

Roope Lappalainen

TUOTANTOLINJAN SAANTOLASKELMIEN TOTEUTTAMINEN

Opinnäytetyö

Liiketalouden ammattikorkeakoulututkinto

Tietojenkäsittelyn koulutus

2024



**Kaakkois-Suomen
ammattikorkeakoulu**

| | |
|-----------------|-----------------------------------------------|
| Tutkintonimike | Tradenomi (amk) |
| Tekijä/Tekijät | Roope Lappalainen |
| Työn nimi | Tuotantolinjan saantolaskelmien toteuttaminen |
| Toimeksiantaja | Vaisala Oyj |
| Vuosi | 2024 |
| Sivut | 39 sivua, liitteitä 0 sivua |
| Työn ohjaaja(t) | Janne Turunen, Marjo Puikkonen, Jani Peltola |

TIIVISTELMÄ

Saanto ja sen laskeminen on tärkeää valmistavan teollisuuden tehtaille. Saantolaskelmat auttavat ymmärtämään tuotantolinjan prosessien tehokkuutta ja mahdollisia ongelmia sekä tuotteen kustannuksia.

Tämän opinnäytetyön tavoitteena oli luoda toimivat saantolaskelmat opinnäytetyön toimeksiantajana toimineen Vaisalan anturitehtaan tarpeisiin. Vaisala valmistaa muun muassa erilaisia mittalaitteita ja -järjestelmiä ympäristömittaukseen sekä erilaisiin teollisuuden mittaustarpeisiin, eikä ennen toimeksianton aloitusta Vaisalan anturitehtaalla ollut tapaa hyödyntää anturitehtaalla käyttöönotetun uuden tuotannonohjausjärjestelmän dataa tuotantolinjan saantolaskelmissa.

Opinnäytetyö oli luonteeltaan kehittämistehtävä ja sen kehittämisingelmana on, kuinka toteutetaan tuotantolinjan saantolaskelmat. Opinnäytetyön menetelminä oli SQL-tietokantakyselyiden toteuttaminen, saantolaskelmien implementointi tuotannonohjausjärjestelmän tarjoamia raportointitoiminnallisuuksia hyödyntäen sekä aihepiiriin liittyvän teoreettisen tiedon käsittely kirjallisten lähteiden kautta.

Opinnäytetyön teoriaosuudessa tutustuttiin työn keskeisiin käsitteisiin, joihin lukeutuivat data, tietokannat, SQL sekä tuotannonohjaus ja saanto. Pääpaino teoriaosuudessa oli tietokannat, joiden osalta keskityttiin työssä käytettäviin relaatiotietokantoihin sekä SQL, jonka osalta tutkittiin opinnäytetyön tavoitteiden täyttämiseen tarvittavia SQL-lausekkeita ja funktioita.

Opinnäytetyön kehitysosuudessa toteutettiin SQL-kyselyt saantolaskelmiin tarvittavan datan hakemiseksi tietokannasta. Opinnäytetyöhön ei kuitenkaan dokumentoitu työn varsinaiseen lopputuotokseen tarvittavia SQL-tietokantakyselyitä tai niiden tuottamia tulosjoukkoja tietojen luottamuksellisuuden takia.

Opinnäytetyön tuloksena syntyi Vaisalan anturitehtaalte toimivat saantolaskelmat anturitehtaan tuotantolinjan kokonaissaannon sekä tuotekohtaisen saannon seuraamiseen uuden käyttöönotetun tuotannonohjausjärjestelmän mahdollistamaa dataa hyödyntäen.

Asiasanat: data, relaatiotietokannat, SQL, tuotannonohjaus

| | |
|------------------|------------------------------------------------------------|
| Degree title | Bachelor of Business Administration |
| Author (authors) | Roope Lappalainen |
| Thesis title | Implementation of yield calculations for a production line |
| Commissioned by | Vaisala Oyj |
| Time | 2024 |
| Pages | 39 pages, 0 pages of appendices |
| Supervisor | Janne Turunen, Marjo Puikkonen, Jani Peltola |

ABSTRACT

Yield and calculating it is important for manufacturing factories. Yield calculations help factories to understand the efficiency and potential problems of the processes in the production line, as well as the cost of the product.

The objective of the thesis was to create functional yield calculations for the needs of Vaisala's sensor factory. Vaisala manufactures, among other things, various measuring devices and systems for environmental measurement and for various industrial measurement needs. Before the start of the thesis, Vaisala's sensor factory had no way of utilizing the data from the recently implemented manufacturing execution system for the yield calculations of the production line.

The nature of the thesis was a development task, and its development problem was how to implement the yield calculations of the production line. The methodology of the thesis was the implementation of the SQL queries and yield calculations using the reporting features provided by the manufacturing execution system, and the processing of theoretical information related to the topic through written sources.

The theoretical part of the thesis explored the key concepts of the thesis, which included data, databases, SQL, as well as production control and yield. The theory section mostly concentrated on databases, where the focus was on relational databases used in the work, and SQL, for which the SQL clauses and functions needed to meet the objectives of the thesis were studied.

In the development part of the thesis, SQL queries were developed to access the data needed for yield calculations from the database. However, the thesis did not document the SQL queries required for the actual final output of the work or the set of results produced by them due to the confidentiality of the data.

The result of the thesis was functional yield calculations for Vaisala's sensor factory to monitor the total and the product-specific yields of the production line of the sensor factory. This was achieved by using data enabled by the recently implemented manufacturing execution system.

Keywords: data, relational databases, SQL, production control

SISÄLLYS

| | | |
|-----|---------------------------------------|----|
| 1 | JOHDANTO..... | 5 |
| 2 | DATA..... | 6 |
| 2.1 | Mitä on data?..... | 6 |
| 2.2 | Datan visualisointi..... | 7 |
| 2.3 | Visualisointi prosessina | 9 |
| 3 | TIETOKANTA..... | 11 |
| 3.1 | Mikä on tietokanta?..... | 11 |
| 3.2 | Relaatiotietokanta | 12 |
| 3.3 | Tietokannan hallintajärjestelmä | 14 |
| 3.4 | SQL | 16 |
| 4 | TUOTANNONOHJAUS | 21 |
| 4.1 | Tuotannonohjausjärjestelmä..... | 21 |
| 4.2 | Saanto | 23 |
| 5 | KEHITYSTYÖ..... | 25 |
| 5.1 | Toimeksiantajan esittely | 25 |
| 5.2 | Työn aloitus | 27 |
| 5.3 | Kokonaissaantolaskelma | 28 |
| 5.4 | Tuotekohtainen saantolaskelma | 32 |
| 6 | PÄÄTÄNTÖ | 35 |
| | LÄHTEET..... | 37 |

1 JOHDANTO

Tämän opinnäytetyön aiheena on tuotantolinjan saantolaskelmien toteuttaminen. Aihe opinnäytetyölle valikoitui työn toimeksiantajana toimivan Vaisala Oyj:n kautta, joka valmistaa mittalaitteita ja -järjestelmiä ympäristömittaukseen sekä erilaisiin teollisuuden mittaustarpeisiin.

Työn tavoitteena on luoda toimivat saantolaskelmat, jotka vastaavat Vaisalan anturitehtaan tarpeisiin. Työlle löytyy oikea tarve, sillä Vaisalan anturitehtaalla ei ole tällä hetkellä tapaa hyödyntää uuden käyttöönotetun tuotannonohjausjärjestelmän dataa tuotantolinjan saantolaskelmissa. Saantolaskelmat ovat tärkeitä tehtaalte, sillä ne auttavat ymmärtämään tuotantolinjan prosessien tehokkuutta ja mahdollisia ongelmia sekä tuotteen kustannuksia.

Luonteeltaan opinnäytetyö on kehittämistehtävä. Sen menetelminä ovat SQL-tietokantakyselyiden toteuttaminen saantolaskelmien vaatimaa dataa varten, saantolaskelmien implementointi Vaisalan anturitehtaan uuden käyttöönotetun tuotannonohjausjärjestelmän tarjoamia raportointitoiminnallisuuksia hyödyntäen sekä aihepiiriin liittyvän teoreettisen tiedon käsittely kirjallisten lähteiden kautta.

Opinnäytetyön kehittämisiongelmana on kuinka toteuttaa tuotantolinjan saantolaskelmat. Tutkimuskysymyksinä työlle ovat miten tuotantolinjan saantoa voidaan laskea, mistä saantolaskelmien vaatima data saadaan haettua sekä miten dataa tulee käsitellä, jotta sitä voidaan hyödyntää saantolaskelmissa.

Opinnäytetyön teoriaosuudessa tutustutaan työn keskeisiin käsitteisiin, joihin lukeutuu data, tietokannat, SQL sekä tuotannonohjaus ja saanto. Lähteinä teoriatte olen käyttänyt muun muassa elektronisia lähteitä, kuten virallisia dokumentaatioita sekä olemassa olevaa kirjallisuutta.

Opinnäytetyön kehitysosuudessa työn lopputuloksen varsinaiset SQL-tietokantakyselyt sekä tietokannan tietomalli on jätetty tarkoituksellisesti työn

tausta-aineistoon niiden luottamuksellisuuden myötä. Näin ollen dokumentoidut SQL-tietokantakyselyt sekä niiden palauttamien tulosjoukkojen data eivät ole sellaisenaan valideja, eivätkä täten myöskään ole osa työn varsinaista lopputuotosta.

2 DATA

2.1 Mitä on data?

Data ja informaatio ovat termejä, joihin törmää nykypäivänä jatkuvasti. Joskus näitä termejä tulkitaan viittaavan samaan asiaan, mikä on kuitenkin väärin (Computer Hope 2020). Yleisesti ottaen data on kokoelma yksittäisiä faktoja tai tilastoja (Galbraith 2023). Dataa voi olla monenlaista ja se voi olla esimerkiksi tekstin, numeroiden, totuusarvojen, havaintojen, kuvien tai symbolien muodossa (Computer Hope 2023; Galbraith 2023).

Galbraithin (2023) mukaan data voidaan jakaa kahteen päätyyppiin, määrälliseen ja laadulliseen dataan. Määrällinen data esitetään numeerisessa muodossa, kuten tuotteen hinta, ja laadullinen data on kuvailevaa, mutta ei numeerista, kuten tuotteen nimi.

Data on siis käsittelemätöntä, eli raakaa tiedon muotoa ja sellaisenaan sillä ei ole mitään merkitystä. Se voi olla hyvin yksinkertaista ja vaikuttaa jopa hyödyttömältä, ennen kuin sitä analysoidaan, järjestellään ja tulkitaan, jolloin siitä saadaan informaatiota. Informaatio on siis datan analysoinnin ja tulkitsemisen tulos (kuva 1). (Galbraith 2023.)

| Data | Informaatio |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • Kokoelma yksittäisiä faktoja tai tilastoja • Järjestämätöntä • Ilman kontekstia • Voi olla määrällistä (numeerinen) tai laadullista (kuvaileva) • Esimerkiksi verkkosivuston kävijöiden määrä, yksittäiset asiakaskyselyn vastaukset tai tuotteen hinta | <ul style="list-style-type: none"> • Datan analysoinnin ja tulkinnan tulos • On konteksti • Voidaan käyttää apuna päätöksenteossa • Esimerkiksi verkkosivuston liikenteen muutokset, asiakkaiden mielipide kyselytulosten perusteella tai tuotteiden hintavertailu |

Kuva 1. Data ja informaatio (mukaillen Galbraith 2023)

Nykypäivän digitalisoituneen maailman myötä dataa syntyy valtavia määriä muun muassa erilaisissa mittauspisteissä, tuotantolaitteissa sekä ohjaus- ja tietojärjestelmissä (Pakkasela & Porri-Henttinen 2021). Suurten datamäärien myötä datan tulkitseminen voi olla monimutkaista (Pengon Oy 2016).

Dataa, joka sisältää pelkkiä prosentteja tai lukuja, on yleensä hankalaa vertailla, eikä se pääsääntöisesti ole sen lukijalle kiinnostavaa. Jotta datasta olisi sen lukijalle hyötyä, pitää siitä pystyä nostamaan esille tärkeimmät luvut ja trendit esimerkiksi yrityksen liiketoiminnan tueksi (Pakkasela & Porri-Henttinen 2021; Pengon Oy 2016).

