Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

PASI KANTO

Datan esikäsittely ja ETL-putkien suunnittelu

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
TUTKINTO-OHJELMA
2022

Tekijä(t) Kanto, Pasi	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 25.5.2022
	Sivumäärä 29	Julkaisun kieli Suomi
Datan esikäsittely ja ETL-putkien suunnittelu		
Tutkinto-ohjelma Sähkö- ja Automaatiotekniikka		
Opinnäytetyön keskeisin tavoite on lyhyesti havainnollistaa mitä vaiheita kuuluu datan esiprosessointiin aina datan hankinnasta sen tallentamiseksi tietokantaan. Nykyisin monet päätökset perustuvat suurelta osin jo hankittuun dataan ja siitä tehtäviin johtopäätöksiin, kuten lainat ja vakuutuspäätökset. Se, onko tuo data kuinka tarkkaa ja puhdasta, perustuu suurelta osin juuri esikäsittelyyn ja sen tallentamiseen oikein. Tavoitteena on asian perusperiaatteiden tutkiminen teoreettisesta näkökulmasta ja käyttää esimerkkinä kuvitteellista tietokantaa, johon on tallennettu jäsenneltyä ja puhdistettua dataa. Esimerkkinä käytetty omaa pientä projektia, jonka tarkoitus oli sekä havainnollistaa tekniikan mahdollisuudet mutta samalla tuoda esiin haasteet, jotka se tuo mukanaan. Projektissa on tarkoitus selvittää, onko VR:n junat aina myöhässä ja kuinka paljon ne tuolloin ovat myöhässä. Suomen junaliikenne on runsasta koko maassa, joten aiheen rajaamiseksi valitsin kaupungeista Helsingin.		
Avainsanat Data, tietokanta, ETL-putki, Python, R-ohjelmointi, SQL, No SQL, API		

Author(s) Kanto, Pasi	Type of Publication Bachelor’s thesis	Date 5/25/22
	Number of pages 29	Language of publication: Finnish

Title of publication
Data preprocessing and designing ETL-pipelines

Degree program
Electrical and automation engineering

The main goal of the thesis is to briefly illustrate the steps involved in preprocessing data from acquiring the data to storing it in a database. Today, many decisions are largely based on data already obtained and the conclusions drawn from it, such as loans and insurance decisions. Whether that data is accurate and clean is largely based on pre-processing and storing it correctly.

The aim is to study the basic principles of the matter from a theoretical point of view and to use as an example an imaginary database in which structured and refined data is stored.

I used my own small project as an example, which was intended both to illustrate the possibilities of the technology but at the same time to highlight the challenges it brings.

The purpose of the project is to find out whether VR's trains are always late and how much they are late. As train traffic is very abundant in Finland, Helsinki and not the whole of Finland was introduced as an example city.

Keywords
Data, database, ETL-pipeline, Python, R-programming, SQL, NoSQl, API

SYMBOLI- JA LYHENNELUETTELO

ETL = Extract, transform, load

SQL = Structured query language

NoSQL = Not only structured query language

JSON = JavaScript Object Notation

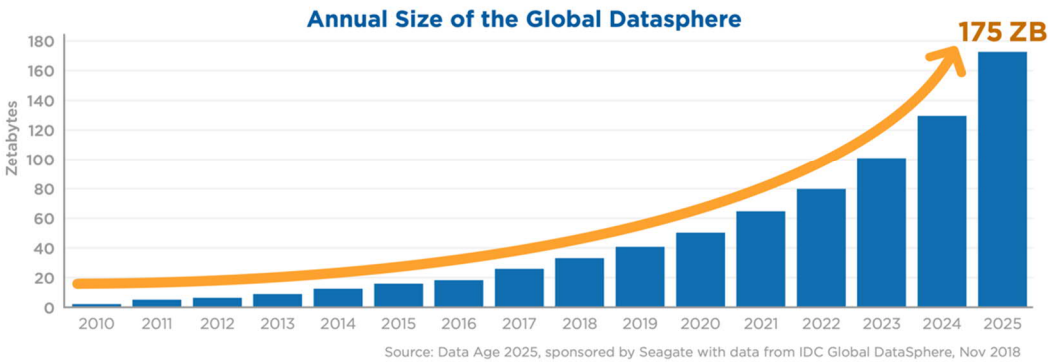
API = Application programming interface

Sisällysluettelo

1 ENSIMMÄINEN LUKU/ JOHDANTO

Tämän vuosituhannen puolella ihmisten tuottaman datan määrä on kasvanut räjähdysmäisesti. Keskiverto ihminen tuottaa dataa keskimäärin 1,5 gigatavua päivässä ja älytehdas, eli automaatiolla toimiva, omaa kuntoa valvova ja tuotantoaan tietojärjestelmien avulla tehostava tehdas, tuottaa sitä yhden petatavun verran. Kaikki tämä on mahdollista kiitos hyvien verkkoyhteyksien, kehittyvien IoT tekniikoiden, älypuhelinten sekä tallennustilan halpuuden ansiota. (Collin & Saarelainen 2016, 43)

Figure 1 – Annual Size of the Global Datasphere



Kuva 1. Datan määrän kasvu 2010–2025. (Seagate www-sivut 2018)

Erilaiset sivustot ja laitteet keräävät ihmisistä tietoa ja tallentavat ne palvelimille, tehtaat tuottavat taukoamatta erilaisista antureista saatuja mittaustuloksia. Puolestaan sosiaalinen media kerää talteen satoja kuvia ja twiittauksia joka sekunti. Tutkimusten mukaan vuonna 2020 maailmalla tuotetun datan määrä on 50 tsettatavua. Tämä tarkoittaa, että ihmiskunta luo kuukaudessa saman verran dataa kuin kirjoitustaidon kehittämisestä oli luotu vuoteen 2003 mennessä. (Merilehto 2018, 129)

Kaikki data ei ole tekstiä ja numeroita, vaan myös kuvia ja ääntä. Myöskään kaikki data ei ole tuotettuna yhtä helposti luettavissa olevaa joka paikassa. Tästä syystä pelkkä datan määrä ei ratkaise, vaan sen on oltava luettavaa ja jollain tavalla myös hyödynnettävää, että sitä kannattaisi säilyttää tai sitä voitaisiin käyttää johonkin. Tästä

syystä dataa täytyy esikäsitellä, puhdistaa, yhdistellä eri datalähteitä, arvoalueet määritellä, puuttuvia arvoja täydentää tai vaihtaa arvot sellaisiksi, jotka voidaan joko laskea tai muuten hyödynnettävään muotoon (Tamminen 2021).

