

TIETOVARASTON VALVONTA

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2017
Niko Salomaa

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

SALOMAA, NIKO:

Tietovaraston valvonta

Ohjelmistotekniikan opinnäytetyö, 42 sivua

Kevät 2017

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli luoda tietovaraston latauksien seurantaraportti käyttäen Power BI -työkalua. Tämän lisäksi tavoitteena oli luoda Azuren palveluista järjestelmä, jolla tietovaraston latauksien tiedot saadaan siirrettyä asiakkaan ympäristöstä yhteiseen tietokantaan. Työ tehtiin Evolvit Oy:lle, joka on suomalainen IT-alan yritys. Evolvit tarjoaa asiakkailleen finanssiteknologia sekä business intelligence -palveluita.

Yrityksen tiedot ovat usein useissa eri operatiivisissa järjestelmissä, jolloin tietojen yhdistämiseen käytetään tietovarastoa. Ennen tietovarastoon siirtämistä tiedot ladataan lähdejärjestelmistä työalueelle. Työalueella tiedot yhdenmukaistetaan, minkä jälkeen ne voidaan ladata tietovarastoon. Tietovarastossa tiedot sijoitetaan fakta- ja dimensiotauluihin, minkä jälkeen tiedoista voidaan tehdä kattavaa analysointia BI-työkaluilla.

Azure on Microsoftin julkaisema pilvessä toimiva sovelluspalvelu. Työssä käytiin läpi Azuren palveluista Event Hubs, Stream Analytics sekä Azure DB. Tietovaraston latauksien tiedot siirretään Event Hubsin ja Stream Analyticsin avulla Azuren tietokantaan. Tämän lisäksi esitellään Power BI -raportointityökalua.

Työosuudessa toteutettiin Azuren palveluista järjestelmä, jolla siirrettiin tietovaraston latauksien tiedot asiakkaan ympäristöstä yhteiseen tietokantaan. Tämän jälkeen latauksien tiedoista luotiin raportti, jolta nähtiin epäonnistuneet lataukset sekä niiden historiatietoja.

Asiasanat: tietovarasto, Azure, Power BI

Lahti University of Applied Sciences
Degree Programme in Information Technology

SALOMAA, NIKO:

Monitoring a data warehouse

Bachelor's Thesis in Software Engineering, 42 pages

Spring 2017

ABSTRACT

The goal of this thesis was to create a report that shows the results of loads from a data warehouse. The report was built using Power BI. A secondary goal was to create a system using Azure services where data warehouse load data would be sent from the customer infrastructure to Azure, where the data would be stored in a database. The work was made for Evolvit Oy, which is Finnish IT service company. Evolvit offers finance technology and business intelligence services for its clients.

Companies often have their data stored in several operative systems, in which case a data warehouse is the place where that data will be combined. Before loading data to a data warehouse, the data is transformed to the right form in the staging area. In the data warehouse the data are inserted to fact and dimension tables, and after that the data can be analysed using business intelligence tools.

Azure is cloud computing platform published by Microsoft. The Azure services dealt with in this thesis were Event Hubs, Stream Analytics and Azure DB. The data warehouse's load data are sent to the Event Hubs, and after that the Stream Analytics reads data from the Event Hubs and inserts it to the Azure database. Besides that, the thesis presents reporting tool Power BI.

The empirical part of the thesis started by creating a system using Azure services to move data warehouse load data to an Azure database from the customer infrastructure. After that, a report was made using load data. The report shows failed loads and history data of loads.

Key words: data warehouse, Azure, Power BI

SISÄLLYS

1	JOHDANTO	1
2	TIETOVARASTOINTI	2
2.1	Yleistä	2
2.2	Historia	2
2.3	Rakenne	3
2.3.1	Työalue	4
2.3.2	Data vault	4
2.3.3	Data mart	5
2.3.4	Tähti- ja lumihuutalemalli	6
2.3.5	Faktataulu	8
2.3.6	Dimensiotaulut	8
3	AZURE	10
3.1	Azuren palvelut	10
3.2	Event Hubs	11
3.2.1	Lähetäminen	12
3.2.2	Osiot	12
3.2.3	Vastaanottaminen	13
3.3	Stream Analytics	14
3.4	Datan liikkuminen	14
3.5	Azure SQL Database	15
3.6	Resurssiryhmä	16
4	POWER BI	17
4.1	Raportoinnin vaiheet	17
4.2	Lataaminen	18
4.2.1	Datan tuonti	18
4.2.2	Suora kysely	18
4.2.3	Analysis Service	19
4.3	Mallintamien	19
4.4	Muokkaaminen	20
4.5	DAX	21
4.6	Visualisointi	22
4.7	Power BI Service	24

5	VALVONTASOVELUS	25
5.1	Valvonta	25
5.2	Arkkitehtuuri	26
5.3	Event Hubin luominen	27
5.4	Datan siirtäminen Event Hubiin	29
5.5	Stream Analytics	30
5.6	Tietokannan luominen	33
5.7	Tietokannan taulut	35
5.8	Datan esittäminen raportilla	36
5.8.1	Datan lataaminen	36
5.8.2	Mallintaminen	36
5.8.3	Datan visualisointi raportiksi	38
6	YHTEENVETO	40
	LÄHTEET	41

1 JOHDANTO

Nykyään tietovarastojen kysyntä on suurta ja BI-yritysten asiakkaille rakentamat ja ylläpitämät tietovarastot sijaitsevat omilla palvelimillaan, jolloin tietovarastojen valvonta on työlästä ja vie paljon resursseja.

Automatisoidut lataukset käynnistyvät yleensä vähintään kerran päivässä, ja jos lataukset jostain syystä epäonnistuvat, pitäisi siitä mahdollisimman nopeasti saada tieto, jotta vikaa päästään tutkimaan ja korjaamaan.

Evolvit Oy on suomalainen IT-alan yritys, joka työllistää Suomessa noin 100 henkilöä ja toimii kolmessa eri kaupungissa: Espoossa, Tampereella ja Lahdessa. Evolvit tarjoaa asiakkailleen finanssiteknologia- ja business intelligence -palveluita, joista yhtenä palveluna on tietovarastojen luominen ja niiden ylläpitäminen.

Työn tavoitteena on luoda tietovaraston latauksien seurantaraportti, jolla näytetään kuluvan päivän epäonnistuneet lataukset sekä latauksien historiatietoja. Tämän lisäksi luodaan Azuren omista palveluista järjestelmä, jolla tietovarastojen latauksien tiedot saadaan kuljetettua asiakkaiden ympäristöistä yhteiseen tietokantaan. Tässä työssä käydään läpi tietovarastointia ja sen vaiheita, Microsoftin Azure-pilvipalvelua sekä Power BI -raportointityökalua.

Ensimmäisessä luvussa kerrotaan tietovarastoinnista, sen rakenteesta ja mallintamisesta. Toisessa Luvussa kerrotaan Azuren pilvipalvelusta ja sen tietyistä palveluista, joita opinnäytetyön työosuudessa käytetään.

Kolmannessa luvussa kerrotaan raportoinnista Power BI -sovelluksella.

Viidennessä luvussa esitellään työn vaiheet datan lähettämisestä Azureen ja sieltä raportille. Viimeinen luku sisältää yhteenvedon ja tietoa sovelluksen tulevaisuudesta.

2 TIETOVARASTOINTI

Yleensä yritysten ja organisaatioiden data sijaitsee operatiivisten järjestelmien tietokannoissa, jotka eivät sovi hyvin tietojen raportointiin ja analysointiin. Datan hyödyntäminen analyyseissä ja raporteissa vaatii yrityksen käyttötarpeita vastaavan tietovaraston, jolloin yrityksen dataa voidaan käsitellä Business Intelligence -työkaluilla. (Hovi 2009, 14.)

2.1 Yleistä

Tietovaraston tarkoituksena on saada yhdistettyä yrityksen operatiivisten järjestelmien tiedot yhteiseen tietovarastoon yhtenäisessä muodossa, jolloin pystytään tekemään kattavia raportteja yrityksen tarpeisiin.

Yrityksillä saattaa olla esimerkiksi monissa eri operatiivisissa järjestelmissä asiakkaiden tietoja ja näihin tietoihin ei pääse käsiksi yhdestä paikasta, jolloin yrityksen koko asiakaskunnasta on hankala tehdä analyysia. Tällöin on parasta tuoda kaikkien järjestelmien tiedot yhteen tietovarastoon. Tietovarastossa samaa tarkoittavat tiedot yhdistetään yhteen tauluun ja tiedot yhdenmukaistetaan, sillä yleensä eri järjestelmissä on käytetty eri nimityksiä samoille asioille. Esimerkiksi yhdessä järjestelmässä suokupuolikoodit voi olla muodossa "M" ja "N" ja toisessa "1" ja "0". Kun tiedot ovat yhdessä paikassa, on helppoa tehdä raportointia esimerkiksi koko asiakaskunnasta ja asiakkaiden kannattavuudesta. (Hovi 2009, 14 - 15.)

2.2 Historia

Tietotekniikan alkuaajoista asti on puhuttu, kuinka tietotekniikasta on hyötyä tietojen raportoinnissa. Kun tekniikka kehittyi, alettiin kehittämään erilaisia järjestelmiä johdon ja päätöksenteon tueksi, kuten DSS-järjestelmä (Decision Support Systems). Näihin järjestelmiin tiedot tallennettiin usein korkealle summatasolle, joka koitui ongelmaksi, sillä johdolla oli usein tarve porautua yksityiskohtaisempiin tietoihin.

Tietotekniikan kehittyessä siihen vaiheeseen, että jokaisella alkoi olla oma henkilökohtainen tietokone ja taulukkolaskentaohjelma, tultiin pisteeseen, jossa ajateltiin, että jokainen voisi analysoida tietoja itsenäisesti.

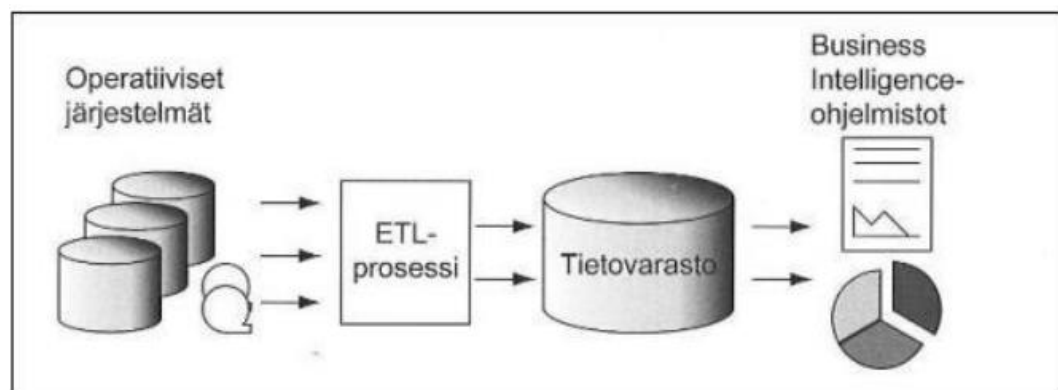
Ongelmaksi huomattiin, että tiedot olivat usein puutteellisia tai vanhentuneita, minkä johdosta tehtiin pahimmillaan vääriä päätöksiä.

Sana tietovarasto esiintyi ensimmäisen kerran 1988-luvulla Devlinin ja Murphyn IBM Systems Journal -lehteen kirjoittamassaan artikkelissa. Ensimmäinen tietovarastokirja kirjoitettiin vuonna 1990 W. H. Inmonian toimesta, jota pidetään myös tietovarastoinnin isänä.

2.3 Rakenne

Tietovarasto on yleistermi koko tietovarastointitoteutukselle, mutta jokaista aluetta kutsutaan myös termillä tietovarasto. Tietovaraston rakenne riippuu siitä, kuinka iso projekti on tai millä tavalla tieto halutaan mallintaa.

Tietovaraston osia voivat olla muun muassa työalue (Staging Area), Data Vault, sekä Data Mart. Tietovaraston rakentamisessa ainoa pakollisena osana on Data Mart, muiden osien tarpeellisuuden määrittää se, millaista arkkitehtuuria tietovaraston mallintamisessa käytetään. Työalue ei varsinaisesti ole osa tietovarastoa, sillä sitä käytetään vain tiedon esikäsittelyyn. Alla oleva Kuvio 1 kuvaa tietojen etenemisen asiakkaan järjestelmistä tietovarastoon ja sieltä BI-ohjelmistoihin analysoitavaksi ja raportoitavaksi. (Hovi 2009, 14 - 15.)



