



# **TIETOVARASTON RAPORTOINTITÄULUJEN OPTIMOINTI PILVIALUS- TALLA**

Lappeenrannan–Lahden teknillinen yliopisto LUT

Tuotantotalouden diplomityö

2024

Aleksi Oja

Tarkastajat: Tutkijatohtori, dosentti Lasse Metso

Tutkijatohtori Antti Ylä-Kujala

## TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tuotantotalous

Aleksi Oja

### **Tietovaraston raportointitaulujen optimointi pilvialustalla**

Tuotantotalouden diplomityö

2024

67 sivua, 21 kuvaa, 7 taulukkoa ja 9 liitettä

Tarkastajat: Tutkijatohtori, dosentti Lasse Metso & Tutkijatohtori Antti Ylä-Kujala

Avainsanat: Pilvitietovarastointi, raportointitaulu, optimointi

Pilvialustalla sijaitseva tietovarastointi on kasvavassa määrin yritysten suosima tietovarastointiratkaisu, joka on tuonut uudenlaisia haasteita vastaan. Eräs näistä on pilvialustojen käytön mukaan laskutettavien kulujen optimointi, johon tämä tutkimus liittyy. Tämä toimeksiantona tehty diplomityö sai alkunsa organisaatiossa meneillään olevasta kolmannen osapuolen ylläpitämän on-premise tietovarastointiratkaisun siirrosta organisaation hallinnoimalle pilvialustalle. Pilvitietovaraston käyttökustannukset halutaan pitää alhaisina, ja tätä varten tarvitaan tietoa vaikuttavista tekijöistä.

Tutkimuksen tavoitteena oli selvittää pilvitietovarastossa sijaitsevien raportointitaulujen kyselyjen optimointitavat kulujen ja nopeuden osalta sekä muunto- että tallennusvaiheessa. Muuntovaiheen optimoinnin pääteltiin liittyvän tehokkaan SQL:n käyttämiseen muunto-operaatioissa. Tehokkaat SQL-pohjaiset muunto-operaatiot selvitettiin kirjallisuuskatsauksella ja nämä, SQL-kieltä, tehostavat keinot koottiin taulukkoon.

Tallennusvaiheen optimoinnin selvitystä varten luotiin ensin kirjallisuuskatsaus optimointikeinoihin ja tämän pohjalta luotiin vuokaavio, joka pyrkii ohjaamaan optimaaliseen raportointitaulun tallennustapaan kyselyjen kannalta. Vuokaaviota testattiin simuloinnin kautta. Lisäksi suorituskykytulosten kautta pääteltiin, ohjaako vuokaavio aina optimaaliseen raportointitaulun tallennustapaan.

Tutkimuksen tulokset sisältävät koottua tietoa tekijöistä, jotka vaikuttavat pilvitietovaraston raportointitaulujen optimointiin, sekä tallennustapaa ohjaavan vuokaavion. Vuokaavio onnistui lähes aina ohjaamaan optimaaliseen tallennustapaan, mutta muutamassa tapauksessa vain tehostamaan tallennusmuotoa kyselyjen kannalta.

## ABSTRACT

Lappeenranta–Lahti University of Technology LUT

School of Engineering Science

Industrial Engineering and Management

Alexsi Oja

### **Optimization of data warehouse reporting tables on a cloud platform**

Master's thesis

2024

67 pages, 21 figures, 7 tables and 9 appendices

Examiners: Post-doctoral Researcher, Docent Lasse Metso &

Post-doctoral Researcher Antti Ylä-Kujala

Keywords: Cloud data warehousing, reporting table, optimization

Data warehousing located on cloud platforms is increasingly becoming a preferred solution for companies, bringing along new challenges. One of these challenges is optimizing the costs billed according to the usage of cloud platforms, which is the focus of this research. This master's thesis originated from an ongoing migration of a third-party maintained on-premises data warehousing solution to the organization's managed cloud platform. The objective is to keep the cost level of the cloud data warehouse usage low, requiring insights into factors affecting this.

The aim of the research was to investigate optimization methods for reporting tables located in the cloud data warehouse in terms of costs and speed, both in the transformation and storage phase concerning queries. From the optimization of the transformation phase, it was inferred to be associated with the efficient use of SQL. Efficient SQL-based transformation operations were identified through a literature review of previous studies, and these SQL-enhancing methods were compiled into a table.

For the investigation of storage phase optimization, a literature review of optimization methods was conducted, and based on this, a flowchart was created aiming to guide towards an optimal storage method for reporting tables from a query perspective. The flowchart was tested through simulation of its usage and performance results were analyzed to determine if the flowchart consistently guides towards an optimal storage method for reporting tables.

The results of the study provide compiled information on factors influencing the optimization of cloud data warehouse reporting tables, as well as a flowchart guiding the storage method. The flowchart almost always succeeded in directing towards the optimal storage method, but in a few cases, it only enhanced the storage format from a query perspective.

## ALKUSANAT

Kiitos Lappeenrannan-Lahden teknilliselle yliopistolle mahdollisuudesta joustaviin opintoihin sekä laadukkaasta opetuksesta. Vuonna 2020 alkaneet opinnot saadaan tämän työn myötä päätökseen. Matka ei ole aina ollut helpoin töiden ja perhe-elämän ohessa opiskellessa, mutta olen erittäin kiitollinen siitä, että ajan puutteesta huolimatta opinnot saatiin joustavasti järjestettyä.

Haluan myös kiittää perhettäni, joka on ollut vankkumaton tukipilari opintojeni aikana. Kiitos rohkaisusta ja tuesta, jotka ovat olleet korvaamattomia matkallani kohti tämän hetken saavuttamista.

Suuret kiitokset myös diplomityöni ohjaajalle Pasi Ruhaselle arvokkaasta ohjauksesta ja tuesta. Hänen kannustuksensa ja kiinnostuksensa työtäni kohtaan ovat olleet ratkaisevassa roolissa tämän työn edistymisessä.

## LYHENNE- JA KÄSITELUETTELO

### Lyhenteet

BI	Business Intelligence, liiketoimintatiedon hallinta/hyödyntäminen
DDL	Data Definition Language
DML	Data Manipulation Language
DSR	Design Science Research
ELT	Extract, load, transform. Datan haku, lataus ja muunto.
GCP	Google Cloud Platform
SQL	Structured Query Language
SSB	Star Schema Benchmark
TPC	Transaction Processing Performance Council

### Käsitteet

<i>Data mart</i>	Tietovaraston osajoukko, joka vastaa tietyn liiketoiminta-alueen tarpeeseen
<i>Granulariteetti</i>	Tiedon granulariteetti tai karkeusaste viittaa tietojen tarkkuuden tasoon. Mitä suurempi tai hienojakoisempi granulariteetti, sen yksityiskohtaisempi on tiedon taso.
<i>Kardinaliteetti</i>	Viittaa tietokantataulun attribuutin uniikkien arvojen määrään. Kardinaliteetti voi olla korkea tai matala. Mitä korkeampi kardinaliteetti, sen enemmän uniikkeja arvoja on datassa.
<i>On-premise</i>	IT-infrastruktuurin sijoitus ja ylläpito omassa fyysisessä tilassa tai datakeskuksessa
<i>Raportointitaulu</i>	Tietovarastossa sijaitsevaa objekti, johon raportointityökalu yhdistetään ja johon raportointityökalu kohdistaa ajamansa kyselyt.
<i>Siirräntä</i>	Input / Output, I/O – informaation siirtämistä tietokoneen komponenttien välillä
<i>Skaala</i>	Viittaa työn empiirisen osuuden testiaineiston kokoon. Mitä suurempi skaala, sen suurempi on testiaineisto kooltaan.

## Sisällysluettelo

Tiivistelmä

Abstract

Alkusanat

Lyhenne- ja käsiteluettelo

1	Johdanto.....	8
1.1	Työn tausta.....	8
1.2	Rajaukset ja tavoitteet .....	9
1.3	Aineisto ja menetelmät .....	11
1.4	Työn rakenne .....	12
1.5	Tekoälyn hyödyntäminen työssä .....	13
2	Datan muuntamisen optimointi pilvitietovarastossa .....	14
2.1	Modernin tietovarastoinnin ja raportoinnin arkkitehtuuri .....	14
2.2	Pilvitietovaraston kustannusrakenne.....	16
2.3	Datan muuntaminen.....	17
2.4	SQL ja sen optimointi .....	19
3	Kolumnaarisen tietovaraston raportointitaulujen optimointi.....	23
3.1	Kolumnaariset tietovarastot .....	23
3.2	Tietovarastojen raportointitaulut.....	26
3.3	Raportointitaulujen optimoinnin keinot.....	27
4	Raportointitaulun tallennustapaa ohjaava vuokaavio.....	32
4.1	Design Science Research tutkimusmenetelmänä.....	32
4.2	Vuokaavion suunnittelu .....	35
4.3	Vuokaavion käytön simulointi.....	39
4.3.1	Käytetty data .....	39
4.3.2	Testijärjestelyjen kuvaus.....	41
4.3.3	Testauksen mittarit ja mittaus .....	45
4.3.4	Vuokaavion ohjaamat tallennustavat, tulokset ja tulosten analysointi .....	46
4.4	Vuokaavion jatkokehitys .....	53
4.5	Vuokaavion jatkotestaus ja -arviointi .....	57
5	Johtopäätökset .....	60

5.1	Tutkimuksen luotettavuus ja rajoitteet.....	62
5.2	Jatkotutkimusehdotukset.....	63
	Lähteet .....	65

## **Liitteet**

Liite 1. Star Schema Benchmarkin alkuperäiset ja tutkimusta varten uudelleenkirjoitetut kyselyt

Liite 2. Vuokaavion ensimmäisen version ohjaamat tallennustavat Q1.1 osalta

Liite 3. Vuokaavion ensimmäisen version ohjaama tallennustapa Q1.2 osalta

Liite 4. Vuokaavion ensimmäisen version ohjaama tallennustapa Q2.3 osalta

Liite 5. Vuokaavion ensimmäisen version ohjaama tallennustapa Q3.2 osalta

Liite 6. Vuokaavion toisen version ohjaamat tallennustavat Q1.1 osalta.

Liite 7. Vuokaavion toisen version ohjaama tallennustapa Q1.2 osalta.

Liite 8. Vuokaavion toisen version ohjaama tallennustapa Q2.3 osalta.

Liite 9. Vuokaavion toisen version ohjaama tallennustapa Q3.2 osalta.

# 1 Johdanto

## 1.1 Työn tausta

Jatkuvasti digitalisoituvassa maailmassa organisaatioiden trendi on IT-palvelujen pilveen siirtäminen, kertoo Gartnerin (2023a) raportti. He ennustavat, että vuoteen 2026 mennessä 75 % organisaatioista olisi adoptoinut pilvipohjaisen alustan ydinpalveluillaan. Gartnerin (2023b) mukaan pankki- ja finanssisektori, johon tämän työn toimeksiantaja kuuluu, noudattaa myös tätä trendiä. IT-palveluihin käytettävä rahoitus on tällä alalla kasvusuunnassa ja suurin osa rahoituksesta ohjataan kyberturvallisuuteen, dataan ja data-analytiikkaan, integraatioteknologioihin sekä pilvessä sijaitseviin palveluihin.

Tämän diplomityön toimeksiantajaorganisaatio (myöhemmin organisaatio) on tunnettu Suomessa toimiva pankkiryhmä, jolla on edessään yksi haastavimmista IT-projekteista, joka pankeille voi ylipäättään tulla: peruspankkijärjestelmän vaihtaminen. Tämän vaihdon myötä uusiutuu osa nykyisistä pankin käyttämissä IT-järjestelmistä ja jäljelle jäävät integroidaan soveltuvien osien uuteen peruspankkijärjestelmään ja muihin IT-järjestelmiin. Tässä työssä keskitytään yhteen uusiutuvista järjestelmistä, tietovarastoon. Organisaation nykyisin käyttämä tietovarasto on kolmannen osapuolen ylläpitämä on-premise tietovarasto, kun taas uutta tietovarastoa rakennetaan pilvialustan päälle. Pilvitietovarastojen kustannusrakenne on oleellisesti erilainen verrattuna on-premise-tietovarastoihin ja kohdatut rajoitteet sekä ongelmatkin eroavat toisistaan. Esimerkiksi on-premise tietovarastolle tehtävien kyselyjen tehokkuudella ei ole ollut tähän asti suurta merkitystä, sillä huonosti rakennetusta raportointitaulusta ja raportointityökalujen tekemisestä kyselyistä on koitunut ainoastaan raportoinnin hitautta, mutta ei lisälaskua. Pilvitietovarastot kerryttävät kustannuksia yleensä käytön mukaan, joten raportointitaulujen muodostukseen ja tallennustapaan on kiinnitettävä aiempaa enemmän huomiota.

Tietovarastomuutoksen lisäksi tietovarastoa hyödyntävän raportointi- ja analytiikkatiimin tehtäväalue laajenee. Projektin myötä tiimi osallistuu jatkossa myös datan muunto-osuuteen ELT-ketjussa (extract, load, transform eli haku, lataus ja muunto) ja näin ollen tulee osallistumaan raportointitaulujen luontiin, joihin raportit tullaan yhdistämään. Tehtäväalueen ja roolin ollessa tiimille uusi, puuttuu raportointitaulun luomisesta vielä parhaita käytäntöjä

noudattava ohjeistus, jolla taataan yhteneväinen tapa nopean ja matalakuluisen raportointitaulun luontiin. Tämä työ pyrkiikin selvittämään parhaat tavat muuntaa ja tallentaa raportointitaulut, jonka pohjalta organisaatio voi muodostaa itselleen ohjeistuksen.

Tutkimuksen aikana uusi tietovarasto on jo osittain otettu käyttöön organisaatiossa. Raportointitaulujen toteutustyyliä on tällä hetkellä organisaation raportointi- ja analytiikkatiimillä kuitenkin useita ja näissä on optimointivaraa. Koska tietovarasto on organisaatiossa suhteellisen tuore, on nyt järkevää optimoida raportointitaulut, jotta raportoinnin kulutaso pysyy jatkossakin maltillisena.

## 1.2 Rajaukset ja tavoitteet

Tutkimuksen rajauksissa keskityttiin niihin asioihin, jotka koskevat organisaation raportointi- ja analytiikkatiimin uutta tehtäväaluetta. Näin ollen tutkimuksesta päätettiin rajata pois ELT-ketjun kaksi ensimmäistä osaa, eli tiedon haku (extract) ja lataus (load). Syy tähän on se, että kyseessä olevan tiimin tehtäviin ei pääsääntöisesti kuulu tiedon haku- ja lataustehtävät ulkoisista lähteistä. Toisin sanoen keskitytään datan muunnon ja sen lopputuotteena olevan raportointitaulun optimointiin, jota raportointityökalut hyödyntävät. Datan muunnon osalta rajattiin tutkimus koskemaan vain SQL-pohjaisia muuntoja, sillä se on yleisesti sovellettu tapa suorittaa muuntoja, eikä se ole sidoksissa mihinkään tiettyyn muuntotyökaluun. Raportointitauluja tallentavien tietovarastojen osalta tutkimus rajattiin koskemaan ainoastaan tietovarastoja, jotka tallentavat datan sarakepohjaisesti. Viimeisenä rajauksena päätettiin, että keskitytään julkipilven alustalla olevaan ympäristöön, sillä näiden kustannusrakenne ja näin ollen optimointi kustannusten osalta eroaa merkittävästi on-premise ratkaisun optimoinneista.

Tutkimuksen empiirinen osuus suoritettiin Google Cloud Platformin tarjoamien työkalujen, eli Cloud Storagen, BigQueryn ja Looker Studion avulla, rajaten pois muut pilvialustat empiirisestä osuudesta. Kaikkien suurimpien pilvialustojen sisällyttäminen tutkimuksen käytännön osuuteen olisi paisuttanut käytännön tutkimuksen osuuden liian suureksi. Googlen pilvialustan työkaluihin päädyttiin siitä syystä, että organisaatiolla on nämä työkalut käytössä.

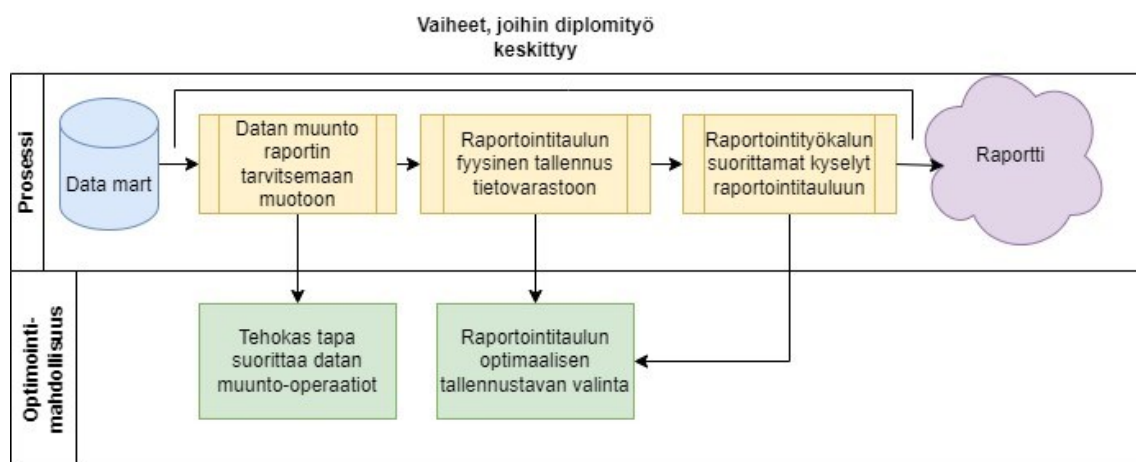
Tutkimuksen tavoitteena on selvittää pilvitietovarastossa sijaitsevien raportointitaulujen optimointitavat kulutehokkuuden ja nopeuden osalta sekä muunto että tallennusvaiheessa kyselyjen kannalta. Tämän lisäksi tavoitteena on luoda oikeaan tallennustapaan ohjaava ratkaisu, joka helpottaa käyttäjää optimoinnissa. Tavoitteeseen pääsyä varten tutkimusta on jäsennelty seuraavilla tutkimuskysymyksillä:

*Millä tavoin SQL-pohjaista datan muuntoa voidaan optimoida pilvitietovarastossa?*

*Millä tavoin pilvitietovaraston raportointitaulujen tallennusta voidaan optimoida kyselyjen kannalta?*

*Kuinka valitaan optimaalinen raportointitaulun tallennustapa?*

Tutkimuksen tulokset hyödyttävät organisaatiota siten, että niiden pohjalta organisaatio voi laatia raportointitaulujen muuntoon ja tallennustapaan liittyvän ohjeistuksen. Tulosten pohjalta laadittu ohjeistus tarjoaa rakenteellisemmän tavan raportointi- ja analytiikkatiimin uusien työtehtävien suorittamiseen. Tämän lisäksi ohjeistus nostaa tiimin osaamista raportointitaulun optimoinnissa, ja tätä kautta organisaation tietovaraston käyttökustannukset pysyvät maltillisina. Työn tuloksista saattaa olla hyötyä myös laajemmalle yritysjoukolle, mikäli heillä on käytössään tutkimuksen rajauksia vastaava tietovarastointiympäristö. Alla oleva kuva 1 pyrkii havainnollistamaan vaihteita, joihin tämä työ keskittyy.



Kuva 1. Tietovarastoinnin ja raportoinnin vaiheet, joihin työ keskittyy.

### 1.3 Aineisto ja menetelmät

Tutkimuksessa keskitytään kahteen eri vaiheeseen, joissa on optimointimahdollisuuksia: datan muuntaminen ja raportointitaulun optimoitu tallennus kyselyjen kannalta. Datan muuntamiseen liittyvä osuus pohjautuu kirjallisuuskatsaukseen ja raportointitaulun tallentamisen optimointiin perehtyvä osa tulee kirjallisuuskatsauksen lisäksi tutkimuksen empiirisestä osuudesta. Datan muunnon osalta tehtiin rajaus, että keskitytään ainoastaan SQL-pohjaisiin muuntoihin, eli toisin sanoen SQL-kielen tehokkuuteen. Tämä osa-alue on jo laajasti tutkittu, joten tähän tutkimuksen osuuteen katsottiin kirjallisuuskatsauksen olevan riittävä.

Tutkimusmenetelmänä empirian osalta käytettiin Peffers, Tuunanen, Rothenberger & Chatterjeen (2007) kehittämää Design Science Research -metodologian prosessimallia, sillä raportointitaulun optimointiin liittyy valintoja ja näiden ohjaamiseen haluttiin luoda selkeä ratkaisu. Design Science Researchin (myöhemmin DSR) avulla luodaan vuokaavio ohjaamaan optimaalisen raportointitaulun tallennusta. Tämä vuokaavio on tutkimuksen artefakti, jonka toimivuutta analysoitiin sen käytön simuloinnin kautta ja mittaamalla vuokaavion ohjaamien tallennustapojen tehokkuutta. Tutkimuksen artefaktin ensimmäinen versio muodostettiin kirjallisuuskatsauksen pohjalta, joten kirjallisuuskatsauksessa käsitellään sekä datan muuntamista, että tietovarastointia ja niiden raportointitauluja.

