

# Tietokantasovellus: Tavarantvaihto harjoitustyö

# 1. Johdanto

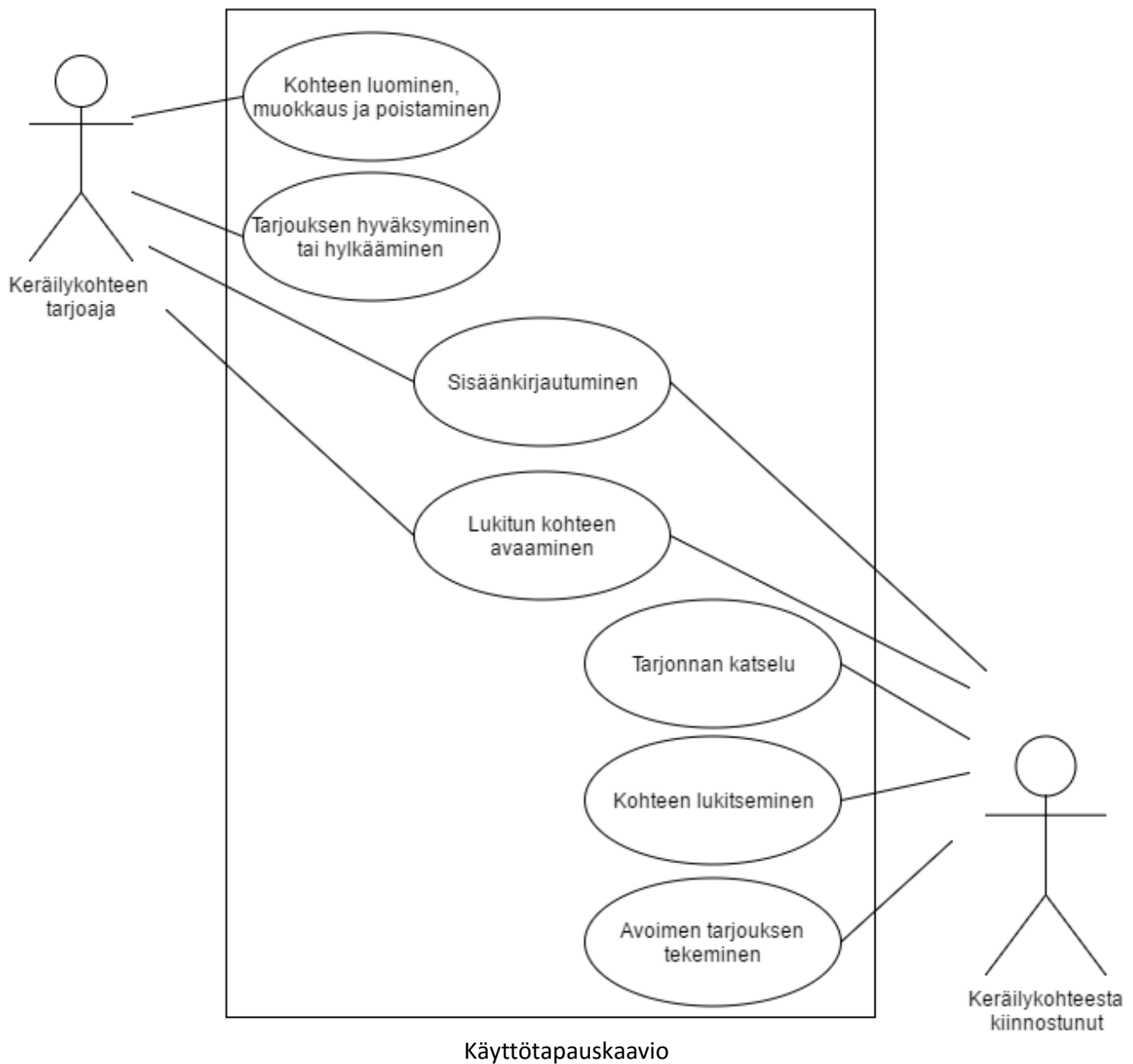
## Järjestelmän tarkoitus

Järjestelmän tarkoitus on tarjota keräilijöille keräilykohteiden vaihtoa helpottava sovellus. Sovellukseen voi tarjota vaihtoon keräilykohteita. Kohteen tarjoaja voi joko esittää toiveen tilalle saatavasta kohteesta tai ottaa vastaan pelkästään avoimia tarjouksia. Avoimia tarjouksia voi kuitenkin tehdä myös kohteista, joissa on esitetty toive tilalle saatavista kohteista. Jos tilalle saatava kohde on määritelty voi tarjontaa tutkiva jäsen lukita tarjouksen. Tarjoaja ja lukon asettaja voivat molemmat poistaa lukon halutessaan. Tarjottavana voi olla erityyppisiä kohteita.

## Toteutus-/toimintaympäristö

Sovellus toteutetaan laitoksen user-spalvelimella, Apache-palvelimen alla. Web-sovelluksen alustajärjestelmän tulee tukea PHP ohjelmointikieltä. Tietokannaksi on valittu postgreSQL.

## 2. Yleiskuva järjestelmästä



### Käyttäjärühmät

#### Jokamies

Jokamies näkee sisäänkirjautusmissivun ja etusivun, jolta ohjataan kirjautumaan sisään. Lisäksi jokamies voi katsella tarjontaa ja yksittäisiä kohteita, mutta ei voi lähettää tarjouksia.

#### Käyttäjä

Käyttäjä on palveluun rekisteröitynyt henkilö, joka voi selailla tarjontaa, lähettää tarjouksia ja asettaa omia kohteita vaihdettavaksi.

#### Keräilykohteen tarjoaja

Käyttäjä, joka on asettanut kohteen tarjolle vaihtoon.

#### Keräilykohteesta kiinnostunut

Käyttäjä joka on lähettää tarjouksen jostain tarjolla vaihtoon olevasta kohteesta.

