkin aikataulu Esim. pysäkin 300.

```
Select lähtöPäätepysäkiltä, linja, suunta, tyyppi  
From PysäkkienLähdot  
Where pysäkkitunnus = 300  
Order By tyyppi, lähtöPäätepysäkiltä;  
07:00 1 0 Arki
```

```
08:00 1 1 Arki
09:00 2 0 Arki
10:00 2 1 Arki
15:00 1 0 Lauantai
16:00 1 1 Lauantai
17:00 2 0 Lauantai
18:00 2 1 Lauantai
11:00 1 0 Pyhä
12:00 1 1 Pyhä
```

6.2 Kuljettajan työvuorot

Haetaan kuljettajan 1 työvuorot

```
SELECT *
FROM Työvuoro
WHERE kuljettaja=1;
1 2025-05-06 07:00 2025-05-06 15:00
```

6.3 Asiakkaan henkilötiedot

Etsitään asiakkaan henkilötiedot, jotta voidaan varmistaa esim. henkilöllisyys tai opiskelijastatus. Esimerkkinä asiakkaalle tunnuksella 101.

```
Select henkilötiedot, alennusryhmä
From Asiakas
Where tunnus = 101;
Teppo Testaaja Opiskelija
```

6.4 Linjan pysäkit

Haetaan linjalle 1 kuuluvat pysäkit

```
SELECT pysäkkitunnus
FROM PysäkitLinjalla
WHERE linjatunnus=1;
100
200
300
```

6.5 Asiakkaan tekemät nousut ajoneuvoihin

Haetaan asiakkaan tekemät nousut asiakkaalle tunnuksella 101

```
Select pysäkki, ajankohta, linja
From Nousu
Join Ajokerta ON ajokerta = ajotunnus
Join Lippu L ON käytettylippu = L.tunnus
Join Asiakas A ON omistaja = A.tunnus
Where A.tunnus = 101;
```

6.6 Ajokerran nousut

Haetaan ajokertojen tuotto eli jokaiselle ajokerralle nousseiden asiakkaiden ostamien lippujen yhteishinta

```
SELECT ajokerta, sum(hinta)
FROM Nousu JOIN Lippu ON käytettylippu=tunnus NATURAL JOIN Lipputyyppi
GROUP BY ajokerta;

1 3
2 5
3 4
```

6.7 Lipun kattavuus

Haetaan kaikki pysäkit, joille pääsee lipulla 201:

```
SELECT pysäkkitunnus
FROM Pysäkki JOIN KäyttöalueVyöhyke on vyöhyke=vyöhyketunnus
WHERE aluetunnus in
      (select käyttöalue
       from Lippu
       where tunnus=201);

100
200
600
700
1100
1200
```

6.8 Lipun hinta

Haetaan liput hintajärjestyksessä asiakkaalle, siten että sisältää vyöhykkeen 1 ja henkilölle asiakastunnusella 102.

```
SELECT nimi, hinta
FROM Lipputyyppi
Natural Join Asiakas
WHERE käyttöalue in
    (select aluetunnus
     from käyttöalueVyöhyke
      where vyöhyketunnus=10)
     AND tunnus = 102
Order By hinta ASC;
```

```
Kertalippu alue 1 3
Kertalippu alue 3 4
```

6.9 Kuskien sairas- ja lomapäivät

Haetaan yhden kuskin poissaolot esimerkiksi kuskielle tunnuksella 1.

```
Select alku, loppu, tyyppi
From LomaSairausPäivä
Where kuljettaja = 1;
```

```
2025-05-01 2025-05-05 loma
2025-06-01 2025-06-07 loma
```

6.10 Mahdolliset kuskit

Haetaan kaikki kuskit jotka ovat töissä linjan vuoron alkaessa esim. ajankohtana 2025-05-06 10:00.

```
Select *
From Työvuoro
Where alku <= '2025-05-06 10:00' AND loppu > '2025-05-06 10:00';
```

```
1 2025-05-06 07:00 2025-05-06 15:00
```

6.11 Kuskien suorittamat ajot

Haetaan mitkä matkat kuski on tehnyt esimerkiksi kuski tunnusnumerolla 3.

```
Select ajotunnus, ajankohta, linja, suunta, käytettyajoneuvo
From Ajokerta
Where kuski = 3;
```

```
3 2025-05-06 12:00 3 0 DEF-456
6 2025-05-07 06:30 3 1 ABC-123
9 2025-05-07 07:45 5 1 DEF-456
13 2025-05-07 10:00 2 1 ABC-123
```

