Tietokantojen perusteet 2013, Periodi III
Kirjastotietokanta

Mika Viinamäki
Paavo Rohamo
John Lång
Eero Antila
Juha Koiranen

21. helmikuuta 2013

1 Tietosisältökartoitus

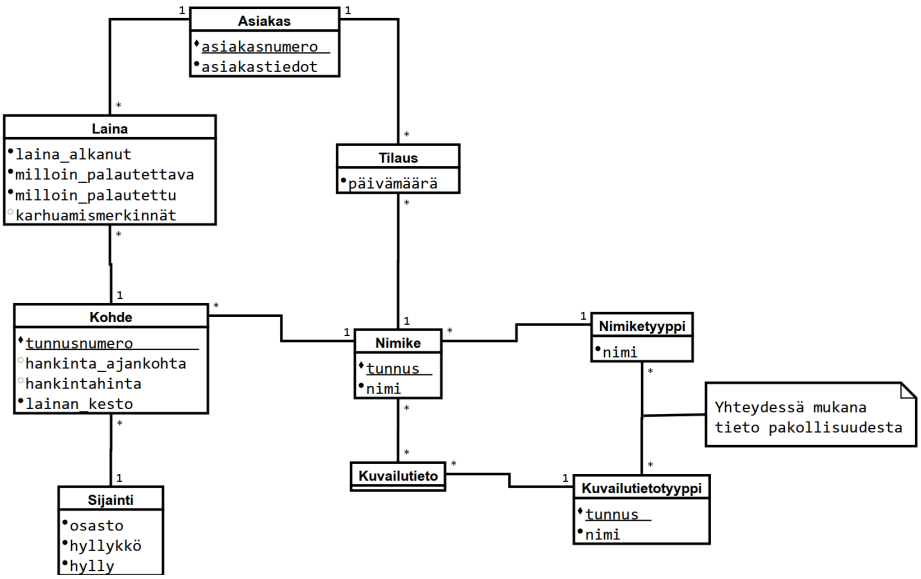
Tietokohteet **lihavoituna**, attribuutit *kursivoituna*.

Kirjaston kokoelmissa on noin 120000 eri **kohdetta**. Suurin osa kohteista on *lainattavia*, mutta osa kuuluu käsikirjastoon. Erilaisia **nimikkeitä** on noin 80000. Kullakin nimikkeellä on *tunnus*, *nimi* ja *tyyppi*. Lisäksi nimikkeeseen voi liittyä runsaasti erilaista **kuvailutietoa**, esimerkiksi tekijä, kustantaja, sivulukumäärä, jne. Kullakin **kuvailutietotyypillä** on yksikäsitteinen tunnus. Esimerkiksi kustantajatiedon *tyyppitunnus* on PUBL. **Nimiketyyppi**kohtaisesti (tyyppejä esimerkiksi kirja ja elokuva) on määritelty, mitä kuvailutietoja kyseisen tyypin nimikkeisiin liittyy sekä mitkä niistä ovat *pakollisia* ja mitkä valinnaisia. Uusia kuvailutietotyyppejä pitää pystyä lisäämään ja kytkemään nimikkeisiin muuttamatta tietokannan rakennetta. Kuvailutietoja käytetään lähinnä nimiketietojen haussa.

Kohdekohtaiset tiedot eivät riipu kohteen tyypistä. Kullakin kohteella on yksikäsitteinen *tunnusnumero*. Lisäksi kohteesta säilytetään **sijaintitietoa** (*osasto*, *hyllykkö*, *hylly*) ja tietoa *hankinta-ajankohdasta* ja *hankintahinnasta*.

Kirjaston asiakaskunta muodostuu noin 20000 **asiakkaasta**. Asiakkaasta on taltioitu *normaalit asiakastiedot*. Asiakkaalla on yksikäsitteinen *asiakasnumero*. Asiakkaat voivat **lainata** kohteita. He voivat myös **tilata** nimikkeitä. Palautuksen yhteydessä järjestelmä ilmoittaa ensimmäiselle jonottavalle asiakkaalle kohteen saapuomisesta. Lainasta tallennetaan *lainausajankohta* ja *palautuspäivä*. Lainaustietoihin voi liittyä myös *karhuamismerkintöjä*. Kun laina palautetaan, lainaustietoa ei poisteta, vaan se jää kantaan historiatiedoksi.

