ETL-prosessin parhaita käytäntöjä tietovaraston rakentamisessa

Juha-Pekka Honkavaara

Tampereen yliopisto
Informaatiotieteiden yksikkö
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaaja: Marko Mäkipää
Kesäkuu 2013

Tampereen yliopisto

Informaatiotieteiden yksikkö

Tietojenkäsittelyoppi

Juha-Pekka Honkavaara: ETL-prosessin parhaita käytäntöjä tietovaraston rakentamisessa

Pro gradu -tutkielma, 58 sivua

Kesäkuu 2013

ETL-prosessia käytetään integroimaan tietojärjestelmien tietoja yhteen. Prosessissa siirretään tietojärjestelmien dataa tietokannoista toisiin tietokantoihin. Prosessin avulla rakennetaan esimerkiksi tietovarastoja joita käytetään yritysten päätöksentekojärjestelmissä.

Tässä tutkielmassa keskitytään erityisesti yrityksessä tapahtuvaan tietovaraston rakentamiseen ETL-prosessin avulla. Tulen esittelemään perustiedot tietovarastoista, yrityksen tietojärjestelmistä ja ETL-prosessista. Reflektiivistä tutkimusmetodia apuna käyttäen esittelen parhaita toimintatapoja ja käytäntöjä ETL-prosessin suunnittelussa ja rakentamisessa. Tämä tutkielma on hyödyksi kenelle tahansa tietovarasto projektia suunnittelevalle tai ETL-prosessien parissa työskenteleville.

Avainsanat ja -sanonnat: ETL-prosessi, Tietovarastot, Asiakasrekisteri, Toiminnanohjausjärjestelmä, Yrityksen tietojärjestelmät, Liiketoimintatiedon hallinta, Business Intelligence, Tietokanta, SQL

Sisällysluettelo

1. Johdanto

Yritysten tietojärjestelmät sisältävät paljon tietoa, joka on tärkeää yritykselle [Ruohonen & Salmela, 1999]. Tiedot sijaitsevat useassa paikassa hajautettuina ja ne ovat vaikeasti hallittavissa. Suomen tilastokeskuksen tekemän tutkimuksen mukaan vuonna 2012 yrityksistä 33 prosentilla oli käytössä toiminnanohjausjärjestelmä (ERP) ja 37 prosentilla asiakkuudenhallintajärjestelmä (CRM) [Tilastokeskus, 2012]. Näistä 57 prosenttia jakoi tietoa sähköisesti muiden tietojärjestelmien kanssa. Yrityksen johdon on tärkeää saada kaikki tämä tieto käyttöönsä, jotta he voivat tehdä tärkeitä päätöksiä perustuen todelliseen dataan [Mundy et al., 2006]. ”Yritys, joka pystyy hallitsemaan ja hyödyntämään tietonsa saa merkittävän kilpailuedun” [Hovi, 1997].

Tietovarastoja rakennetaan yrityksiin varastoimaan ja keräämään yhteen tätä pirstaloitua dataa [Kimball, 1996]. Tietovarastojen dataa käytetään tiedon analysointiin. Analyysia voidaan suorittaa esimerkiksi asiakassuhteiden tilanteesta, asiakkaiden saamisen/menettämisen syistä, myynnistä, ostoista, sekä yrityksen vahvuuksista ja heikkouksista. Tietoa käytetään yrityksen johdon hyväksi, jotta he pystyisivät tekemään oikeita päätöksiä käyttäen analysointi/raportointityökaluja. Tietovarastot toimivat myös tärkeän yritystiedon varmuuskopiona, historiatietona sekä tiedon jakamisessa yrityksen eri osastojen, osien ja järjestelmien kesken. Eräiden työkalujen toiminta pohjautuu nimenomaan tietovarastossa olevaan dataan, kuten esimerkiksi liiketoimintatiedon hallintaohjelmisto (eng. Business Intelligence, lyh. BI). BI yhdistelee tietovarastossa olevaa dataa selvästi luettavissa olevaksi esitykseksi, jolloin pystytään saamaan yleiskuva yrityksestä tai tietystä yrityksen osa-alueesta [Moss & Atre, 2003].

Tietovarastoon tuodaan dataa yrityksen muista järjestelmistä, joita ovat esimerkiksi aiemmin mainitut asiakkuudenhallinta- [Anderson & Kerr, 2002] tai toiminnanohjausjärjestelmät [Garg & Venkitakrishnan, 2004]. Tietovarastoprojekteissa on käytetty ETL-prosessia (Extract, Transform, Load) tietojen siirrossa tietokantojen välillä viime vuosina [Mundy et al., 2006]. Tiedon määrä tietojärjestelmissä kasvaa kuitenkin valtavasti koko ajan. Esimerkiksi Syncsort ja HP siirsivät ennätyksellisesti 5.4 terabittiä tietoa alle tunnissa ETL-prosessilla [Syncsort, 2013]. Tiedon määrän kasvamisen vuoksi tietojen luku- ja siirtoprosessin täytyy muuttua samankaltaisesti ja sitä tulisi uudistaa ja tehostaa.

Tämän vuoksi tarvitaan tehokkaampia ja käytettävämpiä tekniikoita ja käytäntöjä tietojen siirron toteuttamiseksi.

Tietovaraston rakentaminen vaatii projektin, jossa käydään läpi yrityksen eri osastojen tarpeet, tarvittavat raportit, tietolähteet, tietomäärät, käytettävät avaimet, paikka ja tapa [Mundy et al., 2006]. ETL-prosessi on yksi tavoista millä voidaan yhdistää useiden eri tietolähteiden data yhteen tietokantaan (tietovarastoon) [Kimball & Caserta, 2004]. Nimensä mukaisesti prosessissa on tarkoituksena koota tieto eri tietolähteistä (Extract), siivota, muokata ja korjata tieto (Transform), ja lopuksi siirtää tieto määränpäähän (Load), joka voi olla tietovarasto tai muu tietokanta. ETL-prosessia voidaan käyttää yrityksessä myös tiedon vientiin tietovarastosta takaisin yrityksen tietojärjestelmiin [Mundy et al., 2006].

Tietovarastoprojektin ja samalla ETL-prosessin onnistumistekijät ovat pääosin samat kuin Hovin [1997] mainitsemat tietojärjestelmäprojektin onnistumistekijät. Tietojärjestelmäprojekti mitataan onnistuneeksi esimerkiksi seuraavien tekijöiden perusteella. Tietojärjestelmän tekninen laatu, tietojärjestelmän tuottaman informaation laatu, tietojärjestelmän positiivinen vaikutus käyttäjän työhön ja päätöksentekoon, vaikutus liiketoimintaprosesseihin ja vaikutus yrityksen kilpailukykyyn [Hovi, 1997]. Tätä tukee myös Wixomin ja Watsonin [2001] tekemä tutkimus niistä kriittisistä tekijöistä, jotka tekevät tietovarastoprojektista onnistuneen.

Tämän tutkielman tarkoituksena on keskittyä tähän ETL-prosessiin joka vahvasti liittyy yrityksissä nykypäivänä tapahtuvaan tietovarastojen luontiin, tiedon hallintaan ja BI-ympäristöihin. Keskityn erityisesti asiakastiedon ja myyntitiedon hallintaan yrityksissä ja siihen kuinka ETL-prosessin avulla voidaan parantaa näitä osa-alueita. ETL-prosessi on tärkeä vaihe, jonka avulla pystytään tehokkaasti hyödyntämään yrityksen sisäistä dataa. ETL-prosessi/projekti jakautuu suunnitteluvaiheeseen, kehittämisvaiheeseen ja käyttöönottoon, jotka käyn läpi tutkielmassa. Käytän lähtökohtana omaa kokemustani ETL-projektien parissa. Olen ollut mukana kahden vuoden aikana kahdeksan eri asiakkaan projekteissa, joissa on integroitu yrityksen eri järjestelmiä yhteen käyttäen ETL-prosessia.

Tutkimusmetodina tutkielmassani on reflektiivinen käytäntö (eng. Reflective practice) [Schön, 1983]. Kiinnitän huomiota seikkoihin, jotka ovat tärkeitä prosessissa onnistumisen kannalta ja esitän mahdollisia ongelmia joita prosessin kehittämisen aikana voi ilmaantua. Tulen esittelemään kuinka näistä asioista voi oppia ja mitä tulisi ottaa huomioon myöhemmin, kun tarvitaan parhaita toimintatapoja muissa tietovarastoprojekteissa. Tutkimuskysymyksenä tässä tutkielmassa on: Miten saadaan tiedot eri tietojärjestelmissä tehokkaasti koottua yhteen, käyttäjien saataville ja ymmärrettävään muotoon?

Luvussa 2 esittelen ETL-prosessiin ja tutkielmaan yleisesti liittyvät osa-alueet. Siinä myös esitetään taustatietoja tietovarastoista, tietovarastoja apuna käyttävistä BI-ohjelmista ja ETL-prosessista tietovarastojen suunnittelusta. Luvun 2 lopuksi esittelen työkaluja, joilla ETL-prosessi voidaan toteuttaa ja ETL-prosessin tavoitteet. Luvussa 3 kerron enemmän käytetystä tutkimusmetodista. Luku 4 koskee ETL-prosessin suunnittelua ja suunnitteluun kuuluvia osa-alueita. Luvussa 5 esittelen ETL-prosessin kehittämistä ja käyn läpi koko ETL-prosessin rakentamisen vaiheittain. Luvussa 6 esittelen tuloksia peilattuna ETL-prosessin tavoitteisiin ja esittelen parhaita käytäntöjä. Luvussa 7 käyn läpi johtopäätökset.

Tämän tutkielman luettuaan lukijalla on yleinen kuva siitä, mitä ETL-prosessi ja projekti vaatii tietovaraston rakentamisessa. Lukija saa kuvan myös siitä, miten prosessista saadaan onnistunut, jolloin saadaan yrityksen tehokkuus tätä myötä kasvamaan. Tutkielmasta on apua esimerkiksi projektipäällikölle, joka suunnittelee tietovarastoprojektia, järjestelmäasiantuntijalle, joka tulee toteuttamaan projektin tai kenelle tahansa joka on kiinnostunut tietovarastoprojektista ja sen vaiheista.

2. ETL-prosessin lähtökohdat ja taustat

2.1 Tietovarastot

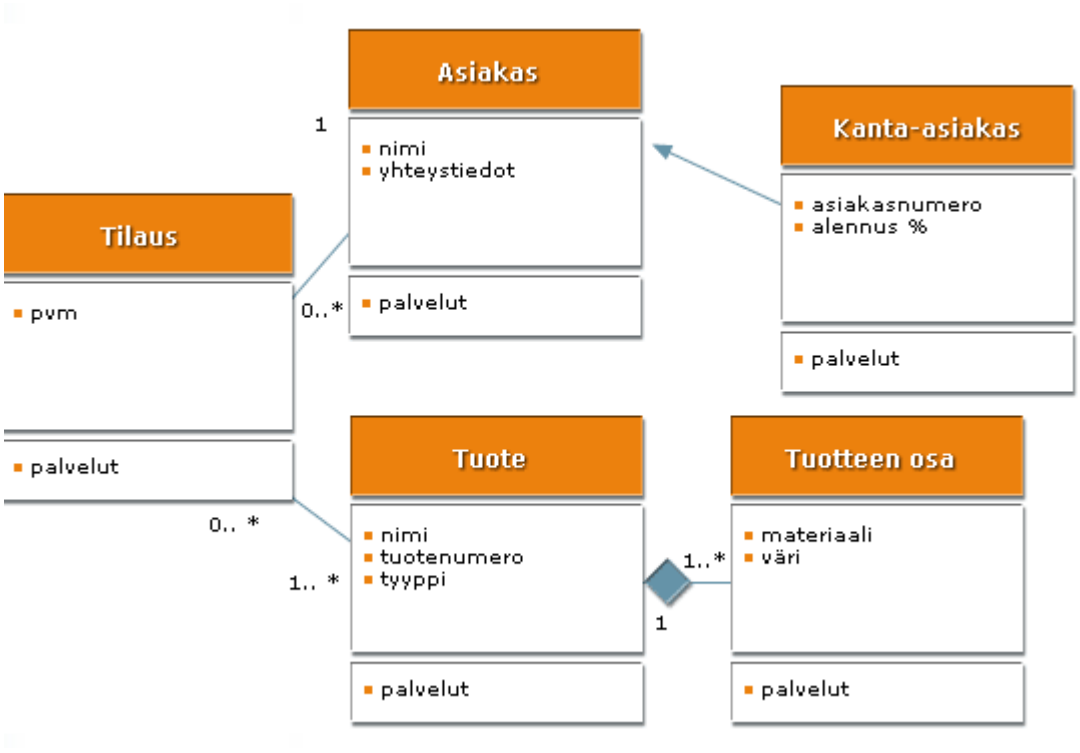
ETL-prosessi liittyy osaltaan tietovarastojen (eng. Data warehouse) rakentamiseen, joten on tärkeää tietää perustiedot tietovarastoista. Tietovarastoihin viitataan myöhemmässä tekstissä myös useasti. ETL-prosessia voidaan tietenkin käyttää myös muussakin yhteydessä, kuten järjestelmien integraatioihin.

Ruohonen ja Salmela [1999] mainitsevat, että yrityksillä on käytössään paljon tietoa eri tietokannoissa ja niillä on myös tarve saada tämä tieto koottua yhteen sekä tehokkaasti saataville yrityksen eri osastoille ja henkilöille. Tieto on tärkeää yrityksen päätöksenteossa. Tarpeellinen tieto yrityksessä kannattaa saattaa yritysjohdon käyttöön, että he pystyisivät tekemään mahdollisimman tarkkoja päätöksiä [Mundy et al., 2006]. Tiedon pitää olla helposti luettavissa ja analysoitavissa jotta päätöksen teko olisi mahdollisimman osuvaa. Faktatietoihin perustuvat oikeat päätökset ovat kriittisiä yrityksen toiminnassa kun toimitaan kilpailevassa kvartaalitaloudessa. Yrityksen johto käyttää tietoa esimerkiksi budjetin analysoimisessa, resurssien ohjaamisessa, toiminnan ennustamisessa ja suunnittelussa [Kulkarni et al., 2010].

Mundy ja muut [2006] mainitsevat, että tietovarasto on paikka mihin kerätään kaikki tieto yrityksen eri tietojärjestelmistä. Tiedon sijaitseminen tietovarastossa auttaa yritystä ja sen johtajia päätöksissä ja työssään, koska tietovarastossa olevaan dataan voidaan käyttää analysointityökaluja ja muita toimintoja. Mitä paremmin tieto on järjestetty ja mitä täydellisempää sekä ajantasaisempaa se on, sitä nopeammin ja helpommin se on analysoitavissa. Mundy ja muut [2006] mainitsevat, että tietovarastossa olevaa tietoa voidaan myös käyttää raportoinnissa ja tiedon yhdistämisessä muiden tietokantojen kanssa. Tiedon vieminen eheytettynä takaisin tietovarastosta työntekijöiden päivittäin käyttämiin tietojärjestelmiin on myös mahdollista. Eheyttäminen tarkoittaa esimerkiksi osoitteiden päivittämistä yritysrekisteristä ja lakkautettujen yritysten poistoa.

Tietovarasto sijaitsee useimmissa tapauksissa SQL-kielellä toteutetussa tietokannassa. Tietokanta taas koostuu tauluista (eng. tables), jotka sisältävät kenttiä (eng. fields). Näiden taulujen koko riippuu niiden sisältämien kenttien määrästä. Yksi taulu voi sisältää

esimerkiksi kaikki perustiedot yrityksestä. Kentät tauluissa sisältävät erilaisia tietoja tauluun liittyvistä asioista. Kentät ovat erityyppisiä ja yksi esimerkki kentästä yritystaulussa on tekstikenttä joka sisältää yrityksen nimen. Tietokannan tai ohjelman rakennetta voidaan kuvata UML-kaaviossa kuten kuvassa 1. Kuvassa yhtä objektia, kuten asiakasta on kuvattu laatikolla. Tämä laatikko sisältää kenttiä (nimi, yhteystiedot). Kuvan 1 esimerkin mukaisesti objektien välillä on yhteyksiä, kuten asiakas voi tilata useita tuotteita yhdessä tilauksessa. UML-kaavio toimii kommunikointivälineenä kehittäjien välillä kuvauksena siitä miten tietokanta on rakennettu. Käytetty SQL-ympäristö määrittelee tekniikat, joita voidaan soveltaa tietokannan analysointiin.



Kuva 1. UML kaavio tauluista, kentistä ja yhteyksistä [UML, 2013].

Tietovaraston tiedot koostuvat yritysmaailmassa tunnetuista objekteista kuten kontakti, osoite, lasku, tuote jne. Tässä esimerkissä on keskitytty asiakas ja taloustietoihin. Näille jokaiselle on omat tunnisteet (y-tunnus, henkilötunnus) ja avaimet (asiakasnumero, tietokanta ID). Myös tiedon tyyppi on määritelty tarkkaan ja se on tarkistettu tietovarastoon vientivaiheessa. Eri objektit yhdistetään toisiinsa avaimilla joista käy ilmi esimerkiksi ketkä työntekijät työskentelevät missäkin yrityksessä tai mitä tuotteita yritys on tilannut. Myöhemmin tässä tutkimuksessa keskitytään enemmän sellaisen tiedon hallin-

taan missä avaimet ja ID numerot ovat saatavilla, eli tieto on hyvin strukturoitua. Muutoin tietokannoista riippuen näitä avaimia pitää erikseen tuottaa (eng. generate), jotta ne osattaisiin yhdistää.

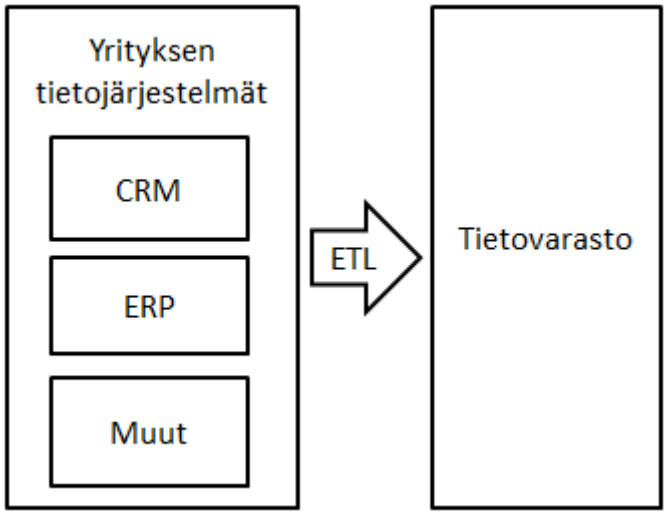
Mundy ja muut [2006] mainitsevat, että tietovaraston kenties tärkein tieto on aika, varsinkin kun puhutaan tiedon analysoinnista. Aika määrittelee analysointivaiheessa, mitä on tehty esimerkiksi tietyssä asiakkuudessa milloinkin ja miten asiakkuus on syventynyt tai etääntynyt ajan saatossa. Vertailemalla tapahtumia keskenään aikatietojen avulla saadaan tietoon syyt asiakkuuden muutoksiin. Hyvin suunniteltu tietovarasto pitää kirjaa muuttuneista tiedoista, jolloin jokaisella muutoksella on oma avain ja rivi tietovarastossa tallessa. Tällöin nähdään esimerkiksi kaikki tapahtumat liittyen tiettyyn objektiin tai saadaan haettua objektin tila tiettyyn aikaan [Mundy et al., 2006].

Tietovarastoon kertyy ajan saatossa valtavasti tietoa. Tietovaraston koko kasvaa nopeasti varsinkin suurissa yrityksissä joissa on useita henkilöitä syöttämässä ja muuttamassa arvoja satoja kertoja työpäivän aikana. Uutta tietoa viedään varastoon joka päivä, kuten esimerkiksi jokaisen työntekijän päivän kirjaukset. Kaikki tämä tietomäärä kasvattaa tietokannan kokoa, ja se vaatii suunnitelmallista tiedon järjestelyä ja laskentatehoa. Tätä voidaan tietovaraston ja ETL-prosessin hyvällä suunnittelulla tehokkaasti auttaa. Arora ja muut [2009] mainitsevat, että kaikki tieto pitää ottaa tietovarastossa talteen, järjestellä ja linkittää, jotta sitä voidaan analysoida/hakea tehokkaasti. Tietovarasto toimii samankaltaisesti kuin tavallinen varasto ja jos tavarat eivät ole järjestyksessä ja kirjattu niin tavaroita ei löydetä helposti.

