

# Tietokantaohjelmointi- kevät 2020

Harjoitustyön dokumentaatio – vaihe 2

Ryhmä 32 – Joonas Ilvonen & Lauri Viitanen

2.5.2020

<https://tikoht-frontend.herokuapp.com/>

<https://github.com/joonasil/Tietokantaohjelmointi>

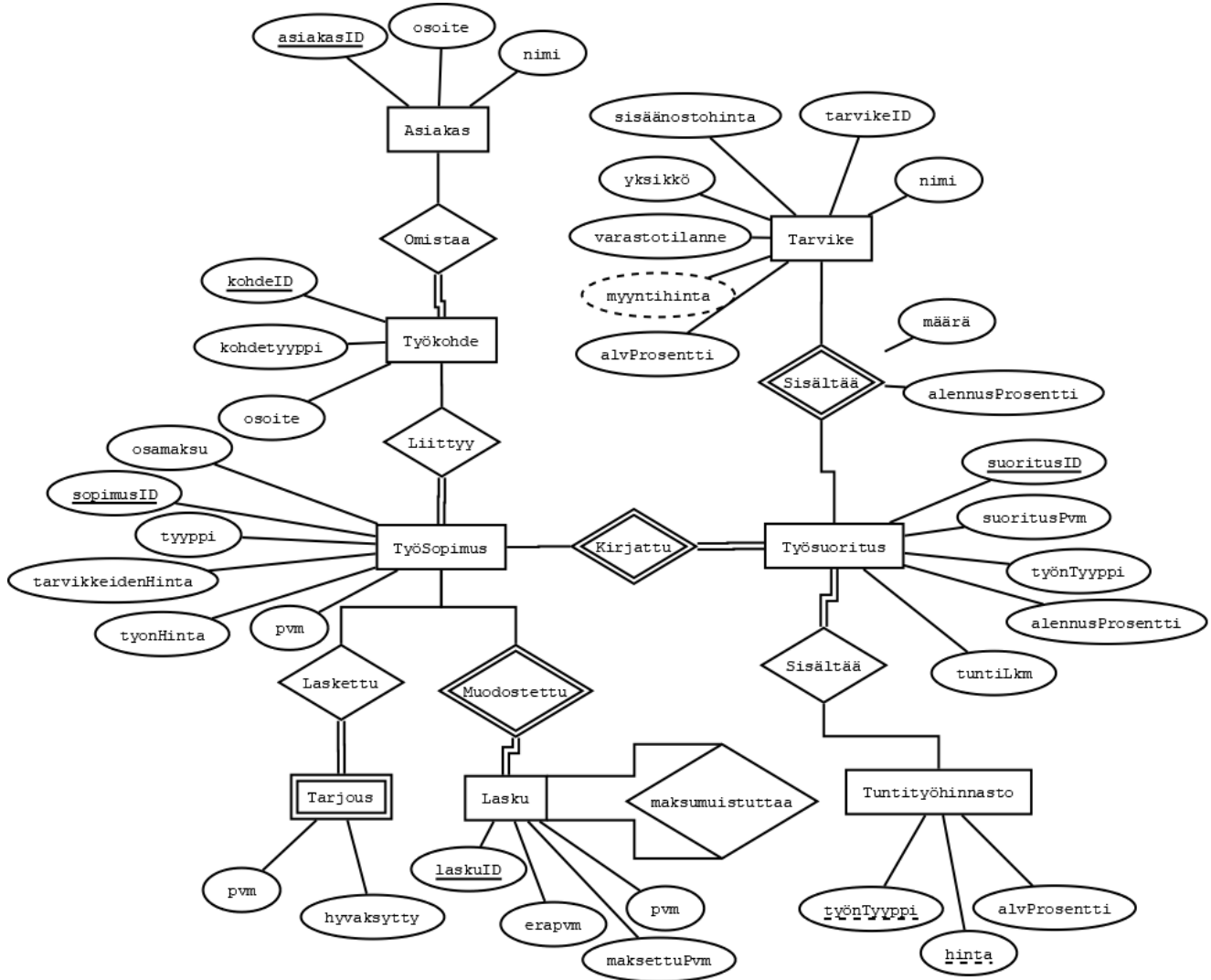
Sisältö:

1. Tehdyt oletukset
2. Tietokannan 1. vaiheen ER-kaavio
3. Tietokannan 1. vaiheen relaatiokaavio
4. Korjattu 2. vaiheen ER-kaavio
5. Tietokannan 2. vaiheen relaatiokaavio
6. Tietokannan luontilauseet
7. Johdetut tiedot
8. Tapahtumat T1-T5
9. Raportit R1-R5
10. Toteutustapa
11. Jäsenten välinen työ
12. Reflektointi ja arvio omasta työstä