6.12 Asiakkaan liput

Haetaan nimet asiakkaan 101 ostamille lipuille, jotka ovat alkaneet 1.5.2025 jälkeen:

```
SELECT nimi
FROM Lippu NATURAL JOIN Lipputyyppi
WHERE omistaja = 101 AND aloitusaika >= "2025-05-02 00:00";
```

30 min Opiskelija Alue 1

6.13 Ajoneuvotyyppien määrä

Haetaan kappalemääät kaikille ajoneuvotyypeille:

```
SELECT ajoneuvotyyppi, sum(kappalemäärä)
FROM Ajoneuvotyyppi NATURAL JOIN Ajoneuvomalli
GROUP BY ajoneuvotyyppi
```

Bussi 2
Raitiovaunu 1

6.14 Linjapätevyys

Haetaan kuljettajan 1 linjapätevyydet:

```
SELECT linja
FROM Linjapätevyydet
where kuljettaja=1
```

1

6.15 Ajoneuvon ajot

Haetaan ajotunnus, ajankohta ja linja ajoille, jotka on merkitty ajoneuvolle rekisteritunnuksella "ABC-123"

```
Select ajotunnus, ajankohta, linja
From Ajokerta
Where käytettyajoneuvo = 'ABC-123'
```

```
Order by ajankohta;
```

```
1 2025-05-06 08:00 1  
6 2025-05-07 06:30 3  
4 2025-05-07 09:00 2  
13 2025-05-07 10:00 2
```

6.16 Vyöhykkeellä toimivat linjat

Haetaan kaikki linjat, jotka toimivat tutkittavalla vyöhykkeellä. Esimerkiksi vyöhykkeellä 10 toimivat linjat

```
SELECT linjatunnus  
FROM PysäkitLinjalla NATURAL JOIN Pysäkki  
Where vyöhyke = 10
```

```
1  
3  
4
```

6.17 Pätevät työvuorot linjalle

Etsii kaikki työvuorot joiden työntekijällä on oikeus ajaa kyseistä linjaa. Esimerkiksi linjalle 1 olevat työvuorot:

```
SELECT henkilötiedot, alku, loppu, linja  
FROM Työvuoro T  
JOIN Linjapätevyydet L ON L.kuljettaja = T.kuljettaja  
JOIN Kuljettaja K ON K.tunnus = T.kuljettaja  
Where linja = 1;
```

```
Maija Meikäläinen 2025-05-06 07:00 2025-05-06 15:00 1  
Kari Kuljettaja 2025-05-06 14:00 2025-05-06 22:00 1
```

7 Tietorakenteen luominen

7.1 Taulut

```
CREATE TABLE Alennusryhmä(  
    ryhmä TEXT PRIMARY KEY  
)  
CREATE TABLE Asiakas(  
    tunnus integer not null primary key,  
    henkilötiedot text,  
    alennusryhmä text not null,  
    foreign key (alennusryhmä) references Alennusryhmä(ryhmä)
```

```

);
create table Lippu(
    tunnus integer not null primary key,
    aloitusaika text,
    omistaja text,
    lipunkesto INTEGER,
    alennusryhmä TEXT,
    käyttöalue INTEGER,
    FOREIGN KEY (lipunkesto, alennusryhmä, käyttöalue)
        REFERENCES Lipputyyppi(lipunkesto, alennusryhmä, käyttöalue)
);
create table Lipputyyppi(
    lipunkesto integer not null,
    alennusryhmä text not null references alennusryhmä(ryhmä),
    käyttöalue integer not null references Käyttöalue(aluetunnus),
    hinta INTEGER,
    nimi TEXT,
    primary key(lipunkesto, alennusryhmä, käyttöalue)
);
create table Käyttöalue(
    aluetunnus integer not null,
    PRIMARY KEY(aluetunnus)
);
create table Vyöhyke(
    vyöhyketunnus integer not null primary key
);
create table KäyttöalueVyöhyke(
    aluetunnus integer not null REFERENCES Käyttöalue(aluetunnus),
    vyöhyketunnus integer not null REFERENCES Vyöhyke(vyöhyketunnus),
    PRIMARY KEY(aluetunnus, vyöhyketunnus)
);
create table Pysäkki(
    pysäkkitunnus integer not null primary key,
    vyöhyke integer not null references Vyöhyke(vyöhyketunnus)
);
create table PysäkitLinjalla(
    linjatunnus integer not null references Linja(linjatunnus),
    pysäkkitunnus integer not null references Pysäkki(pysäkkitunnus),
    PRIMARY KEY(linjatunnus, pysäkkitunnus)
);

