

Suoriin SQL-kyselyihin perustuvien raporttien kehittäminen  
Suomen ympäristökeskuksen (SYKE) tarpeisiin ympäristöhal-  
linnon VAHTI-järjestelmässä

Tiina Leppänen



Tietojenkäsittelyn koulutusohjelma

<p>Tekijä tai tekijät Tiina Leppänen</p>	<p>Ryhmä tai aloitusvuosi 2007</p>
<p>Opinnäytetyön nimi Suoriin SQL-kyselyihin perustuvien raporttien kehittäminen Suomen ympäristökeskuksen (SYKE) tarpeisiin ympäristöhallinnon VAHTI-järjestelmässä</p>	<p>Sivu- ja liitesivumäärä 25 + 19</p>
<p>Ohjaaja tai ohjaajat Ari Kangas, Lauri Etelämäki, Sauli Isonikkilä</p>	
<p>Tämä opinnäytetyö tehtiin Suomen ympäristökeskuksen (SYKE) tarpeesta tehdä suoria SQL-kyselyjä VAHTI –tietokannasta. Tietokanta sisältää raportteja mm. vesi-, ilma- ja jätetiedoista, mutta osa tarvittavista tiedoista ei näy valmiissa raporteissa. Tästä syystä SYKE pyysi minua tekemään haut suoraan tietokannasta.</p> <p>Tämän projektin tarkoituksena on myös auttaa jatkossa, kun SYKE kehittää TIVA2 –hanketta eteenpäin. TIVA2 –hankkeen tarkoituksena on luoda tietovarasto VAHTI –tietokannan päälle siten, että tiedon käytöstä tulisi helpompaa ja avoimempaa kaikille. Hankkeen tarkoituksena on valmistua loppuvuonna 2011.</p> <p>Opinnäytetyössä esitetään VAHTI –tietokannan ongelmakohtia ja kehittämisideoita TIVA2 –hanketta varten. Suurimpina ongelmina käsittelyssä ovat tietokannan huonot taulujen sisällön nimeämiset ja puuttuvien tietojen etsimisen hankaluus.</p> <p>Projektin toteutus eteni erittäin sujuvasti hieman hitaan alun jälkeen ja aikataulusta oli helppo pitää kiinni. Tietokantoihin tutustuminen vei eniten aikaa, suurimmaksi osaksi siitä syystä, että VAHTI –tietokannan lisäksi Asiakas- ja Asiakasrekisteri –tietokannat liittyivät myös projektiin. Varsinainen hakujen tekeminen oli helppoa. Kyselyihin sain vastaukset, jos se vain oli tietokannan takia mahdollista.</p> <p>Opinnäytetyön tulokset olivat ennalta arvattavia jo etukäteistietojen perusteella, vain tietokannan joustamattomuus joissain asioissa tuli pienenä yllätyksenä. Tulokset etenivät vaihe vaiheelta monimutkaisempia hakuja kohti, ensimmäisissä hauissa haettiin pelkästään VAHTI –tietokannan tauluista tietoja ja viimeisissä hauissa haettiin 12 eri tauluista tietoja, joista 5 taulua sijaitsivat toisissa tietokannoissa. Käytössä olleilla työkaluilla saadut tulokset ovat kohtuullisen luotettavia.</p>	
<p>Asiasanat tietokannat, SQL, ympäristöhallinto</p>	

Tietojenkäsittelyn koulutusohjelma

<p>Author or authors Tiina Leppänen</p>	<p>Group or year of entry 2007</p>
<p>The title of thesis Developing reports based on direct SQL queries</p>	<p>Number of pages and appendices 25 + 19</p>
<p>Supervisor or supervisors Ari Kangas, Lauri Etelämäki, Sauli Isonikkilä</p>	
<p>The purpose of this thesis was to develop reports based on direct SQL queries to the VAHTI database of Finland's Environmental Administration. This thesis project was assigned by the Finnish Environment Institute, SYKE.</p> <p>The VAHTI database contains reports concerning e.g. water, air and waste data, but some pieces of the needed information are not included in these reports. Therefore, it was important for SYKE to have the system developed further.</p> <p>Furthermore, this thesis project aimed at helping SYKE to further develop the use of environmental data in the future in its TIVA2-project. The purpose of the TIVA2-project is to create a data warehouse on VAHTI database to make the data easier and more open for everybody to use. This project is scheduled to be completed at the end of the year 2011.</p> <p>This thesis investigated the problem areas of the VAHTI database and introduced some development ideas for the TIVA2 project. The biggest problems in the usage were the poor naming of the content of tables and the difficulty of searching missing data.</p> <p>As a result of this thesis project, the reports were developed according to the needs of the Finnish Environment Institute. The results began in a simple way by getting information out of the tables in VAHTI –database. Later the results were much more complex, the last query contained information from 12 different tables of which 5 were located in different databases.</p>	
<p>Key words SQL, databases, environmental administration</p>	

# Sisällys

1	Johdanto .....	1
1.1	Projektin tausta .....	1
1.2	Projektin tavoitteet .....	1
1.3	Tehtävänasettelu .....	2
1.4	Käsitteet .....	2
2	Teoreettinen viitekehys.....	4
2.1	Tietokannat .....	4
2.1.1	Tietojärjestelmät.....	5
2.1.2	Tietovarastotekniikat ja niiden kehittyminen.....	5
2.1.3	Tietovarastotekniikoiden edut ja haitat.....	6
2.1.4	Tietomallit.....	7
2.1.5	Relaatiotietokannat .....	8
2.1.6	Tietokannanhallintajärjestelmä (DBMS).....	9
2.1.7	Tietokannanhallintajärjestelmän käyttäjäryhmät.....	10
2.2	SQL .....	11
2.2.1	Microsoft SQL Server .....	11
2.2.2	SQL Management Studio .....	12
2.3	VAHTI-tietokanta.....	12
2.4	TIVA2-hanke .....	16
3	Empiirinen osa.....	18
3.1	Esittely .....	18
3.2	Ongelmat ja kehittäminen .....	18
3.3	Toteutus.....	19
3.4	Tulokset .....	20
4	Pohdinta .....	22
4.1	Tulosten tarkastelu .....	22
4.2	Tulosten luotettavuus .....	23
4.3	Yhteenveto ja johtopäätökset .....	23
4.4	Jatkokehittämissuhteet .....	24
4.5	Oppiminen .....	24
4.5.1	Opinnäytetyöprosessin arviointi.....	24

4.5.2 Oman osuuden arviointi .....	25
Lähteet .....	26
Liitteet .....	28
Kuvaluettelo	
Kuva 1 Tietokannan käyttäjäryhmät .....	10
Kuva 2 Asiakastiedot .....	12
Kuva 3 Asiakkaan tiedot .....	13
Liiteluettelo	
Liite 1 Hakuskripti 1 .....	28
Liite 2 Hakuskripti 2 .....	29
Liite 3 Hakuskripti 3 .....	30
Liite 4 SQL-hakujen selitys .....	32
Liite 5 VAHTI-ohje .....	34

# 1 Johdanto

Tässä opinnäytetyössä esitellään Suomen ympäristökeskuksen (SYKE) käyttämä ympäristönsuojelun tietojärjestelmän (VAHTI) tietokanta ja sen raporttien kehittäminen SYKEN tietotarpeisiin sopivaksi. Työssä esitellään lisäksi ongelmat ja parannusehdotukset, jotka tämän opinnäytetyön tekemisessä kohdattiin ja keksittiin.

## 1.1 Projektin tausta

Projekti käynnistyi Suomen ympäristökeskuksen (SYKE) tarpeesta saada mahdollisimman kattavia ja heidän omiin toimiinsa tarpeellisia tietoja ulos VAHTI-tietokannasta. Etenkin raporteista puuttuvien tietojen saaminen ulos tietokannasta oli SYKElle erittäin tärkeää myös jatkon kannalta. Oma tarpeeni oli löytää opinnäytetyöpaikka, jossa pääsisi tekemään töitä tietokantojen parissa sekä oppimaan IT:tä varsinaisessa työelämässä.

SYKE käyttää VAHTI-järjestelmää saadakseen sieltä raportteja, joita myöhemmin käytetään SYKEN tarvitsemalla tavalla eri projekteissa ja töissä. VAHTI on ympäristöhallinnon ylläpitämä ympäristönsuojelun tietojärjestelmä, joka pitää sisällään mm. jätevedenpuhdistamotietoja. Muita VAHTIsta saatavia tuloksia ovat esimerkiksi vesi-, ilma-, polttoaine- ja jätetiedot.

## 1.2 Projektin tavoitteet

Nykyiset VAHTIsta saatavat raportit eivät aina täytä kaikkia SYKEN projekteittain vaihtelevia tietotarpeita ja tästä syystä minua pyydettiin tekemään jätevedenpuhdistamoja koskevia hakuja suoraan VAHTI-tietokannasta. Suoraan tietokannasta hakemalla on mahdollista saada näkyviin arvoja, jotka eivät tulostuisi mihinkään valmiiseen raporttiin. Näiden tekemieni hakujen on tarkoitus auttaa myöhemmin, kun VAHTI – tietokantaan perustuen tehdään uusi tietovarasto ympäristövarantojen käytön kehittämishankkeen toisen vaiheen (TIVA2 –hanke) yhteydessä. Uudistusprojektin on tarkoi-

tus valmistua tietokannan osalta 30.6.2011 mennessä. Tavoitteena on, että tekemieni hakujen avulla SYKEssä on helpompi suunnitella, mihin ongelmakohtiin uudistusprojektissa on jätevedenpuhdistamojen osalta tarkoitus keskittyä. Työstä on mahdollisesti apua myös TIVA2-hankkeen muilla osa-alueilla.

### 1.3 Tehtävänasettelu

Projektissani keskityn vain jätevedenpuhdistamotietoihin ja niihin liittyviin arvoihin. VAHTIn tietokanta on niin suuri, että aihetta on pakko rajata vain yhteen kokonaisuuteen, jotta projektista ei tulisi liian suuri.

### 1.4 Käsitteet

projekti	Ennalta tietyllä aikavälillä suunniteltu hanke, jonka tavoitteena on saavuttaa tietty päämäärä.
(relaatio)tietokanta	Tietovarasto, jossa tiedot on esitetty toisiinsa yhteydessä olevana kokoelmana. Tietokantojen koko voi vaihdella suuresti muutamman rivin mittaisista miljoonien rivien kantoihin. Tietokantaan tallennetut tiedot voivat olla mm. tekstiä, kuvia, ääniä ja videoita.
SQL	=structured query language. Hakujen teon mahdollistava kyselykieli, jonka avulla on mahdollista tehdä hakuja, kyselyitä ja muokkauksia lähes mihin tahansa relaatiotietokantaan.
SYKE	=Suomen ympäristökeskus. SYKE kokoaa, muokkaa ja tuottaa ympäristötietoa sekä tekee ympäristöalan tutkimus- ja kehittämistyötä.
Suomen ympäristöhallinto	Muodostuu ympäristöministeriöstä, Suomen ympäristökeskuksesta sekä asumisen rahoitus- ja kehittämiskeskuksesta.
VAHTI	Ympäristönsuojelun tietojärjestelmä, jonka tehtävänä on pitää sisällään tiedot valvonta- ja kuormitustiedot koskien mm. vesi-, ilma- ja jätearvoja.

VAHTI-raporttiosa	Käytössä toistaiseksi vain intranetissä ja rajoitettuna asiakkaiden ekstranetissä testikäytössä. Tulostaa pyydettyjä tietoja valmiiden raporttipohjien mukaan.
intranet	Eristetty lähiverkko vain tiettyjen ihmisten tai ryhmien käyttöön. Toimii hyvin yrityksen sisäiseen tiedotukseen.
ekstranet	Intranet, jonka oikeuksia on laajennettu myös (tiettyjen) asiakkaiden tai ryhmien käyttöön. Silti verkkopalvelu on suljettu eli ilman oikeuksia sinne eivät ulkopuoliset pääse. Hyvä väylä esimerkiksi asiakkaiden tiedottamiseen.
SQL Server Management Studio 2005	Tietokantaohjelma, jonka avulla voi luoda, muokata, hakea ja poistaa tauluja ja/tai niiden sisällä olevia tietoja. Ohjelma toimii SQL Serverin ylläpitotyökaluna.



## 2 Teoreettinen viitekehys

Tämän opinnäytetyön viitekehystenä esitellään tietokannat yleisesti, tietojärjestelmät, tietovarastotekniikoiden kehittyminen, tietovarastotekniikoiden edut ja haitat, tietomallit, relaatiotietokannat, tietokannanhallintajärjestelmän sekä sen käyttäjäryhmät. Lisäksi esitellään SQL:n ohessa lyhyesti työssä käytössä olevat Microsoft SQL Server ja SQL Management Studio. Näiden lisäksi tutustutaan VAHTI-tietokantaan ja TIVA2 –hankeeseen.