## 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 1. vaiheen ER-kaavio

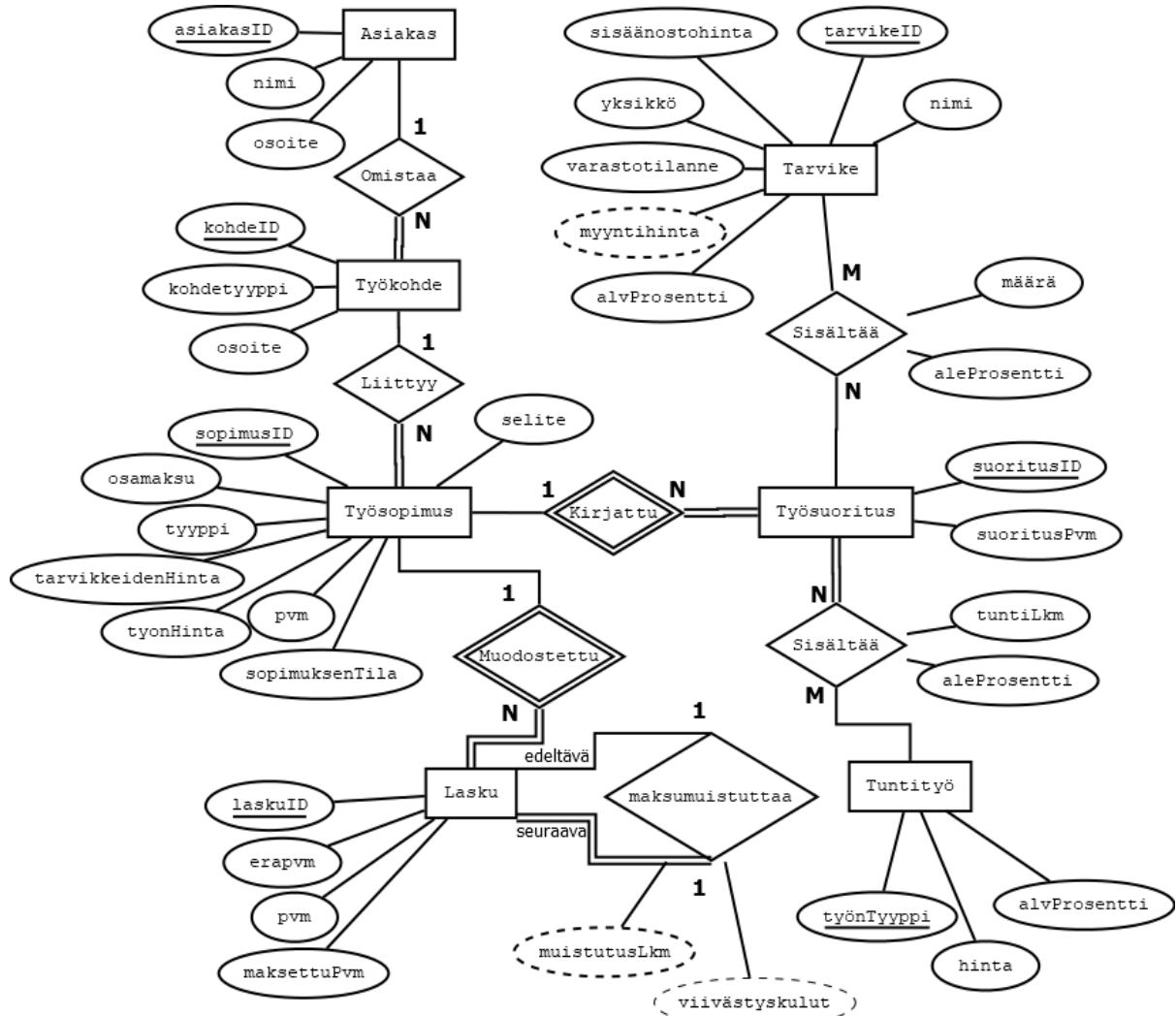


## 3. Tietokannan 1. vaiheen relaatiokaaviot

- 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 (tyyppi, hinta, alv)
- TYOSUORITUS (suoritusID, sopimusID, tyolaji, pvm, tuntiLkm, aleprosentti)
- TARVIKE (tarvikeID, nimi, sisaanostohinta, myyntihinta, yksikko, varastoLkm, alv)

- TARVIKELUETTELO (suoritusID, tarvikeID, lkm, aleProsentti)

