

Metropolia Ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Saku Kirves

Laitehallintarekisteri

Insinöörityö 25.11.2010

Ohjaaja: johtava konsultti Juha Hirvonen
Ohjaava opettaja: lehtori Olli Hämäläinen

Tekijä Otsikko	Saku Kirves Laitehallintarekisteri
Sivumäärä Aika	34 sivua 25.11.2010
Koulutusohjelma	tietotekniikan
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	johtava konsultti Juha Hirvonen lehtori Olli Hämäläinen
<p>Nykyajan yrityksissä on erilaisia laitteita vuosien saatossa kertynyt paljon, eivätkä tiedot laitteista välttämättä ole helposti löydettävissä. Tiedot voivat olla kansioissa paperille tulostettuna tai lukuisissa tiedostoissa. Eri henkilöt yrityksessä ovat saattaneet kirjata hankittujen laitteiden tiedot eri tavalla, ja tiedot ovat hajautuneet.</p> <p>Tässä insinööriyössä suunniteltiin ja toteutettiin keskitetty hallintajärjestelmä yrityksen laitteille. Keskitetyllä hallintajärjestelmällä laitteiden kirjaaminen tapahtuu hallitusti ja samojen periaatteiden mukaisesti. Niiden hallinnointi sekä ylläpito nopeutuvat, ja usea käyttäjä voi yhtä aikaa ylläpitää laitteiden tietoja. Laitetietojen keskitetty sijainti mahdollistaa myös nopeat ja täsmälliset hakutoiminnot, joiden avulla pystyy tarkistamaan täsmällisiä tietoja laitteista ja niiden haltijoista. Järjestelmän käyttöönottoa sekä käyttöä helpottaa sen käyttö web-selaimella, jolloin välttyään ohjelman asentamiselta jokaisen käyttäjän koneelle sekä mahdollistetaan sovelluksen käyttö jokaiselta koneelta, jolla on pääsy yrityksen sisäverkkoon.</p> <p>Sovelluksen toteutuksessa keskityttiin käyttämään WebFOCUS-tuoteperheen sovelluskehittäjiä WebFOCUS Developer Studiota ja Maintain Development Environmentia. Lisäksi käytettiin web-pohjaisissa sovelluksissa yleisesti käytettyjä teknologioita, kuten HTML, CSS ja JavaScript. Tietojen varastointiin käytettiin Microsoft SQL Server -relaatiotietokantaa.</p> <p>Sovelluksen toteutuksessa suuren huomion sai monikielisyys, joka mahdollistaa saman sovelluksen käytön monikansallisen yrityksen eri toimipisteissä. Monikielisyys helpottaa sovelluksen käyttöönottoa ja mahdollistaa tehokkaan käytön. Myös tietoturva, käyttäjän tunnistaminen ja salasanojen salaus olivat keskeisiä toteutuksessa.</p> <p>Työssä toteutettiin laitehallintajärjestelmä keskisuuren yrityksen tarpeisiin. Laitetiedot saatiin päivitettyä, ja jatkossa niiden ylläpito on helppoa. Työ opetti käyttämään valittuja sovelluskehittäjiä tehokkaasti monikielisen sovelluksen tekemiseen sekä paljasti työmäärän kasvun yksikieliseen sovellukseen verrattuna. Tiedoista on hyötyä jatkokehitystyössä sekä monikielisuuden aiheuttaman työmäärän arvioinnissa.</p>	
Hakusanat	laitehallinta, sovelluskehitys, relaatiotietokanta, raportointi, WebFOCUS, SQL

Author Title	Saku Kirves Device management system
Number of Pages Date	34 pages 25 November 2010
Degree Programme	Information Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Juha Hirvonen, Senior Consultant Olli Hämäläinen, Senior Lecturer
<p>Numerous devices tend to accumulate in companies. The data regarding the devices may be stored as prints, or in multiple files, and it can be fragmented due to employees' personal styles in keeping the records.</p> <p>The aim was to build a centralised management system for the devices of a company. In a centralised system, the recording of data is controlled and follows given principles. The management and maintenance of devices becomes more efficient, and several users can participate in updating the information.</p> <p>Centralised storage of data enables fast and precise searches producing accurate information regarding the devices and the persons possessing them. The system is used with a web browser, making the implementation and usage easier: the application is available to every computer in the company's intranet without installation of a new program.</p> <p>The application design tools selected for the project were WebFOCUS product family members WebFOCUS Developer Studio and Maintain Development Environment. Technologies, widely used in web-based applications, such as HTML, CSS and JavaScript were applied. A relational database, Microsoft SQL Server, was used for storage of data.</p> <p>Multilinguality, allowing the use of the same application everywhere in a multinational company, was in the focus of the design. Multilinguality is essential to the implementation and enables efficient use of the system. In addition, data security, identification of the user and encryption of passwords were emphasized.</p> <p>In this project, a device management system for a mid-size company was developed. The device data were updated, and the future maintenance will be easy. The project taught the effective use of the selected tools to create multilingual application. A substantial increase in workload due to the multilinguality was revealed. This information will be useful for calculation of the workload of application development.</p>	
Keywords	device management, application development, relational database, reporting, WebFOCUS, SQL

Sisällys

Tiivistelmä

Abstract

1	Johdanto	5
2	Teknologiat	7
3	Työvälineiden käyttö	9
3.1	WebFOCUS Developer Studio	9
3.2	Maintain Development Environment	10
3.3	Microsoft SQL Server Management Studio	13
4	Tietokanta	15
4.1	Relaatiotietokannan perusteita	15
4.2	Suunnittelu	16
4.2.1	Kielitaulu	16
4.2.2	Laitetaulu	18
4.2.3	Henkilötaulu	19
4.2.4	Yritystaulu	19
4.2.5	Ohjaustaulut	20
4.2.6	Taulujen suhteet	20
4.3	Toteutus	21
4.3.1	Taulujen luonti ja muokkaus	21
4.3.2	Tietojen vienti tietokantaan	23
5	Sovelluksen toteutus	25
5.1	Kirjautuminen	25
5.2	Kielenvalinta	26
5.3	Laitehaku	26
5.4	Lisäys	29
5.5	Päivitys	29
5.6	Poisto	30
5.7	Raportit	30
6	Yhteenveto	32
	Lähteet	34

1 Johdanto

Jo keskisuurissa tietotekniikkayrityksissä on huomattava määrä erilaisia laitteita. Osa on henkilökunnalla omassa käytössä ja osa toimistolla yhteiskäytössä. Yleensä laitteet on hankittu pitkän ajanjakson aikana ja hankkijoina ovat voineet toimia monet henkilöt, joilla on ollut omat tapansa kirjata tiedot muistiin.

Laitekannasta tarvitaan ajoittain tietoja esimerkiksi uusia hankintoja suunniteltaessa ja työntekijöiden vaihtuessa. Tarvittavia tietoja voivat olla esimerkiksi aktiivisessa käytössä olevien tietokoneiden lukumäärä, kuinka vanhoja käytössä olevia puhelimia löytyy tai mitä yrityksen omaisuutta tietyllä työntekijällä on hallussaan. Jos laitehallintajärjestelmä ei ole toimiva ja ajantasainen, näiden tietojen saaminen tarvittaessa on lähes mahdotonta tai ainakin haastavaa.

InfoBuild Oy on vuonna 1986 Suomessa perustettu yritys, jolla on nykyään myyntikonttoreita myös Tallinnassa, Riiassa, Vlnassa, Oslossa ja Tukholmassa. Yritys edustaa Information Builder Inc:n ja iWay Software Inc:n tarjontaa. Yrityksen toiminta kattaa ohjelmistojen lisenssien myynnin, tilaustyönä tehtyjen sovellusten kehittämisen, teknisen konsultoinnin, tuen ja koulutuksen. InfoBuild kuuluu United Software Distributors Oy-konserniin.

Laitehallinnan suhteen yrityksen toimintatapa ei vastannut nykyajan vaatimuksia: keskitettyä laitehallintajärjestelmää ei ollut lainkaan, vaan tiedot olivat hajallaan lukuisissa Excel-taulukoissa. Tilanteen korjaamiseksi haluttiin luoda laitteiden hallintajärjestelmä, joka täyttäisi nykyiset vaatimukset laitetietojen helposta ylläpidosta ja sujuvasta tietojen hakemisesta.

Tehtävänä oli suunnitella ja toteuttaa sovellus yrityksen laitetietokannan ylläpitämiseen ja tietojen hakemiseen. Sovellusta käytettäisiin selaimella, joten käyttäjän koneelle ei tarvitsisi asentaa uutta ohjelmaa, vaan sovellus ajettaisiin yrityksen sovelluspalvelimella.

Sovelluksen määrittelyssä otettiin huomioon sen käyttö eri kielillä, sillä InfoBuildillä on toimintaa monessa maassa. Ohjelman monikielisyys oli välttämättömyys, jotta ohjelma voitaisiin ottaa kattavasti käyttöön.

Työkaluina sovelluksen kehityksessä käytettiin InfoBuildin maahantuomia WebFOCUS-tuoteperheen tuotteita.

2 Teknologiat

WebFOCUS

WebFOCUS kuuluu Information Builders Inc:n tuoteisiin. Tuote on kokenut pitkän kehityskaaren ja on syntynyt jo vuonna 1975 ennen web-aikaa nimellä FOCUS. Tuote koostuu kahdesta kehitystyökalusta, palvelimelle asennettavista komponenteista sekä tietokanta-adapttereista.

