

SUUNNITELMA

Ryhmä 28

Jani Aakio

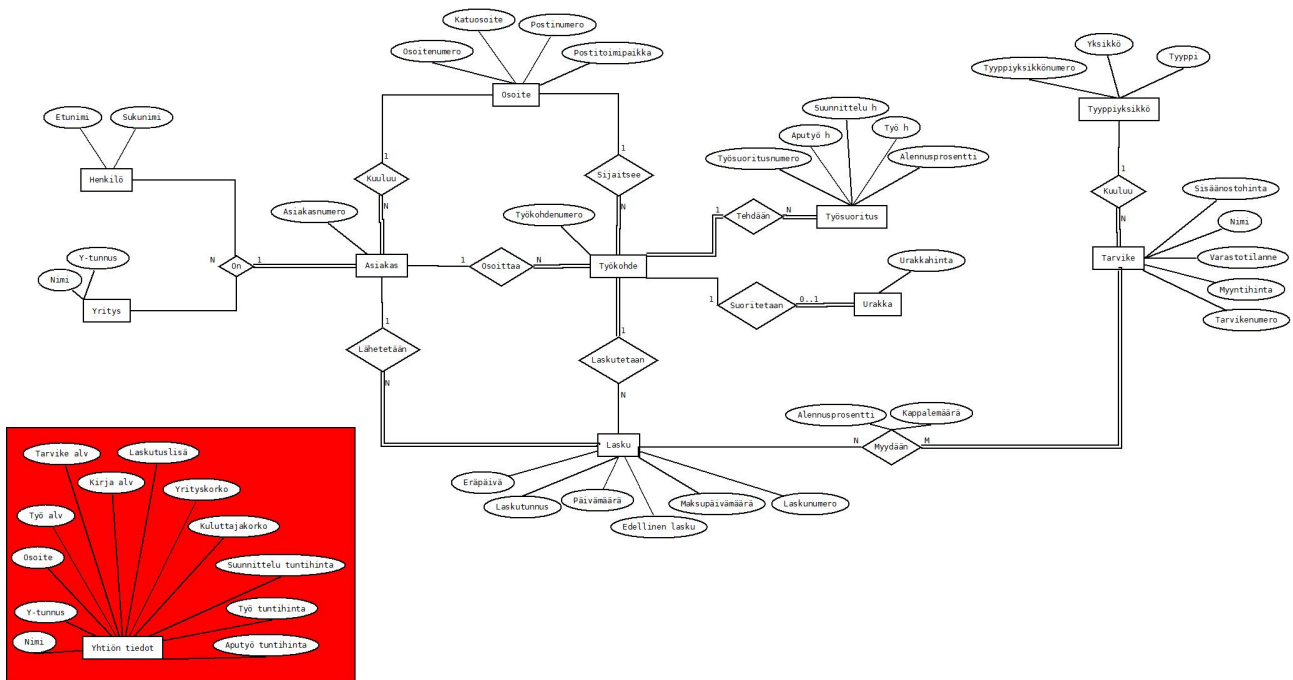
Niko Alanko

Juha-Pekka Salonen

Yhteyshenkilön sähköposti:

jani.aakio@tuni.fi

ER-kaavio



Huomio ER-kaavioon: Punaisella taustalla oleva taulu on tehty helpottamaan ohjelman samanaikaista käyttöä. Ymmärrämme että se ei välttämättä kuuluisi tähän ER-kaavioon mutta se on lisätty tähän kaavioon jotta kaikki ryhmän jäsenet ymmärtäisivät mistä tietyt tiedot tulevat.

Lasku-taulua tullaan käyttämään niin että sinne kerätään laskuun kuuluvia tarvikkeita. Päivämäärä-attribuutti kertoo onko lasku valmis vai ei, koska laskulle annetaan päivämäärä vasta kun se on valmis. Laskunumero kertoo monesko lasku tämä on samasta aiheesta, joten kaikki laskut (normaalit- tai karhulaskut) tulevat samaan tauluun.

