



Ydintietojen hallinta tietojärjestelmissä
Henkilötietojen hallinta Kemi-Tornionlaakson
koulutuskuntayhtymä Lappiassa

Jouni Koivumaa

Teknologiaosaamisen johtaminen opinnäytetyö
Kunnossapito
Insinööri (ylempi AMK)

KEMI 2013

ALKUSANAT

Tämä opinnäytetyö tehtiin Kemi-Tornionlaakson koulutuskuntayhtymä Lappialle. Kiitän tietohallintojohtaja Markku Koivumaata opinnäytetyön mahdollistamisesta. Kiitän työn ohjauksesta sekä kaikesta saamastani tuesta ja palautteesta yliopettaja Tapani Ruokasta. Kiitän myös tuntiopettaja Päivi Honkaa ja lehtori Martta Ruottua raportin kieliasun tarkastamisesta. Iso kiitos kuuluu vaimolleni Soilille kaikesta tuesta ja kannustamisesta koko opinnäytetyöprosessin ajalta.

Tervola 11.12.2013

Jouni Koivumaa

TIIVISTELMÄ

KEMI-TORNION AMMATTIKORKEAKOULU, Tekniikan koulutusala

Koulutusohjelma:	Teknologiaosaamisen johtamisen koulutusohjelma
Opinnäytetyön tekijä:	Jouni Koivumaa
Opinnäytetyön nimi:	Ydintietojen hallinta tietojärjestelmissä Henkilötietojen hallinta Kemi-Tornionlaakson koulutuskuntayhtymä Lappiassa
Sivuja (joista liitesivuja):	72 (13)
Päiväys:	11.12.2013
Opinnäytetyön ohjaaja:	DI Tapani Ruokanen
<p>Tämä opinnäytetyö tehtiin Kemi-Tornionlaakson koulutuskuntayhtymä Lappialle. Työn tarkoituksena oli selvittää, miten eri tietojärjestelmissä sijaitsevien ydintietojen hallinta saataisiin nykyistä tehokkaammaksi, laadukkaammaksi ja joustavammaksi prosesseja muuttamalla sekä tiedonsiirtoja automatisoimalla tietojärjestelmien välillä.</p> <p>Tutkimusosassa keskityttiin tiedonhallintaan sekä perehdyttiin tietokantoihin, jotka ovat yleisimpiä tiedontallennusratkaisuja. Tässä osassa pohdittiin erillisiin tietojärjestelmiin tallennettuihin tietoihin liittyviä ongelmia, niiden ajantasaisuutta ja oikeellisuutta.</p> <p>Esimerkkitapauksena käytettiin henkilötietojen hallintaa keskeisimmissä tietojärjestelmissä sekä tietojen synkronointia järjestelmien välillä. Työkaluna käytettiin Microsoft SQL Server – tuotetta ja erityisesti sen Data Tools - sekä Master Data Service -ominaisuuksia.</p> <p>Ydintietojen hallinta on kokonaisuudessaan hyvin laaja asia. Tässä opinnäytetyössä käsiteltiin aihetta enimmäkseen teknisestä näkökulmasta. Tekniikkaa tärkeämpää on käytännössä tiedostaa ja tunnistaa prosessit ja käytännöt. Oikein toteutettuna MDM-projekti koskee koko organisaatioita ja vaatii riittävät resurssit tämän mahdollistamiseksi. Pienestäkin voi lähteä varovasti liikkeelle. Tässäkin tapauksessa on muistettava ohjeistaa ja tiedottaa toiminnoista. Kaikilla tietoja käsittelevillä osapuolilla tulee olla riittävä ymmärrys siitä, minne kaikkialle tietojen tallennus ja muokkaaminen vaikuttaa.</p>	
Asiasanat: ydintieto, ydintietojen hallinta, tietokanta, tietovarasto.	

ABSTRACT

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES, Technology

Degree programme:	Technology Competence Management
Author:	Jouni Koivumaa
Thesis title:	Master Data Management Personal data management in Kemi-Tornionlaakso Municipal Education and Training Consortium Lappia
Pages (of which appendixes):	72 (13)
Date:	11 December 2013
Thesis instructor:	Tapani Ruokanen, M.Sc. (Tech.)
<p>This study was made for Kemi-Tornionlaakso Municipal Education and Training Consortium Lappia. The purpose of the study was to find out the means to intensify and improve the quality and flexibility in the master data management within different data systems by changing processes and automating data transfers between the systems.</p> <p>The theory part of the study focused on data management and databases which are generally used for data storage solutions. The problems with the data saved in separate data systems were discussed in this part.</p> <p>Personal data management and synchronization within common used data systems were used as an example. Microsoft SQL Server and its Data Tools, especially Master Data Service, were used in the test environment.</p> <p>Master data management is a very broad topic as a whole. In this study the topic was mostly dealt from the technical point of view. In practice, more important than technology is to identify processes and methods. Implemented in a right way the MDM project applies to the whole organization and needs sufficient resources to make it possible. The project can be started carefully from a small segment, but it is important to brief and inform about the functions. All participants must have sufficient understanding about the extents of the impacts.</p>	
Keywords: master data, master data management, database, data warehouse.	

SISÄLLYS

ALKUSANAT	2
TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
KÄYTETYT MERKIT JA LYHENTEET	7
1 JOHDANTO	8
2 TIETOKANTAJÄRJESTELMÄT	10
2.1 Tieto ja tiedonhallinta.....	10
2.2 Tietokanta.....	10
2.3 SQL-tietokannat	11
3 TIETOVARASTOT.....	17
3.1 Raportoinnin haasteet	17
3.2 Ratkaisuna tietovarasto.....	19
4 MASTER DATA MANAGEMENT	21
4.1 Ydintieto.....	21
4.2 Ydintiedon hallinta	21
4.3 Ydintiedon hallintamalli.....	22
4.4 Tiedon laatu	22
4.5 Datastandardi.....	23
4.6 MDM-arkkitehtuurit.....	24
5 MICROSOFT MDS – JÄRJESTELMÄ.....	27
5.1 Mallit	27
5.2 Entiteetit	28
5.3 Attribuutit	29
5.4 MDS-järjestelmän hallinta.....	30
6 YRITYKSEN TIETOJEN HALLINTA	32
6.1 ETL-prosessi	33
6.2 Henkilötietojen vienti MDS-järjestelmään.....	34
6.3 MDS-järjestelmän valmistelu.....	36
7 SSIS-PROJEKTI.....	38
7.1 Extract-vaihe	39
7.2 Transfer-vaihe	43

7.3	Load-vaihe.....	46
7.4	ETL-prosessin suorittaminen.....	50
7.5	Tietojen vienti MDS:sta muihin järjestelmiin.....	53
8	POHDINTA.....	54
	LÄHTEET.....	55
	LIITTEET.....	56

KÄYTETYT MERKIT JA LYHENTEET

DB	Database
SQL	Structured Query Language
SSIS	SQL Server Integrated Services
MDS	Master Data Services
MDM	Master Data Management
ETL	Extract Transform and Load
MDM	Master Data Management
MDS	Master Data Service
CSV	Comma Separated Value
AD	Active Directory
FIM	Forefront Identity Manager
API	Application Programming Interface
FTP	File Transfer Protocol
BI	Business Intelligence
DG	Data Governance
EIM	Enterprise Information Management

1 JOHDANTO

Tämän opinnäytetyön tilaajana oli Kemi-Tornionlaakson koulutuskuntayhtymä Lappia. Koulutuskuntayhtymän tehtävänä on ylläpitää Kemi-Tornion ammattikorkeakoulua ja Ammattiopisto Lappiaa. Kokoaikaista henkilöstöä Lappiassa on noin 710 ja opiskelijoita noin 6 400 (Lappian www-sivut 2013, hakupäivä 17.11.2013). Kemi-Tornion ammattikorkeakoulu on kuulunut osaksi Lapin korkeakoulukonsernia (Lapland University Consortium, LUC) vuodesta 2008 lähtien. Korkeakoulukonsernin muodostavat kolme lappilaista korkeakoulua, Kemi-Tornion ammattikorkeakoulu, Lapin yliopisto sekä Rovaniemen ammattikorkeakoulu (LUC:n www-sivut 2013, hakupäivä 17.11.2013). Konsernilla on yhteisiä tietojärjestelmiä, joita käyttää myös Ammattiopisto Lappia sekä Lappian yhtiöpalvelut. Konsernin yhteisissä tietojärjestelmissä on yhteensä noin 25 000 käyttäjää.

Lappialla on kymmeniä tietojärjestelmiä, joihin tallennettujen tietojen määrä on yhteensä useita teratavuja (TeraByte, TB). Tiedot syntyvät ja tallentuvat operatiivisiin järjestelmiin, jotka ovat pääosin itsenäisiä saarekkeita. Järjestelmissä käsitellään paljon samaa tarkoitettavia tietoja, jotka eivät välity tai päivitty järjestelmien välillä automaattisesti. Joitakin tiedonsiirtoja on toteutettu suoraan kahden eri operatiivisen järjestelmän välille, mutta toteutuksissa on huomioitu vain kyseisten järjestelmien vaatimat yksittäiset tarpeet.

Viime aikoina on yhä enenevässä määrin lisääntynyt tarve tietojen keräämiselle ja yhdistämiselle useista eri lähdejärjestelmistä sekä organisaation sisäisiin että ulkopuolisten tahojen tarpeisiin. Sisäisiä tarpeita ovat esimerkiksi käyttäjätietojen hallinta ja johdon tarvitsemat raportit. Ulkopuolisia ovat mm. opetusministeriö, opetushallitus ja tilastokeskus. Tietojen kerääminen ja yhdistäminen useista eri järjestelmistä on usein hankalaa ja aikaa vievää työtä.

Opinnäytetyön tavoitteena oli selvittää, miten eri järjestelmissä olevaa tietoa saataisiin nykyistä helpommin koottua ja yhdistettyä eri tarpeisiin. Lisäksi työssä oli tarkoituksena pohtia ne prosessit ja menetelmät, joiden avulla järjestelmissä oleva samaa tarkoitettava tieto saataisiin yhtenäistettyä keskenään.

Tutkimuksen kohteeksi rajattiin henkilötietojen käsittely ja tallennus eri järjestelmissä. Henkilötietoja tallennetaan useisiin erillisiin järjestelmiin mm. uuden henkilön aloittaessa työsuhteen. Tietojen kirjauksessa saattaa tulla virheitä ja tiedot voivat muuttua työsuhteen

aikana, joten tiedot eivät pysy ajan tasalla kaikissa tietojärjestelmissä. Työsuhteen päättyessä henkilötietojen tulisi myös päivittyä järjestelmin välillä ja poistua näkyviltä osassa tietojärjestelmistä. Myös tietojärjestelmien käyttäjätilien tulisi lukittua työsuhteen päätymisen jälkeen organisaation tietoturvapoliitikan mukaisesti.

Tarkastelun kohteena oli tutkia Microsoft Data Management Service -tuotteen (MDS) soveltuvuutta käsiteltävän tiedon hallintaan. MDS ei ole itsenäinen sovellus, vaan se kuuluu osaksi Microsoft SQL Server Enterprise -tuotetta. Tuotteen mukana tulee myös Data Tools -komponentit ja Data Quality Services -sovellus, joita hyödynnetään yhdessä MDS:n kanssa.

2 TIETOKANTAJÄRJESTELMÄT

2.1 Tieto ja tiedonhallinta

Organisaation kannalta tiedot ovat oleellinen resurssi, joten niiden varastointi edellyttää investointeja. Tietojärjestelmissä käytetään yleisesti tietokantatekniikkaa tietojen tallentamiseen. Käytännössä organisaatiot ovat täysin riippuvaisia tietokannoistaan. Johdon tekemät tärkeät päätökset perustuvat järjestelmiin tallennettuihin tietoihin. Siksi on tärkeää, että tiedoista on saatavissa nopeasti koosteita johdon tarvitsemiin raportteihin. (Hovi 2005, 4.)

Tiedonhallinta tarkoittaa tallennetun tiedon määrittelemiseen sekä käyttämiseen liittyvää toimintaa. Esimerkiksi materiaali ja henkilöstö ovat tärkeitä resursseja, joiden yhteydessä puhutaankin materiaalin hallinnasta ja henkilöstön hallinnasta. Tietoresurssien hallinta on täten tiedonhallintaa. (Hovi, 4.)

2.2 Tietokanta

Tietokanta (Database, DB) pitää sisällään yhteen liittyä tietoa. Tiedot vastaavat asioita, jotka voidaan kirjata ja joilla on jokin merkitys. Esimerkiksi reseptikokoelma tai videorekisteri voivat olla tietokantoja. Tietokannoilla on seuraavia ominaisuuksia:

- Tietokanta esittää jotain reaali maailman asioita, joiden muutokset vaikuttavat tietokantaan.
- Tietokanta sisältää loogisesti yhteenkuuluvaa tietoa, jolla on jokin merkitys.
- Tietokanta on suunniteltu ja rakennettu jotakin tiettyä tarkoitusta varten ja se on kohdennettu tietylle käyttäjäryhmälle.

Tietokantojen koot vaihtelevat hyvinkin pienestä ja yksinkertaisesta erittäin massiiviseen ja monimutkaiseen. Tietosisältö voi olla vaikkapa puhelinluettelo, joka sisältää nimet, puhelinnumerot ja osoitteet. Toisaalta se voi olla esimerkiksi internetin hakukone, joka pitää sisällään hakusanoja koko WWW:n sisällöstä. Tietokannan vaatimuksia ovat muun muassa seuraavat:

- Kukin tieto tallennetaan kannassa kertaalleen, jolloin tietokannassa ei esiinny turhaa toistoa.

- Tietoa voidaan hakea joustavasti monin perustein.
- Tietokannan rakenteen tulee olla helposti muutettavissa.

(Lahtonen 2002, 2.)

2.3 SQL-tietokannat

Lähes kaikkien tietokantojen taustalla käytetään nykyisin SQL-kieltä (Structure Query Language). Tätä tukevat tunnetuimmat ohjelmistotoimittajat, kuten IBM, Oracle ja Microsoft. SQL on hyvin standardoitu ja perusmuodossaan lähes samanlaista eri tuotteissa. Sen vuoksi se on melko hyvin siirrettävää eri tuotteiden välillä. (Hovi 2004, 2.)

Nykyisin SQL-kieltä käytetään myös avoimen lähdekoodin tuotteissa, kuten MySQL. SQL onkin siis yleinen ja tärkeä kieli, joka on aluksi helpon näköistä, mutta sillä voi saada aikaan myös aidon näköisiä, mutta väärää tuloksia. (Hovi 2004, 2.)

Nimi SQL on lyhenne sanoista Structured Query Language eli strukturoitu kyselykieli. Nimestään huolimatta se kattaa kyselyjen lisäksi seuraavat alueet.

- tietokannan rakenteen määrittely ja muuttaminen (CREATE, ALTER, DROP)
- kyselyt (SELECT)
- lisäykset, muutokset ja poistot (INSERT, UPDATE, DELETE)
- tapahtumakäsittelyt (COMMIT, ROLLBACK)
- valtuuksien ja turvallisuuden hallinta (GRANT, REVOKE)
- upotettu SQL ja kohdistimien hallinta (DECLARE, CURSOR, FETCH)
- API-rajapinnat ohjelmointikieliin. (Hovi 2004, 14.)

Tietokannan kentät

Kentille voidaan määritellä seuraavan laisia ominaisuuksia.

- kentän nimi (fieldname)
- tietotyyppi (data type), joka voi olla numeerinen, alfanumeerinen (merkkijono), päivämäärä ja/tai kellonaika, looginen (true/false) tai bittijono
- maksimipituus ja mahdollinen tarkkuus
- pakollisuus (required)
- oikeellisuuden tai sallittujen arvojen tarkistus (validy)
- oletusarvo (default value)
- syöttömaski (input mask), joka määrittelee kelpuutettavat merkit ja niiden muodon.

Tietotyypit ja niiden nimet vaihtelevat jonkin verran tietokantaohjelmistojen kesken. Alla on lueteltu esimerkkejä Microsoft SQL Server -järjestelmän tietotyypeistä.

Taulukko 1. Esimerkkejä SQL Server tietotyypeistä

(W3schools.com www-sivut 2013, hakupäivä 24.11.2013)

char(n)	kiinteän mittainen merkkijono
varchar(n)	vaihtelevan mittainen merkkijono
text	vaihtelevan mittainen merkkijono(pitkä tekstitieto, esim. selite-kentässä)
smallint	kokonaisluku väliltä -32.768 ja 32.767
int	kokonaisluku väliltä -2.147.483.648 ja 2.147.48.647
float(n)	liukuluku väliltä -1,79E + 308 ja 1,79E + 308
datetime	aika väliltä 1.1.1753 ja 31.12.9999 3,33 millisekunnin tarkkuudella
date	päivämäärä väliltä 1.1.0001 ja 31.12.9999
time	kellonaika 100 nanosekunnin tarkkuudella

Merkkijonokenttään voi tallentaa myös numeroita. Mikäli numeroita käytetään laskemiseen, tulee kentän tietotyyppi olla numeerinen.

Relaatiomalli

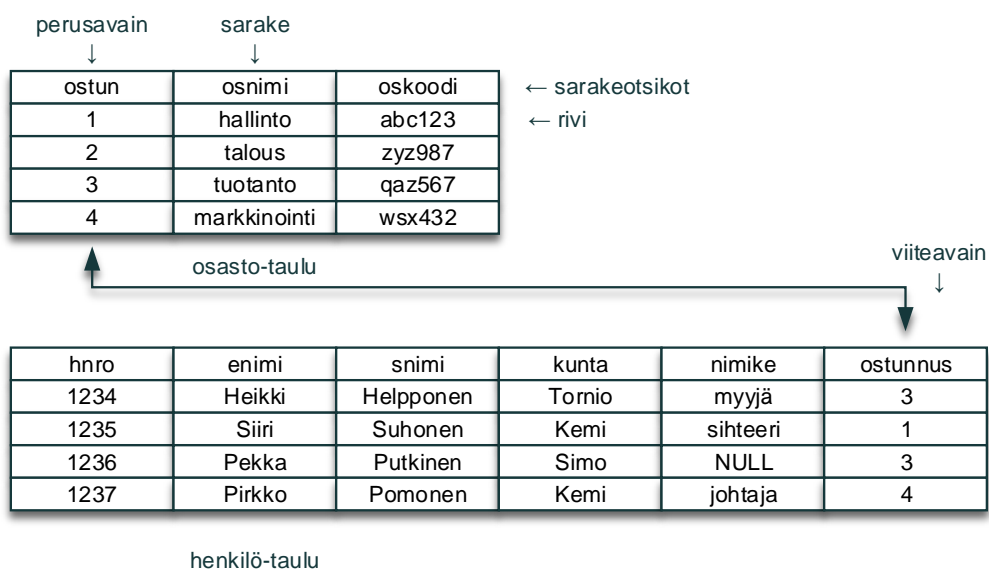
Kaikki relaatiotietokannat perustuvat IBM:n tutkija E.F. Coddin vuonna 1970 julkaise-
maan relaatiomalliin. Malli määrittelee relaatiotietokantojen teorian ja se perustuu
joukko-oppiin, matematiikkaan ja predikaattilogiikkaan. Coddin relaatiomalli onkin syr-
jäyttänyt aiemmin käytetyt hierakiset ja verkkomalliset tietokantatyypit. Kaikissa re-
laatiotietokantatuotteissa käytetään SQL-kieltä ja suurin osa uusista tietojärjestelmistä ra-
kennetaan relaatiokantatuotteiden avulla.

Relaatiomalli jakautuu kolmeen osaan.

- rakenne
- käsittely
- eheyssäännöt (Hovi 2004, 5.).

Tietokannan taulu

Alla olevassa kuviossa (Kuvio 1) on lueteltu osastoja ja niihin liittyviä henkilöitä. Osas-
toon voi liittyä monta henkilöä, mutta henkilö kuuluu aina vain yhteen osastoon. (Hovi
2004, 5.)



Kuvio 1. Esimerkki kahdesta taulusta, osasto ja henkilö

Osastot ja henkilöt ovat asiakokonaisuuksia ja ne esitetään relaatiotietokannoissa tauluina (table). Taulussa on sarakkeita (column), kuten yllä ”ostun”, ”osnimi” ja ”oskoodi” sekä rivejä (row). Tauluja voidaan kutsua myös taulukoiksi, sarakkeita kentiksi ja rivejä tietueiksi. Sarakkeilla on tietotyyppi sekä pituus. Tietotyyppi voi olla esimerkiksi numeerinen tai merkkimuotoinen. Kaikki tiedot tallentuvat vain ja ainoastaan tauluihin. Taulut perustetaan SQL-kielen DDL-osalla (Data Definition Language). (Hovi 2004, 5.)

Perus- ja viiteavain

Jokaisessa taulussa on yleensä tunnisteiden perusavain (primary key, PK), kuten ostun ja hnro. Perusavaimen on oltava yksilöivä, eli kahdella tai useammalla rivillä ei saa olla samaa arvoa. Perusavain voi koostua myös useammasta sarakkeesta. Avain ei ole kuitenkaan pakollinen, kuten myöhemmistä esimerkeistä ilmenee. (Hovi 2004, 6.)

Henkilöt liittyvät osastoihin, joilla voi olla monta henkilöä. Yksi henkilö kuuluu aina vain yhteen osastoon. Tätä kutsutaan isä-lapsi-yhteydeksi tai yksi-moneen-yhteydeksi. Liitoksen toteuttamiseksi lapsitaulussa on linkkikenttä eli viiteavain (foreign key) ”ostunnus”, joka viittaa osasto-taulun perusavaimen ”ostun”. Viiteavainta tarvitaan tauluja yhdistettäessä, eli liitoksia tehtäessä. (Hovi 2004, 6). Perusavaimen ja viiteavaimen kenttien nimet eivät tarvitse olla välttämättä samoja.

