

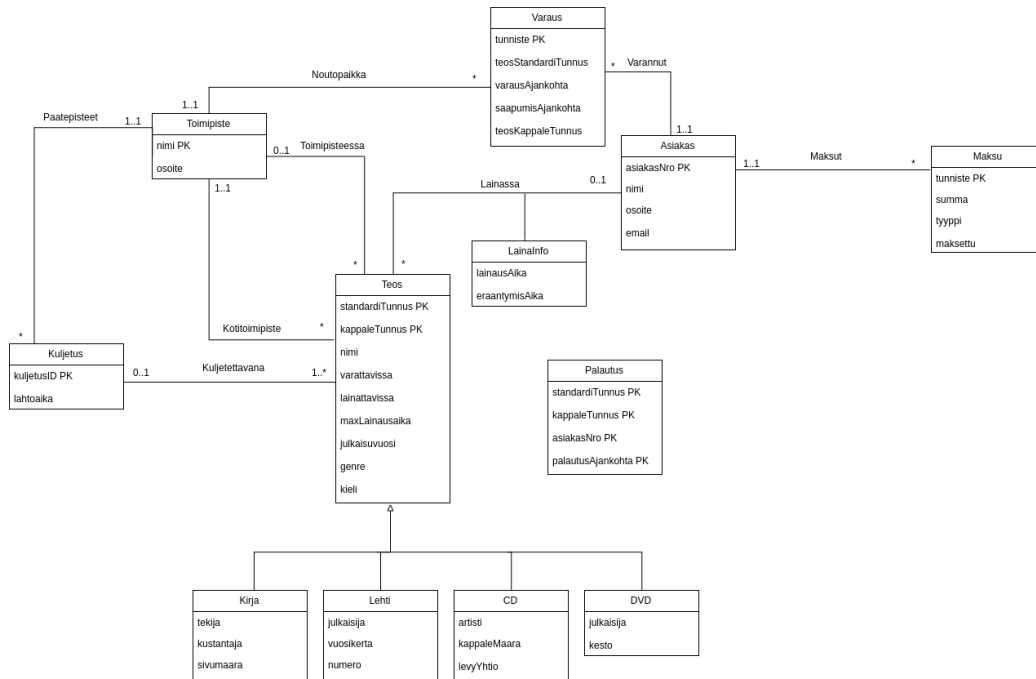
Tietokannat - harjoitustyö (osa 2)

Miska Kananen (652102, miska.kananen@aalto.fi)

Teemu Mäkinen (628835, teemu.v.makinen@aalto.fi)

14. toukokuuta 2019

1 UML-kaavio



2 Relaatiokaavio

- Teos(standardiTunnus, nimi, julkaisuvuosi, genre, kieli)
- Kappale(standardiTunnus, kappaleTunnus, varattavissa, lainattavissa, maxLainausaika, kotitoimipiste)
- Kirja(standardiTunnus, tekija, kustantaja, sivumaara)
- Lehti(standardiTunnus, julkaisija, vuosikerta, numero)
- CD(standardiTunnus, artisti, kappaleMaara, levyYhtio)
- DVD(standardiTunnus, julkaisija, kesto)
- Asiakas(asiakasNro, nimi, osoite, email)
- Varaus(tunniste, teosStandardiTunnus, teosKappaleTunnus, varausAjankohta, saapumisAjankohta, varaajaAsiakasNro, noutoToimipiste)

- `Palautus(standardiTunnus, kappaleTunnus, palautusAjankohta, asiakasNro)`
- `Maksu(tunniste, summa, tyyppi, maksettu, asiakasNro)`
- `Toimipiste(nimi, osoite)`
- `Kuljetus(kuljetusID, lahtoaika, lahtoToimipiste, paateToimipiste)`
- `Lainassa(standardiTunnus, kappaleTunnus, lainausAika, eraantymisAika, asiakasNro)`
- `Toimipisteessa(standardiTunnus, kappaleTunnus, toimipisteNimi)`
- `Kuljetettavana(standardiTunnus, kappaleTunnus, kuljetusID)`

3 Selostus ratkaisusta

Kirjaston teoksia kuvataan **Teos**-relaatiolla, jonka yksi monikko kuvaa tietyn teoksen yksittäistä kappaletta. Relaation avain koostuu standarditunnuksesta, joka yksilöi teoksen ja kappaletunnuksesta, joka yksilöi yksittäisen kappaleen tietyistä teoksesta. Standarditunnus on esim. kirjoilla ISBN ja lehdillä ISSN. Kappaletunnuksena voi toimia esimerkiksi juokseva numerointi. Yksittäiselle kappaleelle voi määrittää sen attribuutteina, onko se varattavissa ja lainattavissa ja jos on, kuinka pitkäksi aikaa.

Teos-relaatio sisältää kaikille teoksille yhteiset ominaisuudet, kuten kieli ja genre, ja teostyypeille spesifit ominaisuudet on määriteltä **Kirja**, **Lehti**, **CD** ja **DVD** -relaatioissa. Kaikki teokset ovat joko kirjoja, lehtiä, CD:itä tai DVD:itä.

Yksittäinen teoksen kappale on kulloinkin joko jossakin toimipisteessä, lainassa tai kuljetettavana. Relaatiot **Toimipisteessa**, **Lainassa** ja **Kuljetettavana** kertovat, mitä teoksia on kulloinkin kyseisissä paikoissa. Kun kappaleen paikka muuttuu esimerkiksi lainauksen tai kuljetuksen yhteydessä, päivitetään relaatioita vastaavasti.

Kirjaston asiakkaita kuvataan **Asiakas**-relaatiolla, ja yksittäinen asiakas yksilöidään asiakasnumerolla, joka voi olla esimerkiksi juokseva numerointi. Asiakkaista tallennetaan oleelliset henkilötiedot, kuten nimi ja osoite.

Asiakkaiden maksettavia maksuja kuvataan **Maksu**-relaatiolla. Kullakin maksulla on yksiselitteinen tunniste, joka voi olla esim. juokseva numerointi, summa, tyyppi, kuten varausmaksu tai myöhästymismaksu ja tieto siitä, onko asiakas maksanut maksun.

Relaatio **Varaus** kuvaa tiettyyn teokseen kohdistuvaa varausta. Varaukseen tallennetaan teoksen standarditunnus, joka määrittää, mihin teokseen varaus kohdistuu, sekä varausajankohta, joka määrittää varauksen sijainnin varausjonossa.

