

OPINNÄYTETYÖ

Heidi Tiuraniemi 2012

**REFERENSSITIEKANTA –
SUUNNITTELU JA TOTEUTUS**



Rovaniemen
ammattikorkeakoulu
University of Applied Sciences

LUC

TIETOJENKÄSITTELYN KOULUTUSOHJELMA

ROVANIEMEN AMMATTIKORKEAKOULU
LUONNONTIETEIDEN ALA

Tietojenkäsittelyn koulutusohjelma

Opinnäytetyö

**REFERENSSITIEKANTA –
SUUNNITTELU JA TOTEUTUS**

Heidi Tiuraniemi

2012

Ohjaaja Eija Turunen

Hyväksytty _____ 2012 _____

Tekijä Heidi Tiuraniemi Vuosi 2012

Toimeksiantaja

Työn nimi Referenssitietokanta – Suunnittelu ja toteutus

Sivu- ja liitemäärä 34+ 4

Tämän opinnäytetyön aiheena on referenssitietokanta ja tavoitteena on suunnitella ja toteuttaa toimiva tietokantasovellus. Sovelluksen tarkoituksena on auttaa esimerkiksi opinnäytetyöntekijää pitämään käyttämänsä lähteet ja suorat viiteteksti lainaukset tallessa yhdessä paikassa. Sovelluksen ansiosta opinnäytetyöntekijä voi helposti tallentaa, etsiä ja muokata lähteitä sekä viedä kaikki tallennetut lähteet Microsoft Office Wordille, joka voidaan liittää suoraan opinnäytetyöhön lähdeluetteloksi.

Tässä opinnäytetyössä käytetään Microsoft Office Access-ohjelmaa, koska se on tuttu ohjelma Rovaniemen ammattikorkeakoulun tietokantakursseilta. Accessin lisäksi työssä käytetään myös SQL-ohjelmointikieltä, jolla sovelluksesta saadaan monipuolisempi.

Opinnäytetyössä pohditaan, mikä on tietokannan suunnittelun merkitys tietokantaa rakennettaessa. Valmis tietokanta on helppokäyttöinen referenssitietokanta, jolla voi hallita esimerkiksi opinnäytetyössä käytettyjä lähteitä. Opinnäytetyöntekijän suureksi avuksi on myös lähdeluettelo, jonka Access luo ja vie Wordiin muutamalla napin painalluksella. Hyvin suunniteltu tietokannan rakenne vie työtä eteenpäin ja saa lopputuloksesta selkeän ja toimivan.

Avainsana(t)

Tietokanta, tietokannan suunnittelu, Microsoft Office Access, SQL

Author	Heidi Tiuraniemi	Year	2012
---------------	------------------	-------------	------

Commissioned by

Subject of thesis Reference database – design and realization

Number of pages 34 + 4

The subject of this thesis is reference database and the goal of the thesis is to design and create a workable database application. The purpose of the application is to for example help the author of the thesis to keep all the sources and reference texts in a safe place. Thanks to the application the author of the thesis can easily save, search and edit the sources as well as take all the sources to Microsoft Office Word document which can be added directly to the thesis as a bibliography.

In this thesis the program Microsoft Office Access is used because it is a familiar program from database courses at Rovaniemi University of Applied Sciences. In addition to Access the SQL-programming language is used whereby the application will be more versatile.

In the thesis the meaning of the design when making a database is discussed. The completed database is a user-friendly reference database which can control for example the sources used in the thesis. Great support to the author of the thesis is the bibliography which Access creates and takes to Word with a few presses of a button. Well designed structure of database leads the work and makes the result clear and workable.

Key words

Database, database design, Microsoft Office Access, SQL

SISÄLTÖ

KUVIOLUETTELO	1
1 JOHDANTO	2
2 TIETOKANTAJÄRJESTELMÄT	3
2.1 TIETOKANTARAKENTEET	3
2.3 RELAATIOMALLI/RELAATIOTIETOKANTA.....	4
2.4 REFERENSSITIETOKANTA	5
3 TIETOKANTATYÖKALUT	6
3.1 MICROSOFT OFFICE ACCESS	6
3.2 YHTEYDET.....	7
3.3 YHTEYSTYYPIT	8
3.4 VIITE-EHEYS.....	9
3.5 SQL-KIELI MS ACCESS-TYÖVÄLINEESSÄ.....	10
3.6 VISUAL BASIC-OHJELMOINNIN KÄYTTÖ MS ACCESS-TYÖVÄLINEESSÄ	11
4 TIETOKANNAN SUUNNITTELU	13
4.1 TIETOKANNAN SUUNNITTELUN MERKITYS	13
4.2 ESISELVITYS	14
4.3 MÄÄRITTELYVAIHE	14
4.3.1 Käyttötapauskaavio	14
4.3.2 ER-kaavio.....	15
4.4 SUUNNITTELUVAIHE.....	15
4.4.1 Tietokannan sisällön kuvaus	15
4.4.2 Näyttöjen kuvaukset	16
4.4.3 Raporttien kuvaukset.....	19
5 TIETOKANNAN TOTEUTUS	20
5.1. TIETOKANNAN PERUSTAMINEN	20
5.2 LOMAKKEIDEN JA KYSELYIDEN LUONTI	22
5.3 RAPORTIN JA PÄÄVALIKON LUONTI	27
5.4 TESTAUS.....	29
6 TIETOKANTASUUNNITELMA KÄYTÄNNÖSSÄ	30
7 POHDINTA	31
LÄHTEET	32
LIITTEET	34

KUVIOLUETTELO

KUVIO 1. HIERARKKINEN TIETOKANTAMALLI	4
KUVIO 2. VERKKOTIETOKANTAMALLI	4
KUVIO 3. RELAATIOTIETOKANTAMALLI	5
KUVIO 4. TAULUKKONÄKYMÄ.....	6
KUVIO 5. RAKENNENÄKYMÄ	7
KUVIO 6. VISUAL BASIC OHJELMOINTIYMPÄRISTÖ	12
KUVIO 7. KÄYTTÖTAPAUSKAAVIO	14
KUVIO 8. ER-KAAVIO	15
KUVIO 9. TIETOKANNAN TAULUT	16
KUVIO 10. PÄÄVALIKKO-NÄYTTÖ	16
KUVIO 11. LISÄÄ UUSI LÄHDE-NÄYTTÖ	17
KUVIO 12. LISÄÄ VIITETEKSTI-NÄYTTÖ	17
KUVIO 13. ASIASANOJEN HALLINTA-NÄYTTÖ.....	18
KUVIO 14. SELAA/MUOKKAA LÄHTEITÄ-NÄYTTÖ	18
KUVIO 15. HAKU-NÄYTTÖ	19
KUVIO 16. TIETOKANTA YHTEYDET	20
KUVIO 17. YHTEYKSIEN MUOKKAUS-IKKUNA.....	21
KUVIO 18. YHTEYKSIEN LIITOKSEN LAJI-VALINTAIKKUNA.....	21
KUVIO 19. ASIASANOJEN HALLINTA-LOMAKE	22
KUVIO 20. UUSI LÄHDE-LOMAKE.....	23
KUVIO 21. LISÄÄ VIITETEKSTI-LOMAKE	24
KUVIO 22. SELAA/MUOKKAA LÄHTEITÄ-LOMAKE	26
KUVIO 23. HAKU-LOMAKE	27
KUVIO 24. LÄHTEET-RAPORTTI.....	28
KUVIO 25. PÄÄVALIKKO-LOMAKE	29

1 JOHDANTO

Opinnäytetyöni aiheen sain suorittaessani 402T20A Sovellustuotannon projektityö-kurssia. Koska kurssilla oppimani suunnittelumenetelmä oli mielestäni helppo ja yksinkertainen, päätin sen avulla toteuttaa hieman isomman ja monimutkaisemman tietokannan.

Opinnäytetyöni aihe oli suunnitella ja toteuttaa referenssitietokanta. Tavoitteena oli luoda tietokantasovellus, jonka avulla voi hallita esimerkiksi opinnäytetyössä käytettyjä lähteitä. Sovellus tulisi sisältää lomakkeita, joilla voidaan lisätä, muokata ja poistaa lähdetietoja sekä lähteistä kopioituja viitekstejä. Sovelluksen tärkein osa olisi luoda valmis lähdeluettelo tietokantaan tallennetuista lähteistä. Lähdeluettelo avautuisi suoraan Microsoft Office Wordille, josta se voidaan liittää opinnäytetyöhön.

Tietokantasovelluksen toteutuksessa käytin Microsoft Office Access 2007-ohjelmaa sekä Visual Basic-ohjelmointiympäristöä, jossa ohjelmointikielinä toimivat Visual Basic ja SQL. Valitsin nämä työvälineet, koska ne olivat minulle entuudestaan tuttuja Rovaniemen ammattikorkeakoulun kursseilta.

Lopullinen referenssitietokanta onnistui suunnitelmien mukaisesti. Pohdinkin opinnäytetyössäni tietokannan suunnittelun merkitystä tietokannan toteutuksessa. Tässä opinnäytetyössä suunnitelmien mukaisesti toteutettu tietokanta on selkeä ja yksinkertainen. Sovellus on helppokäyttöinen ja siinä on alussa lueteltujen ominaisuuksien lisäksi mahdollisuus yksilöidä lähteitä asiasanojen avulla. Asiasanojen avulla esimerkiksi lähteen haku helpottuu.

Sovelluksen tärkein osa, lähdeluettelo, oli vaikeampi toteuttaa, koska siihen ei voinut liittää ohjelmointikieltä. Niinpä Accessista tuodun lähdeluettelon ulkoasua täytyy vielä hieman muokata Wordissa.

2 TIETOKANTAJÄRJESTELMÄT

2.1 Tietokantarakenteet

Tietokanta (Database) tarkoittaa tietovarastoa, jossa tiedoilla on yhteys toisiinsa. Tietokanta on yleinen termi tietotekniikassa, mutta esimerkiksi perinteinen kalenterikin voi olla tietokanta. Tietokantoja hallinnoidaan tietokannan hallintajärjestelmillä (Database Management System).

Tietokannan hallintajärjestelmän avulla tietokanta voi olla

- yhteiskäyttöinen: toisin sanoen useat eri sovellukset käyttävät yhteistä kantaa
- ajantasainen: yhden käyttäjän tekemät päivitykset näkyvät heti kaikille muille
- ei-toisteinen: tiedot on tallennettu kertaalleen
- eheä: tiedot ovat ristiriidattomia ja kuvaavat mahdollisimman tarkasti reaalia maailmaa. (Hovi-Huotari-Lahdenmäki 2005, 5)

Tänä päivänä tietokannan hallintajärjestelmät ovat pääosin relaatiotietokantoja (Relational Database). Relaatiotietokannat ovat SQL-pohjaisia (Structured Query Language). Relaatiotietokantoja hallinnoidaan relaatiotietokannan hallintajärjestelmillä, kuten Microsoft Office Access, Oracle ja MySQL.

Ennen relaatiotietokantamallin kehittymistä käytettiin tietojen säilyttämiseen ja käsittelyyn hierarkkista tietokantamallia ja verkkotietokantamallia.

Hierarkkisen tietokantamallin sisältämät suhteet kuvataan termeillä isä/lapsi. Isätaulu voi liittyä moniin lapsitauluihin, mutta lapsitaululla voi olla vain yksi isätaulu. Tällaisesta tietokannasta tietoja voi hakea nopeasti, mutta se vaatii tietokannan rakenteen tuntemisen.

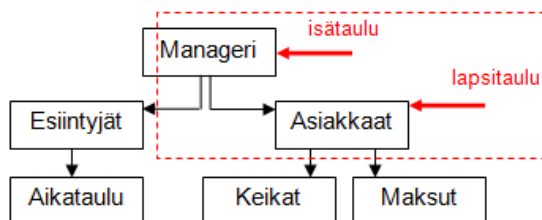
Hierarkkisen tietokannan ongelmien vuoksi kehitettiin verkkotietokantamalli korjaamaan ilmenneitä ongelmia. Verkkotietokantamallissa suhteet kuvataan joukkorakenteen avulla. Joukkorakenteessa kaksi taulua liitetään yhteen niin, että toinen taulu on omistaja ja toinen jäsen. Joukkorakenteet mahdollistavat yhdestä moneen-yhteydet, mikä tarkoittaa, että tietyn joukon omistajataulussa oleva tietue voi olla yhteydessä moniin jäsentaulun tietueisiin, mutta jäsentaulun yksittäinen tietue on yhteydessä vain yhteen omistajataulun tie-

tueeseen (Hernandez 2000, 9). Verkkotietokantamallissa pitää myös tuntea tietokannan rakenne, jotta sieltä voi löytää nopeasti tietoa.

