



SAVONIA

SSIS-TEKNIikka JA SEN HYÖDYNTÄMINEN

TEKIJÄ: Janne Vänttinen

Koulutusala Luonnontieteiden ala	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Työn tekijä(t) Janne Vänttinen	
Työn nimi SSIS-tekniikka ja sen hyödyntäminen	
Päiväys	17.3.2013
Sivumäärä/Liitteet	36
Ohjaaja(t) Marja-Riitta Kivi	
Toimeksiantaja/Yhteistyökumppani(t)	
<p>Tiivistelmä</p> <p>Opinnäytetyössä tutustutaan teoriatasolla SSIS-tekniikkaan, SQL-kieleen, Microsoft SQL SERVER –ohjelmaan sekä SSIS-liittymien toteutukseen käytetään Microsoft Intelligence Studio-ohjelmaa. Teoriaosuudessa käydään läpi SSIS-tekniikan hyödyntämismahdollisuuksia erilaisissa tietokantapohjaisissa toteutuksissa ja tutustutaan ETL-prosessiin.</p> <p>Opinnäytetyön teoriaosio keskittyy myös Microsoft SQL SERVER –ohjelman sekä SQL-kielen historiaan, jotka ovat mahdollistaneet SSIS-tekniikan kehittymisen sen nykyiseen tilanteeseen.</p> <p>Tämän jälkeen opinnäytetyössä on luotu tietokanta Microsoft SQL SERVER –ohjelmalla. Tietokanta on luotu teoriaosuudessa läpikäytyjen asioiden pohjalta. Tietokannan yhteyteen on rakennettu kaksi erilaista SSIS-liittymää, joiden toiminta ja rakenne käydään läpi askel kerrallaan. Kaikki edellämainittu materiaali on myös saatavilla internetistä. Linkki materiaaliin löytyy opinnäytetyöstä.</p> <p>SSIS-tekniikka on erittäin varteenotettava vaihtoehto ETL-prosessien suorittamiseen kun tietokantaa ylläpidetään Microsoft SQL SERVER –ohjelmalla.</p>	
Avainsanat SSIS, ETL, SQL SERVER, SQL SERVER Intelligence Studio	

Field of Study Natural Sciences			
Degree Programme Degree Programme in Information Technology			
Author(s) Janne Väättinen			
Title of Thesis SSIS-technology and benefits of its use			
Date	17.2.2013	Pages/Appendices	36
Supervisor(s) Marja-Riitta Kivi			
Client Organisation /Partners			
<p>Abstract</p> <p>This thesis concentrates on SSIS-technology. Theory part of the thesis deals with SSIS-technology, SQL-language, Microsoft SQL SERVER- software and creating SSIS-interfaces with Microsoft Intelligence Studio –software. The theory part also includes basics of databases and how SSIS-technology can be benefited in database-based solutions via ETL-processes.</p> <p>The thesis contains a database which was created with Microsoft SQL SERVER –software in accordance with the basic functions that were introduced in the theory part. The database contains two different SSIS-interfaces. Both interfaces and their functionality are described step by step in the thesis. All material including scripts for creating the database and SSIS-interfaces are available on the internet. The link to the material can be found in the thesis.</p> <p>SSIS-technology is probably the best option for running ETL-processes when a database is maintained with Microsoft SQL SERVER –software.</p>			
<p>Keywords SSIS, ETL, SQL SERVER, SQL SERVER Intelligence Studio</p>			

SISÄLTÖ

1	JOHDANTO	6
2	SQL SERVER	7
2.1	SQL SERVER:in perusteet	7
2.2	Opinnäytetyössä käytetty versio.....	7
2.3	Historia	7
3	TIETOKANTA	9
3.1	Mikä on tietokanta?.....	9
3.2	Tietokannan taulut.....	9
3.3	Relaatiot.....	10
3.4	Relaatioesimerkkejä	10
3.5	Triggerit.....	11
3.6	Proseduurit.....	12
4	SQL-KIELI.....	14
4.1	SQL-kieli lyhyesti.....	14
4.2	Historia	14
4.3	Esimerkkilauseita	14
5	SSIS-TEKNIikka.....	16
5.1	Mikä on SSIS?.....	16
5.2	Vaihtoehdot SSIS-tekniikalle	16
5.3	ETL-prosessi	16
5.3.1	Extract eli purkuvaihe.....	17
5.3.2	Transform eli muutosvaihe	18
5.3.3	Load eli latausvaihe.....	18
5.4	SSIS-liittymän toteuttaminen SQL SERVER Business Intelligence Studio -ohjelmalla	19
5.5	SSIS-liittymän toimintaperiaate	19
5.6	Virhetilanteet liittymän suorituksessa.....	21
5.7	Liittymän suorittamisen ajastaminen	21
5.8	SSIS-liittymän suorittaminen SQL SERVER Management Studio -ympäristössä	22
5.9	Uuden job-elementin asentaminen SQL SERVER Management Studio -ympäristöön	23
5.10	Jobin suorittaminen SQL SERVER Management Studio -ympäristössä	24

6	SSIS-TEKNIIKAN HYÖDYNTÄMINEN HENKILÖSTÖHALLINNAN SOVELLUKSESSA.....	26
6.1	Yleistä.....	26
6.2	Tietokanta.....	26
6.2.1	Tietokannan taulut.....	26
6.2.2	Tietokannan relaatiot.....	27
6.2.3	Tietokannan trigger.....	28
6.2.4	Tietokannan luominen.....	28
6.3	Tietokannan sisältämä tieto.....	29
6.3.1	Työntekijä-taulun testiarvot.....	29
6.3.2	Nimike-taulun testiarvot.....	29
6.3.1	Yksikkö-taulun testiarvot.....	29
6.3.2	Esimerkkitoteutuksen organisaatorakenne.....	30
6.4	SSIS-liittymä työntekijätietojen tuomiseen siirtotiedostosta.....	30
6.4.1	Liittymän rakenne.....	31
6.4.2	Siirtotiedosto.....	32
6.5	SSIS-liittymä työntekijä tietojen hakemiseen kannasta.....	33
6.5.1	Liittymän rakenne.....	33
6.5.2	Tekstitiedoston rakenne.....	33
7	POHDINTA.....	34

LÄHTEET

LIITTEET

1 JOHDANTO

Tässä opinnäytetyössä tutustutaan SSIS-tekniikkaan, ETL-prosessiin ja SSIS-tekniikan hyödyntämiseen ETL-prosessissa. Opinnäytetyöhön kuuluu tiedostopaketti, joka sisältää opinnäytetyössä läpikäytäviä tiedostoja.

Opinnäytetyöhön liittyvän materiaalipaketin voi ladata osoitteesta
https://dl.dropbox.com/u/20386846/Oppari/Opinnaytetyomateriaali_Janne_Vanttinen.zip

ETL-prosessia käytetään tietojen siirtämiseen tietokantojen välillä. Opinnäytetyön tarkoituksena on tutustua ETL-prosessiin ja käytännön esimerkkien kautta ETL-prosessin toteuttamiseen SSIS-tekniikalla.

Opinnäytetyössä tutustutaan ensin teoriatasolla Microsoft SQL Server- ja SQL SERVER Intelligence Studio -ohjelmistoihin sekä ETL-prosessiin. Tämän jälkeen käydään käytännön esimerkkien kautta läpi SSIS-tekniikalla luotu SSIS-liittymä, jolla suoritetaan ETL-prosessi.

2 SQL SERVER

2.1 SQL SERVER:in perusteet

SSIS eli SQL SERVER INTEGRATION SERVICES on osa Microsoft SQL SERVER -ohjelmistoa. Microsoft SQL SERVER on tietokanta-alustaohjelmisto, jolla voidaan toteuttaa relaatiotietokantoja. Yksinkertaisesti sanottuna Microsoft SQL SERVER on siis tietokanta-alusta, jossa säilytetään tietoja. SQL SERVER -pohjaisiin tietokantoihin voi lisätä tai hakea kannan sisältämää tietoa erillisillä ohjelmilla. Tietojen lisääminen ja hakeminen onnistuu myös ohjelmiston oman käyttöliittymän kautta, mutta tämä tapa ei ole käytännöllinen etenkin suuremmissa tietokantakokonaisuuksissa. (FunctionX, 2013)

Microsoft SQL SERVER -ohjelmisto soveltuu eri kokoisten tietokantojen ylläpitoon. Pienimuotoisissa toteutuksissa ohjelmisto voidaan asentaa paikallisesti yhdelle tietokoneelle ja tietokannan hallinnointi tapahtuu tässä tapauksessa paikallisesti.

Suurissa toteutuksissa Microsoft SQL SERVER -ohjelmisto voidaan asentaa palvelimelle, jolloin tietokantaa pystytään hallinnoimaan useilla eri tietokoneilla internetin välityksellä. (Microsoft, 2013a)

Kun Microsoft SQL SERVER -ohjelmisto asennetaan palvelimelle niin järjestelmässä tapahtuvat työt jaetaan asiakas- ja palvelintietokoneiden välille. Käytännössä tämä tarkoittaa siis sitä, että asiakastietokoneet, joita voi tässä tapauksessa olla useita, vastaavat kannan sisältämän aineiston tuottamisesta. Palvelintietokone puolestaan hallinnoi tietokantaa ja sen ylläpitoa sekä jakaa resursseja asiakaskoneiden käyttöön. (Granroth, 2001)

2.2 Opinnäytetyössä käytetty versio

Tähän opinnäytetyöhön liittyvien tietokantojen ja SSIS-liittymien luonnissa sekä niiden pohjana olen käyttänyt Microsoft SQL SERVER -ohjelmiston versiota 2008 R2 Enterprise edition.

Valitsin kyseisen version käytettäväksi opinnäytetyössäni koska olen töissäni työskennellyt paljon kyseisen ohjelmiston kanssa. Kyseinen ohjelmistoversio on myös työkokemuksieni mukaan tällä hetkellä yleisin käytössä oleva versio ohjelmistosta.

2.3 Historia

Microsoft SQL SERVER -ohjelmiston ensimmäinen versio julkaistiin 1989. 80-luvun lopulla Microsoft ja Sybase ryhtyivät yhteistyöhön luodakseen kilpailevan ohjelmiston IBM:ää ja Oraclea vastaan, jotka tuohon aikaan hallitsivat yritystason tietokantamarkkinoita. (Mohan, 2010)

Microsoftin ja Sybasen yhteistyö loppui 90-luvun alkupuolella. Vuonna 1995 Microsoft julkaisi SQL SERVER -ohjelmistosta version 6.0, joka oli ensimmäinen NT-käyttöjärjestelmille suunniteltu versio.