NULL-arvo

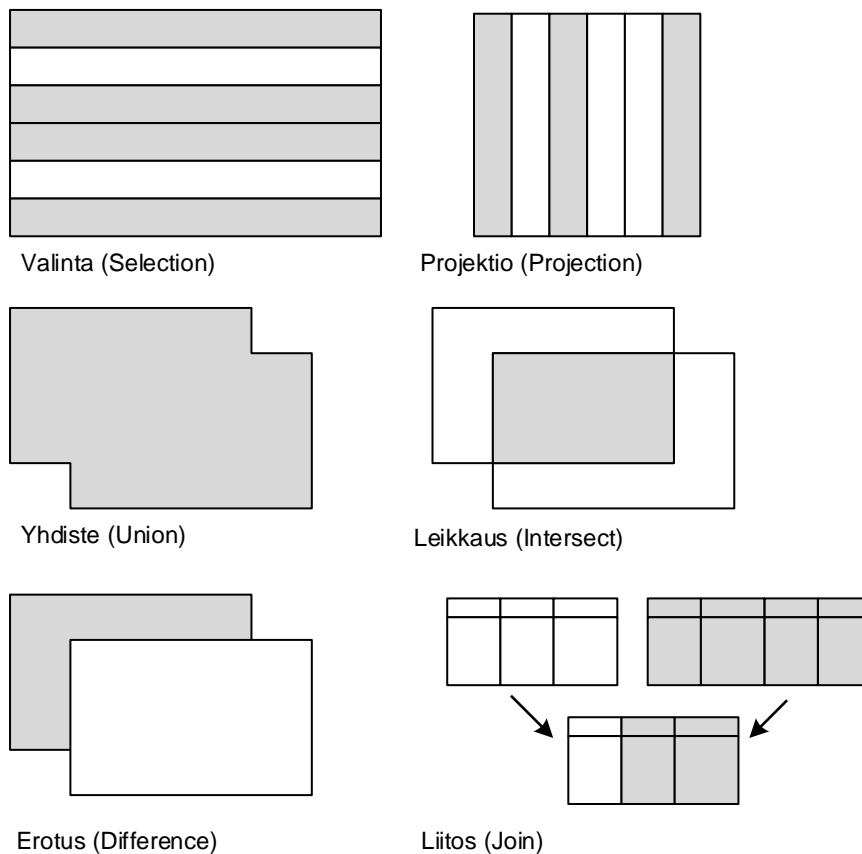
Jos tietoja syötettäessä sarakkeella ei ole arvoa, tulee kyseiseen kohtaan erityinen merkintä, ns. NULL- eli tyhjä arvo. Se ei tarkoita välilyöntiä tai nollaa, vaan tuntematonta arvoa. Jos kysytään, onko henkilön nimike esim. sihteeri, voi vastaus olla tosi, epätosi tai ehkä. (Hovi 2004, 7.)

Jos tietokantaan on tallentunut NULL-arvo, saadaan vastaukseksi ehkä. NULL-arvoja tarvitaan esimerkiksi keskiarvoja (AVG) laskettaessa, koska tällöin NULL-arvot ohitetaan. (Hovi 2004, 7.)

Käsittely

Coddin relaatiomallin mukaisesti tietokannassa olevia tietoja käsitellään joukko-opillisesti. Taulussa olevien rivien joukkoon voi kohdistaa joukko-operaatioita. Haetaan esimerkiksi kaikki tietyn kunnan henkilöt (valinta) ja niistä etunimi sekä sukunimi (projektiio). Joukko-operaatiot voivat kohdistua koko tauluun tai useampaankin samalla kertaa. Tämä koskee myös tietojen päivityksiä, joka mahdollistaa vaikkapa keskeltä sovellusohjelmaa päivittää suuri määrä taulun rivejä ilman silmukoita. (Hovi 2004, 8.)

Seuraavassa kuviossa (Kuvio 2) mainitut joukko-operaatiot voidaan tehdä SQL:n erilaisilla SELECT-käskyn muodoilla.



Kuvio 2. Tärkeimmät joukko-operaatiot (Hovi 2004, 9)

Kaikkia relaatiokannassa olevia sarakkeita voidaan käyttää hakuperusteena SELECT-käskyissä. Hakuoperaatiot eivät edellytä, että kentän tulisi olla erityinen avainkenttä. (Hovi 2004, 9.)

Eheyssäännöt

Relaatiomallissa määritetään myös tietokannan eheys (integrity). Tietokannan eheys edellyttää, että sen tiedot ovat oikein, ristiriidattomia ja vastaavat todellisuutta. Eheys ei toteudu, mikäli esimerkiksi saman henkilön tiedot tallennetaan kahteen kertaan. (Hovi 2004, 9.)

Coddin relaatiomalli sisältää tiettyjä eheysrajoitteita. Ensimmäinen eheyssäntö on avaineheys (entity integrity). Säännön mukaan perusavaimen arvo ei saa olla NULL, eli avaimen arvo on pakollinen. Esimerkiksi osasto-tauluun ei voi lisätä riviä, jossa perusavain (ostun) jäisi tyhjäksi. Kyseiselle osastolle ei voisi lisätä henkilöitä henkilö-tauluun, koska lisättäville henkilöille ei olisi tarvittavaa viiteavaimen arvoa. (Hovi 2004, 9.)

Näkymät

Näkymät (View) ovat ikkunoita varsinaisiin tauluihin, eli niihin ei tallenneta mitään tietoa. Ne näyttävät ulospäin tauluilta ja niihin voidaan kohdistaa SQL-kielen komentoja, kuten tauluihinkin. Näkymät tarjoavat käyttäjille yksinkertaistettuja näkökulmia tietokantaan. Niiden avulla voidaan myös huolehtia tauluihin tallennettujen tietojen suojauksesta, eli näytetään vain rajoitetut rivit ja/tai sarakkeet. Näkymät tarjoavat myös tietoriippumattomuutta, koska tauluihin voidaan tehdä muutoksia niiden vaikuttamatta näkymän kautta saataviin tietoihin. (Hovi 2004, 13.)

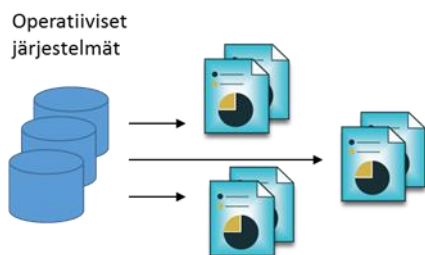
3 TIETOVARASTOT

Organisaatiossa tarvitaan monenlaista tietoa päätöksen teon tueksi. Esimerkiksi myynnin kehityksen seurantaan tarvitaan erilaisia asiakasanalyyssejä. Henkilöstön suunnittelussa puolestaan tarvitaan tietoja henkilöiden lukumääristä, koulutuksista, työajoista jne. pitkiltä aikaväleiltä. Tiedot ovat yleensä olemassa organisaation omissa perusjärjestelmissä, tarkemmin niiden tietokannoissa, joihin on varastoitunut suuria määriä tietoja asiakkaista, tuotteista, henkilöistä ja taloudesta. Tieto muodostaa arvokkaan pääoman, josta kilpailija voi olla kiinnostunut. Julkishallinnossa tiedot tulee lisäksi säilyttää pitkät ajat (Hovi, Heronen & Koistinen 2009, 4.).

Yksi tiedonhallinnan keskeisimpiä ongelmia on se, että tiedot ovat hajallaan organisaation eri järjestelmissä (Kuvio 3). Lisäksi järjestelmissä olevia tietoja ei ole useinkaan kuvattu. Ei siis tiedetä täysin, mitä eri tietokentät tietokannoissa tarkoittavat, eikä tietojärjestelmien tietosisältöjä ole mallinnettu. Toisin sanoen arvokas tietoresurssi on monesti huonossa hallinnassa, koska ei tiedetä mitä tietoja järjestelmät sisältävät, mitä tiedot tarkoittavat ja miten tiedot rakentuvat. (Hovi ym. 2009, 5.)

3.1 Raportoinnin haasteet

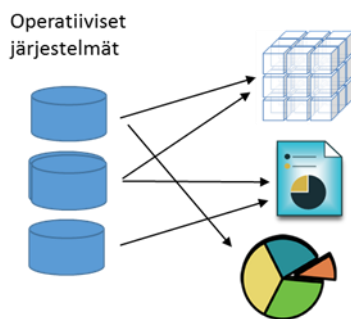
Operatiivisissa järjestelmissä on yleensä valmiita raportteja. Voisiko tätä menetelmää jatkaa ohjelmoimalla lisää raportteja uusiin tarpeisiin? Ongelmana tässä on se, että niiden tekeminen hankalista tai vieraista tietokantarakenteista on yleensä työlästä. Usein joudutaan turvautumaan ulkopuoliseen apuun, jolloin aiheutuu kustannuksia. Ongelmaksi yleensä koituu se, ettei eri järjestelmissä olevia tietoja pystytä yhdistelemään helposti. Lisäksi raportit ja kyselyt ovat raskaita operaatioita, koska ne edellyttävät tietokantojen laajaa selailua ja läpikäyntiä. (Hovi ym. 2009, 6.)



Kuvio 3. Raportit suoraan operatiivisista järjestelmistä (Hovi ym. 2009, 6)

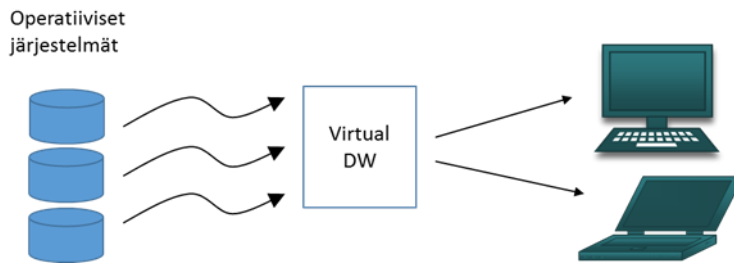
Yleisesti yhteenvetoja ja raportteja kehitetään taulukkolaskentaohjelmalla. Etuna tässä on se, että taulukkolaskentaohjelma on useimmille tuttu ja tietoja on helppo muokata sekä jaella eteenpäin. Ongelmaksi tässä tulee käsityö ja siitä seuraavat virheriskit. Tiedot syötetään käsin tai ladataan tiedostoista ja usein muokataan taulukkolaskentaohjelman makroilla. Tällaisista järjestelmistä kasvaa helposti monimutkaisia, eikä niitä osaa ylläpitää kuin järjestelmän toteuttaja. Lisäksi tämä työllistää IT-henkilöitä, jotka toimittavat käyttäjille erilaisia poimintoja operatiivisten tietojärjestelmien tietokannoista. (Hovi ym. 2009, 7.)

Yhtenä ratkaisumallina voidaan hankkia Business Intelligence -työkaluja ja yrittää kytkeä ne suoraan operatiivisiin järjestelmiin, jolloin ei toteuteta erillistä tietovarastoa (Kuvio 4). Ratkaisu voi sopia pieniin ympäristöihin, missä järjestelmiä on vähän ja ne ovat yksinkertaisia. Kyseisellä mallilla voidaan saada aikaan näyttävää ja toimivaa raportointia. Isommissa ympäristöissä tämä ei kuitenkaan toimi kovin hyvin. Tietojen yhdistelemiseen tarvittava paikka puuttuu, joten BI-välineiden kytkeminen operatiivisiin järjestelmiin on kömpelöä ja hankalaa. Myös virhealttius kasvaa verrattuna tietovarastoratkaisuun, eikä historiatiedoille ole luontevaa paikkaa. (Hovi ym. 2009, 7.)



Kuvio 4. BI-käyttö ilman tietovarastoa (Hovi ym. 2009, 8)

Raportteja voitaisiin koota yhdistelemällä tietoja lennossa matkalla raportteihin, eli käytettäisiin virtuaalista tietovarastoa (Kuvio 5). Tämä tuntuu houkuttelevalta ajatukselta, koska säästyttäisiin erilliseltä tietovarastolta eikä tietoja tarvitsisi toistaa. Tällä mallilla on kuitenkin huomattavia ongelmia. Tietojen yhdisteleminen ja integrointi on monimutkaista, eikä sitä voi tehdä nopeasti.

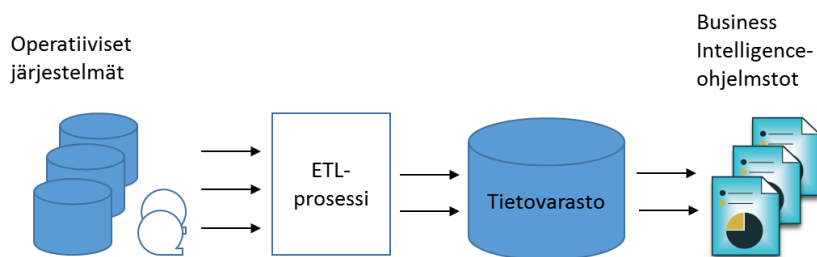


Kuvio 5. Virtuaalinen tietovarasto (Hovi ym. 2009, 8)

Eri tietojärjestelmistä tuleviin tietoihin sisältyy usein päättelyä ja välivarastointitarvetta. On siis parempi selvittää tiedot valmiiksi tietovarastoon, josta ne ovat nopeasti käytettävissä erilaisiin tarpeisiin. Ongelmaksi tulee myös tietojen historiointi, jota ei yleensä tehdä operatiivisissa järjestelmissä. Lisäksi ratkaisu kuormittaa operatiivisia järjestelmiä aiheuttaen mahdollisesti vastausaikojen pidentymistä. (Hovi ym. 2009, 8.)

3.2 Ratkaisuna tietovarasto

Tietojen hyödyntäminen analyyseissa ja raporteissa edellyttää omaa tietokantaansa, joka on erityisesti rakennettu ja suunniteltu tällaista käyttöä varten. Tietovarastotietokanta tukee hyvin myös BI-käyttöä ja -työkaluja. Alla oleva kuvio (Kuvio 6) kuvaa tietojen jalostusketjua perusjärjestelmistä tietovarastoon. Raakamuodossa olevia tietoja muokataan jalostettuun raportointi ja kyselykäyttöön sopivaan muotoon ja ladataan tietovarastoon yleensä kerran vuorokaudessa. (Hovi ym. 2009, 14.)



Kuvio 6. Tietojen kulku BI-ympäristöön (Hovi ym. 2009, 14)

Tietovaraston tiedot ovat yleensä vain lukukäytössä. Niitä ei päivitetä suoraan, koska operatiiviset tietokannat ja tietovarasto eivät olisi ajan tasalla keskenään. Muutokset tulevat siis aina operatiivisista järjestelmistä. Osa niistä voi olla myös organisaation ulkopuolella, jolloin tietovarastoon ladataan myös ulkoista tietoa yhdistettäväksi omiin tietoihin. (Hovi ym. 2009, 14.)

Edellä kerrottua prosessia kutsutaan tietovarastoinniksi, johon liittyy seuraavia asioita:

- Tiedot luetaan operatiivista järjestelmästä ja muokataan ne tietovaraston edellyttämään muotoon, joka edellyttää yhdenmukaistamista ja yhdistämistä.
- Tietovarasto (Data Warehouse) on tietojen helppoa ja nopeaa hakua varten suunniteltu tietokanta. Tietovarastoon tallennetaan monen vuoden tiedot, jolloin mahdollistetaan trendianalyysit.
- Tietovaraston tietoja kysellään, analysoidaan ja raportoidaan BI-työkaluilla. Tietovarastosta ajetaan valmisraportteja, parametroitavia raportteja sekä uusia kyselyjä.

ETL-vaiheessa operatiivisten järjestelmien koodeja yhdistetään ja käännetään ymmärrettävämpään muotoon. Joissakin järjestelmissä voi olla esimerkiksi sukupuolikoodit muodossa ”M” ja ”N” ja toisessa vastaavasti ”1” ja ”0”, joten tietovarastossa voidaan päättää käytettäväksi ”M” ja ”N”. (Hovi ym. 2009, 14.)

Tietoja jalostetaan raakatiedoista raportteja ja kyselyjä varten paremmin palveleviksi. Esimerkiksi perusjärjestelmässä oleva henkilötunnus on hankala, mikäli halutaan ryhmitellä henkilöt ikäryhmiin. ETL-prosessissa voidaan poimia päivämäärä ja laskea sen perustella henkilön ikä. Prosessin aikana voidaan tehdä myös oikeellisuustarkistuksia, jotta mahdolliset virheet saataisiin havaittua. Kaiken kaikkiaan eri operatiivisten järjestelmien tiedot saadaan selkeään ja helppolukuiseen muotoon käyttäjien ulottuville BI-alueen kysely ja raporttivälineillä. (Hovi ym. 2009, 14.)

4 MASTER DATA MANAGEMENT

Julkisen hallinnon toiminta on hyvin tietointensiivistä, eli perustuu vahvasti tiedon käsittelyyn ja hallintaan. Toiminnan tarvitseman tiedon laatu ja saatavuus sekä tiedon hallinta korostuvat entisestään toimintaa tukevien tietojärjestelmien merkityksen lisääntyessä. Koko nykyaikainen yhteiskunta perustuukin tiedon jäsentyneeseen hallintaan. Organisaatio ei voi toimia tehokkaasti ilman laadukasta tietoa ja siitä saatavaa informaatiota. (JHS 179 2011, 2.)

4.1 Ydintieto

Ydintieto (Master Data), on tietoa, joka on koko organisaation toiminnan kannalta välttämätöntä perustietoa. Ydintiedoksi luokitellaan tieto, jota organisaatio tarvitsee toiminnassaan samanlaisena ja -laatuisena ja on tarve hyödyntää eri prosesseissa. Ydintieto on organisaation toiminnan näkökulmasta pysyvää ja kuvaa sellaisia tietokokonaisuuksia, joista on tai tulisi olla yhtenäinen käsitys koko organisaatiossa. (JHS 179 2011, 2.)

Esimerkkejä organisaation ydintiedosta ovat mm.

- osapuolitiedot (asiakkaat, henkilöt, työntekijät, toimittajat, myyjät)
- paikkatiedot (sijainti, toimistot, organisaatiot)
- muut asiat ja esineet (tilitiedot, omaisuus, toimintaperiaatteet, tuotteet ja palvelut). (JHS 179 2011, 2.)

Ydintiedoksi voidaan luokitella myös erinäiset luokittelu- ja ryhmittelytekijät, kuten esimerkiksi organisaatorakenne, palvelualueet, palvelu- ja tuoteryhmät, hinnastot, jne. (JHS 179 2011, 2.)

4.2 Ydintiedon hallinta

Ydintiedon hallinta (Master Data Management, MDM), on toiminto ja prosessi, jonka tehtävänä on hallita, kehittää ja ylläpitää toiminnan tarvitsemaa ydintietoa ja laatua. Ydintiedonhallinnan vastuulla on ylläpitää tiedon eheyttä eri järjestelmien välillä. Se vastaa

keskeisen tiedon käytettävyydestä organisaatiossa, koordinoi tietoa ja sen ylläpitoa toiminnassa käytettävien tietojärjestelmien välillä tarkoituksen mukaisten menetelmien ja mallien avulla. (JHS 179 2011, 2.)

4.3 Ydintiedon hallintamalli

Ydintiedon hallintamalli (Data Governance, DG) koostuu prosesseista, joilla varmistetaan tarvittavien tietojen määrämuotoinen hallinta organisaatiossa. Tällaisia prosesseja ovat esimerkiksi tiedonhallintaprosessit, tiedonhallintalinjausten noudattamisen- ja hyödyntämisen prosessit sekä muutostenhallintaprosessit. Ydintiedonhallintamalli onkin laadunvarmistusmenetelmä, jolla organisaation tärkeää tietoa ylläpidetään sisältäen kaikki ihmiset, prosessit ja tietojenkäsittelytekniikat, jotka tarvitaan yhtenäisen ja hyvän organisaation tiedonhallinnan luomiseksi. (JHS 179 2011, 2.)

DG on kehitteillä oleva toimintamalli, joten yleisesti tunnustettua määritelmää ei ole toistaiseksi olemassa. DG kattaa koko organisaation tiedon hallinnan, tiedon laadun, liiketoimintaprosessien johtamisen ja riskienhallinnan. DG:n tarkoituksena on saada aikaan positiivinen kontrolli prosesseihin ja toimintatapoihin, joilla tietoa hallinnoivat henkilöt käsittelevät tietoa sen koko elinkaaren aikana. (Niemi & Kontra, hakupäivä 16.1.2013)

DG koostuu prosesseista, toimintatavoista, rooleista ja vastuista. Näillä on tarkoitus varmistaa ohjeistuksen mukainen hallinta tiedolle koko organisaation laajuisesti. DG:n avulla varmistetaan, että tieto on luotettavaa. Siinä määritellään myös vastuuhenkilöt huonolle tiedon laadulle sekä parantamiseen tähtääville toimenpiteille. DG:n avulla määritellään myös iteratiivinen prosessi, jolla se voi muuttaa ajattelutapoja sekä varmistaa, että samaa informaatiota voidaan hyödyntää koko organisaatiossa. (Niemi & Kontra 2012, hakupäivä 16.1.2013)

4.4 Tiedon laatu

Tiedon laadun (Data Quality) merkitys kasvaa jatkuvasti päätöksenteossa ja operatiivisissa prosesseissa datamäärien huiman kasvun myötä. Viimeisten vuosikymmenien ai-

kana tietojärjestelmät ovat muuttuneet keskitetyistä järjestelmistä verkostomaiseksi, jolloin käytössä olevin potentiaalisten tietolähteiden määrä ja laajuus ovat kasvaneet räjähdysmäisesti. (Laatikainen & Niemi 2012, hakupäivä 16.1.2013)

Tämä on johtanut siihen, että tiedon laatu heikkenee nopeasti, ellei prosesseissa ja tiedon alkuperäisissä lähteissä ole tiukkaa laadun kontrollia. Toisaalta tietolähteiden hajaantuneisuus ja verkostoituminen on mahdollistanut uusia keinoja tiedon laadun hallintaan valitsemalla laadullisesti parasta tietoa sisältävä järjestelmä ensisijaiseksi tietolähteeksi. Myös virheiden löytäminen voi helpottua vertailemalla eri järjestelmien tietoja keskenään. (Laatikainen & Niemi 2012, hakupäivä 16.1.2013)

Tiedon laadun hallinnalla tarkoitetaan niitä prosesseja ja teknologioita, joilla varmistetaan, että tieto täyttää sille asetetut vaatimukset ja hyväksymiskriteerit. Tavoitteena ei ole välttämättä täydellinen laatu. Tavoitteiden toteutumista voidaan tarkastella esimerkiksi seuraavien ominaisuuksien avulla.

