

Santeri Niemi

# **Business Intelligence Markup Language hyödyntäminen tietovarastojen kehityksessä**

Opinnäytetyö

Kevät 2020

SeAMK Tekniikka

Tietotekniikka



SEINÄJOEN AMMATTIKORKEAKOULU  
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

## Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikka

Tutkinto-ohjelma: Tietotekniikan insinööri

Suuntautumisvaihtoehto: Tietoverkkotekniikka

Tekijä: Santeri Niemi

Työn nimi: Business Intelligence Markup Language hyödyntäminen tietovarastojen kehityksessä

Ohjaaja: Hilkka Niemelä

Vuosi: 2020

Sivumäärä: 64

---

Tämän opinnäytetyön aiheena oli tutkia Biml-merkintäkielen käyttöä sekä rakentaa framework, jota voidaan käyttää työkaluna tietovarastoja rakennettaessa. Opinnäytetyö toteutettiin toimeksiantona Oiwa Solutions Oy:lle. Työn tarkoitus oli selvittää, soveltuuko Biml-merkintäkieli yrityksen toimintatapoihin tietovarastojen kehityksessä.

Työssä tehtiin Biml-kielellä framework, jota voi käyttää työkaluna tietovarastojen ja ETL-prosessien kehittämisessä. Työssä hyödynnettiin Microsoftin BI-työkaluja kuten Microsoft SQL Serveriä ja SQL Server Integration Serviceä. Työn lopputuloksena syntyi framework, jonka avulla saadaan automatisoitua suuri osa tietovaraston kehitysprosessista sekä monistettavissa oleva tietovarasto kokonaisuus LogForce-lähdejärjestelmälle.

Avainsanat: Tietovarastointi, Business Intelligence, BIML, SSIS, ETL

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## **Thesis abstract**

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Network Technology

Author: Santeri Niemi

Title of thesis: Utilization of Business Intelligence Markup Language in Data Warehouse development

Supervisor: Hilikka Niemelä

Year:2020

Number of pages: 64

---

The subject of the thesis was to study the use of Business Intelligence Markup language and to build a framework that can be used as a tool when building data warehouses. The thesis was conducted as an assignment from Oiwa Solutions Oy. The purpose of the work was to find out if Biml is suitable for the company's operating methods in the development of data warehouses.

The practical part of the thesis utilized Microsoft BI tools such as Microsoft SQL Server, Integration Service and Biml. The result of the practical part was a framework that can be used to automate parts of the data warehouse development process such as the creation of ETL-packages.

Keywords: Data Warehouse, Business Intelligence, BIML, SSIS, ETL

## SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	4
Kuvaluettelo .....	5
Käytetyt termit ja lyhenteet .....	7
1 JOHDANTO .....	8
1.1 Työn taustaa .....	8
1.2 Työn tavoite .....	9
1.3 Työn rakenne .....	9
1.4 Yritysesittely .....	9
2 TIETOVARASTO .....	11
3 BUSINESS INTELLIGENCE .....	14
4 BI- TYÖKALUT .....	15
4.1 Microsoft SQL Server.....	15
4.2 SQL-kyselykieli.....	15
4.3 SQL Server Integration Services.....	16
4.4 Power BI .....	18
5 BUSINESS INTELLIGENCE MARKUP LANGUAGE .....	20
5.1 Kielen rakenne .....	20
5.2 XML.....	22
5.3 BimlScript.....	23
5.4 Käyttökohteet .....	23
5.5 Esimerkkejä Biml-kielen käytöstä.....	23
5.5.1 Lastausalueen luominen .....	24
5.5.2 Datan tuominen lastausalueelle .....	31
5.5.3 Metatietojen määrittely ja lukeminen.....	35
6 CASE LOGFORCE - salattu.....	44
7 YHTEENVETO JA POHDINTA .....	45
LÄHTEET .....	46

## Kuvaluettelo

Kuva 1. Tietojen jalostusprosessi.....	11
Kuva 2. ETL-paketti Control flow -tasolla. ....	17
Kuva 3. ETL-paketti Data flow -tasolla. ....	18
Kuva 4. Power BI -raporttien suunnittelunäkymä .....	19
Kuva 5. Biml-kielen rakenne .....	21
Kuva 6. Biml-kääntäjän prosessi.....	21
Kuva 7. Esimerkki paketin sisältämästä XML-dokumentista. ....	22
Kuva 8. Lastausalueen luonti T-SQL skriptillä .....	25
Kuva 9. Luo lastausalue.biml. ....	25
Kuva 10. Yhteydet ja paketin nimi.....	26
Kuva 11. Yhteyden luominen AdventureWorks-kantaan BimlScriptillä .....	26
Kuva 12. BimlScriptin käyttö taskien luomisessa .....	27
Kuva 13. Esikatseluikkunasta voidaan havaita GetDropAndCreateDdl-metodin luoma SQL-lause. ....	28
Kuva 14. Esikatseluikkunan käyttö.....	28
Kuva 15. Visual Studion projektikansio ja paketin kääntäminen. ....	29
Kuva 16. Biml-tiedostosta käännetty SSIS-paketti.....	30
Kuva 17. Lastausalue paketin suorittamisen jälkeen .....	30
Kuva 18. Taulujen lataus .....	31
Kuva 19. Taulujen lataukset suoritetaan Dataflow-tasolla.....	32

Kuva 20. BimlScriptin käyttö Dataflow-komponenttien kanssa .....	32
Kuva 21. Käännetty paketti Control Flow -tasolta .....	33
Kuva 22. Lataa Department -taski Data Flow -tasolla .....	34
Kuva 23. Sarakkeiden kartoitukset Department-taululle. ....	34
Kuva 24. Data Flow -näkyvä suorituksen aikana. ....	35
Kuva 25. Department-taulu paketin suorittamisen jälkeen. ....	35
Kuva 26. DimAddress-näkymän luonti .....	36
Kuva 27. Metatiedot Excel-taulukossa .....	36
Kuva 28. Metadata.biml .....	37
Kuva 29. Onnistunut tietojen haku Excel-taulukosta .....	38
Kuva 30. Ympäristö.biml .....	39
Kuva 31. DimAddress-taulu esikatseluikkunassa.....	39
Kuva 32. SQL-lauseen luonti ja esikatseluikkuna .....	40
Kuva 33. Paketti.biml .....	41
Kuva 34. Paketin kääntäminen .....	41
Kuva 35. Lataa dimAddress-paketin control ja-dataflow .....	42
Kuva 36. Sarakkeiden kartoitukset.....	42
Kuva 37. Temp.dimAddress-taulu paketin ajamisen jälkeen.....	43

## Käytetyt termit ja lyhenteet

<b>BI</b>	Liiketoimintatiedon hallinta (Business Intelligence). Systemaattista yrityksen suorittamaa liike-elämän tietojen hankintaa, tallennusta ja analysointia.
<b>BIML</b>	BI-tarkoituksiin kehitetty XML-pohjainen merkintäkieli.
<b>ETL</b>	Prosessi, jossa tietoa haetaan ja muokataan (Extract – Transform – Load)
<b>SCD</b>	Tietovaraston taulutyyppejä (Slowly changing dimension).
<b>SQL</b>	Kyselykieli (Structured Query Language).
<b>SSIS</b>	Ohjelma, jolla ETL-ratkaisuja toteutetaan. (SQL Server Integration Services)

# 1 JOHDANTO

## 1.1 Työn taustaa

Tiedolla on nykyään entistä suurempi merkitys yritysmaailmassa. Tietoa yrityksille tuottavat erilaiset toiminnanohjausjärjestelmät, kuten esimerkiksi yrityksen varastonohjausjärjestelmä ja yrityksen verkkokauppa. Usein tätä tietoa ei kuitenkaan hyödynnetä täysin tai tieto on niin monimutkaista, ettei yritys pysty sitä itse analysoimaan.

Oiwa Solutions on tiedolla johtamiseen keskittyvä yritys. Tiedolla johtamisen tehtäviin kuuluu tietovarastointi, ylläpito ja raportointi erinäisillä Business Intelligence (BI) -työvälineillä.

Tietovarastoihin kerätään tietoa eri järjestelmistä yhtenäiseen muotoon, josta on helppo muodostaa haluttuja raportteja. Tietovarastointi parantaa myös datan eheyttä, jolloin mahdolliset virheet tiedossa on helpompi paikantaa.

Tietovarastointiprojektin yksi tärkein ja eniten aikaa syövä vaihe on ETL-prosessin kehittäminen. ETL-prosessissa tieto haetaan ohjelmallisesti järjestelmistä, sitä muokataan ja sitten se siirretään tietovarastoon.

Nykyisillä työkaluilla ETL-prosessit tehdään yleensä käsin ja ~~se~~ siihen kuluu suurin osa tietovaraston kehitykseen käytetystä ajasta. Prosessia on kuitenkin mahdollista nopeuttaa tai automatisoida lähes täysin. Biml-merkintäkieli on teknologia, joka mahdollistaa tämän. Biml on merkintäkieli, jolla kirjoitetaan ohjeet ETL-prosessille. Sen tavoitteena on helpottaa metatiedon käyttöä ja hyödyntämistä ETL-prosessin tekemisessä ja tekee prosessista myös paljon aiempaa modulaarisemman.



## 1.2 Työn tavoite

Tämän opinnäytetyön tavoitteena on tutkia, mitä Business Intelligence ja tietovarastointi on, sekä miten ETL-prosessin kehittämistä voidaan nopeuttaa ja osittain automatisoida Biml-kielen avulla. Työn tavoitteena on erityisesti tutkia, miten Biml-kieli soveltuu tietovarastokokonaisuuksien rakentamiseen.

## 1.3 Työn rakenne

Luvussa 2 kerrotaan tietovarastoista yleisellä tasolla, sekä selitetään ETL-prosessin kulkua sekä tietovaraston rakennetta. Luvussa 3 kerrotaan Business Intelligencen käsitteestä ja käyttökohteista. Luvussa 4 käydään läpi projektissa käytettyjä BI-työkaluja ja kerrotaan miten ne toimivat.

Luvussa 5 kerrotaan Biml-kielestä. Kielestä käydään läpi sen historia ja käyttö sekä työkalut. Lisäksi esitellään sen käyttöä toteuttamalla datan tuominen lastausalueelle operatiivisesta järjestelmästä, sekä luodaan ETL-paketti metadatan avulla.

Luvussa 6 kerrotaan, miten Biml-kieltä voidaan käyttää tietovarastojen rakentamisessa ja käydään läpi, miten projektin tietovarasto on kehitetty. Tämä luku on jouduttu jättämään pois liikesalaisuuksien vaarantumisen vuoksi opinnäytetyön julkisesta versiosta.

## 1.4 Yritysesittely

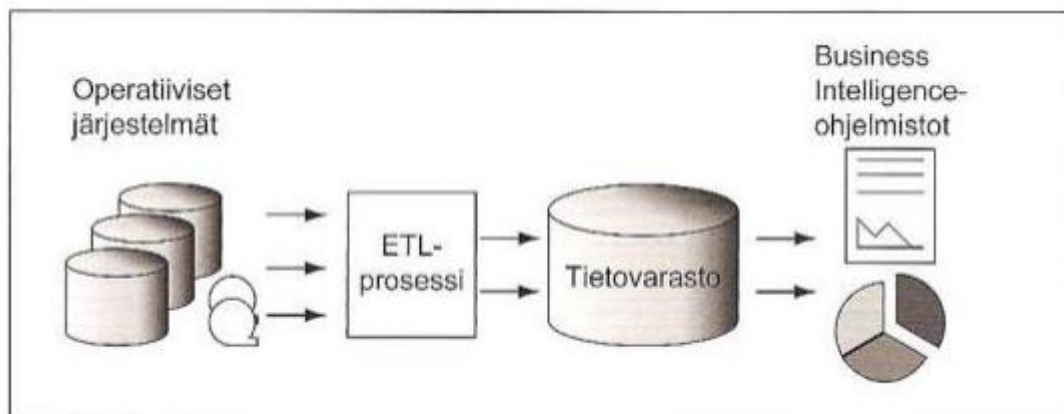
Oiwa Solutions on Seinäjokinen 2017 perustettu tiedolla johtamiseen keskittyvä yritys. Oiwa Solutions tarjoaa Microsoft tuotteilla rakennettuja ja ylläpidettäviä tietovarasto- ja BI-ratkaisuja sekä Net-sovelluskehitystä. Ratkaisut rakennetaan pilvipalveluna Azure-pilvipalveluun tai asiakkaan omaan Windows-palvelinympäristöön. (Oiwa Solutions [Viitattu 13.4.2020].)

Oiwa Solutionsin tarjoamat Proof of Concept -projektit ovat asiakkaalle kustannustehokas ja nopea tapa kokeilla tietovarastokonseptia. POC-projekteissa rajattu osa-

alue tehdään tuotantokäyttöön asti toimivaksi kokonaisuudeksi, jota voi myöhemmin laajentaa pala palalta. (Oiwa Solutions [Viitattu 14.4.2020].)

## 2 TIETOVARASTO

Tietovarastoon poimitaan ja integroidaan eri lähdejärjestelmistä tietoja yhteen. Tietovarasto on oma erillinen tietokantansa, joka tarjoaa monipuolisen ja entistä helpomman tavan tietojen raportointiin. Tietovaraston lataus tapahtuu yleensä joka päivä automaattisesti. Näin tiedot saadaan käyttöön jalostettuina ja riippuvuus tietojärjestelmätoimittajista vähenee. Keskitetyn tietovaraston tietoja voi hyödyntää sekä organisaation sisällä tai sen ulkopuolella. (Hovi 21.11.2012.)



Kuva 1. Tietojen jalostusprosessi (Hovi, Hervonen & Koistinen 2009, 86).

Kuvassa 1 kuvataan jalostusketjua lähdejärjestelmien tietokannoista tietovarastoon. Tätä prosessia kutsutaan tietovarastoinniksi.

Alla on kuvattu kuvaan liittyviä vaiheita:

- ETL tulee sanoista Extract – Transform – Load eli poiminta, muokkaus ja lataus. ETL-prosessissa tiedot luetaan operatiivisista järjestelmistä tai siirtotiedostoista, muokataan tietovarastotietokannan muotoon ja lopuksi kirjoitetaan tietovarastoon. Muokkauksessa on monia osatehtäviä, kuten virheiden tarkistus, koodien muunnokset ja tietojen historiointi. (Hovi ym. 2009, 48.)
- Tietovaraston tietoja kysellään, analysoidaan ja raportoidaan erilaisilla BI-työkaluilla. Tietovarastosta ajetaan kyselyitä ja parametroitavia raportteja. (Hovi ym. 2009, 14.)