KUVIO 1. Tietovarastoinnin perusarkkitehtuuri (Hovi 2009, 14)

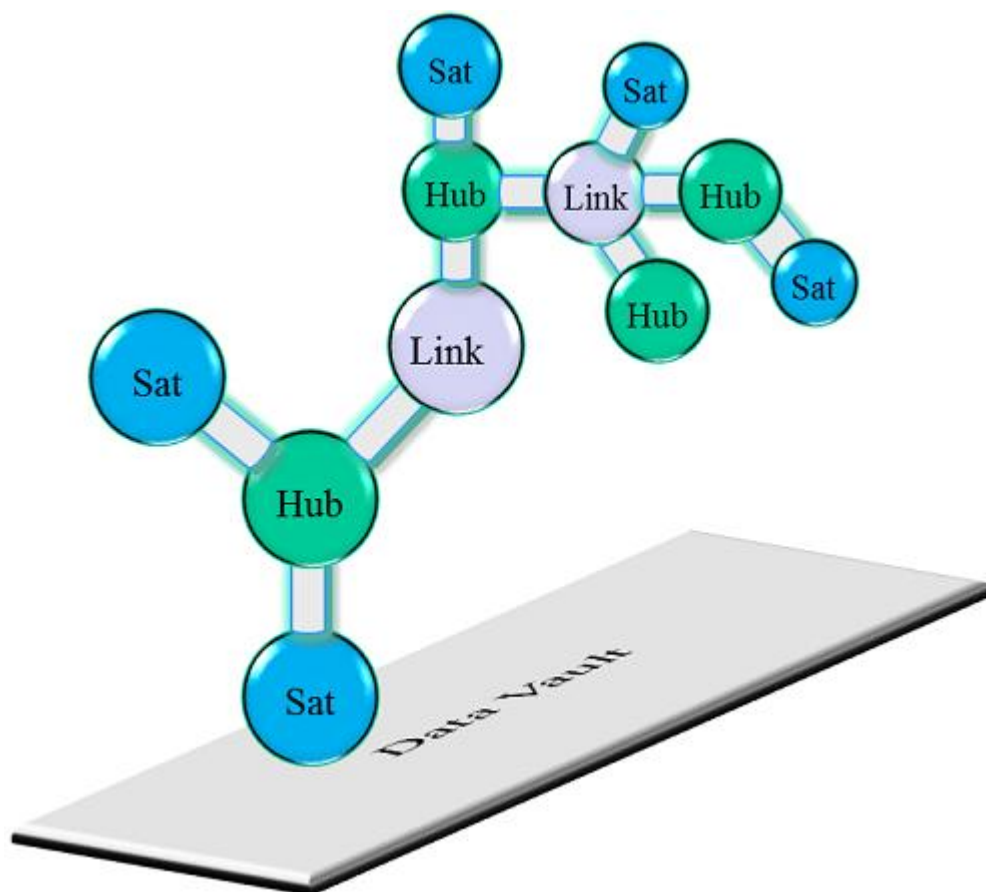
2.3.1 Työalue

Tietojen lataaminen työalueelle on yleensä ensimmäinen vaihe tietovarastointiprojektia. Työalueen tarkoituksena on tukea ETL-prosessia (Extract-Transform-Load). Prosessissa tiedot ladataan ensin lähteistä työalueelle, jossa ne yhdenmukaistetaan ja yhdistellään omiin tauluihinsa, minkä jälkeen tiedot ladataan tietovarastoon. Lähdejärjestelmät ovat usein monimutkaisia, jolloin järjestelmän ylläpitäjän on parasta määrittellä, mitä tauluja lähteestä ladataan. (Hovi 2009, 25.)

2.3.2 Data vault

Data Vault on Dan Linstedtin kehittämä datanmallinnustapa, joka on suunniteltu erityisesti laajoille tietovarastointiprojekteille. Data Vault -menetelmä tukee hyvin tietovaraston laajentamista uusilla lähteillä, mikä mahdollistaa tietovaraston tekemisen pienemmissä palasissa valmiiksi, jolloin saadaan nopeammin dataa asiakkaan käyttöön. Menetelmässä myös datan historiointi on suoraviivaista ja selkeää. Data Vault perustuu kolmen perusmallinnuskäsitteen ympärille; näitä ovat hubit, linkit ja satelliitit (KUVIO 2). (Hyötyläinen 2014.)

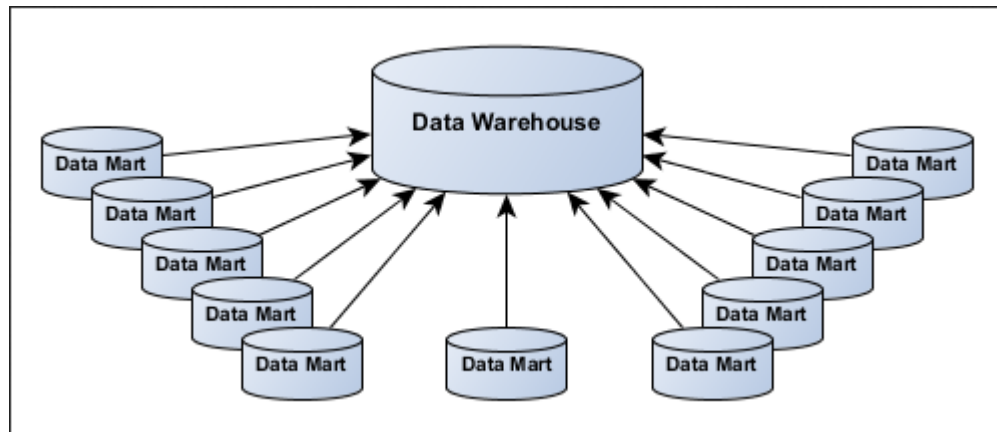
Hubit sisältävät yleensä esimerkiksi asiakkaan muuttumattomia liiketoiminnan avaimia, joita ovat esimerkiksi yrityksen tai asiakkaan tunnistenumerot tai tuotteen tuotekoodit. Linkeissä voi olla esimerkiksi tuotteiden myyntieriä tai asiakkaalle lähetettyjä laskuja. Satelliitit sisältävät hubien ja linkkien muut tiedot sekä erityisesti historiatiedon. (Hyötyläinen 2014.)



KUVIO 2. Data Vault –malli (Hyötyläinen 2014)

2.3.3 Data mart

Data mart eli suomeksi datamartti on tietovaraston osa, joka on suunniteltu BI-työkalujen käyttöön, erityisesti raporttien tekemiseen. Datamartit sisältävät usein vain yhden aihealueen datan. Esimerkiksi yksi datamartti sisältää tietoa yrityksen rahaliikenteestä ja toinen yrityksen henkilöstöstä. Datamartti voi olla joko yhden sovelluksen tiedoista kasattu erillinen datamartti, tai isosta keskitetystä tietovarastosta rakennettu datamartti (KUVIO 3). (Hovi. 2009, 24.)

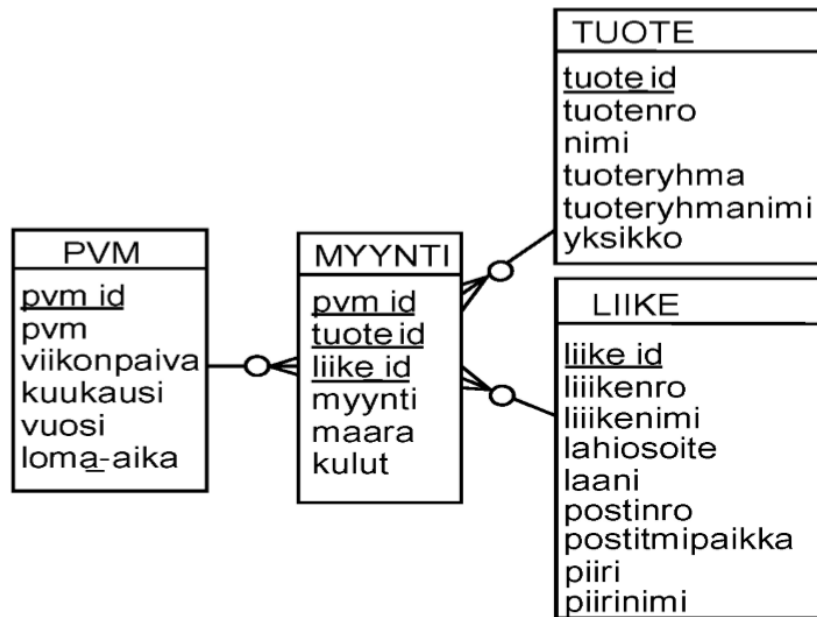


KUVIO 3. Tietovarasto ja sen datamartit (DataOnFucus 2015)

Datamartit ovat usein summa- tai tapahtumatasoisia ja ne suunnitellaan tähtimallin mukaiseksi. Toinen vaihtoehto datamartin mallintamiseen on lumihiihtalemallia. (Hovi 2009, 25.)

2.3.4 Tähti- ja lumihiihtalemalli

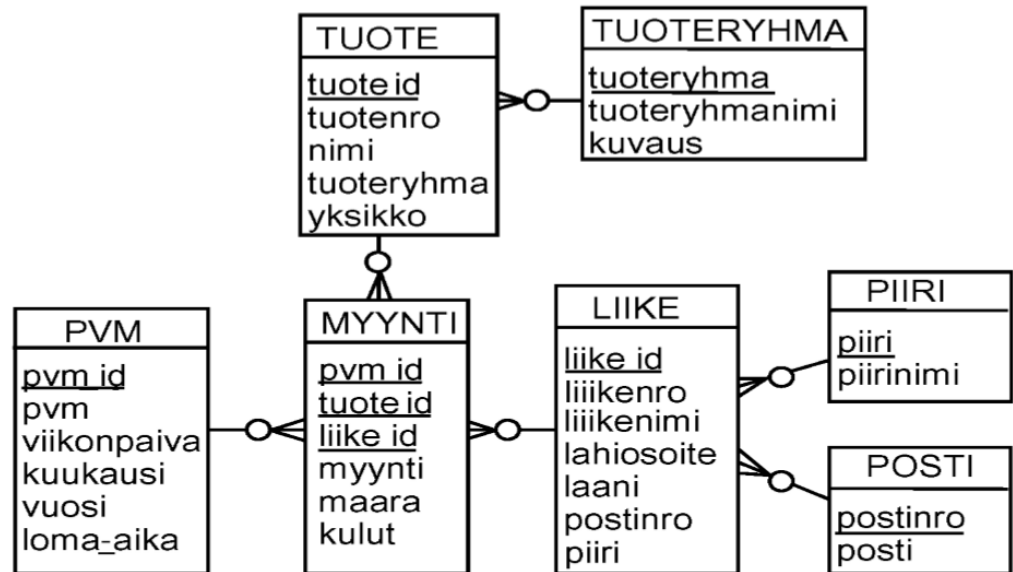
Tähtimalli (Star Schema) on Ralph Kimbalin suosittelu tapa mallintaa datamartteja. Tähtimalli sopii parhaiten moniulotteiselle ja numeeriselle tiedolle, kuten esimerkiksi myynnin seurantaan. Tähtimalli on hyvin selkeä ja käyttäjien on helppo ymmärtää sen toimintaa. Nimi tähtimalli tulee sen tähtimäisestä rakenteesta, sillä faktataulu on keskusta ja dimensiot ovat tähden sakarat (KUVIO 4). Tähtimallissa dimensiossa tietoa toistetaan, mikä tarkoittaa sitä, että hierarkiarakenne toistetaan kokonaisuudessaan joka rivillä. (Hovi 2009, 36.)



KUVIO 4. Tähtimalli (Hovi 2003 - 2005, 8)

Lumihiutalemallissa (snowflake) dimensioiden tietojen toistaminen poistetaan eli taulu normalisoidaan. Lumihiutalemallissa hierarkia hajotetaan omiin tauluihinsa, jolloin esimerkiksi kuvion 4 tuotetaulusta hajotetaan tuoteryhmä omaan tauluunsa (KUVIO 5).

Tähtimallia pidetään lumihiutalemallia parempana, sillä se on yleensä tehokkaampi ja helpommin ymmärrettävä. Molemmat mallinnustavat sisältävät yhden faktataulun sekä usean dimensiontaulun, joista kerrotaan seuraavassa luvussa. (Hovi 2009, 36 - 37.)



KUVIO 5. Lumihutalemalli (Hovi 2003 - 2005, 11)

2.3.5 Faktataulu

Faktataulu on dimensionaalisen mallinnuksen päätaulu, joka sisältää pääavaimen (primary key), joka on kooste kaikkien faktataulun viiteavaimista (foreign key), sekä viiteavaimen dimensiontauluihin. Faktataulussa on myös yrityksen numeerinen data, joita voi olla esimerkiksi tuotteiden myyntiin liittyvät luvut, kuten myyntierän koko sekä hinta. Faktataulussa voi myös olla teksti-arvoja, kuten selitteitä, mutta jos teksti-arvot eivät ole uniikkeja jokaisella rivillä, tulisi ne siirtää omaan dimensioonsa. (Kimball 2002, 19.)

2.3.6 Dimensiotaulut

Dimensiotaulut ovat usein faktatietoa kuvaavia tauluja, kuten esimerkiksi asiakastaulu tai tuotetaulu, jotka yhdessä kertovat, kuka on ostanut minkäkin tuotteen. Yhtenä dimensiona on usein myös päivämäärä-taulu, jonka avulla voidaan tarkastella vuosittaista tai kuukausittaista myyntiä. Päivämäärätaulu on myös hyvä esimerkki dimensioiden hierarkioista,

jossa hierarkia alkaa esimerkiksi vuositason ja päättyy päivätasolle. (Hovi 2009, 37.)

Dimensiontaulut ovat yleensä huomattavasti pienempiä kuin faktataulut, sillä ne ovat kooltaan alle 10 prosenttia faktojen ja dimensioiden yhteenlasketusta koosta. (Kimball 2002, 21.)

3 AZURE

Azure on Microsoftin julkaisema pilvessä toimiva sovellusalustapalvelu, joka tarjoaa laajalta alueelta erilaisia palveluita. Azuressa voi rakentaa, julkaista ja hallita projekteja, melkein mihin tahansa tarkoitukseen. Azure tarjoaa paljon mahdollisuuksia, oli sitten suuri yritys, joka tarvitsee palvelimen, joka kestää suuria työkuormia, tai pieni yritys, joka haluaa vain julkaista nettisivut. Azure tarjoaa alustan sovellusten rakentamiseen, jota yritykset voivat hyödyntää parantaakseen liiketoimintojaan. (Tulloch 2013, 1.)

If you ask, "What is Windows Azure?" the best answer might simply be this: Windows Azure can be anything you want it to be (Tulloch 2013, 1).