2 Käsitekaavio



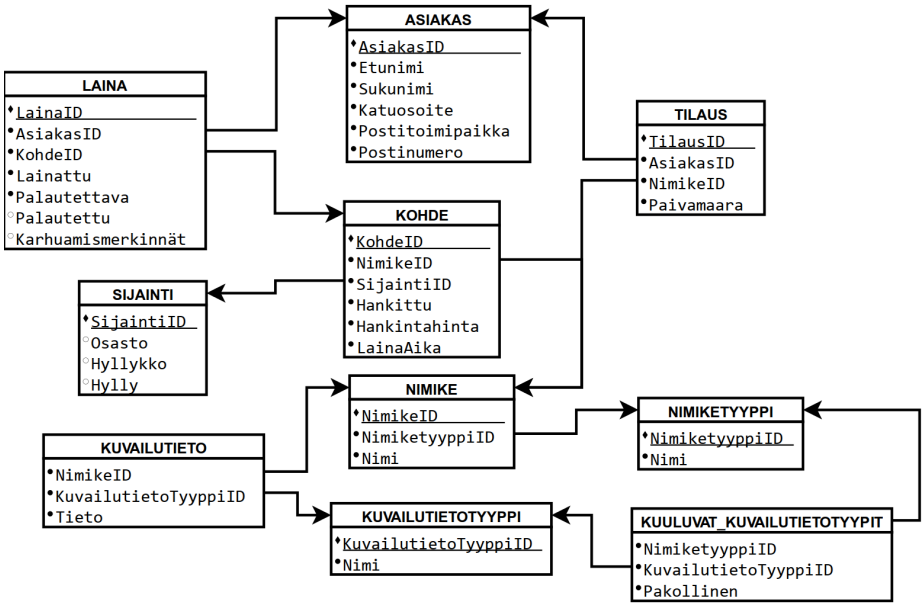
Nimike kuvaa jotain kirjaston valikoimissa olevaa teosta, ja kohde kuvaa yhtä fyysistä kopiota jostain tällaisesta nimikkeestä.

Nimiketyyppi kuvaa jotain muotoa, jota nimike edustaa - esimerkiksi kirja tai elokuva. Jokaiseen nimiketyyppiin liittyy tietyt kuvailutietotyypit, jotka voivat olla pakollisia tai vapaaehtoisia. Kuvailutieto tarkoittaa jotain nimikkeeseen liittyvää tietoa, kuten esimerkiksi nimikkeen kustantajaa tai ohjaajaa. Kuvailutietotyyppi kuvaa tämän tiedon tyyppiä.

Tilaus kuvaa varausta johonkin nimikkeeseen.