Työkalut ovat raportointiin keskittynyt WebFOCUS Developer Studio sekä sovellusten ylläpitoon kehitetty WebFOCUS Maintain. Molemmat toimivat tietokanta-adaptteri-rajapinnan yläpuolella, joten ne ovat tietokantariippumattomia ja sama lähdekoodi toimii kaikkien tietokantojen kanssa.

Palvelinohjelmistot ovat WebFOCUS Client ja Reporting Server, joka on yhteydessä tietokantoihin natiiviadapterien kautta. Omien komponenttien lisäksi palvelimella on oltava jokin sovelluspalvelinohjelmisto, kuten Apache Tomcat, BEA Weblogic tai IBM WebSphere. Lisäksi useimmiten palvelimelle asennetaan web-palvelinohjelmisto, joka yleisimmin on joko Microsoft IIS (Internet Information Services) tai Apache HTTP Server. Lisäksi palvelimella tarvitaan Java Virtual Machine (JVM). [1, s. 28.]

Web-sovelluksen täydentävät tekniikat

HTML (HyperText Markup Language) on käytetyin merkintäkieli, jolla kuvataan WWW (World Wide Web) -sivuja [2, s. 2–3]. HTML ei ole ohjelmointikieli, vaan siinä käytetään tekstin seassa merkintäelementtejä, joilla määritellään sivun ominaisuuksia. Merkintäelementtien avulla WWW-sivuilla näkyy muun muassa muotoiltua tekstiä, linkkejä, kuvia ja lomakkeita. [3.]

CSS-kieli (Cascading Style Sheets) on mekanismi, jonka avulla määritellään web-sivujen ulkoasu. [2, s. 243; 4.] Koodi sijoitetaan joko suoraan HTML-koodiin tai linkitetään erillinen tiedosto HTML-sivulle. CSS-kielillä pystyy yhdellä määrittelyllä antamaan tyyliasetukset useille komponenteille eri WWW-sivuilla.

JavaScript on komentosarjakieli, jota käytetään HTML-sivuille upotettuna [5, s. 296]. Kielen avulla sivuista pystytään tekemään dynaamisemmat ja käyttäjäystävällisemmät kuin pelkällä HTML-koodilla. JavaScript-koodia ei käännetä, vaan selaimessa oleva JavaScript-tulkki suorittaa koodin rivi kerrallaan. Kielen vahvuuksia on laitteistoriippumattomuus, keveys ja yksinkertaisuus. [6.]

Microsoft SQL Server

Microsoft SQL Server on yleisesti yritysmaailmassa käytetty relaatiotietokantapalvelin. Microsoft kehitti SQL Serveriä yhdessä Sybasen kanssa 1993 asti, jonka jälkeen molemmat yritykset ovat kehittäneet omaa versiotaan ohjelmistosta [7, s.14–15; 8, s. 10–11]. Microsoft SQL Serverin vahvuuksia ovat joustavuus ja laajennettavuus pienistä ympäristöistä massiivisiin klusteroituihin ympäristöihin [9, s. 3].

Internet Information Services

Internet Information Services on Microsoftin kehittämä web-palvelinohjelmisto, joka tulee palvelimien Windows-käyttöjärjestelmien mukana ja on näin yleisesti yritysmaailmassa käytetty.

Apache Tomcat

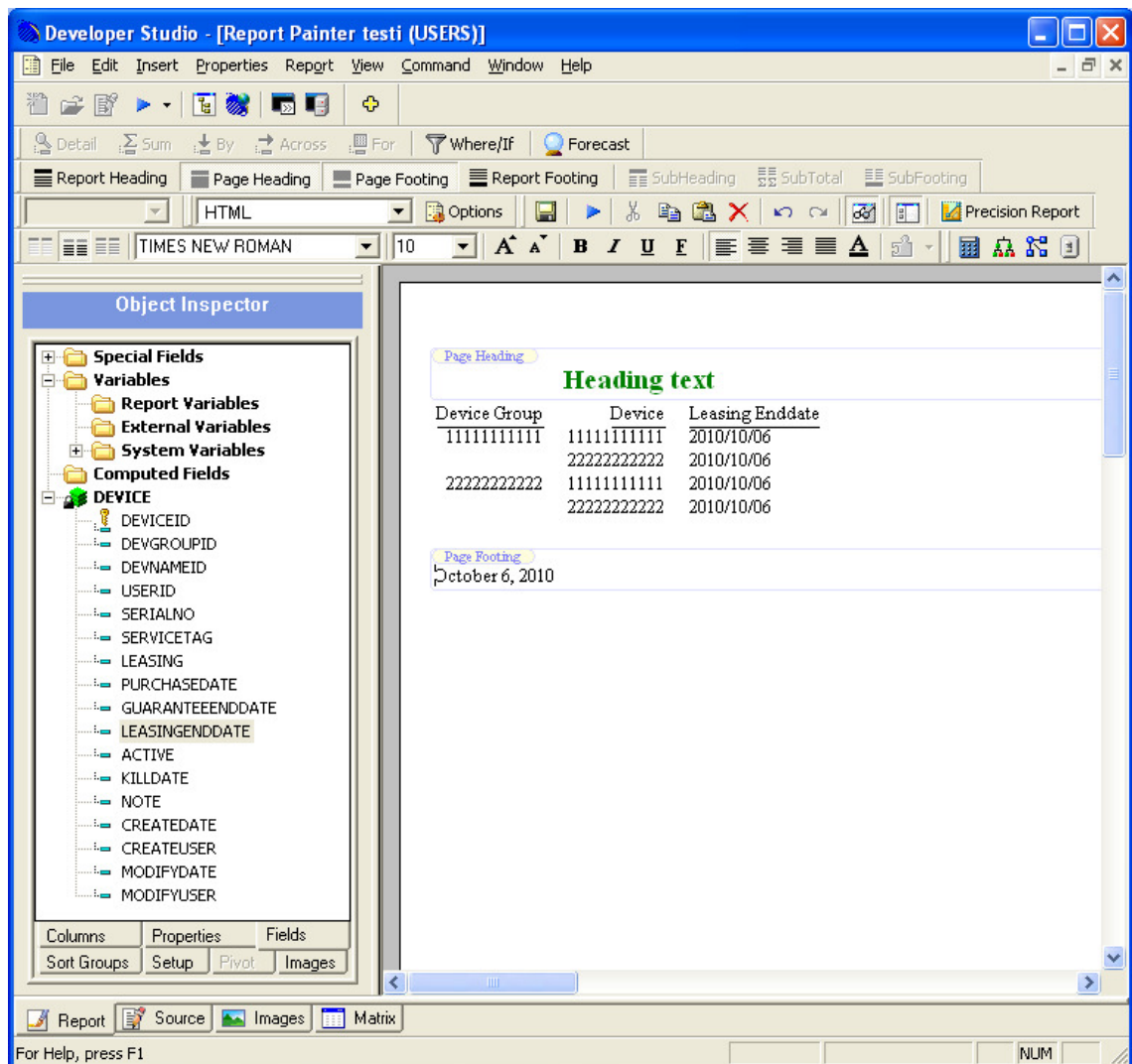
Apache Tomcat on avoimeen lähdekoodiin perustuva Apache Software säätiön kehittämä sovelluspalvelin, jossa voidaan suorittaa Java Servlet ja JavaServer Pages ohjelmia, jotka tuottavat web-sivuja [10]. Se on yksinkertainen ja toimiva ilmaisohjelma, minkä vuoksi se on levinnyt erittäin suosituksi.

3 Työvälineiden käyttö

3.1 WebFOCUS Developer Studio

WebFOCUS Developer Studio on tietokantaraporttien kehittämistyökalu. Sillä pystyy tekemään monipuolisia raportteja eri tulostusmuotoihin. Yleisimpinä raportin muotoina ovat HTML, PDF, Excel ja graafit, joita voi näyttää sellaisenaan kuvatiedostona tai niitä voi upottaa useita yhteen PDF-dokumenttiin tai HTML-sivulle [11, s. 462–463].

Kuvassa 1 on työkalun graafinen käyttöliittymä. Sillä pystyy helposti tekemään monipuolisia raportteja. Kokeneemmat ohjelmoijat käyttävät kuitenkin graafisen työkalun ohella tekstieditoria tehostaakseen työskentelyä käyttämällä kehittyneitä tekniikoita, joita ei ole toistaiseksi pystytty graafiseen liittymään sisällyttämään. Ohjelmointikielenä käytetään FOCUS-kieltä, joka on varsin monipuolinen ja voimakas kieli. Se on syntynyt kauan ennen web-aikaa ja kehittynyt ajan kuluessa vastaamaan nykyisiä vaatimuksia. Kieli on monipuolinen, ja erityisesti raportin muotoilu on hyvin kattavaa. Vaikka FOCUS-kieli eroaa huomattavasti SQL-kielestä, se on helposti luettavaa ja sitä ymmärtää melko vaivattomasti, vaikka ei olisikaan kokemusta kyseisestä kielestä.

