

Aki Mäkitalo

TANSSIURHEILUSEURA OIKEA RYTMİ RY:N  
JÄSENREKISTERISOVELLUKSEN TOTEUTUS JA  
KÄYTTÖÖNOTTO

Tietotekniikan koulutusohjelma  
Ohjelmistotekniikan suuntautumisvaihtoehto  
2012

# TANSSIURHEILUSEURA OIKEA RYTMİ RY:N JÄSENREKISTERISOVELLUKSEN TOTEUTUS JA KÄYTTÖÖNOTTO

Mäkitalo, Aki  
Satakunnan ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Joulukuu 2012  
Ohjaaja: Aarinen, Reino  
Sivumäärä: 59

Asiasanat: Qt, ohjelmointi, relaatiotietokannat, MySQL, jäsenrekisteri

---

Tämän opinnäytetyön tavoitteena oli ensin suunnitella ja sitten toteuttaa Tanssiurheiluseura Oikea Rytmī ry:lle uusi jäsenrekisterisovellus korvaamaan vanhan Excel-ohjelmalla toteutetun jäsenrekisterin ylläpidon.

Uusi jäsenrekisterisovellus sisältää MySQL-tietokannan ja itse ohjelman. Ohjelma ottaa tietokantaan yhteyttä ja käyttää ja ylläpitää sen tietoja. Tietokantoja on vain yksi, ja se sijaitsee internetissä Nebulan ylläpitämässä web-hotelli palvelimessa. Itse ohjelma voi olla useamman henkilön käytössä yhtä aikaa.

Tietokanta toteutettiin käyttäen MySQL-tietokantaa ja itse ohjelma käyttäen Qt-kehitysympäristöä ja siihen sisäänrakennettua C++-ohjelmointikieltä.

Oikea Rytmī ry:n käyttämä vanha jäsenrekisteri toimi Excelissä. Nyt kehitetty uusi järjestelmä sisältää käyttöä helpottavia ominaisuuksia kuten monipuolisia etsimistöimintoja, mahdollisuuden tulostaa laskut automaattisesti, erilaisia raporttien tuloksia ja muita käyttöä helpottavia automaatioita.

Uuden sovelluksen hyvänä puolena on tietokannan sijaitseminen internetissä. Näin sovelluksen ei tarvitse huolehtia versionhallinnasta, koska sovellus hakee aina uusimmat tiedot. Vanhassa Excel-tyylisessä ratkaisussa tiedot olivat monessa eri tiedostossa. Tämä teki versionhallinnasta vaikeaa, kun tiedostoja lähetettiin jatkuvasti sähköpostitse eri käyttäjien välillä.

Työn tuloksena kehitettiin toimiva jäsenrekisterisovellus, joka täytti asiakkaan vaatimukset ja joka helpottaa jäsentietojen ylläpitoa.

# THE IMPLEMENTATION AND DEPLOYMENT OF A MEMBER REGISTRY APPLICATION FOR OIKEA RYTMI.

Mäkitalo, Aki

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technology

December 2012

Supervisor: Aarinen, Reino

Number of pages: 59

Keywords: Qt, programming, relational databases, MySQL, member registry

---

The purpose of this thesis was to first design and then to implement a new member registry application for the dance team Oikea Rytmii ry, to replace the old Excel-style method previously used.

The new member registry uses MySQL-database, and the program itself. The program contacts the database and utilises the information stored within. There is only one database, and it is on the internet running on a web-host server maintained by the provider Nebula. The program can be used by multiple persons simultaneously.

The database was implemented by using MySQL-database and the program itself was made using Qt-development environment and the inbuilt C++-programming language.

The old member registry used by Oikea Rytmii ry, was merely an Excel-file. The new system includes multiple features that assist the user, such as versatile search functions, the possibility to print bills automatically, versatile report printing possibilities and multiple other automations to improve usage.

In addition, one good benefit of the new application is the way the database is stored on the internet. Because of this, the user does not need to worry about version management, because the application always retrieves the newest version of the data from the internet. In the old Excel-style solution, data was stored in multiple different files spread between multiple different users. This made version management difficult, because files had to be constantly sent between users through email.

The result of this project was a functioning member registry, which fulfilled the requirements of the customer, and which should make it easier to maintain member information.

# SISÄLLYS

LYHENNE- JA TERMILUETTELO .....	6
1 YHDISTYSESITTELY .....	7
2 TYÖN JOHDANTO.....	7
2.1 Työn tausta ja käyttötarkoitus.....	7
2.2 Valittu ohjelmointikieli ja tietokanta .....	8
3 QT-OHJELMOINTIKIELI .....	9
3.1 Perusteet ja historiaa .....	9
3.2 Qt:lla ohjelmointi .....	10
3.3 Miksi Qt? .....	11
3.4 Qt:n lisenssi.....	11
4 RELAATIOTIETOKANNAT.....	13
4.1 Yleistä relaatiotietokannoista.....	13
4.2 SQL-kieli.....	15
4.2.1 SELECT .....	16
4.2.2 INSERT .....	16
4.2.3 UPDATE .....	16
4.2.4 DELETE .....	17
4.3 Tietotyypit.....	17
4.4 MySQL-esittely.....	18
5 SUUNNITTELU .....	18
5.1 Suunnitteluprosessi .....	18
5.2 Ohjelman suunnittelu .....	19
5.3 Tietokannan suunnittelu.....	19
5.4 Käyttöliittymän suunnittelu .....	20
5.5 Testaus .....	20
6 TYÖKALUT .....	21
6.1 Toteutusympäristö.....	21
6.2 Tietokannan asennus .....	21
6.3 Qt-ohjelmointi.....	22
6.3.1 Qt Creatorin asennus ja siihen MySQL tuki.....	22
6.4 MySQL Workbench.....	22
7 TOTEUTUS .....	23
7.1 Tietokannan toteutus .....	23
7.2 Taulut .....	23
7.2.1 Jaskaus .....	23

7.2.2	Jasenrooli .....	24
7.2.3	Jasenryhma .....	24
7.2.4	Jasentieto .....	25
7.2.5	Kausi .....	25
7.2.6	Rooli .....	26
7.2.7	Ryhma .....	26
7.2.8	Yhteydet .....	26
7.3	Käyttöliittymä Qt Creatorilla .....	27
7.4	Ohjelmalogiikan toteutus .....	29
8	TYÖN ESITTELY JA TULOKSET .....	30
8.1	Ohjelman yleiset ominaisuudet .....	30
8.2	Tiedostot .....	32
8.3	Toiminnot .....	33
8.4	Raportit .....	49
8.5	Laskut .....	52
8.6	Ohje .....	56
9	YHTEENVETO .....	57
9.1	Yleistä .....	57
9.2	Ongelmia työn aikana .....	57
9.3	Muutoksia suunnitelmassa .....	58
9.4	Kehitysideat tulevaisuudessa .....	58
	LÄHTEET .....	59

## LYHENNE- JA TERMILUETTELO

Mysql	MySQL AB:n kehittämä suosittu relaatiotietokanta.
Qt	Alustariippumaton ohjelmistojen ja graafisten käyttöliittymien kehitysympäristö.
C++	Yksi yleisimmistä ohjelmointikielistä. Tällä kielellä on kirjoitettu esimerkiksi suurin osa maailman käyttöjärjestelmistä.
Relaatiotietokanta	Nykyaikana ylivoimaisesti yleisin käytetty tietokantamalli. Tiedot ovat tauluissa, jotka sisältävät rivejä. Rivit koostuvat kentistä, jotka sisältävät tietoa. Käyttää SQL-kieltä.
PHP	PHP on ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luonnissa
SQL	Structured Query Language on IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä. Käytännössä kaikki relaatiotietokannat ymmärtävät SQL-kieltä.
Webhotelli	Webhotelli on palvelu, jossa asiakas vuokraa palveluntarjoajan www-palvelimelta kiintolevytilaa omia verkkosivujaan ja muita verkkopalveluita (kuten tietokantoja) varten.
Ohjelmointikieli	Ohjelmointikielien ovat formaaleja kieliä, joita ohjelmoijat käyttävät tietokoneen ohjelmointiin.
Excel	Microsoft Excel on taulukkolaskentaohjelma, joka on osa Microsoft Officea.
Lähdekoodi	Lähdekoodilla tarkoitetaan ohjelmoinnissa tietokoneohjelman tekstimuotoista ohjelmointikielistä listausta.
Kääntäjä/kääntäminen	Ohjelmointikielen kääntäjä on tietokoneohjelma, joka luo tietokoneohjelman ohjelmointikielisen lähdekoodin perusteella konekielisen ajettavan binääritiedoston (eli kääntää ohjelman).
Nebula	Nebula Oy on suomalainen Internet-palveluntarjoaja, joka on perustettu vuonna 1997.

## 1 YHDISTYSESITTELY

Opinnäytetyö tehtiin tanssiurheiluseura Oikea Rytmi ry:lle. Oikea Rytmi ry on Porissa toimiva tanssiurheiluseura, joka on perustettu vuonna 1995. Seuran toiminnassa on mukana lapsia, nuoria ja aikuisia, jotka harrastavat tanssiurheilua sen eri muodoissa.

Tanssiurheiluseura Oikea Rytmi ry:n päätavoitteena on lisätä tanssiurheilun tunnettavuutta Satakunnan alueella. Seuran toiminta perustuu palkattomaan työhön. Täten jäsenmaksuista saadut varat käytetään lasten, nuorten ja aikuisten tanssiurheilutoiminnan ja tanssiliikunnan tukemiseen ja kehittämiseen. (Tanssiurheiluseura Oikea Rytmi ry www-sivut, 2012)

## 2 TYÖN JOHDANTO

### 2.1 Työn tausta ja käyttötarkoitus

Opinnäytetyön tarkoituksena oli toteuttaa Oikea Rytmi ry:lle jäsenrekisterisovellus, jonka avulla olisi mahdollista hallinnoida paremmin seuran jäsenten tietoja. Ennen tätä sovellusta seura käytti yksinkertaista Excel-tiedostoa, joka oli primitiivinen ratkaisu jäsentietojen säilytykseen.

Tekemällä tarkoitukseen räätälöity sovellus käyttäen Qt/C++ ohjelmointikieltä ja MySQL-relaatiotietokantaa oli mahdollista toteuttaa monia käyttöä helpottavia ominaisuuksia, jotka eivät olleet mahdollisia pelkkää Excel-tiedostoa käytettäessä. Ylläpitäjää helpottavia ominaisuuksia ovat muun muassa monipuoliset hakutoiminnot sekä erilaiset tulostettavissa olevat raportit ja laskut, joihin on automaattisesti täytetty tarvittavat tiedot. Myös versionhallinta tulee automaattisesti kuntoon, koska käytetään internetissä sijaitsevaa keskitettyä tietokantaa. Tarkemmin sovelluksen ominaisuudet kuvataan myöhemmin tässä dokumentissa.

## 2.2 Valittu ohjelmointikieli ja tietokanta

Suunnitteluvaiheessa, ennen ohjelmoinnin aloittamista, piti valita mihin tietokantaan jäsentiedot tullaan tallentamaan, ja mitä ohjelmointikieltä/ympäristöä käytetään varsinaisen käyttöliittymä-ohjelman toteuttamiseen. Asiakkaan kanssa neuvottelussa mietittiin mikä olisi paras ratkaisu ja mitä ohjelmointikieltä/ympäristöä asiakas halusi käytettävän.

Nykyaikana ainoa järkevä ratkaisu tietokannalle on relaatiotietokanta, joka käyttää standardiksi kehittynyttä SQL kyselykieltä. Keskeinen mietittävä asia olikin, että mitä relaatiotietokantajärjestelmää tässä työssä käytetään. Mahdollisuuksia oli monia, mutta mietinnän jälkeen valittiin käytettäväksi suosittu ja hyvin tuettu MySQL-relaatiotietokanta.

Tietokanta päätettiin luoda Nebula-webhotelliin, koska asiakkaalla oli jo ennestään siellä toimivat kotisivut. Nebulaan oli saatavilla pieneen lisähintaan myös MySQL-tietokantapalvelin. Käyttämällä Nebulan valmista tietokantapakettia ei asiakkaan tarvitse huolehtia oman palvelimen rakentamisesta eikä myöskään palvelimen ja tietokannan päivittämisestä. Itse käyttöliittymäohjelmalla otetaan yhteyttä tähän tietokantaan internetin kautta.

Tietokannan lisäksi valittiin myös ohjelmointikieli, jolla varsinainen käyttöliittymä toteutetaan. Asiakkaan kanssa keskusteltaessa tuli esiin kaksi vaihtoehtoa. Ensimmäisenä vaihtoehtona oli sovelluksen toteuttaminen PHP-ohjelmointikielellä, jolloin sovellus asennettaisiin internetiin ja käyttäjä voisi ottaa ohjelmaan yhteyttä millä tahansa tietokoneella, jossa olisi Web-selain. Toisena vaihtoehtona oli toteuttaa ohjelma käyttäen perinteisempää ohjelmointikieltä, joka käännettäessä tuottaisi käyttäjän omalla koneella suoritettavan ohjelman.

Lopulta toteutettavaksi valittiin toinen vaihtoehto eli omalla koneella ajettava ohjelma. Ohjelmointikieleksi valittiin Qt-kehitysympäristö C++-ohjelmointikieltä käyttäen. Qt on alustariippumaton kieli ja sillä toteutetut sovellukset toimivat esimerkiksi Windowsissa, Applen käyttöjärjestelmissä, tai Symbian puhelimissa. (Wikipedia: Qt www-sivut, 2012)



Qt:n valintaan vaikutti myös se, että asiakkaan yhteyshenkilöllä oli ennestään jo hieman kokemusta ko. kielestä. Siksi haluttiin kokeilla kielen toimivuutta ja hyödyntää avoimen lähdekoodin ohjelmistokehitystä.

## 3 QT-OHJELMOINTIKIELI

### 3.1 Perusteet ja historiaa

Qt on Digia Oyj:n omistama alustariippumaton kehitysympäristö, jonka avulla on mahdollista toteuttaa erilaisia ohjelmia. Qt:llä voidaan toteuttaa konsolisovelluksia tai rakentaa monimutkaisia graafisia käyttöliittymiä. Tämän dokumentin kuvailema opinnäytetyö on yksi esimerkki Qt:llä rakennetusta graafisen käyttöliittymän sisältämästä sovelluksesta.