SQL SERVER 6.0 oli myös ensimmäinen SQL SERVER -ohjelmiston versio, jonka kehityksessä Sybase ei ollut mukana. (Redware, 2001)

Vuonna 2000 julkaistiin Microsoft SQL SERVER 2000 -ohjelmisto, joka oli ensimmäinen täysipainoinen relaatiotietokantojen hallintaan tarkoitettu ohjelmisto. Tähän versioon mennessä kaikki ohjelmiston sisältämä koodi oli uudelleenkirjoitettu, joten Sybianin perintönä tullutta koodia ei enää ohjelmistosta löytynyt. (Mohan, 2010)

Seuraava suuri päivitys julkaistiin vuonna 2005. Microsoft SQL SERVER 2005 -version suurimmat uudistukset liittyivät sen tarjoamaan XML-tukeen sekä virheiden tunnistukseen SQL-lauseita suoritettaessa. (Mohan, 2010)

Vuonna 2008 julkaistiin Microsoft SQL SERVER 2008. Tämän version suurimpia uudistuksia olivat suorituskyvyn paraneminen sekä helpompi käytettävyys. Tämän version myötä Microsoft toi myös ohjelmistoon uusia datatyyppisiä, jotka helpottivat tietokannan tietojen raportoimista käyttäjille. (Mohan, 2010)

Vuonna 2010 julkaistu Microsoft SQL SERVER 2008 R2 on 2008-version pohjalle rakennettu paranneltu versio. Tämän version myötä ohjelmiston teknisiä ominaisuuksia päivitettiin vastaamaan 2010-luvun vaatimuksia. (Mohan, 2010)

3 TIETOKANTA

3.1 Mikä on tietokanta?

Tietokanta on yksinkertaisimmillaan lista tiedoista. Tietokannat ovat osa jokapäiväistä elämäämme, vaikka emme sitä aina tule ajatelleeksi. Puhelinluettelo on yksi esimerkki tietokannasta. Tietokantoja käytetään laajasti myös esimerkiksi verkkokaupoissa. Esimerkiksi tuotetiedot ovat tallennettuna tietokantoihin, joista tieto haetaan käyttäjän määrittämien hakuehtojen mukaan näkyville. Microsoft SQL SERVER on siis ohjelmisto, jolla hallinnoidaan tietokantoja. (Allen, 2004)

3.2 Tietokannan taulut

Tietokanta sisältää tauluja, jonne tiedot tallennetaan. Tietokanta muodostuu vähintään yhdestä tietokannan sisältämästä taulusta. Tietokannan sisältämät taulut muodostuvat taulun sisältämistä kentistä. (Microsoft, 2013b)

Jokaisella taulun sisältämällä kentällä on oma datatyyppinsä, joka määrittelee missä muodossa kyseiseen kenttään voidaan syöttää tietoa. Esimerkiksi integer-tyyppiseen kenttään ei voi syöttää tekstiä, koska integer-datatyyppi vaatii syötetyn tiedon numeerisessa muodossa. Taulukossa 1 on listattu yleisimmät datatyypit. (Microsoft, 2013c)

TAULUKKO 1. Yleisimmät tietokantojen tauluissa käytetyt datatyypit

Datatyyppi	Vaadittu tiedon syöttömuoto	Esimerkki
Integer	Numeerinen tieto	16
nvarchar(255)	Tekstimuotoinen tieto, suluisissa oleva luku tarkoittaa tiedon maksimipituutta	Esimerkki
datetime	Päivämäärä ja aikamuoto	2012-12-12 16:00:00
ntext	Tekstimuotoinen tieto. Datatyyppin maksimipituus on suuri, joten ntext on käytännöllisempi datatyyppi pitkille teksteille kuin nvarchar.	Esimerkki
decimal	Numeerinen tieto, jossa tarvitaan desimaaleja. Esitettävien desimaalien lukumäärä on muutettavissa.	16,46

Kuvassa 1 on esimerkki tyontekija-taulusta SQL SERVER Management Studio Design-näkymässä.

	Column Name	Data Type	Allow Nulls
🔑	tyontekija_id	int	<input type="checkbox"/>
	tyontekija_nimi	nvarchar(50)	<input checked="" type="checkbox"/>
	tyontekija_titteli	nvarchar(50)	<input checked="" type="checkbox"/>
	tyontekija_yksikko_id	int	<input checked="" type="checkbox"/>
	tyontekija_lahiosoite	ntext	<input checked="" type="checkbox"/>
	tyontekija_lisatty_ts	datetime	<input checked="" type="checkbox"/>

KUVA 1. Esimerkkikuva tietokantataulusta

Microsoft SQL SERVER -ohjelmistossa on sisäänrakennettu tiedon oikeellisuuden tarkastus, joka antaa käyttäjälle virheilmoituksen mikäli tietokantaan yritetään syöttää tietoa väärässä muodossa. Tässä tapauksessa käyttäjän syöttämiä tietoja ei luonnollisesti lisätä tietokantatauluun.

3.3 Relaatiot

Microsoft SQL SERVER -ohjelmiston sisältämät tietokannat ovat relaatiotietokantoja. Tämä tarkoittaa sitä, että tietokanta sisältää useita eri tauluja, jotka ovat yhteydessä toisiinsa. Relatio tarkoittaa siis kahden taulun välistä yhteyttä. (Kettunen, 2001)

Kaksi tietokannan sisältämää taulua voivat olla yhteydessä toisiinsa myös muiden taulujen kautta. Tämä tarkoittaa ettei kyseisten taulujen välillä ole suoraa relaatiota, mutta molemmista tauluista on esimerkiksi relatio tauluun X ja taulut ovat yhteydessä toisiinsa tätä kautta. Kuvassa 15 työntekijän esimiestieto haetaan yksikkö-taulun kautta. Työntekijän tiedoissa ei ole esimiestietoa, mutta työntekijän yksikkötiedon kautta haetaan yksikön ja samalla työntekijän esimies.

3.4 Relatioesimerkkejä

Kuvassa 2 on esitetty Työntekijä-taulun ja Yksikkö-taulun välinen relatio. Työntekijä-taulu sisältää työntekijän perustiedot. Työntekijän ID-kenttä on perusavainkenttä. Tämä tarkoittaa sitä, että jokaiselle työntekijälle asetetaan automaattisesti ID-numero, joka on yksilöllinen. Kahdella työntekijällä ei siis voi olla samaa ID-numeroa.



KUVA 2. Esimerkkikuva relaatiosta

Yksikkö-taulussa on tieto yksiköiden nimistä sekä yksilöllinen ID-numero jokaiselle yksikölle.

Kun yksikkötiedot ovat omassa taulussaan, niin Työntekijä-taulun Yksikkö-kentän ei tarvitse olla tekstimuotoinen. Yksikköä ei siis tarvitse erikseen kirjoittaa jokaiselle työntekijälle. Työntekijä-taulun yksikkökenttä sisältää numerotiedon, joka viittaa Yksikkö-tauluun sen yksikön ID-numeroon, johon työntekijä kuuluu.

Yksikkö-taulu sisältää 3 yksikköä. Yksiköt ovat Huolto (1), Ylläpito (2) ja Käyttäjätuki (3), suluissa olevat numerot ovat yksikköjen ID-numerot. Työntekijä-taulussa on työntekijän ”Erkki Esimerkki” työntekijätiedot. Erkki kuuluu Ylläpito-yksikköön, joten Erkin tiedoissa tyontekija_yksikko_id-kentän arvo on 2. Kun kyseisestä kentästä on relaatio Yksikkö-tauluun, niin tätä kautta voidaan selvittää Erkin yksikkö ID-viittauksen avulla.

Relaatioiden käyttäminen säästää siis huomattavasti aikaa ja helpottaa tietokannan sisältämien tietojen hallinnointia. On myös huomion arvoista, että jos yksikköjen nimet muuttuvat niin nimen muuttaminen Yksikkö-tauluun riittää. Työntekijä-tauluun haetaan uusi yksikkönimi samalla ID-viittauksella kuin aiemminkin. Jos kuitenkin yksikkönimi olisi kirjoitettu suoraan tekstinä työntekijätauluun, niin yksikkönimi pitäisi päivittää yksitellen jokaiselle kyseiseen yksikköön kuuluvalla työntekijälle.

3.5 Triggerit

Triggerit ovat oleellinen osa tietokantaa ja sen toimintaa. Suomenkielistä nimitystä triggerille on vaikea keksiä, mutta kuten nimestä voi päätellä niin triggeri laukeaa tietyssä tilanteessa kun tietokantaa käsitellään. (Wells, 2001)

Triggeri sisältää SQL-lauseita, joilla voidaan muokata tietokannan sisältöä. Triggerit ovat taulukohtaisia ja niiden laukeaminen on sidottu niiden isäntätaulun toimintaan. On kuitenkin huomioitava, että triggereillä voidaan hallinnoida kaikkea tietokannan sisältämää tietoa vaikka niiden laukeaminen onkin sidottu yhteen tauluun. (Wells, 2001)

Triggerit voidaan asettaa laukeamaan kolmessa eri tilanteessa. Nämä tilanteet ovat tiedon lisääminen tauluun, taulussa olevan tiedon muokkaaminen tai taulussa olevan tiedon poistaminen. Triggerin voi myös asettaa laukeamaan esimerkiksi kaikissa edellämainituissa tilanteissa.

Alla olevassa kuvassa on SQL SERVER Management Studiosta otettu kuva, jossa esitetään tyontekija_insert_trigger-nimisen triggerin SQL-koodi. Kyseinen trigger on tyontekija-tauluun kuuluva trigger joka laukeaa kun tyontekija-tauluun lisätään uusi rivi. Tässä yhteydessä trigger lisää tyontekija-taulun tyontekija_lisatty_ts-kenttään aikaleiman, joka kertoo milloin kyseinen rivi on lisätty tauluun. Aikaleiman lisäämiseen käytetään GETDATE-funktiota.

Esimerkkitriggerissä on työntekijärivi, jolle aikaleima lisätään käyttämällä inserted-taulua kuvan 3 mukaisella tavalla. Rivin lisäyksen yhteydessä inserted-tauluun muodostuu identtinen rivi lisätyn työntekijärivin kanssa. Aikaleiman asettamista varten oikea työntekijärivi voidaan siis selvittää etsimällä työntekijä-taulusta rivi, jonka yksilöivä ID-numero vastaa inserted-taulun sisältämän rivin ID-numeroa.

```
-- =====  
-- Author:      Janne Vääntinen  
-- Create date: 22.12.2012  
-- Description: Esimerkkitriggeri opinnäytetyöhön  
-- =====  
ALTER TRIGGER [dbo].[tyontekija_insert_trigger]  
ON [dbo].[tyontekija]  
AFTER INSERT  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    UPDATE tyontekija  
    SET tyontekija_lisatty_ts = GETDATE()  
    FROM tyontekija  
    INNER JOIN inserted  
    ON tyontekija.tyontekija_id = inserted.tyontekija_id  
    WHERE tyontekija.tyontekija_id = inserted.tyontekija_id  
  
END
```

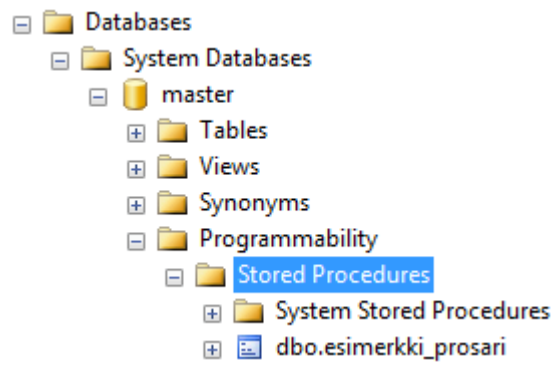
KUVA 3. Kuvassa on esitetty SQL SERVER Management Studio –ympäristöstä otettu kuva triggeristä. Kyseinen trigger on osa tyontekija-taulua ja laukeaa kun tyontekija-tauluun lisätään uutta tietoa.