ETL-prosessien toteutus on luonteeltaan tietokantapohjaista eräajosovelluksen ohjelmointia. Työssä käytetään yleensä erityisiä ETL-työkaluja, kuten Microsoftin SQL Server Integration Services, Azure Datafactory sekä Oracle Data Integrator. Nämä työkalut sisältävät paljon valmista toiminnallisuutta työtä helpottamaan ja nopeuttamaan. (Hovi ym. 2009, 48.)

Tietovarastoinnilla saavutetut keskeisimmät edut ovat:

- Eri lähteissä olevat tiedot yhdistetään eli integroidaan keskitettyyn ja yhteiskäyttöiseen tietokantaan.
- Tietovarasto on riippumaton liiketoiminnan prosesseista, joita ei tarvitse mitenkään muuttaa tietovarastoa toteuttaessa. Näin raportoinnista tulee joustavampaa ja tehokkaampaa verrattuna operatiivisten järjestelmien suoraan raportointiin.
- Johdettuja tietoja ja tunnuslukuja voidaan laskea valmiiksi, esimerkiksi asiakaskatteita, keskisaldoja tai kannattavuutta. Yhteisesti sovitulla kaavoilla lasketut tiedot muodostavat yhteiskäyttöisen ja ainoan oikean version, puhutaan yhden totuuden periaatteesta. Toisin sanoen kaikilla on samat tunnusluvut, ei käytetä erilaisia lukuja eri raporteissa.
- Tietovarastossa säilytettävät historiatiedot mahdollistavat aikasarja-analyytit.
- Tietovaraston tietoja voidaan hyödyntää monella tavoin, kuten kyselyin ja raportein.
- Tietovarastosta voidaan laatia tilastoja, viranomaisraportteja sekä poimintoja talon ulkopuolelle, esim. yhteistyökumppaneille.
- Tietojen laatua voidaan helpommin valvoa, kun tiedot ovat keskitetysti saatavilla.
- Operatiivisista järjestelmistä voidaan turvallisesti poistaa vanhaa tietoa, kun ne on ladattu tietovarastoon ja niitä ei enää tarvita operatiivisen järjestelmän puolella. Näin saadaan parannettua operatiivisten järjestelmien suorituskykyä.
- ETL-prosessit voidaan automatisoida, jolloin ne käynnistyvät ajastetusti esim. joka päivä tai joka tunti. Virheiden määrä vähenee käsityönä tehdyn työn määrän vähentyessä. (Hovi ym. 2009, 15–16.)

Tietovarastolla saadaan helpotettua tietojen oikeellisuuden tarkistamista. Operatiivisissa järjestelmissä tietoja tarkastellaan vain ennalta määritellyistä ja ohjelmoituista näkökulmista, jolloin virheellisiä tietoja ei usein huomata. Tietovarastosta tietoja katsellaan jatkuvasti vaihtuvista näkökulmista, jolloin luokitteluihin ja ryhmitelyihin liittyvät virheet ja epävarmuudet paljastuvat ja ne voidaan korjata. Usein korjaaminen hoituu ETL-puolella, mutta joskus virheet täytyy korjata suoraan operatiiviseen järjestelmään. (Hovi ym. 2009, 15.)

Tietovarastoja voidaan rakentaa eri filosofioilla. Kaksi yleisintä ja vanhinta filosofiaa ovat Bill Inmonin malli ja Ralph Kimballin dimensiomalli, jossa on kahden tyyppisiä tauluja, faktatauluja ja dimensiotauluja. (Naeem 16.12.2019.)

Inmonin mallissa tietovaraston suunnittelu alkaa yrityksen tietomallin muotoilulla, siinä tunnistetaan yrityksen pääasialliset aihealueet sekä kokonaisuudet. Näitä kokonaisuuksia ovat esimerkiksi asiakas, tuote tai myyjä. Kokonaisuuksista rakennetaan looginen malli, jossa kartoitetaan kokonaisuuksien kaikki attribuutit. Kokonaisuuksien tiedot tallennetaan noudattaen kolmatta normaalimuotoa. Normalisointi helpottaa tietojen lataamista, mutta kyselyjen teko vaikeutuu, kun suuri määrä tauluja pitää liittää toisiinsa. (Naeem. 16.12.2019.)

Kimballin dimensiomallin suurin ero Inmonin malliin on se että, tietoa ei normalisoida. Dimensiomallinnuksen keskeisin konsepti on tähtiskeema. Tähtiskeemassa faktatauluja ympäröi tyypillisesti monta dimensiotaulua. Dimensiotaulut koostuvat yleensä tekstimuotoisesta tiedosta, jolla voi vastata kysymyksiin faktataulujen tiedoista, kuten kuka, missä ja mikä. Faktataulut koostuvat dimensiotaulujen vierasavaimista, sekä kaikista aihealueen mittareista, kuten tilausten summista ja myyntitapahtuman kellonajoista. Dimensiomallisen tietovaraston integrointi tapahtuu mukautetuilla dimensioilla, jotka voidaan yhdistää moneen eri faktatauluun. (Rangarajan 1.11.2016.)

Tämän opinäytetyön työ toteutetaan dimensiomallia noudattaen.

### 3 BUSINESS INTELLIGENCE

Business Intelligence on kattava termi, johon sisältyy sovellukset, infrastruktuuri, työkalut ja parhaat käytännöt, jotka mahdollistavat tiedon saannin ja analysoinnin ratkaisujen tehostamiseen, parantamiseen ja optimointiin (Rad 2014, 7 - 8).

Business Intelligence -ratkaisujen avulla yritysten ja julkisten organisaatioiden henkilöstö saa haltuunsa (liike)toimintaa kuvaavaan informaation. Tämä informaatio auttaa heitä tekemään valistuneempia päätöksiä ja ohjaamaan toimintaa oikeaan suuntaan. Organisaatioissa on entistä vähemmän aikaa tehdä liiketoimintaan liittyviä päätöksiä. Entisen vuositason suunnittelun sijaan raportointi täytyy toteuttaa kuukausi-, viikko- tai päivätasolla. BI-ratkaisujen kannalta tämä tarkoittaa tietovarastojen nopeita lataussyklejä ja raporttien automatisoituja latauksia. (Hovi ym. 2009, 74, 76.)

Määritelmistä voidaan huomata, että BI-järjestelmän tuottaman data-analytiikan on tarkoitus auttaa päätöksentekijöitä tekemään oikeat päätökset. BI-ratkaisut voivat parhaimmillaan tuoda johdon ja päätöksentekijöiden ulottuville tärkeitä tietoja operatiivisten järjestelmien syövereistä, joiden olemassaolosta eivät ole tiedneet kuin järjestelmän arkkitehdit. (Hovi ym. 2009, 74.)

Moni yritys kerää tietoa, mutta ei osaa, pysty, halua tai ehdi hyödyntää sitä kaikkea. Tiedon jatkojalostaminen, analysoiminen ja hyödyntäminen ovatkin haasteita, johon Business Intelligence vastaa. (Hovi ym. 2009, 74.)

## 4 BI- TYÖKALUT

BI-työkaluihin kuuluvat kaikki välineet, joiden avulla toteutetaan BI-ratkaisuja. Tässä opinnäytetyössä keskitytään Microsoftin tarjoamiin BI-työkaluihin.

### 4.1 Microsoft SQL Server

Tietovarastot muodostuvat tietokannoista. Tietokannat ovat tietokokoelma, jossa tieto on järjestetty nopeasti etsittäväksi. Tietokannat on rakennettu helpottamaan tietojen tallentamista, muokkaamista, hakemista ja poistamista erilaisten tietojenkäsittelytoimintojen yhteydessä. (Encyclopeadia Britannica [Viitattu 13.4.2020])

Microsoft SQL Server on relaatiotietokannan hallinta- ja analysointijärjestelmä. Relaatiotietokannat koostuvat tauluista, joihin tieto tallennetaan. Taulut koostuvat sarakkeista ja riveistä. Tauluja voidaan liittää toisiinsa avaimilla. Hallintajärjestelmä sisältää työkalut tiedon rakenteen hallintaan, käsittelyyn ja muuttamiseen. (Schlichting 2008.)

Taulujen rivien järjestyksellä ei ole merkitystä relaatiotietokannoissa. Tiedon hakuun ja hallintaan käytetään SQL-kyselykieltä, jolla tiedot saa esiteltä oikeassa järjestyksessä. Microsoft SQL Server -palvelua hallitaan Microsoft SQL Management Studio -sovelluksella. (Schlichting 2008.)

### 4.2 SQL-kyselykieli

SQL-kieli tarjoaa myös mahdollisuuden käsitellä taulun tietoja. Tietoja noudetaan tietokannoista kyselykielten avulla. Structured Query Language (SQL) on IBM:n 1970-luvulla kehittämä standardoitu kyselykieli. SQL-kieli mahdollistaa kyselyt SQL-tietokantoihin. Kyselyt toteutetaan tiedusteluina (Query). Tietoja voi käsitellä lisäämällä, muuttamalla ja poistamalla. Tärkeimpiä operaatioita ovat Select, Insert, Delete ja Update. (W3schools [Viitattu 11.4.2020].)

Select-kyselyillä voidaan hakea tietoa tietokannasta. Kyselylle voi määrittää halutut sarakkeet tai käyttää \*-merkkiä, jos halutaan hakea kaikki taulun sarakkeet. Haettu tieto tallentuu tulostaulukkaan. (W3schools [Viitattu 12.4.2020].)

Insert-kyselyillä tietoa voidaan tallentaa tietokantaan, ja ne ovat muotoa INSERT INTO taulun nimi (sarake1, sarake2, sarake3, ...) VALUES (arvo1, arvo2, arvo3, ...) (W3schools [Viitattu 13.4.2020]).

Update-kyselyillä tietokannoissa olevaa tietoa voi muuttaa. Ne ovat muotoa UPDATE taulun nimi SET sarake1 = arvo 1, sarake2 = arvo2, ... WHERE-ehto. Where-lausekkeella määritellään ehdot, joiden perusteella rivit päivitetään. (W3schools [Viitattu 14.4.2020].) Delete-kyselyillä tauluista voidaan poistaa tietoja, ja ne ovat muotoa DELETE FROM taulun nimi WHERE-ehto (W3schools [Viitattu 15.4.2020]).

Näkymät ovat virtuaalisia tauluja, joihin on tallennettu SQL-kyselyn tulostaulukko. Näkymät sisältävät rivejä ja sarakkeita kuten tavallisetkin taulut. Näkymien kentät koostuvat yhden tai useamman taulun kentistä. (W3schools [Viitattu 16.4.2020].)

### 4.3 SQL Server Integration Services

SQL Server Integration Services (SSIS) on Microsoftin kehittämä alusta yritystason tietointegraatoratkaisujen ja tiedonmuunnosratkaisujen valmistamiseksi. SSIS-alustan avulla voidaan lukea ja muuntaa tietoja monista eri lähteistä. Näihin lähteisiin kuuluu mm. XML-tiedostot, CSV-tiedostot, Excel-tiedostot ja yleisimpänä reaali-tietokannat. (Microsoft 2018a.)

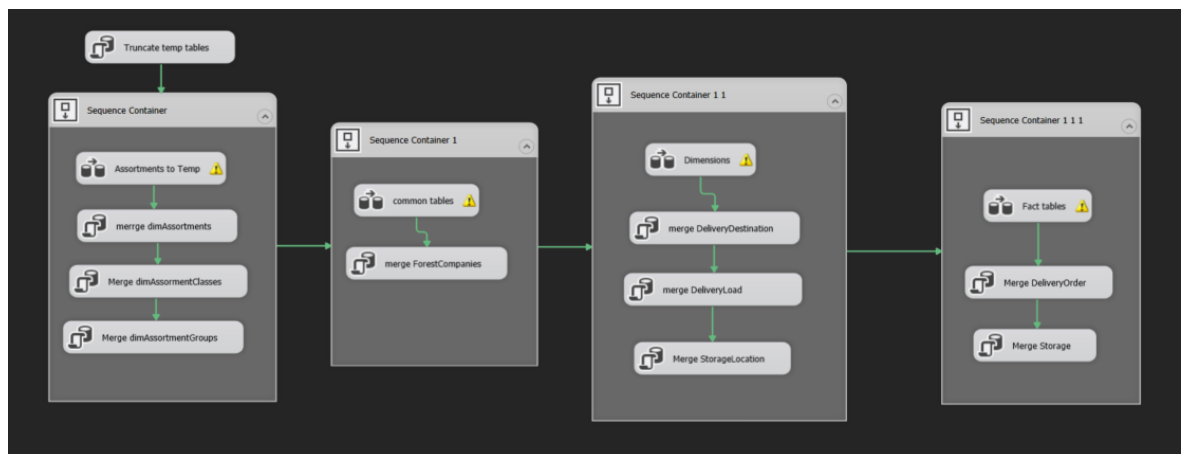
SSIS-alusta on periaatteessa graafinen raahaa ja pudota (Drag and drop)-tyyppinen käyttöliittymä ETL-pakettien tekemiselle. Paketit koostuvat erinäisistä komponenteista, johon kuuluu erinäisiä taskeja, datamuunnoksia ja lähteitä ja päämääriä. (Microsoft 2016.)

Kuvassa 2 näkyy tämän työn prototyypivaiheen ETL-paketti Control flow -tasolta. Control flow -taso koostuu kolmesta päätoiminnallisuudesta:

- Containerit eli astiat antavat paketeille rakenteellisuutta ja mahdollistavat silmukoiden toteuttamisen paketin muuttujia hyväksikäyttäen.