2 ETL-PUTKEN SUUNNITTELU

Extract, transform ja load eli lyhyemmin ETL on kokonaisuus, jossa saatetaan yhteen datan kerääminen, muokkaaminen ja tallentaminen. ETL-putkia käytetäänkin useimmiten datamäärien pienentämisen ja datan luettavuuden parantamisen takia. (Panoplyn www-sivut 2022)

ETL-putkien hyöty perustuu automaation hyväksikäyttöön, jolloin dataa saadaan monesta lähteestä kerättyä saman aikaisesti ja tallennettua suunnitellusti oikeaan paikkaan. Näiden dataputkien käytöstä on tullut käytännössä välttämättömyys kaikille yrityksille, joiden tavoitteena on tiedolla johtaminen. Ilman hyvin suunniteltuja ETL-putkia liiketoiminnan kannalta kriittiset ja vaikeat päätökset joudutaan tekemään ilman luotettavaa dataa. (Databricksin www-sivut 2022)

Suunnittelu lähtee liikkeelle arkkitehtuurisesti suunnittelusta, jossa päätetään raakadatan lähteet, datan muokkaamisesta aina sinne latausvaiheeseen saakka. Suunnittelu on tärkeää, jotta lopullisen datan käyttö olisi mahdollisimman helppoa ja nopeaa. Suunnitteluvaiheessa pystytään vielä tekemään tarvittavia muutoksia ennen toteutusta.

ETL-putkien käytössä on muutamia keinoja, joilla sen saa tehtyä

1. Kartoitus ja ohjeet
2. Käsitteelliset rakenteet
3. Entiteettikartoitus
4. UML-merkinnät

Näistä keinoista toinen ja kolmas keino ovat omiaan kehitettävyyden ja laajentamisen mahdollisuus huomioitaessa. Käytettäessä entiteettikartoitusta ja UML-merkintöjä voidaan kartoitusoperaattoreihin soveltaa entiteetteihin (liitos, leikkaus tai ero) tai entiteettien attribuutteihin kuten yhteen-, vähennys- tai tietotyyppimuunnokset. (Tiwari 2022)

Suunnitteluvaiheessa päätetään, siirretäänkö data jatkuvana virtana, hybridimallilla vai kertasiirrolla. Hybridimalli yhdistää kertasiirron ja virran eli striimin. Tällöin dataa kerätään välipistesäilytykseen, jonka jälkeen kaikki tarvittava tieto siirretään loppusäilytyspaikkaan. Erityisesti IoT-laitteilla tämä on hyvä käytäntö. Esimerkiksi automaatio linjastolla valmistetaan tuotteita ja sensori tarkkailee laatua tutkien jokaisen tuotteen havaiten huonot ja lähettäen jatkuvasti dataa eteenpäin, että nämä huonot tuotteet kerätään pois linjasta. Lopulta tämä data lähetään vielä eteenpäin yhtenä pakettina.

Tietovarastoprojektissa juuri ETL-suunnittelu ja toteutus vievät suuren osan ajasta ja runsaasti resursseja, vaikka kyseinen tietokanta ei olisikaan suuri. Putken suunnittelu vaatii myös testausta ennen lopullista valintaa. On tärkeää tallentaa virheelliset testaukset testin tulosten seurannan ja analysoimisen vuoksi.

3 DATAN KERÄÄMINEN

Dataa kertyy päivittäin suunnattomia määriä. Lähteenä on yleistyneet internet of things eli IoT laitteet, sosiaalinen media sekä tallennustilan yleinen halpuus. Dataa hyödynnetään markkinoinnissa, lääketieteessä, vakuutus- ja pankkialalla, koneoppimisessa sekä rikosten ehkäisyssä. Yritykset ovatkin nyt 2000-luvulla heränneet ymmärtämään datan tärkeyden liiketoiminnalle. Data mahdollistaa yksittäisille yrityksille rahallista säästöä, parempaa laitekannan ylläpito-ominaisuuksia ja liikevaihdon kasvua. (Collin & Saarelainen 2016, 18)

Lääketieteessä käytetään paljon dataa eri muodoissa. Potilastiedot, hoitokertomukset, erilaiset hoitotyön tutkimukset verikokeista painon punnitsemiseen ja röntgenkuvat ovat hyvä esimerkki siitä mihin dataa tarvitaan. Tätä dataa sitten käyttävät joko lääkärit tai hoitajat tai tutkija tai sitten ne menevät tekoälylle joka konenäön avulla pystyy auttamaan lääkäreitä sairauksien havainnoinnissa, sairauden hoidossa tarvittavaan muutokseen tai kutsumaan potilaan tapaamiseen. (Merilehto 2018, 120–121)

Dataa kertyy päivässä suuria määriä, joka käytännössä on arvotonta, ellei sitä jalosteta hyötykäyttöön. Tämä taas on mahdotonta, ellei tiedetä mistä data on lähtöisin, mitä varten se on kerätty tai mihin tarkoitukseen se tulee. (Tikkanen 2021).

Datan saattamisessa hyötykäyttöön on tärkeää ensin selvittää ja suunnitella sen käyttötarkoitus sekä minkälaista dataa halutaan kerätä. Toisin sanoen tarpeen mukaan suunnitellaan datan hyötykäyttöön saattaminen.

3.1 Datan kerryttämisen keinot

Dataa kerätään nykyään kaikkialta ja kaikissa muodoissa. Ihmiset luovuttavat monesti dataa suurille yrityksille ja valtioille edes tietämättään. Nykyiset evästeselosteet ovat usein hyvin vaikeaselkoisia ja käyttäjäehdot ovat yrityksen etua ajatellen tehty. Erityisesti lapsille ja ikääntyville tämä on hankalaa, sillä teksti on vaikeaselkoista luettavaa. (Vänskä, Härkönen, Suomalainen 2020)

Yksi suurista datalähteistä löytyy erilaisista sähköisistä toiminnoista sekä koneisiin ja laitteisiin sijoitetuista antureista. Esimerkiksi jo pelkästään formula-autossa on satoja antureita, jotka lähettävät reaaliaikaista dataa palvelimille ja teknikoille suunnattomat datamassat. (Collin & Saarela 2016, 151–152)

Myös sosiaalisen median yritykset keräävät tietoa ihmisistä kuvien, klikkausten ja julkaisuiden muodossa. Sosiaalinen media saa pelkästään kuvista useimmiten poimittua ihmisten paikkatiedot sekä kuvaukseen käytetyn laitteen tiedot. Sosiaalisen median yritykset tuottavat dataa rakentaakseen tunnettua brändiä, kasvattaakseen myyntiä ja nettisivuillaan liikennettä.

3.1.1 Datan louhinta

Datanlouhinta on yksi keino kerätä dataa. Louhinnan avulla dataa kerätään joko hyödyntäen internetiä suoraan tai hyväksikäyttäen rajapintoja, mutta on huomioitava, että osa datasta on kerättävissä ilmaiseksi, osa on maksumuurin takana. Paljon tietokantoja löytyy julkisista lähteistä mutta on myös salattuja ja ei julkisia tietokantoja. Datanlouhinta ei vaadi tuekseen ohjelmaa joko valmiina tai itse ohjelmoituna. Kaikkein yksinkertaisimmillaan dataa voidaan kerätä lukemalla internetsivuja läpi ja sieltä kopioimalla tietoa omaan tai yleiseen käyttöön. Kyseinen keino on tosin hidas ja kömpelö keino tiedon hankintaa varten.