Niin kauan kuin asiakkaalle ei ole tarjottu varausta noudettavaksi, varaus kohdistuu kaikkiin teoksen kappaleisiin ja mitään yksittäistä kappaletunnusta ei ole määritelty. Kun jokin teoksen kappale vapautuu, kohdistetaan varaus kyseiseen kappaleeseen, eikä varaus enää vaikuta teoksen muihin kappaleisiin.

Saapumisajankohta kertoo, onko varauksen teos saapunut noudettavaksi, ja jos on, milloin. Sen perusteella määritetään, milloin asiakkaan on viimeistään noudettava varaus. Jos varausta ei noudeta ajoissa, se siirtyy seuraavan jonottajan noudettavaksi, tai jos varauksia ei enää ole, vapautuu vapaasti lainattavaksi ja palautuu tarvittaessa kotitoimipisteeseen.

Teoksien kappaleita voidaan joutua kuljettamaan toimipisteiden välillä, jos kappale palautetaan muualle kuin sen kotitoimipisteeseen, tai jos kappale varataan noudettavaksi jostakin muusta toimipisteestä. Kuljetuksia kahden toimipisteen välillä kuvataan **Kuljetus**-relaatiolla, jonka monikko vastaa yksittäistä kuljetusta. Kuljetuksella on yksiselitteinen tunniste, joka voi olla esimerkiksi juokseva numerointi, ja lähtöaika.

Jos jokin kappale pitää kuljettaa toiseen toimipisteeseen, voidaan **Kuljetus**-relaatiosta selvittää, onko jo olemassa sopivaa kuljetusta, joka ei ole lähtenyt vielä. Tällöin kappale voidaan laittaa kyseisen kuljetuksen kyytiin, muussa tapauksessa järjestetään uusi kuljetus.

Kun asiakas lainaa teoksen kappaleen, se lisätään **Lainassa**-relaatioon ja poistetaan **Toimipisteessa**-relaatiosta. Relaatioon tallennetaan lainauspäivämäärä ja lainan erääntymispäivämäärä, joka voidaan määrittää esimerkiksi kappaleen maksimilainausajan perusteella.

Kun asiakas palauttaa teoksen, se poistetaan **Lainassa**-relaatiosta ja lisätään asianmukaiseen **Toimipisteessa**-relaatioon. Palautustapahtumaa kuvaava monikko lisätään **Palautus**-relaatioon, joka toimii kirjaston palautuslokina.

Uusia toimipisteitä, asiakkaita, teoksia ja kappaleita voidaan lisätä yksinkertaisesti lisäämällä vastaaviin relaatioihin uusi monikko.

Ne toimipisteet, joissa on vapaana teoksen kappale, voidaan selvittää et-

simällä kullekin toimipisteelle **Toimipisteessa**-relaatiosta teoksen standarditunnusta vastaavat monikot ja ottamalla **Teos**-relaatiosta niitä vastaavat monikot, jotka kuvaavat toimipisteessä olevia kappaleita. Kappaleissa voi olla kuitenkin vielä varauksia, joten poistetaan monikoista ne kappaleet, joiden standarditunnus ja kappaletunnus esiintyy jossakin **Varaus**-relaation monikossa.

Käyttäjällä lainassa olevat teokset ja niiden erääntymispäivät voi kysyä suoraan **Lainassa**-relaatiosta.

Tietyn kappaleen edelliset lainaajat voi selvittää **Palautus**-relaatiosta teoksen standarditunnuksen ja kappaletunnuksen perusteella järjestämällä monikot palautusajankohdan mukaan laskevaan järjestykseen ja ottamalla vastaavat asiakasnumerot.

Teoksia voidaan hakea erilaisten hakuehtojen mukaan kohdistamalla haun **Teos**-relaatioon sekä **Kirja**, **Lehti**, **CD** ja **DVD**-relaatioihin.

Kun teoksen kappale palautetaan johonkin toimipisteeseen, tarkistetaan ensin **Varaus**-relaatiosta, kohdistuuko teokseen varauksia. Jos kohdistuu, etsitään aikaisin varaus, joka odottaa kappaleen vapautumista eli jolle ei ole vielä määritelty kappaleen tunnusta, ja yhdistetään palautettu kappale varaukseen. Jos kappale on määritelty varauksessa noudettavaksi toisesta toimipisteestä, järjestetään kappaleelle kuljetus. Jos teokseen ei kohdistu varauksia, mutta se on väärässä toimipisteessä, kuljetetaan se takaisin kotitoimipisteeseen.

Myöhästyneet lainat voidaan selvittää **Lainassa**-relaatiosta etsimällä ne lainat, joiden erääntymisaika on mennyt. Teoksen lainannut asiakas saadaan selville samasta relaatiosta.

Asiakkaan maksamatta olevat maksut voi selvittää **Maksut**-relaatiosta etsimällä ne **Maksu**-relaation monikot, joita ei ole vielä merkitty maksetuiksi.

4 Funktionaaliset riippuvuudet

Teoksen standarditunnus yksilöi teoksen, ja kaikilla tietyn teoksen kappaleilla on samoja ominaisuuksia:

- Relaatiossa **Teos** *standardiTunnus* \rightarrow *nimi julkaisuvuosi kieli genre*.
- Relaatiossa **Kirja** *standardiTunnus* \rightarrow *tekija kustantaja sivumaara*.
- Relaatiossa **Lehti** *standardiTunnus* \rightarrow *julkaisija vuosikerta numero*.

- Relaatiossa CD $standardiTunnus \rightarrow artisti\ kappaleMaara\ levyYhtio$.
- Relaatiossa DVD $standardiTunnus \rightarrow julkaisija\ kesto$.

Lisäksi kaikissa relaatioissa avainattribuutit yhdessä määräävät muut attribuutit.

5 Anomaliat

Kun teoksen sijainti muuttuu, eli esimerkiksi teos lainataan, palautetaan tai kuljetetaan toiseen toimipisteeseen, pitää huolehtia, että relaatiot Toimipisteessa, Lainassa, Kuljetettavana ja Varaus pysyvät ajan tasalla.

Jos toimipisteen nimi vaihtuu, joudutaan päivittämään myös relaatiot Paatepisteet, Kuljetettavana, Kotitoimipiste, Sijaitsee ja Noutopaikka.