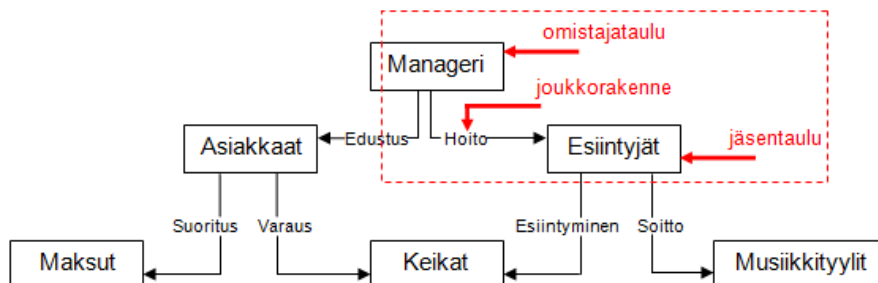
2.3 Relaatiomalli/Relaatiotietokanta

Relaatiotietokannat perustuvat IBM:n tutkija E. F. Coddin vuonna 1970 julkaisemaan relaatiomalliin (the relational model). Relaatiomalli määrittelee, miten tiedot tulisi tallentaa tietokantaan ja miten niitä käsitellään. Relaatiomalli määrittelee myös tietokannan rakenteen, käsittelyn ja eheysäännöt. Relaatiomalli perustuu kahteen matematiikan haaraan; joukkoteoriaan ja prediakaattilogiikkaan.

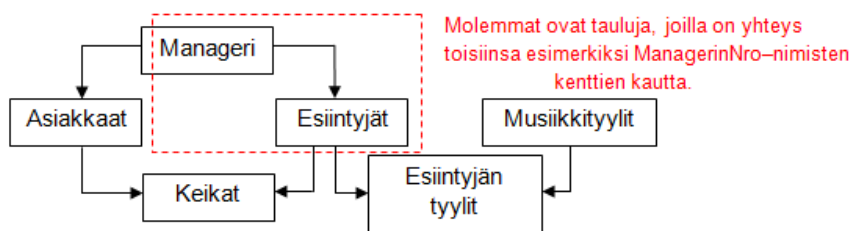
Relaatiotietokantamallissa (Kuvio 3.) relaatio tarkoittaa taulua, johon tiedot tallennetaan. Relaatiot muodostuvat monikoista (tietue) ja attribuuteista (kenttä). Taulun tietueet tunnistetaan kenttien avulla kun kenttä sisältää muista poikkeavan arvon. Tämä mahdollistaa tiedon haun, ilman että käyttäjän tarvitsee tietää tietueen sijaintia. Tämän takia relaatiotietokantamalli eroaa hierarkkisesta tietokantamallista (Kuvio 1.) ja verkkotietokantamallista (Kuvio 2.).



Kuvio 1. Hierarkkinen tietokantamalli



Kuvio 2. Verkkotietokantamalli



Kuvio 3. Relaatiotietokantamalli

2.4 Referenssitetokanta

Referenssitetokanta eli viitetietokanta tarkoittaa tässä tapauksessa opinnäytetyöhön keskittyntä tiedostoa, joka sisältää esimerkiksi käytetyn kirjallisuuden tiedot, tekstiviitteet ja asiasanat.

Referenssi- ja viitetietokantoja löytyy Internetistä useita, mutta vain yksi vastaa opinnäytetyön teon avuksi suunnittelemani tietokantaa.

RefWorks viitteidenhallintaohjelmalla voi muun muassa kerätä, hallita, tallentaa ja jakaa tietoja. Viitteitä voi tallentaa manuaalisesti tai ne voi siirtää esimerkiksi toisesta tietokannasta RefWorksiin sähköisesti. RefWorksissa omia tallentamia viitteitä voi selata, tulostaa tai poimia liitettäväksi omaan tekstiin. Ohjelmalla on mahdollista myös tulostaa valmis lähdeluettelo Microsoft Office Wordille. Muutama miinus tulee ohjelman englanninkielisyydestä ja siitä, ettei tulostettava lähdeluettelo ole Rovaniemen ammattikorkeakoulun opinnäytetyömallin vaatimusten mukainen.

3 TIETOKANTATYÖKALUT

3.1 Microsoft Office Access

Microsoft Office Access 2007 on relaatiotietokannan hallintajärjestelmä. Access-tietokannalla luodaan, muokataan, käsitellään ja ylläpidetään relaatiotietokantoja. Valitsin Access 2007 version siksi, että se on yleisesti opiskelukäytössä oleva ohjelmaversio. Access-tietokannassa on samanlainen käyttöliittymä kuin muissakin Microsoft Office-ohjelmissa, joten muun muassa pikatyökalurivi (Quick Access Toolbar) ja valintanauha (Ribbon) ovat tuttuja.

Tietokantaikkuna, joka avautuu kun tietokanta luodaan tai avataan, on melko helppokäyttöinen käyttöliittymä. Yksinkertaisilla tietokantaohjelmilla pystyy tallentamaan tietoa vain yhteen tauluun. Accessilla, joka on monimutkaisempi tietokanta, pystyy tallentamaan tietoja useampaan toisiinsa liittyviin taulukoihin. Siksi Access on relaatiotietokanta.

Access-tietokanta sisältää erilaisia objektityyppejä. Näitä ovat taulukko, lomakkeet, kyselyt, raportit, makrot ja moduulit. Näistä objekteista vain taulukkoa käytetään tietojen tallentamiseen. Muut objektit ovat tarkoitettu taulukon tallennettujen tietojen hallintaan, käsittelyyn, analysointiin, noutamiseen, näyttämiseen ja julkaisemiseen.

Taulukossa olevat tiedot on järjestetty sarakkeisiin ja riveihin. Niitä kutsutaan kentiksi ja tietueiksi. Jokaisella objektilla on kaksi tai useampi näkymä. Taulukon kaksi yleisintä näkymää ovat taulukkonäkymä (Kuvio 4.) ja raken-
nenäkymä (Kuvio 5.). Taulukkonäkymässä muokataan taulukon tietoja ja raken-
nenäkymässä muokataan taulukon rakennetta.

tyyppi_id	nimi	Lisää uusi kenttä
1	kirja	
2	lehtiartikkeli	
3	opinnäyte	
4	esitelmä/luento	
5	haastattelu	
6	sähköinen lähde	
* tietue	Uusi	

Kuvio 4. Taulukkonäkymä

Kentän nimi	Tietotyyppi
tyyppi_id	Laskuri
nimi	Teksti

Kuvio 5. Rakennenäkymä

Lomakkeiden avulla voidaan syöttää, noutaa, näyttää ja tulostaa taulukoihin tallennettuja tietoja. Luomalla ja suorittamalla kysely, voidaan helposti löytää tarvittavat tiedot Access-tietokannasta. Kyselyn tulosta voi käyttää perustana esimerkiksi raportin luomiseen. Makrot ovat yksinkertaisia ohjelmia, joilla luodaan tietokantatoimintoja. Esimerkiksi lomakkeilla tarvittavat tallenna tai tulosta-painikkeet tehdään makrojen avulla.

Moduulit ovat Microsoft Visual Basic for Applications (VBA)-ohjelmia. Sitä voidaan käyttää monimutkaisten tehtävien suorittamiseen, johon makrot eivät pysty. Tässä opinnäytteessä emme kuitenkaan perehdy tämän tarkemmin VBA:han.

Tietokannan täyttäminen on helppoa. Sen voi tehdä joko tuomalla tietoja toisesta ohjelmasta tai esimerkiksi luodun lomakkeen avulla. Tietoja voi tuoda muun muassa toisesta Access-tietokannasta ja Microsoft Office Excel-laskentataulukosta. Access-tietokannasta voi myös viedä tietoja muihin ohjelmiin. Itse asiassa kaikki Microsoft Office-tuotteet voivat jakaa tietoja eri tavoin.

3.2 Yhteydet

Yhteys tarkoittaa kahden taulun välille luotua liitosta. Yhteyden muodostamiseen tarvitaan perus- ja viiteavain.

Perusavain (Primary Key) on tärkeä osa relaatiotietokannan suunnittelussa. Kussakin taulussa täytyy olla perusavain, joka yksilöi kyseisen tietueen. Yksilöinti mahdollistaa sen, ettei kenttään synny tupla-arvoja. Poikkeustapauksena perusavain voidaan jättää pois, jos tietokannassa tarvitaan aputaulua. Aputauluun syntyy helposti tupla-arvoja.

Kun taulukon perusavain lisätään toiseen tauluun kentäksi, syntyy näiden kahden taulun välille yhteys. Tätä toisen taulun kenttää kutsutaan viite-avaimeksi.

Esimerkki:

Henkilöt-taulun perusavain asiakasnro on tilaukset-taulussa viite-avaimena. Kun tiedot voidaan hakea toisesta taulusta perus- ja viite-avaimen avulla, taulut voidaan pitää siistinä ja yksinkertaisina.

Henkilöt-taulu

<u>asiakasno</u>	<u>etunimi</u>	<u>sukunimi</u>	<u>osoite</u>	<u>paikkakunta</u>
455	Maija	Meri	Kuusitie 4	Kemi
908	Matti	Metsä	Kääpäkuja 789	Kittilä
233	Pirkko	Järvi	Koivukatu 6	Rovaniemi
245	Pekka	Koivu	Lehmuserä 38	Kittilä

Tilaukset-taulu

<u>tilausno</u>	<u>asiakasno</u>	<u>hinta €</u>	<u>maksettu</u>
43	455	39,70	kyllä
45	908	50,00	ei
36	233	109,90	kyllä
29	245	22,40	kyllä

3.3 Yhteystyypit

Taulujen välisiä yhteyksiä eli suhteita (relationship) voi kuvailla kolmella eri tavalla:

Yksi-yhteen-yhteys on melko harvinainen, mutta mahdollinen. Esimerkkinä maan ja pääkaupungin yhteys, jossa jokaisella maalla on vain yksi pääkaupunki ja jokainen pääkaupunki kuuluu vain yhdelle maalle.

Yksi-moneen-yhteys on yleisin käytetty yhteys. Sitä voidaan kutsua myös isä-lapsi-yhteydeksi eli isällä voi olla monta lasta, mutta lapsella vain yksi isä.

Moni-moneen-yhteys on myös yleinen tietokannoissa. Yhteyttä ei kuitenkaan voida muodostaa suoraan, vaan yhteys pitää jakaa kahdeksi yksi-moneen-yhteydeksi. Näiden yksi-moneen-yhteyksien väliin täytyy luoda aputaulu. Esimerkiksi henkilöllä voi olla monta pankkitiliä ja pankkitiliä voi käyttää useampi henkilö.

3.4 Viite-eheys

Viite-eheys (referential integrity) on joukko sääntöjä, joilla varmistetaan että yhdistettyjen taulujen kenttien väliset yhteydet ovat kunnossa ja ettei vahingossa poisteta tai muuteta liitettyjä tietoja.

Viite-eheys voidaan ottaa käyttöön kun seuraavat ehdot täyttyvät:

- Molempien taulujen on kuuluttava samaan Microsoft Office Access-tietokantaan.
- Yhdistettyjen kenttien tietotyypit on oltava samanlaiset. Poikkeuksena laskuri-kenttä voidaan liittää luku-kenttään.
- Ensisijaisen taulukon yhteys kentän on oltava perusavain tai sillä pitää olla ainutkertainen indeksi.

Kun viite-eheys on käytössä, voimassa ovat seuraavat säännöt:

- Taulun viite-avainkenttään ei voi kirjoittaa arvoja, joita ei ole toisen taulun perusavainkentässä.
- Taulusta ei voi poistaa tietuetta, jos sitä vastaavia tietueita on toisessa taulussa.
- Taulun perusavainkentän arvoa ei voi muuttaa, jos kentän tietueisiin on liitetty muita tietueita.
- Kenttien johdannaispäivitys ja tietueiden johdannaispoisto voidaan asettaa päälle.

Kenttien johdannaispäivityksen ollessa valittuna Access tekee automaattisesti muutokset kun päivität perusavainkentän tietoja. Esimerkiksi henkilöt-tilauksen asiakasnumero muuttuu, Access päivittää tiedon myös tilaukset-tilaukseen.