- accuracy: tieto kuvastaa oikein reaali maailman tilannetta
- completeness: kaikki tarvittava tieto on annettu
- consistency: tieto on yhdenmukaista esim. järjestelmien välillä
- duplication: samaa tietoa ei ole kahdesti
- timeliness: tieto on saatavilla riittävän nopeasti.

(Laatikainen & Niemi 2012, hakupäivä 16.1.2013)

4.5 Datastandardi

Datastandardin avulla muodostetaan yhteinen, ymmärrettävä ja yksiselitteinen kieli ydintiedoille organisaatiossa. Se kuvaa sovitun merkityksen, rakenteen, sisällön sekä laatuvaatimukset ydintiedolle. Standardointia tarvitaan luotettavan raportoinnin pohjaksi, tiedon tehokkaaseen hyödyntämiseen sekä kommunikointiin eri kohderyhmien kanssa. (Talent Base 2012, 37.)

Datastandardista tulisi ilmetä määritelmät entiteeteille ja määritteille (attribuuteille), jotka kuvaavat reaali maailman objekteja. Lisäksi siitä tulisi ilmetä mahdolliset yhteydet eri entiteettien välillä, objektikohtaiset määritteet sekä niiden sisältöjen kuvaukset ja vaatimukset. (Talent Base 2012, 37.)

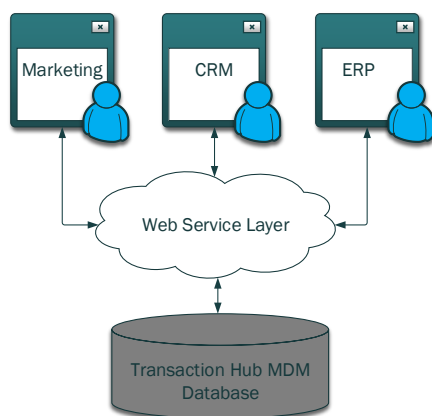
Esimerkiksi henkilötietojen tapauksessa törmää usein tietojen ristiriitaisuuteen. Varsinkin kutsumanimi tuottaa usein ongelmia. Joihinkin järjestelmiin on kirjattu etunimen ensimmäinen nimi ja toisissa on käytetty henkilön varsinaista kutsumanimeä. Myös sukunimi-tieto on sellainen, jota ei välttämättä päivitetä tarvittaviin järjestelmiin sen muuttuessa. Esimerkki attribuuttitaso standardista löytyy liitteestä 1. Lyhenteet C, R, U ja D kuvaavat käyttöoikeustasojen create, read, update ja delete.

4.6 MDM-arkkitehtuurit

MDM hub on järjestelmä, jolla hallitaan tietokantaan tallennettua ydintietoa ja pidetään se yhteneväisenä operatiivisten tietojärjestelmien kanssa. Perusarkkitehtuureja on kolme: transaction hub (repository), registry ja hybrid, joka on edellä mainittujen yhdistelmä.

Transaction Hub

Transaction Hub -arkkitehtuuri on helpoin selittää, mutta se on ehkä vaikein toteuttaa. Ideana tässä on yksi keskitetty varasto, johon kukin sovellus voi kytkeytyä suoraan ja hakea tarvitsemansa ydintiedon (Kuvio 7). Esimerkiksi CRM-sovellus voidaan määrittellä hakemaan asiakastiedot Cust-nimisestä taulusta CRM-nimisestä SQL-tietokannasta. (Kashel, Kent & Bullerwell 2011, 18.)

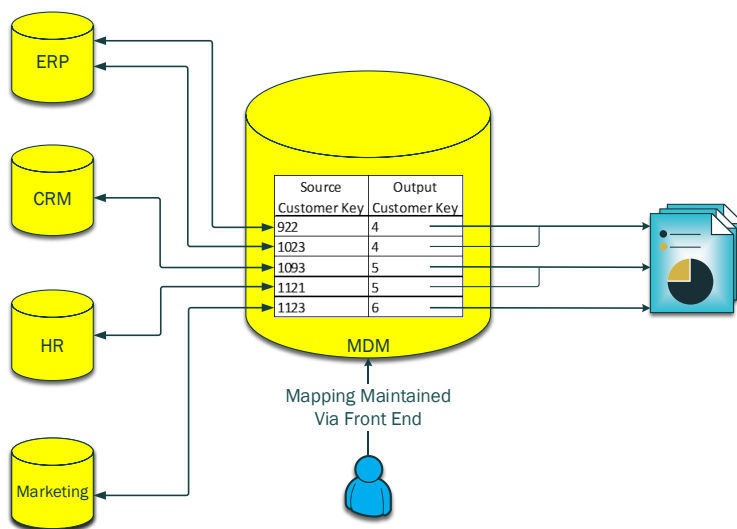


Kuvio 7. Transaction Hub MDS (Kashel ym. 2011, 18)

Sovelluksilla tulee olla mahdollisuus lisätä, päivittää ja poistaa tietueita tarpeen mukaan. Tämä sallitaan Transaction Hub -toteutuksessa alkuperäiselle sovelluksella muutosten helpottamiseksi, mutta muokattavat tiedot tallennetaan keskustietokantaan. Etuna tällä toteutuksella on se, että tietojen päällekkäisyyksiltä vältytään. Ongelma tässä toteutuksessa on integrointi operatiivisiin sovellusohjelmiin. (Kashel ym. 2011, 19.)

Registry

Registry-toteutuksessa on tietokanta, jossa linkitetään sovelluksista tuleva tieto yksilölliseksi ydintietonäkymäksi (Kuvio 8) ydintietoja hallitaan operatiivisissa järjestelmissä ja tiedot linkitetään yhdeksi tietueeksi MDM-järjestelmässä. Tätä voidaan käyttää raportointiin tai muuhun vastaavaan tarpeeseen, jossa tarvitaan näkymä ydintietojen kokonaisuudesta. (Kashel ym. 2011, 19.)



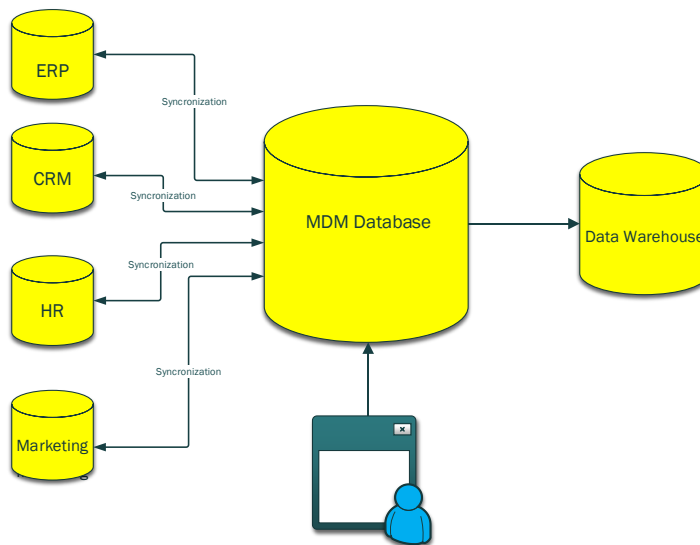
Kuvio 8. Registry MDS (Kashel ym. 2011, 20)

Mitään lisämääreitä (esim. nimi, osoite) ei tallenneta rekisteriin, vaan ne saadaan suoraan lähdejärjestelmistä. Rekisteri onkin tyypillisesti vain lukutilassa oleva ydintietojen sovellus. Koska järjestelmässä ylläpidetään vain linkityksiä, mitään käyttöliittymää tai muuta mekanisme ei tarvita käyttäjille tietojen päivittämiseksi. (Kashel ym. 2011, 20.)

Ongelmana rekisterissä voi olla tietojen päällekkäisyydet, koska samoja tietoa voidaan päivittää useammassa eri lähdejärjestelmässä. Tätä varten järjestelmässä tulee luoda säännöt mahdollisten päällekkäisyyksien eliminoimiseksi. Etuna tässä toteutuksessa on se, että tarvittavat tiedot saadaan yhden näkymän kautta tarvitsematta hakea tietoja eri lähdejärjestelmistä. (Kashel ym. 2011, 21.)

Hybrid

Hybrid on sekoitus kahdesta edellä esitetyistä toteutuksesta. Hybrid-toteutuksessa sovel-lusohjelmistot käyttävät omia tietokantoja, kuten ennen MDM-järjestelmän olemassa-oloa. Hybridin tehtävänä on kytkeytyä sovellusten tietokantoihin ja tunnistaa ydintietoja koskevat muutokset. Tietojen poiminnassa voidaan tehdä useita toimenpiteitä, kuten tietojen laadun tarkistaminen, tietojen rikastaminen ja päällekkäisyyksien poistaminen. (Kashel ym. 2011, 21.)



Kuvio 9. Hybrid MDS (Kashel ym. 2011, 23)

Hybridissä voidaan lisätä määritteitä, joita ei ole lähdejärjestelmissä (Kuvio 9). Ne voidaan yhdistää ydintietoihin ja viedä edelleen esimerkiksi tietovarastoon (Data Warehouse). Haasteita tässä toteutuksessa voi aiheutua tietojen päivittämisestä takaisin lähdejärjestelmiin, kuten Transaction Hub -toteutuksessakin. (Kashel ym. 2011, 23.)

5 MICROSOFT MDS – JÄRJESTELMÄ

Microsoft Master Data Services on SQL Server – toteutus ydintietojen hallinnalle. Sen avulla voidaan hallita organisaatiossa koottuja ydintietojen listoja. MDS:n avulla voi hallita mitä hyvänsä asialuokkia, eli se ei rajoitu pelkästään asiakas-, tuote- tai käyttäjätietojen hallintaan. MDS ei sisällä valmiita asialuokkia, vaan ne määritellään itse omiin tarpeisiin perustamalla niille mallit (model). (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013.)

MDS sisältää seuraavat komponentit ja työkalut:

- ”*MDS Configuration Manager*” on työkalu, jota käytetään MDS-tietokantojen ja -sovellusten hallintaan
- ”*Master Data Manager*” on pääkäyttäjätason sovellus mallien, sääntöjen ja käyttöoikeuksien hallintaan
- ”*MDSModelDeploy.exe*” on työkalu, jolla paketoidaan mallin oliot ja tiedot siirrettäväksi toiseen ympäristöön
- ”*Master Data Services*” on käyttöliittymä MDS-sovellusten räätälöintiin
- ”*MDS Add-in for Excel*” on työkalu tietojen hallintaan ja uusien entiteettien ja attribuuttien perustamiseen. (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013.)

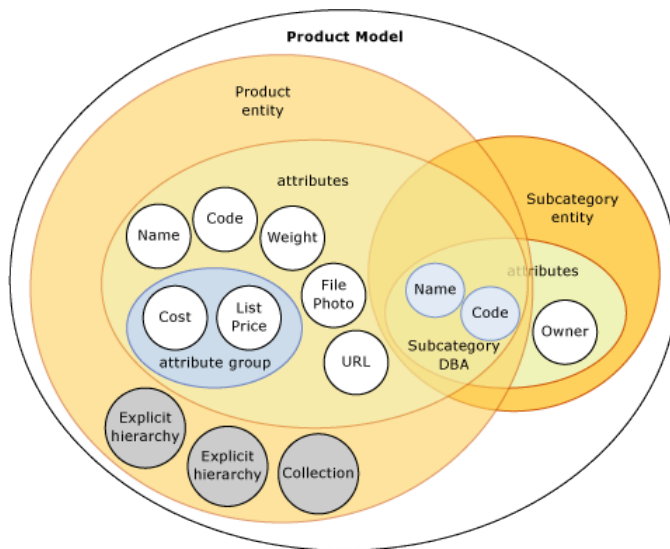
MDS sisältyy osaksi MS SQL Server 2012 Enterprise -tuotetta samoin, kuin SQL Server Data Quality Services (DQS), jota voidaan käyttää MDS:n yhteydessä muun muassa tietojen puhdistamiseen ja oikeellisuuden tarkistamiseen.

5.1 Mallit

Mallit (models) ovat MDS:n korkein tiedon taso. Malli määrittelee tiedon rakenteen MDS-sovelluksessa. Malli sisältää seuraavat oliot (object).

- entiteetit (entities)
- attribuutit (määritteet) ja attribuutti-ryhmät (attributes, attribute groups)
- eksplisiittiset ja johdetut hierarkiat (explicit, derived hierarchies)
- koosteet (collections).

MDS-toteutus voi sisältää yhden tai useita malleja, joista kukin sisältää samantyyppistä tietoa. Malleille voidaan määritellä käyttöoikeuksia tietojen lukemiseen tai päivittämiseen. Mikäli mallille ei ole annettu oikeuksia, mallia ei näytetä. Malli sisältää entiteettejä. Entiteetit sisältävät attribuutteja, eksplisiittisiä hierarkioita ja koosteita. Alla olevassa esimerkissä (Kuvio 10) on esitetty objektien väliset suhteet mallissa. (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013.)



Kuvio 10. MDS-malli (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013)

Ydintieto on tietoa, joka sisältyy mallin objekteihin. MDS:ssa ydintieto tallennetaan entiteetin jäsenenä (members). Mallin objekteja hallitaan System Administration -alueella MDM käyttöliittymässä.

5.2 Entiteetit

Entiteetit (entities) ovat objekteja, jotka sisältyvät MDS:n malleihin. Jokainen entiteetti sisältää jäseniä (members), jotka ovat hallittavia ydintiedon rivejä. Mallit voivat sisältää useita entiteettejä joihin kootaan samantapaista tietoa, esimerkiksi ensisijainen lista henkilökunnasta. Entiteeteistä voidaan tehdä myös vakioitu lista attribuutteja, joita käytetään jäsenten tietojen täydentämisessä. (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013.)

5.3 Attribuutit



Attribuutit sisältyvät entiteetteihin ja niiden arvot kuvaavat entiteetin jäseniä. Attribuutin voidaan ajatella olevan entiteetti-taulun sarake. Attribuutit voidaan koostaa myös ryhmiksi. Pakollisia attribuutteja ovat nimi (name) ja koodi (code), joita ei voi poistaa eikä nimetä uudelleen.

Attribuutteja on kolmea tyyppiä.

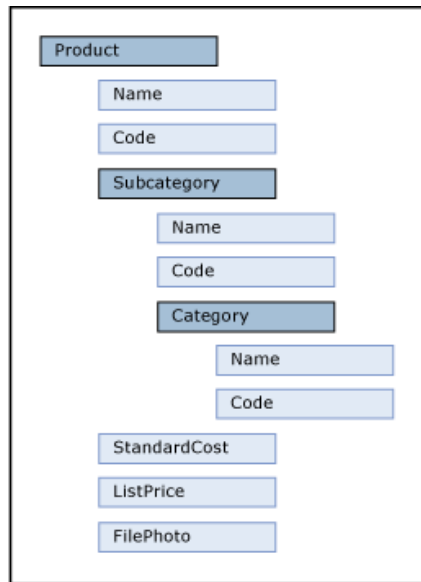
- vapaamuotoinen (Free-form), johon voi tallentaa vapaasti tekstiä, numeroita päivämääriä tai linkkejä
- toimialuepohjainen (Domain-based), jonka arvot ovat tallennettu toiseen entiteettiin (valintalista)
- tiedosto (File), johon voi tallentaa dokumentin tai kuvan. (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013)

Alla olevassa esimerkissä (Taulukko 2. Esimerkki MDS-on attribuutit: Name, Code, Subcategory, StandardCost, ListPrice ja FilePhoto. Nämä attribuutit kuvaavat jäseniä (members). Jokainen jäsen edustaa yhtä riviä määritteiden arvoja. (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013)

Taulukko 2. Esimerkki MDS-attribuuteista

Name	Code	Subcategory	StandardCost	ListPrice	FilePhoto
Mountain-100	BK-M101	Mountain Bikes {5}	1912.15	3399.99	
Mountain-200	BK-M201	Mountain Bikes {5}	1265.62	2319.99	
Road-150	BK-R150	Road Bikes {6}	2171.29	3578.27	
Road-450	BK-450	Road Bikes {6}	884.71	1457.99	

Esimerkissä Subcategory on entiteetti, jota käytetään toimialuepohjaisena attribuuttina Product-entiteetille. Sekä Product-, Subcategory- että Category-entiteeteillä on Name- ja Code-määritteet (Kuvio 11). (Microsoft MSDN www-sivut 2012, hakupäivä 5.3.2013)



Kuvio 11. Product-entiteetti

5.4 MDS-järjestelmän hallinta

MDS-järjestelmää hallitaan selainpohjaisella käyttöliittymällä (liite 2). Käyttöliittymä on monikielinen ja käytettävä kieli riippuu selaimen maa-asetuksista (selaimen käyttöliittymän kielisyydellä ei ole vaikutusta). Käyttöliittymä jakautuu seuraaviin osiin.

- hallintatyökalu (explorer)
 - o ydintietojen ja hiarkiasuhteiden käsittely
- Microsoft Excelin Master Data Services -apuohjelma
 - o linkki Microsoftin latauspalveluun, josta voi ladata apuohjelman
 - o tietojen käsittely MS Excel-taulukkolaskentaohjelmalla
 - o järjestelmänvalvojat voivat ladata tietoja, luoda entiteettejä ja päivittää määritteitä.
- järjestelmän hallinta (system administration)
 - o mallien ja liiketoimintasääntöjen hallintaan
- integroinnin hallinta (integration management)
 - o tietojen tuonti MDS:ään ja SQL-näkymien hallinta

- versioiden hallinta (version management)
 - o ydintietojen versioiden luontiin
- käyttäjän ja ryhmän oikeudet (user and group permissions)
 - o ydintietojen käyttöoikeuksien määrittely.

Tässä yhteydessä jätetään käsittelemättä tarkempi järjestelmän hallinta. Järjestelmään on perustettu LappiaKayttaja-niminen malli, joka vastaa määritteilään ETL-prossista tuotavaa tietoa. Malli on sinänsä yksinkertainen, eikä siis sisällä hierarkioita tai toimialuepohjaisia attribuutteja.

6 YRITYKSEN TIETOJEN HALLINTA

Yrityksen tietojen hallintaan (Enterprise Information Management, EIM) liittyy joukko tietojärjestelmiä. Microsoft SQL Server 2012 tarjoaa yhdellä tuotteella kaikki komponentit, joita tarvitaan tehokkaaseen EIM -ratkaisuun. Komponentit EIM:n rakentamiseen ovat:

- SQL Server Integration Services (SSIS)
- SQL Server Data Quality Services (DQS)
- SQL Server Master Data Services (MDS).

SSIS mahdollistaa tehokkaan, joustavan ja laajennettava alustan tietojen integroimiselle useista tietolähteistä ETL-ratkaisulla, joka tukee työkulkuja, tietovarastoja tai ydintietojen hallintaa.

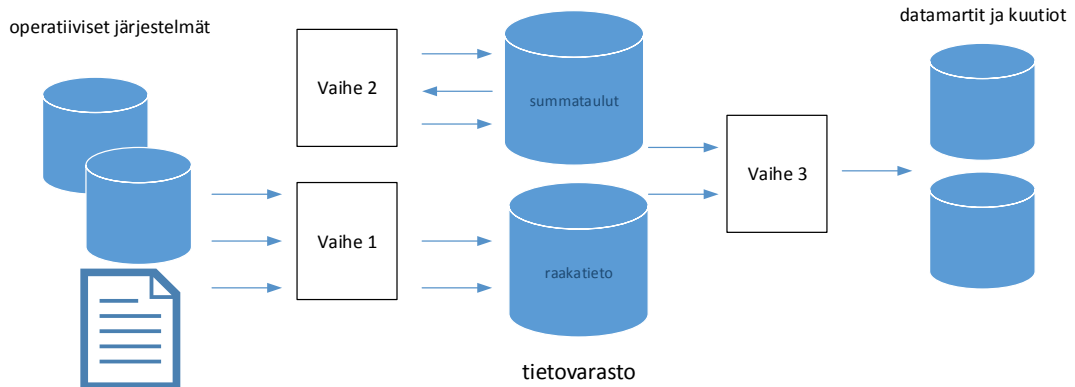
DQS mahdollistaa tietojen puhdistamisen, yhdistämisen, standardoimisen ja rikastamisen, jolloin voidaan jakaa luotettavaa tietoa liiketoiminnalle, tietovarastoon ja tapahtumien käsittelyä varten.

MDS tarjoaa keskitetyn pisteen, jolla varmistetaan tietojen eheys ja yhdenmukaisuus vakioituna eri sovelluksissa (Pelluru & Alva Bravo 2012, 5).

6.1 ETL-prosessi

ETL tulee sanoista Extract, Transfer ja Load eli poiminta, muokkaus ja lataus. Tiedot luetaan joko suoraan operatiivisista järjestelmistä tai niitä tuotetuista siirtotiedostoista. Tämän jälkeen ne muokataan tietovarastoon sopivaan muotoon ja lopuksi kirjoitetaan tietovarastoon. Muokausvaiheessa voi olla useita vaiheita, kuten virheiden tarkistus, muunnokset ja tietojen historiointi. (Hovi ym. 2009, 48.)