Qt:tä on käytetty muun muassa Google Earth, Skype, VirtualBox ja VLC media player toteutuksissa. Qt ei siis ole täysin tuntematon ratkaisu vaan sitä on käytetty monissa suosituissa ja tunnetuissa. (Wikipedia: Qt [www-sivut](#), 2012)

Qt kehitysympäristö on vaihtanut omistajaa useaan otteeseen. Alun perin Qt:n kehitti norjalainen yritys nimeltä Trolltech mutta Nokia osti kyseisen yrityksen vuonna 2008 ja sai täten myös Qt:n omistukseensa. Vuonna 2011 suomalainen ohjelmistoyhtiö Digia osti Nokialta Qt:n lisensointi- ja palveluliiketoiminnan. Tämän jälkeen Nokia jatkoi Qt-teknologian kehittämistä. Digia oli kuitenkin vastuussa kaikesta kaupallisesta lisensoinnista ja palveluliiketoiminnasta. Elokuussa vuonna 2012 Digia osti Nokialta koko Qt-kehitysympäristön ja tämän kaupan myötä Digia vastaakin nyt kaikista Qt:hen liittyvistä asioista, kuten esimerkiksi tuotekehityksestä. (Qt Digia: About us [www-sivut](#), 2012)

Qt:hen on sisäänrakennettu tuki C++ ohjelmointikielelle mutta se tukee myös muitakin ohjelmointikieliä, kuten Javaa, Pythonia, Rubya ja C#:aa. Tuotekehityksen jatku-

essa tuki uusille ohjelmointikielille varmasti lisääntyy tulevaisuudessa. (Wikipedia: Qt www-sivut, 2012)

Qt on vapaa ja avoimen lähdekoodin ohjelmisto ja se on saatavilla GNU Lesser General Public License-lisenssillä, sekä kaupallisellakin lisenssillä. (Wikipedia: Qt www-sivut, 2012)

### 3.2 Qt:lla ohjelmointi

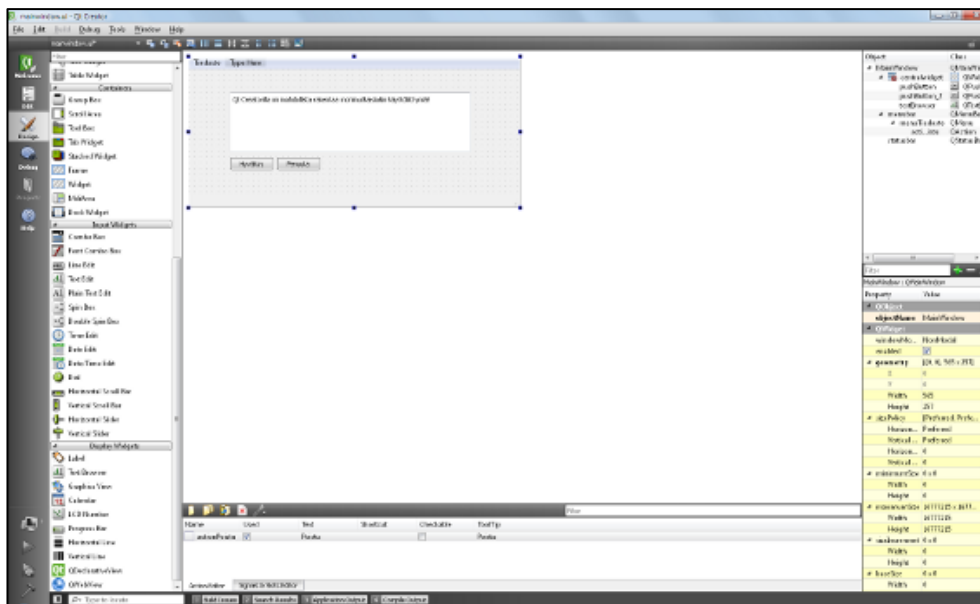
Qt ei ole oma ohjelmointikielensä, vaan se on varsinaisesti luokkakirjasto ja ohjelmointiympäristö, joka helpottaa ohjelmien toteuttamista. Alla esimerkkinä yksinkertainen ohjelma Qt:llä toteutettuna, joka tulostaa standardiksi kehittyneen ”Hello world” tekstin. (Wikipedia: Qt www-sivut, 2012)

```
#include <QApplication>
#include <QLabel>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QLabel label("Hello, world");
    label.show();
    return app.exec();
}
```

QApplication ja QLabel ovat Qt:n sisältämiä kirjastoja ja näiden avulla käyttäen normaalia C++ ohjelmointikielen syntaksia saadaan toteutettua ohjelma, joka tulostaa tekstin ”Hello world”.

Qt:lle löytyy useita valmiita työkalusovelluksia, joita käyttämällä Qt:lla työskentely on helppoa. On myös mahdollista lisätä Qt:lle tuki monille yleisimmille ohjelmointityökaluille. Tästä esimerkkinä on Microsoftin Visual Studio ohjelmankehitysympäristö, johon on mahdollista asentaa lisäosa, joka mahdollistaa Qt-sovelluksien tekemisen kyseisellä työkalulla. Tässä opinnäytetyössä käytettiin kuitenkin Qt Creator ohjelmaa, joka on vartavasten rakennettu Qt-ohjelmointia varten.



Kuva 1 Qt Creator käyttöliittymä

### 3.3 Miksi Qt?

Qt oli asiakkaalle jo ennestään tuttu ohjelmointikieli, joten päätettäessä, millä kielellä sovellus toteutetaan, oli Qt asiakkaalle mieluinen vaihtoehto. Qt ei ollut ennestään tuttu työn tekijälle, joten Qt tarjosi tilaisuuden opetella opinnäytetyötä tehtäessä uusi tapa toteuttaa sovelluksia, josta voi olla hyötyä tulevaisuudessa.

Ainoa toinen mietitty vaihtoehto Qt:lle oli toteuttaa sovellus käyttäen PHP-ohjelmointikieltä ja asentaa se Nebulan web-palvelimelle. Asiakkaalle kuitenkin oli mieluisempaa valita Qt, joten se loppujen lopuksi valittiin kieleksi, jolla sovellus toteutettiin.

### 3.4 Qt:n lisenssi

Lisenssi on ohjelmiston käyttöön oikeuttava sopimus. Eri ohjelmilla voi olla eri lisenssejä ja niissä määritellään lailliset asiat kuten se miten ohjelmistoa voi käyttää, mihin sitä voi käyttää, voiko sitä jakaa muille ja niin edelleen.

Lisenssejä voi olla erilaisia. Esimerkiksi ”freeware”-lisenssi tarkoittaa, että ohjelmaa saa käyttää täysin vapaasti, mutta sitä ei saa myydä edelleen. ”Public domain”-

lisenssi tarkoittaa, että tekijä on käytännössä luopunut oikeuksistaan niin paljon kuin mahdollista ja PD:n alla olevia ohjelmia voikin muokata vapaasti ja jopa myydä muille maksua vastaan. (Wikipedia: Lisenssi www-sivut, 2012)

Mainittakoon vielä GNU General Public License –lisenssi, joka on tarkoitettu vapaiden ohjelmistojen julkaisemiseen. GPL antaa kenelle tahansa oikeuden käyttää ja muokata sen alla julkaistuja lähdekoodeja. Lisenssi myös takaa sen, että kaikki nämä vapaudet säilyvät myös GPL-koodiin pohjautuvissa teoksissa. (Wikipedia: GNU GPL www-sivut, 2012)

Qt oli alun perin saatavilla vain kaupallisella lisenssillä, joka tarkoitti sitä, että käytöstä ja levittämisestä piti maksaa tekijälle. Vuodesta 2009 lähtien Qt on kuitenkin tullut saataville myös uudella vapaalla lisenssillä. Kaupallinenkin lisenssi on silti edelleen käytössä ja sen ostaja saa ylimääräisiä etuuksia ja oikeuksia verrattuna vapaan lisenssin käyttäjään. (Wikipedia: Qt www-sivut, 2012)

Tällä hetkellä kaupallisen lisenssin lisäksi Qt:n saa avoimen lähdekoodin GNU Lesser General Public License 2.1 -lisenssillä. LGPL on muunnos edellä mainittuun GPL lisenssiin ja sen tarkoituksena oli tehdä kompromissi GPL:n ja yksinkertaisempien lisenssien välillä. Pääasiassa LGPL sallii asioita, joita tiukempi GPL ei salli. LGPL:n alla ei esimerkiksi kaupallisten ohjelmien tarvitse julkaista lähdekoodia, vaikka ohjelma käyttäisikin LGPL-lisenssoituja komponentteja. (Wikipedia: GNU LGPL www-sivut, 2012)

LGPL lisenssin alla Qt:llä voi tehdä avoimen ja suljetin lähdekoodin ohjelmia ohjelmia. Kuitenkin mahdolliset Qt-kirjastoon tehdyt muutokset pitää jakaa LGPL-lisenssiehtojen mukaisesti. Kaupallisen lisenssin ostettaessa, ei Qt-kirjastoihinkaan tehtyjä muutoksia tarvitse jakaa.

## 4 RELAATIOTIETOKANNAT

### 4.1 Yleistä relaatiotietokannoista

Relaatiotietokannat ovat nykyaikana tietotekniikassa yleisin tapa tallettaa tietoja. Relaatiotietokannat koostuvat käytännössä tauluista, jotka sisältävät sarakkeita tai kenttiä, jotka voivat koostua eri tietotyypeistä.

Esimerkki yksinkertaisesta relaatiotietokannan taulusta on esitetty ohessa. Taulun nimi on **Jäsenet**, ja se sisältää neljä eri kenttää.

JäsenID	Etunimi	Sukunimi	Ikä
1	Aaro	Ahonen	22
2	Kalle	Korhonen	59
3	Jaakko	Järvinen	11

Yksi tietokanta voi koostua useista tauluista, jotka sisältävät eri tietoja, ja jotka usein ovat jossain yhteydessä toisiinsa. Edellisessä esimerkissä jokaisella henkilöllä on yksilöllinen ID-numero, johon voitaisiin viitata toisessa taulussa. Oletetaan, että jokainen edellisen esimerkkitaulun jäsen voi olla mukana monessa ryhmässä yhtä aikaa. Tällöin toinen esimerkkitaulu voisi olla **Ryhmät** joka sisältää kaksi kenttää.

RyhmäID	Ryhmänimi
1	Ryhmä A
2	Ryhmä B

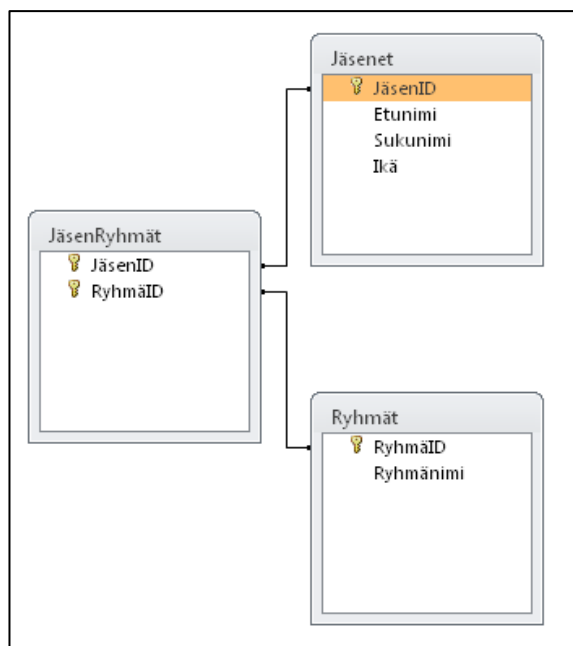
Molemmissa tauluissa on yksilöllinen ID-numero, mutta taulut eivät vielä ole yhteydessä toisiinsa. On kuitenkin mahdollista tehdä kolmas taulu, jossa jäsenet ja ryhmät liitetään toisiinsa luomalla yhteys niiden välille. Tämän taulun nimi voisi olla vaikka **JäsenRyhmät**, jossa on kaksi kenttää, ja jotka viittaavat edellisten taulujen yksilöllisiin ID-numeroihin ja täten yhdistävät ne.

JäsenID	RyhmäID
1	1
1	2
2	2
3	1
3	2

Alla olevassa esimerkissä viitataan molempiin tauluihin. Jos yhdistämme taulut, näemme seuraavanlaisen tuloksen:

JäsenID	Etunimi	Sukunimi	Ikä	Ryhmänimi
1	Aaro	Ahonen	22	Ryhmä A
1	Aaro	Ahonen	22	Ryhmä B
2	Kalle	Korhonen	59	Ryhmä B
3	Jaakko	Järvinen	11	Ryhmä A
3	Jaakko	Järvinen	11	Ryhmä B

Ohjelmointikielessä on kuitenkin helpompaa käyttää tämänkaltaisia yhteyksiä ja useita liitettyjä tauluja, kuin luoda yksi iso taulu, jossa sama tieto on tallennettu useaan kertaan. Edellinen esimerkki oli yksinkertainen, mutta todelliset tietokannat sisältävät yleensä monia taulua ja useita eri tietueita.



**Kuva 2** Esimerkki taulujen liitoksista

Suurissa järjestelmissä tietokannat ovat valtavia ja ne koostuvat erittäin suuristakin määristä tietoa, ja vaativat näin satoja tai jopa tuhansia kovalevyjä. Tämän opinnäytetyön tietokanta on kuitenkin suhteellisen pieni, mutta sisältää silti tietokannan, joka koostuu useasta eri taulusta, jotka kukin sisältävät useita kenttiä.

## 4.2 SQL-kieli

SQL on lyhenne sanoista Structured Query Language ja se on standardiksi kehittynyt yksinkertainen ohjelmointikieli, jolla relaatiotietokantoihin voi tehdä hakuja, lisäyksiä, muutoksia ja poistoja. Nykyaikana käytännössä kaikki relaatiotietokannat ymmärtävät SQL-kieltä.

SQL-kielillä voi tehdä monimutkaisiakin toimintoja, mutta yleisimmät komennot ovat SELECT (hae tieto), UPDATE (päivitä tieto), INSERT (lisää tieto) ja DELETE (poista tieto). Näihin komentoihin voidaan myös liittää WHERE (missä) komento, jolla voidaan tarkentaa mihin riveihin komento kohdistuu. (w3schools: SQL tutorial www-sivut, 2012)

#### 4.2.1 SELECT

SELECT komennolla haetaan tietoa tietokannan tauluista. Viitaten edellisen kohdan tauluesimerkkeihin voisi yksinkertainen hakukomento olla:

```
SELECT Etunimi, Sukunimi FROM Jäsenet WHERE JäsenID = 1
```

Tämä komento näyttäisi **Jäsenet** taulun **Etunimi** ja **Sukunimi** kentän tiedot riviltä jossa **JäsenID:n** arvo on 1. Eli käytännössä kysely tulostaisi nimen ”Aaro Ahonen”. Jos WHERE komento jätettäisiin pois, tulostaisi kysely kaikki taulun sisältämät nimet. On huomattavaa, että itse taulu sisältää myös tiedon jäsenen iästä, mutta koska sitä ei ole mainittu SELECT lauseessa ei se myöskään näy lopputuloksessa. (Patrick 2008, 33)

#### 4.2.2 INSERT

INSERT komennolla voidaan lisätä uusia rivejä tietokannan tauluihin. Jos esimerkiksi lisätään täysin uusi jäsen tietokantaan, käytetään INSERT komentoa. Viitaten taas edellisen kohdan esimerkkeihin voisi yksinkertainen lisäyskomento olla:

```
INSERT INTO Jäsenet (JäsenID, Etunimi, Sukunimi, Ikä) VALUES (4, 'Laura', 'Lahtinen', 20)
```

Tämä komento lisää **Jäsenet** tauluun uuden rivin, jossa **JäsenID:n** arvo on 4, jäsenen nimi Laura Lahtinen ja jäsenen ikä 20 vuotta. (Patrick 2008, 151)

#### 4.2.3 UPDATE

UPDATE komennolla päivitetään jo olemassa olevia tietoja ja muutetaan niitä. Jos esimerkiksi edellisessä kohdassa lisätyn Laura Lahtisen ikää haluttaisiin kasvattaa yhdellä, voitaisiin käyttää seuraavanlaista päivityskomentoa:

```
UPDATE Jäsenet SET Ikä = 21 WHERE JäsenID = 4
```