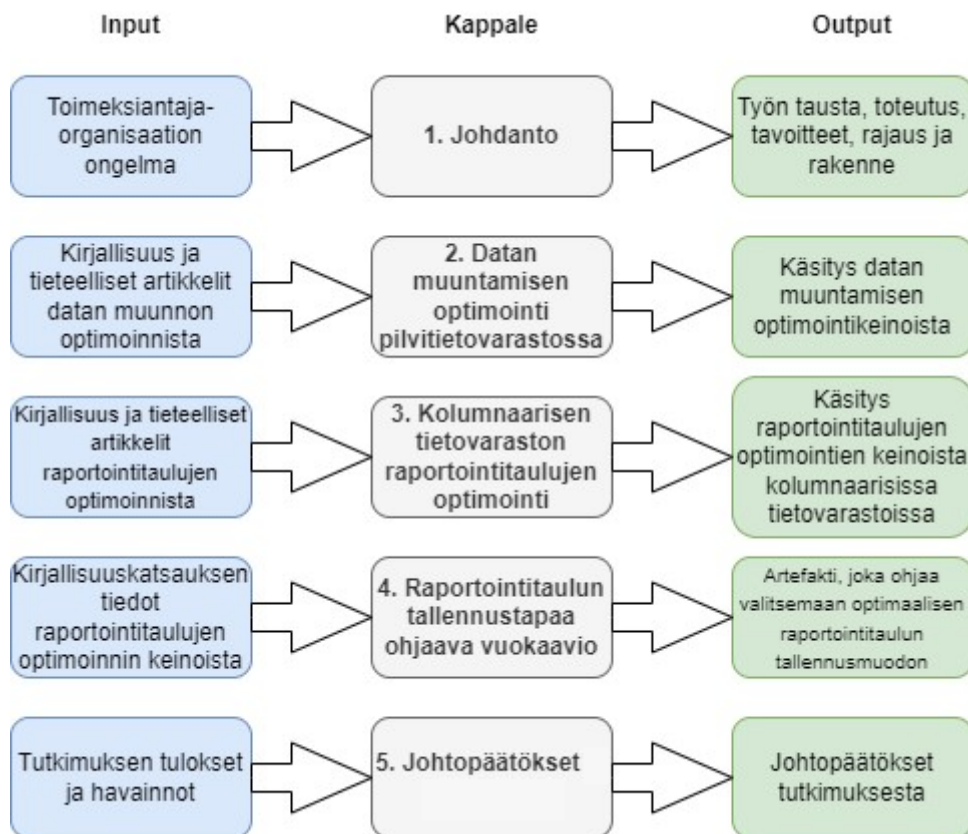
Tiedon etsintään käytettyjä avainsanoja ja -käsitteitä on koottu alla olevaan listaan, joiden lisäksi näitä on yhdistelty ja muunneltu soveltuvien osin. Tutkimuksessa käytettiin lähteinä kirjallisuutta, tieteellisiä artikkeleita sekä verkkosivuja.

- SQL optimization
- Data transformation
- Columnar database
- Data warehouse optimization
- Table clustering
- Table partitioning
- Cloud data warehouse
- Cloud platform costs
- Big data optimization
- Data warehouse architecture
- BI tool components
- Cloud computing
- Modern data warehouse
- Database optimization
- Reporting table optimization

## 1.4 Työn rakenne

Työ koostuu viidestä pääkappaleesta: johdannosta, kahdesta kirjallisuuskatsauksen kappaleesta, empiirisestä tutkimuksesta ja sen tuloksista ja lopuksi johtopäätöksistä. Johdannossa käsitellään työn taustaa, kerrotaan tavoitteet ja rajaukset sekä kuvataan aineisto, menetelmät ja rakenne kokonaisuudessaan. Toisessa pääkappaleessa, eli ”Datan muuntamisen optimointi pilvitietovarastossa”, perehdytään ensin pilvitietovaraston arkkitehtuuriin, jonka jälkeen käsitellään näiden kustannusten muodostumisen periaatetta. Tämän jälkeen syvennyttään yhteen kustannuksiin vaikuttavaan osa-alueeseen, eli datan muuntamiseen, joka käydään läpi työn rajaukset huomioiden.

Kolmas kappale, eli ”Kolumnaarisen tietovaraston raportointitaulujen optimointi” luo pohjan tutkimuksen empiiriselle osuudelle. Kappaleessa luodaan ensin katsaus kolumnaarisiin tietovarastoihin ja niiden toimintaperiaatteisiin. Tämän jälkeen perehdytään raportointitauluihin ja niiden optimoinnin keinoihin. Kirjallisuuskatsausta seuraa kappale, jossa ensin kuvataan DSR tutkimusmenetelmänä, jonka jälkeen kappaleessa toteutetaan menetelmän mukaiset aktiviteetit, joiden lopputuloksena on tutkimuksen artefakti. Viimeisenä kappaleena on ”Johtopäätökset” -kappale, jossa käydään läpi tutkimuksen tuloksia, pohditaan tutkimuksen tavoitteisiin pääsyä ja arvioidaan tutkimuksen onnistumista. Työn rakennetta havainnollistaa kuvassa 2 näkyvä input/output -kaavio työstä.



Kuva 2. Työn input/output -kaavio.

### 1.5 Tekoälyn hyödyntäminen työssä

Tätä työtä tehdessä on hyödynnetty ChatGPT-nimistä tekoälymallia. Palvelua on käytetty apuna yksittäisten sanojen ja lauseiden käänöksissä, lähteiden etsinnässä, työn rakenteen muodostamisessa ja aiheiden ideoinnissa. Mitään ChatGPT:n tuottamaa tekstiä ei ole hyödynnetty sellaisenaan, eikä sen tuottamia käänöksiä ole myöskään käytetty arvioimatta näitä ensin kriittisesti.

## 2 Datan muuntamisen optimointi pilvitietovarastossa

### 2.1 Modernin tietovarastoinnin ja raportoinnin arkkitehtuuri

Aloitetaan katsauksella modernin tietovaraston ja raportoinnin arkkitehtuuriin, joka on toimeksiantajaorganisaation valitsema tietovarastoinnin arkkitehtuuri. Tämä auttaa hahmottamaan pilvialustalla sijaitsevan tietovaraston kaikki osa-alueet ja valmistaa keskittymään niihin osa-alueisiin, joihin työssä tarkemmin syvennyttään. Senna (2024) käsittelee kirjassaan useita eri tietovarastoinnin arkkitehtuureja 2010-luvulta alkaen, joista yhtä hän kutsuu moderniksi tietovarastoksi (engl. The Modern Data Warehouse, MDW). Hän mainitsee, että vaikka modernien tietovarastojen arkkitehtuuri toteutetaan usein kokonaan pilvialustalla, se ei kuitenkaan ole tälle arkkitehtuurille välttämätöntä. Tästä syystä työssäkin puhutaan arkkitehtuurin yhteydessä modernista tietovarastosta ja muutoin pilvitietovarastosta.

Sennan (2024, s. 137–142) mukaan modernin tietovarastoinnin arkkitehtuuri koostuu kahden komponentin yhdistelmästä, joista molemmat voivat olla myös itsenäisiä tietovarastoinnin arkkitehtuureja: tietoaltaista (engl. data lake) ja relaatiotietokannoista. Yksinään pelkän tietoaltaan tai tietokannan käytössä tulee vastaan ongelmia muun muassa skaalautuvuuden, nopeuden tai kustannusten kanssa. Kuitenkin yhdistämällä nämä kaksi, saadaan Sennan mukaan arkkitehtuuriratkaisu, joka on:

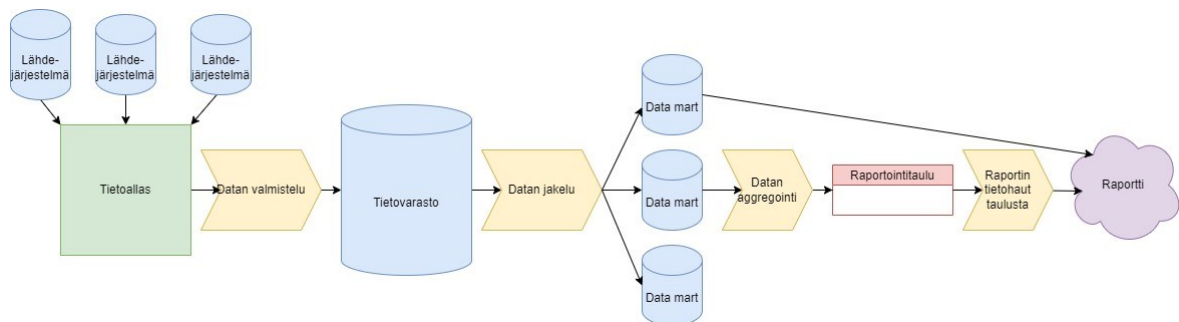
- skaalautuva
- suorituskykyinen
- reaaliaikaista analytiikkaa tukeva
- joustava
- useita datalähteitä tukeva
- ja tietoturvallinen

Santoso (2017, s. 95) täydentää edellä mainittua listaa sillä, että moderneille tietovarastoille on ominaista se, että ne pystyvät käsittelemään strukturoitua, puolistrukturoitua sekä strukturoimatonta dataa. Tämän lisäksi hän mainitsee, että modernit tietovarastot tukevat monien eri tiedostotyyppien, kuten ääni- ja videotiedostojen, käsittelyä.

Tämän arkkitehtuurin potentiaalisiksi haittapuoliksi Senna (2024, s. 137–142) listaa monimutkaisuuden, hinnan suhteessa halvimpiin arkkitehtuureihin, henkilökunnan taitotason

vaatimukset ja datan siiloutumisen. Dibouliya (2023, s. 296–297) täydentää haittapuolia mainitsemalla kulujen optimoinnin haastavuuden sekä jatkuvasti muuttuvan teknologian perässä pysymisen.

Myös Braake (2021) käyttää termiä modernin tietovarastoinnin arkkitehtuuri, lisäten loppuun myös raportoinnin. Alla olevassa kuvassa 3 on Braaken kuvaus modernin tietovarastoinnin ja raportoinnin arkkitehtuurista:



Kuva 3. Modernin tietovarastoinnin ja raportoinnin arkkitehtuuri. (Braake 2021, s. 304)

Senna (2024, s. 137–139) jakaa datan matkan modernin tietovaraston läpi lähdejärjestelmistä raportille viiteen eri vaiheeseen: (datan) hakuun, tallennukseen, muuntamiseen, mallintamiseen ja visualisointiin. Nämä vaiheet ovat nähtävillä myös Braaken kuvailemassa modernin tietovarastoinnin ja raportoinnin arkkitehtuurissa, joskin hieman eri nimillä. Braaken kuvan ensimmäinen vaihe, jossa lähdejärjestelmistä siirtyy dataa tietoaltaan, vastaa Sennan mainitsemaa datan haun vaihetta. Tallennusta vastaavia vaiheita on useita: tietoaltaan tallennetaan lähdejärjestelmien data sen raakassa muodossa, tietovarastoon tallennetaan siivottua dataa ja data marteihin sekä raportointitauluihin tallennetaan liiketoiminnalle ja raportointityökaluille käyttökelpoista dataa.

Datan muuntamisen ja mallintamisen vaiheet osuvat eri tallennusvaiheiden väliin. Datan muuntamisen vaiheella Senna tarkoittaa kuvan ”Datan valmistelu” kohtaa, jossa dataa siivotaan ja valmistellaan ennen tietovarastoon tallennusta. Datan mallintamisella hän taas tarkoittaa datan jakelua ja sen lopputuotteena olevia data marteja. Data mart on tietovaraston osajoukko, joka on tarkoitettu vastaamaan tietyn osaston tai liiketoiminta-alueen tarpeisiin. Sen tarkoitus on tarjota käyttäjille helpommin ymmärrettävä ja saavutettava näkymä tietoihin. Viimeisenä vaiheena on datan visualisointi, joka kattaa kuvan datan aggregoinnin, raportointitaulun ja itse raportin osa-alueet. (Senna 2024, s. 76, 138–139)

Tämä tutkimus keskittyy kuvatun tietovarastoinnin ja raportoinnin arkkitehtuurin loppupäähän alkaen data martista. Luodaan seuraavaksi katsaus pilvitietovarastojen kustannusten muodostumiseen, jotta saadaan käsitys, miten kuluja voidaan madaltaa pilvitietovarastoissa.

## 2.2 Pilvitietovaraston kustannusrakenne

Emergen Researchin (2023) ja Sennan (2024, s. 137) mukaan markkinaosuuksiltaan suurimpiin pilvitietovarastojen tarjoajiin lukeutuvat muun muassa Snowflake, Google, Amazon ja Microsoft. Kahn et al. (2021, s. 599) mukaan pilvipohjaisten palvelujen kustannukset usein muodostuvat käytettyjen resurssien mukaan. Yleistä on, että palveluita voidaan kytkeä päälle ja pois tarpeen mukaan pilviympäristöissä.

Pilvitietovarastointiin liittyvissä hinnoitteluissa on eroavaisuuksia palveluntarjoajien välillä, mutta kaksi pääkomponenttia hinnoittelussa toistuu: datan tallennustilan ja laskentatehon käyttö maksaa aina. Datan tallennustilaa käytetään silloin, kun pilvialustalle tai pilvitietovarastoon tallennetaan tiedostoja tai tauluja. Bellin et al. (2021, s. 131) mukaan datan tallennustila on suhteessa halvempaa verrattaessa laskentatehon hintaan. Datan tallennustilan käytön kustannuksiin vaikuttaa tallennetun tiedon suuruuden lisäksi tallennuspaikan maantieteellinen sijainti. (Amazon Web Services, 2023; Google 2024b; Microsoft, 2024; Snowflake Inc., 2024)

Laskentatehoa käytetään datan käsittelyssä, esimerkiksi laskentatehoa kuluttavista operaatioista mainittakoon SQL-kyselyiden prosessointi, tietokantataulujen muodostukseen ja manipulointiin käytettyjen skriptien prosessointi sekä käyttäjien määrittelemien funktioiden prosessointi. Palveluntarjoajien laskentatehon käytön hinnoittelut ovat vaikeasti verrattavissa keskenään, sillä kaikilla on toisistaan eroavat termit sekä mahdollisesti myös eroavat yksiköt laskentatehon suhteen. Lähes kaikki palveluntarjoajat tarjoavat myös mahdollisuutta ostaa tietty kapasiteetti laskentatehoa ja tallennustilaa etukäteen halvempaan hintaan. Tämä tarjoaa säästämahdollisuuksia, mikäli tuleva käyttö on organisaatiossa hyvin ennustettavissa. (Amazon Web Services, 2023; Google 2024b; Microsoft, 2024; Snowflake Inc., 2024)

Dibouliya (2023, s. 297) mainitsee myös datan tallennustilan ja laskentatehon käytön kuluja aiheuttavina operaatioina, mutta hän jakaa laskentatehon käytön vielä tarkemmalle tasolle:

datan siirtoon liittyviin kuluihin ja datan prosessointiin käytettyihin kuluihin. Näiden lisäksi hän listaa seuraavia pilvitietovarastoinnin kokonaisuuden kuluihin vaikuttavia asioita:

- Infrastruktuurikulut
- Ohjelmistojen lisenssikulut
- Tietohallintoon ja tietoturvaan liittyvät kulut
- Koulutuskulut
- Palveluntarjoajan tuki- ja palvelukulut

Tutkittaessa raportointitaulun optimointia sekä muunto- että tallennusvaiheessa kyselyjen kannalta, on kyse pääosin laskentatehon käytön optimoinnista. Laskentatehoa kuluu raportointitaulun muuntovaiheessa sekä raportointityökalun suorittaessa kyselyjä raportointitauluun. Raportointitaulu vie toki tallennustilaa, mutta suurempi optimointipotentiaali on laskentatehon käytön vähentämisessä. Luodaan seuraavaksi katsaus datan muuntamiseen.

### 2.3 Datan muuntaminen

Reis & Housley (2022, s. 309) mukaan datan muuntaminen antaa datalle sen arvon skaalautuvalla, luotettavalla ja kulutehokkaalla tavalla. Datan muuntamisen tarpeellisuutta he havainnollistavat seuraavalla esimerkillä: kuvitellaan, että jouduttaisiin suorittamaan kysely aina kun halutaan nähdä tuloksia tietystä tietojoukosta. Kyselyä ajettaisiin kymmeniä tai satoja kertoja päivässä. Joka ajolla kysely kävisi läpi parsinnan, puhdistuksen, yhdistämisen, liittämisen ja koostamisen vaiheet. Sen suorittaminen kuluttaisi merkittävästi resursseja ja aiheuttaisi huomattavia pilvipalvelumaksuja. Tämä turhauttaisi käyttäjää hitauden takia ja organisaatiota kulujen takia. Datan muunnon avulla tämän tilanteen saa ratkaistua: kyselyn tulokset voi tallentaa tauluun tai vähintään laskennallisesti intensiivisimmät osat voi suorittaa vain kerran, joka johtaa säästöihin ja nopeampien kyselyjen kautta mukavampaan käyttökokemukseen.

Datan muuntoa tapahtuu muutamassa eri vaiheessa datan matkalla lähdejärjestelmistä raportille, kuten modernin tietovarastoinnin ja raportoinnin arkkitehtuurin kuvasta näkee. Sherman & Imhoffin (2015, s. 12) mukaan, ennen kuin data on käyttökelpoista, sen tulee olla puhdasta, johdonmukaista, yhdenmukaista, oikea-aikaista ja kattavaa. Foxwell (2020, s. 3–8) asettaa teoksessaan myös datalle vaatimuksia, ennen kuin se on käyttökelpoista, mutta hän käyttää hieman eri määritelmää kuin Sherman & Imhoff. Foxwell puhuu ”hyvästä datasta”, joka omaa seuraavat piirteet: tarkkuus, asianmukaisuus, edustavuus, täydellisyys ja

näiden lisäksi datan on oltava hyvin määriteltyä. Datan on hänen mukaansa oltava ”hyvää” tai riskinä on datan pohjalta tehtyjen analyysien epäonnistuminen tai väärin johtopäätösten tekeminen.

Data tulee siis jotenkin muuntaa kriteerit täyttäväksi, ennen kuin sitä voidaan hyödyntää. Sherman & Imhoff (2015, s. 88) jakavat datan muuntamisen kahteen eri vaiheeseen: datan valmisteluun ja datan jakeluun. Datan valmistelun jälkeen puhtauden ja johdonmukaisuuden kriteerit on täytetty, kun taas datan jakeluvaiheen jälkeen kolme muuta kriteeriä, eli yhdenmukaisuus, oikea-aikaisuus ja kattavuus on kuitattu. Densmore (2021, s. 105–106) jakaa datan muuntamisen myös kahteen vaiheeseen, mutta hän käyttää vaiheista eri nimityksiä.

Datan valmisteluun kuuluu sen uudelleenmuotoilu, yhdistäminen ja puhdistaminen. Datan jakelun vaiheeseen kuuluu datan muuntaminen sellaiseen muotoon, jossa se on käyttökelpoista liiketoiminnan henkilöille ja raportointisovelluksille. Jakeluvaiheen muunto-operaatioihin kuuluu loogisen tai fyysisen datamallin mukainen uudelleenjärjestely, liiketoimintametriikoiden laskenta, datan summaus ja -laskuoperaatiot ja lopuksi tallennus data marttiin tai raportointitauluun. (Sherman & Imhoff, s. 88) Yleisiksi haasteiksi edellä mainituissa operaatioissa Foxwell (2020, s. 75–78) listaa puuttuvan datan käsittelyn, datalähteen ymmärtämisen, poikkeavien arvojen tarkoituksen ja käsittelyn sekä virheet datan keräämisen ja tallentamisen vaiheissa.

Sherman & Imhoff (2015, s. 93–94) toteavat, että raportointitaulujen data tulisi muuntaa ja tallentaa mahdollisimman summattuun tai aggregoituun muotoon suorituskyvyn parantamiseksi. Heidän mukaansa eri raportointityökaluissa on eroja sen suhteen, kuinka paljon tämä parantaa suorituskykyä, mutta tämä kuitenkin auttaa aina raportointityökalun suorittamien kyselyiden nopeuteen ja siirrännän vähentämiseen, kun data on mahdollisimman aggregoidussa muodossa.