Yleensä tiedot ladataan ja jalostetaan eräajoina yöaikaan, jotta ne olisivat helposti ja nopeasti käytettävissä tietojen kyselyjä varten päivisin. Prosessi toteutetaan yleensä tietokantapohjaisella eräajosovellukseen tarkoitettulla ETL-työkalulla. Alla olevassa kuviossa (Kuvio 12) on esitetty periaatteellinen tietojen kulku prosessissa. (Hovi ym. 2009, 48.)



Kuvio 12. ETL-prosessin vaiheet (Hovi ym. 2009, 48)

Vaihe 1 kuvaa raakatietojen poimintaa operatiivisten järjestelmien tietokannoista tai siirtotiedostoista sekä tietojen muokkaamista ja lataamista tietovarastoon. Vaiheessa 2 muodostetaan summataulut perustiedoista, esimerkiksi kuukausisummien laskeminen päiväkohtaisista myynneistä. Vaiheessa 3 muodostetaan mahdolliset datamartit ja kuutiot raportointia ja analysointia varten. (Hovi ym. 2009, 48.)

6.2 Henkilötietojen vienti MDS-järjestelmään

Opinnäytetyön aiheena oli tutkia MS MDS-järjestelmän soveltuvuutta henkilötietojen hallitsemiseksi Lappiassa. Tuotteen soveltuvuus kiinnostaa, koska Lappialla on käytössä MS SQL -tuotteita ja olemassa oleva käyttöoikeussopimusmalli mahdollistaa tuotteen käyttöönoton joustavasti.

Lähtötietoina tässä käytettiin kahta tekstimuotoista siirtotiedostoa. Toinen tiedosto tulee henkilöstöhallinnon tietojärjestelmästä ja toinen puhelinvaihdejärjestelmästä. Henkilöstöhallinnon järjestelmästä tuleva tiedosto on alun perin toteutettu Reportronic-työajanhallintajärjestelmän tarpeisiin. Tiedosto muodostetaan lähdejärjestelmässä ajastetusti kerran vuorokaudessa. Puhelinvaihdejärjestelmästä saatava saadaan myös ajastetusti kerran vuorokaudessa. Tiedostomuotona molemmissa on yleisesti käytetty CSV (Comma Separated Value).

Henkilötietoja esiintyy useissa operatiivisissa järjestelmissä, joten niiden tiedot ovat tarkoituksen mukaista pitää samanlaisina ja ajan tasalla. Yhtenä tärkeimpänä tavoitteena on saada siirrettyä tiedot henkilöstöhallinnon järjestelmästä käyttäjähallinnon järjestelmään, jolla perustetaan käyttäjille verkkotunnukset ja sähköpostilaatikat. Myös käyttäjätilien luokitsemiset työsuhteiden päättyessä ovat olennainen osa käyttäjähallintoa sekä tietoturva- ja politiikkaa. Tätä varten tarvitaan tieto työsuhteiden päättymispäivämääristä.

Organisaation käytössä on käyttäjätilien hallintaan Microsoft FIM -järjestelmä, jota on käytetty opiskelijoiden käyttäjätilien hallintaan jo aiemmin. Lähtötiedot poimitaan opetushallinnon järjestelmistä (opiskelijarekisterit). Henkilöstön käyttäjätilit on sen sijaan perustettu manuaalisesti käyttäjähakemistoon (AD), jolloin tietojen kulku ei ole ollut ajan tasasta työsuhteissa tapahtuneiden muutosten kanssa. MDS-järjestelmää on tarkoitus käyttää FIM-järjestelmässä henkilötietojen lähdejärjestelmänä, jolloin AD-käyttäjätilit saataisiin pysymään ajan tasalla työsuhtetietojen mukaisesti.

Esimerkkitiedoissa on käytetyt nimi-, tunnus- ja numerotiedot ovat kuvitteellisia. Osa lähdejärjestelmistä saatavista tiedoista on jätetty pois, koska ne eivät ole tässä käytetyissä esimerkeissä oleellisia tietoja. Tarkoituksena on saada yhdistettyä edellä mainitut tiedot yhdeksi Master Data -tietueeksi, joka viedään käyttäjähallinnan järjestelmään. Henkilös-

töhallinnon järjestelmästä tulevan siirtoaineiston sisältö muodostuu siten, että siihen poimitaan henkilötiedot \pm 60 vrk työsuhteen alkamisesta tai päättymisestä. Se ei sisällä täten työsuhteiden historiatietoja. Sen sijaan kohdejärjestelmässä (MDS) ei poisteta tietoja, vaan ainoastaan lisätään tai päivitetään tiedot. Täten historiatiedot saadaan tallennettua Master Data -järjestelmään.

Koska järjestelmät ovat usein suljettuja järjestelmiä, on niihin hankala viedä tietoja automatisoidusti. Master Data -järjestelmästä saadaan kuitenkin näissäkin tapauksissa tarjottua tarvittavat tiedot tietokantanäkymillä tai raporteilla, joiden käyttöä voidaan sallia tai rajoittaa käyttöoikeuksilla.

Testiympäristönä opinnäytetyössä käytettiin alla olevan taulukon (Taulukko 3) mukaista kokoonpanoa.

Taulukko 3. Testiympäristön kokoonpano

Palvelin:	Dell PowerEdge 2850
Käyttöjärjestelmä:	Windows Server 2012 Standard
Tietokannan hallintajärjestelmä:	SQL Server 2012 Enterprise (MDS, DQS, SSDT)
Taulukkolaskentaohjelmisto:	Microsoft Excel 2010 + Master Data Service Add-in fir Microsoft Excel

6.3 MDS-järjestelmän valmistelu

MDS, DQS ja SSIS ovat valinnaisia ominaisuuksia SQL Server Enterprise -tuotteen asennuksessa. Ominaisuudet voidaan asentaa perusasennuksen yhteydessä tai jälkeensä lisäämällä ominaisuuksia olemassa olevaan SQL-palvelimen instanssiin (Microsoft MSDN www-sivut 2013, hakupäivä 5.3.2013). Perusasennuksella tarkoitetaan tässä Database Engine Services -ominaisuutta. Microsoft SQL Server Management Studio on ohjelmisto, jolla hallitaan tietokantajärjestelmää. Se sisältyy Management Tools -ominaisuuteen, joka asennetaan tyypillisesti myös perusasennuksen yhteydessä.

DQS- ja MDS-ominaisuus edellyttävät asennuksen jälkeistä erillistä määrittelyä. Määrittelyiden yhteydessä muun muassa perustetaan palveluille omat tietokannat. Tässä yhteydessä ei käsitellä tarkemmin asennuksiin liittyviä toimenpiteitä.

Tietokantojen hallinnointi vaatii järjestelmään pääkäyttäjätasoiset oikeudet, jotka määritellään roolipohjaisesti. Järjestelmän sisään rakennettu Sysadmin-rooli mahdollistaa korkeimman tason oikeudet tietokantojen hallinnointiin. Kaikki jatkossa mainitut SQL-komennot suoritettiin Management Tools -ohjelmiston SQL Editorilla.

Testiympäristössä käytettiin Lappia_HR-nimistä tietokantaa, joka perustettiin komenolla CREATE DATABASE [Lappia_HR].

Tietojen siirtoprosessissa perustettiin ensin extract-vaiheessa tarvittavat siirtotiedostoja vastaavat kohdetaulut SQL-järjestelmään. Henkilötietoja sisältävä taulu on nimeltään T_LappiaHlo_I ja vastaavasti puhelintietoja sisältävä taulu on nimeltään T_LappiaPuh_I.

Taulut perustettiin SQL-kielen CREATE TABLE -komennoilla.

```
CREATE TABLE [dbo].[T_LappiaHlo_I](
    [Henkilonumero] [varchar](7) NULL,
    [Henkilotunnus] [varchar](11) NULL,
    ...
CREATE TABLE [dbo].[T_LappiaPuh_I](
    [ID] [varchar](50) NULL,
    [Luoja] [varchar](50) NULL,
    ...
```

Seuraavaksi perustettiin transfer-vaiheen taulut T_LappiaPuh_W ja T_LappiaHlo_W.

```
CREATE TABLE [dbo].[T_LappiaPuh_W](  
    [Sukunimi] [varchar](100) NULL,  
    [Etunimi] [varchar](100) NULL,  
    ...
```

```
CREATE TABLE [dbo].[T_LappiaHlo_W](  
    [Henkilonumero] [varchar](50) NULL,  
    [Koodi] [varchar](50) NULL,  
    ...
```

Komennoissa annettiin taulun nimi, kenttien nimet, - tietotyypit ja – pituudet. Kenttien oletusarvona on NULL, eli määrittelemätön sisältö. Täydelliset listaukset taulujen rakenteesta ja komennoista ovat liitteessä 3.

7 SSIS-PROJEKTI

ETL-prosessi toteutetaan SQL-serverin Data Tools (SSDT) -komponenttiin sisältyvällä Microsoft Visual Studio – kehitystyökalulla. Kehitysympäristö sisältää valmiita mallipohjia (template) Business Intelligence -prosessien toteuttamiseen. ELT-prosessin toteuttamiseen valitaan mallipohjaksi Integration Services Project. Mallipohjassa on mukana työkaluvalikoima (Toolbox), jolla saadaan toteutettua kyseiseen projektiin liittyvät toiminnot. Kehitystyökalun käyttöliittymässä on SSIS Toolbox -valikko, jonka sisältö vaihtuu kohteen mukaan. Tarvittavat komponentit vedetään Designer-näkymään kulloisenkin tilanteen mukaisesti. ETL-prosessin kuvaus ja toimintaperiaate näkyvät liitteessä 4.

Extract-vaihe, lähdetietojen tuonti tietokantaan

1. Tyhjennetään import-taulujen sisällöt aiemmista tiedoista (Execute SQL Task).
2. Kopioidaan lähdetiedot sellaisenaan tekstitiedostosta tietokannan tauluun (Data Flow Task).
3. Mikäli extract-vaiheessa tapahtuu virhe, keskeytyy prosessi ja tapahtumasta lähetetään sähköpostiviesti, muussa tapauksessa jatketaan seuraavaan vaiheeseen (Send Mail Task).

Transform-vaihe, muokataan ja suodatetaan lähtötietoja tarpeiseen sopivaksi

4. Viedään tiedot Work-tauluihin (Data Flow Task)
 - Lisätään puuttuvat tietueet.
 - Päivitetään muuttuneet tietueet.
5. Mikäli Transform-vaiheessa tapahtuu virhe, keskeytyy prosessi ja tapahtumasta lähetetään sähköpostiviesti, muussa tapauksessa jatketaan seuraavaan vaiheeseen (Send Mail Task).

Load-vaihe, viedään tiedot MDS-järjestelmään

6. Tyhjennetään MDS Staging-taulun sisältö (Execute SQL Task).
7. Kopioidaan yhdistetyt tiedot Staging-tauluun (Data Flow Task).
8. Ladataan Staging-taulun tiedot MDS-järjestelmän LappiaHenkilo_Leaf-tauluun (Execute SQL Task).
9. Mikäli extract-vaiheessa tapahtuu virhe, keskeytyy prosessi ja tapahtumasta lähetetään sähköpostiviesti, muussa tapauksessa prosessin suoritus päättyy (Send Mail Task).

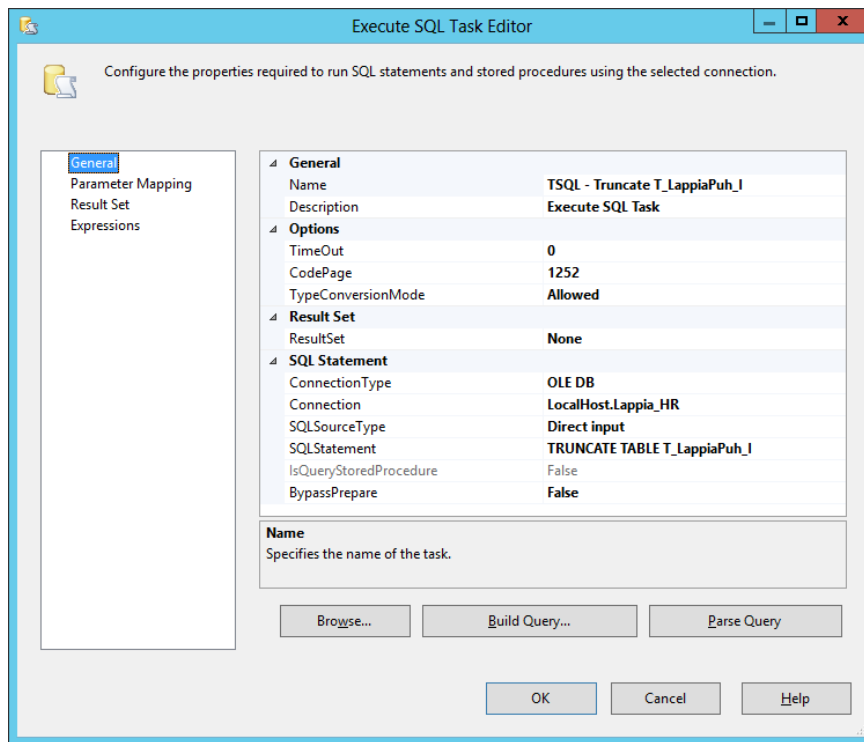
Lähtötietoina tässä prosessissa on kaksi tekstitiedostoa, jotka on tuotettu operatiivisissa järjestelmissä ns. PUSH-tekniikalla. Siirtotiedostojen liittymät ovat järjestelmätoimittajien toteuttamia asiakaan tilauksesta. Tämä on yleinen käytäntö kaupallisissa sovelluksissa, koska järjestelmien toimittajat eivät yleensä luovuta asiakkailleen kuvauksia omista järjestelmistään. Täten tietoja ei pääse hakemaan suoraan lähdetietokannoista esimerkiksi SQL-kyselyillä.

Tiedostot muodostetaan lähdejärjestelmistä kerran vuorokaudessa siten, että uusi aineisto korvaa aina kokonaan edellisen aineiston. Tiedostot muodostetaan lähdejärjestelmän tiedostojärjestelmään, josta ne haetaan FTP-tiedonsiirto-ohjelmalla käyttökohteeseen. Tässä prosessikuvauksessa ei ole kuvattu kyseistä vaihetta, vaan lähdetään siitä tilanteesta, että tiedostot ovat MDS-palvelimen tiedostojärjestelmässä.

Prosessi on jaettu tässä tapauksessa jaksoihin (sequence), jolloin prosessista on saatu selkeämpi ja hallittavampi. Jakso myös ryhmittelee prosessin toiminnot siten, että kaikki jakson sisällä olevat tehtävät suoritetaan ennen seuraavaan jaksoon siirtymistä.

7.1 Extract-vaihe

Extract-vaiheessa tyhjennetään ensin kohdetaulujen sisältö ”Execute SQL Task” -komponentilla. Tietokantaan kohdistuva toiminto määritellään ”Execute SQL Task Editorilla” antamalla komponentin ominaisuuksissa ”SQLStatement” arvoksi SQL-komento (Kuvio 13).

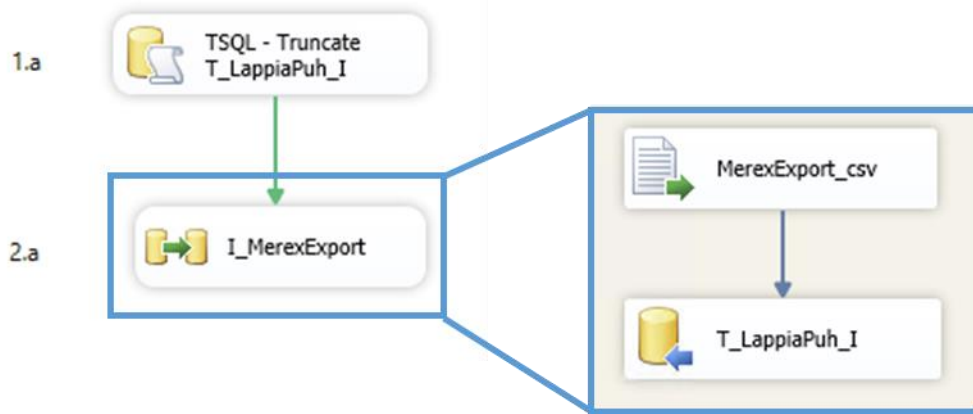


Kuvio 13. Execute SQL Task Editor

Truncate-komennolla tyhjennetään kohdetaulun sisältö kerralla kokonaan. Vastaava lopputulos saataisiin DELETE FROM ”taulu” -komennolla. Tämä on kuitenkin hieman hitaampi, koska se käsittelee taulun rivi kerrallaan. SQL-komennot ovat tässä seuraavat.

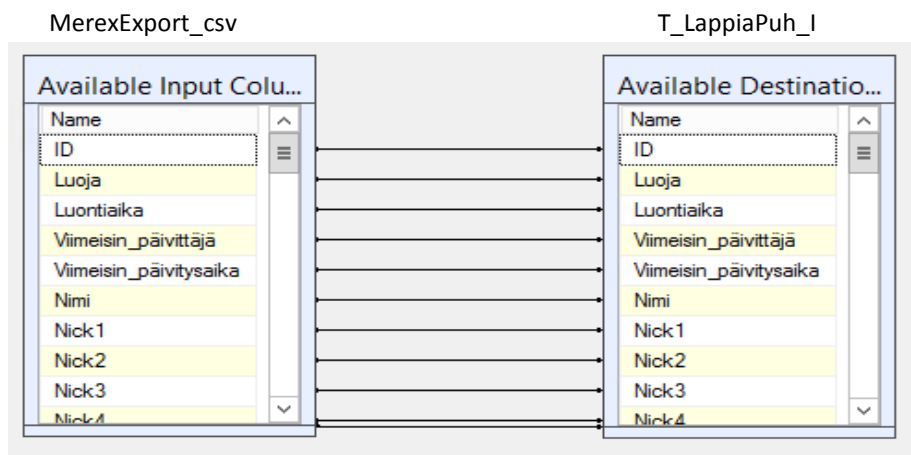
- TRUNCATE TABLE T_LappiaPuh_I (1.a)
- TRUNCATE TABLE T_LappiaHlo_I (1.b)

Tämän jälkeen kopioidaan tekstitiedostojen sisällöt SQL-tietokannan tauluihin. Puhelinvaiheesta saadut tiedot viedään I_merexExport -”Data Flow” -tehtävällä (2.a, Kuvio 14). Tekstitiedostojen ja tietokantataulujen kentät vastaavat toisiaan ja tiedot kopioidaan tauluihin sellaisenaan merkkijonoina. Kuvio 15 havainnollistaa tekstitiedoston ja tietokantataulun kenttien vastaavuudet.



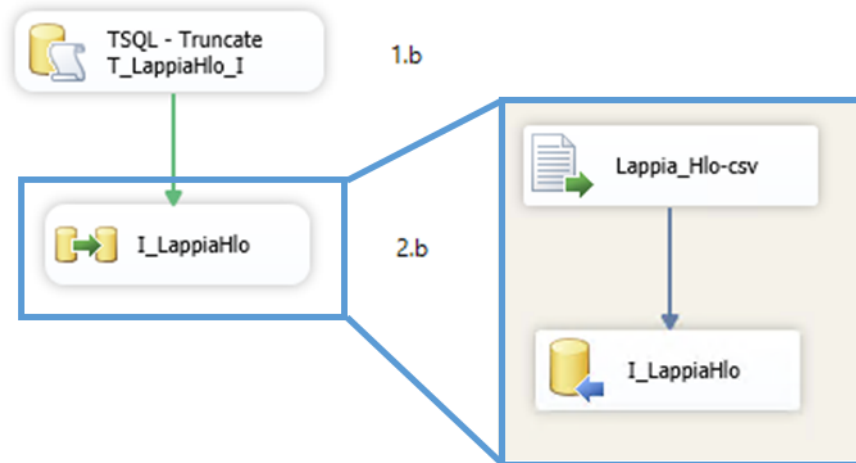
Kuvio 14. I_MerexExport Data Flow Task

MerexExport_csv-komponentissa määritellään tietolähde, joka on tässä ”Flat File Source” -tyyppinen. Tietolähteelle on tehty ”Connection Manager”, jonka ominaisuudet puolestaan määritellään ”Flat File Connection Manager Editorilla” liitteen 5 mukaisesti. Myös kohteelle on tehty ”Connection Manager”, joka on tyypiltään ”OLE DB Destination” -komponentti. Tämän määrittelyt näkyvät liitteessä 6.