2.1.1 Tietovarastot ja ETL

Tietovarastoon voidaan kerätä tietoa monesta eri tietolähteestä yrityksessä. Ruohonen ja Salmela [1999] ovat kirjassaan esitelleet erilaisia yrityksen tietojärjestelmiä ja niiden sisältämiä tietoja. Tällaisia ovat esimerkiksi asiakkuudenhallintajärjestelmä (CRM), jossa myyjät voivat hallinnoida asiakassuhteita ja tehdä myyntityötä. Samalla päivitetään asiakaskontaktien yhteystietoja ja kirjoitetaan muistiin keskusteltuja asioita [Anderson & Kerr, 2002]. Toiminnanohjausjärjestelmässä (ERP) ovat kaikki asiakkaisiin liittyvät myyntisaatavat ja tilaushistoria. Markkinoinnilla on omissa järjestelmissään tai Excel-taulukoissa tieto kaikista markkinointiaktiviteeteista joita on kohdistettu asiakkaaseen, kuten esimerkiksi postituksista. Tuotannolla on tieto mitä on tuotantokannassa

tai mitä valmiita tuotteita on sillä hetkellä varastossa. Asiakaspalvelulla on tieto asiakkaan kyselyistä ja avoimista tai ratkaistuista ongelmista (reklamaatiot). Tästä kokonaisuudesta voi huomata, että yrityksen eri osastot ja osastojen työntekijät tuottavat tietoa, mitkä yhteen tuomisen jälkeen näyttävät kaiken mitä yrityksessä tapahtuu. Kuva 2 näyttää miten ETL-prosessi on sijoittunut tässä tietovaraston tai muun järjestelmän rakentamisen vaiheessa, eli kaikkien yrityksessä olevien tietojen viemisessä tietovarastoon. Kuvassa 2 vasemmalla on kaikki yrityksen tietojärjestelmät ja niiden tietokannat. Oikealla on tietovarasto. ETL-prosessi sijaitsee näiden välissä ja sen kautta kaikki tieto ajetaan tiedon siirtyessä tietovarastoon.

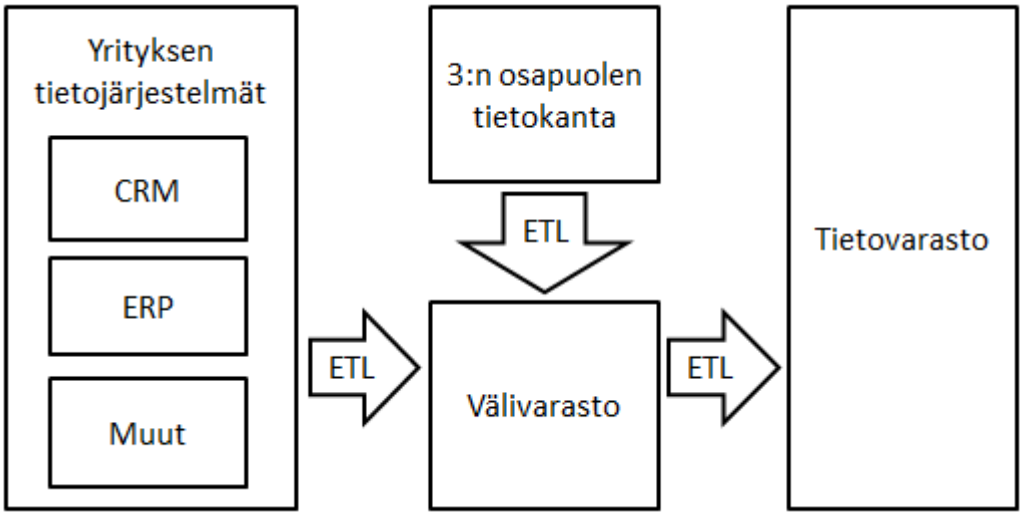


Kuva 2. ETL-prosessi tietovaraston rakentamisessa.

Tietovaraston rakentaminen ja eheys riippuu tästä vaiheesta, joten ETL-prosessin täytyy olla niin hyvin suunniteltu ja toteutettu, että siitä ei aiheudu ongelmia tietovarastoa käyttäville. Tämä on yksi ehto tietovarastoprojektin onnistumiselle joka mainittiin johdannossa.

Ennen tiedon keräämistä tietovarastoon tieto voidaan joissain tapauksissa joutua kierrättämään välitietokannan (eng. staging area) kautta, joka yhdistää tietoja useasta eri tietokannasta ja eheyttää tietoa [Mundy et al., 2006]. Näin tehdään sen takia, että tietovarastoa lukee suoraan muut järjestelmät, joten siihen ei kannata viedä väliaikaista tietoa.

Kuva 2 näyttää asiat yksinkertaisimmillaan. Suurten yritysten tietokannat vaativat useita eri ETL-prosesseja ja välitietokannan, sekä testiympäristön rakentamista, jota havainnollistetaan kuvassa 3. Tietoa joudutaan eri prosesseissa yhdistelemään, eheyttämään ja päivittämään. Yrityksen johdon avustamista ja toiminnan kehittämistä ajatellen pitää ottaa kaikki muuttuneet tiedot talteen, kuten aikaisemmin on esitelty. Kuva 3 näyttää esimerkin isomman tietovaraston luomisesta ja kannattaa huomata, että ETL-prosesseja on useampia koko projektissa. ETL-prosesseja on jokaisen eri tietokannan ja ulkoisen osapuolen datan välillä. Mundy ja muut [2006] esittävät, että tämä on tilanne, kun siirrytään isompien tietovarastojen rakennushankkeisiin.



Kuva 3. Isomman tietovarastoprojektin kuvaus [Wikipedia tietovarasto, 2013].

Kuva 3 eroaa edellisestä kuvasta siten, että yrityksen omia tietoja täydennetään kolmannen osapuolen tiedoilla. Kolmannen osapuolen tarjoaman tietokannan tiedot (eng. external data) ajetaan ETL-prosessin läpi, jotta voidaan tarkistaa, että tieto on täysin oikeata ja oikeassa muodossa. Tämä kolmannen osapuolen tietokanta on esimerkiksi yritysrekisteri joka tarjoaa viimeisimmät yhteys- ja luottotiedot yrityksestä. Kuvasta 3 on nähtävissä, että turvallisinta on siirtää tiedot välivarastoon ja sovittaa tietoja yrityksen tietojärjestelmien kanssa. Mundy ja muut [2006] suosittelevat tätä tapaa. Kolmannen osapuolen tiedot siirretään ETL-prosessin avulla, joka vaatii mahdollisesti muita tekniikoita, kuin tietokannasta lukua SQL-komennoin. Näitä on esimerkiksi XML ja/tai WWW-sovelluspalvelukutsut (eng. Webservice). Näitä tekniikoita ei käydä tässä tutkielmassa läpi sen tarkemmin.

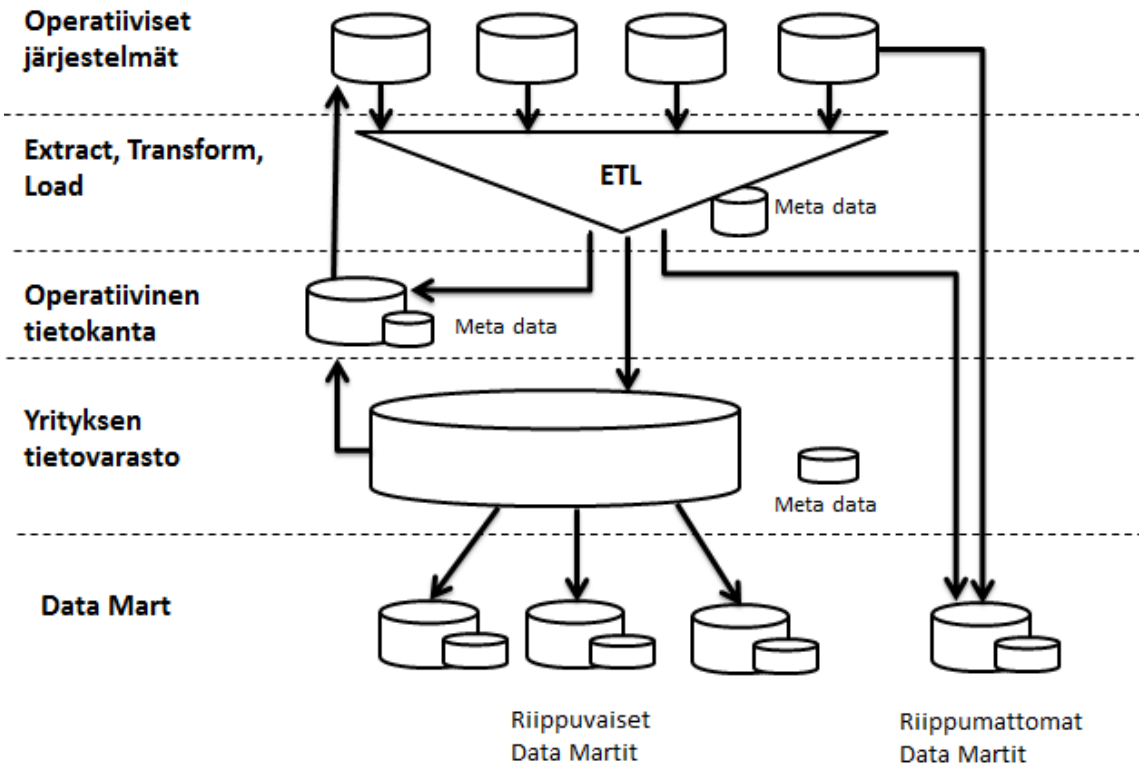
Aina ei ole tarpeen tehdä tietovarastoa ja tällöin rakentaa ETL-prosessia. Pienten ja keskisuurten yritysten käytössä voi olla vain yksi järjestelmä, jolloin tietovarasto ei ole tarpeen. Heillä voi olla esimerkiksi asiakasrekisterijärjestelmä, josta käyttäjät näkevät kaikki tiedot tietovaraston tavoin. Näitä järjestelmiä on mahdollista laajentaa moduuleilla ja ne tarjoavat analysointi-työkalut yrityksen käyttöön. Talon omat ohjelmointiresurssit voivat tehdä omia kyselyitä ja analysointeja suoraan SQL-tietokantaan. Yrityksen kannattaa siis miettiä tarkasti, ovatko heidän tarpeensa niin suuria, että tietovarastoprojekti on välttämätön. Tietovaraston rakentaminen sitoo työtunteja ja vaatii ylläpitoa. Tietovaraston ylläpito sitoo silloin yrityksestä myös työntekijöitä käyttöönoton jälkeenkin, koska tietovarasto halutaan pitää eheänä ja ajan tasalla.

2.1.2 Tietovarastojen pilkkominen paikallisvarastoihin

Tietovarasto ei välttämättä ole tiedon lopullinen varastointipaikka josta analysoinnit tehdään. Tietovarasto voi olla joissakin yrityksissä varsin laaja jolloin se on parasta pilkkoa osiin joillakin kriteereillä. Yrityksessä tällaisia kriteerejä ovat esimerkiksi yrityksen eri osastot. Tietovaraston pilkottuja osia kutsutaan nimellä paikallisvarasto (eng. Data Mart) [Moody & Kortink, 2000]. ”Datamart on tietovarastoa pienempi, usein osastokohtainen paikallisvarasto” [Hovi, 1997]. Paikallisvarastoon voidaan soveltaa kyselyitä ja analyyseja, jotka koskevat vain yhtä osaa (osastoa) yrityksestä. Osastot voivat tehdä omia analysointeja ja ylläpitää dataa erillään muista osastoista jolloin myös datan siivous helpottuu ja nopeutuu. Moody ja Kortink [2000] mainitsevat tämän yhdeksi paikallisvarastojen suurimmiksi hyödyiksi, koska erittäin laaja tietovarasto on vaikea ylläpitää ja analysoida. Tietovarasto sisältää sellaista tietoa mikä ei ole hyödyllistä jokaiselle osastolle yrityksessä. Paikallisvarastoon pääsee käsiksi suoraan jopa loppukäyttäjä, sillä tietovarasto on melkein aina suojattuna suoralta lukemiselta ja kirjoittamiselta [Moody & Kortink, 2000].

Kuva 4 havainnollistaa paikallisvarastojen sijoittumista suhteessa yrityksen tietojärjestelmiin ja ETL-prosessiin. Kuva 4 on samankaltainen kuvan 3 kanssa, mutta esitysmuoto on kerroksittain. Ylimmällä tasolla näkyvät yrityksen työntekijöiden käyttämät tietojärjestelmät (esim. asiakasrekisteri). Näistä tieto luetaan ETL-prosessiin, josta se taas voidaan lukea operatiiviseen tietokantaan (välitietokanta), tietovarastoon tai suoraan paikallisvarastoon. Kuvassa riippumaton ”Data Mart” on siis olemassa jopa ilman tieto-

varastoa. Kuvan 4 alimmaiselta tasolta löytyvät kaikki paikallisvarastot (eng. Data Mart), jotka on myös pilkottu tietovarastosta ja täten riippuvaisia sen tiedoista.



Kuva 4. Tietovarasto-prosessin tasot [DW Glossary, 2007].

DM Review [1998] mainitsee, että paikallisvarasto ei kuitenkaan ole sama asia kuin tietovarasto. Siinä määritellään kaksi paikallisvaraston tyyppiä, joista toinen on riippuvainen tietovarastosta ja käyttää sen dataa, kuten kuva 4 havainnollistaa. Jokainen yksikkö yrityksessä voi suorittaa oman riippumattoman paikallisvarastoprojektinsa, jolloin yhden IT-henkilön tai ryhmän vastuu pienentyy ja projektin sekä ETL-prosessin onnistuminen paranee. Tietohallinnon johtajan täytyy olla suunnitelmallinen, jos paikallisvarastot halutaan yhdistää myöhemmin koko yritystä koskevaan tietovarastoon. Ruohonen ja Salmela [1999] mainitsevat, että tietohallinnon johto toteuttaa tällöin ennalta määriteltyä yrityksen tietohallintostrategiaa. Paikallisvarastojen ollessa täysin erilaisia saman yrityksen sisällä, ne eivät palvele koko yrityksen strategiaa myöhemmästä yhdistämisestä. Paikallisvarastojen ollessa valmiita voidaan myöhemmin tietovarastoprojektissa käyttää näitä hyväksi, jolloin projektissa säästetään huomattavasti resursseja. Tällöin taas otetaan käyttöön ETL-prosessi, jolla ne yhdistetään toisiinsa.

Paikallisvarastot saattavat sisältää itsessään vielä lisää pienempiä paikallisvarastoja joita käytetään hienojakoisempaan lajitteluun vielä osaston sisällä. Paikallisvarastotyyppinen ajattelu monimutkaistuu siinä vaiheessa, kun yksittäisten paikallisvarastojen pitää lukea tietoja toisistaan, jolloin tietovaraston rakentaminen on paljon hyödyllisempää. DM Review [1998] esittää kuitenkin vahvoja kommentteja siitä, että paikallisvarasto ja tietovarastoympäristöt ovat niin erilaiset, että on turha ajatella muuttaa paikallisvaraston tietokantaa tietovarastoksi, kun se on saavuttanut tietyn koon. Samoin Hovi [1997] esittää kritiikkiä siitä, että paikallisvarastot voivat olla epäyhteensopivia keskenään.

2.1.3 Tietovarasto vai paikallisvarasto?

Tietovarastoja suunniteltaessa voidaan käyttää jo olemassa olevia paikallisvarastoja hyödyksi. Edellisessä kappaleessa mainitsin, että paikallisvarastot voivat olla olemassa ennen tietovarastoa, joten kaikkea paikallisvarastojen sisältämää tietoa ei tarvitse heittää hukkaan. Tätä koskien tietovaraston suunnittelussa katsotaan olevan kaksi yleistä koulukuntaa [DWH, 2013]. Näiden kehittäjiä pidetään myös tietovarastojen ”isinä”, kuten Exforsys [2005] ja DM Review [1998] mainitsevat. Näistä toista edustaa Ralph Kimball ja Kimball Group [Kimball Group, 2013]. He kannattavat alhaalta-ylös (eng, bottom-up) suunnittelua, jossa ensin tulevat paikallisvarastot ja näiden jälkeen luodaan vasta tietovarastot. Paikallisvarastojen suunnitteluun käytetty aika käytetään myöhemmin hyödyksi, jolloin tietovaraston suunnittelu tapahtuu nopeammin ja yritys saa paikallisvarastojen hyödyt käyttöönsä tällöin myös nopeammin. Paikallisvarastot sisältävät jo tiedot kaikista niistä objekteista mitä eri yrityksen osastoilla on asiakkaista. Näitä ei ole vain vielä yhdistetty muihin tietoihin [DWH, 2013].

Bill Imnon edustaa toista koulukuntaa, jonka mielestä ylhäältä-alas (eng. top-down) ajattelu on parempi ratkaisu [DWH, 2013], [DM Review, 1998]. Tässä tietovarasto rakennetaan ensin hyödyntäen koko yrityksen tietoja asiakkaista. Tavoitteen ollessa paikallisvarastojen toteuttaminen, projekti tulee olemaan pitkäkestoisempi, mutta tietovaraston ollessa valmiina on paikallisvarastojen pilkkominen taas suoraviivaisempaa. Tästä syystä yrityksen tietojärjestelmiä suunniteltaessa kannattaa miettiä otetaanko lähtökohdaksi ensin tietovarasto vai paikallisvarasto.

Exforsys [2005] mainitsee vielä kolmannen vaihtoehdon, joka on "Hybrid design". ”Hybrid design” on paras tapa silloin, jos käytössä on jo esimerkiksi asiakkuudenhallintajärjestelmä, joka kerää tietoja myyjiltä ja asiakaspalvelulta. Tällöin voidaan lukea tätä tietoa suoraan näistä kannoista, joissa se on jo järjestetty jollakin avaimella ja mahdollisesti aikaleimoilla. Tässä tapauksessa tiedot viedään välivarastoon, johon ne tallennetaan ja järjestellään väliaikaisesti, sekä muokataan ennen siirtoa toiseen tietokantaan.

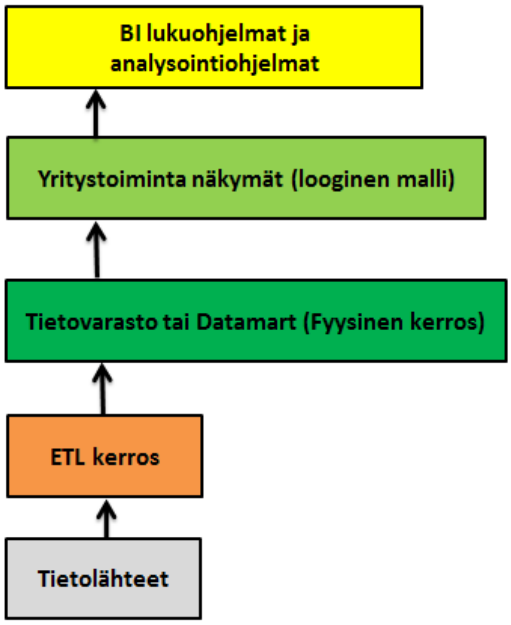
2.2 Liiketoimintatiedon hallinta

Liiketoimintatiedon hallinnan englanninkielistä termiä Business Intelligence (lyh. BI) käytti ensimmäisen kerran IBM:n tutkija Hans Peter Luhn [1958]. Tässä tutkielmassa viittaan tähän jatkossa lyhenteellä BI. Tuohon aikaan tietotekniikka ei ollut niin kehittynyttä, että sitä olisi voitu käyttää yrityksen ratkaisuissa. Tietovarastojen luomisen jälkeen, grafiikan parantuessa ja laskentatehon kasvaessa BI koki kuitenkin uuden nousun. BI-ratkaisuilla tulee olemaan tutkimusyhtiö Gartnerin mukaan 7 prosentin nousu vuonna 2013 [Gartner, 2013a].