Tietueiden johdannaispoiston avulla voit poistaa tietueita, vaikka ne olisi yhdistetty toiseen tauluun. Esimerkiksi henkilöt-tilauksesta poistetaan yhden asi-

akkaan kaikki tiedot, poistaa Access sen asiakkaan tiedot myös tilauksettaulusta.

3.5 SQL-kieli MS Access-työvälineessä

SQL (Structured Query Language) on 1970-luvun lopulla IBM:n laboratoriossa kehitetty relaatiotietokannan määrittely- ja käsittelykieli. SQL ei ole rakenteellinen kieli, joten sen avulla tehtävät voidaan määrittellä kysymyksellä "mitä" ennemminkin kuin "miten". Esimerkiksi SQL määrittelee mitkä tiedot haetaan tai lisätään, ei niinkään, miten toiminto tapahtuu. (Koulutus- ja konsultointipalvelu KK Mediat 2012.) SQL:n avulla voidaan muun muassa muokata tietokannan rakennetta, suorittaa kyselyjä tietokannasta ja päivittää tietokantaa.

SQL-käskyjä ovat muun muassa:

- CREATE, ALTER, DROP (tietokannan rakenteen määrittely ja muuttaminen)
- SELECT (kyselyt)
- INSERT, UPDATE, DELETE (lisäykset, muutokset ja poistot)

SQL:n käytetyin kyselykäsky SELECT voi koostua seuraavista osista:

- SELECT (mitkä sarakkeet halutaan)
- FROM (mistä taulusta/tauluista/näkymästä/näkymistä haetaan)
- WHERE (mitkä rivit haetaan)
- GROUP BY (miten ryhmitellään)
- HAVING (mitkä ryhmittelyn tuloksena syntyneistä riveistä haetaan)
- ORDER BY (miten lajitellaan). (Hovi 2004, 27.)

Esimerkki:

Henkilöt-taulu

<u>asiakasno</u>	etunimi	sukunimi	paikkakunta
455	Maija	Meri	Kemi
908	Matti	Metsä	Kittilä
233	Pirkko	Järvi	Rovaniemi
245	Pekka	Koivu	Kittilä

Haetaan Henkilöt-taulusta kaikki Kittilässä asuvat:

SELECT etunimi, sukunimi, paikkakunta

FROM Henkilöt

WHERE paikkakunta = 'Kittilä'

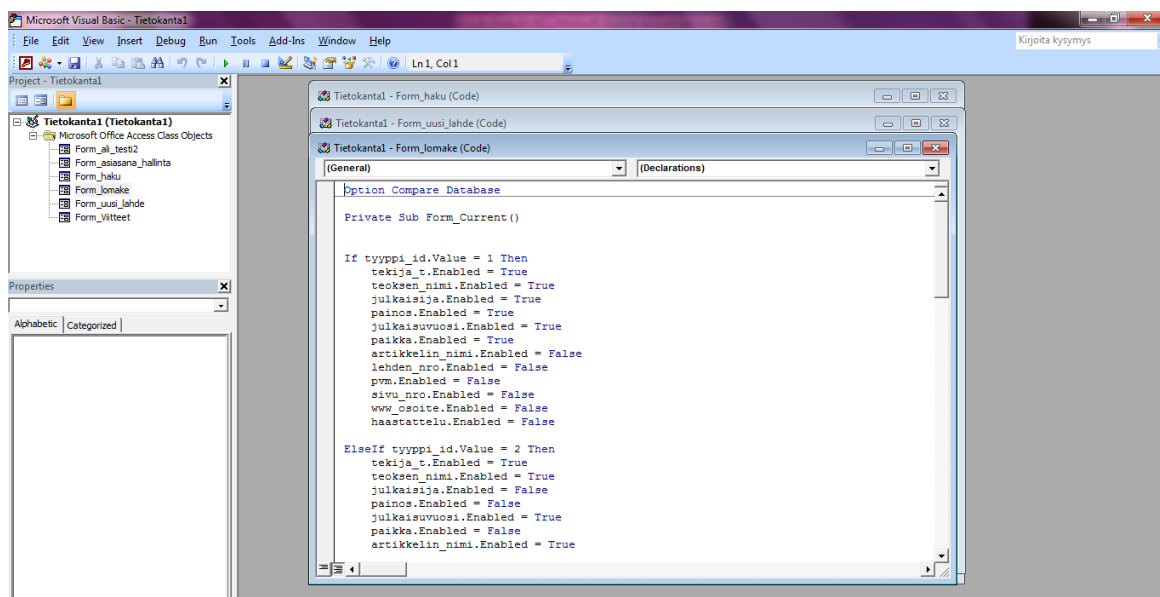
Vastaukseksi saadaan luettelo:

etunimi	sukunimi	paikkakunta
Matti	Metsä	Kittilä
Pekka	Koivu	Kittilä

3.6 Visual Basic-ohjelmoinnin käyttö MS Access-työvälineessä

Visual Basic (VB) on vuonna 1991 Microsoftin julkaisema ohjelmointikieli. Ohjelmointiympäristö, jossa VB ohjelmointikieltä käytetään, on myös nimeltään Visual Basic. Microsoft on julkaissut vuosien aikana VB –kielen versiot 1.0-6.0, jotka toimivat vain Windows-käyttöjärjestelmissä. Vuonna 2002 julkaistiin kielen nykyinen versio Visual Basic.NET (VB.NET), jota käytetään Visual Studio ohjelmointiympäristössä. VB.NET –kieli toimii Microsoftin kehittämässä .NET ympäristössä, mitä varten täytyy ladata .NET Framework (ohjelmistokomponenttikirjasto).

Tässä opinnäytetyössä käytän Visual Basic 6.5-versiota sekä Visual Basic ohjelmointiympäristöä. Koska VB-kielen viimeisin versio on 6.0, VB 6.5 viittaa Visual Basic for Applications (VBA) 6.5 versioon. VBA on isäntäohjelman sisällä toimiva ohjelmointikieli ja se on hyvin samankaltainen VB 6.0 -kielen kanssa.



Kuvio 6. Visual Basic ohjelmointiympäristö

4 TIETOKANNAN SUUNNITTELU

4.1 Tietokannan suunnittelun merkitys

Tietokannan suunnittelu (database design) käsittää laajan kirjon asioita vaatimusmäärittelystä tietokannan mallinnukseen (database modelling) ja fyysiseen suunnitteluun (Hovi ym. 2005, 20).

Suunnittelun merkityksen tärkeys kasvaa mitä laajempi ja monimutkaisempi tietokanta on. Hyvän tietokannan rakenteen keskeisiä ominaisuuksia ovat

- kattavuus: sisältää kaikki järjestelmissä tai kyselyissä tarvittavat tiedot ja yhteydet
- selkeys ja ymmärrettävyys: yksinkertainen rakenne, ilmaisuvoima; helppo kysellä
- muutosjoustavuus: laajennettavuus minimoiden nykyisten ohjelmien muutokset
- yleiskäyttöisyys: soveltuvuus erilaisiin ympäristöihin ja eri asiakkaille tarvitsematta muuttaa tietokannan rakennetta
- eheys: toisteisuuden välttäminen; oikeellisuus; sisäinen ristiriidattomuus
- ohjelmointimukavuus: selkeät tietorakenteet, sarakkeilla kiinteä merkitys (sarakkeen merkitys ei saisi riippua toisesta sarakkeesta)
- suorituskyky eli tehokkuus: riittävä vastausaika tapahtumille ja riittävän tehokkaat eräajot. (Hovi ym. 2005, 21.)

Tietokannan suunnittelu kannattaa ottaa tosissaan, koska huonosti suunnitellusta tietokannasta voi olla vaikea noutaa tiettyjä tietoja. Virheellinen informaatio on kenties vahingollisin huonon tietokantasuunnittelun tulos (Hernandez 2000, 22).

Ainahan voi hypätä suunnitteluvaiheen yli ja oppia virheistä. Tämä vie kuitenkin turhan paljon aikaa. eikä takaa, että tietokannan rakenteesta tulisi toimiva kokonaisuus. Hyvän suunnittelumenetelmän opettelu etuna ovat siitä saatavat taidot suunnitella kestävä tietokantarakenne. Suunnittelumenetelmän avulla oppii erilaisia tekniikoita, jotka auttavat selviämään suunnitteluprosessin alusta loppuun. Suunnittelumenetelmän seuraaminen vähentää virheitä ja tarvetta tehdä suunnitelmia uudelleen.

Tässä opinnäytetyössä käytän 402T20A Sovellustuotannon projektityökurssilla oppimaani suunnittelumenetelmää, joskin hieman sovellettuna.

Suunnittelumenetelmän vaiheet:

1. Esiselvitys
2. Määrittelyvaihe
3. Suunnitteluvaihe
4. Toteutus
5. Testaus

4.2 Esiselvitys

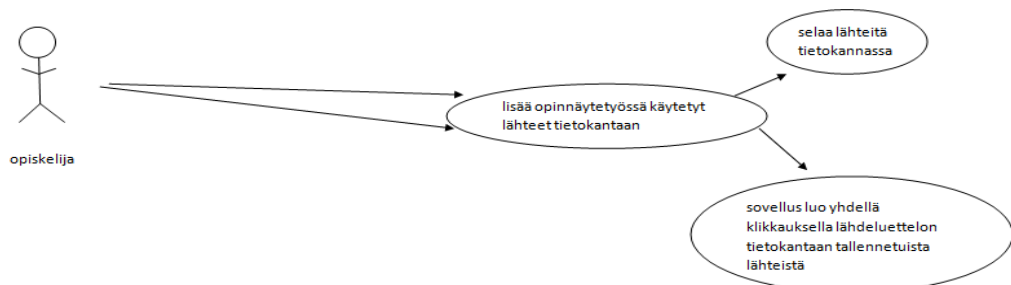
Esiselvityksen tarkoituksena on päättää mitä tehdään, miten tehdään ja miksi tehdään. Tässä opinnäytetyössä päätös oli luoda referenssitietokanta Microsoft Office Access-ohjelmalla. Tietokantasovelluksen tarkoituksena on auttaa esimerkiksi opinnäytetyöntekijää lähteiden hallinnassa.

4.3 Määrittelyvaihe

4.3.1 Käyttötapauskaavio

Määrittelyvaiheessa voi soveltaa esimerkiksi UML (Unified Modeling Language)-kuvaustekniikan käyttötapauskaaviota (Use Case Diagram). Se on selkeä kaavio (Kuvio 7.) käyttäjistä (Actor) ja käyttötapauksista (Use Case) ja se kuvaa kuka tietokantaa tulee käyttämään ja mitä tietoa hän sieltä tarvitsee.

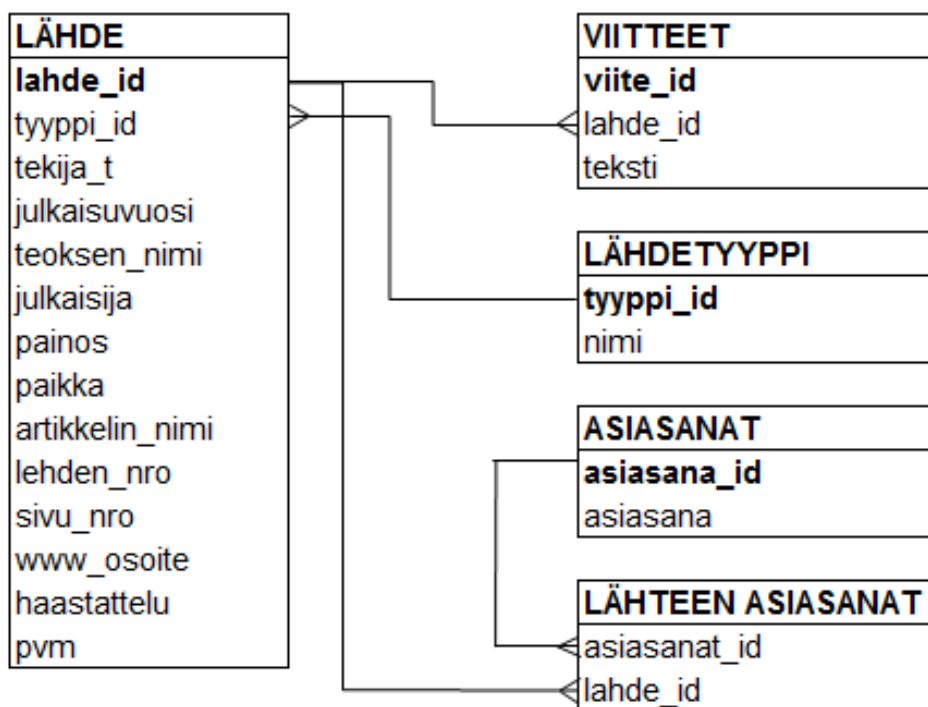
Käyttäjä kuvataan tikku-ukkona, joka voi syöttää tai vastaanottaa tietoa järjestelmästä. Käyttötapaus kuvataan ovaalina ja se sisältää toiminnan tai toiminnan osan.