```

```

create table Linja(
    linjatunnus integer not null primary key,
    ajoneuvotyyppi text not null references Ajoneuvotyyppi(ajoneuvotyyppi)
);

create table Aikataulu(
    linja INTEGER REFERENCES Linja(linjatunnus),
    lähtöaika TEXT NOT NULL,
    tyyppi TEXT CHECK (tyyppi IN ("Arki", "Lauantai", "Pyhä")),
    suunta INTEGER CHECK (suunta IN (0,1)),
    PRIMARY KEY(linja, lähtöaika, tyyppi, suunta)
);

create table Ajoneuvotyyppi(
    ajoneuvotyyppi TEXT PRIMARY KEY
);

create table Ajoneuvomalli(
    malli text not null PRIMARY KEY,
    ajoneuvotyyppi text not null REFERENCES Ajoneuvotyyppi(ajoneuvotyyppi),
    kappalemäärä integer check (kappalemäärä>=0),
    kapasiteetti integer check (kapasiteetti>0)
);

create table Ajoneuvo(
    rekisteritunnus text not null primary key,
    ajoneuvomalli text not null references Ajoneuvomalli(malli),
    tila text not null check(tila in("käytössä", "vapaa", "rikki", "huollossa"))
);

create table Kuljettaja(
    tunnus integer not null primary key,
    henkilötiedot text
);

create table Ajokerta(
    ajotunnus integer not null primary key,
    käytettyajoneuvo text references ajoneuvo(rekisteritunnus),
    kuski integer references kuljettaja(tunnus),
    ajankohta text,
    suunta integer check(suunta in(0,1)),
    linja INTEGER references Linja(linjatunnus)
);

create table Linjapätevyydet(
    kuljettaja integer not null references Kuljettaja(tunnus),

```

```

linja integer references Linja(linjatunnus),
PRIMARY KEY(kuljettaja, linja)
);

create table Ajoneuvopätevyydet(
    kuljettaja integer not null references Kuljettaja(tunnus),
    ajoneuvotyyppi TEXT references Ajoneuvotyyppi(ajoneuvotyyppi),
    PRIMARY KEY(kuljettaja, ajoneuvotyyppi)
);

create table Työvuoro(
    kuljettaja integer not null references Kuljettaja(tunnus),
    alku text not null,
    loppu text not null,
    primary key(kuljettaja,alku,loppu)
);
create table LomaSairausPäivä(
    kuljettaja integer not null references Kuljettaja(tunnus),
    alku text not null,
    loppu text not null,
    tyyppi text not null check(tyyppi in("loma","sairaus")),
    primary key(kuljettaja,alku,loppu)
);
create table Nousu(
    nousutunnus integer not null primary key,
    ajokerta integer not null references Ajokerta(ajotunnus),
    käytettylipu integer not null references Lippu(tunnus),
    pysäkki integer not null references Pysäkki(pysäkkitunnus)
);

CREATE INDEX kuskienlomasairaus ON LomaSairausPäivä(kuljettaja);

CREATE INDEX kuskintyöt ON Työvuoro(kuljettaja);

CREATE INDEX mahdollisetKuskit ON Työvuoro(alku, loppu);

CREATE INDEX kuskiennajot ON Ajokerta(kuski);

CREATE INDEX pysäkkiNro ON PysäkitLinjalla(pysäkkitunnus);

--- Ei ole PRIMARY KEY

CREATE INDEX lippujenomistajat ON Lippu(omistaja);

CREATE INDEX ajankohdanajot ON Ajokerta(ajankohta);