Kaikkea sitä tietoa mitä tietovarastosta löytyy, voidaan käyttää yrityksen johtajien ja päättäjien avuksi, jotta he pystyisivät tekemään tarkkoja suunnitelmia yrityksen tulevaisuudesta [Morris, 2008], [Mundy et al., 2006]. Tietovaraston tiedoilla pystyy myös tekemään suunnitelmia tulevista toiminnoista, ennustamaan näiden toimivuutta tai katsomaan historiatiedoista miten ne ovat toimineet. Näitä suunnitelmia ja tuloksia voidaan seurata ainakin BI-työkaluilla. Data-analyysin tekeminen suoraan kohdetietokannoista ei ole aina mahdollista, jos tieto on niissä hajautettuna, vaan BI-ohjelman tietojen lähteenä kannattaa olla tietovarasto. Khan ja muut [2012] mainitsevat, että ilman tietovarastoa BI-analyysin tekemisessä 80 prosenttia ajasta menee tiedon muokkaamiseen.

BI on laaja käsite joka kattaa useita eri osa-alueita. Kokonaisuudessaan hyvä BI-järjestelmä yhdistää yrityksessä olevaa raakaa dataa eri toimintojen avulla helposti luettavaksi analyysiksi tai grafiikaksi. Datalla tässä tarkoitetaan tietovarastossa olevaa dataa, koska se sisältää parhaan ja lajitelluimman tiedon yrityksen eri tietojärjestelmistä. Tämä tieto voidaan lukea BI-työkaluun helposti ja kattavasti. BI-ohjelmistoja on markkinoilla useita. Työkalut toimivat itsenäisinä ohjelmistoina tai sitten ne ovat lisäosia joissakin moderneissa CRM ja ERP -järjestelmissä. Lisäosissa on puutteena, että ne

pystyvät lukemaan vain tuon järjestelmän dataa. Henry ja muut [2005] mainitsevat teoksessaan, että ETL-prosessi on elämänlanka mille tahansa BI-ratkaisulle. Kuva 5 esittää eri kerrokset tietolähteestä BI-lukuohjelmaan. Toiminta lähtee liikkeelle tietolähteistä, jotka ovat yrityksessä päivittäisiä tiedon keräämiseen tarkoitettuja järjestelmiä. Näistä tiedot noudetaan ETL-prosessin kautta ja siirretään tietovarastoon tai paikallisvarastoihin (Datamart). Tästä tiedosta voidaan kerätä ja yhdistää yritystoiminnan näkymässä (eng. business view) tietoja esimerkiksi yrityksen eri osastoista. Näkymät voivat sijaita tietovaraston tietokannassa jota luetaan BI-lukuohjelmilla.

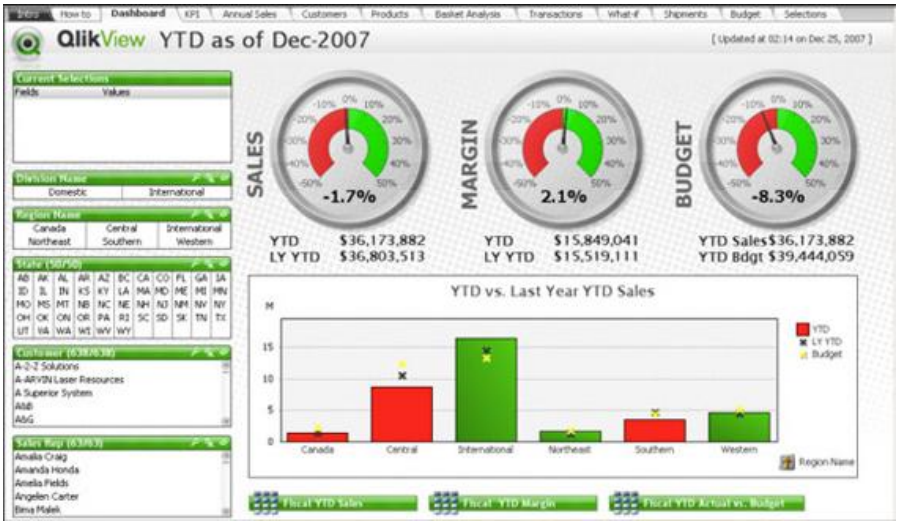


Kuva 5. Business Intelligence -arkkitehtuuri [Wu, 2007].

BI on tietovaraston jälkeen luonnollisesti seuraavana projektina tietohallinnon listalla, koska tietovarasto sisältää paljon sellaista dataa, joka olisi sellaisenaan hyödyllistä vain koodaajalle. BI-työkalut esittävät tämän datan muodossa, jota on helppo lukea ja hakea ilman laajaa ymmärrystä tietotekniikasta. Yksi esimerkki BI-ohjelmasta on QlickView, joka näyttää tiedon helposti luettavana "kojelautana" (eng. dashboard) [QlickView, 2013]. Kojelautanäkymä onkin kopioitu moneen muuhun BI-ohjelmaan hyvän luettavuuden ansiosta. Tämän näkymän voi liittää muihin yrityksen järjestelmiin, tulostaa tai liittää esimerkiksi PowerPoint esitykseen. Kuvassa 6 on nähtävillä kuva QlickView ko-

jelaudasta. Siinä näytetään valitusta tiedosta (kuvassa joulukuu 2007) mm. myynti, budjetti ja vertailu edelliseen vuoteen samaan ajankohtaan.

Mundy ja muut [2006] mainitsevat, että BI-ympäristön käyttö ja sujuva hallinta ei olisi helppoa ilman hyvin rakennettua tietovarastoa. ETL-prosessi auttaa tietovaraston rakentamisessa jotta siitä saadaan sellainen, että BI-järjestelmä pystyy hyötymään siitä. Henryn ja muiden [2005] mukaan ETL- ja tietovarastoprojekti on jo 60-80 % koko BI-projektista. Isoin osa työstä on siis tehty jo taustalla ETL-ajolla, jolla on yhdistelty eri tietolähteet samaan tietokantaan jota BI-ohjelma lukee reaaliaikaisesti.



Kuva 6. QlickView-kojelauta [QlickView, 2013].

2.3 Tiedonlouhinta

Tiedonlouhinta tarkoittaa prosessia jolla haetaan oleellinen tieto tietovarastosta. Mundy ja muut [2006] mainitsevat, että tiedonlouhinnan pääperiaate on löytää tunnistettavia kaavoja etsittävästä tiedosta. Siihen voidaan käyttää useita eri louhintamenetelmiä, analyyseja ja matematiikkaa. Pelkkä tietovaraston olemassaolo ei riitä tiedon käsittelemiseen mahdollisimman tehokkaasti.

Aroran ja muiden [2009] mukaan tiedonlouhintaa käyttävät yritykset ovat usein erikoistuneet jälleenmyyntiin, talouteen, kommunikaatioon tai markkinointiin. Otetaan taas esimerkkinä asiakkuudenhallintajärjestelmä. Siinä tiedonlouhinnan käytössä voidaan törmätä skenaarioihin, jossa myyjien aikaa ei haluta käyttää ottamaan yhteyttä mahdollisiin asiakkaisiin, jotka eivät ole todennäköisesti tekemässä mitään hankintoja, joita

yrityksen myyjät yrittävät heille kaupata. Tällöin voidaan tehokkaasti hyödyntää tiedonlouhintaa ja algoritmeja, jotka löytävät mahdollisesti ne asiakkaat joilla olisi huomattu tarve ja joihin on tärkeä ottaa yhteyttä pikaisesti. Näin säästetään työntekijöiden aikaa, joka samalla tarkoittaa säästöjä kustannuksissa ja lisää yrityksen tuloja silloin, kun datalouhinta on onnistunutta.

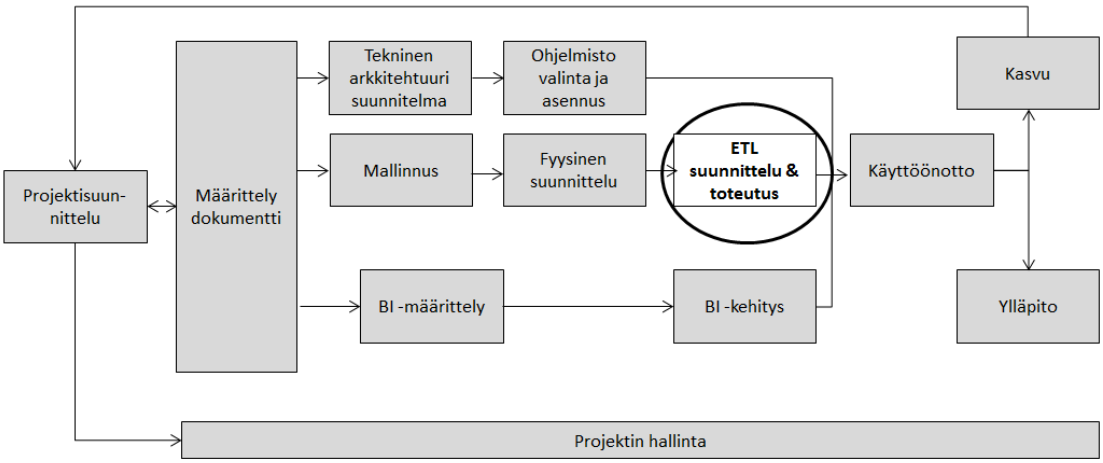
Paikallisvarastoja voi olla yrityksessä useita, jokaisella osastolla omansa. Sama pätee myös tiedonlouhintamalleihin. Mundy ja muut [2006] mainitsevat, että jokaisella osastolla yrityksessä voi olla oma tiedonlouhintamallinsa, joita he itse ylläpitävät. Tähän pätee myös sama toteamus kuin paikallisvarastojen rakentamisessa, että yrityksen osasto itsessään tietää parhaiten miten he haluavat tietoja käyttää ja miten niitä pitäisi louhia. Osaston itsessään tehdessä kehitystyön he saavat myös parhaan lopputuloksen.

Tiedonlouhintaa käytetään tänä päivänä useissa paikoissa. Esimerkiksi vähittäistavarakaupan asiakkaan ostoskäyttäytymistä seurataan etukorttien avulla. Nyt kiinnostuneet saavat selville esimerkiksi asiakkaan puolen vuoden ostot, jolloin yrityksellä on erittäin hyvät mahdollisuudet ennustaa mitä asioita asiakkaalle pitäisi markkinoida suoraan, jotta saadaan mahdollisimman hyvä palaute markkinointibudjetille [Ruohonen & Salmela, 1999].

2.4 ETL

ETL-prosessi on osa tietovaraston ja muiden suurien tietokokonaisuuksien luomista [Mundy et al, 2006]. ETL on yksinkertaistettuna prosessi, jossa lähdetietokannasta kerätään tarvittavat tiedot, muokataan tiedot sopivaksi ja viedään lopuksi tiedot kohdetietokantaan. Mundy ja muut [2006] mainitsevat, että ETL-prosessi suoritetaan yleensä eräajona kerran yössä jos tietomäärät ovat suuria. Muutoin ne voi ajaa osissa useita kertoja päivässä. Tietomäärä on kuitenkin niin suuri, että se vaikuttaa käyttäjien toimintaan työympäristössä, koska ajo aiheuttaa hidastelua tietojärjestelmissä joista tietoa luetaan. Tämä pitää ottaa huomioon kehitystyössä [Mundy et al, 2006]. ETL-prosessilla luodaan tietovarasto, mutta myös päivitetään sitä jatkuvasti. ETL-prosessin kehittäminen ei pääty tietovarastoprojektin päättymiseen vaan se on jatkuvan kehityksen ja ylläpidon alaisena. Kaikki ETL-prosessit ovat omanlaisiaan ja ne vastaavat sen yrityksen tarpeita, joka prosessin tilaa. Morris [2008] mainitsee, että ETL-projektissa voidaan käyttää ai-

empaa kokemusta esimerkiksi tietystä ERP-järjestelmästä, mutta yritysten arkkitehtuurit ovat sen verran erilaisia, että niihin ei voida suoraan käyttää mitään valmista pakettia.



Kuva 7. ETL-prosessin sijainti tietovarasto ja BI projektissa. [Mundy et al., 2006].

Kuva 7 havainnollistaa koko BI-projektin elinkaarta. Kuvasta näkyy ETL-prosessin sijainti verrattuna muuhun määrittelyyn. BI-projekti alkaa suunnittelusta ja koko projektin ajan projektia hallitaan projektin johdon tasolta. Projektissa kirjoitetaan määrittelydokumentti, joka esittää miten BI-työkalu tulee toimimaan ja mitkä sen tavoitteet ovat. Yleisen suunnitelman jälkeen ETL-prosessi voidaan suunnitella ja rakentaa. ETL-prosessilla tiedot viedään tietovarastoon [Mundy et al.,2006]. BI-työkalun kehitys jatkuu tämän rinnalla. Käyttöönotossa BI-työkalulla luetaan tietovarastossa olevia tietoja ja ylläpito alkaa. BI-työkalun ja ETL-prosessin kehitys on jatkuvaa. Järjestelmän piiriin voidaan tuoda uusia tietolähteitä jolloin työkalusta saadaan enemmän hyötyjä yritykselle.

ETL-prosessissa ”Extract” (lyh. E) tarkoittaa tiedon noutamista yrityksen tietokannoista tai muista lähteistä, missä tietoa sijaitsee. Dataa voidaan hakea mistä tahansa lähteestä joka on yrityksen tietoverkon sisällä ja johon on lukuoikeudet. Lähde voi olla myös internetin kautta saatavilla. Lähteitä voivat olla esimerkiksi eri relaatiotietokannat, taulukkolaskennan ohjelmat, osoitekirjat, henkilörekisterit, internetsivustot ja webpalvelut. Extract-vaihe on tärkeä, koska se vaikuttaa kaikkiin tuleviin vaiheisiin joten on tarpeen, että tiedot haetaan tässä vaiheessa oikein ja, että kaikki tieto on saatavilla. Extract-

vaiheen epäonnistuessa muut tämän jälkeen tulevat vaiheet todennäköisesti eivät pysty korjaamaan tämän vaiheen virheitä.

”Transform” (lyh. T) -vaihe on se missä suurin osa työstä tapahtuu. Siinä käydään läpi kaikki tuodut tiedot. Tietojen läpi käynti aloitetaan siivoamisella. Tiedoista pitää poistaa esimerkiksi Null-arvot, tehdä tyyppimuunnokset, muuttaa tiedot kohdetietokannan ymmärtämään muotoon, suorittaa tiedon täydentäminen eli mahdollinen lukeminen toisesta tietokannasta. Vaiheessa voidaan myös verrata muistiin luettua tietoa ja tarkistaa, onko tietoa olemassa objektista tai esimerkiksi täydentää tietoa lähteestä. Transform-vaiheessa tehdään merkkijonon muutokset tai suoritetaan aritmeettisia laskuoperaatioita. Tietoa voidaan järjestää ja yhdistää, sekä tallentaa muuttuneet tiedot muistiin.

Tietovirrasta otetaan pois esimerkiksi duplikaattiarvot jotka tarkoittavat lähdetietokannassa esiintyvää samaa objektia kahteen kertaan. Duplikaatti voi olla tietokannassa jopa täysin samoilla tiedoilla tai hiukan eriävästi. Duplikaatteja syntyy lähdetietokantaan siinä tapauksessa, jos niitä ei tarkisteta ennen uuden luontia manuaalisesti tai automaattisesti. Duplikaatteja voi syntyä myös aiemmin tehdyn tietokantojen yhdistämisen johdosta tai muissa ETL-prosesseissa.

Transform-vaiheessa tiedossa hallinnoidaan myös ”likainen data”. Kuten Arora ja muut [2009] mainitsevat, likainen data on tietokannasta löytyvää väärää, epäjohdonmukaista ja ei haluttua tietoa. Likainen data tulee ETL-prosessiin lähdejärjestelmistä. Transform-vaiheessa tällöin tullaan siivoamaan data, jotta se on tarpeeksi hyvää varastoitavaksi.

Transform-vaihe on erittäin tehokas ja riippuen käytetystä ETL-työkalusta siinä voidaan tehdä tiedolle paljonkin muutoksia ja tarkistuksia. Suunnittelu on kuitenkin tärkeää, sillä virheet voivat pidentää tiedon muuttamiseen tarvittavaa aikaa eksponentiaalisesti. Virheet tässä vaiheessa vaikuttavat seuraavan vaiheen (Load) läpimenoon, sekä tiedon eheyteen luettaessa sitä kohdetietokannasta (tietovarastosta).

”Load” (lyh. L) -vaihe tarkoittaa muutettujen tietojen kohdetietokantaan siirtämistä, joka voi olla minkä tahansa tyyppinen tietokanta, yleensä kuitenkin relaatiotietokanta kuten tietovarasto [Mundy et al., 2006]. Yksinkertaisimmassa tapauksessa tämä tapahtuu suoraviivaisesti, jolloin prosessille kerrotaan, mikä tieto sijoitetaan kohteessa mihinkin kenttään. Suunnitteluvaiheessa määritellään ollaanko tietoja ylikirjoittamassa vai

halutaanko säilöä tieto siitä, mikä arvo kentässä on ennen muutoksia. Pelkkä ylikirjoitus ei vaadi muita toimenpiteitä. Yleensä kuitenkin halutaan pitää historiatietoja muistissa, joka vaatii historiataulun lisäämistä tietokantaan ja avaimet joista tiedetään, mikä arvo kentässä on mihinkin aikaan ollut. Mundy ja muut [2006] mainitsevat, että tällöin jokaiseen muutokseen pitää laittaa aikaleima, jotta pystytään seuraamaan muutoksia. Tästä lisää luvussa 4.

Kaikki edellä mainittu liittyy yleisesti tiedon eheyteen ja on tärkeää, että ETL-prosessi on hyvin suunniteltu, jotta sen kohteena ollutta tietokantaa voidaan myöhemmin jatkojalostaa esimerkiksi BI-analyysiin.

2.5 ETL vai ELT?

Ei voida sanoa, että ETL-prosessi olisi ainoa oikea ratkaisu kun kehitetään tietovarastoa ja käsitellään sen informaatiota. On olemassa myös muita koulukuntia joita eri työkalujen valmistajat kannattavat ja ajavat eteenpäin. Esimerkiksi Oracle on siirtymässä enemmän ELT-ajatteluun, mikä tarkoittaa, että ladataan tiedot ensin tietovarastoon ja sen jälkeen vasta käsitellään tietoja. On olemassa myös ELTL-koulukunta, jossa ajatellaan, että tietovarastossa käsitellyn tiedon voi taas siirtää eteenpäin. Kaikilla tavoilla on hyvät ja huonot puolensa ja taulukossa 1 on lista niistä molemmista. [Davenport, 2008]

ETL		**ELT ja ELTL**	
Hyvät +	**Huonot -**	**Hyvät +**	**Huonot -**
Nopea kehitystyö	Ei joustava, kun tarvitsee lisätä uutta	Pystytään purkamaan projekti osiin	Ei ole yleinen tapa
Voidaan keskittyä tiettyyn alueeseen	Tarvitsee paljon tehoa palvelimelta	Transform-vaiheeseen pystytään puuttumaan myöhemmin	Kehitystyökaluja ei saatavilla paljoakaan
Paljon työkaluja saatavilla	Korkea oppimiskäyrä	Vaiheet (E,L,T) voidaan erotella.	

Taulukko 1. ETL vai ELT? [Davenport, 2008].

