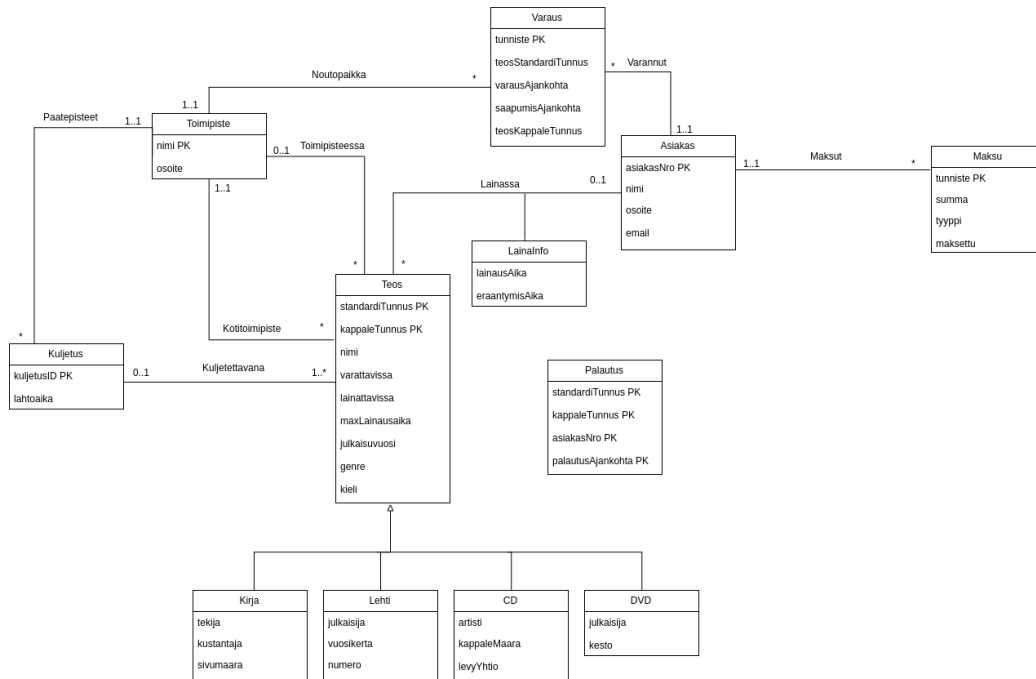


Tietokannat - harjoitustyö (osa 2)

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 määritelty **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 hakuhtojen 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

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         Saapumisajankohta
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),
117         FOREIGN KEY (noutoToimipiste) REFERENCES Toimipiste(
            nimi)
118     );
119

```

```

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  /*
142      Maksun tunniste voi olla esimerkiksi juokseva
143      numerointi.
144      Summa annetaan sentteinä.
145  */
146  CREATE TABLE Maksu (
147      tunniste INT NOT NULL,
148      summa INT,
149      tyyppi VARCHAR(256),
150      maksettu BOOLEAN DEFAULT FALSE,
151      asiakasNro INT,
152      PRIMARY KEY (tunniste),
153      FOREIGN KEY (asiakasNro) REFERENCES Asiakas(
154          asiakasNro)
155  );
156
157  CREATE TABLE Toimipiste (
158      nimi VARCHAR(256) NOT NULL,
159      osoite VARCHAR(256),
160      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
181     FOREIGN KEY (asiakasNro) REFERENCES Asiakas(
        asiakasNro)
182 );
183
184 /*
        Taulu kertoo, mitkä kappaleet ovat tällä hetkellä
        missäkin toimipisteessä. Kun kappale poistuu

```