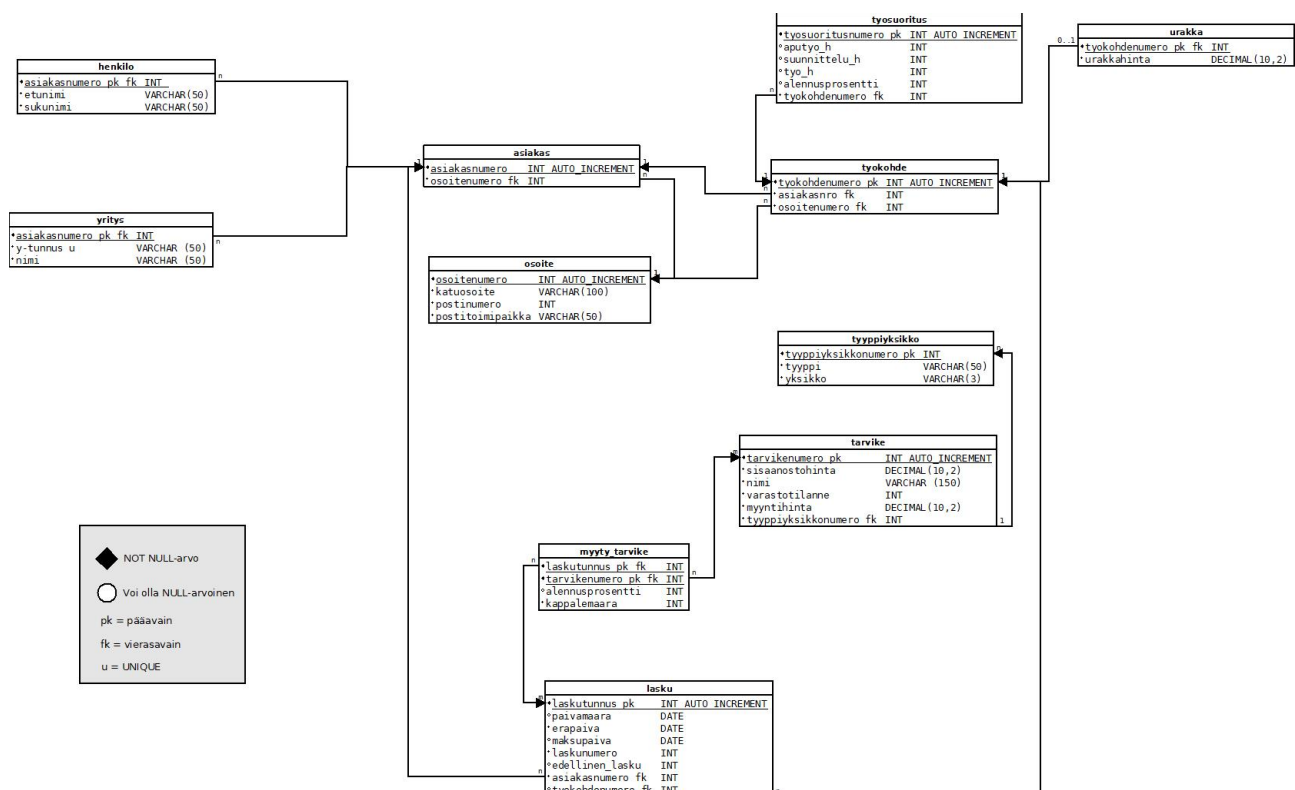
Alennusprosentteja ei urakkatöistä anneta vaikka tehtävänanto niin saattoi ilmaista. Emme pidä sitä loogisena.

ER-kaavion muunnos tietokantakaavioksi

- Asiakas-entiteetistä tehdään asiakas-taulu, pääavaimena asiakasnumero.
- Asiakas-entiteettiin IS-A-suhteessa olevista Henkilö-entiteetistä ja Yritys-entiteetistä tehdään omat taulut. Molemmat taulut saavat pääavaimiksi ja vierasavaimiksi asiakas-taulun pääavaimen NOT NULL-määreillä.
- Osoite-entiteetistä tehdään oma taulu, pääavaimena osoitenumero.
- Asiakkaan ja Osoitteen suhteesta syntyy asiakas-tauluun osoite-taulun pääavain vierasavaimeksi NOT NULL-määreellä.
- Työkohde-entiteetistä tehdään taulu, pääavaimena työkohdenumero.