Kuvio 7. Käyttötapauskaavio

4.3.2 ER-kaavio

Sovelluksen tietorakenteiden suunnittelussa sovelsin ER-kaaviona (Kuvio 8.) kuvattavaa käsittemallia (Entity-Relationship Model). ER-kaavion on kehittänyt Peter Chen vuonna 1976. ER-kaavio on käsitekaavio, jossa kohde-alue mallinnetaan kuvana, mikä on helppo hahmottaa. Kaavio sisältää käsitteitä (entity), yhteyksiä (relationship) ja käsitteisiin liittyviä ominaisuuksia (attribute).



Kuvio 8. ER-kaavio

4.4 Suunnitteluvaihe

4.4.1 Tietokannan sisällön kuvaus

Suunnittelun aluksi päätetään tietokannan taulujen sisältö ja yhteydet (Kuvio 9.). Tietokantaan tulee 5 taulua, joista yksi on aputaulu. Aputaulussa ei ole perusavainta. Jotta tietokannan taulut toimivat oikein, on hyvä nimetä kentät ilman välilyöntejä, ääkkösiä tai muita erikoismerkkejä. Taulujen perusavaimet on kuvassa lihavoituna. Kaikkien perusavainten tietotyypeiksi on valittu laskuri, joka antaa kentän jokaisessa uudessa tietueessa seuraavan käytettävissä olevan numeron. Luku-tietotyyppi on vain luku-muodossa, eli siihen ei voi lisätä tekstiä. Teksti-tietotyyppiin voi lisätä tekstiä, numeroita ja erikoismerkkejä, maksimimerkkimäärä on 255. Pvm./klo-tietotyypin ominaisuuksista voi

valita, missä muodossa päivän tai ajan haluaa esitettävän. Tässä tietokannassa valittuna on lyhyt pvm., joka näyttää päivämäärän muodossa 1.1.2001. Memo-tietotyyppiin voi tallentaa suuren määrän tekstiä, numeroita ja erikoismerkkejä.

LÄHDE-TAULU		
Lähde_id	lahde_id	Laskuri
Tyyppi_id	tyyppi_id	Luku
Tekijä(t)	tekija_t	Teksti
Julkaisuvuosi	julkaisuvuosi	Luku
Teoksen nimi	teoksen_nimi	Teksti
Julkaisija	julkaisija	Teksti
Painos	painos	Teksti
Paikka	paikka	Teksti
Artikkelin nimi	artikkelin_nimi	Teksti
Lehden numero	lehden_nro	Teksti
Sivunumero	sivu_nro	Teksti
www-osoite	www_osoite	Teksti
Haastattelu	haastattelu	Teksti
Päivämäärä	pvm	Pvm./klo

VIITTEET-TAULU		
Viite_id	viite_id	Laskuri
Lähde_id	lahde_id	Luku
Viiteteksti	teksti	Memo

LÄHDETYYPPI-TAULU		
Tyyppi_id	tyyppi_id	Laskuri
Lähdetyypin nimi	nimi	Teksti

ASIASANAT-TAULU		
Asiasana_id	asiasana_id	Laskuri
Asiasana	asiasana	Teksti

LÄHTEEN ASIASANAT-TAULU		
Asiasanat_id	asiasanat_id	Luku
Lähde_id	lahde_id	Luku

Kuvio 9. Tietokannan taulut

4.4.2 Näyttöjen kuvaukset

Suunnittelun toisessa vaiheessa suunnitellaan luotujen tietokannan taulujen perusteella tarvittavat lomakkeet Microsoft Office Excel-ohjelmalla.

Ensimmäisenä suunnittelin päävalikkonäytön ja mitä lomakkeita sitä varten täytyy luoda (Kuvio 10.).

PÄÄVALIKKO

Lisää uusi lähde

Lisää viiteteksti

Asiasanojen hallinta

Selaa/muokkaa lähteitä

Haku

Luo lähdeluettelo

Kuvio 10. Päävalikko-näyttö

Lisää uusi lähde-näytön (Kuvio 11.) kautta lisätään kaikki uudet lähteet. Halutessa lähteeseen voi liittää asiasanoja, jotka helpottavat lähteiden hakemista haku-näytöllä.

Kuvio 11. Lisää uusi lähde-näyttö

Lisää viiteteksti-näytöllä (Kuvio 12.) lisätään viitetekstejä jo tallennettuihin lähteisiin. Viitetekstit ovat siis suoria lainauksia, esimerkiksi kirja-lähteestä, jotka on liitetty opinnäytetyöhön.

Kuvio 12. Lisää viiteteksti-näyttö

Asiasanojen hallinta-näytöllä (Kuvio 13.) lisätään ja poistetaan asiasanoja. Näytölle pääsee päävalikkonäytöstä, jolloin asiasanoja voi tallentaa etukäteen. Asiasanojen hallinta-näytölle pääsee myös lisää uusi lähde- ja selaa/muokkaa lähteitä-näytöltä.

Kuvio 13. Asiasanojen hallinta-näyttö

Selaa/muokkaa lähteitä-näytöllä (Kuvio 14.) voi selata ja muokata kaikkia tallennettuja lähteitä ja niihin liitettyjä asiasanoja ja viitetekstejä. Ainoa, mitä ei pysty muokkaamaan, on näytön vasemmassa yläreunassa oleva lähdetyyppi. Jos valittuna on väärä lähdetyyppi, pitää koko lähde poistaa.

Kuvio 14. Selaa/muokkaa lähteitä-näyttö

Haku-näytön (Kuvio 15.) avulla voi etsiä tietyn lähteen käyttämällä eri haku-ehdoita. Jos esimerkiksi tekijän nimeä ei muista kokonaan, riittää kun kirjoittaa vaikka sukunimen ensimmäisen kirjaimen. Kun hakutulokset antavat oikean lähteen, voi siirtyä muokkaa lähdetä-painikkeen kautta suoraan selaa/muokkaa lähteitä-näytölle.

HAKU-LOMAKE

Tekijä(t)

Teoksen nimi

Asiasana

Lähdetyyppi

Hakutulos:

1.	<input type="text"/>	<input type="button" value="Muokkaa lähdettä"/>
2.	<input type="text"/>	<input type="button" value="Muokkaa lähdettä"/>
3.	<input type="text"/>	<input type="button" value="Muokkaa lähdettä"/>

Kuvio 15. Haku-näyttö

4.4.3 Raporttien kuvaukset

Suunnitteluvaiheen kolmannessa vaiheessa mietitään, millaisia raportteja tietokannasta tullaan saamaan. Tässä opinnäytetyössä tarkoituksena on saada tietokannasta Rovaniemen ammattikorkeakoulun opinnäytetyömallin mukainen lähdeluettelo liitettäväksi valmiiseen opinnäytetyöhön.

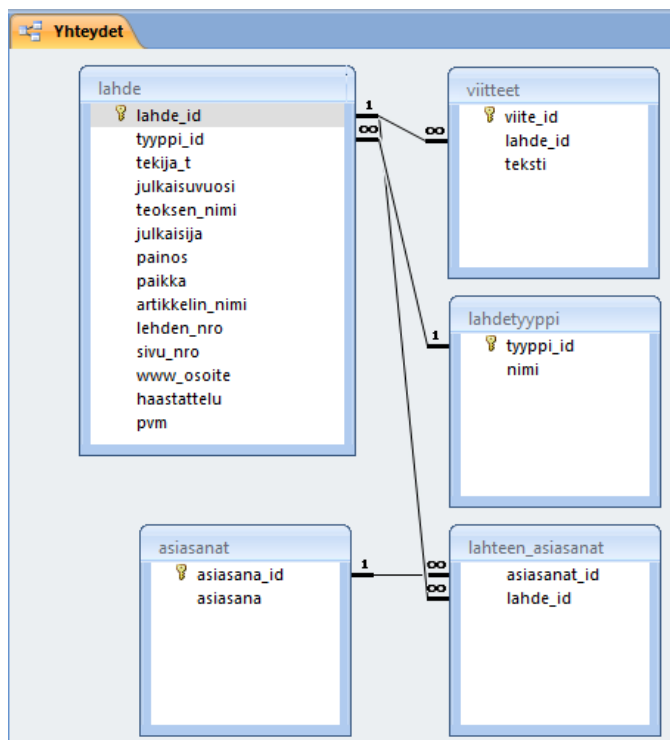
5 TIETOKANNAN TOTEUTUS

5.1. Tietokannan perustaminen

Uuden tietokannan luominen käy helposti Microsoft Office Access 2007-ohjelman avulla. Accessissa on tarjolla valmiita tietokantamalleja, mutta tässä opinnäytetyössä luomme referenssitetokannan tyhjästä tietokannasta.

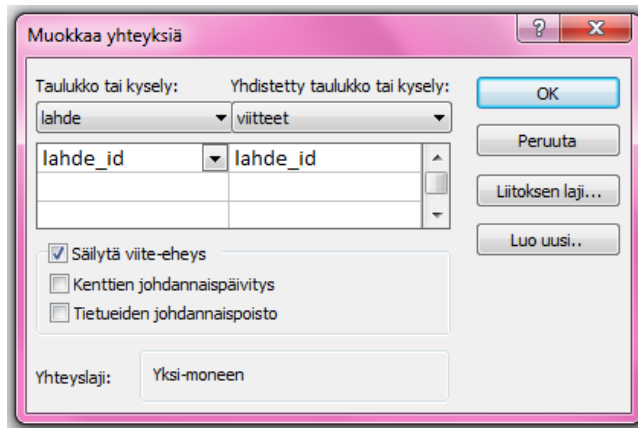
Kun uusi tietokanta on luotu ja nimetty, voidaan aloittaa tietokannan täyttö. Ensin luodaan taulut. Suomenkielisessä Accessissa käytetään termiä taulukko, mutta tässä opinnäytetyössä käytetään termiä taulu. Taulu ja taulukko tarkoittavat samaa. Taulut luodaan luvussa 4.4.1 esitettyjen suunnitelmien mukaisesti. Kun kaikki 5 taulua on luotu, voidaan määrittää taulujen väliset yhteydet.

Yhteydet luodaan perus- ja viite-avainten välille, jotka yleensä on nimetty samannimisiksi. Yhteyksiä luodessa täytyy olla tarkka, jotta yhteys muodostuu oikein. Esimerkiksi kuviossa 16. oleva lahde- ja viitteet-tilujen välinen yksi-moneen yhteys on luotava vetämällä lahde-tilun perusavain lahde_id viitteet-tilun lahde_id-viite-avaimeen. Tällöin saadaan yksi-moneen-suhde, jossa lähteellä voi olla monta viitettä, mutta viitteellä vain yksi lähde.



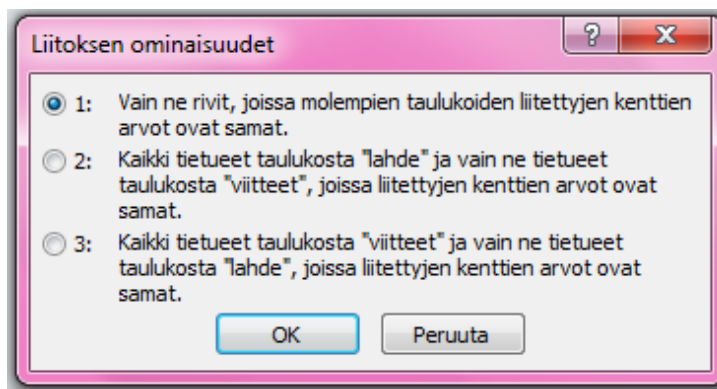
Kuvio 16. Tietokanta yhteydet

Klikkaamalla kahden taulun välissä olevaa, yhteyttä kuvaavaa viivaa, avautuu muokkaa yhteyksiä-valintaruutu. Kuvioista 17. nähdään lahde- ja viitteet-taulujen yhteys, joka muodostuu lahde-taulun perusavaimesta lahde_id ja viitteet-taulun viite-avaimesta lahde_id. Valinta-ikkunan kautta voi myös määrittää liitoksen lajin, säilyttää viite-eheyden, valita kenttien johdannaispäivityksen tai tietueiden johdannaispoiston.