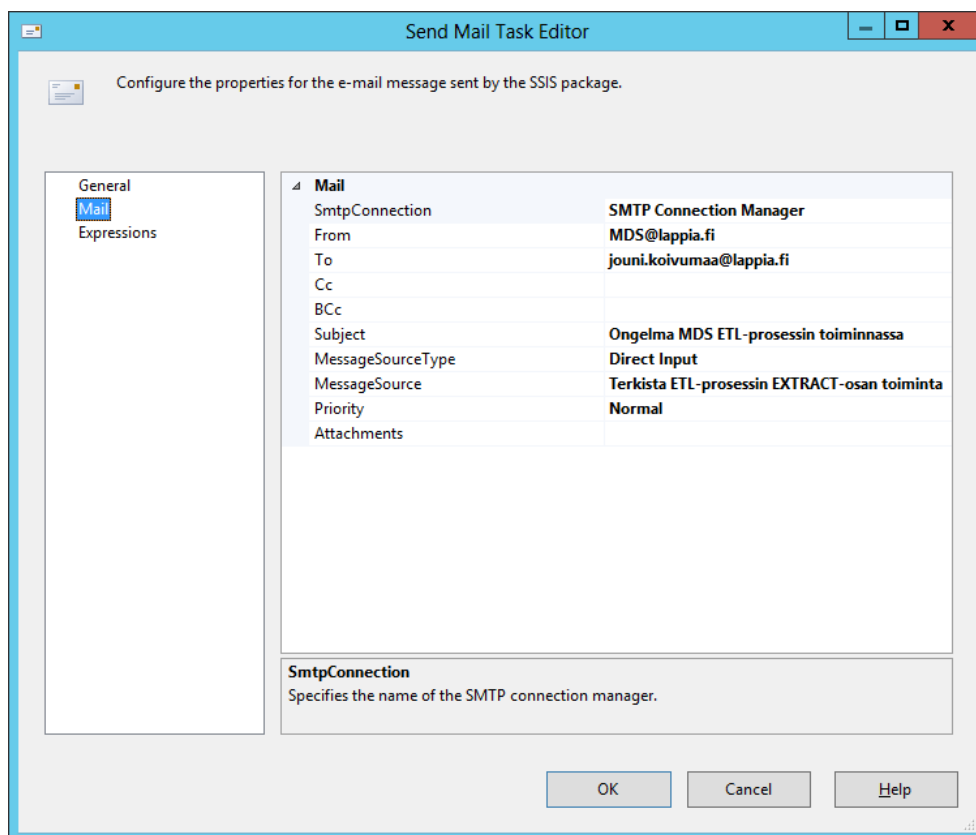
Kuvio 15. Tekstiedoston ja tietokantataulun kenttien vastaavuudet

Henkilöstöhallinnosta saadut tiedot viedään vastaavasti tietokannan tauluun ”I_LappiaHlo” – ”Data Flow Taskilla” (2.b, Kuvio 16). Lappia_Hlo-csv-komponentissa määritellään tietolähde, joka on myös tässä ”Flat File Source” -tyyppinen. Tietolähteelle on tehty ”Connection Manager”, jonka ominaisuudet puolestaan määritellään ”Flat File Connection Editorilla” liitteen 7 mukaisesti. Kohteelle käytetään edellä tehtyä ”OLE DB Connection Manageria” (liite 6).



Kuvio 16. I_LappiaHlo Data Flow Task

Mikäli Extract- vaiheessa tulee jokin virhe, ohjautuu prosessi E_virheilmoitus-tehtävään, jonka jälkeen prosessin suoritus keskeytyy. Muussa tapauksessa prosessi jatkaa Transfer-vaiheeseen. E_virheilmoitus on ”Send Mail Task”, jonka määrittymiset näkyvät alla (Kuvio 17).



Kuvio 17. Send Mail Task

7.2 Transfer-vaihe

Transfer-vaiheessa päivitetään tiedot ns. Work-tauluihin T_LappiaPuh_W ja T_LappiaHlo_W. Tietoja ei lueta suoraan edellisistä tauluista, vaan niistä on muodostettu erilliset näkymät (view), joissa muokataan valmiiksi osa tiedoista siirtoa varten. SQL-kyselelyssä on käytetty CASE-, LEFT-, RIGHT-, SUBSTRING-, CHARINDEX- ja REVERSE-funktioita.

Puhelinvaihteesta saatavat tiedot poimitaan SQL-kyselyllä ja muotoillaan seuraavasti (V_LappiaPuh_IW liite 8):

- Nimi-kentässä oleva ”Sukunimi Etunimi” – merkkijono puretaan erillisiksi sukunimi- ja etunimi-merkkijonoiksi ja sijoitetaan ne omiin sarakkeisiin.
- Alanumeroon, joka on lyhytvalintanumero, lisätään eteen kuntayhtymän ulkoverkkotunnus 010 383.
- Matkapuhelinnumero muokataan merkkijonosta 040-1234567 muotoon 040 123 4567.
- Muodostetaan toinen matkapuhelinnumero, joka on kansainvälisessä muodossa +358 40 123 4567.

Puhelintietojen siirto tehdään T_LappiaPuh_I to T_LappiaPuh_W – ”Taskilla” (Kuvio 18). Lähteen ja kohteen tietoja verrataan Lookup-komponentilla. Tietojen vertailussa käytetään henkilönumero-kenttää. Mikäli vastaavaa henkilönumeroa ei löydy kohteesta, lisätään tietue T_LappiaPuh_W-tauluun. Mikäli henkilönumero vastaava henkilönumero löytyy, päivitetään nimi- tai numerotiedot kyseiseen tauluun alla olevalla UPDATE-lauseella.

```
UPDATE [dbo].[T_LappiaPuh_W]
```

```
  SET [Sukunimi] = ?
```

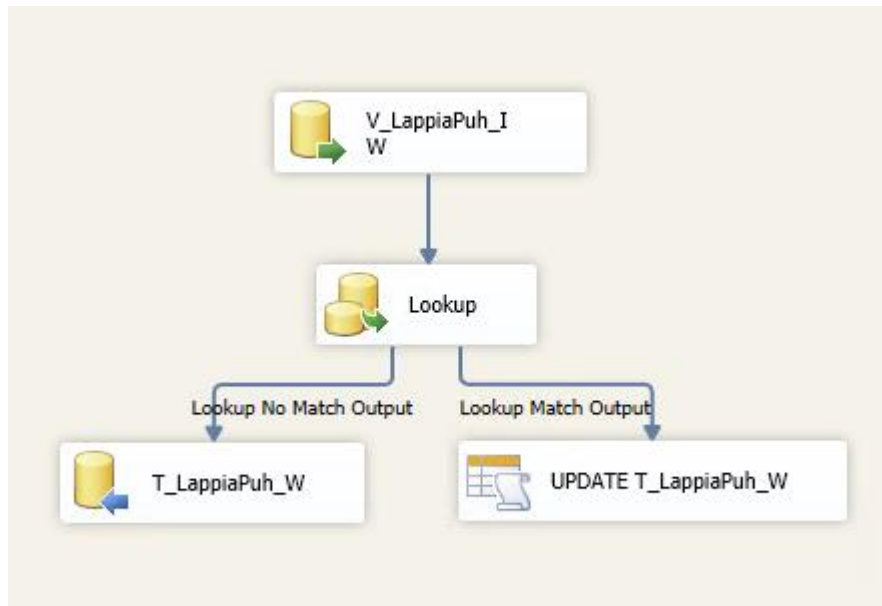
```
    ,[Etunimi] = ?
```

```
    ,[Matkapuhelin] = ?
```

```
    ,[Matkapuhelin2] = ?
```

```
    ,[Puhelin] = ?
```

```
WHERE [Henkilönumero] = ?
```



Kuvio 18. T_LappiaPuh_I to T_LappiaPuh_W Task

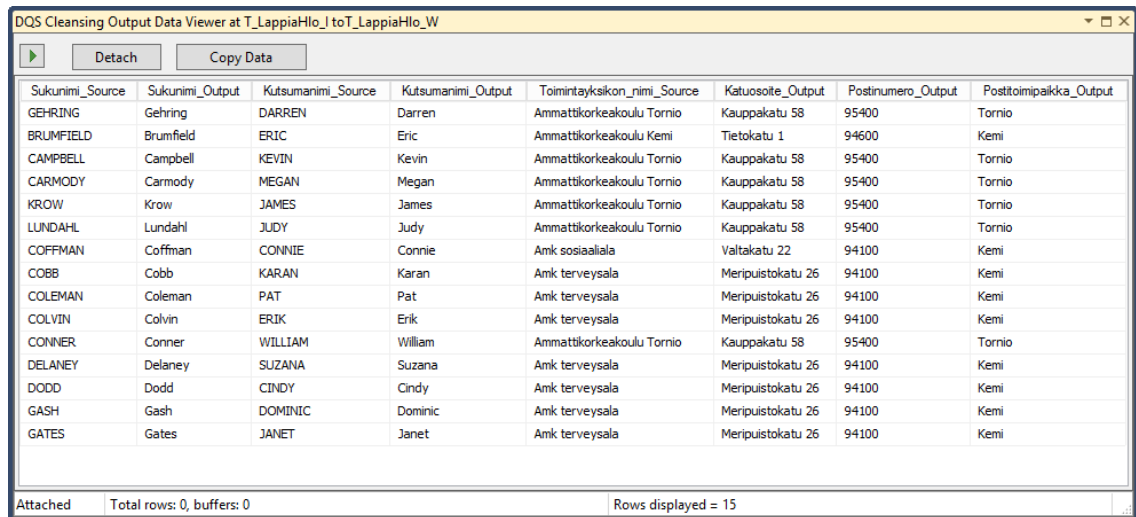
Henkilöstöhallinnon järjestelmästä saatavat tiedot poimitaan SQL-näkymään V_LappiaHlo_IW (liite 9), jossa ne muotoillaan seuraavasti.

- Henkilötunnus salataan MD5-algoritmilla, jotta selväkielistä henkilötunnusta ei tarvitse käyttää yksilöivänä tietona järjestelmissä (IETF www-sivut, hakupäivä 15.9.2013).
- Kutsumanimeksi poimitaan etunimet-kentästä ensimmäinen nimi. Mikäli kentässä on vain yksi nimi, poimitaan se sellaisenaan.
- Organisaatio-kentän arvo poimitaan yksikkönimen alkuosan perusteella. Arvona voi olla AO (ammattipisto), AMK (ammattikorkeakoulu) tai KY (kuntayhtymä).
- Osasto-kentän arvoksi poimitaan yksikkönimen loppuosa.
- Osoite-kentät muodostetaan NULL-arvoilla, jotka täytetään myöhemmin DQS-komponentin avulla.

Lähdeaineistossa voi olla samasta henkilöstä useampi tietue (rivi), joten tässä poimitaan tietueet vain kertaalleen. Tietueet ovat yksilöllisiä, vaikka sama henkilö voi esiintyä tietueissa useampaan kertaan. Esimerkiksi määräaikaisen työsuhteen jatkuessa tulee uusi rivi siirtoaineistoon uudesta työsuhteesta johtuen. Kuten aiemmin mainittiin, kerätään siirtoaineistoon tiedot ± 60 vrk työsuhteen alkamisesta ja loppumisesta. Poiminta tehdään

työsuhtetietojen päivämäärien perusteella vertaamalla niitä nykyhetkeen. Poiminnassa käytetään SQL:n MIN, MAX, CONVERT ja GETDATE-funktioita.

Seuraavaksi henkilötiedot viedään DQS-järjestelmään, jossa muotoillaan nimitiedot isoille alkukirjaimille (Capitalize) ja muodostetaan toimintayksikön nimen avulla katuosoite. DQS-komponentissa määritellään Data quality connection manager ja Data Quality Knowledge Base (liite 10). Kyseisten määritysten jälkeen DQS Cleansing Transformation Editorissa näkyy käytettävissä olevat määreet. DQS:n puolella attribuutti-käsitettä vastaa domain-käsite. Kyseinen DQS-komponentti toimii alla olevan esimerkin (Kuvio 19) mukaisesti.



DQS Cleansing Output Data Viewer at T_LappiaHlo_I to T_LappiaHlo_W

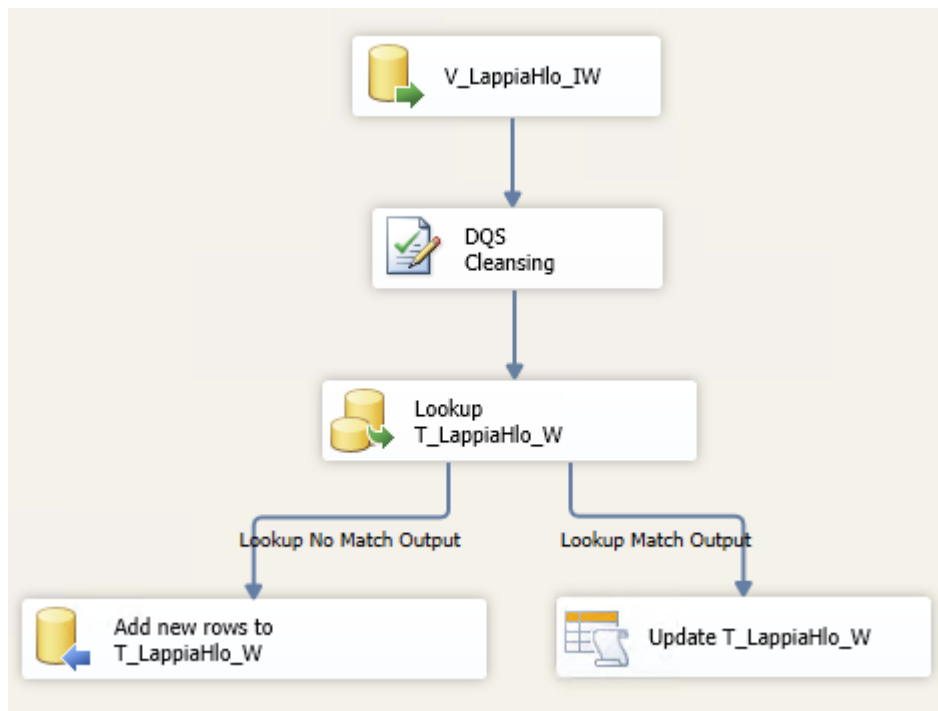
Detach Copy Data

Sukunimi_Source	Sukunimi_Output	Kutsumanimi_Source	Kutsumanimi_Output	Toimintayksikon_nimi_Source	Katuosoite_Output	Postinumero_Output	Postitoimipaikka_Output
GEHRING	Gehring	DARREN	Darren	Ammattikorkeakoulu Tornio	Kauppakatu 58	95400	Tornio
BRUMFIELD	Brumfield	ERIC	Eric	Ammattikorkeakoulu Kemi	Tietokatu 1	94600	Kemi
CAMPBELL	Campbell	KEVIN	Kevin	Ammattikorkeakoulu Tornio	Kauppakatu 58	95400	Tornio
CARMODY	Carmody	MEGAN	Megan	Ammattikorkeakoulu Tornio	Kauppakatu 58	95400	Tornio
KROW	Krow	JAMES	James	Ammattikorkeakoulu Tornio	Kauppakatu 58	95400	Tornio
LUNDAHL	Lundahl	JUDY	Judy	Ammattikorkeakoulu Tornio	Kauppakatu 58	95400	Tornio
COFFMAN	Coffman	CONNIE	Connie	Amk sosiaalityö	Valtakatu 22	94100	Kemi
COBB	Cobb	KARAN	Karan	Amk terveysala	Meripuistokatu 26	94100	Kemi
COLEMAN	Coleman	PAT	Pat	Amk terveysala	Meripuistokatu 26	94100	Kemi
COLVIN	Colvin	ERIK	Erik	Amk terveysala	Meripuistokatu 26	94100	Kemi
CONNER	Conner	WILLIAM	William	Ammattikorkeakoulu Tornio	Kauppakatu 58	95400	Tornio
DELANEY	Delaney	SUZANA	Suzana	Amk terveysala	Meripuistokatu 26	94100	Kemi
DODD	Dodd	CINDY	Cindy	Amk terveysala	Meripuistokatu 26	94100	Kemi
GASH	Gash	DOMINIC	Dominic	Amk terveysala	Meripuistokatu 26	94100	Kemi
GATES	Gates	JANET	Janet	Amk terveysala	Meripuistokatu 26	94100	Kemi

Attached Total rows: 0, buffers: 0 Rows displayed = 15

Kuvio 19. DQS input ja output

Henkilötietojen siirto tehdään T_LappiaHlo_I to T_LappiaHlo_W -"Taskilla" (Kuvio 20). DQS:n jälkeen tiedot lisätään tai päivitetään T_LappiaHlo_W-tauluun vastaavalla tavalla, kuin edellä tehty puhelintietojen päivittäminen.



Kuvio 20. T_LappiaHlo_I to T_LappiaHlo_W Task

Mikäli Transfer-vaiheessa tulee jokin virhe, ohjautuu prosessi Virheilmoitus_T-tehtävään, jonka jälkeen prosessin suoritus keskeytyy. Muussa tapauksessa prosessi jatkaa LOAD-vaiheeseen. Virheilmoitus_T on vastaavanlainen Send Mail Task, kuin aiemmassa extract-vaiheessa.

7.3 Load-vaihe

Load-vaiheessa viedään kahdesta tietolähteestä yhdistetyt tiedot MDS-tietokantaan. Tiedot yhdistetään ensin V_LappiaHlo_O -SQL-näkymällä (liite 11). Yhdistämisessä otetaan kaikki tietueet T_LappiaHlo_W -taulusta ja tietueisiin yhdistetään sarakkeet T_LappiaPuh_W -taulusta, mikäli henkilönnumero, sukunimi ja etunimi ovat samat. Ehto määritellään SELECT-lauseen FROM-osassa seuraavasti.

```
SELECT
```

```
...
```

```
FROM
```

```
dbo.T_LappiaHlo_W LEFT OUTER JOIN dbo.T_LappiaPuh_W
```

```
ON dbo.T_LappiaHlo_W.Henkilonumero = dbo.T_LappiaPuh_W.Henkilonumero
```

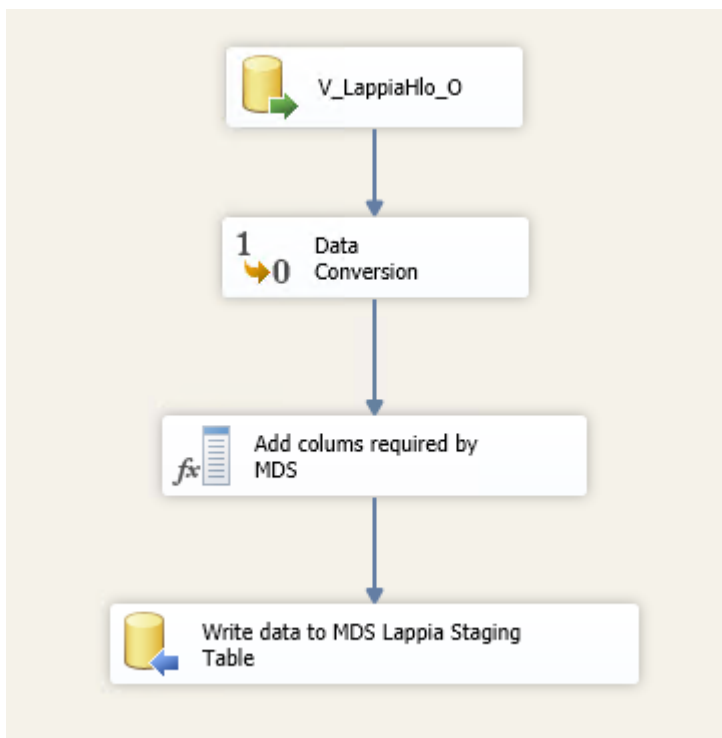
```
AND dbo.T_LappiaHlo_W.Sukunimi = dbo.T_LappiaPuh_W.Sukunimi
```

```
AND dbo.T_LappiaHlo_W.Kutsumanimi = dbo.T_LappiaPuh_W.Etunimi
```

LEFT OUTER JOIN määrittelee, että T_LappiaHlo-W-taulusta otetaan kaikki rivit. AND-operaattoreilla määritellään, että kaikki kolme lueteltua ehtoa tulee täyttyä. Mikäli vastaavuutta ei löydy, jäävät puhelintiedot kyseisillä riveillä tyhjiksi. Nimitietojen lisäksi ehtoihin on määritelty henkilönnumero, koska organisaatiossa voi olla samannimisiä henkilöitä. Ilman henkilönnumeroehtoa tiedot yhdistettäisiin järjestyksessä ensimmäisen henkilön mukaan ja toisen kohdalla tulisi virhetilanne. Toteutus edellyttää, että henkilönumerot ovat tallennettu puhelinvaihejärjestelmään.

Load-vaiheessa tyhjennetään ensi kohdetaulu stg_LappiaHenkilo_leaf. Tämä on MDS-järjestelmän muodostama taulu, joka määräytyy LappiaHenkilo-nimisestä mallista. Taulun tyhjennys tehdään SQL Taskin TRUNCATE -lauseella, TRUNCATE TABLE stg.LappiaHenkilo_Leaf.

Seuraavaksi kopioidaan tietueet näkymästä tauluun I_LappiaHlo_O to stg_LappiaHenkilo_Leaf -Data Flow -"Taskilla" (Kuvio 21).



