

Tietokantaohjelmointi - kevät 2020

Harjoitustyön suunnitteludokumentti 1

Ryhmä 32 – Joonas Ilvonen & Lauri Viitanen

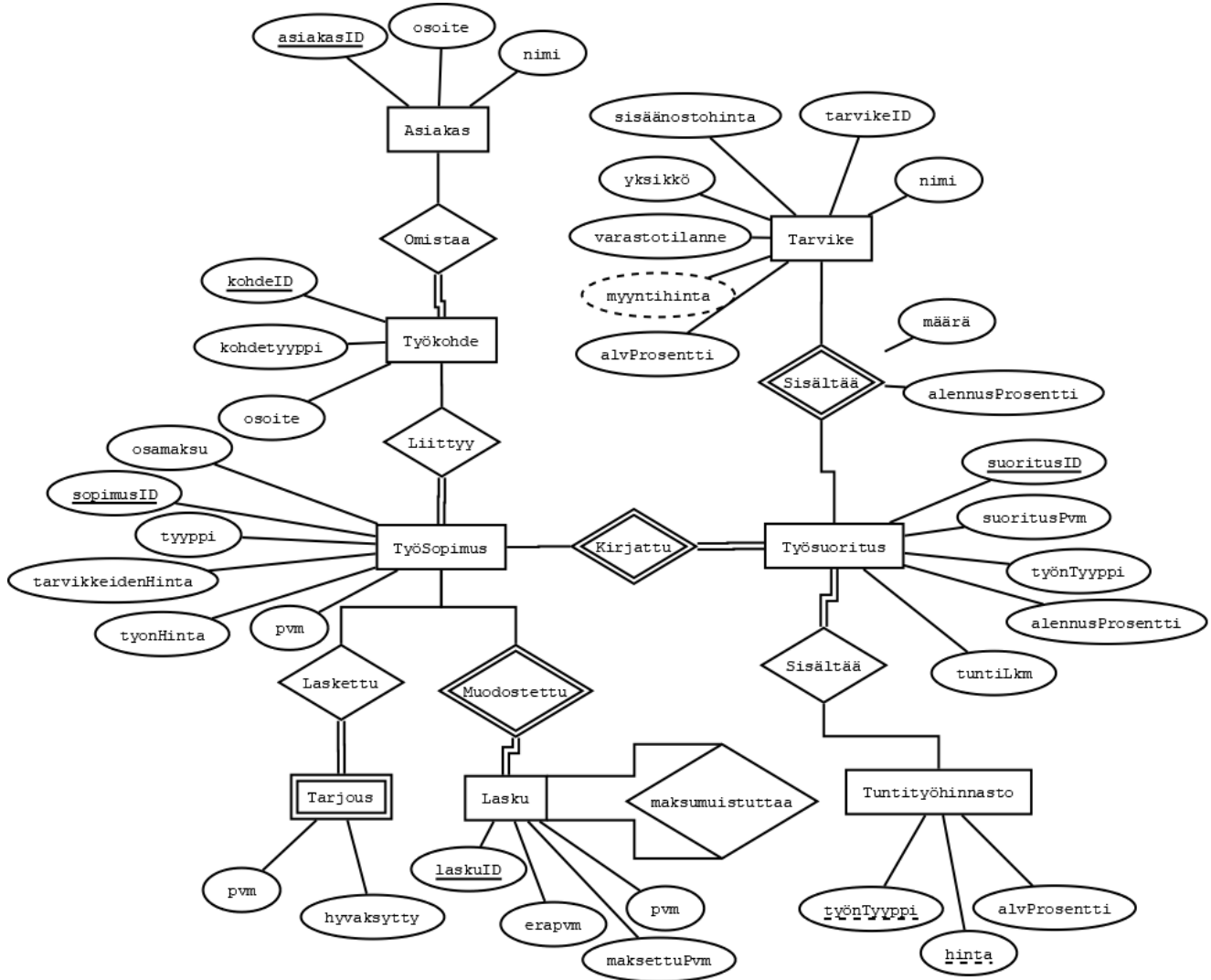
Sisältö:

1. Tehdyt oletukset
2. Tietokannan ER-kaavio
3. Tietokannan relaatiokaavio
4. Tietokannan luontilauseet
5. Johdetut tiedot
6. Tapahtumat T1-T5
7. Raportit R1-R5
8. Toteutustapa

1. Tehdyt oletukset

- ”Tmi Sähkötärsky suorittaa sähkötoita sekä välittää asentamiaan sähkötarvikkeita”
 - Tmi Sähkötärsky ei siis myy tuotteita suoraan asiakkailleen ilman asennuspalveluita
- Tuntityösopimusta ei laskuteta urakan ollessa kesken. Esim. jokaisen kk-lopussa ei laskuteta siihen mennessä tehtyjen tuntien ja tarvikkeiden määrää