3.6 Proseduurit

Proseduureissa voidaan suorittaa SQL-komentoja samalla tavalla kuin triggereissä. Proseduureissa suoritettavia SQL-komentoja tai SQL-komentojen kohdetauluja ei ole rajoitettu millään tavalla.

Proseduurit eivät ole riippuvaisia tietokannan tauluista tai niihin tehdyistä muutoksista. Proseduureja kutsutaan "EXEC proseduurin_nimi" -komennolla. Proseduureja voidaan kutsua ja suorittaa esimerkiksi tietokantaa käsittelevästä sovelluksesta, triggereistä tai toisista proseduureista.

Microsoft SQL SERVER -ohjelmistossa proseduurit löytyvät tietokannan "alta" Stored procedures-kansiosta. Kuvassa 4 on esitetty proseduurien sijainti SQL SERVER Management Studio –ympäristössä.



KUVA 4. Proseduurien sijainti Microsoft SQL Server Management Studiossa

4 SQL-KIELI

4.1 SQL-kieli lyhyesti

SQL-kieli on ohjelmointikieli, jolla hallinnoidaan relaatiotietokantoja. Tietokannan sisältämään aineistoon voidaan kohdistaa hakuja käyttämällä SQL-kieltä. Myös tietokannan sisältämän tiedon lisääminen, muokkaaminen ja poistaminen tapahtuu käyttämällä SQL-kieltä.

4.2 Historia

IBM kehitti SQL-kielen 1970-luvun lopulla hallinnoidakseen tietoja IBM System R-tietokantaohjelmistossa. American National Standard Instituten (ANSI) standardisoi SQL-kielen vuonna 1986. (Simba technologies, 2013)

Nykyään SQL-pohjaiset tietokannat ovat erittäin suosittuja. SQL-kieltä pidetäänkin tänä päivänä tiedon hallinnoinnin kivijalkana. (Simba technologies, 2013)

4.3 Esimerkkilauseita

Taulukossa 2 on esitetty muutamia yksinkertaisia SQL-lauseita sekä selitetty niiden toimintaa.

TAULUKKO 2. Esimerkkilauseita SQL-kielillä

SQL-lause	Selite
SELECT * FROM tyontekija	Kyselylause, joka palauttaa kaiken tiedon tyontekija-tilusta
SELECT tyontekija_nimi FROM tyontekija WHERE tyontekija_yksikko = 'Yksikkö 1'	Kyselylause, joka palauttaa tyontekija-tilusta tyontekija_nimi-kentän sisältämän tiedon niiltä riveiltä, joissa tyontekija_yksikko-kentän arvo on "Yksikkö 1"
INSERT INTO tyontekija SELECT 'teksti1', 'teksti2', 1988, 'teksti3'	Lisäyslause, joka lisää tyontekija-tiluun riviin, SELECT-lauseen jälkeisillä arvoilla. On tärkeää huomioida, että tyontekija-tilun kenttien 1, 2 ja 4 on oltava datatyypiltään tekstimuotoisia ja kentän 3 numeromuotoinen. Muuten tiedon lisäys tiluun epäonnistuu.
DELETE FROM tyontekija	Poistolause, joka poistaa kaikki rivit tyontekija-tilusta.
DELETE FROM tyontekija WHERE tyontekija_nimi = 'Erkki Esimerkki'	Poistolause, joka poistaa tyontekija-tilusta kaikki rivit joissa tyontekija_nimi-kentän arvona on "Erkki Esimerkki"
UPDATE tyontekija SET tyontekija_nimike = 'Työntekijä' WHERE tyontekija_nimike =	Päivityslause, joka päivittää tyontekija_nimike-kenttään arvon "Työntekijä" niille riveille, joilla

'Alainen'	kyseisen kentän arvona on päivityshetkellä "Alainen".
-----------	--

5 SSIS-TEKNIikka

5.1 Mikä on SSIS?

SSIS on lyhenne sanoista SQL SERVER INTEGRATION SERVICES. SSIS-liittymiä käytetään ETL-prosessien toteuttamiseen. ETL on lyhenne sanoista Extract, transform ja load. SSIS-liittymien luomiseen tarvitaan SQL SERVER Business Intelligence Studio -ohjelma. Tämä ohjelma kuuluu Microsoft SQL SERVER -ohjelmistopakettiin ja sen asentaminen on onnistuu SQL SERVER -ohjelmiston asentamisen yhteydessä. Microsoft SQL SERVER -ohjelmistosta on myös esimerkiksi opiskelukäyttöön tarkoitettuja karsittuja versioita, kuten esimerkiksi Microsoft SQL SERVER 2008 R2 Express Edition.

SQL SERVER Business Intelligence Studio -ohjelma ei kuulu kaikkiin SQL SERVER-ohjelmistoversioihin. Esimerkiksi SQL SERVER 2008 R2 Express Edition ei sisällä Intelligence Studio-ohjelmaa.

5.2 Vaihtoehdot SSIS-tekniikalle

SSIS-tekniikalle on olemassa useita eri vaihtoehtoja. Muita ETL-prosesseissa käytettäviä ohjelmia ovat esimerkiksi Oracle Warehouse Builder ja Pentaho ETL. Kaikkien vaihtoehtoisten ohjelmien toimintaidea on sama. Suurimmat erot ovat ohjelmien toteutuksessa ja käyttöliittymissä. Esimerkiksi Pentaho ETL on toteutettu Java-tekniikalla. (Linstedt, 2013)

ETL-prosessissa käytettävä ohjelmisto on järkevää valita käyttötarkoituksen ja muiden käytettävien ohjelmistojen mukaan. Käytettäessä Microsoft SQL Serveriä on viisainta käyttää ETL-prosessiin SQL SERVER Integration Services -ohjelmaa. SQL SERVER Integration Services -ohjelma on suunniteltu toimimaan SQL SERVER -ohjelmiston kanssa, joten SQL SERVER -ohjelmistoa käyttäessä se on paras valinta. Tässä tilanteessa on turha käyttää ETL-prosessiin ulkopuolisia ohjelmia joiden yhteensopivuus SQL SERVER -ohjelmiston kanssa ei ole yhtä hyvä kuin SQL SERVER Integration Services-ohjelmiston. (Linstedt, 2013)

5.3 ETL-prosessi

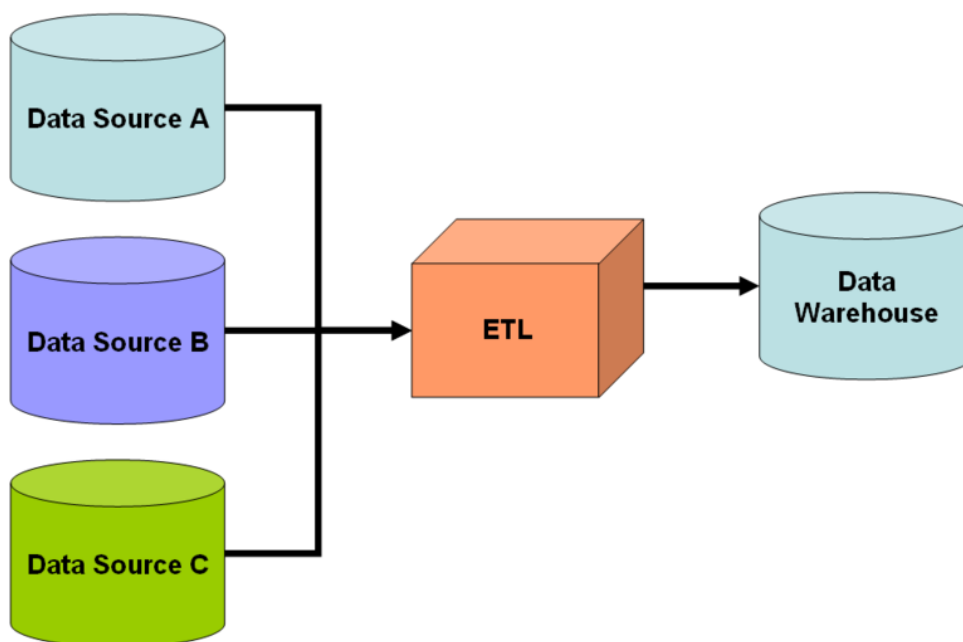
SSIS-liittymiä käytetään ETL-prosessien hallintaan. ETL-prosessin tarkoittaa tiedon siirtämistä tietokannasta A tietokantaan B. (Passionned group, 2012)

Esimerkiksi kauppaketjulla voi olla tietokannassa asiakasrekisteri kaikista kauppaketjun asiakkaista. Kauppaketjuun kuuluu 3 liikettä, joiden omien asiakasrekistereiden pohjalta päivitetään koko kauppaketjun asiakasrekisteri.

Tässä esimerkkitalanteessa kauppojen asiakasrekisterit ovat rakenteeltaan yhdenmukaisia, mutta kauppaketjun asiakasrekisteri on rakenteeltaan erilainen verrattuna kauppojen asiakasrekistereihin.

Tietojen siirtäminen suoraan kauppojen tietokannoista kauppaketjun tietokantaan ei siis ole mahdollista.

Tämä ongelma voidaan ratkaista luomalla SSIS-liittymä, joka suorittaa ETL-prosessin, jonka seurauksena kauppaketjun asiakasrekisteri päivitetään ajan tasalle. ETL-vaiheet on käyty läpi tämän esimerkkitapauksen avulla seuraavissa luvuissa. Kuvassa 5 on esitetty yksinkertaisen ETL-prosessin rakenne.



KUVA 5. Havainnollistava kuva ETL-prosessista (Datazoomers, 2013)

5.3.1 Extract eli purkuvaihe

Extract eli purkuvaiheessa SSIS-liittymä noutaa kauppojen asiakasrekisteritiedot kauppaketjun tietokantaan. Tietoja ei kuitenkaan tässä vaiheessa viedä varsinaiseen asiakasrekisteritauluun vaan esimerkiksi tietokantaan luotuun välitauluun. Välitaulu tarkoittaa tässä yhteydessä tietokannan taulua, jonka tarkoituksena on säilyttää kauppojen asiakasrekistereistä tulevia tietoja SSIS-liittymän suorittamisen ajan.

Yleinen tapa tämän tyyliseen tietojen siirtämiseen on siirtotiedoston käyttäminen. Kaupan järjestelmä siis generoi esimerkiksi tekstitiedoston, jossa asiakasrekisteritiedot ovat tietyssä järjestyksessä. SSIS-liittymä noutaa siirtotiedoston sille määritellystä sijainnista ja lukee sen sisältämät tiedot tietokannan välitauluun.