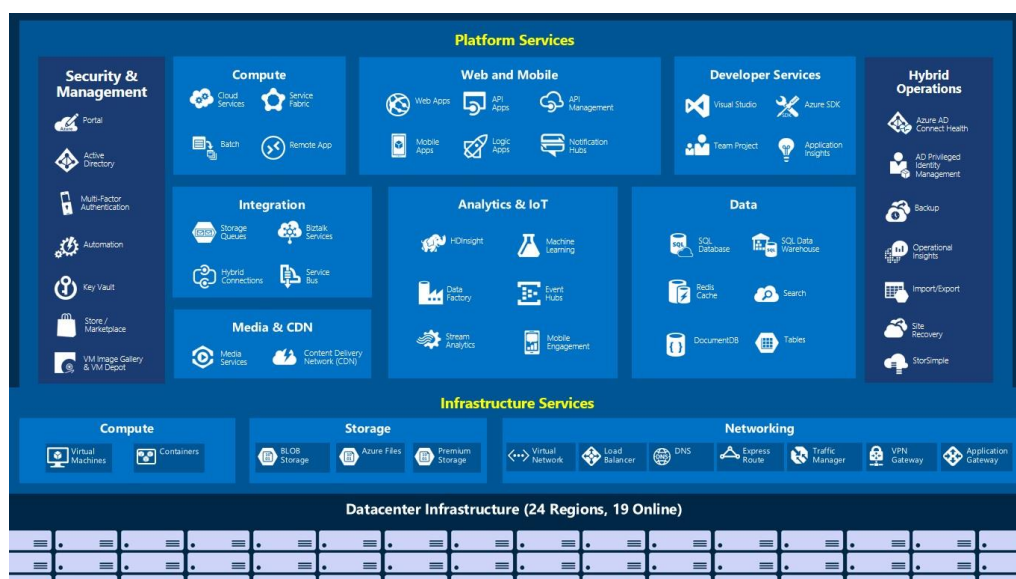
3.1 Azuren palvelut

Azuren palveluiden (KUVIO 6) määrä on suuri, ja ne voidaan jaotella esimerkiksi PaaS- (*Platform as A Service*) sekä IaaS (Infrastructure as a Service) -palveluihin (Forbes 2014).

Alustapalvelut (PaaS) tarjoavat mahdollisuuden rakentaa ja julkaista moderneja applikaatiota hyödyntäen pilveä. Näihin kuuluvat muun muassa www- ja mobiilisovellukset. Azuren alustapalvelut mahdollistavat sovelluksien julkaisemisen ilman palvelimen ja muun arkkitehtuurin ostamista ja ylläpitämistä. Tällöin sovellukset saadaan nopeasti julkaistua ja voidaan keskittyä vain sovelluksen kehittämiseen. (Forbes 2014.)

IaaS-palvelut tarjoavat virtualisointipalveluita niin Windows kuin Linux ympäristöihin, sekä erilaisia tiedon tallennus ratkaisuja. IaaS-ratkaisut pohjautuvat usein virtualisointi-tekniikoihin, kuten virtuaalikoneen luomiseen ja tietokannan asentaminen sinne, joita voidaan sitten hallinnoida yrityksen omasta infrastruktuurista. (Forbes 2014.)

Seuraavaksi esitellään muutamia Azuren palveluita: Event Hubs ja Stream Analytics, jotka ovat osa Azuren sovelluspalveluita, sekä datapalveluihin kuuluvaa Azure SQL Database.

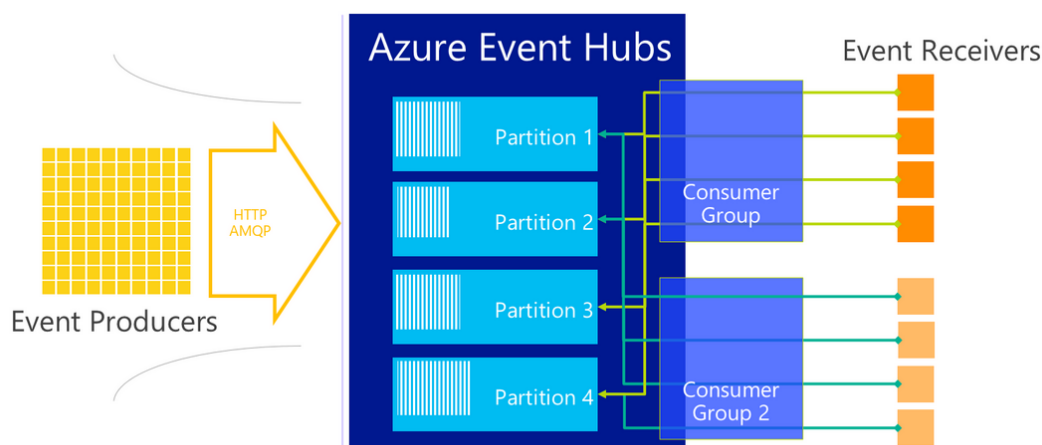


KUVIO 6. Azure palvelut (Van Den Berg 2016)

3.2 Event Hubs

Event Hubs on Azuren palvelu, joka on hyvin skaalautuva datankokoamispalvelu, joka pystyy vastaanottamaan miljoonia viestejä sekunnissa. Event Hubsia käytetään silloin, kun dataa tulee paljon eri lähteistä. Tämänkaltaisia tilanteita ovat esimerkiksi anturidatan tai sovelluksen käyttödatan kerääminen. Event Hubs onkin kuin etuovi datan kululle: kun data on ensin kerätty Event Hubsiin, niin sen jälkeen data voidaan analysoida ja säilöä muualle. (Microsoft 2017a.)

Event Hubsin alle luodaan yksittäisiä Event Hubeja, joista tieto liikkuu lähettäjältä (Publisher) Event Hubiin ja sieltä vastaanottajalle (Consumer) kuten kuviossa 7. Event Hubin sisällä on osioita (partition), jotka ovat lohkoja, joihin data säilötään. Tiedon siirtoon Event Hub käyttää AMQP- sekä HTTPS-protokollaa. AMQP-protokolla (The Advanced Message Queuing Protocol) on suunniteltu erityisesti yritysten tietojen siirtämiseen. (Microsoft 2017a.)



KUVIO 7. Event Hubs (Microsoft 2017a)

3.2.1 Lähettäminen

Viestejä lähetetään joko käyttäen HTTPS- tai AMQP 1.0 -tekniikkaa.

Viestien lähettäjät ja vastaanottajat käyttävät SAS-tunnistetta (Shared Access Signature) tunnistaakseen itsensä Event Hubiin. Tunniste voi olla joko yleinen, tai jokaisella lähettäjällä voi olla oma uniikki tunniste.

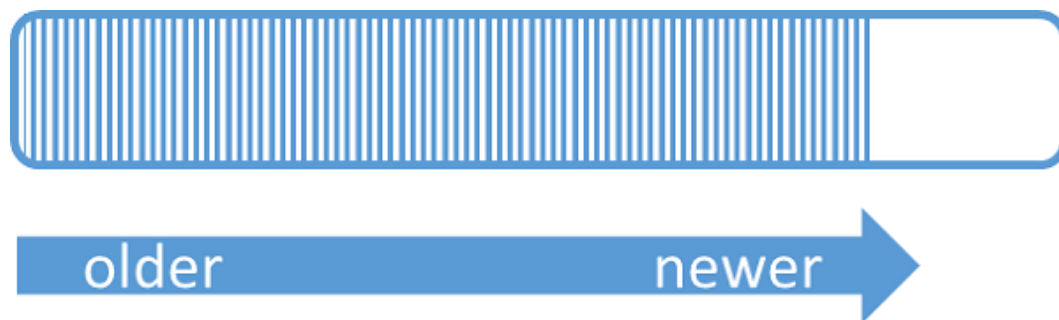
Jokaiselle tunnisteelle voidaan antaa erilaisia oikeuksia, joita ovat hallinta, lähettäminen sekä vastaanottaminen. (Microsoft 2017a.)

Viestien julkaisemiseen .NET-ympäristössä on tarjolla Microsoftin tekemä Azure Service Bus -luokka, joka tarjoaa kaiken tarvittavan viestien lähettämiseen. Muissa ympäristöissä voi käyttää mitä tahansa AMQP 1.0 clienttiä, esimerkiksi Apache Qpidiä. (Microsoft 2017a.)

3.2.2 Osiot

Osiolla tarkoitetaan Event Hubissa olevia lohkoja, jotka ovat järjestettyjä viestijonoja, joihin saapuvat viestit menevät aina jonon perälle (KUVIO 8).

Osiin saapuvat viestit säilyvät yhdestä seitsemään päivään, eikä niitä voi poistaa erikseen. (Microsoft 2017a.)



KUVIO 8. Event Hub osio (Microsoft 2017a)

Event Hubia luodessa käyttäjän pitää valita osioiden määrä, ja se voi olla 2 ja 32 välillä, mutta oletuksena tämä määrä on 4. Osioiden määrää ei voi muuttaa jälkikäteen muuten kuin ottamalla yhteyttä palvelun tarjoajaan. (Microsoft 2017.)

Osiot ovat datan organisointi mekanismi, joiden määrän määrää se, kuinka paljon Event Hubista ladataan dataa, eikä se kuinka paljon sitä sinne lähetetään. Jokaisella osiolla voi olla vain yksi vastaanottaja kerrallaan. (Microsoft 2017a.)

3.2.3 Vastaanottaminen

Jokainen, joka lukee viestejä Event Hubista, on viestien vastaanottaja. Kun viestejä halutaan lukea, vastaanottaja avaa AMPQ 1.0 session, jolloin viestit lähetetään session läpi heti, kun ne ovat saatavilla. (Microsoft 2017a.)

Osioiden ja vastaanottajan välissä on ryhmiä (KUVIO 7), joihin vastaanottajat kuuluvat ja ryhmän asetuksista voidaan päättää mistä osiosta kyseinen ryhmä pystyy vastaanottamaan viestejä. Event Hubissa on aina oletuksena yksi ryhmä, jota voidaan käyttää, tai jokaiselle vastaanottajalle voidaan tehdä oma ryhmä. Jokaisella osiolla voi olla vain yksi aktiivinen vastaanottaja yhdestä ryhmästä, jolloin muut joutuvat odottamaan vuoroaan. (Microsoft 2017a.)

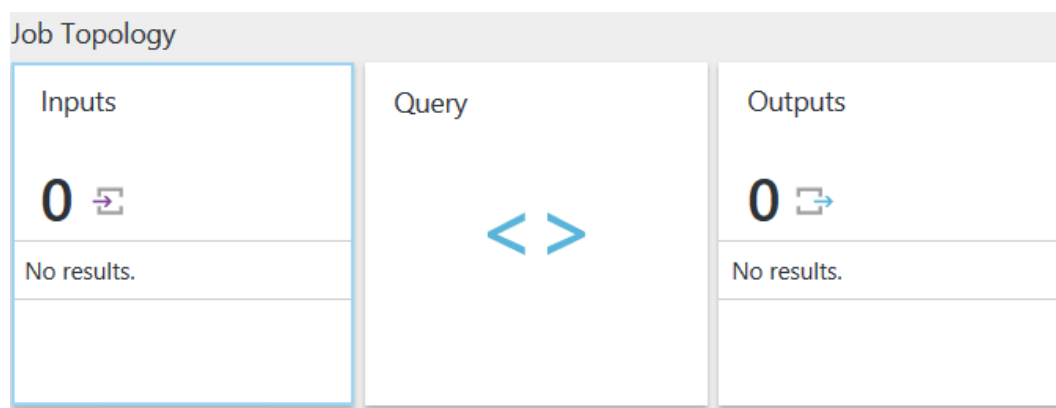
3.3 Stream Analytics

Stream Analytics on Azuren palvelu, joka on täysin hallittava reaaliaikainen datan prosessointi moottori. Stream Analytics mahdollistaa helposti asennettavan reaaliaikaisen ja analyttisen laskennan erilaisista lähteistä, kuten laitteista, sensoreista, nettisivuilta ja sovelluksista. (Microsoft 2017b.)

3.4 Datan liikkuminen

Stream Analytics koostuu kolmesta komponentista (KUVIO 9): lähteestä (Inputs), kyselystä (Query) ja määränpäästä (Outputs). Ensimmäisenä valitaan datan lähde. Lähteinä Stream Analytics tarjoaa tällä hetkellä kolme eri vaihtoehtoa, Event Hub, Blob Storage sekä IoT Hub, joiden tuettuja dataformaatteja ovat CSV, JSON ja Avro. (Microsoft 2017b.)

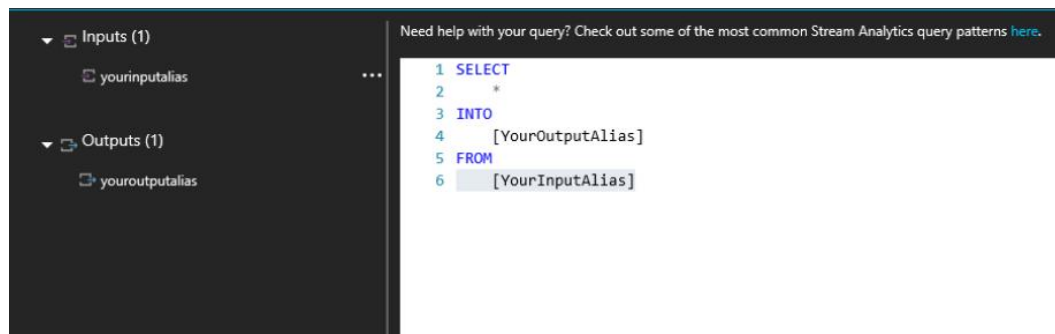
Lähteen valitsemisen jälkeen valitaan määränpää eli se, mihin data viedään. Määränpäinä Azure tarjoaa tällä hetkellä muun muassa SQL Databasen, Event Hubin, Power BI:in ja DocumentDB-tietokannan. (Microsoft 2017b.)