Yleisimpiä käytettyjä ohjelmointikieliä tiedon louhimista varten ovat Python tai Java. Molemmista löytyy valmiita kirjastoja Pythonilla Selenium tai Beatifulsoup ja Javalla Jsoup tai Selenium. Näissä on tarkoitus löytää selaimella hyödyllisiä tietoja joko

suoraan front-end puolelta eli suoraan HTML-tiedostoformaatista tai sitten back-end puolelta käyttäen hyväksi API: a.

3.1.2 IoT tiedon lähteenä

2000-luvulla esineiden internet, jossa yhdistyy kuluttajille ja teollisuuteen tarkoitetut älylaitteet, on noussut kuumaksi puheenaiheeksi. Esineiden internetin läpimurron on mahdollistanut monien eri asioiden summa, kuten antureiden ja komponenttien hinnan laskeminen sekä verkko- ja analysointitekniikoiden kehitys.

Ruotsalaisen tutkimuksen mukaan jo pelkästään teollisuuden automaatiojärjestelmissä olevat internetiin yhdistyneiden langattomien laitteiden määrä olisi ollut 43,5 miljoonaa laitetta. (Collin & Saarelainen. 2016, 21)

Laitteista ja sensoreista kertynyt data täytyy myös tallettaa johonkin. Tietokannan valinta onkin tärkeimpiä tehtäviä, jotta siitä ei muodostu pullonkaulaa järjestelmässä tai ettei siitä koidu haittaa, kun dataa analysoidaan. (Collin & Saarelainen 2016, 196)

Iot tiedonlähteenä on myös epävarma, koska toimitusketjun kyberturvallisuus voi vaarantua missä vaiheessa tahansa ja aiheuttaa haasteita erilaisten toimijoiden kanssa. Näihin haasteisiin toki voidaan ennakoivasti puuttua. (Kettunen 2020.)

3.1.3 Asiakaskyselyt ja tutkimukset

Myös erilaiset asiakaskyselyt ja palautteet ovat hyvä lähde tiedolle. Lähes jokaiseen kauppaan on nykyään sijoitettu se pieni laite, jossa on kysely kaupassakäyntikokemuksesta. Moni yritys toivoo ja haluaa asiakaspalautetta, jolla kehittää liiketoimintaa.

Tunnettuna esimerkkinä hyvin toteutetusta ja asiakkaita hyvin palvelevasta kyselystä on Netflixin keräämä palaute. Palaute annettiin alun perin asteikolla 1–5, mutta

muutettiin myöhemmin muotoon peukku ylös tai alas. Netflixin kysely tavoittaa keskimäärin 100 miljoonaa käyttäjää. (Merilehto 2018, 35) Yksinkertainen palautteen antamisen muoto kannustaa asiakkaita vastaamaan kyselyyn.

3.2 Datan eri muodot

Vaikka dataa kerätään monista eri lähteistä, on tieto muodoltaan jäsenneltyä eli structured, semistrukturoitua eli semi-structured tai jäsentelemätöntä eli nonstructured. Relaatiotietokannat ovat hyvä esimerkki jäsennellystä datasta. Siinä tiedot kerätään riveihin ja sarakkeisiin, joilla tiedon saa helposti jäsenneltyä ja se on nopeasti haettavissa indeksien avulla. Selkeimpinä vaihtoehtoina on erilaiset SQL-tietokannat tai Excel-taulukot. Usein yrityksissä keskitytään strukturoidun datan analysoimiseen sen helpon käytettävyyden vuoksi.

Semistrukturoitu data on myös osittain jäsenneltyä. Hyvänä esimerkkinä xml eli extended markup language, joka on järjestettynä loogisella rakenteella käyttäen itse määritettyjä elementtejä. Xml-kieltä käytetään rajapinnoissa tiedonvälitykseen sovellusten välillä. Semistrukturoitu data voi myös olla avainsanoilla varustettua videokuvaa. Tällöin itse video on strukturoimatonta dataa, mutta avainsanat luovat useista videoista koostuvalle datamassalle struktuurin.

Jäsentelemätön data saattaa sisältää kuvaa, ääntä, tekstiä tai kaikkea tätä ja se on ennalta määrittelemätöntä dataa. Hyvänä esimerkkinä on ihmisten sosiaalisessa mediassa luodut tiedot: kuvat ja videot, kuten esimerkiksi Facebookissa tai Instagramissa.
.

3.3 Työkalut datan esikäsittelyssä

Datan esikäsittelyyn on monia työkaluja. Osa sovelluksista on ilmaisia kuten Open Office tai OpenRefine tai sitten maksullisia kuten Excel. Nämä sovellukset ovat tarkoitettu taulukoiden muokkaamiseen, siivoamiseen sekä analysointiin. (Helsinki Region Infosharen www-sivut, 2022)

Ohjelmoinnista on hyötyä, kun dataa aletaan käsittelemään ja käytännöllisimmät ohjelmointikielet tähän ovat Python ja R. R-ohjelmointikieli on kehitetty nimenomaan tilastolliseen laskentaan ja datan visualisointiin ja se pitää sisällään lukuisia kirjastoja koskien datan putsausta ja analysointia. (Weston & Yee 2022)

Python taas on yleishyödyllinen kieli, jossa on laajempi kirjastokokonaisuus kehitetty pelkästään datan analysointia ja visualisointia varten. Erityisesti Pandas, Numpyja ja matplotlib ovat tähän tarkoitukseen sopivia (Karczewski 2021).

Oracle tarjoaa valmiin sovelluksen nimeltä Oracle Data Integrator, jossa on graafinen käyttöliittymä, ETL-putkien luomiseksi alusta loppuun. Oraclen työkalu tarjoaa myös dokumentaation, visuaalisen työkalun datan liikkeiden tarkkailuun sekä graafisen käyttöliittymän. (Oracle-www sivut 2022)

3.4 Datan siistiminen

Maailmalta kerätty data on lähes aina epäpuhdasta ja vaatii siistimistä, jotta siitä olisi hyötyä jatkokäsittelyä varten. Datan siistimisen tarkoitus on poistaa datasta tyypillisiä merkityksetöntä tai väärää tietoa. Epäjohdonmukaisuuksia esiintyy aika ajoin ja niiden poimiminen voi olla hyvinkin vaikeaa, sillä nämä tiedot eivät periaatteessa ole ”väärin”, mutta ei vain ole mahdollista niitä käyttää. Esimerkiksi kuukaudet ja päivämäärät merkitään eri tavalla Yhdysvalloissa ja Euroopassa. Yhdysvalloissa merkataan aika muodossa kk/pp/vvvv ja Euroopassa pp/kk/vvvv.