### 2.1 Tietokannat

Tietokannat, toiselta nimeltään tietovarastot, ovat kokoelma tietoja yhdistettynä toisiinsa. Tietokannat toimivat nykyisin suurimmaksi osaksi sähköisessä muodossa, mutta joissain tapauksissa tietokannat saattavat olla vielä paperiversioita. Etenkin yksinkertaiset ja pienet tietokannat ovat helposti hallittavia kokonaisuuksia. Tietokantojen koko voi vaihdella suuresti muutaman rivin pätkästä miljoonia rivejä sisältäviin tauluihin. Yleisimmin tietokantoja käytetään tekstin varastointiin, mutta sen lisäksi niihin voi tallentaa esimerkiksi kuvia, ääntä ja videoita. (Tietokanta. 2010.)

Suurimpia syitä yrityksille hankkia tietojärjestelmä hallinnoimaan heidän tietojensa ovat tuotteiden ja palveluiden laadun parantaminen, työntekijöiden työn kannattavuuden maksimoiminen sekä markkinoiden mahdollisuuksien tarkkailu ja hyväksikäyttö. Tietovarastoja käyttämällä tiedon päivitettävyyden ja pysyvyyden käyttötavat nousevat esille aivan uudenaikaisina ominaisuuksina. (Elmasri, R. & Navathe, S. 2006.)

Tietovarastot jaetaan säilytystapansa perusteella kahteen eri luokkaan: tietokoneella hoidettavat ja käytettävät atk-varastot sekä ihmisten käsitetovarastot. Tässä opinnäytetyössä keskitytään atk-varastoihin. Tietovarastoissa säilytettäviä erityyppisiä tietoja ovat yleisimmin määrämuotoiset tietoalkiot, asiakirjoina tallentuvat tekstit, kuvat ja äänitteet. Tiedon rakenne jaetaan kolmeen eri luokkaan: rakenteinen data, osittain rakenteinen data sekä rakenteeton data. Rakenteinen data kattaa esimerkiksi tilien saldot, laskun tiedot sekä eri tapahtumien tiedot. Osittain rakenteinen data käsittää määrämuotoiset

dokumentit, esimerkiksi manuaalit, ohjeet ja sopimukset. Rakenteettomalla datalla ei ole toistuvaa rakennetta nimensä mukaisesti eli se voi kattaa kaikenlaista tietoa puhelinnumeroista kuviin. (Elmasri, R. & Navathe, S. 2006.)

### 2.1.1 Tietojärjestelmät

Tietojärjestelmäksi eli tietosysteemiksi kutsutaan tiettyä toiminnan osaa palvelevaa tietojenkäsittelykokonaisuutta. Tietojärjestelmässä voi olla mukana myös atk-järjestelmiä, mutta ne eivät ole pakollisia. Tietojärjestelmään liittyy aina ihmisen suorittama osuus eli automatisoimaton osuus. Erilaisia tietojärjestelmiä on nimetty neljä: tapahtumien käsittelyjärjestelmät, raportoinnin- ja analysoinnin tietojärjestelmät, toimistoautomaatiojärjestelmät sekä tietotyön tukijärjestelmät. Tapahtumien käsittelyjärjestelmän tunnusmerkkejä ovat tiedon jatkuva muuttuminen, ajantakaisuuden tärkeys sekä jatkuva päivityskäyttö. Se palvelee hyvin operatiivista toimintaa. Raportoinnin- ja analyysien tietojärjestelmät ovat joko johdon informaatiojärjestelmiä tai päätöksenteon tukijärjestelmiä. Niissä tiedonhaun nopeus on keskeisessä roolissa ja ne saattavat sisältää suuria datamääriä. Toimistoautomaatiojärjestelmien tavoitteena on parantaa työn tuotettavuutta mm. tekstinkäsittelyn ja taulukkolaskennan avulla. Tietotyön tukijärjestelmät toimivat uuden tiedon luomisessa ja integroinnissa apuna. Nämä kaikki eri tietojärjestelmät voivat toimia omatoimisin tai ne voi integroida toisiinsa. (Elmasri, R. & Navathe, S. 2006.)

### 2.1.2 Tietovarastotekniikat ja niiden kehittyminen

Tiedostopohjaiset ratkaisut elivät kulta-aikaansa 1950- ja 1960-luvuilla, jolloin niiden kukin sovellus määritteli oman datansa. Ne olivat käytettävissä vain yhden käyttäjän sekä sovelluksen toimesta kerrallaan. Samoihin aikoihin peräkkäistiedostot kehittyivät reikäkorteista ja –nauhoista magneettinauhoihin, jotka ovat yhä tähän päivään asti käytössä lähinnä varmuuskopioinnissa ja päivityslokeissa. Tiedot kirjoitetaan peräkkäisjärjestyksessä, joten tiedostoista voi tulla erittäin suuria. Useampi sovellus voi käyttää tätä tiedostotapaa, mutta perinteisesti sitä käyttää vain yksi. (Garcia-Molina, H. & Ullman, J. & Widom, J. 2008.)

Suorasaantitiedostot kehittyivät 1960-luvulla, jolloin niille syntyivät rumpu ja levymuistit. Tiedoissa täytyy olla yksilöivä ominaisuus, jota käytetään hakuperusteena eli avaimena. Suorasaantitiedostot toimivat joko hakemistorakenteisena peräkkäistiedostona tai hajarakenteena. Tiedoston pystyy luomaan, poistamaan, kopioimaan, siirtämään, nimeämään uudelleen, avaamaan, sulkemaan sekä suojaamaan yksinkertaisilla tavoilla. (Garcia-Molina, H. & Ullman, J. & Widom, J. 2008.)

Vuonna 1960 kehitelty tietokantojen edeltäjä oli suurimmaksi osaksi Charles Bachmanin kehittely tulos. Kymmentä vuotta myöhemmin Edgar F. Codd kehitti relaatiotietokantamallin sekä käsittelyteorian. Hän oli töissä IBM:n laboratoriossa ja täten IBM:llä on suuri vaikutus tietokantojen kehitykseen. IBM:ltä julkaistiinkin vuonna 1982 ensimmäinen kaupallinen tietokantatuote nimeltään DB2. (Tietokanta. 2010.)

Tietokantapohjaiset ratkaisut kehittyivät 1970 luvulla. Tarkoituksena oli, että varsinaisesta tiedonhausta vastaa tiedonhallintajärjestelmä, jonka ansiosta usea käyttäjä sekä sovellus voi käyttää tietovarastoa samanaikaisesti. Lisäksi erona aiempaan on, että sovellusohjelma käsittelee tietoa tiedonhallintajärjestelmän kautta eikä suoraan kuten ennen. Tämä aikaansaa tietovaraston ja sovelluksen riippumattomuuden. 1970-luvulla tietokantapohjaisissa ratkaisuissa käytettiin verkkotietokantoja sekä hierarkkisia tietokantoja, jotka kehittyivät 1980-luvulla relaatiotietokannoiksi. Oliotietokannat ovat uusin muoto tietokannoista. (Garcia-Molina, H. & Ullman, J. & Widom, J. 2008.)

### 2.1.3 Tietovarastotekniikoiden edut ja haitat

Suurimmat ongelmat tiedostopohjaisissa ratkaisuissa olivat tiedon riippuvuus, tiedon monistus ja tiedon suojauksen hankaluus. Tiedon riippuvuus esiintyi siten, että tietovaraston rakenne oli kuvattu vain ohjelmassa, joka sitä käsitteli. Jos tietovaraston rakennetta muutti, se aiheutti muutoksia myös sitä käsittelevään ohjelmaan. Tiedon monistus esiintyi ongelmana, kun moneen kertaan tallennettu tieto saattoi olla ristiriitaista keskenään sekä se vaati suuren tallennustilan. Tiedon suojausta hankaloitti se, että tiedon

eheydestä huolehtivat mekanismit, virhetilanteista toipuminen sekä käyttöoikeuksien valvonta tuli ohjelmoida itse. (Garcia-Molina, H. & Ullman, J. & Widom, J. 2008.)

Tietokannanhallintajärjestelmän palveluita ovat tiedon varastointi, haku, päivitys, eheyden valvonta, transaktiokäsittely, samanaikaisuuden hallinta, toipuminen, käyttöoikeuksien hallinta, tietohakemisto, tietoliikennetuki sekä apuohjelmat. Etuina tietokantapohjaisissa ratkaisuissa ovat tiedon ohjelmariippumattomuus, eheys, suojaus ja ohjelmoinnin tehokkuus. Tiedon ohjelmariippumattomuus aikaansaa tiedon paremman saavutettavuuden, eri sovellusten tietojen yhdistämisen sekä helpomman ylläpidettävyyden. Tiedon eheys koostuu toiston eliminoinnista, eheyssääntöjen määrittelystä ja toipumismekanismien luomisesta virhetilanteiden varalta. Tämän ansiosta tiedon yhteiskäyttöisyys helpottuu. Tiedon suojaus mahdollistaa käyttöoikeuksien hallinnan erittäin tarkalla tasolla. Ohjelmoinnin tehokkuus koostuu standardista käsittelykielestä ja mekanismeista, jotka vastaavat eheyttä, tietoturvaa ja toipumista, ovat valmiina. Haittoina tietokantapohjaisilla ratkaisuilla ovat suurimpana hinta, monimuotoisuus sekä suorituskyky. (Garcia-Molina, H. & Ullman, J. & Widom, J. 2008.)

#### 2.1.4 Tietomallit

Tietomalleja on useita erilaisia, mm. hierarkkinen malli, verkkomalli ja relaatiomalli. Tietomalli kuvaa tavan, jolla tietokannan tiedot organisoidaan. Se koostuu käsitteistä, säännöistä ja operaatioista, joiden tarkoituksena on luoda mallista mahdollisimman yksinkertainen ja selkeä. (Elmasri, R. & Navathe, S. 2006.)

Tietokantoja suunniteltaessa kannan rakenteen tulee olla eheä, yksinkertainen, ilmaisuvoimainen ja rakenteessa ei saa esiintyä toistoa. Tavoitteena on tiedon luotettavuuden varmistaminen, käyttäjien tietotarpeisiin vastaaminen, hyvä suorituskyky sekä jatkokehitettävyyden. (Elmasri, R. & Navathe, S. 2006.)

## 2.1.5 Relaatietietokannat

Relaatietietokannat ovat SQL-kielen avulla hallittavia tietokantoja. Tietokantatauluja luodaan, haetaan ja poistetaan käyttämällä SQL-kielen määrittelylauseita, kuten CREATE-, SELECT- ja DROP-lauseita. Relaatietietokantajärjestelmiä ovat mm. Access, DB2, MySQL, Oracle ja SQL Server. (Tietokanta. 2010.)

Relaatietietokannat ovat yleisimmin käytettyjä tietokantoja. Niissä monen taulun tiedot sidotaan yhteen käyttämällä kenttiä, jotka ovat tauluille yhteisiä siten, että tiedot voidaan hakea useasta taulusta samanaikaisesti. (Morley, D. & Parker, C. 2007.)

Relaation tyypillisimpiä ominaisuuksia ovat jokaisen relaation yksikäsitteinen nimi, relaatio ei saa sisältää kahta täysin identtistä riviä, jokaisella rivillä tulee olla sama määrä attribuutteja, kukin attribuutti sisältää arvoja vain yhdestä arvoalueesta sekä rivien tai attribuuttien ei tarvitse olla tietyssä järjestyksessä. (Connolly T. & Begg, C: 2004.)

Attribuutin ominaisuuksia ovat yksikäsitteinen nimi relaation sisällä, relaatiotietomallissa arvojoukot voidaan määritellä attribuuttien merkityksen mukaan sekä attribuutit esitellään perustietotyyppien ja niiden maksimipituuksien mukaan. Monikko on attribuutti-arvo –parien joukko eli se koostuu arvoista ja niiden attribuuteista. Tyhjä arvo eli null-arvo on puuttuva arvo, jota joko ei tiedetä tai sitä ei ole. Näitä arvoja on mahdollista vertailla tai laskea. (Connolly T. & Begg, C: 2004.)

Avaimet toimivat relaatioissa pienimpinä mahdollisina joukkoina attribuutteja, jotka yksilöivät jokaisen relaation rivin. Avaimen arvo ei voi olla tyhjä, mutta avaimia voi olla relaatioissa useampi kuin yksi. Pääavain on relaation avain ja sellaiseksi kannattaa valita attribuutti, jonka arvo ei usein muutu. Viiteavain on attribuutti relaatioissa, joka viittaa jonkun muun relaation pääavaimeen, se siis sisältää saman arvoalueen kuin toisen relaation pääavain. Viiteavaimia käytetään relaation välisten yhteyksien luomiseen. Viiteavain sijaitsee aina yhteyden moni-päässä, kun pääavain on (0..)1-päässä. Viiteavaimen arvo voi olla tyhjä vain siinä tapauksessa, että yhteys relaatioiden välillä oleva yhteys ei ole pakollinen. (Connolly T. & Begg, C: 2004.)