- Työkohteen ja Osoitteen suhteesta syntyy työkohde-tauluun osoite-taulun pääavain vierasavaimeksi NOT NULL-määreellä.
 - Työsuoritus-entiteetistä tehdään oma taulu, pääavaimena työsuoritusnumero.
 - Urakka-entiteetistä tehdään oma taulu.
 - Työkohteeseen suhteista Urakkaan ja Työsuoritukseen muodostetaan molempiin tauluihin vierasavaimiksi työkohde-taulun pääavain NOT NULL-määreellä. Tämä on tulee lisäksi urakka-taulun pääavaimeksi.
 - Lasku-entiteetistä tehdään oma taulu, pääavaimena laskutunnus.
 - Asiakkaan ja Laskun suhteesta muodostuu lasku-tauluun vierasavaimeksi asiakas-taulun pääavain NOT NULL-määreellä.
 - Työkohteen ja Laskun suhteesta muodostuu lasku-tauluun vierasavaimeksi työkohde-taulun pääavain ilman NOT NULL-määrettä, johtuen Lasku-entiteetin osittaisesta osallistumisesta suhteeseen.
 - Tarvike-entiteetistä tehdään oma taulu, pääavaimena tarvikenumero.
 - Laskun ja tarvikkeen suhteesta luodaan suhdetaulu, jonka pääavaimiksi tulevat lasku-taulun ja tarvike-taulun pääavaimet.
 - Nämä toimivat myös vierasavaimina. Suhteen attribuutit tulevat taulun sarakkeiksi.
- Tyyppiyksikkö-entiteetistä tehdään oma taulu, pääavaimena tyyppiyksikkönumero. Tyyppiyksikön ja Tarvikkeen suhteesta johtuen tarviketauluun tulee vierasavaimeksi tyyppiyksikkö-taulun pääavain NOT NULL-määreellä

Tietokanta-kaavio



Tapahtumien kuvaukset

T1:

1. Pyydetään asiakkaan nimi tai numero sekä tarvittavat tiedot mitä lisätä
2. Tarkistetaan että asiakas on olemassa tekemällä kysely asiakas-tauluun
3. Lisätään työkohde-tauluun uusi rivi jossa tiedot sekä viittaus asiakkaaseen
4. Lisätään lasku-tauluun uusi rivi jossa viittaus työkohteeseen sekä asiakkaaseen

T2:

1. Pyydetään työkohteen nimi tai tunnus sekä tarvittavat tiedot
2. Tarkistetaan että työkohde on olemassa tekemällä kysely työkohde-taulusta.
3. Lisätään rivi työsuoritus-tauluun jossa annetut tunnit ja viitataan se käskettyyn työkohteeseen.
4. Etsitään lasku-taulusta työkohteeseen liitetty rivi jossa ei ole vielä päiväystä.
5. Jos riviä ei löydy, sellainen luodaan.
6. Tarkistetaan että käytetyt esineet löytyvät tarvike-taulusta.
6. Lisätään jokaiselle esineelle uusi rivi myyty_tarvike-tauluun joka viittaa tarvike-tauluun sekä lasku-tauluun.

T3:

1. Etsitään laskut taulusta kaikki laskut joissa eräpäivä < nykyinen päivä, maksupäivä = NULL, ja laskun numero on 1
2. Etsitään laskut-taulusta kaikki laskut jotka ovat viittauksia näihin laskuihin
3. Poistetaan kohdan 1. hakutuloksista kaikki joille löytyy viittaus.
4. Lisätään jokaista kohdan 1. hakutulosta kohti uusi lasku joka viittaa vanhaan laskuun ja annetaan laskujen numeroksi 2.

T4.

1. Etsitään laskut taulusta kaikki laskut joissa eräpäivä < nykyinen päivä, maksupäivä = NULL, ja laskun numero on 2
2. Luodaan jokaista löydettyä laskua kohti uusi lasku joka viittaa alkuperäiseen laskuun ja lisätään laskun numeroksi 3

Toteutuksesta

Toteutamme työn Java-kielisenä. Kiinnostusta olisi tehdä sovellukseen hyvin yksinkertainen graafinen käyttöliittymä, mutta emme sitoudu tähän vielä.

