

-- keskusdivari

```
CREATE TABLE Asiakas(  
    asiakas_id INT,  
    etunimi VARCHAR(20) NOT NULL,  
    sukunimi VARCHAR(20) NOT NULL,  
    a_osoite VARCHAR(64) NOT NULL,  
    sposti VARCHAR(32) UNIQUE,  
    puh_nro VARCHAR(15) NOT NULL,  
    salasana VARCHAR(20) NOT NULL,  
    taso INT DEFAULT 0,  
    PRIMARY KEY (asiakas_id));
```

```
CREATE TABLE Kirja(  
    isbn VARCHAR(20),  
    tekija VARCHAR(32) NOT NULL,  
    k_nimi VARCHAR(64) NOT NULL,  
    tyyppi VARCHAR(32) NOT NULL,  
    genre VARCHAR(32) NOT NULL,  
    vuosiluku INT,  
    PRIMARY KEY(isbn));
```

```
CREATE TABLE Teos(  
    teos_id INT,  
    hinta DECIMAL(7,2) NOT NULL,  
    tila BOOLEAN DEFAULT TRUE,  
    paino INT NOT NULL,  
    isbn VARCHAR(20),  
    PRIMARY KEY(teos_id),  
    FOREIGN KEY(isbn) REFERENCES kirja(isbn));
```

```
CREATE TABLE Tilaus(  
    -- keskusdivari
```

```
tilaus_id INT,  
tilaaja VARCHAR(32),  
hinta DECIMAL(7,2) NOT NULL,  
paino INT NOT NULL,  
myyty BOOLEAN DEFAULT FALSE,  
PRIMARY KEY(tilaus_id),  
FOREIGN KEY(tilaaja) REFERENCES asiakas(sposti));
```

```
CREATE TABLE Tilaustiedot(  
    tilaus_id INT,  
    teos_id INT,  
    hinta DECIMAL(7,2) NOT NULL,  
    paino INT NOT NULL,  
    tilausvalmis INT DEFAULT 0,  
    CONSTRAINT tilaus_teos_id PRIMARY KEY (tilaus_id, teos_id),  
    FOREIGN KEY(tilaus_id) REFERENCES Tilaus(tilaus_id),  
    FOREIGN KEY(teos_id) REFERENCES Teos(teos_id));
```

```
CREATE TABLE Lahetys(  
    lahetys_id INT,  
    tilaus_id INT,  
    hinta DECIMAL(7,2) NOT NULL,  
    paino INT NOT NULL,  
    pvm DATE,  
    CONSTRAINT lahetys_tilaus_id PRIMARY KEY(lahetys_id, tilaus_id),  
    FOREIGN KEY (tilaus_id) REFERENCES Tilaus(tilaus_id));
```

```
-- single divari
```

```
CREATE TABLE kirja(  
    isbn VARCHAR(20),  
    tekija VARCHAR(32) NOT NULL,  
    k_nimi VARCHAR(64) NOT NULL,  
    tyyppi VARCHAR(32) NOT NULL,  
    genre VARCHAR(32) NOT NULL,  
    vuosiluku INT,  
    PRIMARY KEY(isbn));  
  
CREATE TABLE teos(  
    teos_id INT,  
    ostohinta DECIMAL(7,2) NOT NULL,  
    hinta DECIMAL(7,2) NOT NULL,  
    paino INT NOT NULL,  
    tila BOOLEAN DEFAULT TRUE,  
    isbn VARCHAR(20),  
    PRIMARY KEY(teos_id),  
    FOREIGN KEY(isbn) REFERENCES kirja(isbn));
```

-- views

```
CREATE VIEW kirjat_lkm AS  
  
    SELECT k_nimi, tekija, tyyppi, genre, kirja.isbn, hinta, paino, COUNT(teos_id) as teokset_lkm  
    FROM keskus.Kirja, keskus.Teos  
  
    WHERE kirja.isbn = teos.isbn  
  
    GROUP BY k_nimi, tekija, tyyppi, genre, kirja.isbn, hinta, paino  
  
    ORDER BY k_nimi ASC;
```

```
CREATE VIEW kirjat AS  
  
    SELECT tekija, k_nimi, tyyppi, genre, kirja.isbn, hinta, paino, tila, teos_id  
    FROM keskus.Kirja, keskus.Teos  
  
    WHERE kirja.isbn = teos.isbn
```

```
ORDER BY tekija ASC;
```

```
-- triggers
```

```
CREATE TRIGGER lisaaKirja
```

```
AFTER INSERT ON haara.Teos
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE lisaaKirja();
```

```
CREATE OR REPLACE FUNCTION lisaaKirja()
```

```
RETURNS trigger AS $$
```

```
DECLARE
```

```
    tempKirja keskus.Kirja%ROWTYPE;
```

```
    tempTeos keskus.Teos%ROWTYPE;
```

```
BEGIN
```

```
    IF(TG_OP = 'INSERT') THEN
```

```
        SELECT * INTO tempKirja FROM keskus.Kirja WHERE keskus.Kirja.isbn = NEW.isbn;
```

```
        IF NOT FOUND THEN
```

```
            SELECT * INTO tempKirja FROM haara.Kirja WHERE haara.Kirja.isbn = NEW.isbn;
```

```
            INSERT INTO keskus.Kirja VALUES(tempKirja.isbn, tempKirja.tekija,
```

```
            tempKirja.k_nimi, tempKirja.tyyppi, tempKirja.genre, tempKirja.vuosiluku);
```

```
        END IF;
```

```
        SELECT * INTO tempTeos FROM keskus.Teos WHERE keskus.Teos.teos_id = NEW.teos_id;
```

```
        IF NOT FOUND THEN
```

```
            INSERT INTO keskus.Teos VALUES(NEW.teos_id, NEW.hinta, NEW.tila,
```

```
            NEW.paino, NEW.isbn);
```

```
        END IF;
```

```
    END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