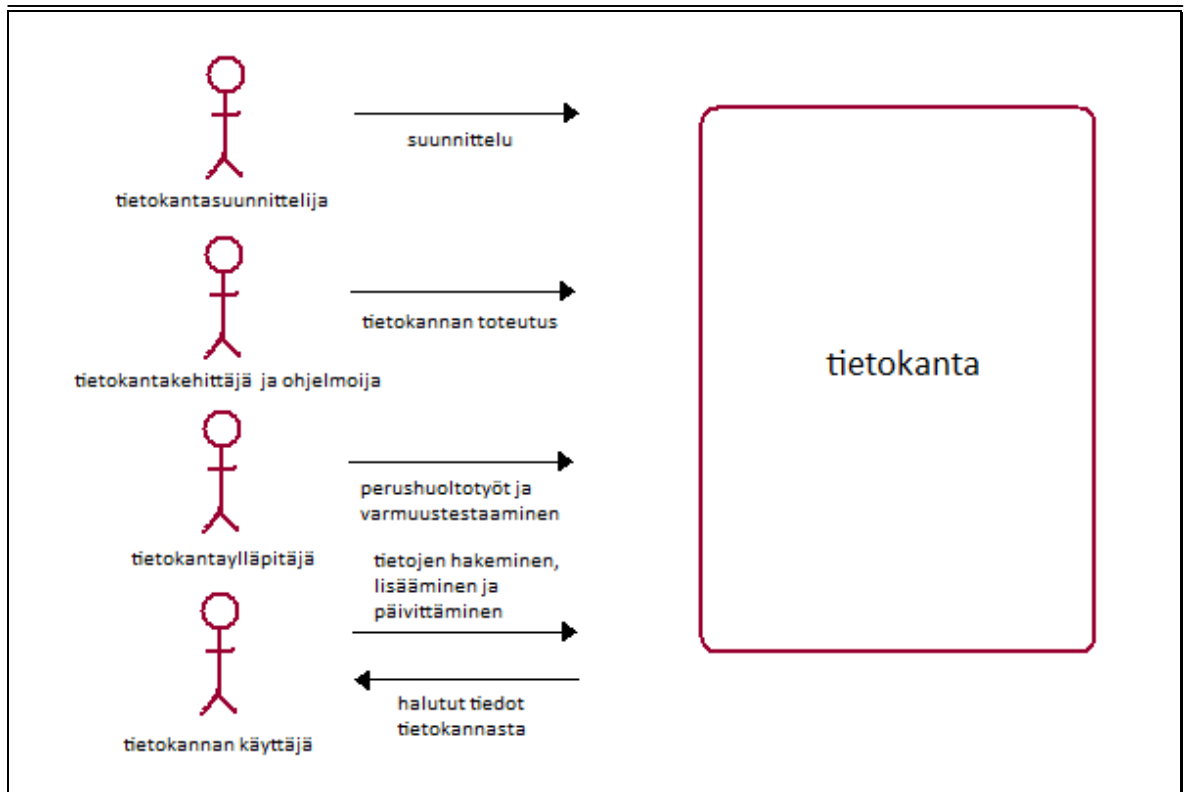
Tallennetut relaatiot eli taulut ovat tiedon tallennusta varten kehitetyt varastot. Taulun luonnissa sarakkeet, avaimet sekä taulun nimi määritellään. Kyselyt ovat keinoja, joilla tauluista haetaan tietoja. Usein hauille määritellään tietyt hakuehdot, joiden mukaan tietokanta hakee kyseiset arvot. Rivit ja sarakkeet voidaan järjestää itse haluamalla tavalla ja niistä voidaan hakea halutessa vain osa tiedoista. (Garcia-Molina, H. & Ullman, J. & Widom, J. 2008.)

#### 2.1.6 Tietokannanhallintajärjestelmä (DBMS)

Tietokannanhallintajärjestelmä (DBMS) luo, ylläpitää ja tarjoaa pääsyn tietokantoihin, lisäksi se kontrolloi tiedon järjestämistä sekä suojelee tiedon turvallisuutta ja eheyttä. Tiedon tarkoituksena on pysyä paikkansa pitävänä ja suojeltuna sekä tahallaan että vahingossa aiheutetulta tuholta. DBMS:n perusosa on ns. tietokantakone, joka on se varsinainen osa ohjelmaa, joka varastoi ja palauttaa halutut tiedot. Vaihtoehtona tietokantakoneelle monet DBMS:t ovat suunniteltuja siten, että tietokanta on jo valmiiksi yhdistetty tarvittaviin välineisiin, jotka tekevät suuren osa välttämättömistä tehtävistä. Tällaisia ovat mm. käyttäjärajapintojen teko, lomakkeiden ja raporttien luonti sekä kysely- ja ohjelmointikielien liittäminen tietokantaan monimutkaisia hakuja varten. (Morley, D. & Parker, C. 2007.)

Tietokantoja käytetään sekä erien käsittelyssä että reaaliaikakäsittelyssä. Nimensä mukaisesti eräkäsittelyssä tapahtumat kerätään pitkällä aikavälillä eriksi, joissa ne käsitellään esimerkiksi päivän päätteeksi. Reaaliaikakäsittelyssä tapahtumat käsitellään sitä mukaan kun ne tapahtuvat. Täten käyttäjän ja tietokoneen välillä on jatkuvaa vuorovaikutusta keskenään. Eräkäsittelyssä vuorovaikutusta ei synny niin kauan kuin prosessi on meillä. Reaaliaikainen käsittely on huomattavasti suositumpaa nykyisin kuin eräkäsittely. (Morley, D. & Parker, C. 2007.)

## 2.1.7 Tietokannanhallintajärjestelmän käyttäjäryhmät



Kuva 1 Tietokannan käyttäjäryhmät

DBMS:ään liittyviä käyttäjäryhmiä ovat tietokannan käyttäjät, tietokantasuunnittelijat, tietokantakehittäjät ja ohjelmoijat sekä tietokantaylläpitäjät. Luontivaiheessa tietokantasuunnittelijat suunnittelevat tietokannan vastaamaan mahdollisimman hyvin asiakkaan tarpeita ja suunnitelman pohjalta tietokantakehittäjät pyrkivät toteuttamaan sen mahdollisimman suurelta osin. Ohjelmoijat toimivat yhteistyössä tietokantakehittäjien kanssa lisäämällä tarpeelliset ohjelmakoodit, joiden avulla mahdolliset muut ohjelmat toimivat yhteistyössä tietokannan kanssa. Tietokantaylläpitäjät tekevät tietokantaan säännöllisesti varmuustestejä tietokannan eheyden ja luotettavuuden varmistamiseksi sekä tarvittaessa perushuoltotyöitä siihen liittyen. Käyttäjät ovat ne, joita varten tietokanta on luotu. He hakevat, lisäävät ja päivittävät tietoa tietokannan sisältä. (Morley, D. & Parker, C. 2007.)

## 2.2 SQL

SQL on tietojen käsittelyä, jossa käytetään yleisimmin insert-, update-, delete- ja select –käskyjä tietokannan taulujen, attribuuttien ja rivien komentoihin. Insert –käskyä käytetään rivien lisäämiseen tauluun, updatea tauluissa olevien tietojen päivittämiseen, deleteä rivien poistoon tauluista ja selectiä yhteen tai useampaan tauluun kohdistuvaan kyselyyn. Lisäksi SQL:llä on useita eri funktioita, esimerkiksi merkkijono- ja päivämäärä-funktiot, joiden tehtäviin kuuluvat mm. suurten kirjaimien muuttamisen pieneksi, merkkijonon pituuden palauttaminen sekä ajan palauttamisen päivämäärän ja ajan sekunnin perusteella. Tyypillinen haku on select –käsky, jossa haetaan jonkin attribuutin arvo jostain taulusta. From –käskyn avulla määritellään taulu, josta tieto haetaan. Where –käskyllä valitaan halutut rivit vertaamalla lauseketta toiseen. Esimerkiksi voidaan hakea Palkka –taulusta vain niiden työntekijöiden nimet ja palkat, joiden palkat ylittävät 3500€/kk. Lopuksi tulokset voidaan järjestellä order by –komennon avulla, jossa esitetään esimerkiksi asiakkaiden nimien mukaan järjestetyt palkat. (Elmasri, R. & Navathe, S. 2006.)

### 2.2.1 Microsoft SQL Server

Ensimmäinen versio Microsoftin SQL Serveristä kehitettiin jo vuonna 1989 yhteistyössä Sybasen ja Ashton-Taten kanssa. Ensimmäisen version nimi oli SQL Server 1.0 for OS/2, jota seurasivat Microsoft SQL Server 4.21, Microsoft SQL Server v6.0, SQL Server 7.0, SQL Server 2000, SQL Server 2005 ja SQL Server 2008. Microsoft julkaisi SQL Server 2005:n marraskuussa 2005 Suomessa samanaikaisesti Visual Studio 2005:n ja BizTalk Server 2006:n kanssa (Microsoft 2005). SQL Server 2005 on tietokanta tietojen käsittelyä ja analysointia varten, se pystyy aiemmasta poiketen käsittelemään myös xml-tietoa jo aiemmin käsitellyn relaatiotiedon lisäksi. Lisätoimintoina siinä myös on sisäinen raportointi- ja tietoanalyysityökalujen käyttömahdollisuus, joka oikein käytettynä lisää hallittavuutta yritysten liiketoimintatietojen käsittelyssä. (MSDN 2008.)



## 2.2.2 SQL Management Studio

SQL Management Studio on Microsoft SQL Server 2005:een liittyvä työkalu, jonka avulla pystyy hallinnoimaan, muokkaamaan ja päivittämään Microsoft SQL Server 2005:n komponentteja. SQL Management Studio toimii myös yhteistyössä uudempien SQL Serverien kanssa. Se näyttää sekä skripti-näkymän että graafisen näkymän ja työkalut, joten sen käyttö on todella selkeää. (MSDN 2008.)

## 2.3 VAHTI-tietokanta

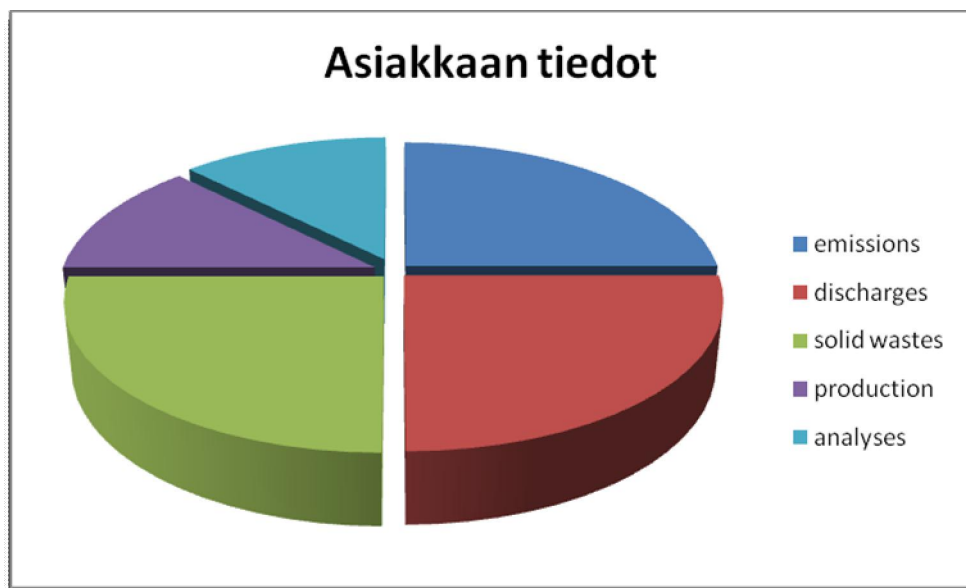
VAHTI on Suomen ympäristöhallinnon aluehallinnon tietohallintopalveluyksikön ylläpitämä valvonta ja kuormitustietojärjestelmä, joka palvelee Suomen kolmeatoista alueellista ELY-keskusta ympäristölupien valvonnassa ja tarkkailussa. Järjestelmä on julkaistu vuonna 1997 ja se sisältää tietoja mm. jätevesikuormituksista, ilman epäpuhtauksista, päästöistä ja jätteistä (Valvonta ja kuormitustietojärjestelmä (VAHTI). 2010). VAHTI toimii osana Ympäristönsuojelun tietojärjestelmää ja täten sinne tallennetaan myös ympäristönsuojelulainsäädännön mukaisten lupien ja ilmoitusten tietoja. Vuonna 2005 jokapäiväisiä käyttäjiä VAHTIlla oli 800. (SYKE 2005.)