Kuvio 17. Yhteyksien muokkaus-ikkuna

Liitoksen laji-vaihtoehtoja on kolme, joista ensimmäinen vaihtoehto on aina oletusasetuksena (Kuvio 18.). Liitoksen laji kannattaa valita huolella, koska se vaikuttaa kyselyiden lopputulokseen. Liitoksen lajin valinnan jälkeen Access tietää, mitkä tietueet kyselyn tuloksessa näytetään.



Kuvio 18. Yhteyksien liitoksen laji-valintaikkuna

Kuviosta 16. nähdään, että kaikki tietokannan yhteydet ovat yksi-moneen-yhteyksiä. Myös liitoksen laji on kaikilla sama, oletusasetus. Viite-eheys on myös kaikissa yhteyksissä käytössä. Johdannaisasetuksissa on pieniä eroja, lahde-taulun lahteen_asiasanat-yhteydessä on käytössä tietueiden johdan-

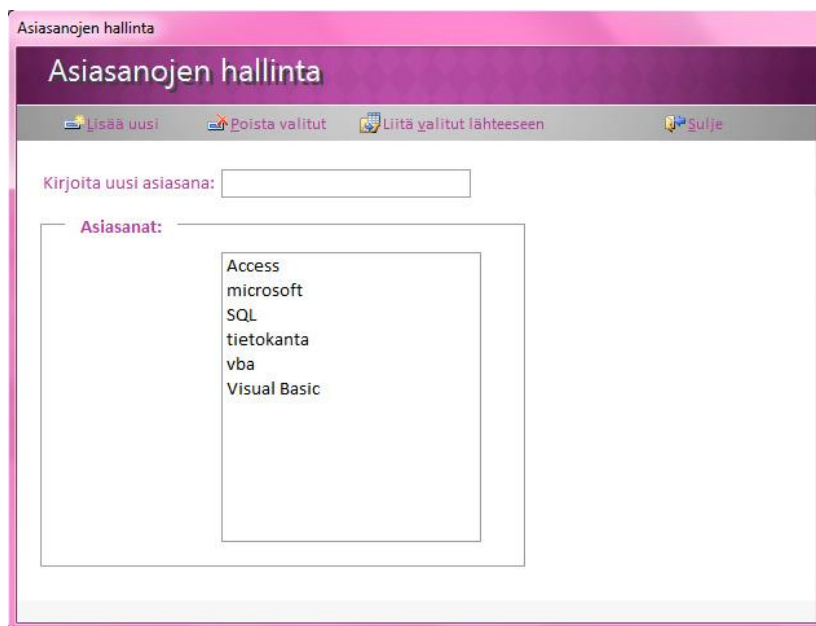
naispoisto ja asiasanat-taulun lahteen_asiasanat-yhteydessä on käytössä sekä kenttien johdannaispäivitys sekä tietueiden johdannaispoisto.

5.2 Lomakkeiden ja kyselyiden luonti

Lomakkeet luodaan ensin Microsoft Office Access-ohjelmalla suunnitelmien mukaisesti ja sen jälkeen niitä muokataan SQL-ohjelmointikielen avulla. Kyselyitä tarvitaan esimerkiksi alilomakkeiden luontiin, joita tarvitaan muun muassa selaa/muokkaa lähteitä- ja haku-näytöillä.

Asiasanojen hallinta-lomake (Kuvio 19.) luodaan ohjatulla toiminnolla. Ohjatun toiminnon avulla lomakkeelle tuodaan kaikki tallennetut asiasanat asiasanat-taulusta. Lomakkeella on yksi piilotettu muokkausruutu asiasanojen suodatusta varten.

Visual Basic-ohjelman kautta voidaan liittää SQL-ohjelmointikieltä tietokannan lomakkeille. Asiasanojen hallinta-lomakkeen SQL on liitteenä opinnäytetyön lopussa (Liite 1).



Kuvio 19. Asiasanojen hallinta-lomake

Uusi lähde-lomake (Kuvio 20.) tehdään ohjatulla lomakkeen luomisella. Ohjatulla toiminnolla lomakkeelle valitaan kaikki kentät lahde-taulusta, joista tyyppi_id ja lahde_id ovat piilotettu. Lähdetyyppi on yhdistelmäruutu ja siihen on liitetty nimi-kenttä lähdetyyppi-taulusta. Kun yhdistelmäruudusta valitaan

esimerkiksi kirja, avautuvat vain kirja-lähteessä tarvittavat muokkausruudut; tekijä(t), julkaisuvuosi, teoksen nimi, painos, paikka ja julkaisija.

Uusi lähde-lomakkeen SQL on opinnäytetyön liitteenä (Liite 2).

Kuvio 20. Uusi lähde-lomake

Lisää viiteteksti-lomake (Kuvio 21.) luodaan tyhjälle lomakkeelle. Lomakkeella on yhdistelmäruutu, josta löytyvät kaikki jo tallennetut lähteiden tekijät ja teosten nimet. Kun tietty lähde valitaan, voidaan ko. lähteelle kirjoittaa ja tallentaa viiteteksti. Yhdelle lähteelle voi tallentaa useamman viitetekstin. Lomakkeella on kaksi piilotettua kenttää, viite_id ja lahde_id.

Lisää viiteteksti-lomakkeen SQL:

Option Compare Database

```
'tallenna viite -painiketta painettaessa
Private Sub tallenna_viite_Click()

'tarkistetaan että lähde on annettu
If IsNull(lahde_id.Value) Then
    MsgBox ("Valitse ensin lähde, jolle viiteteksti lisätään.")
    Exit Sub

Else
    'tallennetaan viite ja lisätty uusi rivi
    On Error GoTo Err
    DoCmd.RunCommand acCmdSaveRecord
```

```

DoCmd.RunCommand acCmdRecordsGoToNew

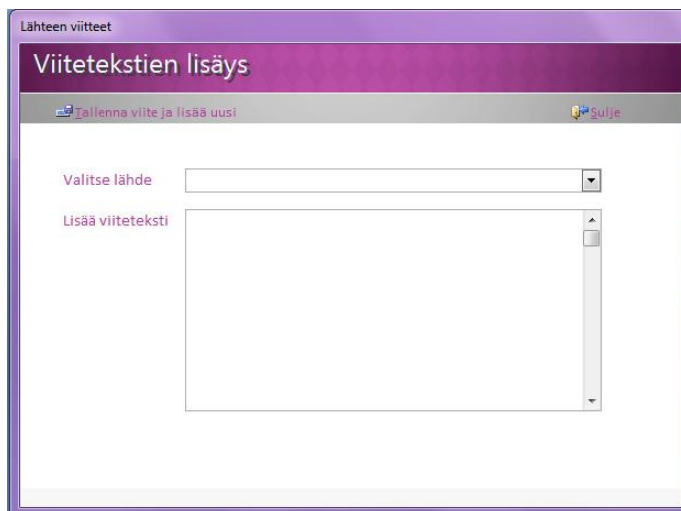
End If

Exit_This:
    Exit Sub

'virheen käsittely
Err:
MsgBox Err.Description
Resume Exit_This

End Sub

```



Kuvio 21. Lisää viiteteksti-lomake

Lähteen selaus/muokkaus-lomake (Kuvio 22.) luodaan ohjatulla toiminnolla. Lomakkeelle valitaan kaikki kentät lahde-taulusta, joista tyyppi_id ja lahde_id ovat piilotettuja. Lahdetyyppi-taulun nimi-kenttä on näkyvässä, muttei käytössä eli sitä ei voi muokata. Myös tällä lomakkeella vain ne kentät ovat käytössä lähdetiedoissa, joita tarvitaan kyseessä olevassa lähdetyypissä.

Viiteteksti(t) on alilomake ja sitä varten on luodaan oma lomake ohjatulla toiminnolla. Lomakkeelle valitaan teksti-kenttä viitteet-taulusta. Lomakkeelle luodaan myös Edellinen ja Seuraava-painikkeet. Viiteteksti-lomakkeen SQL:

```
Option Compare Database
```

```
Private Sub Form_Current()
```

```
'disabloidaa/enabloidaan seuraava ja edellinen -painikkeet
```

```
'sitä mukaa onko nykyinen rivi viimeinen tai ensimmäinen
Me.RecordsetClone.MoveLast
Me.seuraava_viite.Enabled = Not (Me.CurrentRecord =
Me.RecordsetClone.RecordCount)
Me.edellinen_viite.Enabled = Not (Me.CurrentRecord = 1)

End Sub
```

Asiasana(t) luetteloruutuja on lomakkeella kaksi, joista toinen on näkyvä ja toinen piilotettu. Piilotettua luetteloruutua varten luodaan asiasanat kysely, joka tuo alkuperäiset, lähteelle tallennetut asiasanat.

Lähteen selaus/muokkaus-lomaketta varten luodaan myös kolme kyselyä, jotka laskevat lähteiden, viitetekstien ja asiasanojen lukumäärän. Kyselyiden SQL:

Lähteiden lukumäärä-kysely

```
SELECT '(' & Count(0) & ' kpl) ' AS lkm
FROM lahde;
```

Lähteen asiasanojen lukumäärä-kysely

```
SELECT '(' & Count(0) & ' kpl) ' AS lkm
FROM lahteen_asiasanat
WHERE
(((lahteen_asiasanat.lahde_id)=[Forms].[lahteen_muokkaus].[lahde_id]));
```

Lähteen viitteiden lukumäärä-kysely

```
SELECT '(' & Count(0) & ' kpl) ' AS lkm
FROM viitteet
WHERE (((viitteet.lahde_id)=[Forms].[lahteen_muokkaus].[lahde_id]));
```

Selaa/muokkaa lähteitä-lomakkeen SQL on liitteenä opinnäytetyön lopussa (Liite 3).

Kuvio 22. Selaa/muokkaa lähteitä-lomake

Haku-lomake (Kuvio 23.) luodaan tyhjälle lomakkeelle. Lähdetyyppi-hakueto on yhdistelmäruutu, johon tuodaan lähdeyyppi-taulusta nimi-kenttä. Hakutulokset on alilomake ja sitä varten luodaan ensin kysely, johon otetaan mukaan lahde-taulusta lahde_id, tekija_t, teoksen_nimi ja lähdeyyppi-taulusta nimi. Tästä kyselystä luodaan ohjatulla toiminnolla lomake, johon otetaan mukaan vain tekija_t ja teoksen_nimi kentät. Lomakkeelle luodaan myös komentopainike, joka avaa lähteiden selaus/muokkaus-lomakkeen. Lähteiden selaus/muokkaus-lomake aukaisee sen lähteen tiedot, mikä hakutulos on haku-lomakkeella valittu.

Hakutulokset kyselyn SQL:

```
SELECT DISTINCT lahde.lahde_id, lahde.tekija_t, lahde.teoksen_nimi, lahde-
tyyppi.nimi
FROM lahdeyyppi INNER JOIN lahde ON lahdeyyppi.tyyppi_id=lahde.tyyppi_id
WHERE (((lahde.tekija_t) Like '*' & Forms!haku!tekija_t & '*') And
(nz(lahde.teoksen_nimi,'') Like '*' & Forms!haku!teoksen_nimi & '*') And
((lahdeyyppi.nimi) Like '*' & Forms!haku!lahdeyyppi & '*'));
```

Haku-lomakkeen SQL on opinnäytetyön liitteenä (Liite 4).