2.2 Datan visualisointi

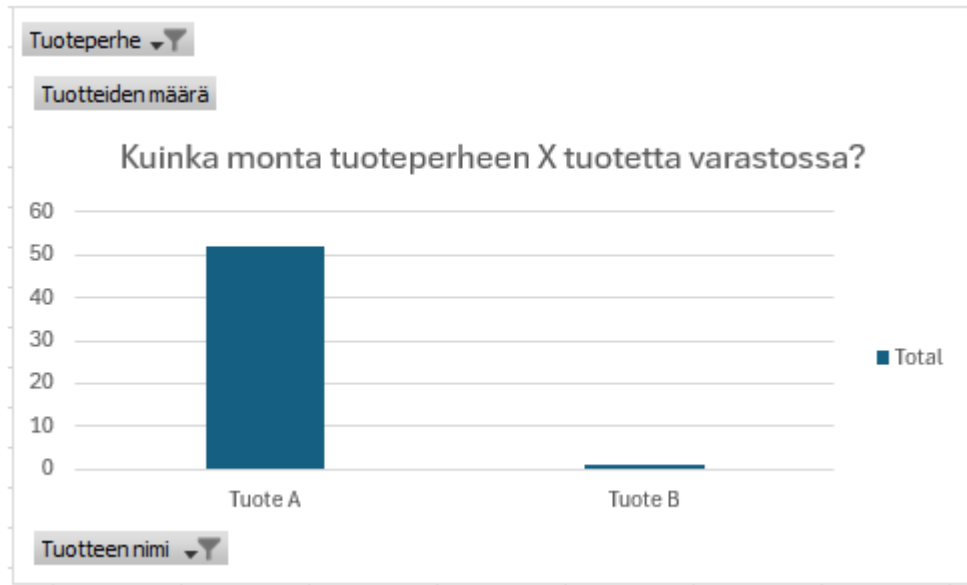
Datan vertailu helpottuu, kun prosenttiosuudet ja lukumäärät muutetaan kuviksi ja kuvioiksi, eli se visualisoidaan (kuva 2; Pengon Oy 2016). Jalosen ja Kososen (2020, 32) mukaan datan visualisoinnin pohjimmainen tarkoitus on kuitenkin tuottaa ymmärrystä, ei kuvia.



Kuva 2. Esimerkki visualisoidusta datasta

Näin ollen dataa esittäessä on hyvä muistaa, että kaikkea dataa, jota voidaan visualisoida, ei välttämättä kannata visualisoida. Jos saman asian voi ilmaista sanallisesti yhtä selkeästi tai selkeämmin, on visualisointi todennäköisesti turhaa, tai jopa haitallista. (Jalonen & Kosonen 2020, 33.)

Visualisoitu data voi olla myös interaktiivista, jolloin dataa lukeva henkilö voi esimerkiksi klikkaamalla tutkia pylväsdiaagrammin yhden pylvään sisältämiä tarkempia tietoja (kuva 3). Näin esityksestä sekä sen sisältämästä datasta voidaan saada sen lukijalle kiinnostavampaa sekä helpommin ymmärrettävää. (Pengon Oy 2016.)

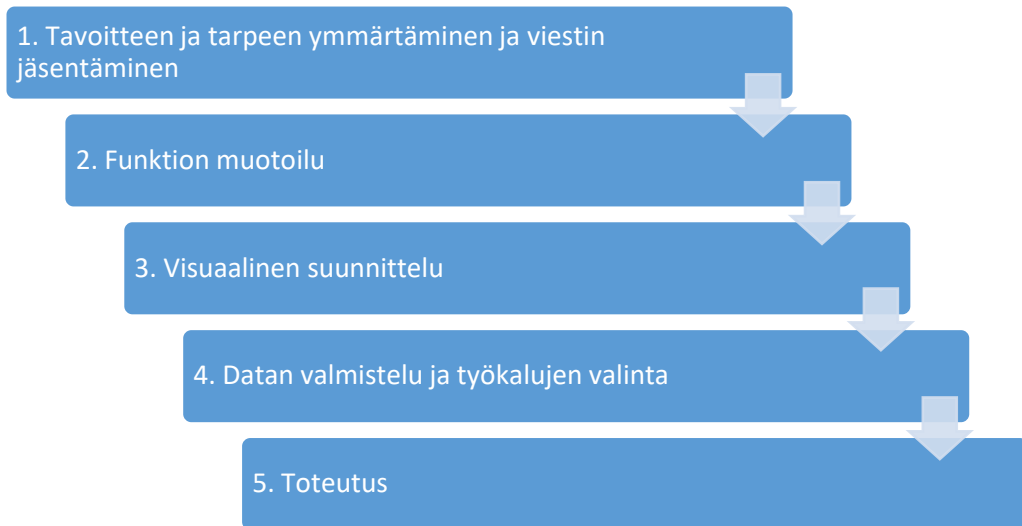


Kuva 3. Esimerkki visualisoinnin interaktiivisuudesta

Hannus ja Lampi (s.a.) toteavatkin, että visualisointi on samaan aikaan datan jäsentämistä ja analysointia kuin myös viestintää ja esittämistä. Visualisoinnin avulla data muuttuu muotonsa tiedoksi, joka mahdollistaa sen havainnollistamisen ja valitun esitystavan kautta jalostumisen ihmisen toiminnan kautta ymmärrykseksi ja tietämykseksi.

2.3 Visualisointi prosessina

Yksi suurimpia kompastuskiviä datan visualisoinnissa on se, että ihmiset huolehtivat liikaa, miltä lopputulos tulee näyttämään (Jotform 2023). Visualisoinnin toteutuksessa voidaankin esimerkiksi hyödyntää Hannuksen ja Lammen (s.a.) esittämää kuvassa 4 näkyvää visualisoinnin prosessimallia.



Kuva 4. Visualisoinnin prosessimalli (mukaillen Hannus & Lampi s.a.)

Ensimmäisenä ja tärkeimpänä vaiheena on tunnistaa visualisoinnin tavoitteet, eli mihin tarpeeseen se vastaa, kenelle, miksi, ja mitä viestiä halutaan välittää, sillä kaikki myöhemmät visualisointia koskevat päätökset ja toimenpiteet tulevat perustumaan näihin määrityksiin. Tavoitteiden ja välitettävän viestin määrittämyksen myötä muodostuu hahmotus siitä, mitä dataa tarvitaan. (Hannus & Lampi s.a.)

Visualisoinnin tehtävä ja tarkoitus, eli sen funktio, muodostuu määritetyn tavoitteen perusteella. Funktiolla määritetään muun muassa mitä faktoja visualisoinnin pitää esittää, onko sen sisältämä data vertailukelpoista, miten data luokitellaan tai ryhmitellään sekä miten visualisointia on tarkoitus lukea. Funktio siis määrittää millaisella lopputuloksella saavutetaan visualisoinnin tavoite ja miten lopputulos saadaan aikaan. Erilaisia valintoja ja päätöksiä on kuitenkin paljon ja näitä tehdessä kannattaakin pitää mielessä aiemmin määritelty tavoite ja välitettävä viesti. (Hannus & Lampi s.a.)

Funktion määrittämisen jälkeen voidaan visuaalinen suunnittelu aloittaa. Suunnittelussa mietittäviä tasoja ovat muun muassa graafinen esitystapa, joka käsittelee perinteisempää graafista suunnittelua, ymmärrettävyyttä ja ulkoasun miellyttävyyttä, informaatiomuotoilu, joka vastaa tiedon esittämistavan suunnitteluun sekä interaktiosuunnittelu, jonka tärkeys korostuu, jos visualisoinnista halutaan tehdä interaktiivinen. (Hannus & Lampi s.a.)

Kun tavoite, välitettävä viesti sekä niiden muodostama funktio tiedetään, on datan valmistelu ja erilaisten teknologiaratkaisujen valintojen tekeminen helpompaa. Tässä on kuitenkin tärkeä huomioitava se, että valintojen tulisi aina tähdätä funktion ja tavoitteiden täyttämiseen. (Hannus & Lampi s.a.) Dataa on myös usein saatavilla suuria määriä, joten on tärkeää tunnistaa, mikä on merkityksellistä (Jotform 2023).

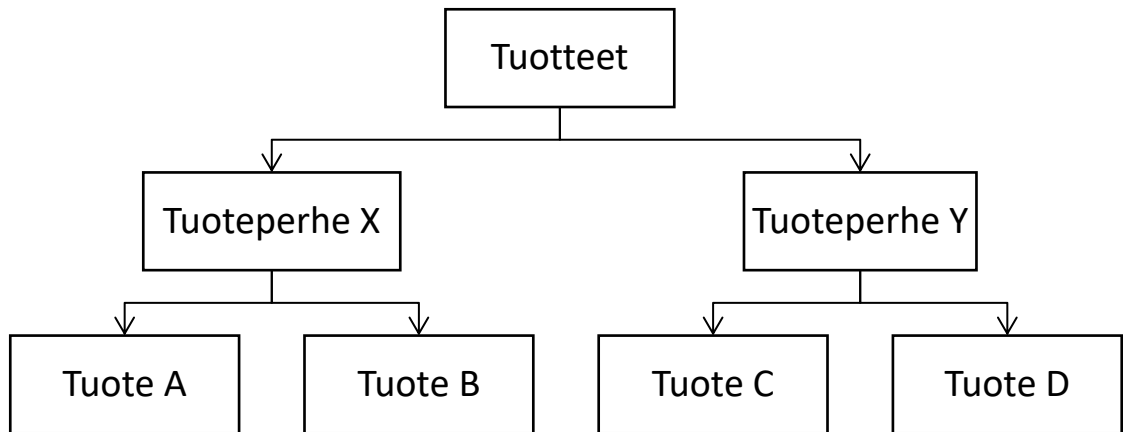
Lopulta itse visualisoinnin tekninen toteutus voidaan aloittaa. Visualisointia kannattaa myös testata koeyleisöllä, jonka rooliin Jotform (2023) suosittelee sellaisia henkilöitä, jotka eivät paljoa aiheesta tiedä. Tällä tavoin voidaan saada tärkeää palautetta, jonka perusteella on mahdollista tehdä visualisoinnin uudelleen suunnittelua, eli prosessia voidaan siis käydä läpi myös iteratiivisesti (Hannus & Lampi s.a.). Eräs yleisimmin esiintyvä ongelma Jotformin (2023) mukaan on väärän tyyppisen kaavion käyttäminen esitettävälle tiedolle.

3 TIETOKANTA

3.1 Mikä on tietokanta?

Tietokanta (eng. database) on kokoelma, joka sisältää toisiinsa liittyvää tietoa tai dataa. Yksinkertaisena esimerkkinä tietokannasta voidaan pitää vaikkapa paperille kirjoitettua ostoksia sisältävää ostoslistaa. (Microsoft s.a.) Tietojenkäsittelyn kontekstissa tietokannalla tarkoitetaan kuitenkin tyypillisesti tietokoneelle tallennettua dataa sisältävää kokoelmaa. (Microsoft s.a.; What Is a Database? s.a.)

Ensimmäiset tietokannat kehitettiin jo 1960-luvun alkupuolella. Nämä varhaiset tietokannat perustuivat hierarkkiseen (kuva 5) tai verkkopohjaiseen tietokantamalliin. Hierarkkinen tietokanta perustui puumalliin, joka mahdollisti tiedon liittämisen yhden tietyn käsitteen alle. (What is a Database? s.a.)



Kuva 5. Hierarkkinen tietokanta

Tiedon tallentaminen ja hakeminen hierarkkisesta tietokannasta oli tehokasta, mutta monimutkaisempi tiedon haku tämän tyyppisestä tietokannasta ei ollut mahdollista sen rakenteen myötä, sillä kantaa analysoidiin aina yksisuuntaisesti (Björklund 2019, 7). Oraclen (What Is a Database? s.a.) mukaan tietokannat ja niiden mallit ovat kuitenkin kehittyneet todella paljon niiden alkuun verrattuna.

Nykypäivän tietokannat voidaankin yleisesti jakaa kahteen eri ryhmään, relaatiotietokantoihin, ja ei-relaatiotietokantoihin. Ei-relaatiotietokantoja kutsutaan myös usein NoSQL (Not only SQL) -tietokannoiksi. (Microsoft s.a.) Koska tässä työssä käsitellään relaatiotietokantoja, emme perehdy ei-relaatiotietokantoihin tämän tarkemmin.

3.2 Relaatiotietokanta

Kaikki relaatiotietokannat perustuvat vuonna 1970 IBM:n tutkija E.F. Coddin julkaisemaan relaatiomalliin (Taylor 2013, 11). Se määrittelee relaatiotietokantojen teoreettisen pohjan ja perustuu joukko-oppiin, matematiikkaan ja predikaattilogiikkaan (Hovi 2004, 5).

Relaatiotietokantojen tietojen käsittelyssä käytetään tietokantaan tallennettuja kaksiulotteisia taulukoita, jotka sisältävät sarakkeita (eng. column) ja rivejä (eng. row). Näitä taulukoita kutsutaan usein tauluiksi (eng. table). (What is a Relational Database (RDBMS)? s.a.) Taulun sarakkeille määritellään kuhunkin

sarakkeeseen talletettavan tiedon tietotyyppi, kuten numeerinen tai merkki-
muotoinen sekä pituus (Hovi 2004, 6). Hovin (2004, 6) mukaan sarakkeita kut-
sutaan joskus myös kentiksi (eng. field) ja rivejä tietueiksi (eng. record).

Jokaisella relaatiotietokannan taululla on määritettynä oma yksilöllinen pää-
avain (eng. primary key), jonka avulla tietokannan taulut identifioidaan. Jokai-
nen taulun rivi tulee sisältää siis oma uniikki pääavaimensa, eikä samaa arvoa
sisältävää pääavainta saa olla useammalla taulun rivillä (taulukko 1). (Hovi
2004, 6; What is a Relational Database (RDBMS)? s.a.)