Kuva 2 Asiakastiedot

Yläpuolella olevassa kuvassa on kuvattu esimerkki mahdollisista tiedoista, joita asiakkaasta tallennetaan VAHTIin. Kuvassa asiakkaasta on tallennettu asiakkaan tunnistaminen (customer identification), lisenssit (licences), dokumenttikansiot (document folders) ja asiakkaan kuvaus (customer description). Vuonna 2003 VAHTI sisälsi tiedot 31 000 asiakkaasta, nykyisin tuo luku on paljon suurempi. (SYKE 2005.)

Vaikka VAHTI onkin julkaistu vasta 90-luvun lopulla, se sisältää ympäristönkuormitus-tietoja ja arvoja jo 1970-luvulta lähtien. Nämä vanhemmat tiedot eivät kuitenkaan ole aina kovinkaan kattavia tai luotettavia. Yleensä arvot, joita VAHTIsta halutaan, esitetään vuosiarvoina. (SYKE 2005.)



Kuva 3 Asiakkaan tiedot

Yläpuolella olevassa kuvassa on kuvattu esimerkkitapaus tiedoista, joita asiakkaalla voi olla VAHTI-tietokannassa. Kuvassa näkyvät päästöt (emissions), virtaamat (discharges), kiinteät jätteet (solid wastes) ja tuotanto (production), joka on jaettu yhteiseksi kokonaisuudeksi analyysitulosten (analyses) kanssa. (SYKE 2005.)

VAHTIlla on erillinen raporttiosa, johon pääsee ympäristöhallinnon ja SYKE:n intranetin kautta. Testikäytössä on kokeilussa rajoitettu raporttiosa asiakkaita varten ekstranetissä. VAHTI:n raporttiosasta saa tulostettua valmiiksi tehtyjä raporttipohjia ja valita niihin halutut tiedot, jotka on koodattu valittaviksi kentiksi käyttöliittymään. Raport-

tipohjia on tehty vuosien aikana ympäristöhallinnon omana työnä VAHTIin käyttäjien toiveiden ja tarpeiden mukaan. Raporttiosa palvelee suurta osaa käyttöryhmien tarpeista, mutta ei kaikkia. Muutamit henkilöt SYKEllä tekevät hakuja suoraan tietokannasta tai muiden ohjelmien mm. Accessin ja SAS:n kautta.

VAHTI koostuu kaiken kaikkiaan 202 taulusta. Tietokanta on jaettu neljään päätauluun: ASIAKAS, OSIO, PISTE ja JAKSO. Taulujen alla olevien alakokonaisuuksien määrä vaihtelee kahdeksan ja 22 välillä, alakokonaisuudet sisältävät alatauluja yhdestä 14:ään. Alla olevassa taulukossa on kuvattu päätaulut, niiden alakokonaisuudet ja alakokonaisuuksien alataulujen määrät.

päätaulut	alakokonaisuudet	alataulujen lukumäärä alakokonaisuuksissa
ASIAKAS	vedenotto	2kpl
	raaka-aineet	2 kpl
	energiankäyttö	2 kpl
	ymp.suojeluinvestoinnit	2 kpl
	tuotantoyksiköt	1 kpl
	tuotanto	2 kpl
	kemikaali	1 kpl
	kalenteri	2 kpl
	ympäristövahinkovakuutus	2 kpl
	tarkastuspöytäkirjat	7 kpl
	VPLS-ilmoitukset	2 kpl
	VOC-ilmoitukset	6 kpl
	häiriö- ja yleisöilmoitukset	6 kpl
	dokumentit	1 kpl
	VAHTI-asiakastiedot	4 kpl
	laatu- ja johtamisjärjestelmät	2 kpl
	raportointi	7 kpl
	tarkkailuohjelma	3 kpl
	satamailmoitukset	2 kpl

	luvat ja velvoitteet vesiensuojelumaksut valvontaviranomaiset	14 kpl 2 kpl 5 kpl
OSIO	polttoaine ja kemikaalivarastot kiinteistöt eläinsuojat jätekuljettajat kalankasvatus turvetuotanto kaatopaikat eläinsuojat	3 kpl 3 kpl 11 kpl 14 kpl 3 kpl 6 kpl 4 kpl 6 kpl
PISTE	energiatuotanto johtamisvuorokaudet viikkovirtaamat käyttöaika päästöt laitekatkokset polttoaineet jäteseuranta jätevedenpuhdistamot näytteet puhdistusprosessit kattilat luotettavuus	2 kpl 1 kpl 1 kpl 1 kpl 1 kpl 2 kpl 2 kpl 8 kpl 4 kpl 4 kpl 2 kpl 2 kpl 3 kpl
JAKSO	koostuu PISTE-, OSIO- ja ASIA- KAS- päätaulujen kokonaisuuksista sekä lisäksi yksiköt parametrit jakso	 3 kpl 3 kpl 2 kpl

VAHTI –tietokanta on tietojärjestelmätyypiltään lähimpänä tapahtumien käsittelyjärjestelmää jatkuvan päivitettävyyden ja tietojen muuttumisen vuoksi. Myös tiedon ajanta-

kaisuus ja oikeellisuus ovat erittäin tärkeitä ominaisuuksia VAHTIlla. Lisäksi siinä näkyy raportoinnin- ja analyysien tietojärjestelmän tunnusmerkkejä suuren tietokannan koon ja tiedon nopeuden osalta.

VAHTI ei käytä DBMS:ään tietokantakonetta vaan se tietokanta on jo valmiiksi yhdistetty tarvittaviin ohjelmointi- ja raportointivälineisiin. VAHTI:ssa käytetään reaaliaikais- ta käsittelyä, jotta tieto pysyy jatkuvasti ajantasaisena.

## 2.4 TIVA2-hanke

TIVA2-hanke on Suomen ympäristöministeriön vuonna 2008 käynnistämä hanke uudistaa ympäristötietojen tietovarasto VAHTI –tietokannan pohjalta. Määräaika hankkeelle on 31.12.2011 asti. TIVA2:lla on kolme osaprojektia: VAHTI - tietovarastoprojekti, uhanalaisten eliöiden Internet-raportit ja karttapalvelinalustan kehittäminen. Uuden ympäristötietovaraston on tarkoitus palvella käyttäjiään, mm. valvojia, laitoksia, asiakkaita ja peruskansalaisia, mahdollisimman monipuolisesti. Tietovarasto rajataan käyttöoikeuksilla joka käyttäjäryhmän tarpeisiin sopivaksi. (SYKE 2010b.)

Hankkeen tavoitteena on toteuttaa VAHTI –tietokannan tietoihin perustuva tietovarasto, jonka lisäksi tietoja on tarkoitus pystyä hyödyntämään raportointiin ja sen kehittämiseen. Lisäksi hankkeen on tarkoitus organisoida tietovarastot ja niistä saatavat raportoinnin ylläpito ja jatkokehitys. Toisena päätavoitteena on kehittää pohja keskitetylle ympäristötietovarastolle, johon on tarkoitus tulevaisuudessa ruveta lisäämään muitakin kuin VAHTI:n tietoja. Tietojen yhdistelemisen ja jalostamisen avulla on tarkoitus saada parempaa ja kattavampaa tietoa ympäristön tilasta. (SYKE 2010a.)

Tällä hetkellä TIVA2-hanke on vielä suunnittelu- ja määrittelyvaiheessa. Hankkeen projektiryhmä miettii vielä käyttötapauksia ja muita vastaavia asioita (SYKE 2010b). Tekemäni projekti saattaa auttaa TIVA2-hankkeen projektiryhmää uuden tietovaraston suunnittelussa.

Lopputuloksina hankkeelta odotetaan VAHTI –jätevesitietoja sisältävä tietovarasto sekä tietovarastosta tehdyt raportit ja välineet itsenäistä käyttöä varten. Tarkoituksena on, että TIVA2 –hankkeesta tulee työkalu raportointia varten ja sen lisäksi siitä saadaan selkeä kuvaus sekä jatkokehittämissuunnitelmat. (SYKE 2010a.)

## 3 Empiirinen osa

### 3.1 Esittely

Projekti kohdistuu VAHTI -tietokannan jätevesiä koskeviin tauluihin ja niihin kytköksissä oleviin tietoihin. Jätevedet sijaitsevat VAHTI -tietokannassa PISTE-päätaulun alaisuudessa. Alakokonaisuuden nimi on Jätevedenpuhdistamot ja sen sisällä on neljä alataulua. Määrätietoja haetaan Näyteanalyysi –alataulusta, joka sijaitsee myös PISTEen alla. JAKSO-päätaulun alta haetaan parametri- ja yksikkötiedot. Tarkoituksena oli tehdä hakuja VAHTI –tietokantaan ja saada raportteja, jotka palvelisivat SYKE:n tietotarpeita. Lisäksi haut dokumentoitiin ja ohjeistettiin siten, että kokemattomammakin tietokantaohjelmien käyttäjät pystyvät samoja työkaluja käyttämällä tekemään vastaavanlaisia hakuja.

### 3.2 Ongelmat ja kehittäminen

VAHTI:n suurimpana ongelmana mielestäni on, että se on erittäin raskas käyttää. Jo pelkästään VAHTI-tietokannassa on 202 taulua, jonka lisäksi asiakastiedot sijaitsevat erillisissä Asiakas- ja Asiakasrekisteri –tietokannoissa. Lisäksi taulujen sisällä olevat attribuutit on nimetty usein samalla nimellä kuin toisissa tauluissa olevat. Esimerkiksi Status-attribuuttia käytetään erittäin monessa taulussa, joten sen selityksen jäljittäminen tarvittavassa yhteydessä on hankalaa. Usein hakujen sisällä haettavat tiedot vaihtelevat suuresti eri päätaulujen alaisuudessa, tästä syystä joutuu yhdistelemään JOIN-lauseiden avulla useita lisätauluja. Ongelmana on liitosten suuri määrä, joka tekee haun liitososista hyvin pitkän. Tällöin sen luominen vie suuresti aikaa ja vaatii laajaa selvitystyötä ennen kuin selviää, mikä liitos kytkeytyy mihinkin siten, että haku toimii yhä.

Puuttuvien tietojen etsiminen tietokannasta on hankalaa, koska rivit luodaan sitä mukaan, kun tiedot niihin syötetään. Ilmeisesti tietokannassa ei ole aktiivisilla laitoksilla mitään pakollista syötettävää vuositietoa, joten puuttuvia tietoja on mahdoton löytää vuoden perusteella. Toistaiseksi puuttuvat tiedot yritetään etsiä vertailemalla tauluja toisiinsa, esimerkiksi kaikki aktiiviset laitokset sekä kaikki laitokset, joilla on tietoja vuodelta 2010 ovat hyvä vertailupari.

Asiakastietojen listauksessa ongelmana ovat asiakkaiden erilaiset nimeämiset vuosien saatossa. Välillä nimet on kirjoitettu isoilla kirjaimilla ja välillä pienillä, joissain nimissä on ylimääräisiä välilyöntejä sekä vanhat nimet jäävät tietokantaan aktiivisina. Lisäksi yrityksen nimen muuttamisen yhteydessä tietokanta luo yritykselle kokonaan uuden asiakastietorivin ja samalla jättää vanhan tiedon sinne.

Taulukoiden tietojen listauksessa vastaan tuli ongelma, ettei eriarvoisia saman parametrin omaavia arvoja saa aseteltua rinnakkain taulukossa vaan ne tulostuvat allekkain. Vertailun vuoksi olisi kätevämpi, jos arvot näkyisivät taulukossa rinnakkain.

### 3.3 Toteutus

Projekti aloitettiin 6.9.2010 Suomen ympäristökeskuksen Mechelininkadun toimipisteessä. Käytössäni on oma huone ja siellä tietokone. Projekti alkoi kaikkien tarpeellisten tunnusten ja oikeuksien saamisella tarvittaviin tietoihin, ohjelmiin ja tietokantaan. Seuraavat pari viikkoa menivät ohjelmiin ja tietokantaan tutustumisessa. Tein yksinkertaisia SQL-hakuja VAHTI -tietokannasta linkittäen useampia tauluja toisiinsa vähitellen (Liite 1 Hakuskripti 1). Sen jälkeen tutustuin Asiakas- ja Asiakasrekisteri -tietokantoihin. Aloin tehdä haastavampia hakuja yhdistäen kaikki kolme tietokantaa toisiinsa (Liite 2 Hakuskripti 2). Lisäsin haulle ehtoja ja määreitä WHERE -käskyjen avulla (Liite 3 Hakuskripti 3). Tulostin hauista saatuja raportteja ulos SYKEN tarpeiden mukaan. Lisäksi tein esimerkkihakuun ohjeet ja selityksen, jotta kokemattomamminkin tietokannan käyttäjät pystyvät samojen käytössä olevien työkalujen avulla tulkitsemaan tehtyjä SQL-lauseita ja muuttamaan niitä omia tarpeitaan vastaaviksi (Liite 5 VAHTI-ohje).