Välitaulu tyhjennetään aina liittymän suorittamisen aloituksen yhteydessä. Tämän toimenpiteen seurauksena välitaulussa on vain tuoreimmat siirtotiedot, eikä taulussa turhaan säilytetä vanhaa tietoa. Välitaulun tyhjennys suoritetaan liittymän ajamisen aloituksen yhteydessä siitä syystä, että edellisen siirron yhteydessä siirtyneet tiedot ovat välitaulussa tallessa seuraavaan siirtoon asti. Mahdollisissa virhetapauksissa siirrettyjä tietoja voidaan siis tarkastella tietokannan välitaulusta.

Purkuvaiheen jälkeen ollaan tilanteessa, jossa kauppojen asiakasrekisterit on siirretty sellaisenaan kauppaketjun tietokannan välitauluun.

5.3.2 Transform eli muutosvaihe

Transform eli muutosvaiheessa välitaulussa olevan tiedon oikeellisuus tarkastetaan. Välitaulun tieto myös muokataan tässä vaiheessa oikeaan muotoon niin, että se on valmis siirrettäväksi kauppaketjun varsinaiseen asiakasrekisteritauluun.

Tässä vaiheessa voidaan esimerkiksi liittää tietokannan tauluja välitaulun aineistoon mikäli kauppojen asiakasrekistereistä tulleissa tiedoissa on id-viittauksia, jotka vaativat liitoksia muihin tauluihin. Myös tiedon päivittäminen ja oikeellisuuden tarkistus suoritetaan tässä vaiheessa.

Välitaulussa olevia tietoja ei välttämättä tarvitse päivittää välitauluun vaan tietojen muokkaus onnistuu myös tietojen kanta-asiakastauluun siirron yhteydessä. Esimerkiksi tekstimuotoisessa kentässä oleva numero voidaan muuttaa numeromuotoiseen kenttään sopivaksi tässä yhteydessä käyttämällä SQL:n CONVERT-funktiota.

Kun edellämainitut vaiheet on suoritettu niin lopuksi määritellään ehdot välitaulusta varsinaiseen asiakasrekisteriin siirrettävälle tiedolle. Ehdot voivat olla esimerkiksi seuraavanlaisia:

- Kanta-asiakaskortin numeron tulee olla tietyn kaavan mukainen
- Asiakkaan henkilötietokentät eivät saa olla tyhjiä
- Vastaavalla henkilötunnuksella ei ole vielä henkilöä kauppaketjun kanta-asiakastaulussa

5.3.3 Load eli latausvaihe

Muutosvaiheen jälkeen rajausehdot täyttävät rivit ladataan kauppaketjun asiakasrekisteritauluun. Edellisessä luvussa asetettujen rajausehtojen mukaan kauppaketjun asiakasrekisteriin lisättiin siis uudet asiakkaat, joiden henkilötietokentät sisälsivät dataa ja kanta-asiakaskortin numero oli laskentakaavan mukainen.

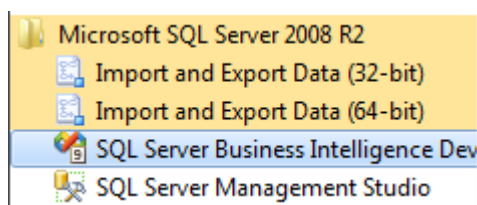
Tässä yhteydessä on myös mahdollista muodostaa logi siirretyistä ja siirrossa hylätyistä riveistä. Logiin voi muodostaa esimerkiksi omaan tietokantatauluunsa. Logiin voidaan lisätä esimerkiksi siirron aikaleima sekä yksinkertainen viesti, jossa kerrotaan kanta-asiakastauluun siirretyt uudet asiakkaat. Yksilöivänä tietona logiin voidaan lisätä asiakkaan nimen lisäksi myös asiakkaan henkilötunnus.

Logi mahdollistaa siirrettyjen ja hylättyjen tietojen tarkastelun jälkepäin. Logi on erittäin hyödyllinen esimerkiksi liittymässä siirrettyä tietoa koskevien virhetilanteiden selvityksessä.

Kun tiedot on siirretty kanta-asiakastauluun, voidaan siirtoaineisto siirtää määriteltyyn arkistointikansioon sekä lisätä tiedostoon aikaleima. Esimerkiksi arkistointikansiossa oleva tiedosto "siirtoaineisto_20121223_050000" on siirretty kansioon 23.12.2012 klo 05:00 . Tämä estää sen, että saman nimisten siirtotiedostojen kanssa tulisi duplikaattiongelmia, koska siirtotiedoston pitää aina olla nimetty samalla tavalla.

5.4 SSIS-liittymän toteuttaminen SQL SERVER Business Intelligence Studio -ohjelmalla

SSIS-liittymät rakennetaan SQL SERVER Business Intelligence Studio -ohjelmalla, joka on osa Microsoft SQL SERVER -ohjelmistoa. Mikäli SQL SERVER Business Intelligence Studio -ohjelma on asennettu Microsoft SQL SERVER -ohjelmiston asennuksen yhteydessä, niin ohjelman voi käynnistää käynnistysvalikon kautta kuvan 6 osoittamalla tavalla.



KUVA 6. SQL SERVER Business Intelligence Studio -ohjelman käynnistäminen käynnistä-valikosta

5.5 SSIS-liittymän toimintaperiaate

Yksinkertaisesti selitettynä SSIS-liittymä koostuu vaihtelevasta määrästä toiminnallisia elementtejä. Näiden elementtien toimintajärjestys määrittellään SQL SERVER Business Intelligence Studio-ohjelmassa. Elementeille määrittellään niiden toiminta sekä tämän jälkeen mitä elementin suorittamisen jälkeen tapahtuu. Elementeille voidaan määrittellä seuraava suoritettava elementti sen mukaan onko edellisen elementin suoritus onnistunut vai epäonnistunut. Myös molempien vaihtoehtojen määrittely yhdelle elementille on mahdollista.

Kuvassa 7 on esitetty näkymä SQL SERVER Business Intelligence Studiosta sekä yksinkertaisen esimerkiliittymän pohja. Liittymän suorittaminen alkaa Script task -elementistä, jossa tarkastetaan siirtotiedoston sijainti määrittelystä tiedostopolusta. Script task -elementit käyttävät oletuksena C#-ohjelmointikieltä. SQL SERVER Business Intelligence Studio mahdollistaa kuitenkin Script task -elementtien käyttämän ohjelmointikielen vaihtamisen myös Visual basic -ohjelmointikielen. (Microsoft, 2013d)

Mikäli Script task -elementin suoritus epäonnistuu tässä vaiheessa, niin liittymän suoritus keskeytyy. Tässä tilanteessa liittymä ei lisää mihinkään tietokannan tauluun virheilmoitusta suorituksen epäonnistumisesta koska kyseistä toiminnallisuutta ei ole esimerkiliittymään rakennettu.

Script task -elementin suorituksen onnistuessa siirrytään seuraavan elementin suoritukseen. Script task -elementistä lähtevä vihreä viiva osoittaa mikä elementti suoritetaan seuraavaksi Script task -elementin onnistuneen suorituksen jälkeen.

Kun Script task -elementissä on todettu siirtotiedoston sijaitsevan sille määritellyssä sijainnissa niin seuraavaksi esimerkkiliittymässä suoritetaan Data flow task -elementti, joka siirtää siirtotiedostossa olevat tiedot tietokannassa olevaan tiedonsiirron välitauluun.

Data flow task -elementeillä on sisäisiä ominaisuuksia, jotka pitää tässä vaiheessa asettaa, jotta liittymä toimisi oikein. Näihin ominaisuuksiin kuuluu esimerkiksi tiedonsiirron "mappaukset", joilla määritellään mikä siirtotiedoston tieto siirretään mihinkin tiedonsiirron välitaulun kenttään.

Kun tiedot on siirretty onnistuneesti siirtotiedostosta tietokannan välitauluun, niin liittymän suorittamisessa siirrytään seuraavaan vaiheeseen. Seuraavassa Data flow task -elementissä siirretään tiedot välitaulusta tietokannan varsinaiseen tauluun. Tässä Data flow task -elementissä voidaan kutsua esimerkiksi tietokannan proseduuria, jossa tarkistetaan onko tietokannan välitaulussa olevia tietoja jo ennestään tietokannan varsinaisessa taulussa. Mikäli välitaulussa on rivejä, jotka löytyvät kannasta jo ennestään niin ne voidaan tässä yhteydessä rajata siirrosta pois ja siirtää tietokannan varsinaiseen tauluun vain uudet rivit.

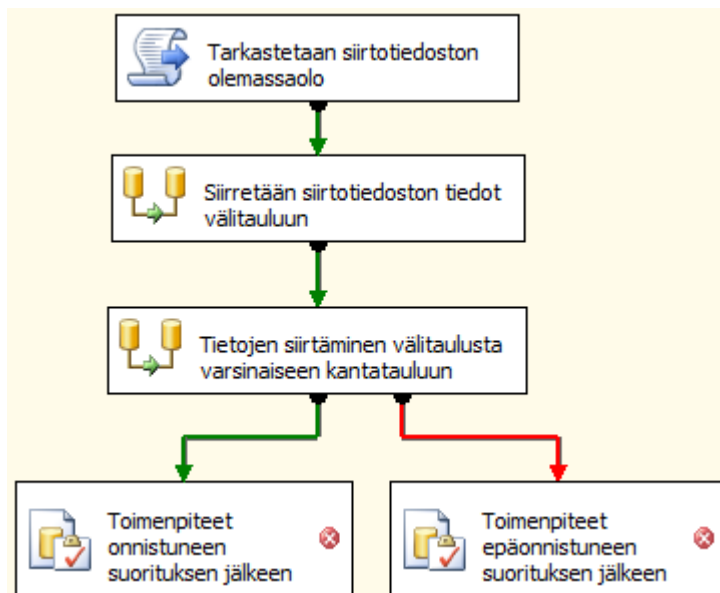
Tämän jälkeen siirrytään Execute SQL task -elementtiin, johon liittymän suorittaminen päättyy. Kuten kuvasta 7 näkyy, liittymän lopussa Data flow task -elementin jälkeen on kaksi Execute SQL task -elementtiä. Data flow task -elementistä lähtevän viivan värikoodi kertoo kumpi Execute SQL task -elementti suoritetaan missäkin tilanteessa.

Kun Data flow task -elementin suoritus onnistuu, niin siirrytään vihreän viivan osoittamaan "Toimenpiteet onnistuneen suorituksen jälkeen" Execute SQL task -elementtiin. Execute SQL task -elementissä suoritetaan SQL-lauseita kuten elementin nimestäkin pystyy päättelemään. Tässä elementissä voidaan esimerkiksi lisätä tietokannan logitauluun tieto tietokantaan lisättyjen rivien lukumäärästä.

Mikäli Data flow task -elementin suoritus epäonnistuu, niin siirrytään punaisen viivan osoittamaan "Toimenpiteet epäonnistuneen suorituksen jälkeen" Execute SQL task -elementtiin. Tässä elementissä voidaan lisätä SQL-lauseilla esimerkiksi tietokannan logitauluun ilmoitus liittymän suorittamisen epäonnistumisesta.