Tässä nähdään, että päivityslauseessa määritellään muutettava taulu, muutettava kenttä ja sen uusi arvo. Lopuksi käytetään WHERE komentoa tarkentamaan mitä ri-



viä muutetaan, eli tässä tapauksessa vain riviä jonka JäsenID on 4. Jos WHERE komento jätettäisiin kokonaan pois, päivittäisi komento kaikkien rivien Ikä-kentän arvoksi 21. (Patrick 2008, 157)

#### 4.2.4 DELETE

DELETE komento yksinkertaisesti poistaa yhden tai useamman rivin taulusta. Poistokomennon syntaksi on yksinkertainen:

**DELETE FROM Jäsenet WHERE JäsenID = 4**

Kuten päivityskomennossa, määritellään poistokomennossakin taulu josta poistetaan, ja WHERE komennolla tarkennetaan poistettava rivi. Jos WHERE ehtoa ei ole annettu, poistaa poistokomento taulusta kaikki rivit ja täten tyhjentää sen kokonaan. Poistoja tehtäessä onkin täten tärkeää pitää huolta siitä, että WHERE komento on aina poistolauseessa mukana, ettei vahingossa tuhota tietoa jota ei saisi tuhota. (Patrick 2008, 159)

#### 4.3 Tietotyypit

Tietokantojen tauluja tehtäessä on mahdollista määritellä taulujen eri kentille tietotyyppejä, jotka kertovat minkälaista tietoa on kyseiseen kenttään mahdollista laittaa. Tietotyypit yleensä vaihtelevat jossain määrin riippuen siitä mitä relaatiotietokantaa käytetään, mutta yleisesti ainakin jotkut tietotyypit ovat standardeiksi kehittyneitä ja täten löytyvät jokaisesta relaatiotietokannasta. Alla esittelen lyhyesti muutaman näistä yleisistä tietotyypeistä. (Patrick 2008, 214–216)

Tietotyyppi	Selitys
INT	INT eli Integer tarkoittaa kokonaislukua, eli ei desimaaleja.
VARCHAR	VARCHAR (tai myös CHAR) tarkoittaa tekstikenttää..
DATETIME	DATETIME on päivämäärekenttä.
DOUBLE	DOUBLE on luku joka sisältää myös desimaalit.
BOOLEAN	BOOLEAN on joko totta (true) tai epätotta (false).

## 4.4 MySQL-esittely

Relaatiotietokantoja on useita eri versioita useilta eri yrityksiltä. MySQL on relaatiotietokanta, jonka ensimmäinen versio julkaistiin vuonna 1995. MySQL on erittäin suosittu tietokanta ja sitä käyttää esimerkiksi Wikipedia, YouTube ja Twitter. Tällä hetkellä MySQL:n omistaa Oracle. Alun perin MySQL:ää kehitti ruotsalainen yritys nimeltä MySQL AB. Sun Microsystems osti kyseisen yrityksen vuonna 2008 ja Oracle taas osti Sunin vuonna 2009. MySQL on saatavissa vapaalla lisenssillä ja sen vaatimat ohjelmistot voi imuroida MySQL:n web-sivustolta. Useat web-hotellit kuitenkin tarjoavat pieneen hintaan mahdollisuuden käyttää MySQL-tietokantaa, jolloin ei ole tarvetta rakentaa omaa yksityistä palvelinta. Tämän opinnäytetyön sovelluksen käyttämä tietokanta on internetissä Nebulan palvelimilla. Tämä helpottaa asiakkaan asemaa, koska Nebula ylläpitää MySQL-palvelinta ja pitää huolen siitä, että tietoturva on kunnossa ja että palvelin on aina toiminnassa. (MySQL www-sivut, 2012)

# 5 SUUNNITTELU

## 5.1 Suunnitteluprosessi

Kun sovellusta aletaan toteuttaa, on tärkeää yrittää määritellä sovelluksen tarpeet ja laatia suunnitelma siitä miten kyseiset tarpeet tullaan toteuttamaan. Tämä helpottaa suuresti sovelluksen toteutusta. Suunnittelun alussa on tärkeää olla yhteydessä asiakkaaseen ja yleensä pitää jonkinlainen aloituspalaveri, jossa selvitetään asiakkaan tarpeet ja halut siitä, millainen sovellus on. Myös käyttöliittymää kannattaa ainakin karkeasti suunnitella tässä vaiheessa.

On kuitenkin muistettava, että suunnitelmat ja jopa tarpeet voivat muuttua sovelluksen kehittyessä ja uusia ongelmia voi tulla eteen, jotka vaativat muutoksia alkuperäisissä suunnitelmissa. Esimerkiksi tämän opinnäytetyön toteutuksen aikana tuli vastaan ongelmia, jotka vaativat muutoksia, ja asiakkaan kanssa kommunikoidessa tuli

mieleen tarpeita, joita ei aloituspalaverissa ollut huomioitu niin hyvin, kuin olisi ehkä pitänyt.

Tästä huolimatta on aina parempi olla edes jonkinasteinen idea siitä, miten sovellus rakennetaan, sen sijaan, että vain aloitettaisiin sovelluksen koodaaminen ilman mitään suunnitelmaa, olipa kyse kuinka pienestä sovelluksesta tahansa. Muutoksiin kannattaa kuitenkin aina varautua.

## 5.2 Ohjelman suunnittelu

Ohjelman suunnittelu koostui ensinnäkin tietokannan suunnittelusta. Tietokantaan tuli määritellä jäsenrekisterin tarvitsemat tiedot ja miten niistä rakennetaan taulut ja taulujen välille liitokset. Lisäksi tuli harkita myös karkeasti millainen käyttöliittymä ohjelmalla tulee olemaan, jotta se on looginen ja asiakkaan ymmärtämä. Itse ohjelmointikoodin rakennetta ei alustavasti ollut, koska Qt oli alun perin vähän tuntematon kieli ja sen käyttö tuli todella tutuksi itse toteutusvaiheessa. Sovelluksen testausta jossain määrin suunniteltiin. Alla olevissa kohdissa selvitetään tarkemmin jokaista yllä mainittua kohtaa.

## 5.3 Tietokannan suunnittelu

Tietokantaa suunniteltaessa on tiedettävä ensinnäkin niin tarkoin kuin mahdollista mitä kaikkia tietoja sovellus tulee tarvitsemaan. Koska kyseessä oli jäsenrekisterin toteutus, oli osa näistä tiedoista itsestään selviä, kuten esimerkiksi jäsenen nimet, osoitteet ja vastaavat yhteystiedot. Oli kuitenkin mahdollista, että muita lisätietoja tarvitaan joidenkin jäsenten kohdalla.

Jokainen jäsen esimerkiksi voi olla mukana eri ryhmissä. Ryhmäjakoja piti pystyä muuttamaan, joten oli mietittävä ryhmätaulut ja tarvittavat yhteydet jäsentaulun ja ryhmätaulun välille. Jäsenentiedot voivat myös sisältää eri rooleja ja jäsenet voivat olla mukana eri kausissa. Myös laskutus piti automatisoida niin paljon kuin mahdollista.

Tietokannan alustava suunnitelma ei ihan sellaisenaan säilynyt vaan toteutuksen aikana tuli vastaan ongelmia, jotka vaativat muutoksia tietokantaan. Osa ongelmista johtui itse käyttöliittymän vaatimuksista, joita ei huomattu suunnitelmaa tehtäessä. Osa taas johtui muutoksista sovelluksen tarpeissa, joita ei huomattu yleisessä suunnitelmassa.

Muutokset eivät kuitenkaan olleet radikaaleja, vaan enemmänkin pieniä korjauksia ja lisäyksiä, joten loppujen lopuksi voitaisiin sanoa, että tietokannan suunnittelu onnistui aika hyvin.

#### 5.4 Käyttöliittymän suunnittelu

Käyttöliittymää suunniteltaessa oli tärkeää ottaa huomioon, että sovelluksen käyttäjissä oli henkilöitä, jotka eivät ole tietotekniikan asiantuntijoita. Tämän vuoksi oli tärkeää pyrkiä pitämään käyttöliittymä loogisena Windows-käyttöliittymien standardin mukaisena, jotta sovelluksen käyttäjä osaisi etukäteen jo hieman odottaa miten asiat toimivat.

#### 5.5 Testaus

Testaussuunnitelma ei voinut ollut kovin monimutkainen, koska oli vain yksi testaaja. Tässä opinnäytetyössä testauksen ja vikojen pääetsimisen hoiti ohjelmoija itse. Koska asiakkaan puoleltakin testaajia oli vain yksi, testauksen merkitys kasvoi. Mikäli toiminnallinen virhe jäi huomaamatta ohjelmoijalta ja asiakkaan testaajalta, muita testaajia ei ollut jotka sen voisivat löytää.

Työn edetessä asiakkaan kanssa keskusteltiin jatkuvasti ja sovelluksen kehittyessä asiakas testasi eri versioita. Asiakkaan testauksessa tuli viimeistään ilmi, onko ohjelmoija ymmärtänyt asiakkaan vaatimukset oikein. Asiakkaan näkemykset ohjelmasta luonnollisesti myös muuttuvat sitä mukaa, kun ohjelmaan kehityksen myötä tulee lisää logiikkaa ja älyä. Suurin osa halutuista muutoksista koskikin ohjelman käytettävyyttä.

## 6 TYÖKALUT

Luotaessa uusia sovelluksia tarvitaan jonkinasteiset työkalut, joilla uusi sovellus tullaan rakentamaan. Koska tämä sovellus käyttää Qt-ohjelmointikieltä täytyi toteutusympäristöön asentaa työkalut, joilla uusi sovellus voidaan toteuttaa mahdollisimman helposti ja tehokkaasti. Koska sovellus käyttää tietokantaa, myös tietokantaa pitää hallinnoida, joka sekin vaatii omat työkalunsa.

Tämän opinnäytetyön vaatima sovellus toteutettiin käyttäen Qt Development Frameworksin toteuttamaa Qt Creator -ohjelmaa, jossa on valmiiksi mukana Qt C++ ohjelmointiin vaadittavat kirjastot ja kääntäjät. (Qt Digia: Qt Creator [www-sivut](#), 2012)

Tietokantana oli MySQL, ja sen palvelin asennettiin toteutusympäristöön. Nebulan palvelimille tilattiin myös toinen tietokantapalvelin, jota käytetään lopullisessa sovelluksen versiossa ja varsinaisessa käytössä.

Molempia tietokantoja ylläpidettiin MySQL Workbench ohjelmalla, joka on visuaalinen MySQL tietokantojen ylläpito-ohjelma. (MySQL [www-sivut](#), 2012)

### 6.1 Toteutusympäristö

Varsinaisena toteutusympäristönä käytettiin Windows 7 käyttöjärjestelmää, johon asennettiin kaikki vaadittavat työkalut.

### 6.2 Tietokannan asennus

Toteutusympäristöön asennettiin MySQL palvelin 5.5 ja sen lähdekoodit. Vaikka virallinen sovellus käyttääkin Nebulan palvelimilla olevaa tietokantaa, oli sovellusta toteutettaessa hyödyllistä olla paikallinen tietokanta, jossa voitiin tietokantamuutoksia ja ratkaisuja testata ennen niiden siirtämistä Nebulan palvelimille. MySQL lähdekoodeja tarvittiin myös Qt Creatoria asennettaessa.

### 6.3 Qt-ohjelmointi

Itse Qt ohjelmointi toteutettiin käyttäen Qt Creator versiota 2.2.0. Tällä ohjelmalla oli mahdollista rakentaa käyttöliittymä käyttäen Qt Creatorin graafisia työkaluja ja varsinainen ohjelmalogiikka käyttäen C++ ohjelmointikieltä.

#### 6.3.1 Qt Creatorin asennus ja siihen MySQL tuki

Standardi vapaan lähdekoodin Qt Creatorin asennus oli helppo ja automatisoitu, mutta se ei sisällä tukea MySQL tietokannalle. Jotta MySQL tuki toimi piti MySQL-palvelimen lähdekoodeja ja Qt Creatorin lähdekoodeja käyttäen rakentaa QMYSQL-moduuli. Tämä aiheutti hieman ongelmia, koska Qt-MySQL moduulin kääntäminen ei heti onnistunut, ja dokumentaatio asiasta oli heikkoa. (Qt Reference Documentati-on: SQL Drivers www-sivut, 2012)

### 6.4 MySQL Workbench

MySQL Workbench on sovellus, jolla on mahdollista rakentaa tietokantoja, toteuttaa tauluja ja tehdä kyselyjä käyttäen graafista käyttöliittymää. Tälle työkalulle löytyy oma automaattinen asennusohjelmansa ja sen käyttö on loogista ja yksinkertaista.

Vaikka relaatiotietokantoja voikin rakentaa ja ylläpitää käyttäen pelkkää käsin toteutettua SQL-kieltä, on se vaikea ja hidas tapa toteuttaa ja muuttaa tietokantaa. Workbench nopeuttaa ja helpottaa toteutusta huomattavasti, joten sitä kannattaa yleensäkin käyttää MySQL-tietokantoja käytettäessä. (MySQL www-sivut, 2012)

## 7 TOTEUTUS

### 7.1 Tietokannan toteutus

Suunniteltu tietokanta toteutettiin käyttäen edellä mainittua MySQL Workbench työkalua. Tällä rakennettiin yksitellen suunnitellut taulut ja luotiin niiden väliset yhteydet. Opinnäytetyön edetessä Workbenchillä toteutettiin myös kaikki muutokset tietokantaan.

### 7.2 Taulut

Opinnäytetyön tietokanta koostuu useasta taulusta. Alla on listattu taulut, taulujen kentät ja lyhyet selitykset siitä mihin taulua käytetään varsinaisessa sovelluksessa

Sanottakoon vielä, että merkintä PK tulee sanoista Primary Key. Tämä tarkoittaa että kyseinen kenttä on taulun yksilöllinen kenttä, jonka perusteella erotellaan eri tiedot toisistaan. Esimerkiksi jassen\_id on yksilöllinen Primary Key ja jokaisella jäsenellä on oma yksilöllinen arvonsa joka ei voi toistua. PK siis estää sen, että tietokantaan lisättäisiin useampia identtisiä arvoja.

#### 7.2.1 Jassenkausi

<b>Nimi</b>	<b>Tyyppi</b>
jassen_id	int(11) PK
vuosi	int(11) PK
maksusumma	double
huom	varchar(1000)
maksettu	varchar(10)
luotupvm	datetime

Tämä taulu sisältää tiedot kausista, joilla jäsenet ovat mukana. Kun tietty jäsen liitetään kaudelle, tehdään siitä merkintä tähän tauluun. Tärkeää tässä taulussa on jäsenelle yksilöllinen maksusumma-kenttä, joka mahdollistaa jokaiselle jäsenelle oman hintansa. Tätä käytetään, koska tietyt jäsenet voivat saada alennuksia maksuihin. Myös maksettu kenttä on tärkeä ja siinä on arvona joko ”kyllä” tai ”ei” sen mukaan, onko jäsen maksanut kyseisen kauden laskun.

### 7.2.2 Jäsenrooli

Nimi	Tyyppi
jasen_id	int(11) PK
rooli_id	varchar(45) PK

Tämä taulu liittyy jäsentaulun ja roolitaulun. Se sisältää käytännössä listauksen rooleista, jotka tietyllä jäsenellä on. Jäsen joko on roolissa tai ei, joten tässä taulussa ei ole enempää kenttiä.