KUVIO 9. Stream Analytics -komponentit

Kun lähde ja määränpää on määriteltä, voidaan dataa hakea ja viedä haluttuun paikkaan. Kysely-osiossa (KUVIO 10), lähdedataa käsitellään T-SQL tapaisilla kyselyillä, joissa kerrotaan mistä lähteestä dataa haetaan ja mihin se viedään. Kyselyssä pystytään valmiiksi muokkaamaan ja

analysoimaan lähdedataa ennen sen viemistä määränpäähänsä. Esimerkiksi sarakkeiden tietotyyppiä voidaan määrittää, sekä summata myyntilukuja ja yhdistellä lähteitä keskenään. (Microsoft 2017b.)



KUVIO 10. Stream Analytics query

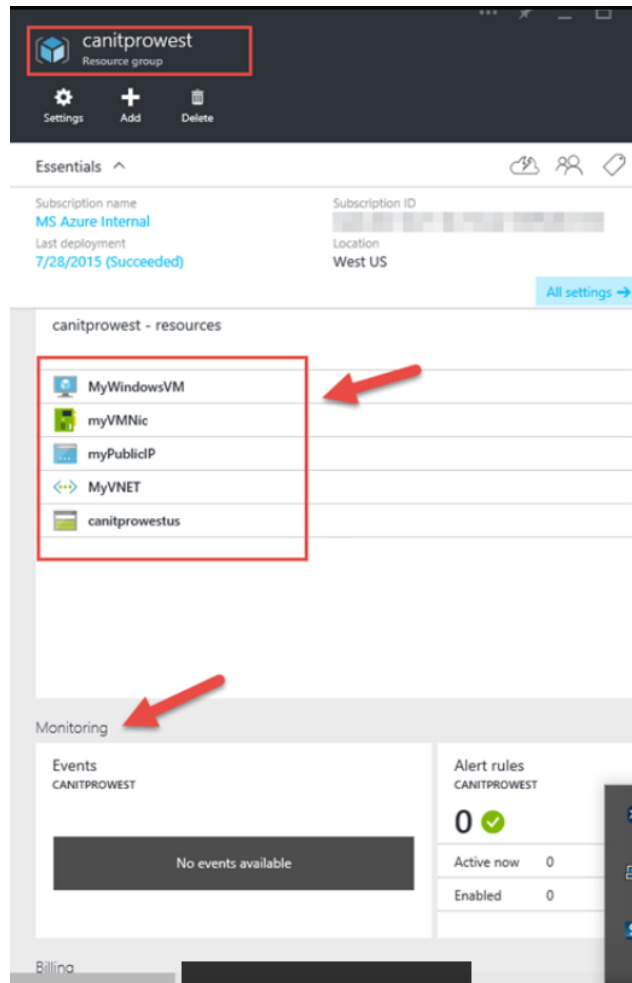
3.5 Azure SQL Database

SQL Database on relaatioihin pohjautuvat Azuren tietokantapalvelu, joka pystyy hallitsemaan suuriakin tietomääriä. SQL Database pohjautuu Microsoftin SQL Serverin moottoriin ja tukee olemassa olevia SQL Server työkaluja, kirjastoja ja ohjelmointirajapintoja. (Microsoft 2017c.)

Azure tarjoaa kaksi erilaista mallia SQL-tietokannan asentamiseen, joita ovat single database tai elastic pools. Single databasea käytetään silloin kun tietokannannan suorituskyky tarpeet tiedetään etukäteen. Elastic pools on taas monen tietokannan "allas", jonka alle voi luoda useita tietokantoja. Elastic pools osaa automaattisesti säädellä kunkin tietokannan suorituskykyä tietokannan sen hetkisten tarpeiden mukaan. Molemmille malleille voidaan valita Basic, Standard ja Premium palvelun väliltä, jotka määräävät sen, kuinka paljon tietokannalla on suorituskykyä ja tilaa datan säilömiseen. (Microsoft 2017c.)

3.6 Resurssiryhmä

Azuren resurssiryhmä (Resource Group) on tarkoitettu luotujen palveluiden ja sovelluksien ryhmittelyyn (KUVIO 11). Ryhmään lisätään kaikki samaan sovellukseen kuuluvat osat, jolloin sovellus kokonaisuutta voidaan helposti hallinnoida, ylläpitää sekä julkaista yhtenä pakettina. (Foulds 2015.)



KUVIO 11. Resurssi ryhmä (Roman 2015)

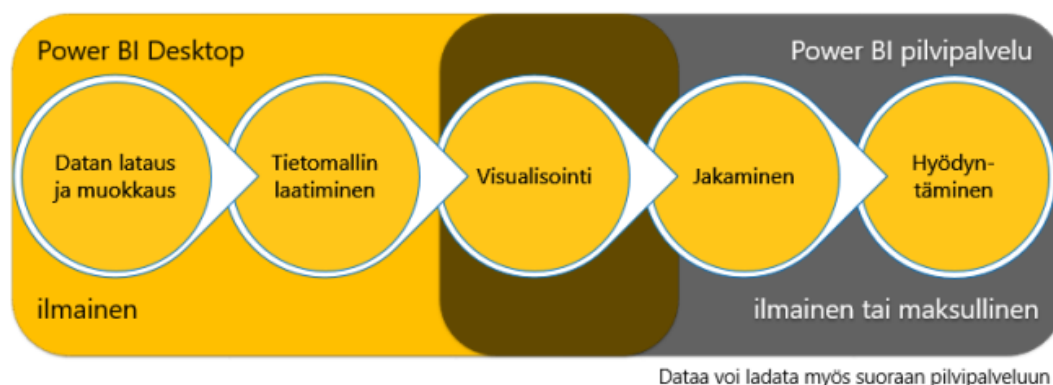
4 POWER BI

Power BI on Microsoftin vuonna 2015 julkaisema raportointi- ja analysointipalvelu. Power BI on joukko ohjelmia ja palveluita, jotka mahdollistavat pääsyn dataan, sen visualisointiin ja jakamiseen muille käyttäjille. Power BI:hin kuuluu kolme tuotetta, joita ovat: Power BI Services, Power BI Mobile ja Power BI Desktop. (Lachev 2016, 3.)

Power BI Service on pilvessä toimiva (powerbi.com) business analytics-palvelu, minne data ja raportit viedään jaettavaksi muille käyttäjille. Power BI Mobile on natiivi puhelinsovellus, joka toimii iOS, Android ja Windows laitteilla ja sillä voi katsoa raportteja, jotka on toimitettu pilveen. Power BI Desktop on taas tarkoitettu datan mallintamiseen ja raporttien luomiseen. Se sisältää Excelistä tutut Power Pivot tyyppiset itsepalvelu BI-toiminnallisuudet. (Lachev 2016, 3.)

4.1 Raportoinnin vaiheet

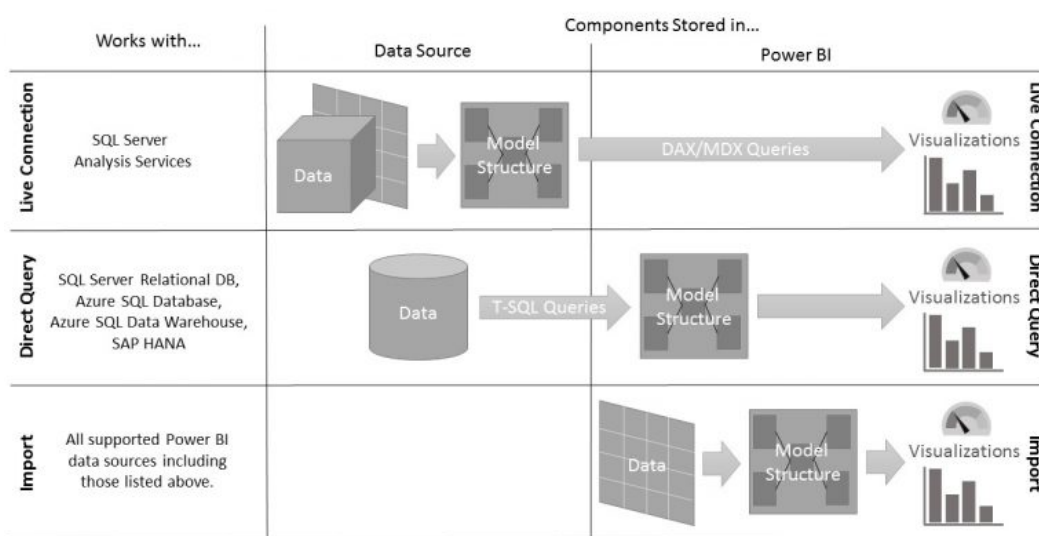
Raporttien tekeminen alkaa datan lataamisella ja muokkaamisella, jonka jälkeen se voidaan mallintaa. Mallintamisen jälkeen, data visualisoidaan raporteiksi, jonka jälkeen raportit voidaan julkaista pilveen muiden nähtäviksi ja jossa muut voivat tehdä omia visualisointeja datasta. Kuviossa 12 esitellään raportoinnin vaiheet. (Enho 2016.)



KUVIO 12. raportoinnin vaiheet. (Enho 2016)

4.2 Lataaminen

Power BI tarjoaa kolme tapaa lähdedatan lataamiselle (KUVIO 13). Näitä ovat: datan tuonti (data import), sekä kaksi tapaa reaaliaikaisille latauksille, joita ovat suorat kyselyt lähteestä (direct query), sekä reaaliaikainen yhteys SQL Serverin Analysis Service lähteille (live connection). Datan lataustapaan vaikuttaa lähteiden määrä, sekä itse lähde, sillä kaikille lähteille ei tueta reaaliaikaista lataamista. (Lachev 2016, 96.)



KUVIO 13. Power BI yhteydet (Larson 2016)

4.2.1 Datan tuonti

Silloin kun lähteitä on useita, tai Power BI ei tue lähteellä reaaliaikaista yhteyttä, niin käytetään datan tuonti menetelmää. Kun datat tuodaan lähteistä, ne tallennetaan ensin pakattuna ohjelman muistiin, jolloin data täytyy päivittää manuaalisesti tai ajastetusti aina, kun halutaan uusimmat tiedot lähteistä. Tämän jälkeen data voidaan muokata, luoda mittareita ja sarakkeita, sekä luoda liitoksia taulujen välille. (Lachev 2016, 96.)

4.2.2 Suora kysely

Kun lähteitä on vain yksi, voidaan data tuoda reaaliaikaisesti suorilla kyselyillä lähteestä. Tällöin valitaan lähteestä halutut taulut, jolloin ohjelma

luo kyselyn lähteen natiivilla kielellä ja kyselee tiedot suoraan lähteestä. Esimerkiksi, jos lähde on SQL Server, niin ohjelma luo T-SQL-kyselyn suoraan lähteestä. Tämän jälkeen Data mallinnetaan ja muokataan raportointia varten. (Lachev 2016, 96.)

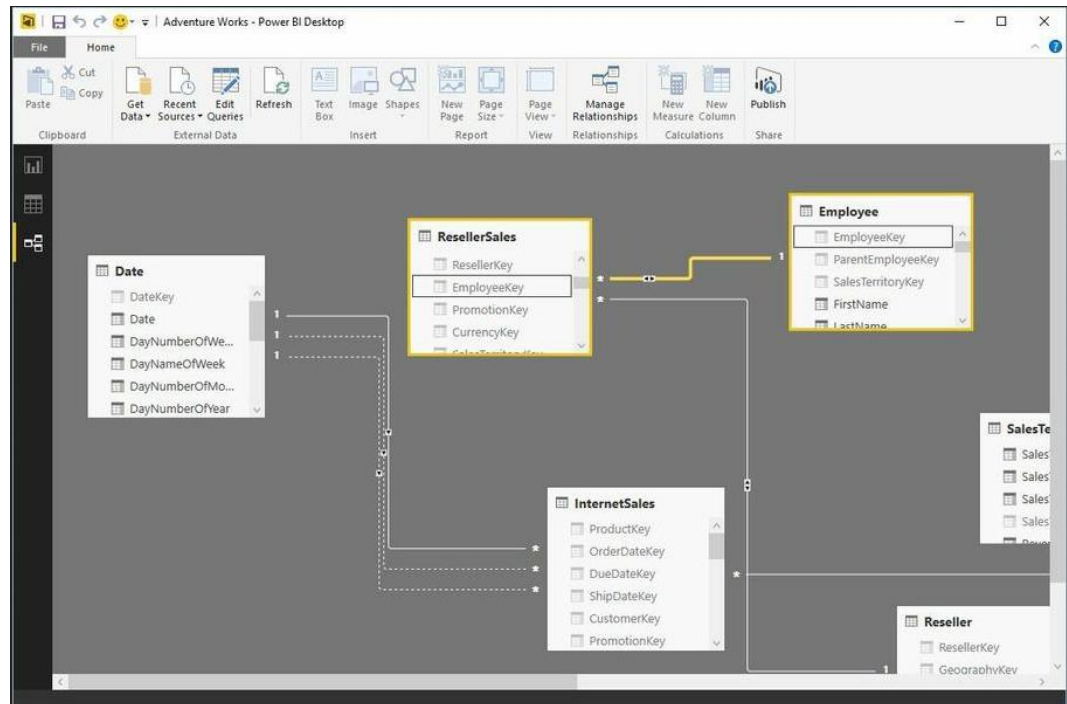
Suoria kyselyitä tehdessä dataa ei tallenneta ohjelman muistiin, vaan se ladataan aina uudestaan, kun raporttia käytetään. Suorien kyselyiden huonona puolena on, että mikäli lähde sisältää paljon dataa, niin raportin latautuminen voi olla hidasta. (Lachev 2016, 96 - 97.)