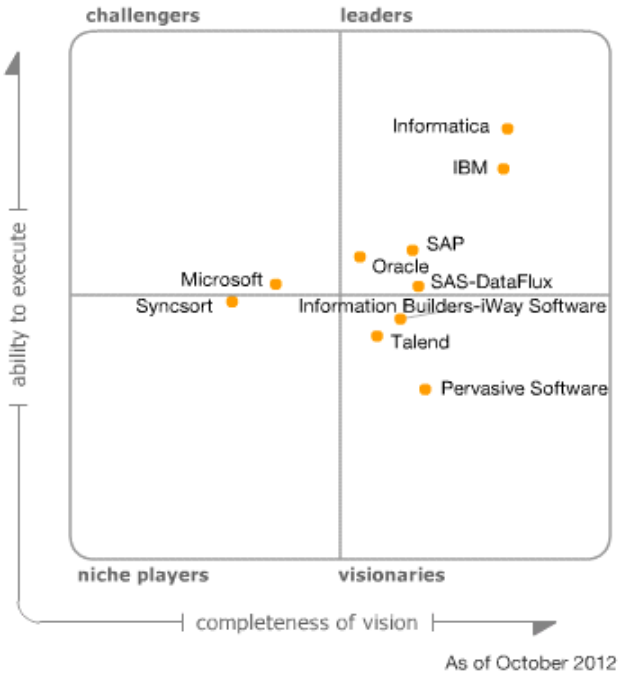
Taulukosta 1 huomataan, että ETL on yleisesti käytössä joten siihen on saatavilla useampia työkaluja ja taustatietoa. ELT ja ELTL ovat innovatiivisia seuraavia tapoja toteuttaa ETL. Tällä hetkellä ELT ja ELTL ovat toimivia vain tietyissä projekteissa ja työkaluja näiden hallinnoimiseen ei vielä ole paljonkaan.

2.6 Esimerkkejä ETL-työkaluista

Kurukunda [2013] mainitsee, että ETL-työkaluja on markkinoilla useita. Internetistä löytyy myös ilmaisia avoimeen lähdekoodiin perustuvia työkaluja, joten alkuun pääseminen on helppoa. Ominaisuuksiltaan ne eivät kuitenkaan vastaa kaupallisten työkalujen toiminnallisuutta ja niiden ominaisuuksien laajuutta. Seuraavissa kappaleissa on mainittu muutamia kaupallisia työkaluja ja listattuna niiden parhaita ominaisuuksia. Käytän ohjelmistojen vertailuun tutkimulaitos Gartnerin [2013b] analyytikkojen mainitsemia hyviä ja huonoja ominaisuuksia eri ohjelmista.

Gartner [2013b] listaa joka vuosi dataintegraatiotyökaluja ja analysoi niiden heikkoudet ja vahvuudet. Kuvassa 8 on nähtävissä kaavio, jossa on mm. kaikki esittelemäni kaupalliset työkalut edustettuina. Vaaka-akselilla määritellään kuinka paljon uusia innovaatioita yritykset ovat kehittäneet työkaluihinsa. Pystyakseli määrittelee mikä on mahdollisuus toteuttaa haluamansa prosessi näillä työkaluilla. Informatica ja IBM ovat taulukossa kärkipäässä ja niitä kuvataan täten innovoiviksi ja tehokkaiksi työkaluiksi. Microsoftin työkalu on jäänyt perustyökaluksi ja puoliväliin. Esittelen syitä näihin kunkin työkalun kohdalla.

Kaikki mainitut ETL-prosessin luomiseen käytettävät työkalut sisältävät graafisen käyttöliittymän, jossa voidaan luoda ETL-prosessin peruspaketti tarvitsematta osata ohjelmointikieliä. Paketilla tarkoitetaan ETL-prosessin ajamiseen tarkoitettuja suoritettavia ohjelmia ja konfigurointi tiedostoja (lisätietoa kappaleessa 4.8.2). Toisin sanoen kuka tahansa voisi tehdä peruspaketin, jossa haetaan tieto tietokannasta ja viedään se toiseen. Dokumentointia seuraamalla pystytään tekemään ETL-prosessi yksinkertaisiin tarpeisiin.



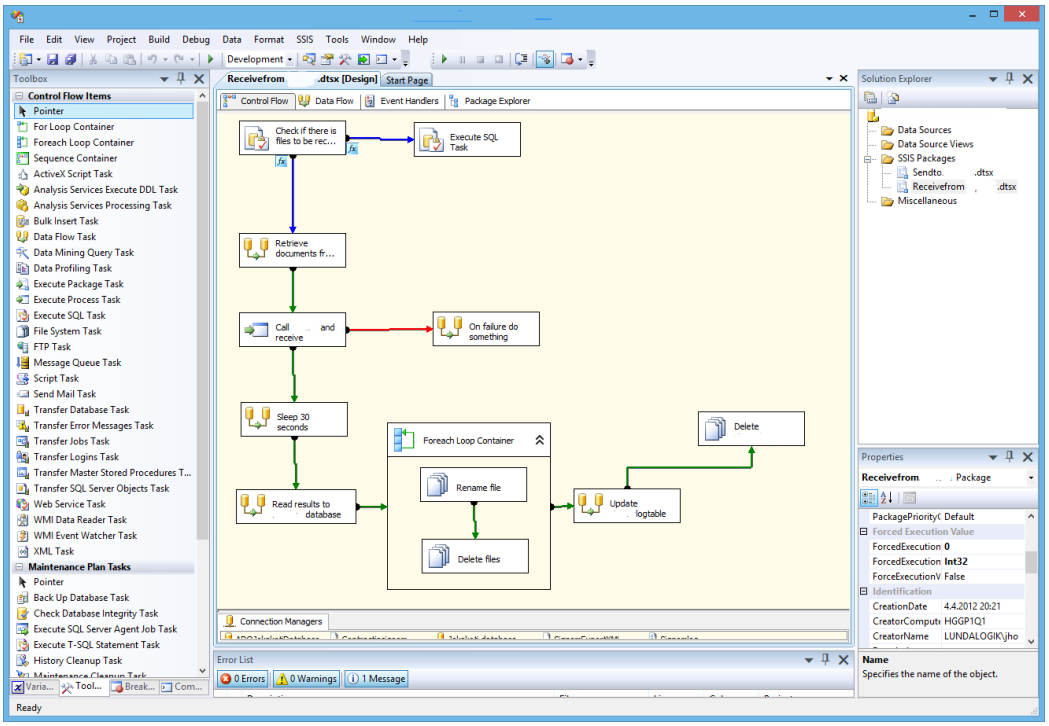
Kuva 8. Gartnerin “Magic Quadrant” Data integraatio-työkaluille [Gartner, 2013b].

2.6.1 Microsoft SQL Server Integration Services

Gartner [2013b] löysi seuraavat vahvuudet. Työkalun perusominaisuudet ovat hyvin hallussa, sitä on helppo käyttää, nopea kehittää ratkaisuja ja se on helppo integroida Microsoftin SQL Server ratkaisuihin. Järjestelmän maine ja mahdollisuuksien laajuus on hyvä. Microsoft yrityksenä on hyvässä kunnossa ja päivittää jatkuvasti järjestelmään parannuksia.

Toisaalta järjestelmästä on löydetty myös heikkouksia. Microsoft kulkee omia polkujaan, eikä sillä ole samaa visiota datan integroimisesta kuin muilla. Joitakin uusia tekniikoita ja tapoja on jäänyt puuttumaan. Datapaketit toimivat vain Windows ympäristössä mikä haittaa eri käyttöjärjestelmien kirjoa yrityksessä.

Yleisesti ohjelmasta voidaan mainita seuraavaa. Yrityksessä ollessa käytössä Microsoft-ympäristö voidaan ottaa käyttöön Microsoftin oma työkalu. Se tulee uusimman SQL Server ohjelmiston kylkiäisenä, joten se on jo valmiiksi ostettu mikäli yritys omistaa kyseisen ohjelmiston. Toiminta ja ajo tapahtuu tekemällä SSIS-paketteja jotka voidaan ajastetusti ajaa SQL palvelimella työjonossa. Ohjelmointityö tapahtuu SQL Server Business Intelligence työkalulla, josta on esimerkki kuvassa 9. Kun kyseessä on Microsoftin työkalu, niin voi hyödyntää kattavasti myös muita Windows ohjelmia, SQL työkalua ja ohjelmointikieliä [Mundy et al., 2006].



Kuva 9. Microsoft Business Intelligence työkalu.

2.6.2 IBM Information Server

Gartner [2013b] esittää seuraavia vahvuuksia IBM:n tuotteessa. Ohjelmisto sisältää laajan määrän erilaisia tapoja suorittaa dataintegraatioita. IBM:n asiakkaat löytävät sopivan työkalun erilaisille projektityypeille. Työkalulla voidaan hallita suurempia ja monimutkaisempia projekteja kuin muilla kilpailevilla työkaluilla. Työkalujen eri osat keskustelevat hyvin keskenään ja näistä löytyy ainakin yksi työkalu, joka sopii jokaiseen tarpeeseen tai ongelmaan.

Työkalujen määrä kuuluu myös yhdeksi heikkoudeksi jonka Gartner löysi ohjelmistosta. Työkalujen määrä monimutkaistaa ohjelmiston käyttöä. Työkalu on myös vaikeampi

oppia ja käytettävyys ei ole niin hyvä. Ohjelmisto on kallis ja hinnoittelu on monimutkainen, jolloin pienellä yrityksellä ei ole varaa saada koko järjestelmää käyttöönsä vaan vain osa työkaluista.

Yleisesti ohjelmistosta voidaan mainita seuraavaa. IBM Information Server koostuu useista eri työkalusta, jotka yhteen koottuna suorittavat koko ETL-prosessin. Esimerkiksi IBM QualityStage auttaa hakemaan lähdetietokannoista tietoa ja myös käsittelemään tätä metadataa, sekä analysoimaan enemmän graafisesti lähdetietokannoista löytyvää tietoa. IBM DataStage on puolestaan ohjelma, mikä käsittelee Transform ja Load -vaiheita [IBM, 2013].

Kummatkin QualityStage ja DataStage toimivat Information Serverin graafisessa käyttöliittymässä, joka näyttää tietoprosessin samanlaisena putkena kuin aiempi Microsoftin tuote. Information Serverin huonona puolena on juuri se, että se koostuu useista eri komponenteista. Jos halutaan rakentaa erittäin tehokas tietovarasto ja BI-prosessi, täytyy hankkia useampi ohjelma. Tämä taas vie rahaa budjetista. Myös nämä ohjelmat ovat huonosti yhteensopivia muiden valmistajien ohjelmistojen kanssa, joten niitä ei voi helposti yhdistää muihin vaan ne keskustelevat vain toistensa kanssa [IBM, 2013].

2.6.3 Informatica Powercenter

Gartner [2013b] löytää seuraavia vahvuuksia Informatican järjestelmästä. Siitä löytyy tuki kaikille data integraatio tavoille, jotka ovat tällä hetkellä yleisesti käytössä. Järjestelmä vastaa hyvin koko ajan kehittyviä tarpeita yritysmaailmassa. Informatica yhdistelee eri integraatiotyylejä siten, että asiakkaat saavat synergiaetuja. Työkalulla on pystytty kehittämään monenlaisia ja monimutkaisiakin projekteja. Informatican tuotestrategia seuraa trendejä ja viimeisimmät teknologiat päivitetään työkaluun.

Joitakin heikkouksiakin löytyy. Vaikka työkalu seuraa trendejä niin Informatica voi yrittää olla liian innovatiivinen ja kehittää työkaluun uusia ja ei niin suosittuja funktioita. Eri työkalujen välisessä integraatiossa on kehittämisen varaa ja muut Informatican työkalut pitäisi ottaa paremmin huomioon. Lähes puolet Informatican työkalun käyttäjistä sanoo, että työkalun hinnoittelussa on selventämisen varaa. Erilaiset lisäosat ovat kalliita ja jotta saa tarvitsemansa työkalun, voi joutua maksamaan enemmän kuin hinnasto näyttää.

Yleisesti Informatican järjestelmästä voidaan mainita seuraavaa. Se sisältää myös graafisen työkalun (Designer), jolla voidaan suorittaa kehitystyötä. Powercenter koostuu kahdesta eri ohjelmasta, jossa tietoprosessi suunnitellaan. Näistä toinen on ”Designer”, jossa koko ETL-prosessi tapahtuu. Designer on paikka, jossa luetaan data lähteistä, tehdään muunnokset ja kirjoitetaan ne kohdetietokantaan [ETL-tools].

2.7 Tutkimuksen lähtökohdat ja tavoitteet

Tämän tutkielman luvussa 4 käyn läpi ETL-prosessin suunnittelun ja luvussa 5 toteuttamisen. Aiemmin tässä luvussa mainitut asiat tietovarastoista, paikallisvarastoista ja BI-ohjelmistoista on tärkeää tietää ennen seuraavien lukujen lukemista koska tässä tutkielmassa viitataan usein ETL-prosessiin tietovaraston rakentamisessa. Lopuksi luvussa 6 esittelen parhaita käytäntöjä niihin kohtiin mitä on esitelty luvuissa 4 ja 5.

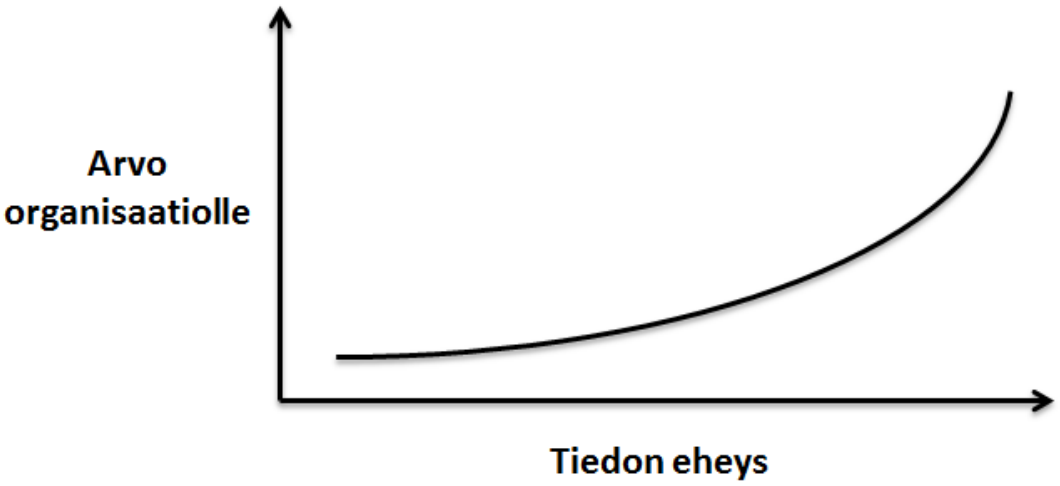
ETL-prosessiin liittyvät luvut on jaettu erilleen suunnittelu ja toteutusosioihin, koska suunnitteluvaihe on erittäin tärkeä myös ETL-projektissa [Mundy et al., 2006]. Suunnitteluvaiheessa käydään läpi toiminta yrityksen liiketoiminnan näkökulmasta ja valmistellaan tulevaa kehitystyötä. Kuten tavallisessakin ohjelmistoprojektissa, niin suunnitteluvaiheessa havaituilla asioilla voidaan suorittaa käytännön työ tehokkaammin. Suunnitteluvaihe sisältää enemmän teoriaa ja kartoitusta kun taas toteuttaminen sisältää varsinaisen ohjelmointityön.

Tutkimuksessa käyn läpi asioita oman työn reflektoinnin näkökulmasta. Tästä tutkimusmetodista löytyy enemmän tietoa luvussa 3. Oman työn reflektointi Case-analyysin tai muun metodin sijasta on valittu siksi, että voidaan kerätä tarvittavaa materiaalia käytännön työstä ja verrata niitä sitten luvussa 6 tavoitteisiin ja muiden hyviksi havaitsemiin käytäntöihin. Reflektointi auttaa vastaamaan kysymyksiin 1. Mitä, 2. Mitä sitten, 3. Mitä seuraavaksi, jota muissa metodeissa ei tällä tapaa käytetä.

Tavoitteena ETL-prosessin käyttöön ottamisella yrityksessä on saavuttaa tietty automaation taso tietojen integraatiossa. Siinä vaiheessa, kun tiedon hallinta manuaalisesti vie enemmän työntekijän aikaa ja yrityksen rahaa, kuin tuottaa hyötyjä niin ETL-prosessi kannattaa kehittää. ETL-prosessin automatisointi vapauttaa työvoimaa ja estää samalla inhimillisiä virheitä tiedonkäsittelyssä. ETL-prosessin käyttöönotto voi tuoda mukanaan myös hyötyjä yritykselle kun useat eri tietolähteet on integroitu yhteen tieto-

kantaan. Tällaisia ovat esimerkiksi kuvassa 10 näkyvä tietojen eheyden arvo yritykselle. Tiedon eheyden (eng. Information maturity) kasvaessa vaaka-akselilla tiedon arvo (eng. Value to organization) yritykselle lisääntyy.

ETL-prosessin käsittelyyn joutuu tulevaisuudessa valtava määrä tietoja. Esimerkiksi internet on paikka johon syötetään jatkuvasti uutta tekstiä, kuvia ja videota. Otetaan esimerkkinä Outlook.com, jossa Microsoft yhdisti eri sähköpostitilit uuteen Outlook.com palveluun. Tällöin käsittelyyn pääsi 150 terabittiä tietoja ja tietojen migraatio kesti 6 viikkoa [Outlook, 2013]



Kuva 10. Tiedon siisteyden arvo yritykselle [Khan et al., 2012]

Tutkimuksen tavoitteena on etsiä parhaita käytäntöjä ETL-prosessia, projektia ja tietovaraston rakentamista koskeviin osa-alueisiin. Näitä tavoitteita projektin aikana ovat esimerkiksi seuraavanlaisia. Projektin näkökulmasta on tärkeää pitää asiakas mukana projektissa jo suunnitteluvaiheessa. Projektin suunnittelu ennen varsinaista toteuttamista pitää hoitaa oikein. Projektin aikana peruskäyttäjää pitää häiritä mahdollisimman vähän. Valmista ohjelmaa pitää pystyä käyttämään myöhemmin ja myös jatkokehittämään. Ohjelman pitää myös toimia oikein ja luotettavasti.

Tietovaraston ja analysoinnin puolella on useita tavoitteita. Muutoksien tallentaminen pitää toteuttaa siten, että siitä saadaan yrityksen tavoitteen mukaista hyötyä. Tiedon tuonti pitää tapahtua nopeasti ja siten, että se ehditään tehdä ja siitä ei aiheudu haittaa

muille järjestelmille. Tieto pitää eheyttää ja täydentää, sekä duplikaattiarvot pitää poistaa.

Tietovaraston arvo ja tavoitteet yritykselle ovat myös omanlaisensa. Tietovarastossa olevasta tiedosta pitää nähdä mitä yrityksessä tapahtuu. Tietovarastossa olevaa tietoa pitää pystyä käyttämään muiden tietojärjestelmien hyväksi eheyttämään tietoa. Tietovarastossa oleva tieto voidaan lukea pohjaksi BI-ohjelmistoihin. Tietoa pitää myös pystyä käyttämään apuna kun tehdään yrityksen toimintaa koskevia päätöksiä.

Tutkimuksessa etsin parhaita käytäntöjä näihin tavoitteisiin ja esittelen niitä tuloksissa reflektoimalla kysymyksillä Mitä, Mitä sitten ja Mitä seuraavaksi.

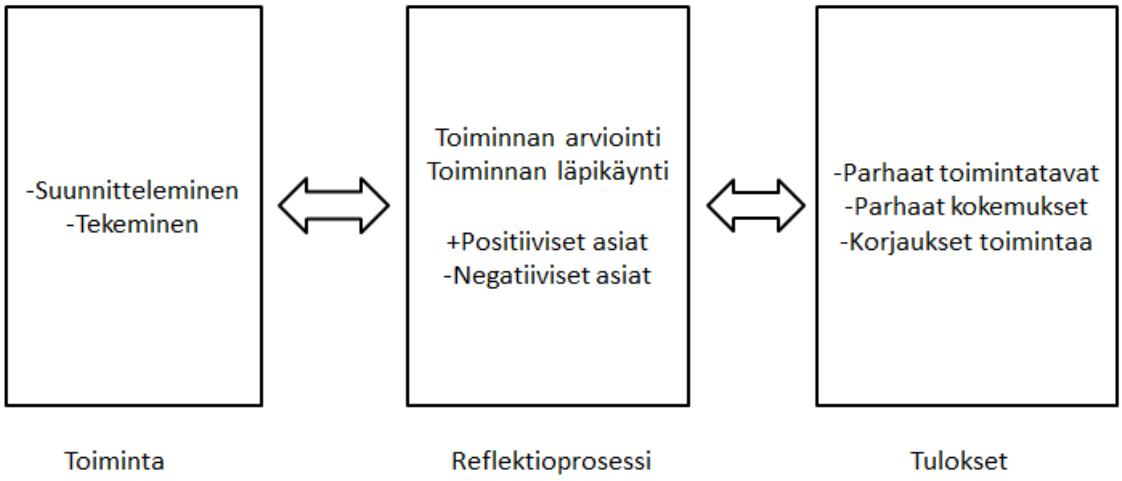
3. Tutkimusmetodi

Tutkimusmetodina tutkielmassa ongelmien ratkaisuun ja parhaiden käytäntöjen löytämiseen tulen käyttämään reflektiivistä käytäntöä (eng. reflective practice). Reflektiivinen käytäntö tarkoittaa oman työn analysointia ja siitä oppimista. Reflektiivisen käytännön käsitteen on ensimmäisenä määritellyt Donald Schön [1983]. Reflektiivisessä käytännössä käydään läpi kysymyksiä, kuten mitä tehtiin, miksi jokin asia tapahtui ja miten sitä voidaan parantaa [Bolton, 2001]. Tällä metodilla opitaan omasta tekemisestä sen sijaan, että kuullaan käytännöt oppitunnilla tai opettaja siirtää tietoa oppilaalle.