#### 4. Korjattu 2. vaiheen ER-kaavio



Tehdyt muutokset:

ER-malli:

- Entiteetit:
  - Tarvike
    - Muutettu tarvikeID avaimeksi
  - Työsuoritus
    - Poistettu attribuutit työnTyyppi, alennusProsentti, tuntiLkm
  - Tuntityöhinnasto
    - Muutettu työnTyyppi avaimeksi
  - Tarjous
    - Poistettu koko entiteetti, sisällytetty Työsopimukseen
  - Työsopimus
    - Lisätty sopimuksenTila -attribuutti ilmaisemaan, mikäli suunnitelma, tarjous tai sopimus
  - Tuntityö

- Nimi muutettu (Tuntityöhinnasto -> Tuntityö)
  - Lasku
    - Lisätty muistutuslkm -attribuutti, lisätty viivästyskulut -attribuutti
- Suhteet
  - Sisältää (Tarvike – Työsuoritus)
    - Muutettu tavanomaiseksi
  - Sisältää (Työsuoritus – Tuntityö)
    - Lisätty attribuutit alennusProsentti ja tuntiLkm
  - Maksumuistuttaa
    - Lisätty johdetut attribuutit muistutusLkm ja viivästyskulut

Muut:

Palautteessa ehdotettu, että voisi olla luontevampaa linkittää lasku työsuoritukseen sopimuksen sijasta, koska sopimus voi sisältää monta suoritusta. Lasku jätetty kuitenkin linkitettyinä työsuoritus sopimukseen, sillä näin on ajateltu mahdollistettavan se, että useampi työsuoritus samalta sopimukselta laskutetaan kerralla. Esimerkiksi työntekijän tehdessä useampana päivänä samassa kohteessa töitä.

Lisätty kardinaalirajoitukset suhteisiin.

SQL-lauseet:

- Muokattu vastaamaan ER-mallia muutosten jälkeen.
- ID attribuuttien SERIAL tyypit muutettu INT, jotta ei vahingossa syötetä DEFAULT arvolla samaa dataa useampaan kertaan.

## 5. Tietokannan 2. vaiheen relaatiokaavio

- ASIAKAS (asiakasID, nimi, osoite)
- TYOKOHDE (kohdeID, omistajaID, kohdetyyppi, osoite)
- TYOSOPIMUS (sopimusID, kohdeID, tyyppi, työnHinta, tarvikkeidenHinta, osamaksu, pvm, selite, sopimuksenTila, sopimuksenSumma)
- LASKU (laskuID, sopimusID, pvm, erapaiva, maksettuPVM, edeltavaLasku, muistutuslkm, viivästyskulut)
- TUNTITYO (tyyppi, hinta, alv)
- TYOSUORITUS (suoritusID, sopimusID, suorituspvm)
- TYOSUORITUKSENTUNTITYO (suoritusID, aleprosentti, tyontyyppi, tuntilkm)
- TARVIKE (tarvikeID, nimi, sisaanostohinta, myyntihinta, yksikko, varastoLkm, alv)
- TARVIKELUETTELO (suoritusID, tarvikeID, lkm, aleprosentti)

## 6. Tietokannan luontilauseet

Annettu erillisessä tiedostossa: Ryhma32-HT-vaihe2\_SQL\_luontilauseet\_ja\_esimerkkidata.pdf

## 7. Johdetut tiedot

Sopimuksen ollessa tuntipohjainen haetaan kaikki sille kirjatut työsuoritukset ja kaikki työsuorituksiin kirjatut tarvikeluettelot. Näistä pystytään summaamaan kokonaismäärä työlle ja tarvikkeille, sekä tarvittaessa muotoilemaan erittely laskulle esimerkiksi päiväkohtaisesti. Työsopimuksissa näytettävistä tiedoista sopimuksen summa ja alv-määrä johdetaan laskemalla muista tiedoista.

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.

## 8. Tapahtumat T1-T5

T1:

- Siirrytään välilehdelle Työkohde
- Syötetään "Lisää tai muokkaa entiteettiä" lomakkeeseen omistajan id, kohdetyyppi ja osoite
- Painetaan Lisää-painiketta

T2:

- Tarkistetaan työkohde välilehdeltä tyokohteen id
- Siirrytään työsopimus välilehdelle, josta katsotaan kyseisen kohteen sopimuksen id
- Siirrytään työsuoritus välilehdelle, jossa lisätään uusi kirjaus sopimukseen
- Siirrytään työsuorituksentuntityö-välilehdelle, jossa lisätään työsuoritukseen tunteja
- Siirrytään tarvikeluettelo-välilehdelle, jossa lisätään suorituksen tuotteet