```

185      toimipisteestä, poistetaan monikko taulusta.
186  */
187  CREATE TABLE Toimipisteessa (
188      standardiTunnus VARCHAR(32),
189      kappaleTunnus INT,
190      toimipisteNimi VARCHAR(256),
191      PRIMARY KEY (standardiTunnus, kappaleTunnus),
192      FOREIGN KEY (standardiTunnus, kappaleTunnus)
193          REFERENCES Kappale(standardiTunnus, kappaleTunnus)
194      ,
195      FOREIGN KEY (toimipisteNimi) REFERENCES Toimipiste(
196          nimi)
197  );
198  /*
199      Taulu kertoo, mitkä kappaleet ovat tällä hetkellä
200      kuljetettavana ja
201      minkä kuljetuksen kyydissä ne ovat. Kun kappale
202      poistuu kuljetuksesta,
203      poistetaan monikko taulusta.
204  */
205  CREATE TABLE Kuljetettavana (
206      standardiTunnus VARCHAR(32),
207      kappaleTunnus INT,
208      kuljetusID INT,
209      PRIMARY KEY (standardiTunnus, kappaleTunnus),
210      FOREIGN KEY (standardiTunnus, kappaleTunnus)
211          REFERENCES Kappale(standardiTunnus, kappaleTunnus)
212      ,
213      FOREIGN KEY (kuljetusID) REFERENCES Kuljetus(
214          kuljetusID)
215  );

```

8.2 Alkuperäinen data

9 Hakemistot ja näkymät

```

1  /*
2      Luodaan näkymä, joka yhdistää yksittäisen kappaleen
      tiedot

```

```

3      teoksen yleisiin tietoihin.
4  */
5  CREATE VIEW KappaleTiedot AS
6      SELECT *
7      FROM Teos NATURAL JOIN Kappale
8
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 /*
34     Luodaan hakemistot standarditunnuksen ja
35     kappaletunnuksen perusteella, jotta
36     kappaleen olinpaikka voidaan selvittää tehokkaasti.
37 */
38 CREATE INDEX ToimipisteessaIndex ON Toimipisteessa(
39     standardiTunnus, kappaleTunnus);
40 CREATE INDEX LainassaIndex ON Lainassa(standardiTunnus,
41     kappaleTunnus);
42 CREATE INDEX KuljetettavanaIndex ON Kuljetettavana(
43     standardiTunnus, kappaleTunnus);
44
45 /*

```

```

38         Luodaan hakemisto asiakkaan asiakasnumeron
39         perusteella, koska yksittäisen asiakkaan
40         henkilötietoja kysytään usein.
41     */
42     CREATE INDEX AsiakasIndex ON Asiakas(asiakasNro);

```

10 Käyttötapaukset

```

1  /*
2      Tietokantaan täytyy pystyä luomaan uusia toimipisteit
3      ä.
4      Lisätään uusi monikko Toimipiste-relaatioon.
5  */
6  INSERT INTO Toimipiste
7  VALUES ('Oodi', 'Töölönlahdenkatu_4');
8
9  /*
10     Tietokantaan täytyy pystyä lisäämään erilaisia
11     teoksia ja niiden kappaleita.
12     Lisätään uusi kirja ja yksittäinen kirjan kappale.
13
14     Ensin lisätään kirjan yleiset tiedot Teos-relaatioon,
15     jonka jälkeen lisätään yksittäisen kappaleen
16     tiedot Kappale-relaatioon.
17     Koska teos on kirja, täytyy teoksen tyypille
18     ominaiset tiedot vielä lisätä Kirja-relaatioon.
19     Tämän jälkeen asetetaan kappale johonkin
20     toimipisteeseen lisäämällä se Toimipisteessa-
21     relaatioon.
22  */
23  INSERT INTO Teos
24  VALUES ('978-951-1-23676-4', 'Kalevala', 2009, '
25      Kaunokirjallisuus', 'suomi');
26
27  INSERT INTO Kirja
28  VALUES ('978-951-1-23676-4', 'Elias_Lönnrot', 'Otava', 475);
29
30  INSERT INTO Kappale
31  VALUES ('978-951-1-23676-4', 0, TRUE, TRUE, '+30_days', 'Oodi
32      ');
33
34  INSERT INTO Toimipisteessa

```