## Käyttötapauskuvaukset

### Keräilykohteen tarjoaja:

#### Kohteen luominen, muokkaaminen ja poistaminen

Käyttäjä voi lisätä järjestelmään oman kohteen ja oman hallintasivun kautta muokata ja poistaa lisäämiään kohteita. Kohteelle annetaan nimi, kuvaus ja mahdollinen vastatarjoustoive. Lisäksi kohdetta luodessa sille voi määritellä tunnisteita, joita ei kuitenkaan voi toistaiseksi jälkikäteen muokata tai poistaa.

#### Tarjouksen hyväksyminen ja hylkääminen

Käyttäjä näkee listauksen saaduista tarjouksista, jotka hän voi hyväksyä tai hylätä. Hyväksymällä tarjouksen hylkää automaattisesti muut kohteesta tulleet tarjoukset. Hylätyn tarjouksen voi poistaa sekä lähettäjä että vastaanottaja. Kun tarjous on hyväksytty ja vaihdosta sovittu, voi kaupan merkitä onnistuneeksi, jolloin kohde poistuu vaihtojärjestelmästä.

### Keräilykohteesta kiinnostunut:

#### Tarjonnan katselu

Käyttäjä näkee listauksen kaikista tarjolla olevista kohteista, joiden kohdalla vaihtokauppaa ei ole merkitty onnistuneeksi. Listauksesta käyttäjä voi valita kiinnostavan kohteen ja tutkia omalla näkymäsivulla sen tietoja tarkemmin.

#### Avoimen tarjouksen tekeminen

Jokaisesta avoinna olevasta kohteesta voi lähettää avoimen tarjouksen eli vapaamuotoisen viestin kohteen omistajalle. Tämän tarjouksen kohteen omistaja voi joko hyväksyä tai hylätä.

#### Kohteen lukitseminen

Jos kohteen tarjolle asettanut on määrittänyt tarkemmin, mitä haluaa vaihdossa, voi kohteen lukita suoraan. Esimerkiksi jos kolikosta halutaan vaihdossa postikortteja, voi postikortteja omistava lukita kohteen ja tarjota näin automaattisesti postikortteja vaihdossa kolikosta.