### 7.2.3 Jäsenryhma

Nimi	Tyyppi
jasen_id	int(11) PK
ryhmanimi	varchar(30) PK
huom	varchar(1000)
maksusumma	double

Tämä taulu sisältää tiedot siitä missä ryhmässä jäsen on. Kuten kausissakin, tässä taulussa tärkeänä on maksusumma-kenttä, johon voidaan asettaa jäsenelle oma alennettu hintansa.



## 7.2.4 Jasentieto

<b>Nimi</b>	<b>Tyyppi</b>
jasen_id	int(11) PK
jasennro	varchar(10)
etunimi	varchar(25)
sukunimi	varchar(25)
katuosoite	varchar(50)
postinumero	varchar(7)
postitoimipaikka	varchar(30)
puhelin	varchar(20)
email	varchar(100)
huoltaja	varchar(100)
sukupuoli	varchar(1)
syntaika	date
huom	varchar(1000)
liittymispvm	datetime
muutospvm	datetime

Tämä taulu sisältää jäsenen varsinaiset tiedot. Näihin kuuluvat osoitetiedot, jäsennumerot, syntymäaika ja vastaavat tiedot.

## 7.2.5 Kausi

<b>Nimi</b>	<b>Tyyppi</b>
kausi_id	int(11) PK
vuosi	int(11)
huom	varchar(1000)
a16maksu	double
y16maksu	double

Tässä taulussa on lista kaikista kausista, nykyisistä ja vanhoista. Vanhat tiedot säilytetään historiasyistä. Tärkeää on a16maksu ja y16maksu, jotka sisältävät kauden oletusmaksun riippuen siitä, onko jäsen alle 16-vuotias vai ei.

### 7.2.6 Rooli

Nimi	Tyyppi
rooli_id	int(11) PK
roolinimi	varchar(30)
huom	varchar(1000)

Tämä taulu sisältää kaikki mahdolliset roolit, jotka jäsenellä voi olla.

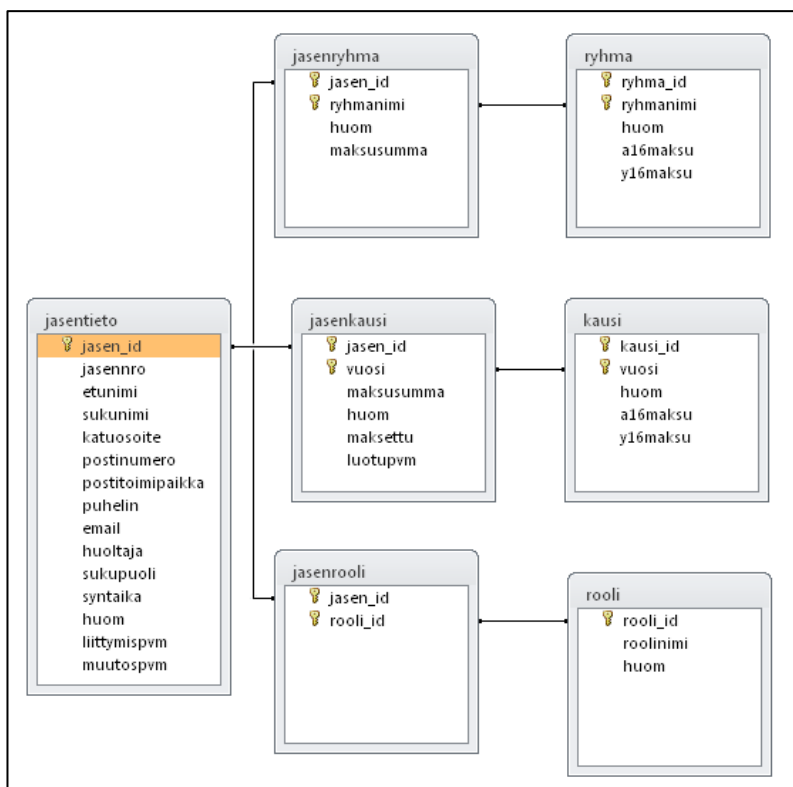
### 7.2.7 Ryhma

Nimi	Tyyppi
ryhma_id	int(11) PK
ryhmanimi	varchar(30)
huom	varchar(1000)
a16maksu	double
y16maksu	double

Tämä taulu sisältää listauksen kaikista ryhmistä. Kuten kausi-taulussa, tässäkin taulussa tärkeänä on a16maksu- ja y16maksu-kentät, jotka sisältävät oletusmaksun alle ja yli 16-vuotiaille.

### 7.2.8 Yhteydet

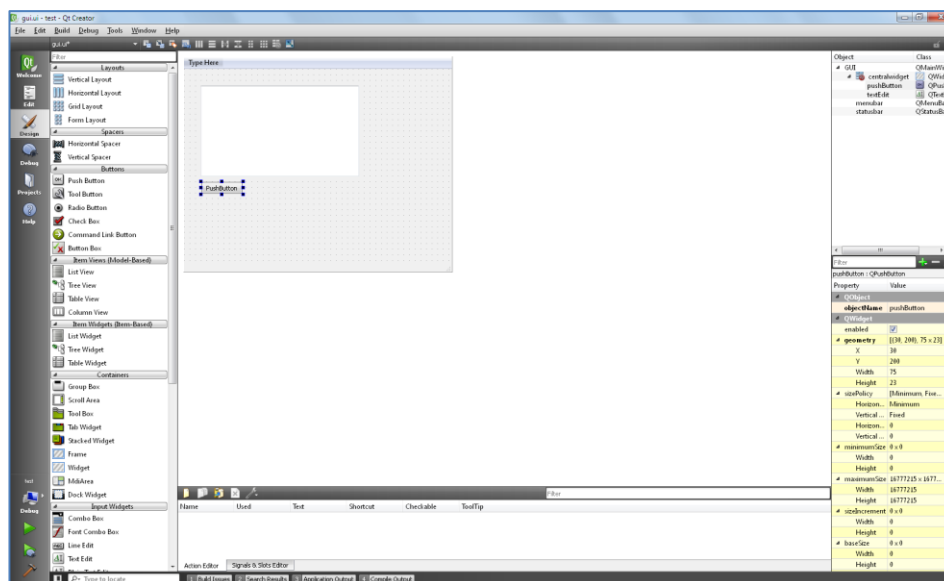
Koska jotkut taulut ovat liitoksissa toisiinsa, alla on kuva siitä miten taulut liittyvät toisiinsa.



Kuva 3 Opinnäytetyön taulujen liitokset

### 7.3 Käyttöliittymä Qt Creatorilla

Qt Creatorilla on mahdollista luoda käyttöliittymä käyttäen siihen rakennettua graafista työkalua, joka toimii samoin kun monet muutkin nykyaikaiset toteutusympäristöt.



Kuva 4 Qt Creator käyttöliittymä

Kuvassa vasemmalla on lista eri komponenteista. Näitä ovat esimerkiksi nappi tai tekstikenttä, jotka on yllä olevassa kuvassa asetettu tyhjään ikkunaan. Qt Creatorissa vetämällä vasemmalta drag-and-drop menetelmällä voi ikkunaan lisätä komponentteja, joita voi sitten muokata tarkemmin. Komponentteihin voi myös liittää tapahtumia, kuten esimerkiksi nappiin ”clicked” eli painettu tapahtuman, joka luo ohjelmakoodiin vastaavan funktion. Mitä nappi tekee, kun sitä painetaan, voidaan sitten toteuttaa käyttäen C++ ohjelmointikieltä.

```

1  #include "gui.h"
2  #include "ui_gui.h"
3
4  GUI::GUI(QWidget *parent) :
5      QMainWindow(parent),
6      ui(new Ui::GUI)
7  {
8      ui->setupUi(this);
9  }
10
11 GUI::~GUI()
12 {
13     delete ui;
14 }
15
16 void GUI::on_pushButton_clicked()
17 {
18     //Tähän kirjoitettu ohjelmakoodi toteutuu, kun nappia painetaan.
19 }
20

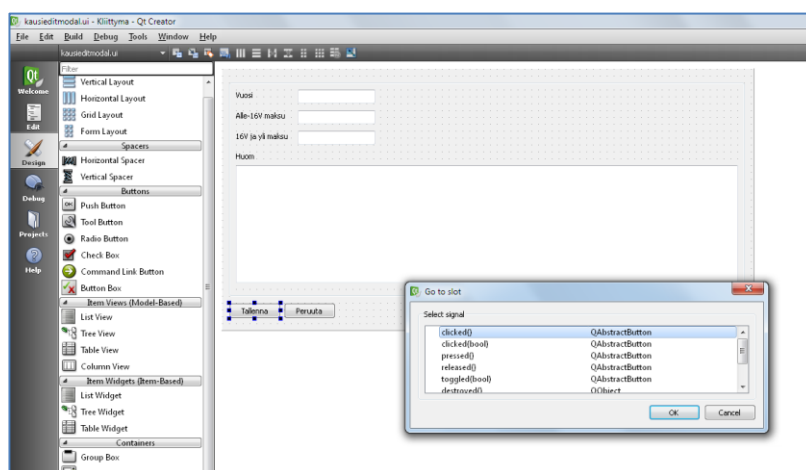
```

Kuva 5 Esimerkki Qt ohjelmakoodista

## 7.4 Ohjelmalogiikan toteutus

Kuten yllä jo mainittiinkin, on Qt Creatorissa mahdollisuus toteuttaa ohjelmalogiikka käyttäen komponentteihin liitettyjä tapahtumia, joiden ohjelmalogiikka toteutetaan käyttäen C++-kieltä.

Jos esimerkiksi halutaan lisätä tietokantaan tietoa, voidaan ikkunaan lisätä eri tekstikenttiä joihin käyttäjä voi kirjoittaa tietoa. Voidaan myös lisätä nappi jota painettaessa toteutuu käyttäjän toteuttama koodinpätkä. Koodissa voidaan sitten esimerkiksi hakea tekstikentissä olevat tiedot, muodostaa tiedoista SQL-lause, ottaa yhteys tietokantaan ja lopuksi ajaa muodostettu SQL-koodi, joka lisää tekstikentissä olevat tiedot tietokantaan.



Kuva 6 Esimerkki Qt clicked tapahtuman asettamisesta nappiin

## 8 TYÖN ESITTELY JA TULOKSET

Tässä kohdassa kuvaillaan mitä opinnäytetyössä rakennettu sovellus eli jäsenrekisteriä ylläpitävä ohjelma tekee. Alla on listattu eri ominaisuuksia, ja vähän selitetty tarkemmin niiden toimintoja.

### 8.1 Ohjelman yleiset ominaisuudet

Yleisesti jäsenrekisterin on tarkoitus tehdä seuraavia asioita.

#### **Jäsenet**

- Lisätä uusia jäseniä rekisteriin, muuttaa jäsenten tietoja tai poistaa jo olemassa olevia jäseniä rekisteristä.
- Mahdollistaa jäsenten listauksen ja listauksen tarkentamisen käyttäen monipuolista hakutoimintoa.
- Kauden vaihtuessa siirtää valitut jäsenet vanhalta kaudelta automaattisesti uudelle kaudelle.

#### **Kaudet**

- Lisätä uusia kausia, muuttaa jo olemassa olevia tai poistaa vanhoja kausia
- Näyttää listaus käytössä olevista kausista.

#### **Ryhmät**

- Lisätä uusia ryhmiä, muuttaa jo olemassa olevia tai poistaa vanhoja ryhmiä.
- Näyttää listaus käytössä olevista ryhmistä.

#### **Roolit**

- Lisätä uusia rooleja, muuttaa jo olemassa olevia tai poistaa vanhoja rooleja.
- Näyttää listaus käytössä olevista rooleista.

**Raportit**

- Tuottaa tulostettavia raportteja jäsenistä joko automaattisesti ryhmittäin, rooleittain, kausittain tai sen mukaan, onko jäsen maksanut kaikki maksunsa vai ei.
- Tai vaihtoehtoisesti tuottaa tulostettava raportti käyttäen monipuolista hakutoimintoa, jolla voidaan hakea raporttiin vain halutut jäsenet.
- Tulostaa raportti paperimuotoisena.

**Laskut**

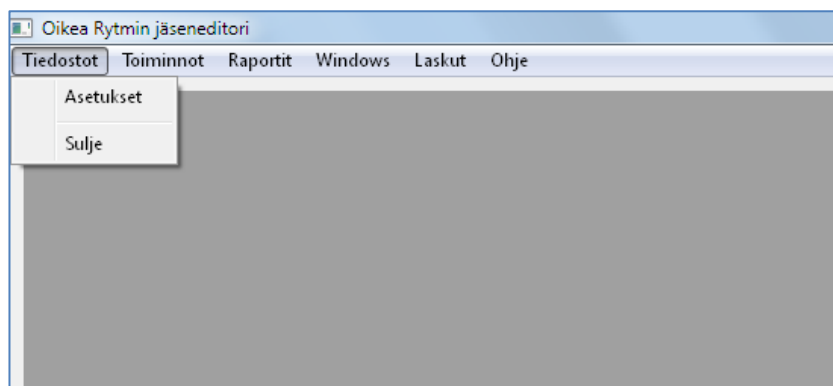
- Tulostaa tyhjä jäsentilisiirto johon ohjelman käyttäjä voi käsin kirjoittaa haluamansa tiedot
- Tuottaa automaattisesti jäsenen lasku käyttäen hyväksy jäsenrekisteriin tallennettuja tietoja jäsenistä, kausista, ikätiedoista ja maksusummista.

Nämä ovat sovelluksen perustoiminnot. Näitä käyttäen on pyritty tekemään helpoksi ylläpitää jäsenrekisteriä, tulostaa raportteja jäsenistä ja helpottaa jäsenten laskutusta. Alla on tarkempi listaus ohjelman eri toiminnoista.

## 8.2 Tiedostot

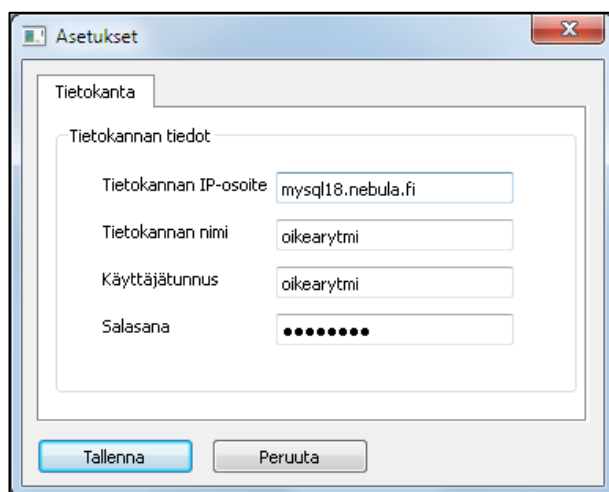
### Tiedostot

Tässä on vain kaksi toimintoa. ”Asetukset” avaa asetusikkunan, jossa määritellään tietokannan tiedot. ”Sulje” sammuttaa ohjelman.