```

```
CREATE INDEX ajoneuvonajot ON Ajokerta(käytettyajoneuvo);
```

7.2 Yksilöt

```
-- Alennusryhmät
INSERT INTO Alennusryhmä VALUES
('Opiskelija'),
('Eläkeläinen'),
('Aikuinen');

-- Käyttöalueet
INSERT INTO Käyttöalue VALUES
(1),
(2),
(3);

-- Vyöhykkeet
INSERT INTO Vyöhyke VALUES
(10),
(20),
(30),
(40),
(50);

-- Käyttöalueen vyöhykkeet
INSERT INTO KäyttöalueVyöhyke VALUES
(1, 10),
(1, 20),
(2, 30),
(2, 40),
(3, 10),
(3, 50);

-- Ajoneuvotyyppit
INSERT INTO Ajoneuvotyyppi VALUES
('Bussi'),
('Raitiovaunu');

-- Ajoneuvomallit
INSERT INTO Ajoneuvomalli VALUES
('Volvo X', 'Bussi', 1, 40),
('Mercedes M', 'Bussi', 1, 35),
('Skoda T', 'Raitiovaunu', 1, 100);

-- Ajoneuvot ja pseudolaukaisimet
```

```

INSERT INTO Ajoneuvo VALUES
('ABC-123', 'Volvo X', 'käytössä'),
('DEF-456', 'Mercedes M', 'huollossa'),
('XYZ-789', 'Skoda T', 'vapaa');

UPDATE Ajoneuvomalli
SET kappalemäärä = kappalemäärä + 2
WHERE malli = "Volvo XX";

UPDATE Ajoneuvomalli
SET kappalemäärä = kappalemäärä + 1
WHERE malli = "Skoda TT";

-- Pysäkit
INSERT INTO Pysäkki VALUES
(100, 10),
(200, 20),
(300, 30),
(400, 40),
(500, 50),
(600, 10),
(700, 20),
(800, 30),
(900, 40),
(1000, 50),
(1100, 10),
(1200, 20),
(1300, 30),
(1400, 40),
(1500, 50);

-- Linjat
INSERT INTO Linja VALUES
(1, 'Bussi'),
(2, 'Raitiovaunu'),
(3, 'Bussi'),
(4, 'Raitiovaunu'),
(5, 'Bussi');

-- Pysäkit linjoilla
INSERT INTO PysäkitLinjalla VALUES
-- Linja 1
(1, 100), (1, 200), (1, 300),
-- Linja 2
(2, 300), (2, 400), (2, 500),
-- Linja 3

```

```

(3, 600), (3, 700), (3, 800),
-- Linja 4
(4, 900), (4, 1000), (4, 1100),
-- Linja 5
(5, 1200), (5, 1300), (5, 1400), (5, 1500);

-- Aikataulut
INSERT INTO Aikataulu VALUES
-- Linja 1
(1, '08:00', 'Arki' ,1),
(1, '16:00', 'Lauantai',1),
(1, '12:00', 'Pyhä',1),
(1, '07:00', 'Arki' ,0),
(1, '15:00', 'Lauantai',0),
(1, '11:00', 'Pyhä',0),
-- Linja 2
(2, '10:00', 'Arki',1),
(2, '18:00', 'Lauantai',1),
(2, '09:00', 'Arki',0),
(2, '17:00', 'Lauantai',0),
-- Linja 3
(3, '07:30', 'Arki',1),
(3, '14:00', 'Lauantai',1),
(3, '06:30', 'Arki',1),
(3, '13:00', 'Lauantai',1),
-- Linja 4
(4, '09:00', 'Arki',0),
(4, '15:30', 'Pyhä',1),
-- Linja 5
(5, '06:45', 'Arki',0),
(5, '13:15', 'Lauantai',0),
(5, '17:00', 'Pyhä',0),
(5, '07:45', 'Arki',1),
(5, '14:15', 'Lauantai',1),
(5, '18:00', 'Pyhä',1);

-- Kuljettajat
INSERT INTO Kuljettaja VALUES
(1, 'Maija Meikäläinen'),
(2, 'Kari Kuljettaja'),
(3, 'Pekka Pätevä');

-- Ajoneuvopätevyydet
INSERT INTO Ajoneuvopätevyydet VALUES
(1, 'Bussi'),

```

```

(2, 'Raitiovaunu'),
(3, 'Bussi'),
(3, 'Raitiovaunu');

-- Linjapätevyydet
INSERT INTO Linjapätevyydet VALUES
(1, 1),
(2, 1),
(2, 2),
(3, 1);

-- Työvuorot
INSERT INTO Työvuoro VALUES
(1, '2025-05-06 07:00', '2025-05-06 15:00'),
(2, '2025-05-06 14:00', '2025-05-06 22:00');

-- Lomat ja sairauspoissaolot
INSERT INTO LomaSairauspäivä VALUES
(3, '2025-05-07', '2025-05-10', 'loma'),
(1, '2025-05-01', '2025-05-05', 'loma'),
(2, '2025-05-06', '2025-05-08', 'sairaus'),
(2, '2025-05-07', '2025-05-10', 'loma'),
(1, '2025-06-01', '2025-06-07', 'loma'),
(3, '2025-06-15', '2025-06-16', 'sairaus'),
(3, '2025-07-01', '2025-07-14', 'loma');

-- Ajokerrat
INSERT INTO Ajokerta VALUES
(1, 'ABC-123', 1, '2025-05-06 08:00', 0, 1),
(2, 'XYZ-789', 2, '2025-05-06 10:00', 1, 1),
(3, 'DEF-456', 3, '2025-05-06 12:00', 0, 3),
(4, 'ABC-123', 1, '2025-05-07 09:00', 0, 2),
(5, 'XYZ-789', 2, '2025-05-07 07:30', 1, 3),
(6, 'ABC-123', 3, '2025-05-07 06:30', 1, 3),
(7, 'DEF-456', 1, '2025-05-07 09:00', 0, 4),
(8, 'XYZ-789', 2, '2025-05-07 06:45', 0, 5),
(9, 'DEF-456', 3, '2025-05-07 07:45', 1, 5),
(10, 'DEF-456', 1, '2025-05-07 08:00', 1, 1),
(12, 'XYZ-789', 2, '2025-05-07 07:00', 0, 1),
(13, 'ABC-123', 3, '2025-05-07 10:00', 1, 2);

-- Asiakkaat
INSERT INTO Asiakas VALUES
(101, 'Teppo Testaaja', 'Opiskelija'),
(102, 'Anna Aikunen', 'Aikunen'),
(103, 'Ella Eläkeläinen', 'Eläkeläinen');