SSIS-liittymissä voi määrittää elementistä seuraavaan siirtymiselle kolme eri ehtoa. Kuvan 7 esimerkissä on esitetty näistä kaksi, eli elementin onnistuneen tai epäonnistuneen suorituksen jälkeen seuraavaan elementtiin siirtyminen.

Lisäksi SSIS-liittymissä voi käyttää elementtien välillä siirtymisessä "Completion"-ehtoa. Tämä siis tarkoittaa, että elementistä siirrytään seuraavaan elementtiin riippumatta elementin suorituksen lopputuloksesta. Nämä siirtymät elementtien välillä on merkitty liittymissä sinisillä viivoilla.



KUVA 7. Esimerkkikuva SQL SERVER Business Intelligence Studio toiminnallisista elementeistä ja suoritusjärjestyksestä

5.6 Virhetilanteet liittymän suorituksessa

Mahdollisissa virhetilanteissa, joissa liittymän suorittaminen ei onnistu on tärkeää, että liittymän hallinnoijalle ilmoitetaan virhetilanteesta. SSIS-liittymään on mahdollista luoda komponentti, joka virhetilanteessa tallentaa virhetilanteen antaman virhekuvaus. Tämä virhekuvaus voidaan tallentaa muuttujaan, joka voidaan lisätä käyttäjälle lähetettävään sähköpostiin, mikäli kyseinen ominaisuus on liittymään rakennettu.

Virhetilanteesta lähtevän sähköpostituksen voi rakentaa esimerkiksi liittymän viimeiseksi osaksi joka suoritetaan jokaisella liittymän ajokerralla. Sähköpostituksen rajauksiin on tässä tapauksessa viisasta lisätä ehto, että sähköpostia ei lähetetä, mikäli virhetilanteen kuvauksen tallentavassa muuttujassa ei ole arvoa. Tällä tavalla liittymä ei lisää sähköpostitauluun lähtevää postia, jos liittymässä ei tapahdu virhetilannetta.

5.7 Liittymän suorittamisen ajastaminen

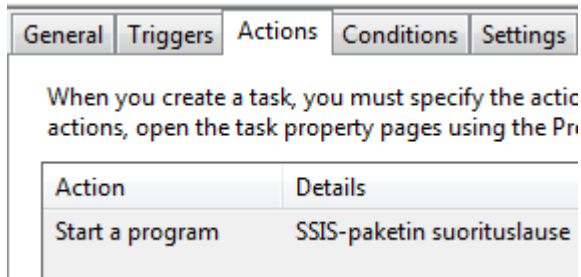
Yleensä tämän tapaisia liittymiä suoritetaan kerran vuorokaudessa. Liittymien suorittaminen on useimmiten ajoitettu yöaikaan, jolloin sovelluksissa eikä kannassa ole juurikaan muuta toimintaa eikä raskautta.

Esimerkkitapauksessa Microsoft SQL SERVER -ohjelmisto on asennettu kauppaketjun erilliselle palvelimelle, jossa tietokantaa sekä SSIS-liittymiä hallinnoidaan. Liittymän ajastaminen toimimaan esimerkiksi kerran vuorokaudessa klo 05:00 onnistuu käyttämällä Windowsin Task scheduler -ohjelmaa.

Name	Status	Triggers	Next Run Time	Last Run Time
🕒 Esimerkki ajastamisesta	Ready	At 5:00 every day	24.12.2012 5:00:00	Never

KUVA 8. Kuva Windows task scheduler -ohjelmasta ja esimerkiajastuksesta

Kuvassa 8 on esitetty esimerkki ajastamisesta. Kuvan esimerkiajastus suorittaa määritellyt toimenpiteet joka päivä klo 05:00. Tarvittaessa ajastuksen voi tehdä toimimaan myös esimerkiksi viikoittain, kuukausittain tai pelkästään arkipäivinä.



KUVA 9. Kuvakaappaus Windows task scheduler-ohjelmasta. Kuvassa on ajastuksen Actions-välilehti, jossa asetetaan ajastuksen suorittamat toimenpiteet.

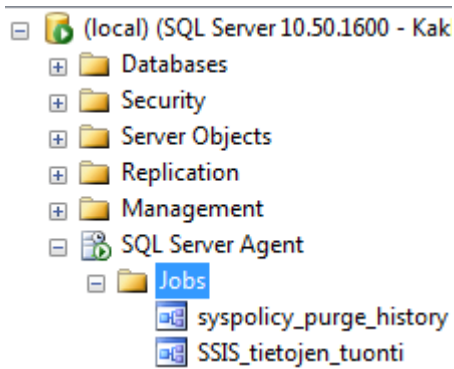
Kuvassa 9 on esitetty kuvakaappaus ajastuksen Actions-välilehdeltä. Ajastuksen yhteyteen voidaan asettaa yksi kolmesta eri toiminnosta. Valittavissa ovat sähköpostin lähettäminen, viestin esittäminen tai ohjelman suorittaminen.

SSIS-liittymän suorittaminen onnistuu valitsemalla ohjelman suorittamisen eli "Start a program" ja lisäämällä toiminnoksi SSIS-liittymän suorituslauseeseen.

SSIS-liittymän voi suorittaa palvelinympäristössä esimerkiksi kutsumalla ajastuksen yhteydessä dtexec.exe-sovellusta ja syöttämällä tälle SSIS-paketin suorittamiseen tarvittavat tiedot. Michael Entinin blogikirjoituksen mukaan tämä on luotettava tapa suorittaa SSIS-liittymiä ajastuksen kanssa toimittaessa. (Entin, 2007)

5.8 SSIS-liittymän suorittaminen SQL SERVER Management Studio -ympäristössä

SSIS-liittymät asennetaan Microsoft SQL SERVER Management Studio SQL Server Agenttiin job-elementteinä. Tämä mahdollistaa liittymien voidaan suorittamisen myös SQL SERVER Management Studio kautta. Kuvassa 10 on esitetty Job-elementtien sijainti SQL SERVER Management Studio -ohjelmistossa.



KUVA 10. Jobien sijainti Microsoft SQL SERVER Management Studiassa

Käytän tässä opinnäytetyössä Job-termiä, koska työkokemukseni mukaan se on työelämässä yleisesti käytetty termi. Job-termille ei myöskään ole olemassa mielestäni tarpeeksi kuvaavaa suomenkielistä termiä.

5.9 Uuden job-elementin asentaminen SQL SERVER Management Studio -ympäristöön

SSIS-liittymän asentaminen SQL SERVERiin jobina tapahtuu lisäämällä uusi jobi SQL Server Agenttiin. Ensimmäisenä jobille annetaan nimi kuvassa 11 näkyvään kenttään.

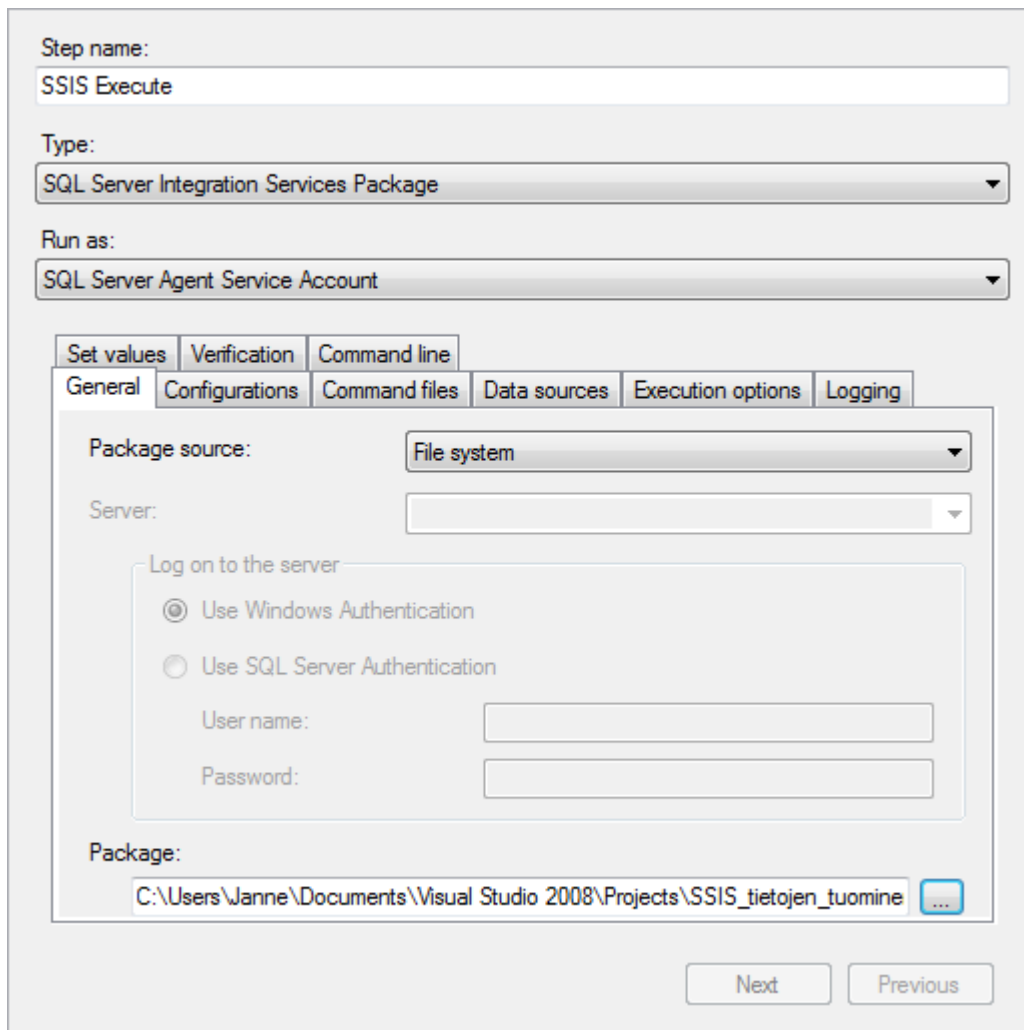
The image shows a dialog box for creating a new job. It has three fields: 'Name' with the text 'SSIS_tietojen_tuonti', 'Owner' with 'Esimerkki\Janne', and 'Category' with a dropdown menu showing '[Uncategorized (Local)]'. There are ellipsis buttons next to the Owner and Category fields.

KUVA 11. Uuden jobin nimeäminen

Tämän jälkeen uuden jobin luonnissa valitaan Steps-välilehti. Steps-välilehdelle luodaan toiminto eli askel, joka käynnistää liittymän suorittamisen. Esimerkkiliittymässä yksi askel riittää liittymän suorittamiseen.

Mikäli liittymä tarvitsee muuttujalle tai muuttujille arvoja liittymän suorittamisen yhteydessä niin jobille voidaan luoda useita askelia, joilla liittymän suorittaminen käynnistetään. Askelille voidaan tässä yhteydessä määritellä muuttujia ja niille arvoja, jotka liittymään vievät jobin käynnistämisen yhteydessä.