4.2.3 Analysis Service

Power BI:in kolmas vaihtoehto datan hakemiselle on reaaliaikainen yhteys Microsoftin Analysis Servicen Multidimensional- tai Tabular-kuutio. Suorien kyselyiden tavoin, dataa ei tallenneta ohjelman muistiin, vaan ne haetaan suoraan lähteestä. Kun data haetaan Analysis Servicestä ei dataa tarvitse myöskään mallintaa, vaan data-malli, laskennalliset sarakkeet, sekä mittarit tulevat lähteessä itsessään. (Lachev 2016, 97.)

4.3 Mallintamien

Taulujen yhdistäminen toisiinsa tehdään sille tarkoitettussa relaatio-ikkunassa (KUVIO 14). Ikkunasta on helppo ymmärtää, kuinka taulut ovat liitoksissa keskenään. Taulujen väliset liitokset esitetään viivoilla taulujen välillä ja viivojen päissä olevat symbolit kertovat liitoksen muodon. Numero yksi tarkoittaa yhden suhde -liitosta ja tähti monen suhde -liitosta, jolloin esimerkiksi kuvion 14, työntekijä-taulu on yhden suhde moneen -liitoksessa jälleenmyynti-tauluun. (Lachev 2016, 157 - 158.)

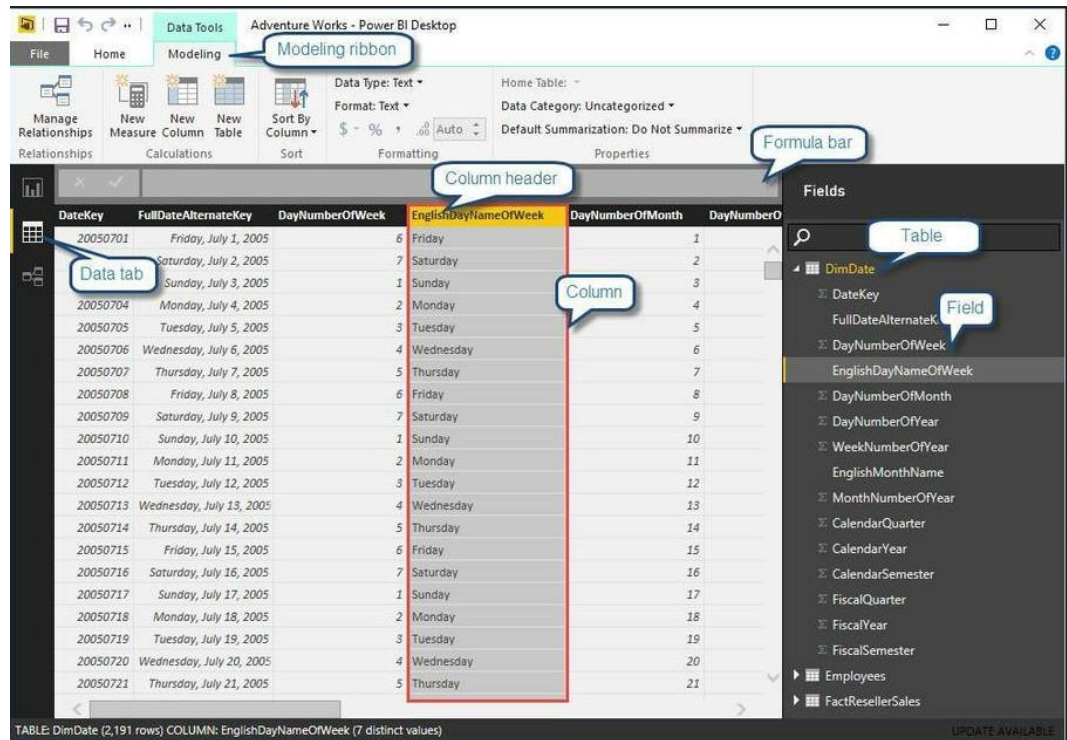


KUVIO 14. Relaatio ikkuna (Lachev 2016, 157)

Taulujen yhdistämiseen toisiinsa voidaan tehdä kahdella tavalla, joko käyttämällä automaattista liitosten tunnistusta, jolloin ohjelma käy taulut läpi ja etsii liitosavaimia taulujen välillä. Toinen tapa taulujen yhdistämiseen on käyttää siihen tarkoitettua ikkunaa, jossa valitaan taulut ja taulujen liitosavaimet sekä taulujen välinen suhde. (Lachev 2016, 158 - 159.)

4.4 Muokkaaminen

Datan muokkaaminen tehdään sille tarkoitetussa data-ikkunassa (KUVIO 15), jossa sarakkeen nimeä ja tietotyyppiä voidaan muokata, jolloin esimerkiksi numeeristen-sarakkeiden tietotyyppi vaihdetaan numeroksi. Sarakkeita voidaan myös piilottaa loppukäyttäjältä, tällöisiä sarakkeita ovat yleensä taulujen liitosavaimet. Datasta voidaan myös tehdä erilaisia laskennallisia sarakkeita, sekä mittareita, joiden luomiseen käytetään DAX-kieltä (Data Analysis Expression). (Lachev 2016, 162.)



KUVIO 15. Data-ikkuna (Lachev 2016, 137)

4.5 DAX

DAX on kaavoihin pohjautuva kieli, jolla pystytään tekemään laskennallisia sarakkeita, sekä mittareita datasta. DAX on hyvin saman kaltainen kieli kuin, mitä Excelissä käytetään ja molemmista löytyy samoja funktiota kuten SUM ja AVARAGE. DAX on suunniteltu data-malleille, joissa käytetään relaatiota, jolloin voidaan tehdä laskentaa käyttäen usean taulun tietoja samanaikaisesti. (Lachev 2016, 162.)

Laskennalliset sarakkeet ovat taulun sarakkeita, joilla voidaan yhdistää ja laskea sarakkeita keskenään, kuten kuviossa 16, yhdistetään etu- ja sukunimi omaan sarakkeeseensa. Laskennallisia sarakkeita käytetään silloin, kun raportilla tarvitsee yhdistellä sarakkeita, tai saraketta halutaan käyttää apuna muussa laskennassa. (Lachev 2016, 162-163.)

Formula bar: FullName = [FirstName] & " " & [LastName]

Title	FirstName	MiddleName	LastName	FullName
	Larry		Gill	Larry Gill
	Geoffrey		Gonzalez	Geoffrey Gonzalez
	Blake			Blake Williams
	Alexa			Alexa Johnson
	Jacquelyn			Jacquelyn Dominguez
	Casey		Gutierrez	Casey Gutierrez
	Colleen		Lu	Colleen Lu
	Jeremiah		Stewart	Jeremiah Stewart

Row Context evaluated for each row

KUVIO 16. Laskennallinen sarake (Lachev 2016, 163)

Mittareita käytetään silloin, kun halutaan saada kokonaissumma myynnistä, tai halutaan saada tuotteita tilanneiden asiakkaiden kokonaismäärä. Toisin kuin laskennallisissa sarakeissa mittareiden arvoja ei näytetä taulun riveillä, vaan arvot näkyvät vasta raportoinnissa. Tällöin arvot jakaantuvat omille soluilleen rajauksien mukaan kuten kuviossa 17, esitetään pyörien vuosittaista myyntiä maittain. Tyypillisimpiä funktioita mittareiden luomiseen ovat muun muassa SUM, COUNT ja AVERAGE. (Lachev 2016, 164 - 165.)

SalesTerritory	Country	2007	2008	Total
Australia			\$43,174.77	\$1,323,820.73
Canada			\$9,709.62	\$4,370,334.95
France		\$1,341,320.07	\$1,111,858.69	\$2,453,178.76
Germany		\$820,513.65	\$722,502.00	\$1,543,015.65
United Kingdom		\$1,230,915.70	\$1,060,662.55	\$2,291,578.25
United States		\$8,933,163.90	\$7,951,335.55	\$16,884,499.45
Total		\$15,467,184.61	\$13,399,243.18	\$28,866,427.79

ProductCategory: is Bikes

- (All) 1
- Accessories 22
- Bikes 60
- Clothing 20
- Components 94

KUVIO 17. Mittarin käyttö (Lachev 2016, 165)

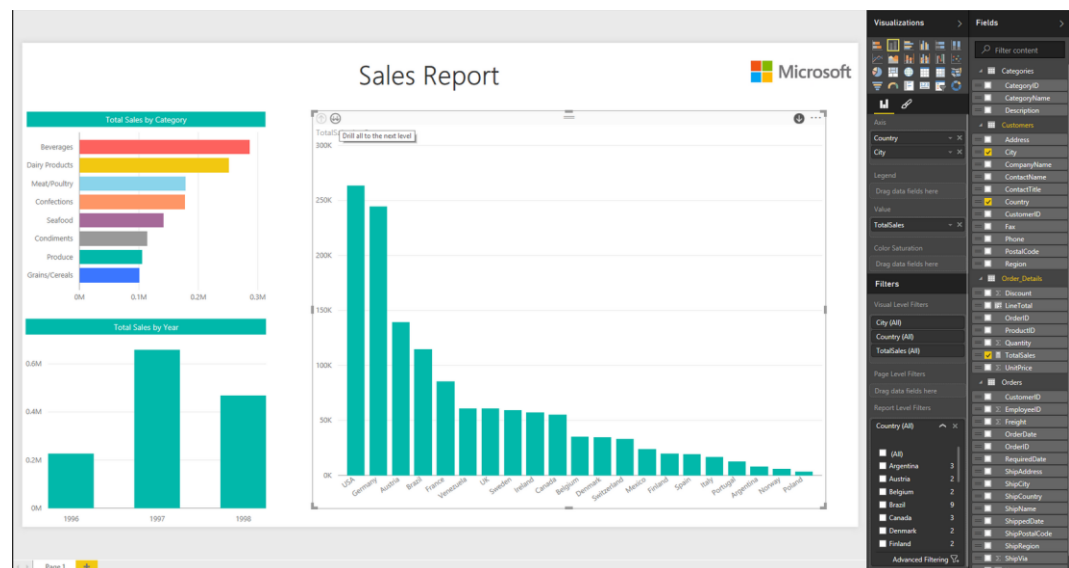
4.6 Visualisointi

Raportit luodaan raportointi-ikkunassa (KUVIO 18), josta löytyy raportointi näkymä, komponentit, joilla dataa voidaan visualisoida, sekä taulut ja niiden sarakkeet ja mittarit. Raportointi näkymä koostuu usein monista eri

datan visualisointi komponenteista, kuten pylväs- ja piirakkakaavioista, sekä perinteisistä taulukoista. (Enho 2016.)

Raportilla olevat komponentit ovat vuorovaikutuksissa toisiinsa. Vuorovaikutuksella tarkoitetaan sitä, kuinka kaikki raportin komponentit ovat yhteydessä toisiinsa. Esimerkiksi voidaan aktivoida pylväs pylväskaaviosta, joka esittää myyjien myyntilukuja. Pylvään aktivoinnin jälkeen kaikki muut komponentit päivittyvät näyttämään vain valitun myyjän lukuja, jolloin esimerkiksi kartalla näkyy, missä päin maailmaa kyseinen myyjä on tehnyt kauppvoja. (Enho 2016.)

Datasta voidaan myös tehdä hierarkioita, kuten laittamalla kalenteri-taulun vuosi- ja kuukausi-sarakkeet samaan hierarkiaan. Tämän jälkeen raportilla voidaan porautua vuositasolta kuukausitasolle. Erilaisia filttareita käytetään myös raporteilla, kuten aika-filtteri. Kun aika-filtterin aikajaksoa kavennetaan, niin silloin myös muiden komponenttien tiedot muuttuvat näyttämään vain kyseisen ajanjakson tietoja. (Lachev 2016, 50, 58.)



KUVIO 18. Raportointi-ikkuna (Llopis 2015)

4.7 Power BI Service

Power BI Service on pilvessä toimiva business analytics -palvelu, joka on tarkoitettu tiedon visuaaliseen analysointiin. Power BI Serviceen voidaan tuoda myös dataa suoraan lähteistä, mutta jos lähdedataa tarvitsee esikäsitellä, kuten luoda liitoksia lähteiden välille, niin silloin joudutaan mallintaminen tekemään ensin Power BI Desktop -ohjelmalla. (Lachev 2016, 10 - 11.)

Power BI Servicessä raportointi voidaan jakaa kahteen alueeseen, joita ovat raportointi sivut, sekä kojelauta (Dashboard). Raportointi sivut sisältävät yleensä vain yhden aihealueen tietoja, jolloin yhdellä sivulla esitellään myynnin lukuja ja toisella henkilöstön tietoja. Kojelaudalle poimitaan raportointi sivuilta vain tärkeimmät komponentit, jolloin kojelaudalta nähdään kokonaiskuva yrityksen koko tilanteesta. (Lachev 2016, 11.)

Power BI Servicellä luodut raportit ja kojelaudat voidaan jakaa muille käyttäjille ja antaa käyttäjille erilaisia oikeuksia. Käyttäjälle voidaan antaa joko vain lukuoikeudet, tai oikeus päästä luomaan tiedoista omia raportteja. Oikeuksia voidaan myös antaa ryhmille jolloin esimerkiksi myynnin-henkilöstö saa oikeudet ainoastaan myyntiin liittyviin tietoihin. (Lachev 2016, 12 - 13.)