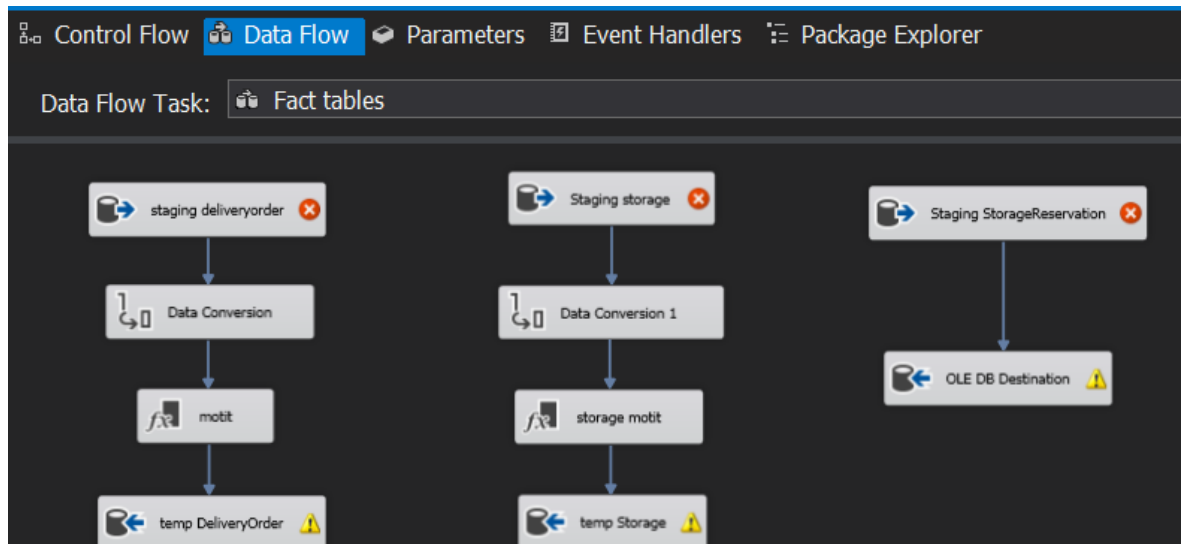


- Taskit eli suoritettavat komponentit antavat paketeille niiden toiminnallisuuden. Esimerkiksi SQL-taskilla on mahdollista ajaa SQL-kyselyitä yhdistetystä tietokannasta.
- Precedence Constraints eli edeltävät rajoitukset yhdistävät taskit ja containerit toisiinsa ja määräämällä niiden suoritusjärjestyksen. Kuvan 2 vihreät nuolet ovat edeltävyysrajoituksia. (Microsoft 2017a.)



Kuva 2. ETL-paketti Control flow -tasolla.

Varsinainen tiedonsiirto tapahtuu pakettien Data flow -taskilla. Kuvassa 3 näkyy esimerkki data flow -tasosta. Data flow -taso koostuu lähteistä, päämääristä ja muunnoksista. Tietoa voidaan hakea esimerkiksi tietokannoista, Excel-taulukoista ja XML-tiedostoista. Muunnoksilla voidaan muokata dataa, kuten suorittaa tyyppi-muunnoksia. Muunnoksilla voi esimerkiksi vaihtaa taulun päivämääräsarakeen tiedot numeromuotoon tai tekstimuotoisen rahamäärän desimaalimuotoon. (Microsoft 2017c.) Datavirran voi myös jakaa jonkin muuttujan perusteella eri päämääriin. Tällä tasolla nuolet määräävät datavirran suunnan. (Microsoft 2017b.)



Kuva 3. ETL-paketti Data flow -tasolla.

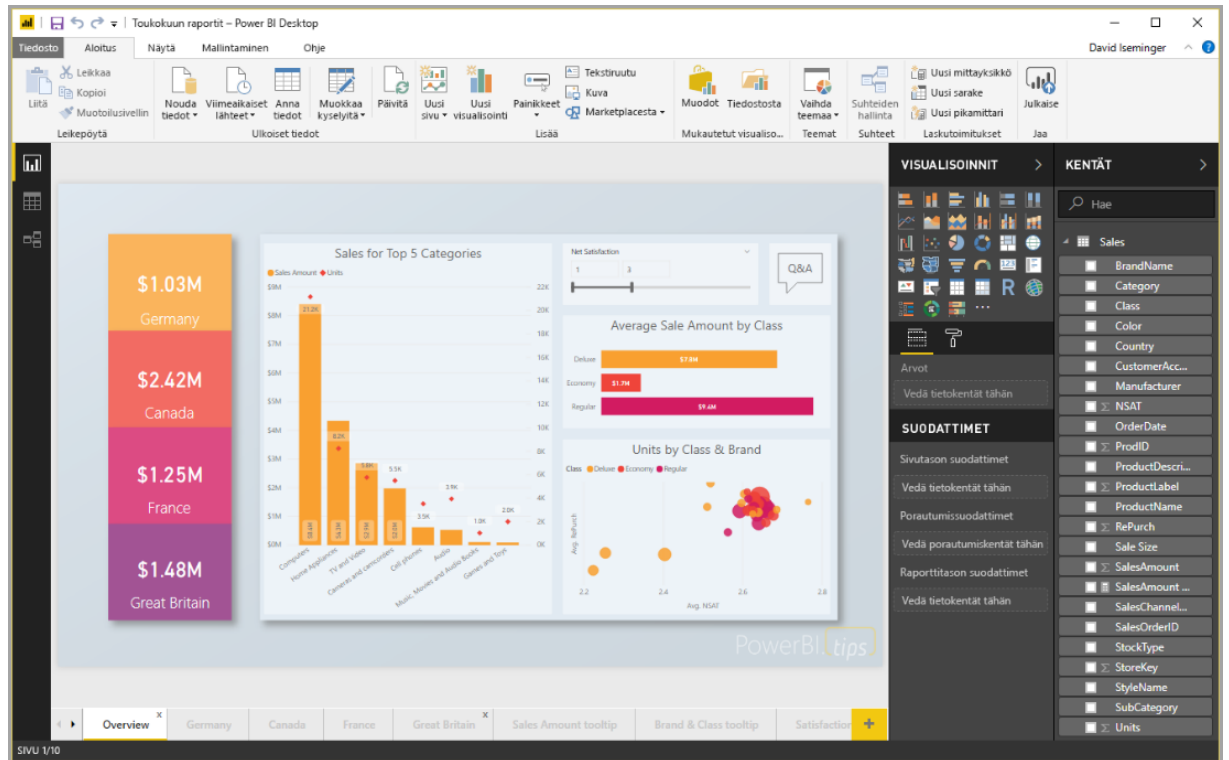
Paketit ovat dtsx-päätteisiä tiedostoja, jotka sisältävät ohjeet datavirralle. Pohjimmiltaan paketit ovat XML-tiedostoja. Näistä kerrotaan tarkemmin osiossa 5.2.

#### 4.4 Power BI

Power BI on sarja liiketoiminnan analysointityökaluja. Power BI tarjoaa työkalun suunnitella ja toteuttaa interaktiivisia raportteja. Power BI tarjoaa kattavan valikoidun eri lähteitä, joista tiedot saa näkyymiin, kuten Excel-taulukoista, paikallisista tietokannoista tai useista eri pilvipalveluista. Power BI tukee myös reaaliaikaisen tiedon esittämistä. Power BI tarjoaa pilvipalvelun, jonne näkymiä ja raportteja voi tallentaa. Näkymiä ja raportteja voi myös upottaa nettisivuille. Näkymiä voi selata mobiilisovelluksella tai web-selaimella. (Power BI [Viitattu 23.4.2020].)

Power BI-palvelun Q&A-palvelu mahdollistaa kysymysten esittämisen tiedolle. Palvelu toimii vain englannin kielellä. Palvelulta voi kysyä esimerkiksi ”How many stores are there?” ja palvelu vastaa kysymykseen kaaviolla tai KPI-mittarilla (Key Performance Indicator). Palvelu osaa vastata, jos tietolähteestä löytyy tiedot vastauksen antamiseen. (Microsoft 2019b.)

Kuvassa 4 on Power BI Desktop, joka on Power BI -raporttien tekemiseen tarkoitettu työkalu. Power BI Desktop tarjoaa visuaalisen Drag and Drop -käyttöliittymän raporttien suunnittelulle. Tieto voi esitellä monilla erityyppisillä graafeilla ja taulukoilla.



Kuva 4. Power BI -raporttien suunnittelunäkymä. (Microsoft 2019b.)

## 5 BUSINESS INTELLIGENCE MARKUP LANGUAGE

Business Intelligence Markup Language (Biml) on Varigencen kehittämä XML-dialekti BI-tarkoituksiin (Varigence [Viitattu 11.4.2020]). Vaikka Biml-merkintäkielen pelkkä deklaratiiivinen muotoilu riittää BI-resurssien kirjoittamiseen, BimlScript mahdollistaa C# tai VB.NET-koodin upottamisen Biml-kieleen. Näin saadaan helpotettua ja vähennettyä kirjoittamista, kun saadaan esimerkiksi for- ja foreach-silmukat käyttöön. (Leonard ym. 2017, 25.)

Työkalut, joilla Biml-ratkaisuja tehdään, kääntävät Biml metatiedot SSIS tai SSAS vastaaviksi, joita SQL Server ymmärtää. Biml on T4-tyyppinen kieli (T4 on tekstimallin muuttamisen kirjasto), joka perustuu tekstin luomiseen mallipohjasta. (Leonard ym. 2017, 26.)

### 5.1 Kielen rakenne

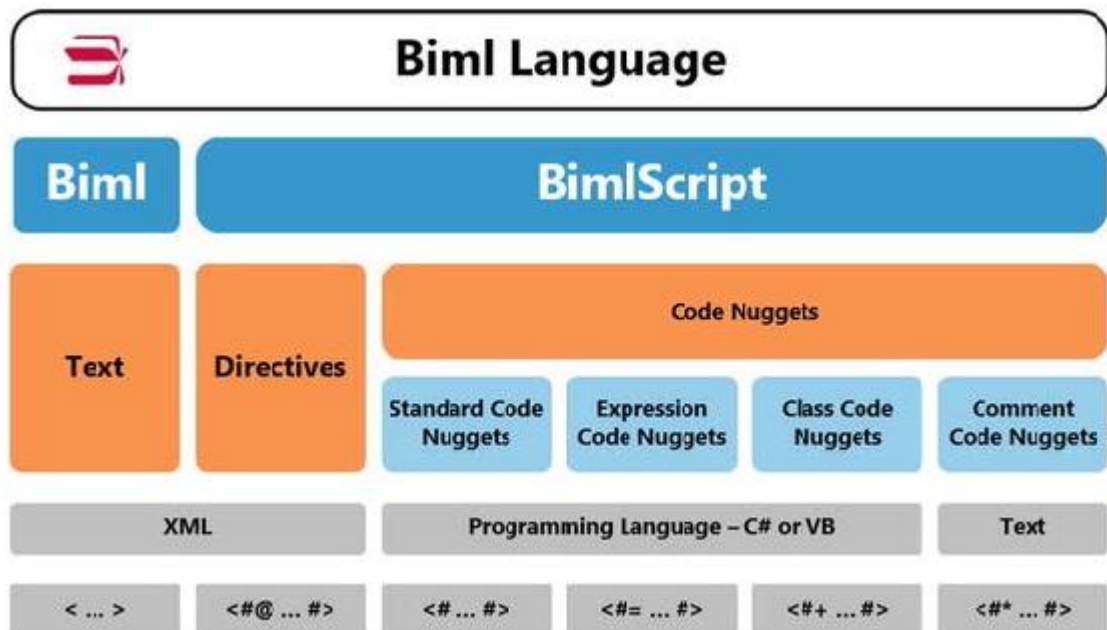
Biml-kielen rakenne voidaan jakaa kahteen pääosaan: Biml-merkintäkieleen ja BimlScript-ohjelmointikieleen. Biml referoi Biml-dokumenttien XML-osioita, joilla määritellään ETL-pakettien rakenne. (Leonard ym. 2017, 25-26.)

Kuvassa 5 havainnollistetaan Biml-merkintäkielen ja BimlScript-ohjelmointikielen rakenteellinen jakautuminen. On syytä huomata, että Biml-merkintäkieli on pieni osa koko kielestä. Alimmista laatikoista näkyy, millä merkinnöillä jokainen osa esiintyy Biml-kielessä.

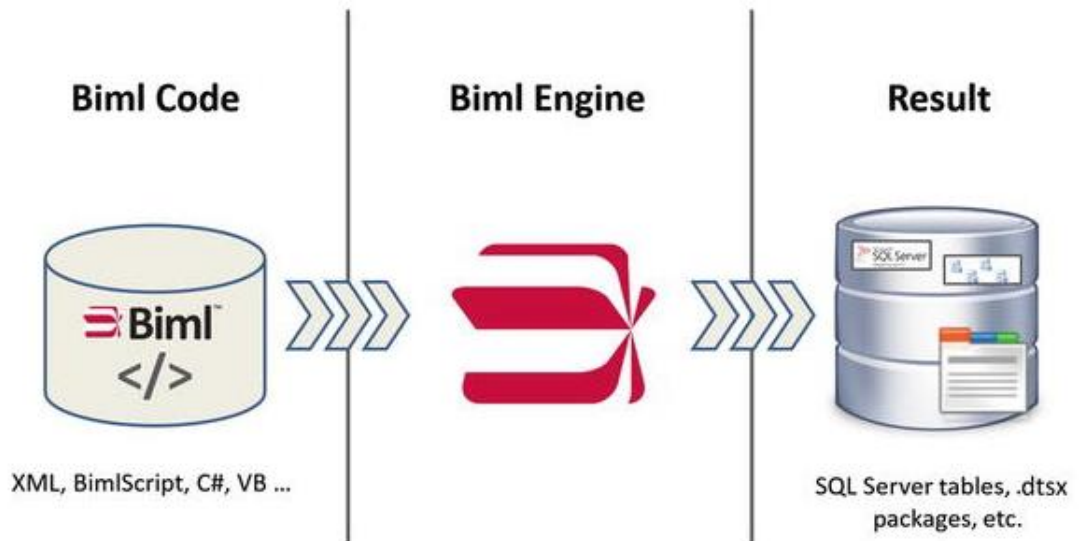
Tärkeimmät näistä merkinnöistä ovat:

- Biml-merkintäkielen ja XML:n käyttämä `< [Biml] >`
- Control koodi -nugettien käyttämä `<# [C#- tai Visual Basic -koodi] #>`. Control nugetit mahdollistavat C#- tai VB-koodin upottamisen Biml-merkintäkieleen ja ne auttavat Biml-kääntäjää generoimaan tekstiä. Tämä mahdollistaa Biml-objektien ohjelmallisen luomisen.
- Teksti koodi -nugettien käyttämä `<#=[C#- tai Visual Basic koodi] #>`. Teksti-nugetit ovat koodin osia, jotka arvioivat niiden sisällä olevan lausekkeen ja kääntävät sen tekstiksi Biml-merkintäkieleen. Teksti-nugeteille

ei voi antaa kuin koodin osia, jotka kääntyvät vain yhdeksi tekstiksi. (Leonard ym. 2017, 36-38.)