T3

- Siirrytään Laskut välilehdelle
- Siirrytään kohtaan "Hallitse Laskuja"
- Jos halutaan tarkastaa erääntyneet laskut, joista ei ole vielä lähetetty muistutusta ja joita ei ole maksettu, painetaan "Näytä erääntyneet laskut".
- Luodaan muistutuslaskut painamalla "Luo muistutuslaskut erääntyneistä" -painiketta
  - Laskuihin lisätään automaattisesti viivästyskulut

T4

- Sama kuin T3.
  - Jos kyseessä on kolmas muistutus, lasketaan sen viivästyskuluihin korko ja maksulisä

T5

- Ei toteutettu

## 9. Raportit R1-R5

R1:

- Siirrytään työkohde, työsopimus, lasku tai tarvike välilehdille
- Siirrytään Laske Hinta-arvio kohteelle-lomakkeeseen
- Täytetään kohdeid ja lisätään halutu tuoterivit ja palvelut painamalla kyseisiä painikkeita
- Kun halutut tuotteet ja palvelut on lisätty, nähdään laskettuna Hinta-arvion kokonaissumat ja muut tiedot.
- Luotaan uusi tarjous painamalla "Muodosta urakka-tarjous"-painiketta

R2:

- Siirrytään välilehdille työsopimus tai lasku

- Syötetään ”lasku sopimuksesta” -lomakkeelle
- Syötetään halutun sopimuksen id, josta halutaan muodostaa lasku
- Painetaan Lasku sopimuksesta-painiketta
- Lasku tulee sivulle näkyviin lomakkeen alapuolelle, josta se voidaan tulostaa

R3

- Siirrytään tarvikeluettelo-välilehdelle ja annetaan työsuorituksen tuotteelle alennus
  - Joko lisäämällä kirjaus tai muokkaamalla valmiiksi olevaa
- Siirrytään työsuoituksetuntityo-välilehdelle ja annetaan työsuorituksen työsuoritteelle alennus
  - Joko lisäämällä kirjaus tai muokkaamalla valmiiksi olevaa
- Siirrytään joko työsopimus- tai lasku -välilehdille
- Täytetään muodosta lasku sopimuksesta -lomakkeeseen sopimuksen id, jolle on kirjattu työsuorituksia ja tarvikeluetteloita
- Lasku tulee näkyviin, se voidaan tulostaa selaimen tulostusominaisuudella

R4

- Siirrytään työkohde, työsopimus, lasku tai tarvike välilehdille
- Siirrytään Laske Hinta-arvio kohteelle-lomakkeeseen
- Täytetään kohdeid ja lisätään halutu tuoterivit ja palvelut painamalla kyseisiä painikkeita
- Kun halutut tuotteet ja palvelut on lisätty, nähdään laskettuna Hinta-arvion kokonaissumat ja muut tiedot.
- Luotaan uusi tarjous painamalla ”Muodosta urakka-tarjous”-painiketta

R5

- Ei toteutettu

## 10. Toteutustapa

Sovellus toteutettu Spring-boot backendin ja React frontend-sovelluksen yhdistelmällä. Backend ottaa yhteyden tietokantaan ja välittää tietoa REST-apin kautta frontend-sovellukselle. Nämä on Mavenilla pakattu yksittäiseksi JAR-tiedostoksi, joka on laitettu serverille pyörimään.

React-app tekee http-pyyntöjä Spring-Boot sovellukselle, joka hoitaa tietokantakyselyt JDBC kirjastolla. Suurin osa laskennasta tapahtuu backend-sovelluksessa. Frontend tekee joitakin yksinkertaisia laskutoimituksia, mutta lähinnä asettelee haetut tiedot näkyville.

## 11. Jäsenten välinen työnjao

Vaihe1:

Yhdessä: Tietokannan suunnittelu, er-kaavio, sql ja dokumentaatio

Vaihe2:

Yhdessä: Ohjelman toimivuuden suunnittelu ja ominaisuuksien miettiminen.

Joonas: Springboot-Backend

## 12. Reflektointi ja oma arvio työstä

### Lauri:

Valitettavasti aloitin harjoitustyön toisen vaiheen tekemisen vasta reilu viikko ennen palautusajankohtaa. Toteutustavaksi valitsimme yhdessä modernin FullStack sovelluksen, jossa käyttäisimme backendin toteutukseen Spring-boot frameworkiä ja frontend sovellukseen reactia.