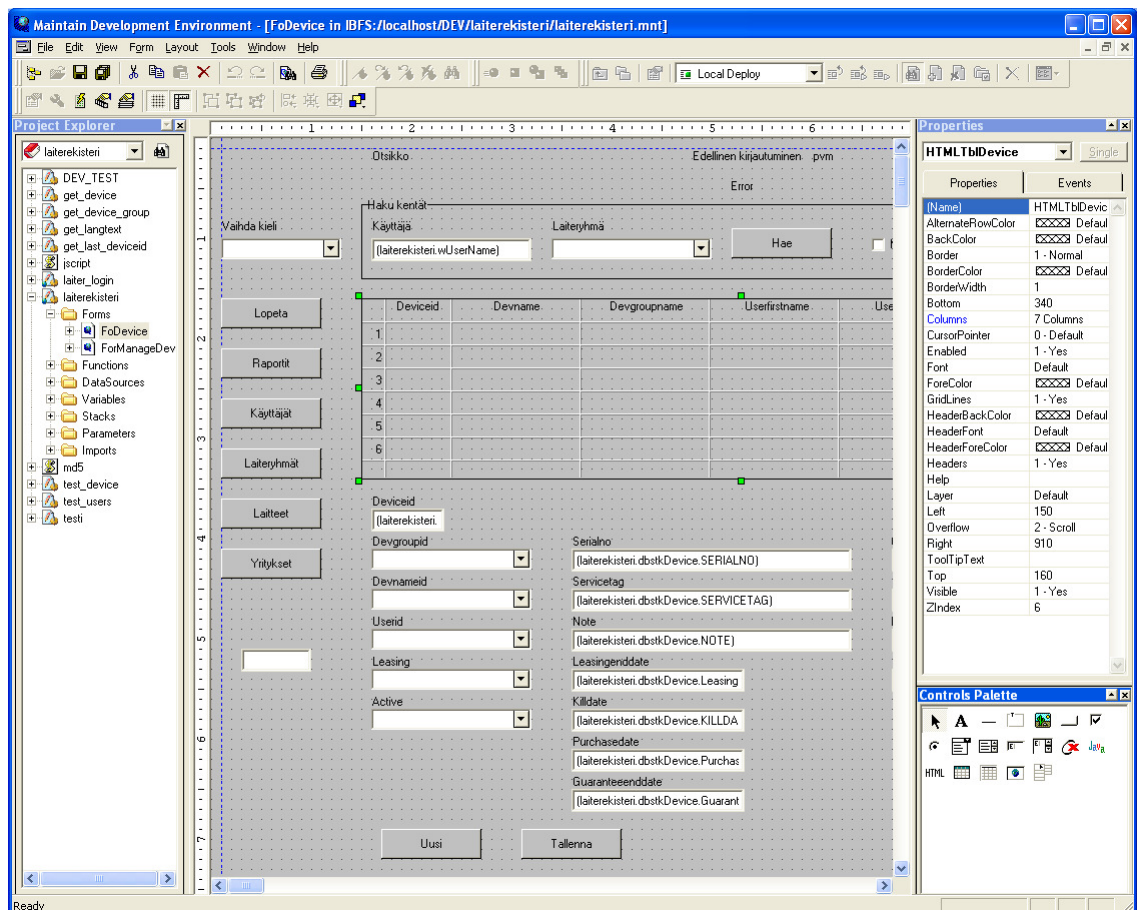


Kuva 1. WebFOCUS Developer Studion graafinen käyttöliittymä

3.2 Maintain Development Environment

WebFOCUS Maintain Environment (WF Maintain) on tietokantojen ylläpitosovellusten suunnitteluun kehitetty työväline [12, s. 1-2]. Se käyttää lähdekoodinaan aivan omaa koodikieltään, joka ei suoranaisesti muistuta mitään muuta kieltä, mutta on helpohko oppia, jos on ohjelmointitaustaa. Kieli on hyvin luettavaa, mikä tekee koodista ymmärrettävää. Ohjelmassa on ns. Language Wizard -toiminto, joka auttaa pääsemään ohjelman teossa alkuun. Myös kokeneemmat WF Maintain -ohjelmoijat käyttävät tätä wizard-ominaisuutta hyödykseen välttääkseen kirjoitusvirheitä ja nopeuttaakseen koodaamista.

Kuvassa 2 on WF Maintain -sovelluksen näyttöjen tekemisessä käytetty näkymä. Se on samantyyppinen kuin VisualBasic-ohjelmassa. Objekteja siirretään näytölle haluttuun kohtaan ja niitä voi siirrellä sekä niiden kokoa muuttaa hiirellä vetämällä tai venyttämällä. Objektien toiminnallisuus ohjelmoidaan pääasiassa työväliseen omalla kielellä, mutta myös JavaScript on yleisesti käytössä. JavaScriptiä hyödynnetään muun muassa tietojen oikeellisuuden tarkastuksiin sekä varoitus- ja vahvistusilmoitusten näyttämiseen. Yleensä pienet asiat pyritään tekemään JavaScript-kielellä, sillä Maintain-koodi suoritetaan aina palvelimella, mikä vaatii näytön uudelleen muodostamisen. Tällöin selaimessa oleva näkymä ladataan palvelimelta uudestaan, ja hitailla yhteyksillä käyttäjä saattaa kokea tämän ongelmalliseksi.



Kuva 2. WebFOCUS Maintain Environment graafinen käyttöliittymä

Tietojen vieni tietokantoihin on toteutettu hyvin yksinkertaisesti. Näytöllä olevat objektit on yleensä liitetty pinoihin (stack), ja pino on muodoltaan samanlainen kuin tietokannan taulu. Pinossa on samat sarakkeet kuin tietokannan taulussa, ja niiden tietotyypit ovat myös samat. Kun tiedot halutaan lisätä, päivittää tai poistaa, tietokannan

taulusta suoritetaan vain yksinkertainen komento, joka tekee kyseisen operaation pinon yhden tai useamman rivin perusteella.

Pinon arvojen lisääminen tietokantaan tapahtuu koodiesimerkissä 1 olevalla komenolla. Include-komennon perässä kerrotaan tietokantataulun perusavain (primary key), jonka perusteella sovellus osaa kohdistaa operaation oikeaan tietokantatauluun. From-komennon jälkeen kerrotaan pinon nimi, jossa tietokantaan vietävä data sijaitsee. Pinon nimen perässä voi olla sulkumerkkien sisällä numero, joka kertoo, monenneltako pinon riviltä operaatio aloitetaan. Ilman sulkumerkkejä ja numeroa operaatio aloitetaan ensimmäiseltä riviltä. Ennen include-komentoa voi olla sanat ”For all”, joilla tarkoitetaan kaikkien aloitusrivin jälkeisten rivien viemistä tietokantaan. Ilman näitä sanoja komento kohdistuu vain aloitusriviin.

Koodiesimerkki 1. Include-komento

```
For all include device.DEVICE.DEVICEID from dbstkDevice(2);
```

Koodiesimerkissä 2 näytetään, miten tietojen päivittäminen tietokantaan tapahtuu Update-komenolla, joka on hyvin samanlainen kuin Include, mutta siinä pitää luetella perusavaimen lisäksi kaikki taulun kentät, joita halutaan päivittää.

Koodiesimerkki 2. Update-komento

```
For all update device.DEVICE.DEVGROUPID device.DEVICE.DEVNAMEID
device.DEVICE.USERID device.DEVICE.SERIALNO device.DEVICE.SERVICETAG
device.DEVICE.LEASING device.DEVICE.PURCHASEDATE
device.DEVICE.GUARANTEEEENDDATE device.DEVICE.LEASINGENDDATE
device.DEVICE.ACTIVE device.DEVICE.KILLDATE device.DEVICE.NOTE
device.DEVICE.CREATEDATE device.DEVICE.CREATEUSER
device.DEVICE.MODIFYDATE device.DEVICE.MODIFYUSER from dbstkDevice(2);
```

Tietojen poistoon tarkoitettu delete-komento, koodiesimerkki 3, on samanmuotoinen kuin include-komento.

Koodiesimerkki 3. Delete-komento

```
For all delete device.DEVICE.DEVICEID from dbstkDevice(2) ;
```

Kaikki tietokantaa päivittävät komennot tekevät lopulliset tietokantamuutokset vasta, kun ne on vahvistettu commit-komennolla tai sovellus suljetaan. Sitä ennen tietokantamuutoksen pystyy kumoamaan rollback-komennolla. Koodiesimerkissä 4 on tyypillinen