Datan muuntamiseen on tarjolla useita työkaluja, mutta Reis & Housley (2022, s. 313) jakavat nämä kahteen kategoriaan: SQL-pohjaisiin muuntoihin ja muihin koodipohjaisiin muuntoihin. Heidän mukaansa näistä kahdesta ensimmäinen, eli SQL-pohjaiset muunnot ovat yleistyneet viime aikoina, johtuen yleisesti niiden soveltuvuudesta suurien datamassojen käsittelyyn. SQL-pohjaisiin muuntoihin lukevat myös sellaiset työkalut, joissa muunto-operaatiot kirjoitetaan SQL:ää hyödyntäen. Myös Densmore (2021, s. 105–106) mainitsee SQL:n olevan suosittu muuntamisen kieli siitä syystä, että data-analyttikot ja analytiikkainsinöörit

ovat aiempaa enemmän vastuussa datan muuntamisesta ja SQL on usein kielenä heille entuudestaan tuttu. Luodaan seuraavaksi katsaus SQL-kielen optimointiin.

## 2.4 SQL ja sen optimointi

SQL on laajasti käytössä oleva tietokantojen yhteydessä käytettävä kieli. Lahtonen (2002, s. 37) jakaa SQL:n kahteen osaan: tietokannan rakenteen määrittelyyn (Data Definition Language, DDL) ja tietokannan sisällön käsittelyyn (Data Manipulation Language, DML). SQL luokitellaan deklarattiiviseksi ohjelmointikieleksi, eli sillä määritellään haluttu tulos, mutta ei askelia sen saavuttamiseksi. SQL:n avulla ei siis voi kirjoittaa täysin toimivia sovelluksia. Tietokannoissa näistä niin kutsutuista askelista päättää tietokannan moottori, jota kutsutaan myös optimoijaksi. Optimoijan työ on lukea sille annetut SQL-lausekkeet ja määritellä tehokkain suorituspolku kyseiselle lausekkeelle. Suorituspolkuihin vaikuttaa taulun määritelty rakenne. Optimoijan kyvykkyydet ovat kuitenkin rajalliset ja se kykenee vain järjestelemään uudelleen sille annetun SQL-lausekkeen järjestystä, joten käyttäjän vastuulle jää kuitenkin suurin osa SQL:n optimoinnista. (Lahtonen 2002, Beaulieu 2020)

Ennen katsausta SQL:n optimointiin, esitellään vielä SQL:n tietokannan sisällön käsittelyyn liittyvät lausekkeet ja katsotaan niiden yleinen suoritusjärjestys. Usein SELECT-lausekkin sanottu kysely jaetaan Beaulieun (2020, s. 60) ja Lahtosen (2002, s. 60) toimesta kuuteen päälausekkeeseen, jotka on kuvattu alla olevassa taulukossa:

Taulukko 1. SELECT-lauseen alalausekkeet. (Beaulieu 2020, s. 60)

Lausekkeen nimi	Tarkoitus
SELECT	Määrittelee, mitkä sarakkeet sisällytetään kyselyn tuloksiin
FROM	Määrittelee taulut, joista data haetaan ja kuinka taulut liitetään toisiinsa
WHERE	Suodattaa pois datan, jota ei haluta tuloksiin
GROUP BY	Käytetään rivien ryhmittelyssä
HAVING	Suodattaa pois ryhmät, joita ei haluta tuloksiin
ORDER BY	Järjestää rivit yhden tai useamman sarakkeen arvojen mukaan

Kaikki taulukossa 1 olevat lausekkeet ovat standardoituja. Vain SELECT-lauseke on pakollinen kyselyn suorittamiseen, mutta normaalissa kyselyssä on noin 2–4 lauseketta. Celko (2010, s. 398) nostaa vielä WITH-lausekkeen, jolla määritellään Common Table Expressioinit (CTE:t), mukaan SELECT-lausekkeeseen kuuluvaksi osaksi. Common Table Expressioinia Celko kuvailee tauluksi tai näkymäksi kyselyn sisällä, johon voi viitata. Vaikka SELECT-lauseke kirjoitetaan taulukossa 1 olevassa järjestyksessä, niin tietokanta ei silti suorita kyselyä sille kirjoitetussa järjestyksessä. Suoritusjärjestys on Celkon mukaan seuraava:

1. WITH-lausekkeessa määritellyt, CTE:n muodostavat kyselyt
2. FROM-lauseke, joka määrittelee käytettävät taulut
3. WHERE-lauseke, joka suodattaa tauluja
4. GROUP BY-lauseke, joka ryhmittelee taulun rivejä
5. HAVING-lauseke, joka suodattaa ryhmiteltyjä rivejä
6. SELECT-lauseke, joka suodattaa pois sarakkeita
7. ORDER BY-lauseke, joka halutessa järjestää rivit järjestykseen

Hayath, Usman, Shafiulla & Dadapeerin (2023, s. 1–3) mukaan SQL:n optimointiin on monia eri keinoja ja SQL-lauseen voi kirjoittaa hyvin monella eri tavalla. Optimointia voidaan suorittaa sekä DDL-lauseilla, että DML-lauseilla. Toisin sanoen osa optimoinnista tapahtuu tietokannan tai -varaston taulujen luonnin ja fyysisen tallennuksen yhteydessä ja osa näitä tauluja kyseltäessä. DML-lauseiden optimoinnin keinoihin vaikuttaa myös oleellisesti se, että miten data on tallennettu. Samaa jakoa optimoinnin osalta käyttää Myalapalli & Savarapu (2014, s. 1), mutta he käyttävät DDL-lauseista käsitettä tietokantojen optimointi.

Hayathin et al. (2023, s. 4) mukaan SQL:n optimoinnilla voidaan saavuttaa parempi tehokkuus, madaltaa kuluja sekä saavuttaa tyytyväisemmät raportoinnin asiakkaat. Tyytyväisemmät asiakkaat ovat seurausta nopeammin suoriutuvista kyselyistä, kun taas kulueta saadaan vähentyneen laskentatehon käytöstä. Hayath et al. mainitsevat SQL:n suorituskyvyn optimoinnin pohjautuvan usein siihen, että pyritään vähentämään resurssien määrää, joilla haetaan tarvittava data. Tehokkain tapa heidän mukaansa on se, joka käsittelee vähimmän määrän rivejä ja sarakkeita. Käsiteltävän datan määrää voi vähentää kirjoittamalla parhaita käytäntöjä noudattavia SQL-lauseita, joita esitellään alla.

SQL:n optimointi on jo laajasti tutkittu alue ja aiheesta löytyy paljon tieteellisiä artikkeleita ja kirjallisuutta. Näiden artikkeleiden ja kirjallisuuden optimointikeinoja on pyritty

listaamaan alla olevaan taulukkoon 2 mahdollisimman monipuolisesti. Taulukkoon on poimittu erityisesti DML-lauseiden ja taulun lukuun liittyviä optimointikeinoja.

Taulukko 2. SQL-lauseiden optimointikeinoja.

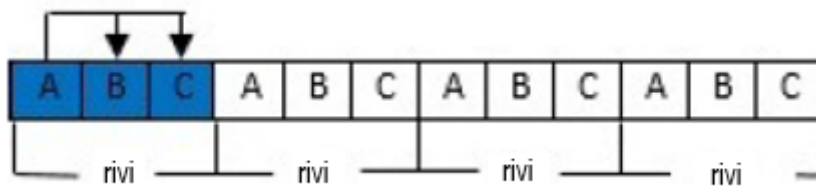
SQL:n optimointikeino	Lähde
Käytä REGEXP_EXTRACT:ia CASE WHEN LIKE:n sijaan	Hayath et al. 2023, s. 3
Konvertoi pitkät IN-listaukset väliaikaisiksi tauluiksi ja käytä niitä	Hayath et al. 2023, s. 3
Järjestä JOIN:it aina suurimmasta taulusta pienimpään	Hayath et al. 2023, s. 4
Käytä MAX:ia RANK:in sijaan	Hayath et al. 2023, s. 4
Käytä REGEXP_LIKE:a LIKE:n sijaan	Hayath et al. 2023, s. 3
Vältä koko taulun skannausta lisäämällä WHERE-ehdoja	Myalapalli & Savarapu 2014 s. 2
Rajaa kyselyjä taulun ositusten mukaan	Myalapalli & Savarapu 2014 s. 4
Vältä HAVING-lauseketta kirjoittamalla kysely uudelleen WITH-lauseketta hyväksi käyttäen	Myalapalli & Savarapu 2014 s. 4
Käytä UNION ALL:ia UNION:in sijaan	Myalapalli & Chakravarthy 2016 s. 2
Vältä useita samanlaisia alakyselyitä	Myalapalli & Chakravarthy 2016 s. 2
Järjestä rivit vain kerran kyselyssä, jos mahdollista	Myalapalli & Chakravarthy 2016 s. 2
Järjestä rivit vasta kyselyn lopussa	Myalapalli & Chakravarthy 2016 s. 2
Käytä vertailuoperaattoreista aina mieluummin ”=” kuin ”!=”, sillä jälkimmäinen johtaa koko taulun skannaukseen	Myalapalli & Chakravarthy 2016 s. 2
Käytä DISTINCT:ia UNION:in sijaan	Celko 2010 s. 748
Vältä CROSS JOIN:eja	Celko 2010 s. 754
Tarvittavien attribuuttien valinta SELECT *:n sijaan	Visweswara, Narechania & Arulraj 2020 s. 4
Vältä RAND-funktiota	Visweswara, Narechania & Arulraj 2020 s. 4
Vältä DISTINCT:ia JOIN:in duplikaattien poiston yhteydessä	Visweswara, Narechania & Arulraj 2020 s. 4
Vältä liian monia JOIN-operaatioita	Visweswara, Narechania & Arulraj 2020 s. 4

Useat pilvitietovarastot hyödyntävät hajautettua tietojenkäsittelyä ja näille on olemassa erityinen JOIN-operaatioon liittyvä optimointikeino. Luu (2018, s. 176–177), Reis & Housley (2022, s. 310) ja Thallam (2020b) mainitsevat, että aina kun mahdollista, tulisi pyrkiä suorittamaan niin sanottu broadcast join, jossa suurimpaan tauluun liitetään ensin pienin taulu, jonka jälkeen laskevassa järjestyksessä loput taulut. Tällöin hajautettua tietojenkäsittelyä hyödyntävien järjestelmien kyselyn käsittelijät joutuvat jakamaan toistensa kesken vähemmän tietoa, tuoden suorituskyky- ja resurssien käyttöhyötyjä. Sellaiset SQL-lauseet, joissa on useita JOIN-operaatioita, tulee siis kirjoittaa siten, että aloitetaan suurimmasta taulusta, jonka jälkeen siihen liitetään pienin taulu. Tämän jälkeen JOIN-operaatiot suoritetaan kooltaan laskevassa järjestyksessä.

### 3 Kolumnaarisen tietovaraston raportointitaulujen optimointi

#### 3.1 Kolumnaariset tietovarastot

Sekä Abadi, Boncz & Harizopoulos (2009, s. 1664) että Wu (2015, s. 1) jakavat tietokannat ja -varastot kahteen kategoriaan sen mukaan, miten ne tallentavat datansa: rivipohjaisiin ja sarakepohjaisiin. Tässä kappaleessa esitellään molempien toimintaperiaatteet, jonka jälkeen perehdytään enemmän sarakepohjaisiin eli kolumnaarisiin tietovarastoihin. Perinteiset tietovarastot ovat yleensä rivipohjaisia ja ne tallentavat dataa rivi kerrallaan. Uudet rivit lisätään yleensä taulun loppupäähän. Taulu saattaisi näyttää levyllä tallennettuna tältä:



Kuva 4. Rivipohjaisen tietovaraston taulun tallennus levyllä. (Morales-Morales et al. 2019, s. 49)

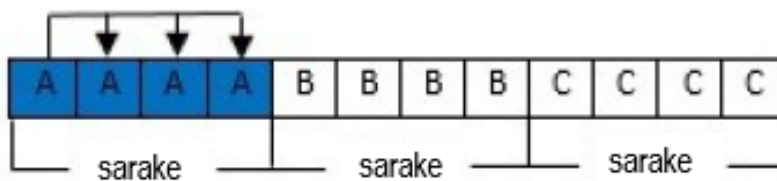
Etuna rivipohjaisessa datan tallennuksessa on se, että uuden rivin lisääminen on helppoa. Ei tarvitse tietää, montako riviä taulussa on jo valmiiksi: voidaan vain hypätä taulun loppuun ja lisätä rivi sinne tai luoda uusi tiedosto ja yhdistää vanha ja uusi ilman ongelmia. Tehottomuuksiin rivipohjaisessa varastoinnissa törmätään kuitenkin silloin, kun halutaan lukea dataa analyttisiin tarkoituksiin. Havainnollistaaksemme tätä tehottomuutta, tarkastellaan seuraavaa Mucchetin (2020, s. 14) käyttämää esimerkkiä ja siihen liittyvää taulua:

Taulukko 3. Esimerkkidata tallennettuna rivipohjaisesti. (Mucchetti 2020, s. 14–15)

Nimi	Väri	Muoto
Omena	Punainen	Pyöreä
Appelsiini	Oranssi	Pyöreä
Banaani	Keltainen	Käyrä

Esimerkkidatan värit pyrkivät havainnollistamaan tallennustapaa, joka myös yllä olevassa kuvassa 4 kuvattiin. Jos haluttaisiin tietää, mitkä kaikki hedelmät ovat pyöreitä, tulisi tietovaraston lukea koko taulu, pysähtyen jokaisessa ”muoto”-sarakeessa, tarkistaa sen arvo ja päätellä, tuleeko rivi ottaa mukaan vai ei. Tämä on luonnollisesti tehotonta siitä syystä, että koko taulu pitää lukea. Perinteisissä tietovarastoissa on joitakin keinoja, esimerkiksi indeksointi, jolla tätä ongelmaa saadaan pienennettyä, mutta esimerkki havainnollistaa rivipohjaisen tietovarastoinnin ongelmat selkeästi.

Kolumnaarisen tietovaraston taulun tallennus levyllä taas näyttää tältä Morales-Morales et al. (2019, s.49) mukaan tältä:



Kuva 5. Kolumnaarisen tietovaraston taulun tallennus levyllä. (Morales-Morales et al. 2019, s.49)

Äskeinen Muchetin (2020, s. 15) esimerkkidata olisi kolumnaarisessa tietovarastossa tallennettu seuraavalla tavalla:

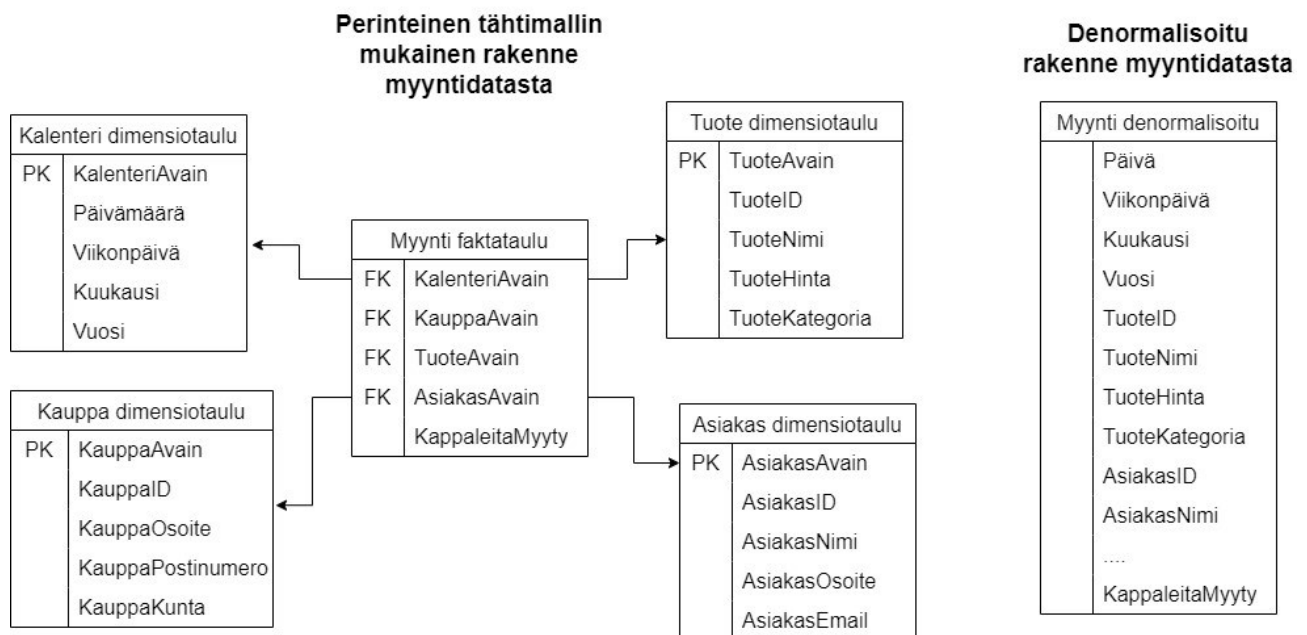
Taulukko 4. Esimerkkidata tallennettuna kolumnaarisesti. (Mucchetti 2020, s. 14–15)

Nimi	Väri	Muoto
Omena	Punainen	Pyöreä
Appelsiini	Oranssi	Pyöreä
Banaani	Keltainen	Käyrä

Jatkaen Muchetin esimerkkiä, nyt jos haluttaisiin jälleen löytää kaikki pyöreät hedelmät, voitaisiin ladata ainoastaan nimi- ja muotosarake ja yhdistää näiden tiedot. Värisaraketta ei koskaan tarvitse ladata, tai mitään muitakaan mahdollisia kymmeniä tai satoja sarakkeita, kuten rivipohjaisessa varastoinnissa. Kolumnaarisessa varastoinnissa törmätään kuitenkin päinvastaiseen ongelmaan: kun halutaan lisätä uusi rivi tauluun, täytyy jokainen taulun sarake avata ja kirjoittaa sinne uusi arvo loppuun.

Wu (2015, s. 1) ja Bhagat & Gopal (2012, s. 195) listaavat kolumnaaristen tietokantojen ja -varastojen eduksi verrattuna rivipohjaisiin tietokantoihin paremman kyselytehokkuuden, vähemmän tallennustilan käytön ja tehokkaammat lukuoperaatiot kun on kyse analyttisistä toimenpiteistä. Näiksi toimenpiteiksi hän listaa tietovarastoinnin käytön, päätöksentekojärjestelmät ja BI-sovellukset. Näitä listattujen etujen väitettä tukee myös Morales-Moralesin et al. (2019) tekemä tutkimus, jossa vertailtiin sekä rivipohjaisen tietokannan, että kolumnaarisen tietokannan suorituskykyä nopeuden kannalta. Kolumnaariset tietokannat suoriutuivat huomattavasti nopeammin monista eri kyselyistä.

Sekä Abadin et al. (2009, s. 1664) että Wun (2015, s. 1) mukaan kolumnaariset tietokannat ja -varastot osittavat taulunsa vertikaalisesti, jakaen kaikki sarakkeet yksittäisiin osiin, jotka tallennetaan erikseen. Tämä on kuitenkin vain datan fyysiseen tallennukseen liittyvä toimenpide, sillä tietokantatauluja kyseltäessä kolumnaarisen tietokannan taulut näyttävät tismalleen samalta kuin rivipohjaisen tietokannan. Jukic et al. (2017, s. 67) toteavat myös, että kolumnaaristen tietokantojen taulujen vertikaalinen ositus mahdollistaa denormalisoidun tietorakenteen menettämättä suorituskykyä. Kuva 6 esittää saman myyntidatan rakenteen perinteisessä tähtimallisissa sekä denormalisoidussa muodossa, jota kutsutaan myös yhdeksi isoksi tauluksi (one big table) tai flatfileksi (tasaiseksi tiedostoksi).



Kuva 6. Myyntidata tähtimallinnettuna ja denormalisoituna. (Jukic et al. 2017, s. 64 & 68)

Jukicin et al. (2017, s. 67) mukaan denormalisoidun rakenteen käyttö on mahdollista suorituskykyä menettämättä siitä syystä, että taulusta voidaan valita aina vain tarvittavat

sarakkeet ja kysellä niiden rivit, toisin kuin rivipohjaisessa tietokannassa. Heidän mukaansa datan analysointi ja siitä liiketoimintaan vaikuttavien päätösten tekeminen on helpompaa ja nopeammin saatavilla, kun data on valmiiksi yhdistetty jo yhteen paikkaan. Myös liiketoiminnan henkilöiden on helpompi käyttää tämän rakenteen mukaista taulua, sillä datan kysely on huomattavasti helpompaa yhdestä paikasta verrattuna usean taulun yhdistämiseen.