Tutkielmassa tulen esittämään asioita siten kuin olen huomannut niitä käytännön työssä ja testauksessa. Metodin avulla pystyn kirjoittamaan kokemuksesta, analysoimaan hyviä ja huonoja puolia, sekä ehdottamaan parhaita käytäntöjä ja korjauksia. Rekflektiivistä käytäntöä voi täten myös kuvata ”Tekemällä oppimiseksi”. Schön [1983] mainitsee, että reklektiivisen käytännön harjoittaja ajautuu jatkuvan oppimisen prosessiin (eng. continuous learning).

Reflektiivinen käytäntö on yleensä käytössä oppimiseen ja opetusympäristöön liittyen. Opettajat käyttävät sitä oman työnsä arviointiin opettajina. Reflektointi auttaa sen harjoittajaa kehittämään omaa päätöksentekoprosessia, tunnistamaan oppimistarpeita ja se pakottaa ottamaan selvää oman tietämyksen puutteista [Bolton, 2001]. Metodin käyttäjä voi käyttää reflektiota jo tekemisen aikana (eng. reflection in action) tai tekemisen jälkeen (eng. reflection upon action) [Schön, 1983]. Tässä tutkielmassa on keskitytty enemmän reflektioon tekemisen jälkeen, jolloin voidaan vertailla tuloksia siihen miten prosessia pitäisi tarkastella.

Kuva 11 esittelee reflektioprosessia. Reflektioprosessi toimii jatkuvasti. Se alkaa toiminnasta, jossa suunnitellaan tai tehdään. Tässä tutkielmassa suunnitellaan ja kehitetään ETL-prosessia. Toiminnan päätyttyä sitä arvioidaan ja listataan opittuja asioita. Toiminnassa on tullut ilmi sekä positiivisia, että negatiivisia asioita. Prosessin jälkeen ollaan selvillä tuloksista, eli on kerätty parhaat toimintatavat ja voidaan suodattaa pois virheellinen toiminta.



Kuva 11: Reflektioprosessi [Boyd et al., 1985].

Reflektiivinen käytäntö auttaa paljon työperäistä oppimista, koska siinä yritetään koko-ajan analysoida mitä on tehty ja opitaan siitä. Bolton [2001] kertoo, että jokapäiväisen työn ohessa saadaan kehityksellistä näkemystä omasta työstä.

Rolfe ja muut [2001] kehittivät aiemmin mainittua mallia ja kehitti kolme kysymystä jolla reflektiivistä käytäntöä voidaan yksinkertaisuudessaan harjoittaa. Täytyy kysyä itseltään kysymykset Mitä?, Mitä sitten?, Mitä seuraavaksi? (eng. What?, So What?, Now What?). Kun näihin kysymyksiin löytää vastauksen niin on päässyt ongelman ytimeen ja oppinut tekemästään.

4. ETL-Prosessin suunnittelu

ETL-prosessi pitää suunnitella hyvin, kuten kaikki tietojärjestelmäprojektit [Ruohonen & Salmela, 1999] ja ensimmäinen vaihe lähtee suunnittelusta yleisellä tasolla. Tulen seuraavissa alakappaleissa mukailemaan suunnittelujärjestystä jonka Mundy ja muut [2006] ovat todenneet hyödylliseksi. Tietovaraston suunnitteluprosessin järjestystä voidaan muokata projektin tarpeiden perusteella. Mundy ja muut [2006] mainitsevat, että suunnitelman lopputuloksena on kaikki mitä tarvitaan ETL-prosessin määrittelydokumenttia varten.

4.1 Graafinen yleiskuvaus

Aluksi täytyy miettiä asioita yleisellä tasolla ja saada kuva alkutilanteesta [Mundy et al., 2006]. On tärkeää myös ymmärtää mikä pitäisi olla projektin lopputulos. Tällöin voidaan piirtää yleissuunnitelma käyttäen prosessikaaviota. Tämän kaavion ei pitäisi olla valtava vaan nimenomaan yleiskuvaus aiheesta, jota voi helposti lukea. Yksi A4-kokoinen kuvaus riittää [Mundy et al., 2006].

Kuvauksella on tavoitteena useita eri asioita. Kuvaus pakottaa miettimään isoja kokonaisuuksia ja havainnoimaan mahdolliset ongelmat alkuvaiheessa. Kaaviosta näkee myös mikä on tietojenkäsittelyn järjestys. Tämä tarkoittaa esimerkiksi sitä, että ensin pitää saada joitain tietoja tietovarastoon, ennen toisen tiedon tuomista [Mundy et al., 2006]. Esimerkiksi toiminnanohjausjärjestelmän (ERP) tietoja tuodessa pitäisi tuoterivit tuoda ensin tietokantaan ennen laskutusrivejä, jolloin voidaan yhdistää laskut oikeisiin tuotteisiin. Näin jälkimmäinen pystytään tekemään yhdellä ja samalla tuontikerralla (ajoprosessin aika puolittuu). Tämä on mahdollista tehdä myös useassa erässä, jolloin tuodaan ensin kaikki objektit järjestelmään ja tämän jälkeen rakennetaan linkitykset. Se kuitenkin kasvattaa työmäärää mahdollisesti jopa moninkertaiseksi. Samoin jälkeenpäin tapahtuva ylläpito vaikeutuu, koska muutoksia pitää tehdä useassa eri paikassa.

Kuvaus toimii myös kommunikaatiovälineenä ETL-prosessin tilaajan ja kehittäjän välillä. ”Kuvausten ensimmäinen tehtävä on toimia yhteisenä kielenä järjestelmän kehittäjien ja tulevien käyttäjien välillä järjestelmää suunniteltaessa” [Ruohonen & Salmela, 1999]. Kuvauksesta tilaaja näkee, että prosessin rakentajat ovat ymmärtäneet tarpeet.

Kuvauksesta nähdään myös prosessin puutteet. Kuvauksen graafinen luonne on avuksi myös niille henkilöille, jotka eivät ole niin teknisiä kuin kehittäjät. Jos tilaajalla on vaikea ymmärtää materiaalia, jota heille toimitetaan niin silloin voi materiaali jäädä lukematta ja tilaaja ei ole samassa ymmärryksessä kehittäjien kanssa. Lopputulos ei silloin aina vastaa sitä mitä tilaaja on toivonut. Tilaaja päättää aina projektin onnistumisesta. Earls [2003] mainitsee, että asiakkaalta pitää osata kysyä oikeat kysymykset. Käyttäjätyytyväisyys on yksi tekijä jonka Ruohonen ja muut [1999] mainitsevat tietojärjestelmäprojektien onnistumisen tekijäksi.

Graafiseen kuvaukseen on kerätty kaikki lähteet ja objektit, jotka on tunnistettu käsiteltäviksi. Mundy ja muut [2006] mainitsevat, että kuvaukseen ei tarvitse kuvata tarkkaan mitä tiedolle tehdään sen jälkeen kun se on kerätty lähteistä. Yksi vaihekuvaus on esimerkiksi virke ”Suoritetaan tyyppimuunnokset”. Myöhemmässä vaiheessa voidaan tehdä tarkempi kuvaus joka koostuu esimerkiksi UML-kaavioista, josta esimerkki kuvassa 1. Näitä tarvitaan kun valmistellaan erityisen monimutkaista dataa vietäväksi järjestelmään.

4.2 Testitietokanta

ETL-prosessin suunnittelun ja toteuttamisen aikana tehdään paljon kyselyitä lähdetietokantoihin tai tiedostoihin. Yrityksissä nämä ovat työaikaan kovassa käytössä ja kyselyt tulisivat kuormittamaan kantoja sen verran, että käyttäjät voisivat huomata järjestelmissään hidastelua. Tästä syystä on parempi luoda kopio tietokannoista testiympäristöön johon kehittäjät voivat käydä tutustumassa ja testata toimintaa. Testiympäristö päivitetään joka yö varsinaisesta tietokannasta, joten viimeisin tieto olisi aina saatavilla kehitysvaiheessa. Yksinkertaisimmillaan testiympäristö rakennetaan ottamalla kopio lähdetietokannoista ja siirtämällä ne testiympäristöön.

Vaikka ETL-prosessi kehitettäisiinkin testiympäristöön jossa tietokannan nimet, palvelimen nimi tai polut eroavat varsinaisesta ympäristöstä, niin tämä ei haittaa käyttöönotossa. Ratkaisu on rakentaa konfigurointitiedostot, jotka sisältävät erot kehitysympäristössä ja tuotantoympäristössä. Mundy ja muut [2006] mainitsevat, että nämä tiedostot koostuvat esimerkiksi XML:stä.

Lopputuloksena on ympäristö, jossa kehitystä tehdään vapaasti välittämättä työajoista, mahdollisista yhteysongelmista ja tiedon saatavuudesta. Kehitysympäristössä työskentely ei vaadi kaiken työn tekemistä uudelleen tuotantoympäristössä vaan voi siirtää tuotetut ETL-prosessipaketit varsinaiseen ympäristöön. Tietokannan tietoja ei täten syväkoodata pakettiin, vaan kuten edellisessä kappaleessa on mainittu, niin käytetään konfigurointitiedostoja.

4.3 Objektien ja niiden välisten yhteyksien tunnistaminen

On hyvin tärkeää, että tiedetään mitä eri objekteja lähdetietokannoissa on ja miten ne ovat yhteydessä toisiinsa [Mundy et al., 2006]. Tarkempaan suunnitelmaan pitää selvittää myös vierasavaimet. Lähteen ollessa tietokanta voidaan yleensä sanoa, että yksi objekti vastaa aina yhtä taulua ja vierasavain arvot näyttävät eri yhteydet. Näin ei jokaisessa tapauksessa kuitenkaan ole. XML- tai taulukkolaskentaohjelmissa olevat arvot eivät ole yleensä eroteltu valmiiksi objekteihin vaan ne pitää erotella tiedostoista, mikä on hankalaa.

Tunnistamisen onnistumiseksi vaaditaan paljon dataan tutustumista. On tärkeää huomata, mitä tietoja on olemassa ja missä muodossa, jotta voidaan rakentaa oikeat vastaavuudet kohdetietokantaan. Objektit ja yhteydet kuvataan graafiseen kaavioon. Tätä käytetään ohjelmoinnin tukena ja sisäisen projektiryhmän suunnitelmassa. Lähde- ja kohdeobjektien selvittyä aloitetaan suunnittelemaan linkityksiä eri lähteiden ja kohteen välillä [Mundy et al., 2006]. Tämä tarkoittaa, että suunnitellaan kohdetietokannan taulut ja kentät, sekä linkitetään lähteistä löytyvät tiedot kohteeseen. Näin myös erotellaan tiedot, jotka tarvitsevat lisähuomiota myöhemmin mikäli ne ovat puutteellisia.

4.4 ETL-prosessin ajoaikataulun suunnittelu

ETL-prosessin ajoaikataulusta riippuu kuinka uutta tietoa tietovarasto sisältää. Mundy ja muut [2006] mainitsevat, että ETL-prosessi ajetaan hyvin usein eräajona kerran yössä, jolloin viimeisimmän työpäivän tiedot ovat tietovarastossa. Osa tiedoista on tarpeen ajaa päivittäin, ja toiset taas kerran kuussa. Tietojen ajamisen aikataulu on tärkeä tietää suunnitteluvaiheessa, koska se vaikuttaa suunnitelmaan. Esimerkiksi joitain tietoja voidaan sitoa kuukausittain saataviin tietoihin. Näitä on toiminnanohjausjärjestelmässä

(ERP) esimerkiksi laskut. Tietojen ajaminen yöllisinä eräajoina estää sen, että ETL-prosessi olisi niin laaja, että sen ajaminen kestäisi yli 10 tuntia [Mundy et al., 2006]. Yli 10 tunnin ajo ei ehtisi mennä läpi yhden yön aikana, jolloin tiedonsiirto ei ole valmis työntekijöiden saavuttua työpisteilleen.

Useasti ajettava ETL-prosessi vaatii enemmän optimointia, eli nopeutta, mikäli käsitellään terabiteittäin tietoa. Liiketoiminnasta riippuen on tarpeetonta ajaa tiedot useammin kuin kerran päivässä mikäli tiedot eivät ole paljon muuttuneet. Tällöin voi tehdä esitarkistuksia siitä ovatko tiedot muuttuneet. Tietoa ei myöskään ole aina saatavilla. Esimerkiksi, jos tarvitsee odottaa jotain muuta järjestelmää, jotta se saa ajettua tietoja ensin tietokantaan, kuten laskutusjärjestelmää, joka laskuttaa kerran kuussa.

Mundy ja muut [2006] mainitsevat, että yleensä yksi päivä on minimi ajanjakso jolla tiedot ajetaan tietokantaan käyttäen ETL-prosessia (edellisen päivän myynti). Poikkeuksia on, ja tietoja voidaan ajaa tietenkin useamminkin sisään tietovarastoon, mikäli tietomäärä pysyy pienenä. Reaaliaikaisesta tiedosta puhuttaessa ei yleensä kannata käyttää ETL-prosessia vaan tallentaa päivittyvät tiedot muulla tavoin. Tässä tutkielmassa on keskitytty tiedon eräajoihin.

4.5 Muutostietojen tallentaminen

Varsinkin BI-ratkaisut pohjautuvat vahvasti siihen tietoon, miten on edistytty aiemmasta tallennetusta tilanteesta [Moss & Atre, 2003]. Esimerkiksi asiakkaista kannattaa tallentaa asiakkuuden tilassa tapahtuneet muutokset. Muutoksia ei voi vertailla, jos aikaisempia tietoja ei ole talletettu muistiin. Tällöin ETL-projektissa on projektista riippuen hyvä tallentaa muutokset tietyistä kentistä aina muutoksen yhteydessä, jotta saadaan kattava kuva esimerkiksi asiakkuudesta. Yleisemmällä tasolla tämä auttaa saamaan kuvauksen yrityksen parantuneista tai heikentyneistä näkymistä ja niihin johtaneista syistä.

Tauluja joihin muutostiedot tallennetaan, kutsutaan aggregaateiksi. Kulkarni ja muut [2010] mainitsevat, että jokaisella taululla joiden muutoksia kerätään, on oma aggregaattitaulunsa/rivinsä. Näitä tauluja voidaan päivittää suoraan käyttäen ETL-prosessia tai tietokannan puolella funktiota. Kaikki tiedot tauluissa voivat olla jo valmiiksi aggregaatteja, jolloin ETL-prosessissa ei tarvitse kuin tarkastella rivien avaimia. On olemassa myös projekteja joissa historiatiedot eivät ole niin tärkeitä. Tällaisissa tapauksissa uutta

dataa ajettaessa on vain korvattava vanhat tiedot uusilla. Vanhat tiedot eivät jää tällöin järjestelmään muistiin, mikä estää muutoksien seuraamisen myöhemmin.

Tietojen talteen ottaminen aina muutoksen yhteydessä vaikuttaa kahteen suunnittelukriteeriin. Muutokset voi ensinnäkin tallentaa samaan tauluun, mutta piilottaa ne näkyvistä. Toiseksi voi tehdä jokaiselle taululle oma historia-taulu (aggregaatti), johon tieto tallennetaan muistiin. Taulujen välillä käytetään avaimia, joilla viitataan vanhoihin tietoihin ja saadaan ne tarvittaessa näkyviin. Näissä historiatiedoissa on tärkein argumentti myös aika, jolla saadaan muutokset näkyviin aikajärjestyksessä, jolloin niitä voidaan analysoida esimerkiksi BI-järjestelmässä. Aikaleima pitää olla aina jokaisessa muutoksessa mukana tässä tapauksessa [Mundy et al., 2006].

Tietojen tallentaminen vaatii paljon levytilaa, koska muutoksia tulee useita päivittäin. Nykyajan levytilakapasiteeteilla tämä ei kuitenkaan ole ongelma. Mundy ja muut [2006] mainitsevat, että kaikkea tietoa ei tarvitse säilyttää ikuisesti ja on hyvä suunnitteluvaiheessa määritellä, kuinka monta vuotta tai kuukautta tietoja säilytetään ennen tietokannan siivousta. BI-toiminnoissa varsinkin yli viisi vuotta vanha tieto ei sinänsä vaikuta niin paljon analyysiin tai yrityksen johdon päätöksiin, että se olisi tarpeellista ottaa mukaan raportteihin päätöksiä tekeville henkilöille. Tietoja kerääntyy päivittäin valtavia määriä, mikäli järjestelmä on käytössä ja tämä saadaan sitten analysointitoiminnassa hyötykäyttöön. Tietoa voidaan myös käyttää virheiden etsimiseen, markkinointiin tai vaikka ostokäyttäytymisen seurantaan.

4.6 Tietolähteiden valinta

Parhaassa tapauksessa eri tietolähteitä, joilla tuleva tietovarasto täytetään, on olemassa vain muutamia ja ne ovat hyvin järjestelty ja samankaltaisia. Useimmiten kuitenkin tietolähteitä on useita erilaisia ja jokainen niistä vaatii oman projektinsa, joissa täytyy tehdä omanlaisensa tietojennoutomenetelmä. Arora ja muut [2009] mainitsevat, että nämä eri tietolähteet voivat olla eri osioista samaa konsernia tai ne voivat olla yrityksen eri osastoja.

Tietoa luetaan suoraan lähteenä olevasta tietokannasta tai mikäli tämä ei ole mahdollista niin täytyy käyttää yhteysajureita tai kolmannen osapuolen tarjoamaa yhteyskirjastoa, jotta voidaan rakentaa yhteys tietokantaan. Suoraan tietokannasta lukeminen on joissain

tapauksissa mahdollista ja nopein tapa, mutta se sisältää eräitä riskejä mitä pitää ottaa huomioon. Tietolähteen ollessa päivittäin käytössä, sen käyttäjät tekevät jatkuvia muutoksia, jotka voivat vaikuttaa paketin tietojen hakuun Extract-vaiheessa. Tietokannan ollessa ulkoisen toimijan käytössä, jolloin se ei ole sisäverkossa, siihen on mahdotonta saada suoraan yhteyttä. Tällöin pitää käyttää toimittajan tarjoamaa rajapintaa ja noudattaa sen ohjeistusta integraation rakentamiseen. Kolmannen osapuolen tietokantaan pääsy on suoraan estetty tietosuojasyistäkin.

Suunnitteluvaiheessa pitää ottaa selville eri lähteet ja selvittää tekniikat joilla lähteiden tietokannat on strukturoitu sekä kuinka saa lähdetietokannoista luettua tarvittavan datan. Oletuksena pitäisi voida lukea tietoja lähdetietokannoista mahdollisimman yksinkertaisin lausein, jotta kuormitus lukuvaiheessa olisi vähäistä. Tätä raakadataa muokataan paremmaksi paikallisessa kehitysympäristössä Transform-vaiheessa.