6 Boyce-Codd -normaalimuoto

Tietokanta on Boyce-Codd -normaalimuodossa lukuunottamatta relaatioita Teos, Kirja, Lehti, CD ja DVD. Ositetaan nämä relaatiot.

Relaatiossa Teos(standardiTunnus, kappaleTunnus, nimi, varattavissa, lainattavissa, maxLainausaika, julkaisuvuosi, genre, kieli) on voimassa riippuvuus $standardiTunnus \rightarrow nimi\ julkaisuvuosi\ genre\ kieli$, mutta $\{standardiTunnus\}^+ = \{standardiTunnus, nimi, julkaisuvuosi, genre, kieli\}$.

Ositetaan relaatio. Relaatioon Teos2 tulee $\{standardiTunnus\}^+ :n$ attribuutit ja relaatioon Teos3 tulee $standardiTunnus$ ja relaation Teos loput attribuutit.

Relaatiot ovat siis Teos2(standardiTunnus, nimi, julkaisuvuosi, genre, kieli) ja Teos3(standardiTunnus, kappaleTunnus, varattavissa, lainattavissa, maxLainausAika).

Teos2:ssa on voimassa vain riippuvuus $standardiTunnus \rightarrow nimi\ julkaisuvuosi\ genre\ kieli$, ja $\{Teos2\}^+ = \{standardiTunnus, nimi, julkaisuvuosi, genre, kieli\}$, joten Teos2 on BCNF:ssä.

Teos3:ssa on voimassa vain riippuvuus $standardiTunnus\ kappaleTunnus \rightarrow varattavissa\ lainattavissa\ maxLainausAika$, ja $\{standardiTunnus, kappaleTunnus\}^+ = \{standardiTunnus, kappaleTunnus, varattavissa, lainattavissa, maxLainausAika\}$, joten Teos3 on BCNF:ssä.

Täysin vastaavalla päättelyllä voidaan osoittaa relaatiot Kirja, Lehti, CD ja DVD aiemmin mainittujen funktionaalisten riippuvuuksien perusteella.

Relaation Kirja(standardiTunnus, kappaleTunnus, tekija, kustantaja, sivumaara) ositus BCNF:ään on Kirja2(standardiTunnus, tekija, kustantaja, sivumaara) ja Kirja3(standardiTunnus, kappaleTunnus).

Relaation Lehti(standardiTunnus, kappaleTunnus, julkaisija, vuosikerta, numero) ositus BCNF:ään on Lehti2(standardiTunnus, julkaisija, vuosikerta, numero) ja Lehti3(standardiTunnus, kappaleTunnus).

Relaation CD(standardiTunnus, kappaleTunnus, artisti, kappaleMaara, levyYhtio) ositus BCNF:ään on CD2(standardiTunnus, artisti, kappaleMaara, levyYhtio) ja CD3(standardiTunnus, kappaleTunnus).

Relaation DVD(standardiTunnus, kappaleTunnus, julkaisija, kesto) ositus BCNF:ään on DVD2(standardiTunnus, julkaisija, kesto) ja DVD3(standardiTunnus, kappaleTunnus).

7 1. osan palautuksen jälkeen tehdyt muutokset

7.1 Relaatiokaavio

1. osan relaatiokaaviota on päivitetty.

Poistetaan tarpeettomia monesta yhteen -assosiaatioista muodostettuja relaatioita. Poistetaan relaatiot Paatepisteet, Maksut, Varannut, Noutopaikka ja Kotitoimipiste ja yhdistetään niiden tiedot vastaaviin monesta-puolen relaatioihin.

Muutetaan Palautus-relaation avainta: asiakasnumeron ei tarvitse olla avaimena, koska tietty teoksen kappale voidaan palauttaa vain kerran tietynä ajanhetkenä.

Erotetaan teoksen yleiset ja kappalekohtaiset tiedot kahdeksi eri relaatioksi Teos ja Kappale. Relaatio on tällöin BCNF:ssä.

Poistetaan kappaleTunnus relaatioista Kirja, Lehti, CD ja DVD. Tällöin relaatiot ovat BCNF:ssä.

8 Tietokannan luominen

8.1 Taulujen luominen

```
1  /*
2      Standarditunnus on tällä hetkellä 10- tai 13-
        merkinen ISBN-merkkijono.
3      Kenttään on jätetty laajennusvaraa mahdollisille
        tuleville tunnisteille.
4  */
5  CREATE TABLE Teos (
6      standardiTunnus VARCHAR(32) NOT NULL,
7      nimi VARCHAR(256),
8      julkaisuvuosi INT,
9      genre VARCHAR(256),
10     kieli VARCHAR(256),
11     PRIMARY KEY (standardiTunnus)
12 );
13
14 /*
15     Standarditunnus vastaa Teos-taulun standarditunnusta.
16     KappaleTunnus yksilöi teoksen kappaleen ja voi olla
        esimerkiksi juokseva
17     numerointi.
18     MaxLainausaika on ajanjakso, esimerkiksi '1 month'.
19     Kotitoimipiste on toimipisteen nimi.
20
21     Teoksen ja kotitoimipisteen, joihin kappale viittaa,
        on oltava olemassa.
22 */
23 CREATE TABLE Kappale (
24     standardiTunnus VARCHAR(32),
25     kappaleTunnus INT NOT NULL,
26     varattavissa BOOLEAN,
27     lainattavissa BOOLEAN,
28     maxLainausaika DATE NOT NULL,
29     kotitoimipiste VARCHAR(256),
30     PRIMARY KEY (standardiTunnus, kappaleTunnus),
31     FOREIGN KEY (standardiTunnus) REFERENCES Teos(
        standardiTunnus),
32     FOREIGN KEY (kotitoimipiste) REFERENCES Toimipiste(
        nimi)
33 );
34
35 /*
```



```

36      Kirjan sivumäärä ei voi olla negatiivinen.
37  */
38  CREATE TABLE Kirja (
39      standardiTunnus VARCHAR(32) NOT NULL,
40      tekija VARCHAR(256),
41      kustantaja VARCHAR(256),
42      sivumaara INT CHECK (sivumaara >= 0),
43      PRIMARY KEY (standardiTunnus),
44      FOREIGN KEY (standardiTunnus) REFERENCES Teos(
          standardiTunnus)
45 );
46
47  /*
48      Vuosikerta on juokseva numerointi.
49      Lehden numero on merkkijono, koska lehdet voivat
          julkaista erikoisnumeroita.
50  */
51  CREATE TABLE Lehti (
52      standardiTunnus VARCHAR(32) NOT NULL,
53      julkaisija VARCHAR(256),
54      vuosikerta INT,
55      numero VARCHAR(256),
56      PRIMARY KEY (standardiTunnus),
57      FOREIGN KEY (standardiTunnus) REFERENCES Teos(
          standardiTunnus)
58 );
59
60  /*
61      CD:n kappalemäärä ei voi olla negatiivinen.
62  */
63  CREATE TABLE CD (
64      standardiTunnus VARCHAR(32) NOT NULL,
65      artisti VARCHAR(256),
66      kappaleMaara INT CHECK (kappaleMaara >= 0),
67      levyYhtio VARCHAR(256),
68      PRIMARY KEY (standardiTunnus),
69      FOREIGN KEY (standardiTunnus) REFERENCES Teos(
          standardiTunnus)
70 );
71
72  /*
73      DVD:n kesto on ajanjakso, esim '2 hours'.
74  */
75  CREATE TABLE DVD (
76      standardiTunnus VARCHAR(32) NOT NULL,