Taulukko 1. Tuoteperhe-taulu

| id (pääavain) | nimi |
|---------------|--------------|
| 1 | Tuoteperhe X |
| 2 | Tuoteperhe Y |
| 3 | Tuoteperhe Z |

Taululla voi myös olla määritettynä viiteavain (eng. foreign key), joka sisältää
jonkin toisen taulun avaimen (taulukko 2). Viiteavaimen sisältävää taulua kut-
sutaan lapsitauluksi ja taulua, johon viitataan, kutsutaan isätauluksi. Tätä yh-
teyttä kutsutaan isä-lapsi-yhteydeksi tai yksi-moneen yhteydeksi. (Hovi 2004,
6–7.)

Taulukko 2. Tuote-taulu

| id (pääavain) | nimi | tuoteperheld (viiteavain) |
|---------------|---------|------------------------------|
| 1 | Tuote C | 2 |
| 2 | Tuote A | 1 |
| 3 | Tuote B | 1 |
| 4 | Tuote D | 2 |

Relaatiotietokantojen eheys (eng. integrity) varmistetaan eheyssäännöillä.
Eheyssäännöillä voidaan esimerkiksi estää duplikaattirivien muodostuminen
tauluun tai estää tyhjien arvojen tallentamisen taulun sarakkeisiin. (What is a
Relational Database (RDBMS)? s.a.)

3.3 Tietokannan hallintajärjestelmä

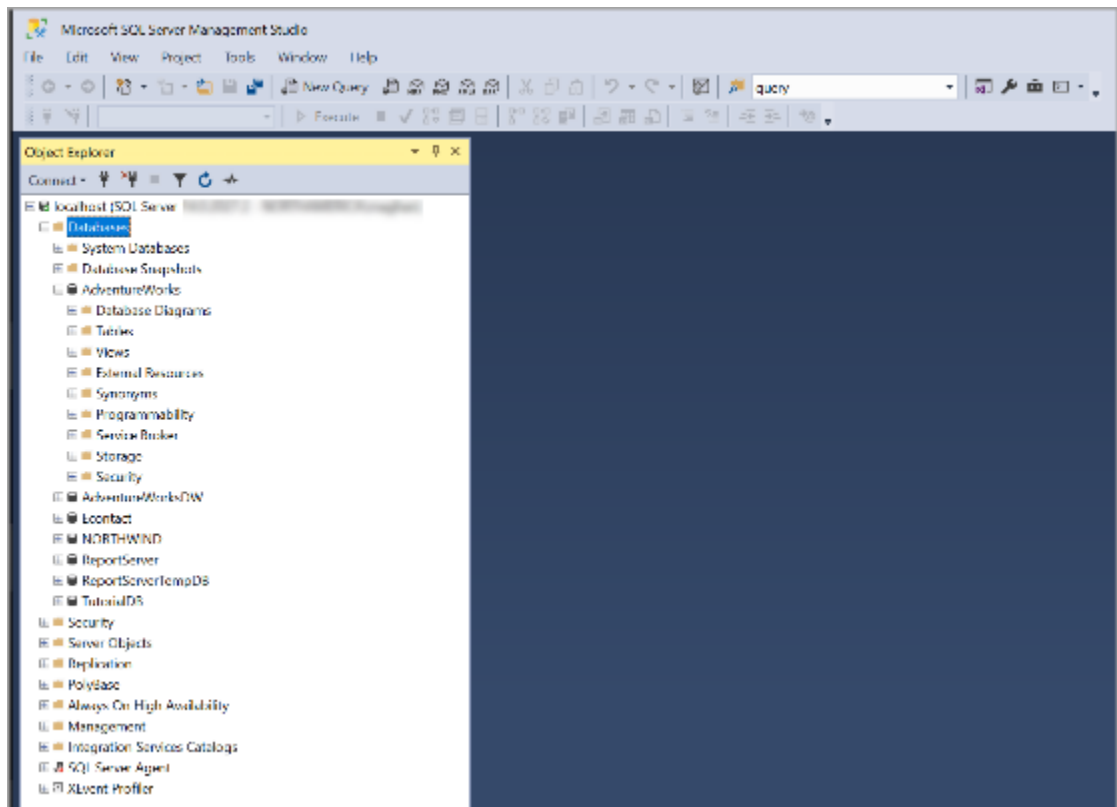
Tietokantoja hallitaan yleensä tietokannan hallintajärjestelmillä (eng. database management system, lyh. DBMS) (What Is a Database? s.a.). Tietokannan hallintajärjestelmät kehitettiin, kun tietokantojen sisältämät datamäärät kasvoivat ja alkoivat olla monimutkaisempia hankaloittaen datan järjestyksessä, saatavilla sekä turvassa pitämistä (Microsoft s.a.).

Tietokannan hallintajärjestelmällä voidaan luoda tietokannan määrytykset ja sisältö. Sen avulla voidaan myös lisätä uutta dataa tietokantaan sekä päivittää tai poistaa tietokantaan jo tallennettua dataa. (Björklund 2019, 7.)

Eri tietokannan hallintajärjestelmä vaihtoehtoja löytyy useita, joista suosituimpia Oraclen (What Is a Database? s.a.) mukaan ovat esimerkiksi MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database sekä dBase.

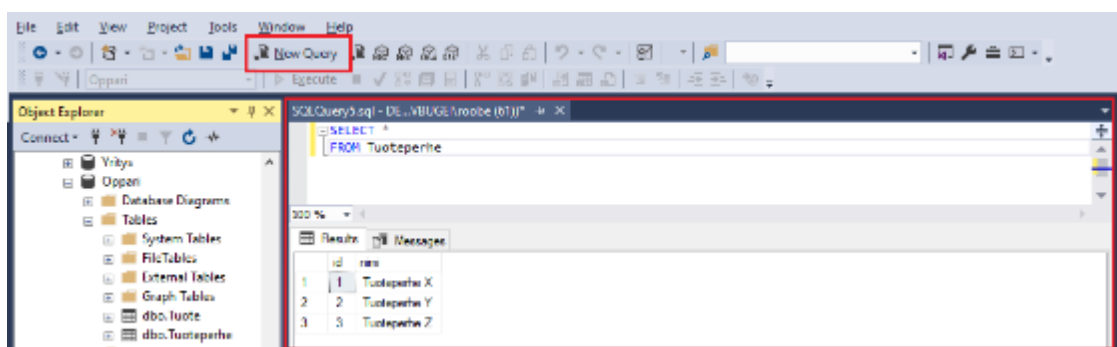
Edellä mainituista esimerkeistä työssä käytettävä Microsoft SQL Server on Microsoftin kehittämä relaatiotietokantojen hallintajärjestelmä (eng. relational database management system, lyh. RDBMS). Se käyttää omaa SQL-kielen toteutusta Transact-SQL:ää, joka tarjoaa lisäominaisuuksia muun muassa muuttujien määrittelyyn ja poikkeusten käsittelyyn. Transact-SQL:ää kutsutaan myös usein T-SQL:äksi. (Microsoft 2024; Peterson 2023.)

Microsoft SQL Serverin pääsääntöisenä käyttöliittymätyökaluna toimii Microsoftin SQL Server Management Studio (kuva 6; Peterson 2023). Se mahdollistaa muun muassa tietokannan käyttäjien, käyttöoikeuksien, tietomallien ja taulujen hallinnan sekä tarjoaa erilaisia graafisia työkaluja. (Microsoft 2023b.)



Kuva 6. Microsoft SQL Server Management Studio (Microsoft 2023a)

Microsoft SQL Server Management Studio mahdollistaa myös SQL-kyselyiden ajamisen suoraan käyttöliittymän sisällä New Query-toimintoa käyttäen (kuva 7). Toiminto avaa uuden ikkunan, johon kyselyt syötetään. Avautuneessa ikkunassa näytetään myös kyselyn tulokset. (Microsoft 2023a.)



Kuva 7. New Query-toiminto (mukaan Microsoft 2023a)

SQL Server onkin vakiinnuttanut jo pitkään asemaansa tietokantaohjelmistojen markkinoilla, sillä sen historia ulottuu jopa 1980-luvulle asti. Alun perin se suunniteltiin pienempiä järjestelmiä varten, mutta nykyään sen suosio myös keskisuurten järjestelmien ohjelmistona on kasvanut. Eräs vaikuttava tekijä

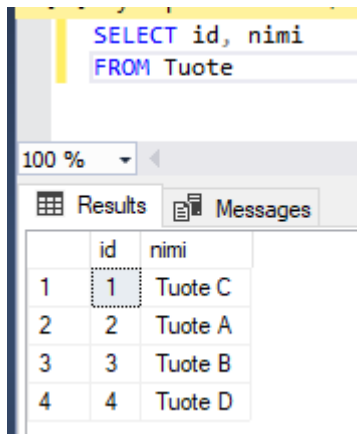
sen tunnetuksi tulemisessa on ollut ohjelmiston helppokäyttöisyys. (Björklund 2019, 11.)

3.4 SQL

SQL eli Structured Query Language on IBM:n kehittämä relaatiotietokannan tietojen tallentamiseen, muokkaamiseen ja hakemiseen tarkoitettu kieli. Se koostuu ennalta määritetyistä englanninkielisistä sanoista, eli SQL-lausekkeista, joilla haluttu toiminnallisuus tiedon käsittelemiseksi toteutetaan. (Taylor 2013, 24.)

SQL-kielen rakenne voidaan jakaa kolmeen eri osaan, joista tässä työssä käsitellään vain DML, eli Data Manipulation Language osaa, joka sisältää niimensä mukaisesti tiedon manipulointiin tarkoitettuja lausekkeitä (Hovi 2004, 14; Taylor 2013, 298). Muut SQL:n osat ovat DDL, eli Data Definition Language, ja DCL, eli Data Control Language. Näistä DDL sisältää tietokannan luomista ja määrittämiä koskevat SQL-lausekkeet, ja DCL tietokannan turvallisuuden liittyvät lausekkeet. (Taylor 2013, 298.)

Tietokannalle tehtäviä SQL-lausekkeista muodostuvia kommunikointi yrityksiä, eli käskyjä kutsutaan SQL-kyselyiksi (eng. query), joihin tietokanta esimerkiksi tietoa hakiessa vastaa palauttamalla kyselyn ehdot täyttävät tiedot, jos sellaista on olemassa. Yleisimpiä ja eniten käytettyjä käskyjä ovat SELECT, jolla haetaan tietoa (kuva 8), INSERT, jolla lisätään tietoa, UPDATE, jolla päivitetään tietoa, ja DELETE, jolla poistetaan tietoa. (Hovi 2004, 14; Taylor 2013, 298.) Tämän työn tavoitteisiin pääsemiseen tulemme tarvitsemaan edellä mainituista käskyistä vain SELECT-käskyä.



The screenshot shows a SQL query editor with the following text:

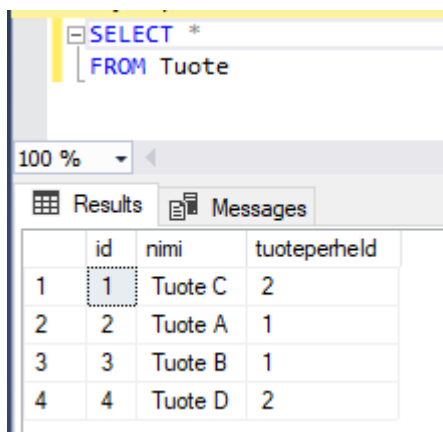
```
SELECT id, nimi
FROM Tuote
```

Below the query, there is a 'Results' tab showing a table with the following data:

| | id | nimi |
|---|----|---------|
| 1 | 1 | Tuote C |
| 2 | 2 | Tuote A |
| 3 | 3 | Tuote B |
| 4 | 4 | Tuote D |

Kuva 8. SELECT-käsky

SELECT on yksi SQL:n keskeisimpiä käskyjä, jonka pakollisia osia ovat SELECT ja FROM. SELECT määrittää mitkä sarakkeet taulusta haetaan, ja FROM mistä taulusta ne haetaan. SELECT-lausekkeen perään luetellaan pilkuilla erotettuna ne sarakkeet, jotka haetaan. Sarakkeiden tulee kuitenkin olla olemassa taulussa, joka löytyy kyselyn FROM-määreestä. Jos taulusta halutaan hakea kaikki sen sisältämät sarakkeet, voidaan lueteltujen sarakkeiden sijaan antaa asteriski, eli *-merkki (kuva 9). (Hovi 2004, 27.)