Kuva 7 Tiedostot menu

### Asetukset



Kuva 8 Tietokannan asetukset

Asetukset ikkunassa voi määrittellä sen mitä tietokantaa käytetään. Tässä ikkunassa voi asettaa tietokannan IP-osoitteen, itse tietokannan nimen ja käyttäjätunnuksen ja salasanan, jolla tietokantaan otetaan kiinni. Nämä tiedot tallennetaan oikearytmi.ini tiedostoon ja salasana kryptataan, jotta se säilyy salaisena.



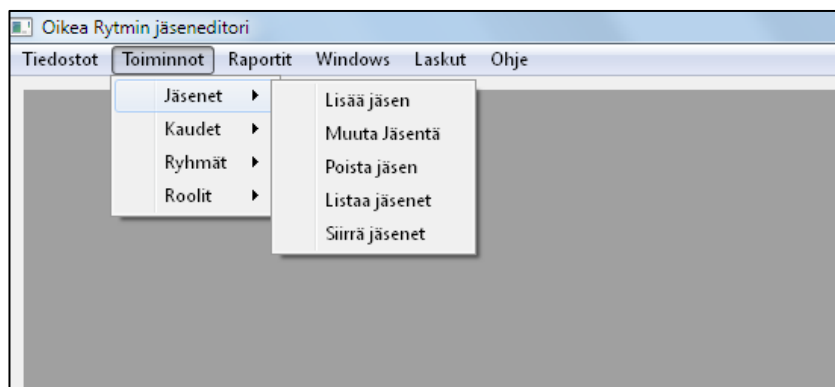
## Sulje

Sulje sammuttaa jäseneditorin.

## 8.3 Toiminnot

### Jäsenet

Jäseneditorissa muokataan itse jäsentietoja. Tässä voi lisätä, muuttaa ja poistaa jäseniä sekä listata ja myös siirtää jäseniä kaudelta toiselle.



Kuva 9 Toiminnot/Jäsenet menu

## Lisää jäsen

**Lisää uusi jäsen**

Jäsenno:  Postitoimipaikka:  Syntymäaika: 1.1.2000

Etinimi:  Puhelin:  Laskettu ikä: 12

Sukunimi:  Email:

Katuosoite:  Huoltaja:

Postinumero:  Sukupuoli: M Valmennusmaksut: 0

Jäsenmaksut: 0

Maksut yhteensä: 0

**Roolit**

Hallitus
Ohjaaja
Kilpailija

**Jäsenen ryhmät**

Ryhman nimi	Maksu	Huom

Lisää ryhmä  
Muuta ryhmän tietoja  
Poista ryhmä

**Jäsenen kaudet**

Vuosi	Maksu	Huom	Maksettu	Luotupvm

Lisää kausi  
Muuta kauden tietoja  
Poista kausi

**Huom**

Tallenna    Peruuta

**Kuva 10 Uuden jäsenen lisäys**

Tässä ikkunassa lisätään uusi jäsen. Uutta jäsentä lisättäessä annetaan ensinnäkin perustiedot jäsenestä, kuten jäsennumero, etu- ja sukunimi, osoitetiedot, yhteystiedot, mahdollinen huoltaja, sukupuoli ja syntymäaika. Syntymäajan perusteella lasketaan kuinka vanha jäsen on. On myös mahdollista kirjoittaa ”Huom” kenttään ihan mitä vaan tekstiä, jota käyttäjä haluaa jäsenestä mainita.

Roolit haetaan muualla määrittelystä listasta ja ne voi tässä valita. Jos jäsen on esimerkiksi hallituksen jäsen, voidaan se valita tässä. Tai jos jäsen toimii myös ohjaajana ja opettaa muita niin senkin voi tässä valita.

Jäsenen ryhmät ja kaudet toimivat samalla tavalla. Molemmista näkyy listaus, johon voi lisätä ryhmiä tai kausia, muuttaa niitä tai poistaa jo olemassa olevia rivejä. Painettaessa esimerkiksi ”Lisää ryhmä” nappia aukeaa seuraavanlainen ikkuna:

	vuosi	alimaksu	huom
1	2009	50	-
2	2010	50	-
3	2011	60	-
4	2012	60	-
5	2013	50	-
6	2014	75	-

**Kuva 11 Uuden kauden lisäys jäsenelle**

Tässä on ylempänä mahdollisuus valita vuosi ja asettaa sen oletustiedot alempana oleviin varsinaisiin tietokenttiin. On myös hakutoiminto, jolla voi tarkentaa mitkä vuodet näkyvät (esimerkiksi huom. kentän perusteella). Jos käyttäjä haluaa esimerkiksi lisätä vuoden 2012 oletustiedot voi käyttäjä kaksoisklikata 2012 riviä, jolloin ”Kauden tiedot” kohtaan menee automaattisesti 2012 vuoden tiedot. Tämän jälkeen tietoja voi muokata tai ne voi hyväksyä sellaisenaan. Mainittakoon vielä, että se näkykö tässä alle 16-vuotiaan maksuhinta vai 16-vuotiaan tai yli maksuhinta riippuu jäsenen iästä.

Muutettaessa vanhaa kautta ainoana erona on se, että tietokannasta haetaan valmiiksi vanhat tiedot riippuen siitä mitä kautta ollaan muuttamassa. Jo asetettuja kausia muutettaessa ei näy oletustietoja eikä hakutoimintoa. Vanhan kauden poistossa vain kysytään onko käyttäjä varma ja sitten poistetaan kausi.

Ryhmätiedot toimivat käytännössä ihan samalla tavalla kuin kaudetkin. Ryhmätoiminnossa aukeaa samalla tapaa ikkuna, jossa näytetään oletustiedot. Ikkunaan voi syöttää tiedot käsin tai valita oletustiedon. Hinnat muuttuvat iän perusteella ja niin edelleen. Ryhmien muutos ja poistokin toimivat samoin kuin kausissa.

## Muuta jäsentä

	jäsens_id	jäsenno	etunimi	sukunimi	katuosoite	postinumero	postitoimipaikka	
1	1	1	Aki	Mäkitalo	Valekatu 12X	28660	Pori	050-5
2	2	2	Ari	Kuusimäki	Vanhatie 22	28660	Pori	050-1
3	3	3	Tiina	Kuusimäki	Vanhatie 22	28660	Pori	02-12
4	18	4	Nilu	Näppääjä	Näpintie 1	28100	Pori	123-4
5	30	5	Taisto	Tango	Tangokatu 2	28600	Pori	050-9

Kuva 12 Muutettavan jäsenen valinta

Muutettaessa jäsentä aukeaa ensin ikkuna, jossa oletuksena näkyy kaikki jäsenet. Tästä ikkunasta valitaan jäsen, jota ollaan muuttamassa. Ruudun alemmassa reunassa näkyy ikkuna, jossa on jäsenlista, ja ylempänä on hakukenttiä, joita käyttäen voi tarkentaa mitä jäseniä näkyy. Jos käyttäjä esimerkiksi kirjoittaa hakukenttään ”Sukunimi” arvon ”Kuusimäki”, niin päivittyy jäsenlista automaattisesti näyttämään vain ne jäsenet, joiden sukunimenä on Kuusimäki. Tällä monipuolisella hakutoiminnolla voi helposti etsiä halutun jäsenen. Kun jäsen on löytynyt, niin joko kaksoisklikkaamalla jäsenen nimeä tai valitsemalla se ja painamalla nappia ”Muuta”, aukeaa varsinainen jäseneditori.

Muuta jäsentä

Jasen\_id 1

Jäsenno 1 Postitoimipaikka Pori Liittymis pvm 1.1.2000 0:00:00

Etunimi Aki Puhelin 050-55544433 Syntymäaika 15.5.1990

Sukunimi Mäkitalo Email aki@ei.ole Laskettu ikä 22

Katuosoite Valekatu 12X Huoltaja - Valmennusmaksut 255

Postinumero 28660 Sukupuoli M Jäsenmaksut 55

Maksut yhteensä 310

Roolit

- Hallitus
- Ohjaaja
- Kilpailija

Jäsenen ryhmät

Ryhmanimi	Maksusumma	Huom
A-luokka	25	-
B-luokka	55	-
C-luokka	90	-

Jäsenen kaudet

Vuosi	Maksusumma	Huom	Maksettu	Luotupvm
2011	75	-	Kyllä	14.11.2012 16:0...
2012	55	-	Ei	14.11.2012 16:0...

Huom

Huomio kenttä

Tallenna Peruuta

Kuva 13 Vanhan jäsenen muutos

Tämä jäsenen muutosikkuna toimii käytännössä ihan samalla tapaa kuin jo edellä kuvailtu jäsenen lisäys ikkuna. Ainoana erona on se, että koska kyseessä on jäsenen muutos, niin kaikki tiedot on jo valmiiksi haettu tietokannasta ja täytetty kenttiin. Näitä tietoja voi sitten muuttaa miten haluaa. Ryhmien ja kausien lisäys toimii myös täysin samalla tapaa kuin jäsenten lisäyksessäkin.

## Poista jäsen

Valitse poistettava jäsen

Jäsenno  Puhelin  Maksettu

Etunimi  Email  Huom

Sukunimi  Huoltaja  Liittymis pvm

Katuosoite  Sukupuoli  Roolit

Postinumero  Syntymäaika  Ryhmät

Postitoimipaikka  Kaudet

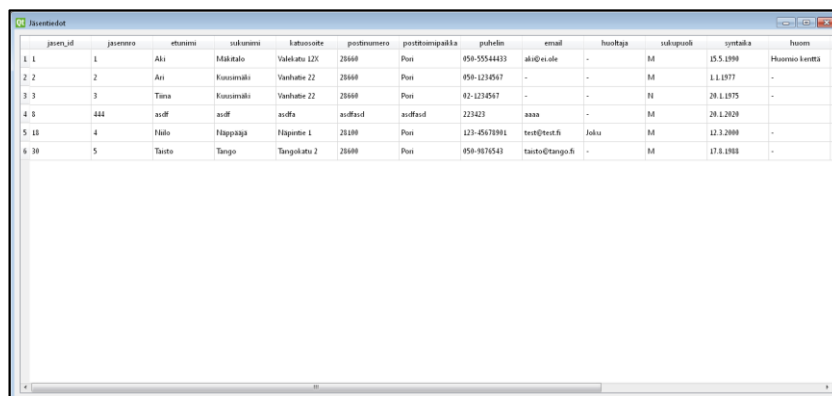
	jasen_id	jasenno	etunimi	sukunimi	katuosoite	postinumero	postitoimipaikka	
1	1	1	Aki	Mäkitalo	Valekatu 12X	28660	Pori	050-5
2	2	2	Ari	Kuusimäki	Vanhatie 22	28660	Pori	050-1
3	3	3	Tiina	Kuusimäki	Vanhatie 22	28660	Pori	02-12
4	18	4	Nilu	Näppääjä	Näpintie 1	28100	Pori	123-4
5	30	5	Taisto	Tango	Tangokatu 2	28600	Pori	050-9

Poista Peruuta Tyhjennä hakukentät

Kuva 14 Poistettavan jäsenen valinta

Poistettaessa vanhaa jäsentä aukeaa ensin ikkuna, jossa valitaan poistettava jäsen. Tämä ikkuna toimii samalla tapaa kuin ikkuna, jossa valitaan muutettava jäsen, eli alempana ruudussa on lista jäsenistä, ja ylempänä on monipuolinen hakutoiminto, jonka kenttiin voi syöttää hakutietoja. Kun jäsen, joka halutaan poistaa, on löydetty, voidaan sitä kaksoisklikata tai valita se ja painaa ”Poista” nappia. Ohjelma vielä kysyy onko käyttäjä varma ja tämän varmistuksen saatuaan poistetaan jäsen tietokannasta.

## Listaa jäsenet

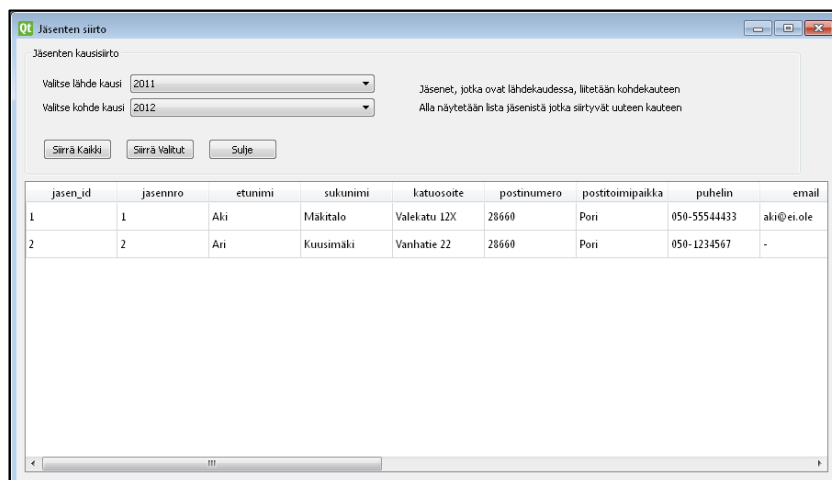


id	jäsenno	etunimi	sukunimi	katuosoite	postinumero	postitoimipaikka	puhelin	email	huoltaja	sukupuoli	syntäika	huono
1	1	Aki	Mäkitalo	Valekatu 12X	28660	Pori	050-5554433	aki@ei.ole	-	M	15.5.1959	Huonoio Lentä
2	2	Ari	Kuusimäki	Vanhatie 22	28660	Pori	050-1234567	-	-	M	1.1.1977	-
3	3	Timo	Kuusimäki	Vanhatie 22	28660	Pori	02-1234567	-	-	M	26.1.1975	-
4	444	asdf	asdf	asdfsd	22343	asdfsd	22343	aaaa	-	M	26.1.2020	-
5	16	Ilmo	Höppäjä	Höppätie 1	21100	Pori	123-4567891	test@test.fi	Joku	M	12.3.2000	-
6	30	Taisto	Tango	Tangokatu 2	28660	Pori	050-9876543	taisto@tango.fi	-	M	17.8.1958	-

Kuva 15 Jäsenten listaus

Tämä toiminto listaa kaikki tietokannan jäsenet ja kaikki jäsenten tiedot. Tämä listaus muistuttaa tavallaan vanhaa Excel-tyylistä näyttöä ja on yksinkertainen tapa tutkia jäsenten tietoja.