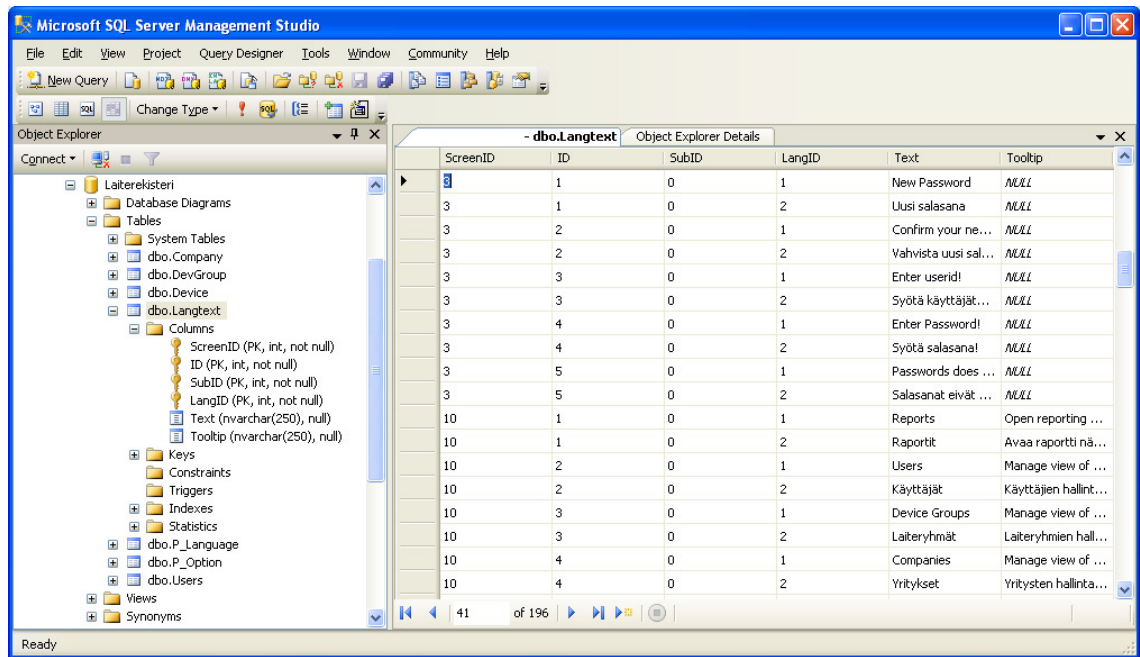
tapa testata tietokantaa päivittävän komennon onnistumista ja tehdä tilanteen mukaan vahvistus tai kumoaminen. WF Maintain tallentaa FocError-nimiseen sisäiseen muuttujaan tiedon päivityskomennon onnistumisesta. Jos muuttujassa on luku 0, päivitys on mennyt oikein ja se voidaan vahvistaa. Muuten kumotaan päivitys ja annetaan sovelluksen käyttäjälle virheilmoitus.

Koodiesimerkki 4. Tietojen vahvistus tietokantaan

```
if FocError eq 0 then begin
    Commit;
endbegin
else begin
    Rollback;
    FuError(1);
endbegin
```

3.3 Microsoft SQL Server Management Studio

Tietokannan rakentaminen ja testiaineiston lisääminen tietokantaan tehtiin Microsoft SQL Server Management Studion avulla. Ohjelma on suhteellisen helppokäyttöinen, mutta monipuolinen. Sen graafinen käyttöliittymä, joka näkyy kuvassa 3, nopeuttaa ja helpottaa tietokannan taulujen tuontia ja niiden muokkaamista verrattuna SQL-komentojen käyttämiseen. Tietojen syöttäminen tietokannan tauluihin on helppoa: taulun sisältö on näkyvässä taulukkomuodossa, jossa voi hiirellä valita muokattavan sarakkeen ja rivin, minkä jälkeen kirjoitetaan soluun haluttu teksti tai numeroarvo. Suora SQL-lause on nopeampi suorittaa, mutta sen kirjoittaminen kestää oman aikansa, ja siihen saattaa helposti jäädä jokin pieni syntaksivirhe, jonka huomaamiseen menee helposti aikaa. Lisäksi SQL-lauseeseen saattaa jäädä virhe, joka tekee tietokantaan ei-toivotun päivityksen. Tämän tyyppisten virheiden etsiminen ja korjaaminen on aikaa vievää käsityötä.



Kuva 3. Microsoft SQL Server Management Studion käyttöliittymä

4 Tietokanta

4.1 Relaatietietokannan perusteita

Nykyään sanalla tietokanta tarkoitetaan lähes yksinomaan SQL-pohjaista relaatiotietokantaa. Aikaisemmin käytettiin myös verkkomallisia sekä hierarkkisia tietokantoja. Relaatietietokannan vahvuuksia ovat helppokäyttöisyys, muokattavuus, tietojen eheys, tietoturva ja parantuneet varmistus- ja palautusominaisuudet. Nykyisin myös suorituskyky on hyvä, ja relaatiotietokannoilla toteutetaan kaikkein vaativimmatkin järjestelmät. [9, s. 3; 13, s. 5.]

Relaatietietokannan peruselementti on taulu, joihin tieto tallennetaan. Taulu voi olla esimerkiksi henkilötietoja sisältävä, jolloin sen nimeksi voisi antaa HENKILO. Taulussa on sarakkeita ja rivejä. Sarakkeet on nimetty tarpeiden mukaan. Henkilötaulun sarakkeita olisivat esimerkiksi henkilönnumero, etunimi, sukunimi, syntymäaika ja yritysnúmero. Taulun sarakkeiden nimissä on tapana välttää erikoiskirjaimia kuten ä- ja ö-kirjaimia, ja nimiä lyhennetään, mutta ne kannattaa pitää havainnollisina. Henkilönúmero-sarake voisi olla nimeltään hlonro tai hnro. Rivillä on aina yhden henkilön tiedot ja uusi rivi syntyy, kun tauluun lisätään uusi henkilö.

Relaatietietokannat eivät vaadi perusavainta, mutta sen käyttö on lähes välttämätöntä ja sen jättäminen pois taulusta on aina erikoistapaus. Perusavain yksilöi rivin, ja näin ollen tietokantaan ei voi tallentua samaa riviä kahteen kertaan. Perusavain on siis aina uniikki. Usein perusavain on taulun ensimmäinen sarake, mutta se voi muodostua useammastakin sarakkeesta. Perusavainten lisäksi taulussa voi olla viiteavaimia, jotka viittaavat toisten taulujen perusavaimiin, ja näiden syntyneiden linkkien avulla tietoja voidaan yhdistää taulujen välillä. Esimerkiksi henkilö-taulussa on sarake ”yritysnúmero”, joka on viiteavain. Se viittaa YRITYS nimisen taulun perusavaimiin. Tämän viiteavain-perusavain -linkin avulla henkilöt liitetään yrityksiin ja saadaan listattua henkilöt yrityksittäin. [13, s. 8–9; 14, s. 26–29.]

4.2 Suunnittelu

4.2.1 Kielitaulu

Tietokannan suunnittelussa lähdettiin liikkeelle miettimällä kielitaulun (Langtext) rakennetta. Koska sovelluksesta rakennettiin monikielistä, oli järkevää käyttää yhtä kielitaulua, joka sisältäisi kaikki sovelluksen tekstit. Tällä ratkaisulla vähennettiin myöhempää sovelluksen koodin muokkaamisen tarvetta, sillä tekstin muuttaminen vaatii vain tietokannassa olevien tekstien päivittämistä. Lisäksi mahdollinen uuden kielen lisäys tapahtuu lisäämällä uuden kielen tekstit tietokannan tauluun.

Kielitaulun rakennettaessa oli selvää, että siinä pitäisi olla kentät tunnistenumeroille (ID), kielikoodille (LangID) ja tekstille (Text). Tarkempi pohdiskelu nosti esille kysymyksen: onko yksi tunnistenumero riittävän selkeä tapa yksilöidä tekstit, kun tauluun tulee kuitenkin tuhansia rivejä? Pelkän tunnistenumeron perusteella on haastavaa löytää oikea rivi taulusta. Lopulta taulun rakenteesta tulikin kuvan 4 mallinen.

Langtext		
ScreenID	int	PK
ID	int	PK
SubID	int	PK
LangID	int	PK
Text	nvarchar(250)	
Tooltip	nvarchar(250)	

Kuva 4. Kielitaulun rakenne

Nyt taulusta saa helposti tulostettua tiettyyn sovelluksen näyttöön liittyvät tekstit, kun jokaisella näytöllä on oma tunnistenumerosa (ScreenID). Tästä on hyötyä sekä sovelluksen toiminnassa että jatkokehittämisessä. Tarvittavat tekstit on helppo hakea, kun sovelluksessa siirrytään näytöltä toiselle. Sovellusta kehittäessä tulee todennäköisesti eteen tilanteita, jossa näytölle tarvitaan uusia elementtejä, jotka tarvitsevat kuvaavia nimiä. Ilman ScreenID-saraketta yhden näytön tekstit eivät saisi peräkkäisiä tunnistenumeroita: uusien tekstien tunnistenumerot olisivat taulussa suurimmilla numeroilla, alkuperäisten tekstien tunnistenumerot olisivat pienemmillä numeroilla ja

välissä olisi jotain muita tekstejä, joilla ei ole mitään tekemistä kyseisen näytön kanssa. Tämä hidastaisi sovelluksen toimintaa ja häiritäisi sovelluksen kehitystä.

Tauluun lisättiin myös SubID-sarake, joka koettiin hyödylliseksi taulukkomuotoisten elementtien sarakeotsikoiden teksteille. Näin taulukolla on vain yksi ID-numero ja sen eri sarakkeet yksilöidään SubID-sarakkeella. SubID-sarake koettiin tarpeelliseksi samoin perustein kuin ScreenID-sarake. Näin näytöllä olevan taulukon tekstit pysyvät jatkossakin tietokannassa peräkkäin, vaikka taulukkoon tulisi uusi sarake. Tauluun lisättiin myöhemmin Tooltip-sarake, kun huomattiin, että moni teksti kaipasi myös pidempää selitettä. Se tulee näkyviin sovelluksessa, kun vie hiiren kohdistimen kyseisen elementin päälle.

Relaatiotietokantatauluissa käytetään yleisesti perusavainta, joka yksilöi taulun rivin. Kielitaulun kohdalla tämä tarkoitti sitä, että perusavain koostuu neljästä sarakkeesta (ScreenID, ID, SubID ja LangID). Koska kaikki perusavaimen sarakkeet ovat luonteeltaan numeerista tietoa, mutta ei kuitenkaan desimaaleja sisältäviä, niiden tietotyyppiä tuli kokonaisluku (int).