```

27 VALUES ('978-951-1-23676-4', 0, 'Dodi');
28
29
30 /*
31      Tietokantaan täytyy pystyä lisäämään uusia asiakkaita
32      .
33      Lisätään uusi monikko Asiakas-relaatioon.
34 */
35 INSERT INTO Asiakas
36 VALUES (123, 'Matti Meikäläinen', 'Otakaari 20', 'matti.
37      meikalainen@gmail.com');
38
39 /*
40      Kirjastosta voidaan lainata yksittäisiä teoksen
41      kappaleita ja tapahtuma on rekisteröitävä
42      tietokannan tarvittaviin relaatioihin.
43      Asiakas lainaa teoksen kappaleen toimipisteestä,
44      jolloin Lainassa-relaatioon lisätään uusi monikko
45      joka sisältää lainauksen tiedot.
46      Oletetaan, että kappale on toimipisteessä ja asiakas
47      voi lainata sen, eli
48      se on lainattavissa eikä ole varattuna kenellekään
49      muulle.
50      Lainaajaksi asetetaan lainaushetki ja erää
51      ntymisajaksi
52      lainaushetki + kappaleen max lainausaika.
53      Koska lainaushetkellä asiakas ottaa teoksen jostakin
54      toimipisteestä,
55      poistetaan tämän jälkeen kappale toimipisteessä-
56      relaatiosta, jossa se oli.
57      Poistetaan myös käyttäjän mahdolliset teokseen
58      kohdistuneet varaukset Varaus-relaatiosta.
59
60      Matti Meikäläinen lainaa yhden kappaleen Kalevalaa.
61 */
62 INSERT INTO Lainassa
63 VALUES ('978-951-1-23676-4', 0, datetime('now'), datetime('
64      now',
65      (SELECT maxLainausaika
66      FROM Kappale
67      WHERE standardiTunnus = '978-951-1-23676-4' AND
68      kappaleTunnus = 0))
69      , 123);
70
71

```



```

58 DELETE FROM Toimipisteessa
59 WHERE standardiTunnus = '978-951-1-23676-4' AND kappaleTunnus
    = 0;
60
61 DELETE FROM Varaus
62 WHERE teosStandardiTunnus = '978-951-1-23676-4' AND
    varaajaAsiakasNro = 123;
63
64 /*
65     Lainattuja teoksia on voitava palauttaa, jolloin
        palautustapahtuman tiedot kirjataan tietokantaan.
66     Palautushistoria tallentuu Palautus-relaatioon, sillä
        sen sisältämiä monikoita ei poisteta.
67     Asiakas palauttaa teoksen kappaleen johonkin
        toimipisteeseen, jolloin kappale lisätään takaisin
        Toimipisteessa-relaatioon.
68     Tämän jälkeen tallennetaan tieto palautustapahtumasta
        Palautus-relaatioon ja poistetaan kappale
        Lainassa-relaatiosta.
69
70     Matti Meikäläinen palauttaa lainatun Kalevalan
        kappaleen Oodiin.
71 */
72 INSERT INTO Toimipisteessa
73 VALUES ('978-951-1-23676-4', 0, 'Oodi');
74
75 INSERT INTO Palautus
76 VALUES ('978-951-1-23676-4', 0, datetime('now'), (SELECT
    asiakasNro
77     FROM Lainassa
78     WHERE standardiTunnus = '978-951-1-23676-4' AND
        kappaleTunnus = 0));
79
80 DELETE FROM Lainassa
81 WHERE standardiTunnus = '978-951-1-23676-4' AND kappaleTunnus
    = 0;
82
83 /*
84     Kirjaston tietokannasta halutaan selvittää, missä
        toimipisteissä on saatavana tietyn teoksen kappale
85     .
        Tämä tapahtuu kysymällä Toimipisteessa-relaatiosta
        niitä monikoita, joiden standarditunnus täsmää
        halutun teoksen standarditunnukseen.
86

```

```

87         Etsitään kaikki toimipisteet joissa on vapaana
           Kalevala.
88     */
89     SELECT DISTINCT toimipisteNimi
90     FROM Toimipisteessa
91     WHERE standardiTunnus = '978-951-1-23676-4';
92
93     /*
94         Halutaan selvittää, mitä teoksia yksittäisellä
           asiakkaalla on lainassa tällä hetkellä ja mitkä
           ovat lainojen erääntymispäivät
95         Tämä tapahtuu tekemällä kysely Lainassa- ja Teos -
           relaatioiden luonnolliseen liitokseen ja
           suodattamalla tuloksesta vain halutun
           asiakasnumeron omaavat monikot.
96
97         Kysytään, mitä lainoja Matti Meikäläisellä on tällä
           hetkellä.
98     */
99     SELECT standardiTunnus, nimi, eraantymisAika
100    FROM Lainassa NATURAL JOIN Teos
101    WHERE asiakasNro = 123;
102
103    /*
104        Kirjaston on voitava selvittää aikajärjestyksessä,
           kenellä teoksen kappale on aiemmin ollut lainassa
105        ja milloin se on palautettu.
106        Tehdään kysely Palautus- ja Asiakas -relaatioiden
           luonnolliseen liitokseen ja suodatetaan tuloksesta
           vain haluttua teoksen kappaletta koskevat monikot
           .
107        Tulos järjestetään vielä lopuksi aikajärjestykseen,
           uusin palautus ensin.
108
109        Kysytään Kalevalan palautushistoria.
110    */
111    SELECT asiakasNro, nimi, osoite, email, palautusAjankohta
112    FROM Palautus NATURAL JOIN Asiakas
113    WHERE standardiTunnus = '978-951-1-23676-4' AND kappaleTunnus
           = 0
114    ORDER BY palautusAjankohta DESC;
115
116    /*
117        Kirjaston teoksista halutaan tehdä varauksia, jotta
           asiakas voi noutaa teoksen lähimmästä

```