Luontilauseet

```
CREATE TABLE tyyppiyksikko (  
    tyyppiyksikkonumero INT NOT NULL,  
    tyyppi VARCHAR(50) NOT NULL,  
    yksikko VARCHAR(3) NOT NULL,  
    PRIMARY KEY (tyyppiyksikkonumero),  
);
```

```
CREATE TABLE osoite (  
    osoitenumero INT NOT NULL AUTO_INCREMENT,  
    katuosoite VARCHAR(100) NOT NULL,
```

```
    postinnumero INT NOT NULL,  
    postitoimipaikka VARCHAR(50) NOT NULL,  
    PRIMARY KEY (osoitenumero)  
);
```

```
CREATE TABLE tarvike (  
    tarvikenumero INT NOT NULL AUTO_INCREMENT,  
    sisanostohinta DECIMAL(10,2) NOT NULL,  
    nimi VARCHAR(150) NOT NULL,  
    varastotilanne INT NOT NULL,  
    myyntihinta DECIMAL(10,2) NOT NULL,  
    tyyppiyksikkonumero INT NOT NULL,  
    PRIMARY KEY (tarvikenumero),  
    FOREIGN KEY (tyyppiyksikkonumero) REFERENCES  
    tyyppiyksikko(tyyppiyksikkonumero)  
);
```

```
CREATE TABLE asiakas (  
    asiakasnumero INT NOT NULL AUTO_INCREMENT,  
    osoitenumero INT NOT NULL,  
    PRIMARY KEY (asiakasnumero),  
    FOREIGN KEY (osoitenumero) REFERENCES osoite(osoitenumero)  
);
```

```
CREATE TABLE henkilo (  
    asiakasnumero INT NOT NULL,  
    etunimi VARCHAR(50) NOT NULL,  
    sukunimi VARCHAR(50) NOT NULL,  
    PRIMARY KEY (asiakasnumero),  
    FOREIGN KEY (asiakasnumero) REFERENCES asiakas(asiakasnumero)  
);
```

```
CREATE TABLE yritys (  
    asiakasnumero INT NOT NULL,  
    y-tunnus VARCHAR(50) NOT NULL UNIQUE,  
    nimi VARCHAR(50) NOT NULL,  
    PRIMARY KEY (asiakasnumero),  
    FOREIGN KEY (asiakasnumero) REFERENCES asiakas(asiakasnumero)  
);
```

```
CREATE TABLE tyokohde (  
    tyokohdenumero INT NOT NULL AUTO_INCREMENT,  
    asiakasnro INT NOT NULL,  
    osoitenumero INT NOT NULL,  
    PRIMARY KEY (tyokohdenumero),  
    FOREIGN KEY (asiakasnro) REFERENCES asiakas(asiakasnumero),  
    FOREIGN KEY (osoitenumero) REFERENCES osoite(osoitenumero)  
);
```

```
CREATE TABLE urakka (  
    tyokohdenumero INT NOT NULL,  
    urakkahinta DECIMAL(10,2) NOT NULL,
```

```
PRIMARY KEY (tyokohdenumero),  
FOREIGN KEY (tyokohdenumero) REFERENCES tyokohde(tyokohdenumero)  
);
```

```
CREATE TABLE tyosuoritus (  
    tyosuoritusnumero INT NOT NULL AUTO_INCREMENT,  
    aputyö_h INT,  
    suunnittelu_h INT,  
    työ_h INT,  
    alennusprosentti INT,  
    tyokohdenumero INT,  
    PRIMARY KEY (tyosuoritusnumero),  
    FOREIGN KEY (tyokohdenumero) REFERENCES tyokohde(tyokohdenumero)  
);
```

```
CREATE TABLE lasku (  
    laskutunnus INT NOT NULL AUTO_INCREMENT,  
    päivämäärä DATE,  
    eräpäivä DATE NOT NULL,  
    maksupäivä DATE,  
    laskunumero INT NOT NULL,  
    edellinen_lasku INT,  
    asiakasnumero INT NOT NULL,  
    tyokohdenumero INT,  
    PRIMARY KEY (laskutunnus),  
    FOREIGN KEY (asiakasnumero) REFERENCES asiakas(asiakasnumero),  
    FOREIGN KEY (tyokohdenumero) REFERENCES tyokohde(tyokohdenumero)  
);
```

```
CREATE TABLE myyty_tarvike (  
    laskutunnus INT NOT NULL,  
    tarvikenumero INT NOT NULL,  
    alennusprosentti INT,  
    kappalemaara INT NOT NULL,  
    PRIMARY KEY (laskutunnus, tarvikenumero),  
    FOREIGN KEY (tarvikenumero) REFERENCES tarvike(tarvikenumero),  
    FOREIGN KEY (laskutunnus) REFERENCES lasku(laskutunnus)  
);
```