Jukicin et al. (2017, s. 78) tutkimuksen mukaan raportointisovellukset toimivat nopeammin, kun raportit rakennetaan denormalisoidun mallin pohjalta. Tutkimuksessaan he loivat yrityksen raportit sekä tähtimallin että denormalisoidun mallin avulla, ja denormalisoitua taulua käyttäneet raportit toimivat 8–25 kertaa nopeammin kuin tähtimallia käyttäneet.

### 3.2 Tietovarastojen raportointitaulut

Raportointitaululla tarkoitetaan tässä työssä sitä tietovarastossa sijaitsevaa objektia, johon raportointityökalut yhdistetään tai sitä objektia, johon raportointityökalu kohdistaa ajamansa kyselyt. Näitä on muutamia erilaisia: tauluja, näkymiä, materialisoituja näkymiä ja useissa pilvitietovarastoissa myös ulkoisia tauluja.

Mucchetti (2020, s. 9) kuvailee tauluja tietokantojen pääkäsitteeksi. Tietokannoissa ja -varastoissa asuva data järjestetään sarakkeisiin ja riveihin. Sarakkeet kuvaavat ominaisuuksia tai attribuutteja ja rivit edustavat kaikkien ominaisuuksien tietuetta. Harringtonin (2009, s. 5) mukaan tietokantojen ja -varastojen on myös pidettävä sisällään tieto siitä, miten taulut ja niissä oleva data liittyvät toisiinsa.

Näkymä on Beulieu (2020, s. 187) mukaan mekanismi datan kyselemiseksi. Näkymiä ei tallenneta fyysisesti levyille, kuten tauluille tehdään. Näkymät ovat tallennettuja SELECT-lauseita, joille muutkin käyttäjät voivat suorittaa tietokantakyselyjä kuten tauluillekin. Käyttäjät eivät välttämättä edes tiedosta käyttävänsä näkymää, sillä niiden käyttö on samanlaista kuin taulujenkin. Näkymää kyseltäessä taustalla kuitenkin suoritetaan näkymään tallennettu kysely, sillä näkymiä ei tallenneta fyysisesti levyille.

Beulieu (2020, s. 189–191) listaa näkymien käyttötarkoituksiksi datan turvallisuuden, aggregoinnin, monimutkaisuuden piilottamisen sekä ositetun datan yhdistämisen. Datan turvallisuudella tarkoitetaan tässä yhteydessä sitä, että näkymillä voi piilottaa osan sen taustalla olevien taulujen datasta. Esimerkiksi Asiakas-taulu kokonaisuudessaan saattaa sisältää

luottamuksellista dataa, mutta nämä tietyt sarakkeet voidaan rajata pois näkymän avulla. Käyttäjälle voidaan sen jälkeen myöntää pääsy näkymään, mutta ei sen taustalla oleviin tietokantatauluihin. Näkymillä voidaan myös aggregoida dataa esimerkiksi raportointisovelluksia varten, jotka usein haluavat datansa mahdollisimman suppeassa muodossa. Niillä voi myös piilottaa monimutkaiset ja pitkät SQL-lausekkeet, jotta datan käyttö säilyy helppona. Ositetun datan yhdistämisellä tarkoitetaan esimerkiksi vanhan järjestelmän ja uuden järjestelmän datojen yhdistämistä näkymän avulla.

Lightstone, Nadeau & Teoreyn (2010, s. 72) mukaan materialisoidut näkymät ovat näkymiä, joiden tulokset tallennetaan levyille. Mikäli näkymä kyselee taustalla useita suuria tauluja ja aiheuttaa näin ollen paljon siirrantää, kannattaa se heidän mukaansa tallentaa ennemmin materialisoituun näkymään. Materialisoidut näkymät ottavat tilannekuvan tiettyä ajanhetkenä taustalla olevan kyselyn tuloksista ja tallentavat sen levyille. Tietojen fyysinen tallennus levyille vähentää tietojen siirrantää ja näin ollen nopeuttaa materialisoidulle näkymille tehtyjä kyselyitä. Materialisoituja näkymiä tulee kuitenkin päivittää aina, kun taustalla olevat taulutkin päivittyvät.

Ulkoiset taulut ovat Mortonin (2022, s. 20) mukaan tauluja, joiden tieto ei ole tallennettu fyysisesti tietovarastoon. Usein tämä tallennuspaikka on tietoaallas, joita on useilla pilvialustoilla käytössä. Ulkoiset taulut ovat hyödyllisiä, kun datan tarpeellisuutta vielä arvioidaan. Niiden avulla voidaan tutkia dataa helposti jalostamatta tätä ensin tietovarastoon asti. Mikäli data osoittautuu tutkimusten jälkeen käyttökelpoiseksi, voidaan data ottaa mukaan tietovaraston muuntoputkeen.


### 3.3 Raportointitaulujen optimoinnin keinot

Sekä Vaisman & Zimányi (2022, s. 13–36) että Lightstone et al. (2010, s. 5–7) jakavat tietokannan ja -varaston suunnittelun ja mallinnuksen kolmeen eri vaiheeseen: käsitteellisen, loogiseen ja fyysisen mallin vaiheisiin. Käsitteellisessä mallissa pyritään hahmottamaan kaikki tietokantaan kuuluvat asiat ja mahdollisesti myös se, miten ne liittyvät toisiinsa. Käsitteelliseen malliin ei kuulu mitään tulevaan käytännön toteutukseen liittyvää, vaan se auttaa kuvaamaan kaikki tietokantaan kuuluvat asiat ja objektit. Looginen malli on käsitteellisestä mallista seuraava mallinnuksen aste, jossa käsitteellinen malli muunnetaan sellaiseen muotoon, että tietokantaan tulevat taulut ja niiden attribuutit kuvataan tarkemmalla tasolla ja

tauluille hahmotellaan pää- ja vierasavaimet, millä tauluja voi liittää toisiinsa. Fyysinen malli on tarkoin mallinnuksen taso, jossa kuvataan tarkasti, kuinka tieto tallennetaan tietokantaan. Taulujen tietotyypit määritellään, attribuutit nimetään ja niille asetetaan tietyt rajoitteet (kuten ”arvon oltava uniikki”, ”ei voi olla tyhjä” tarvittaessa). Fyysiseen mallinnukseen liittyviä käsitteitä on fyysinen datan järjestely ja tallennus, kyselyjen tehokkuus ja tärkeimpänä tätä työtä ajatellen: optimointi, joka käsittää myös raportointitaulujen optimoinnin.

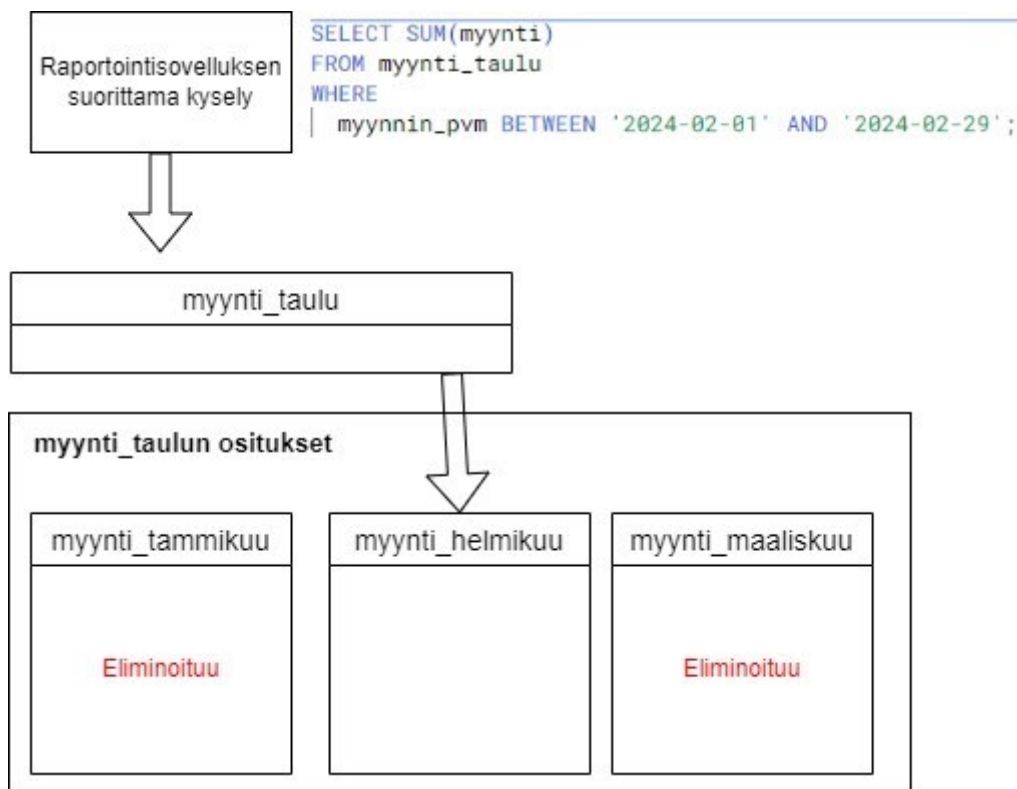
Vaisman & Zimányi (2022, s. 245) listaavat kolme yleistä fyysiseen mallinnukseen liittyvää optimointikeinoa raportointitauluille: materialisoidut näkymät, indeksoinnin ja osituksen. Lightstone et al. (2010, s. 7) mainitsevat näiden lisäksi klusteroinnin optimointikeinona. Näistä mainituista keinoista indeksointi ei kuitenkaan ole kolumnaarisen tietovaraston optimointikeino, johtuen tässä yhteydessä tarkoitettujen indeksien rivipohjaisuudesta. Tämän lisäksi Reis & Housley (2022, s. 324) luonnehtivat materialisoituja näkymiä enemmänkin myöhäiseksi transformaatiovaiheeksi kuin optimointikeinoksi, sillä materialisoituihin näkymiin joudutaan yleensä kääntymään silloin, kun huomataan joidenkin näkymien aiheuttavan runsaasti siirräntää. Toisin sanoen raportointitauluja suunniteltaessa materialisoidut näkymät eivät ole koskaan suoraan vaihtoehtolistalla. Jäljelle jää siis ositus ja klusterointi, joita käsitellään seuraavaksi.

Lightstonen et al. (2010, s. 10, 126) mukaan ositus on fyysisessä mallinnuksessa tapa vähentää yksittäisen laitteen, kuten kovalevyn, työkuormaa osittamalla data monelle laitteelle. Osituksessa yksi taulu jaetaan useaan pienempään samarakenteiseen tauluun. Näitä pienempiä tauluja kutsutaan taulun osiksi tai fragmenteiksi. Datat voidaan osittaa jonkin sarakkeen arvon perusteella. Tällöin kaikki tietyn arvon omaava data tallennetaan samaan paikkaan, kun taas seuraavan arvon saavat datat tallennetaan seuraavaan paikkaan. Datojen ollessa ositettuna tietyn arvon perusteella, voidaan vain tarvittava osa prosessoida, tuoden suorituskykyhyötyjä. Datan voi osittaa periaatteessa minkä tahansa arvon pohjalta, mutta useimmiten osituksessa käytetty arvo on päivämäärä, sillä näistä tulee pääpiirteittäin samankokoisia taulun osia. Kuva 7 esittää kuvitteellisen myynti-aulun sekä ositettuna että ilman ositusta:

myynti_taulu		Ositus kuukauden pohjalta	myynti_taulu_ositettu	
päivä	myynti		päivä	myynti
1.1.2024	100 000		Taulun osa, tammikuu	1.1.2024 100 000
1.3.2024	150 000			15.1.2024 110 000
1.2.2024	120 000		Taulun osa, helmikuu	1.2.2024 120 000
15.1.2024	110 000			15.2.2024 130 000
29.2.2024	140 000			29.2.2024 140 000
15.3.2024	160 000		Taulun osa, maaliskuu	1.3.2024 150 000
15.2.2024	130 000			15.3.2024 160 000

Kuva 7. Myynti\_taulu ilman ositusta sekä ositettuna. (Thallam 2020a)

Lightstone et al. (2010 s. 127, 135–142) listaavat osittamisen hyödyiksi suorituskykyhyödyt, nopeamman datan lisäyksen ja poiston, paremman hallittavuuden sekä mahdollisuuden tal-  
lentaa suurempia tauluja. He suosittelevat ositusta käytettävän silloin, kun kyseessä on suu-  
ret taulut, joita päivitetään tietyn arvojoukon (esimerkiksi päivän) pohjalta. Ositus on heidän  
mukaansa hyödyllistä myös silloin, kun taulua käytetään BI-tyylisissä kyselyissä (esim. ra-  
portointisovellusten tekemät kyselyt), jolloin suorituskykyhyötyjä saadaan erityisesti ositus-  
ten eliminoinnin kautta. Ositusten eliminoinnilla tarkoitetaan sitä, kun SQL-kyselyn suorit-  
tava optimoija kykenee ositetun sarakkeen perusteella hakemaan oikealla tallennussijainnilla  
olevat taulun osat ja jättämään huomioita tai eliminoidaan muut ositukset. Alla oleva kuva  
8 havainnollistaa ositusten eliminointia, kun oikein muodostetulla SQL-kyselyllä haetaan  
tietoja ositetusta taulusta:



Kuva 8. Ositettu taulu ja ositusten eliminointi. (Lightstone et al. 2010, s. 127)

Lightstone et al. (2010 s. 141–142) suosittelevat ositusten koon olevan vähintään 50 Mt per osa, parhaan koon ollessa kuitenkin muutaman Gt:n kokoinen ositus. Ositusten määrän ylärajaksi he suosittelevat 500 ositusta per taulu.

Lightstone et al. (2010, s. 10, 145 & 166) kuvailevat klusterointia keinoksi, jolla voi ryhmitellä dataa dimensioittain, kuten paikan, aikajakson tai tuotetyypin perusteella. Klusterointi jakaa datat omiin fyysisiin tallennuslohkoihin ja sen suorituskykyhyödyt pohjautuvat datan siirrän vähentämiseen. Klusterointia voi tehdä yhden tai usean attribuutin pohjalta ja se on tarkoitettu klusteroitavaksi niiden attribuuttien tai dimensioiden mukaan, joita käytetään kyselyissä paljon. Heidän mukaansa klusterointia kannattaa aina harkita, kun nämä usein käytetyt attribuutit on tunnistettu. Thallam (2020a) täsmentää tätä vielä suosittelemalla klusterointia sellaisten attribuuttien pohjalta, joita käytetään suodattimina kyselyissä tai jos attribuutin pohjalta aggregoidaan dataa. Hänen mukaansa klusteroidun attribuutin pohjalta suodattaminen vähentää siirrantää ja muutoin tämä nopeuttaa kyselyn toimintaa. Lightstone et al. (2010, s. 145 & 166) mainitsevat, että mikäli tietovarastoon tehdään usein hakuja päivämäärärajoja käyttäen, voisi olla hyödyllistä klusteroida data päivämäärän tai kuukauden perusteella, kuten kuvassa 9 on tehty jälkimmäisen datan kohdalla.

Tallennuslohkot klusteroimattomana		Tallennuslohkot klusteroituna kuukauden mukaan	
Tallennuslohko 1	Tallennuslohko 2	Tallennuslohko 1	Tallennuslohko 2
Tietue 1.1.2024	Tietue 30.1.2024	Tietue 1.1.2024	Tietue 4.2.2024
Tietue 4.2.2024	Tietue 11.1.2024	Tietue 9.1.2024	Tietue 20.2.2024
Tietue 9.1.2024	Tietue 15.2.2024	Tietue 4.1.2024	Tietue 8.2.2024
Tietue 4.1.2024	Tietue 19.1.2024	Tietue 30.1.2024	Tietue 15.2.2024
Tietue 20.2.2024	Tietue 1.2.2024	Tietue 11.1.2024	Tietue 1.2.2024
Tietue 8.2.2024	Tietue 28.2.2024	Tietue 19.1.2024	Tietue 28.2.2024

Kuva 9. Datan fyysinen klusterointi. (Lightstone et al. 2010, s. 146)

Kuvasta 9 voidaan tehdä kolme tärkeää havaintoa:

1. Kyseltäessä tammikuun dataa, tarvitsee tietovaraston hakea tiedot vain yhdestä tallennuslohkosta, mikäli haku tehdään klusteroidulle datalle. Mikäli data olisi klusteroimatonta, molemmat tallennuslohkot tulisi skannata. Klusteroidun datan siirranta on siis puolet klusteroimattoman datan siirrännästä tässä esimerkissä.
2. Data klusterin sisällä ei ole järjestyksessä.
3. Kyseltäessä kaikkea dataa, joudutaan silti hakemaan kaikista tallennuslohkoista tietoja, joten datan siirrännän osalta ei saada hyötyjä näissä tapauksissa.

Costa, Costa & Santos (2019, s. 34–36) tutkivat osituksen, klusteroinnin ja näiden yhdistämisen hyötyjä suorituskykyyn sekä tähtimallinnetuille tauluille että denormalisoiduille tauluille. Johtopäätöksissään he listasivat seuraavia huomioita:

1. Denormalisoidut mallit olivat suorituskyvyltään parempia verrattuna tähtimallinnettuun dataan
2. Osituksesta saatiin useammin ja suurempia suorituskykyhyötyjä verrattuna klusterointiin
3. Ositusta ja klusterointia harkitessa on tärkeää tutustua datan attribuuttien kardinaliteettiin, jotta tunnistetaan sopivat attribuutit ositukseen ja klusterointiin
  - a. Ositukseen sopivat matalan kardinaliteetin ja tasaisesti riveiltään jakautuneet attribuutit (useimmiten päivämäärät)
  - b. Klusterointiin sopii korkean kardinaliteetin omaavat attribuutit
4. Taulua ei tule yliosittaa tai klusteroida, sillä tällöin suorituskykyhyödyt mitätöityvät
5. Klusterointia kannattaa erityisesti harkita silloin, kun attribuutti esiintyy ”GROUP BY” tai ”ORDER BY” -operaatioissa

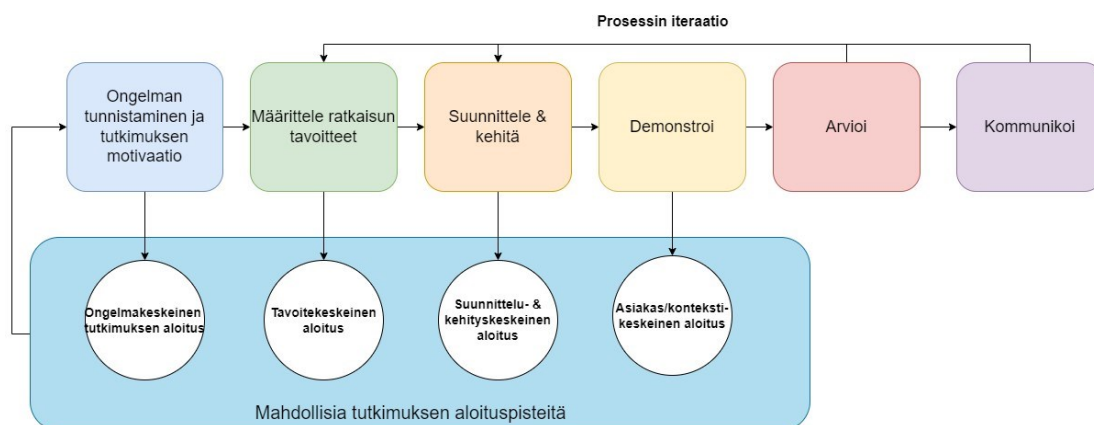
Kardinaliteetilla viitataan attribuutin uniikkien arvojen määrään. Sukupuoli-attribuutti on matalakardinaliteettinen, sillä uniikkeja arvoja on vain kaksi. Esimerkki korkeakardinaliteetisesta attribuutista on postinumero, jonka uniikkeja arvoja on satoja.

## 4 Raportointitaulun tallennustapaa ohjaava vuokaavio

### 4.1 Design Science Research tutkimusmenetelmänä

Peffer, Tuunanen, Rothenberger & Chatterjee (2007, s. 49) määrittelevät Design Science Researchin (myöhemmin DSR) seuraavalla tavalla: DSR luo ja arvioi artefakteja, jotka on tarkoitettu ratkaisemaan tunnistettuja organisaation ongelmia. Se noudattaa tiettyä prosessia havaittujen ongelmien ratkaisemiseksi artefaktien suunnittelussa, niiden arvioinnissa sekä tulosten viestinnässä. Nämä artefaktit voivat olla konstruktioita, malleja, menetelmiä ja erilaisia ilmentymiä. Ne voivat myös sisältää sosiaalisia innovaatioita tai uusia teknisiä, sosiaalisia tai informatiivisia resurssien ominaisuuksia. Lyhyesti sanottuna tämä määritelmä sisältää minkä tahansa suunnitellun objektin, jossa on sisäänrakennettu ratkaisu ymmärrettyyn tutkimusongelmaan. DSR on heidän mukaansa vasta muutamia vuosikymmeniä vanha tutkimusmenetelmä ja näin ollen kehittyvä vielä jatkuvasti. Saman kuvauksen DSR:stä antavat Brocke, Hevner & Maedche (2020, s. 1–2). He mainitsevat lisäksi, että DSR pyrkii luomaan tietoa siitä, kuinka asiat tulisi rakentaa tai järjestellä saavuttaakseen asetetut tavoitteet ja tätä tietoa kutsutaan suunnittelutiedoksi (engl. design knowledge).