2. Tietokannan ER-kaavio



3. Tietokannan relaatiokaaviot

- **ASIAKAS** (asiakasID, nimi, osoite)
- **TYOKOHDE** (kohdeID, omistajaID, kohdetyyppi, osoite)
- **TYOSOPIMUS** (sopimusID, kohdeID, tyyppi, tyonHinta, tarvikkeidenHinta, osamaksu, pvm, selite)
- **TYOTARJOUS** (sopimusID, pvm, hyväksytty)
- **LASKU** (laskuID, sopimusID, pvm, erapaiva, maksettuPVM, edeltavaLasku)
- **TUNTITYOHINNASTO** (tyyppe, hinta, alv)
- **TYOSUORITUS** (suoritusID, sopimusID, tyolaji, pvm, tuntiLkm, aleprosentti)
- **TARVIKE** (tarvikeID, nimi, sisaanostohinta, myyntihinta, yksikko, varastoLkm, alv)

- TARVIKELUETTELO (suoritusID, tarvikeID, lkm, aleProsentti)

4. Tietokannan luontilauseet

```
-- Uudet tietotyypit sopimuslajille ja yksiköille
CREATE TYPE sopimuslaji AS ENUM ('urakka', 'tunti');
CREATE TYPE yksikot AS ENUM ('kpl', 'kg', 'm', 'cm', 'g', 'l', 'kela');

CREATE TABLE asiakas (
    asiakasID SERIAL PRIMARY KEY,
    nimi VARCHAR (50) NOT NULL,
    osoite VARCHAR (100) NOT NULL
);

CREATE TABLE tyokohde (
    kohdeID SERIAL PRIMARY KEY,
    omistajaID INT,
    kohdetyyppi VARCHAR (50) NOT NULL,
    osoite VARCHAR (100) NOT NULL,
    FOREIGN KEY (omistajaID) REFERENCES asiakas(asiakasID)
);

CREATE TABLE tyosopimus (
    sopimusID SERIAL PRIMARY KEY,
    kohdeID INT,
    tyyppi SOPIMUSLAJI NOT NULL,
    tyonHinta NUMERIC,
    tarvikkeidenHinta NUMERIC,
    -- laskujen määrä -> arvo 1 = ei osamaksua, arvo 4 = maksetaan 4 osassa
    osamaksu INT NOT NULL,
    pvm DATE,
    selite VARCHAR (50),
    CONSTRAINT jos_urakkatyo_hinnat_NOTNULL
        CHECK ( (tyyppi = 'tunti' AND tyonHinta IS NULL
                AND tarvikkeidenHinta IS NULL )
              OR (tyyppi = 'urakka' AND tyonHinta IS NOT NULL
                AND tarvikkeidenHinta IS NOT NULL ) ),
    FOREIGN KEY (kohdeID) REFERENCES tyokohde(kohdeID)
);

CREATE TABLE tyotarjous (
    sopimusID INT,
    pvm DATE NOT NULL,
    hyväksytty BOOLEAN,
    FOREIGN KEY (sopimusID) REFERENCES tyosopimus(sopimusID)
);

-- Laskussa tarkistus, ettei laskun eräpäivä ole myöhemmin kuin laskun pvm
CREATE TABLE lasku (
    laskuID SERIAL PRIMARY KEY,
    sopimusID INT,
    pvm DATE NOT NULL,
    erapaiva DATE NOT NULL,
    maksettuPvm DATE,
    edeltavaLasku INT,
    CONSTRAINT check_dates CHECK (erapaiva > pvm),
    FOREIGN KEY (edeltavaLasku) REFERENCES lasku(laskuID),
    FOREIGN KEY (sopimusID) REFERENCES tyosopimus(sopimusID)
);
```

```

CREATE TABLE tuntityhinnasto (
    tyyppi VARCHAR (50) NOT NULL UNIQUE,
    hinta NUMERIC NOT NULL,
    alv INT NOT NULL,
    PRIMARY KEY(tyyppi, hinta, alv)
);

CREATE TABLE tyosuoritus (
    suoritusID SERIAL PRIMARY KEY,
    sopimusID INT,
    tyolaji VARCHAR (50),
    pvm DATE NOT NULL,
    tuntiLkm INT,
    aleProsentti INT,
    FOREIGN KEY (sopimusID) REFERENCES tyosopimus(sopimusID)
);

CREATE TABLE tarvike (
    tarvikeID SERIAL PRIMARY KEY,
    nimi VARCHAR (50),
    sisaanostohinta NUMERIC,
    myyntihinta NUMERIC,
    yksikko yksikot,
    varastoLkm INT,
    alv INT
);

CREATE TABLE tarvikeluettelo (
    suoritusID INT,
    tarvikeID INT,
    lkm INT,
    aleProsentti INT,
    FOREIGN KEY (suoritusID) REFERENCES tyosuoritus(suoritusID),
    FOREIGN KEY (tarvikeID) REFERENCES tarvike(tarvikeID)
);