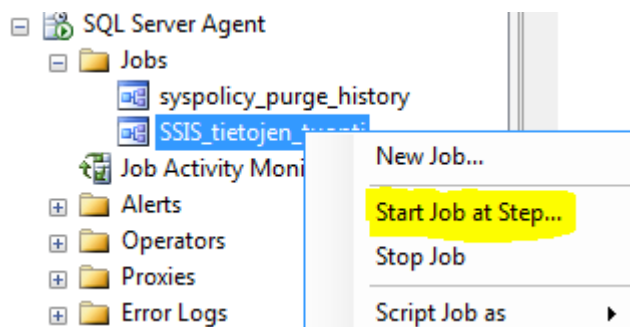
Kuvassa 12 on esitetty Steps-välilehdelle lisättävät asetukset, jotka mahdollistavat SSIS-liittymän suorittamisen jobin avulla. Aluksi askelille annetaan kuvaava nimi. Tämän jälkeen askeleen tyyppiä pitää vaihtaa "SQL Server Integration Services Package". Tällä tavalla määritellään, että kyseessä on SSIS-paketin suorittaminen. Package source -valikosta valitaan "File system", jonka jälkeen Package-kenttään noudetaan SSIS-liittymäpaketin .dtsx-tiedoston polku. Tämän jälkeen askeleen asetukset hyväksytään klikkaamalla OK. Jobiin ei tarvitse tehdä muita asetuksia, joten jobin asetuksista voi myös poistua klikkaamalla OK.



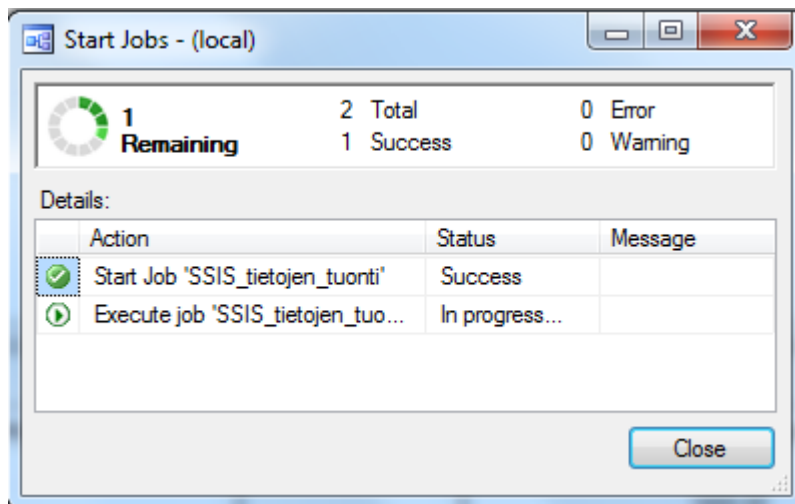
KUVA 12. Askeleen asetukset SSIS-liittymän suorittamista varten

5.10 Jobin suorittaminen SQL SERVER Management Studio -ympäristössä

Jobien suorittaminen SQL SERVER Management Studiossa tapahtuu kuvan 13 osoittamalla tavalla. Jobia klikataan hiiren oikealla painikkeella ja valitaan "Start job at step". Koska luodussa jobissa ei ole kuin yksi askel, alkaa liittymän suorittaminen välittömästi klikkauksen jälkeen. Mikäli liittymässä olisi useampia askelia, niin tässä vaiheessa käyttäjän pitäisi valita mistä askeleesta liittymän suorittaminen aloitetaan. Kuvassa 14 on esitetty kuvankaappaus jobin suorittamisesta.



KUVA 13. Jobin suorittamisen aloittaminen



KUVA 14. Kuvankaappaus jobin suorittamisesta

6 SSIS-TEKNIIKAN HYÖDYNTÄMINEN HENKILÖSTÖHALLINNAN SOVELLUKSESSA

6.1 Yleistä

Tässä osiossa käydään läpi käytännön esimerkkien kautta SSIS-liittymien toimintaa. SSIS-liittymiä käytetään tätä opinnäytetyötä varten luodun tietokannan kanssa. Tietokannan luomiseen tarvittavat SQL-lauseet, SSIS-liittymät sekä SSIS-liittymien toimintaa havainnollistavat videot ovat opinnäytetyön materiaalipaketissa.

Opinnäytetyöhön liittyvän materiaalipaketin voi ladata osoitteesta

https://dl.dropbox.com/u/20386846/Oppari/Opinnaytetyomateriaali_Janne_Vanttinen.zip

Opinnäytetyömateriaali on pakattu .zip-pakettiin, jonka sisällön voi purkaa esimerkiksi Winzip- tai Winrar-purkuohjelmalla.

6.2 Tietokanta

SSIS-liittymien esimerkkien pohjaksi loin yksinkertaisen relaatiotietokannan, joka käsittää henkilöstöhallinnan perustiedot. Pyrin pitämään kannan mahdollisimman yksinkertaisena, jotta sen rakenne ja toiminta olisivat helposti ymmärrettävissä myös tietokantojen parissa vähemmän työskennelleille. Kannan toteutuksessa otin kuitenkin huomioon, että kannassa olisi tiettyjä perustoiminnallisuuksia kuten triggerit ja relaatiot samaan tauluun.

Kantaan sisältyvät tiedot työntekijöistä, työntekijöiden yksiköistä, työntekijöiden nimikkeistä, työntekijöiden palkoista, yksikköjen esimiehistä sekä yksikköjen pääyksiköistä eli organisaatorakenteessa ylempänä olevasta yksiköstä.

6.2.1 Tietokannan taulut

Tässä luvussa esitetään tietokannan sisältämät taulut. Taulukossa 3 on esitetty Työntekijä-tilun rakenne. Taulukossa 4 on esitetty Nimike-tilun rakenne. Taulukossa 5 on esitetty Yksikkö-tilun rakenne.

TAULUKKO 3. Esimerkkitoiteutuksen Työntekijä-tilu

Kentän nimi	Datatyppi	Selite
tyontekija_id	Numeerinen tieto	Perusavain, yksilöivä ID-numero työntekijälle.
tyontekija_nimi	Tekstimuotoinen tieto	Työntekijän etu- ja sukunimi. Pakollinen tieto.
tyontekija_henkilonumero	Numeerinen tieto	Yksilöivä henkilönnumero jokaiselle työntekijälle. Pakollinen tieto.
tyontekija_nimike_id	Numeerinen tieto	ID-viittaus Nimike-tiluun, josta haetaan työntekijän nimike. Pakollinen tieto.
tyontekija_yksikko_id	Numeerinen tieto	ID-viittaus Yksikkö-tiluun, josta haetaan

		työntekijän yksikkö. Pakollinen tieto.
tyontekija_palkka	Numeerinen tieto	Työntekijän kuukausipalkka
tyontekija_lisatty_ts	Päivämäärä ja aika	Aikaleima milloin työntekijän tiedot on lisätty tietokantaan, täytetään taulun triggerissä

TAULUKKO 4. Esimerkkitoteutuksen Nimike-taulu

Kentän nimi	Datatyyppe	Selite
nimike_id	Numeerinen tieto	Perusavain, yksilöivä ID-numero nimikkeelle
nimike_nimi	Tekstimuotoinen tieto	Nimikkeen nimi. Pakollinen tieto.

TAULUKKO 5. Esimerkkitoteutuksen Yksikkö-taulu

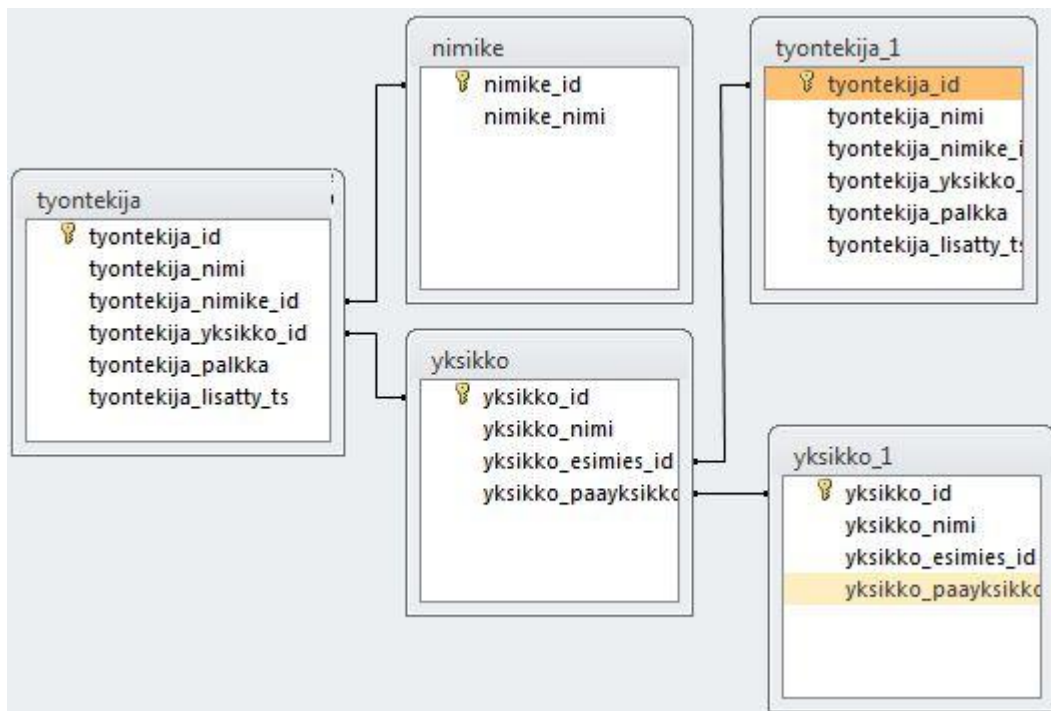
Kentän nimi	Datatyyppe	Selite
yksikko_id	Numeerinen tieto	Perusavain, yksilöivä ID-numero yksikölle.
yksikko_nimi	Tekstimuotoinen tieto	Yksikön nimi. Pakollinen tieto.
yksikko_esimies_id	Numeerinen tieto	ID-viittaus Työntekijä-tauluun, josta haetaan yksikön esimiestieto. Pakollinen tieto.
yksikko_paayksikko_id	Numeerinen tieto	ID-viittaus Yksikkö-tauluun, mistä haetaan yksikön pääyksikkö jos yksiköllä sellainen on.

6.2.2 Tietokannan relaatiot

Kuvassa 15 esitetään tietokannan sisältämät relaatiot. Työntekijä-taulu sisältää relaatiot Nimike- ja Yksikkö-tauluihin, joista työntekijälle haetaan nimike- ja yksikkötiedot.

Yksikkö-taulu sisältää relaation Yksikkö-tauluun eli Yksikkö-taulussa on kenttä, jossa määritellään yksikön ylemmän yksikön ID. Tästä kentästä on relatio Yksikkö-taulun ID-kenttään. Yksikkö-taulun relatio itseensä tällä tavalla mahdollistaa organisaatorakenteen luonnin.

Yksikkö-taulusta on lisäksi relatio Työntekijä-tauluun, jonka kautta haetaan yksikölle esimies.