Samaten numeerisille arvoille, esimerkiksi ikäryhmät, on helpompi antaa yksi numeroarvo sen sijaan, että ne olisivat vain arvojoukko. Epäselvyyksien välttämiseksi täytyy nämä tiedot yhdistää ja saattaa samaan formaattiin. (Withee 2010)

3.5 Puuttuvien tietojen korvaaminen

Dataa jää monesta syystä puuttumaan. Joko tietoa ei ole, tieto puuttuu inhimillisen virheen takia tai dataan on lisätty rivejä, mutta alkuperäisiin ei ole uusia arvoja. Suurimmilta osin puuttuvat arvot on merkattu data frameihin merkinnällä NaN tai sitten vain tyhjänä ruutuna.

Puuttuviin arvoihin ja niiden käsittelyyn on oikeastaan kaksi tapaa. Nämä rivit voidaan joko kokonaan poistaa tai sitten korvata puuttuvat arvot jollain.

Suurissa taulukoissa, joissa on tuhansia rivejä, voidaan rivit poistaa kokonaan, koska näiden vaikutus lopputulokseen on kovin pieni. Korvaavaksi arvoksi voidaan käyttää joko eniten käytetty arvo, keskiarvo tai käyttää todennäköisintä arvoa, mutta tämä edellyttäisi joko päätöspuuta tai jotain muuta koneoppimisen tekniikkaa.
(Rasku 2017)

3.6 Virheellisten arvojen ja poikkeamien korjaaminen

Arvojen poikkeama tarkoittaa sitä, että taulukossa esiintyy muista arvoista liian paljon eroava data, jolloin se ei ole enää uskottava arvo. Virheellinen data tarkoittaa, että arvot eivät ole uskottavia tai muuten mahdollisia eli ovat jollain tapaa epänormaaleja. Lisäksi samasta arvosta tai riveistä saattaa olla tullut kaksoiskappaleita. Virheellisen datan käyttö heikentää datataulukoiden arvoa. Siksi on tärkeää puuttua virheellisten arvojen korjaamiseen, jotta taulukoiden arvo säilyy ja siitä on hyötyä käyttäjälle. Virheellisten ja poikkeavien arvojen määrä kasvaa mitä suuremmista taulukoista on kysymys. (Vasarhelyi, Kogan & Tuttle 2015, 381-396.)

3.7 Lähteiden luotettavuus ja datan laatu

Dataa on tarjolla nykyään paljon ja sitä on saatavilla monesta paikkaa. Avoimia datalähteitä on tarjolla pilvin pimein ja netistä saa kerättyä suoraan eri rajapinnoista. Avoimen datanlähteistä onkin syytä valita jokin tunnettu lähde käytettäväksi. Suomessa avoimia datalähteitä ovat ainakin tilastokeskuksella, VR:llä, Trafi sekä Ilmatieteen laitoksella.

Datan laadusta vastaa datan tuottaja aina ja jos se on virheellistä, saattaa siitä aiheutua imago-ongelmia sen tuottajalle, siksi on tärkeää, että data sisältää mahdollisimman vähän virheitä. Vanhentunut tai virheellinen tieto tulisi joko poistaa tai korjata.

Avoimen datan tarjoaja usein tarjoaa myös metadataa mistä on apua, kun halutaan tietää miten dataa voi käyttää oikein. (Avoin data-www sivut 2022)

Osa datasta on maksumuurin takana, jolloin sen hankinta vaatii maksukykyä tai yrityksen panostusta. Maksullista tietoa on tarjolla paljonkin, mutta sen ongelma on siinä, että ei ole tietoa missä muodossa tuo data on ja mitä tietoa tuolla maksulla on saatavilla. (Koski, Honkanen, Luukkonen, Pajarinen & Ropponen 2017)

Internetistä kerättäessä tietoja on oltava tarkka, että noudattaa palveluehtoja ja tietosuojalainsäädäntöä. Monella sivustolla on tietojen haravoinnista joko kielto tai rajoituksia käyttäjäehdoissa, jotka joko rajoittavat tai kokonaan kieltävät sivustolta kerätyn tiedon käytön.

4 DATAN MUUNTAMINEN MALLINNUSTA VARTEN

4.1 Normalisointi

Kerättäessä dataa monesta lähteestä saadaan dataa, jonka arvoväli saattaa olla hyvinkin erilainen ja näin ollen ei ollenkaan verrokkikelpoista.
Datan normalisoinnilla on tarkoitus skaalata arvot käyttämään samaa arvoväliä, jotta eri lähteistä saatu data on vertailukelpoista. Data muutetaan noudattamaan kapeaa arvoväliä 0–1. (Rasku 2017)

4.2 Tiedon vähentäminen

Tiedon vähentäminen tarkoittaa, että datasta poistetaan rivejä tai sarakkeita, jotka eivät ole oleellisia, joista puuttuu liikaa arvoja, joiden korrelaatio on liian samankaltaista tai joilla liian alhainen varianssi. Dataa poistettaessa täytyy olla tarkkana, ettei datan analyyttinen arvo katoa. Tällöin datasta saatava hyöty jää saamatta kokonaan tai tulokset ovat vääristyneitä.

Liiallinen datan volyymi hidastaa myös datan prosessointia ja ajoittain aiheuttaa ongelmia myös tallennuskapasiteetin riittävyyden kanssa. Prosessoritehot riittävät datan laskentaan, mutta siihen tarvittava aika kasvaa datan määrän kasvun myötä. Myöskään datan pakkaaminen ei ole ratkaisu, sillä se vaatii myös aikaa ja resursseja ja datan analysointi hidastuisi tästä syystä entisestään. (Andrian 2013)

Varsinkin IoT-laitteet tuottavat suuret määrät dataa, kun sensorit mittaavat jatkuvasti ja lähettävät tietoa palvelimille. Näissä tapauksissa volyymi on kasvanut suunnattomiin mittakaavoihin ja voi hyvinkin tuottaa tuhansia rivejä dataa päivässä. Kyseinen määrä on ihmiselle mahdoton tehtävä lukea ja sisäistää.

5 DATAN LATAAMINEN TIETOKANTAAN

5.1 Tietokantatyypin valinta

Kun data on ensin hankittu ja se on siistitty ja muokattu käyttöön sopivaksi, täytyy se myös tallentaa käyttöä varten. Datan voi varastoida joko yhteen paikkaan tai sen voi hajauttaa eri palvelimille, joka on toki paljon luotettavampi tapa datan saatavuuden ja eheyden takia. Varastoinnin tarkoitus on pitää data mahdollisimman nopeasti luettavissa ja kirjoitettavissa olevana ja jotta järjestelmän toiminta on mahdollisimman luotettavaa. Tietokantatyypin valinta vaikuttaa siihen missä muodossa data lopulta tallennetaan.