Sen lisäksi pidin ensin ohjaajalleni erillisen demon keskiviikkona 27.10. tekemistäni hauista ja mahdollisuuksista käyttää tietokantaa omatoimisesti käyttäen apuna tekemiäni ohjeita. Demossa kävimme läpi alusta alkaen ohjelman käynnistämisen ja tarvittavat etukäteistiedot, jotta tietokannasta löytäisi haluamansa tiedot. Teimme muutaman yksinkertaisen haun, jossa haimme yhdestä tai kahdesta taulusta muutaman tiedon. Sitten keskityimme tarkemmin jo valmiiksi tekemääni hakuskriptiin, jota käytetään jatkossa



hakujen tekemisessä. Kävimme läpi skriptin ja selitin mitä mikäkin käsky tarkoittaa ja tekee. Sitten katsoimme läpi kohdat, joita muutetaan, kun tehdään toisia hakuja. Lopuksi ohjaajani haki itsenäisesti valmista skriptiä itse muuttaen haluamiaan tietoja tietokannasta.

Pidin maanantaina 15.11. suuremman demon, jossa paikalla oli neljä henkilöä lisäkseni. Heistä kolmelle asia oli uusi ja lähdimmekin liikkeelle aivan perusasioista. Ensin haimme tarvittavan SQL Server Management Studio –ohjelman demossa käytettävään koneeseen. Sen jälkeen etenimme samalla tavoin kuin ensimmäisessä demossa, mutta keskityimme enemmän valmiin skriptin muokkaamiseen kuin itse alusta asti tehtäviin hakuihin. Tämän demon yhteydessä kävi ilmi, että kaikille jaetuista ohjeista puuttui tietojen siirto Exceliin, joten lisäsin sen sinne.

Lisäksi demossa selvisi mahdollisia syitä siihen miksi monissa taulukoissa oli poikkeavia arvoja. Esimerkiksi parametri\_id –taulukossa biokemiallisen hapenkulutuksen (BOD) määrittämisen osalta on neljä eri mahdollisuutta valita BOD-parametri: BOD, BOD<sub>5</sub>, BOD<sub>7</sub> sekä BOD<sub>7</sub>ATU. Vaihtoehdot ovat listassa kaukana toisistaan, joten kokemattomammat käyttäjät (tietojen syöttäjät/tietojen hakijat) valitsevat helposti listan ensimmäisen BOD-parametrin. Lisäksi yksikkö\_id –listasta on erittäin helppo valita alusvetopalkista väärä yksikkö, esimerkiksi mg/kg vaihtuu nopeasti g/kg. Myös tarkoituksella valitut eri yksiköt, esimerkiksi osassa on käytetty prosenttiarvoja ja osassa mg/kg-arvoja, sekoittavat taulukoiden tekoa huomattavasti.

### 3.4 Tulokset

SQL-lauseiden luonti alkoi hyvin yksinkertaisista hauista. Ensimmäisissä hauissa hain (SELECT-käsky) tietoja monesta VAHTIn taulusta samanaikaisesti (Liite 1 Hakuskripti 1). Ensimmäisessä tapauksessa hain Jateseuranta-, Jateseurantaalkupera-, Piste- ja Laite-laitekatkos-tauluista (FROM-käsky) tietoja, joita en jäsennellyt mitenkään. Seuraavassa haussa hain Kattila, Piste- ja Osio-tauluista tietoja. Kriteereiksi (WHERE-käsky) hauille tulivat, ettei selitekenttä saa olla tyhjä, toisen selitteen tulee olla eläinsuoja, kattilatyyppin

tulee olla arvo 3, järjestysluvun tulee olla 4, hakuun tulevat vain vuoden 2010 tulokset sekä nimen tulee olla eläinsuoja.

Seuraavaksi tein hieman monimutkaisempia hakuja, joissa hain VAHTI –tietokannan lisäksi Asiakas- ja Asiakasrekisteri –tietokannoista tietoja (Liite 2 Hakuskripti 2). Asiakas- ja Asiakasrekisteri –tietokantojen taulujen nimet tuli ilmaista muodossa tietokannan nimi.dbo.taulun nimi.

Viimeinen hauista olikin jo kohtalaisen pitkä ja monimutkainen (Liite 3 Hakuskripti 3). Haku haki 12 eri taulusta tietoja, joista viisi sijaitsivat toisissa tietokannoissa kuin VAHTI. Juuri tästä syystä WHERE -määrittelystä tuli todella pitkä. Taulujen yhdistäminen Pää- ja viiteavainten suhteen oli hankalaa vain selvitysosuudella, varsinainen tekeminen sujui pohjatöiden jälkeen helposti. Yhdistelyiden lisäksi määrittelin parametrille kaksi arvoa, jotka tulivat mukaan hakuun, lajeiksi yhdyskuntajätevedenpuhdistamoja tarkoittavat AS:t, vuodeksi 2009, laitosten tuli olla aktiivisia sekä selite alkoi Tuleva-etuliitteellä. Lopuksi vielä järjestelin (ORDER BY-käskey) tulokset yrityksen nimen mukaan.

## 4 Pohdinta

### 4.1 Tulosten tarkastelu

Suunnittelu käynnistyi hitaasti, koska alussa oli hieman epäselvää mitä tietoja SYKE halusi ulos tietokannasta. Apunani suunnittelussa oli VAHTIn perusteella tehty käsitekaavio, joka kuitenkin oli niin suuri, että oli mahdotonta liittää sitä tähän mukaan. Käsitekaavion avulla oli mahdollista alkuvaiheessa käsittää kuinka suuri tietokanta oli kyseessä ja nähdä miten mikäkin taulu liittyy toisiinsa. Suuren tietokannan, jossa on jo itsessään 202 taulua, hallinnointiin oli hyvä olla apuna esimerkiksi käsitekaavio etenkin, kun jouduttiin tekemään lisäksi linkityksiä muihin tietokantoihin. Kaaviosta myös näkyvät pääavaimet ja siten sai linkityksistä jonkinlaisen käsityksen.

Varsinainen SQL-käskyjen luominen onnistui hyvin ja kohtuullisen nopeasti. Katsoin mallia vanhoista kurssitehtävistä, jos en muistanut kaikkea tarvitsemaani. Myös internetistä löysin tarvittaessa apua. Etenin helpommista kyselyistä vähitellen kohti vaikeampia lisäten aina jonkin ominaisuuden lisää hakuun. Virheilmoitukset olivat suurimmaksi osaksi joko kirjoitusvirheitä tai vääriä parametrien nimeämisiä. Osa yrittämistäni hauista ei onnistunut, koska tietokanta ei pystynyt vastaamaan niihin. Esimerkiksi, kun yritin tulostaa kaikki laitokset, joilta puuttuivat tulokset vuodelta 2009, tietokanta ei pystynyt tekemään sitä. Jo edellä ongelmassa kuvattu ongelma siitä miten tietokantaan ei ole luotu mitään pakollista kenttää täytettäväksi, joka tallentaisi vuoden, aiheutti tämän tilanteen.

Toistuvuus oli suurin ongelma tulosten vertailussa. Yhdellä yrityksellä saattoi olla yhden tuloksen sijaan viisi tulosta, koska sen nimi oli kirjoitettu hieman eri muodossa joka kerta tietokantaan. Exceliin siirtämällä osan toistuvuudesta sai poistettua, mutta esimerkiksi kirjoitusvirheelliset nimet tulostuivat useamman kerran. Lopulta tulokset piti tarkastaa manuaalisesti, jotta saattoi olla varma, ettei samoja rivejä ollut montaa kertaa.

Raporttien tulostaminen sujui hyvin. Osa raporteista oli liian leveitä, joten ne jatkuivat toiselle sivulle, mutta muokkaamalla niitä ne tulostuivat siististi. Raportit olivat välillä

todella suuria, joten sivuja kertyi sadoittain. Yritin usein rajata haut mahdollisimman suppeiksi, jotta raporttien tulostaminen onnistuisi helpommin ja nopeammin.

Edellä mainitut ongelmat olivat mielestäni arvattavissa jo etukäteen, kun kuulin, ettei VAHTIa ole aiemmin raportoitu millään muotoa. Tietokannan koko tuli minulle hie- man yllätyksenä, en ollut alussa aivan tajunnut miten suuri se on ja miten se on yhtey- dessä Asiakas- ja Asiakasrekisteri –kantoihin.

Itse SQL-käskyjen luomiset sujuivat hyvin ja suurin osa kyselyistä tuotti halutut rapor- tit. Muutamia raportteja ei ollut tietokannan rakenteen vuoksi mahdollista toteuttaa, joka tuli pienenä yllätyksenä. Suurin ongelma johtuikin juuri siitä syystä, että tietokan- nassa ei ollut pakollisia vuosittain täytettäviä kenttiä. Tästä johtuen tietokannasta on mahdotonta saada puuttuvia tietoja selville, ellei tietokantaa uudista tai tee kokonaan uutta.

## 4.2 Tulosten luotettavuus

Tulokset ovat mielestäni luotettavat, jos käytössä ovat samat työkalut kuin minulla. Kenties muilla (lisä)ohjelmilla tuloksia saisi enemmän irti ja järkevämmässä muodossa ulos kuin näillä. Myös tietokantaosaajat, joilla on enemmän kokemusta tietokannoista, saattavat osata joitain ominaisuuksia, joita minä en vielä osaa. Kaiken kaikkiaan uskon tulosten silti esittävän kattavan kuvan siitä millainen tietokanta VAHTI on ja millaisia raportteja siitä saa ulos.

## 4.3 Yhteenveto ja johtopäätökset

Projekti onnistui odotetulla tavalla määräajassa. Muutamat raportit, joita ei saanut tie- tokannasta ulos, eivät johtuneet minusta vaan tietokannan luontivaiheessa tehdyistä ratkaisuksista. Projekti oli onnistunut ja palveli SYKE:n tarpeita hyvin. Ohjaajani olivat tyytyväisiä projektin lopputulokseen ja sain kiitosta tekemästäni työstä.

## 4.4 Jatkokehittämis ehdotukset

Projektia tässä muodossa tuskin jatketaan, mutta TIVA2-hankkeen tarkoituksena on poistaa tässä projektissa havaittuja puutteita koskien VAHTI-tietokantaa. TIVA2-hanke etenee pian suunnitteluvaiheesta varsinaiseen tekovaiheeseen ja tuloksia pitäisi näkyä vuoden 2011 puolella. Lisäksi tekemiäni ohjeiden ja pitämieni demojen perusteella neljän ihmisen tulisi osata nyt käyttää tietokantaa tarpeisiinsa muokkaamalla valmiiksi tekemiäni hakulauseita oman tarpeen mukaiseksi. Demossa olleiden ihmisten joukossa oli myös muutama, jotka ovat mukana suunnittelemassa TIVA2-hanketta ja he saivat lisätietoja siitä mitä VAHTI:ssä tulisi kehittää.

## 4.5 Oppiminen

Tässä määritellään ensiksi opinnäytetyöprosessin onnistuminen sekä tarkastellaan pitkö alun perin sovittu aikataulu paikkansa. Sen jälkeen tarkastellaan omaa oppimistani opinnäytetyön suhteen. Sen yhteydessä tarkastellaan heikkouksiani ja vahvuuksiani työn suhteen.

### 4.5.1 Opinnäytetyöprosessin arviointi

Opinnäytetyöprosessi sujui mielestäni hyvin sujuvasti alun pienen kangertelun jälkeen. Juuri alku oli haastavin, koska minulla ei ollut käsitystä millainen tietokanta on, mitä minun pitää sieltä hakea ja osaanko tehdä sen. Siitä eteenpäin kaikki sujuikin omalla painollaan.

Mielestäni aikaa oli juuri sopivasti varattu opinnäyteprosessia varten ja projektin aikataulu oli hyvin realistinen. Varsinainen opinnäytetyön kirjoittaminen sujui hieman hitaammin kuin alun perin oli tarkoitus, mutta työ itsessään valmistui ajallaan. Sain tarvittavaa ohjausta aina, kun sitä tarvitsin ja työt etenivät sujuvasti.

#### 4.5.2 Oman osuuden arviointi

Opin paljon uutta tietokantoihin liittyen ja monet opintojen perusteella tiedetyt teoriat vakiintuivat käytännön taidoiksi. SQL Server Management Studio –ohjelman käyttö myös muuttui huomattavan paljon helpommaksi mitä edemmäs projekti eteni.

Vahvuuksinani pidän kykyä keskittyä saman asian tekemiseen monta tuntia kerrallaan. Samoin lauseiden luonnissa jaksoin hyvin tehdä pitkiä ketjutuksia ja yhdistää tietokantoja toisiinsa.

Heikkouksinani pidin kokemuksen puutetta vastaavista töistä, siitä johtuen alussa kului enemmän aikaa tietokantaan ja ohjelmiin tutustumisessa. Samoin jouduin testaamaan mitä toimintoja tietokannalla oli mahdollista tehdä, kun pätevämpi ihminen olisi jo pystynyt suoraan sanomaan mihin tietokanta pystyy. Lisäksi opinnäytetyön kirjoittaminen oli haastavampaa kuin aiemmat raportit, joita olen tehnyt, joten aikaa kului hieman kauemmin kuin alun perin olin luullut.