5 VALVONTASOVELUS

Valvontasovellus tehdään helpottamaan tietovaraston päivittäisten latausten seuraamista, jolloin työntekijöiden ei tarvitse käydä fyysisesti katsomassa, onko jokin lataus epäonnistunut. Latauksien seuranta mahdollistaa myös tietovarastointi projektin yleisen seurannan, jolloin voidaan havainnoida, missä vaiheessa projektia tulee eniten virheitä.

Opinnäytetyössä ei käydä läpi, sitä kuinka latauksien tiedot kerätään tietovarastosta vaan, kuinka ne siirretään yhteiseen paikkaan säilöttäväksi ja sieltä raportille. Azure valittiin tietojen säilöntään, sillä se tarjoaa hyviä valmiita palveluita, joiden avulla latauksien tiedot saadaan turvallisesti siirrettyä ja säilöttyä.

Huonona puolena toteutuksessa on, että datan siirtäminen Azureen tarvitaan internet-yhteys, mutta tämä ongelma voidaan ratkaista sillä, että latauksista kerätty data tallennetaan tiedostoon ja tiedosto siirretään ympäristöön, josta on pääsy internetiin. Tällöin tietovaraston reaaliaikainen valvonta ei onnistu, vaan dataa voidaan käyttää vain tietovaraston latausten yleiseen seuraamiseen.

Tavoitteena on luoda Power BI -raportti, josta nähdään sen hetkinen latauksien tilanne, jolloin voidaan reagoida nopeasti epäonnistuneisiin latauksiin. Tämän lisäksi halutaan nähdä latauksien historiatietoa, kuten latauksien keston muuttuminen ajan kuluessa. Toteutus on suunniteltu toimimaan Windows-ympäristössä ja tukee tällä hetkellä vain Microsoftin SQL Serveriä.

5.1 Valvonta

Tietovarastossa on yleensä paljon lähteitä, joista tiedot ladataan päivittäin tietovarastoon ja joskus saattaa olla tilanteita, että lataukset epäonnistuvat. Epäonnistumisen saattaa aiheuttaa useat syyt. Esimerkiksi lähdejärjestelmään on mennyt virheellistä tietoa, tietovaraston jokin komponentti on toteutettu väärin tai palvelin, jossa tietovarasto sijaitsee, kaatuu äkillisesti. Latauksien epäonnistuessa, on tärkeää, että

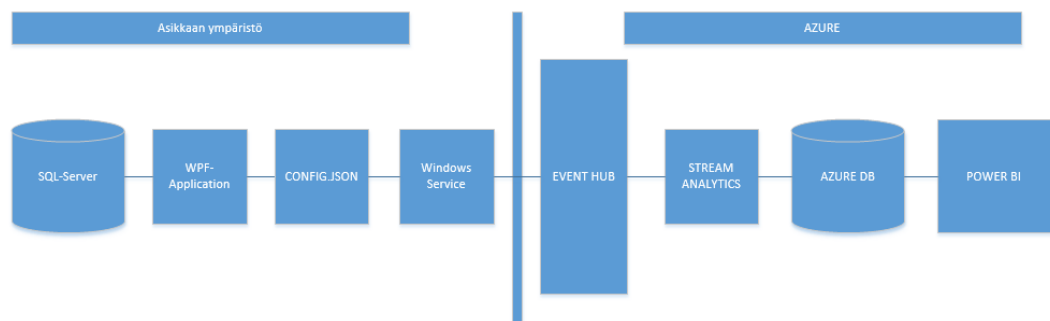
epäonnistuminen huomataan mahdollisimman nopeasti. Tällöin vikaa päästään tutkimaan ja korjaamaan, jotta tiedot saadaan taas raporteille asiakkaan käyttöön.

5.2 Arkkitehtuuri

Toteutuksen arkkitehtuuri (KUVIO 19) voidaan jakaa kolmeen osioon, asiakkaan ympäristössä oleviin sovelluksiin, Azuren palveluihin sekä Power BI -raporttiin. Asiakkaan ympäristössä olevia sovelluksia ei käydä läpi tässä työssä.

Latauksien tiedot kerätään SQL-Serveristä. Tietojen keräämistä varten luotiin WPF-sovellus, joka luo tiedoston, jossa määritellään, minkä tietokannan latauksia halutaan seurata. Windows Service lukee ensin tiedoston, jonka jälkeen ohjelma hakee tietokannasta latauksien tiedot ja lähettää ne Event Hubiin.

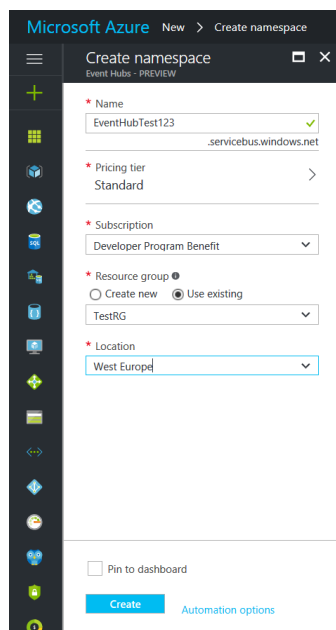
Kun latauksien tiedot ovat saapuneet Event Hubiin, Stream Analytics siirtää tiedot Azure DB -tietokantaan varastoitavaksi. Tämän jälkeen latauksien tiedot haetaan tietokannasta ja esitetään Power BI -raportilla.



KUVIO 19. Arkkitehtuurikuvaus

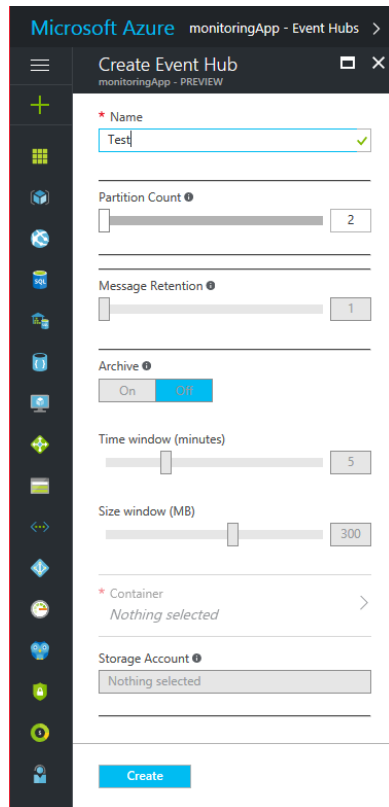
5.3 Event Hubin luominen

Jotta dataa voidaan lähettää Event Hubiin, täytyy ensin luoda Azureen Event Hubs -nimiavaruus, jonka alle luodaan Event Hub. Event Hubs -nimiavaruus luodaan (Kuvio 20) antamalla sille ensin uniikki nimi (koko Azuressa). Tämän jälkeen valitaan, mitä tilausta käytetään, sekä resurssiryhmä ja Azuren palvelimen sijainti (location). Palvelimen sijainti kannattaa valita tarkkaan, sillä datan siirtäminen Azuren komponenttien välillä on maksutonta, jos ne sijaitsevat samalla palvelimella ja kaikkia Azuren palveluja ei ole mahdollista asentaa kaikille palvelimille.



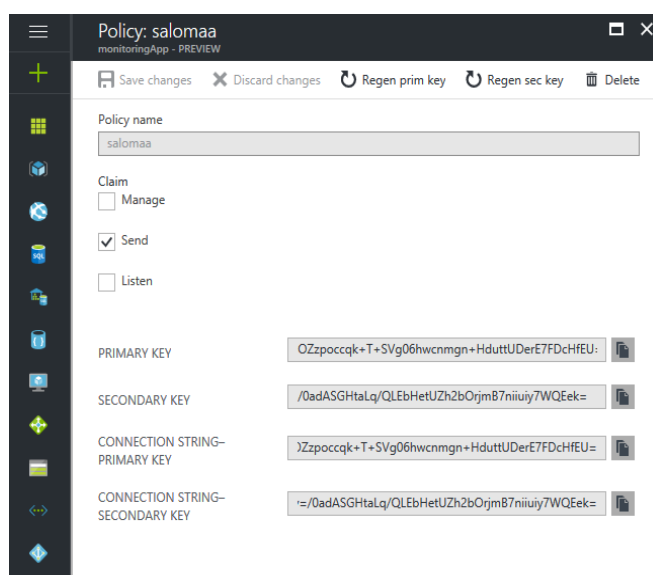
Kuvio 20. Event Hubs -nimiavaruuden luominen

Event Hubs -nimiavaruuden alle luodaan Event Hubin (Kuvio 21), johon data lähetetään. Event Hubille annetaan nimi, jonka täytyy olla uniikki Event Hubs -nimiavaruuden alla. Lisäksi valitaan osioiden lukumäärä, aika kuinka kauan dataa säilötään (Message retention), sekä halutaanko data arkistoida (Archive).



Kuvio 21. Event Hubin luomien

Seuraavaksi luodaan oikeudet, joilla määritellään ketkä voivat hallita, lähettää sekä lukea viestäjä Event Hubista (Kuvio 22). Kun oikeudet on luotu, saadaan osoite (connection string) sekä SAS-tunniste, joilla voidaan lähettää viestejä Event Hubiin.



KUVIO 22. Shared access policy

5.4 Datan siirtäminen Event Hubiin

Datan lähettämiseen Azuren Event Hubiin .NET:illa Microsoft on luonut WindowsAzure.ServiceBus -kirjaston, joka sisältää valmiita metodeja datan lähettämiseen. Kuviossa 23 esitellään, kuinka lähetys voidaan toteuttaa. Rivillä kolme annetaan parametreina Event Hubs -nimiavaruuden osoite ja SAS-tunniste (KUVIO 15), sekä Event Hubin nimi. Komennolla `eventHubClient.Send()`, viesti lähetetään Event Hubiin.

```
static void SendingRandomMessages()
{
    var eventHubClient = EventHubClient.CreateFromConnectionString(connectionString, eventHubName);
    while (true)
    {
        try
        {
            var message = Guid.NewGuid().ToString();
            Console.WriteLine("{0} > Sending message: {1}", DateTime.Now, message);
            eventHubClient.Send(new EventData(Encoding.UTF8.GetBytes(message)));
        }
        catch (Exception exception)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("{0} > Exception: {1}", DateTime.Now, exception.Message);
            Console.ResetColor();
        }

        Thread.Sleep(200);
    }
}
```

KUVIO 23. Event hubiin lähettäminen

Viesti sisältää Json-muotoisen objektin (KUVIO 24), jossa kerrotaan asiakkaan nimi salattuna merkkijonona, ladatun paketin kansiorakenne, ajetun paketin nimi ja latauksen tulos, sekä alku- ja loppuaika. Yksi viesti sisältää aina yhden Json-objektin, jolloin Event Hubiin lähetetään niin monta viestiä, kuin latauksia on tehty.

```
1  [
2  {
3      "Customer": "D971410C25B59EF1CB3BEABB89E47359DE5F83CD9C98DD3CCBE366491E5C45A7",
4      "FolderName": "DEV",
5      "ObjectName": "SA_test_asiakkaat",
6      "PackageName": "SA_Test_asiakkaat_CHAIN.dtsx",
7      "Status": "4",
8      "StartTime": "2016-08-01 15:54:47.0326630 +0",
9      "EndTime": "2016-08-01 15:54:48.5198081 +0"
10 } ,
11 {
12     "Customer": "D971410C25B59EF1CB3BEABB89E47359DE5F83CD9C98DD3CCBE366491E5C45A7",
13     "FolderName": "DEV",
14     "ObjectName": "SA_test_tunnit",
15     "PackageName": "SA_test_tunnit_CHAIN.dtsx",
16     "Status": "7",
17     "StartTime": "2016-07-14 09:42:36.1458201 +0",
18     "EndTime": "2016-07-14 09:42:51.1030590 +0"
19 }
20 ]
21
```

KUVIO 24. Json-muotoinen data

5.5 Stream Analytics

Datan siirtämiseen Event Hubista Azure DB -tietokantaan, käytetään Stream Analytics työtä (job), joka luodaan kuviossa 25 vasemmalla. Kun työ on luotu, päästään päänäkymään (KUVIO 25, oikealla) jossa näkyy komponentit: lähteet, kysely, määränpäätt sekä statistiikkaa datan liikkumisesta.

The screenshot displays the 'New Stream Analytics Job' configuration window. On the left, there is a sidebar with navigation icons. The main configuration area includes the following fields:

- Job name:** A text input field with the placeholder 'Enter job name'.
- Subscription:** A dropdown menu currently set to 'Developer Program Benefit'.
- Resource group:** Radio buttons for 'Create new' (selected) and 'Use existing', followed by an empty text input field.
- Location:** A dropdown menu currently set to 'North Europe'.
- Pin to dashboard:** An unchecked checkbox.
- Create:** A blue button.
- Automation options:** A link.

On the right, the 'Job Topology' section shows a flow from 'Inputs' (1 instance of 'test') to 'Query' (represented by a double arrow icon) to 'Outputs' (1 instance of 'EventTestDB'). Below this is the 'Monitoring' section, which features a line chart titled 'Input Events, Output Events and one more metric past week'. The chart shows a significant spike on Feb 27. Below the chart, there are three summary cards: 'INPUT EVENTS' with a value of 20, 'OUTPUT EVENTS' with a value of 19, and 'RUNTIME ERRORS' with a value of 2.