Muut taulussa olevat sarakkeet Text ja Tooltip ovat puolestaan tekstiä sisältäviä, joten niille piti valita sopiva tekstitietotyyppi. Kun lähtökohtana oli monikielinen sovellus, päätettiin valita sellainen tyyppi, johon pystyy tallentamaan tekstin Unicode-merkistön merkkejä käyttäen. Tällä ratkaisulla vältyttäisiin merkistöongelmilta tulevaisuudessa, jos ohjelmaan liitetään kieliä, joiden merkkejä ei ole vanhemmissa suppeammassa merkistöissä. Valittavaksi jäi siten nchar ja nvarchar, joista oli helposti valittavissa nvarchar, sillä tietokantaan vietävät tekstit olisivat hyvin erimittaisia ja sarakkeen maksimipituudeksi valittiin tekstien monipuolisen käyttötarkoituksen takia 250 merkkiä pitkäksi. Tietotyyppi nvarchar ei varaa ylimääräistä tilaa, vaan sarakkeeseen tallentuu vain syötetty teksti ja tieto tekstin pituudesta. Tietotyyppi nchar puolestaan tallentaa aina sarakkeen täyteen, joten lyhyen tekstin perään tallennetaan välilyöntejä. [13, s.112.]

4.2.2 Laitetaulu

Seuraavaksi hahmoteltiin sovelluksella syötettävälle laitetiedolle sopivaa päätaulua, johon pitäisi voida tallentaa laitteista monenlaista tietoa. Tämän taulun perusavaimen tekeminen oli helppoa. Luotiin surrogaattiavain, joka on automaattisesti kasvava numerosarja. Joka kerta, kun uusi laite lisätään järjestelmään, se saa yhtä suuremman lukuarvon kuin ennen lisäystä taulussa oleva suurin avainarvo. [13, s.115.]

Laitteella pitää olla nimi, joten sitä varten tarvitaan sarake, mutta nimestä ei tehtykään tekstimuotoista tietotyyppiä vaan kokonaisluku. Tämä sarake toimii viiteavaimena kielitauluun, jonne myös laitteiden nimet laitettiin. Syitä tähän ratkaisuun on useita. Laitteiden nimet saattavat olla erilaiset eri kielillä, ja tällä tavalla sovelluksen käyttäjä näkee laitteiden nimet omalla kielellään. Toinen syy tähän ratkaisuun oli nimien vakiointi. Yrityksissä on yleensä useampia samannimisiä ja -mallisia laitteita, kuten tietty tietokonemalli. Raportoinnin kannalta olisi hankalaa, jos saman tietokonemallin eri yksilöille olisi annettu nimiä vahingossa eri kirjoitusasuissa. Kolmas syy nimen sijaitsemiseen kielitaulussa on välttää turhaa tiedon monistamista. Kielitaulussa kaikkien samannimisten laitteiden nimet ovat vain kertaalleen tallennettuna, kun laitetaulussa ne olisivat jokaisen laitteen rivillä.

Laitteita ajatellaan usein ryhminä, kuten tietokoneet, puhelimet ja tulostimet. Tämän vuoksi tauluun lisättiin laiteryhmälle oma tieto, joka on toteutettu samalla tavalla kuin laitteen nimi. Laiteryhmästä on laitteen tiedoissa vain kokonaisluku, joka viittaa kielitauluun.

Muista sarakkeista myös henkilöön liittyvät tiedot on toteutettu viiteavaimella henkilötäuluun. Näitä sarakkeita on kolme kappaletta, joista laitteen käyttäjä on ilmeisin. Lisäksi laitetaulussa on laitteen perustaja sekä viimeisen tallentajan tieto.

Loput sarakkeet sisältävät tarkempaa tietoa laitteesta, joten ne sijaitsevat suoraan laitetaulussa rivitietona. Tekstitietotyyppiä käytetään sarjanumero-, huoltokoodi- ja huomautus-sarakkeissa. Taulussa on kaksi saraketta, jotka sisältävät vain kyllä- tai ei-

tiedon. Sarakkeiden tietotyypinä on yhden merkin mittainen char, ja niihin tallennetaan vain Y- tai N-kirjain. Sarakkeet sisältävät tiedon, onko laite vuokralaite sekä onko laite vielä käytössä. Lisäksi taulussa on kuusi päivämäärä-saraketta, jotka ovat hankintapäivä, takuun päättymispäivä, vuokran päättymispäivä, käytöstä poistumispäivä, perustamispäivä ja muutospäivä.

4.2.3 Henkilötaulu

Tietokantaan tehtiin myös henkilötaulu, joka sisältää perustiedot henkilöistä, jotka voivat käyttää ohjelmaa tai joiden hallussa on yrityksen laitteita. Taulua käytetään myös sovelluksen käyttöoikeustauluna, joten siellä on myös käyttäjätunnus ja salasana. Salasana-saraketta ei välttämättä tarvitse käyttää, jos sovellus käyttää Single Sign-on (SSO) -ominaisuutta. Jos salasana-saraketta käytetään, siihen tallennetaan salasanasta laskettu MD5-kooste, joten salasana-sarakkeen pitää olla riittävän pitkä, jotta 32 merkkiä pitkä tiiviste siihen mahtuisi. Henkilötaulun perusavaimena käytettiin surrogaattivainta samaan tyyliin kuin laitetaulussa. Henkilötaulussa on myös yksi viiteavain, joka viittaa yritystauluun.

4.2.4 Yritystaulu

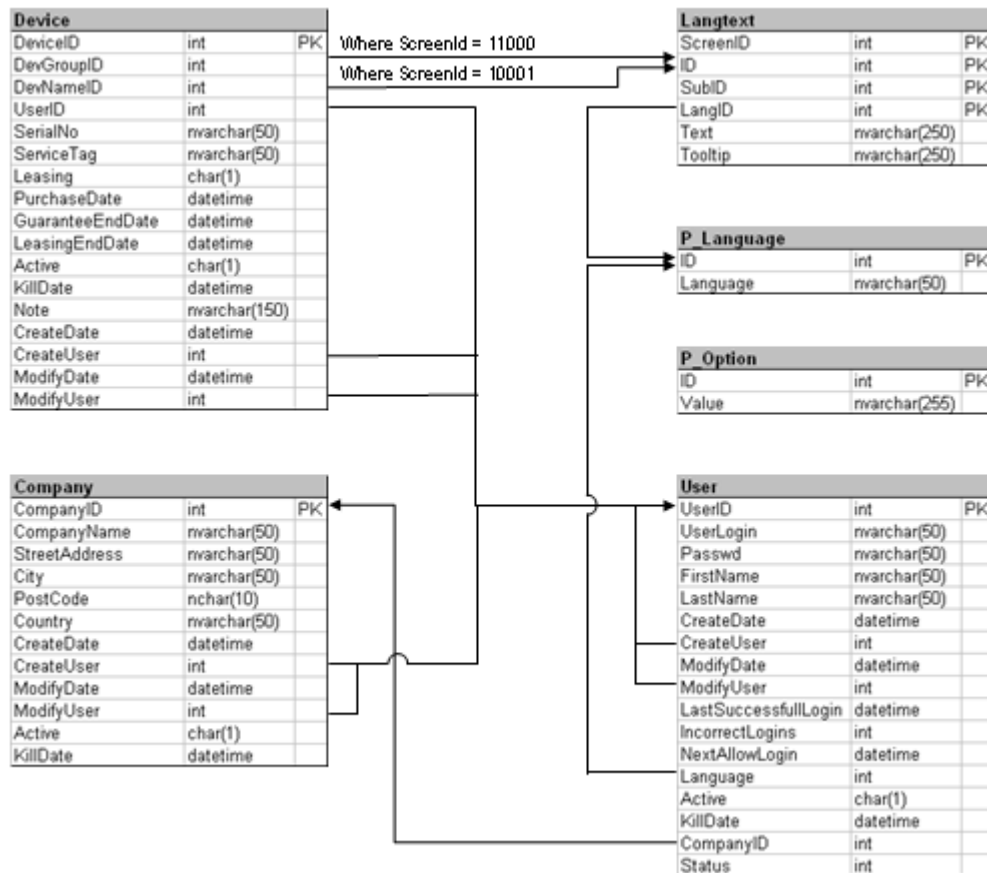
Yritystaulussa on perustiedot yrityksestä, johon henkilö kuuluu. Yritystiedon avulla sovellukseen voi lisätä useamman yrityksen laitteita sekoittamatta niitä keskenään. Yritystaulu katsottiin tarpeelliseksi, sillä sovellus otetaan tulevaisuudessa käyttöön koko United Software Distributors -konsernissa. Jos sovellusta käytettäisiin isommassa yrityksessä, yritystaulua voisi käyttää toimipisteen tai osaston käyttöön, mutta silloin ratkaisu ei enää sovi konserniratkaisuksi. Tällöin sovellukseen pitäisi lisätä uusi osastotaulu henkilö- ja yritystaulun väliin.

4.2.5 Ohjaustaulut

Tietokannassa on vielä kaksi taulua, joita käytetään vain sovelluksen ohjaukseen. Toisessa taulussa on listattuna sovelluksen käyttämät kielet. Toisessa puolestaan on tieto ohjelman oletuskielestä sekä ohjelman käyttämästä kirjautumistavasta. Näiden taulujen avulla sovelluksen toimintaa voi muuttaa ilman ohjelman koodin muuttamista.