Tyyppiä valittaessa on tärkeää ottaa huomioon, minkä kaltaista tietoa tallennetaan, sillä kaikki tyypit eivät ole ollenkaan sopivia keskenään ja tietokantatyypin vaihtaminen kesken projektin on äärimmäisen vaikeaa ja aikaa vievää.

5.1.1 Rakenteellinen tietokanta

Rakenteellisella tietokannalla tarkoitetaan useimmiten relaatiotietokantoja, joista tunnetuin lienee SQL eli structured query language. SQL-kielelle tyypillistä on ennalta tarkkaan määritetty rakenne sekä taulukoiden muodostamat suhteet. Taulukot ovat yhteneväisiä ja tarkkoja ja ne sopivat hyvin yritysten liiketoiminnan analysointiin.

Rakenteellisen tietokannan valinnalla on paljon vahvoja puolia. Kyseinen keino on hyvin vanha tapa tallentaa tietoa, joten osaamista löytyy ja sitä on myös tutkittu ja käytetty paljon. Tietokantoihin on myös helppo lisätä dataa ja samoin poistaa sitä.
Yksi tärkeimmistä ominaisuuksista on helppo luettavuus ja hallinta sekä vuosikymmenten saatossa omaisuuksia on kehitetty lisää vastaamaan tarpeita.

Rakenteellisen tietokannan huonoina puolina voidaan pitää joustamattomuutta ja skaalautumattomuutta. Jo olemassa olevien tietokantojen mallin muuttaminen on hankalaa ja tietokantoihin tiedon lisääminen vertikaalisesti on helppoa, mutta silti ne eivät laajennu horisontaalisesti yhtä helposti.

Toinen huono puoli on kustannuskysymys, sillä vaikka ammattilaisia on paljon ja helposti saatavilla, on heidän palkkaamisensa kallista. Relaatiokantojen ylläpito vaatii kuitenkin paljon ammattitaitoa ja osaamista.

Kuva 2. Esimerkki rakenteellisen tietokannan skeemasta

5.1.2 Semirakenteellinen tietokanta

Puolirakenteellinen tietokanta sijoittuu nimensä mukaan rakenteellisen ja ei rakanteellisen tietokannan välimaastoon. Hyvänä esimerkkinä voi pitää HTML-dokumenttia tai sähköpostia, jossa tiedot tallennetaan noudattamaan jotain tiettyä määrittelyä tai johdonmukaisuutta, mutta kuitenkin niin, ettei se sisällä samankaltaista jäykkää rakennetta kuin relaatiotietokanta. (Marr 2019)

Sähköpostissa on esimerkiksi aina lähettäjä ja vastaanottaja ja aikaleima, mutta itse postin sisältö saattaa olla mitä tahansa aina kuvista tai tekstistä pelkäksi liitetiedostoksi.

5.1.3 Rakenteeton tietokanta

Rakenteeton tietokanta NoSQL eli Not only sequal query language on tarkoitettu tietokannoille, jotka eivät perustu perinteiselle relaatiomallille. NoSQL-tietokannoille

sopiva käyttötarkoitus on silloin, kun tarvitsee tallentaa ääntä, kuvaa ja tekstiä sekaisin.

Tietokantakehitys on ollut suurten yritysten kuten Google, Facebook ja Amazon ansiota. Näiden toimijoiden tarpeet eivät täyttyneet perinteisimmillä tietokantamalleilla, sillä dataa kertyi niin paljon.

NoSQL tallentaa dataa kolmella eri tavalla: dokumenttitietokanta, avain/arvoparitietokantana sekä saraketietokantana. Näistä avain/arvoparitietoa haetaan avaimilla. Tietokanta on perinteisiä relaatiomalleja helpommin laajennettavissa ja haku toimii nopeammin.
.
Sarakemallissa data tallennetaan tauluihin aivan kuten relaatiomallissa, mutta rivien sijaan tiedot tallennetaan sarakkeisiin. Sarakemallissa haku on huomattavasti nopeampaa, koska koko taulua ei tarvitse käydä läpi. Tarvittavat tiedot voidaan hakea yhdestä sarakkeesta. (Stonebraker, Abadi, Batkin, Chen, Cherniack & Ferreira 2005)

Dokumenttivarastossa tiedot tallennetaan asettamalla tiedolle jokin avainarvo, jolla tietoja voidaan hakea. Toisin kuin avain-arvotietokannat voidaan arvoksi asettaa osoittimia, listoja tai vieläkin monimutkaisempia tietorakenteita. Dokumenttivarastojen vahvuus onkin nimenomaa talletuksen vapaus. Dokumenttivaraston heikkoutena voidaan sen sijaa pitää hakujen nopeutta. Dokumenttivarastosta dataa voidaan hakea avaimen lisäksi myös arvolla itsellään.

5.2 Tietojen lataaminen

ETL-putkien luomisessa viimeinen vaihe on yksinkertaisesti datan siirtäminen valittuun tietokantaan noudattaen alkuperäistä rakennesuunnitelmaa. Lataus aloitellaan jo siinä vaiheessa, kun osa tiedoista on valmiina eli toinen ja ensimmäinen vaihe on vielä kesken. (Mohammed 2021)

Tiedot yleensä lisätään vanhojen tietojen päälle taikka taulukon perään riippuen tietotyypeistä tai suunnitelmasta. Latausta varten SQL-kielellä on INSERT-komento, jota voi käyttää tai Oraclella on valmis työkalu myös tätä vaihetta varten

Suunnitteluvaiheessa on myös tärkeää huomioida datan lataamisen ajankohta, jotta sen aiheuttama kuormitus palvelimille ei olisi liian suuri. Useimmiten siirto tapahtuu ilta- tai yöaikaan, koska monissa sovelluksissa on jonkinlainen ajonhallintaohjelmisto tätä ohjaamaan. Osa datasta saattaa olla ajankohtaista, jolloin siirtoja tehdään pitkin päivää.

Siirrosta tehdään myös lokitiedosto, josta näkee minkä verran rivejä on siirretty ja kauan siirtoihin meni aikaa. Näistä muodostuu metatietoa käyttäjille. (Hovi, Ylinen, Koistinen 2009, 54)

6 KÄYTÄNNÖSSÄ

Oma käytännön projekti liittyi usein otsikoissa olevan VR:n liikenteen myöhästelyyn ja haluun luoda uniikki projekti itselle omaan portfolioon. Tavoitteena projektissa tuottaa suunnaton arvo työelämään siirtymisen kannalta. Projektin tietolähteinä halusin käyttää kotimaisia ja luotettavia lähteitä, jotta tiedon oikeellisuus olisi varmistettu.