```

118         toimipisteestään.
119         Tehdään uusi teokseen kohdistuva varaus lisäämällä
120         uusi monikko Varaus-relaatioon.
121
122         Matti Meikäläinen varaa kappaleen Kalevalaa
123         toimitettavaksi Oodiin.
124
125     */
126     INSERT INTO Varaus (tunniste, teosStandardiTunnus,
127         varausAjankohta, varaajaAsiakasNro, noutoToimipiste)
128     VALUES (12345678, '978-951-1-23676-4', datetime('now'), 123,
129         'Oodi')
130
131     /*
132         Halutaan selvittää, mitkä lainat ovat myöhässä ja
133         kysytään lainaajien henkilötiedot.
134         Tehdään Lainassa-, Asiakas- ja Teos- relaatioiden
135         luonnolliseen liitokseen kysely, josta suodatetaan
136         näytettäväksi vain ne lainat, joiden erää
137         ntymisaika on ennen nykyhetkeä.
138         Lasketaan myös, montako päivää laina on myöhässä
139         viimeisestä palautuspäivästä ja näytetään tieto
140         omana sarakkeena.
141
142     */
143     SELECT asiakasNro, Asiakas.nimi AS asiakasNimi, osoite, email
144         , standardiTunnus, Teos.nimi AS teosNimi, eraantymisAika,
145         julianday(datetime('now'))-julianday(eraantymisAika) AS
146         pvMyohassa
147     FROM (Lainassa NATURAL JOIN Asiakas) JOIN Teos ON Lainassa.
148         standardiTunnus = Teos.standardiTunnus
149     WHERE eraantymisAika < datetime('now')
150
151     /*
152         Tietokantaan voidaan lisätä uusia maksuja asiakkaille
153         . Tälläisiä voivat olla esimerkiksi myöhästymis-
154         ja varausmaksut.
155         Lisätään uusi monikko Maksu-relaatioon.
156
157         Lisätään Matti Meikäläiselle 3.50e suuruinen myöhä
158         stymismaksu.
159
160     */
161     INSERT INTO Maksu (tunniste, summa, tyyppi, asiakasNro)
162     VALUES (3456, 350, 'myohastyminen', 123)
163
164     /*

```

```

144      Tietokannasta voidaan selvittää asiakkaan maksamatta
      olevat maksut
145      Tämä tapahtuu tekemällä Maksu-relaatioon kysely,
      jossa ehtona on, että maksua ei ole maksettu ja
      monikon asiakasnumero täsmää.

146
147      Kysytään, mitä maksuja Matti Meikäläisellä on
      maksamatta.
148  */
149  SELECT summa, tyyppi
150  FROM Maksu
151  WHERE asiakasNro = 123 AND maksettu = FALSE;
152
153  /*
154      Halutaan selvittää asiakkaat, joilla on varauksia
      tiettyyn teokseen.
155      Tämä voidaan selvittää tekemällä kysely Varaus- ja
      Asiakas- relaatioiden liitokseen määrittämällä
      ehdoksi, että teoksen standarditunnuksen on
156      täsmättävä haettuun teokseen.
157      Järjestetään lopuksi monikot varausajankohdan mukaan.
158
159      Kysytään, ketkä ovat varanneet Kalevalan.
160  */
161  SELECT asiakasNro, nimi, varausAjankohta
162  FROM Varaus JOIN Asiakas ON varaaajaAsiakasNro = asiakasNro
163  WHERE teosStandardiTunnus = '978-951-1-23676-4'
164  ORDER BY varausAjankohta ASC
165
166  /*
167      Selvitetään, missä toimipisteissä teoksen kappaleita
      on ja kuinka monta.
168      Tehdään kysely Toimipisteessa-relaatioon ja määritetä
      än kyselyn ehdoksi, että teoksen
      standarditunnuksen on täsmättävä haetun teoksen
      kanssa.
169      Ryhmitellään monikot toimipisteen nimen mukaan, jotta
      saadaa selville haluttu lukumäärä.
170
171      Kysytään, kuinka monta kappaletta Kalevalaa on
      kussakin toimipisteessä.
172  */
173  SELECT toimipisteNimi, COUNT(*) AS lukumaara
174  FROM Toimipisteessa
175  WHERE standardiTunnus = '978-951-1-23676-4'

```