4.2.6 Taulujen suhteet

Sovelluksen relaatiotietokantataulujen relaatiot näkyvät kuvassa 6. Vaikka tauluja sovelluksessa ei ole kuin kuusi, relaatioita on monia. Suurin osa näistä kohdistuu käyttäjätauluun, sillä kolmessa taulussa on sarake sekä rivin luojalle että sen päivittäjälle. Laitteen ja laiteryhmän nimien sijainti kielitaulussa aiheutti epätavallisen linkkauksen, sillä niiden id-numero viittaa vain osaan kielitaulun perusavaimesta. Päätettiin, että laitteiden nimet ovat kielitaulun sarakkeessa ScreenID numerolla 10001 ja laiteryhmien nimet numerolla 11000. Käyttäjätaulusta on luonnollisesti linkki yritystauluun.



Kuva 5. Tietokannan relaatiot

4.3 Toteutus

4.3.1 Taulujen luonti ja muokkaus

Tietokannan luonnissa sekä tietojen syöttämisessä tietokantaan käytettiin yksinomaan Microsoft SQL Server Management Studio -työkalua. Taulun luonti tapahtuu yksinkertaisesti painamalla Table-kansion kohdalla hiiren kakkospainiketta ja valitsemalla komento ”New Table”. Tämän jälkeen aukeaa taulukko, johon kirjoitetaan tauluun määritellyt kentän nimet ja valitaan pudotusvalikosta oikeat tietotyypit. Lisäksi valitaan, sallitaanko kentän arvoksi null, joka tarkoittaa tyhjää arvoa.

Olemassa olevaa tietokantataulua tarvitsee ennen pitkää muokata. Sovellusta kehitettäessä saatetaan huomata uuden kentän tarve, tai jonkun kentän nimeä tai tietotyyppiä pitäisi muuttaa. Tämä tapahtuu valitsemalla muutoksen kohteena oleva taulu ja

poimimalla hiiren kakkospainikkeella valikosta komento ”Design”, jolloin aukeaa kuvassa 5 oleva näkymä. Se on samanlainen kuin uuden taulun luontinäkymä, mutta kentät on tietenkin valmiiksi määritelty.

Kentän nimen muutostarve tuli sovellusta tehdessä esiin, kun huomattiin, ettei kentän käyttötarkoitus vastannutkaan täysin suunniteltua. Nimen muutos tapahtuu yksinkertaisesti valitsemalla oikea sarake ja muokkaamalla kentän nimeä. Tämä tapa on tuttu kaikille, jotka ovat muuttaneet joskus Windows-käyttöjärjestelmässä tiedoston nimeä.

	Column Name	Data Type	Allow Nulls
▶ 🔑	ScreenID	int	<input type="checkbox"/>
🔑	ID	int	<input type="checkbox"/>
🔑	SubID	int	<input type="checkbox"/>
🔑	LangID	int	<input type="checkbox"/>
	Text	nvarchar(250)	<input checked="" type="checkbox"/>
	Tooltip	nvarchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Column Properties	
(General)	
(Name)	ScreenID
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
Indexable	Yes
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4

Kuva 6. Taulun muokkaaminen Microsoft SQL Server Management Studio -työkalulla

Tietotyypin muutos tuli ajankohtaiseksi, kun taulua luotaessa oli tehty virhe tietotyypin valinnassa, ja se piti myöhemmin korjata. Operaatio on tässäkin tapauksessa hyvin yksinkertainen. Valitaan pudotusvalikosta uusi tietotyyppi char(1) vanhan int-tietotyypin tilalle. Vaikka taulussa on muutoshetkellä tietoja, se ei haittaa tietotyypin muutosta: kaikki tieto pysyy taulussa ja se muutetaan vastaamaan uutta tietotyyppiä automaattisesti. Jos muutos olisi pitänyt tehdä toisinpäin merkkimuotoisesta tietotyypistä numeeriseen tietotyyppiin, pitäisi ensin varmistaa, että kentässä on arvoina vain numeroita.

Kielitauluun lisätään taulun luonnin jälkeen Tooltip-sarake. Lisäys tapahtuu työkalun Desig-tilassa valitsemalla ensimmäinen tyhjä rivi taulun taulukkomuotoisen näkymän lopusta ja kirjoittamalla kentälle nimi ja antamalla tiedostotyyppi. Uuden kentän voisi tehdä muuallekin kuin taulun loppuun, lisäämällä ensin tyhjä rivi taulukossa oikeaan väliin, mutta tässä tapauksessa paikka viimeisenä on looginen.

4.3.2 Tietojen vienti tietokantaan

Sovellusta kehitettäessä lisätään tietokannan tauluihin usein tietoa. Varsinkin kielitaulun sisällöin muokkaaminen on yleistä. Kun sovellukseen lisätään uusi objekti, siihen liittyvä teksti viedään samalla kielitauluun, jotta sovellusta testattaessa voidaan heti varmistaa tekstien hakurutiinin toimivuus.

Tietojen vienti tehdään avaamalla halutun taulun sisältö Microsoft SQL Server Management Studio -työkalulla, jolloin saadaan kuvassa 7 näkyvä taulukko, johon tulostuvat kaikki taulun sarakkeet ja rivit. Taulukon lopussa on tyhjä rivi, johon voi lisätä uuden rivin arvot kirjoittamalla suoraan sarakkeisiin halutut arvot. Normaalisti SQL-koodissa pitää merkkimuotoisen sarakkeen arvo olla puolilainausmerkkien sisällä, mutta tähän taulukkoon puolilainausmerkkejä ei tarvitse kirjoittaa. Kun tietokannan tauluun tulee paljon rivejä, taulun aukaiseminen kestää jonkin aikaa. Tällöin saattaa olla nopeampaa kirjoittaa SQL-koodin insert-lause, mutta reilun tuhannen rivin taulussa ei ole vielä havaittu oleellista hidastumista.

ScreenID	ID	SubID	LangID	Text	Tooltip
1	1	0	1	Wrong user id or...	
1	1	0	2	Väärä tunnus tai...	
1	2	0	1	Login	Login to the pro...
1	2	0	2	Kirjaudu	Nappia painamall...
1	4	0	1	UserId	
1	4	0	2	Käyttäjätunnus	
1	5	0	1	Password	
1	5	0	2	Salasana	
1	6	0	1	Enter userid!	
1	6	0	2	Syötä käyttäjät...	
1	7	0	1	Enter password!	
1	7	0	2	Syötä salasana!	
1	8	0	1	Change Password	Click to change ...
1	8	0	2	Vaihda salasana	Nappia painamall...

Kuva 7. Tietokantataulun sisällön näyttäminen Microsoft SQL Server Management Studio työkalussa

5 Sovelluksen toteutus

5.1 Kirjautuminen

Sovellukseen suunniteltiin kaksi erilaista tapaa kirjautua. Toisessa hyödynnetään Windowsin Active Directory (AD) -ominaisuutta ja toisessa on käyttäjätunnus-salasana-pari. Sovellukseen pystyy kirjautumaan vain jommallakummalla tavalla, ja se kerrotaan sovelluksen P_Option taulussa, joka sisältää erilaisia sovelluksen ohjausparametreja.

Windows Active Directory (AD) -ominaisuus vaatii, että yrityksellä on Windows Domain Controller, jota käytetään kirjautumispalvelimena. Kun tietokoneelle on kirjaututtu yrityksen verkkotunnuksella, osaa Internet Explorer välittää tunnuksen IIS-webpalvelimelle tämän sitä pyytäessä. IIS käy AD:sta tarkistamassa, että käyttäjällä on oikeus käyttää webpalvelinta. Sovelluksen vastuulle jää vain tarkistaa henkilötaulusta, onko kyseisellä tunnuksella oikeutta tähän ohjelmaan. Näin käyttäjän ei tarvitse kuin kirjautua koneelleen, ja tämän jälkeen hän pääsee ilman erillisiä kirjautumisia sovellukseen. Muut selaimet kuin Internet Explorer eivät pysty koneelle kirjautuneen henkilön käyttäjätunnusta välittämään automaattisesti IIS:lle, joten ne kysyvät sovellusta käynnistäessä tunnuksen ja salasanan, joka sitten kerrotaan eteenpäin IIS:lle.

Käyttäjätunnus-salasana-parilla kirjaututtaessa on otettu tietoturva huomioon salasanan salauksella. Tietokannassa ei näy tunnusten salasanvoja selväkielisenä vaan niistä lasketaan selaimessa JavaScript-funktiolla toteutetulla MD5-algoritmilla kooste heti, kun käyttäjä on salasanan kirjoittanut. Koosteen laskemisen jälkeen salasanakenttä tyhjennetään ja kooste lähetetään palvelimelle, joka käy tarkistamassa, onko se sama kuin tietokannassa oleva salasanan kooste. Näin salasana ei liiku tietoverkoissa selväkielisenä missään vaiheessa.

5.2 Kielenvalinta

Kielenvalinta on käyttäjäkohtainen, ja sitä pystyy muuttamaan milloin tahansa. Muutos tallentuu käyttäjän tietoihin ja on voimassa siihen asti, kun käyttäjä seuraavan kerran muuttaa kieltä.

Ohjelmassa on määriteltävä myös ohjelman oletuskieli, joka käytännössä määrittelee vain sen, millä kielellä kirjautumissivu näytetään, jos ohjelmassa käytetään erillistä kirjautumista eikä Single Sign-on (SSO) -ominaisuutta. SSO:ta käytettäessä ohjelman oletuskieli tulee esiin vain, jos ohjelmaa yritetään käynnistää tunnuksilla, joilla ei ole oikeutta ohjelmaan. Tällöin tieto käyttöoikeuden puuttumisesta tulee oletuskielellä.