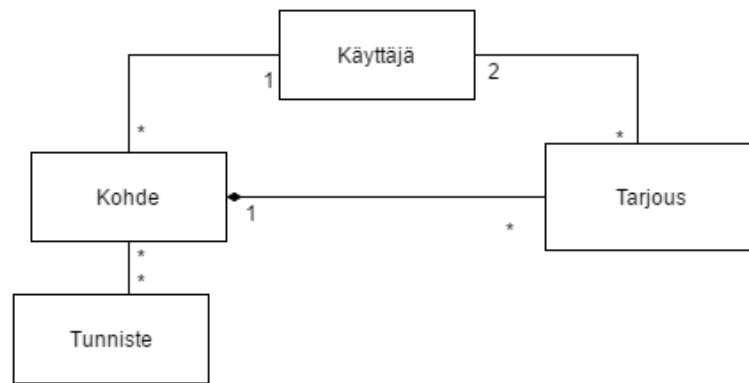
### Molemmat:

#### Sisäänkirjautuminen

#### Lukitun kohteen avaaminen

Kun kohde on lukittu, voi sekä kohteen omistaja että kohteen lukitsija avata asetetun lukon, jolloin kohde vapautuu jälleen vaihtokauppaan. Kun kohde on lukittuna, ei muut käyttäjät voi tarjota siitä myöskään avoimia tarjouksia.

### 3. Järjestelmän tietosisältö



Käsitekaavio

#### Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku	Käyttäjän id
Nimi	merkkijono	Käyttäjän nimi
Sähköpostiosoite	merkkijono	Käyttäjän sähköpostiosoite
Salasana	merkkijono	Käyttäjän salasana

Käyttäjä voi asettaa tarjolle vaihtoon useampia kohteita, mutta yhdellä kohteella voi olla vain yksi käyttäjä eli sen omistaja. Yksi käyttäjä voi myös lähettää useita tarjouksia ja yhteen tarjoukseen liittyy aina kaksi käyttäjää, lähettäjä ja vastaanottaja.

#### Tietokohde: Kohde

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku	kohteen id
Nimi	merkkijono	Kohteen nimi
Kuvaus	merkkijono	Kohteen kuvaus
Status	avoinna, lukittu, valmis	Kohteen status
Omistaja	kokonaisluku	Omistajan id
Vastatarjous	merkkijono	Kuvaus, mitä kohteesta halutaan vastatarjouksena takaisin
Lisätty	päivämäärä	Milloin kohde on lisätty

Yhdellä kohteella voi olla vain yksi omistaja eli käyttäjä, mutta yhdellä käyttäjällä voi olla useampi kohde tarjolla vaihtoon. Kohteeseen voi liittyä useampia tarjouksia, mutta tarjouksella on aina yksi kohde, johon ko. tarjous liittyy. Kohteeseen voi liittyä useampia tunnisteita ja yhteen tunnisteeseen voi liittyä useita kohteita.

### *Tietokohde: Tarjous*

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku	Viestin id
Lähtettäjä	kokonaisluku	Lähtetäjän id
Vastaanottaja	kokonaisluku	Vastaanottajan id
Viesti	merkkijono	Tarjouksen sisältö
Kohde id	kokonaisluku	Tarjoukseen liittyvän kohteen id
Tyyppi	avoin tarjous, vastatarjous	Tarjouksen tyyppi
Lähetetty	päivämäärä	Milloin viesti on lähetetty