Kuva 5. Biml-kielen rakenne (Leonard ym. 2017, 25).



Kuva 6. Biml-kääntäjän prosessi (Leonard ym. 2017, xxiv.)

BimlScript luo sisältöä pelkästään suunnitteluympäristössä, eikä sitä päädy valmiiden pakettien puolelle. Kuvassa 6 havainnollistetaan käänösprosessi, Biml-tiedosto voi sisältää jokaista kuvassa 5 näkyvää syntaksia ja se käännetään normaaleiksi BI-objekteiksi kuten SSIS-pakettitiedostoiksi. (Leonard ym. 2017, xxix.)

## 5.2 XML

XML on lyhenne sanoista eXtensible Markup Language. XML on yksinkertainen laitteistosta tai alustasta riippumaton standardi, jolla tietoja voidaan kuvata jäseneltyssä muodossa. XML-kielen tavoite on olla helppolukuista koneille sekä ihmisille. (Richards 2004, 1.)

SSIS-paketit ovat dtsx-päätteisiä tiedostoja, joihin tallennetaan ohjeet tietovirran käsittelemiseksi lähtöpisteistä määränpäähänsä. Dtsx on Microsoftin kehittämä XML-pohjainen tiedostomuoto. (Microsoft 2019a.)

Biml on XML-dialekti, jonka avulla on mahdollisuus automatisoida ja helpottaa SSIS-pakettien ylläpitoa ja kirjoittamista. Yksinkertainenkin paketti, joka vain tyhjää taulun, voi olla jopa 400 riviä XML-tekstiä sisältävä dtsx-tiedosto, eikä sen käsin muokkaaminen ole kovin tuotteliasta. (Leonard ym. 2017, xxvii-xxviii.)

Kuvassa 7 on esimerkki, miltä SSIS-paketin sisältämä XML näyttää.

```

Luo laetusluodot (XML)
<?xml version="1.0" encoding="utf-8"?>
<DTS:Executable DTS:CreationDate="02/12/2020 15:01:02" DTS:CreationName="Microsoft.Package" DTS:CreatorComputerName="PC" DTS:CreatorName="santeri" DTS:DTSID="{5A22D623-1317-4D17-B732-49F...}"
  <DTS:Property DTS:Name="PackageFormatVersion">8</DTS:Property>
  <DTS:ConnectionManagers>
    <DTS:ConnectionManager DTS:CreationName="OLEDB" DTS:DTSID="{C21C03A2-41AD-4901-ADD8-B9882DF0F113}" DTS:ObjectName="Staging" DTS:refId="Package.ConnectionManagers[Staging]">
      <DTS:ObjectData>
        <DTS:ConnectionManager DTS:ConnectionString="data source=pc\SQL EXPRESS;initial catalog=Staging;provider=SQLNCLI11.1;integrated security=SSPI;auto translate=False" />
      </DTS:ObjectData>
    </DTS:ConnectionManager>
  </DTS:ConnectionManagers>
  <DTS:Executables>
    <DTS:Executable DTS:CreationName="Microsoft.ExecuteSQLTask" DTS:DTSID="{DA5EAE63-3816-4365-B7A9-B856449CBFDC}" DTS:ExecutableType="Microsoft.ExecuteSQLTask" DTS:LocaleID="-1" DTS:Obj...>
      <DTS:ObjectData>
        <SQLTask:SqlTaskData SQLTask:Connection="Staging" SQLTask:SqlStatementSource="&xA;" SET ANSI_NULLS ON&xA;SET QUOTED_IDENTIFIER ON&xA;GO&xA;&xA;----
      </DTS:ObjectData>
    </DTS:Executable>
    <DTS:Executable DTS:CreationName="Microsoft.ExecuteSQLTask" DTS:DTSID="{596DC7B0-8130-406B-8828-4175C7F18D85}" DTS:ExecutableType="Microsoft.ExecuteSQLTask" DTS:LocaleID="-1" DTS:Obj...>
      <DTS:ObjectData>
        <SQLTask:SqlTaskData SQLTask:Connection="Staging" SQLTask:SqlStatementSource="&xA;" SET ANSI_NULLS ON&xA;SET QUOTED_IDENTIFIER ON&xA;GO&xA;&xA;----
      </DTS:ObjectData>
    </DTS:Executable>
    <DTS:Executable DTS:CreationName="Microsoft.ExecuteSQLTask" DTS:DTSID="{61928EF8-51F5-48E2-8A51-FEC388732889}" DTS:ExecutableType="Microsoft.ExecuteSQLTask" DTS:LocaleID="-1" DTS:Obj...>
      <DTS:ObjectData>
        <SQLTask:SqlTaskData SQLTask:Connection="Staging" SQLTask:SqlStatementSource="&xA;" SET ANSI_NULLS ON&xA;SET QUOTED_IDENTIFIER ON&xA;GO&xA;&xA;----
      </DTS:ObjectData>
    </DTS:Executable>
    <DTS:Executable DTS:CreationName="Microsoft.ExecuteSQLTask" DTS:DTSID="{6AEAE9A-9EE5-4A03-A258-2EFE1665F5F5}" DTS:ExecutableType="Microsoft.ExecuteSQLTask" DTS:LocaleID="-1" DTS:Obj...>
      <DTS:ObjectData>
        <SQLTask:SqlTaskData SQLTask:Connection="Staging" SQLTask:SqlStatementSource="&xA;" SET ANSI_NULLS ON&xA;SET QUOTED_IDENTIFIER ON&xA;GO&xA;&xA;----
      </DTS:ObjectData>
    </DTS:Executable>
    <DTS:Executable DTS:CreationName="Microsoft.ExecuteSQLTask" DTS:DTSID="{E265D1E5-0A36-4437-BEEF-26897CDC6EDA}" DTS:ExecutableType="Microsoft.ExecuteSQLTask" DTS:LocaleID="-1" DTS:Obj...>
      <DTS:ObjectData>
        <SQLTask:SqlTaskData SQLTask:Connection="Staging" SQLTask:SqlStatementSource="&xA;" SET ANSI_NULLS ON&xA;SET QUOTED_IDENTIFIER ON&xA;GO&xA;&xA;----
      </DTS:ObjectData>
    </DTS:Executable>
    <DTS:Executable DTS:CreationName="Microsoft.ExecuteSQLTask" DTS:DTSID="{D95798B0-DE56-4F92-BEFD-2C608BA6CC2D}" DTS:ExecutableType="Microsoft.ExecuteSQLTask" DTS:LocaleID="-1" DTS:Obj...>
      <DTS:ObjectData>
        <SQLTask:SqlTaskData SQLTask:Connection="Staging" SQLTask:SqlStatementSource="&xA;" SET ANSI_NULLS ON&xA;SET QUOTED_IDENTIFIER ON&xA;GO&xA;&xA;----
      </DTS:ObjectData>
    </DTS:Executable>
  </DTS:Executables>

```

Kuva 7. Esimerkki paketin sisältämästä XML-dokumentista.

### 5.3 BimlScript

BimlScriptin avulla saadaan sisällytettyä Biml-tiedostoihin klassisten ohjelmointikielten elementtejä ja toiminnallisuuksia, kuten silmukat ja funktiot. Näitä kutsutaan yleensä koodi nugeteiksi ja direktiiveiksi. Näitä nugetteja voi ohjelmoida C#- ja Visual Basic -ohjelmointikielillä. Samassa projektissa on mahdollista käyttää kumpaa-kin ohjelmointikieltä, mutta ei samassa Biml-dokumentissa. Tässä opinnäytetyössä käytetään vain C#-kieltä. (Leonard ym. 2017, 36.)

### 5.4 Käyttökohteet

Biml soveltuu parhaiten sellaisten prosessien automatisointiin, joissa noudatetaan samaa kaavaa toistuvasti. Yksi tällainen prosessi on tietovaraston latausalueen luominen ja lataus. Käyttämällä SSIS-ohjelmistoa taulun kopioiminen tietokannasta toiseen on helppoa, eikä siihen kulu aikaa 5 minuuttia enempää. Tämä ei kuitenkaan skaalaudu hyvin, kun tauluja on useampia, varsinkin kun joissain operatiivisissa järjestelmissä voi olla jopa satoja tauluja. (Leonard ym. 2017, xxvii.)

Biml myös helpottaa täysin metatietoon pohjautuvien tietovarastojen rakentamista. Tietovaraston rakenteen metatiedot voivat olla tallennettuna vaikka erilliseen konfiguraatiotauluun tai Excel-tiedostoon. Rakenteellinen metatieto sisältää ohjeet tietovaraston rakentamiseen. Metatietoa on esimerkiksi taulujen nimet ja taulujen sisältämien sarakkeiden nimet ja niiden tietotyypit. (Leonard ym. 2017, 234-235.)

### 5.5 Esimerkkejä Biml-kielen käytöstä

Biml-kieltä voi kirjoittaa millä tahansa tekstieditorilla, mutta sen kääntäminen vaatii jonkun Varigencen työkaluista. (Varigence [Viitattu 10.4.2020]).

Varigence on kehittänyt kolme työkalua Biml-ratkaisujen tekemiseen:

1. BimlExpress on yrityksen ainoa ilmainen tuote. Sen saa käyttöönsä ilman lisenssimaksuja Visual Studion lisäosaksi. Se tarjoaa koodieditorin syn-

taksin korostamisella, mikä mahdollistaa Biml-kielen helpomman kirjoittamisen, sekä pakettigeneraattorin, jolla Biml käännetään SSIS-paketeiksi. (Varigence [Viitattu 14.4.2020])

2. BimlStudio tarjoaa graafisen käyttöliittymän Biml-ratkaisujen kehittämiseksi. BimlStudio toimii Drag and Drop -periaatteella kuten SSIS. (Varigence [Viitattu 12.4.2020])
3. BimlFlex on kokoelma malleja, metatietojen määritelmiä ja työkaluja, joiden avulla voidaan rakentaa kokonaisia tietovarastomalleja kirjoittamatta yhtään riviä koodia. BimlFlex-ohjelmalla on mahdollista mallintaa tietovarastoja joko Kimballin tai Inmonin mallinnusfilosofialla tai DataVault-tekniikalla (Varigence [Viitattu 13.4.2020]).

Tässä opinnäytetyössä tutustutaan vain BimlExpress-työkalun käyttöön johtuen muiden työkalujen suurista lisenssimaksuista (BimlStudio 249 \$/kuukausi, BimlFlex 995 \$/kuukausi).

### **5.5.1 Lastausalueen luominen**

Tässä esimerkissä käytetään AdventureWorks2017-tietokantaa. AdventureWorks on Microsoftin keksimä kuvitteellinen yritys ja AdventureWorks2017 on tämän kuvitteellisen yrityksen OLTP-tietokanta (Online Transaction Processing). (Microsoft 2018b.)



Aluksi luodaan Staging-tietokanta ja operatiivisen järjestelmän skeemat T-SQL skriptillä, kuvan 8 mukaisesti. Tätä esimerkkiä varten luodaan skeemat Humanresources, person ja temp.

```
USE [MASTER]
GO
CREATE DATABASE Staging
GO
USE [STAGING]
GO
CREATE SCHEMA HumanResources
GO
CREATE SCHEMA Person
GO
CREATE SCHEMA temp
```

Kuva 8. Lastausalueen luonti T-SQL skriptillä

Projektiin lisätään uusi Biml-tiedosto ja nimetään se "Luo lastausalue". Biml-dokumentit alkavat aina <Biml>-elementillä, joka näkyy kuvassa 9. Attribuutti xmlns määrittelee dokumentin nimiavaruuden. Attribuuteilla voidaan erottaa muuten saman nimiset elementit (tai attribuutit) eri nimiavaruuksiin, jolloin eri XML-dokumentteja yhdisteltäessä vältetään nimien yhteentörmäyksiltä (W3schools [Viitattu 17.4.2020]).

```
Luo lastausalue.biml  + X
1 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
2 </Biml>
```

Kuva 9. Luo lastausalue.biml.

Ensimmäiseksi lisätään Biml-dokumenttiin yhteydet ja paketin nimi kuvan 10 mukaisesti. Tässä paketissa ei tarvita kuin yksi yhteys, joka näkyy kuvassa 10 nimellä Staging. Lisäksi määritellään myös Packages-elementti, eli paketit, jotka Biml luo, kun se käännetään. Nimetään paketti "Luo lastausalue" luomalla Package-elementti ja antamalla sille attribuutti Name, jonka arvoksi "Luo lastausalue" annetaan.

```
Luo lastausalue.biml* - X
1 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
2   <Connections>
3     <Connection Name="Staging" ConnectionString="data source=pc\SQLSERVER;initial catalog=Staging;" />
4   </Connections>
5   <Packages>
6     <Package Name="Luo lastausalue">
7   </Package>
8   </Packages>
9 </Biml>
```

Kuva 10. Yhteydet ja paketin nimi.

Seuraavaksi käytetään BimlScriptiä hakemaan metatiedot AdventureWorks-kannasta kuvan 11 mukaisesti. SchemaManager.CreateConnectionNode-metodi ottaa parametreikseen yhteyden nimen ja yhteyden tiedot, kuten palvelimen nimen, tietokannan nimen ja kirjautumistiedot.

ImportDB-metodin kaksi ensimmäistä parametriä ovat IncludeSchemas ja IncludeTables. IncludeSchemasin avulla määritellään skeemat, joiden metatiedot haetaan. IncludeTablesin avulla määritellään kopioitavat taulut, tässä tapauksessa parametrien arvo on pelkkä "" eli tyhjä merkkijono, joka tarkoittaa sitä, että kopioidaan kaikki taulut annetun skeeman alta.

Kolmas parametri on ImportOptions jolla voidaan määrittää, mitä lähtötietokannasta ei kopioida, kuten näkymiä, indeksejä tai vierasavaimia.

