

# Tietokantojen perusteet 2013, Pershowspaces=false, iodi III Kirjastotietokanta

Mika Viinamäki  
Paavo Rohamo  
John Lång  
Eero Antila  
Juha Koiranen

17. helmikuuta 2013

## 1 Tietosisältökartoitus

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

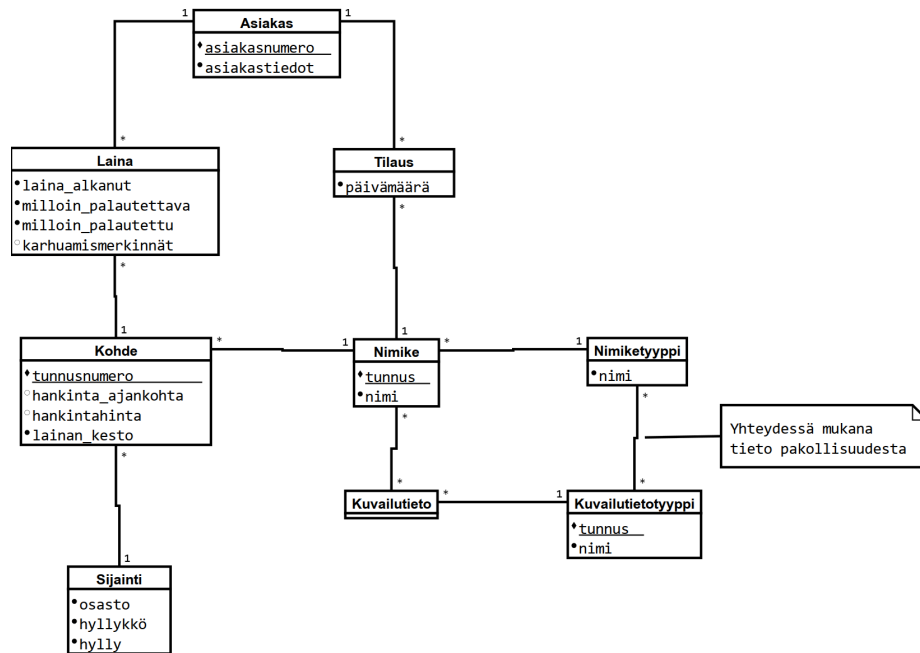
Kirjaston kokoelmissa on noin 120000 eri **kohdetta**. Suurin osa kohteista on *lainattavia*, mutta osa kuuluu käsikirjastoon. Erilaisia **nimikkeitä** on noin 80000. Kullakin nimikkeellä on *tunnus*, *nimi* ja *tyyppi*. Lisäksi nimikkeeseen voi liittyä runsaasti erilaista **kuvailutietoa**, esimerkiksi tekijä, kustantaja, sivulukumäärä, jne. Kullakin **kuvailutietotyyppillä** on yksikäsitteinen tunnus. Esimerkiksi kustantajatiedon *tyyppitunnus* on PUBL. **Nimiketyyppikohtaisesti** (tyyppejä esimerkiksi kirja ja elokuva) on määritelty, mitä kuvailutietoja kyseisen tyypin nimikkeisiin liittyy sekä mitkä niistä ovat *pakollisia* ja mitkä valinnaisia. Uusia kuvailutietotyyppieitä 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 saapumisesta. Lainasta tallennetaan *lainausajankohta* ja *palautuspäivä*.

Lainaustietoihin voi liittyä myös *karhuamismerkintöjä*. Kun laina palautetaan, lainaustietoa ei poisteta, vaan se jää kantaan historiatiedoksi.

## 2 Käsitekaavio



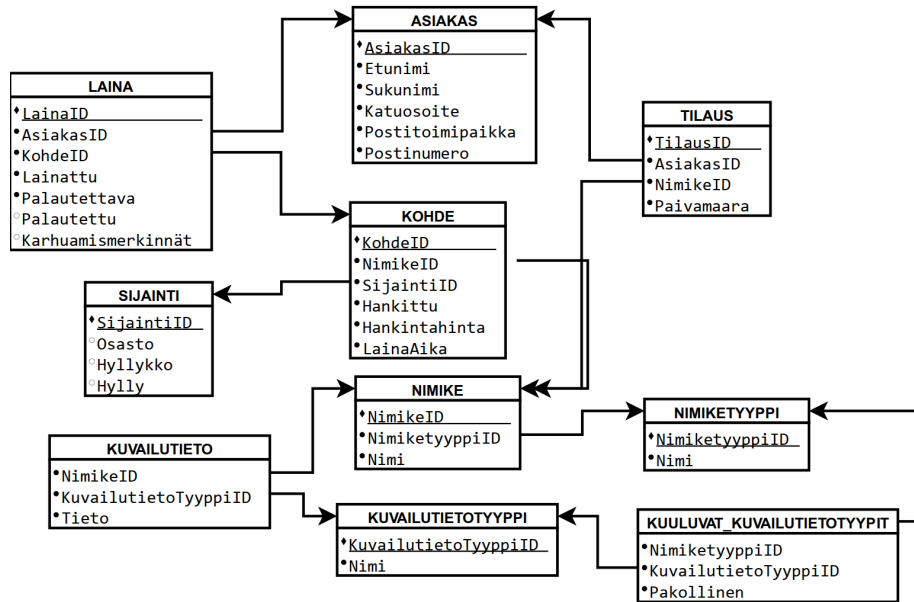
Nimike kuvaa jotain kirjaston valikoimissa olevaa teosta, ja kohde kuvaa yhtä fyysistä kopiota jostain tällaisesta nimikkeestä.

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

Tilaua 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 postinumeronumeroita.
 */
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 lö
 * ytyä
 * 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 kuvailutietotyyppillä on neljä-kirjaiminen koodi ja
 * tekstimuotoinen kuvaus.
 */
CREATE TABLE KUVAILUTIETOTYYPI (
    KuvailutietoTyypID CHAR(4) PRIMARY KEY,
    Nimi VARCHAR(64) NOT NULL
);

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

/*
 * Kuvailutieto on johonkin nimikkeeseen liittyvää ns. metadataa. Esimerkkein
 * ä
 * 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,
    KuvailutietoTyypID CHAR(4) NOT NULL REFERENCES KUVAILUTIETOTYYPI,
    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 päivämäärä, milloin lainaus pitäisi viimeist
    ään
* 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-tilun 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', 'Lainaja', '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 muistelmät'),
(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;

```