Käytännön työ aloitettiin suunnittelemalla tiedon lähteet, joita käytetään projektissa. Sen lisäksi rajattiin hakukohteiden käyttö sisältämään vain yhden kaupungin aikataulut helpottamaan projektin hallittavuutta. Data päätettiin myös ladata kerralla koko päivän tiedot, sen sijaan, että olisi käytetty jatkuvaa datan virtaa: Tämä valinta vähensi puuttuvien tietojen määrän, joka haettiin rajapinnasta.

Suunnittelussa oleellista oli myös tietokantatyypin valinta. SQL-tietokanta oli tässä tapauksessa selvästi paras ratkaisu. Työkalujen valintana suunnittelussa otettiin huomioon oma osaaminen ja mitkä työkalut olivat jo entuudestaan tuttuja. Samalla päätin hyödyntää avoimen lähdekoodin ohjelmia, jotka ovat ilmaisia. Jupyter Notebook valittiin tästä syystä alustaksi datan siistimistä varten ja muihin ohjelmointia vaativille tehtäville. Kyseinen sovellus pitää jo valmiiksi sisällään tarvittavat kirjastot ja niiden liitännäiset kuten Pandas ja Seaborn kirjastot. Tietokantatyökaluna SQLite sopi tehtävään hyvin, koska se on riittävän pieni tietokanta ja se on linkitetty suoraan sovellukseen, jolloin erillistä ODBC-yhteyttä ei tässä tapauksessa tarvita käyttöönotossa.

Tietokannasta haluttiin yksinkertainen ja helposti luettava, joten suunnitelmissa tämä otettiin huomioon ottamalla vain välttämättömät tiedot. Suunnitelmassa käytettiin UML-sovellusta, joka auttoi rakentamaan skeeman, jota projektissa käytettiin.

Projektissa tiedot kerättiin suoraan Digitraficin rajapinnasta osoitteesta "https://rata.digitraffic.fi/api/v1/live-trains/station/HKI", jolloin erillistä sovellusta ei tarvittu vaan käyttöön riitti pelkkä Jupiter Notebook-ohjelmointiympäristö ja siellä requests-kirjasto, joka antaa tehdä rajapintakyselyitä verkkosivustoille. Kyseinen rajapinta on vapaasti kaikkien käytössä olevaa avointa dataa. Ohjelmointikielenä käytettiin Python3-kieltä, koska se on itselle paljon tutumpi kieli kuin Java tai JavaScript.

```
[1]: import pandas as pd
     import numpy as np
     import requests
     import sqlite3

     url = "https://rata.digitraffic.fi/api/v1/live-trains/station/HKI"

     req = requests.get(url=url)
     req.status_code
     data = req.json()
     taulukko = pd.DataFrame(data)
```

Kuva 3. Sisältää datan hakemista varten kirjoitetun koodi osuuden

Projektissa data haetaan tunnetulta verkkosivustolta, joten se on luotettavaa ja luokiteltu avoimeksi datalähteeksi eli se on myös ilmainen ja sitä saa vapaasti käyttää, jolloin sen käyttäminen oikein on helpompaa. Sivusto ei ole myöskään vielä estänyt jatkuvia hakuja tietystä osoitteesta. Näin yritykset voivat toimia, mikäli tietystä osoitteesta tulee paljon kyselyitä.

Metadataa ei saanut ladattua suoraan tuolta kohtaa ja se vaikeuttaa hiukan asiaa, koska nyt ei aukea asemakoodit tai mitä merkitystä joillakin sarakkeilla oli. Tämä vahvistaa metadataa ja sen käytön tärkeyttä. Kyseiseen kohtaan olisi voinut myös merkitä peruuntumisen syiden koodiston, joka avaisi dataa enemmän. Nyt tiedetään vain, että jokin junavuoro oli peruttu, mutta ei syytä siihen.

Data haetaan sivustolta JSON-muodossa ja sen jälkeen muutetaan Pandas-kirjaston avulla dataframiksi. Pelkkä dataframiksi muutto ei kuitenkaan riitä, koska taulukossa on soluja, joiden arvona on toisia taulukoita. Projektin lopulliset tiedot haettiin juuri näistä soluista. Kyseinen solu piti sisällään aina yhden junan koko aikataulun, joten nyt saatiin tarkempaa tietoa millä pysäkeillä tuli viivettä ja saatiinko jollain pysäkillä aikaa kurottua kiinni.

Datan siistiminen oli projektissa aikaa vievä vaihe. Metadataa ei ollut juurikaan tarjolla, joten kaikki taulukossa käytettävät lyhenteet eivät auenneet tiedostoja lukiessa. Aikataulu piti myös sisällään junia, jotka eivät olleet matkustajakäytössä vaan olivat muussa käytössä, kuten tavarajunia. Nämä junat aiheuttivat eniten harmia, sillä ne sisälsivät paljon tyhjiä arvoja. Tavarajunista puuttui monesti myös actualTime, joten ne karsittiin lopullisesta taulukosta tiputtamalla kaikki junat pois, joilla ei tuota arvoa ollut määritelty.


{"stationShortCode":"JVS","stationUICCode":1272,"countryCode":"FI","type":"ARRIVAL","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:08:00.000Z","causes":[]},
{"stationShortCode":"JVS","stationUICCode":1272,"countryCode":"FI","type":"DEPARTURE","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:08:00.000Z","causes":[]},
{"stationShortCode":"HVS","stationUICCode":1021,"countryCode":"FI","type":"ARRIVAL","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:18:00.000Z","causes":[]},
{"stationShortCode":"HVS","stationUICCode":1021,"countryCode":"FI","type":"DEPARTURE","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:18:00.000Z","causes":[]},
{"stationShortCode":"OV","stationUICCode":190,"countryCode":"FI","type":"ARRIVAL","trainStopping":true,"commercialStop":true,"commercialTrack":"2","cancelled":false,"scheduledTime":"2022-04-29T14:29:00.000Z","causes":[]},
{"stationShortCode":"OV","stationUICCode":190,"countryCode":"FI","type":"DEPARTURE","trainStopping":true,"commercialStop":true,"commercialTrack":"2","cancelled":false,"scheduledTime":"2022-04-29T14:30:00.000Z","causes":[]},
{"stationShortCode":"TRK","stationUICCode":1283,"countryCode":"FI","type":"ARRIVAL","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:38:00.000Z","causes":[]},
{"stationShortCode":"TRK","stationUICCode":1283,"countryCode":"FI","type":"DEPARTURE","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:38:00.000Z","causes":[]},
{"stationShortCode":"TV","stationUICCode":1270,"countryCode":"FI","type":"ARRIVAL","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:42:00.000Z","causes":[]},
{"stationShortCode":"TV","stationUICCode":1270,"countryCode":"FI","type":"DEPARTURE","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:42:00.000Z","causes":[]},
{"stationShortCode":"LÄP","stationUICCode":203,"countryCode":"FI","type":"ARRIVAL","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:47:00.000Z","causes":[]},
{"stationShortCode":"LÄP","stationUICCode":203,"countryCode":"FI","type":"DEPARTURE","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:47:00.000Z","causes":[]},
{"stationShortCode":"LPR","stationUICCode":1149,"countryCode":"FI","type":"ARRIVAL","trainStopping":false,"commercialTrack":"","cancelled":false,"scheduledTime":"2022-04-29T14:53:00.000Z","causes":[]},

Kuva 4. Haettu data ennen siistimistä ja muokkausta

Projektissa kerätty data tuli yhdestä ja luotettavasta lähteestä. Sivustolla on myös sertifikaatti eli se tarjoaa myös turvallisen yhteyden sivustolle. Kyseinen sivusto on valtion ylläpitämä sivusto, joka antaa sieltä haetulle tiedolle lisäarvoa.