The screenshot shows a SQL query editor with the following text:

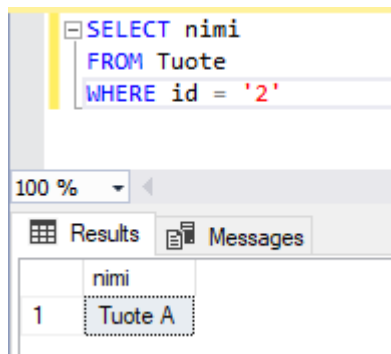
```
SELECT *
FROM Tuote
```

Below the query, there is a 'Results' tab showing a table with the following data:

| | id | nimi | tuoteperheld |
|---|----|---------|--------------|
| 1 | 1 | Tuote C | 2 |
| 2 | 2 | Tuote A | 1 |
| 3 | 3 | Tuote B | 1 |
| 4 | 4 | Tuote D | 2 |

Kuva 9. Taulun kaikkien sarakkeiden haku

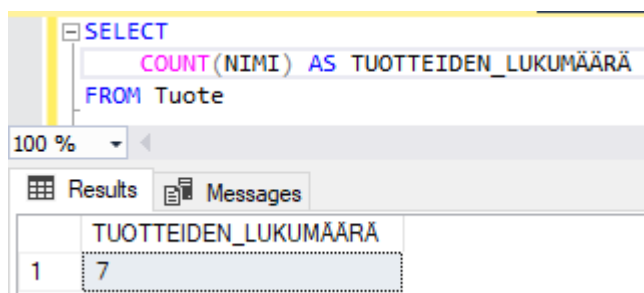
Muita valinnaisia osia käskyyn ovat WHERE, jolla määritetään, mitkä rivit haetaan (kuva 10), GROUP BY, jolla määritetään, miten haettava tieto ryhmitellään, HAVING, jolla määritetään, mitkä ryhmittelyn tuloksena syntyneistä riveistä haetaan sekä ORDER BY, jolla määritellään, miten haettava tieto lajitellaan. (Hovi 2004, 27.)



Kuva 10. Esimerkki WHERE-ehdolauseen käytöstä

SQL sisältää myös muutamia erilaisia koostefunktioita, joiden yhteydessä käytetään yleensä GROUP BY -määrettä (Microsoft 2023d). Näiden avulla voidaan laskea sarakkeiden arvoista esimerkiksi summia, keskiarvoja, minimiä ja maksimia. Tässä työssä käytettäviä koostefunktioita ovat COUNT, jolla lasetaan ehdot täyttävien sarakkeiden arvojen lukumäärä sekä SUM, jolla lasetaan numeeristen sarakkeiden arvojen summa. (Hovi 2004, 44–46; Microsoft 2023d.)

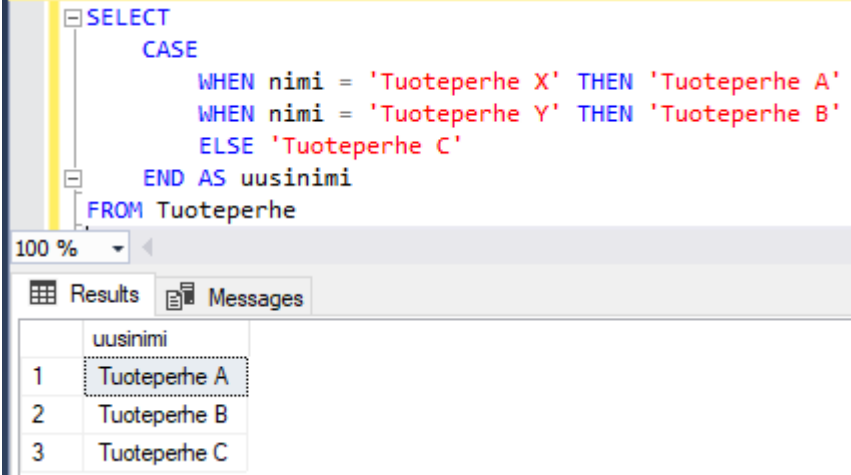
Koostefunktioiden palauttama arvo ei sisällä sarakkeen otsikkoa eli nimeä, joten sarakkeelle täytyy antaa uusi nimi, eli aliasnimi. Aliasnimi voidaan määrittää kirjoittamalla se välilyönnillä erotettuna sarakenimen perään. Suositeltavampi tapa on kuitenkin Hovin (2004, 29–30) mukaan käyttää AS-syntaksia (kuva 11) mahdollisten erehdyksien välttämiseksi.



Kuva 11. Esimerkki sarakkeen aliasnimestä

Muita työssä käytettäviä SQL funktioita ovat konkatenointi funktio CONCAT, jolla saadaan yhdistettyä eri merkkijonoja yhdeksi merkkijonoksi, NULL-funktio COALESCE, joka palauttaa sille parametreinä syötetyistä arvoista ensimmäisen arvon, joka ei ole NULL sekä CASE-lause (kuva 12), jonka avulla voidaan

tarkastella eri ehtoja SQL-lauseissa (Hovi 2004, 50–56; Microsoft 2023e; Microsoft 2023f; Microsoft 2023g). CASE-lause alkaa CASE-määreellä, jonka jälkeen määritetään ”jos X, niin Y” tyyppisiä ehtoja WHEN- ja THEN-määreiden avulla. CASE-lauseessa voidaan myös määrittää ELSE-määreellä sarakkeen arvo niissä tilanteissa, joissa mikään ehto ei täyty. CASE-lause lopetetaan END-määreeseen, jonka jälkeen voidaan lauseen palauttamalle arvolle määrittää sarakkeen aliasnimi. (Hovi 2004, 54–55; Microsoft 2023e.)



```
SELECT
  CASE
    WHEN nimi = 'Tuoteperhe X' THEN 'Tuoteperhe A'
    WHEN nimi = 'Tuoteperhe Y' THEN 'Tuoteperhe B'
    ELSE 'Tuoteperhe C'
  END AS uusinimi
FROM Tuoteperhe
```

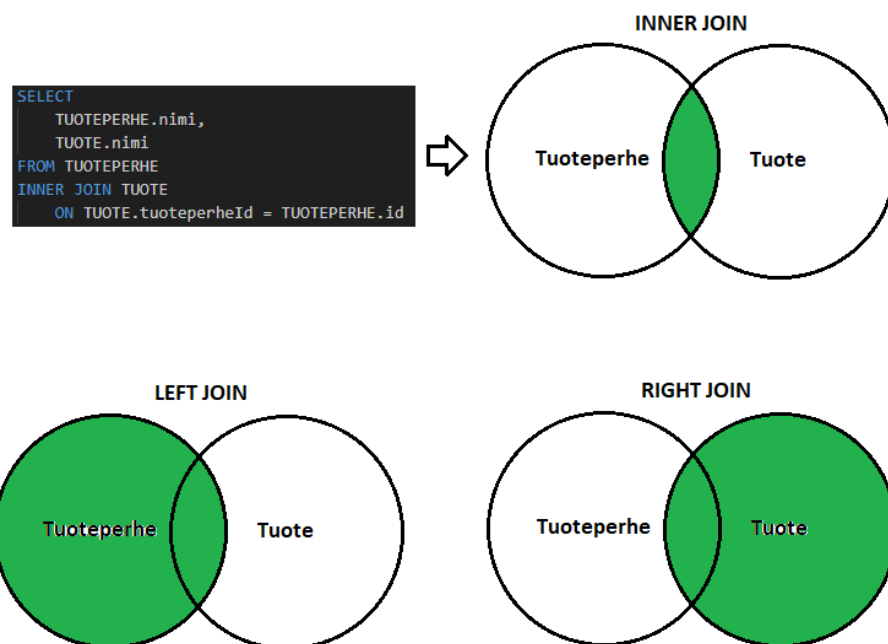
100 %

Results Messages

| | uusinimi |
|---|--------------|
| 1 | Tuoteperhe A |
| 2 | Tuoteperhe B |
| 3 | Tuoteperhe C |

Kuva 12. Esimerkki CASE-lauseesta

Koska relaatiotietokannan sisältämät tiedot ovat useissa eri tauluissa, joiden välillä on yhteyksiä, käytetään usein SELECT-lausekkeita sisältävien kyselyiden yhteydessä erilaisia JOIN-liitoksia (kuva 13), jotta eri taulujen sarakkeiden tietoja saadaan yhdistettyä yhteen tulosjoukkoon. Taulujen sarakkeen arvo, jonka perusteella taulut liitetään, määritellään ON-ehdon yhteydessä, joka toimii samaan tapaan kuin WHERE-ehto. (Microsoft 2023c; Taylor 2013, 249–266.)



Kuva 13. Esimerkkejä taulujen JOIN-liitoksista (mukaillen Dybka 2016)

Kun taulujen välinen liitos tehdään INNER JOIN -liitoksella, palauttaa tietokanta tuloksina vain ne rivit tauluista, joista molemmista löytyy ON-ehdossa määritellyt keskenään vastaavat arvot. LEFT JOIN -liitos palauttaa tuloksena kaikki rivit vasemmanpuoleisesta taulusta, eli taulusta johon toinen taulu liitetään sekä ne rivit oikeanpuoleisesta taulusta, joka täyttää ON-ehdossa määritetyn vastaavuuden. RIGHT JOIN -liitos toimii samalla periaatteella kuin LEFT JOIN, mutta käänteisesti, eli se palauttaa oikeanpuoleisesta taulusta kaikki rivit, mutta vasemmanpuoleisesta taulusta vain osan. (Microsoft 2023c; Taylor 2013, 255–259.)

Taulut voidaan liittää myös tarvittaessa toisiinsa ilman liitosehtoa CROSS JOIN -liitoksella. CROSS JOIN -liitoksella taulun kaikki rivit liitetään toisen taulun kaikkiin riveihin, jolloin liitoksena saadaan tulosjoukkojen karteesinen tulo (eng. cartesian product) (Hovi 2004, 78).

Tietokannan tauluja voidaan myös yhdistää UNION-määreellä, jos tulosjoukkojen sarakemäärät sekä sarakkeiden tietotyypit täsmäävät. UNION-määre kirjoitetaan SELECT-käskyjen väliin, jolloin lopullinen tulosjoukko sisältää tu-

losjoukkojen rivit allekkain. Jos yhdistettyä tulosjoukkoa halutaan lajitella ORDER BY-lauseella, asetetaan ORDER BY viimeisenä olevaan SELECT-käskyyn. (Hovi 2004, 86–87.)

4 TUOTANNONOHJAUS

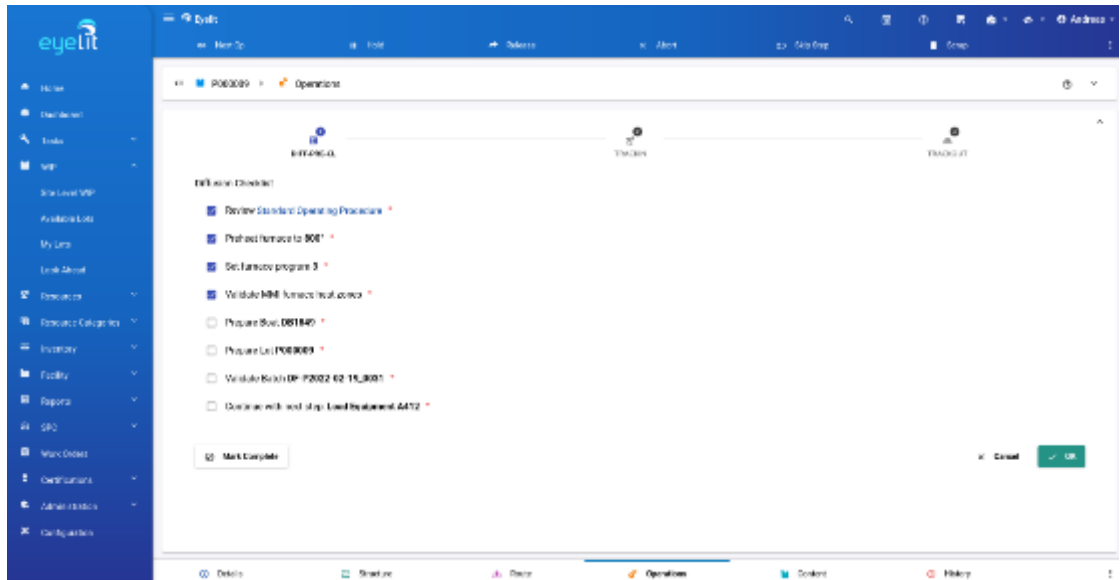
4.1 Tuotannonohjausjärjestelmä

Tuotannonohjaus on yrityksen jokapäiväisen tuotantotoiminnan ohjaamista. Se sisältää tuotannon osatoimien ja tehtävien toteutukseen liittyvää suunnittelua, päätöksentekoa, tiedottamista, ja valvontaa. Kun tuotannonohjaus on tehokasta, voi se tarjota yritykselle merkittävän kilpailuedun. (Nikupeteri 2016, 18–19.)

Tuotannonohjausjärjestelmä tai MES-järjestelmä (eng. Manufacturing Execution System) on reaaliaikainen tietojärjestelmä, joka on tarkoitettu tuotannon ohjaamiseen, johtamiseen ja kehittämiseen. Eniten järjestelmän käytöstä hyötyvät ne yritykset, joissa tehdään sarjatuotantoa riippumatta siitä, onko tuotanto automaattista vai puoliautomaattista. Tuotannonohjausjärjestelmän avulla saadaan kerättyä dataa jokaisesta tuotantovaiheesta tietokantaan. Tätä dataa voidaan hyödyntää tuotantoon liittyvien päätöksien teossa sekä sen jatkuvassa parantamisessa. (Fikuro 2023.)

Järjestelmän automaattisesta datan keruusta huolimatta on tuotannossa työskentelevillä henkilöillä kuitenkin tärkeä rooli tiedon rikastuttamisessa sekä raportoisissa. Kun mahdolliset poikkeamat raportoidaan, parantaa se myös tuotteiden jäljitettävyyttä. (Arkkola 2023.)