```

176 GROUP BY toimipisteNimi
177
178 /*
179      Selvitetään, missä kuljetuksissa tietyn teoksen
          kappaleita on kyydissä.
180      Tehdään kysely Kuljetettavana- ja Kuljetus-
          relaatioiden luonnolliseen liitokseen ja määritell
          ään ehdoksi, että standarditunnuksen tulee täsmätä
          haetun teoksen kanssa.
181 */
182 SELECT lahtoToimipiste, paateToimipiste, lahtoaika
183 FROM Kuljetettavana NATURAL JOIN Kuljetus
184 WHERE standardiTunnus = '978-951-1-23676-4'
185
186 /*
187      Tietokannasta voidaan hakea teoksia eri hakuehtojen
          mukaan.
188      Haut voidaan kohdistaa aina Teos- ja Kirja-
          relaatioiden luonnolliseen liitokseen.
189      Kysytään esimerkiksi, mitä kirjoja Elias Lönnrotilta
          on.
190 */
191 SELECT nimi, julkaisuvuosi
192 FROM Teos NATURAL JOIN Kirja
193 WHERE tekija = 'Elias_Lönnrot'
194
195 /*
196      Teoksen kappaleiden olinpaikan selvitys voidaan
          toteuttaa etsimällä kappaleita
197      Kysytään, mitkä yksittäisen teoksen kappaleet eivät
          ole kotitoimipisteessä, vaan toisessa
          toimipisteessä, kuljetuksessa tai lainassa.
198 */
199 SELECT kappaleTunnus
200 FROM Lainassa
201 WHERE standardiTunnus = '978-951-1-23676-4'
202      UNION
203 SELECT kappaleTunnus
204 FROM Kuljetettavana
205 WHERE standardiTunnus = '978-951-1-23676-4'
206      UNION
207 SELECT kappaleTunnus
208 FROM Toimipisteessa
209 WHERE standardiTunnus = '978-951-1-23676-4' AND
          toimipisteNimi != (

```

```

210         SELECT kotitoimipiste
211         FROM Kappale
212         WHERE Kappale.standardiTunnus = Toimipisteessa.
                standardiTunnus AND
213                 Kappale.kappaleTunnus =
                        Toimipisteessa.kappaleTunnus)
214
215
216  /*
217      Selvitetään jokaiselle asiakkaalle viimeisen vuoden
                aikana
218      palautettujen lainojen määrä.
219  */
220  SELECT asiakasNro, nimi, COUNT(*) AS palautuksia
221  FROM Palautus NATURAL JOIN Asiakas
222  WHERE palautusAjankohta > (datetime('now', '-1_year'))
223  GROUP BY asiakasNro
224  ORDER BY nimi
225
226
227  /*
228      Selvitetään jokaiselle asiakkaalle maksettujen
                maksujen kokonaissumma.
229  */
230  SELECT asiakasNro, nimi, SUM(summa) AS maksettuYhteensa
231  FROM Maksu NATURAL JOIN Asiakas
232  WHERE maksettu = TRUE
233  GROUP BY asiakasNro
234  ORDER BY nimi
235
236  /*
237      Selvitetään viime kuun 10 suosituinta teosta
                palautusten määrän perusteella.
238  */
239  SELECT nimi, julkaisuvuosi, genre, COUNT(*) AS palautuksia
240  FROM Palautus NATURAL JOIN Teos
241  WHERE palautusAjankohta > (datetime('now', '-1_month'))
242  GROUP BY standardiTunnus
243  ORDER BY palautuksia DESC
244  LIMIT 10

```