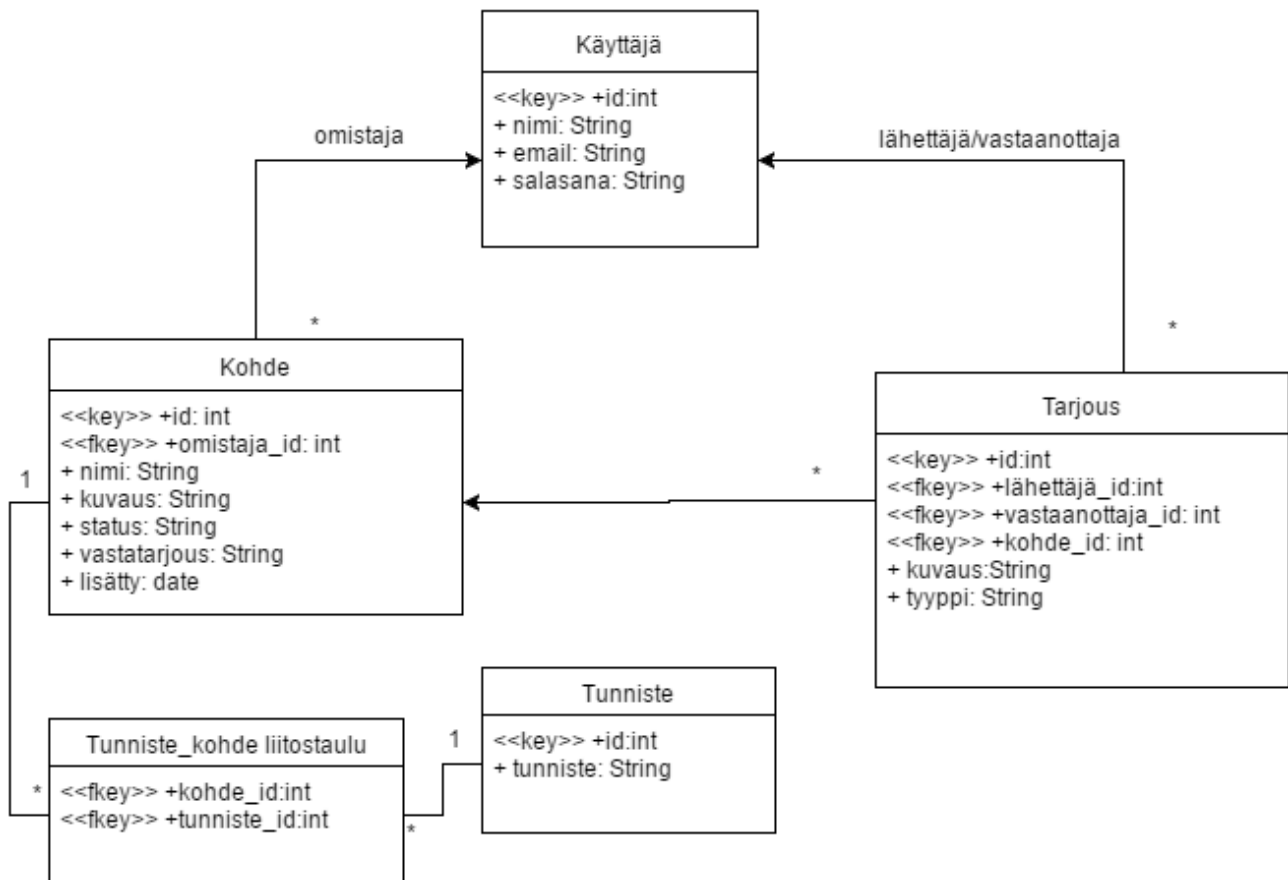
Yhdellä tarjouksella on aina kaksi käyttäjää eli lähettäjä ja vastaanottaja. Yhdellä käyttäjällä voi olla useampia tarjouksia. Tarjoukseen liittyy aina yksi kohde, mutta samaan kohteeseen voi liittyä useampia tarjouksia.

### *Tietokohde: Tunniste*

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku	Tunnisteen id
tunniste	merkkijono	Tunnisteen nimi (esim. antiikki)

Yksi tunniste voi liittyä useaan kohteeseen ja yhdellä kohteella voi olla monta eri tunnistetta.