Vaisalan anturitehtaan käytössä oleva tuotannonohjausjärjestelmä Eyelit MES (kuva 14) on seuraavan sukupolven alustasta riippumaton järjestelmä, joka tarjoaa laajan valikoiman eri toimintoja yrityksen liiketoiminnan- ja tuotantoprosessien koordinointiin, ohjaukseen ja optimointiin.



Kuva 14. Kuvankaappaus Eyelit MES-järjestelmän käyttöliittymästä (Eyelit s.a.)

Näihin toimintoihin lukeutuvat esimerkiksi järjestelmän sisältämä varastonhallintamoduuli, joka mahdollistaa kulutusmateriaalien ja varaosien seurannan jäljitettävyyden parantamiseksi sekä raportointitoiminnallisuudet, joiden avulla käyttäjät voivat luoda ja tarkastella erilaisia raportteja helposti (kuva 15). (Eyelit s.a.)



Kuva 15. Kuvankaappaus Eyelit MES-järjestelmän raportista (Eyelit s.a.)

Näiden lisäksi Eyelit MES -järjestelmän tarjoamia muita ominaisuuksia ja hyötyjä ovat muun muassa tiedonhallinta ja jakelu, jotta oikeat tiedot saadaan toimitettua niitä tarvitseville henkilöille oikeaan aikaan tuotantoympäristön muut-

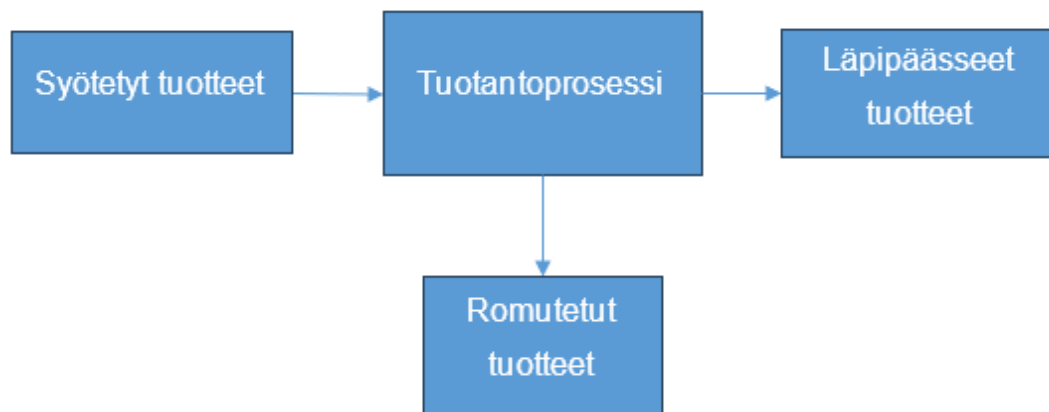
tuviin olosuhteisiin reagoimista varten, yksilölliset näkymät, jotta eri loppukäyttäjryhmät voivat luoda henkilökohtaisia näkymiä sisältäen vain heille olennaiset tuotantotoiminnot ja elementit sekä laadunvalvonta ja -hallinta ominaisuuksia, joiden avulla lopputuotteen laatua saadaan parannettua. (Eyelit s.a.)

4.2 Saanto

Toimintaa ei pystytä kehittämään, jos nykyistä suoritustasoa ei ymmärretä, eikä tavoitteita kehitykselle voida asettaa ilman mittareita. Mittaamisen tavoitteena on selkeiden tavoitteiden asettaminen, jotta mahdolliset poikkeamat ja ongelmat havaittaisiin nopeasti, samalla mahdollistaen kehitystoimenpiteiden vaikutusten seurannan. (Nikupeteri 2016, 50.)

Perinteiset prosessimittarit ovat yleensä olleet rahaan perustuvia, mutta tutkijat ovat havainneet, että tällaisilla mittareilla on paljon rajoitteita, joiden myötä ne ovat huonosti soveltuvia nykypäivänä yleistyneisiin joustaviin toimintamalleihin. Mittareiden tulisi sen sijaan tarjota reaaliaikaista tietoa sekä olla joustavia ja relevantteja jokaisella johtotasolla. (Nikupeteri 2016, 50–51.) Mittari, jonka eri variaatioita tässä työssä käsitellään, on saanto (eng. yield).

Ensisaannolla tarkoitetaan tuotteiden läpäisemistä tuotantoprosessissa, eli kuinka suuri osa tuotantoprosessiin syötetyistä tuotteista pääsee prosessin loppuun onnistuneesti (kuva 16) (Mäkelä 2015, 24; Precognize s.a.). Mäkelän (2015, 24) mukaan ensisaannolle tyypillisiä tarkastelupisteitä ovat tuotteiden laadun tarkastaminen tai tuotteiden testaaminen.



Kuva 16. Esimerkki saannosta (mukaiillen Gygi 2016)

Mäkelän (2015, 25) mukaan ensisaannon laskemiseen ja mittaamiseen käytetään muutamia erilaisia tapoja, mutta näiden kaikkien pääasiallinen tarkoitus on kuitenkin sama, eli kuvata ensisaanto prosenttilukuna. Riippuen mitä laskentatapaa käyttää, tulee tuloksista hieman erilaisia. Tyypillisimpiä laskentatapoja ja niistä käytettäviä termejä ovat FPY eli First Pass Yield, jota joskus kutsutaan myös termillä TPY eli Throughput Yield, FTY eli First Time Yield ja RTY eli Rolled Throughput Yield.

Klassinen saanto eli FTY kuvaa tuotantoprosessin yhden vaiheen saantoa, niin että uudelleen prosessoituja eli korjattuja tuotteita ei huomioida, joten vain romutetut tuotteet vähentävät saantoa. Jos siis esimerkiksi prosessi sisältää kolme eri vaihetta, voidaan eri vaiheiden A, B, ja C saanto laskea jakamalla kaavan 1 mukaisesti prosessivaiheesta valmistuneiden tuotteiden kappalemäärä prosessivaiheeseen syötettyjen tuotteiden kappalemäärälle. Esimerkiksi prosessin ensimmäiseen vaiheeseen A syötetään 100 kpl tuotetta, mutta kyseisen vaiheen läpi pääsee onnistuneena 90 kpl. Tällöin prosessin A vaiheen FTY-saanto on $90/100 = 90\%$. Sama laskentasääntö pätee myös muihin prosessin vaiheisiin. Jos siis seuraavaan vaiheeseen B syötetään vaiheesta A läpipäässeet 90 tuotetta ja niistä pääsee onnistuneena vaiheen B läpi, on B vaiheen FTY-saanto tällöin $80/90 = 88,89\%$. Jos taas vaiheesta C läpi pääsee 75 tuotetta onnistuneesti, on C vaiheen FTY-saanto $75/80 = 93,75\%$. Prosessin kokonais-FTY saadaan laskettua jakamalla koko prosessin läpipäässeiden kappalemäärä koko prosessiin syötettyjen määrällä. (Mäkelä 2015, 26; Nikupeteri 2016, 57)

$$FTY = \frac{\textit{läpipäässeet tuotteet}}{\textit{syötetyt tuotteet}} \cdot 100 \% \quad (1)$$

Ensisaanto eli FPY toimii muuten samalla periaatteella kuin FTY, mutta tämän laskemisessa kaavan 2 mukaan otetaan huomioon romutettujen tuotteiden lisäksi myös korjatut tuotteet, eli prosessista läpipäässeistä tuotteista vähennetään korjatut tuotteet, jonka jälkeen saatu luku jaetaan prosessiin syötettyjen tuotteiden kappalemäärällä. FPY siis kertoo, kuinka suuri määrä tuotteita pääsevät prosessin läpi ensimmäisellä kerralla hyväksytysti ilman korjaustarpeita. (Mäkelä 2015, 25.)

$$FPY = \frac{\textit{läpipäässeet tuotteet} - \textit{korjatut tuotteet}}{\textit{syötetyt tuotteet}} \cdot 100 \% \quad (2)$$

Läpivyörytetty saanto eli RTY kuvaa todennäköisyyttä, jolla tuotantoprosessista valmistuu tuotteita ilman yhtäkään virhettä. Se voidaan laskea kaavan 3 mukaisesti kertomalla eri prosessien FPY-saannot keskenään, jonka tuloksena syntyy osuus prosessoiduista tuotteista, jotka ovat läpipäässeet monivaiheisen prosessin ilman yhtäkään virhettä.

$$RTY = FPY_1 \cdot FPY_2 \cdot \dots \cdot FPY_n \cdot 100 \% \quad (3)$$

Nikupeterin (2016, 58) mukaan RTY on mittarina erityisen tärkeä, jos prosessissa on havaittu liikaa tarpeita tuotteiden korjauksille, eli toisin sanoen tehdään siis paljon turhaa työtä. Mäkelän (2015, 27) mielestä jokaisen valmistavan yrityksen tulisi mitata ensisaantoa RTY:n avulla, sillä pelkkä FPY tai FTY ei yksistään kuvaa prosessin laadukkuutta riittävästi.

5 KEHITYSTYÖ

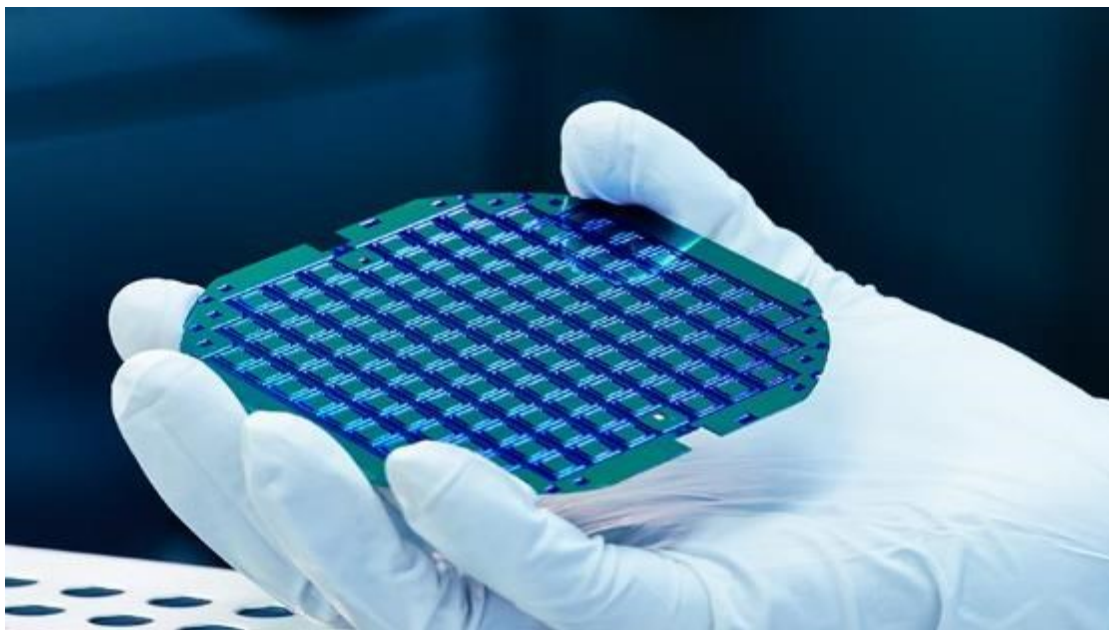
5.1 Toimeksiantajan esittely

Opinnäytetyön toimeksiantajana toimiva Vaisala Oyj on suomalainen yritys, joka suunnittelee, kehittää ja valmistaa laitteita ja järjestelmiä ympäristön ja teollisuuden mittaustarpeisiin. Vaisala tarjoaa laajan valikoiman tuotteita muun

muassa ilmanlaadun seurantaan, meteorologiaan ja muihin sääolosuhteiden mittauksiin, sisältäen esimerkiksi kosteus- ja lämpötila-antureita sekä ilmanpaineantureita. Tuotteet soveltuvat monenlaisiin käyttökohteisiin, kuten lentoliikenteen turvallisuuden parantamiseen, teollisuusprosessien optimointiin ja ilmastonmuutoksen tutkimukseen. (Vaisala Oyj 2024.)

Vaisalan antureiden tutkimus-, kehitys- ja valmistustyö tapahtuu yrityksen omissa puhdastiloissa (eng. cleanroom) Vantaalla, jossa sijaitsee myös yrityksen pääkonttori. Puhdastilalla tarkoitetaan aluetta, jossa eri olosuhteita kuten ilmanlaatua, kosteutta ja lämpötilaa kontrolloidaan. Puhdastilan ilmaa suodattetaan jatkuvasti HEPA-suodattimien avulla, jotta puhdastilan ilmassa ei esiintyisi partikkeleita, jotka voivat mahdollisesti aiheuttaa kontaminaatioita antureiden valmistusprosessien aikana. Puhdastilassa työskentelevät henkilöt käyttävät myös puhdastilaan tarkoitettua erityisvaatetusta, jotta vältetään ihmislähtöisiä kontaminaatioita, kuten esimerkiksi karvojen tai ihosolujen päätymistä puhdastilan ilmaan tai tuotteisiin. (Vaisala 2021.)