Lähdetietojen haku

Etsi Tyhjennä hakehdot Sulje

Lähdetyyppi:

Tekijä(t):

Teoksen nimi:

Asiasana:

Hakutulokset:

Lambert, S.	Access 2007 Tehokas hallinta	Muokkaa
Hernandez, M.	Tietokannat suunnittelu ja toteutus	Muokkaa
Hovi, A. – Huotari, J. – Lahdenmäki, T.	Tietokantojen suunnittelu ja indeksointi	Muokkaa
Stephens, R. – Plew, R. – Morgan, B. – P	SQL-tietokantaohjelmointi	Muokkaa
Hovi, A.	SQL-opas	Muokkaa
Halvorson, M.	Visual Basic 2008	Muokkaa

Kuvio 23. Haku-lomake

5.3 Raportin ja päävalikon luonti

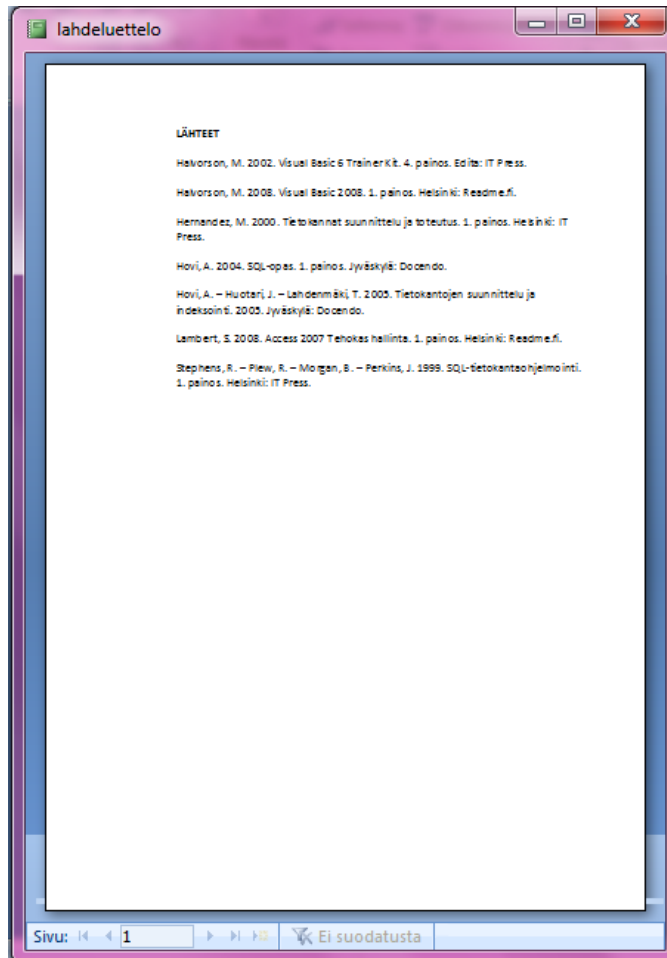
Raporttia varten tehdään ensin kysely, jossa valitaan, mitä tietoja raporttiin tarvitaan. Lähdeluettelo-raporttia (Kuvio 24.) varten tarvitaan kaikki muut kentät lahde-taulusta, paitsi lahde_id ja tyyppi_id-kentät. SQL:n avulla muokataan, missä muodossa tiedot raportille tulostuvat. Lahteet-kyselyn SQL:

```
SELECT Switch(tyyppi_id=1,tekija_t & ' ' & julkaisuvuosi & '. ' & teoksen_nimi & '. ' & painos & '. ' & paikka & ': ' & julkaisija & '.',tyyppi_id=2,tekija_t & ' ' & julkaisuvuosi & '. ' & artikkelin_nimi & '. ' & teoksen_nimi & ' ' & lehden_nro & ', ' & sivu_nro & '.',tyyppi_id=3,tekija_t & ' ' & julkaisuvuosi & '. ' & teoksen_nimi & '. ' & julkaisija & '.',tyyppi_id=4,tekija_t & ' ' & julkaisuvuosi & '. ' & teoksen_nimi & '. ' & 'Esitelmä ' & julkaisija & ' ' & paikka & ' ' & pvm & '.',tyyppi_id=5,tekija_t & ' ' & julkaisuvuosi & '. ' & haastattelu & ' haastattelu ' & pvm & '.',tyyppi_id=6,tekija_t & ' ' & julkaisuvuosi & '. ' & teoksen_nimi & '. ' & 'Osoitteessa ' & www_osoite & '. ' & pvm & '.') & Chr(13) & Chr(10) AS LÄHTEET
FROM lahde
ORDER BY lahde.tekija_t;
```

Raportti luodaan kyselyn pohjalta. Raportin on melkein Rovaniemen ammatikorkeakoulun opinnäytetyömallin mukainen, mutta raportin vieminen Microsoft Office Wordiin tuo eteen ongelmia. Helposti korjattavissa olevat ongelmat liittyvät tekstin ulkoasuun. Fontin tulisi olla Arial, mutta Accessista

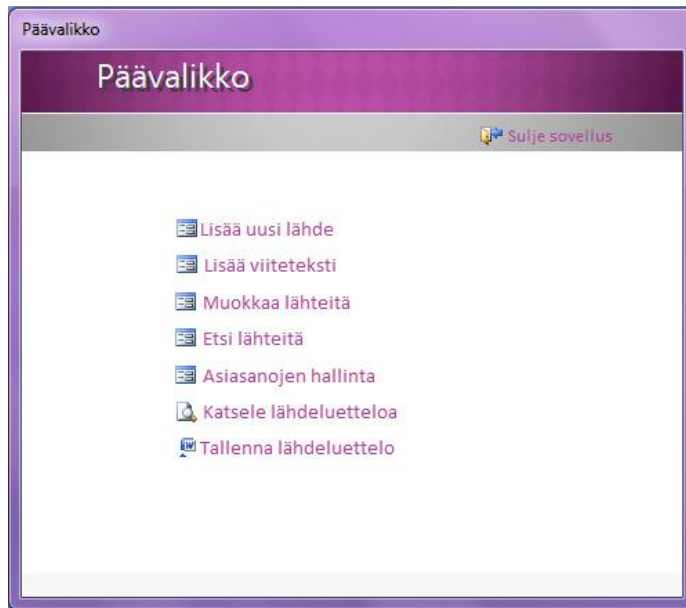
Wordiin tuotaessa ääkköset häviävät, joten teksti täytyy tuoda alkuperäisessä, Calibri, muodossa, ja vaihtaa Wordissa Arialiin. Myös tekstin taseus ja tavutus täytyy tehdä vasta Wordissa. Jos lähde on kahdella rivillä, jälkimmäinen rivi täytyy sientää ja se tehdään vasta Wordissa.

Vaikein ongelma tulee eteen kun Accessin raporttia tarkasteltaessa kaikki näyttää olevan hyvin, mutta Wordiin vietäessä osa tekstistä häviää. Tähän ongelmaan ei löydy vastausta.



Kuvio 24. Lähteet-raportti

Kun kaikki lomakkeet, kyselyt ja raportit ovat valmiina, voidaan luoda päävalikko (Kuvio 25.). Accessissa tätä kutsutaan valikkonäytöksi ja siihen on toiminto, jonka avulla valikkonäytön hallinta sujuu helposti. Tässä opinnäytetyössä valikkonäyttö luodaan kuitenkin normaalilla lomakkeella, koska Tallenna lähdeluettelo-painikkeen lisääminen ei onnistunut Accessin valikkonäyttö hallinnan kautta.



Kuvio 25. Päävalikko-lomake

5.4 Testaus

Testausta on hyvä tehdä koko ajan tietokannan edetessä. Testaaminen on tärkeä osa työtä ja ilman sitä ei tiedetä, saadaanko lopputuloksesta onnistunut. Olisi hyvä antaa jonkun ulkopuolisen tehdä lopullinen testaus, jotta tiedetään, onko sovellus toimiva ja helposti ymmärrettävä.

Tässä opinnäytetyössä testausta tehtiin koko ajan tietokannan edetessä. Tämän opinnäytteen lopussa oleva lähdeluettelo on tehty luodulla tietokantasovelluksella. Lopullista lähdeluettelo on kuitenkin pitänyt hieman muokata, jotta se on Rovaniemen ammattikorkeakoulun opinnäytetyömallin mukainen.

6 TIETOKANTASUUNNITELMA KÄYTÄNNÖSSÄ

Tietokannan suunnittelu ei välttämättä vaadi aikaisempaa kokemusta. Huolellinen tutustuminen tietokannan suunnittelun vaiheisiin ja tietokannan rakenteeseen auttaa työssä eteenpäin. Microsoft Office Access on helppokäyttöinen ohjelma, jolla ensikertalainenkin saa luotua toimivan tietokannan.

Rovaniemen ammattikorkeakoulussa oppimaani suunnittelumenetelmää oli helppo seurata ja kolme ensimmäistä kohtaa, esiselvitys, määrittelyvaihe ja suunnitteluvaihe, valmistuivat nopeasti. Tietokannan toteutus suunnitelmien mukaisesti oli alkuun haastavaa, mutta työn edetessä myös taidot karttuivat.

Tietokantasovelluksen lopputulos oli suunnitelmien mukainen, pieniä yksityiskohtia lukuun ottamatta. Esimerkiksi alkuperäiset Microsoft Office Excel-ohjelmalla suunnitellut lomakkeet olivat yksinkertaisia ja pelkistettyjä sisältäen vain tarpeellisen. Työtä testattiin koko ajan työn edetessä ja lopuksi tietokanta testattiin tässä opinnäytetyössä käytetyillä lähteillä.

Työssä on varmasti kehittämistarpeita, jotka ovat jääneet huomaamatta. Ulkoasu on helposti muutettavissa, rakenteen muuttaminen vaatii hieman enemmän.

7 POHDINTA

Heti alkuun voisi sanoa että, aina ei mene niin kuin on suunniteltu. Opin, että tietokantaa suunniteltaessa täytyy ottaa huomioon ehdottomasti myös se, mitä itse osaa. Matkan varrella tuli vastaan useita ongelmia, joita taitoni eivät riittäneet ratkaisemaan ja aikataulu venyi ja venyi.

Minulla oli jonkin verran kokemusta tietokannoista, mutta monen vuoden tauko käytännön töistä oli tehnyt tehtävänsä ja minun piti tutustua tietokannan suunnitteluun ja toteutukseen perusasioista alkaen. Itseopiskelu ei ole minun vahvimpia alueita, joten kaiken uudelleen oppiminen ei toteutunut niin kuin toivoin. Ratkaisujen etsiminen Internetistä ja avun pyytäminen ongelmatilanteissa sai kuitenkin työn etenemään ja lopulta myös valmistumaan.

Tietokantajärjestelmä on testattu tämän opinnäyteraportin yhteydessä. Mielestäni työn tärkein osa, lähdeluettelon luominen Rovaniemen ammattikorkeakoulun opinnäytetyömallin mukaisesti, ei onnistunut täysin haluamallani tavalla. Tästä huolimatta sovelluksesta tuli toimiva kokonaisuus, koska Wordissa olevan tuloksen ulkoasua voi muokata edelleen annettujen ulkoasuohjeiden mukaan.

Sovellus on mielestäni helppokäyttöinen ja selkeä. Sovelluksen käytöstä voisi olla apua muillekin opinnäytetyön parissa työskenteleville.

LÄHTEET

- About.com 2006. Using MS Access/Access Search Form. Osoitteessa <http://en.allexperts.com/q/Using-MS-Access-1440/Access-Search-Form.htm>. 14.8.2012.
- Bytes.com 2012. Delete message - Can is selectively be turned off?. Osoitteessa <http://bytes.com/topic/access/answers/198835-delete-message-can-selectively-turned-off>. 4.8.2012.
- Demand Media, Inc. 2012. What Is Microsoft Visual Basic 6.5?. Osoitteessa http://www.ehow.com/facts_7244653_microsoft-visual-basic-6_5_.html. 12.9.2012.
- Ekonoja, A. – Lahtonen, T. – Mäntylä, J. 2011. Microsoft Access: lomakkeet ja raportit. Osoitteessa <http://appro.mit.jyu.fi/tiedonhallinta/demot/demo4/#TOC2>. 4.8.2012.
- Halvorson, M. 2008. Visual Basic 2008. 1. painos. Helsinki: Readme.fi.
- Halvorson, M. 2002. Visual Basic 6 Trainer Kit. 4. painos. Edita: IT Press.
- Hernandez, M. 2000. Tietokannat suunnittelu ja toteutus. 1. painos. Helsinki: IT Press.
- Hovi, A. 2004. SQL-opas. 1. painos. Jyväskylä: Docendo.
- Hovi, A. – Huotari, J. – Lahdenmäki, T. 2005. Tietokantojen suunnittelu ja indeksointi. 2005. Jyväskylä: Docendo.
- Itä-Suomen yliopisto 2012. RefWorks. Osoitteessa <http://www.uef.fi/kirjasto/ref-works>. 4.7.2012.
- Jelsoft Enterprises Ltd. 2012. How validate mandatory fields new record data entry form. Osoitteessa <http://www.dbforums.com/microsoft-access/972466-how-validate-mandatory-fields-new-record-data-entry-form.html>. 4.8.2012.
- Koulutus- ja konsultointipalvelu KK Mediat 2012. Johdatus SQL:n maailmaan. Osoitteessa <http://www.2kmediat.com/sql/al-keet.asp>. 30.8.2012.
- Lambert, S. 2008. Access 2007 Tehokas hallinta. 1. painos. Helsinki: Readme.fi.
- Microsoft Corporation 2011. Power tip: count the number of rows in a subform. Osoitteessa <http://blogs.office.com/b/microsoft-access/archive/2010/04/15/>. 4.8.2012.