```

```

77         julkaisija VARCHAR(256),
78         kesto DATE,
79         PRIMARY KEY (standardiTunnus),
80         FOREIGN KEY (standardiTunnus) REFERENCES Teos(
            standardiTunnus)
81     );
82
83     /*
84         Asiakasnumero voi olla esim. juokseva numerointi.
85     */
86     CREATE TABLE Asiakas (
87         asiakasNro INT NOT NULL,
88         nimi VARCHAR(256),
89         osoite VARCHAR(256),
90         email VARCHAR(256),
91         PRIMARY KEY (asiakasNro)
92     );
93
94     /*
95         Kun asiakas tekee varauksen, se kohdistuu teokseen,
96         muttei vielä
97         mihinkään kappaleeseen, joten kappaleTunnus on NULL.
98         Saapumisajankohtaa
99         ei myöskään tiedetä vielä, joten sekin on NULL.
100
101         Kun varatun teoksen kappale saapuu johonkin
102         toimipisteeseen, yhdistetään varaus
103         kyseiseen kappaleeseen ja kappaleTunnus ja
104         saapumisAjankohta saavat arvonsa.
105     */
106     CREATE TABLE Varaus (
107         tunniste INT NOT NULL,
108         teosStandardiTunnus VARCHAR(32),
109         teosKappaleTunnus INT DEFAULT NULL,
110         varausAjankohta DATE,
111         saapumisAjankohta DATE DEFAULT NULL,
112         varaajaAsiakasNro INT,
113         noutoToimipiste VARCHAR(256),
114         PRIMARY KEY (tunniste),
115         FOREIGN KEY (teosStandardiTunnus, teosKappaleTunnus)
            REFERENCES Kappale(standardiTunnus, kappaleTunnus)
            ,
116         FOREIGN KEY (varaajaAsiakasNro) REFERENCES Asiakas(
            asiakasNro),

```

```

113         FOREIGN KEY (noutoToimipiste) REFERENCES Toimipiste(
114             nimi)
115     );
116     /*
117     Asiakasnumeron ei tarvitse olla avainattribuuttina,
118     koska kaksi asiakasta ei
119     voi palauttaa samaa kappaletta samalla hetkellä. Sama
120     asiakas voi palauttaa saman
121     teoksen useasti, koska palautusAjankohta on
122     avainattribuutti.
123
124     Taulu säilyttää lainaushistorian, eikä sen monikkoja
125     ole normaalisti tarkoitus poistaa.
126     */
127     CREATE TABLE Palautus (
128         standardiTunnus VARCHAR(32),
129         kappaleTunnus INT,
130         palautusAjankohta DATE,
131         asiakasNro INT,
132         PRIMARY KEY (standardiTunnus, kappaleTunnus,
133             palautusAjankohta),
134         FOREIGN KEY (standardiTunnus, kappaleTunnus)
135             REFERENCES Kappale(standardiTunnus, kappaleTunnus)
136         ,
137         FOREIGN KEY (asiakasNro) REFERENCES Asiakas(
138             asiakasNro)
139     );
140     /*
141     Maksun tunniste voi olla esimerkiksi juokseva
142     numerointi.
143     Summa annetaan sentteinä.
144     */
145     CREATE TABLE Maksu (
146         tunniste INT NOT NULL,
147         summa INT,
148         tyyppi VARCHAR(256),
149         maksettu BOOLEAN DEFAULT FALSE,
150         asiakasNro INT,
151         PRIMARY KEY (tunniste),
152         FOREIGN KEY (asiakasNro) REFERENCES Asiakas(
153             asiakasNro)
154     );

```

```

147 CREATE TABLE Toimipiste (
148     nimi VARCHAR(256) NOT NULL,
149     osoite VARCHAR(256),
150     PRIMARY KEY (nimi)
151 );
152
153 /*
154     KuljetusID voi olla esim. juokseva numerointi.
155 */
156 CREATE TABLE Kuljetus (
157     kuljetusID INT NOT NULL,
158     lahtoaika DATE,
159     lahtoToimipiste VARCHAR(256),
160     paateToimipiste VARCHAR(256),
161     PRIMARY KEY (kuljetusID),
162     FOREIGN KEY (lahtoToimipiste) REFERENCES Toimipiste(
        nimi),
163     FOREIGN KEY (paateToimipiste) REFERENCES Toimipiste(
        nimi)
164 );
165
166 /*
167     Taulu kertoo, mitkä kappaleet ovat tällä hetkellä
        lainassa. Kun kappale palautetaan, poistetaan se
        taulusta. Asiakas voi lainata saman teoksen
        useasti, koska tieto aiemmasta lainauksesta
        poistetaan taulusta, kun laina palautetaan.
        Lainaushistoria säilyy Palautus-tilussa.
168
169     Laina ei voi erääntyä ennen sen alkamista.
170
171 */
172 CREATE TABLE Lainassa (
173     standardiTunnus VARCHAR(32),
174     kappaleTunnus INT,
175     lainausAika DATE NOT NULL,
176     eraantymisAika DATE NOT NULL CHECK (lainausAika <=
        eraantymisAika),
177     asiakasNro INT,
178     PRIMARY KEY (standardiTunnus, kappaleTunnus),
179     FOREIGN KEY (standardiTunnus, kappaleTunnus)
        REFERENCES Kappale(standardiTunnus, kappaleTunnus)
        ,
180     FOREIGN KEY (asiakasNro) REFERENCES Asiakas(
        asiakasNro)
181 );