Vaisalan anturitehtaan puhdastilassa antureiden valmistus aloitetaan tyhjiltä kiekoilta, joiden materiaalina toimii muun muassa pii-, alumiinioksidi-, tai lasi. Anturin rakenne muodostetaan kiekon pinnalle (kuva 17) käyttämällä erilaisia teknologioita, kuten esimerkiksi ohutkalvopinnoitus, litografia ja etsaus. Valmistuksen aikana anturiekot viettävät aikaa puhdastilassa jopa kolme kuukautta, käyden läpi yli 100 tilastollista prosessinsäätömittausta. Yhdelle kielelle muodostuu lopulta useita anturisiruja, joiden rakenteet ovat mikrometreissä mitattavia, eli todella pieniä. Anturiekon sirut sahataan erillisiksi ja valmistuessaan ne siirtyvät Vaisalan omiin kokoonpanolinjoihin, jossa anturit kootaan. (Vaisala 2021.)



Kuva 17. Vaisalan Barocap-kiekkko (Vaisala BAROCAP® -teknologia s.a.)

Opinnäytetyössä keskitytäänkin nimenomaan Vaisalan anturitehtaaseen ja siellä käyttöönotetun uuden tuotannonohjausjärjestelmän tarjoaman datan hyödyntämiseen anturitehtaan saannon laskemisessa.

5.2 Työn aloitus

Opinnäytetyötä käynnistäessä pidimme palaverin yrityksen ohjaajan sekä Vaisalan anturitehtaan johtajan kanssa, jossa kävimme läpi, mitä dataa haluttaisiin nähdä, miltä aikaväliltä ja miten. Päätimme, että kehitettävä saantolaskelma toteutettaisiin tuotannonohjausjärjestelmän raportointitoiminnallisuuksia hyödyntäen. Päätimme myös, että anturikiekkot ja erilleen sahatut anturisirut eriteltäisiin eri raporteihin, sillä niiden tuotantolinjat sekä prosessointivuot poikkeavat toisistaan paljon.

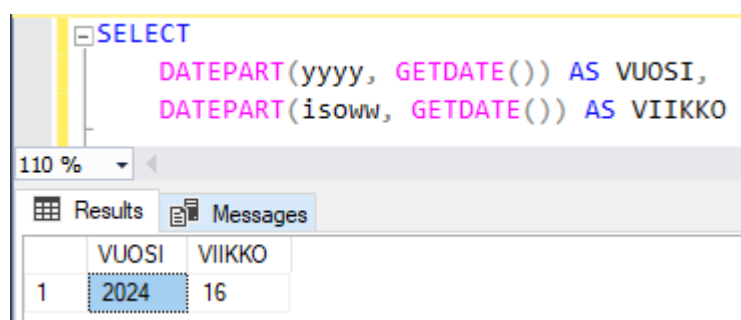
Kokouksen aikana kävi myös muun muassa ilmi se, ettei tällä hetkellä tuotannonohjausjärjestelmän datan perusteella nähdä ollenkaan tietoa siitä, että kuinka monta tuotetta on aloitettu tuotantoon tai valmistunut tuotannosta kullakin viikolla. Myös eri tuotteiden väliset erilaiset tuotantovolyymit, eli missä määrin kyseistä tuotetta valmistetaan sekä läpimenoajat, eli missä ajassa tuote valmistuu tuotantolinjalta, tulisivat vaikuttamaan laskelmiin, joten perinteiset saannon laskemistavat eivät välttämättä tulisi toimimaan sellaisenaan.

Sovimme, että toteutan muutaman erilaisen testilaskelman, joiden katselmoinnin perusteella päättäisimme mihin suuntaan saantolaskelmia haluttaisiin kehittää. Näin ollen, aloin kehittämään SQL-kyselyitä, jotta haluttu data saataisiin haettua tietokannasta.

Laskelmat joihin lopulta päädyimme, olivat anturitehtaan ylemmän tason mittarina toimiva kokonaissaantolaskelma, joka sisältäisi kaikkien aloitettujen, valmistuneiden ja romutettujen tuotantoerien määrät viikoittain sekä tuotekohtaisempi saantolaskelma, joka mahdollistaisi eri tuotteiden saannon tarkemman seuraamisen juurisyiden selvittämiseksi. Työtä päätettiin myös rajata niin, että laskelmissa huomioitaisiin vain ne tuotantoerät, jotka sisältävät anturikiekkaja.

5.3 Kokonaissaantolaskelma

Koska data haluttiin ryhmitellä viikoittain, loin ensin SQL-kyselyn, joka hakisi kaikki viikkonumerot sekä vuodet. Selvisi kuitenkin, että näiden hakemiseen oli jo valmis ratkaisu toteutettuna, mutta kyselyn palauttama tulosjoukko ei sisältänyt nykyistä viikkoa. Nykyisen viikkonumeron ja vuoden hakeminen toteutui SQL Serverin sisäänrakennettuja GETDATE- sekä DATEPART-funktioita käyttäen (kuva 18).



Kuva 18. SQL-kysely nykyisen vuoden ja viikkonumeron hakemiseen

Nykyisen viikkonumeron ja vuoden yhdistäminen muiden viikkonumeroiden ja vuosien sisältämään tulosjoukkoon toteutui UNION-määreellä (kuva 19). Näin saatiin luotua pohja, johon muu data tultaisiin liittämään ja jonka mukaan se jäsenneltäisiin.

```

SELECT
    VUOSI,
    VIIKKO
FROM Vuosi_ja_viikko
UNION
SELECT
    DATEPART(yyyy, GETDATE()) AS VUOSI,
    DATEPART(isoww, GETDATE()) AS VIIKKO

```

| | VUOSI | VIIKKO |
|---|-------|--------|
| 1 | 2024 | 11 |
| 2 | 2024 | 12 |
| 3 | 2024 | 13 |
| 4 | 2024 | 14 |
| 5 | 2024 | 15 |
| 6 | 2024 | 16 |

Kuva 19. Tulosjoukkojen yhdistäminen UNION-määreellä

Seuraavaksi tulisi selvittää, kuinka paljon kunakin viikkona tuotetta on aloitettu tuotantoon. Jokaisesta anturikiekk- tai anturisiruerälle suoritettavasta toiminnosta tallentuu rivi tietokannan tauluun, joka sisältää muun muassa anturikiekk- tai anturisiruerän tunnisteen, sille suoritettun toiminnon, toiminnon aikaleiman, toiminnon aikaisen erän kiekko- tai sirumäärän sekä tiedon siitä, että sisältääkö kyseinen tuotantoerä anturikiekkoja vai anturisiruja.

Kyseisessä taulussa ei kuitenkaan ole valmiina vuotta tai viikkonumeroa, joten näiden saamiseksi täytyi hyödyntää taas DATEPART-funktiota. Loin ensin WITH-määreen avulla tilapäisen taulun, jossa muunsin suoritettun tapahtuman aikaleiman viikkonumeroksi ja vuodeksi. Rajasin myös tilapäisen taulun tuloksia WHERE ehdon avulla, jotta taulu sisältäisi vain ne rivit, joissa suoritettu toiminto on tuotantoon aloitus ja tuotantoerän sisältö on anturikiekkot. Näin tilapäisestä taulusta saatiin aloitettujen anturikiekkojen määrä selville ryhmittelemällä rivit GROUP BY -lausekkeella vuoden ja viikon mukaan sekä laskemalla määrät yhteen SUM-funktiolla (kuva 20).

```

WITH AloitetutTuotantoerät AS
(
    SELECT
        ERÄTUNNISTE,
        MÄÄRÄ,
        DATEPART(yyyy, AIKALEIMA) AS VUOSI,
        DATEPART(isoww, AIKALEIMA) AS VIIKKO
    FROM TUOTANTOERILLE_SUORITETUT_TOIMINNOT
    WHERE KIEKKO_VAI_SIRU = 'KIEKKO'
    AND Suoritettu_Toiminto = 'aloitaTuotantoon'
)

SELECT
    VUOSI,
    VIIKKO,
    SUM(MÄÄRÄ) AS MÄÄRÄ
FROM AloitetutTuotantoerät
GROUP BY VUOSI, VIIKKO

```

| | VUOSI | VIIKKO | MÄÄRÄ |
|---|-------|--------|-------|
| 1 | 2024 | 11 | 90 |
| 2 | 2024 | 12 | 30 |
| 3 | 2024 | 13 | 30 |
| 4 | 2024 | 14 | 120 |
| 5 | 2024 | 16 | 30 |

Kuva 20. Summan laskeminen SUM-koostefunktiolla

Valmistuneiden anturikiekkojen määrän selvittämiseksi muutin WHERE-ehdossa määritellyn suoritettua toiminnon (kuva 21). Näin tulosjoukkoon saatiin sisällytettyä vain valmistuneiksi merkityt tuotantoerät, joiden määrän laskeminen tapahtui samalla periaatteella kuin aloitettujen kiekkojen määrä.

```

WITH ValmistuneetTuotantoerät AS
(
    SELECT
        ERÄTUNNISTE,
        MÄÄRÄ,
        DATEPART(yyyy, AIKALEIMA) AS VUOSI,
        DATEPART(isoww, AIKALEIMA) AS VIIKKO
    FROM TUOTANTOERILLE_SUORITETUT_TOIMINNOT
    WHERE KIEKKO_VAI_SIRU = 'KIEKKO'
    AND SUORITETTU_TOIMINTO = 'merkitseValmiiksi'
)

SELECT
    VUOSI,
    VIIKKO,
    SUM(MÄÄRÄ) AS MÄÄRÄ
FROM ValmistuneetTuotantoerät
GROUP BY VUOSI, VIIKKO

```

| | VUOSI | VIIKKO | MÄÄRÄ |
|---|-------|--------|-------|
| 1 | 2024 | 13 | 30 |
| 2 | 2024 | 16 | 96 |

Kuva 21. SQL-kysely valmistuneiden tuotantoerien hakemiseen

Kun aloitettujen ja lopetettujen kiekkojen määrät olivat selvillä, täytyi laskelmaa varten vielä selvittää millä viikolla tuotantoeristä on kiekkoja romutettu.

Näiden selvittämiseen pystyin hyödyntämään valmiiksi luodun SQL-kyselyn pohjaa, jonka olin Vaisalassa jo aiemmin työskennellessäni kehittänyt romutettujen kiekkojen raportointia varten. Kyseisessä raportissa käytetyn SQL-kyselyn yksittäinen rivi vastaa yhtä romutettua kiekkoa, sisältäen muun muassa kiekontunnisteen ja romutuksen aikaleiman, joten romutettujen määrän laskemiseen käytin COUNT-funktiota (kuva 22).

```
RomutetutKiekot AS
(
  SELECT
    KIEKKOTUNNISTE,
    DATEPART(yyyy, AIKALEIMA) AS VUOSI,
    DATEPART(isoww, AIKALEIMA) AS VIIKKO
  FROM Kiekkojen_Romutukset
),

RomutettujenMääräViikoittain AS
(
  SELECT
    VUOSI,
    VIIKKO,
    COUNT(KIEKKOTUNNISTE) AS MÄÄRÄ
  FROM RomutetutKiekot
  GROUP BY VUOSI, VIIKKO
)
```

Kuva 22. Lukumäärän laskeminen COUNT-koostefunktiolla

Jotta aloitettujen, lopetettujen ja romutettujen määrät saataisiin yhteen SQL-kyselyyn yhdistettyä, täytyi tilapäiset taulut yhdistää yhden kyselyn sisään, jossa ne liitettäisiin vuodet ja viikot sisältävään tauluun LEFT JOIN -liitoksilla (kuva 23). Päätimme yrityksen ohjaajan sekä anturitehtaan johtajan kanssa, että laskisimme tämän datan perusteella ylemmän tason viikoittaista kokonaisuutta, joten jaoin valmistuneiden määrän valmistuneiden ja romutettujen määrän summalla ja kerroin sen sadalla saadakseni prosenttiosuuden selville. Laskutoimituksessa ei tarvinnut huolehtia nolllalla jakamisesta, sillä jos kyseisellä viikolla ei ole olisi yhtään valmistunutta, olisi tämän kolumnin arvona NULL. Datan luettavuuden kannalta käytin FORMAT-funktiota, jolla saanto-prosentti pyöristettiin kahden desimaalin tarkkuuteen sekä CONCAT-funktiota, jolla yhdistin vuoden ja viikon yhdeksi kolumniksi.

```

77 SELECT
78   CONCAT(A.VUOSI, '-', A.VIIKKO) AS VUOSI_VIIKKO,
79   B.MÄÄRÄ AS ALOITETTUJEN_MÄÄRÄ,
80   C.MÄÄRÄ AS VALMISTUNEIDEN_MÄÄRÄ,
81   D.MÄÄRÄ AS ROMUTETTUJEN_MÄÄRÄ,
82   FORMAT(C.MÄÄRÄ / (C.MÄÄRÄ + COALESCE(D.MÄÄRÄ, 0)) * 100, 'N2') AS SAANTO
83 FROM VuodetJaViikot A
84 LEFT JOIN AloitetttujenMääräViikoittain B
85   ON B.VUOSI = A.VUOSI
86   AND B.VIIKKO = A.VIIKKO
87 LEFT JOIN ValmistuneidenMääräViikoittain C
88   ON C.VUOSI = A.VUOSI
89   AND C.VIIKKO = A.VIIKKO
90 LEFT JOIN RomutettujenMääräViikoittain D
91   ON D.VUOSI = A.VUOSI
92   AND D.VIIKKO = A.VIIKKO
93 ORDER BY A.VUOSI, A.VIIKKO

```

100 %

Results Messages

| | VUOSI_VIIKKO | ALOITETTUJEN_MÄÄRÄ | VALMISTUNEIDEN_MÄÄRÄ | ROMUTETTUJEN_MÄÄRÄ | SAANTO |
|---|--------------|--------------------|----------------------|--------------------|--------|
| 1 | 2024-13 | 120.0 | 28.0 | 1.0 | 96.55 |
| 2 | 2024-14 | 150.0 | 30.0 | 6.0 | 83.33 |

Kuva 23. Tilapäisten taulujen liittäminen tulosjoukkoon LEFT JOIN -liitoksella

SQL-kyselyn kehittämisen jälkeen täytyi tuotannonohjausjärjestelmässä vielä luoda uusi taulukkomuotoinen raportti, johon kysely liitettiin. Näin meillä oli syntynyt anturitehtaan mittarina toimiva kokonaissaantolaskelma, jonka avulla on mahdollista seurata viikkotasolla aloitettujen, valmistuneiden ja romutettujen määriä sekä valmistuneiden ja romutettujen suhdetta toisiinsa.