Kuvio 21. I_LappiaHlo_O to stg_LappiaHenkilo_Leaf -Data Flow Task

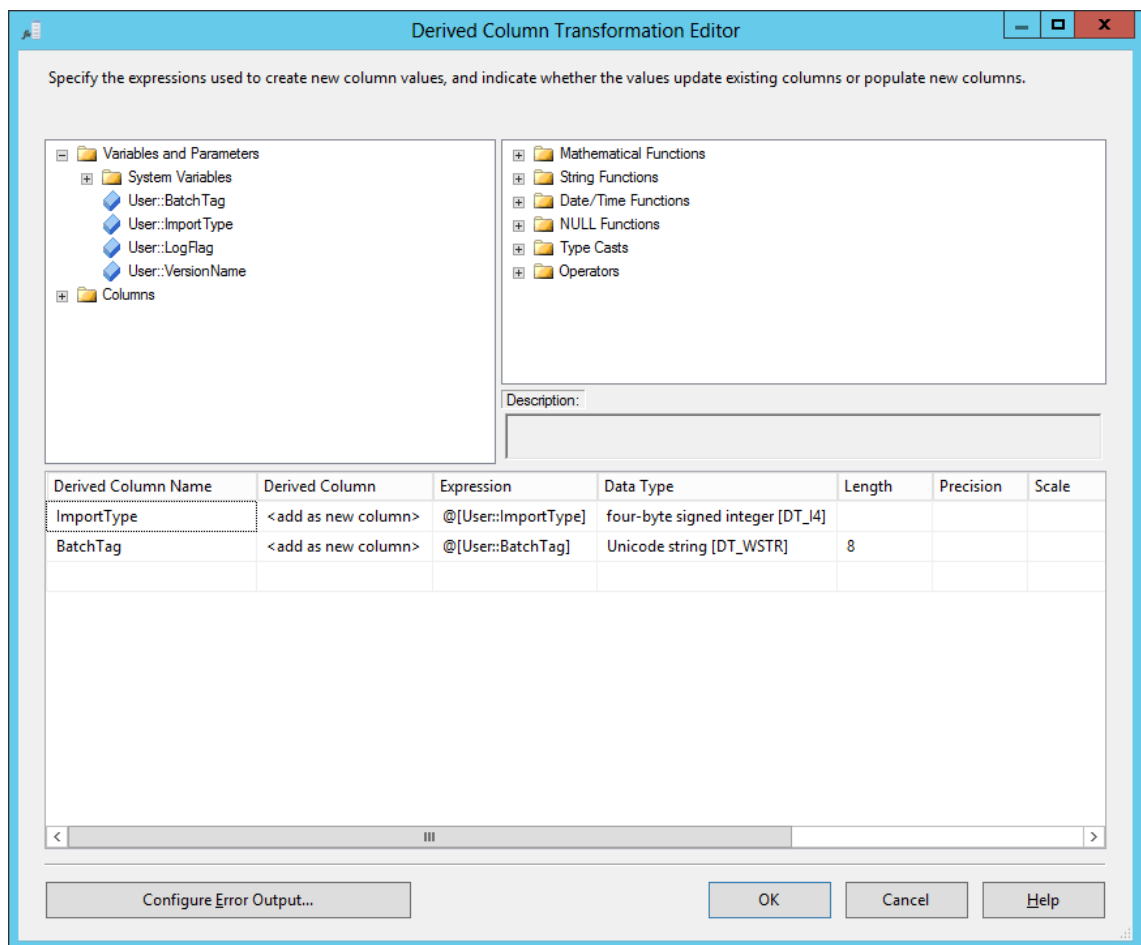
Tehtävässä haetaan tiedot V_LappiaHlo_O-näkymästä ja tehdään niille seuraavaksi tietotyypin muunnos (Data Conversion). MDS tukee vain unicode-merkistöä (DT-WSTR)

ja tässä yhteydessä lähdetieto on Suomessa yleisesti käytettävän 1252-merkistön mukainen (Korpela 2002, hakupäivä 17.11.2013). Konversio-modulissa määritellään muunneltavat kentät ja niille alias-nimet sekä tietotyyppi (Kuvio 22).

Input Column	Output Alias	Data Type	Length
Henkilonumero	Henkilonumero_DCT	Unicode string [DT_WSTR]	50
Koodi	Koodi_DCT	Unicode string [DT_WSTR]	50
Sukunimi	Sukunimi_DCT	Unicode string [DT_WSTR]	50
Etunimet	Etunimet_DCT	Unicode string [DT_WSTR]	50
Kutsumanimi	Kutsumanimi_DCT	Unicode string [DT_WSTR]	50
Ammattinimike	Ammattinimike_DCT	Unicode string [DT_WSTR]	50

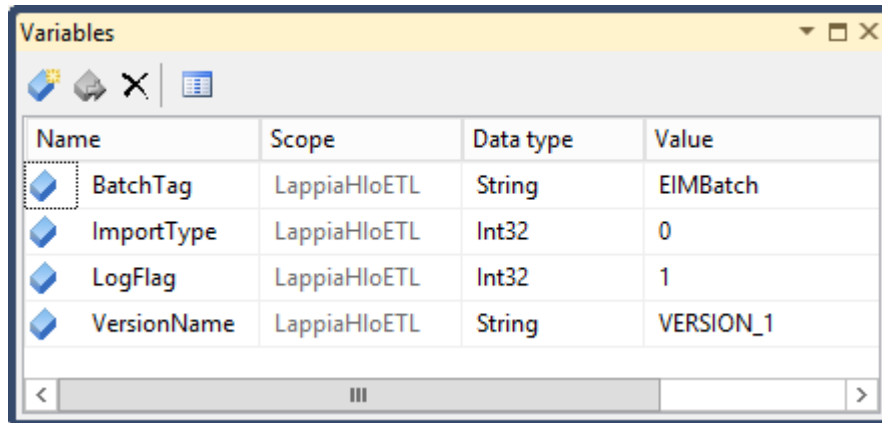
Kuvio 22. Data Conversion

Seuraavaksi lisätään MDS:n vaatimat kaksi saraketta, ImportType ja BatchTag, joiden määritykset näkyvät alla (Kuvio 23).



Kuvio 23. MDS-järjestelmän vaatimat lisäkentät

Kenttiin lisättävät arvot määritellään näkyvien muuttujien (variables) avulla (Kuvio 24).



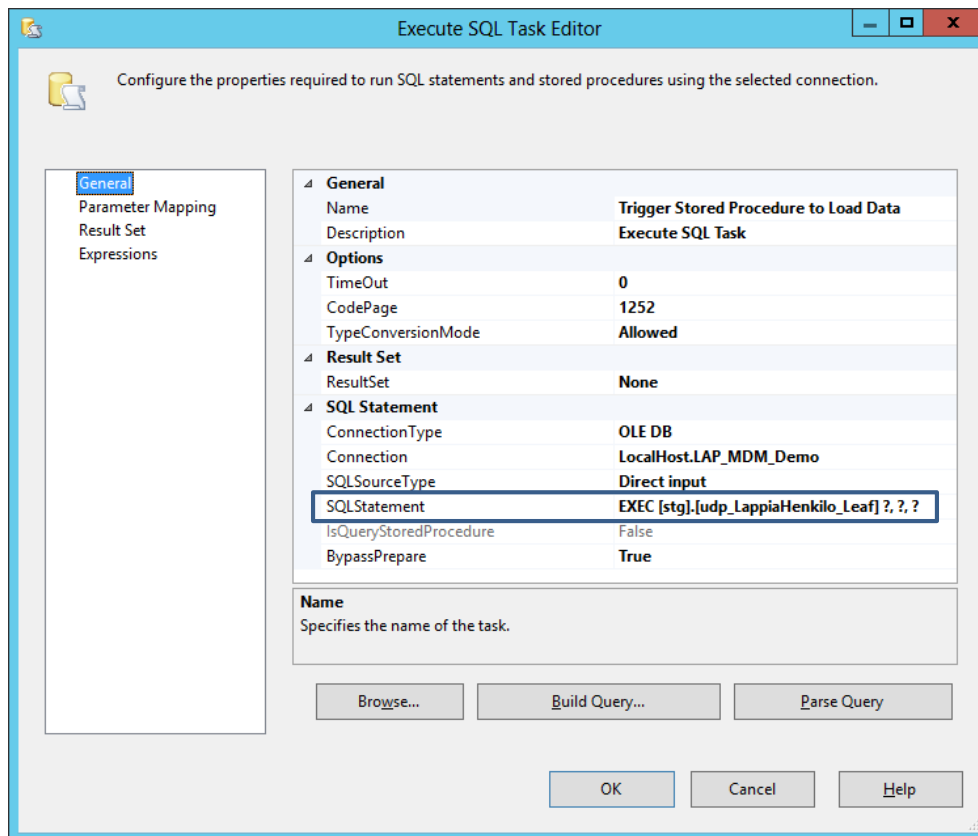
Name	Scope	Data type	Value
BatchTag	LappiaHloETL	String	EIMBatch
ImportType	LappiaHloETL	Int32	0
LogFlag	LappiaHloETL	Int32	1
VersionName	LappiaHloETL	String	VERSION_1

Kuvio 24. Muuttujien määrittelyt

BatchTag-muuttuja tarkoittaa tässä eräajon nimeä, jonka voi itse vapaasti määrittää. ImportType-muuttuja puolestaan määrittelee, miten järjestelmä käsittelee tietoa, joka löytyy ennestään MDS-tietokannasta. Arvo 0 tarkoittaa, että uudet tiedot perustetaan ja muuttuneet tiedot korvataan, ellei korvaava tieto ole NULL (Microsoft MSDN www-sivut, hakupäivä 17.11.2013).

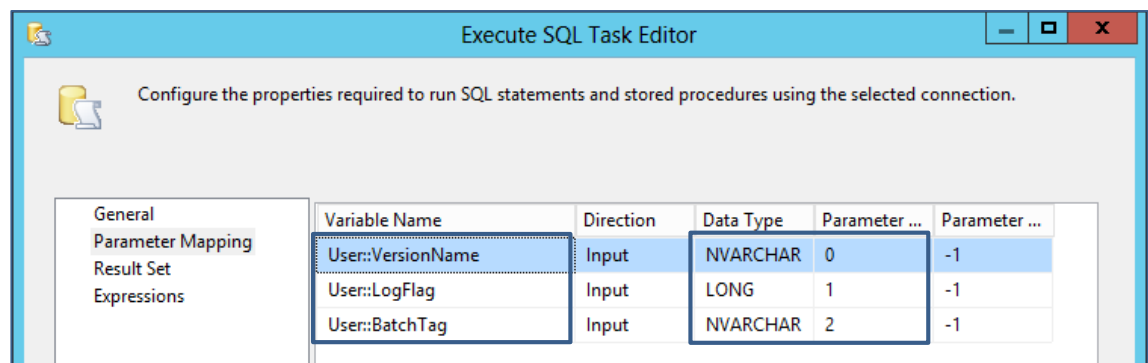
Seuraavaksi tiedot kirjoitetaan MDS:n Leaf Member Staging – tietokantatauluun, jonka MSD-järjestelmä on perustanut automaattisesti MDS-mallin määrittelyn yhteydessä. Tietovirtamäärittelyt tehdään Write data to MDS Lappia Staging -moduulissa liitteen 12 mukaisesti. Staging-taulu toimii väliaikaisvarasto (”lastauslaiturina”) tietojen lataamiselle MDS-järjestelmään. Jokaisen ajon yhteydessä taulu tyhjennetään ennen uusien tietueiden lataamista.

Lopuksi tietueet kopioidaan staging-taulusta MDS-tietokannan kohdetauluun. Toimintoon käytetään tallennettua proseduuria, joka on kytketty kohdetauluun. Proseduurin ohjelmakoodia ei kirjoiteta itse, vaan MDS-järjestelmä on muodostanut sen automaattisesti mallin tekemisen yhteydessä. Tässä yhteydessä proseduurille tarvitsee määrittää vain kaksi parametria, LofFlag ja VersionName, jotka ovat esitelty muuttujien yhteydessä kuviossa 21. Toiminto määritellään Execute SQL Task -moduulilla, jossa annetaan SQL-komennoksi EXEC [stg].[udp_LappiaHenkilo_Leaf] ?, ?, ? (Kuvio 25), jossa hakusuluissa oleva osa tarkoittaa tietokantataulun nimeä ja kysymysmerkit vastaavat tiettyjä parametreja (Kuvio 26).



Kuvio 25. Trigger Stored Procedure to Load Data Task

Parametrien määrittely tehdään Parameter Mapping -välilehdellä (Kuvio 26). Parametrien arvoina käytetään edellä määriteltyjä muuttujia, VersionName, LogFlag ja BatchTag



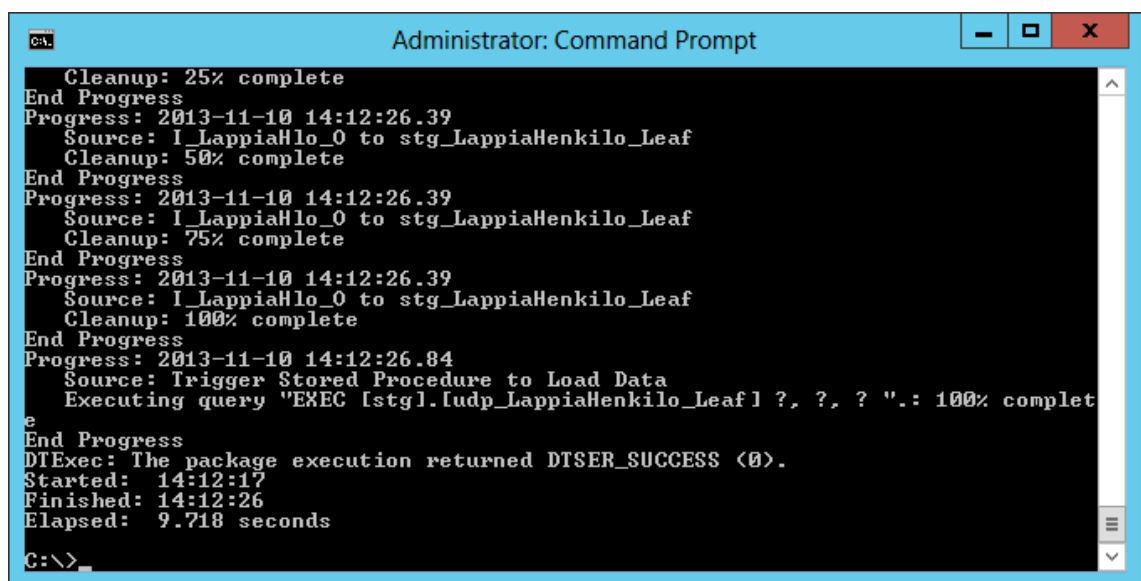
Kuvio 26. Stored Procedure - parametrit

7.4 ETL-prosessin suorittaminen

ETL-prosessin ajamista voidaan testata kehitysympäristössä valitsemalla Debug-valikosta Start Debugging. Output-ikkunassa voi seurata ajon aikaisia ilmoituksia ja jäljittää

mahdollisia virheitä. Onnistuneen ajon lopuksi Output-ikkunaan tulee rivi: SSIS package "D:\Data\Projects\Lappia_hlo_ETL\LappiaHloETL.dtsx" finished. Valmis prosessi voidaan ajaa käyttöjärjestelmän komentokehotteesta (Command Prompt) dtexec-apuohjelmalla, joka tulee SQL Server Data Tools – asennuksen mukana.

Suoritettavan komennon syntaksi on dtexec /f "tiedosto". Komentokehotteesta ajettuna näytölle tulee ajan aikaisia ilmoituksia vastaavasti, kuten kehitysympäristön Output-ikkunassa (Kuvio 27). Dtexec-komennon suorittaminen palauttaa exit-koodin (errorlevel), jota voidaan hyödyntää ajastetun tehtävän suorituksessa. Numero nolla tarkoittaa onnistunutta suoritusta. (Microsoft MSDN www-sivut, hakupäivä 17.11.2013).



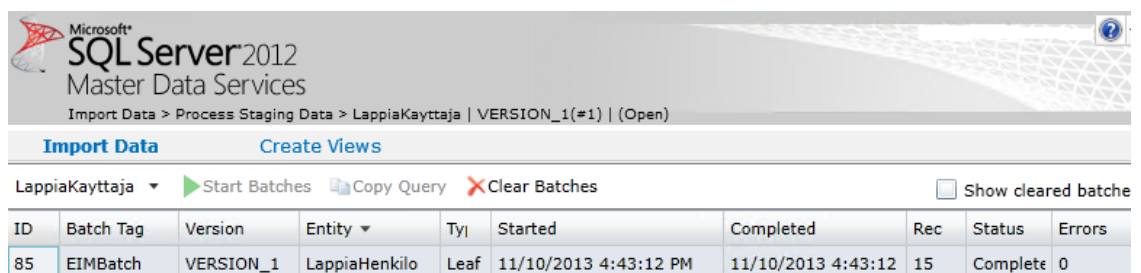
```

Administrator: Command Prompt
Cleanup: 25% complete
End Progress
Progress: 2013-11-10 14:12:26.39
Source: I_LappiaHlo_0 to stg_LappiaHenkilo_Leaf
Cleanup: 50% complete
End Progress
Progress: 2013-11-10 14:12:26.39
Source: I_LappiaHlo_0 to stg_LappiaHenkilo_Leaf
Cleanup: 75% complete
End Progress
Progress: 2013-11-10 14:12:26.39
Source: I_LappiaHlo_0 to stg_LappiaHenkilo_Leaf
Cleanup: 100% complete
End Progress
Progress: 2013-11-10 14:12:26.84
Source: Trigger Stored Procedure to Load Data
Executing query "EXEC [stg].[udp_LappiaHenkilo_Leaf] ?, ?, ? ".: 100% complete
End Progress
DTExec: The package execution returned DTExec_SUCCESS (0).
Started: 14:12:17
Finished: 14:12:26
Elapsed: 9.718 seconds
C:\>

```

Kuvio 27. Dtexec-komennon suorittaminen komentokehotteesta

Tietojen viennin voi tarkastaa MDS:n Integration Management/Import Data- näkymässä (Kuvio 28). Mikäli Status-kentän arvo on Completed, ovat tiedot luettu sisään onnistuneesti.



ID	Batch Tag	Version	Entity	Tyj	Started	Completed	Rec	Status	Errors
85	EIMBatch	VERSION_1	LappiaHenkilo	Leaf	11/10/2013 4:43:12 PM	11/10/2013 4:43:12	15	Complete	0

Kuvio 28. MDS Integration Management/Import data

MDS:n Explorer-näkymässä tuotuja tietueita voidaan tarkastella tarkemmin (Kuvio 29). Tässä esimerkissä näkyy huone-attribuutti, jota ei ole ollenkaan mukana automaattisesti tuoduissa tiedoissa. Järjestelmässä voi lisätä halutut attribuutit manuaalisesti, ellei niitä saada muista lähdejärjestelmistä. Tällöin MDS toimii ensisijaisen tietolähteenä kyseisille määritteille.

The screenshot shows the Microsoft SQL Server 2012 Master Data Services Explorer interface. The main window displays the 'LappiaHenkilo Entity' with a table of members. The 'Huone' column is highlighted in blue. The 'Details' pane on the right shows the attributes for the selected member, 'Gates Janet', with the 'Huone' attribute set to '114'.

	Name	Tyoyksikkonimi	Huone	Puhelin	Matkapuhelin
✓	Coffman Connie	AMK sosiaali ja terv	110	010 383 1257	040 123 4590
✓	Cobb Karan	AMK sosiaali ja terv	111	010 383 1256	040 123 4589
✓	Coleman Pat	AMK sosiaali ja terv	111	010 383 1258	040 123 4591
✓	Colvin Erik	AMK sosiaali ja terv	112	010 383 1259	040 123 4592
✓	Conner William	AMK kaupan ja kultt	109	010 383 1260	040 123 4593
✓	Delaney Suzana	AMK sosiaali ja terv	113	010 383 1262	040 123 4595
✓	Dodd Cindy	AMK sosiaali ja terv	114	010 383 1268	040 123 4601
✓	Gash Dominic	AMK sosiaali ja terv	113	010 383 1270	040 123 4603
✓	Gates Janet	AMK sosiaali ja terv	114	010 383 1271	040 123 4604

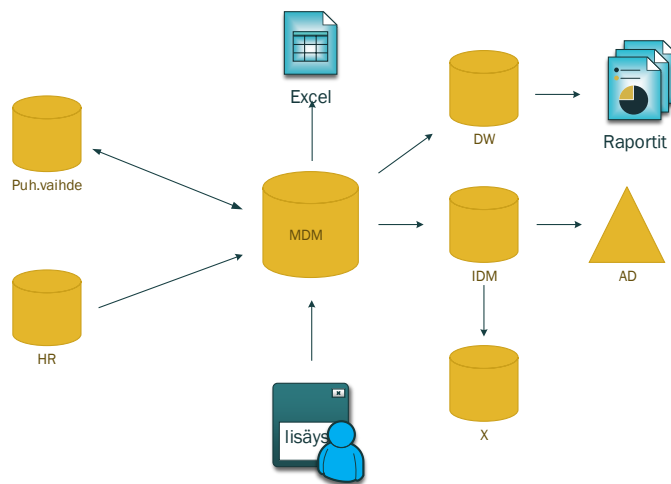
The 'Details' pane shows the following attributes for 'Gates Janet':

- Name: Gates Janet
- Code: A266B83C1BEC2DC
- Toimintayksikkonimi: Amk terveysala
- Tyoyksikkonimi: AMK sosiaali ja terveysala
- Huone: 114
- Puhelin: 010 383 1271
- Matkapuhelin: 040 123 4604
- Katuosoite: Meripuistokatu 26

Kuvio 29. MDS Explorer

7.5 Tietojen vienti MDS:sta muihin järjestelmiin

Tämä esimerkkitalaus voidaan luokitella hybrid-arkkitehtuurin mukaiseksi MDS-ratkaisuksi. Olennainen osa tiedoista tallennetaan lähdejärjestelmissä ja vain huone-attribuutti lisätään MDS-järjestelmässä. MDS-järjestelmään tallennettua ydintietoa voidaan nyt hyödyntää muissa tietojärjestelmissä. Yksi tärkeimmistä kohteista on IDM-järjestelmä, jolla hallinnoidaan mm. Windows AD:n käyttäjätilejä (Kuvio 30). Järjestelmään vietään henkilötiedoista nimitietojen lisäksi sellaiset attribuutit, joiden avulla käyttäjätilit perustetaan, aktivoidaan ja työsuhteen päättyessä suljetaan. IDM:n kautta käyttäjätietoja voidaan ohjata edelleen myös muihin järjestelmiin. MDS-järjestelmä voi toimia myös hyvänä lähdejärjestelmänä tietovarastolle (Hovi ym. 2009, 67).



Kuvio 30. Integrointi muihin järjestelmiin

MDS-tietokannassa olevasta ydintiedosta voidaan muodostaa SQL-näkymiä tai tarvittavat tiedot voidaan poimia ETL-prosessien avulla tiedostoiksi. Excel-taulukkolaskentaohjelmistoon asennettavan lisäominaisuuden avulla tiedot voidaan lukea suoraan MDS-järjestelmästä. Samalla työkalulla voidaan myös päivittää tietoja, kuten edellä kuvatun esimerkin työhuoneita. Excel soveltuu hyvin ydintietojen hallintaan myös silloin, kun MDS toimii ensisijaisena tietolähteenä muille järjestelmille.