```

```

182
183  /*
184      Taulu kertoo, mitkä kappaleet ovat tällä hetkellä
          missäkin toimipisteessä. Kun kappale poistuu
          toimipisteestä, poistetaan monikko taulusta.
185  */
186  CREATE TABLE Toimipisteessa (
187      standardiTunnus VARCHAR(32),
188      kappaleTunnus INT,
189      toimipisteNimi VARCHAR(256),
190      PRIMARY KEY (standardiTunnus, kappaleTunnus),
191      FOREIGN KEY (standardiTunnus, kappaleTunnus)
          REFERENCES Kappale(standardiTunnus, kappaleTunnus)
          ,
192      FOREIGN KEY (toimipisteNimi) REFERENCES Toimipiste(
          nimi)
193  );
194
195  /*
196      Taulu kertoo, mitkä kappaleet ovat tällä hetkellä
          kuljetettavana ja
197      minkä kuljetuksen kyydissä ne ovat. Kun kappale
          poistuu kuljetuksesta,
          poistetaan monikko taulusta.
198  */
199
200  CREATE TABLE Kuljetettavana (
201      standardiTunnus VARCHAR(32),
202      kappaleTunnus INT,
203      kuljetusID INT,
204      PRIMARY KEY (standardiTunnus, kappaleTunnus),
205      FOREIGN KEY (standardiTunnus, kappaleTunnus)
          REFERENCES Kappale(standardiTunnus, kappaleTunnus)
          ,
206      FOREIGN KEY (kuljetusID) REFERENCES Kuljetus(
          kuljetusID)
207  );

```

8.2 Alkuperäinen data

```

1  /*
2      Tietokantaan täytyy pystyä luomaan uusia toimipisteit
          ä.
3      Lisätään uusia monikkoja Toimipiste-relaatioon.
4  */

```

```

5
6 INSERT INTO Toimipiste
7 VALUES ('Oodi', 'Töölönlahdenkatu_4');
8
9 INSERT INTO Toimipiste
10 VALUES ('Kallio', 'Kalliontie_3');
11
12 INSERT INTO Toimipiste
13 VALUES ('Itäskeskus', 'Itäkatu_4');
14
15 INSERT INTO Toimipiste
16 VALUES ('Vuosaari', 'Vuotie_14');
17
18 /*
19      Tietokantaan täytyy pystyä lisäämään uusia asiakkaita
20      .
21      Lisätään uusi monikko Asiakas-relaatioon.
22 */
23 INSERT INTO Asiakas
24 VALUES (982, 'Tero_Teekkari', 'Maarintie_3', 'teetero@hotmail
    .com');
25
26 INSERT INTO Asiakas
27 VALUES (111, 'Minni_Meikäläinen', 'Vuotie_22', 'minni.
    meikalainen@hotmail.com');
28
29 INSERT INTO Asiakas
30 VALUES (123, 'Matti_Meikäläinen', 'Otakaari_20', 'matti.
    meikalainen@gmail.com');
31
32 /*
33      Tietokantaan täytyy pystyä lisäämään erilaisia
34      teoksia ja niiden kappaleita.
35      Lisätään uusi kirja ja yksittäinen kirjan kappale.
36
37      Ensin lisätään kirjan yleiset tiedot Teos-relaatioon,
38      jonka jälkeen lisätään yksittäisen kappaleen
39      tiedot Kappale-relaatioon.
40      Koska teos on kirja, täytyy teoksen tyypille
41      ominaiset tiedot vielä lisätä Kirja-relaatioon.
42      Tämän jälkeen asetetaan kappale johonkin
43      toimipisteeseen lisäämällä se Toimipisteessa-
44      relaatioon.
45 */

```

```

40
41 -- Lisätään teos Kalevala
42
43 INSERT INTO Teos
44 VALUES ('978-951-1-23676-4', 'Kalevala', 2009, '
    Kaunokirjallisuus', 'suomi');
45
46 INSERT INTO Kirja
47 VALUES ('978-951-1-23676-4', 'Elias_Lönnrot', 'Otava', 475);
48
49 -- Lisätään teos Matrix Algebra
50
51 INSERT INTO Teos
52 VALUES ('111-951-1-23676-4', 'Matrix_Algebra', 2019, '
    Tietokirjallisuus', 'englanti');
53
54 INSERT INTO Kirja
55 VALUES ('111-951-1-23676-4', 'Pekka_Professori', 'Sanoma',
    4175);
56
57 -- Lisätään kolme kappaletta Kalevalaa
58
59 INSERT INTO Kappale
60 VALUES ('978-951-1-23676-4', 0, TRUE, TRUE, '+30_days', 'Oodi
    ');
61
62 INSERT INTO Toimipisteessa
63 VALUES ('978-951-1-23676-4', 0, 'Oodi');
64
65 -- Kappale 1 on kuljetuksessa Vuosaaresta Kallioon
66 INSERT INTO Kappale
67 VALUES ('978-951-1-23676-4', 1, TRUE, TRUE, '+30_days', 'Oodi
    ');
68
69 INSERT INTO Kuljetus
70 VALUES (456, '2019-05-12T11:30:00', 'Vuosaari', 'Kallio');
71
72 INSERT INTO Kuljetettavana
73 VALUES ('978-951-1-23676-4', 1, 456);
74
75 INSERT INTO Kappale
76 VALUES ('978-951-1-23676-4', 2, TRUE, TRUE, '+14_days', 'Oodi
    ');
77
78 INSERT INTO Toimipisteessa