3 Tietokantakuvaus

3.1 Tietokantakaavio



3.2 SQL-lauseet

Kaikki tämän dokumentin SQL-lauseet olettavat käytössä olevan PostgreSQL-tietokantaohjelmiston tuoreehko versio (testattu versiolla 9.2.3).

```
/*
 * Asiakas on jokin kirjaston palveluita käyttävä henkilö. Tietokanta
 * olettaa, että järjestelmässä on vain asiakkaiden suomalaisia
 * osoitteita - "Postinumero" -kenttä ei salli kuin 5-merkkisiä
 * postinumeroita.
 */
CREATE TABLE ASIAKAS (
    AsiakasID SERIAL PRIMARY KEY,
    Etunimi VARCHAR(32) NOT NULL,
    Sukunimi VARCHAR(64) NOT NULL,
    Katuosoite VARCHAR(64) NOT NULL,
    Postitoimipaikka VARCHAR(32) NOT NULL,
    Postinumero CHAR(5) NOT NULL
);

/*
 * Kuvaa jotain fyysistä sijaintia Kohteille kirjastossa - Osasto,
 * Hyllykko ja Hylly ovat tarkemmin rajaamattomia numeroita.
 */
CREATE TABLE SIJAINTI (
    SijaintiID SERIAL PRIMARY KEY,
    Osasto INTEGER NOT NULL,
    Hyllykko INTEGER NOT NULL,
```

```
    Hylly INTEGER NOT NULL
);

/*
 * Nimiketyyppejä ovat esimerkiksi elokuva ja kirja. Tämä nimiketyypin
 * tekstimuotoinen kuvaus on attribuutissa "Nimi".
 */
CREATE TABLE NIMIKETYYPPI (
    NimiketyyppiID SERIAL PRIMARY KEY,
    Nimi VARCHAR(32) NOT NULL
);

/*
 * Nimike on jokin kirjaston valikoimista löytyvä teos, josta sitten
 * voi löytyä monia fyysisiä kopioita - kts. taulu KOHDE. Nimikkeellä
 * on aina jokin nimiketyyppi.
 */
CREATE TABLE NIMIKE (
    NimikeID SERIAL PRIMARY KEY,
    NimiketyyppiID INTEGER NOT NULL REFERENCES NIMIKETYYPPI,
    Nimi VARCHAR(255)
);

/*
 * Kuvailutietotyyppi on nimikkeeseen liittyvän kuvailutiedon tyyppi.
 * Jokaisella kuvailutietotyypillä on neljä-kirjaiminen koodi ja
 * tekstimuotoinen kuvaus.
 */
CREATE TABLE KUVAILUTIETOTYYPPI (
    KuvailutietoTyyppiID CHAR(4) PRIMARY KEY,
    Nimi VARCHAR(64) NOT NULL
);

/*
 * Taulu kuvaa johonkin nimiketyyppiin kuuluvia kuvailutietotyyppejä.
 * Yhteydet voivat olla pakollisia tai valinnaisia.
 */
CREATE TABLE KUULUVAT_KUVAILUTIETOTYYPIT (
    NimiketyyppiID INTEGER NOT NULL REFERENCES NIMIKETYYPPI,
    KuvailutietoTyyppiID CHAR(4) NOT NULL REFERENCES KUVAILUTIETOTYYPPI,
    Pakollinen BOOLEAN NOT NULL
);

/*
 * Kuvailutieto on johonkin nimikkeeseen liittyvää ns. metadataa.
 * Esimerkkeinä esimerkiksi kirjan kustantaja tai elokuvan ohjaaja.
 * Kuvailutietoon liittyy aina jokin kuvailutietotyyppi. Itse