Verkon käyttäjätilien automatisointi edellyttää prosessien ja käytäntöjen toimivuutta henkilötietojen työsuhtetietojen kirjaamisissa. Liitteessä 13 on kuvattu periaatteellinen malli uuden työsuhteen sopimiseen liittyvästä prosessista. Vastaava prosessi liittyy tietojen muuttumiseen ja työsuhteen päättymiseen, jolloin käyttäjätili suljetaan.

8 POHDINTA

Ydintietojen hallinta on kaiken kaikkiaan laaja asia. Se koskee koko organisaation toimintaa, joten se vaatii kokonaisvaltaista selvitys- ja suunnittelutyötä. Käytännössä tämä tarkoittaa omaa projektia ja siihen liittyvää tarvittavaa resursointia. Olennaisena osana tähän liittyy luonnollisesti johdon antama tuki. On myös tärkeää saada koko organisaatio ymmärtämään asian merkitys ja sen vaikutukset. Kysehän on laadunvarmistuksesta ja toimintojen sekä prosessien parantamisesta.

Opinnäytetyössäni ydintietojen hallintaa on käsitelty loppujen lopuksi aika kapeasta näkökulmasta ja pääpaino tässäkin kohdistunut tekniseen ratkaisuun. Työlle asetetut tavoitteet saavutettiin sinänsä, mutta aiheesta riittää vielä paljon selvittävää. Vaikka esimerkiksi käytettyjen tietojen käsittely oli periaatteeltaan aika yksinkertaista, tuli tässä hyvin esille prosessin moninaiset vaiheet ja niihin liittyvät tekniset toteutukset.

Suunnittelun ja dokumentoinnin tärkeyttä ei pidä missään vaiheessa unohtaa tai vähätellä. Ydintietojen määrittely on pikkutarkkaa työtä ja vaatii asioiden pohtimista kaikkien asiaan liittyvien osapuolten näkökantojen huomioon ottamista. Liikkeelle voi lähteä pienestäkin osa-alueesta, kuten tässä työssä käsitellyistä henkilötiedoista. Tätäkin koskee kuitenkin samat periaatteelliset säännöt, kuin laajamittaisempaakin toteutusta.

Jatkokehityksenä tämän suunnitelman osalta ehdotan henkilötietojen tallennus- ja muutosprosessin jalkauttamista käytäntöön siten, että tiedot saataisiin kirjattua lähdejärjestelmään aikataulun työsuhteiden elinkaarten mukaisesti. Käytännössä tämä tarkoittaa sitä, että uuden henkilön aloittaessa hänestä olisi tiedot valmiina tarvittavien käyttäjätilien luomiseksi ja niiden luovuttamiseksi ja vastaavasti käyttäjätien sulkemiseksi työsuhteen päättyessä.

Tutkinnan kohteena olleesta MDS-järjestelmän ominaisuudet tulivat tämän tutkimuksen yhteydessä esille vain osittain. Järjestelmän syvällisempi tutkiminen antaisi varmasti lisää tietoa sen laajemmasta hyödyntämisestä. Toisaalta olisi mielenkiintoista tutustua myös johonkin toiseen tuotteeseen ja sen tarjoamiin ominaisuuksiin. Kyseinen tuote on kuitenkin hyvin käyttökelpoinen monine ominaisuuksineen tiedonsiirroissa ja muokkaamisissa.

LÄHTEET

- Hovi, Ari & Hervonen, Henriikki & Koistinen, Heikki 2009. Tietovarastot ja Business Intelligence. Jyväskylä: Docendo Finland Oy
- Hovi, Ari, 2004. SQL-opas. Jyväskylä: Docendo Finland Oy
- Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA), 2011. JHS 179 ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurimenetelmä/Liite 7 Master Data Management. Hakupäivä 13.1.2013. http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179/JHS179_liite7.pdf
- Kashel, Jeremy & Kent, Tim & Bullerwell, Martyn 2011. Microsoft SQL Server 2008 R2 Master Data Services. Birmingham:Packt Publishing Ltd.
<http://site.ebrary.com/lib/token/docDetail.action?docID=10490865>
- Korpela Jukka 2002, TUT www-sivut. Hakupäivä 17.11.2013.
<http://www.cs.tut.fi/~jkorpela/merkit/merkki.html>
- Laatikainen, Tero & Niemi Erkkä, 2012. DATA QUALITY JOHTAVISSA SUOMALAISYRITYKSISSÄ: NYKYTILA JA TARVITTAVAT RAKENTEET. Hakupäivä 16.1.2013 http://www.damafinland.fi/uploads/8/5/2/1/8521225/dama_dataquality_120430.pdf
- Lahtonen, Tommi, 2002. SQL. Jyväskylä: Docendo Finland Oy
- Lapin korkeakoulukonsernin (LUC) www-sivut 2013. Hakupäivä 17.11.2013.
http://www.luc.fi/Suomeksi/Esittely/LUC_pahkinankuossa.iw3
- Lappian www-sivut 2013. Hakupäivä 17.11.2013. <http://www.kkylappia.fi/Suomeksi/Esittely>
- Microsoft MSDN www-sivut 2013. dtexec Utility (SSIS Tool). Hakupäivä 17.11.2013.
<http://technet.microsoft.com/en-us/library/ms162810%28v=SQL.105%29.aspx>
- Microsoft MSDN www-sivut 2013. Install SQL Server 2012. Hakupäivä 5.3.2013
<http://msdn.microsoft.com/en-us/library/bb500395.aspx>
- Microsoft MSDN www-sivut 2013. Leaf Member Staging Table (Master Data Services). Hakupäivä 17.11.2013. <http://msdn.microsoft.com/en-us/library/ee633854.aspx>
- Microsoft MSDN www-sivut 2013. Master Data Services. Hakupäivä 5.3.2013.
<http://msdn.microsoft.com/en-us/library/ee633763.aspx>
- Microsoftin www-sivut, 2012. Features Supported by the Editions of SQL Server 2012. Hakupäivä 15.9.2013. <http://msdn.microsoft.com/en-us/library/cc645993.aspx>
- Niemi, Erkkä & Kontra, Kimmo 2012. DATA GOVERNANCE JOHTAVISSA SUOMALAISYRITYKSISSÄ: NYKYTILA JA YHDEKSÄN SUOSITUSTA. Hakupäivä 16.1.2013. http://www.damafinland.fi/uploads/8/5/2/1/8521225/dama_datagovernance_120212.pdf
- Pelluru, Sreedhar & Bravo, Jaime Alva, 2012. SQL Server 2012 Tutorial: Enterprise Information Management using SSIS, MDS, and DQS Together. Hakupäivä 5.3.2013.
<http://download.microsoft.com/download/5/5/5/5550A5F5-8F61-41CB-9BBE-A574143DE1E4/EIM%20Tutorial.docx>
- TalentBase 2012, MDM käytännössä -kurssimateriaali
w3schools.com www-sivut 2013. Hakupäivä 24.11.2013.
http://www.w3schools.com/sql/sql_datatypes.asp

LIITTEET

- Liite 1. Esimerkki attribuuttitason datastandardista
- Liite 2. MDS-käyttöliittymä
- Liite 3. T_LappiaPuh_I-taulun rakenne
- Liite 4. ETL-prosessi
- Liite 5. Scr_MerexExport Connection Manager
- Liite 6: OLE DB Connection Manager
- Liite 7. Scr_LappiaHlo Connection Manager
- Liite 8. V_LappiaPuh_IW-näkymän SQL-koodi
- Liite 9. V_LappiaHlo_IW SQL-koodi
- Liite 10. DQS Cleansing Transformation Editor
- Liite 11. V_LappiaHlo_O SQL-koodi
- Liite 12. Write data to MDS Lappia Staging – moduuli
- Liite 13. Prosessikaavio, uusi työsuhde

Esimerkki attribuuttitason datastandardista

LIITE 1

Attribuutin nimi	Sukunimi	Etunimet	Kutsunimi	Matkapuhelinnumero
Synonyymit	-	-	-	GSM-numero
Tiedon lähde	HR-järjestelmä	HR-järjestelmä	HR-järjestelmä	Puhelinvaihde
Tiedon ylläpitovastuu	Henkilöstöhallinto (C,R,U,D)	Henkilöstöhallinto (C,R,U,D)	Henkilöstöhallinto (C,R,U,D)	Vaihteenhoitajat (C,R,U,D)
	Käyttäjähallinto (IT) (R)		Käyttäjähallinto (IT)	Käyttäjähallinto (IT) (R)
	Puhelinvaihde (R)	Käyttäjähallinto (IT) (R)	Puhelinvaihde (R)	Henkilöstöhallinto (R)
	eOppimiskeskus (R)	Puhelinvaihde (R)	eOppimiskeskus (R)	eOppimiskeskus (R)
	Opintotoimisto (R)	eOppimiskeskus (R)	Opintotoimisto (R)	Opintotoimisto (R)
	Hankintapalvelut (R)	eOppimiskeskus (R)	Hankintapalvelut (R)	Hankintapalvelut (R)
Tiedon asiakkaat	Tiedotus ja markkinointi (R)	Opintotoimisto (R)	Tiedotus ja markkinointi (R)	Tiedotus ja markkinointi (R)
Tiedon omistaja	Henkilöstöhallinto	Henkilöstöhallinto	Henkilöstöhallinto	Puhelinvaihde
Sallitut arvot	Merkit A-Ö ja a-ö	Merkit A-Ö ja a-ö	Merkit A-Ö ja a-ö	Merkit 0-9, välilyönti ja väliviiva (-)
Esimerkkejä	Muttinen	Veikko Matti	Matti	040-123 4567
Attribuutin määritelmä	Henkilön sukunimi	Henkilön kaikki etunimet	Henkilön virallinen kutsunimi	Matkapuhelinnumero
Tiedon taso		Läpi organisaation käytettävä ydintieto		
Vaatimukset	Pakollinen	Pakollinen	Pakollinen	Vapaaehtoinen
Huomioitavaa	Nimen muuttuessa tiedot muuttuvat kaikissa kohdejärjestelmissä		Käytetään henkilön etunimitietona kaikissa näkyvillä olevissa paikoissa, kuten käyttäjähakemisto, puhelinluettelo jne...	

Microsoft SQL Server 2012
Master Data Services

Information Worker Tasks Refresh cached information

Model: Version: Status: Open

Explorer
Work with master data and with hierarchy relationships.

Install Master Data Services Add-in for Microsoft Excel
- Users can load and bulk edit data.
- Administrators can load data, create entities, and update attributes.

Administrative Tasks

System Administration
Build the model structure, create business rules, and deploy model packages.

Integration Management
Import data into MDS and create SQL views for use by subscription systems.

Version Management
Create versions of master data.

User and Group Permissions
Assign permissions to master data.

```

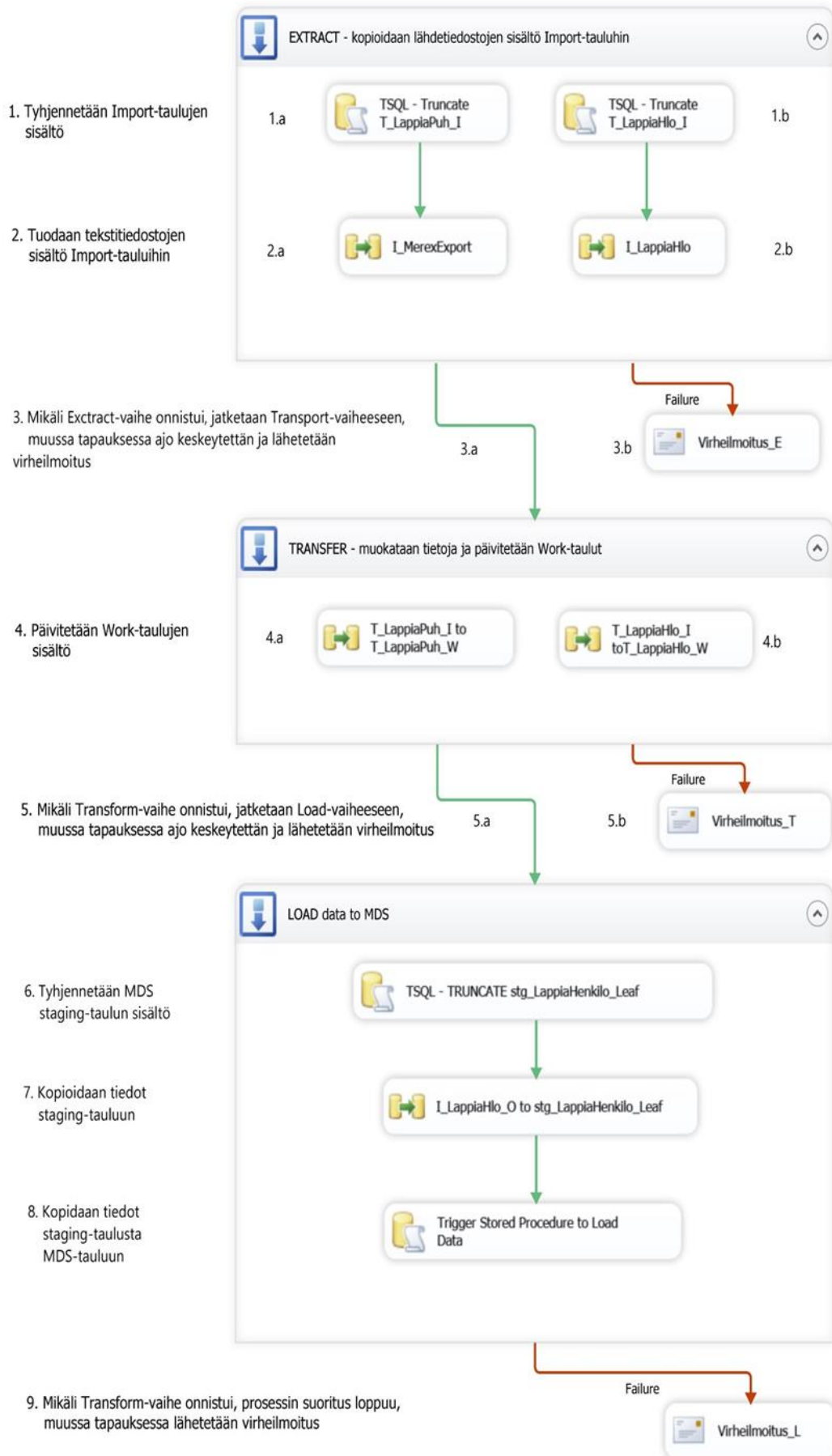
CREATE TABLE [dbo].[T_LappiaPuh_I](
    [ID] [varchar](50) NULL,
    [Luoja] [varchar](50) NULL,
    [Luontiaika] [varchar](50) NULL,
    [Viimeisin_päivittäjä] [varchar](50) NULL,
    [Viimeisin_päivitysaika] [varchar](50) NULL,
    [Nimi] [varchar](100) NULL,
    [Nick1] [varchar](50) NULL,
    [Nick2] [varchar](50) NULL,
    [Nick3] [varchar](50) NULL,
    [Nick4] [varchar](50) NULL,
    [Alanumero] [varchar](50) NULL,
    [Matkapuh] [varchar](50) NULL,
    [privatel] [varchar](50) NULL,
    [Sijainen] [varchar](100) NULL,
    [Sihteeri] [varchar](100) NULL,
    [Vara_nro] [varchar](50) NULL,
    [Lisännumero1] [varchar](50) NULL,
    ...
    [Tehtävä1] [varchar](500) NULL,
    [Tehtävä2] [varchar](255) NULL,
    [Tehtävä3] [varchar](255) NULL,
    ...
    [Nimike1] [varchar](255) NULL,
    [Nimike2] [varchar](255) NULL,
    [Nimike3] [varchar](255) NULL,
    [Käyntiosoite] [varchar](50) NULL,
    [Postiosoite] [varchar](50) NULL,
    [Huone] [varchar](50) NULL,
    [Fax] [varchar](50) NULL,
    [Sijainti] [varchar](50) NULL,
    [Yritys] [varchar](50) NULL,
    [Osasto] [varchar](500) NULL,
    [Ryhmä] [varchar](50) NULL,
    [Henkilönumero] [varchar](50) NULL,
    [KP] [varchar](50) NULL,
    [E-mail] [varchar](50) NULL,
    [Vaihde] [varchar](50) NULL,
    [Sijainen1] [varchar](50) NULL,
    [Sijainen2] [varchar](50) NULL,
    [Sihteeri1] [varchar](50) NULL,
    [Sihteeri2] [varchar](50) NULL,
    [Esimies] [varchar](50) NULL,
    [Piilotustieto] [varchar](50) NULL
) ON [PRIMARY]

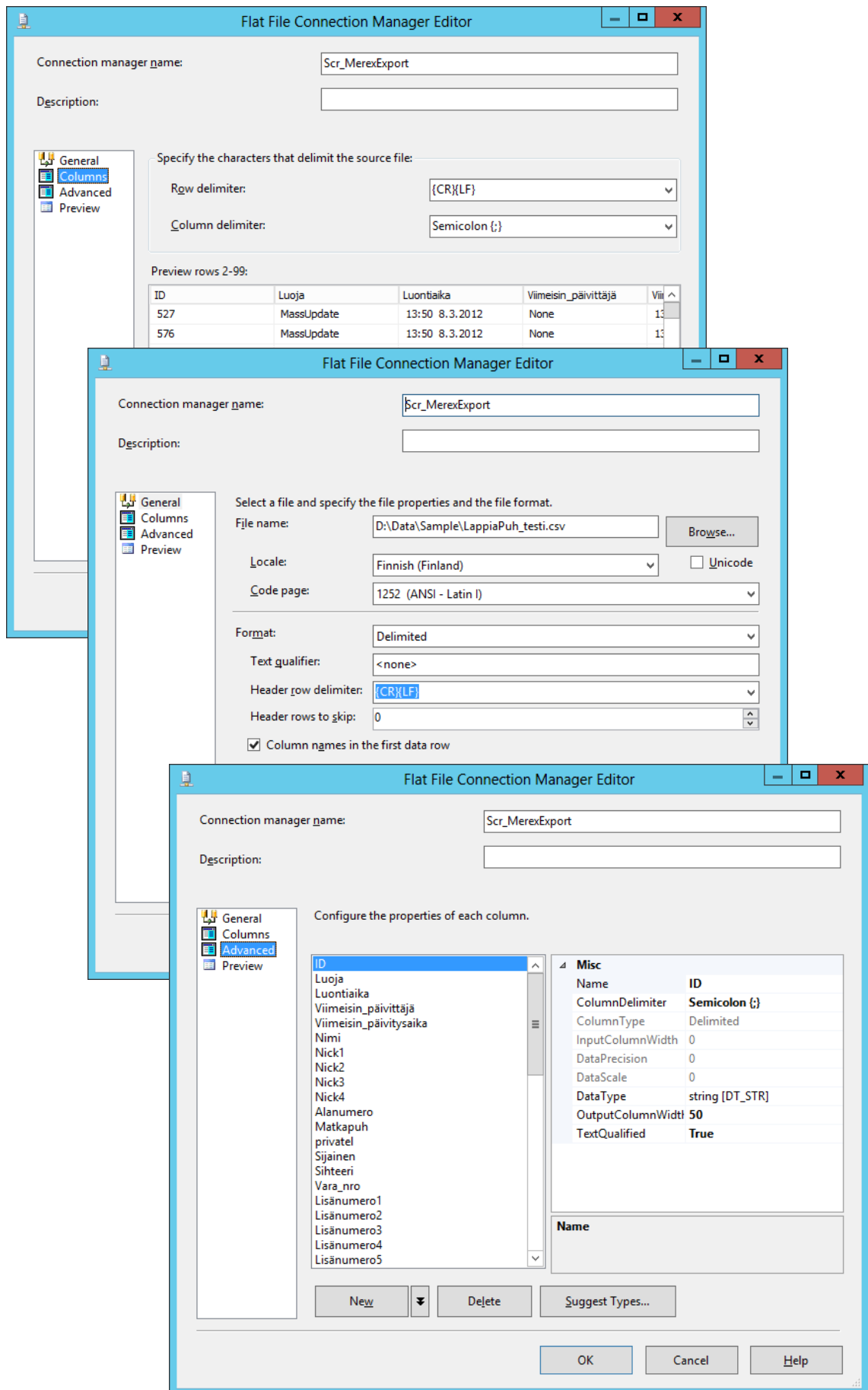
```