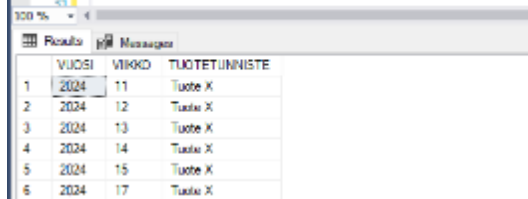
5.4 Tuotekohtainen saantolaskelma

Myös tuotekohtaisen saantolaskelman data ryhmiteltäisiin viikoittain, joten pystyin hyödyntämään vuosien ja viikkojen hakemisessa jo edellistä laskelmaa varten luomaani SQL-kyselyä, johon liitin CROSS JOIN -liitoksella tuotetunnisteet kaikki tuotteet sisältävästä taulusta, jotta jokainen vuotta ja viikkoa kuvaava rivi sisältäisi myös tuotetunnisteen (kuva 24). Tuotannonohjausjärjestelmässä generoiduissa raporteissa pystyy myös käyttämään käyttäjensyöttämiä parametrejä, jotka merkitään : -merkillä. Näin raporttiin saataisiin näkyviin käyttäjän syöttämän tuotteen tiedot, kun parametri poimitaan WHERE-ehdolauseesta. SELECT-käskyn yhteydessä käytin myös DISTINCT-määrettä, jottei tulosjoukko sisältäisi duplikaattirivejä.

```

1 WITH VuodetJaViikot AS
2 (
3     SELECT
4         VUOSI,
5         VIKKO
6     FROM VUOSI_JA_VIIKKO
7     UNION
8     SELECT
9         DATEPART(yyyy, GETDATE()) AS VUOSI,
10        DATEPART(isow, GETDATE()) AS VIKKO
11 ),
12
13 Tuotetunniste AS
14 (
15     SELECT
16         TUOTETUNNISTE
17     FROM TUOTE
18     WHERE TUOTETUNNISTE = 'Tuote X'
19     --WHERE TUOTETUNNISTE = :KAYTTAJAN_SVOTTAMA_PARAMETRI
20 ),
21
22 VuodetJaViikotJaTuotetunniste AS
23 (
24     SELECT DISTINCT *
25     FROM VuodetJaViikot
26     CROSS JOIN Tuotetunniste
27 )
28
29 SELECT *
30 FROM VuodetJaViikotJaTuotetunniste
31

```



| | VUOSI | VIKKO | TUOTETUNNISTE |
|---|-------|-------|---------------|
| 1 | 2024 | 11 | Tuote X |
| 2 | 2024 | 12 | Tuote X |
| 3 | 2024 | 13 | Tuote X |
| 4 | 2024 | 14 | Tuote X |
| 5 | 2024 | 15 | Tuote X |
| 6 | 2024 | 17 | Tuote X |

Kuva 24. Tuotetunnisteen liittäminen vuodet ja viikot sisältävään tulosjoukkoon

Aloitettujen tuotantoerien hakemien tietokannasta onnistui samaan tapaan kuin kokonaissaantolaskelmaa varten luodussa SQL-kyselyssä, mutta valmistuneiden tuotantoerien lisäksi halusimme myös tiedon niistä aloitetuista tuotantoeristä, jotka eivät ole päässeet tuotantolinjan loppuun asti koko tuotantoerän romuttamisen myötä. Kokonaan romutetuista tuotantoeristä tallentuisi myös romutustoiminnon yhteydessä rivi tuotantoerille suoritettujen toimintojen sisältävään tauluun, joten ne saatiin samaan tulosjoukkoon muokkaamalla valmistuneiden tuotantoerien kyselyn WHERE-ehtoa sisällyttämään kaikki ne rivit, joiden suoritettu toiminto on joko "merkitseValmiiksi" tai "romutaTuotantoerä".

Tuotantoerän romutustoiminnon yhteydessä tauluun tallentuva määrä oli kuitenkin romutustoiminnon aikainen kiekkomäärän, joten tämä saatiin ratkaistua CASE-lauseella (kuva 25). Jos tuotantoerä on romutettu kokonaan, mielletään se valmistuneeksi tuotantoeräksi, jonka kiekkomäärä on nolla. Jos taas erä on merkitty valmiiksi, on sen kiekkomäärä toiminnon suoritushetkinen määrä.

```

51 ValmistuneetJaRomutetutTuotantoerät AS
52 (
53     SELECT
54         ERÄTUNNISTE,
55         TUOTETUNNISTE,
56         CASE
57             WHEN SUORITETTU_TOIMINTO = 'merkitseValmiiksi' THEN MÄÄRÄ
58             WHEN SUORITETTU_TOIMINTO = 'romutaTuotantoerä' THEN 0
59         END AS MÄÄRÄ,
60         DATEPART(yyyy, AIKALEIMA) AS VUOSI,
61         DATEPART(isoww, AIKALEIMA) AS VIIKKO
62     FROM TUOTANTOERILLE_SUORITETUT_TOIMINNOT
63     WHERE KIEKKO_VAI_SIRU = 'KIEKKO'
64     AND Suoritettu_Toiminto IN ('merkitseValmiiksi', 'romutaTuotantoerä')
65 ),

```

Kuva 25. Valmistuneiden ja romutettujen tuotantoerien määrät CASE-lauseella

Yhdistämällä valmistuneet tuotantoerät ja romutetut tuotantoerät yhteen tulosjoukkoon pystyttiin tuotantoerien loppumäärät liittämään tuotantoon aloitettuihin tuotantoihin LEFT JOIN -liitoksella. Tällä tavoin pystyimme koostefunktioita käyttäen selvittää tiedon siitä, että kuinka monta tuotantoerää on kullakin viikolla aloitettu, mikä on kunkin viikon tuotantoerien yhteenlaskettu määrä, kuinka monen aloitetun kohtalo on ratkennut, mikä on suurin mahdollinen määrä tähän asti ratkenneista tuotantoeristä, mikä on ratkenneiden tuotantoerien lopullinen kiekkomäärä sekä kuinka paljon ratkenneista tuotantoeristä on kiekkoja (kuva 26).

```

56 LasketutMäärätViikoittain AS
57 (
58     SELECT
59         A.VUOSI,
60         A.VIIKKO,
61         A.TUOTETUNNISTE,
62         --Lasketaan aloitettujen tuotantoerien lukumäärä
63         COUNT(B.ERÄTUNNISTE) AS ALOITETTUJEN_TUOTANTOERIEN_MÄÄRÄ,
64         --Lasketaan aloitettujen tuotantoerien kiekkojen summa
65         SUM(B.MÄÄRÄ) AS ALOITETTUJEN_KIEKKOJEN_MÄÄRÄ,
66         --Jos aloitettujen erien määrä 0, niin ratkenneiden erien määrä NULL, muulloin lasketaan ratkenneiden erien lukumäärä
67         CASE
68             WHEN COUNT(B.ERÄTUNNISTE) = 0 THEN NULL
69             ELSE COUNT(C.ERÄTUNNISTE)
70         END AS RATKENNEIDEN_TUOTANTOERIEN_MÄÄRÄ,
71         --Lasketaan niiden erien aloitus kiekkomäärä yhteen, jotka ovat ratkenneet.
72         SUM(CASE WHEN C.ERÄTUNNISTE IS NOT NULL THEN B.MÄÄRÄ END) AS SUURIN_MAHDOLLINEN_VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ,
73         --Lasketaan valmistuneiden tuotantoerien kiekkojen summa
74         SUM(C.MÄÄRÄ) AS VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ,
75         --Lasketaan ratkenneiden erien kiekkomäärä yhteen, jonka jälkeen vähennetään sitä valmistuneiden kiekkojen määrä.
76         SUM(CASE WHEN C.ERÄTUNNISTE IS NOT NULL THEN B.MÄÄRÄ END) - SUM(C.MÄÄRÄ) AS ROMUTETTUJEN_KIEKKOJEN_MÄÄRÄ
77     FROM VuodetJaViikotJaTuotetunniste A
78     LEFT JOIN AloitetutTuotantoerät B
79         ON B.VUOSI = A.VUOSI
80         AND B.VIIKKO = A.VIIKKO
81         AND B.TUOTETUNNISTE = A.TUOTETUNNISTE
82     LEFT JOIN ValmistuneetJaRomutetutTuotantoerät C
83         ON C.ERÄTUNNISTE = B.ERÄTUNNISTE
84     GROUP BY A.VUOSI, A.VIIKKO, A.TUOTETUNNISTE
85 )
86
87 SELECT *
88 FROM LasketutMäärätViikoittain

```

| | VUOSI | VIIKKO | TUOTETUNNISTE | ALOITETTUJEN_TU... | ALOITETTUJEN_KI... | RATKENNEIDEN_TU... | SUURIN_MAHD... | VALMISTUNEI... | ROMUTETT... |
|---|-------|--------|---------------|--------------------|--------------------|--------------------|----------------|----------------|-------------|
| 1 | 2024 | 13 | Tuote X | 2 | 60 | 2 | 60 | 30 | 30 |
| 2 | 2024 | 14 | Tuote X | 1 | 30 | 1 | 30 | 24 | 6 |
| 3 | 2024 | 13 | Tuote Y | 1 | 30 | 1 | 30 | 28 | 2 |
| 4 | 2024 | 14 | Tuote Y | 3 | 90 | 2 | 60 | 17 | 43 |
| 5 | 2024 | 13 | Tuote Z | 1 | 30 | 1 | 30 | 28 | 2 |
| 6 | 2024 | 14 | Tuote Z | 1 | 30 | 0 | NULL | NULL | NULL |

Kuva 26. Tilapäisten taulujen yhdistäminen määrien selvittämiseksi

Saadun tulosjoukon perusteella määristä pystyttiin laskea kunkin viikon aloitettujen tuotantoerien sen hetkinen saanto jakamalla valmistuneiden kiekkojen määrä suurimmalla mahdollisella valmistuneiden kiekkojen määrällä ja kertomalla se sadalla. Lisäksi saannon sen hetkinen tarkkuus eli kuinka iso osa aloitettujen tuotantoerien kohtaloista on tiedossa ja sen myötä huomioitu saannossa pystyttiin laskea jakamalla ratkenneiden tuotantoerien määrä aloitettujen tuotantoerien määrällä ja kertomalla se sadalla (kuva 27).

```

87 SELECT
88     CONCAT(VUOSI, '-', VIIKKO) AS VUOSI_VIIKKO,
89     TUOTETUNNISTE,
90     ALOITETTUJEN_TUOTANTOERIEN_MÄÄRÄ,
91     ALOITETTUJEN_KIEKKOJEN_MÄÄRÄ,
92     RATKENNEIDEN_TUOTANTOERIEN_MÄÄRÄ,
93     SUURIN_MAHDOLLINEN_VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ,
94     VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ,
95     ROMUTETTUJEN_KIEKKOJEN_MÄÄRÄ,
96     /*Kun suurin mahdollinen valmistuneiden kiekkojen määrä on erisuuri kuin nolla,
97     jaetaan valmistuneiden kiekkojen määrä suurimmalla mahdollisella valmistuneiden kiekkojen määrällä ja kerrotaan se sadalla.*/
98     CASE
99         WHEN SUURIN_MAHDOLLINEN_VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ != 0
100        THEN FORMAT((VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ * 1.0) / (SUURIN_MAHDOLLINEN_VALMISTUNEIDEN_KIEKKOJEN_MÄÄRÄ * 1.0) * 100, 'N2')
101        END AS SAANTO,
102     /*Kun ratkenneiden erien määrä on erisuuri kuin nolla,
103     jaetaan ratkenneiden erien määrä aloitettujen erien määrällä ja kerrotaan se sadalla.*/
104     CASE
105         WHEN RATKENNEIDEN_TUOTANTOERIEN_MÄÄRÄ != 0
106        THEN FORMAT((RATKENNEIDEN_TUOTANTOERIEN_MÄÄRÄ * 1.0) / (ALOITETTUJEN_TUOTANTOERIEN_MÄÄRÄ * 1.0) * 100, 'N2')
107        END AS SAANNON_TARKKUUS_ELI_RATKENNEIDEN_OSUUS_ALOITETUISTA
108 FROM LasketutMäärätViikoittain
109 ORDER BY VUOSI, VIIKKO, TUOTETUNNISTE