- Microsoft Corporation 2012. Taulukoiden yhteyksien opas. Osoitteessa <http://office.microsoft.com/fi-fi/access-help/taulukoiden-yhteyksien-opas-HA010120534.aspx>. 9.10.2012.
- MsGroups.Net 2010. Count Records in a SubForm. Osoitteessa <http://msgroups.net/microsoft.public.access/count-records-in-a-subform/39687>. 14.8.2012.
- ProQuest LLC 2012. RefWorks. Osoitteessa <https://www.refworks.com/refworks2/?r=authentication::init>. 4.8.2012.
- Ratol 2012. Access 2003 perusteet - Yleistä. Osoitteessa <http://www.ratol.fi/opensource/ac/yleista/index.html>. 26.9.2012.
- Ratol 2012. Access 2003 jatko - Yhteydet. Osoitteessa <http://www.ratol.fi/opensource/acj/yhteydet/index.html>. 9.10.2012.
- Ratol 2012. Access 2003 perusteet - Yhteydet. Osoitteessa <http://www.ratol.fi/opensource/ac/yhteydet/index.html>. 9.10.2012.
- Savonia ammattikorkeakoulu 2006. UML. Osoitteessa webd.savonia.fi/home/ktrasse/mat_tiedostot/thtietojar/UML.ppt. 12.9.2012.
- Stephens, R. – Plew, R. – Morgan, B. – Perkins, J. 1999. SQL-tietokantaohjelmointi. 1. painos. Helsinki: IT Press.
- Utter Access 2010. Add A 'blank' Row To Combo Box - Query Rowsource. Osoitteessa <http://www.utteraccess.com/forum/-Add-blank-Row-Combo-B-t1939222.html>. 11.8.2012.
- Wikipedia 2012. Tietokanta. Osoitteessa <http://fi.wikipedia.org/wiki/Tietokanta>. 4.8.2012.

LIITTEET

Asiasanojen hallinta-lomakkeen SQL	Liite 1
Uusi lähde-lomakkeen SQL	Liite 2
Selaa/muokkaa lähteitä-lomakkeen SQL	Liite 3
Haku-lomakkeen SQL	Liite 4

ASIASANOJEN HALLINTA-LOMAKKEEN SQL

Liite 1

```
Option Compare Database
```

```
Private Sub Form_Load()
```

```
'tarkistetaan onko näyttö avattu Muokkaa lähde vai Uusi lähde -näytöltä
```

```
If Not IsNull(Me.OpenArgs) Then
```

```
    If CInt(Me.OpenArgs) = 1 Or CInt(Me.OpenArgs) = 2 Then
```

```
        'jos on, asetetaan Liitä valitut -painike näkyviin
```

```
        Me.liita_valitut.Visible = True
```

```
    Else
```

```
        'muutoin Liitä valitut -painike ei näy.
```

```
        Me.liita_valitut.Visible = False
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub lisää_asiasana_Click()
```

```
'muuttujien esittely
```

```
Dim insertSQL As String
```

```
Dim hakutulos As Integer
```

```
Dim SelectSql As String
```

```
Dim db As DAO.Database
```

```
Dim rs As DAO.Recordset
```

```
'tarkistetaan oliko asiasanaa syötetty
```

```
'jos ei ollut annetaan virheilmoitus
```

```
If IsNull(uusi_asiasana.Value) Then
```

```
    MsgBox ("Kirjoita ensin asiasana Lisää asiasana-ruutuun.")
```

```
    Exit Sub
```

```
End If
```

```
On Error GoTo Err
```

```
Set db = CurrentDb
```

```
'asetetaan default varoitusilmoitukset pois näkyvistä
```

```
DoCmd.SetWarnings False
```

```
'asetetaan focus Uusi asiasana -kenttään, jotta voidaan viitata
```

```
'kentän teksti arvoon
```

```
Me.uusi_asiasana.SetFocus
```

```
' tarkistetaan kyselyllä kannasta ettei syötettyä asiasanaa ole jo olemassa
```

```
SelectSql = "SELECT COUNT(*) As lkm FROM asiasanat " & _
```



```

" WHERE UCASE(asiyasana) = UCASE ('" & uusi_asiyasana.Text & "');"

Set rs = db.OpenRecordset(SelectSql, dbOpenSnapshot)
hakutulos = rs("lkm")

' asiasanaa ei ollut
If hakutulos = 0 Then
    ' lisätään uusi asiasana kantaan
    ' tyhjennetään muokkausruutu päivitysten jälkeen
    ' ja päivitetään asiasanat lista, jotta lisätty asiasana näkyy siellä
    insertSQL = "INSERT INTO asiasanat (asiyasana)" & _
    " VALUES ('" & uusi_asiyasana.Text & "');"
    DoCmd.RunSQL (insertSQL)

    uusi_asiyasana.Text = ""
    Me.asiasanat_lista.Requery
    ' asiasana oli jo olemassa
Else
    MsgBox ("Asiasana on jo olemassa.")

End If

' suljetaan recordset ja tietokanta yhteys
rs.Close
db.Close

' näytetään mahdolliset virheet
Err:
If Err.Number <> 0 Then
    MsgBox Err.Number & ":" & " " & Err.Description, , "Error"
End If

End Sub

Private Sub poista_asiyasana_Click()

    ' muuttujien esittely
    Dim deleteSQL As String
    Dim asiasana_id As Integer
    Dim vastaus As Integer

    ' asetetaan default varoituseroitukset pois näkyvistä
    DoCmd.SetWarnings False
    On Error GoTo Err

    ' alustetaan delete lause
    deleteSQL = "DELETE FROM asiasanat WHERE asiasana_id = "

```

```

' tarkistetaan onko valittu rivejä asiasanat listalta
If asiasanat_lista.ItemsSelected.Count = 0 Then
    MsgBox ("Yhtään asiasanaa ei ole valittuna.")
Else
    ' varmistetaan poisto
    vastaus = MsgBox("Haluatko varmasti poistaa valitut asiasanat?",
vbYesNo)
    If vastaus = vbYes Then
        'käydään läpi valitut rivit ja poistetaan ne kannasta
        For Each rivi In asiasanat_lista.ItemsSelected
            deleteSQL = "DELETE FROM asiasanat WHERE asiasana_id = " &
asiasanat_lista.ItemData(rivi) & ";"
            DoCmd.RunSQL deleteSQL
        Next rivi
        ' päivitetään asiasanat lista, jotta poistetut
        'asiasanat eivät enää näy
        Me.asiasanat_lista.Requery
    End If

End If

' näytetään mahdolliset virheet
Err:
If Err.Number <> 0 Then
    MsgBox Err.Number & ":" & " " & Err.Description, , "Error"
End If

End Sub

Private Sub liita_valitut_Click()

' muuttujien esittely
Dim asiasana_id As Integer
Dim asiasana As String
Dim onLiitetty As Boolean

' käydään läpi valitut rivit
For Each rivi In asiasanat_lista.ItemsSelected
    'otetaan talteen asiasana_id ja asiasana
    asiasana_id = Me.asiasanat_lista.Column(0, rivi)
    asiasana = Me.asiasanat_lista.Column(1, rivi)
    onLiitetty = False

    ' tarkistetaan, oliko asiasana jo liitetty
    ' uusi lähde -näytölle, jos avattu sieltä
    If CInt(Me.OpenArgs) = 1 Then

```

```

For i = 0 To Forms!uusi_lahde!asiasanat.ListCount - 1

    If asiasana_id = Forms!uusi_lahde!asiasanat.Column(1, i) Then
        ' asiasana oli jo, ei lisätä uudestaan
        onLiitetty = True
        Exit For
    End If

Next i

' asiasanaa ei vielä ole, lisätään asiasanat listalle
If onLiitetty = False Then
    ' sarakkeet uusi lähde -näytön asiasanat -listalla
    ' 0=asiasana, 1=asiasana_id
    Forms!uusi_lahde!asiasanat.AddItem (asiasana & ";" &
        asiasana_id)
    End If
    ' lähteen muokkaus -näytölle, jos avattu sieltä
ElseIf CInt(Me.OpenArgs) = 2 Then
    For i = 0 To Forms!lahteen_muokkaus!lahteen_asiasanat2.ListCount - 1

        If
            asiasana_id
Forms!lahteen_muokkaus!lahteen_asiasanat2.Column(2, i) Then
            ' asiasana oli jo, ei lisätä uudestaan
            onLiitetty = True
            Exit For
        End If

    Next i

    ' asiasanaa ei vielä ole, lisätään asiasanat listalle
    If onLiitetty = False Then
        ' sarake -järjestys: asiasana, lahde_id, asiasana_id,
        Forms!lahteen_muokkaus!lahteen_asiasanat2.AddItem (asiasana &
";" & Forms!lahteen_muokkaus!lahde_id & ";" & _
            asiasana_id)
        End If
    End If

Next rivi

' suljetaan näyttö
DoCmd.RunCommand acCmdCloseWindow

End Sub

'suodatetaan asiasana listaa sitä mukaa kun
'uusi_asiasana -kenttään kirjoitetaan

```

```
Private Sub uusi_asiasana_Change()  
  
    ' muuttujien esittely  
    Dim vSearchString As String  
  
    ' otetaan talteen muuttujaan uusi_asiasana - tekstikenttään syötetty arvo  
    vSearchString = uusi_asiasana.Text  
  
    ' asetetaan uusi_asiasana - tekstikenttään syötetty arvo piilotettuun  
    ' teksti-kenttään (etsi_asiasana), joka  
    ' toimii hakuehtona asiasanat -listalle  
    etsi_asiasana.Value = vSearchString  
  
    ' ajetaan kysely asiasanat-listalle, jotta tekstikenttään syötetyn tekstin  
    ' perusteella löydetyt asiasanat suodattuvat asiasanat -listalle  
    Me.asiasanat_lista.Requery  
  
    ' pidetään kursorin focus uusi_asiasana -tekstikentässä  
    Me.uusi_asiasana.SetFocus  
  
End Sub
```

UUSI LÄHDE-LOMAKKEEN SQL

Liite 2

Option Compare Database

```
' avataan asiasanojen hallinta -näyttö
Private Sub hae_asiasana_Click()

DoCmd.OpenForm "asiasana_hallinta", acNormal, , , , acDialog, "1"
End Sub

Private Sub tallenna_Click()
' muuttujien esittely
Dim insertSQL As String
On Error GoTo Err

' lähdetyyppi ja tekijä ovat pakollisia tietoja,
' tarkistetaan että on syötetty
If IsNull(Me.tyyppi_id) Or IsNull(Me.tekija_t) Then
    MsgBox ("Aseta ensin lähdetyyppi ja tekijä.")
    Exit Sub
End If

' asetetaan default varoitusilmoitukset pois näkyvistä
DoCmd.SetWarnings False
'tallennetaan lähteen tiedot
DoCmd.RunCommand acCmdSaveRecord

'tallennetaan asiasanat jos niitä oli lisätty
If Me.asiasanat.ListCount > 0 Then
    For i = 0 To Me.asiasanat.ListCount - 1

        ' sarakkeet uusi lähde asiasanat-listalla 0=asiasana, 1=asiasana_id
        insertSQL = "INSERT INTO lahteen_asiasanat (asiasanat_id, lahde_id)
" & _
            " VALUES (" & Me.asiasanat.Column(1, i) & ", " &
Me.lahde_id.Value & ");"
        DoCmd.RunSQL (insertSQL)

    Next i
End If

' alustetaan uusi rivi
DoCmd.RunCommand acCmdRecordsGoToNew

'tyhjennetään asiasanat lista
If Me.asiasanat.ListCount > 0 Then
    For i = 0 To Me.asiasanat.ListCount - 1
```

```

        Me.asiasanat.RemoveItem 0
    Next i
End If

' näytetään mahdolliset virheet
Err:
If Err.Number <> 0 Then
    MsgBox Err.Number & ":" & " " & Err.Description, , "Error"
End If
End Sub

' valittujen asiasanojen poistaminen
Private Sub poista_asiasana_Click()

' muuttujien esittely
Dim counter As Integer
Dim rivit As String
Dim poistettavatRivit() As String

' tarkistetaan onko rivejä valittu
If Me.asiasanat.ItemsSelected.Count > 0 Then
    ' alustetaan muuttujat
    counter = 0
    rivit = ""

    'käydään läpi valitut rivit ja kerätään talteen rivinumerot
    'rivit poistetaan myöhemmin, koska poisto täytyy tehdä viimeisestä
    'valitusta rivistä alkaen
    For Each rivinro In Me.asiasanat.ItemsSelected
        counter = counter + 1
        rivit = rivit & rivinro

        If counter < Me.asiasanat.ItemsSelected.Count Then
            rivit = rivit & ","
        End If
    Next rivinro

    ' viedään poistettavat rivit taulukkoon
    poistettavatRivit = Split(rivit, ",")

    ' käydään läpi poistettavat rivit taulukko viimeisestä
    ' sarakkeesta alkaen ja poistetaan rivi(t)
    For i = UBound(poistettavatRivit())
        To LBound(poistettavatRivit()) Step -1
            Me.asiasanat.RemoveItem Val(poistettavatRivit(i))
        Next i
    End For
End Sub