```

```

-- Lipputyyppit
INSERT INTO Lipputyyppi VALUES
(30, 'Opiskelija', 1, 3, '30 min Opiskelija Alue 1'),
(60, 'Aikuinen', 2, 5, '1h Aikuinen Alue 2'),
(0, 'Aikuinen', 1, 3, 'Kertalippu alue 1'),
(0, 'Aikuinen', 2, 3, 'Kertalippu alue 2'),
(0, 'Aikuinen', 3, 4, 'Kertalippu alue 3'),
(120, 'Eläkeläinen', 3, 4, '2h Eläkeläinen Alue 3');

-- Liput
INSERT INTO Lippu VALUES
(201, '2025-05-06 08:05', '101', 30, 'Opiskelija', 1),
(202, '2025-05-06 10:10', '102', 60, 'Aikuinen', 2),
(203, '2025-05-06 11:00', '103', 120, 'Eläkeläinen', 3);

-- Nousut
INSERT INTO Nousu VALUES
(301, 1, 201, 100),
(302, 2, 202, 400),
(303, 3, 203, 500);

```

7.3 Hakemistot

```

CREATE INDEX kuskienlomasairaus ON LomaSairausPäivä(kuljettaja);

CREATE INDEX kuskintyöt ON Työvuoro(kuljettaja);

CREATE INDEX mahdollisetKuskit ON Työvuoro(alku, loppu);

CREATE INDEX kuskienajot ON Ajokerta(kuski);

CREATE INDEX pysäkkiNro ON PysäkitLinjalla(pysäkkitunnus);

--- Ei ole PRIMARY KEY

CREATE INDEX lippujenomistajat ON Lippu(omistaja);

CREATE INDEX ajankohdanajot ON Ajokerta(ajankohta);

CREATE INDEX ajoneuvonajot ON Ajokerta(käytettyajoneuvo);

```

Ensimmäisestä hakemistosta löytää kunkin kuljettajan loma- tai sairauspäivät,

jotka ovat tarpeellisia esimerkiksi työvuorojen suunnittelun. Esimerkiksi kuljettajat voivat hakea työvuoronsa toisesta hakemistosta. Kolmas hakemisto on hyvä työkalu ajokertojen kuskien suunnittelun. Neljännestä hakemistosta voidaan tarkistaa, kuka kuljettaja ajaa mitäkin linjaan tietyllä ajanjäädikillä. Viidennessä hakemistosta löytää nopeasti pysäkillä pysähtyvät linjat. Kuudennesta hakemistosta voidaan esimerkiksi tarkistaa, onko tietyllä asiakkaalla voimassa oleva lippu. Seitsemännestä hakemistosta voi nähdä tietyllä ajanjäädikillä lähtevät ajot, mikä auttaa kuljettajien sijoittelussa ajoihin. Viimeisestä hakemistosta löytää nopeasti esimerkiksi ne ajot, joihin ei ole vielä suunniteltu ajoneuvoa.

7.4 Näkymät

Näyttää milloin minkäkin pysäkin kohdalla pysähtyvä ajokerta on lähtenyt ensimmäiseltä pysäkiltä

```
CREATE VIEW PysäkkienLähdot AS
    Select pysäkkitunnus, linjatunnus AS linja, suunta,
          lähtöaika AS lähtöPäätelypysäkiltä, tyyppi
    From PysäkitLinjalla
    Natural Join Linja
    Join Aikataulu ON linjatunnus = linja;
```