Brocke et al. (2020, s. 2–12) esittävät muutamia eri viitekehyksiä DSR:n suorittamiseen, joista tähän työhön valittiin Pefferin et al. (2007, s. 52–56) luoma viitekehys. He kehittivät DSR:ää varten kuuden kohdan viitekehyksen, jota hyödyntäen DSR-tutkimukset voidaan suorittaa. Tätä viitekehystä he kutsuvat DSR-metodologian prosessin malliksi (engl. DSRM Process Model), jota havainnollistaa kuva 10. Mallin kohtia kutsutaan aktiviteeteiksi.



Kuva 10. DSR-metodologian prosessin malli. (Peffer et al. 2007, s. 54)

DSR-metodologian prosessin vaiheet ovat:

**Aktiviteetti 1: Ongelman tunnistaminen ja tutkimuksen motivaatio.** Tässä vaiheessa määritellään tarkasti tutkimusongelma ja perustellaan ratkaisun tuoma arvo. Ongelman määrittelyä käytetään artefaktin kehittämiseen, joka pyrkii ratkaisemaan ongelman tehokkaasti. Ratkaisun arvon perustelemisella saavutetaan kaksi asiaa: se motivoi tutkijaa ja tutkimuksen yleisöä ratkaisun tavoitteluun ja tulosten hyväksymiseen, ja se auttaa ymmärtämään tutkijan käsitystä ongelman merkityksestä. Tämän aktiviteetin suorittamiseen tarvitaan tieto ongelmasta ja sen ratkaisun tärkeydestä. (Peffer et al. 2007, s. 52–55)

Tämän tutkimuksen ongelman tunnistaminen ja tutkimuksen motivaatio on kuvattu johdannon kappaleissa ”Työn tausta” ja ”Tavoitteet ja rajaus”. Tutkimuksessa keskitytään pilvitietovarastossa sijaitsevien raportointitaulujen optimointitapoihin nopeuden ja kustannustehokkuuden osalta sekä muunto että tallennusvaiheessa. Tutkimusta on jäsennelty seuraavilla tutkimuskysymyksillä:

- Millä tavoin SQL-pohjaista datan muuntoa voidaan optimoida pilvitietovarastoissa?
- Millä tavoin pilvitietovaraston raportointitaulujen tallennusta voidaan optimoida kyselyjen kannalta?
- Kuinka valitaan optimaalinen raportointitaulun tallennustapa?

Näistä tutkimuskysymyksistä ensimmäinen ei liity tutkimuksen empiiriseen osuuteen. Tutkimuksen motivaationa toimii toimeksiantajaorganisaation uuden pilvitietovaraston kulutuksen maltillisena pitäminen sekä järjestelmällisempi raportointitaulujen luontitapa organisaation sisällä.

**Aktiviteetti 2: Määrittele ratkaisun tavoitteet.** Ratkaisun tavoitteet päätellään ongelman määritelmän avulla ottaen huomioon sen, mikä on mahdollista ja toteuttamiskelpoista. Tavoitteet voivat olla joko määrällisiä tai laadullisia. Tavoitteet tulee päätellä rationaalisesti ongelman määritelmästä. Tämän aktiviteetin suorittamiseen tarvitaan tieto ongelman tilasta sekä nykyisistä ratkaisuista, jos sellaisia on. Mikäli nykyisiä ratkaisuja on olemassa, tarvitaan myös tieto niiden tehokkuudesta. (Peffer et al. 2007, s. 55)

Tutkimuksen tavoitteisiin päästään, kun löydetään vastaukset tutkimuskysymyksiin. Toiseen tutkimuskysymykseen, eli ”Millä tavoin pilvitietovaraston raportointitaulujen tallennusta voidaan optimoida kyselyjen kannalta?”, löydetään vastaus kirjallisuuskatsauksesta, jossa esitellään raportointitaulujen optimointien keinoja eli ratkaisu löytyy sieltä. Kolmas tutkimuskysymys vaatii empiiristä tutkimusta, johon keskitytään seuraavissa kappaleissa.

Tavoite on siis löytää vastaus kysymyksen ” Kuinka valitaan optimaalinen raportointitaulun tallennustapa?” ja ratkaisuna tähän pyritään luomaan raportointitaulun optimaalista tallennustapaa ohjaava vuokaavio, jota tietovaraston fyysistä mallia luova henkilö voi seurata valitakseen oikean tallennustavan raportointitaululle. Vuokaavion ensimmäinen versio pohjautuu kirjallisuuskatsauksen tietoihin, jonka jälkeen luotua vuokaaviota testataan soveltuvassa ympäristössä. Ratkaistuksi ongelmaa pidetään siinä vaiheessa, kun vuokaavio ohjaa demonstrointiaktiviteetissa aina optimaaliseen tallennustapaan.

**Aktiviteetti 3: Suunnittele ja kehitä.** Tässä vaiheessa luodaan artefakti. Esimerkkejä artefakteista ovat rakenteet, mallit, menetelmät ja teknisten, sosiaalisten tai informatiivisten resurssien uusia ominaisuuksia. Periaatteessa artefakti voi olla mikä tahansa suunniteltu objekti, jonka suunnitteluun on hyödynnetty tutkimusta. Tässä aktiviteetissa määritellään artefaktin haluttu toiminnallisuus ja sen arkkitehtuuri, jonka jälkeen luodaan itse artefakti. Tämän aktiviteetin suorittamiseen tarvitaan teoriatietoa, jota voidaan hyödyntää ratkaisun suunnittelussa. (Peffer et al. 2007, s. 55)

Tässä tutkimuksessa luotava artefakti on vuokaavio, joka pyrkii ohjaamaan raportointitaulun optimaaliseen tallennustapaan. Vuokaavion suunnittelu kuvataan tarkemmin kappaleessa 4.2, jossa käydään läpi teoriapohja, joka vaikutti vuokaavion ensimmäisen version luomiseen. Tämän lisäksi kappaleessa käydään läpi vuokaavion ensimmäisen version kohdat ja sen toiminta.

**Aktiviteetti 4: Demonstro.** Luotua artefaktia sovelletaan ongelman ratkaisemiseksi. Demonstroinnin voi tehdä esimerkiksi käytön kokeilun, simuloinnin, tapaustutkimuksen tai muun sopivan toiminnan kautta. Tämän aktiviteetin suorittamiseen tarvitaan artefakti, tieto siitä miten sitä käytetään ja soveltuva ympäristö sen käyttöön. (Peffer et al. 2007, s. 55)

Kappaleessa ”4.3 Vuokaavion testaus ja jatkokehitys” testataan vuokaavion toimintaa sen käytön simuloinnin kautta. Käytön simulointia varten tarvitaan soveltuva ympäristö, dataa, raportointitauluja, raportointityökalu, soveltuvat kysymykset sekä sopiva mittaustapa. Edellä mainitut asiat avataan kappaleessa, jonka lisäksi suoritetaan itse käytön kokeilu.

**Aktiviteetti 5: Arvioi.** Demonstroinnin jälkeen mitataan tai tarkkaillaan, kuinka hyvin luotu artefakti soveltuu ongelman ratkaisemiseen. Tässä aktiviteetissa verrataan ratkaisun tavoitteita todellisiin havaittuihin tuloksiin. Aktiviteetin suorittamista varten tulee tietää soveltuvat mittarit ja analyysitekniikat. Ongelman luonteesta ja artefaktista riippuen arviointi voi

tapahtua monin eri tavoin. Periaatteessa arviointi voi sisältää mitä tahansa sopivaa empiiristä näyttöä tai loogista todistetta. Esimerkkeinä mittareista Peffers et al. (2007, s. 56) mainitsevat suorituskykymittarit kuten budjetit ja tuotetut kohteet, simuloinnin, asiakaspalautteet, tyytyväisyyskyselyiden tulokset ja järjestelmän suorituskyvyn määrälliset mittarit kuten vasteaika ja käytettävyys. Aktiviteetin lopussa päätetään, että halutaanko edelleen pyrkiä parantamaan luotua artefaktia palaamalla aktiviteettiin kolme, vai jatketaanko seuraavaan aktiviteettiin eli kommunikointiin. Tutkimusympäristö saattaa myös vaikuttaa siihen, onko iterointi mahdollista vai ei.

Kappaleessa ”4.3 Vuokaavion testaus” yhdistyy aktiviteetit 4 & 5 ja kappale jatkuu käytön simuloinnin tulosten arvioimisella. Tulosten pohjalta arvioitiin, että päästiinkö ratkaisun tavoitteisiin.

**Aktiviteetti 6: Kommunikoi.** Lopussa viestitään ongelma ja sen tärkeys, artefakti, sen hyödyllisyys ja tehokkuus asiankuuluville henkilöille. Peffers et al. (2007, s. 56) mainitsevat, että tieteellisissä tutkimusjulkaisuissa tutkijat voivat käyttää tämän prosessin rakennetta tutkimuksen rakenteena. Tätä aktiviteettia varten tulee tietää asianmukaiset viestintätavat ja yleisö, jolle viestiä tulokset. Tämän tutkimuksen ”Kommunikoi”-aktiviteetti tapahtuu kappaleessa 5.

## 4.2 Vuokaavion suunnittelu

Tässä kappaleessa kuvataan DSR:n kolmas aktiviteetti, eli artefaktin suunnittelu ja kehitys. Tässä tutkimuksessa luotu artefakti on vuokaavio, jonka suunnittelun pohjana toimii kirjallisuuskatsauksen kappaleet 2.3, 3.2 ja 3.3. Vuokaavion ensimmäisen version syntymiseen vaikuttivat seuraavat kirjallisuuskatsauksessa mainitut asiat, johon viitataan myöhemmin tekstissä listana:

1. Denormalisoitu raportointitaulu on suorituskyvyltään parempi verrattuna tähtimallinnettuihin tauluihin (Jukic et al. 2017, s. 67 & 78, Costa et al. 2019, s. 34–36)
2. Data tulisi tallentaa mahdollisimman summattuun ja aggregoituun muotoon (Sherman & Imhoff 2015, s. 93–94)
3. Raportointitaulun optimointikeinoina kannattaa harkita ositusta, klusterointia tai näiden yhdistelmää (Lightstone et al. 2010; Costa et al. 2019, s. 34–36; Vaisman & Zimányi 2022, s. 245)

4. Löytääkseen sopivat attribuutit ositukseen ja klusterointiin tulee raportointitaulun dataa analysoida kardinaliteettien osalta (Costa et al. 2019, s. 34–36)
5. Tasaisesti jakautuneet matalakardinaliteettiset attribuutit, joiden pohjalta raportointitaulua suodatetaan, kannattaa osittaa (Costa et al. 2019, s. 34–36)
6. Korkeakardinaliteettiset attribuutit, joiden pohjalta raportointitaulua suodatetaan, kannattaa klusteroida (Costa et al. 2019, s. 34–36)
7. Ositusten kokojen tulisi olla vähintään 50 Mt (Lightstone et al. 2010, s. 141–142)
8. Klusterointia kannattaa aina harkita, sillä se vähentää siirräntää oikein käytettynä (Lightstone et al. 2010, s. 166)
9. Ositus on klusterointia tehokkaampaa, joten pyritään ensisijaisesti ohjaamaan siihen (Costa et al. 2019, s. 34–36)

Lähtöpisteenä vuokaavioille on se kohta, kun jonkin liiketoiminta-alueen dataa halutaan muodostaa raportointitaulu raportointisovelluksen käytettäväksi. Loppupiste vuokaavioilla on kyseisen raportointitaulun optimaalinen tallennustapa. Näitä voi olla viisi kappaletta, kuitaten yllä mainitun listan kohdat kolme ja kahdeksan:

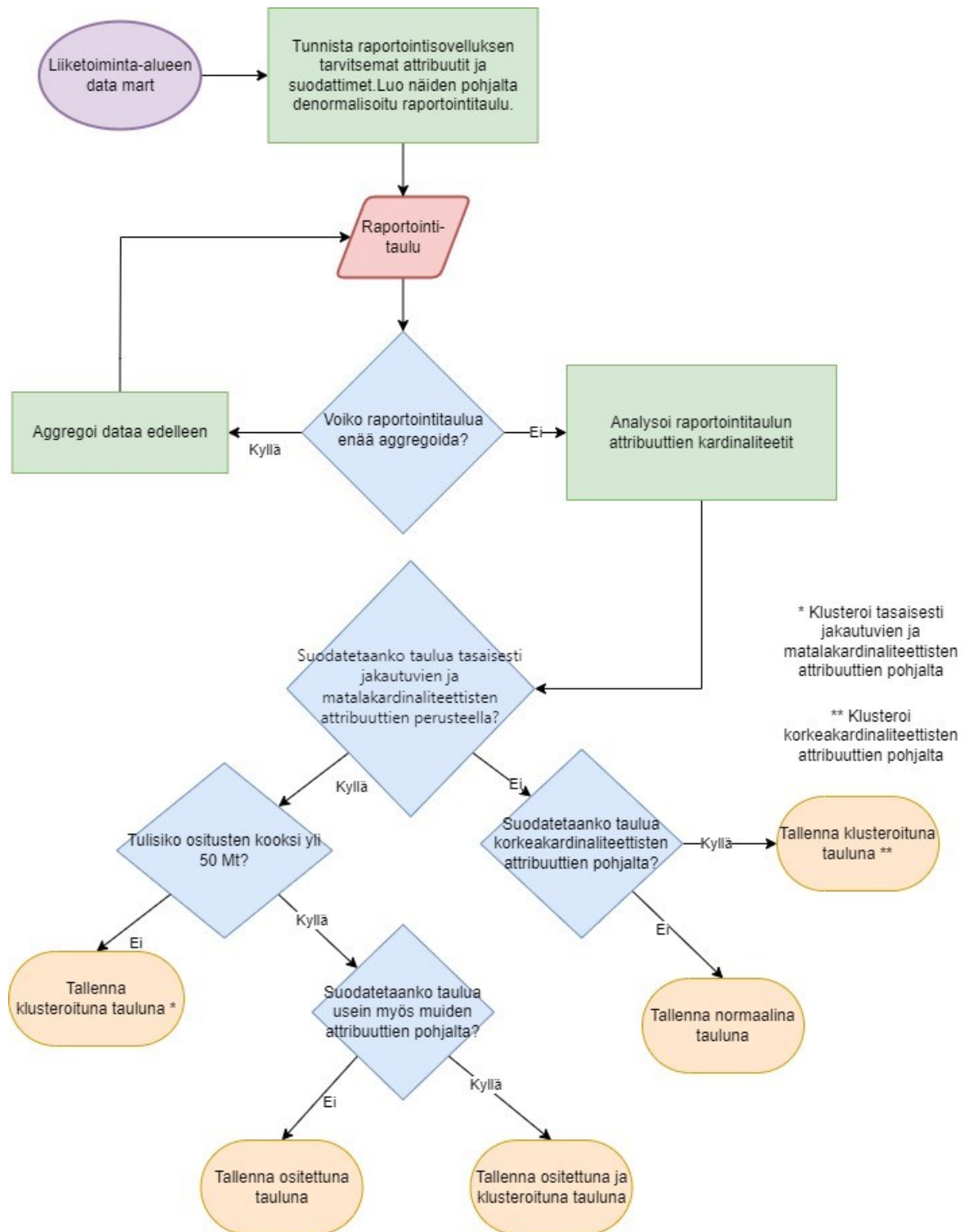
- Normaali taulu
- Ositettu taulu
- Klusteroitu taulu korkeakardinaliteettisten attribuuttien pohjalta
- Klusteroitu taulu matalakardinaliteettisten attribuuttien pohjalta
- Ositettu ja klusteroitu taulu

Lähtö- ja loppupisteiden väliin upotettiin listan kohdat prosessi- ja päätöspisteiden muodossa, aloittamalla kohdasta yksi: denormalisoidun raportointitaulun mallinnuksen ja luonnin prosessipiste. Tämä sisältää raportointisovelluksen tarvitsemien attribuuttien ja suodattimien kartoituksen, joista käytettävien suodattimien tietoa tarvitaan vuokaavion myöhemässä vaiheessa. Kohta kaksi lisättiin vuokaavioon päätöspisteenä, kysyen käyttäjältä, voiko raportointitaulua enää aggregoida enempää? Mikäli voi, ohjataan käyttäjä vielä aggregoimaan dataa prosessipisteen kautta, jonka jälkeen palataan takaisin samaan kysymykseen. Kun raportointitaulun dataa ei voi enää aggregoida, pääsee käyttäjä datan analysoinnin prosessipisteeseen eli listan kohtaan neljä.

Kun analysointi on tehty, pääsee vuokaavion käyttäjä päätöspisteeseen, jossa kysytään ”Suodatetaanko taulua tasaisesti jakautuvien ja matalakardinaliteettisten attribuuttien pohjalta?”. Tämä kysymys esitetään ensimmäisenä siitä syystä, että käyttäjää koitetaan ohjata ensin osituksen suuntaan klusteroinnin sijasta, kuitaten listan kohdan yhdeksän. Käyttäjän pitäisi pysyä vastaamaan tähän aiempien prosessipisteiden selvitysten ja analysointien pohjalta.

Käyttäjän vastatessa ”kyllä”, siirrytään seuraavaan päätöspisteeseen, jossa kysytään ositusta käyttäessä ositusten koosta: ”Tulisiko ositusten kooksi yli 50 Mt?”. Tällä kysymyksellä kuitataan listan kohta seitsemän. Tähän vastatessa ”ei” päästään loppupisteeseen ”Tallenna (raportointitaulu) klusteroituna tauluna”. Tähän valikoitui klusteroitu taulu normaalin taulun sijaan siitä syystä, että kohdan 8 mukaan klusterointi vähentää oikein käytettynä (eli mikäli dataa suodatetaan attribuutin mukaan) datan siirrääntää. Mikäli osituksen koko -kysymykseen vastataan kyllä, päästään vielä yhteen päätöspisteeseen ennen loppupistettä, joka on ”Suodatetaanko taulua usein myös muiden attribuuttien pohjalta?”. Vastauksen ollessa ei, ohjataan käyttäjää tallentamaan raportointitaulu ositettuna. Vastauksen ollessa kyllä, ohjataan tekemään sekä ositettu että klusteroitu taulu.

Mikäli päätöspisteessä ”Suodatetaanko taulua tasaisesti jakautuvien ja matalakardinaliteettisten attribuuttien pohjalta?” vastataan ei, päädytään seuraavaan päätöspisteeseen, johon vastatessa vuokaavio ohjaa loppupisteen kautta tallennustapoihin. Viimeisessä päätöspisteessä kysytään ”Suodatetaanko taulua korkeakardinaliteettisten attribuuttien pohjalta?”. Mikäli vastataan kyllä, ohjaa vuokaavio tallentamaan raportointitaulun klusteroituun tauluun, kuitaten listan viimeisen kohdan kuusi. Vastatessa ei, vuokaavio ohjaa tallentamaan raportointitaulun normaaliksi tauluksi, tarkoittaen sitä, että analyysissa ei tunnistettu hyviä osituksen tai klusteroinnin attribuutteja tai sitä, että raportointitaulua ei suodateta raportointisovelluksessa. Kuvassa 11 on esitelty edellä kuvattu vuokaavio, eli tämän DSR:n artefakti.



Kuva 11. Raportointitaulun optimaalista tallennustapaa ohjaavan vuokaavion ensimmäinen versio.