 * tietosisältö - eli siis vaikkapa kustantajan tai ohjaajan nimi - on
 * taulun "Tieto"-attribuutissa.
 */
CREATE TABLE KUVAILUTIETO (
    NimikeID INTEGER NOT NULL REFERENCES NIMIKE,
    KuvailutietoTyyppiID CHAR(4) NOT NULL REFERENCES KUVAILUTIETOTYYPPI,
    Tieto VARCHAR(255)
);


/*
 * Kohde on jokin fyysinen kopio jostain nimikkeestä. Kohteella on
 * aina jokin sijainti, joka ei muutu vaikka kirja olisikin lainassa.
 * "Hankittu"-attribuutti kuvaa päivämäärää, jolloin kohde on hankittu
 * kirjastoon. "Hankintahinta"-attribuutti on kohden hankintahinta
 * euroina.  "Hankittu" ja "Hankintahinta" voivat olla NULL, jos ne
 * eivät ole tiedossa.  "LainaAika"-attribuutti on lainan kesto
 * päivinä tätä kohdetta lainattaessa.  Jos "LainaAika" on 0, kohteen
 * katsotaan kuuluvan käsikirjastoon eikä sitä voi lainata.
 */
CREATE TABLE KOHDE (
```

```
    KohdeID SERIAL PRIMARY KEY,
    NimikeID INTEGER NOT NULL REFERENCES NIMIKE,
    SijaintiID INTEGER NOT NULL REFERENCES SIJAINTI,
    Hankittu DATE,
    Hankintahinta NUMERIC(6, 2),
    LainaAika INTEGER NOT NULL
);

/*
 * Laina kuvaa yhden asiakkaan lainaustapahtumaa yhdestä kohteesta.
 * "Lainattu"-attribuutti on päivämäärä, milloin lainaus on alkanut.
 * "Palautettava"-attribuutti on päivämäärö, milloin lainaus pitäisi
 * viimeistään palauttaa. "Palautettu" on päivämäärä, milloin kohde on
 * palautettu kirjastoon tai NULL, jos kohdetta ei ole palautettu.
 * "Karhuamismerkinnät" on kertojen lukumäärä, milloin asiakasta on
 * muistutettu palauttamaan kohde kirjastoon sen ollessa myöhässä.
 */
CREATE TABLE LAINA (
    LainaID SERIAL PRIMARY KEY,
    AsiakasID INTEGER NOT NULL REFERENCES ASIAKAS,
    KohdeID INTEGER NOT NULL REFERENCES KOHDE,
    Lainattu DATE NOT NULL DEFAULT CURRENT_DATE,
    Palautettava DATE NOT NULL,
    Palautettu DATE,
    Karhuamismerkinnat INTEGER NOT NULL DEFAULT 0
);

/*
 * Tilaus on jonkin asiakkaan varaus jollekin nimikkeelle.
 * "Paivamaara"-attribuutti on päivä, jolloin varaus on tehty. Kun
 * varaus perutaan tai se on täytetty - eli asiakas on lainannut
 * varaamansa nimikkeen - kyseinen tilaus on tarkoitus poistaa
 * taulusta.
 */
CREATE TABLE TILAUS (
    TilausID SERIAL PRIMARY KEY,
    AsiakasID INTEGER NOT NULL REFERENCES ASIAKAS,
    NimikeID INTEGER NOT NULL REFERENCES NIMIKE,
    Paivamaara DATE NOT NULL
);
```