```

```
End If

End Sub

' kun lähdetyyppi tieto on valittu
Private Sub lahde_tyyppi_AfterUpdate()

' asetetaan kenttien näkyvyys lähdetyyppin mukaan
If lahde_tyyppi.Value = 1 Then 'kirjalähde
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = True
    painos.Enabled = True
    julkaisuvuosi.Enabled = True
    paikka.Enabled = True
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = False
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 2 Then 'lehtiartikkeli
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = False
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = True
    lehden_nro.Enabled = True
    pvm.Enabled = False
    sivu_nro.Enabled = True
    www_osoite.Enabled = False
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 3 Then 'opinnäyte
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = True
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = False
    sivu_nro.Enabled = False
```

```
www_osoite.Enabled = False
haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 4 Then 'esitelmä/luento
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = True
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = True
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = True
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 5 Then 'haastattelu
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = False
    julkaisija.Enabled = False
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = True
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
    haastattelu.Enabled = True

ElseIf lahde_tyyppi.Value = 6 Then 'sähköinen lähde
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = False
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = True
    sivu_nro.Enabled = False
    www_osoite.Enabled = True
    haastattelu.Enabled = False

End If
End Sub
```


SELAA/MUOKKAA LÄHTEITÄ-LOMAKKEEN SQL

Liite 3

Option Compare Database

Private Sub Form_Current()

```
' asetetaan kenttien näkyvyys lähdetyyppin mukaan
If lahde_tyyppi.Value = 1 Then 'kirjalähde
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = True
    painos.Enabled = True
    julkaisuvuosi.Enabled = True
    paikka.Enabled = True
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = False
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 2 Then 'lehtiartikkeli
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = False
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = True
    lehden_nro.Enabled = True
    pvm.Enabled = False
    sivu_nro.Enabled = True
    www_osoite.Enabled = False
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 3 Then 'opinnäyte
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = True
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = False
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
```

```
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 4 Then 'esitelmä/luento
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = True
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = True
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = True
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
    haastattelu.Enabled = False

ElseIf lahde_tyyppi.Value = 5 Then 'haastattelu
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = False
    julkaisija.Enabled = False
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = True
    sivu_nro.Enabled = False
    www_osoite.Enabled = False
    haastattelu.Enabled = True

ElseIf lahde_tyyppi.Value = 6 Then 'sähköinen lähde
    tekija_t.Enabled = True
    teoksen_nimi.Enabled = True
    julkaisija.Enabled = False
    painos.Enabled = False
    julkaisuvuosi.Enabled = True
    paikka.Enabled = False
    artikkelin_nimi.Enabled = False
    lehden_nro.Enabled = False
    pvm.Enabled = True
    sivu_nro.Enabled = False
    www_osoite.Enabled = True
    haastattelu.Enabled = False

End If

' päivitetään lomake, jotta muutokset näkyvät
```

Me.Refresh

```
' näytöllä on kaksi asiasanat listaa:
' 1. lahteen_asiasanat -lista alkuperäisiä arvoja varten,
' lista on piilotettu
' 2. lahteen_asiasanat2 -lista muokkausta varten, lista näkyvissä

' tyhjennetään asiasana luettelo ennen täyttämistä
If Me.lahteen_asiasanat2.ListCount > 0 Then
    For i = 0 To Me.lahteen_asiasanat2.ListCount - 1
        Me.lahteen_asiasanat2.RemoveItem 0
    Next i
End If

' asetetaan asiasanat luetteloon näkyviin
If Me.lahteen_asiasanat.ListCount > 0 Then

    For i = 0 To Me.lahteen_asiasanat.ListCount - 1
        Me.lahteen_asiasanat2.AddItem (Me.lahteen_asiasanat.Column(0, i) &
";"      &      Me.lahteen_asiasanat.Column(1,      i)      &      ";"      &
Me.lahteen_asiasanat.Column(2, i))
    Next i

End If

' disabloidaa/enabloidaan seuraava ja edellinen -painikkeet
' onko nykyinen rivi viimeinen tai ensimmäinen
Me.RecordsetClone.MoveLast
Me.seur.Enabled = Not (Me.CurrentRecord = Me.RecordsetClone.RecordCount)
Me.edellinen.Enabled = Not (Me.CurrentRecord = 1)

'asetetaan seuraava viite ja edellinen viite -painikkeet disabloiduksi
'jos ei viiterivejä 'nykyisellä lähteellä
If Me.lahteen_viitteet.Form.RecordsetClone.RecordCount = 0 Then
    Me.lahteen_viitteet.Form.edellinen_viite.Enabled = False
    Me.lahteen_viitteet.Form.seuraava_viite.Enabled = False
End If

End Sub

' avataan asiasanojen hallinta näyttö
Private Sub hae_asiasana_Click()

DoCmd.OpenForm "asiasana_hallinta", acNormal, , , , acDialog, "2"

End Sub
```

```

' asiasanojen poistaminen listalta
Private Sub poista_asiasana_Click()

' muuttujien esittely
Dim counter As Integer
Dim rivit As String
Dim poistettavatRivit() As String

' tarkistetaan onko rivejä valittu
If Me.lahteen_asiasanat2.ItemsSelected.Count > 0 Then
    ' alustetaan muuttujat
    counter = 0
    rivit = ""

    'käydään läpi valitut rivit ja kerätään talteen rivinumerot
    'rivit poistetaan myöhemmin, koska poisto täytyy tehdä viimeisestä
    'valitusta rivistä alkaen
    For Each rivinro In Me.lahteen_asiasanat2.ItemsSelected
        counter = counter + 1
        rivit = rivit & rivinro

        If counter < Me.lahteen_asiasanat2.ItemsSelected.Count Then
            rivit = rivit & ","
        End If
    Next rivinro

    ' viedään poistettavat rivit taulukkoon
    poistettavatRivit = Split(rivit, ",")

    ' käydään läpi poistettavat rivit taulukko viimeisestä sarakkeesta
    'alkaen ja poistetaan rivi(t)
    For i = UBound(poistettavatRivit()) To LBound(poistettavatRivit()) Step
-1
        Me.lahteen_asiasanat2.RemoveItem Val(poistettavatRivit(i))
    Next i

End If

End Sub

Private Sub poista_Click()
' muuttujien esittely
Dim vastaus As Integer
On Error GoTo Err

' varmistetaan poisto
vastaus = MsgBox("Halutanko varmasti poistaa valitun lähteen?", vbYesNo)

```

```

If vastaus = vbYes Then
    ' asetetaan oletus varoitustekstit pois näkyvistä.
    DoCmd.SetWarnings False
    ' poistetaan lähde
    DoCmd.RunCommand acCmdDeleteRecord

    ' päivitetään lomake, jotta kaikki näytöllä olevat tiedot ovat
    'ajan tasalla poiston jälkeen
    Me.Form.Refresh

End If

' näytetään mahdolliset virheet
Err:
If Err.Number <> 0 Then
    MsgBox Err.Number & ":" & " " & Err.Description, , "Error"
End If

End Sub

' tallenna muutokset
Private Sub tallenna_lahde_Click()

' muuttujien esittely
Dim insertSQL As String
Dim deleteSQL As String

On Error GoTo Err

' ei näytetä accessin oletusilmoituksia
DoCmd.SetWarnings False

' tallentaa perustiedot ja viitteet
DoCmd.RunCommand acCmdSaveRecord

' asiasanojen tallennus, ei tarkistella oliko asiasanoihin
' tehty muutoksia vaan :
' 1. poistetaan lähteeseen mahdollisesti liitetyt asiasanat
' 2. lisätään asiasanat uudelleen
deleteSQL = "DELETE FROM lahteen_asiasanat WHERE lahde_id = " &
Me.lahde_id.Value & ";"
DoCmd.RunSQL (deleteSQL)

' käydään läpi asiasanat, jos niitä on
If Me.lahteen_asiasanat2.ListCount > 0 Then
    ' asiasana, lähde_id, asiasana_id,
    For i = 0 To Me.lahteen_asiasanat2.ListCount - 1

```

```
' tallennetaan kantaan rivi kerrallaan
insertSQL = "INSERT INTO lahteen_asiasanat (asiasanat_id, lahde_id)
" & _
" VALUES (" & Me.lahteen_asiasanat2.Column(2, i) & "," &
Me.lahde_id.Value & ");"
DoCmd.RunSQL (insertSQL)

Next i
End If

' päivitetään piilotettu asiasanat lista, jotta tiedetään tallennetut
Me.lahteen_asiasanat.Requery

' näytetään mahdolliset virheet
Err:
If Err.Number <> 0 Then
MsgBox Err.Number & ":" & " " & Err.Description, , "Error"
End If

End Sub
```

HAKU-LOMAKKEEN SQL

Liite 4

```

'public muuttuja alkuperäiselle sql -lauseelle
'näkyy kaikille funktioille
Dim OriginalSql As String

Option Compare Database
Private Sub Form_Load()
    ' asetetaan alkuperäinen kysely talteen
    OriginalSql = "SELECT DISTINCT lahde.lahde_id, lahde.tekija_t, lah-
de.teoksen_nimi, lahdetyyppi.nimi " & _
        " FROM lahdetyyppi INNER JOIN lahde ON lahdetyyppi.tyyppi_id
= lahde.tyyppi_id " & _
        " WHERE      (((lahde.tekija_t)      Like      '*'      &
[Forms]![haku]![tekija_t] & '*') " & _
        " AND      (nz(lahde.teoksen_nimi, ''))      Like      '*'      &
[Forms]![haku]![teoksen_nimi] & '*' ) " & _
        " AND      ((lahdetyyppi.nimi)      Like      '*'      &
[Forms]![haku]![lahdetyyppi] & '*'))"

End Sub

' Etsi -painiketta painettaessa
Private Sub etsi_Click()
' muuttujien esittely
Dim SelectSql As String

' tarkistetaan onko asiasana hakuehto annettu, jos on lisätään
'alkuperäiseen kyselyyn uusi hakuehto
If Not IsNull(Me.asiasana.Value) Or Me.asiasana.Value <> "" Then

    SelectSql = OriginalSql + " AND lahde.lahde_id in (SELECT
lahteen_asiasanat.lahde_id " & _
        " FROM asiasanat INNER JOIN lahteen_asiasanat ON asiasa-
nat.asiasana_id=lahteen_asiasanat.asiasanat_id " & _
        " WHERE asiasanat.asiasana Like '*' & [Forms]![haku]![asiasana] & '*' )
"

' asiasanaa ei ollut, käytetään alkuperäistä kyselyä
' asiasanat taulua ei voi kytkeä kyselyyn, muutoin lähteet
' joilla ei ole asiasanoja 'jäisivät pois hakutuloksista
Else

    SelectSql = OriginalSql

End If

```

```
' asetetaan sql -lause ja suoritetaan kysely
Me.lahde_KyselyLomake.Form.RecordSource = SelectSql
Me.lahde_KyselyLomake.Requery

End Sub

' Tyhjennä hakuehdot -painiketta painaessa
Private Sub tyhjenna_Click()

' tyhjennetään hakuehto -kentät
Me.tekija_t = Null
Me.teoksen_nimi = Null
Me.lahdetyyppi = Null
Me.asiasana = Null

' asetetaan alkuperäinen sql -lause ja suoritetaan kysely
Me.lahde_KyselyLomake.Form.RecordSource = OriginalSql
Me.lahde_KyselyLomake.Requery

End Sub
```