4.7 Duplikaattien hallinta

Aina kun kerätään useasta erillään olleesta tietokannasta objektit yhteen, löytyy samankaltaisuuksia. Arora ja muut [2009] mainitsevat, että tietokannat ovat täten heterogeenisiä. Nämä objektit ovat samoja yrityksiä tai henkilöitä ja näillä voi myös olla esimerkiksi eri yhteystiedot tai tittelit. Tiedon transformaatiovaiheessa pitää yhdistää nämä tuplaarvot (duplikaatit) ja on pääteltävä mikä tieto on oikea [Mundy et al., 2006]. Tässä auttaa, kun selvitetään mikä on tiedolle päätietokanta (eng. Master), jossa on parhaiten ylläpidetty tieto. Tieto esimerkiksi laskutusosoitteesta on paras siinä järjestelmässä, josta laskut lähtee asiakkaalle, koska tieto päivitetään osoitetiedon saamisen jälkeen. Jollekin toiselle tiedolle taas voi löytyä paras tieto toisesta tietojärjestelmästä. Jos paras arvo on vaikea määritellä, voidaan tukeutua aikaleimaan, joka tiedon päivittyessä tallentuu. Aikaleima kertoo uusimman tiedon sijainnin.

Duplikaatit voi paikallistaa ajamalla tiedot algoritmeista läpi, joka pystyy erottamaan samankaltaisuudet. Useassa ETL-työkalussa on mukana duplikaattien erottelu ominaisuuksia, joita voidaan käyttää suhteellisen helposti. Nämä algoritmit toimivat yleensä siten, että ne vertailevat merkkijonoja toisiinsa ja antavat niille sopivuusprosentin, jonka perusteella ne voidaan käydä tarkemmin läpi. Arora ja muut [2009] mainitsevat, että algoritmit eivät toimi aina täysin oikein kun käsitellään reaalimaailman dataa.

4.8 Lisätoiminnallisuudet

Suunniteltaessa ETL-prosessia ei aina voi tietää tarkalleen missä lähdepalvelimet sijaitsevat tai minkä nimisiä ne ovat, tai edes sitä missä ETL-prosessi tullaan ajamaan tulevaisuudessa. Suunnittelun jälkeen myös pitää ottaa huomioon, että ympäristö voi vaihtua kehityksen aikana tai sen jälkeen ja ETL-prosessin pitää toimia myös tämän jälkeen. Toisin sanoen suunnitelmaa tehdessä ETL-järjestelmässä on hyvä ottaa huomioon erilaiset kohdat, jotka voivat muuttua myöhemmin, koska näitä on vaikea hallita jos ne ovat koodattuja sisään järjestelmään. ETL-työkaluissa on omat asetukset kehitys, testi ja tuotantoympäristöille, jotka voidaan tallettaa ulkoiseen tiedostoon, missä jatkomuokkausta tehdään [Mundy et al., 2006].

4.8.1 Lokitiedostot ja varmuuskopiot

ETL-prosessin suunnitteluvaiheessa pitää myös selvittää, miten voidaan seurata, että kaikki on mennyt oikein ETL-ajon aikana. On olemassa paljon kysymyksiä mitä pitää ottaa suunnitelmassa huomioon. Mitä tapahtuu jos paketti epäonnistuu jostain syystä? Voimmeko tällöin selvittää nopeasti ongelman lähteen ja korjata sen? Apuna tähän voidaan käyttää lokitiedostojen rakentamista erillisiin tiedostoihin tai suoraan tietokantaan lokitauluun. Mitä tarkemmin ajon aikana tapahtuneet asiat on rekisteröity, sitä helpompi on sitten selvittää ongelman sattuessa virheen lähde.

Virheen sattuessa tarvitaan myös ilmoitus siitä, että virhe on tapahtunut. Jos ei vastaanoteta mitään ilmoitusta asiasta, ei voida tehdä mitään ennen kuin saadaan viestiä viimeistään käyttäjältä, joka on huomannut virheen. Virheen jo tapahduttua sitä on vaikeampi korjata, koska esimerkiksi järjestelmä on ollut käytössä ja tietoja on jo ehditty päivittämään. Järjestelmän ylläpitäjät eivät aina tarkkaile joka ajoa sen aikana vaan prosessin rakentaminen kannattaa tehdä niin, että virhetilanteet on huomioitu. Eräissä ETL-prosessien rakennusohjelmissa on mahdollisuus, että lähetetään ilmoitus sähköpostitse tai tekstiviestillä mikäli paketti on onnistunut tai epäonnistunut [Mundy et al., 2006]. Tähän voidaan liittää esimerkiksi epäonnistumisen jälkeen lokitiedot siitä, mikä on johtanut epäonnistumiseen.

ETL-prosessin lähteenä on myös erilaisia tiedostoja. Nämä voivat olla tekstiä tai XML-tiedostoja. Esimerkiksi laskutustiedot voidaan tuoda kerran kuussa lähdekansioon mistä

ne ajetaan sisään. Se mitä näille tiedostoille tapahtuu ajon jälkeen pitää ottaa huomioon. Mikäli myöhemmin tulee epäselvyyksiä prosessissa ja täytyy tehdä virheen etsintää, on hyvä jos tiedostot on siirretty talteen. Voidaan esimerkiksi suorittaa tiedostojärjestelmän funktio, joka nimeää tuotavat tiedot päivämäärän mukaan ja tallettaa ne varmuuskopiokansioon.

4.8.2 ETL-pakettien sijainti

Lopuksi tulee määritellä, että on olemassa paikka mihin tallentaa rakennetut ETL-paketit jotta ne säilyvät suojassa vuosien ajan [Mundy et al., 2006]. Mikäli paketit poistettaisiin tai muutettaisiin hakemistorakennetta niin ETL-prosessin suoritus ei onnistuisi. Paketit sijoitetaan yleensä tiedostojärjestelmään, jossa niille merkitään suunnitelmaan pysyvä loppusijoituspaikka ja asetetaan suojaukset.

4.9 Yhteenveto

Kehitystyön aikana voi tietenkin ilmetä asioita mitä ei ole onnistuttu ottamaan huomioon, koska niitä ei ole ajateltu tarpeeksi. Suunnittelun ohella kehitetään kuvaus prosessista paperille, joka auttaa kehittäjää ja asiakasta olemaan samalla viivalla siitä mitä yritetään saavuttaa. Tämä estää yllätyksiä ja väärinkäsityksiä. Suunnitelman jälkeen myös kehitystyö sujuu tehokkaammin kun rajakohdat on jo saatu mietittyä ja ennakoitua.

5. ETL-prosessin kehittäminen

Seuraavassa käyn läpi edellisessä luvussa esitetyn suunnitelman pohjalta luotavan ETL-prosessin kehitysvaiheita. Järjestys missä esittelen kehitystoimenpiteitä noudattaa hyväksi havaitsemaani tapaa siitä, missä järjestyksessä kehittäminen on parasta toteuttaa.

5.1 Aloitustoimenpiteet

Hyvän suunnitelman kehittämisen ja sen hyväksymisen jälkeen voidaan aloittaa varsinainen kehitystyö. Kehitystyötä varten on hankittava tarvittavat ohjelmistot, laitteistot ja yhteydet palvelimille. Kehitystyö tehdään kehitysympäristössä ja se siirretään ennen käyttöönottoa tuotantopalvelimelle. Kehitysympäristö on myös mahdollista muuttaa tuotantoympäristöön. Tällöin pitää ottaa huomioon, että kehitys- ja tuotantopalvelimilla tulee olla samanlaiset yhteydet lähdetietokantoihin, jotta käyttöönotto on myöhemmin mahdollista. Vähintäänkin kehitysympäristöön on rakennettava lähdeympäristöä vastaava tietolähde, jota voidaan lukea aloitusvaiheessa, kuten edellisen luvun alussa mainitsin.

Lähdetietokantoihin tulee olla pääsyoikeudet ja näistä lukemisen tulee onnistua palvelimelle. Tietoa ei tarvitse jokaisen ajokerran aikana aina lukea uudestaan lähdetietokannasta vaan kannattaa perustaa ”hiekkalaatikko” (eng. Sandbox), joka tarkoittaa ajantasaista kopiota lähdetietokannan tiedoista, jolloin lähdetietokantaa ei tarvitse kuormittaa [Mundy et al., 2006]. Kuormittuminen näkyisi käyttäjien koneella hidasteluna. ”Hiekkalaatikko”-ympäristön luominen tapahtuu testiympäristöön, johon luodaan ensimmäinen versio tietovaraston tietokantarakenteessa. Tämän tietovaraston taulukkokuvaukset voi luoda myöhemmin kehitysympäristöön tai sitten voi kopioida koko testiympäristössä luodun tietokannan sisällön tuotantoympäristöön.

Alkuvalmisteluiden jälkeen aloitetaan tietokannan lukukyselyjen luominen, joka tarkoittaa ETL-prosessissa Extract-vaihetta. Tämän jälkeen muokataan tietoa ja yhdistellään sitä Transform-vaiheessa ja lopuksi ajetaan kaikki kohdetietokantaan Load-vaiheessa.

5.2 Extract

ETL-prosessin ensimmäinen osa alkaa siitä kun luetaan (Extract) dataa prosessiin. On olemassa vaihtoehtoisia tapoja tiedon lukemisessa sisään. Silloin, kun lukemiseen käytetty aika ei ole este, tiedot voidaan lukea suoraan muistiin välittämättä muutoksista lukemisen aikana ja tiedot voidaan muokata vasta myöhemmässä Transform-vaiheessa. Suoraan lukeminen ei kuormita lähdetietokantaa kovinkaan paljon, koska mitään muutoksia tai funktioita ei käytetä lukulauseessa, eli tällöin otetaan data ulos lähteestä suoraan.

Tietokannoissa hankalia ovat esimerkiksi Null-arvot, eli kentät jotka eivät sisällä mitään, eivät edes tyhjää merkkiä. Harva tietokanta myös hyväksyy Null-arvoja joten ne täytyy käydä läpi ja muuttaa oletusarvoltaan ei tyhjiksi. Tämä toimenpide säästää aikaa myöhemmissä ETL-prosessin vaiheissa, mikäli se tehdään jo Extract-vaiheessa, eli kirjoitetaan Null-arvojen tarkistus ja korjaukset SQL-lukukomentoon.

On tärkeää huomata, että käytössä voi olla välitietokanta (eng. Staging area), johon tiedot kerätään [Mundy et al., 2006]. Silloin kun ei käytetä välitietokantaa, käytetään tiedon liittämiseen SQL:n UNION lausetta, joka yhdistää tietokantojen tietoja lukemisen aikana. Tämä ehkäisee pakettien rakentamiseen kuluvaa työtä. Tämänkaltainen tietojen yhdistely onnistuu mikäli molemmat tietokannat sijaitsevat samalla palvelimella ja ovat samantyyppisiä-SQL tietokantoja.

Yksi vaihtoehto on lukea sisään prosessiin toisen taulun tiedot ja tehdä Transform-vaiheessa tietojen haku (eng. lookup) toiseen tietokantaan, jolloin ei ole väliä missä tietokannat sijaitsevat, kunhan niihin pääsee käsiksi. Tällöin Extract-vaiheessa luetaan vain toinen tietokanta millä tahansa menetelmällä ja yhdistetään toisen tiedot myöhemmin seuraavassa Transform-vaiheessa.

5.3 Transform

5.3.1 Tiedon siivous ja muunnokset

Tiedon lukemisen jälkeen muokataan sisään tuotua dataa. Muokkaus alkaa datan siivoamisesta, jolloin on käytössä Transform-vaiheessa siivotut tiedot. Arora ja muut [2009] määrittelevät tiedon siivouksen aktiviteetiksi, missä määritellään ja etsitään data

joka on ei haluttua, korruptoitunutta, epämääräistä ja viallista. Datan siivoamisen aikana käydään läpi jokainen arvo ja tarkistetaan, että arvot eivät ole esimerkiksi Null-arvoja. Samalla tieto muokataan sopivaksi ja yleisesti luettavaksi myöhempiä vaiheita varten. Aina ei esimerkiksi voi pitää mukana tietotyyppejä joille on asetettu maksimipituus, koska ne eivät mahtuisi kohdetietokannassa olevaan kenttään. Ei kannata olettaa kentän tyypin tai sen sisältävän arvon olevan jotain, koska se voi muuttua myöhemmin, vaan tiedot tulisi käydä läpi.

Kenttien tai sarakkeiden nimet voivat olla tuotavassa datassa jonkin toisen järjestelmän nimeämiskäytännön mukaisia ja samaa käytäntöä ei ole tarpeen jatkaa, varsinkaan jos se on vaikeasti luettava. Transform-vaiheessa nimetään kentät uudestaan omalla nimeämiskäytännöllä, jota käytetään myös tietovaraston loppuniminä [Mundy et al., 2006]. Mikäli sarakkeiden nimet on tässä vaiheessa muutettu samaksi kuin tietovarastossa olevat kohdesarakkeet niin Load-vaihe helpottuu huomattavasti. Lähdekenttä osataan tämän johdosta yhdistää automaattisesti kohdekenttään. Tämä myös auttaa osaltaan estämään sitä, että ei onnistuta vahingossa yhdistämään väärää lähdettä kohteeseen samankaltaisen nimen vuoksi.

Muutoksien tekemisen aikana korvataan vanhat tiedot uusilla, eikä arvoista tehdä duplikaatteja uusilla nimillä. Muutoin myöhemmässä vaiheessa saatettaisiin vahingossa muokata väärää tietoa ja lisäksi ETL-prosessin käsittelemä tieto varaisi enemmän muistia kuin olisi tarpeen. Virheiden etsintä (eng. Debug) onnistuu myös paremmin silloin, kun ei ole ylimääräisiä sarakkeita tietokannassa.

Tietovarasto ei aina välttämättä kopioi lähdetietokannassa olevaa tapaa tehdä asioita, vaan on mahdollista tehdä tietovarastosta parempi kuin lähteestä. Esimerkiksi jos tietolähteessä sukupuoliarvo-kentässä lukee ”Mies” niin tämä voidaan muokata vastaamaan oikeaa arvoa tietovarastossa. Esimerkiksi tietovarastossa tieto voisi vastata erilaista kenttätyyppiä (Int) tai siellä voisi olla erikseen kenttä miehelle tai naiselle (Boolean). Täten saavutetaan se, että tietovarastoa ei sidota yleisesti esimerkiksi suomenkieleen, koska se on oletus joka ei toimi toisten järjestelmien tietojen kanssa yhteen.

Edellisen esimerkin mukaisesti arvon datatyyppiä voidaan muokata tyypistä String tyyppiin Boolean tai Int. Tyyppimuunnoksissa ei aina voida korvata vanhaa arvoa vaan

on pakko tehdä uusi arvo, koska tyypit eivät ole yhteensopivia. Tätä varten voidaan uudelleennimetä vanha arvo liittämällä esimerkiksi _old kentän nimen loppuun. Tyyppimuunnoksia tehdessä täytyy olla varovainen siitä mitä tekee, koska tyyppimuunnokset voivat aiheuttaa ajonaikaisen virheen.

Kussakin kohdassa ETL-prosessia on mahdollista hyödyntää virheenseurantaa, jossa tietovirrasta poimitaan virhetilanteet ja käsitellään ne erikseen. On mahdollista, että esimerkiksi ETL-paketti epäonnistuu sen vuoksi, että käyttäjä on kirjoittanut kenttään tekstiä mitä lähdetietokannan ei olisi pitänyt sallia. Käsittelen virhetilanteiden hallintaa enemmän luvussa 5.5.

5.3.2 Tiedon täydentäminen

Seuraavassa vaiheessa täydennetään puuttuvat arvot dataan. Näitä löytyy muista tietokannoista, tiedostoista, tai vaikkapa www-palvelun kautta. Jos esimerkiksi halutaan tuoda yrityksen henkilön tietoja mukaan käsittelyyn, henkilötaulusta saadaan vierasavain yritystauluun mistä tarvittava tieto löytyy [Mundy et al., 2006]. Vierasavaimia käytetään silloin yhdistämään käsiteltävään tietokantariviin uusia kenttiä toisista tietokanta-tauluista. Tämä helpottaa työtä kun voi käsitellä tietoja yhdessä ETL-prosessissa, eikä erillisesti hallittavia paketteja tarvitse tehdä.

UNION ja JOIN -komentoja voi tehdä myös kohdetietokantaa vasten silloin kun on ajettu tietoja jo aikaisemmin sisään. Tätä tapaa voidaan käyttää esimerkiksi siinä tapauksessa, jos tarkistetaan onko tieto ajettu sisään tietovarastoon jo aikaisemmin. Tiedon löytyessä voi tehdä uusia toimintoja, kuten verrata onko tapahtunut muutoksia tietokannassa. Nämä rivit voi siirtää käsiteltäväksi erilleen normaalista tietovirrasta. Tiedon löytyessä tarvitsee ottaa muutos talteen, mikäli se on prosessin suunnitelmassa määritelty ja tallettaa muutostieto aggregaatti-tauluun. Tässä tapauksessa tietovirta voi haarautua ensimmäisen kerran eri toimintoihin. Tiedon puuttuessa tiedetään sen olevan uusi rivi, minkä jälkeen luodaan tämä rivi tietokantaan Load-vaiheessa.

Tietovirran haarautuessa prosessi mutkistuu ja se tulee vaikeaselkoisemmaksi. Tehokkaamman kehittämisen etuna olisi, että haarautuminen tapahtuisi mahdollisimman myöhään. Tiedon eri haaroille on tarkoitus tehdä samanlaisia toimenpiteitä, kuten esimerkiksi lajitella tai yhdistellä tietoa. Mikäli tarvitsee tehdä samoja toimenpiteitä myöhem-

min, niin se tarkoittaa tuplamääräistä työtä ja muutosten sattuessa pitää tehdä muutokset kahteen eri paikkaan. Tämä sotii uudelleenkäytettävyyttä ja ylläpitämistä vastaan. Tietovirrat voi jälkeenpäin yhdistää samaksi virraksi ennen Load-vaihetta tai yhteisiä operaatioita. Tässä tapauksessa tietovirtojen pitää kuitenkin olla samanlaiset, jotta yhdistäminen olisi mahdollista. Mikäli tietovirtaa on esimerkiksi lajiteltu, niin lajittelu samalla kriteerillä pitää tehdä kummassakin paikassa.

Tiedon löytyessä Lookup-toiminnolla pitää ottaa erilainen lähestymistapa riippuen suunnitelmasta. Mahdollisesti ylikirjoitetaan olemassa oleva tieto tai otetaan talteen sekä uusi, että vanha arvo. Tällöin siirretään vanhat arvot talteen aggregaatti-tauluun. Tämä uusi rivi saa avaimen, johon viitataan kentästä joka on sillä hetkellä käsittelyssä. Tehtäessä tällaista tiedon päivitystä yöllisinä ETL-ajoina on tarpeen muistaa, että viimeisen ajon jälkeen jotkut arvot ovat voineet muuttua useammin kuin kerran. Näin tapahtuu esimerkiksi yritysmaailmassa, jossa tietoja käsitellään jatkuvasti. Mikäli näistä otetaan talteen vain viimeksi voimassa oleva arvo, niin menetetään kaikki välissä tapahtunut.

Jotkin ohjelmat itsessään keräävät historiatietoja jokaisen päivityksen jälkeen. Tällöin ETL-prosessi monimutkaistuu sen verran, että pitää ensin lukea tätä historiataulua ja ottaa sen tiedot talteen ennen kuin otetaan nykyinen arvo taulusta. Se, että lähdetietokanta pitäisi yllä muuttuneita arvoja on sinänsä jo valtava etu, sillä edellisessä kappaleessa mainittuja tietoja ei menetetä.

Tarkasteltaessa sitä onko jokin arvo uusi vai vanha pitää olla tarkkana. Lähdetietokannasta voidaan ottaa viite tiettyyn riviin aina talteen ja luottaa siihen, että viitteen taustalla oleva objekti säilyy. Tässäkin piilee vaaransa, jos oletetaan objektien sijainnin pysyvän samoina. Yleensä tietokantoja siivotaan ja sieltä poistetaan vanhoja arvoja päivittäin. Mikäli nämä ehtivät poistua kokonaan niin tietokanta voi uudelleen järjestää tietokannan indeksit (avaimet), jolloin nämä indeksit ovat taas jaossa toisille riveille. Jos linkitys tapahtuu, niin tietokannassa viitataan väärään objektiin tämän jälkeen.