## Siirrä jäsenet



Jäsenten siirto

Jäsenten kausisiirto

Valitse lähde kausi: 2011

Valitse kohde kausi: 2012

Jäsenet, jotka ovat lähdekaudessa, liitetään kohdekauteen  
 Alla näytetään lista jäsenistä jotka siirtyvät uuteen kauteen

Siirrä Kaikki Siirrä Valitut Suoje

jäsen_id	jäsenno	etunimi	sukunimi	katuosoite	postinumero	postitoimipaikka	puhelin	email
1	1	Aki	Mäkitalo	Valekatu 12X	28660	Pori	050-5554433	aki@ei.ole
2	2	Ari	Kuusimäki	Vanhatie 22	28660	Pori	050-1234567	-

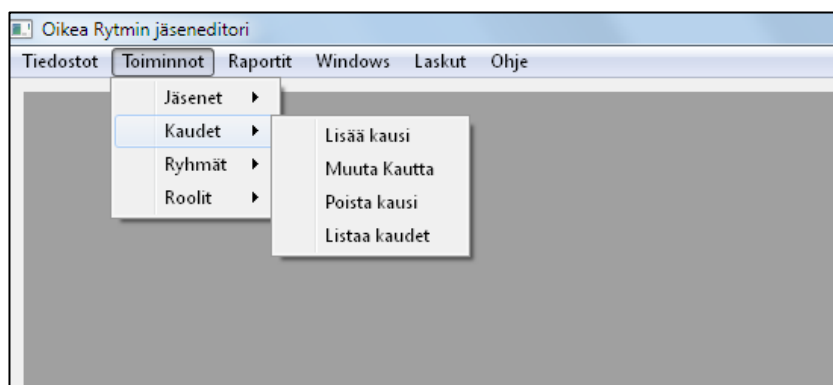
Kuva 16 Jäsenten siirto

Tässä ruudussa voidaan automaattisesti siirtää jäseniä määritetyltä kaudelta toiselle kaudelle. Siirrettäessä jäsenen vanhan kauden tiedot siirtyvät automaattisesti uudelle kaudelle ja vain vuosi vaihtuu. Jos esimerkiksi on 10 jäsentä, jotka kaikki ovat jäseniä kaudella 2012, ja nämä jäsenet halutaan siirtää uudelle kaudelle 2013, niin lähdekausi kenttään valitaan arvo 2012. Tällöin ikkunaan päivittyy lista kaikista jäsenistä, jotka ovat kaudella 2012. Kohdekausi kohtaan asetetaan sitten arvo 2013 (tai haluttaessa muukin vuosi), joka kertoo mille kaudelle jäseniä on tarkoitus siirtää. Lopuksi

voidaan ”Siirrä Kaikki” nappia painamalla siirtää kaikki näkyvät jäsenet kaudelle 2013 tai haluttaessa valita klikkaamalla vain tietyt jäsenet ja siirtää vain ne uudelle kaudelle painamalla ”Siirrä Valitut” nappia. Tämä toiminto helpottaa uudelle kaudelle siirtymistä.

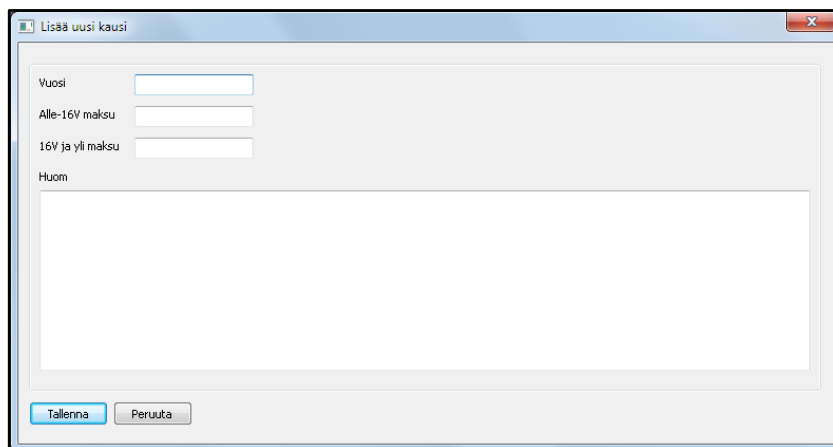
## Kaudet

Kaudet ovat käytännössä vuosia.. Näitä kausia voidaan liittää jäseniin ja tässä kohtaa muokataan oletuskausia.



Kuva 17 Toiminnot/Kaudet menu

## Lisää kausi



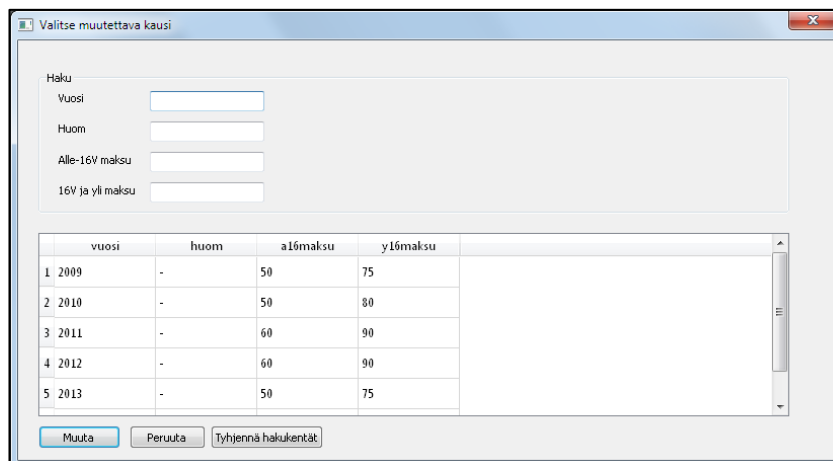
Kuva 18 Uuden kauden lisäys

Tässä ikkunassa on mahdollista lisätä uusi oletuskausi. Tätä tietoa käytetään, kun lisätään kausia jäsenelle. Kuten ylempänä jäseneditorissa kuvattiin, näkyy jäseneditorissa oletustietoja, jotka voi vain valita. Kyseiset oletustiedot asetetaan tässä. Koska lapset, eli alle 16-vuotiaat, saavat yleensä alennetun hinnan verrattuna aikuisiin, voi



tässä kohtaa asettaa kaksi eri maksuarvoa, alle 16-vuotialle ja 16-vuotialle tai sen yli oleville. Se kumpaa arvoa jäsen käyttää lasketaan sitten jäsentiedoissa olevan syntymäajan perusteella.

## Muuta kautta



Valitse muutettava kausi

Haku

Vuosi

Huom

Alle-16V maksu

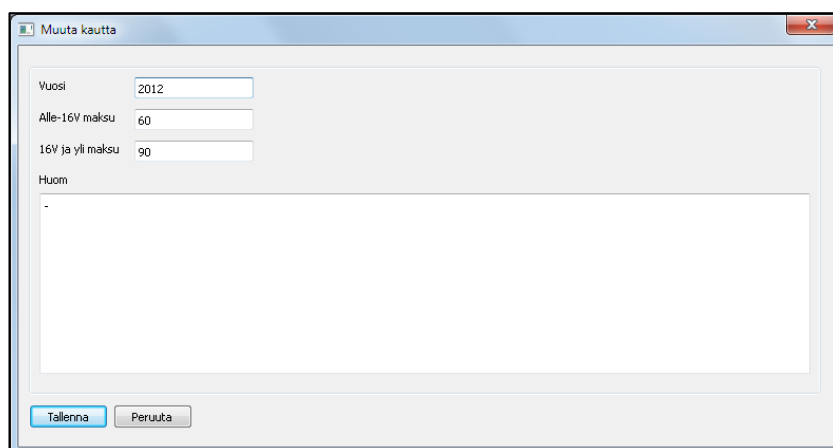
16V ja yli maksu

	vuosi	huom	al16maksu	y16maksu
1	2009	-	50	75
2	2010	-	50	80
3	2011	-	60	90
4	2012	-	60	90
5	2013	-	50	75

Muuta Peruuta Tyhjennä hakukentät

Kuva 19 Muutettavan kauden valinta

Kautta muutettaessa näkyy ensin ikkuna, joka toimii hyvin samoin kuin jäsenten muutoskin. Ikkunassa alempana on lista kaikista kausista ja ylempänä hakukenttiä. Hakukenttiin voi syöttää tietoja tarkentamaan listaa ja kun muutettava kausi on löytynyt, voidaan kaksoisklikkaamalla tai valitsemalla kausi ja painamalla ”Muuta” nappia siirtyä itse kausieditoriin.



Muuta kautta

Vuosi

Alle-16V maksu

16V ja yli maksu

Huom

-

Tallenna Peruuta

Kuva 20 Vanhan kauden muutos

Tämä ikkuna toimii samoin kuin kauden lisäyskin. Ainoa ero on, että vanhat oletustiedot kuten esim. maksutiedot on haettu jo valmiiksi näkyviin. Näitä tietoja voi sitten muutella miten halutaan.

## Poista kausi

	vuosi	huom	al6maksu	yl6maksu
1	2009	-	50	75
2	2010	-	50	80
3	2011	-	60	90
4	2012	-	60	90
5	2013	-	50	75

Kuva 21 Poistettavan kauden valinta

Kausien poisto toimii samoin kuin kausien muutoskin eli ensin näkyy lista kaikista kausista hakutoiminnon kanssa. Kun kausi valitaan, ei kuitenkaan aukea muutosnäyttö, vaan ohjelma kysyy onko käyttäjä varma siitä että kausi poistetaan. Varmistuksen saatuaan ohjelma poistaa vanhan kauden.

## Listaa kaudet

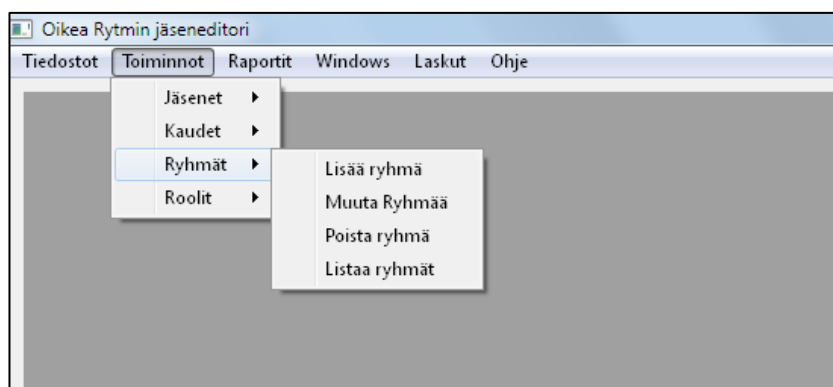
	kausi_id	vuosi	huom	al6maksu	yl6maksu
1	1	2009	-	50	75
2	2	2010	-	50	80
3	3	2011	-	60	90
4	6	2012	-	60	90
5	7	2013	-	50	75
6	8	2014	-	75	100

Kuva 22 Kausien listaus

Tämä listaa kaikki kaudet ja niiden tiedot.

## Ryhmät

Ryhmät sisältävät eri vaikeusasteita, ja hinnat vaihtelevat. Voi olla esimerkiksi ryhmä ”alkeet”, johon kirjataan aloittavat jäsenet. Tai toisaalta voi olla vaikka ryhmä ”A-luokka”, johon kirjataan ne jäsenet, jotka ovat saavuttaneet tietyn taitotason. Ryhmien editointi on lähes identtinen kausien editoinnin kanssa, eli ryhmiä voi lisätä, muuttaa, poistaa ja listata.



Kuva 23 Toiminnot/Ryhmät

## Lisää ryhmä

 The screenshot shows a dialog box titled "Lisää uusi ryhmä". It contains three input fields: "Ryhmänimi", "Alle-16V maksu", and "16V ja yli maksu". Below these is a text area labeled "Huom". At the bottom of the dialog are two buttons: "Tallenna" and "Peruuta".

Kuva 24 Uuden ryhmän lisäys

Tämä lisäysikkuna toimii samalla tapaa kuin kausissakin. Lisättäessä syötetään halutut arvot ja lopuksi tallennetaan ryhmä. Kuten kausissakin, on ryhmissä kaksi eri hintatietoa lapsille, ja aikuisille.

## Muuta ryhmää

ryhman nimi	huom	al6maksu	y16maksu
1. Allkeet-luokka	-	55	70
2. A-luokka	-	50	75
3. B-luokka	-	50	100
4. C-luokka	-	30	90
5. D-luokka	-	75	90

Kuva 25 Muutettavan ryhmän valinta

Kuten kausissakin, näkyy muuttaessa ensin lista kaikista ryhmistä, sekä hakutoiminto, jolla voi tarkentaa mitkä ryhmät näkyvät. Kaksoisklikkaamalla ryhmää tai valitsemalla se ja painamalla ”Muuta” nappia aukeaa ryhmäeditori.

Kuva 26 Vanhan ryhmän muutos

Tässä näkyy ryhmäeditori, johon on haettu vanhat oletustiedot valmiiksi. Näitä voi muutella miten halutaan ja lopuksi tallentaa tiedot.

## Poista ryhmä

	ryhmanimi	huom	al6maksu	y16maksu
1	Alkeet-luokka	-	55	70
2	A-luokka	-	50	75
3	B-luokka	-	50	100
4	C-luokka	-	30	90
5	D-luokka	-	75	90

Kuva 27 Poistettavan ryhmän valinta

Poisto toimii taas samoin kuin muutoskin eli näkyy ikkuna, jossa on hakutoiminto ja lista kaikista ryhmistä. Kaksoisklikkaamalla ryhmää tai valitsemalla se ja painamalla ”Poista” nappia aukeaa varmistus siitä, onko käyttäjä varma, että ryhmä poistetaan. Kun ohjelma saa myöntävän vastauksen, poistaa se vanhan ryhmän.

## Listaa ryhmät

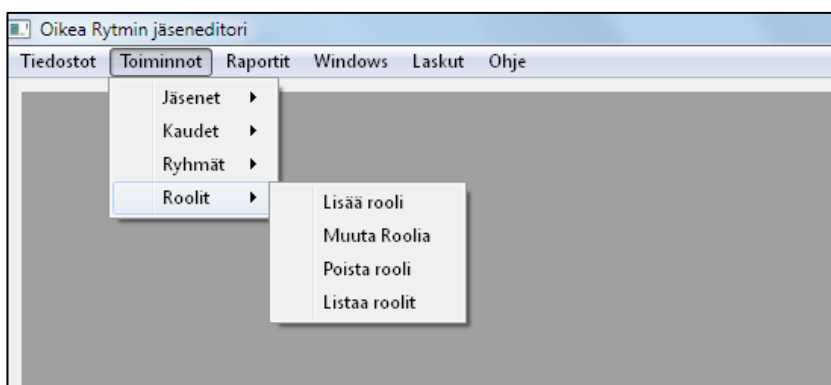
	ryhma_id	ryhmanimi	huom	al6maksu	y16maksu
1	12	Alkeet-luokka	-	55	70
2	13	A-luokka	-	50	75
3	14	B-luokka	-	50	100
4	15	C-luokka	-	30	90
5	16	D-luokka	-	75	90
6	17	E-luokka	-	75	90
7	18	F-luokka	-	55	90
8	19	Jatko-luokka	-	55	75

Kuva 28 Ryhmien listaus

Tämä on lista kaikista ryhmistä ja niiden tiedoista.

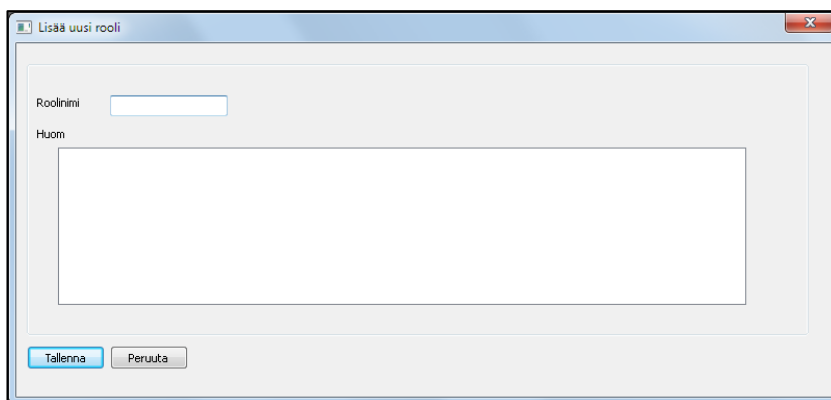
## Roolit

Roolit, kuten nimikin sanoo, ovat eri rooleja joita jäsenellä voi olla. Jäsen voi esimerkiksi olla hallituksen jäsen, jolloin jäsen on mukana roolissa ”hallitus”, tai ehkä jäsen on ohjaaja, jolloin hän on mukana roolissa ”ohjaaja”. Se missä rooleissa jäsen on, valitaan jäseneditorissa, ja tässä kohtaa editoidaan listaa siitä mitä kaikkia rooleja on olemassa.