SourceMetaData-muuttuja sisältää nyt AdventureWorks-kannan HumanResources-skeeman alta löytyvien taulujen metatiedot.

```
Luo lastausalue.biml* - X
1 var sourceConnection = SchemaManager.CreateConnectionNode("AW",@"data source=pc\SQLSERVER;
2   initial catalog=AdventureWorks;
3   provider=SQLNCLI11.1;
4   integrated security=SSPI;
5   auto translate=False");
6 var sourceMetaData = sourceConnection.ImportDB("HumanResources","",ImportOptions.ExcludeViews | ImportOptions.ExcludeIndex
7 | ImportOptions.ExcludeColumnDefault | ImportOptions.ExcludeForeignKey | ImportOptions.ExcludeCheckConstraint
8 | ImportOptions.ExcludePrimaryKey | ImportOptions.ExcludeUniqueKey);
9
10 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
11   <Connections>
12     <Connection Name="Staging" ConnectionString="data source=pc\SQLSERVER;initial catalog=Staging;provider=SQLNCLI11.1;" />
13   </Connections>
14   <Packages>
15     <Package Name="Luo lastausalue">
16   </Package>
17   </Packages>
18 </Biml>
```

Kuva 11. Yhteyden luominen AdventureWorks-kantaan BimlScriptillä

Kuvassa 12 BimlScript-kieltä käyttämällä ohjelmoidaan foreach-silmukka, joka käy läpi SourceMetada-kokoelmaa. Silmukan muuttujaksi annetaan nimi table ja se sisältää yksittäisen taulun metatiedot.

Kuvassa 12 käytetään kahta erityyppistä koodi nugettia. <#...#> Control nugetti ja <#=...#> Text nugettia. Control nugettit antavat mahdollisuuden sisällyttää C#- tai VB-koodia Biml-kieleen ja Text nugettit arvioivat annetun lausekkeen ja kääntävät sen tekstiksi Biml-merkintäkieleen. (Leonard ym. 2017, 36.)

Silmukan sisälle sijoitetaan ExecuteSQL-taski, joka luo taulut Staging-kohdekantaan. ExecuteSQL suorittaa SQL-lauseita SQL Serverin puolella. ExecuteSQL voi käyttää joko SQL-tiedostoja, SSIS-muuttujia tai DirectInput-tagien sisään kirjoitettua SQL-lausetta. Tässä esimerkissä GetDropAndCreateDdl-metodi generoi SQL-lauseen taulun poistamiseksi ja luomiseksi.

Tekstinugettien toiminnallisuus voidaan nähdä kuvan 13 ympyröidyllä alueella olevasta ExecuteSQL-taskin nimestä, joka tulee kuvan 12 mukaisesta taulun skeemasta ja nimestä.

```
<Package Name="Luo lastausalue">
  <Tasks>
    <#foreach (var table in sourceMetaData.TableNodes){#>
      <ExecuteSQL Name="Poista ja luo <#=table.Schema#> <#=table.Name#>" ConnectionName="Staging">
        <DirectInput>
          <#=table.GetDropAndCreateDdl()#>
        </DirectInput>
      </ExecuteSQL>
    <#}#>
  </Tasks>
```

Kuva 12. BimlScriptin käyttö taskien luomisessa

Kuvassa 13 esitellään GetDropAndCreateDdl-metodin generoimaa taulujenluontiskriptiä. Skripti tarkistaa ensin, onko kohdekannassa jo olemassa samannimistä taulua. Jos on, se poistetaan ja luodaan uudestaan määritellyillä sarakkeilla, jotka näkyvät kuvassa Column Definition -kohdassa.

```

<Package Name="Luo-Laajentaminen">
  <Tasks>
    <ExecuteSQL Name="Poista ja luo HumanResources_Department" ConnectionName="Staging">
      <DirectInput>
        SET QUOTED_IDENTIFIER ON
        GO
        -----
        IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[HumanResources].[Department]') AND type IN (N'U'))
        DROP TABLE [HumanResources].[Department]
        GO
        -----
        CREATE TABLE [HumanResources].[Department]
        (
          -- Columns Definition
          [DepartmentID] smallint IDENTITY(1,1) NOT NULL
          , [Name] nvarchar(50) NOT NULL
          , [GroupName] nvarchar(50) NOT NULL
          , [ModifiedDate] datetime NOT NULL
          -- Constraints
        )
        ON "default"
        WITH (DATA_COMPRESSION = NONE)
        GO
      </DirectInput>
    </ExecuteSQL>
  </Tasks>
</Package>

```

Kuva 13. Esikatseluikkunasta voidaan havaita GetDropAndCreateDdl-metodin luoma SQL-lause.

Kuvassa 14 havainnollistaa esikatseluikkunan toiminnallisuutta. Ikkunasta näkee, mitä Biml-tekstiä BimlScript luo. Kuvan esimerkin silmukka käy läpi sourceMetadata-kokoelman ja tallentaa taulujen skeeman ja nimen XML-komentiksi. Tällä tavoin voidaan nopeasti tarkistaa, onko yhteys kunnossa tai mitä tauluja lähdetietokannasta löytyy. Esikatseluikkuna on voimakas työkalu vikojen etsimisessä ja Biml-kielen kirjoittamisessa.

```

1  <#var sourceConnection = SchemaManager.CreateConnectionNode("AW",@"data source=pc\SQLEXPRESS;
2     initial catalog=AdventureWorks;
3     provider=SQLNCLI11.1;
4     integrated security=SSPI;
5     auto translate=False");
6  var sourceMetadata = sourceConnection.ImportDB("HumanResources", "", ImportOptions.ExcludeViews | ImportOptions.ExcludeIndex
7     | ImportOptions.ExcludeColumnDefault | ImportOptions.ExcludeForeignKey | ImportOptions.ExcludeCheckConstraint
8     | ImportOptions.ExcludePrimaryKey | ImportOptions.ExcludeUniqueKey);
9  #>
10 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
11   <#foreach (var table in sourceMetadata.TableNodes){#>
12     <!--<#=table.Schema#> <#=table.Name#> -->
13     <#>
14 </Biml>

```

```

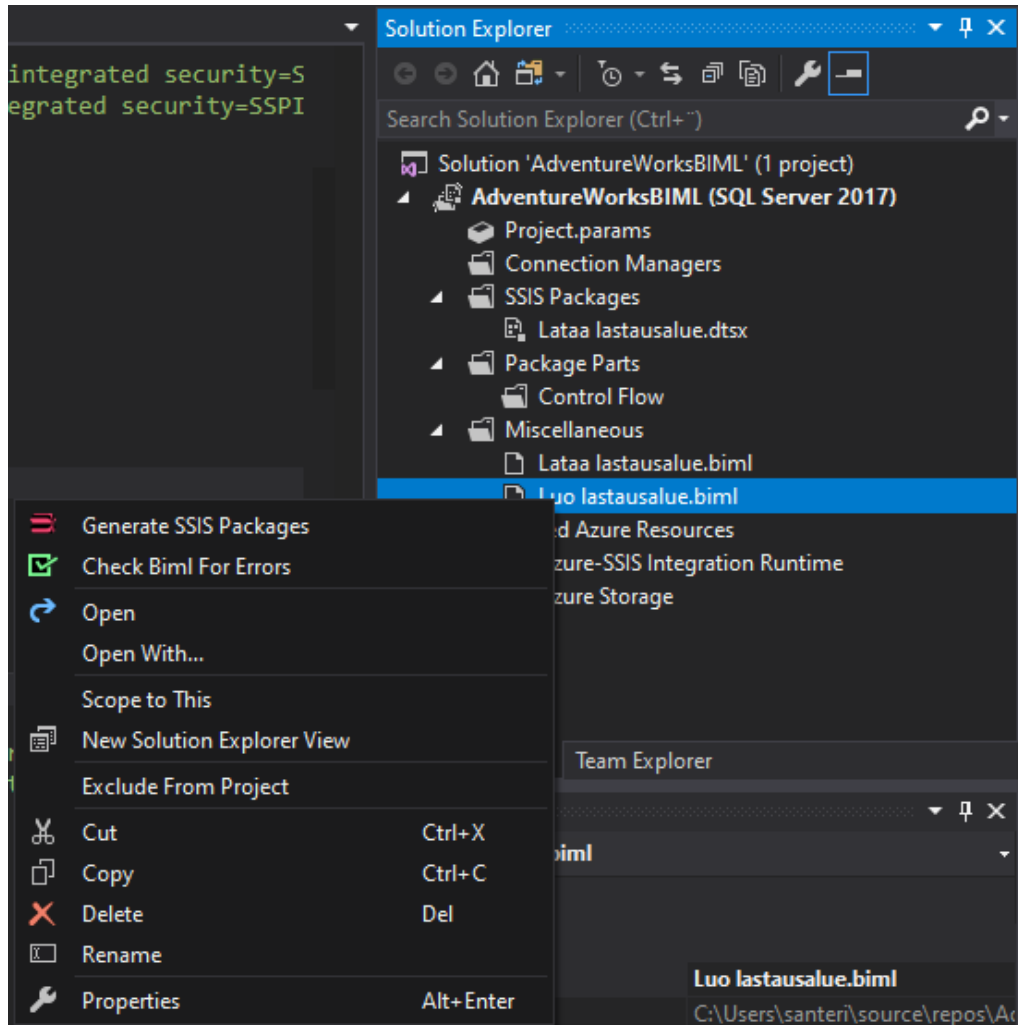
1  <Biml xmlns="http://schemas.varigence.com/biml.xsd">
2     <!--HumanResources Department -->
3     <!--HumanResources Employee -->
4     <!--HumanResources EmployeeDepartmentHistory -->
5     <!--HumanResources EmployeePayHistory -->
6     <!--HumanResources JobCandidate -->
7     <!--HumanResources Shift -->
8 </Biml>

```

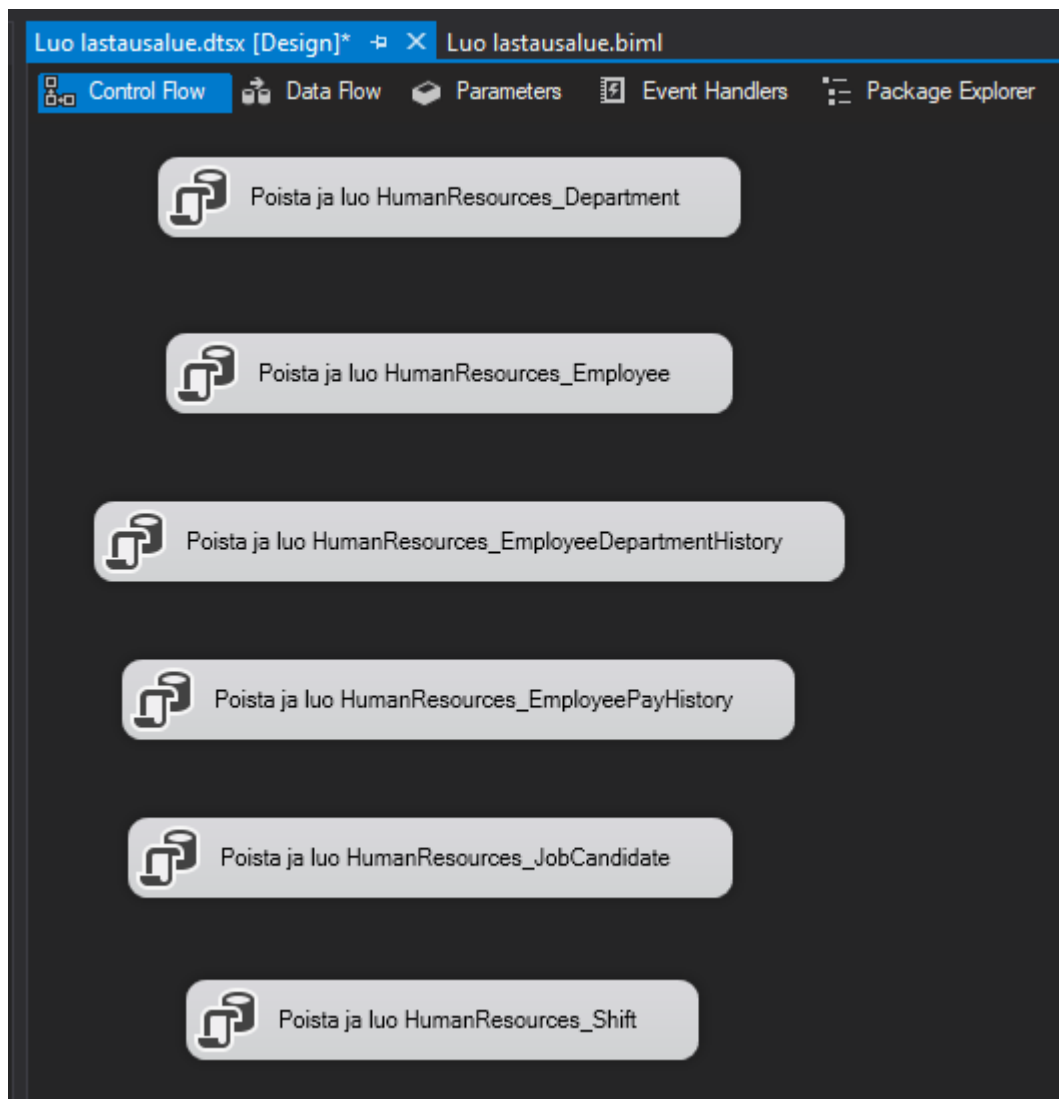
Kuva 14. Esikatseluikkunan käyttö

Biml-tiedostot käännetään SSIS-paketeiksi kuvan 15 mukaisesti. Visual Studio projektikansiossa Biml-tiedostot ovat Miscellaneous-alikansiossa. Valitsemalla Biml-tiedosto hiiren oikealla painikkeella avautuu kuvan mukainen valikko. Paketti käännetään valitsemalla Generate SSIS packages -painike. Käännetty paketti nä-

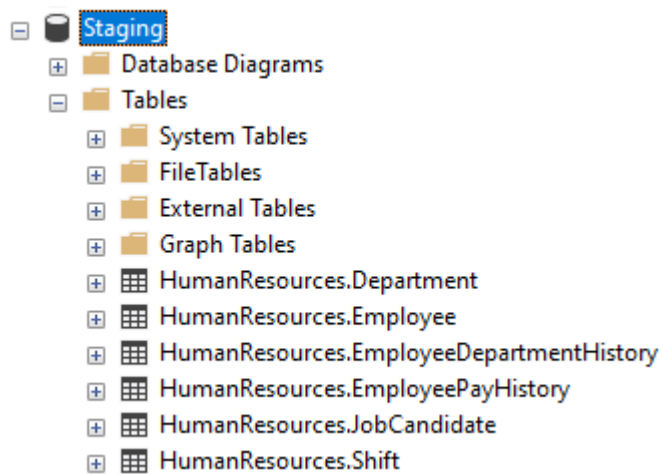
kyä kuvassa 16. Luodun paketin toiminnallisuus koostuu pelkistä ExecuteSQL-taskeista, jotka poistavat ja luovat kohdekantaan tauluja. Kun paketti ajetaan, aiemmin luotu Staging-kanta näyttää kuvan 17 mukaiselta.