Tiedon yhdistämisessä kannattaa olla useampi kuin yksi kriteeri mihin verrataan muuttunutta dataa. Pitää yrittää käyttää aina parasta mahdollista ja muuttumatonta vertauskohdetta. Esimerkiksi jos verrataan laskutusta tuotteittain ja kuukausittain, voi yhdistää

kaikki avaimet, eli lasku, tuote ja kuukausi. Tästä tullaan taas tärkeimpään attribuuttiin, eli aikaan, jolloin sen uniikkia arvoa voi käyttää hyödyksi. Menneen tapahtuman aikaleima pysyy muuttumattomana.

Usein ETL-prosesseissa tehdään monta tiedon yhdistämistä peräkkäin. Jokaiselle avaimelle täytyy löytää vastaavuus, jotta tieto olisi mahdollisimman täydellistä. Sijaistaulua voi käyttää esimerkiksi sellaisista lähdejärjestelmistä haettaessa, joissa on paljon sisäisiä viittauksia (käytännössä alasveto-valikkoja). Näissä pitää yhdistää alasvetovalikon ID-numero vastaavaan tekstiin toisessa taulussa. Aiempien tietojen haku ja yhdistämistoimintojen tietoja voi käyttää seuraavan pohjana, joten ajojärjestys kannattaa suunnitella etukäteen.

5.3.3 Dimensiot

Mundy ja muut [2006] mainitsevat, että tietovaraston luomisessa tärkeä osa-alue on dimensiot. Ne ovat löydettyjä objektiryhmiä. Dimensioille rakennetaan oma taulunsa tietokantaan. Yleisiä dimensioita ovat esimerkiksi aika ja paikka. Kaikki aikaleimat yhdessä muodostavat aika-dimension. Osoite, postinumero, kaupunki ja maa yhdessä muodostavat geografisen-dimension. Joissakin järjestelmissä myös demografia voi olla oma dimensionsa. Dimensioilla on oma avaimensa tietokannassa. Uusi dimensio luo aina uuden ulottuvuuden tietovarastoon. Suunnitteluvaiheessa määritellään, mitkä tiedot ovat dimensioita, joita halutaan seurata. Nämä tiedot käsitellään omassa osiossaan ETL-prosessissa kun ajetaan päivitys tietokantaan. Muuttuneet arvot tallennetaan samaan tai erilliseen tauluun ja niille annetaan uusi ID-numero, johon voidaan viitata päätaulusta. Näin ne löytyvät analysointivaiheessa myöhemmin ja asioita voidaan tarkastella asioita valittujen muuttujien kautta [Mundy et al, 2006].

Mundy ja muut [2006] mainitsevat, että eräissä ETL-työkaluissa on mukana erillinen funktio, jolla voi hallita dimensioiden muutoksia. Tämä auttaa vähentämään manuaalista työtä, mitä työn tekeminen käsin alusta loppuun saattaisi aiheuttaa. Voi siis sanoa, että kalliimman ohjelman hankkiminen tietovaraston luomiseen saattaa säästää rahaa projektissa henkilötyötunteina.

Dimensioita käytetään esimerkiksi data-analyyseissä ja BI-prosesseissa, joita ajetaan tietovarastosta. Historiatiedoista saadaan poimittua kaikki tarpeellinen tieto, jotta tietoja yhdistämällä saamme tarkasteltua miten tallennetut tiedot eroavat toisistaan.

5.3.4 Tiedon eheyden tarkistaminen

Joskus lähdetietokantoihin on voinut kerääntyä turhaakin tietoa. Käyttäjät lisäävät päivittäin useita rivejä ja näissä ei ole oleellista tietoa tietovaraston kannalta. Esimerkiksi jos joku ei ole vastannut puhelimeen tiettyyn kellonaikaan. Näitä tietoja voidaan poistaa Transform-vaiheessa, jonka jälkeen tietovarasto sisältää oleelliset tiedot.

Tiedon eheyttäminen on yksi isoista tehtävistä, jotka tehdään Transform-vaiheessa. Suunnitelmassa on täytynyt määritellä mitä kriteerejä tai esimerkiksi aktiviteetteja tullaan seuraamaan tietovarastossa. Muita tietoja ei kannata tuoda. Pitää myös ottaa huomioon, että tarpeet saattavat muuttua myöhemmin, joten mitään ei kannata syväkoodata. Myöhemmin joudutaan ehkä tuomaan mukaan esimerkiksi lisätietoja muista tauluista. Yleisimpiä aktiviteetteja mitä esimerkiksi asiakasrekisterijärjestelmässä syötetään sisään, on kuinka paljon myyntipuheluita tai tapaamisia työntekijät ovat suorittaneet mihinkin aikaan. Nämä tiedot voivat löytyä ainoastaan asiakasrekisterijärjestelmästä josta ne pitää poimia tietovarastoon. Yrityksessä tuotteet eivät myy itseään ja jos asiakkaaseen ei ole otettu yhteyttä pitkään aikaan, niin tuoteostoja asiakkaan taholta ei myöskään varmaan tapahdu.

Tieto voi myös olla vajavaista tai muuten huolimattomasti täytettyä. Tietoa kannattaa yrittää täydentää käyttäen samaa tietokantaa tai muita tietokantoja. Jos esimerkiksi on perustettu asiakas ilman osoitetietoja toiseen järjestelmään, mutta sama asiakas löytyy myös toisesta järjestelmästä, tietovarastoon kannattaa silloin nämä tiedot ilman muuta yhdistää. Tietovarasto on se paikka yrityksessä, jossa viimeisin, eheytetty, täydennetty ja siivottu tieto on tallennettuna. Tätä voi myöhemmin käyttää hyödyksi, kuten päivittää tietoja tietovarastosta ulospäin täydentämään työntekijöiden käytössä olevia järjestelmiä.

Inhimillisistä virheistä johtuvat täyttövirheet ovat myös yleisiä CRM ja ERP -järjestelmissä. Esimerkiksi henkilöiden titteleitä täytettäessä, voi sama titteli olla kirjoitettu monella eri tavalla. Näiden tyylien samankaltaisuuksia on vaikea löytää tekstialgo-

ritmeillakin. Agrawal [2008] mainitsee, että kaikki nykyiset algoritmitkään eivät onnistu käsittelemään tosielämän dataa. Postituksiin valitaan joissakin tapauksissa titteli mukaan, jolloin sen pitää olla oikein kirjoitettu. Pitää olla mietittynä kuinka paljon aikaa tiettyyn kohtaan prosessia kannattaa käyttää ja priorisoida ajankäyttöä.

5.4 Load

Load-vaiheen pitäisi olla ETL-prosessin nopein ja suoraviivaisin vaihe tai näin usein oletetaan. Merkitäänhän siinä vain mikä tieto tietovirrasta viedään mihinkin kenttään kohdetietokannassa. On kuitenkin olemassa paljon asioita, mitä pitää ottaa tässäkin vaiheessa huomioon.

5.4.1 Tietotyyppien yhteensopivuus

Tietotyyppien pitää olla yhteensopivat toistensa kanssa Load-vaiheessa. Yleinen syy ETL-ajon kaatumiseen on, että tapahtuu virhe tietojen viemisessä tietokantaan. Datatyyppiä ei yritetä muuttaa automaattisesti, vaan se on täytynyt muuttaa jo aikaisemmissa vaiheissa. Tietotyyppiä sovitettaessa tulee siis ajonaikainen virhe. Mikäli tietotyypit ovat samat, mutta kentän koko on eri, jos esimerkiksi kohde on pienempi kuin siihen tuotava arvo, tapahtuu virhetilanne (eng. truncation error). Nämä virhetilanteet pitää käsitellä, jotta koko paketin ajo ei epäonnistu. Onkin tärkeää jo Transform-vaiheessa ottaa huomioon mahdollisia virhetilanteita ja tehdä tyyppimuunnokset. Ei siis voida olettaa, että tiedolle joka on syötetty kenttään, olisi mahdollista suorittaa tyyppimuunnos. Joskus tiedon voi muuntaa tekstimuotoiseksi, mutta siitä takaisinpäin on tarkempaa. Tyyppimuunnoksen voi tehdä vaikka se ei näyttäisi tarpeelliselta. Näin varmistetaan, että tieto on oikeantyyppinen ja vältetään virhetilanteet.

5.4.2 Päivitys vai uuden luonti

Load-vaihe voi päättyä usealla eri tavalla. Luodaan uudet tiedot Insert-komennolla tai päivitetään vanhat Update-komennolla. Joskus joutuu tekemään molemmat, koska kumpikin tilanne on jokapäiväinen. Uutta dataa syntyy ja vanhaa päivittyy.

Toinen asia mihin tietovirta voi päättyä on täysin uuden tiedon luominen ja sen historian tallettaminen tietokantaan. Yleensä hyvin rakennettu tietokanta hoitaa indeksoinnin ja vierasavaimet, mutta toisaalta tämän voi hoitaa manuaalisesti ensin toisessa tietovirras-

sa. Sen jälkeen kun tieto on tallennettu, Load-vaihe päättyy aina siihen. Tietovirta päättyy ja sitä ei ole tarpeellista jatkaa. Mikäli on tarpeellista jatkaa tästä, niin prosessi on väärin suunniteltu, ja silloin ei pitäisi olla Load-vaiheessa vielä. ETL-työkaluilla ei aina ole mahdollistakaan jatkaa. Tietojen tallentamisessa väliaikaisesti tietokantaan käytetään Transform-vaihetta ja sen työkaluja. Tämän datan lisäksi voi käyttää myös tietokannan Update-lausetta.

5.4.3 Ajojärjestys

On tärkeää ajaa prosessin eri komponentit tarkkaan mietityssä järjestyksessä. Suunniteltaessa ajojärjestys säästetään aikaa ajossa sekä koodaustyössä. Esimerkiksi jos ERP ja CRM -järjestelmissä on olemassa lista tuotteista, joita viedään ETL-prosessilla. Näitä esimerkiksi hallitaan ERP-järjestelmässä, mutta on tärkeää linkittää CRM-järjestelmän tietoja tuotteisiin. Tämän perusteella saadaan tietoja esimerkiksi tuotekiinnostuksista. Jotta pystytään tekemään linkityksiä ERP ja CRM -järjestelmien tiedoista tietovarastoon tuotaviin tietoihin, niin pitää olla ensin ajettuna pohjatiedoiksi tuotteet ja niiden tuoteryhmät. Nämä pitää löytyä ainakin välitietokannassa, josta tietoa päivitetään.

5.5 Virhetilanteiden hallinta

Tietojärjestelmäprojektissa pitää käsitellä virhetilanteet kun tekee kehitystä. Jokaisesta virhetilanteesta pitää pystyä jatkamaan. Käyttäjän aiheuttamat syötteet tai väärinkäytökset eivät saa aiheuttaa koko ohjelman kaatumista. Käyttäjä päättää lopuksi itse miten järjestelmää käyttää ja mitä arvoja sinne syöttää. Virhetilanteet pitää tunnistaa ja hoitaa asian mukaisesti. Ei voida olettaa automaattisesti tietojen oikeellisuutta, vaan lähdetietokannoissa voi olla mitä tahansa tietoa ja tiedot pitää tarkistaa.

Toisaalta on toivottavaa, että tiedot tallennettaisiin välitietokantaan tai ajettaisiin tiedostoon, jossa voi tarkastella virheellistä tietoa ja oppia virheistä. Tämän jälkeen voi ajaa nämä tiedot erillisessä prosessissa, joka ottaa paremmin huomioon nämä tiedon erilaisuudet. Kuitenkaan tämä ei ole kestävä ratkaisu vaan pitäisi pystyä muuttamaan prosessia mahdollisimman helposti. ETL-prosessi pitäisi rakentaa niin dynaamiseksi, että sitä voi myöhemmin muuttaa helposti. Eri prosessin osa-alueet voi erotella omiin ohjelmiinsa (paketteihin), joissa yhden paketin epäonnistuminen ei johda koko prosessin epäon-

nistumiseen. Tämä taas onnistuu hyvin mikäli ETL-prosessi on rakennettu erillisistä paketeista ja yksi pääpaketti ajaa nämä kaikki.

Lokitietojen luominen tiedostoon tulee viemään jonkin verran tilaa kovalevyltä, mutta jos ei perusta tietokantataulua palvelimelle niin tämä on nopein tapa suorittaa virhetietojen tallentamista. Lokitiedostoon voi kirjoittaa esimerkiksi jokaisen paketin välissä, että kuinka monta riviä on viety ja kuinka monta on käsittelyssä jolloin näkee suoraan, jos jokin ei mennyt läpi. Tällöin ei tarvitse etsiä viallisen tiedon lähdettä läpikäymällä koodia, vaan voi mennä suoraan ongelman lähteelle vain lokitiedoston tietojen perusteella. Lokitiedosto on myös oivallinen paikka kertoa mikä tarkalleen meni vikaan tietojen siirrossa ja myös kirjata ylös virheelliset rivit.

Virhetilanteisiin pitää tarttua kiireellisesti, koska yritysmaailma ei odota ja onkin hyvä suorittaa korjaukset ennen varsinaista työaikaa, jolloin käyttäjät eivät välttämättä edes saa tietoonsa, että ongelmia on esiintynyt tietojen tuonnissa. Onkin hyvin tärkeää, että käyttäjiin vaikuttavat mahdollisimman vähän eri virhetilanteet. Virhetilanteisiin tulisi voida pystyä puuttumaan täten nopeasti.

On myös mahdollista tehdä sähköinen ilmoitus (sähköposti, tekstiviesti) paketin epäonnistumisesta. Tämä ei ole niin pakollinen kehitysvaiheessa, vaan hyödyllisempi myöhemmin tuotantovaiheessa. Useimmat palvelinohjelmistot sallivat sähköisen ilmoituksen ja kattavan raportin siitä mitä lähetetään, kun jokin menee vikaan prosessissa.

Ei ole kuitenkaan järkevää laittaa sähköpostin lähetystä prosessiksi jokaiseen virhetapaukseen, vaan kannattaa koota sähköposti vasta lopuksi. Näin vältytään massasähköpostilta. Sähköpostin voi myös lähettää, kun paketti onnistuu ja silloin saadaan vähän kerättyä statistiikkaa siitä, kuinka paljon dataa meni läpi ja kuinka kauan asiat kestivät. Vaikka paketti menisi läpi virheettömästi, se ei välttämättä tarkoita, että kaikki olisi sujunut virheettömästi. Saadulla statistiikalla voi havainnoida mitä kaikkia tietoja on käsitelty ja verrata sitä tietoon siitä, kuinka paljon pitäisi olla käsiteltynä.

5.6 Yhteenveto

ETL-prosessin kehitysvaiheessa kannattaa tukeutua suunnitteluvaiheessa koostettuun tietoon mahdollisimman paljon. Tällöin asioita ei tarvitse kerrata useaan otteeseen vaan

voi aloittaa kehitystyön välittömästi. Varsinkin jos on kyse ETL-prosessista, jossa käydään läpi asiat tietyssä järjestyksessä voidaan nojautua tiettyyn tapaan tehdä asioita, eikä työmetodeja tarvitse muuttaa jokaista projektia varten. Tällä hetkellä viimeisimmät järjestelmät, joihin tehdään integraatioita muistuttavat niin paljon toisiaan, että kun tuntee yhden järjestelmän niin voi olla helppoa integroida toiseen.

6. Tulokset

Olen aiemmissa luvuissa esittänyt ETL-prosessin kehityksen vaiheita perustuen kirjallisuuteen ja omaan työkokemukseen. Olen koonnut oman työn reflektiota apuna käyttäen useita hyviä käytäntöjä, kun rakennetaan tietovarastoa ETL-prosessin avulla. Olen tunnistanut eri kohtia, jotka huomioimalla saadaan tietovarastoprojektin onnistumismahdollisuudet paremmiksi. Esittelen kohdat ETL-prosessin suoritusjärjestyksessä.

1. Mitä: Tilaajan sisällyttäminen projektiin. Mitä sitten: Asiakas ei ole tietoinen siitä mitä aiotaan kehittää. Mitä seuraavaksi: Tietovaraston tilaajalle pitää kommunikoida suunnitelma prosessista.

Tilaajan kanssa täytyy määritellä tarkasti, se mitä he haluavat tietovarastolta, mitä tietoja se sisältää ja miten niitä pitää pystyä käyttämään. Tilaaja määrittelee lopuksi tietovarastoprojektin onnistuneeksi tai epäonnistuneeksi. Tietovaraston pitää siis vastata käyttäjän tarpeita ja tehostaa työntekoa yrityksessä.

Projektin onnistumista parantaa se, että asiakas sidotaan projektiin mukaan ja pidetään ajan tasalla siitä mitä on sovittu. Suunnittelu projektissa paranee ja saadaan tarkempi kuva tarpeista ja siitä miten ETL-prosessista saadaan yrityksen tietostrategian mukaista hyötyä.

2. Mitä: ETL-prosessin suunnittelu. Mitä sitten: ETL-prosessi muuttuu liian monimutkaiseksi ja hankalasti hallittaviksi. Mitä seuraavaksi: ETL-prosessin pitää olla mahdollisimman tiivis ja hyvin suunniteltu.

Tällä tarkoitetaan, että ETL-prosessin suunnittelussa pitää havaita kaikki mahdolliset objektit ja miten niitä voidaan yhdistää ajon aikana. Yhdellä ETL-paketilla voidaan viedä monta objektia tietovarastoon kerralla. Tämän vastakohta on tehdä jokaiselle objektille oma pakettinsa ja vielä yksi paketti missä nämä objektit yhdistetään toisiinsa. Suunnittelemalla voidaan parantaa aloitustietoja siten, että kehittämistyöhön tarvittava aika lyhenee, jolloin projekti pysyy aikataulussa ja myös budjetissa.

Ennen koodaustyön aloittamista suunnitelman on tärkeää olla hyvin tehty. Paras käytäntö on aina suunnitella ja sitten vasta rakentaa. Yleensä ETL-prosessia ei ajatella projektina, jossa suunnitellaan kokonaista järjestelmää vaan se on vain osa isompaa järjestel-

mää, jolloin ETL-prosessin suunnitelma voi jäädä tekemättä. Se ei kuitenkaan ole toivottavaa.

3. Mitä: Tietojen hakeminen lähteestä. Mitä sitten: Lähdetietokanta kuormittuu joka kerta kun tiedot haetaan, jolloin kaikki tiedon käsittely hidastuu. Mitä seuraavaksi: Luo testitietokanta lähdetiedoista.

Testitietokanta pitää olla sen vuoksi, että varsinainen lähdetietokanta ei kuormitu. Näin toimitaan, koska muuten tietokantaa käyttävät työntekijät voivat huomata hidastelua työnsä aikana. Tietovaraston rakentamisesta aiheutuva haitta käyttäjille pitää minimoida kehityksen aikana. Projektin aiheuttama haitta voi saada tilaajan ajattelemaan projektia negatiivisesti.

4. Mitä: Rakennettujen ETL-pakettien siirto ja uudelleenkäyttö. Mitä sitten: ETL-paketti ei ole helposti siirrettävissä palvelimen vaihdon yhteydessä ja pienetkin muutokset vaativat koodin muutoksia. Mitä seuraavaksi: Käytä konfigurointitiedostoja, älä syväkoodaa muuttujia.

Käytä konfigurointitiedostoja määrittelemään erot testiympäristön ja tuotantoympäristön eroista. Palvelimen nimi ja myös mahdollisesti tietokannan nimi voivat erota erilaisissa ympäristöissä. Jotta pystytään vaihtamaan ympäristöä mahdollisimman tehokkaasti ja vähiten aikaa vievästi, niin käyttämällä konfigurointitiedostoja voidaan ottaa eri asetukset käyttöön helposti. Muut muuttujat kuin edellä mainitut voivat erota riippuen ympäristöstä, joten pitää tunnistaa kaikki mahdolliset eroavaisuudet eri ympäristöjen välillä ja liittää ne konfigurointitiedostoihin.

5. Mitä: Virhetilanteiden hallinta. Mitä sitten: Virhetilanne aiheuttaa tilanteen jossa tieto korruptoituu ja sitä ei ehditä korjata ajoissa. Mitä seuraavaksi: Tarkista virhetilanteet, luo ajonaikainen loki, sekä lähetä ilmoitus ylläpitäjille.