Projektissa työkaluiksi valittiin Python3 ohjelmointikieli, joka on itselle kaikkein tutuin ohjelmointikieli. Anaconda-sovellus käytettiin koodausalustaksi ja SQLite tietokannan hallinnan työkaluksi. SQLite valittiin, koska se toimii hyviin pieniin tietokantoihin sekä se ei vaadi erillistä hallintaa ja on siksi helppokäyttöinen.

Projektissa arvoja skaalattiin näyttämään vain päivämäärä, kuukausi sekä kellon aika. Näin saadaan riittävän tarkkoja ennusteita siitä, onko jokin junan myöhästyminen mahdollista ensi kerralla vai kulkeeko juna ajallaan. True/False arvot muutettiin laskettavaksi 1/0 arvoiksi, jotta niitä voidaan myöhemmin hyötykäyttää, kun otetaan käyttöön koneopin malleja.

Virheellistä dataa etsittiin aikapoikkeamista visualisoimalla dataa. Tällä keinoilla saatiin etsittyä arvoja, jotka selkeästi poikkesivat muista arvoista. Näitä virheellisiä arvoja etsittiin tarkastelemalla lähtöajan arvoa, joka tulkittiin taulukossa nolla-arvoksi ja siitä laskemalla lähtöajan ja pysäkillä olevan ajan erotusta. Tässä jouduttiin käyttämään tietysti oletusta, jossa junan ei oleteta olevan reilusti myöhässä yhdellä pysäkillä ja olevan täysin ajoissa edellisillä ja seuraavalla pysäkillä, jos pysäkkien välit ovat vain

muutaman minuutin matkalla. Virheellisten arvojen etsintä on projektissa haastavaa, koska junat saattoivat hajota tai tulla muita syitä miksi ne eivät pääse perille, jolloin poikkeavien arvojen tarkastelu on tästä syystä vaikeaa, kun ei koskaan tiedä onko trainstopping oikeasti tosi vai ei.

Datasta poistetaan sarakkeita runsaasti, koska niitä ei tarvita ja koska sarakkeessa timeTableRows sisälsi suurimman osan tarvittavaa tietoa pois lukien trainNumber sarake. Jäljelle jätetään käytettäväksi ainoastaan stationShortCode, type, trainstopping, cancelled, scheduledTime, actualTime ja differenceInMinutes. Vain näillä riveillä oli myöhästymiseen liittyviä syitä ja samalla näkyy suoraan myöhästyminen minuutteina.

[5]:

	stationShortCode	type	trainStopping	cancelled	scheduledTime	actualTime	differenceInMinutes	liveEstimateTime
0	LPV	DEPARTURE	True	False	2021-12-08T18:56:00.000Z	2021-12-08T18:56:13.000Z	0.0	NaN
1	8201	ARRIVAL	True	False	2021-12-08T18:57:30.000Z	2021-12-08T18:58:18.000Z	1.0	NaN
2	8378	DEPARTURE	True	False	2021-12-08T18:58:00.000Z	2021-12-08T18:58:42.000Z	1.0	NaN
3	8383	ARRIVAL	True	False	2021-12-08T18:59:30.000Z	2021-12-08T18:59:57.000Z	0.0	NaN
4	8494	DEPARTURE	True	False	2021-12-08T19:00:00.000Z	2021-12-08T19:00:22.000Z	0.0	NaN

Kuva 5. Sisältää lopullisen datataulukon 5 ylintä riviä ennen poistoa

Projektissa data lisätään suoraan tietokantaan, joka tarkastellaan päivän tai ajon päätteeksi.

Taulukoista jäi arvoja puuttumaan harvemmin, koska koko taulu on haettu päivän päätteeksi ja suurin osa tiedoista on jo keretty lisäämään tai korjaaman. Puuttuvat arvot korvattiin aikatauluissa joko aikaisempien päivien aikataululla tai vastaavasti jos tarkka-aika eli actualTime puuttui. Tämä arvo saatiin lisäämällä edellisten pysäkkien aikaerotuksen keskiarvo, joka lisättiin scheduledTime arvoon. Näin saatiin kohtalaisen tarkkoja arvioita junien aikataulusta.

Tietokantaan haluttiin lisätä tuloksia vanhojen arvojen perään, koska se oli käytännöllistä ja ei vaatinut huomattavasti enempää laskentatehoja tai resursseja ja latausajat olivat siedettävän lyhyet, johtuen tietokannan käyttäjämäärän pienuudesta ja se että tiedot sijaitsivat fyysisesti samalla koneella. Tietokanta toteutettiin relaatiotietokantana, koska kaikki tiedot haluttiin olevan helposti haettavissa ja tiedot olivat selkeästi

yhteydessä toisiinsa. Jokaisella junalla oli oma yksilöivä tunnistetieto, jota käytettiin pääavaimena relaatiotietokannassa.

Muodostettua putkea on tarkoitus päivittää automaattisesti. Normaalisti tämä päivittäminen voidaan hoitaa automaatiolla, joka vähentää tarvittavaa rutiinityön määrää, kun huolehditaan kerätystä datasta ja sen sijoittelusta tietokantavarastoon. Tällä hetkellä automaation virka hoituu sillä, että sovellus on ohjelmoitu ajamaan itseään aina 86400 sekunnin sykleissä eli 24-tunnin välein.

Käyttöön olisi voinut ottaa myös kyseiseen tarkoitukseen kaupallisessa tarkoituksessa olevia palveluita, vaikkapa Googlelta tai Microsoftilta, mutta nämä palvelut ovat maksullisia. Valmiita työkaluja koko ETL-putken tekemiseen olisi myös ollut tarjolla, mutta niiden käyttöä haluttiin välttää, jotta saadaan parempi kokonaiskuva mitä yksittäisen kokonaisuuden rakentaminen ja ratkaiseminen vaati. Kehitettävää jäi vielä paljon ja vaikka ihan perusteiltaan putki on onnistunut, ei sen hyötykäyttö ole kuin vain aivan minimaalinen.