```

| | VUOSI_VIIKKO | TUOTETUNNISTE | ALOITETTUJEN_TU... | ALOITETTUJEN_KI... | RATKE... | SUURIN_M... | VALMISTUN... | ROMUTETTUJ... | SAANTO | SAANNON_TARKK... |
|---|--------------|---------------|--------------------|--------------------|----------|-------------|--------------|---------------|--------|------------------|
| 1 | 2024-13 | Tuote X | 2 | 60 | 2 | 60 | 30 | 30 | 50.00 | 100.00 |
| 2 | 2024-13 | Tuote Y | 1 | 30 | 1 | 30 | 28 | 2 | 93.33 | 100.00 |
| 3 | 2024-13 | Tuote Z | 1 | 30 | 1 | 30 | 28 | 2 | 93.33 | 100.00 |
| 4 | 2024-14 | Tuote X | 1 | 30 | 1 | 30 | 24 | 6 | 80.00 | 100.00 |
| 5 | 2024-14 | Tuote Y | 3 | 90 | 2 | 60 | 17 | 43 | 28.33 | 66.67 |
| 6 | 2024-14 | Tuote Z | 1 | 30 | 0 | NULL | NULL | NULL | NULL | NULL |

Kuva 27. Lopullinen tulosjoukko ja tuotekohtaisen saannon laskeminen

SQL-kyselyn kehittämisen jälkeen liitin sen omaan uuteen raporttiinsa tuotannonohjausjärjestelmässä. Lopullinen raportti on ulkoasultaan taulukkomuotoinen, eli se muistuttaa kuvassa 27 näkyvää SQL Server Management Studion tulostaulua. Kyseisen raportin avulla anturitehtaalla pystytään seuraamaan kunkin viikon aloitettujen tuotantoerien päivittyvää saantoa mahdollistaen tarkemman saannon seurannan tuotekohtaisesti syvempiä analyysejä varten.

6 PÄÄTÄNTÖ

Opinnäytetyö onnistui mielestäni hyvin, vaikkakaan se ei aivan täysin sen alkuperäistä tavoitetta täytä. Lopputuloksena syntyneet saantolaskelmat piti alun perin olla visualisoidun datan muodossa, mutta huomasimme kehitystyön

aikana, että tuotannonohjausjärjestelmän mahdollistamien raporttien skaalautuvuus ei ollut tarpeeksi kustomoitavissa, mikä teki niiden lukemisesta ja ymmärtämisestä vaikeaa. Näin ollen päätimme, että laskelmat noudattaisivat perinteisempää taulukkomuotoa. Tämä toki mahdollistaa sen, että laskelmien sisältämä data on tarvittaessa mahdollista siirtää muihin datan visualisointiin ja analysointiin tarkoitettuihin ohjelmiin.

Alkuperäisten suunnitelmien ja opinnäytetyön rajauksen mukaan saantolaskelmiin piti sisällyttää anturikiekkaja sisältävien tuotantoerien lisäksi myös anturisiruja sisältävät tuotantoerät. Nämä kuitenkin päätettiin rajata työstä pois työn loppu metreillä minusta riippumattomista syistä.

Työn teoria sisältää mielestäni kattavasti tietoa datasta ja sen visualisoinnista sekä tietokannoista ja SQL-kyselykielestä. Tuotannonohjauksen ja siihen liittyvien käsitteiden osuus teoriassa jäi kuitenkin hieman suppeammaksi, mihin varmasti vaikutti myös se, että työn muu aihepiiri on opintojen myötä tullut tummaksi.

Myös kehitystyön dokumentointi onnistui mielestäni olosuhteisiinsa nähden hyvin, vaikkakaan se ei sisällä varsinaisessa työn lopputuoksessa käytettyjä ratkaisuja tietojen luottamuksellisuuden myötä. Dokumentaation perusteella lukijalle kuitenkin mielestäni syntyy selkeä kuva siitä, miten työ on edennyt sekä minkä tyyppisiä ratkaisuja on tehty.

Yksi selkeä jatkokehitysidea mielestäni olisi anturisiruja sisältävien tuotantoerien sisällyttäminen omiin vastaaviin saantolaskelmiinsa. Näiden kehityksessä myös pystytään hyvin todennäköisesti hyödyntämään opinnäytetyön aikana kehitettyjen SQL-tietokantakyselyiden pohjia, jolloin kehitystyössä säästetään aikaa.

LÄHTEET

- Arkkola, J. 2023. MES-järjestelmän tuomat hyödyt valmistavassa teollisuudessa. Blogi. Päivitetty 28.9.2023. Saatavissa: <https://blog.pinja.com/mes-jarjestelman-tuomat-hyodyt-valmistavassa-teollisuudessa> [viitattu 7.2.2024].
- Björklund, J. 2019. SQL Serverin ja MongoDB:n vertailu ja transaktioiden hallinta. Haaga-Helia ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:amk-2019051810483> [viitattu 15.2.2024].
- Computer Hope. 2020. What is the difference between data and information? WWW-dokumentti. Päivitetty 31.8.2020. Saatavissa: <https://www.computerhope.com/issues/ch001629.htm> [viitattu 4.3.2024].
- Computer Hope. 2023. Data. WWW-dokumentti. Päivitetty 29.12.2023. Saatavissa: <https://www.computerhope.com/jargon/d/data.htm> [viitattu 4.3.2024].
- Dybka, P. 2016. SQL JOINS Explained with Venn Diagrams. Blogi. Päivitetty 21.11.2016. Saatavissa: <https://learnsql.com/blog/sql-joins/> [viitattu 1.3.2024].
- Eyelit. s.a. Eyelit Manufacturing Execution System (MES). WWW-dokumentti. Saatavissa: <https://www.eyelit.com/products/mes/> [viitattu 7.3.2024].
- Fikuro. 2023. Mikä on MES-järjestelmä ja miten pk-yritys siitä hyötyy? Blogi. Päivitetty 6.11.2023. Saatavissa: <https://www.fikuro.fi/blogi/mes-jarjestelma> [viitattu 7.2.2024].
- Galbraith, B. 2023. Data vs. Information: What's the Difference? Blogi. Päivitetty 5.6.2023. Saatavissa: <https://bloomfire.com/blog/data-vs-information/> [viitattu 4.3.2024].
- Gygi, C., Williams, B., DeCarlo, N. & Covey, S. 2016. How to Measure Yield for Six Sigma. WWW-dokumentti. Päivitetty 26.3.2016. Saatavissa: <https://www.dummies.com/article/business-careers-money/business/project-management/how-to-measure-yield-for-six-sigma-170183/> [viitattu 7.2.2024].
- Hannus, E. & Lampi, M. s.a. Datavisualisointiopas. Kaakkois-Suomen ammattikorkeakoulu. WWW-dokumentti. Saatavissa: <https://www.xamk.fi/tutkimus-ja-kehitystoiminta/dataopas/> [viitattu 5.3.2024].
- Hovi, A. 2004. SQL-opas. Jyväskylä: Docendo Finland. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/Record/kaakkuri.171926?sid=4063989818> [viitattu 26.2.2024].
- Jalonen, H. & Kosonen, M. 2020. Visualisointi saa datan puhumaan äänellä, joka kuullaan. *Tietoasiantuntija* 2–3, 32–33. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi-fe2020062445588> [viitattu 7.2.2024].
- Jotform. 2023. How to do data visualization in 7 steps. Blogi. Päivitetty 15.12.2023. Saatavissa: <https://www.jotform.com/blog/how-to-do-data-visualization/> [viitattu 5.3.2024].

Microsoft. 2023a. SQL Server Management Studio components and configuration. WWW-dokumentti. Päivitetty 1.3.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/ssms/tutorials/ssms-configuration?view=sql-server-ver16> [viitattu 23.2.2024].

Microsoft. 2023b. What is SQL Server Management Studio (SSMS)? WWW-dokumentti. Päivitetty 31.3.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16> [viitattu 23.2.2024].

Microsoft. 2023c. Joins (SQL Server). WWW-dokumentti. Päivitetty 13.7.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/relational-databases/performance/joins?view=sql-server-ver16> [viitattu 23.2.2024].

Microsoft. 2023d. Aggregate Functions (Transact-SQL). WWW-dokumentti. Päivitetty 23.5.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver16> [viitattu 22.4.2024].

Microsoft. 2023e. CASE (Transact-SQL). WWW-dokumentti. Päivitetty 13.7.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/t-sql/language-elements/case-transact-sql?view=sql-server-ver16> [viitattu 22.4.2024].

Microsoft. 2023f. CONCAT (Transact-SQL). WWW-dokumentti. Päivitetty 25.9.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/t-sql/functions/concat-transact-sql?view=sql-server-ver16> [viitattu 22.4.2024].

Microsoft. 2023g. COALESCE (Transact-SQL). WWW-dokumentti. Päivitetty 23.5.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/t-sql/language-elements/coalesce-transact-sql?view=sql-server-ver16> [viitattu 22.4.2024].

Microsoft. 2023h. Set Operators - UNION (Transact-SQL). WWW-dokumentti. Päivitetty 23.5.2023. Saatavissa: <https://learn.microsoft.com/en-us/sql/t-sql/language-elements/set-operators-union-transact-sql?view=sql-server-ver16> [viitattu 22.4.2024].

Microsoft. 2024. What is SQL Server? WWW-dokumentti. Päivitetty 14.2.2024. Saatavissa: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16> [viitattu 23.2.2024].

Microsoft. s.a. What are databases? WWW-dokumentti. Saatavissa: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-databases/> [viitattu 15.2.2024].

Mäkelä, V. 2015. Ensisaannon automaattisen laskennan mahdollisuudet Epec Oy:n elektroniikkatuotannon automaattisilla testauslaitteistoilla. Seinäjoen ammattikorkeakoulu. Yrittäjyyden ja liiketoimintaosaamisen koulutusohjelma. Opinnäytetyö. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:amk-201505229653> [viitattu 6.3.2024].

Nikupeteri, S. 2016. Mittarointi tuotannonohjauksen ja kehittämisen tukena. Oulun yliopisto. Teknillinen tiedekunta. Diplomityö. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:oulu-201606022118> [viitattu 6.3.2024].

Pakkasela, M. & Porri-Henttinen, H. 2021. Data-analytiikka ja sen visualisointi. WWW-dokumentti. Päivitetty 17.11.2021. Saatavissa: <https://www.tiedollajohdaminen.fi/uutiset/data-analytiikka-ja-sen-visualisointi/> [viitattu 1.3.2024].

Pengon Oy. 2016. Tiedon visualisoinnin hyödyt. Blogi. Päivitetty 16.3.2016. Saatavissa: <https://blogi.pengon.fi/arkisto/tiedon-visualisointi-hyodyt> [viitattu 7.2.2024].

Peterson, R. 2023. What is SQL Server? Introduction, History, Types, Versions. Blogi. Päivitetty 21.11.2023. Saatavissa: <https://www.guru99.com/sql-server-introduction.html> [viitattu 20.2.2024].

Precognize. s.a. Manufacturing Yield. WWW-dokumentti. Saatavissa: <https://precog.co/glossary/manufacturing-yield/> [viitattu 7.2.2024].

Taylor, A. 2013. SQL for Dummies. 8. uudistettu painos. John Wiley & Sons, Incorporated. E-Kirja. Saatavissa: <https://ebookcentral.proquest.com/lib/xamk-ebooks/detail.action?docID=1335264> [viitattu 6.2.2024].

Vaisala. 2021. Virtual visit to Vaisala's cleanroom. Youtube. Videoleike. Päivitetty 20.5.2021. Saatavissa: <https://www.youtube.com/watch?v=bMXbSeBpvSM> [viitattu 15.4.2024].

Vaisala BAROCAP® -teknologia s.a. Vaisala Oyj. WWW-dokumentti. Saatavissa: <https://www.vaisala.com/fi/vaisala-barocapr-technology> [viitattu 16.4.2024].

Vaisala Oyj. 2024. Tietoa meistä. WWW-dokumentti. Saatavissa: <https://www.vaisala.com/fi/vaisala-company> [viitattu 15.4.2024].

What Is a Database? s.a. Oracle. WWW-dokumentti. Saatavissa: <https://www.oracle.com/database/what-is-database/> [viitattu 15.2.2024].

What is a Relational Database (RDBMS)? s.a. Oracle. WWW-dokumentti. Saatavissa: <https://www.oracle.com/database/what-is-a-relational-database/> [viitattu 15.2.2024].