Itselläni ei ollut juurikaan kokemusta kummastakaan, joten alkuun jouduin käyttämään useamman päivän ymmärtääkseni, kuinka full stack ja spring boot toimii. Tämän jälkeen viimeiset 1,5 viikkoa meni frontend-sovelluksen kirjoittamiseen. Haasteita tuli jälleen heikkojen ennakkotietojen takia. Lisäksi alkuun sovellusta ei oltu suunniteltu kunnolla, vaan siihen lisättiin ominaisuuksia sitä mukaan kun kerrettiin. Tämä aiheutti myöhemmin ongelmia, kun koodin ylläpidettävyyks alkoi kärsimään ja lopulta kiireen takia kirjoitin sen hyvin epäpuhtaasti.

Työ on visuaalisesti toimiva. Toiminnallisuuksiltaan se on suppea, eikä vastaa modernin käyttöliittymän kaikkia suunnitteluperiaatteita. Kuitenkin suurimmilta osiltaan se toimii tietokannan web-pohjaisena ohjaimena ja jotenkuten kuvitteellisen toiminimen laskutusjärjestelmänä.

Olin halunnut hioa frontendiä vielä pidemmälle ja lisätä siihen ominaisuuksia, kuten kirjautusmissivun ja käyttöoikeuksien hallinnan. Nämä kuitenkin jäivät pois aikasyistä, varsinkin kun työ on valmiiksi myöhässä.

Kaikkia ominaisuuksia ei ole keretty hiomaan aivan loppuun asti. Tietokantaan olisi ollut hyvä lisätä näkymiä, joita voisi hyödyntää nettisivulla.

Valitettavasti suurin osa ajasta kului tietokantaohjelmoinnin kannalta epärelevantteihin asioihin, eli JavaScriptin ja Reactin opetteluun ja niiden ongelmien ratkontaan. Pelkästään ”valmiin sovelluksen” verkkoon julkiseksi asettaminen vei pari päivää, sillä sekin oli täysin uusia asia opetella. Valittu toteutustapa ei tässä tapauksessa ollut työn tarkoituksen kannalta optimaalinen. Työn 2. vaiheeseen käytetty n. 115h.

### Joonas:

Ryhmän välinen kommunikaatio tippui todennäköisesti muiden kiireiden takia noin kuukaudeksi minkä takia toisen vaiheen toteutus jäi hieman viimetippaan. Olimme alun perin suunnitelleet tekemämme työn javalla ja halusimme tehdä jonkinlaisen graafisen käyttöliittymän sovelluksen käytön yksinkertaistamiseksi.

Olin ehdottanut ensimmäiseksi vaihtoehtoksi Javan omaa JavaFx kirjastoa, jota olin käyttänyt ennenkin, mutta se olikin poistettu uusimmista JRE versioista ja sen lisäämisessä oli hieman ongelmia, joten päätin että tehdään sovellus vähän tulevaisuuden työelämää enemmän hyödyttävänä Spring Boot MVC sovelluksena. Backendin kanssa ei ollut kovin suuria ongelmia olin ennakkin tehnyt MVC mallin mukaisen sovelluksen Java Springin avulla ja Spring Boot on tehty helpommaksi pistää pystyyn.

Aika paljon meni toisen vaiheen alussa aikaa taas javan ja mavenin kertaamiseen, kun niiden käyttämisestä on mennyt muutamia vuosia aikaa. Frontendiä ei oltu kumpikaan oikeastaan tehty aikaisemmin joten sen kanssa tuli eniten ongelmia. Meille ehdotettiin käyttää Reactia ja rupesimme molemmat opettelemaan React frontin tekoa ja lopulta Lauri sai sen toimimaan, mutta koska siinä

vaiheessa oli jo deadline käytännössä saavutettu tuli frontin koodista kiireessä sen verran vaikeasti ylläpidettävä että en pystynyt auttamaan sen kirjoittamisessa oikeastaan ollenkaan ja näin ollen se jäi kokonaan Laurin tehtäväksi.

Meidän olisi pitänyt käyttää enemmän aikaa sovelluksen suunnitteluun sillä nyt suurin osa ajasta kului uusien ominaisuuksien toteuttamiseen vanhan koodin päälle ja niistä aiheutuvien ongelmien korjaamiseen. Olen ihan tyytyväinen työstä jonka saimme tehtyä ja olosuhteisiin nähden Lauri sai todella hyvän käyttöliittymän kasattua sovellukseen. Olisin halunnut vielä toteuttaa laskujen datan keräyksen paremmin backendin puolella koska nyt laskujen esitys ei ota huomioon osamaksuja eikä niissä näy mahdollisia viivästyskuluja.