```

```

79 VALUES ('978-951-1-23676-4', 2, 'Vuosaari');
80
81
82 -- Lisätään kaksi kappaletta Matrix Algebraa
83 INSERT INTO Kappale
84 VALUES ('111-951-1-23676-4', 0, TRUE, TRUE, '+30_days', '
    Vuosaari');
85
86 INSERT INTO Toimipisteessa
87 VALUES ('111-951-1-23676-4', 0, 'Vuosaari');
88
89 INSERT INTO Kappale
90 VALUES ('111-951-1-23676-4', 1, TRUE, TRUE, '+30_days', '
    Vuosaari');
91
92 -- Toinen kappale on lainassa Tero Teekkarilla ja laina on my
    öhässä
93 INSERT INTO Lainassa
94 VALUES ('111-951-1-23676-4', 1, '2017-11-07', '2018-01-21',
    982);
95
96 /*
97     Tietokantaan voidaan lisätä uusia maksuja asiakkaille
        . Tälläisiä voivat olla esimerkiksi myöhästymis-
        ja varausmaksut.
98     Lisätään uusi monikko Maksu-relaatioon.
99
100     Lisätään Matti Meikäläiselle 3.50e suuruinen myöhä
        stymismaksu ja 1.00e suuruinen varausmaksu.
101 */
102 INSERT INTO Maksu (tunniste, summa, tyyppi, asiakasNro)
103 VALUES (3456, 350, 'myohastyminen', 123);
104
105 INSERT INTO Maksu (tunniste, summa, tyyppi, asiakasNro)
106 VALUES (3457, 100, 'varaus', 123);
107
108 INSERT INTO Maksu (tunniste, summa, tyyppi, asiakasNro)
109 VALUES (3458, 350, 'myohastyminen', 111);
110
111 -- Lisätään tietokantaan myös maksuja, jotka on jo maksettu
112 INSERT INTO Maksu
113 VALUES (3459, 250, 'myohastyminen', TRUE, 111);
114
115 INSERT INTO Maksu
116 VALUES (3460, 550, 'myohastyminen', TRUE, 123);

```



```

117
118 INSERT INTO Maksu
119 VALUES (3461, 150, 'varaus', TRUE, 982);
120
121 INSERT INTO Maksu
122 VALUES (3462, 250, 'myohastyminen', TRUE, 982);

```

9 Hakemistot ja näkymät

```

1  /*
2      Luodaan näkymä, joka yhdistää yksittäisen kappaleen
3      tiedot
4      teoksen yleisiin tietoihin.
5  */
6  CREATE VIEW KappaleTiedot AS
7      SELECT *
8      FROM Teos NATURAL JOIN Kappale
9
10 /*
11     Luodaan hakemistot eri teostyyppien
12     standarditunnuksien perusteella, koska
13     tauluista haetaan usein yksittäisen teoksen tietoja
14     standarditunnuksen
15     avulla.
16 */
17 CREATE INDEX TeosIndex ON Teos(standardiTunnus);
18 CREATE INDEX KirjaIndex ON Kirja(standardiTunnus);
19 CREATE INDEX LehtiIndex ON Lehti(standardiTunnus);
20 CREATE INDEX CDIndex ON CD(standardiTunnus);
21 CREATE INDEX DVDIndex ON DVD(standardiTunnus);
22
23 /*
24     Luodaan hakemisto kappaleen standarditunnuksen ja
25     kappaletunnuksen perusteella,
26     koska taulusta haetaan usein yksittäisen kappaleen
27     tietoja (standardiTunnus,
28     kappaleTunnus) -parin perusteella.
29 */
30 CREATE INDEX KappaleIndex ON Kappale(standardiTunnus,
31     kappaleTunnus);
32
33 /*

```

```

29      Luodaan hakemistot standarditunnuksen ja
30      kappaletunnuksen perusteella, jotta
31      kappaleen olinpaikka voidaan selvittää tehokkaasti.
32  */
33  CREATE INDEX ToimipisteessaIndex ON Toimipisteessa(
34      standardiTunnus, kappaleTunnus);
35  CREATE INDEX LainassaIndex ON Lainassa(standardiTunnus,
36      kappaleTunnus);
37  CREATE INDEX KuljetettavanaIndex ON Kuljetettavana(
38      standardiTunnus, kappaleTunnus);
39
40  /*
41      Luodaan hakemisto asiakkaan asiakasnumeron
42      perusteella, koska yksittäisen asiakkaan
43      henkilötietoja kysytään usein.
44  */
45  CREATE INDEX AsiakasIndex ON Asiakas(asiakasNro);

```

10 Käyttötapaukset

```

1  /*
2      Kirjastosta voidaan lainata yksittäisiä teoksen
3      kappaleita ja tapahtuma on rekisteröitävä
4      tietokannan tarvittaviin relaatioihin.
5      Asiakas lainaa teoksen kappaleen toimipisteestä,
6      jolloin Lainassa-relaatioon lisätään uusi monikko,
7      joka sisältää lainauksen tiedot.
8      Oletetaan, että kappale on toimipisteessä ja asiakas
9      voi lainata sen, eli
10     se on lainattavissa eikä ole varattuna kenellekään
11     muulle.
12     Lainaajaksi asetetaan lainaushetki ja erää
13     ntymisajaksi
14     lainaushetki + kappaleen max lainausaika.
15     Koska lainaushetkellä asiakas ottaa teoksen jostakin
16     toimipisteestä,
17     poistetaan tämän jälkeen kappale Toimipisteessa-
18     relaatiosta, jossa se oli.
19     Poistetaan myös käyttäjän mahdolliset teokseen
20     kohdistuneet varaukset Varaus-relaatiosta.
21
22     Matti Meikäläinen lainaa yhden kappaleen Kalevalaa.

```

```

13 */
14 INSERT INTO Lainassa
15 VALUES ('978-951-1-23676-4', 0, datetime('now'), datetime('
    now',
16         (SELECT maxLainausaika
17         FROM Kappale
18         WHERE standardiTunnus = '978-951-1-23676-4' AND
            kappaleTunnus = 0))
19 , 123);
20
21 DELETE FROM Toimipisteessa
22 WHERE standardiTunnus = '978-951-1-23676-4' AND kappaleTunnus
    = 0;
23
24 DELETE FROM Varaus
25 WHERE teosStandardiTunnus = '978-951-1-23676-4' AND
    varaajaAsiakasNro = 123;
26
27 /*
28     Halutaan selvittää, mitä teoksia yksittäisellä
        asiakkaalla on lainassa tällä hetkellä ja mitkä
        ovat lainojen erääntymispäivät
29     Tämä tapahtuu tekemällä kysely Lainassa- ja Teos -
        relaatioiden luonnolliseen liitokseen ja
        suodattamalla tuloksesta vain halutun
        asiakasnumeron omaavat monikot.
30
31     Kysytään, mitä lainoja Matti Meikäläisellä on tällä
        hetkellä.
32 */
33 SELECT standardiTunnus, nimi, eraantymisAika
34 FROM Lainassa NATURAL JOIN Teos
35 WHERE asiakasNro = 123;
36
37 /*
38     Lainattuja teoksia on voitava palauttaa, jolloin
        palautustapahtuman tiedot kirjataan tietokantaan.
39     Palautushistoria tallentuu Palautus-relaatioon, sillä
        sen sisältämiä monikoita ei poisteta.
40     Asiakas palauttaa teoksen kappaleen johonkin
        toimipisteeseen, jolloin kappale lisätään takaisin
        Toimipisteessa-relaatioon.
41     Tämän jälkeen tallennetaan tieto palautustapahtumasta
        Palautus-relaatioon ja poistetaan kappale
        Lainassa-relaatiosta.