KUVIO 25. Stream Analytics Jobin luonti ja päänäkymä

Seuraavaksi valitaan datan lähde, sekä määränpää (KUVIO 26). Tässä toteutuksessa lähteenä käytetään Event Hubia, joka luotiin aiemmassa luvussa. Lisäksi Valitaan mitä oikeuksia käytetään, sekä missä muodossa data tulee. Määränpääksi valitaan SQL-tietokanta, sekä tietokannan taulun nimi, johon tiedot halutaan viedä.

The image shows two side-by-side configuration windows for Stream Analytics. The left window is titled 'New input' and the right window is titled 'New output'.

New input configuration:

- * Input alias: (empty text field)
- * Source Type: Data stream (dropdown)
- * Source: Event hub (dropdown)
- * Subscription: Use event hub from current subscription (dropdown)
- * Service bus namespace: monitoringApp (dropdown)
- * Event hub name: monitoringhub (dropdown)
- * Event hub policy name: Main (dropdown with a red warning icon)
- Event hub consumer group: (empty text field)
- * Event serialization format: JSON (dropdown)
- Encoding: UTF-8 (dropdown)

New output configuration:

- * Output alias: (empty text field)
- * Sink: SQL database (dropdown)
- * Subscription: Use SQL database from current subscription (dropdown)
- * Database: EventTest (dropdown)
- Server name: salomaatestdb.database.windows.net (text field)
- * Username: (empty text field)
- * Password: (empty text field)
- * Table: (empty text field)

Both windows have a 'Create' button at the bottom.

KUVIO 26. Stream Analytics input ja output luonti

Lähteen ja määränpään valinnan jälkeen voidaan luoda kysely, jossa määritellään, mistä lähteestä dataa luetaan (KUVIO 27, rivi 11 - 12), sekä minne data viedään (KUVIO 27, rivi 9 - 10). Select-osuudessa määritellään, mitkä sarakkeet datasta viedään tietokantaan ja muutetaan nimet vastaamaan tietokannassa olevan taulun sarakkeita. Kun kysely on määritelty, voidaan Stream Analytics käynnistää, jolloin se on toiminnassa kaiken aikaa. Stream Analytics voidaan myös käynnistää ajastetusti haluttuun aikaan halutuksi ajaksi.

Save Discard Test

Inputs (1)
test

Outputs (1)
EventTestDB

Need help with your query? Check out some of the most common Stream

```
1 SELECT
2     Customer AS customerHash
3     ,foldername as folderName
4     ,objectname AS objectName
5     ,packagename AS packageName
6     ,status
7     ,starttime AS startTime
8     , endtime AS endTime
9 INTO
10 EventTestDB
11 FROM
12 Test
```

KUVIO 27. Stream Analytics -kysely

5.6 Tietokannan luominen

Azureen luodessa SQL-tietokanta, luodaan ensin SQL Server, johon tietokanta ja sen taulut luodaan. SQL Server luodaan (KUVIO 28, vasemmalla) antamalla serverille nimi, tunnukset, tilaaja, resurssi ryhmä, sekä palvelimen sijainti. Kun SQL serveri on luotu, luodaan SQL-tietokanta (KUVIO 28, oikealla).

The image displays two side-by-side screenshots of the Azure portal configuration interface. The left screenshot is titled "SQL Server (logical server o...)" and shows the configuration for a new SQL Server. The right screenshot is titled "SQL Database" and shows the configuration for a new SQL Database.

SQL Server (logical server o...) Configuration:

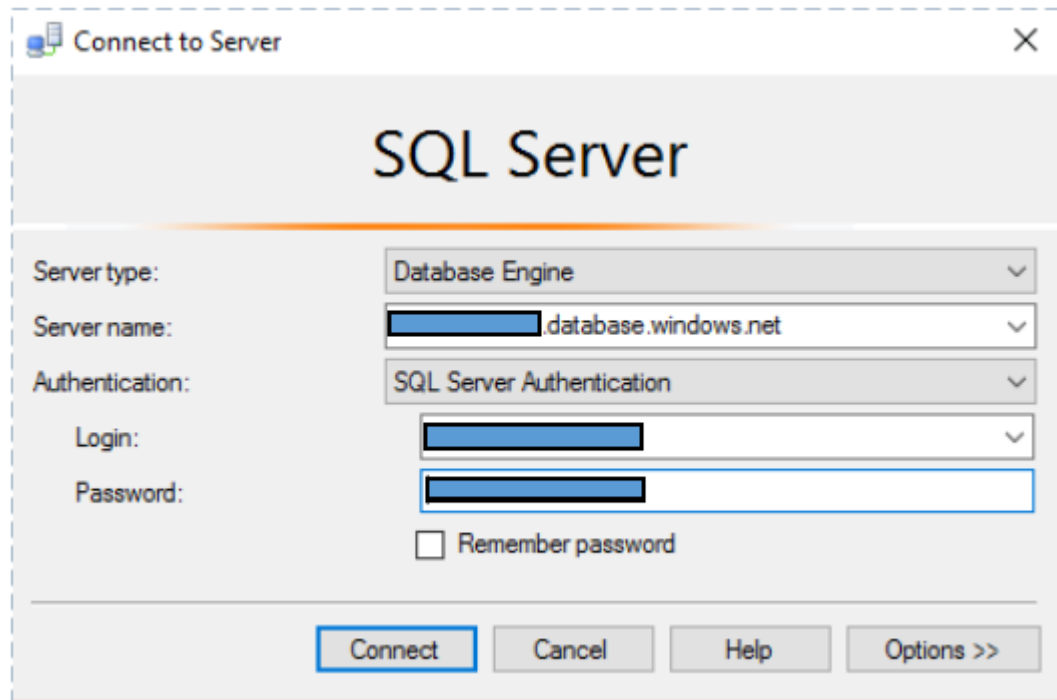
- Server name: testing123456 (with a green checkmark and ".database.windows.net" suffix)
- Server admin login: Enter user name (with a red error bar)
- Password: (with a red error bar)
- Confirm password: (with a red error bar)
- Subscription: Developer Program Benefit
- Resource group: Create new (selected) / Use existing
- Location: West Europe
- Allow azure services to access server:
- Buttons: Create, Automation options

SQL Database Configuration:

- Database name: testing (with a green checkmark)
- Subscription: Developer Program Benefit
- Resource group: Create new / Use existing (selected) / TestRG
- Select source: Blank database
- Server: salomaatestdb (West Europe)
- Want to use SQL elastic pool?: Yes / Not now (selected)
- Pricing tier: Basic (highlighted in blue)
- Collation: SQL_Latin1_General_CP1_CI_AS
- Buttons: Create, Automation options

KUVIO 28. SQL serverin ja tietokannan luominen

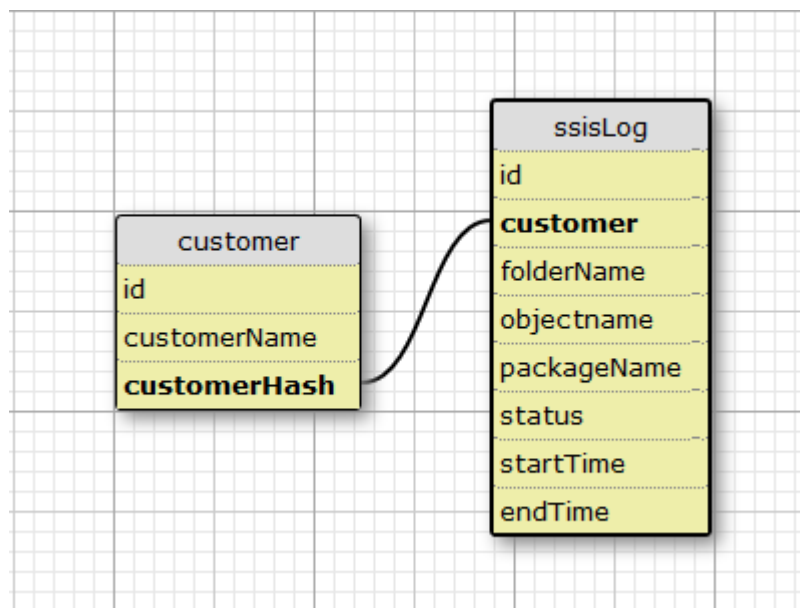
Tietokannan luomisen jälkeen, tietokantaa voidaan hallinnoida Microsoftin SQL Server Management Studio -ohjelmalla. Ennen kuin Management Studiolla päästään kirjautumaan tietokantaan, täytyy SQL serverin palomuuuri asetuksiin käydä lisäämässä tietokoneen IP-osoite. Management studiolla kirjaututaan (KUVIO 29) antamalla Azureen luodun SQL serverin osoite, sekä käyttäjätunnus ja salasana.



KUVIO 29. SQL Server Management Studioon kirjautuminen

5.7 Tietokannan taulut

Tietokantaan luodaan kaksi taulua: customer ja ssisLog (KUVIO 30). Stream Analytics täyttää ssisLog-taulua ja customer-taulua täytetään manuaalisesti aina, kun uusia asiakkaita tulee valvonnan piiriin.



KUVIO 30. Tietokannan taulu rakenne

5.8 Datat esittäminen raportilla

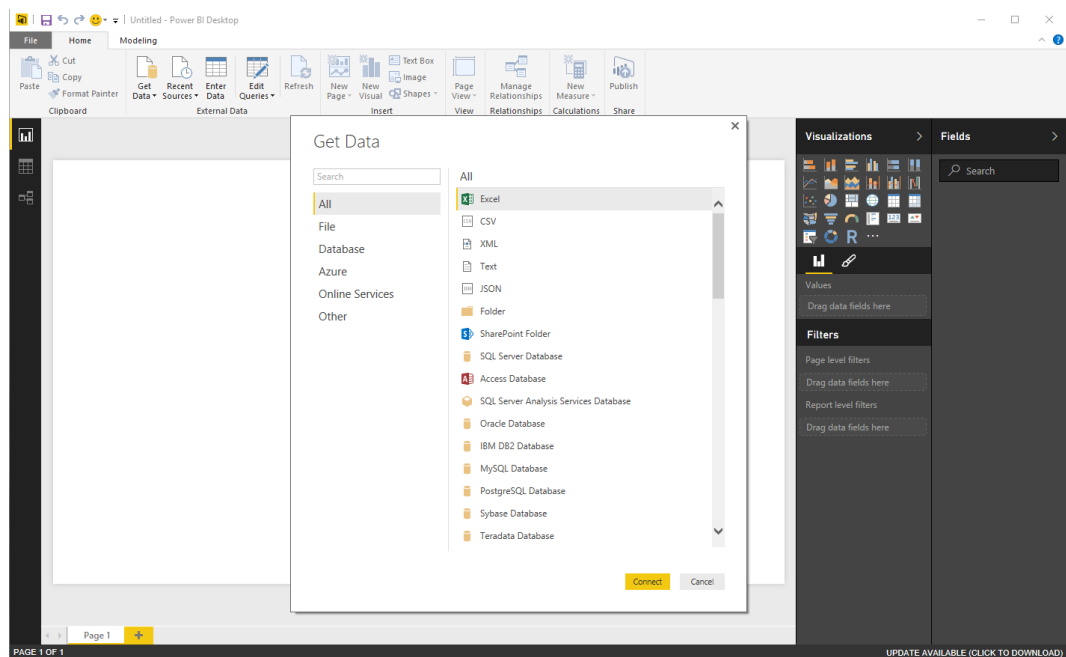
Kun tietokantaan on saatu datat ladattua, voidaan aloittaa raportin tekeminen Power BI Desktopilla. Ensimmäisenä data ladataan ohjelmaan, jonka jälkeen se mallinnetaan, eli yhdistellään taulut keskenään, sekä luodaan tarvittavat sarakkeet ja mittarit. Viimeiseksi data visualisoidaan valmiiksi raportiksi.

5.8.1 Datat lataaminen

Data ladataan ohjelmaan valitsemalla lähde lähdeluettelosta (KUVIO 31).

Data ladataan Azuren palveluista, joten valitaan Azuren alta Microsoft Azure SQL Database, jonka jälkeen annetaan tunnistautumistiedot.

Tämän jälkeen valitaan taulut, jotka halutaan tietokannasta ladata.



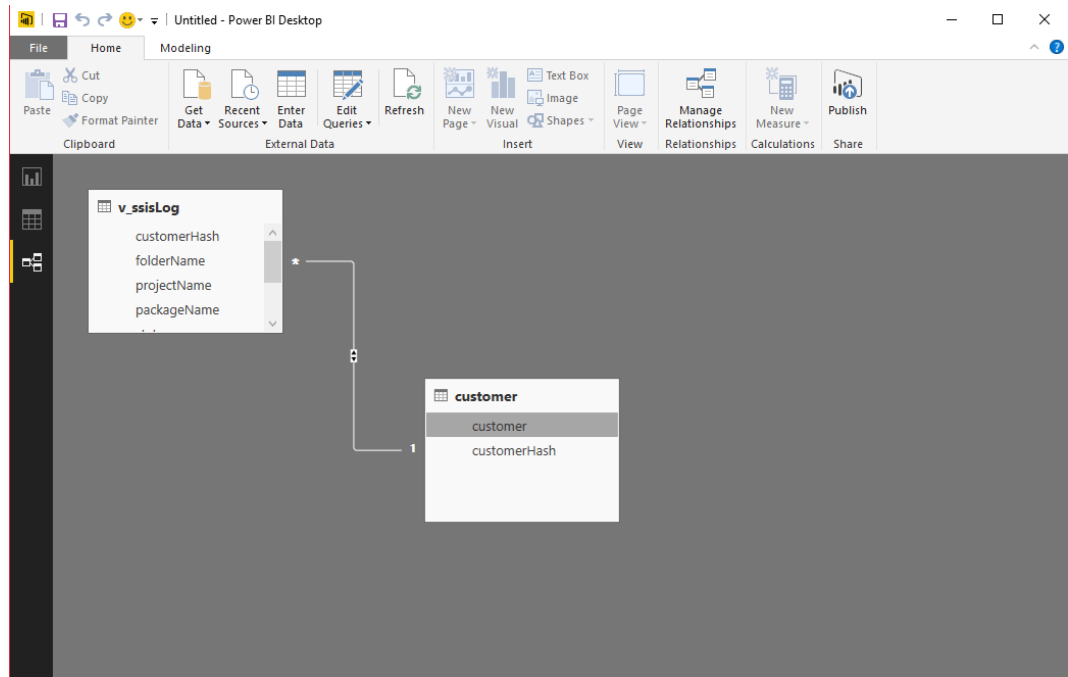
KUVIO 31. Lähteen valitseminen

5.8.2 Mallintaminen

Kun data on ladattu ohjelmaan, voidaan data mallintaa eli yhdistellä taulut toisiinsa. Yhdistämistä varten avataan relaatio-ikkuna (KUVIO 32).