5.3 Laitehaku

Laitteiden haku käsiteltäväksi on tehty käyttäjälle mahdollisimman helpoksi. Kuten kuvassa 8 näkyy, valintakriteereitä on vain kaksi: laiteryhmä ja käyttäjä. Laiteryhmä on pudotusvalikko, jossa on listattuna yrityksen laiteryhmät. Käyttäjä on tekstikenttä, jonka perusteella haetaan nimen osalla ilman erillisiä jokerimerkkejä. Lisäksi haku kohdistuu sekä etu- että sukunimeen eikä ole merkkikokoriippuvainen.

Haku kentät

Käyttäjä Laiteryhmä Myös poistetut laitteet

	Laitenro	Laitenimi	Laiteryhmänimi	Käyttäjän etunimi	Käyttäjän sukunimi	Sarjanumero
1	1	Nokia E71	Matkapuhelin	Saku	Kirves	xxxxxxxxxxxx
2	4	Nokia E71	Matkapuhelin	Admin	User	xxxxxxxxxxxx
3	10	Nokia E71	Matkapuhelin	Admin	User	dgjhsfjggfj4

Kuva 8. Sovelluksen laitehakunäkymä

Haku hakee vain aktiivikäytössä olevat laitteet, mutta yhden valintaruudun valitsemalla saa haettua myös poistetut laitteet.

Hakutulokset näytetään hakukenttien alapuolella pienessä taulukossa, josta linkkiä painamalla saadaan halutun laitteen tiedot näytölle muokattavaksi.

Haku on toteutettu normaalista tavasta poiketen pienen mutkan kautta. Normaalisti Maintain-kielen haku toimii erinomaisesti, mutta jos samalla pitää hakea tietoja useammasta taulusta tai hakuehto on poikkeava normaalista yhtä suuri -vertailusta, on haku helpompi toteuttaa seuraavasti. Maintain koodissa suoritetaan koodiesimerkin 5 exec-komento, jolla suoritetaan WebFOCUS-raportti. Raportilla tehdään normaalit tietokantahaut ja tulostusmuodoksi laitetaan XML, jonka Maintain-ohjelma osaa ottaa vastaan ja käyttää tulosta jatkossa aivan samoin kuin tulos olisi Mantain-kielen oman haun suorittama. Näin raportille on huomattavasti helpompi tehdä monipuolisia hakuja ja taulujen yhdistelyä kuin Maintain-puolella.

Koodiesimerkki 5. Maintain-kielen exec-komento

```
Exec get_device From wUserlang  
stkDevGroupSearch(stkDevGroupSearch.FocIndex).DevGroupId wUserName1  
wUserName2 wRemovedDevice Into stkDevice;
```

Komennossa on kerrottu, että suoritettava raportti on nimeltään get_device, jonka lähdekoodi on koodiesimerkissä 6. From-sanan jälkeen on lueteltu raportille välitettävät parametrit ja Into-sanan jälkeen on kerrottu pinon nimi, johon raportti palauttaa tiedot.

Koodiesimerkki 6. Maintain-koodista kutsuttu WebFOCUS-raportti

```

-* Asetetaan muuttujille oletusarvot
-DEFAULT &1=1
-DEFAULT &2=1
-DEFAULT &3=''
-DEFAULT &4=''
-DEFAULT &5=''

-* Alustetaan nimi muuttujat tyhjiksi
-SET &NIMI1 = '';
-SET &NIMI2 = '';

-* Jos käyttäjä parametri on tyhjä ei tehdä hakuehtoja käyttäjälle
-SET &NIMI1 = IF &3 EQ '' THEN 'x' ELSE &3;
-SET &NIMI2 = &4;

JOIN CLEAR *
JOIN DEVICE.USERID IN DEVICE TO
      USERS.USERID IN USERS AS J1
END
JOIN DEVICE.DEVGROUPID IN DEVICE TO
      LANGTEXT.ID IN LANGTEXT TAG DGROUP AS J3
END
JOIN DEVICE.DEVNAMEID IN DEVICE TO
      LANGTEXT.ID IN LANGTEXT TAG DNAME AS J4
END

TABLE FILE DEVICE
PRINT
      DNAME.TEXT AS 'DevName'
      DGROUP.TEXT AS 'DevGroupName'
      USERS.FIRSTNAME AS 'UserFirstName'
      USERS.LASTNAME AS 'UserLastName'
      DEVICE.SERIALNO
      DEVICE.SERVICETAG
      DEVICE.LEASING
      DEVICE.PURCHASEDATE
      DEVICE.GUARANTEEEENDDATE
      DEVICE.LEASINGENDDATE
      DEVICE.ACTIVE
      DEVICE.KILLDATE
      DEVICE.NOTE
BY DEVICEID
-IF &2 EQ 0 THEN GOTO Y_DEVGROUP
WHERE DEVICE.DEVGROUPID EQ &2;
-Y_DEVGROUP
-IF &4 EQ '' THEN GOTO Y_NIMIHAKU
WHERE UPCASE(50, USERS.FIRSTNAME, 'A50V') CONTAINS '&NIMI1' OR
      UPCASE(50, USERS.FIRSTNAME, 'A50V') CONTAINS '&NIMI2' OR
      UPCASE(50, USERS.LASTNAME, 'A50V') CONTAINS '&NIMI2';
-Y_NIMIHAKU
-IF &5 EQ 1 THEN GOTO Y_POISTETUT
WHERE DEVICE.ACTIVE EQ '1';
-Y_POISTETUT
WHERE DNAME.SCREENID EQ 11000 AND DNAME.LANGID EQ &1 ;
WHERE DGROUP.SCREENID EQ 10001 AND DGROUP.LANGID EQ &1 ;
ON TABLE SET ASNAMES ON
ON TABLE PCHOLD
END

```

5.4 Lisäys

Uuden laitteen lisäyksessä painetaan ensin painiketta Uusi, joka alustaa täytettävät kentät. Samassa yhteydessä sovellus käy hakemassa laitteelle yksilöllisen laitenumeron, joka on aina yhtä suurempi kuin viimeksi syötetyn laitteen numero. Osa kentistä on harmaana ja niiden tiedot ohjelma täyttää itse, kuten laitenumero ja laitteen luontipäivämäärä. Käyttäjä valitsee muutamasta pudotusvalikosta oikeat vaihtoehdot ja kirjoittaa syöttökenttiin tarvittavat tiedot, jonka jälkeen painetaan Tallenna-painiketta. Tällöin ohjelma laittaa ensin pinon Perustaja- ja Muuttaja-kenttiin käyttäjän tunnuksen sekä Perustuspäivä- ja Muutospäivä-kenttiin kyseisen hetken aikaleiman. Tämän jälkeen tiedot viedään pinosta tietokantaan.

5.5 Päivitys

Laitetietoja päivitettäessä haetaan ensin laitehaulla kyseisen laitteen perustiedot näytölle taulukkoon, josta linkkiä painamalla saadaan halutun laitteen tarkat tiedot kuvassa 9 näkyvälle näytölle muokattavaksi.

Laitenumero	1	Sarjanumero	xxxxxxxxxxxx	Luoja	Admin User
Laiteryhmä	Matkapuhelin	Huoltotunnus		Luontipäivä	2009/01/01 00:00:00.000
Laitenimi	Nokia E71	Huomautus		Muuttaja	Admin User
Käyttäjä	Saku Kirves	Leasingin päättymispäivä		Muutospäivä	2010/05/29 14:32:41.000
Leasing	Kyllä	Poistopäivä			
Aktiivinen	Kyllä	Hankintapäivä	2006/11/01		
		Takuun päättymispäivä	2008/11/01		

Buttons: Uusi, Tallenna

Kuva 9. Sovelluksen laitetiedot-näkymä

Tässä tilassa käyttäjä voi muokata tietoja, jotka eivät ole harmaina. Painamalla Tallenna-painiketta Muuttaja- sekä Muutospäivä-kentät päivittyvät automaattisesti. Laitenumeron, perustajan sekä perustuspäivän tiedot pysyvät luonnollisesti ennallaan.

5.6 Poisto

Varsinaista poistoa ohjelmassa ei ole, sillä käytöstä poistettujen laitteiden tietoja saatetaan tarvita myöhemmin, joten laitteet merkitään vain poistuneiksi. Tämä tapahtuu menemällä muokkaamaan laitetta ja valitsemalla Aktiivinen-pudotusvalikosta Ei ja painamalla Tallenna. Tällöin ohjelma laittaa poistopäiväksi automaattisesti kyseisen päivän.

5.7 Raportit

Raportointi on sovelluksen keskeinen toiminto, joka tulee kehittymään jatkossa käyttäjien tarpeiden mukaan. Ensimmäiseen versioon ohjelmasta toteutettiin kaksi tärkeintä tarvetta: lista tietyn laiteryhmän tai laitetyypin laitteista sekä lista tietyn henkilön hallussa olevista laitteista.

Sovellukseen lisättiin raportti-painike, joka aukaisee javascript-komennolla uuden selainikkunan, jonka sisältönä on raportoinnin parametrisivu. Siinä valitaan haluttu raportti, minkä jälkeen sivulle tulevat näkyviin kyseiseen raporttiin liittyvät hakuvalinnat, raportin tulostusmuodon valinta sekä suorita-painike.