```

```

42
43         Matti Meikäläinen palauttaa lainatun Kalevalan
           kappaleen Oodiin.
44 */
45 INSERT INTO Toimipisteessa
46 VALUES ('978-951-1-23676-4', 0, 'Oodi');
47
48 INSERT INTO Palautus
49 VALUES ('978-951-1-23676-4', 0, datetime('now'), (SELECT
           asiakasNro
50         FROM Lainassa
51         WHERE standardiTunnus = '978-951-1-23676-4' AND
           kappaleTunnus = 0));
52
53 DELETE FROM Lainassa
54 WHERE standardiTunnus = '978-951-1-23676-4' AND kappaleTunnus
           = 0;
55
56 /*
57         Kirjaston tietokannasta halutaan selvittää, missä
           toimipisteissä on saatavana tietyn teoksen kappale
58         .
           Tämä tapahtuu kysymällä Toimipisteessa-relaatiosta
           niitä monikoita, joiden standarditunnus täsmää
           halutun teoksen standarditunnukseen.
59
60         Etsitään kaikki toimipisteet, joissa on vapaana
           Kalevala.
61 */
62 SELECT DISTINCT toimipisteNimi
63 FROM Toimipisteessa
64 WHERE standardiTunnus = '978-951-1-23676-4';
65
66 /*
67         Kirjaston on voitava selvittää aikajärjestyksessä,
           kenellä teoksen kappale on aiemmin ollut lainassa
           ja milloin se on palautettu.
68         Tehdään kysely Palautus- ja Asiakas -relaatioiden
           luonnolliseen liitokseen ja suodatetaan tuloksesta
           vain haluttua teoksen kappaletta koskevat monikot
69         .
           Tulos järjestetään vielä lopuksi aikajärjestykseen,
           uusin palautus ensin.
70
71         Kysytään Kalevalan palautushistoria.

```

```

72 */
73 SELECT asiakasNro, nimi, osoite, email, palautusAjankohta
74 FROM Palautus NATURAL JOIN Asiakas
75 WHERE standardiTunnus = '978-951-1-23676-4' AND kappaleTunnus
    = 0
76 ORDER BY palautusAjankohta DESC;
77
78 /*
79         Kirjaston teoksista halutaan tehdä varauksia, jotta
            asiakas voi noutaa teoksen lähimmästä
            toimipisteestään.
80         Tehdään uusi teokseen kohdistuva varaus lisäämällä
            uusi monikko Varaus-relaatioon.
81
82         Matti Meikäläinen varaa kappaleen Kalevalaa
            toimitettavaksi Oodiin.
83 */
84 INSERT INTO Varaus (tunniste, teosStandardiTunnus,
            varausAjankohta, varaajaAsiakasNro, noutoToimipiste)
85 VALUES (12345678, '978-951-1-23676-4', datetime('now'), 123,
            'Oodi');
86
87 /*
88         Halutaan selvittää, mitkä lainat ovat myöhässä ja
            kysytään lainaajien henkilötiedot.
89         Tehdään Lainassa-, Asiakas- ja Teos- relaatioiden
            luonnolliseen liitokseen kysely, josta suodatetaan
            näytettäväksi vain ne lainat, joiden erää
            ntymisaika on ennen nykyhetkeä.
90         Lasketaan myös, montako päivää laina on myöhässä
            viimeisestä palautuspäivästä ja näytetään tieto
            omana sarakkeena.
91 */
92 SELECT asiakasNro, Asiakas.nimi AS asiakasNimi, osoite, email
            , standardiTunnus, Teos.nimi AS teosNimi, eraantymisAika,
            julianday(datetime('now'))-julianday(eraantymisAika) AS
            pvMyohassa
93 FROM (Lainassa NATURAL JOIN Asiakas) JOIN Teos ON Lainassa.
            standardiTunnus = Teos.standardiTunnus
94 WHERE eraantymisAika < datetime('now');
95
96 /*
97         Tietokantaan voidaan lisätä uusia maksuja asiakkaille
            . Tälläisiä voivat olla esimerkiksi myöhästymis-
            ja varausmaksut.

```

```

98         Lisätään uusi monikko Maksu-relaatioon.
99
100        Lisätään Matti Meikäläiselle 4.00e suuruinen myöhä
           stymismaksu.
101    */
102    INSERT INTO Maksu (tunniste, summa, tyyppi, asiakasNro)
103    VALUES (9876, 400, 'myohastyminen', 123);
104
105    /*
106        Tietokannasta voidaan selvittää asiakkaan maksamatta
           olevat maksut
107        Tämä tapahtuu tekemällä Maksu-relaatioon kysely,
           jossa ehtona on, että maksua ei ole maksettu ja
           monikon asiakasnumero täsmää.
108
109        Kysytään, mitä maksuja Matti Meikäläisellä on
           maksamatta.
110    */
111    SELECT summa, tyyppi
112    FROM Maksu
113    WHERE asiakasNro = 123 AND maksettu = FALSE;
114
115    /*
116        Halutaan selvittää asiakkaat, joilla on varauksia
           tiettyyn teokseen.
117        Tämä voidaan selvittää tekemällä kysely Varaus- ja
           Asiakas- relaatioiden liitokseen määrittämällä
           ehdoksi, että teoksen standarditunnuksen on
118        täsmättävä haettuun teokseen.
119        Järjestetään lopuksi monikot varausajankohdan mukaan.
120
121        Kysytään, ketkä ovat varanneet Kalevalan.
122    */
123    SELECT asiakasNro, nimi, varausAjankohta
124    FROM Varaus JOIN Asiakas ON varaajaAsiakasNro = asiakasNro
125    WHERE teosStandardiTunnus = '978-951-1-23676-4'
126    ORDER BY varausAjankohta ASC;
127
128    /*
129        Selvitetään, missä toimipisteissä teoksen kappaleita
           on ja kuinka monta.
130        Tehdään kysely Toimipisteessa-relaatioon ja määritetä
           än kyselyn ehdoksi, että teoksen
           standarditunnuksen on täsmättävä haetun teoksen
           kanssa.