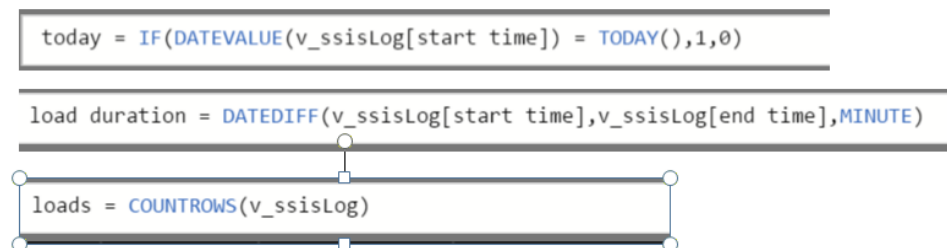
Ohjelman automaattinen liitosten tunnistus tunnisti liitoksen ladattujen

taulujen välillä ja muodosti liitoksen taulujen välille automaattisesti. Muuten tauluja voidaan yhdistellä, joko hiirellä vetämällä taulusta toiseen tai valitsemalla Manage Relationships -painikkeen alta, yhdistettävät taulut.



KUVIO 32. Relaatio-ikkuna

Mittarit ja sarakkeet luodaan data-ikkunassa, jossa näkyy taulut ja niiden sisältämä rivitiedot. Mittareita ja sarakkeita luodaan käyttämällä DAX-kieltä. Raportointia varten luodaan laskennalliset sarakkeet (KUVIO 33) kertomaan latauksien kestot, mitkä lataukset on tänään suoritettu, sekä sarake, joka kertoo tekstimuodossa latauksen tilan. Tämän jälkeen luodaan myös mittari laskemaan latauksien määrä.



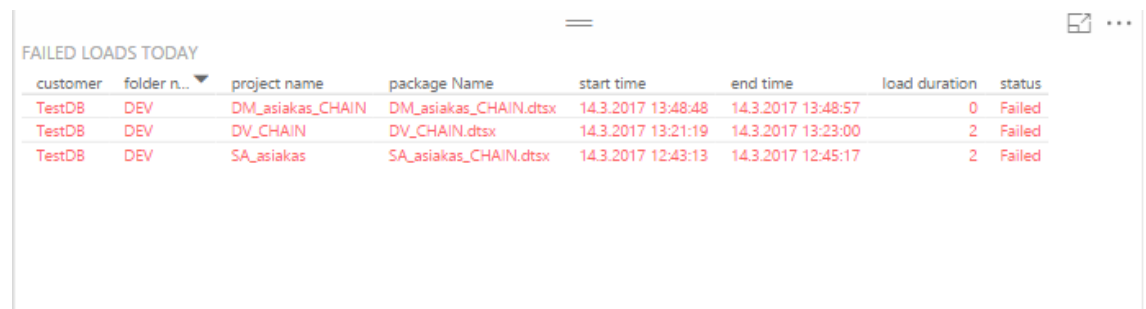
Kuvio 33. Sarakkeiden ja mittareiden luonti

5.8.3 Datan visualisointi raportiksi

Visualisointi tehdään raportointi-ikkunassa, jonka oikeasta reunasta löytyy taulut, niiden sarakkeet ja mittarit sekä visualisointi komponentit.

Ensimmäisellä raportointi sivulla näytetään kaikki kuluvan päivän kaatuneet lataukset ja toisella sivulla esitetään lataajien historiatietoja.

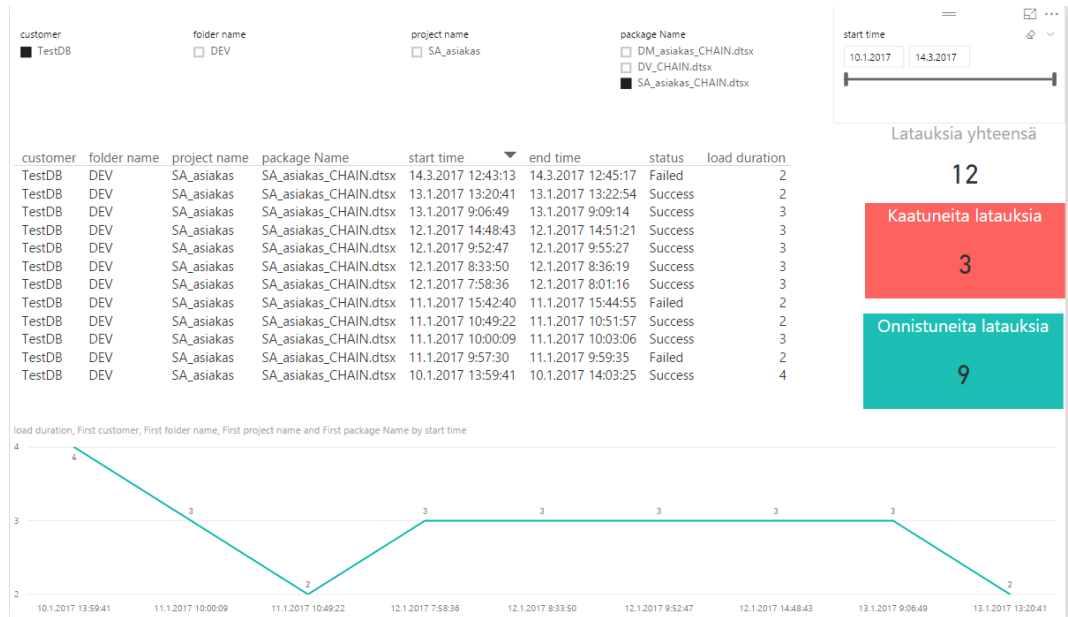
Latauksien kaatumisten seuranta varten luodaan taulukko, painamalla taulukuvaketta komponenttilistalta. Taulukkoon lisätään halutut sarakkeet, sekä rajataan tauluun vain ne rivit, joissa lataus on tänään epäonnistunut (KUVIO 34).



customer	folder n...	project name	package Name	start time	end time	load duration	status
TestDB	DEV	DM_asiakas_CHAIN	DM_asiakas_CHAIN.dtsx	14.3.2017 13:48:48	14.3.2017 13:48:57	0	Failed
TestDB	DEV	DV_CHAIN	DV_CHAIN.dtsx	14.3.2017 13:21:19	14.3.2017 13:23:00	2	Failed
TestDB	DEV	SA_asiakas	SA_asiakas_CHAIN.dtsx	14.3.2017 12:43:13	14.3.2017 12:45:17	2	Failed

Kuvio 34. Epäonnistuneet lataukset

Toiselle raportointisivulle (KUVIO 35) lisätään komponentteja, joiden avulla voidaan seurata latauksien historiatietoja. Ylälaitaan lisätään filttareita, kuten asiakas ja aika. Näillä filttareilla voidaan valita, miten laajasti latauksien historiaa halutaan seurata. Lisätään myös viivadiagrammi, joka kertoo latauksien keston, jolloin nähdään, onko paketin lataus aika kasvanut projektin edetessä. Näiden lisäksi lisätään taulukko kertomaan rivitasolla latauksien tietoja, sekä kolme korttia, jotka kertovat epäonnistuneiden ja onnistuneiden latauksien kokonaismäärän.



KUVIO 35. Toinen raportointi sivu

6 YHTEENVETO

Opinnäytetyössä käytiin läpi tietovarastointia, Azure-pilvipalvelua, sekä raportointia Power BI -ohjelmistolla. Tavoitteena oli saada tietovaraston latauksien tiedot esitettyä Power BI -raportilla. Tämän lisäksi tavoitteena oli luoda Azureen tarvittavat palvelut, joiden avulla latauksien tiedot saadaan siirrettyä asiakkaiden ympäristöistä yhteiseen tietokantaan. Tavoitteet saavutettiin, mutta testaaminen jäi vähäiseksi, sillä sovellukset, joilla tietovaraston latauksien tiedot kerätään ja lähetetään ovat vielä kesken.

Tietovaraston latauksien seuranta saatiin tehtyä ja lopputuloksena saatiin raportti, josta nähdään epäonnistuneet lataukset, sekä latauksien historiatietoja. Jos työn aloitaisi uudelleen, täytyisi raportin tekemiseen käyttää enemmän aikaa, jolloin raportista saataisiin paremman näköinen. Tämän lisäksi osan raportilla olevista laskennoista voisi tehdä jo tietokannan näkymiin.

Jatkokehityksenä, asiakkaan ympäristössä olevat sovellukset tullaan tekemään valmiiksi, jotta tietovaraston valvonta saataisiin ylläpidon käyttöön. Myös raportin ulkoasua tullaan kohentamaan.

Seuraavaan versioon tullaan lisäämään uusia seurannankohteita, kuten taulujen koko, raporttien määrä ja niiden käytettävyys, sekä levytilan seuranta. Tarkoituksena olisi myös laajentaa seurantaa, esimerkiksi Qlik-ohjelmistolle.

LÄHTEET

DataOnFocus 2015. Data Mart vs Data Warehouse [viitattu 12.3.2017].
Saatavissa: <http://www.dataonfocus.com/data-mart-vs-data-warehouse/>

Enho, H. 2016. Power BI – kaikki mitä sinun tulee tietää aloittaessasi [viitattu 12.1.2017]. Saatavissa: <https://hexcelligent.fi/2016/01/30/power-bi-kaikki-mita-sinun-tulee-tietaa-aloittaessasi/>

Forbes 2014. Microsoft's Azure Cloud Platform Explained - Part 1 [viitattu 17.2.2017]. Saatavissa:
<https://www.forbes.com/sites/greatspeculations/2014/12/19/microsofts-azure-cloud-platform-explained-part-1/#6ec77ebd19e7>

Foulds, I. & Sqillace, R. 2017. Azure resource group guidelines for Windows VMs [viitattu 13.3.2017]. Saatavissa:
<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/infrastructure-resource-groups-guidelines>

Hovi, A., Hervonen, H. & Koistinen, H. 2009. Tietovarastot ja Business Intelligence. Jyväskylä: WSOY.

Hovi, A., Huotari, J. & Lahdenmäki, T. 2003, 2005. Tietovarastojen suunnittelu [viitattu 15.1.2017]. Saatavissa: <http://docplayer.fi/5659571-Tietovarastojen-suunnittelu.html>

Hyötyläinen, T. 2014. Data Vault 2.0 – tietovarastojen villin lännen ”pieni suuri” metodi? [viitattu 3.2.2017]. Saatavissa: <https://gofore.com/data-vault-2-0-tietovarastojen-villin-lannen-pieni-suuri-metodi/>

Kimball, R. & Ross, M. 2002. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling Second Edition. Hoboken, NJ, USA: John Wiley and Sons, Inc. Saatavissa:
<https://books.google.fi/books?id=XoS2oy1lcB4C&printsec=frontcover&dq=data+warehouse&hl=fi&sa=X&ei=Fc36VLHzFout7gbf84HIBw&sqj=2&ved=0CDoQ6AEwAQ#v=onepage&q&f=false>

- Lachev, T. 2016. Applied Microsoft Power BI. Prologika Press.
- Larson, B. 2016. Power BI Connection Types [viitattu 12.1.2017].
Saatavissa: <https://www.teamscs.com/2016/08/power-bi-connection-types/>
- Llopis, M. 44 New Features in the Power BI Desktop September Update [viitattu 13.1.2017]. Saatavissa: <https://powerbi.microsoft.com/en-us/blog/44-new-features-in-the-power-bi-desktop-september-update/>
- Microsoft 2017a. What is Azure Event Hubs [viitattu 20.1.2017]?
Saatavissa: <https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-what-is-event-hubs>
- Microsoft 2017b. What is Stream Analytics [viitattu 22.1.2017]?
Saatavissa: <https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction>
- Microsoft 2017c. What is SQL Database? Introduction to SQL Database [viitattu 23.1.2017]. Saatavissa: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-technical-overview>
- Roman, P. 2015. Azure Cloud Service VS Azure Resource Manager [viitattu 13.3.2017]. Saatavissa:
<https://blogs.technet.microsoft.com/canitpro/2015/09/17/azure-cloud-service-vs-azure-resource-manager/>
- Tulloch, M. 2013. Introducing Windows Azure for IT Professionals. Microsoft. Saatavissa: <https://mva.microsoft.com/ebooks>
- Van Den Berg, M. 2016. An overview of cloud vision and solutions of VMware and Microsoft. Part one: IaaS [viitattu 12.1.2017]. Saatavissa:
<http://up2v.nl/2016/09/04/an-overview-of-cloud-vision-and-solutions-of-vmware-and-microsoft-part-one-iaas/>