4 Riippuvuusanalyysi

Asiakas: Taulussa on funktionaalisia riippuvuuksia esimerkiksi postinumerolla ja postitoimipaikalla - jälkimmäinen riippuu suoraan ensimmäisestä. Taulu ei siis täytä kolmatta normaalimuotoa.

Sijainti, Nimiketyyppi, Nimike, Kuvailutietotyyppi, Kuuluvat_Kuvailutietotyypit, Kuvailutieto, Kohde, Tilaus: Attribuuteilla ei funktionaalisia riippuvuuksia kuin pääavaimiin.

Laina: Palautettava-attribuutin voi päätellä lainattu-attribuutin ja Kohde-taulun laina-ajan perusteella - olettaen että kohteen laina-aika ei koskaan muutu. Kohteen laina-aika voi kuitenkin muuttua, jolloin kyseessä ei ole funktionaalinen riippuvuus.

5 Esimerkkidata

```
INSERT INTO ASIAKAS (Etunimi, Sukunimi, Katuosoite, Postitoimipaikka, Postinumero)
     VALUES
    ('Lasse', 'Lainaaja', 'Lainaamokatu 33', 'Helsinki', '00100'),
    ('Heikki', 'Heijari', 'Kellarikuja 77', 'Veteli', '69700'),
    ('Hermanni', 'Hiiri', 'Hiiritie 9', 'Helsinki', '00920');

INSERT INTO SIJAINTI (Osasto, Hyllykko, Hylly) VALUES
    (1, 1, 1),
    (1, 1, 2),
    (1, 1, 3),
    (1, 1, 4),
    (1, 2, 1),
    (2, 1, 1);

INSERT INTO NIMIKETYYPPI (Nimi) VALUES
    ('Kirja'),
    ('Elokuva'),
    ('Musiikki');

INSERT INTO NIMIKE (NimiketyyppiID, Nimi) VALUES
    (1, 'Kaapon muistelmat'),
    (1, 'Säästöpossu'),
    (1, 'Monivitamiini-hivenainetabletti ja 10 muuta suosikkitarinaa'),
    (2, 'Pelottava elokuva'),
    (2, 'Pelottavampi elokuva'),
    (3, 'DISCO PRINCE');

INSERT INTO KUVAILUTIETOTYYPPI (KuvailutietoTyyppiID, Nimi) VALUES
    ('PUBL', 'Kustantaja'),
    ('YEAR', 'Julkaisuvuosi'),
    ('ARTI', 'Esittäjä'),
    ('AUTH', 'Kirjoittaja'),
    ('DIRE', 'Ohjaaja');

INSERT INTO KUULUVAT_KUVAILUTIETOTYYPIT (NimiketyyppiID, KuvailutietoTyyppiID,
     Pakollinen) VALUES
    (1, 'YEAR', TRUE),
    (1, 'PUBL', FALSE),
    (1, 'AUTH', TRUE),
    (2, 'YEAR', TRUE),
    (2, 'DIRE', TRUE),
    (3, 'YEAR', TRUE),
    (3, 'ARTI', TRUE);

INSERT INTO KUVAILUTIETO (NimikeID, KuvailutietoTyyppiID, Tieto) VALUES
    (1, 'YEAR', '1992'),
    (1, 'PUBL', 'Petterin kustantamo'),
    (1, 'AUTH', 'Petteri Peijonen'),
    (1, 'AUTH', 'Vessapaperimies'),
    (2, 'YEAR', '1939'),
    (2, 'AUTH', 'Keijo Suuri'),
    (3, 'YEAR', '1999'),
    (3, 'AUTH', 'Rainbow'),
    (4, 'YEAR', '1924'),
    (4, 'DIRE', 'Hurja Ohjaaja'),
    (5, 'YEAR', '1925'),
    (5, 'DIRE', 'Hurja Ohjaaja'),
    (6, 'YEAR', '2005'),
    (6, 'ARTI', 'Katamari Damacy');

INSERT INTO KOHDE (NimikeID, SijaintiID, Hankittu, Hankintahinta, LainaAika)
     VALUES
    (1, 1, CURRENT_DATE, NULL, 30),
    (1, 2, CURRENT_DATE - 10, NULL, 30),
    (1, 2, CURRENT_DATE - 30, NULL, 0),
```

```
    (2, 3, '1992-01-01', 30.00, 3),
    (3, 2, '2000-03-02', NULL, 30),
    (4, 1, NULL, NULL, 20),
    (5, 2, NULL, NULL, 10),
    (6, 3, NULL, NULL, 1);

INSERT INTO LAINA (AsiakasID, KohdeID, Lainattu, Palautettava, Palautettu) VALUES
    (1, 2, CURRENT_DATE, CURRENT_DATE + 10, NULL),
    (2, 4, '1999-01-01', '1999-02-02', '1999-01-15'),
    (3, 3, '1999-02-02', '2000-02-02', NULL);

INSERT INTO TILAUS(AsiakasID, NimikeID, Paivamaara) VALUES
    (1, 3, CURRENT_DATE);
```

6 Käyttötapauksia

6.1 Kaikkien kohteiden listaaminen

Kaikki tietokannassa olevat kohteet, niiden nimet ja tyypit selkokielisenä sekä sijaintitiedot saa seuraavalla kyselyllä:

```
SELECT NIMIKE.Nimi,
    NIMIKETYYPPI.Nimi,
    SIJAINTI.Osasto,
    SIJAINTI.Hyllykko,
    SIJAINTI.Hylly
FROM KOHDE
INNER JOIN NIMIKE
    ON KOHDE.NimikeID = NIMIKE.NimikeID
INNER JOIN NIMIKETYYPPI
    ON NIMIKE.NimikeTyyppiID = NIMIKETYYPPI.NimikeTyyppiID
INNER JOIN SIJAINTI
    ON KOHDE.SijaintiID = SIJAINTI.SijaintiID;
```

6.2 Nimikkeen kuvailutietojen listaaminen

Tietyn nimikkeen (tässä tapauksessa NimikeID = 1) kuvailutiedot kuvailutietotyyppien selkokielisten nimien kera:

```
SELECT KUVAILUTIETOTYYPPI.Nimi,
    KUVAILUTIETO.Tieto
FROM KUVAILUTIETO
INNER JOIN KUVAILUTIETOTYYPPI
    ON KUVAILUTIETO.KuvailutietoTyyppiID = KUVAILUTIETOTYYPPI.KuvailutietoTyyppiID
WHERE
    KUVAILUTIETO.NimikeID = 1
ORDER BY Nimi;
```