```

```

131      Ryhmitellään monikot toimipisteen nimen mukaan, jotta
      saadaa selville haluttu lukumäärä.
132
133      Kysytään, kuinka monta kappaletta Kalevalaa on
      kussakin toimipisteessä.
134  */
135  SELECT toimipisteNimi, COUNT(*) AS lukumaara
136  FROM Toimipisteessa
137  WHERE standardiTunnus = '978-951-1-23676-4'
138  GROUP BY toimipisteNimi;
139
140  /*
141      Selvitetään, missä kuljetuksissa tietyn teoksen
      kappaleita on kyydissä.
142      Tehdään kysely Kuljetettavana- ja Kuljetus-
      relaatioiden luonnolliseen liitokseen ja määritell
      ään ehdoksi, että stardarditunnuksen tulee täsmätä
      haetun teoksen kanssa.
143  */
144  SELECT lahtoToimipiste, paateToimipiste, lahtoaika
145  FROM Kuljetettavana NATURAL JOIN Kuljetus
146  WHERE standardiTunnus = '978-951-1-23676-4';
147
148  /*
149      Tietokannasta voidaan hakea kirjoja eri hakuehtojen
      mukaan.
150      Haut voidaan kohdistaa Teos- ja Kirja- relaatioiden
      luonnolliseen liitokseen.
151      Kysytään esimerkiksi, mitä kirjoja Elias Lönnrotilta
      on.
152  */
153  SELECT nimi, julkaisuvuosi
154  FROM Teos NATURAL JOIN Kirja
155  WHERE tekija = 'Elias_Lönnrot';
156
157  /*
158      Teoksen kappaleiden olinpaikan selvitys voidaan
      toteuttaa etsimällä kappaleita Lainassa-,
      Kuljetettavana- ja Toimipisteessa- relaatioista.
159      Kysytään esimerkiksi, mitkä yksittäiset Kalevalan
      kappaleet eivät ole kotitoimipisteessä, vaan
      toisessa toimipisteessä, kuljetuksessa tai
      lainassa.
160  */
161  SELECT kappaleTunnus

```

```

162 FROM Lainassa
163 WHERE standardiTunnus = '978-951-1-23676-4'
164 UNION
165 SELECT kappaleTunnus
166 FROM Kuljetettavana
167 WHERE standardiTunnus = '978-951-1-23676-4'
168 UNION
169 SELECT kappaleTunnus
170 FROM Toimipisteessa
171 WHERE standardiTunnus = '978-951-1-23676-4' AND
    toimipisteNimi != (
172     SELECT kotitoimipiste
173     FROM Kappale
174     WHERE Kappale.standardiTunnus = Toimipisteessa.
        standardiTunnus AND
175         Kappale.kappaleTunnus =
            Toimipisteessa.kappaleTunnus);
176
177
178 /*
179     Selvitetään jokaiselle asiakkaalle viimeisen vuoden
        aikana
180     palautettujen lainojen määrä tekemällä kysely
        Palautus- ja Asiakas- relaatioiden luonnolliseen
        liitokseen ja asettamalla ehdoksi, että
        palautusajankohta on ollut vuoden sisällä
        nykyhetkestä.
181     Ryhmitellään monikot asiakasnumeron mukaan jotta
        saadaan haluttu lukumäärä laskettua, ja järjestetä
        än lopuksi monikot nimen mukaan aakkosjä
        rjestykseen.
182 */
183 SELECT asiakasNro, nimi, COUNT(*) AS palautuksia
184 FROM Palautus NATURAL JOIN Asiakas
185 WHERE palautusAjankohta > (datetime('now', '-1_year'))
186 GROUP BY asiakasNro
187 ORDER BY nimi;
188
189
190 /*
191     Selvitetään jokaiselle asiakkaalle maksettujen
        maksujen kokonaissumma tekemällä kysely Maksu- ja
        Asiakas- relaatioiden luonnolliseen liitokseen ja
        asettamalla ehdoksi, että maksu on maksettu.

```



```

192         Ryhmitellään monikot asiakasnumeron mukaan, jotta
           saadaa haluttu summa laskettua, ja järjestetään
           lopuksi monikot nimen mukaan aakkosjärjestykseen.
193 */
194 SELECT asiakasNro, nimi, SUM(summa) AS maksettuYhteensa
195 FROM Maksu NATURAL JOIN Asiakas
196 WHERE maksettu = TRUE
197 GROUP BY asiakasNro
198 ORDER BY nimi;
199
200 /*
201         Selvitetään viime kuun 10 suosituinta teosta
           palautusten määrän perusteella tekemällä kysely
           Palautus- ja Teos- relaatioiden luonnolliseen
           liitokseen
202         ja asettamalla ehdoksi, että palautusajankohta on
           kuukauden sisällä nykyhetkestä.
203         Ryhmitellään monikot standarditunnuksen mukaan, jotta
           saadaa haluttu lukumäärä laskettua, ja järjestet
           ään lopuksi monikot lukumäärän mukaan laskevaan jä
           rjestykseen.
204         Rajoitetaan haettavien monikkojen määrä kymmeneen.
205 */
206 SELECT nimi, julkaisuvuosi, genre, COUNT(*) AS palautuksia
207 FROM Palautus NATURAL JOIN Teos
208 WHERE palautusAjankohta > (datetime('now', '-1┐month'))
209 GROUP BY standardiTunnus
210 ORDER BY palautuksia DESC
211 LIMIT 10;

```

11 SQLite-ympäristön käskyt

Tietokanta luodaan luomalla ensin taulut `tietokannan_luonti.sql`-tiedoston käskyillä ja sen jälkeen lisäämällä dataa `alkuperainen_data.sql`-tiedoston käskyillä. Kummankin tiedoston kaikkien käskyjen pitäisi onnistua.

Tämän jälkeen `kayttotapaukset.sql`-tiedoston käyttötapauksia voidaan ajaa. Käyttötapaukset kannattaa käydä läpi järjestyksessä. Kaikkien käskyjen pitäisi onnistua ja kyselyjen tuottaa epätyhjä vastaus, jos käyttötapaukset on suoritettu järjestyksessä.