Projektissa lopputuloksena saatiin päivittäin itsestään verkkosivulta tiedot hakeva, itsenäisesti datan putsaava ja oikeaan tietokantaan tallentava ETL-putki, joka toimii testissä hyvin.

7 JÄLKIPOHDINTA

Datasta sekä sen taltioinnista on tullut tärkeä osa meidän yhteiskuntaamme. Sen vaikutukset jokapäiväiseen elämään ovat poikkeuksellisen suuret, vaikka sitä ei välttämättä päällepäin näekään. Datan käyttö liiketoiminnan, lääketieteen, tekniikan ja vaikka elokuvien valinnan työkaluna on tärkeänä osana päätöksentekoa. Tekoäly, koneoppimisen mallit sekä BI-sovellukset vaativat runsaasti puhdasta dataa toimiakseen.

Hyvin toteutettua dataputkia on helppo hyötykäyttää pitkälle tulevaisuuteen. Jokainen dataputki on kuitenkin uniikki. Valmiita dataputkia ei ole ja suunnittelu on aloitettava alusta joka kerta. Markkinoilta löytyy onneksi runsaasti työskentelyä helpottavia työkaluja, mutta silti manuaalista työtä joutuu aluksi tekemään paljon ennen kuin automaation saa ajamaan koodia itsenäisesti.

Koko pienen projektin tärkein päämäärä oli osoittaa datan esikäsittelyn ja siistimisen tärkeys ja kuinka pelkällä raakadatalla ei päästä kovinkaan pitkälle. Datan esikäsittelyssä joutui tekemään paljon pohdintaa ja päättelemään, mitkä annetuista arvoista ovat tärkeitä ja mitkä eivät. Monesti taulukoista joutui poistamaan ja sitten palauttamaan rivejä ja sarakkeita, kun lopputulokset alkoivat vaikuttaa epäsopivilta tai epäloogisilta kokonaisuutta ajatellen.

8 LÄHTEET

Abhishek T. 2018. ETL workflow modeling. Viitattu 20.3.2022. Saatavissa: https://www.abhishek-tiwari.com/etl-workflow-modeling/

Alexandru A. 2013. Big data challenges. Viitattu: 7.12.2021. Saatavilla: https://www.dbjournal.ro/archive/13/13_4.pdf

Arsalan M. 2021. Viitattu 12.12.2021. Saatavilla: https://hevodata.com/learn/data-loading/

Avoin Data www-sivut 2022. Organisaation jäsenet ja ylläpitäjä. Viitattu: 7.12.2021 Saatavilla: https://www.avoindata.fi/fi/kayttoohjeet/yllapitajan-rooli#yllapitaja

Bernard M. 2019. What's the difference between structured, semi-structured and unstructured data? Viitattu 7.12.202. Saatavilla:https://www.forbes.com/sites/bernardmarr/2019/10/18/whats-the-difference-between-structured-semi-structured-and-unstructured-data/?sh=162c9a492b4d

Collin J. & Saarelainen A. 2016. Teollinen Internet. Jyväskylä: Talentum

Databricks. 2022. Extract Transform Load (ETL). Viitattu 20.2.2022. Saatavilla: https://databricks.com/glossary/extract-transform-load

Helsinki region infoshare 2017. Työkalut. Viitattu 5.12.2021. Saatavilla:https://hri.fi/fi/ohjeet/datan-hyodyntajalle/tyokalut/

Hovi A, Hervonen H. & Koistinen H. 2009 Tietovarastot ja business intelligence. Jyväskylä: WSOYpro.

Karczewski D. 2021. Python for data-analysis. Viitattu 14.3.2022. Saatavilla: https://www.ideamotive.co/blog/python-for-data-analysis

Kettunen, M. 2020. IoT-järjestelmien kyberturvallisuutta parantamassa. Digitaalinen talous 4/2020. XAMK. Viitattu 3.5.2022. Saatavissa: https://read.xamk.fi/2020/digitaalinen-talous/iot-jarjestelmien-kyberturvallisuutta-parantamassa/

Koski H., Honkanen M., Luukkonen J., Pajarinen M. & Ropponen T. 2017. Avoimen datan hyödyntäminen ja vaikuttavuus. Viitattu 6.12.2021 Saatavilla: https://tietokayttoon.fi/documents/10616/3866814/40_avoimen+datan+16032017.pdf/0444467d-5400-4f0c-8728-2447cef039ad,

Merilehto A. 2018. Tekoäly matkaopas johtajalle. Helsinki: Alma Talent.

Panoply 2022. 3 Ways to Build an ETL Process with Examples. Viitattu 20.2.2022. Saatavilla: https://panoply.io/data-warehouse-guide/3-ways-to-build-an-etl-process/

Rasku J. 2017. Data-analytiikan perusteet. Viitattu 5.12.2021. Saatavilla: https://coss.fi/wp-content/uploads/2017/12/3-Data-analytiikan-perusteet.pdf

Seagate 2018 The Digitalization of the World from edge to the core. Viitattu 4.5.2022 Saatavilla https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

Stonebraker, M., Abadi, D. J., Batkin, A., Chen, X., Cherniack, M. & Ferreira, M. 2005. Proceedings of the 31st international conference on very large data bases. Viitattu 7.12.2021. Saatavilla: https://dl.acm.org/doi/proceedings/10.5555/1083592

Tamminen S. 2019. Datan rooli koneoppimessa ja tekoälyssä Viitattu 4.12.2021. Saatavissa: https://www.slideshare.net/Solita_Oy/datan-rooli-koneoppimisessa-ja-tekolyss

Tikkanen S. 2021. Data ei ole mitään, vasta datasta jalostetulla tiedolla on merkitystä. Viitattu 2.12.2021. Saatavissa: https://www.dimecc.com/data-ei-ole-mitaan/

Vasarhelyi M. A., Kogan A. & Tuttle B. M. 2015. Big Data in Accounting: An Overview. Viitattu 7.12.2021 Saatavissa: https://meridian.allenpress.com/accounting-horizons/article-abstract/29/2/381/99282/Big-Data-in-Accounting-An-Overview

Weston S, Yee D. 2017. Why you should become a user. A brief introduction to R. Viitattu 4.12.2021. https://www.psychologicalscience.org/observer/why-you-should-become-a-user-a-brief-introduction-to-r

Withee K. 2010. Microsoft Business Intelligence for dummies. New Jersey: John Wiley & Sons.

Vänskä R., Härkönen T., Suomalainen K. 2020. Ihmisistä kerätty data uppoaa monimutkaisiin verkostoihin. Viitattu 3.12.2021. Saatavissa: https://www.sitra.fi/artikkelit/ihmisista-keratty-data-uppoaa-monimutkaisiin-verkostoihin/