KUVA 15. Esimerkitoteutustietokannan relaatiot

6.2.3 Tietokannan trigger

Tietokanta sisältää yhden triggerin. Kyseessä on Työntekijä-tauluun liitetty triggeri, joka laukee kun Työntekijä-tauluun lisätään uusi rivi. Kyseessä on siis insert-triggeri. Triggerin nimi on `tyontekija_insert_trigger`. Triggerin sisältö on esitetty kuvassa 3. Triggerin luova koodi on opinnäytetyöhön kuuluvan tiedostopaketin tiedostossa `01_tietokannan_luonti.txt`.

Triggerillä päivitetään Työntekijä-tauluun aikaleima tiedon lisäämishetkestä kyseiseen tauluun. Suuremmissa toteutuksissa on mielestäni viisasta lisätä erilliset aikaleimat tiedon lisäämiselle sekä edelliselle muokkaukselle. Esimerkiksi virhetilanteita selvittäessä aikaleimat helpottavat huomattavasti ongelman selvittäjän työtä kun kantaan vietyjä muutoksia voi tarkastella aikaleimojen perusteella.

6.2.4 Tietokannan luominen

Tietokannan luomiseen Microsoft SQL SERVER Management studiossa tarvittavat scriptit ovat opinnäytetyön liitteenä. Samaan tiedostoon on koottu kaikkien taulujen ja Työntekijä-taulun triggerin luomiseen tarvittavat scriptit. Tämän scriptin suorittaminen Microsoft SQL SERVER Management Studiossa luo siis valittuun tietokantaan Työntekijä-, Yksikkö- ja Nimike-taulut sekä Työntekijä-taululle triggerin `tyontekija_insert_trigger`.

Scripti löytyy tiedostosta `01_tietokannan_luonti.txt`.

6.3 Tietokannan sisältämä tieto

Opinnäytetyön liitteenä on scriptti 02_taulujen_tayttaminen.txt. Tällä scriptillä lisätään kaikkiin tietokannan tauluihin rivejä. Rivien lisääminen tietokannan tauluihin helpottaa SSIS-liittymien testitapauksien luomista sekä liittymän toimintojen havainnollistamista erilaisissa tilanteissa.

Pyrin pitämään tiedot mahdollisimman yksinkertaisina, että liittymien ja kannan toimimisen ymmärtäminen olisi helppoa myös SQL SERVER –ohjelmistoa vähän käyttäneille henkilöille.

Seuraavissa luvuissa on esitelty taulukoittain jokaiseen tietokannan tauluun lisättävät testiarvot.

6.3.1 Työntekijä-tilun testiarvot

Taulukossa 6 on esitetty Työntekijä-tilun sisältämät tiedot.

TAULUKKO 6. Esimerkkiteutuksen Työntekijä-tiluun lisättävät testiarvot

Työntekijän nimi	Nimike	Henkilönum.	Yksikkö	Palkka
Teemu Testaaja	Esimies	10	Alimman tason yksikkö A	1800
Tiina Testaaja	Esimies	11	Alimman tason yksikkö B	1800
Teppo Testaaja	Esimies	12	Alimman tason yksikkö C	1800
Erkki Esimerkki	Alainen	13	Alimman tason yksikkö A	1600
Eino Esimerkki	Alainen	14	Alimman tason yksikkö B	1600
Henna Henkilö	Alainen	15	Alimman tason yksikkö C	1600
Heikki Henkilö	Esimies	16	Keskitason yksikkö A	2200
Hanna Henkilö	Alainen	17	Keskitason yksikkö A	1800
Hilma Henkilö	Esimies	18	Keskitason yksikkö B	2200
Walter White	Esimies	19	Ylin yksikkö	2800

6.3.2 Nimike-tilun testiarvot

Taulukossa 7 on esitetty Nimike-tilun sisältämät tiedot.

TAULUKKO 7. Esimerkkiteutuksen Nimike-tiluun lisättävät testiarvot

Nimike
Alainen
Esimies

6.3.1 Yksikkö-tilun testiarvot

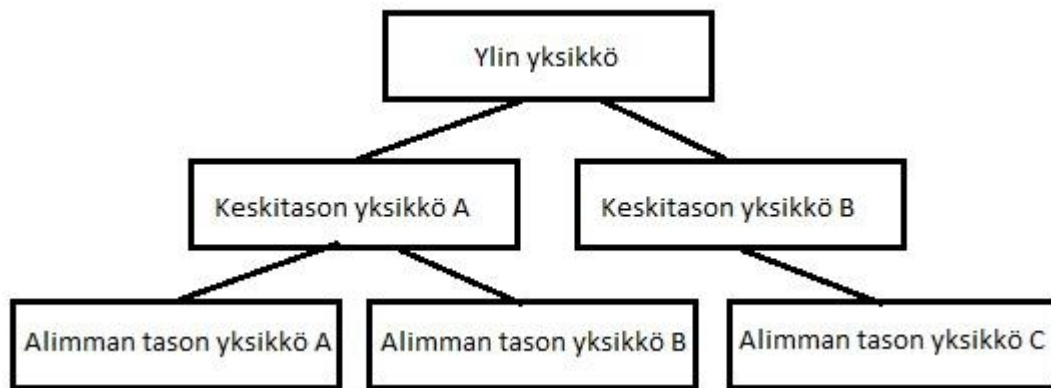
Taulukossa 8 on esitetty Yksikkö-tilun sisältämät tiedot.

TAULUKKO 8. Esimerkkiteutuksen Yksikkö-tiluun lisättävät testiarvot

Yksikön nimi	Yksikön esimies	Pääyksikkö
Ylin yksikkö	Walter white	
Keskitason yksikkö A	Heikki Henkilö	Ylin yksikkö
Keskitason yksikkö B	Hilma Henkilö	Ylin yksikkö
Alimman tason yksikkö A	Teemu Testaaja	Keskitason yksikkö A
Alimman tason yksikkö B	Tiina Testaaja	Keskitason yksikkö A
Alimman tason yksikkö C	Teppo Testaaja	Keskitason yksikkö B

6.3.2 Esimerkkiteutuksen organisaatorakenne

Työntekijän esimiestä ei ole merkattu työntekijän omiin tietoihin koska relaatiotietokanta mahdollistaa tämän tiedon hakemisen toisesta taulusta relaatioiden kautta. Tässä tapauksessa siis työntekijän yksikkötiedon kautta haetaan yksikön esimies, joka samalla on yksikköön kuuluvan työntekijän esimies. Esimerkkiteutuksen organisaatorakenne on esitetty kuvassa 16.



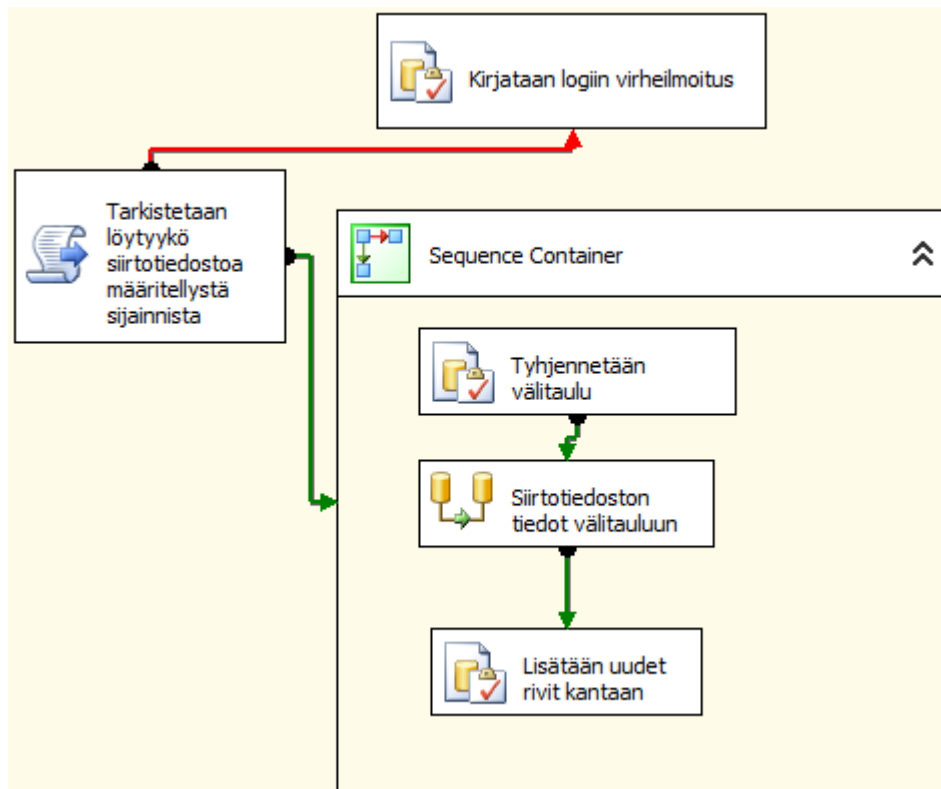
KUVA 16. Esimerkkiteutuksen organisaatorakenne

6.4 SSIS-liittymä työntekijätietojen tuomiseen siirtotiedostosta

Opinnäytetyön liitteenä olevalla "SSIS_tietojen_tuominen_kantaan"-liittymällä tuodaan siirtotiedostosta työntekijätietoja tietokannan Työntekijä-tauluun. Siirtotiedoston sijainnin ja rakenteen pitää olla määritellyn mukainen, että liittymä toimii oikein. Seuraavissa luvuissa on esitetty liittymän rakenne sekä tiedonsiirron rajaukset. Opinnäytetyön liitteenä on videotiedosto "liittymän_suurittaminen_kehitysympäristössä", jossa tämä liittymä suoritetaan SQL SERVER Intelligence Studio-ympäristössä. Video auttaa havainnollistamaan SSIS-liittymien toimintaa.

Valitsin esimerkiliittymäni siirtotiedostomallin, koska työkokemukseni mukaan se on yleisin tapa siirtää tietoja kahden erilaisen järjestelmän välillä.

6.4.1 Liittymän rakenne

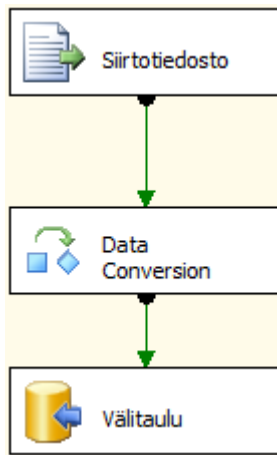


KUVA 17. SSIS-liittymän rakenne

Kuvassa 17 näkyy SSIS-liittymän rakenne. Liittymän suoritus alkaa Script task -elementistä, jossa tarkastetaan onko siirtotiedosto sille määritellyssä sijainnissa. Siirtotiedoston sijainniksi liittymässä on määritelty tiedostopolku C:\SSIS\03_siirtotiedosto.txt. Mikäli siirtotiedostoa ei tästä sijainnista löydy niin liittymä siirtyy Execute SQL task -elementtiin, joka kirjaa tietokannan tapahtumalogitauluun virheilmoituksen siirtotiedoston puuttumisesta sekä aikaleiman.

Mikäli siirtotiedosto löytyy määritellystä sijainnista, niin liittymän suorittaminen siirtyy Sequence container -elementtiin, joka sisältää elementit joita käytetään siirtotiedoston sisältämien tietojen siirtämisessä tietokantaan.