```

5. Johdetut tiedot

Tietokantaa suunnitellessa on pyritty minimoimaan redundantin ja ylimääräisen informaation tallentaminen. Esimerkiksi Lasku-relaatiossa ei ole tallennettu mitään summia, sillä ne voidaan aggregoida työsopimuksen takaa löytyvistä tiedoista. Sopimuksen ollessa tuntipohjainen haetaan kaikki sille kirjatut työsoritukset ja kaikki työsorituksiin kirjatut tarvikeluettelot. Näistä pystytään summaamaan kokonaismäärä työlle ja tarvikkeille, sekä tarvittaessa muotoilemaan erittely laskulle esimerkiksi päiväkohtaisesti.

Lisäksi muistutus- ja karhulaskuihin liittyvä informaatio johdetaan muusta tiedosta. Lasku-relaatiossa on attribuutit laskuID eräpäivä, maksettuPvm ja edellinenLasku. Näistä pystytään hakemaan ensin kaikki erääntyneet laskut (eräpäivä < nykyinen päivä), joista voidaan rekursiivisesti hakea kaikki edeltävät laskut. Mikäli laskulla ei ole edeltäviä laskuja, luodaan järjestelmään uusi lasku, jossa on lisätty muistutusmaksu ja tarvittaessa maksulisä (yllä olevissa malleissa ja luontilauseissa puuttuu Lasku-relaatiosta 'laskulisä'-attribuutti, johon lasketaan muistutuksista ja muista kertyvät kasvavat kulut. Tämä lisätty jälkikäteen).

Tarvikkeen myyntihinta on tässä kohtaa johdettu suoraan sisäänostohinnasta kertomalla se 1.5 kertoimella. Myöhemmin on tarkoitus antaa käyttäjälle mahdollisuus määrittää myyntihinta suoraan eksplisiittisesti tai antaa kerroin mikä lisätään sisäänostohintaan.

6. Tapahtumat T1-T5

T1:

- Luetaan asiakas relaatiosta id haetulle asiakkaalle (nimi ja osoite täsmäävät)
- Luetaan tyokohde relaatiosta viimeisin id
- Kirjoitetaan tyokohde relaatioon uusi merkintä, jossa id yhden suurempi kuin viimesin, asiakas id haettu, kohdetyyppi ja osoite käyttäjän syöttämiä

T2:

- Luetaan sopimus relaatiosta kyseisen kohteen sopimuksen id
- Luetaan tyosuoritus relaatiosta viimeisin id
- Kirjoitetaan tyosuoritus relaatioon uusi merkintä, jossa id yhden suurempi kuin viimeisin, kohde id haettu, pvm nykyinen ja työlaji, tuntimäärä ja aleprosentti käyttäjän syöttämiä

T3

- Luetaan lasku relaatiosta kaikki laskut joiden eräpäivä on mennyt, joille ei ole merkattu maksettu päivämäärää ja joille ei löydy laskua, jossa olisi merkattu erääntyneen laskun id edeltäväksi laskuksi
- Kirjoitetaan uudet laskut jokaiselle löytyneelle, joissa: pvm nykyinen, eräpäivä 1kk eteenpäin, edeltävän id sama kuin haetun laskun ja laskulisään lisätty muistutuslaskun kustannukset

T4

- Luetaan lasku relaatiosta kaikki laskut, joilla on yksi edeltävä lasku ja joita ei ole kuitattu maksetuksi
- Kirjoitetaan uudet laskut jokaiselle löytyneelle, joissa: pvm nykyinen, eräpäivä 1kk eteenpäin, edeltävän id sama kuin haetun laskun ja laskulisään lisätty karhulaskun kustannukset

T5

- Luetaan tekstimuodossa oleva hinnasto
- Parsitaan tietokantaan sopivaan muotoon
- Yhdistetään tavarantoimittajan tuotteiden id:t ja oman kannan tuotteiden id:t toisiinsa
- Poistetaan vanhat tuotteet ja kirjoitetaan tekstitiedostoon historiatiedoksi
- Päivitetään uudet tuotteet tuote relaatioon

7. Raportit R1-R5

8. Toteutustapa

Näillä näkyminen JDBC.