## Lähteet

Connolly T. & Begg, C: 2004. Database Solutions. Pearson Education Limited. Essex, England.

Elmasri, R. & Navathe, S. 2006. Fundamentals of Database Systems. Addison Wesley. USA.

Garcia-Molina, H. & Ullman, J. & Widom, J. 2008. Database Systems: The Complete Book. Prentice-Hall. USA.

Microsoft 2005. Microsoft julkisti SQL Server 2005:n, Visual Studio 2005:n ja BizTalk Server 2006:n Suomessa. Tiedote. Luettavissa:  
[http://www.microsoft.com/finland/pr/press/microsoftjulkisti3\\_24nov2005.msp](http://www.microsoft.com/finland/pr/press/microsoftjulkisti3_24nov2005.msp).  
Luettu: 1.10.2010.

Morley, D. & Parker, C. 2007. Understanding Computers: Today and tomorrow. Thomson Course Technology. Massachusetts, USA.

MSDN 2010. Database informations from the library. Seloste. Luettavissa:  
<http://msdn.microsoft.com/en-us/library/>. Luettu 5.10.2010.

MSDN 2008. Using SQL Server Management Studio. Seloste. Luettavissa:  
<http://msdn.microsoft.com/en-us/library/ms174173.aspx>. Luettu 5.10.2010.

SYKE 2005. The Compliance Monitoring Data system – VAHTI. Seloste. Luettavissa:  
<http://www.environment.fi/default.asp?contentid=142451&lan=EN>. Luettu:  
30.9.2010.

SYKE 2010a. Ympäristövarantojen käytön kehittäminen, vaihe 2 TIVA2-HANKE.  
Vahti-tietovarasto PROJEKTISUUNNITELMA.

SYKE 2010b. Muistio SYKE:n johtoryhmän kokouksesta 22.9.2010 (14. kokous / 2010).

Tietokanta. 2010. Seloste. Luettavissa: <http://fi.wikipedia.org/wiki/Tietokanta>. Luettu 1.10.2010.

Valvonta ja kuormitustietojärjestelmä (VAHTI). 2010. Seloste. Luettavissa: <http://www.i5.ymparisto.fi/i5/463t.htm>. Luettu: 24.9.2010.



## Liitteet

### Liite 1 Hakuskripti 1

```
select jatelaji, kokonaispaino, selite_suomi, selite, jarjestysluku, piste.paivays, maara,  
laitelaitekatkos.paivays  
from jateseuranta, jateseurantaalkupera, piste, laitelaitekatkos
```

```
select kattilatyyp_i_id, polttoteho, kayttoonottopvm, jarjestysluku,  
p.selite, wanhatunnus, p.paivays, nimi, s.selite, s.paivays  
from kattila, piste p, osio s  
where p.selite is not null  
and s.selite = 'eläinsuoja'  
and kattilatyyp_i_id = 3  
and jarjestysluku = 4  
and datepart(year, s.paivays) = 2010  
and nimi = 'eläinsuoja'
```

## Liite 2 Hakuskripti 2

```
select distinct a.asiakas_id, a.nimi1, b.yritys, c.paivays, d.mitoitusvirtaamatunti,  
d.mitoitusvirtaamapaiva  
from asiakasrekisteri.dbo.asiakas(nolock) a, asiakas.dbo.asiakas(nolock) b, asiakas_osio  
bc, osio bcc, osio_piste bccc, piste c, jatevedenpuhdistamomitoitus d
```

### Liite 3 Hakuskripti 3

```
select
a.asiakas_id,
f.selite,
/*e.nimi,*/
a.yritys,
ad.laji1,
ad.laji2,
ab.selite,
b.selite,
c.parametri_id,
c.maara,
c.yksikko_id,
cb.alkupaivamaara,
cb.loppupaivamaara

from
asiakas.dbo.asiakas(nolock) a,
piste b,
nayteanalyysi c,
asiakas_osio bc,
osio bcc,
osio_piste bccc,
nayte cb,
asiakas.dbo.asiakastyyppi(nolock) ab,
osiolaji ad,
asiakasrekisteri.dbo.asiakastila(nolock) e,
asiakasrekisteri.dbo.asiakas(nolock) ea,
asiakas.dbo.status(nolock) f

where e.tila_id = ea.tila_id
and ea.asiakas_id = a.asiakas_id
```

```
and a.asiakastyyppi_id = ab.asiakastyyppi_id
and f.status_id = a.aktiivinen
and a.asiakas_id = bc.asiakas_id
and bc.osio_id = bcc.osio_id
and ad.laji_id = bcc.laji_id
and bcc.osio_id = bccc.osio_id
and bccc.piste_id = b.piste_id
and b.piste_id = cb.piste_id
and cb.nayte_id = c.nayte_id
and (c.parametri_id = 84
or c.parametri_id = 279)
and ad.laji1 = 'AS'
and ad.laji2 = 'AS'
and datepart(year,cb.alkupaivamaara) = 2009
and f.selite = 'aktiivinen'
and b.selite like 'tuleva%'

order by a.yritys
```

## Liite 4 SQL-hakujen selitys

```
select
a.asiakas_id,
f.selite,
/*e.nimi,*/
a.yritys,
ad.laji1,
ad.laji2,
ab.selite,
b.selite,
c.parametri_id,
c.maara,
c.yksikko_id,
cb.alkupaivamaara,
cb.loppupaivamaara
```

halutut sarakkeet, jotka näkyvät taulukossa, /\*...\*/ tarkoittaa että merkkien välissä oleva teksti on huomautus ja ohjelma ei lue sitä, parametrien edessä olevat kirjaimet ovat kyseisten luokkien nimien lyhenteitä, jotka selitetään auki alla

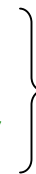
```
from
asiakas.dbo.asiakas(nolock) a,
piste b,
nayteanalyysi c,
asiakas_osio bc,
osio bcc,
osio_piste bccc,
nayte cb,
asiakas.dbo.asiakastyyppi(nolock) ab,
osiolaji ad,
asiakasrekisteri.dbo.asiakastila(nolock) e,
asiakasrekisteri.dbo.asiakas(nolock) ea,
asiakas.dbo.status(nolock) f
```

tässä esitellään edellä haettujen parametrien luokat, joista tiedot löytyvät, luokan nimen perässä on lyhenne (keksitty itse päästä), jota käytetään muissa tilanteissa viitattaessa luokkaan, jos luokka tulee toisesta tietokannasta kuin, jossa itse olet, siihen tulee tehdä johdatus kuten esim. ensimmäisessä kohdassa on tehty

```
where e.tila_id = ea.tila_id
and ea.asiakas_id = a.asiakas_id
and a.asiakastyyppi_id = ab.asiakastyyppi_id
and f.status_id = a.aktiivinen
and a.asiakas_id = bc.asiakas_id
and bc.osio_id = bcc.osio_id
and ad.laji_id = bcc.laji_id
and bcc.osio_id = bccc.osio_id
and bccc.piste_id = b.piste_id
and b.piste_id = cb.piste_id
and cb.nayte_id = c.nayte_id
```

linkitykset tauluista toiseen, tavoitteena on tehdä rikkoutumaton ketju, taulujen pää- ja viiteavaimet on yhdistetty (pääavaimen tunnistaa avaimen kuvasta parametrin edessä)

```
/*and c.parametri_id = 84*/  
and ad.laji1 = 'AS'  
and ad.laji2 = 'AS'  
/*and datepart(year,cb.alkupaivamaara) = 2009*/  
and f.selite = 'aktiivinen'
```



rajauksia ja hakuetoja, joita halutaan asettaa

```
order by a.yritys
```



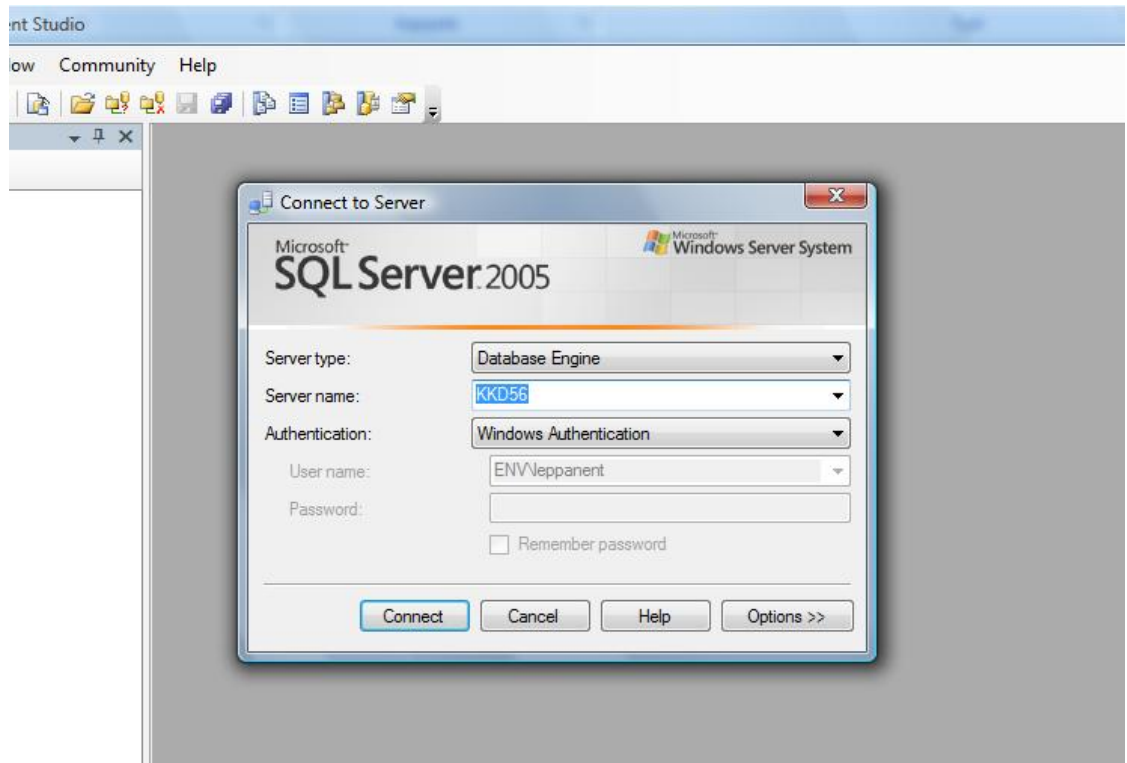
sarake, jonka mukaan tiedot halutaan järjestää

## Liite 5 VAHTI-ohje

Ohjeet SQL Server 2005:n käyttöön VAHTI – tietokannan suhteen

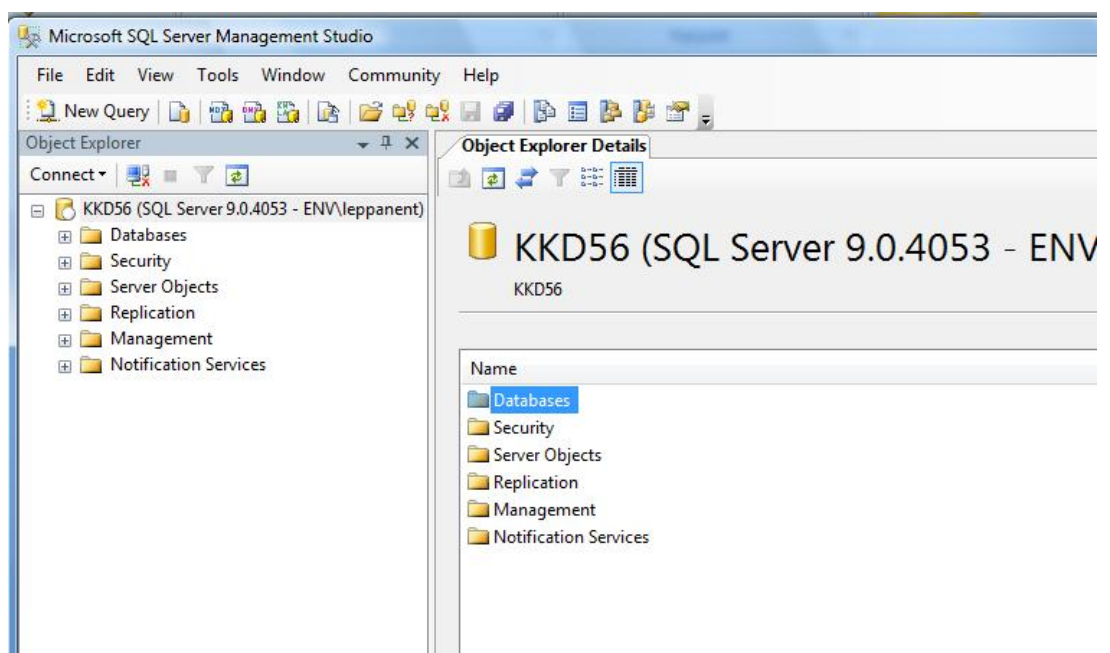
Tiina Leppänen

Käyttäjä avaa IT-tuen asentaman SQL Server Management Studion. Näkyviin tulee alla olevan kaltainen serveririkysely.