Kuva 15. Visual Studion projektikansio ja paketin kääntäminen.



Kuva 16. Biml-tiedostosta käännetty SSIS-paketti



Kuva 17. Lastausalue paketin suorittamisen jälkeen

## 5.5.2 Datan tuominen lastausalueelle

Seuraavaksi luodaan uusi Biml-tiedosto ja nimetään se Lataa lastausalueeksi. Sen pohjaksi otetaan Luo lastausalue-skripti ja muokataan sitä hieman. Yhteyksiin lisätään AdventureWorks-tietokanta nimellä AW ja poistetaan foreach-silmukan sisällä oleva ExecuteSQL-komento. Kuvassa 18 näkyy käytetty pohja.

```

1 <#var sourceConnection = SchemaManager.CreateConnectionNode("AW",@"data source=pc\SQLSERVER;
2                                     initial catalog=AdventureWorks;
3                                     provider=SQLNCLI11.1;
4                                     integrated security=SSPI;
5                                     auto translate=False");
6
7     var sourceMetaData = sourceConnection.ImportDB("HumanResources","",ImportOptions.ExcludeViews);
8
9 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
10     <Connections>
11         <Connection Name="Staging" ConnectionString="data source=pc\SQLSERVER;initial catalog=Staging;pr
12     </Connections>
13     <Packages>
14         <Package Name="Lataa lastausalue">
15             <Tasks>
16                 <#foreach (var table in sourceMetaData.TableNodes){#>
17                     <#>
18                 </Tasks>
19         </Package>
20     </Packages>
21 </Biml>

```

Kuva 18. Taulujen lataus

Esimerkissä käytettävä lähde on OleDbSource ja päämäärä on OleDbDestination kuvan 19 mukaisesti.

Kuvan 19 mukaisesti foreach-silmukan sisään lisätään Dataflow-elementti, joka määrittää Data Flow -taskin. Data Flow -tasolla tapahtuu ETL-prosessin muunnosvaihe, jossa taulujen sisältämää dataa voi muokata lähteen ja päämäärän välillä. Transform-elementin sisältämä OleDbSource-elementti määrittää tiedon lähteen ja OleDbDestination määrittää mihin tieto tallennetaan.

DirectInput-elementin tagien sisään voidaan kirjoittaa SELECT-lauseke, jolla haetaan tiedot tietokannasta. ExternalTableOutput-elementillä määritellään, että mihin kohdetauluun haetut rivit ladataan.

```

<Tasks>
  <#foreach (var table in sourceMetaData.TableNodes){#>
    <Dataflow >
      <Transformations>
        <OleDbSource>
          <DirectInput>

          </DirectInput>
        </OleDbSource>

        <OleDbDestination>
          <ExternalTableOutput />
        </OleDbDestination>
      </Transformations>
    </Dataflow>
  <#}&#gt;
</Tasks>

```

Kuva 19. Taulujen lataukset suoritetaan Dataflow-tasolla.

Elementteihin lisätään attribuutit kuvan 20 mukaisesti BimlScriptiä käyttäen. DirectInput-tagien sisällä oleva SQL-kysely hakee taulun jokaisen sarakkeen ja rivin. ExternalTableOutput-elementti kertoo, mihin kohdetauluun haettu tieto tallennetaan. Koska lähdetaulu ja kohdetaulu ovat saman muotoiset ja nimiset, osaa SSIS itse kartoittaa, mikä lähdetaulun sarake vastaa samaa kohdetaulun saraketta.

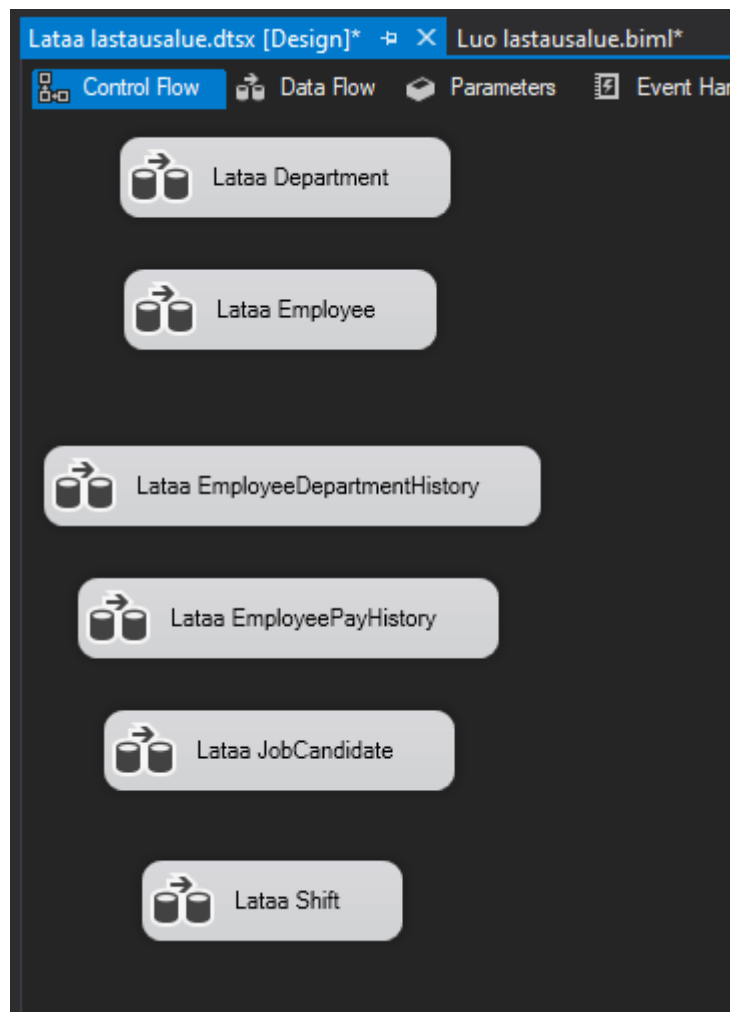
```

<Tasks>
  <#foreach (var table in sourceMetaData.TableNodes){#>
    <Dataflow Name="Lataa <#=table.Name#>">
      <Transformations>
        <OleDbSource Name="Hae taulu AdventureWorks_<#=table.Schema#>_<#=table.Name#>" ConnectionName="AW">
          <DirectInput>
            SELECT *
            FROM
              <#=table.Schema#>.<#=table.Name#>
          </DirectInput>
        </OleDbSource>
        <OleDbDestination Name="Lataa Staging <#=table.Schema#>_<#=table.Name#>" ConnectionName="Staging">
          <ExternalTableOutput Table=" [<#=table.Schema#>]. [<#=table.Name#>]" />
        </OleDbDestination>
      </Transformations>
    </Dataflow>
  <#}&#gt;
&lt;/Tasks&gt;
</pre>
</div>
<div data-bbox="185 788 731 807" data-label="Caption">
<p>Kuva 20. BimlScriptin käyttö Dataflow-komponenttien kanssa</p>
</div>
```

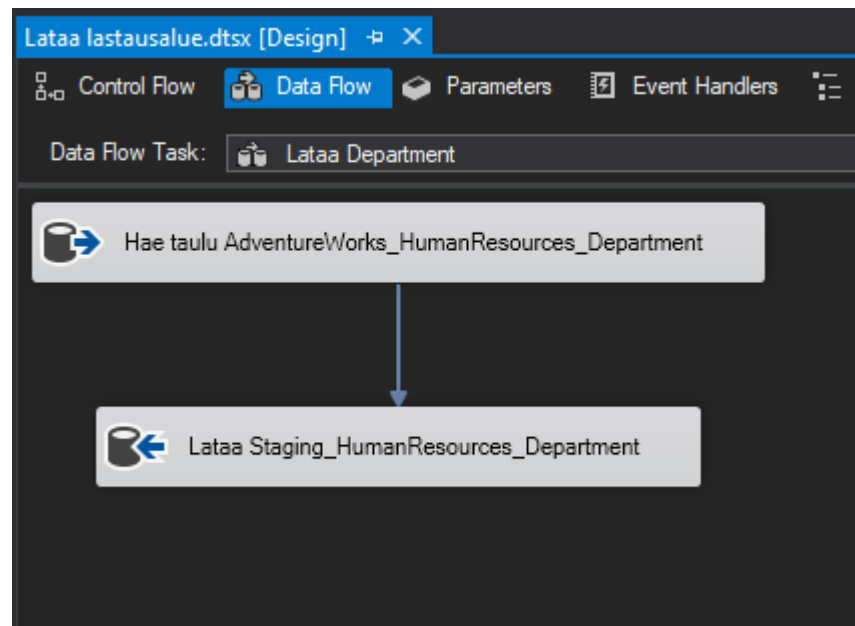


Kuvassa 21 näkyy käännetty paketti Control Flow -tasolla. Paketti koostuu kuudesta Data Flow -taskista. Kuvassa 22 Lataa Department -taski näkyy Data Flow -tasolla ja siitä näkee tiedon lähteen ja päämäärän.

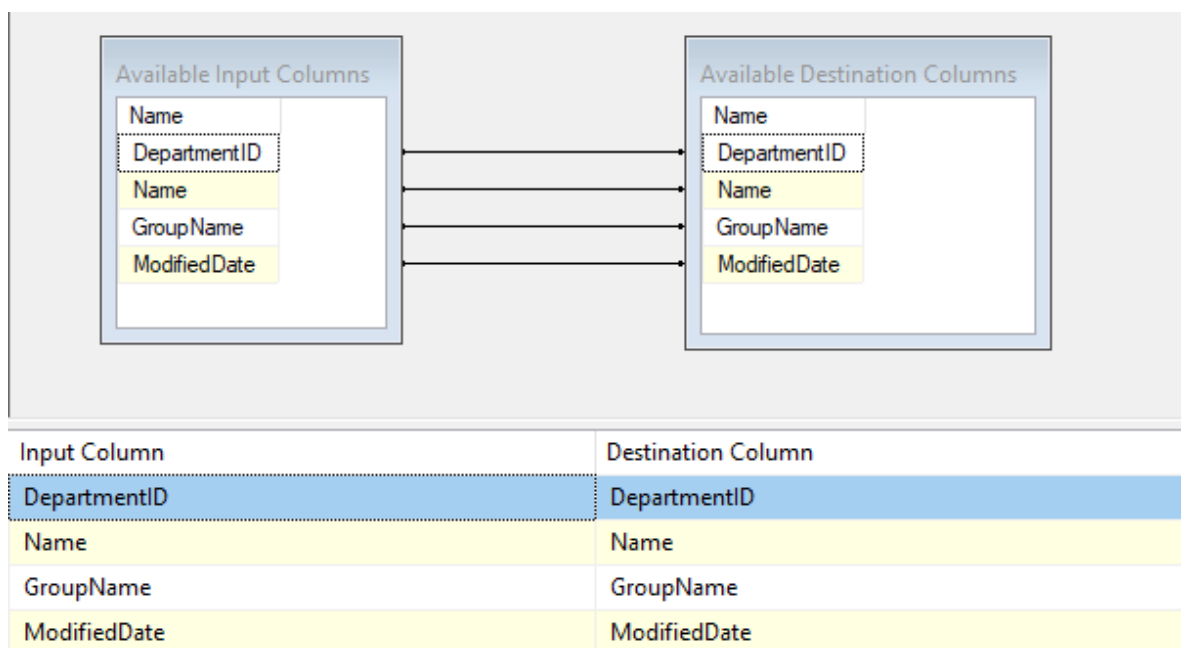
Kuva 23 näyttää, miten sarakkeet on kartoitettu lähteen ja kohteen välillä. Jos sarakkeiden nimet eroavat toisistaan, SSIS ei pysty kartoittamaan niitä automaattisesti. Tämä aiheuttaa töitä pakettien ylläpidossa, koska aina kun sarakkeen nimeä vaihtaa, se pitää kartoittaa paketissa uudestaan.



Kuva 21. Käännetty paketti Control Flow -tasolta

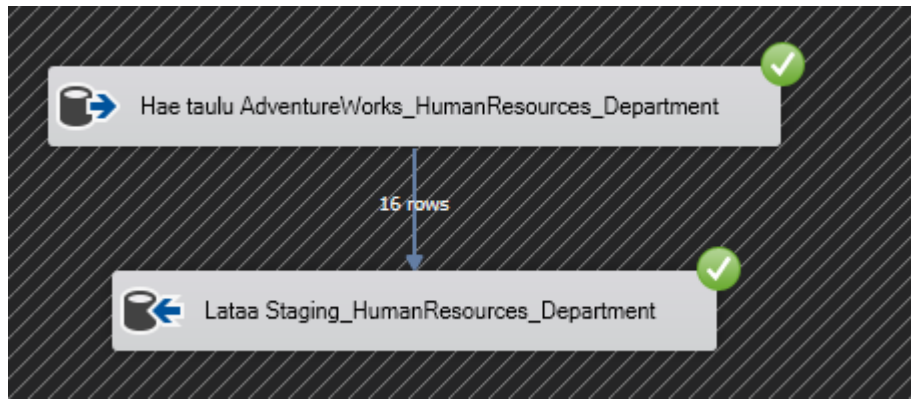


Kuva 22. Lataa Department -taksi Data Flow -tasolla.



Kuva 23. Sarakkeiden kartoitukset Department-taululle.

Kun paketti suoritetaan, kuvasta 24 nähdään, että lähteestä on haettu 16 riviä ja ne on tallennettu onnistuneesti kohdetauluun. Kuvassa 25 näkyy tallennetut rivit Staging-kannan Department-taulussa.



Kuva 24. Data Flow -näkyminen suorituksen aikana.

	DepartmentID	Name	GroupName	ModifiedDate
1	1	Engineering	Research and Development	2008-04-30 00:00:00.000
2	2	Tool Design	Research and Development	2008-04-30 00:00:00.000
3	3	Sales	Sales and Marketing	2008-04-30 00:00:00.000
4	4	Marketing	Sales and Marketing	2008-04-30 00:00:00.000
5	5	Purchasing	Inventory Management	2008-04-30 00:00:00.000
6	6	Research and Development	Research and Development	2008-04-30 00:00:00.000
7	7	Production	Manufacturing	2008-04-30 00:00:00.000
8	8	Production Control	Manufacturing	2008-04-30 00:00:00.000
9	9	Human Resources	Executive General and Administration	2008-04-30 00:00:00.000
10	10	Finance	Executive General and Administration	2008-04-30 00:00:00.000
11	11	Information Services	Executive General and Administration	2008-04-30 00:00:00.000
12	12	Document Control	Quality Assurance	2008-04-30 00:00:00.000
13	13	Quality Assurance	Quality Assurance	2008-04-30 00:00:00.000
14	14	Facilities and Maintenance	Executive General and Administration	2008-04-30 00:00:00.000
15	15	Shipping and Receiving	Inventory Management	2008-04-30 00:00:00.000
16	16	Executive	Executive General and Administration	2008-04-30 00:00:00.000

Kuva 25. Department-taulu paketin suorittamisen jälkeen.