Kuva 29 Toiminnot/Roolit

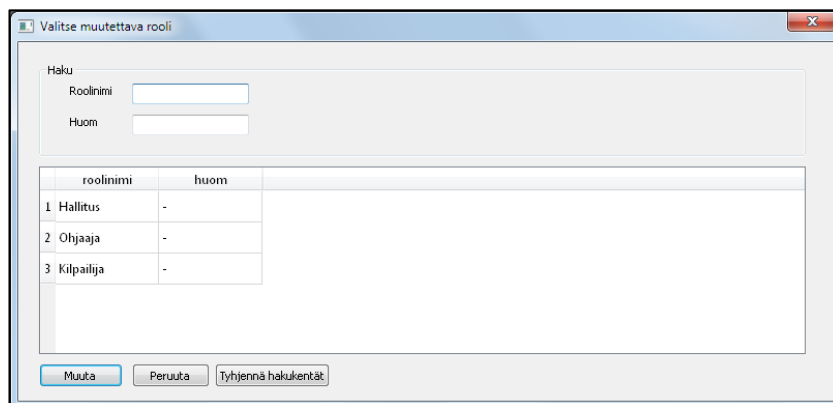
## Lisää rooli



Kuva 30 Uuden roolin lisäys

Yksinkertainen näyttö, jossa lisätään uusi rooli. Syötetään vain roolinimi ja mahdollinen ”Huom” teksti. Muita arvoja ei tässä kohtaa ole.

## Muuta roolia



Valitse muutettava rooli

Haku

Roolinimi

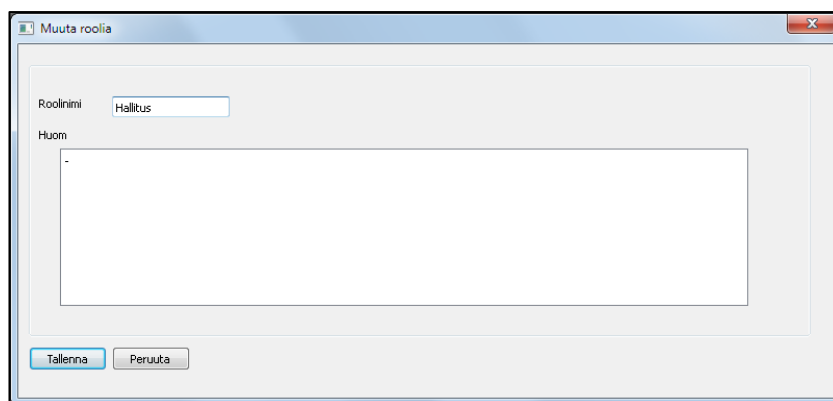
Huom

	roolinimi	huom
1	Hallitus	-
2	Ohjaaja	-
3	Kilpailija	-

Muuta Peruuta Tyhjennä hakukentät

Kuva 31 muutettavan roolin valinta

Kuten ryhmiäkin muutettaessa, näkyy tässä kohtaa ikkuna, jossa on lista eri rooleista ja hakutoiminto. Valitsemalla haluttu rooli listasta, aukeaa muutosikkuna.



Muuta roolia

Roolinimi

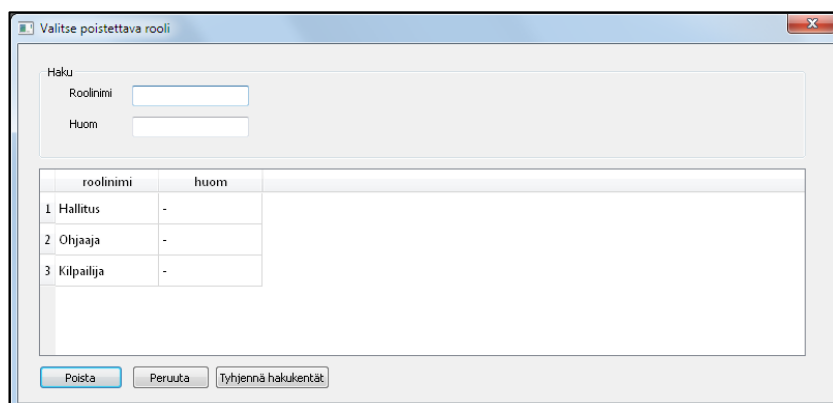
Huom

Tallenna Peruuta

Kuva 32 Vanhan roolin muutos

Tämä ikkuna on identtinen lisäyksen kanssa. Ainoana erona on se, että vanhat tiedot on jo haettu tietokannasta valmiiksi.

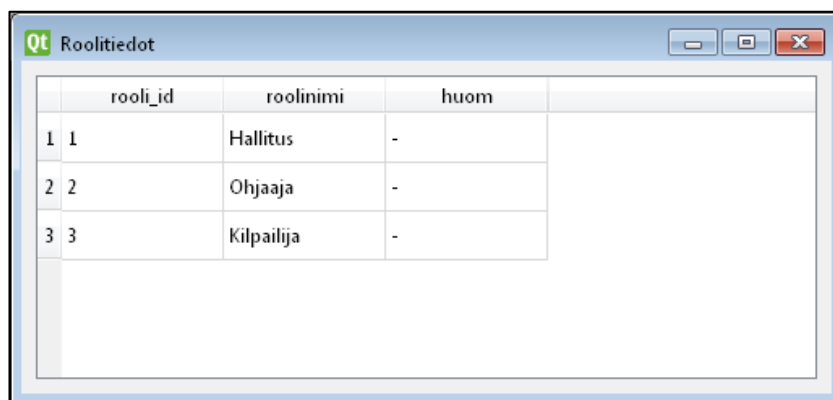
## Poista rooli



Kuva 33 Poistettavan roolin valinta

Poisto toimii taas samoin kuin muutoskin eli näkyy ikkuna, jossa on hakutoiminto ja lista kaikista rooleista. Kaksoisklikkaamalla roolia tai valitsemalla se ja painamalla ”Poista” nappia aukeaa varmistus siitä, onko käyttäjä varma, että rooli poistetaan. Kun ohjelma saa myöntävän vastauksen, poistaa se vanhan roolin.

## Listaa roolit



Kuva 34 Roolien listaus

Tässä listataan kaikki roolit ja niiden tiedot.

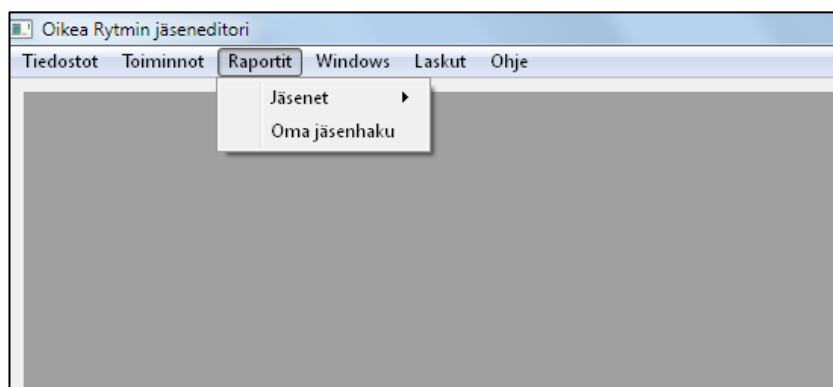


## 8.4 Raportit

### Raportit

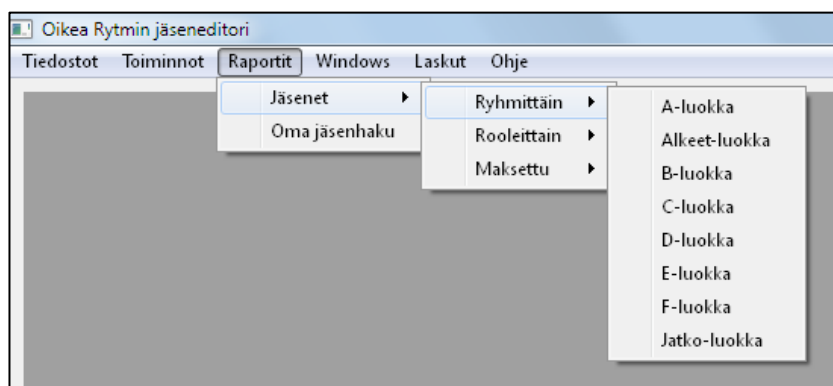
Raportit ovat käytännössä jäsenlistoja ruutumuodossa, joita voidaan tulostaa paperille. Tässä kohtaa on mahdollista valita automaattisesti raportteja siitä, missä ryhmässä tai rooleissa jäsen on tai raportti niistä jäsenistä, jotka ovat joko maksaneet kaikki maksunsa, tai jolla on vielä joku kausi maksamatta.

On myös mahdollisuus tehdä tarkempi raportti käyttäen hakutoimintoa ”Oma jäsenhaku”.



Kuva 35 Raportit menu

### Jäsenet ryhmittäin



Kuva 36 Raportit ryhmittäin valinta

jäsenno	etunimi	sukunimi	katuosoite	postinnumero	postitoimipaikka	puhelin	email	huoltaja	sukupuoli	syntäika	huom	liittymispvm	muutospvm
1	Aki	Mäkilalo	Valekatu 12X	28660	Pori	050-55544433	aj@vei.ole	-	M	15.05.1990	Huomio kerättä	01.01.2000	14.11.2012
2	Ari	Kuusimäki	Vanhate 22	28660	Pori	050-1234567	-	-	M	01.01.1977	-	12.08.2011	08.11.2012

Kuva 37 Raportti ryhmästä

Menussa on mahdollista valita jäsenet ryhmittäin, jolloin raporttiin tulee lista kaikista kyseisen ryhmän jäsenistä. Yllä olevassa kuvassa on esimerkiksi valittu Raportit → Jäsenet → Ryhmittäin → A-luokka, jolloin raporttiin tulee lista jäsenistä jotka ovat ”A-luokka”-ryhmässä. Painamalla ”Tulosta” nappia voi ruudussa näkyvät tiedot tulostaa paperille.

## Jäsenet rooleittain

jäsenno	etunimi	sukunimi	katuosoite	postinnumero	postitoimipaikka	puhelin	email	huoltaja	sukupuoli	syntäika	huom	liittymispvm	muutospvm
1	Aki	Mäkilalo	Valekatu 12X	28660	Pori	050-55544433	aj@vei.ole	-	M	15.05.1990	Huomio kerättä	01.01.2000	14.11.2012
2	Ari	Kuusimäki	Vanhate 22	28660	Pori	050-1234567	-	-	M	01.01.1977	-	12.08.2011	08.11.2012
3	Tina	Kuusimäki	Vanhate 22	28660	Pori	02-1234567	-	-	M	20.01.1975	-	12.08.2011	08.11.2012

Kuva 38 Raportti roolista

Tämä toimii samoin kuin edeltävä ryhmittäin raporttitoiminto. Ainoana erona on, että ryhmien sijaan valitaan rooli. Kuvassa on valittu rooli ”Hallitus”, joten raportissa näkyy jäsenet, jotka ovat hallituksessa. Raportin voi tulostaa paperille.

## Jäsenet sen perusteella onko maksettu

Jäsenno nro	Sukunimi	Kutsumie nimi	Suunnio nro	Sähköpost i-osoite	Sähkö posti	Syntyä päivä	Sukupu li	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	
1	Kui	Makkala	120	20660	Por	050-5554412	and@ui.ole	25.05.1999	puuro harjo	01.01.2000	14.11.2012	Kuusi, Kalle	Kalle	2011,2012	205	85
2	Kui	Kuusimäki	22	20660	Por	050-1234567		01.01.1977		01.08.2011	08.11.2012	Kuusi, Jouko	Kuusi, Jouko	2011,2012,2013	195	82
4	Niilo	Häppälä	1	20100	Por	022-4567890	niilo@niilo.fi	12.03.2000	Siho	05.08.2011	05.11.2012	Kuusi, Kalle	Kalle	2012	75	80
6	Pöytä	Porjo	Kuusimäki	2	20600	Por	050-9876543	porjo@porjo.fi	17.08.1969	05.07.2012	05.11.2012	Kuusi, Kalle	Kalle	2012,2013	80	105

Kuva 39 Raportti niistä, joilla on jotain maksamatta

Tässä raportissa näytetään jäsenet maksutietojen perusteella. On mahdollista näyttää lista jäsenistä, jotka ovat maksaneet kaikki laskunsa tai listata jäsenet joilta on vielä jotain maksamatta. Raportin voi tulostaa paperille.

## Oma jäsenhaku

Jäsenno nro	Sukunimi	Kutsumie nimi	Suunnio nro	Sähköpost i-osoite	Sähkö posti	Syntyä päivä	Sukupu li	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	Jäsen maksu	
1	Kui	Makkala	120	20660	Por	050-5554412	and@ui.ole	25.05.1999	puuro harjo	01.01.2000	14.11.2012	Kuusi, Kalle	Kalle	2011,2012	205	85
2	Kui	Kuusimäki	22	20660	Por	050-1234567		01.01.1977		01.08.2011	08.11.2012	Kuusi, Jouko	Kuusi, Jouko	2011,2012,2013	195	82
4	Niilo	Häppälä	1	20100	Por	022-4567890	niilo@niilo.fi	12.03.2000	Siho	05.08.2011	05.11.2012	Kuusi, Kalle	Kalle	2012	75	80
6	Pöytä	Porjo	Kuusimäki	2	20600	Por	050-9876543	porjo@porjo.fi	17.08.1969	05.07.2012	05.11.2012	Kuusi, Kalle	Kalle	2012,2013	80	105

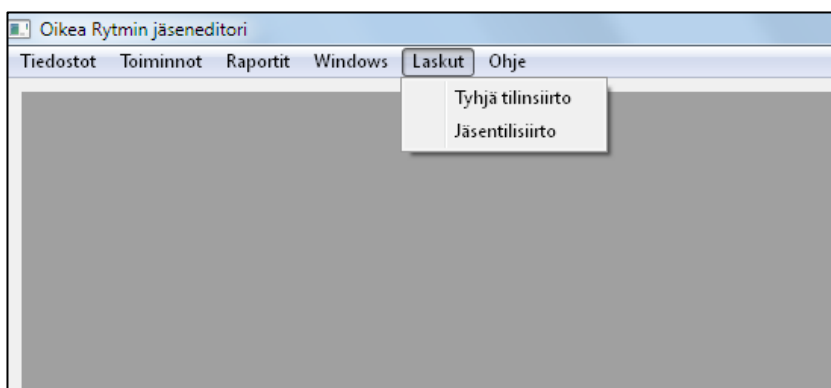
Kuva 40 Raportti hakutoiminnolla

Oma jäsenhaku sisältää ikkunan alemmassa kohtaa raporttimuodossa listan jäsenistä ja ylempänä monipuolisen hakutoiminnon. Hakukenttiin tietoja syöttämällä päivittyy raportti automaattisesti. Tämä toimii samalla tapaa, kuin esimerkiksi jäseneditorin muutostoimintokin. Jos esimerkiksi hakukohtaan ”Sukunimi” syötetään arvo ”Kuusi-  
simäki”, päivittyy raportti niin, että vain ne jäsenet, joiden sukunimi on ”Kuusi-  
simäki”, näkyvät. Kun kaikki halutut hakukentät on annettu, voidaan raportti lopuksi tu-  
lostaa paperille.