Kun suorita-painiketta painetaan, käynnistyy varsinainen tietokantakysely ja raportin muodostus. Raportille tarvitaan tietoja useammasta taulusta ja WebFOCUS-kielessä taulujen yhdistely tehdään aina JOIN-komennolla, josta on malli koodiesimerkissä 7, ennen varsinaista tietokantahakukomentoa.

Koodiesimerkki 7. Tietokantataulujen yhdistelykomento JOIN

```
JOIN DEVICE.USERID IN DEVICE TO UNIQUE USERS.USERID IN USERS
```

Tietojen hakuun käytetty TABLE FILE -lauseke, koodiesimerkki 8, on WebFOCUS-kielessä todella monipuolinen, sillä siihen on liitetty raportin kaikki tulostuksen tyyliin liittyvät muotoilut, mutta perusmuodossa lauseke on yksinkertainen ja helposti ymmärrettävä.

Koodiesimerkki 8 Tietojen hakukomento

```
TABLE FILE DEVICE
PRINT
      'DEVICE.DEVICEID'
      'DEVICE.SERIALNO'
      'DEVICE.PURCHASEDATE'
BY 'USERS.LASTNAME'
BY 'USERS.FIRSTNAME'
WHERE 'DEVICE.DEVGROUPID' EQ &DEVGROUPID;
END
```

Haasteellisin osuus raportin ohjelmoinnissa on monesti muotoilu. Onneksi tässä tuotteessa eri tulostusmuodot saa helposti muodostettua. Kolme haluttua tulostusmuotoa, PDF, Excel ja HTML, vaativat ohjelmoinnissa ainoastaan koodiesimerkin 9 viimeisen sanan korvaamista muuttujalla, jonka sisältö asetetaan oikeaksi jo raportin parametrisivulla valittaessa pudotusvalikosta tulostusmuotoa.

Koodiesimerkki 9. Excel tulostusmuodon komento

```
ON TABLE PCHOLD FORMAT EXL2K
```

Myös raportilla hakuehtoina käytettävät laitteen tyyppi, laitteen nimi ja henkilön nimi välitetään muuttujina raportille, joka käyttää niitä hakuehdoissa. Tosin raportille ei välity esimerkiksi käyttäjän parametrinäytöllä näkemä laitteen nimi, vaan laitteen yksilöllinen numero.

Tietojen hakemisen jälkeen raportin muodostamisessa on jäljellä sen muotoilu, joka suoritetaan osittain CSS-määrittelyllä, jotta eri raporttien ulkoasu säilyy yhtenäisenä ja haluttu muutos saadaan helposti voimaan kaikille raporteille.

6 Yhteenveto

Tässä työssä toteutettiin InfoBuild Oy:n tarpeita vastaava laitteiden keskitetty hallintajärjestelmä. Sen avulla saatiin laitetiedot ajan tasalle, ja jatkossa ylläpito helpottuu. Hankkeen aikana opittiin käytetyistä ohjelmointityökaluista uusia piirteitä sekä käyttämään niitä tehokkaasti monikielisen sovelluksen kehityksessä.

Vaikka tässä työssä käytettiin pääasiassa ohjelmointityökaluja, joista tekijällä oli monen vuoden kokemus, työkalusta löytyi ennen kokemattomia ominaisuuksia. Suurimmat haasteet toivat sovelluksen monikielisyys ja sen rakentaminen mahdollisimman joustavaksi ja helpoksi ylläpitää sekä tulevien kielien lisäämisen helppous. Paljon ponnistelua vaati saada kaikki sovelluksen tekstit aseteltua näytölle tietokannasta luettuna, kun siihen ei ollut kaikissa tapauksissa selvää tapaa, eikä dokumentoitua ohjeistustakaan löytynyt. Tietyissä tapauksissa jouduttiin yrittää ja erehdy -menetelmällä hakemaan oikea tapa. Aidosti monikielinen sovellus oli yllättävän työläs toteuttaa. Jos vastaavan tyyppisestä työstä ei ole aikaisempaa kokemusta, voi helposti aliarvioida työhön menevän ajan.

Vaikka monikielisyyttä on sovelluksen suunnittelussa ja kehityksessä mietitty tarkasti, myös jatkokehityksen osalta monikielisyys nousee esiin. Sovelluksen ensimmäisessä versiossa kielitaulua pitää päivittää suoraan tietokannassa SQL-komennoin tai tietokantapalvelimen graafisen työkalun Microsoft SQL Server Management Studion avulla. Jatkossa sovellukseen voisi lisätä osion, jossa sovelluksen ylläpitäjällä olisi mahdollisuus päivittää kielten tekstejä sekä lisätä uusia kieliä. Tässä sovelluksen osassa voisi valita lähtökielen, mistä lähtisi tekstejä kääntämään uudelle kielelle.

Ensimmäisessä versiossa sovelluksesta pystyy selvittämään vain laitteen nykyisen käyttäjän, mutta mahdollista käyttäjähistoriaa ei ole mahdollista saada. Tämä voi olla tarpeellinen ominaisuus esimerkiksi tilanteessa, jossa pitää selvittää, onko käyttäjä palauttanut jonkun hallussaan olleen laitteen. Nyt, jos toinen käyttäjä on merkitty laitteen käyttäjäksi, sovelluksesta ei löydy tietoa, että laite on aikaisemmin ollut toisen käyttäjän hallussa ja hän on sen palauttanut.

Sovellukseen olisi hyödyllistä kehittää jatkossa myös muutoshistorian tallentuminen. Tällöin jokainen sovelluksessa tehty muutos, lisäys ja poisto tallentuisivat, ja mahdollisten ongelmatilanteiden selvittäminen olisi helpompaa. Tällöin pystyttäisiin tarkistamaan, kuka on tehnyt ja minkälaisia muutoksia sovelluksessa. Näiden perusteella pystyttäisiin helpommin selvittämään ongelman syytä.

Mielenkiintoinen jatkokehityskohde on sähköpostitoimintojen liittäminen sovellukseen. Tämä tosin vaatii uuden ohjelmiston asentamista palvelimelle. ReportCaster-niminen ohjelmisto kuuluu samaan tuoteperheeseen jo palvelimelle asennettujen WebFOCUS Clientin ja Reporting Serverin kanssa. Helposti toteutettavissa olisi muistutus laitevastaavan sähköpostiin, kun laitteen takuu- tai leasingaika olisi päättymässä. Suurempi kehityskohde olisi sähköpostin kautta toimiva sähköinen kuittaus. Tässä käyttäjä saisi sähköpostiinsa viestin, jossa kerrotaan hänen vastaanottamansa laitteen tiedot ja pyydetään varmistamaan laitteen vastaanotto valitsemalla viestissä oleva web-linkki tai vaihtoehtoisesti vastaamalla viestiin. Tämän jälkeen laitevastaava saisi tiedon sähköpostiinsa käyttäjän vahvistuksesta ja sovelluksessa laitteen tila muuttuisi.

Raportoinnin kehitys on todennäköisesti nopeaa, mikäli siihen halutaan panostaa. Työväline, jolla sovellus on tehty, antaa siihen hyvät puitteet, ja käyttäjiltä varmasti tulee ideoita uusista raporteista nopeasti, kun he ovat päässeet tutustumaan sovellukseen.

Lähteet

- 1 WebFOCUS and ReportCaster Installation and Configuration for Windows Version 7 Release 6.1 and Higher. New York: Information Builders, 2009.
- 2 Duckett, Jon. Beginning HTML, XHTML, CSS, and JavaScript. Indianapolis: Wiley, cop. 2010.
- 3 Introduction to HTML. (WWW-dokumentti.) W3Schools.
<http://www.w3schools.com/html/html_intro.asp>. Luettu 10.2010.
- 4 CSS. (WWW-dokumentti.) The World Wide Web Consortium (W3C).
<<http://www.w3.org/Style/CSS/>>. Luettu 10.2010.
- 5 Harsu, Maarit. Ohjelmointikielien periaatteet, käsitteet, valintaperusteet. Helsinki: Talentum, 2005.
- 6 Saarikumpu, Osmo. JavaScript-kielen alkeet - osa 1. (WWW-dokumentti.)
<<http://weppipakki.com/js/opas/alkeet1.htm>>. 24.3.2010. Luettu 19.10.2010.
- 7 Kline, Kevin E. Gould, Lee. Zanevsky, Andrew. Transact-SQL programming. Sebastopol: O'Reilly & Associates, Inc. 1999.
- 8 Fernando G. Guerrero. Carlos Eduardo Rojas. Microsoft SQL Server 2000 programming by example. United States of America: Que Publishing, 2001.
- 9 Gunderloy, Mike. Jordan, Joseph L. Tschanz David W. Mastering Microsoft SQL Server 2005. Indianapolis: Wiley Publishing, Inc. 2006.
- 10 The Apache Software Foundation. (WWW-dokumentti.)
<<http://tomcat.apache.org/>>. Luettu 8.11.2010.
- 11 WebFOCUS Developing Reporting Applications With Graphical Tools Version 7 Release 6. New York: Information Builders, 2009.
- 12 WebFOCUS Maintain Getting Started Version 7 Release 6. New York: Information Builders, 2009.
- 13 Hovi, Ari. Huotari, Jouni. Lähdesmäki, Tapio. Tietokantojen suunnittelu ja indeksointi. Jyväskylä: Docendo Finland Oy, 2005.
- 14 Connolly, Thomas M. Begg, Carolyn E. Database systems : a practical approach to design, implementation, and management. Boston: Addison-Wesley, 2005.