### 5.5.3 Metatietojen määrittely ja lukeminen

Tässä esimerkissä yhdistellään AdventureWorks-kannan Person-skeeman alla olevia osoitetietoja sisältäviä tauluja yhdeksi dimensiotauluksi. Taulut yhdistetään ja tallennetaan dimAddress-nimiseen näkymään. Näkymän luonti suoritetaan kuvan 26 SQL-lausekkeella. Näkymän metatiedot tallennetaan Excel-tilukseen, josta ne luetaan BimlScript-kielen avulla Biml-objekteiksi, joiden pohjalta generoidaan SQL-lauseke taulujen luomiseen ja poistamiseen, sekä ETL-paketti tiedonsiirrolle näkymästä luotuun tauluun.

Tämän kaltainen keskitetty metatietojen hallinta mahdollistaa laajojenkin ETL-prosessien rakentamisen. Tässä esimerkissä käytetään metatietojen hallintaan vain yksinkertaista yhden sivun Excel-taulukkoa.

```
CREATE OR ALTER VIEW dimAddress
AS
SELECT a.[BusinessEntityID]
      ,d.Name as AddressType
      ,c.AddressLine1 as Address
      ,c.City
      ,c.PostalCode
      ,e.Name as StateName
FROM Person.[BusinessEntity] a
LEFT JOIN Person.BusinessEntityAddress b ON b.[BusinessEntityID] = a.[BusinessEntityID]
LEFT JOIN Person.Address c ON c.AddressID = b.AddressID
LEFT JOIN Person.AddressType d ON d.AddressTypeID = b.AddressTypeID
LEFT JOIN Person.StateProvince e ON e.StateProvinceID = c.StateProvinceID
```

Kuva 26. DimAddress-näkymän luonti

Ensin luodaan Excel-tiedosto ja annetaan sille nimeksi metatieto ja annetaan taulukolle nimeksi Taulut. Taulukkoon lisätään kuvassa 27 näkyvät sarakkeet ja rivit. Ensimmäinen sarake määrittää taulun nimen, josta tieto noudetaan, toinen sarake määrittää taulun sarakkeen nimen ja kolmas sen tyyppin.

1	Taulu	Sarake	Tyyppi
2	dimAddress	BusinessEntityID	Int32
3	dimAddress	AddressType	String
4	dimAddress	Address	String
5	dimAddress	City	String
6	dimAddress	PostalCode	String
7	dimAddress	StateName	String
8			
9			

Taulut (+)

Kuva 27. Metatiedot Excel-taulukossa

Excel-taulukko luetaan sisään kuvan 28 BimlScript-koodia käyttäen. Se on metodityyppinen skripti, jota voi kutsua muista Biml-tiedostoista, ja se palauttaa Excel-taulukosta noudetut rivit. Excel-tiedostojen sisäänluku vaatii Microsoft Access Database Enginen.

ExternalDataAccess.GetDataTable-metodi mahdollistaa Excel-taulukoiden lukemisen ja muokkaamisen SQL-kyselyillä. Metodille annetaan parametreina yhteystiedot sekä kysely. Kuvassa näkyvä kysely `SELECT * FROM [Taulut$]` lukee kuvan 27 taulukon tiedot ja tallentaa ne Taulut-nimiseen muuttujaan. Lopuksi määritellään, mitä BimlScript palauttaa sitä kutsuvalle skriptille CustomOutput-ominaisuuden avulla.

Kuvassa 29 näkyy, miten tämän tyyppistä skriptiä kutsutaan kohdassa CallBimlScriptWithOutPut. CallBimlScriptWithOutput-metodille annetaan parametreina tiedostopolku kutsuttavaan Biml-tiedostoon sekä muuttuja, johon sen palauttama tieto tallennetaan. Muuttujan nimi on kuvassa metadata ja se määritellään dynaamisesti dynamic-tyyppillä.

Kuvassa näkyvällä foreach-silmukalla käydään läpi haetut tiedot ja varmistetaan, että tietojen haku onnistui.

```
<#@ import namespace="System.Data" #>
<#
var MetadataFile = @"C:\Users\SanteriNiemi\Desktop\Metatieto.xlsx";
var MetadataConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
+ MetadataFile
+ ";Extended Properties='"EXCEL 12.0 XML;HDR=YES"'";
DataTable Taulut = ExternalDataAccess.GetDataTable(MetadataConnectionString, String.Format("SELECT * FROM [Taulut$]"));
#>
<#
CustomOutput.Taulut = Taulut;
#>
```

Kuva 28. Metadata.biml

```

<#@ template tier="0" #>
<#@ import namespace="System.Data" #>

<# dynamic metadata; #>
<#=CallBimlScriptWithOutput(@"..\AdventureWorks\Metadata.biml", out metadata) #>
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <#foreach ( var table in metadata.Taulut.Rows){#>
    <!-- <#=table["Taulu"]#> <#=table["Sarake"]#> <#=table["Tyyppi"]#>-->
    <#)#>
  </Biml>

<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <!-- dimAddress BusinessEntityID Int32-->
  <!-- dimAddress AddressType String-->
  <!-- dimAddress Address String-->
  <!-- dimAddress City String-->
  <!-- dimAddress PostalCode String-->
  <!-- dimAddress StateName String-->
</Biml>

```

Kuva 29. Onnistunut tietojen haku Excel-tilukosta

Kuvassa 30 metatietojen pohjalta generoidaan dimAddress-tilu Biml-tiluobjektiksi, joka näkyy kuvassa 31. Table-elementin attribuuteiksi on annettu tilun nimi sekä skeeman nimi, jonka alle tilu luodaan.

Tilujektin sarakkeet määritellään ja luodaan Columns-elementin sisällä olevalla foreach-silmukalla, jolla käydään läpi Excel-tilukon metatiedot. Ensin tarkistetaan if-lauseella, mitä tietotyyppiä sarake on, jotta Biml-kääntäjä osaa kääntää ne oikeiksi SQL-vastikkeiksi. Esimerkiksi kuvan 30 rivillä 25 näkyvän DataType="String" Length="100" -elementin SQL-vastine on NVARCHAR (100).

```

1 <#@ template tier="0" #>
2 <#@ import namespace="System.Data" #>
3
4 <# dynamic metadata; #>
5 <#=CallBimlScriptWithOutput(@"..\AdventureWorks\Metadata.biml", out metadata) #>
6 <Biml xmlns="http://schemas.varigence.com/biml.xsd">
7   <Connections>
8     <Connection Name="Staging" ConnectionString="data source=localhost;Initial catalog=Staging;" />
9   </Connections>
10  <Databases>
11    <Database Name="STG" ConnectionName="Staging" />
12  </Databases>
13  <Schemas>
14    <Schema DatabaseName="STG" Name="temp" />
15  </Schemas>
16  <Tables>
17    <Table Name="dimAddress" SchemaName="STG.temp">
18      <Columns>
19        <#foreach ( var column in metadata.Taulut.Rows) { #>
20          <#if(column["Tyyppi"] == "Int32") { #>
21            <Column Name="<#=column["Sarake"]#>" DataType="Int32" />
22          <#} #>
23          <#if(column["Tyyppi"] == "String") { #>
24            <Column Name="<#=column["Sarake"]#>" DataType="String" Length="100" />
25          <#} #>
26        <#} #>
27      </Columns>
28    </Table>
29  </Tables>
30 </Biml>
31

```

Kuva 30. Ympäristö.biml

```

<Tables>
  <Table Name="dimAddress" SchemaName="STG.temp">
    <Columns>
      <Column Name="BusinessEntityID" DataType="Int32" />
      <Column Name="AddressType" DataType="String" Length="100" />
      <Column Name="Address" DataType="String" Length="100" />
      <Column Name="City" DataType="String" Length="100" />
      <Column Name="PostalCode" DataType="String" Length="100" />
      <Column Name="StateName" DataType="String" Length="100" />
    </Columns>
  </Table>
</Tables>

```

Kuva 31. DimAddress-taulu esikatseluikkunassa

Kuvassa 32 näkyy 2 riviä BimlScript-kieltä, jolla generoidaan SQL-lauseke taulun luomiselle tauluobjektista, ja sen alla esikatseluikkunassa näkyy generoitu SQL-lauseke. Kysely pitää suorittaa SQL-palvelimella ennen varsinaista tiedonsiirtoa, jotta SSIS voi tunnistaa kartoitukset kahden eri taulun sarakkeiden välillä. Kuvan ensimmäisellä rivillä näkyvä template tier näkyy myös kuvassa 30. Template tier kertoo

Biml-kääntäjälle, missä järjestyksessä Biml-tiedostot käännetään. Näin mahdollistaen Biml-tiedostojen pätkimisen useampaan osaan, jotta koodin luettavuus helpottuu.

```
<#@ template tier="10" #>
<#=RootNode.Tables.First().GetDropAndCreateDdl()#>

SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

-----
IF EXISTS (SELECT * from sys.objects WHERE object_id = OBJECT_ID(N'[temp].[dimAddress]'))
DROP TABLE [temp].[dimAddress]
GO

CREATE TABLE [temp].[dimAddress]
(
-- Columns Definition
  [BusinessEntityID] int NOT NULL
, [AddressType] nvarchar(100) NOT NULL
, [Address] nvarchar(100) NOT NULL
, [City] nvarchar(100) NOT NULL
, [PostalCode] nvarchar(100) NOT NULL
, [StateName] nvarchar(100) NOT NULL
```

Kuva 32. SQL-lauseen luonti ja esikatseluikkuna

Kuvassa 33 näkyy Paketti.biml-skripti, jossa tapahtuu tämän esimerkin varsinainen tiedonsiirto ja paketin luonti. Paketin nimeksi annetaan "Lataa dimAddress". Seuraavaksi määritellään ExecuteSQL-task, jolla tyhjätyään taulu aina kun paketti ajetaan. Sen jälkeen määritellään paketin DataFlow-elementti. Elementin sisällä tapahtuva tiedonsiirto koostuu kahdesta komponentista: OleDbSource ja OleDbDestination.

OleDbSource hakee DirectInput-elementin avulla alussa luodusta näkymästä tiedot, josta ne menevät OleDbDestination-elementtiin, jolla tiedot ladataan äskeisessä vaiheessa luotuun tauluun. OleDbDestination-elementin sisällä oleva ExternalTableOutput-elementin Table-attribuutin arvoksi annetaan kohdetaulun skeeman ja taulun nimi. skeeman nimen määrittely on tässä tärkeää, koska näkymän ja äskettäin luodun taulun nimi on sama dimAddress, mutta näkymän skeema on dbo ja taulun temp.

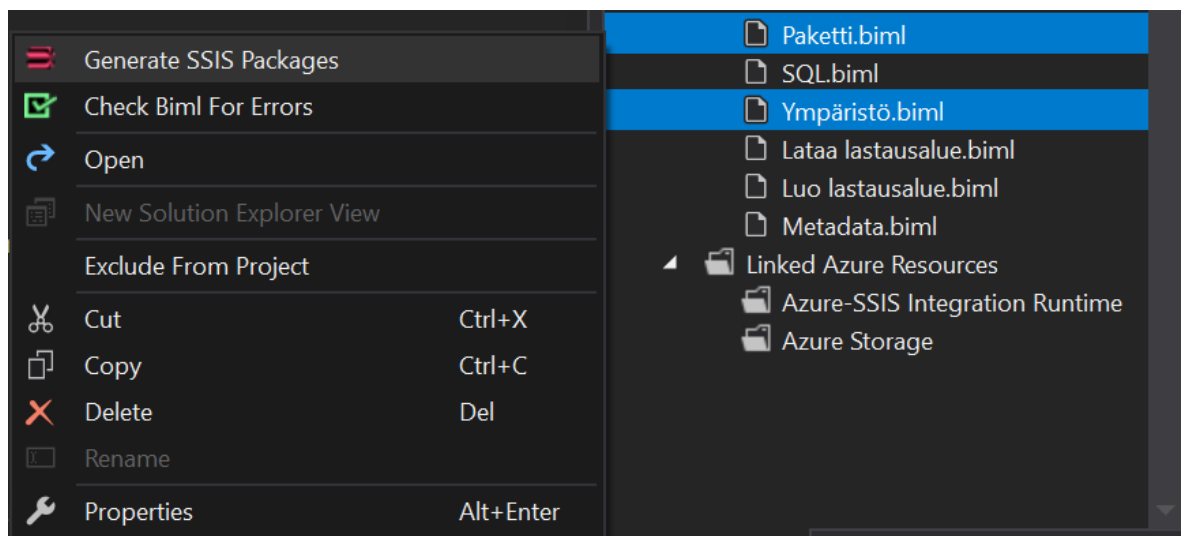


```

<#@ template tier="10" #>
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <Package Name="Lataa dimAddress">
      <Tasks>
        <ExecuteSQL Name="Poista ja luo taulu" ConnectionName="Staging">
          <DirectInput>
            <#=RootNode.Tables.First().GetDropAndCreateDdl()#>
          </DirectInput>
        </ExecuteSQL>
        <Dataflow Name="dimAddress">
          <Transformations>
            <OleDbSource Name="Get Data" ConnectionName="Staging">
              <DirectInput>
                select * from dbo.dimAddress
              </DirectInput>
            </OleDbSource>
            <OleDbDestination Name="Load Data" ConnectionName="Staging">
              <ExternalTableOutput Table="temp.dimAddress"/>
            </OleDbDestination>
          </Transformations>
        </Dataflow>
      </Tasks>
    </Package>
  </Packages>
</Biml>