### 4.3 Vuokaavion käytön simulointi

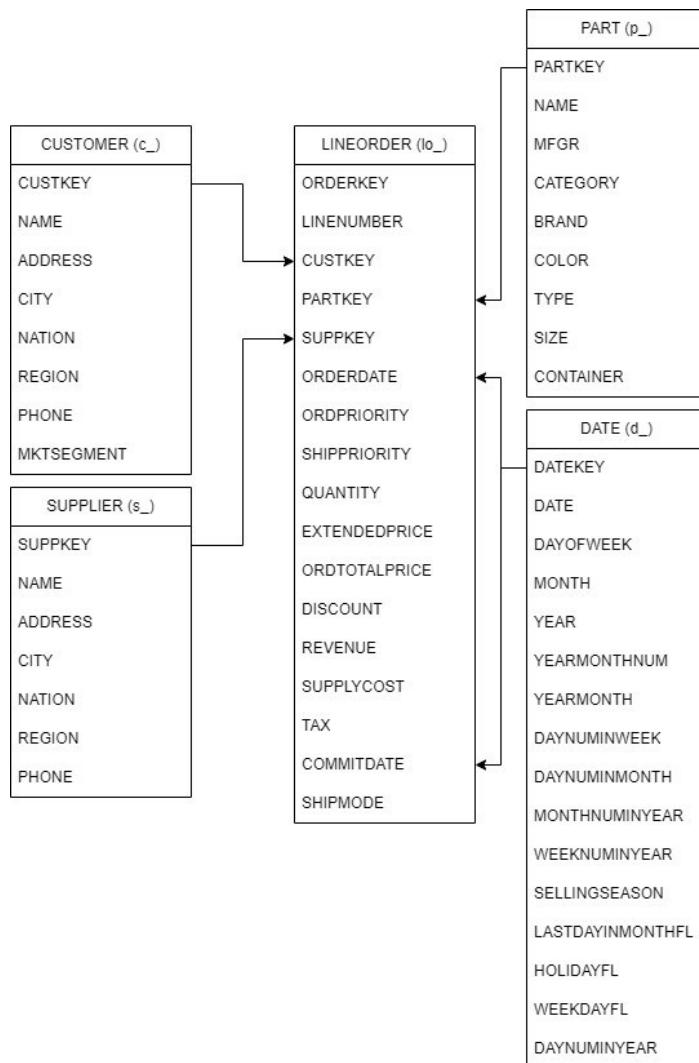
Artefaktin eli vuokaavion ensimmäisen version ollessa valmis, DSR:n seuraavat vaiheet ovat ”demonstroi” ja ”arvioi”, joihin tämä kappale keskittyy. Vuokaavion toimivuutta testataan luomalla realistinen ympäristö, jossa vuokaavio ohjaa raportointitaulun optimaaliseen tallennustapaan. Realistista ympäristöä varten tarvitaan kolumnaarinen pilvitietovarasto, jonne raportointitaulut tallennetaan ja dataa, jonka pohjalta raportointitaulut muodostetaan. Tämän lisäksi tarvitaan muutamia hypoteettisia liiketoiminnan kysymyksiä, jotka vaikuttavat siihen, miten datasta tehdään raportointitauluja. Näiden lisäksi tarvitaan raportointityökalu, joka hyödyntää raportointitauluja sekä tapa mitata eri raportointitaulujen tehokkuutta.

Testaukseen soveltuvaksi pilvitietovarastoksi valittiin Google Cloud Platformin (myöhemmin GCP) BigQuery ja raportointityökaluksi Looker Studio. Valintaan vaikutti se, että toimeksiantajayrityksellä on nämä työkalut käytössään. Valitut työkalut sopivat teknisesti kaiken tarvittavan testaamiseen, sillä BigQueryssa pystyy tallentamaan raportointitaulut sekä osittaen että klusteroiden. BigQuery on kolumnaarisesti datansa tallentava tietovarasto, joten siltäkin osin soveltuva. Tämän lisäksi BigQueryssa voidaan myös monitoroida käytetyn laskentatehon tai prosessoitujen tavujen määrää ja kyselyjen nopeutta, jotka ovat tämän työn optimaalisuuden mittareita.

#### 4.3.1 Käytetty data

Datan osalta tutkimuksessa käytettiin Star Schema Benchmarkiksi (myöhemmin SSB) kutsuttua dataa, joka on O’Neil, O’Neil & Chenin (2009) muunnos Transaction Processing Performance Councilin (TPC) luomasta TPC-H -nimisestä päätöstukijärjestelmien suorituskykytestistä. Sanchez (2016) ehdottaa artikkelissaan, että SSB:sta tulisi uusi päätöstukijärjestelmien suorituskyvyn testauksen standardi, syrjäyttäen TPC-H:n, jota pidettiin ennen alan standardina Lightstonen et al. mukaan (2010, s. 103).

Alkuperäisessä TPC:n suorituskykytestissä on kahdeksasta tietokantataulusta koostuva tietokanta, joka kuvaa varaosien tilauksia ja niiden tilausrivejä. SSB:ssä tätä skeemaa on denormalisoitu tähtimallin muotoon ja siinä on samat tiedot viidessä taulussa, jotka ovat; lineorder, customer, part, supplier ja date. Näistä lineorder on faktataulu ja loput dimensiotauluja.



Kuva 12. Star Schema Benchmarkin skeema. (O'Neil et al. 2009)

Datan luontiin käytettiin GitHubista löytynyttä SSB generaattoria (Phillips & Sundstrom 2010), jolle pystyy antamaan argumenttina käytetyn skaalan, joka vaikuttaa luotujen taulujen suuruuteen. Tutkimusta varten luotiin neljä eri kokoista datasettiä skaaloilla 0,1, 1, 10 ja 25. Luonnin jälkeen datasetit ladattiin GCP:n Cloud Storageen eli tietoahtaaseen csv-tiedostoina, joista luotiin tämän jälkeen BigQueryn tietokantatauluja.

Taulut edustavat kirjallisuuskatsauksessa esitettyä pilvitietovarastoinnin arkkitehtuurin ja myös vuokaaviossa olevaa data mart -kohtaa. Seuraavaksi data mart -tauluista luotiin yksi denormalisoitu raportointitaulu, jossa on kaikki attribuutit. Tämän raportointitaulun optimaalista tallennustapaa eri kyselyjen kannalta tutkitaan. Alla oleva taulukko 5 kertoo tämän optimoimattoman raportointitaulun rivimäärän ja koon eri datasettien skaaloilla.

Taulukko 5. Denormalisoidun taulun rivimäärä ja koko.

Skaala	Rivimäärä	Koko (Mt)
<b>0,1</b>	600 597	341,27
<b>1</b>	6 001 173	3 440,64
<b>10</b>	59 986 217	34 539,52
<b>25</b>	149 996 354	86 507,52

#### 4.3.2 Testijärjestelyjen kuvaus

O’Neil et al. (2009) esittelevät suorituskykytestausta varten 13 eri SQL-kyselyä, joiden tuloksilla pyritään vastaamaan hypoteettisiin liiketoiminnan esittämiin kysymyksiin. Osa näistä SSB:n SQL-kyselyistä toimivat testiympäristössämme raportointisovelluksen ajamina kyselyinä raportointitaulua vasten, vaikka raportointisovelluksen toimintaa simuloidaankin ajamalla vain näitä SQL-kyselyjä tietovarastoa vasten.

Osoittaakseni, että raportointisovelluksen ajamia kyselyitä voi simuloida vain ajamalla SQL-kyselyitä tietovarastoa vastaan, käydään läpi esimerkki demoraportin kautta. Alla on kuva yksinkertaisesta Looker Studio -raportista, jossa on yksi taulukko ja yksi suodatin. Demoraportin data on kuvitteellista dataa. Raportti on yhdistetty BigQueryn tauluun nimeltä masters-thesis-ao.performance\_tracking.card\_external\_non\_partitioned.

Resurssien käytön monitoroinnin demo

File Editing View Insert Page Arrange Resource Help

↶ ↷ 🔍 Add page 📄 Add data 📊 Add a chart

🔽 + Add quick filter

card\_provider

	card_provider	Record Count
1.	VISA 16 digit	16,651
2.	JCB 16 digit	16,589
3.	JCB 15 digit	8,464
4.	Discover	8,412
5.	Maestro	8,381
6.	VISA 19 digit	8,381
7.	Diners Club / Carte Blanche	8,362
8.	American Express	8,284
9.	Mastercard	8,268
10.	VISA 13 digit	8,208

1 - 10 / 10 < >

Kuva 13. Yksinkertainen Looker Studio -taulukko demoa varten.

Kun raportin card\_provider-suodattimesta valitaan yksi arvo (tässä demossa Mastercard), ajaa Looker Studio taustalla seuraavan SQL-kyselyn BigQuerya vasten ja esittää kyselyn tulokset raportilla:

SELECT

```
t0.card_provider,
COUNT(1) AS t0_qt_j7jbkbnndd
FROM `masters-thesis-ao.performance_tracking.card_external_non_partitioned` AS t0
WHERE
t0.card_provider IN ('Mastercard')
GROUP BY t0.card_provider
ORDER BY t0_qt_j7jbkbnndd DESC
LIMIT 2000001;
```

Looker Studion ajaman kyselyn löytää, kun BigQueryn puolella tekee kyselyn INFORMATION\_SCHEMA.JOBS-nimiseen tauluun, joka tallentaa kaikki BigQuery-tiliä ja näkyymiin tehdyt kyselyt. Kuten demosta näemme, Looker Studio -raportointisovellus suorittaa myös taustalla vain SQL-kyselyitä ja näin ollen SSB:ssä esiteltyt SQL-kyselyt soveltuvat hyvin simuloimaan raportointisovelluksen toimintaa.

Palaten takaisin näiden raportointisovelluksen toimintaa simuloivien kyselyiden valintaan; SSB:n SQL-kyselyistä pyrittiin löytämään vuokaavion testausta varten sopivat. Kysymysten valitsemista ennen suoritettiin raportointitaulun datalle analyysi sen jakaumista ja kardinaliteeteista, kuten Costa et al. (2019, s. 36) suosittelevat tehtävän ennen optimaalisen tallennustavan valitsemista ja kuten vuokaaviossakin ohjataan. Datan analysoinnista paljastui, että p\_brand -attribuutti omaa korkeimman kardinaliteetin datasta, kun taas tasaisesti jakautunut ja matalan kardinaliteetin attribuutti löytyi d\_year-attribuutista. Näiden tietojen pohjalta esitettiin siis SSB:n kysymykset, jotka täyttivät seuraavat vaatimukset:

- kumpaakaan attribuuttia ei hyödynnetä kysymyksessä
- molempia attribuutteja hyödynnettiin kysymyksessä
- korkean kardinaliteetin attribuuttia hyödynnettiin kysymyksessä
- matalan kardinaliteetin attribuuttia hyödynnettiin kysymyksessä

Nämä vaatimukset täyttävillä kysymyksillä saadaan oletusarvoisesti kaikki vuokaavion raportointitaulujen tallennustavoista testattua. Vuokaavion testaukseen löytyi kolme suoraan sopivaa SQL-kyselyä: Q1.1, Q1.2 ja Q2.3. Näiden kysymysten alkuperäiset versiot sekä testissä käytetyt versiot löytyvät liitteestä 1. Q1.1 on kysymys, joka suodattaa dataa d\_yearin pohjalta eli sillä saadaan testattua matalan kardinaliteetin attribuuttia hyödyntävän raportointitaulun optimaalinen tallennustapa. Q1.2 ei suodata dataa kummankaan attribuutin kohdalta, joten se testaa sellaisen raportointitaulun optimaalista tallennustapaa, joka ei suodata dataa minkään kyselyissä käytetyn attribuutin pohjalta. Q2.3 suodattaa dataa korkean kardinaliteetin attribuutin eli p\_brandin pohjalta, vastaten sen tyylisten kysymysten raportointitaulun optimaaliseen tallennustapaan.

SSB:n suorituskyvyn testauksen kyselyistä ei löytynyt sellaista kyselyä, joka suodattaisi dataa sekä d\_yearin pohjalta, että p\_brandin pohjalta, jättäen aukon sellaisen raportointitaulun optimaaliseen tallennustavan testaukseen, joka suodattaa dataa sekä matala- että korkeakardinaliteettisen attribuutin pohjalta. Näin ollen SSB:n kysymystä Q3.2 muokattiin siten, että vaihdettiin alkuperäinen WHERE-ehto ”s\_nation = 'UNITED STATES' ” seuraavaan: ”p\_brand = 'MFGR#2221' ”. Näin muokattuna Q3.2 suodattaa dataa sekä d\_yearin että p\_brandin pohjalta.

Raportointisovellusten edustamien kyselyjen sekä ositusta ja klusterointia varten sopivien attribuuttien löydyttyä pystytään muodostamaan suorituskykytestausta varten kaikki raportointitaulut, joiden tehokkuutta kysymyskohtaisesti tutkitaan. Testausta varten tehtiin viisi

eri tavalla tallennettua raportointitaulua per skaala (eli 20 yhteensä), kuten vuokaaviossakin on kuvattu:

1. Normaali taulu eli alkuperäinen, optimoimaton taulu
2. Ositettu taulu d\_yearin pohjalta
3. Klusteroitu taulu p\_brandin pohjalta
4. Klusteroitu taulu d\_yearin pohjalta
5. Ositettu ja klusteroitu taulu, ositus d\_yearin pohjalta ja klusterointi p\_brandin pohjalta

Suorituskykymittaukset ajetaan kaikkia näitä raportointitauluja vasten, jonka jälkeen analysoidaan tuloksia kysymyskohtaisesti.

### 4.3.3 Testauksen mittarit ja mittaus

Testattavien raportointitaulujen ollessa valmiita tarvitaan tapa, jolla mitata niiden tehokkuutta. Raportointitauluja pyritään optimoimaan siten, että ne olisivat sekä kulutehokkaita että nopeita, kun raportointisovellukset suorittavat niihin kyselyitä. Näitä varten tarvitaan mittarit, jotta kulutehokkuutta ja nopeutta voidaan arvioida. Sopivat mittarit löytyvät jo kertaalleen mainitusta BigQueryn INFORMATION\_SCHEMA.JOBS-taulusta. Jokaisesta suoritusta kyselystä tallennetaan paljon metatietoa, mutta tätä tutkimusta varten löytyi kaksi olennaista: kyselykohtaiset `total_bytes_billed` ja `total_slot_ms` -tiedot. Google (2024a) BigQueryn dokumentaatiosta nähdään, että `total_bytes_billed` -kenttä kertoo tavumäärän, jonka pohjalta BigQuery laskuttaa kyseessä olevasta kyselystä. Tätä tietoa tarkastelemalla päästään kiinni kulupuolen analysointiin. Toinen soveltuva mittari, eli `total_slot_ms`, kertoo dokumentaation mukaan millisekunteina ajan, jonka kysely on ollut ”RUNNING”-tilassa eli suoritustilassa. Tätä tietoa seuraamalla päästään mittaamaan nopeusaspektia.

Ennen mittauksia määriteltiin vielä, että mikä on optimaalisuuden määritelmä, sillä mitataan kahta eri tietoa. Odotusarvoisesti prosessoidut tavut (`total_bytes_billed`) ja nopeus (`total_slot_ms`) ovat linjassa keskenään, mutta mikäli näin ei ole, tarvitaan määritelmä, jolla eri raportointitaulujen tallennustavat saadaan tehokkuusjärjestykseen. Toimeksiantajaorganisaation kanssa sovittiin, että priorisoidaan prosessoitujen tavujen vähimmäismäärää ja nopeus tulee mukaan sellaisissa tilanteissa, joissa prosessoitujen tavujen määrä on sama eri raportointitaulujen välillä. Organisaation kanssa pohdittiin myös eri painoarvojen antamista kahdelle mitatulle tiedolle, mutta lopulta päädyttiin kuitenkin kulutehokkuuden priorisointiin.

Mittaus suoritettiin ajamalla tietovarastoa vasten kappaleessa 4.3.2 valitut ja liitteestä 1 löytyvät SQL-kyselyt kaikille raportointitauluille ja niiden skaaloille, joita on yhteensä 20 kappaletta. Ajettuja uniikkeja kyselyitä on siis 80 kappaletta. Uniikit kyselyt ajettiin viisi kertaa, jonka jälkeen raportointitaulu- ja skaalakohtaisista tuloksista otettiin keskiarvot valittujen mittareiden osalta. Toistoilla ja keskiarvon ottamisella pyrittiin vähentämään mahdollisen yhden kauemmin kestäneen kyselyn vaikutusta tuloksiin.

Mittauksen jälkeen tulosten tulkintaa helpottamaan päätettiin laskea kaksi uutta mittaria: prosessoidut tavut suhteessa normaaliin tauluun ja aika suhteessa normaaliin tauluun. Ajatuksena on verrata aina eri tallennusmuotojen toimivuutta normaaliin tai toisin sanoen

optimoimattomaan tauluun. Lasketut mittarit ovat prosentuaalisia mittareita suhteessa normaaliin tauluun, jotka on laskettu yksinkertaisesti jakamalla raportointitaulun prosessoidut tavut tai nopeus suhteessa sen skaalan normaalin taulun vastaaviin. Alla oleva kuva 14 havainnollistaa kuvatun mittauksen tuloksia ja raportointitaulujen tehokkuuden järjestämistä, joista ensimmäisen sijan omaava on aina optimaalinen vaihtoehto.

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
0,1	Ositettu d_yearin pohjalta	10 485 760	79,2	35,71 %	53,66 %	1
0,1	Ositettu (d_year) ja klusteroitu (p_brand)	10 485 760	81,4	35,71 %	55,15 %	2
0,1	Normaali	29 360 128	147,6	100,00 %	100,00 %	3
0,1	Klusteroitu d_yearin pohjalta	29 360 128	171,6	100,00 %	116,26 %	4
0,1	Klusteroitu p_brandin pohjalta	29 360 128	209,8	100,00 %	142,14 %	5
1	Ositettu d_yearin pohjalta	44 040 192	532,6	15,27 %	23,22 %	1
1	Ositettu (d_year) ja klusteroitu (p_brand)	44 040 192	631,0	15,27 %	27,50 %	2
1	Klusteroitu d_yearin pohjalta	44 040 192	1 086,2	15,27 %	47,35 %	3
1	Normaali	288 358 400	2 294,2	100,00 %	100,00 %	4
1	Klusteroitu p_brandin pohjalta	288 358 400	2 914,2	100,00 %	127,03 %	5

Kuva 14. Esimerkki mittaustuloksista.

#### 4.3.4 Vuokaavion ohjaamat tallennustavat, tulokset ja tulosten analysointi

Mittausten ollessa valmiit, päästään analysoimaan tuloksia ja tarkastelemaan vuokaavion ohjaamien tallennustapojen optimaalisuutta. Tässä kappaleessa käydään valittujen kysymysten osalta läpi seuraavat vaiheet:

1. Vuokaavion ohjaama raportointitaulun tallennustapa kysymyskohtaisesti
2. Kysymykseen liittyvät mittaustulokset
3. Mittaustulosten analysointi

Liitteistä 2–5 löytyy vuokaavion ohjaamat kysymyskohtaiset tallennustavat havainnollistettuna.

#### Kysymys Q1.1:

Q1.1 suodattaa dataa d\_yearin pohjalta, mutta ei p\_brandin pohjalta. Kysymys suodattaa siis dataa matalakardinaliteettisen attribuutin pohjalta. Vuokaaviota seurattaessa voidaan päätyä kahteen eri raportointitaulun tallennustapaan, jotka ovat ajanjakson pohjalta klusteroitu taulu ja ajanjakson pohjalta ositettu taulu. Näiden oletetaan siis olevan optimaaliset tallennustavat, riippuen taulun koosta. Valinta näiden kahden välillä pohjautuu (potentiaalisen) ositusten kokoon; laskennallisen osituksen koon ollessa yli 50 Mt luodaan ositettu taulu, muussa

tapauksessa klusteroitu taulu. Taulukossa 6 on laskennalliset osien koot, jotka on laskettu sillä olettamalla, että kaikki d\_year-attribuutin arvot ovat rivimäärällisesti tasaisesti jakautuneet. Datassa on 7 vuoden ajalta vuosilukuja (1992–1998), joten yhden osituksen koko on laskettu: (taulun koko / 7). Alla olevassa taulukossa 6 on taulujen koot ja oletetut ositusten koot eri skaaloilla:

Taulukko 6. Ositusten koot eri skaaloilla.

Skaala	Taulun koko (Mt)	Osituksen koko (Mt)
0,1	341,27	48,75
1	3 440,64	491,52
10	34 539,52	4 934,22
25	86 507,52	12 358,22

Liitteessä 2 on korostettu vuokaavioon vihreillä nuolilla sitä polkua, johon vuokaavio ohjaa kysymyksen Q1.1 osalta. Liitteestä käy ilmi, että vuokaavio voi ohjata kahteen eri tallennustapaan riippuen ositusten koosta. Ainoastaan skaalalla 0,1 olevan datasetin taulu tulisi vuokaavion mukaan tallentaa ajanjakson, eli d\_yearin pohjalta klusteroituun tauluun. Loppujen optimaalisin tallennustapa olisi vuokaavion mukaan d\_yearin pohjalta ositettu taulu. Alla olevassa kuvassa 15 on Q1.1 mittaustulokset.