6.3 Nimikkeiden hakeminen kuvailutietojen perusteella

Esimerkkinä kaikki nimikkeet, joiden kustantaja on "Pekka":

```
SELECT DISTINCT NIMIKE.NimikeID, NIMIKE.Nimi
FROM NIMIKE
INNER JOIN KUVAILUTIETO
    ON NIMIKE.NimikeID = KUVAILUTIETO.NimikeID
WHERE
    KUVAILUTIETO.KuvailutietoTyyppiID = 'PUBL'
AND
    KUVAILUTIETO.Tieto = 'Pekka'
```

6.4 Kohteen lainaaminen

Lainaustapahtuman yhteydessä tiedämme kohteen `KohdeID`:n - se saadaan esimerkiksi kohteen takana olevasta viivakoodista. Esimerkeissä `KohdeID` on 1.

Lisäksi asiakkaan `AsiakasID` on tiedossa - se saadaan vaikkapa asiakkaan kirjastokortista. Esimerkeissä `AsiakasID` on niin ikään 1.

Ihan ensin tarkistetaan, onko kohde ylipäätään paikalla ja onko se lainattavissa — jos ei ole, lainausta ei voida tehdä.

```
SELECT EXISTS (
    FROM KOHDE
    WHERE
        KOHDE.KohdeID = 1
    AND
        KOHDE.LainaAika > 0
    AND
        NOT EXISTS (
            SELECT 1
            FROM Laina
            WHERE
                KohdeID = KOHDE.KohdeID
            AND
                Palautettu IS NULL
        );
)
```

Tästä tarkistuksesta selvittyämme selvitämme kohteen `NimikeID` — sitä tarvitaan useaan otteeseen:

```
SELECT NimikeID
FROM KOHDE
WHERE KohdeID = 1
LIMIT 1;
```

Oletetaan, että kohteen `NimikeID`:ksi saatiin 2.

Nyt tarkistetaan, onko nimikkeestä enemmän kohteita paikalla kuin avoimia tilauksia - jos ei ole, lainausta ei voida tehdä. Tilausten määrä tälle nimikkeelle saadaan seuraavan kyselyn avulla:

```
SELECT count(*)
FROM TILAUS
WHERE TILAUS.NimikeID = 2;
```

Lainattavissa ja paikalla olevien kyseisen nimikkeen kohteiden määrä taas seuraavasti:

```
SELECT count(*)
FROM KOHDE
WHERE
    KOHDE.NimikeID = 2
AND
    KOHDE.LainaAika > 0
AND
    NOT EXISTS (
        SELECT 1
        FROM Laina
        WHERE
            KohdeID = KOHDE.KohdeID
        AND
            Palautettu IS NULL
    );
```

Jos tilauksia on enemmän tai yhtä monta kuin saatavilla olevia kohteita, varausta ei voida tehdä - ellei asiakas ole niin monen varauksen joukossa kuin

saatavilla olevia kohteita on. Oletetaan, että paikalla olevia kohteita halutulle nimikkeelle on 5:

```
SELECT EXISTS (
    SELECT 1
    FROM TILAUS
    WHERE
        NimikeID = 2
    AND
        AsiakasID = 1
    ORDER BY Paivamaara DESC
    LIMIT 5
);
```

Jos kysely palauttaa TRUE, voidaan jatkaa vaikka saatavilla olevia kohteita ei olisikaan enempää kuin varauksia kyseiselle nimikkeelle. Nyt kun kaikki on kunnossa, voidaan kirjata uusi lainaus:

```
INSERT INTO LAINA (AsiakasID, KohdeID, Lainattu, Palautettava) VALUES (
    1,
    1,
    CURRENT_DATE,
    CURRENT_DATE + (SELECT LainaAika FROM KOHDE WHERE KohdeID = 1)
);
```

Lopuksi voidaan poistaa kyseisen asiakkaat kyseiselle nimikkeelle tekemät varaukset, koska asiakas on saanut kohteensa:

```
DELETE FROM TILAUS WHERE AsiakasID = 1 AND NimikeID = 2;
```