```

Kuva 33. Paketti.biml

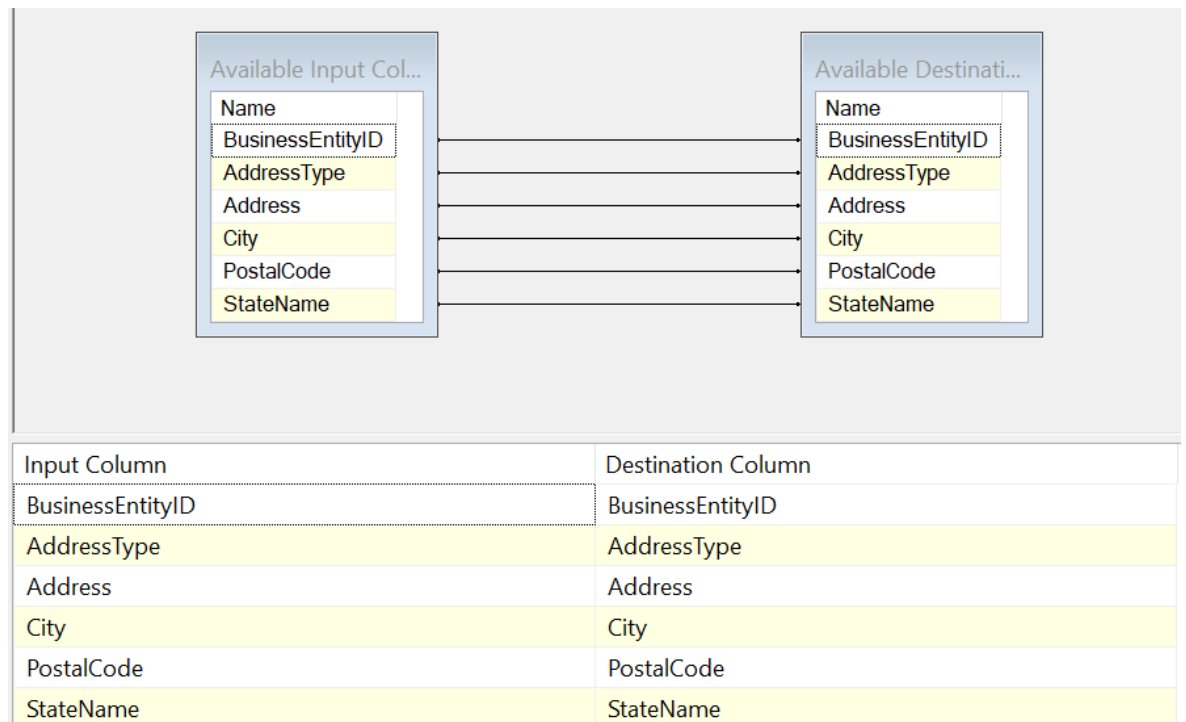


Kuva 34. Paketin kääntäminen

Kun Biml-tiedosto on pätkitty useampaan osaan template tier -ominaisuutta käyttäen, tiedostot pitää kääntää paketeiksi kuvan 34 mukaisesti. Kaikki skriptin osat valitaan ja klikataan generate SSIS packages -painiketta. Käännetyn paketin Control Flow näkyy kuvassa 35 vasemmalla puolella ja Data Flow oikealla puolella. Kuvassa 35 näkyy näkymän ja taulun väliset kartoitukset. Kuvassa 36 näkyy näkymän ja taulun väliset kartoitukset, sekä kuvassa 37 on nähtävillä taulun sisältö paketin suorittamisen jälkeen.



Kuva 35. Lataa dimAddress-paketin control ja-dataflow



Kuva 36. Sarakkeiden kartoitukset

```

1 SELECT TOP (1000) [BusinessEntityID]
2     , [AddressType]
3     , [Address]
4     , [City]
5     , [PostalCode]
6     , [StateName]
7 FROM [Staging].[temp].[dimAddress]

```

BusinessEntityID	AddressType	Address	City	PostalCode	StateName
12	Home	1970 Napa Ct.	Bothell	98011	Washington
123	Home	9833 Mt. Dias Blv.	Bothell	98011	Washington
285	Home	7484 Roundtree Drive	Bothell	98011	Washington
251	Home	9539 Glenside Dr	Bothell	98011	Washington
124	Home	1226 Shoe St.	Bothell	98011	Washington
210	Home	1399 Firestone Drive	Bothell	98011	Washington
88	Home	5672 Hale Dr.	Bothell	98011	Washington

Kuva 37. Temp.dimAddress-taulu paketin ajamisen jälkeen

Tässä luvussa näytettiin esimerkkien avulla, mitä Biml-kielellä voi tehdä. Luvussa 5.5.1 luotiin ja valmisteltiin lastausalue SQL-palvelimelle datan lataamiselle varten AdventureWorks-tietokannasta. Luvussa 5.5.2 kirjoitettiin Biml-tiedosto, joka generoi SSIS-paketteja, joilla tietoa voi siirtää AdventureWorks-kannasta lastausalueelle. Luvussa 5.5.3 näytettiin, miten Excel-taulukosta pystyy lukemaan mukautettuja metatietoja, sekä tutustuttiin Biml-tauluobjektiin, jonka pohjalta pystyy luomaan skriptejä, joilla luoda tauluja SQL-palvelimelle. Luvussa luotiin myös paketti, joka siirsi tietoa näkymästä tauluun.

Luku 6 pohjautuu äskeisessä kappaleessa näkyviin esimerkkeihin, sekä mukautettujen metatietojen hallinta on seuraavassa luvussa hyvin keskeisessä osassa. Luvussa rakennetaan tietovarastokokonaisuus ETL-paketteineen pelkkien metatietojen pohjalta.

## 6 CASE LOGFORCE - salattu

## 7 YHTEENVETO JA POHDINTA

Työssä esiteltiin tietovarastoinnin ja Business Intelligencen tavoitteet. Työssä tutkittiin Business Intelligence Markup Languageen käyttöä tietovaraston kehitysprosessissa. Tietovarastoinnista käytiin läpi sen periaatteita sekä toteutusfilosofioita. Business Intelligencestä käytiin läpi lyhyesti BI-ratkaisujen keskeisimmät hyödyt. Työssä esiteltiin BI-työkaluja, joilla työn BI-ratkaisut toteutettiin. Lisäksi esiteltiin SQL Server Integration Services -ohjelmistoa, jotta voitaisiin näyttää, miten Biml-kieltä voi käyttää apuna ohjelmalliseen ETL-prosessien tekemiseen. Biml-kielestä käytiin läpi sen rakennetta ja sen suhdetta XML-merkintäkieleen. Biml-kielen käyttöä käytiin läpi esimerkkien avulla.

Työtä varten toteutettiin framework, jonka käyttöä esitellään luvussa kuusi. Framework mahdollistaa tietovarastokokonaisuuksien rakentamisen ja ylläpidon metatietoa käyttäen. Frameworkin käyttö ei kuitenkaan ole kuitenkaan kovin helppoa ja se vaatii vielä jatkokehittämistä sen käytettävyyden helpottamiseksi ja lisäominaisuuksien lisäämiseksi. Metatietojen hallintaan ja muokkaamiseen tällä hetkellä käytetty Excel on myös hieman kömpelö työkalu metatietojen hallitsemiseen. Itsekoodattu hallintaohjelma mahdollistaisi vielä helpomman kehityksen ja helpottaisi tietovaraston versionhallintaa.

Kokonaisuuden suunnittelu ja toteutus oli alkuun haastavaa, koska Biml-kieli oli ennestään tuntematon. Haastavaa työstä teki myös Biml-kielen dokumentoinnin ja esimerkkien puute. Projekti oli kuitenkin mielenkiintoinen ja palkitseva. Havaitsin Biml-kielen olevan erinomainen lisä SSIS-pakettien tekemisen helpottamiseksi, koska sen avulla ainakin yksinkertaisimmat aikaa vievät operaatiot saa suoritettua lähes automaattisesti.

## LÄHTEET

- Encyclopaedia Britannica. Ei päiväystä. Database. [Verkkosivu]. [Viitattu 13.4.2020]. Saatavana: <https://www.britannica.com/technology/database>
- Hovi, A. 21.11.2012. Dataa kaikille – kertaratkaisulla. [Blogikirjoitus]. Ari Hovi. [Viitattu 13.4.2020]. Saatavana: <http://www.arihovi.com/dataa-kaikille-kertaratkaisulla/>
- Hovi, A., Hervonen, H. & Koistilainen, H. 2009. Tietovarastot ja Business Intelligence. Jyväskylä: Docendo.
- Kimball, R. 21.8.2008. Slowly Changing Dimensions. [Blogikirjoitus]. Kimball group. Saatavana: <https://www.kimballgroup.com/2008/08/slowly-changing-dimensions/>
- Leonard, A., Currie, S., Alley, J., Andersson, M., Avenant, P., Fellows, B., Peck, S., Smith, R., Sondak, R., Weissman, B. & Wilhelmsen, C. 2017. The Biml Book: Business Intelligence and Data Warehouse Automation. New York: Springer Science+Business Media.
- LogForce. Ei päiväystä. Palvelun kuvaus. [Verkkosivu]. [Viitattu 23.4.2020] Saatavana: <https://www.logforce.fi/palvelun-kuvaus/>
- Microsoft. 2016. SSIS Designer. [Verkkosivu]. Microsoft Corp. [Viitattu 2.5.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/integration-services/ssis-designer?view=sql-server-ver15>
- Microsoft. 2017a. Control Flow. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/integration-services/control-flow/control-flow?view=sql-server-ver15>
- Microsoft. 2017b. Data Flow Task. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/integration-services/control-flow/data-flow-task?view=sql-server-ver15>
- Microsoft. 2017c. Derived Column Transformation. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/integration-services/data-flow/transformations/derived-column-transformation?view=sql-server-ver15>

- Microsoft. 2017d. Parse and Transform JSON Data with OPENJSON. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/relational-databases/json/convert-json-data-to-rows-and-columns-with-openjson-sql-server?view=sql-server-ver15>
- Microsoft. 2018a. SQL Server Integration Services. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15>
- Microsoft. 2018b. AdventureWorks. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15>
- Microsoft. 2019a. [MS-DTSX]: Data Transformation Services Package XML File Format. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: [https://docs.microsoft.com/en-us/openspecs/sql\\_data\\_portability/ms-dtsx/235600e9-0c13-4b5b-a388-aa3c65aec1dd](https://docs.microsoft.com/en-us/openspecs/sql_data_portability/ms-dtsx/235600e9-0c13-4b5b-a388-aa3c65aec1dd)
- Microsoft. 2019b. Use Power BI Q&A to explore your data and create visuals. [Verkkosivu]. Microsoft Corp. [Viitattu 13.4.2020]. Saatavana: <https://docs.microsoft.com/en-us/power-bi/power-bi-tutorial-q-and-a>.
- Naeem, T. 16.12.2019. Data Warehouse Concepts: Kimball vs. Inmon Approach. [Blogikirjoitus]. Astera. [Viitattu 13.4.2020]. Saatavana: <https://www.astera.com/type/blog/data-warehouse-concepts/>
- Oiwa Solutions. Ei päiväystä. Sovelluskehitys. [Verkkosivu]. [Viitattu 13.4.2020]. Saatavana: <https://oiwasolutions.fi/tiedolla-johtamisen-palvelut/sovelluskehitys/>
- Oiwa Solutions. Ei päiväystä. Proof of concept. [Verkkosivu]. [Viitattu 14.4.2020]. Saatavana: <https://oiwasolutions.fi/tiedolla-johtamisen-ratkaisut/proof-of-concept-poc/>
- Power BI. Ei päiväystä. What is Power BI? [Verkkosivu]. Microsoft Corporation. [Viitattu 23.34.2020]. Saatavissa: <https://powerbi.microsoft.com/en-us/what-is-powerbi/>
- Rad, R. 2014. Microsoft SQL Server 2014 Business Intelligence Development. Birmingham, United Kingdom: Packt Publishing.
- Rangarajan, S. 1.11.2016. Data Warehouse Design – Inmon versus Kimball. [Blogikirjoitus]. TDAN. [Viitattu 13.4.2020]. Saatavana: <https://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>
- Richards, R. 2004. Pro PHP XML and Web Services. New York: Apress.

Schlichting, D. 2008. What is SQL Server. [verkkosivu]. Database Journal. [Viitattu 15.4.2020]. Saatavana: <https://www.databasejournal.com/features/mssql/article.php/3769211/What-is-SQL-Server.htm>

Varigence. Ei päiväystä. Biml. [Verkkosivu]. [Viitattu 10.4.2020]. Saatavana: <https://varigence.com/biml>

Varigence. Ei päiväystä. FAQ. [Verkkosivu]. [Viitattu 11.4.2020]. Saatavana: <https://varigence.com/FAQ>

Varigence. Ei päiväystä. BimlStudio. [Verkkosivu]. [Viitattu 12.4.2020]. Saatavana: <https://varigence.com/BimlStudio>

Varigence. Ei päiväystä. BimlFlex. [Verkkosivu]. [Viitattu 13.4.2020]. Saatavana: <https://varigence.com/BimlFlex>

Varigence. Ei päiväystä. BimlExpress. [Verkkosivu]. [Viitattu 14.4.2020]. Saatavana: <https://varigence.com/BimlExpress>

W3school. Ei päiväystä. Sql intro. [Verkkosivu]. [Viitattu 11.4.2020]. Saatavana: [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)

W3school. Ei päiväystä. Sql select. [Verkkosivu]. [Viitattu 12.4.2020]. Saatavana: [https://www.w3schools.com/sql/sql\\_select.asp](https://www.w3schools.com/sql/sql_select.asp)

W3school. Ei päiväystä. Sql insert. [Verkkosivu]. [Viitattu 13.4.2020]. Saatavana: [https://www.w3schools.com/sql/sql\\_insert.asp](https://www.w3schools.com/sql/sql_insert.asp)

W3school. Ei päiväystä. Sql update. [Verkkosivu]. [Viitattu 14.4.2020]. Saatavana: [https://www.w3schools.com/sql/sql\\_update.asp](https://www.w3schools.com/sql/sql_update.asp)

W3school. Ei päiväystä. Sql delete. [Verkkosivu]. [Viitattu 15.4.2020]. Saatavana: [https://www.w3schools.com/sql/sql\\_delete.asp](https://www.w3schools.com/sql/sql_delete.asp)



W3school. Ei päiväystä. Sql views. [Verkkosivu]. [Viitattu 16.4.2020]. Saatavana: [https://www.w3schools.com/sql/sql\\_view.asp](https://www.w3schools.com/sql/sql_view.asp)

W3school. Ei päiväystä. XML Namespaces. [Verkkosivu]. [Viitattu 17.4.2020]. Saatavana: [https://www.w3schools.com/xml/xml\\_namespaces.asp](https://www.w3schools.com/xml/xml_namespaces.asp)