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
0,1	Ositettu d_yearin pohjalta	10 485 760	79,2	35,71 %	53,66 %	1
0,1	Ositettu (d_year) ja klusteroitu (p_brand)	10 485 760	81,4	35,71 %	55,15 %	2
0,1	Normaali	29 360 128	147,6	100,00 %	100,00 %	3
0,1	Klusteroitu d_yearin pohjalta	29 360 128	171,6	100,00 %	116,26 %	4
0,1	Klusteroitu p_brandin pohjalta	29 360 128	209,8	100,00 %	142,14 %	5
1	Ositettu d_yearin pohjalta	44 040 192	532,6	15,27 %	23,22 %	1
1	Ositettu (d_year) ja klusteroitu (p_brand)	44 040 192	631,0	15,27 %	27,50 %	2
1	Klusteroitu d_yearin pohjalta	44 040 192	1 086,2	15,27 %	47,35 %	3
1	Normaali	288 358 400	2 294,2	100,00 %	100,00 %	4
1	Klusteroitu p_brandin pohjalta	288 358 400	2 914,2	100,00 %	127,03 %	5
10	Ositettu d_yearin pohjalta	437 256 192	4 902,2	15,19 %	33,42 %	1
10	Ositettu (d_year) ja klusteroitu (p_brand)	437 256 192	5 160,2	15,19 %	35,18 %	2
10	Klusteroitu d_yearin pohjalta	437 256 192	9 507,6	15,19 %	64,83 %	3
10	Normaali	2 879 389 696	14 666,6	100,00 %	100,00 %	4
10	Klusteroitu p_brandin pohjalta	2 879 389 696	27 212,6	100,00 %	185,54 %	5
25	Ositettu (d_year) ja klusteroitu (p_brand)	1 092 616 192	12 805,6	15,17 %	56,65 %	1
25	Ositettu d_yearin pohjalta	1 092 616 192	23 778,2	15,17 %	105,20 %	2
25	Klusteroitu d_yearin pohjalta	1 092 616 192	27 428,0	15,17 %	121,35 %	3
25	Normaali	7 200 571 392	22 603,4	100,00 %	100,00 %	4
25	Klusteroitu p_brandin pohjalta	7 200 571 392	79 934,8	100,00 %	353,64 %	5

Kuva 15. SSB Q1.1 mittaustulokset.

Tuloksista voidaan todeta ainakin seuraavat asiat: Skaalan 0,1 datasetissä taulu, joka on klusteroitu d\_yearin pohjalta, ei ole optimaalinen tallennustapa. Se on normaaliakin taulua

hitaampi vaihtoehto, molempien kuluttaessa kuitenkin saman verran tavuja. Sen sijaan ositettu taulu ja ositettu sekä klusteroitu taulu ovat normaalia taulua tehokkaampia tallennustapoja skaalan 0,1 datasetissä huolimatta alle 50 Mt:n ositusten ko'oista.

Skaalojen 1 ja 10 tulokset ovat samanlaisia ja näiden tapauksessa vuokaavio ohjaa optimaaliseen raportointitaulun tallennustapaan. Näissä skaaloissa myös klusteroinnin hyödytkin alkavat näkymään, sen käyttäessä saman verran tavuja kuin ositettu taulu sekä ositettu ja klusteroitu taulu. Tämä siis viittaisi siihen, että klusteroidussakin taulussa saattaa olla taulun minimikoon raja, ennen kuin sitä kannattaa harkita. Klusteroitu taulu on kuitenkin näistä kolmesta vaihtoehdosta hitain, vahvistaen Costan et al. (2019, s. 34–36) johtopäätöstä siitä, että ositus on klusterointia tehokkaampaa.

Skaalan 25 tuloksia analysoidessa huomataan, että vuokaavio ohjaa jälleen väärään vaihtoehtoon, sillä ositettu ja klusteroitu taulu on näistä tehokkain. Kaikki taulut prosessoivat jälleen saman verran tavuja, mutta ositettu ja klusteroitu taulu on näistä kaikista nopein. Jossain kohtaa kulkee siis raja, kun hyödytään vielä hienojakoisemmasta datasta kuin mitä pelkkä ositus voi tarjota. Tulosten perusteella se on jossain 4 934,22 Mt:n ja 12 358,22 Mt:n välissä, jotka ovat skaalan 10 ja 25 taulun ositusten koot. Vuokaavio vaatii siis vielä hienosäätöä, sillä se ohjasi optimaaliseen tallennustapaan vain kahdessa tapauksessa neljästä.

### **Kysymys Q1.2**

Kysymys Q1.2 ei suodata dataa d\_yearin eikä p\_brandin pohjalta, kuvastaen tilannetta, jossa raportointitaulua ei suodateta raportointisovelluksen toimesta tai analyysikohdassa ei löytynyt soveltuvia attribuutteja ositukseen tai klusterointiin. Tällaisissa tapauksissa vuokaavio ohjaa aina taulun tallentamiseen normaalina tauluna eli sen pitäisi olla optimaalinen tallennustapa. Liitteessä 3 korostetaan tähän kysymykseen liittyvää vuokaavion polkua. Kaikkien tallennustapojen oletetaan prosessoivan saman verran tavuja, mutta normaalin taulun oletetaan olevan näistä nopein. Alla olevassa kuvassa 16 on Q1.2 mittaustulokset, jotka todettiin kuitenkin virheellisiksi.

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
0,1	Ositettu (d_year) ja klusteroitu (p_brand)	10 485 760	143,6	35,71 %	269,93 %	1
0,1	Klusteroitu p_brandin pohjalta	29 360 128	51,8	100,00 %	97,37 %	2
0,1	Klusteroitu d_yearin pohjalta	29 360 128	52,4	100,00 %	98,50 %	3
0,1	Normaali	29 360 128	53,2	100,00 %	100,00 %	4
0,1	Ositettu d_yearin pohjalta	29 360 128	118,6	100,00 %	222,93 %	5
1	Ositettu (d_year) ja klusteroitu (p_brand)	44 040 192	523,2	15,27 %	34,71 %	1
1	Klusteroitu d_yearin pohjalta	44 040 192	839,0	15,27 %	55,66 %	2
1	Ositettu d_yearin pohjalta	288 358 400	996,6	100,00 %	66,11 %	3
1	Normaali	288 358 400	1 507,4	100,00 %	100,00 %	4
1	Klusteroitu p_brandin pohjalta	288 358 400	1 749,4	100,00 %	116,05 %	5
10	Ositettu (d_year) ja klusteroitu (p_brand)	437 256 192	4 890,6	15,19 %	69,97 %	1
10	Klusteroitu d_yearin pohjalta	437 256 192	7 020,8	15,19 %	100,45 %	2
10	Normaali	2 879 389 696	6 989,6	100,00 %	100,00 %	3
10	Ositettu d_yearin pohjalta	2 879 389 696	9 175,8	100,00 %	131,28 %	4
10	Klusteroitu p_brandin pohjalta	2 879 389 696	16 285,8	100,00 %	233,00 %	5
25	Ositettu (d_year) ja klusteroitu (p_brand)	1 092 616 192	11 302,8	15,17 %	76,80 %	1
25	Klusteroitu d_yearin pohjalta	1 092 616 192	21 121,0	15,17 %	143,52 %	2
25	Normaali	7 200 571 392	14 716,8	100,00 %	100,00 %	3
25	Ositettu d_yearin pohjalta	7 200 571 392	22 046,6	100,00 %	149,81 %	4
25	Klusteroitu p_brandin pohjalta	7 200 571 392	45 003,4	100,00 %	305,80 %	5

Kuva 16. SSB Q1.2 tulokset ennen yearmonthnum-rajauksen poistoa.

Tuloksia katsoessa huomataan, että ositettu ja klusteroitu taulu sekä skaaloilla 1, 10 ja 25 d\_yearin pohjalta klusteroitu taulu prosessoivat normaaliin tauluun nähden huomattavasti vähemmän tavuja. Analysoinnin jälkeen syyksi paljastui se, että kysymyksessä on year-monthnum-rajaus, joka viittaa tietyn vuoden kuukauteen. Vaikka kyselyä ei suodatettu klusteroidun attribuutin (d\_year) pohjalta, niin BigQuery löysi kaikki tarvittavat rivit skannaamalla vain osan klustereista, prosessoiden vähemmän tavuja normaaliin tauluun verrattuna. Tätä ei kuitenkaan haluttu tällä kysymyksellä testata, vaan tarkoitus oli testata sellaisia tapauksia, joissa raportointitaulua ei suodateta minkään attribuutin pohjalta. Näin ollen Q1.2 tuloksien mittaus suoritettiin uudelleen sen jälkeen, kun kysymyksestä oli poistettu year-monthnum-rajaus. Tämän jälkeen tulokset näyttävät kuvan 17 mukaisilta:

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
0,1	Normaali	24117248	177,4	100,00 %	100,00 %	1
0,1	Klusteroitu p_brandin pohjalta	24117248	203,2	100,00 %	114,54 %	2
0,1	Klusteroitu d_yearin pohjalta	24117248	214,8	100,00 %	121,08 %	3
0,1	Ositettu d_yearin pohjalta	24117248	442,8	100,00 %	249,61 %	4
0,1	Ositettu (d_year) ja klusteroitu (p_brand)	24117248	443,2	100,00 %	249,83 %	5
1	Normaali	240123904	2885,8	100,00 %	100,00 %	1
1	Klusteroitu d_yearin pohjalta	240123904	3169,6	100,00 %	109,83 %	2
1	Ositettu d_yearin pohjalta	240123904	3235	100,00 %	112,10 %	3
1	Ositettu (d_year) ja klusteroitu (p_brand)	240123904	3247	100,00 %	112,52 %	4
1	Klusteroitu p_brandin pohjalta	240123904	3783,8	100,00 %	131,12 %	5
10	Normaali	2400190464	17555,8	100,00 %	100,00 %	1
10	Klusteroitu d_yearin pohjalta	2400190464	29241	100,00 %	166,56 %	2
10	Klusteroitu p_brandin pohjalta	2400190464	30339	100,00 %	172,82 %	3
10	Ositettu d_yearin pohjalta	2400190464	30565	100,00 %	174,10 %	4
10	Ositettu (d_year) ja klusteroitu (p_brand)	2400190464	36018,8	100,00 %	205,17 %	5
25	Normaali	5999951872	28193	100,00 %	100,00 %	1
25	Ositettu d_yearin pohjalta	5999951872	71920,2	100,00 %	255,10 %	2
25	Klusteroitu d_yearin pohjalta	5999951872	84089,4	100,00 %	298,26 %	3
25	Klusteroitu p_brandin pohjalta	5999951872	86784,2	100,00 %	307,82 %	4
25	Ositettu (d_year) ja klusteroitu (p_brand)	5999951872	90243,2	100,00 %	320,09 %	5

Kuva 17. SSB Q1.2 tulokset yearmonthnum-rajauksen poiston jälkeen.

Nyt tulokset näyttävät odotetun mukaisilta, sillä kaikki tallennustavat prosessoivat saman verran tavuja. Normaali taulu on näistä kaikista myös nopein tallennustapa kaikilla skaaloilla. Tämä tarkoittaa sitä, että mikäli raportointitaulua ei tulla suodattamaan minkään attribuutin pohjalta, osituksesta, klusteroinnista ja näiden yhdistelmästä on vain haittaa. Tämän kysymyksen osalta vuokaavio onnistui siis ohjaamaan optimaalisen tallennustavan valintaan.

### Kysymys Q2.3

Kysymys Q2.3 suodattaa dataa p\_brandin eli korkean kardinaliteetin attribuutin pohjalta, mutta ei d\_yearin pohjalta. Kysymys ryhmittelee dataa d\_yearin pohjalta, mutta tästä ei kirjallisuuskatsauksen perusteella ole suorituskykyhyötyjä. Tällaisessa tapauksessa vuokaavio ohjaa tallentamaan raportointitaulun aina klusteroituna tauluna korkeakardinaliteettisen attribuutin pohjalta. Liite 4 korostaa Q2.3 osalta vuokaavion suosittamaa polkua. P\_brandin pohjalta klusteroidun taulun odotetaan aina prosessoivan vähimmän määrän tavuja sekä olevan nopein vaihtoehto. Alla on Q2.3 mittaustulokset:

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
0,1	Klusteroitu p_brandin pohjalta	26 214 400	147,6	100,00 %	89,67 %	1
0,1	Normaali	26 214 400	164,6	100,00 %	100,00 %	2
0,1	Klusteroitu d_yearin pohjalta	26 214 400	196,6	100,00 %	119,44 %	3
0,1	Ositettu d_yearin pohjalta	26 214 400	341,4	100,00 %	207,41 %	4
0,1	Ositettu (d_year) ja klusteroitu (p_brand)	26 214 400	405,8	100,00 %	246,54 %	5
1	Klusteroitu p_brandin pohjalta	10 485 760	604,0	4,00 %	21,52 %	1
1	Ositettu (d_year) ja klusteroitu (p_brand)	33 554 432	381,4	12,80 %	13,59 %	2
1	Normaali	262 144 000	2 806,8	100,00 %	100,00 %	3
1	Ositettu d_yearin pohjalta	262 144 000	3 314,8	100,00 %	118,10 %	4
1	Klusteroitu d_yearin pohjalta	262 144 000	3 390,6	100,00 %	120,80 %	5
10	Klusteroitu p_brandin pohjalta	10 485 760	5 238,0	0,40 %	32,58 %	1
10	Ositettu (d_year) ja klusteroitu (p_brand)	32 505 856	415,6	1,24 %	2,59 %	2
10	Normaali	2 616 197 120	16 078,4	100,00 %	100,00 %	3
10	Klusteroitu d_yearin pohjalta	2 616 197 120	30 597,4	100,00 %	190,30 %	4
10	Ositettu d_yearin pohjalta	2 616 197 120	31 374,8	100,00 %	195,14 %	5
25	Klusteroitu p_brandin pohjalta	10 485 760	17 027,0	0,16 %	74,07 %	1
25	Ositettu (d_year) ja klusteroitu (p_brand)	30 408 704	396,2	0,47 %	1,72 %	2
25	Normaali	6 536 822 784	22 988,8	100,00 %	100,00 %	3
25	Ositettu d_yearin pohjalta	6 536 822 784	71 420,4	100,00 %	310,68 %	4
25	Klusteroitu d_yearin pohjalta	6 536 822 784	89 048,2	100,00 %	387,36 %	5

Kuva 18. SSB Q2.3 tulokset.

Tuloksia analysoidessa voidaan nopealla vilkaisulla todeta, että vuokaavio onnistui aina ohjaamaan optimaaliseen tallennustapaan, sillä p\_brandin pohjalta klusteroitu taulu on aina optimaalinen vaihtoehto. Tarkemman analyysin jälkeen esille nousee kuitenkin muutama huomionarvoinen asia. Ensimmäiseksi, 0,1 skaalalla klusteroitu taulu ei vähentänyt prosessoitujen tavujen määrää ollenkaan. Tällä skaalalla taulu kuitenkin suoriutui nopeimmin, ansaiten parhaan sijoituksen. Klusteroinnilla vaikuttaisi siis olevan joku minimiraja sille, milloin siitä alkaa saamaan hyötyjä irti laskentatehon käytön vähentämisessä. Toisena huomiona: klusteroitu taulu ei ollut millään muulla skaalalla nopein, paitsi pienimmällä 0,1 skaalalla. Ositettu ja klusteroitu taulu oli kaikilla muilla skaaloilla nopein vaihtoehto, nopeuseron vain kasvaessa mitä suurempaan skaalaan mennään. Tämä ei kuitenkaan ole optimaalinen vaihtoehto siitä syystä, että klusteroidut taulut prosessoivat vähemmän tavuja, kuin ositetut ja klusteroidut.

Kuten aiemmin on jo kuvattu, niin tässä testauksessa on prosessoitujen tavujen pienin määrä ensisijainen optimaalisuuden mittari, nopeuden tullessa mukaan vasta tasatilanteissa. Tulokset näyttäisivät tämän kysymyksen osalta kuitenkin erilaisilta sijoitusten puolesta, mikäli prosessoitujen tavujen määrälle ja nopeudelle annettaisiin jonkinlaiset painoarvot.

### Muokattu kysymys Q3.2

Muokattu kysymys Q3.2 suodattaa dataa sekä d\_yearin että p\_brandin pohjalta. Toisin sanoen suodatuksia on sekä matala- että korkeakardinaliteettisen attribuutin pohjalta ja näissä

tilanteissa vuokaavio ohjaa tallentamaan raportointitaulun klusteroituna tauluna matalakardinaliteettisen attribuutin pohjalta, kun osituksen kooksi jää alle 50 Mt, muutoin ohjataan aina ositettuun ja klusteroituun tauluun. Tähänkin kysymykseen pätee taulukko 6:n mukaiset raportointitaulun ositusten koot, eli 0,1 skaalan vuokaavion ohjaama optimaalisin raportointitaulun tallennustapa on d\_yearin pohjalta klusteroitu taulu ja kaikissa muissa tapauksissa ositettu ja klusteroitu taulu. Alla olevassa kuvassa 19 on kysymyksen Q3.2 tulokset:

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
mt_ssb_0_1	Klusteroitu d_yearin pohjalta	40894464	118,2	88,64 %	149,24 %	1
mt_ssb_0_1	Ositettu d_yearin pohjalta	41943040	282,4	90,91 %	356,57 %	2
mt_ssb_0_1	Ositettu (d_year) ja klusteroitu (p_brand)	41943040	306,8	90,91 %	387,37 %	3
mt_ssb_0_1	Normaali	46137344	79,2	100,00 %	100,00 %	4
mt_ssb_0_1	Klusteroitu p_brandin pohjalta	46137344	118,2	100,00 %	149,24 %	5
mt_ssb_1	Klusteroitu p_brandin pohjalta	10485760	469,6	2,30 %	20,51 %	1
mt_ssb_1	Ositettu (d_year) ja klusteroitu (p_brand)	49283072	267,6	10,81 %	11,69 %	2
mt_ssb_1	Klusteroitu d_yearin pohjalta	371195904	2316,8	81,38 %	101,21 %	3
mt_ssb_1	Ositettu d_yearin pohjalta	415236096	2472,6	91,03 %	108,01 %	4
mt_ssb_1	Normaali	456130560	2289,2	100,00 %	100,00 %	5
mt_ssb_10	Klusteroitu p_brandin pohjalta	10485760	4224,2	0,23 %	27,74 %	1
mt_ssb_10	Ositettu (d_year) ja klusteroitu (p_brand)	48234496	266	1,06 %	1,75 %	2
mt_ssb_10	Klusteroitu d_yearin pohjalta	3708813312	23826,8	81,50 %	156,45 %	3
mt_ssb_10	Ositettu d_yearin pohjalta	4146069504	24311	91,11 %	159,63 %	4
mt_ssb_10	Normaali	4550819840	15229,4	100,00 %	100,00 %	5
mt_ssb_25	Klusteroitu p_brandin pohjalta	12582912	13138,8	0,11 %	71,29 %	1
mt_ssb_25	Ositettu (d_year) ja klusteroitu (p_brand)	46137344	233,2	0,41 %	1,27 %	2
mt_ssb_25	Klusteroitu d_yearin pohjalta	9273606144	65949,4	81,50 %	357,85 %	3
mt_ssb_25	Ositettu d_yearin pohjalta	10366222336	59238,4	91,11 %	321,43 %	4
mt_ssb_25	Normaali	11378098176	18429,6	100,00 %	100,00 %	5

Kuva 19. Muokatun SSB Q3.2 tulokset.

Tarkastellaan ensin skaalaa 0,1, sillä tällä on vuokaavion mukaan eri optimaalinen vaihtoehto kuin muilla. D\_yearin pohjalta klusteroitu taulu on tulosten mukaan optimaalisin tallennustapa, sillä se prosessoi pienimmän määrän tavuja, joten vuokaavio onnistui ohjaamaan parhaaseen tallennustapaan. Huomionarvoista on kuitenkin jälleen kerran nopeuteen liittyvät tekijät: kaikki ositetut ja klusteroidut taulut olivat pienessä raportointitaulussa hitaampia kuin normaali taulu. Prosessoitujen tavujen määrää ja nopeutta painoavottaessa tulokset voisivat taas olla eriävät. Klusteroitu taulu ei myöskään prosessoinut kovin merkittävää määrää vähemmän tavuja kuin normaali taulu.

Siirtyessä muiden skaalojen analysointiin huomataan nopealla silmäyksellä, että vuokaavio ei onnistunut lopuissa tapauksissa ohjaamaan optimaalisimpaan tallennustapaan, joka oli oletusarvoisesti sekä ositettu että klusteroitu taulu. P\_brandin pohjalta klusteroitu taulu prosessoi kaikista vähiten tavuja skaaloilla 1, 10 ja 25 ollen tämän testauksen optimaalinen

vaihtoehto. Tämä tallennustapa prosessoii vain noin neljäosan seuraavaksi parhaan, eli ositetun ja klusteroidun taulun tavuista. Nopeutta tarkastellessa huomataan, että ositettu ja klusteroitu taulu oli kuitenkin kaikista nopein jokaisella skaalalla. Jälleen kerran mittareiden painotuksilla tulokset näyttäisivät erilaisilta, joka on hyvä pitää mielessä. Kokonaisuutta tarkastellessa vuokaavio ohjasi oikeaan suuntaan ja auttoi optimoinnissa, mutta ei kuitenkaan tuottanut optimaalista tulosta. Muokkauksia siis tarvitaan.