#### 4. Relaatiotietokantakaavio



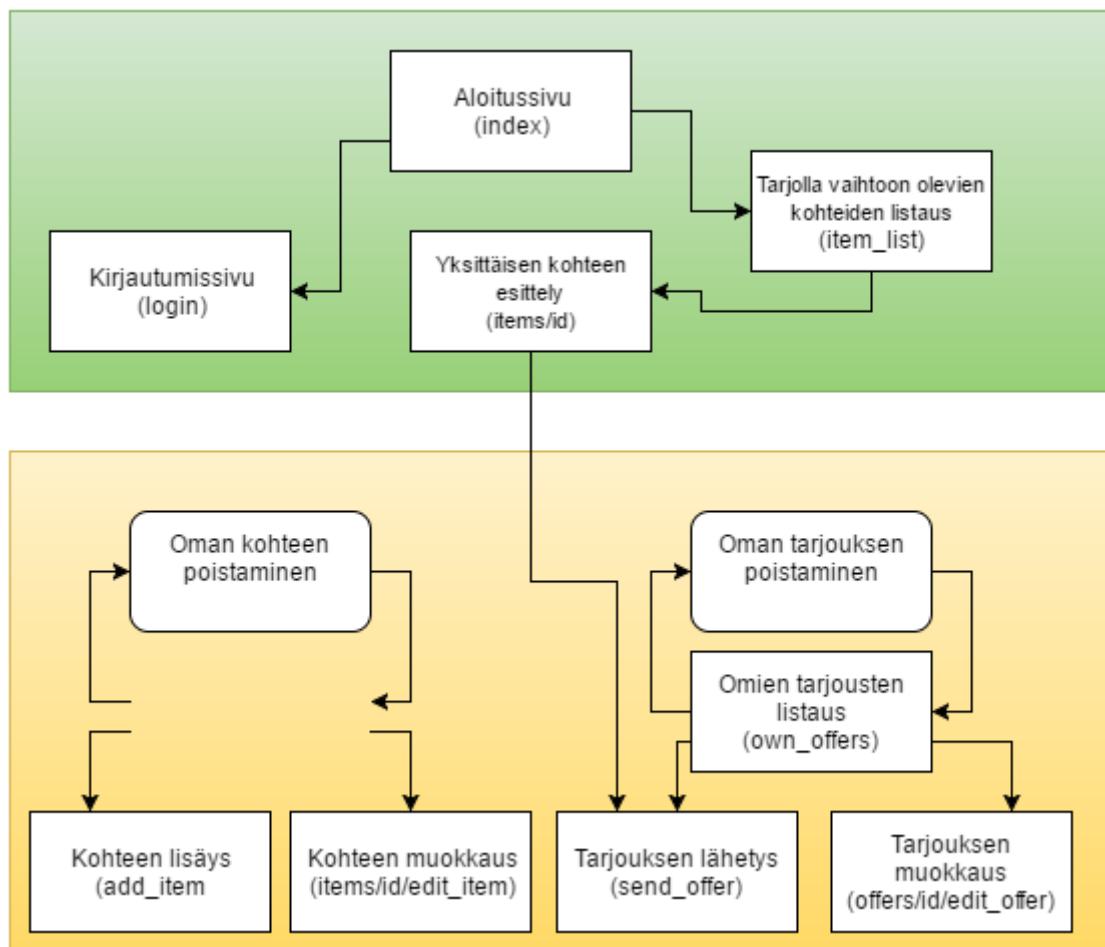
Relaatiotietokantakaavio

## 5. Järjestelmän yleisrakenne

Hakemistossa app on jaoteltu MVC-mallin mukaan kontrollerit hakemistoon controllers, mallit hakemistoon models ja näkymät hakemistoon views. Tässä hakemistossa näkymät on jaoteltu vielä erikseen omiin hakemistoihinsa kokonaisuuksien (items, offers, users) perusteella.

Hakemistosta assets löytyy css ja javascript tiedostot, hakemistosta config löytyy tietokanna, ympäristön ja eri reittien tiedot, hakemistosta doc sovelluksen dokumentaatio, hakemistosta lib tarvittavat pohjat php-tiedostoille, hakemistossa sql tarvittavat tiedostot tietokannan luomiseen, testidatan asettamiseen sekä tuhoamiseen. Kaikki tiedostot on nimetty pienellä alkukirjaimella.