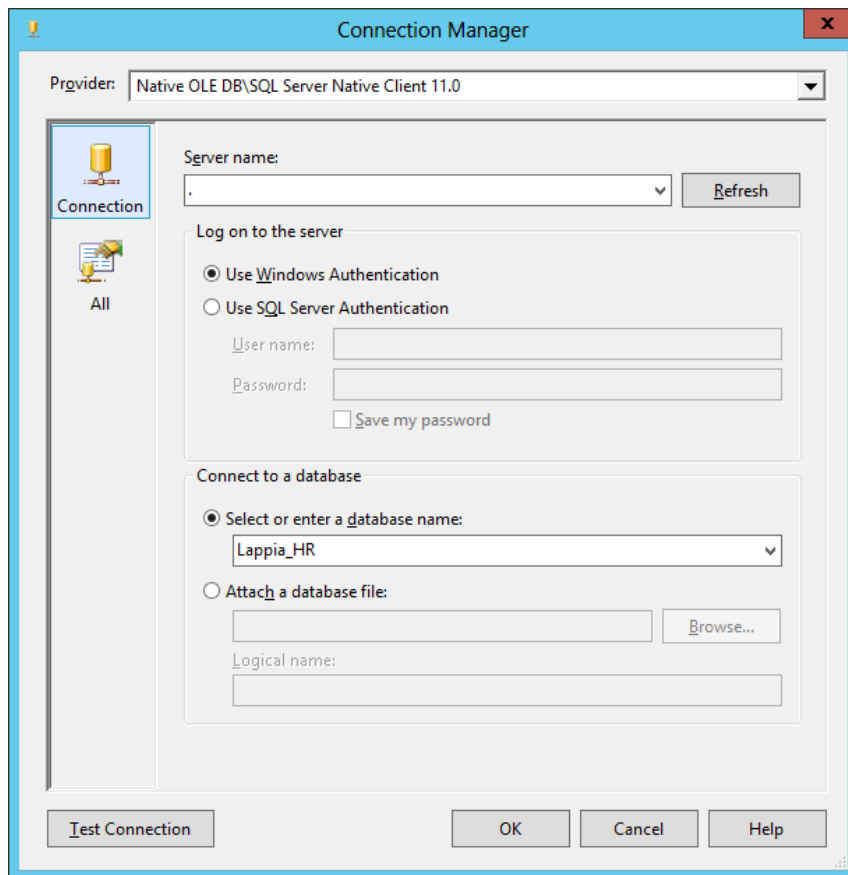
```
CREATE TABLE [dbo].[T_LappiaHlo_I](
    [Henkilonumero] [varchar](7) NULL,
    [Henkilotunnus] [varchar](11) NULL,
    [Palvelussuhdenro] [varchar](6) NULL,
    [Palvelussuhdejnro] [varchar](1) NULL,
    [Sukunimi] [varchar](50) NULL,
    [Kutsumanimi] [varchar](20) NULL,
    [Etunimet] [varchar](50) NULL,
    [Yksikko] [varchar](20) NULL,
    [Yksikkonimi] [varchar](50) NULL,
    [Toimintayksikko] [varchar](14) NULL,
    [Toimintayksikon_nimi] [varchar](50) NULL,
    [Ammattinimike] [varchar](50) NULL,
    [Viikkotyoaika] [varchar](5) NULL,
    [Esimies] [varchar](7) NULL,
    [Tyosuhteen_alkamispvm] [varchar](8) NULL,
    [Tyosuhteen_paattymispvm] [varchar](8) NULL,
    [Tili] [varchar](4) NULL
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[T_LappiaPuh_W](
    [Sukunimi] [varchar](100) NULL,
    [Etunimi] [varchar](100) NULL,
    [Matkapuhelin] [varchar](50) NULL,
    [Puhelin] [varchar](58) NULL,
    [Henkilönumero] [varchar](50) NULL,
    [Matkapuhelin2] [varchar](16) NULL
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[T_LappiaHlo_W](
    [Henkilönumero] [varchar](50) NULL,
    [Koodi] [varchar](50) NULL,
    [Sukunimi] [varchar](50) NULL,
    [Kutsumanimi] [varchar](50) NULL,
    [Etunimet] [varchar](50) NULL,
    [Ammattinimike] [varchar](50) NULL,
    [Organisaatio] [varchar](50) NULL,
    [Osasto] [varchar](50) NULL,
    [Työyksikkö] [varchar](50) NULL,
    [Työyksikön_nimi] [varchar](50) NULL,
    [Toimintayksikkö] [varchar](50) NULL,
    [Toimintayksikön_nimi] [varchar](50) NULL,
    [Esimies] [varchar](50) NULL,
    [Työsuhteen_alkamispvm] [varchar](50) NULL,
    [Työsuhteen_paattymispvm] [varchar](50) NULL,
    [Katuosoite] [varchar](50) NULL,
    [Postinumero] [varchar](50) NULL,
    [Postitoimipaikka] [varchar](50) NULL
) ON [PRIMARY]
```

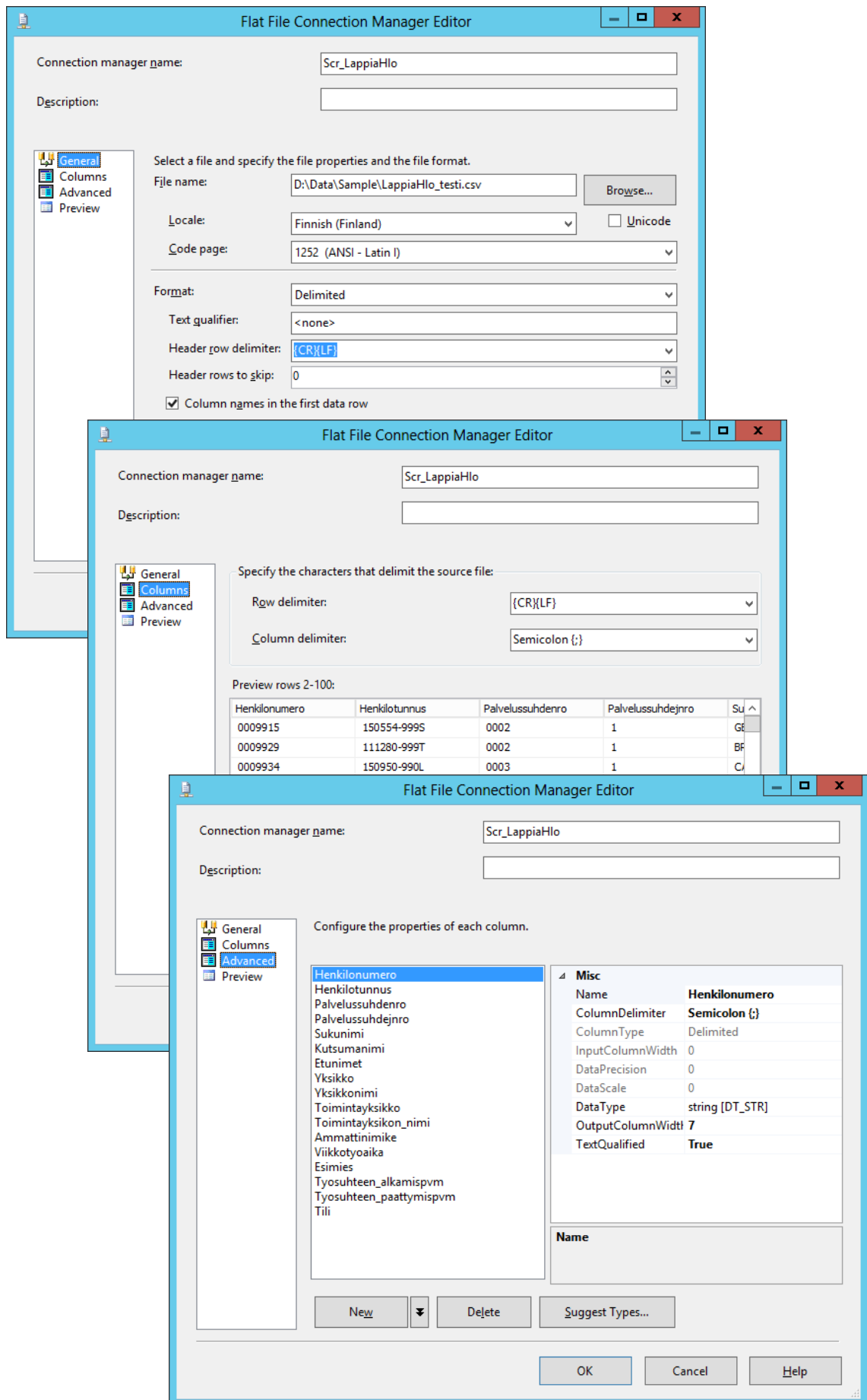






Server name – kentässä oleva piste tarkoittaa paikallista palvelinta.

Log on to server: Use Windows Authentication tarkoittaa, että tietokantaan otetaan yhteys käyttämällä käyttäjätiliä, jolla ollaan kirjaututtu Windows-istuntoon.



```

SELECT
-- Poimitaan henkilönnumero sellaisenaan
Henkilönnumero,
-- Sukunimi
-- Tarkistetaan nimi-kentästä, sisältääkö se välilyönnin (space)
CASE
-- mikäli välilyönti löytyy, poimitaan sen vasemmalla puolella oleva merkkijono suku nimeksi
  WHEN Nimi LIKE '% %' THEN LEFT(Nimi, Charindex(' ', Nimi) - 1)
  -- muussa tapauksessa asetetaan kentän arvoksi NULL
  ELSE NULL
END AS Sukunimi,
-- Etunimi
CASE
  -- mikäli välilyönti löytyy, poimitaan sen oikealla puolella oleva merkkijono etunimeksi
  WHEN Nimi LIKE '% %' THEN RIGHT(Nimi, Charindex(' ', Reverse(Nimi)) - 1)
  -- muussa tapauksessa asetetaan kentän arvoksi NULL
  ELSE NULL
END AS Etunimi,
-- Puhelin
CASE
  -- Mikäli alanumero-kentän arvo on NULL, asetetaan sen arvoksi tyhjä merkkijono
  WHEN Alanumero IS NULL THEN ''
  -- muussa tapauksessa asetetaan lyhytvalintanumeron eteen vakioarvo '010 383 '
  ELSE '010 383 ' + Alanumero
END AS Puhelin,
-- Matkapuhelin
CASE
  -- Mikäli matkapuhelin-kentän arvo on NULL, asetetaan sen arvoksi tyhjä merkkijono
  WHEN Matkapuh IS NULL THEN ''
  -- muussa tapauksessa muotoillaan merkkijono 040-1234567 merkkijonon 040 123 4567
  mukaiseksi
  ELSE SUBSTRING(Matkapuh, 1, 3) + ' ' + SUBSTRING(Matkapuh, CHARINDEX('-', Matkapuh) +
1, 3) + ' ' + SUBSTRING(Matkapuh, CHARINDEX('-', Matkapuh) + 4, 4)
END AS Matkapuhelin,
-- Matkapuhelin2
-- muotoillaan matkapuhelinnumero kansainväliseen muotoon
CASE
  WHEN Matkapuh = '' THEN ''
  -- lisätään merkkijonon eteen vakioarvo '+358 ' ja tehdään matkapuhelinta vastaavat muo-
  toilut
  ELSE '+358 ' + SUBSTRING(Matkapuh, 2, 2) + ' ' + SUBSTRING(Matkapuh, CHARINDEX('-', Mat
kapuh) + 1, 3) + ' ' + SUBSTRING(Matkapuh, CHARINDEX('-', Matkapuh) + 4, 4)
  END AS Matkapuhelin2
-- Poimitaan tiedot T_LappiaPuh_I-taulusta
FROM T_LappiaPuh_I
-- Järjestetään tulosjoukko sukunimen mukaiseen aakkosjärjestykseen
ORDER BY Sukunimi

```

```

SELECT
T1.Henkilonumero,
-- salataan henkilötunnus MD5-algoritmilla
convert(nvarchar(32),HashBytes('MD5', T1.Henkilotunnus),2) as Koodi,
T1.Sukunimi,
T1.Etunimet,
-- Poimitaan kutsumanimi
CASE T1.Kutsumanimi
-- Jos kutsumanimi-kentän arvo on tyhjä, haetaan etunimet-kentästä välilyönnin paikka.
    WHEN '' THEN CASE CHARINDEX(' ',T1.Etunimet)
        -- Jos sitä ei ole, poimitaan koko etunimet kentän arvo (yksiosainen etuni-mi).
        WHEN '0' THEN T1.Etunimet
    -- Muussa tapauksessa poimitaan etunimet-kentässä oleva ensimmäinen nimi.
        ELSE SUBSTRING(T1.Etunimet, 1, CHARINDEX(' ', T1.Etunimet) - 1)
    END
    -- Jos kentässä on arvo, poimitaan se sellaisenaan.
    ELSE T1.Kutsumanimi
END AS Kutsumanimi,
Ammattinimike,
-- Poimitaan organisaatio yksikkönimestä.
CASE T1.Yksikkonimi
    -- Jos kentän arvo on tyhjä, jätetään organisaatiokin tyhjäksi.
    WHEN '' THEN ''
    -- Muussa tapauksessa etsitään kentästä-välilyönnin paikka
    ELSE CASE CHARINDEX(' ',T1.Yksikkonimi)
        -- Jos sitä ei ole, poimitaan koko yksikkönimi.
        WHEN '0' THEN T1.Yksikkonimi
        -- Muussa tapauksessa poimitaan yksikkönimen ensimmäinen osa.
        ELSE SUBSTRING(T1.Yksikkonimi, 1,CHARINDEX(' ',T1.Yksikkonimi) - 1)
    END
END AS Organisaatio,
T1.Yksikko AS Tyoyksikko,
T1.Yksikkonimi AS Tyoyksikon_nimi,
-- Poimitaan osasto yksikkönimestä.
CASE T1.Yksikkonimi
    -- Jos kentän arvo on tyhjä, jätetään osastokin tyhjäksi.
    WHEN '' THEN ''
    -- Muussa tapauksessa etsitään kentästä-välilyönnin paikka
    ELSE CASE CHARINDEX(' ',T1.Yksikkonimi)
        -- Jos sitä ei ole, poimitaan koko yksikkönimi.
        WHEN '0' THEN T1.Yksikkonimi
        -- Muussa tapauksessa poimitaan yksikkönimen jälimmäinen osa.
        ELSE SUBSTRING(T1.Yksikkonimi,CHARINDEX(' ',T1.Yksikkonimi)
            + 1,LEN(T1.Yksikkonimi))
    END
END AS Osasto,
```

```
T1.Toimintayksikko,
T1.Toimintayksikon_nimi,
T1.Esimies,
-- Asetetaan katuosoitteen arvoksi NULL.
NULL AS Katuosoite,
-- Asetetaan postinumeron arvoksi NULL.
NULL AS Postinumero,
-- Asetetaan postitoimipaikan arvoksi NULL.
NULL AS Postitoimipaikka,
T1.Tyosuhteen_alkamispvm,
T1.Tyosuhteen_paattymispvm

FROM T_LappiaHlo_I T1

-- Yhdistetään samasta taulusta muodostettu virtuaalitaulu T2, jonka avulla poimitaan vain yksi
tietue/henkilö

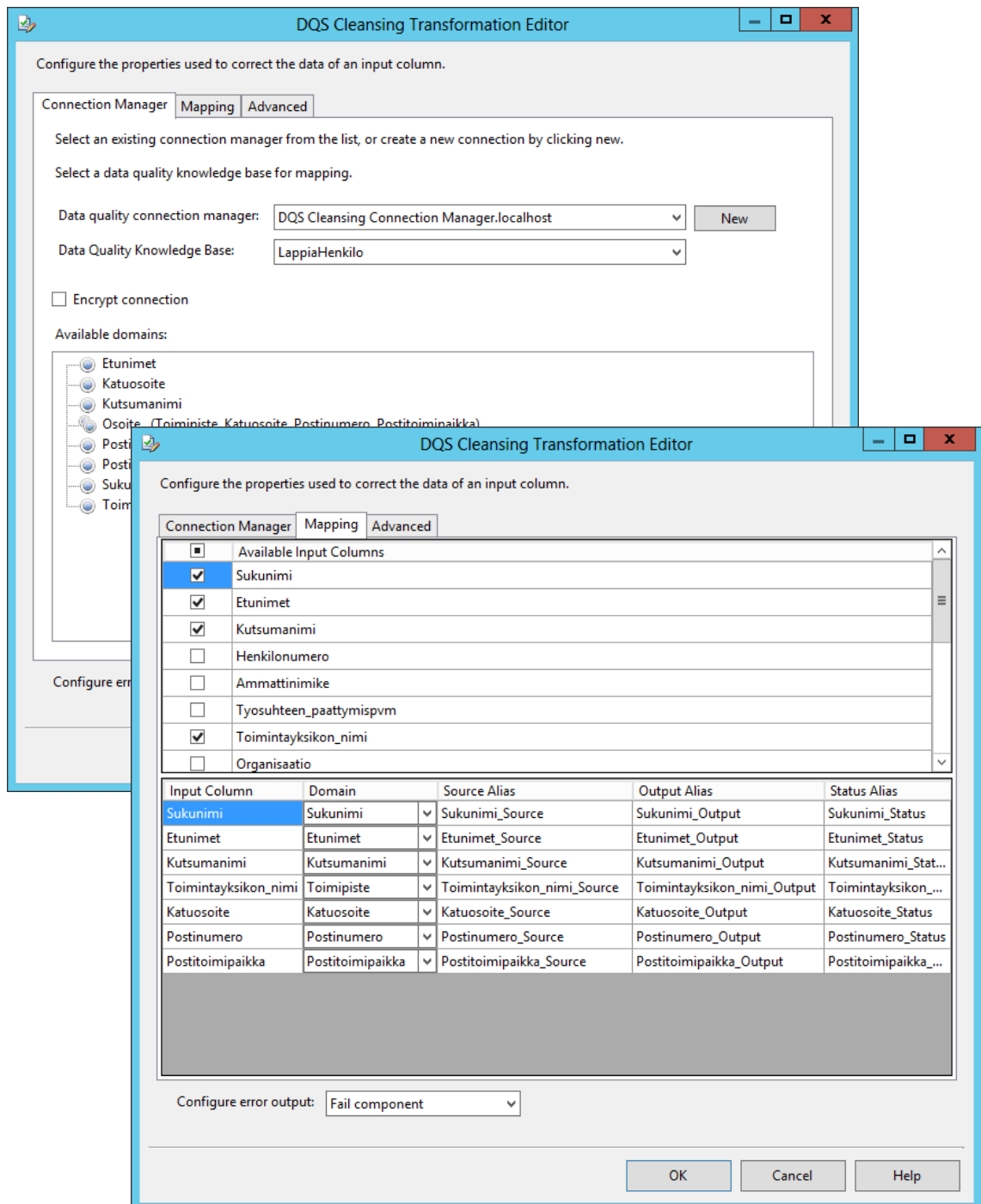
INNER JOIN

(SELECT
Henkilotunnus,
CASE
-- Mikäli henkilötunnus esiintyy useammalla, kuin yhdellä rivillä,
WHEN COUNT(Tyosuhteen_paattymispvm) > 1 THEN
CASE
WHEN MIN(Tyosuhteen_paattymispvm) >= CONVERT(char, GETDATE(), 112) THEN
MIN(Tyosuhteen_paattymispvm)
ELSE MAX(Tyosuhteen_paattymispvm)
END
ELSE MIN(Tyosuhteen_paattymispvm)
END AS Tyosuhteen_paattymispvm

FROM T_LappiaHlo_I

GROUP BY Henkilotunnus, Esimies) T2

ON T1.Tyosuhteen_paattymispvm = T2.Tyosuhteen_paattymispvm AND T1.Henkilotunnus =
T2.Henkilotunnus
```



```
SELECT
```

```
    T_LappiaHlo_W.Henkilonumero,  
    T_LappiaHlo_W.Koodi,  
    T_LappiaHlo_W.Sukunimi + ' ' + T_LappiaHlo_W.Kutsumanimi AS Nimi,  
    T_LappiaHlo_W.Sukunimi,  
    T_LappiaHlo_W.Etunimet,  
    T_LappiaHlo_W.Kutsumanimi,  
    T_LappiaHlo_W.Ammattinimike,  
    T_LappiaHlo_W.Organisaatio,  
    T_LappiaHlo_W.Osasto,  
    T_LappiaHlo_W.Tyoyksikko,  
    T_LappiaHlo_W.Tyoyksikon_nimi,  
    T_LappiaHlo_W.Toimintayksikko,  
    T_LappiaHlo_W.Toimintayksikon_nimi,  
    T_LappiaHlo_W.Esimies,  
    T_LappiaHlo_W.Tyosuhteen_alkamispvm,  
    T_LappiaHlo_W.Tyosuhteen_paattymispvm,  
    T_LappiaPuh_W.Puhelin,  
    T_LappiaPuh_W.Matkapuhelin,  
    T_LappiaPuh_W.Matkapuhelin2,  
    T_LappiaHlo_W.Katuosoite,  
    T_LappiaHlo_W.Postinumero,  
    dbo.T_LappiaHlo_W.Postitoimipaikka
```

```
FROM T_LappiaHlo_W LEFT OUTER JOIN T_LappiaPuh_W  
ON T_LappiaHlo_W.Henkilonumero = T_LappiaPuh_W.Henkilönumero  
AND T_LappiaHlo_W.Sukunimi = T_LappiaPuh_W.Sukunimi  
AND T_LappiaHlo_W.Kutsumanimi = T_LappiaPuh_W.Etunimi
```

Input Column	Destination Column ^A
<ignore>	Batch_ID
BatchTag	BatchTag
Koodi_DCT	Code
<ignore>	ErrorCode
Esimies_DCT	Esimies
Etunimet_DCT	Etunimet
Henkilonumero_DCT	Henkilonumero
<ignore>	Huone
<ignore>	ID
<ignore>	ImportStatus_ID
ImportType	ImportType
Katuosoite_DCT	Katuosoite
Kutsumanimi_DCT	Kutsumanimi
Matkapuhelin_DCT	Matkapuhelin
Nimi_DCT	Name
<ignore>	NewCode
Postinumero_DCT	Postinumero
Postitoimipaikka_DCT	Postitoimipaikka
Puhelin_DCT	Puhelin
Sukunimi_DCT	Sukunimi
<ignore>	Tili
Toimintayksikko_DCT	Toimintayksikko
Toimintayksikon_nimi_DCT	Toimintayksikkonimi
Tyosuhteen_alkamispvm_DCT	Tyosuhteen_alkamispvm
Tyosuhteen_paattymispvm_DCT	Tyosuhteen_paattymispvm
Tyoyksikko_DCT	Tyoyksikko
Tyoyksikon_nimi_DCT	Tyoyksikkonimi
<ignore>	Viikkotyoaika

Input Column – sarake

- kenttien nimet vastaavat Data Conversion – moduulin Output Alias – nimiä

Destination Column – sarake

- kenttien nimet vastaavat stg.LappiaHenkilo_Leaf – tietokantataulun kenttien nimiä

Prosessikaavio, uusi työsuhde

LIITE 13