ETL-prosessi on kuten kaikki ohjelmistoprojektit, jossa pitää hallinnoida virhetilanteet siten, että ohjelman suoritus ei vaarannu. Pitää tunnistaa virhetilanteet, joita ETL-paketti ajon aikana tulee aiheuttamaan ja hallinnoida nämä virheet. Virheistä ja muista tapahtumista pitää luoda ajonaikainen loki, josta voidaan seurata prosessin suorittamista, sekä korjata ohjelman toimintaa. Mikäli mahdollista, niin prosessin ajon epäonnistumisesta

lähetetään ilmoitus sähköisesti järjestelmän ylläpitäjille, jotta pystytään nopeasti reagoimaan ongelmiin.

6. Mitä: Tiedon muutoksien seuraaminen. Mitä sitten: Ei pysty seuraamaan sitä, mikä tieto on muuttunut ja menetetään hyödyt seurata kehitystä tilastoissa. Mitä seuraavaksi: Talleta datassa tapahtuneet muutokset tietovarastoon.

Tietovaraston yhtenä valtavana etuna on seurata datan muutoksia päivittäin. Tätä dataa voi käyttää pohjana yrityksen analysointityökaluissa. Tietovarastosta löytyvät muutokset kertovat esimerkiksi asiakkaiden muuttuneista tilanteista ja sen vaikutuksesta esimerkiksi ostokäyttäytymiseen. Tätä tietoa voidaan hyödyntää myös yrityksen prosessien parantamisessa. Tietovarasto toimii myös BI-analyysissä ja muutosten tallentamisen vuoksi tästä analyysistä tulee tarkempi ja hyödyllisempi.

7. Mitä: Tiedon tuonti. Mitä sitten: Tiedon muokkaaminen Transform -vaiheessa on monimutkaista ja vaatii muokkausta jokaisen lisätyn tiedon kohdalla. Mitä seuraavaksi: Siivoa ETL-prosessiin sisään tuotava data jo tuontivaiheessa.

Dataa sisään tuodessa se pitää siistiä ja eheyttää, jotta sitä voidaan käsitellä myöhemmin paremmin. Data sisältää erilaisia virheitä, jotka myöhemmissä vaiheissa aiheuttavat suurempia ongelmia. Tällaisia ovat esimerkiksi Null-arvot, tyhjät arvot ja väärät arvot. Arvoja voi muokata suoraan luettaessa tai heti Transform-vaiheen alussa. Tiedon siivoamisen jälkeen tietoa on paljon tehokkaampaa jatkojalostaa ja täydentää.

8. Mitä: Tiedon eheys. Mitä sitten: Tieto on puutteellista ja väärää analysointivaiheessa, eikä tule mukaan tuloksiin. Mitä seuraavaksi: Eheytä, yhdistä ja täydennä data.

Tieto on usein lukuvaiheessa epätäydellistä ja sisältää paljon tyhjiä, arvoja joita yrityksen toisissa tietokannoissa mahdollisesti on. Tieto kannattaa täydentää käytettävissä olevien tietokantojen tiedoilla, jotta se on mahdollisimman hyvin analysoitavissa jälkikäteen. Tietovarastossa olevan tiedon pitää sisältää kaikki yrityksessä olevien tietojärjestelmien tieto, joten duplikaattien estämiseksi nämä tiedot pitää yhdistää.

9. Mitä: Tiedon duplikaattiarvot. Mitä sitten: Tietoja täytetään ja muutetaan kahteen eri objektiin. Tämä aiheuttaa tiedon vääristymistä ja esimerkiksi tuplapostituksia. Mitä seuraavaksi: Poista duplikaatit tietovirrasta Transform-vaiheessa.

Duplikaattien poisto on tärkeää, jotta osataan yhdistää analyysi oikeaan objektiin. Duplikaattien poistoon voi käyttää esimerkiksi tekstialgoritmia tai kolmannen osapuolen työkaluja. Yrityksen eri tietojärjestelmässä voi olla oma objektikorttinsa jokaiselle asiakkaalle ym. Näistä tietojärjestelmistä pitää valita se, joka sisältää viimeisimmän ja parhaan tiedon.

Näihin kaikkiin kohtiin perustuu onnistunut ETL-prosessi. Tämän suosituksen noudattaminen vastaa myös kysymykseen ”Miten saadaan tiedot eri tietojärjestelmissä tehokkaasti koottua yhteen, käyttäjien saataville ja ymmärrettävään muotoon?”, joka oli tutkimuskysymykseni.

6.1 Vertailu muihin tutkimuksiin

Varsinaista tutkimusta ETL-prosessin parhaista käytännöistä ei löydy Suomesta. Maailmalla liikkeellä on kuitenkin erittäin paljon suosituksia eri konsultti ja IT-alan yrityksiltä. Eniten viitataan Ralph Kimball ryhmän tekemään teokseen [Kimball, 1996], mikä sisältää useita eri suosituksia ETL-prosessin tehokkaasta suunnittelusta ja kehittämisestä.

Kimball [1996] mainitsee, että tiedon luku ja kirjoituskerrat kannattaa pitää niin vähäisinä kuin mahdollista. Tämä vie aikaa ETL-prosessilta. Kannattaa suodattaa ja järjestää tiedot aikaisessa vaiheessa, jolloin käsiteltävien rivien määrä vähenee. Useat tietovirrat kannattaa käsitellä rinnakkain jos laskentateho ja muisti riittävät, koska tämä tehostaa kapasiteetin käyttöä. Kannattaa käsitellä vain tietoa mitä tarvitsee, eli pitää rajata käsiteltävän tiedon määrää. Eliminoi kaikki ylimääräinen verkkoliikenne, joka voi haitata ETL-prosessia, koska tällöin ei olla niin riippuvaisia huonosta verkkoliikenneyhteydestä. Anna ETL-prosessin tehdä kaikki työ, eli vältä kutsumasta ulkoisia funktioita tai ohjelmia vaan keskitä kaikki toiminnallisuus ETL-pakettiin.

Kimball keskittyy tämän vuoksi tekniseen osuuteen ja kehittämiseen. Fokus on nopeudessa ja tehokkuudessa. Toisaalta yrityksen data integraatio konsultit ovat tehneet suosituksia ETL-prosessiin, jossa keskitytään projektiin kokonaisuudessaan [Millbrook, 2013]. Heidän mielestään ETL-prosessin onnistumisen näkökulmasta kriittisin asia on osaava työvoima. Tietovaraston tai BI-työkalun tilaavalla asiakkaalla ei yleensä ole osaavaa työvoimaa itsellään käytössä, joten kehittäjien jotka projektiin valitaan pitää

olla taitavia. Rakennettaessa BI-työkalua, joka vaatii datan analysointia, tarvitaan projektiryhmässä henkilöitä, jotka osaavat analysoida yritysdataa ja saada sieltä eristettyä tärkeimmät asiat asiakkaalle.

Tämän tutkielman tulokset noudattelevat muita suosituksia ETL- ja tietovarastoprojektien parhaisiin käytäntöihin. Asiakkaalla voi tietenkin olla erityisiä tarpeita, jotka tietenkin pitää ottaa huomioon ETL-prosessin kehityksessä.

7. Johtopäätökset

Kaikki rahaan ja yrityksen liiketoimintaan liittyvä tieto on oleellista yrityksen omistajille ja johtajille, joten tämän tiedon analysointi on korvaamatonta ja se on myös kannustin siihen, miksi tietovarastoprojektiin edes ryhdytään. Tietovarasto kerää hyödyllistä tietoa siitä miten yritystoimintaa ja rahavirtaa voitaisiin parantaa. Tietovarastoprojektilla voi olla siten hyvä palautus verrattuna tehtyyn investointiin.

ETL-prosessi ja tietovaraston luominen vaatii jo jonkin verran yritysdataa, että se olisi järkevä kehitysalue yrityksessä. Kaikissa yrityksissä ei siihen kannata vielä ryhtyä. Yritys voi rakentaa oman yksinkertaisen tietokannan, jota käyttää tietovaraston pohjana. Tähän voi lukea tiedot eri järjestelmistä. Kaikkien pienten ja keskisuurien yritysten ei siten tarvitse palkata konsultteja, rahoittaa ja rakentaa tätä varten suurta ETL-pakettia vaan pelkkä tiedon siirto ja pienimuotoinen muokkaus SQL-lausekkein riittää.

ETL-prosessin toisena äärilaitana voi pitää prosessia missä tietoa luetaan ja viedään reaaliaikaisesti. ETL-prosessin tuottama tieto ei ole reaaliaikaista, kun se ajetaan eräajoina, mutta tarve reaaliaikaisuuteen riippuu yrityksen toimialasta. Esimerkiksi osakekaupassa viimeisin tieto täytyy olla (milli) sekunnilleen oikein.

Reaaliaikaisessa tiedon hallinnassa ei voida aina käyttää jo rakennettuja ETL-prosesseja vaan pitää rakentaa jotain täysin uutta, jotta tämä olisi toteutettavissa hyödyllisimmin. ETL-prosessia käytetään yleisesti tietovaraston, paikallisvaraston tai muun tietokannan päivittämisessä, eikä datan lähettämisessä yrityksen reaaliaikaiseen datan seurantaan. Reaaliaikaisuus johtaa projektin aikataulun kasvamiseen ja samalla budjetin leviämiseen, joten pitää olla erittäin hyvin perusteltua miksi reaaliaikaista dataa on pakko saada ja mitä hyötyjä siitä saataisiin. Mundy ja muut [2006] toteavat, että ETL-prosessia ei suositella käytettävän reaaliaikaisesti tai niin usein, että se vaikuttaa käyttäjien käyttönopeuteen tai raportointeihin.

ETL-prosessia suunniteltaessa pitää ottaa huomioon, että jokin tieto on tärkeää yhdelle käyttäjälle, mutta ei niin tärkeää jollekin toiselle. Saatetaan tehdä ratkaisevia virheitä kun jätetään jotain historiatietoja keräämättä ja sitten yritetään tehdä tarpeeksi kattavaa analyysiä järjestelmään. Suunnitteluvaiheessa on tärkeää kommunikoida tämä tilaajalle.

Tulee siis erotella jo suunnitteluvaiheessa, että mikä tieto on sellaista mitä ei ole tarpeen kerätä talteen ja mille ei tarvitse suorittaa vertailuja myöhemmässä vaiheessa.

ETL-ohjelmiston käytössä on selviä hyötyjä. Käyttämällä esimerkiksi tutkielmassa mainittuja ETL-työkaluja ei edes tarvitse osata monimutkaisia tietokantakomentoja vaan tiedot voi yhdistellä graafisella käyttöliittymällä.

Olen esittänyt tässä tutkielmassa aiemmin esimerkkejä kohteista, joita voi parantaa ja kertonut miten saavutetaan mahdollisimman tehokas toiminta, kun suoritetaan ETL-prosessia. Mikäli prosessia halutaan parantaa, mikä pitäisi olla aina tavoitteena, on useita eri kohteita missä voidaan suorittaa jatkokehitystä ETL-prosessin kehityksessä. Yksi tapa on kerätä statistiikkaa ajon aikaisesta suorituksesta. Kysymyksiä voi olla esimerkiksi, miten monta eri muutosta on tapahtunut, kuinka kauan aikaa ajaminen on kestänyt, sekä tietenkin kuinka tiedot ovat menneet läpi. Ei kannatta vain rakentaa prosessia, olettaa kaiken sujuvan tulevaisuudessa ja käyttää työntekijöitä testihenkilöinä heidän varsinaisen työnsä ohessa. ETL- ja tietovarastoprojektit voivat epäonnistua myös jälkikäteen, mikäli niitä ei ylläpidetä.

Tietovaraston rakentaminen on vasta yksi vaihe prosessia. Kaikkien yrityksen eri järjestelmien pitäisi keskustella keskenään, jolloin tiedonvaihto yrityksen sisällä parantuu. Tietovaraston rakentaminen on ensi askel yrityksen laajempaan tiedonvaihtoon ja BI-projektin ennakointia. Tietovarasto on se paikka yrityksessä, missä on viimeisin ja paras data jokaisesta järjestelmästä. Tietovarastosta voi lähettää dataa takaisin toisiin kantoihin, koska kaikki tiedot on puhdistettu ja täydennetty.

Tietovarastoprojekti ei ole sellainen, että jokaisen yrityksen kannattaa siihen ryhtyä. Pitää ottaa selville hyödyt ja haitat, kuten kustannukset. Mikäli tietovarastoprojektiin ryhdytään, niin kannattaa aloittaa yrityksen tietojärjestelmien siivous ajoissa jo projektiin valmistautuessa. Ennen projektin aloittamista kannattaa myös ottaa selville ETL-prosessiin käytettävissä olevat parhaat työkalut.

Lähdeluettelo

[Agrawal, 2008] Himanshu Agrawal, An enhanced extract-transform-load system for migrating data in telecom billing, *Conference on Data Engineering (7-12.4.2008)*, 1277-1286.

[Anderson & Kerr, 2002] Kristin Anderson and Carol Kerr, *Customer Relationship Management*, McGraw Hill Professional, 2002.

[Arora et al., 2009] Rajiv Arora, Payal Pahwa and Shubha Bansal, Alliance rules for data warehouse cleansing, *2009 International Conference on Signal Prosessing Systems*, 743-747.

[Bolton, 2001] Gillie Bolton, *Reflective Practice: Writing and Professional Development*, Paul Chapman Publishing Ltd, 2001.

[Boyd et al., 1985] David Boud, Rosemary Keogh and Donald Walker, *Reflection. Turning experience into learning.* London: Kogan Page, 1985.

[DM Review, 1998] DM Review Managezine article, http://www.information-management.com/infodirect/19991120/1675-1.html, Tarkistettu 20.4.2013.

[DWH, 2013] Data Warehouse Wiki, http://en.dwhwiki.info/ Tarkistettu 17.4.2013.

[Davenport, 2008] Robert J. Davenport, ETL vs ELT a subjective view, Insource Commercial aspects of BI whitepaper, 2008.

[DW Glossary, 2007] Data warehouse glossary, http://hubpages.com/hub/Data_Warehouse_Glossary. Tarkistettu 13.3.2013.

[Earls, 2003] Alan R. Earls, ETL: Preparation is the best bet, *Computerworld (2003)*, Vol. **37**, Issue 34.

[ETL-tools] ETL-tools Info, http://etl-tools.info/informatica/programs.html, Tarkistettu 11.2.2013.

[Exforsys, 2005] Design of data warehouse: Kimball vs. Inmon http://www.exforsys.com/tutorials/msas/data-warehouse-design-kimball-vs-inmon.html Tarkistettu 17.4 .2013.

[Garg & Venkitakrishnan, 2004] Vinod Kumar Garg, N.K. Venkitakrishnan, *Enterprise Resource Planning: Concepts and Practice*, PHI Learning Pvt. Ltd, 2004.

[Gartner, 2013a] Top 10 Strategic Technology Trends for 2013, http://www.gartner.com/technology/research/top-10-technology-trends/, Tarkistettu 20.4.2013.

[Gartner, 2013b] Comparison between data integration tools, http://www.gartner.com/technology/reprints.do?id=1-1CYG9N1&ct=121127&st=sb , Tarkistettu 23.5.2013.

[Henry et al., 2005] Henry Scott, Sherlynn Hoon, Meeky Hwang, Diane Lee and Michael D. DeVore, Engineering trade study: extract, transform, load tools for data migration. *Systems and information Engineering Design Symposium (29.4.2005)*, 1-8.

[Hovi, 1997] Ari Hovi, *Data Warehousing – Tietovarastotekniikka*, Gummerus Kirjapaino Oy, 1997.

[IBM, 2013] IBM Information Server, http://www-01.ibm.com/software/data/integration/info_server/, Tarkistettu 20.4.2013.

[Kimball, 1996] Ralph Kimball, *The Data Warehouse Toolkit*. Wiley, 1996.

[Kimball Group, 2013] http://www.kimballgroup.com/2003/09/17/the-bottom-up-misnomer/. Tarkistettu 14.4.2013.

[Kimball & Caserta, 2004] Ralph Kimball and Joe Caserta, *The Data Warehouse ETL Toolkit*, Wiley, 2004.

[Khan et al., 2012] Abeer Khan, Nadeem Ehsan, Ebtisam Mirza and Zahoor Sarwar, Integration between customer relationship management (CRM) and data warehousing, *Procedia Technology 1*, (2012), 239-249.

[Kulkarni et al., 2010] Manashree Kulkari, Meiliu Lu and Du Zhang, A case-based data warehousing courseware. *IEEE IRI 2010*, **August 4-6** (2010), 245-248.

[Kurukunda, 2013] Prakash Kurukunda, Planning the Move, *Best's Review*, Mar2013, Vol. 113, Issue **11**, 71-73.

[Luhn, 1958] Hans Peter Luhn, A Business Intelligence System. *IBM Journal* **2** (4): 314.

[Millbrook, 2013] ETL best practices, http://www.millbrookinc.com/files/Millbrook_ETL_wp_010511.pdf, Tarkistettu 25.5.2013.

[Moody & Kortink, 2000] Daniel L. Moody, Mark A. R. Kortink, From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design, *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000) Stockholm*, Sweden, June 5-6, 2000.

[Morris, 2008] Huong Morris, Bringing Business Objects into Extract-Transform-Load (ETL) Technology, *Conference on e-Business Engineering*, (22-24.10.2008), 709-714.

[Moss & Atre, 2003] Larissa T. Moss, Shaku Atre, *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*, Addison-Wesley Professional, 2003.

[Mundy et al., 2006] Joy Mundy, Warren Thornthwaite and Ralph Kimball, *The Microsoft Data Warehouse Toolkit: With SQL Server 2005 and the Microsoft Business Intelligence Toolset*, Wiley Publishing, Inc, 2006.

[Outlook, 2013] Case about Outlook.com migration process, http://blogs.office.com/b/microsoft-outlook/archive/2013/05/02/outlook-com-400-million-active-accounts-hotmail-upgrade-complete-and-more-features-on-the-way.aspx Tarkistettu 21.5.2013.

[QlickView, 2013] QlickView BI –ohjelmisto, http://www.qlikview.com/fi, Tarkistettu 20.4.2013.

[Rolfe et al., 2001] Gary Rolfe, Dawn Freshwater and Melanie Jasper, *Critical Reflection for Nursing and The Helping Professions*, Basingstoke, U.K: Palgrave, 2001.

[Ruohonen & Salmela, 1999] Mikko J. Ruohonen, Hannu Salmela, *Yrityksen tietohallinto*, Oy Edita Ab, 1999.

[Syncsort, 2013] Syncort and Vertica set new world record, http://www.syncsort.com/Portals/0/Resources/Solution/DMX_Solution_WorldRecord.pdf, Tarkistettu 7.4.2013.

[Schön, 1983] Schön, Donald A., *The Reflective Practitioner : How Professionals Think in Action*, Basic Books, New York, 1983.

[Tilastokeskus, 2012] Suomen virallinen tilasto (SVT): Tietotekniikan käyttö yrityksissä [verkkojulkaisu]. 2012, *Laatuseloste: Tietotekniikan käyttö yrityksissä 2012*. Saatavilla: http://www.stat.fi/til/icte/2012/icte_2012_2012-11-27_laa_001_fi.html, Tarkistettu 25.5.2013.

[UML, 2013] UML-kaavio esimerkki, http://elearn.ncp.fi/materiaali/uimonenj/VirtAMK/anraportti.html, Tarkistettu 25.5.2013.

[Wikipedia tietovarasto, 2013] Tietovaraston kuvaus Wikipedia, http://en.wikipedia.org/wiki/Data_warehouse, Tarkistettu 14.4.2013.

[Wixom & Watson, 2001] Barbara H. Wixom, Hugh J. Watson, An Empirical Investigation of the Factors Affecting Date Warehousing Success, *MIS Quarterly* Vol. **2** No. 1. s. 17-41, Maaliskuu 2001.

[Wu, 2007] Liya Wu, A Service-oriented Architecture for Business Intelligence. In: *IEEE International Conference on Service-Oriented Computing and Applications*, (2007), 279-285.