## 6. Käyttöliittymän ja järjestelmän komponentit



Sivustolla on käytössä navigaatiopalkki, joten liikkuminen etusivun, tarjolla vaihtoon olevien kohteiden, omien kohteiden listauksen, omien tarjousten listauksen sekä kirjautumissivun välillä on vapaata. Kohteiden lukitseminen toimii samalla tavalla kuin tarjouksen lähetys, joten sitä ei



## 7. Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon. Aseta sen jälkeen config/enviroment.sh tiedostoon oma käyttäjätunnus ja sekä haluttu kansion nimi ja config/database.php tiedostoon tietokannan tiedot. Tietokantataulut voi luoda ajamalla create\_tables.sh tiedoston ja testidataa sovellukseen saa ajamalla add\_test\_data.sh tiedoston. Sovellus käynnistyy ja muutokset voi lisätä ajamalla deploy.sh tiedoston.

## 8. Käynnistys- / Käyttöohje

Työ löytyy osoitteesta: <http://jknurmi.users.cs.helsinki.fi/tsoha/>

Sisäänkirjautuminen suoraan: <http://jknurmi.users.cs.helsinki.fi/tsoha/login>

Sovellusta voi testata useammalla käyttäjällä, jotta tarjousten lähettämistä ja vastaanottamista on helpompi tarkastella.

**Käyttäjätunnus:** Kalle Keräilijä

**Salasana:** Keräilijä1

**Käyttäjätunnus:** Hanna Hamstraaja

**Salasana:** Hamstraaja1

**Käyttäjätunnus:** Veikko Valitsija

**Salasana:** Valitsija1

## 9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

### Testaus

Ohjelmaa on testattu lähinnä kokeilemalla eri toimintoja itse sivustolla. Sovellusta on kokeillut kehittäjän lisäksi myös toinen henkilö. Koodipuolella testausta ei ole erikseen tehty sen enempää kuin mitä valmiissa pohjassa oli.

### Tunnetut bugit

Tiedossa ei ole, että sovelluksessa olisi mainittavia bugeja.

### Puutteet & jatkokehitysideat

- Tällä hetkellä sovellus luottaa, että kohteen tarjoaja ja kohteesta kiinnostunut sopivat lopulta kaupan yksityiskohdat sähköpostitse. Käyttäjäystävällisempää olisi kuitenkin kehittää tarjoussysteemiä niin, että sovelluksessa voisi käydä enemmänkin viestin vaihtoa ja hoitaa neuvottelut kokonaan siinä.
- Tunnisteita voi tällä hetkellä lisätä ainoastaan kohdetta luodessa. Olisi hyvä, jos tunnisteita voisi muokata, poistaa ja lisätä samalla kuin muokkaa kohdetta.
- Tunnisteiden avulla voisi jatkossa kehittää sovellukseen listauksen tunnisteiden perusteella tai jonkinlaisen haun.
- Kohteisiin voisi myöhemmin lisätä myös kuvan, mutta se vaatisi muutoksia koodin lisäksi myös tietokantaan.

## 10. Omat kokemukset

Tietokantasovelluksen tekemisessä helppoa oli itse koodaaminen, kun vauhtiin pääsi. Hankalaa taas oli suunnittelu ja suurimmat ongelmat tulivatkin vastaan siinä, että tietokannan rakenne oli aluksi suunniteltu vähän sinnepäin. Niinpä jos tekisin työn uudestaan, panostaisin ehdottomasti enemmän ensimmäisillä viikoilla tietokannan rakenteen ja sovelluksen toiminnan miettimiseen jo ennen kuin lähden sitä muuten toteuttamaan. Tietokantasovellusta tehdessä opin, miten MVC-malli toimii (aluksi pidin sitä työläänä ja monimutkaisena, mutta lopuksi tajusin, kuinka se oikeastaan selkeyttä paljon koko hommaa). Myös SQL-kyselyiden tekeminen yms. palautui hyvin mieleen, vaikka aluksi olikin hieman hankalaa, sillä tikapen käymisestä oli jo yli vuosi.