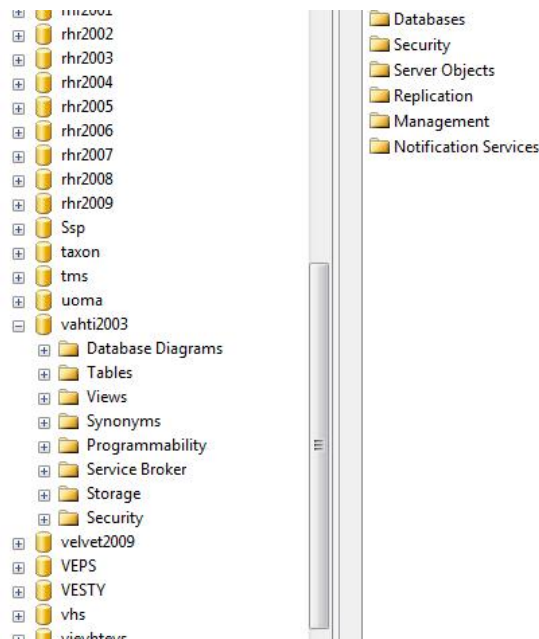
Server name –kenttään kirjoitetaan KKD56 ja painetaan Connect.

Seuraavanlainen näkymä ilmestyy:

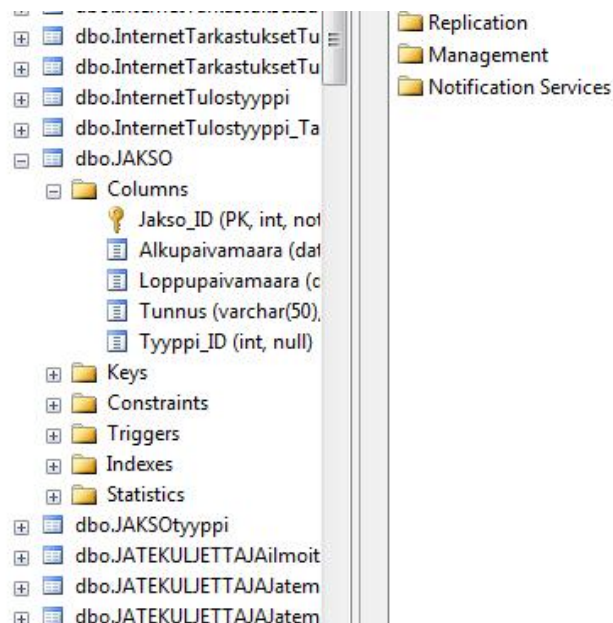




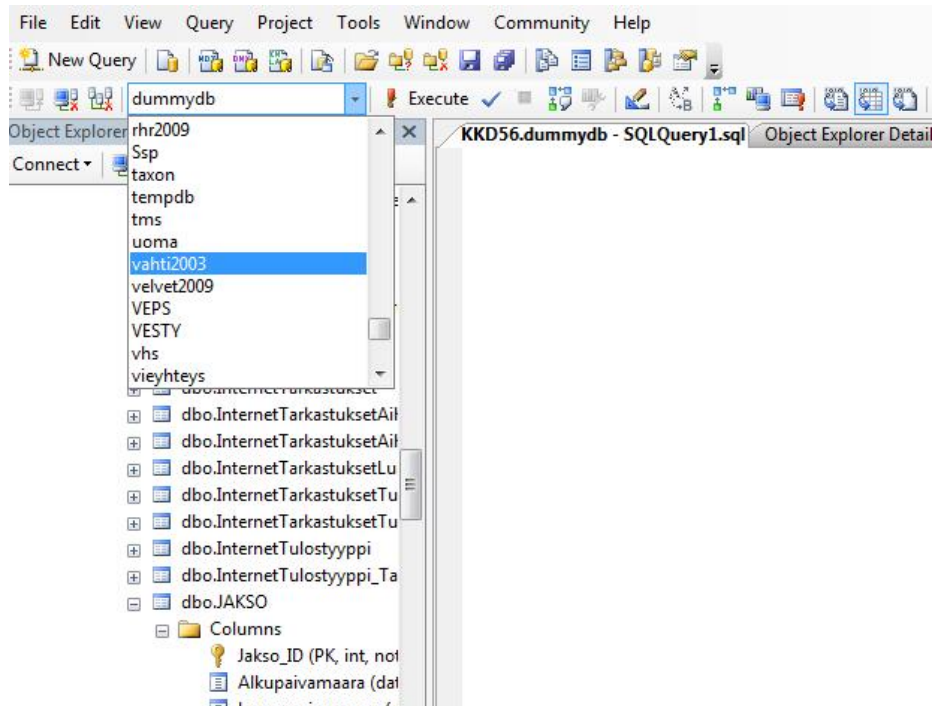
Kaikessa käsittelyssä kannattaa käyttää vasemman puoleista tietokantavalikkoo. Aluksi klikataan Databases edessä olevaa + -merkkiä, sitten sen alta paljastuvasta pitkästä tietokantalistasta vahti 2003 edessä olevaa + -merkkiä (*HUOM! Asiakastiedot löytyvät asiakas ja asiakasrekisteri tietokannoista!*). Näkymä on nyt tämän näköinen:



Seuraavaksi klikataan Tables -kansion edessä olevaa + -merkkiä. Nyt kaikki Vahti 2003 -tietokannassa olevat taulut tulevat näkyviin. Otetaan esimerkkinä taulu JAKSO, joka klikataan auki + -merkistä ja Column myös + -merkistä. Näkymä on nyt tällainen:



Tässä näkymässä siis näkee jakso-taulun sarakkeet. Jos haluaa nähdä tauluun tallennetut tiedot, painetaan yläpalkin alta New Query –painiketta. Sen alla on alasvetovalikko, jossa on valittuna dummydb, se pitää vaihtaa vahti 2003:ksi (*jos haluat mennä asiakastietoihin niin sitten valitset joko asiakas tai asiakasrekisteri*). Näkymä näyttää nyt tältä:



Nyt alustukset on tehty ja SQL –käskyjen kirjoittaminen voi alkaa. Yksinkertainen käsky, jolla näkee taulujen sisällön on `select * from jakso`(*eli jakson voi korvata halutun taulun nimellä*). Sen kirjoittamisen jälkeen painetaan Execute -painiketta, jossa punainen huutomerkki. Kirjoitetun käskyn alle tulee näkyviin taulukko tietoineen.

Jakso_ID	Alkupaivamaara	Loppupaivamaara	Tunnus	Tyyppi_ID
1	1944-12-01 00:00:00.000	1944-12-31 00:00:00.000	1944 12/12	1
2	1964-01-01 00:00:00.000	1964-12-31 00:00:00.000	1964	12
3	1965-01-01 00:00:00.000	1965-12-31 00:00:00.000	1965	12
4	1966-01-01 00:00:00.000	1966-12-31 00:00:00.000	1966	12
5	1967-01-01 00:00:00.000	1967-12-31 00:00:00.000	1967	12
6	1968-01-01 00:00:00.000	1968-12-31 00:00:00.000	1968	12

Jos halutaan näkyviin vain tunnus, tyyppi\_id ja alkupaivamaara muokataan käskyä hie-  
 man: `select tunnus, tyyppi_id, alkupaivamaara from jakso`. Näin saa näkyviin tau-  
 lukon, jossa on tulokset tietokoneen valitsemassa järjestyksessä. Jos haluaa lajitella ne  
 esim. tunnuksen mukaan käänteisessä järjestyksessä (eli uusimmat ensin) niin kirjoite-  
 taan edellisen käskyn perään `order by tunnus desc` (desc on käänteinen, asc on nor-  
 maali järjestys, normaali tosin tulee ilman mitään päätettä tunnuksen jälkeen).

```

select tunnus, tyyppi_id, alkupaivamaara
from jakso
order by tunnus desc
  
```

	tunnus	tyyppi_id	alkupaivamaara
1	2020 9/12	1	2020-09-01 00:00:00.000
2	2020 8/12	1	2020-08-01 00:00:00.000
3	2020 7/12	1	2020-07-01 00:00:00.000
4	2020 6/12	1	2020-06-01 00:00:00.000
5	2020 5/12	1	2020-05-01 00:00:00.000
6	2020 4/4	3	2020-10-01 00:00:00.000
7	2020 4/12	1	2020-04-01 00:00:00.000
8	2020 3/4	3	2020-07-01 00:00:00.000
9	2020 3/3	4	2020-09-01 00:00:00.000
10	2020 3/12	1	2020-03-01 00:00:00.000

Jos halutaan määritellä kriteereinä haulle, että näkyviin tulevat vain tulokset, joissa  
 tyyppi\_id on 3, käsky on `select tunnus, tyyppi_id, alkupaivamaara from jakso  
 where tyyppi_id = 3 order by desc`. Oikeassa alareunassa näkyvät aina tulosten rivien  
 määrät.

```

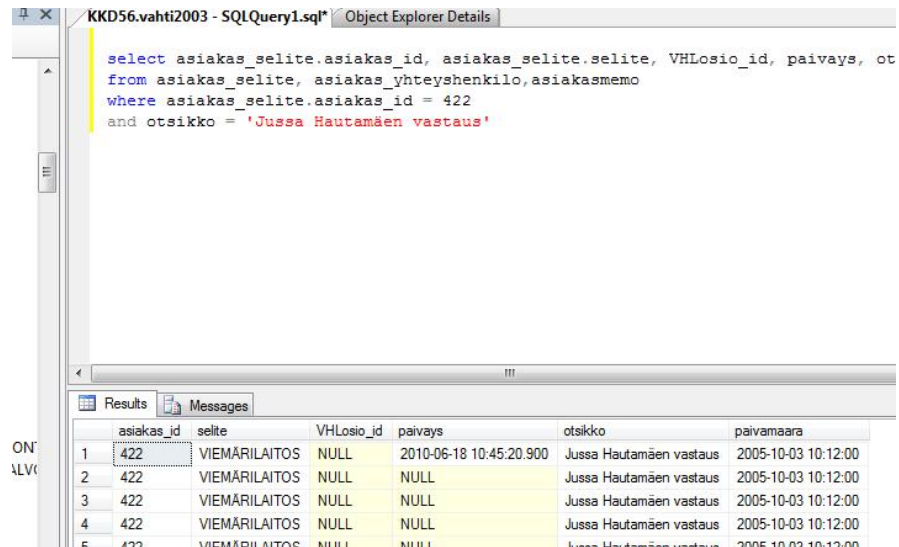
select tunnus, tyyppi_id, alkupaivamaara
from jakso
where tyyppi_id = 3
order by tunnus desc
  
```

	tunnus	tyyppi_id	alkupaivamaara
1	2020 4/4	3	2020-10-01 00:00:00.000
2	2020 3/4	3	2020-07-01 00:00:00.000
3	2020 2/4	3	2020-04-01 00:00:00.000
4	2020 1/4	3	2020-01-01 00:00:00.000
5	2019 4/4	3	2019-10-01 00:00:00.000

Query executed successfully. KKD56 (9.0 SP3) ENV\neppanent (97) vahti2003 00:00:00 145 rows

Määritellessä kriteereinä tekstiä, kyseinen sana/sanat/lause pitää pistää lainausmerkkeihin eli kahden ' -merkin väliin. Sisällön pitää vastata tarkasti määriteltyä kriteeriä.

```
select asiakas_selite.asiakas_id, asiakas_selite.selite, VHLosio_id, paivays,
otsikko, paivamaara
from asiakas_selite, asiakas_yhteyshenkilo,asiakasmemo
where asiakas_selite.asiakas_id = 422
and otsikko = 'Jussa Hautamäen vastaus'
```



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```
select asiakas_selite.asiakas_id, asiakas_selite.selite, VHLosio_id, paivays, ot
from asiakas_selite, asiakas_yhteyshenkilo,asiakasmemo
where asiakas_selite.asiakas_id = 422
and otsikko = 'Jussa Hautamäen vastaus'
```

The results pane displays a table with the following data:

asiakas_id	selite	VHLosio_id	paivays	otsikko	paivamaara
422	VIEMÄRILAITOS	NULL	2010-06-18 10:45:20.900	Jussa Hautamäen vastaus	2005-10-03 10:12:00
422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:00
422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:00
422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:00
422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:00

Jos tarkkaa määritelmää tekstille ei haluta/voida tehdä, = -merkin tilalle kirjoitetaan sana like ja % -merkki kertoo, että ennen/jälkeen sanaa voi olla muuta tekstiä.