Ensimmäisessä elementissä tyhjennetään tietokannan tiedonsiirron välitaulu edellisten siirtojen riveistä. Tämän jälkeen Data flow task -elementissä siirretään siirtotiedostosta tiedot välitauluun. Kuvassa 18 esitetään Data flow task -elementin rakenne.



KUVA 18. Data flow task-elementin rakenne

Viimeisessä vaiheessa SQL-lauseilla suoritetaan tietojen siirto välitaulusta työntekijätauluun. Tähän yhteyteen on myös tehty tiedonsiirron rajaukset. Tietojen siirtämisessä välitaulusta työntekijätauluun on voimassa seuraavat rajaukset:

- Samalla henkilönumerolla ei ole työntekijää tietokannassa
- Työntekijällä on henkilönnumero
- Työntekijän tiedoissa oleva nimike löytyy tietokannasta
- Työntekijän tiedoissa oleva yksikkö löytyy tietokannasta

Edellämainittujen rajausten jälkeen jäljelle jääneet rivit siirretään työntekijätauluun.

6.4.2 Siirtotiedosto

Siirtotiedosto "03_siirtotiedosto.txt" on opinnäytetyöhön kuuluvassa tiedostopakettissa.

Siirtotiedoston pitää olla kansiossa C:\SSIS kun liittymä suoritetaan. Siirtotiedoston ensimmäinen rivi on otsikorivi ja tämän jälkeen työntekijätiedot ovat riveittäin. Tiedot erotetaan toisistaan ;-merkillä. Taulukossa 9 on esitetty siirtotiedoston rakenne sekä sen sisältämät työntekijätiedot.

TAULUKKO 9. Siirtotiedoston sisältö

Nimi	Nimike	Hlönro	Yksikkö	Palkka
Audrey Horne	Alainen	31	Alimman tason yksikkö A	1650,25
Ben Horne	Alainen	32	Alimman tason yksikkö B	1780,50
Sylvia Horne	Alainen	33	Keskitasen yksikkö B	1800,00
Laura Palmer	Alainen	11	Ylin yksikkö	2600,80
Sarah Palmer	Alainen		Keskitasen yksikkö A	1986,90
Leland Palmer	Alainen	34	Keskitasen yksikkö D	2000,00

Edellisessä luvussa esitettyjen tiedonsiirron rajausehtojen mukaan siirtomateriaalista työntekijätauluun siirretään kolme ensimmäistä riviä. Tämä edellyttää sitä, että tietokanta on

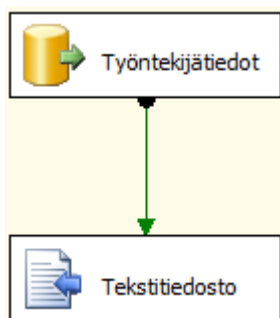
täytetty opinnäytetyön liitteenä olevalla datalla. Työntekijätauluun lisätyt rivit voi tarkastaa työntekijätaulusta aikaleiman perusteella.

6.5 SSIS-liittymä työntekijä tietojen hakemiseen kannasta

Opinnäytetyön liitteenä olevalla "SSIS_tietojen_haku_kannasta"-liittymällä haetaan kaikki työntekijätaulun sisältämät rivit ja viedään ne tekstitiedostoon. Tämä liittymä käyttää tietokannassa olevaa tyontekijatiedot-näkymää tietojen hakemiseen. Liittymän luoma tekstitiedosto löytyy tiedostopolusta C:\SSIS\tyontekijatiedot.txt

6.5.1 Liittymän rakenne

Liittymän rakenne on yksinkertainen. Liittymä sisältää yhden Data flow task -elementin, jonka rakenne on esitetty kuvassa 19.



KUVA 19. SSIS-liittymän Data flow task -elementin rakenne

Työntekijätiedot haetaan tyontekijatiedot-näkymästä. Tässä näkymässä haetaan työntekijätaulusta työntekijätiedot sekä relaatioilla tekstitiedostoa varten työntekijöiden yksiköiden nimet ja nimikkeet. Näkymässä ei ole minkäänlaisia rajoituksia vaan kaikki työntekijätaulun sisältämät rivit viedään tekstitiedostoon.

6.5.2 Tekstitiedoston rakenne

Taulukossa 10 on esitetty liittymän luoman tekstitiedoston rakenne sekä esimerkkietoja.

TAULUKKO 10. Tekstitiedoston rakenne

Nimi	Hlönro	Palkka	Nimike	Yksikkö
Teemu Testaaja	10	1800,00	Esimies	Alimman tason yksikkö A
Tiina Testaaja	11	1800,00	Esimies	Alimman tason yksikkö B

7 POHDINTA

En tehnyt opinnäytetyötä toimeksiantona miltään yritykseltä vaan teoriapohjaisena projektina. Olen työssäni tutustunut SSIS-tekniikkaan sekä ETL-prosessiin ja tätä kautta kiinnostuin tutustumaan lähemmin SSIS-tekniikan mahdollisuuksiin ja ETL-prosessiin.

Aloitin opinnäytetyön tutustumalla teoriatasolla SSIS-tekniikkaan sekä Microsoft SQL SERVER –ohjelmaan, johon SQL SERVER Intelligence Studio-kuuluu. Tämän jälkeen loin tietokannan, jossa käytin opinnäytetyössä läpikäytyjä Microsoft SQL SERVER –ohjelman perustoimintoja. Tietokantaan liittyen loin kaksi SSIS-liittymää, joiden kautta havainnollistin SSIS-tekniikan mahdollisuuksia ja ETL-prosessia.

Opinnäytetyötä tehdessä tutustuin myös muihin tekniikoihin, joita voi käyttää ETL-prosessin toteutukseen. Opinnäytetyötä tehdessäni mietin olisiko mikään ohjelmisto yksinkertaisesti paras ETL-prosessin toteutukseen. Päädyin lopputulokseen, että ohjelmistoa valittaessa pitää aina miettiä mitä muita ohjelmistoja tietokannan ylläpidossa käytetään. Microsoft SQL SERVER –ohjelmalla ylläpidettyjen tietokantojen yhteydessä SSIS-tekniikka on mielestäni paras vaihtoehto, koska se on suunniteltu toimimaan saumattomasti yhteen Microsoft SQL SERVER –ohjelman kanssa. Jos tietokantaa ylläpidettäisiin esimerkiksi Oraclella niin SSIS-tekniikka ei varmasti olisi paras vaihtoehto. Siinä tapauksessa mieltäisin tietokannan kokoa ja käyttötarkoitusta ja valitsisin niiden kautta toteutukseen parhaiten sopivan tekniikan ETL-prosessin toteuttamiseen.

Opinnäytetyön tekeminen oli mielenkiintoista koska olen kiinnostunut SSIS-tekniikasta ja sen tarjoamista mahdollisuuksista tietokantojen kanssa työskenneltäessä. Opinnäytetyötä tehdessä tutustuin myös paremmin ETL-prosessiin.

Suurimpia ongelmia opinnäytetyötä tehdessä aiheuttivat tekniset ongelmat Microsoft SQL SERVER –ohjelman kanssa. Microsoft SQL SERVER –ohjelma ei aluksi toiminut omalla tietokoneellani, joten minulla oli vaikeuksia käytännön toteutuksien aloittamisen kanssa. Sain kuitenkin ongelmat selvitettyä ja ohjelman toimimaan.

Opinnäytetyön tekeminen sujui teknisistä ongelmista huolimatta sujuvasti ja opinnäytetyö syntyi lopulta yllättävän nopeasti. Mielestäni sain käytyä kattavasti läpi SSIS-tekniikan ominaisuuksia ja tutustuin samalla itse enemmän SSIS-tekniikkaan ja ETL-prosessiin.

LÄHTEET

Allen Christopher [2004]. *Database basics* [verkkodokumentti] [viitattu 22.12.2012]. Saatavissa: http://www.mhprofessional.com/downloads/products/0072255404/0072255404_ch01.pdf

Datazoomers [2013]. *Data integration* [verkkodokumentti] [viitattu 17.3.2013]. Saatavissa: <http://www.datazoomers.com/data-integration>

Entin Michael [2007]. *Running SSIS package programmatically* [verkkojulkaisu] [viitattu 23.12.2012]. Saatavissa: <http://blogs.msdn.com/b/michen/archive/2007/03/22/running-ssis-package-programmatically.aspx>

FunctionX [2013]. *Installing Microsoft SQL Server* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: <http://www.functionx.com/sqlserver/Lesson01.htm>

Granroth Kari [2001]. *SQL Kieli* [verkkodokumentti] [viitattu 21.12.2012]. Saatavissa: edu.phkk.fi/Opiskelu/sqlkieli/SQL%20kieli.doc

Kettunen Eero [2001]. *Relaatiotietokantasuunnittelun aakkosia* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: <http://www.lpt.fi/it/opetus/tietokantasuunnittelu/perusteita.html>

Linstedt Dan [2013]. *Informatica, Abinitio, SSIS, OWB, Pentaho ETL, Talend* [verkkojulkaisu] [viitattu 16.2.2013]. Saatavissa: <http://empoweredholdings.com/2010/05/informatica-abinitio-ssis-owb-pentaho-etl-talend/>

Microsoft [2013a]. *Connecting to SQL Server over the Internet* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: [http://msdn.microsoft.com/en-us/library/ms175483\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms175483(v=sql.105).aspx)

Microsoft [2013b]. *Tietokannan suunnittelun perusteet* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: <http://office.microsoft.com/fi-fi/access-help/tietokannan-suunnittelun-perusteet-HA001224247.aspx>

Microsoft [2013c]. *Tietoja Access-tietokannan ja Access-projektien tietotyypien eroista* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: <http://office.microsoft.com/fi-fi/access-help/tietoja-access-tietokannan-ja-access-projektin-tietotyypien-eroista-HP005274573.aspx>

Microsoft [2013d]. *Integration Services in Business Intelligence Studio* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: [http://msdn.microsoft.com/en-us/library/ms174181\(v=sql.100\).aspx](http://msdn.microsoft.com/en-us/library/ms174181(v=sql.100).aspx)

Mohan T K Nithin [2010]. *Evolution of SQL Server from SQL 2000 to SQL Server 2008 R2* [verkkojulkaisu] [viitattu 21.12.2012]. Saatavissa: <http://www.nitrix-reloaded.com/2010/05/09/evolution-of-sql-server-from-sql-2000-to-sql-server-2008-r2/>

Passionned group [2012]. *What is ETL, Extract Transform and Load?* [verkkojulkaisu] [viitattu 23.12.2012]. Saatavissa: <http://www.passionned.com/data-integration/etl-extract-transform-and-load/>

Redware [2001]. *SQL Server Overview* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: http://www.redware.com/handbooks/sql_server_handbook/sql_server_overview.html

Simba technologies [2013]. *SQL* [verkkojulkaisu] [viitattu 22.12.2012]. Saatavissa: <http://www.simba.com/sql.htm>

Wells Garth [2001]. *An Introduction to Triggers – Part I* [verkkojulkaisu] [viitattu 2.2.2013]. Saatavissa: <http://www.sqlteam.com/article/an-introduction-to-triggers-part-i>