## 8.5 Laskut

### Laskut

Tässä kohtaa voi tulostaa joko tyhjän tilisiirron, johon voi sitten käsin kirjoittaa laskutiedot tai voi käyttää jäsentilisiirto-toimintoa, jolloin lasku automatisoidaan käyttäen tietokannassa olevia tietoja siitä missä kausissa ja ryhmissä jäsen on mukana.



Kuva 41 Laskut menu

## Tyhjä tilisiirto

Tilisiirto

Tulosta

**LASKU**

Laskuttaja <b>Tanssiurheiluseura</b>	Päivämäärä	Laskun numero
<b>Oikea Rytmii ry</b>	Viitenumero	
<b>PORI</b>		
Maksaja	Y-tunnus 1025754-7	alv 0 %
	Maksuehdot	
	eräpäivä	
	Huomautusaika	Mivästyskorke
	7 päivää	16 %
	Yhteensä	€
Saajan tilinumero Nordea Pori 200518-45592 IBAN FI12 2005 1800 0455 92	BIC NDEAFIHH	
Saaja Oikea Rytmii ry	Viesti HUOM !	
Maksaja	Maksaessasi käytä viitenumeroa.	
Alekirjoitus	Mitenumero	
Tilitä no	Eräpäivä	€

Tanssiurheiluseura  
Oikea Rytmii ry  
[www.oikearytmi.net](http://www.oikearytmi.net)

Rahastonhoitaja Eeva Juuti  
puh. 040 510 6056  
[eeva.juuti@oikearytmi.net](mailto:eeva.juuti@oikearytmi.net)

Oikkari (seuran sali)  
Peräsimentie 11  
28200 PORI

Kuva 42 Tyhjä tilisiirto

Tämä on tyhjä tilisiirto, jonka voi tulostaa paperille. Se sisältää perustiedot kuten esimerkiksi tilinumerot, Y-tunnuksen ja vastaavat tiedot, jotka eivät muutu, mutta ei sisällä mitään tietoja siitä kenelle lasku on menossa eikä sitä kuinka paljon lasku maksaa. Nämä tiedot pitää sitten kirjoittaa käsin tulostettuun versioon.

## Jäsentilisiirto

The screenshot shows a window titled "Jäsenen tilisiirto" with a form for entering member details and a table listing members. The form includes fields for Jäsenno, Puhelin, Häätty, Etunimi, Email, Hiccn, Sukunimi, Muuttaja, Liityksipvm, Hakausno, Salvoitu, Puhelin, Postinumero, Sotylukko, Puhelin, and Postinpaikka. The table below has columns for jäsen\_id, jäsenno, etunimi, sukunimi, hakausno, postinumero, postinpaikka, puhelin, email, hautilja, sukupuu, syntäjä, hoito, liittymispvm, and eräpäiväm.

jäsen_id	jäsenno	etunimi	sukunimi	hakausno	postinumero	postinpaikka	puhelin	email	hautilja	sukupuu	syntäjä	hoito	liittymispvm	eräpäiväm
1	1	Ale	Mälände	Nahkari 22	2000	Fin	05-5554433	ale@wala	-	M	15.05.1958	Huonekunta	01.01.2000-00.0	04.11.2012 16.8.
2	2	Ari	Kuusimäki	Nahkari 22	2000	Fin	05-1230517	-	-	M	01.01.1977	-	12.08.2011-00.0	01.11.2012 7:49:01
3	3	Timo	Kuusimäki	Nahkari 22	2000	Fin	02-1230517	-	-	M	26.01.1975	-	12.08.2011-00.0	01.11.2012 7:49:00
4	4	Häle	Häppärä	Häppäri 1	2000	Fin	123-4567891	test@wala.fi	Joka	M	12.03.2000	-	05.08.2011 12:4.	01.11.2012 7:45:47
5	5	Taisto	Tango	Tangokatu 2	2000	Fin	05-9876543	taisto@tango.fi	-	M	17.08.1988	-	08.07.2012 10:2.	01.11.2012 7:46:24

Kuva 43 Laskutettavan jäsenen valinta

Tässä kohtaa tehdään maksu automaattisesti tietyn jäsenen tietojen perusteella. Tämän takia pitää haluttu jäsen valita ja valinnassa käytetään jo tässä vaiheessa tuttua valintaikkunaa. Ikkunassa alla näkyy lista jäsenistä ja ylempänä on hakutoiminto, johon tietoja syöttämällä voi tarkentaa jäseniä. Kun haluttu jäsen on löytynyt, voidaan jäsenen lasku avata kaksoisklikkaamalla jäsentä tai valitsemalla jäsen ja painamalla nappia ”Jäsenlasku”. Muista poikkeavaa on se, että ennen kuin lasku aukeaa kysyy ohjelma eräpäivää, joka tulee laskuun. Eräpäivää ei lasketa automaattisesti vaan sen arvo syötetään käsin asiakkaan pyynnöstä.

The screenshot shows a window titled "Syötä laskutiedot" with a form for entering bill details. It includes fields for Eräpäivä, Hintasumma, and Omat hintatiedot. At the bottom, there are buttons for Hyväksy and Peruuta.

Kuva 44 Erikoislaskun tietojen syöttö

Mainittavaa on myös mahdollisuus tehdä asiakkaalle niin sanottu erikoislasku. Normaalisti laskuun tulee kausi ja ryhmätiedot automaattisesti ja hintalistaus ja laskun loppusumma lasketaan kaikki automaattisesti jäsentietojen perusteella. Jos kuitenkin painetaan nappia ”Erikoisjäsenlasku”, on käyttäjän mahdollista syöttää manuaalisesti hintatiedot ja hintasumma, jotka tulevat laskuun. Osoitetiedot ja vastaavat tulevat kuitenkin edelleen jäsenen tiedoista.

Tilisiirto : 1

Tulosta

### LASKU

Laskuttaja <b>Tanssiurheiluseura</b> <b>Oikea Rythmi ry</b> <b>PORI</b>	Päivämäärä 14.11.2012	Laskun numero -
	Viitenumero 1	
Maksaja Aki Mäkitalo Valekatu 12X 28660 Pori	Y-tunnus 1025754-7	alv 0 %
	Maksuehdot eräpäivä	
	Huomautusaika 7 päivää	Mivästyskorke 16 %
Jäsenmaksu 2011		75 €
<b>VALMENNUSMAKSUT</b>		
A-luokka		25 €
B-luokka		55 €
C-luokka		90 €
F-luokka		85 €
	Yhteensä	€ 330 €
Saajan tilinumero Nordea Pori 200518-45592 IBAN FI12 2005 1800 0455 92	BIC NDEAFIHH	
Saaja Oikea Rythmi ry	Mesti HUOM !	
Maksaja Aki Mäkitalo Valekatu 12X 28660 Pori	Maksaessasi käytä viitenumeroa.	
Alekirjoitus	Viitenumero 1	
Tilitä no	Eräpäivä 25.12.2012	€ 330 €

Tanssiurheiluseura  
Oikea Rythmi ry  
[www.oikearythmi.net](http://www.oikearythmi.net)

Rahastonhoitaja Eeva Juuti  
puh. 040 510 6056  
[eeva.juuti@oikearythmi.net](mailto:eeva.juuti@oikearythmi.net)

Oikokari (seuran sali)  
Peräsimentie 11  
28200 PORI

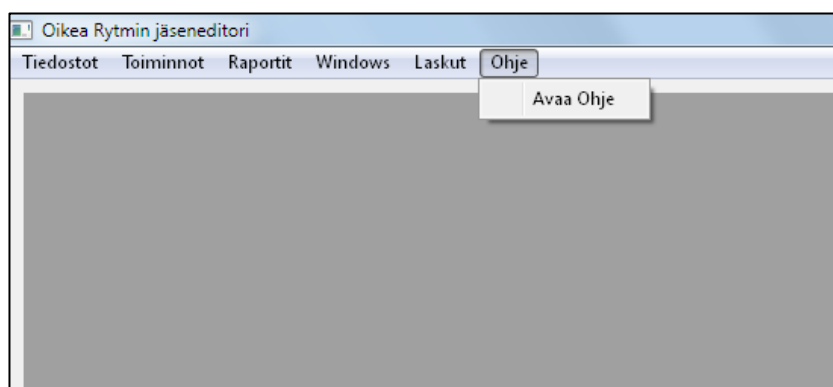
Kuva 45 Täytetty jäsentilisiirto

Kuten kuvasta näkyy, on tässä laskussa asetettu hintatiedot ja muut tiedot automaattisesti valmiiksi. Laskun voi tulostaa paperille ja siihen ei käsin paljon/mitään muutoksia tarvitse tehdä.

## 8.6 Ohje

### Ohje

Tästä aukeaa vain ohje, jossa neuvotaan käyttäjää ohjelman toiminnasta.



Kuva 46 Ohje menu

### Avaa ohje

Tätä painamalla aukeaa erillinen ohje



## 9 YHTEENVETO

### 9.1 Yleistä

Yleisesti ottaen opinnäytetyön tekeminen sujui hyvin. Joitakin ongelmia työn aikana tuli vastaan ja koska Qt oli ohjelmointikieli josta ei ollut kokemusta, oli tietenkin pieni opiskelukynnys kun aloitettiin sovelluksen rakentaminen. Työn tekijä oppi käyttämään Qt:tä, ja samalla tuli myös harjoiteltua lisää SQL:ää ja tietokantojen rakennusta, joista on varmasti apua tulevaisuudessa. Oli myös opettavaista tehdä tällainen projekti alusta loppuun suunnittelusta toteutukseen ja testaamiseen ja dokumentaation kirjoittamiseen. Tästäkin on varmaan apua tulevaisuudessa.

### 9.2 Ongelmia työn aikana

Koko työn aikana suurimpana vaikeutena oli ehkä saada kehitysympäristö toimimaan oikein. Qt Creator ei tukenut oletuksena esimerkiksi MySQL-relaatiotietokantaa ja tuen asentaminen ei ollut mikään helppo asia. Qt Creatorin lähdekoodit piti ladata internetistä ja kääntää kaikki kehityssovellukset MySQL-tuella. Kääntäminen ei kuitenkaan onnistunut heti ja ohjeet olivat hieman heikot. Lopulta eri internetfoorumeilta löytyi tarvittavat tiedot, joidenka avulla saatiin kehitysympäristö pystyyn, mutta tämä asia vei mielestäni liian kauan ja oli liian vaikea. Tosin Qt oli uusi kieli, joten on mahdollista, että kokemuksen puute aiheutti ongelmia. Joskus uudet asiat voivat tuntua vaikeammilta, kuin ne todellisuudessa ovat. On kuitenkin harmillista ettei MySQL tuki ole sisäänrakennettu Qt:ssa. Tämän mahdollisuuden estävät ymmärtääkseni lisenssiyhteensopimattomuudet. (Qt Reference Documentation: SQL Drivers [www-sivut](#), 2012)

Muuten pahempia ongelmia ei ollut. Tietenkin joitakin pieniä vastoinkäymisiä tuli vastaan ohjelmoitaessa, koska Qt oli uusi ja tuntematon ohjelmointikieli, mutta nämä ongelmat yleensä ratkesivat nopeasti joko eri menetelmiä kokeillen tai pyytämällä neuvoja internetissä olevista Qt-keskustelufoorumeista. (Qt Centre [www-sivut](#), 2012)

### 9.3 Muutoksia suunnitelmassa

Alkuperäinen kehityssuunnitelma piti aika hyvin paikkansa, mutta tuli jotain muutoksiakin työtä tehtäessä. Asiakkaan kanssa oltiin yhteydessä jatkuvasti ja ohjelmoija lähetti asiakkaalle testiversioita työn kehittyessä, jotka selvensivät asiakkaalle sitä miten lopullinen sovellus toimii. Nämä testiversiot olivat hyvä idea, koska työn aikana tuli esiin useita väärinymmärryksiä ja muutoksia joita ei ollut huomioitu alkusuunnitelmassa.

Loppujen lopuksi mikään näistä muutoksista ei kuitenkaan ollut hirvittävän iso. Perusidea siitä, mitä sovelluksen kuului tehdä, kehitysympäristö ja ohjelmointikieli eivät muuttuneet. Yleensäkin, kun verrataan lopullista versiota ihan ensimmäisiin mietintöihin asiakkaan tarpeista, niin vastaavat ne aika hyvin toisiaan.

### 9.4 Kehitysideat tulevaisuudessa

Tulevaisuudessa voidaan mahdollisesti muokata käyttöliittymää vielä jos asiakas niin haluaa. On myös suunnitteilla mahdollisuus exportata tietoa jäsenrekistä excel-tiedostoon, jonka sovelluksen käyttäjä voisi sitten tallentaa omalle tietokoneelleen ja tarkastella rauhassa niin sanotussa ”offline-tilassa”.

Myös muut muutokset tai päivitykset ovat tietenkin mahdollisia, jos asiakkaan tarpeet muuttuvat.

## LÄHTEET

Qt Digia: About us. 2012. Viitattu 20.11.2012. <http://qt.digia.com/About-us/>.

Qt Digia: Qt Creator. 2012. Viitattu 1.12.2012.  
<http://qt.digia.com/Product/Developer-Tools/>.

Qt Reference Documentation: SQL Drivers. 2012. Viitattu 1.12.2012.  
<http://doc.qt.digia.com/stable/sql-driver.html>.

Wikipedia. Englanninkielinen. Qt. 2012. Viitattu 20.11.2012.  
[http://en.wikipedia.org/wiki/Qt\\_\(framework\)](http://en.wikipedia.org/wiki/Qt_(framework)).

Qt Centre. 2012. Viitattu 20.11.2012. <http://www.qtcentre.org/content/>.

Johan Telin. 2007. Foundations of Qt Development (Expert's Voice in Open Source). USA: Apress.

MySQL. 2012. Viitattu 20.11.2012. <http://www.mysql.com/>.

w3schools.com. SQL tutorial. 2012. Viitattu 20.11.2012.  
<http://www.w3schools.com/sql/default.asp>.

John J. Patrick. 2008. SQL Fundamentals (3rd Edition). 3. painos. USA: Prentice Hall.

Wikipedia. Suomenkielinen. GNU GPL. 2012. Viitattu 1.12.2012.  
[http://fi.wikipedia.org/wiki/GNU\\_GPL](http://fi.wikipedia.org/wiki/GNU_GPL).

Wikipedia. Suomenkielinen. GNU LGPL. 2012. Viitattu 1.12.2012.  
[http://fi.wikipedia.org/wiki/GNU\\_Lesser\\_General\\_Public\\_License](http://fi.wikipedia.org/wiki/GNU_Lesser_General_Public_License).

Wikipedia. Lisenssi. 2012. Viitattu 1.12.2012. <http://fi.wikipedia.org/wiki/Lisenssi>.

Tanssiurheiluseura Oikea Rytmi ry. 2012. Viitattu 23.11.2012.  
<http://www.oikearytmi.net/>.