```
select asiakas_selite.asiakas_id, asiakas_selite.selite, VHLosio_id, paivays,
otsikko, paivamaara
from asiakas_selite, asiakas_yhteyshenkilo,asiakasmemo
where asiakas_selite.asiakas_id = 422
and otsikko like '%Juss%'
```

The screenshot shows a SQL query window with the following text:

```

select asiakas_selite.asiakas_id, asiakas_selite.selite, VHLosio_id, paivays
from asiakas_selite, asiakas_yhteyshenkilo, asiakasmemo
where asiakas_selite.asiakas_id = 422
and otsikko like '%Juss%'

```

Below the query, the Results pane displays a table with the following data:

	asiakas_id	selite	VHLosio_id	paivays	otsikko	paivamaara
1	422	VIEMÄRILAITOS	NULL	2010-06-18 10:45:20.900	Jussa Hautamäen vastaus	2005-10-03 10:12:01
2	422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:01
3	422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:01
4	422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:01
5	422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:01
6	422	VIEMÄRILAITOS	NULL	NULL	Jussa Hautamäen vastaus	2005-10-03 10:12:01

Jos halutaan löytää rivit, joiden päivämäärä ei ole null eli tyhjä, käsky on

```

select asiakas_selite.asiakas_id, asiakas_selite.selite, VHLosio_id, paivays,
otsikko, paivamaara
from asiakas_selite, asiakas_yhteyshenkilo, asiakasmemo
where asiakas_selite.asiakas_id = 422
and otsikko like '%Juss%'
and paivays is not null

```

Jos halutaan rivit, joiden päivämäärä on tyhjä, poistetaan not –sana välistä.

The screenshot shows a SQL query window with the following text:

```

select asiakas_selite.asiakas_id, asiakas_selite.selite, VHLosio_id, paivays, c
from asiakas_selite, asiakas_yhteyshenkilo, asiakasmemo
where asiakas_selite.asiakas_id = 422
and otsikko like '%Juss%'
and paivays is not null

```

Below the query, the Results pane displays a table with the following data:

	asiakas_id	selite	VHLosio_id	paivays	otsikko	paivamaara
1	422	VIEMÄRILAITOS	NULL	2010-06-18 10:45:20.900	Jussa Hautamäen vastaus	2005-10-03 10:12:00
2	422	VIEMÄRILAITOS	NULL	2008-08-20 11:08:57.997	Jussa Hautamäen vastaus	2005-10-03 10:12:00
3	422	VIEMÄRILAITOS	NULL	2009-03-31 03:40:21.577	Jussa Hautamäen vastaus	2005-10-03 10:12:00
4	422	VIEMÄRILAITOS	1900006692	2008-10-20 12:22:11.977	Jussa Hautamäen vastaus	2005-10-03 10:12:00
5	422	VIEMÄRILAITOS	NULL	2009-04-24 09:32:46.720	Jussa Hautamäen vastaus	2005-10-03 10:12:00

Taulujen yhdistäminen toisiinsa aiheuttaa monimutkaisempia lauseita.

```
select
a.asiakas_id,
a.yritys,
ad.laji1,
ad.laji2,
ab.selite,
b.selite,
c.parametri_id,
c.maara,
c.yksikko_id,
c.paivays
```

ennen haluttuja attribuutteja olevat kirjaimet ovat lyhenteitä taulujen nimistä (itse päätetyt kirjaimet) ja ne on esitelty alempana

```
from
asiakas.dbo.asiakas(nolock) a,
piste b,
nayteanalyysi c,
asiakas_osio bc,
osio bcc,
osio_piste bccc,
nayte cb,
asiakas.dbo.asiakastyyppe(nolock) ab,
osiolaji ad
```

vahti 2003-tietokannasta olevat taulut on esitelty samoin kuin aiemmin, mukaan on vain lisätty kirjainlyhenteet helpottamaan kirjoittamisessa

asiakas –tietokannasta haetut taulut vaativat pidemmän nimeämisen, jossa ensin mainitaan tietokannan nimi, sitten dbo ja lopuksi taulun nimi, jonka perässä lukee nolock ja lyhenne

```
where a.asiakastyyppe_id = ab.asiakastyyppe_id
and a.asiakas_id = bc.asiakas_id
and bc.osio_id = bcc.osio_id
and ad.laji_id = bcc.laji_id
and bcc.osio_id = bccc.osio_id
and bccc.piste_id = b.piste_id
and b.piste_id = cb.piste_id
and cb.nayte_id = c.nayte_id
and datepart(year,c.paivays) = 2010
```

tässä osiossa yhdistetään taulut toisiinsa, sekä eri tietokannoista että samoista, pitää ensin etsiä yhteiset kahdesta peräkkäisestä taulusta löytyvät samat parametrit, jotka sitten yhdistetään toisiinsa, sama toistetaan kunnes kaikki kyselyssä mukana olevat taulut on linkitetty toisiinsa

```

and c.parametri_id = 84
and ad.laji1 = 'AS'
and ad.laji2 = 'AS'
/*and c.maara < 1*/

```

määritellyt kriteerit haulle,  
 /\*...\*/ välissä olevat tekstit  
 ovat kommentteja eli niitä ei  
 oteta mukaan kyselyyn

order by a.yritys

```

asiakas.dbo.asiakastyyppi (nolock) ab,
osiolaji ad

where a.asiakastyyppi_id = ab.asiakastyyppi_id
and a.asiakas_id = bc.asiakas_id
and bc.osio_id = bcc.osio_id
and ad.laji_id = bcc.laji_id
and bcc.osio_id = bccc.osio_id
and bccc.piste_id = b.piste_id
and b.piste_id = cb.piste_id
and cb.nayte_id = c.nayte_id
and datepart(year,c.paivays) = 2010
and c.parametri_id = 84
and ad.laji1 = 'AS'
and ad.laji2 = 'AS'
/*and c.maara < 1*/

```

	asiakas_id	yritys	laji1	laji2	selite	selite	parametri_id	maara	yksikko_id	paiva
1	375	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	10	28	2010-
2	375	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	9,7	28	2010-
3	375	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	12	28	2010-
4	375	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	11	28	2010-
5	375	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Vesistöön	84	0,86	28	2010-
6	375	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Vesistöön	84	0,2	28	2010-

Lopullinen haku, jota muokkailen:

```
select
a.asiakas_id,
f.selite,
/*e.nimi,*/
a.yritys,
ad.laji1,
ad.laji2,
ab.selite,
b.selite,
c.parametri_id,
c.maara,
c.yksikko_id,
cb.alkupaivamaara,
cb.loppupaivamaara

from
asiakas.dbo.asiakas(nolock) a,
piste b,
nayteanalyysi c,
asiakas_osio bc,
osio bcc,
osio_piste bccc,
nayte cb,
asiakas.dbo.asiakastyyppe(nolock) ab,
osiolaji ad,
asiakasrekisteri.dbo.asiakastila(nolock) e,
asiakasrekisteri.dbo.asiakas(nolock) ea,
asiakas.dbo.status(nolock) f

where e.tila_id = ea.tila_id
and ea.asiakas_id = a.asiakas_id
and a.asiakastyyppe_id = ab.asiakastyyppe_id
```



```

and f.status_id = a.aktiivinen
and a.asiakas_id = bc.asiakas_id
and bc.osio_id = bcc.osio_id
and ad.laji_id = bcc.laji_id
and bcc.osio_id = bccc.osio_id
and bccc.piste_id = b.piste_id
and b.piste_id = cb.piste_id
and cb.nayte_id = c.nayte_id
and (c.parametri_id = 84
or c.parametri_id = 279)
and ad.laji1 = 'AS'
and ad.laji2 = 'AS'
and datepart(year,cb.alkupaivamaara) = 2009
and f.selite = 'aktiivinen'
and b.selite like 'tuleva%'

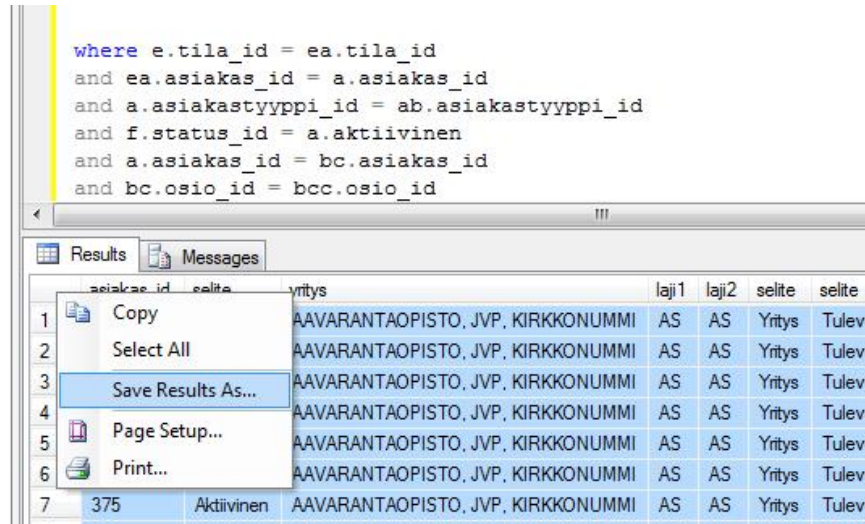
order by a.yritys

```

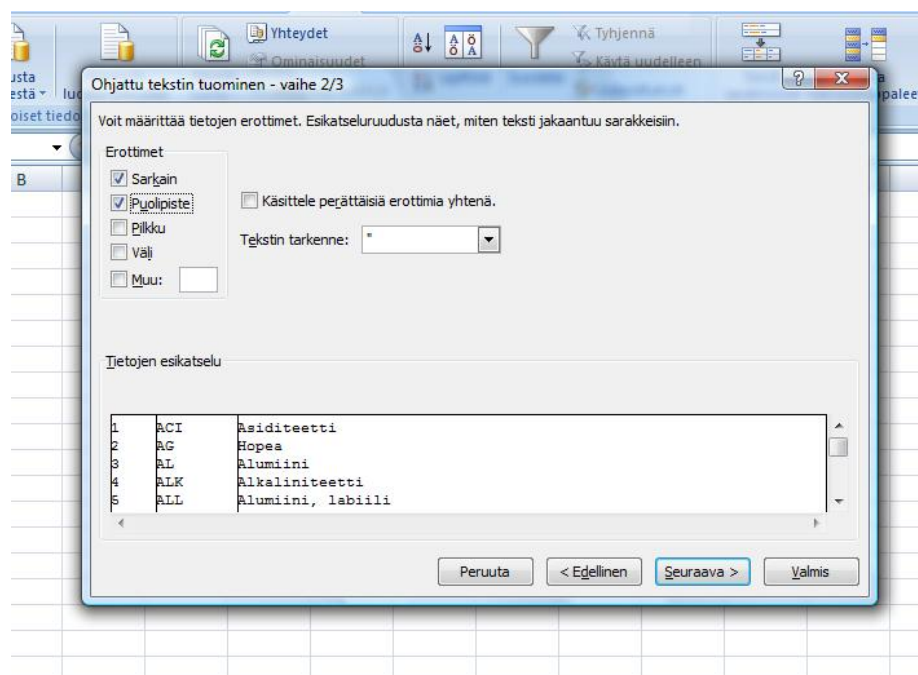
The screenshot shows a SQL query in a text editor and its results in a table. The query filters for active companies in 2009 with specific parameters and company types. The results table shows 6 rows of data for company ID 375, with different parameter IDs and company types.

	asiakas_id	selite	yritys	laji1	laji2	selite	selite	parametri_id	maara	yksikko
1	375	Aktiivinen	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	10	28
2	375	Aktiivinen	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	279	220	28
3	375	Aktiivinen	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	14	28
4	375	Aktiivinen	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	279	710	28
5	375	Aktiivinen	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	84	12	28
6	375	Aktiivinen	AAVARANTAOPISTO, JVP, KIRKKONUMMI	AS	AS	Yritys	Tuleva (laskenta)	279	220	28

Taulukon siirtäminen Excelliin tapahtuu maanaamalla koko taulukon painamalla ennen ensimmäistä otsikkosaraketta olevaa tyhjää ruutua, seuraavaksi painetaan hiiren oikealla näppäimellä Save Results As. Tallenna taulukko haluamaasi paikkaan .csv-muotoisena.



Avaa Excelliin normaali tyhjä sivu. Tiedot-välilehdeltä valitaan vasemmasta reunasta Tekstistä-painike. Sieltä etsitään tallennettu taulukko ja valitaan se. Seuraavaksi avautuvasta ikkunasta valitaan ylempi kohta eli Erotettu. Seuraavaa painamalla valitaan erottimeksi Puolipiste ja sen jälkeen voidaankin painaa Valmis-painiketta.



Sen jälkeen tiedot ovat Excelissä ja niistä voi tehdä esimerkiksi diagrammeja.