#### 4.4 Vuokaavion jatkokehitys

Vuokaavion testauksessa havaittiin muutamia tapauksia, jolloin vuokaavio ei ohjannut optimaaliseen tallennustapaan, joten vuokaaviota tulee muokata mahdollisuuksien mukaan siten, että havaituissakin tapauksissa vuokaavio löytäisi parhaan vaihtoehdon. DSR metodologian prosessin mallin mukaisesti palataan siis aktiviteettiin 3 eli ”Suunnittele ja kehitä”. Havaittuja puutteita olivat:

1. Q1.1 skaala 0,1 – vuokaavio ohjasi klusteroituun tauluun matalakardinaliteettisen attribuutin pohjalta, optimaalisen ollessa ositettu taulu.
2. Q1.1 skaala 25 – vuokaavio ohjasi ositettuun tauluun, optimaalisen ollessa ositettu ja klusteroitu taulu
3. Q3.2 skaalat 1–25 – vuokaavio ohjasi ositettuun ja klusteroituun tauluun, optimaalisen ollessa klusteroitu taulu korkeakardinaliteettisen attribuutin pohjalta

Yllä mainittujen kohtien lisäksi huomionarvoisina asioina todettiin, että pienillä datamäärillä klusterointi ei välttämättä vähennä prosessoitujen tavujen määrää (Q1.1, Q1.2 virheelliset tulokset & Q2.3), mutta se saattaa lisätä vähäisesti nopeutta (Q1.2 virheelliset tulokset & Q2.3 tulokset). Myös Q3.2 osalta klusteroinnin hyödyt alkoivat korostua vasta suuremmissa datamäärissä.

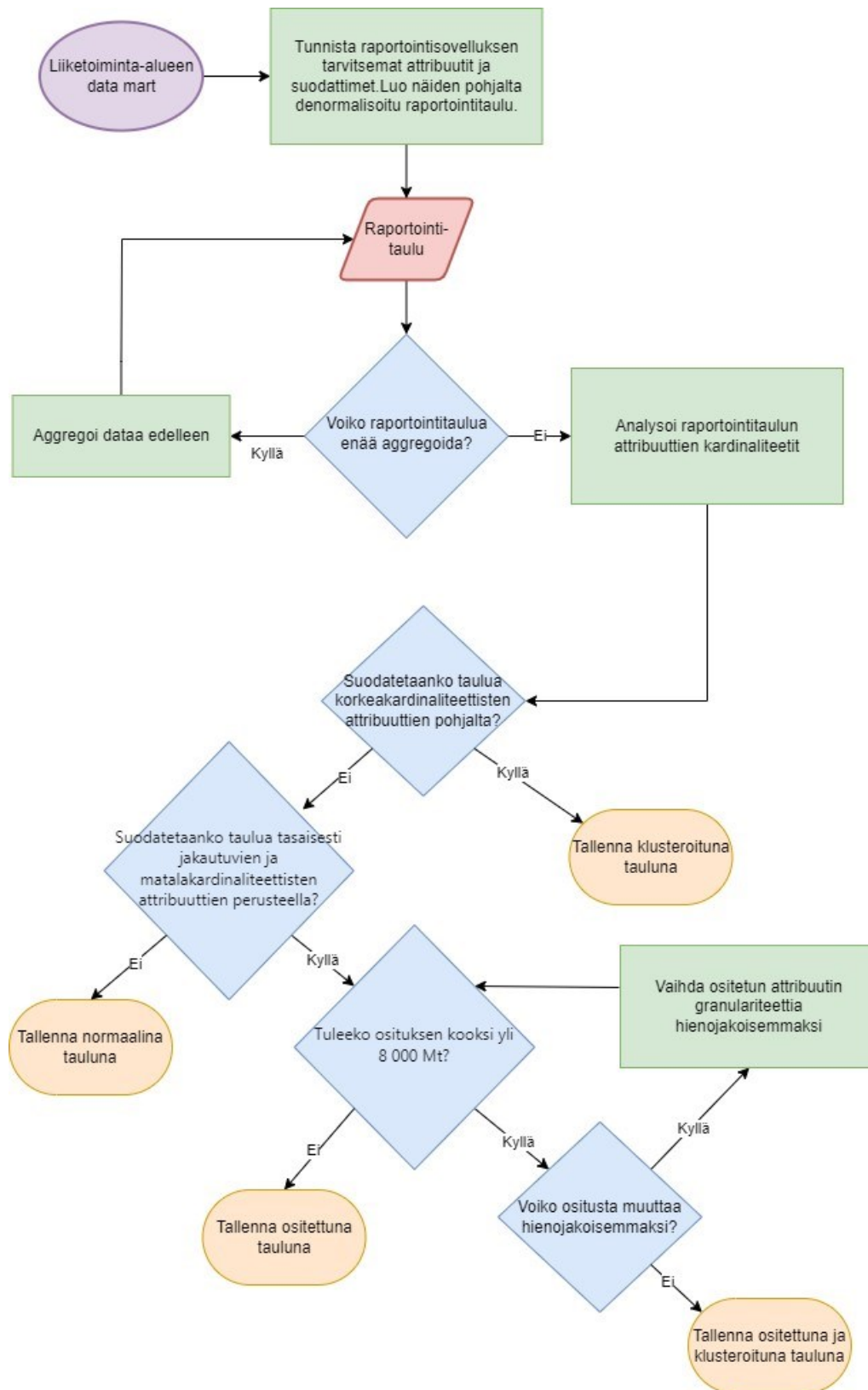
Q1.1:n tuloksista huomattiin, että suodattaessa dataa ainoastaan matalakardinaliteettisen attribuutin pohjalta, klusteroitu taulu ei ollut koskaan optimaalinen vaihtoehto, joten tämä vaihtoehto pitää muokata vuokaaviosta pois. Tämän lisäksi todettiin, että osituksesta oli hyötyä ja oli myös optimaalinen vaihtoehto siitä huolimatta, että osituksen koko oli alle 50 Mt. Vuokaaviota tulee siis muokata siten, että mikäli raportointitaulua suodatetaan matalakardinaliteettisen attribuutin pohjalta, se ei koskaan ohjaa klusteroidun taulun luomiseen. Tämän lisäksi ositusten koon tarkastelu näyttää olevan turha, joten poistetaan tämäkin päätöskohta vuokaaviosta. Näillä muutoksilla ratkaistaan ongelmakohta 1.

Q1.1 tuloksia analysoidessa todettiin myös, että jossain kohtaa kulkee raja, jolloin tarvitaan hienojakoisempaa datan tallennusta. Vaikka Q1.1 ei suodattanut dataa muun kuin matalakardinaliteettisen attribuutin pohjalta, niin ositettu ja klusteroitu taulurakenne oli tästä huolimatta tehokkain vaihtoehto. Tämä selittyy sillä, että hienojakoisempi tallennustapa mahdollisti hajautetumman tietojenkäsittelyn, joka johti nopeampaan käsittelyyn. Vuokaaviota tulee siis muokata siten, että se ohjaa jonkun raja-arvon ylittäessä hienojakoisempaan tallennustapaan. Tämä tallennustapa voi olla osituksen muuttaminen hienojakoisemmaksi (kuten vuodesta kuukauteen) tai sitten osituksen lisäksi klusterointi. Raja-arvo on tutkimustulosten perusteella jossain 4 934,22 Mt:n ja 12 358,22 Mt:n välillä, kun tarkastellaan ositusten kojoja. Tarkka raja-arvo vaihtelee oletettavasti tilanteen mukaan, joten vuokaavioon asetetaan raja-arvoksi 8 000 Mt, joka on 12 358,22 Mt:n ja 4 943,22 Mt:n keskellä. Vuokaaviota muokataan myös siten, että se ohjaa ensisijaisesti osittamaan hienojakoisemmin, mikäli mahdollista ja jos tämä ei onnistu, niin vuokaavio ohjaa osittamaan ja klusteroimaan. Näillä muutoksilla ratkaistaan ongelmakohta 2.

Q3.2 tuloksista huomattiin, että prosessoitujen tavujen osalta klusterointi on aina paras vaihtoehto, vaikka taulua suodatettiin sekä korkea- että matalakardinaliteettisen attribuutin pohjalta. Tähän löytyi selitys Costan et al. (2019, s. 34) tutkimuksesta, jossa todettiin, että hyödyntäessä sekä ositusta että klusterointia, suoritetaan ositus ensin ja klusterointi jälkeensä. Tämä tarkoittaa käytännössä sitä Q3.2:n osalta, että data on ensin ositettu `d_yearin` mukaan, jonka jälkeen klusteroitu `p_brandin` mukaan. `P_brandin` arvot saattavat ja todennäköisesti ovatkin eri osituksissa ennen niiden klusterointia ja tämä aiheuttaa ylimääräisen datan prosessoinnin, kun dataa suodatetaan tietyn `p_brandin` arvon pohjalta klusterointiin verrattuna. Ollessaan pelkästään klusteroituna, suodatuksenmukainen `p_brand`-arvo löytyy suoraan yhdestä tallennuslohkosta, näin ollen prosessoiden vähemmän tavuja. Vuokaavion muokkausten osalta tämä tarkoittaa sitä, että mikäli dataa suodatetaan korkeakardinaliteettisen attribuutin pohjalta, kannattaa se aina tallentaa ainoastaan klusteroituun tauluun, kun ensisijaisesti optimoidaan prosessoitujen tavujen kannalta – niissäkin tapauksissa, jossa taulua suodatettaisiin korkeakardinaliteettisen attribuutin lisäksi matalakardinaliteettisellä attribuutilla.

Epäjohdonmukaisuuksia tuloksiin aiheuttaa pienimmän skaalan tulokset klusteroinnin osalta, joka hankaloittaa yksiselitteisesti optimaalisimman tavan valintaa. Q3.2 osalta pienimmän skaalan optimaalinen raportointitaulu on klusteroitu taulu, mutta `d_yearin` pohjalta

eikä p\_brandin pohjalta. Tämän kysymyksen kohdalla klusterointi d\_yearin pohjalta vähensi prosessoitujen tavujen määrää jopa pienimmällä skaalalla. Tämä on ristiriidassa Q1.2 ja Q2.3 tulosten kanssa, joissa klusterointi pienimmällä skaalalla ei vähentänyt prosessoituja tavuja. Tähän epä johdonmukaisuuteen ei löytynyt selitystä tai ratkaisuja, sillä tuloksia katsoessa tietyn raja-arvon asettaminenkaan klusteroinnille ei ratkaise tätä ongelmaa yksiselitteisesti. Tämä ongelma siis vain tiedostetaan ja vuokaavion muokkaus vielä paremmaksi jää mahdollisen jatkotutkijan tehtäväksi. Tutkimus tarvitsee lisämittauksia pienempien data-määrien osalta, jotta löydetään selitys tälle ongelmalle. Näiden havaintojen ja muutosten pohjalta luotiin vuokaavion toinen versio, joka on esitelty kuvassa 20:



Kuva 20. Vuokaavion toinen versio.

#### 4.5 Vuokaavion jatkotestaus ja -arviointi

Vuokaavion toisen version ollessa valmis, siirryttiin DSR:n aktiviteetteihin 4 ja 5, eli ”demo” ja ”arvio”. Vuokaavion testaukseen käytettiin samoja SSB:n kysymyksiä kuin aiemminkin, joten ensimmäisen suorituskykytestauksen tulokset ovat suurilta osin edelleen päteviä. Vuokaavio on kuitenkin muuttunut, joten on syytä arvioida kysymyskohtaisesti uudelleen, että mihin tallennustapaan vuokaavio ohjaa ja varmistaa, että vuokaaviota muokattiin onnistuneesti ohjaamaan optimaaliseen tallennustapaan.

Päivitetessä vuokaaviossa on nyt myös uusi päätöskohta, jonka pohjalta vuokaavio voi ohjata vaihtamaan ositetun attribuutin granulariteettia hienojakoisemmaksi. Tämä vaikuttaa tutkimuksen kysymykseen Q1.1, jossa dataa suodatetaan matalakardinaliteettisen attribuutin pohjalta. Kysymyksen Q1.1 osalta pitää suorittaa lisäsuorituskykytestausta, jotta saadaan vuokaavion ohjaaman hienojakoisemmin ositetun taulun suorituskyky myös testattua. Vuokaaviota käytettiin uudelleen ohjaamaan kysymyksien Q1.1, Q1.2, Q2.3 ja muokatun Q3.2 raportointitaulujen optimaaliseen tallennustapaan ja nämä kysymyskohtaiset vuokaaviot on esitelty liitteissä 6–9.

Aloitetaan ensin niiden kysymysten vuokaavioiden arvioinnilla, jotka eivät vaadi lisäsuorituskykytestausta, eli kysymyksien Q1.2, Q2.3 ja Q3.2 vuokaavioiden ohjaamista tallennustavoista. Nämä löytyvät liitteistä 7–9. Q1.2 tuloksien pohjalta normaali taulu oli aina optimaalinen vaihtoehto kaikissa skaaloissa, joten vuokaavion toisen version pitää myös ohjata tähän tallennusmuotoon. Koska raportointitaulua ei kysymyksessä suodateta korkea- eikä matalakardinaliteettisen attribuutin pohjalta, ohjaa vuokaavio tallentamaan raportointitaulun normaalina tauluna, kuten liitteestä 7 näkyy. Q1.2 osalta vuokaavio ohjaa siis edelleen optimaaliseen tallennustapaan.

Q2.3 ja Q3.2 osalta molempia kysymyksiä suodatetaan korkeakardinaliteettisen attribuutin pohjalta, joten vuokaavio ohjaa näiden kysymysten osalta aina klusteroituun tauluun korkeakardinaliteettisen attribuutin pohjalta. Liitteissä 8 ja 9 on kuvattu vuokaavioiden ohjaamat tallennuspolut näiden kysymysten osalta. Tuloksiin peilattaessa tätä tallennustavan valintaa voidaan todeta, että vuokaavio ohjaa optimaaliseen tallennustapaan kaikissa tapauksissa paitsi yhdessä. Poikkeama on Q3.2 tuloksissa pienimmässä skaalassa, jossa tulosten mukaan optimaalinen tallennustapa on kylläkin klusteroitu taulu, mutta `d_yearin` pohjalta eli matalakardinaliteettisen attribuutin pohjalta. Tämä poikkeama on siis kappaleen 4.4 lopussa

mainittu tulosten epäjohdonmukaisuus, jonka pohjalta ei saatu muokattua vuokaaviota yksiselitteisesti ohjaamaan aina optimaaliseen tallennustapaan.

Q1.1 osalta tehtiin jatkosuorituskykytestausta eri osituksen hienojakoisuuksilla, sillä vuokaavio ohjaa nyt muuttamaan ositetun attribuutin granulariteettia hienojakoisemmaksi, mikäli ositusten kooksi tulisi yli 8 000 Mt. Aiemmin suoritetuissa suorituskykytestauksissa raportointitaulu ositettiin d\_yearin eli vuoden pohjalta, mutta tämän hienojakoisempi ositus joko kuukauden tai päivän pohjalta on mahdollista. Hienojakoisempien ositetujen taulujen ositusten koot arvioitiin samaa periaatetta käyttäen kuin aiemminkin, eli kuukauden pohjalta ositetun taulun osituksen koko laskettiin: taulun koko / 7 (datan vuosien määrä) / 12 ja päivän pohjalta edellä mainittua kaavaa jatkettiin vielä jakamalla se 30:llä. Alla olevassa taulukossa 7 on laskennalliset ositusten koot eri skaaloilla:

Taulukko 7. Ositusten koot eri granulariteeteilla skaaloittain.

Skaala	Taulun koko (Mt)	Osituksen koko, vuosi (Mt)	Osituksen koko, kuukausi (Mt)	Osituksen koko, päivä (Mt)
<b>0,1</b>	341,27	48,75	4,06	0,14
<b>1</b>	3 440,64	491,52	40,96	1,37
<b>10</b>	34 539,52	4 934,22	411,19	13,71
<b>25</b>	86 507,52	12 358,22	1029,85	34,33

Taulukosta 7 nähdään, että skaalan 25 osituksen koko vuositasolla ylittää vuokaavion raja-arvon 8 000 Mt, joten tämän skaalan osalta vuokaavion mukainen optimaalinen tallennustapa on eri kuin muilla skaaloilla. Liitteestä 6 näkee vuokaavion toisen version ohjaamat tallennustavat Q1.1 osalta. Liitteestä nähdään, että polkuja on kaksi, mutta ne päätyvät samaan tallennustapaan eli ositettiin tauluun, mutta eri granulariteeteilla ositettuna. Skaalojen 0,1–10 vuokaavion ohjaama tallennustapa on ositettu taulu d\_yearin pohjalta ja skaalan 25 osalta muutetaan vuositaso hienojakoisemmaksi, eli kuukauden pohjalta ositetuksi tauluksi.

Kuukauden pohjalta ositettua taulua ei ollut vielä suorituskykytestattu, joten uusi raportointitaulu luotiin, joka tallennettiin kuukausitason ositettuna tauluna. Samaan aikaan luotiin myös päivätason ositettu taulu testausta varten samalla vaivalla, vaikka vuokaavio ei niin hienojakoiseksi ohjaakaan tallentamaan. Uusille raportointitauluille tehtiin samat suorituskykytestit kuin muillekin tauluille ja uusien taulujen tulokset liitettiin osaksi muita Q1.1 kysymyksen tuloksia. Päivitetyt tulokset löytyvät kuvasta 21:

Skaala	Raportointitaulu	Prosessoidut tavut	Aika (millisek.)	Prosessoidut tavut suhteessa normaaliin tauluun	Aika suhteessa normaaliin tauluun	Sijoitus omassa skaalassa
0,1	Ositettu d_yearin pohjalta	10 485 760	79,2	35,71 %	53,66 %	1
0,1	Ositettu (d_year) ja klusteroitu (p_brand)	10 485 760	81,4	35,71 %	55,15 %	2
0,1	Ositettu d_monthin pohjalta	10 485 760	365,6	35,71 %	247,70 %	3
0,1	Ositettu päivän pohjalta	10 485 760	8 534,4	35,71 %	5782,11 %	4
0,1	Normaali	29 360 128	147,6	100,00 %	100,00 %	5
0,1	Klusteroitu d_yearin pohjalta	29 360 128	171,6	100,00 %	116,26 %	6
0,1	Klusteroitu p_brandin pohjalta	29 360 128	209,8	100,00 %	142,14 %	7
1	Ositettu d_yearin pohjalta	44 040 192	532,6	15,27 %	23,22 %	1
1	Ositettu (d_year) ja klusteroitu (p_brand)	44 040 192	631,0	15,27 %	27,50 %	2
1	Ositettu d_monthin pohjalta	44 040 192	977,2	15,27 %	42,59 %	3
1	Klusteroitu d_yearin pohjalta	44 040 192	1 086,2	15,27 %	47,35 %	4
1	Ositettu päivän pohjalta	44 040 192	9 191,0	15,27 %	400,62 %	5
1	Normaali	288 358 400	2 294,2	100,00 %	100,00 %	6
1	Klusteroitu p_brandin pohjalta	288 358 400	2 914,2	100,00 %	127,03 %	7
10	Ositettu d_yearin pohjalta	437 256 192	4 902,2	15,19 %	33,42 %	1
10	Ositettu (d_year) ja klusteroitu (p_brand)	437 256 192	5 160,2	15,19 %	35,18 %	2
10	Ositettu d_monthin pohjalta	437 256 192	5 817,4	15,19 %	39,66 %	3
10	Klusteroitu d_yearin pohjalta	437 256 192	9 507,6	15,19 %	64,83 %	4
10	Ositettu päivän pohjalta	437 256 192	11 762,2	15,19 %	80,20 %	5
10	Normaali	2 879 389 696	14 666,6	100,00 %	100,00 %	6
10	Klusteroitu p_brandin pohjalta	2 879 389 696	27 212,6	100,00 %	185,54 %	7
25	Ositettu päivän pohjalta	1 092 616 192	11 280,6	15,17 %	49,91 %	1
25	Ositettu (d_year) ja klusteroitu (p_brand)	1 092 616 192	12 805,6	15,17 %	56,65 %	2
25	Ositettu d_monthin pohjalta	1 092 616 192	14 094,8	15,17 %	62,36 %	3
25	Ositettu d_yearin pohjalta	1 092 616 192	23 778,2	15,17 %	105,20 %	4
25	Klusteroitu d_yearin pohjalta	1 092 616 192	27 428,0	15,17 %	121,35 %	5
25	Normaali	7 200 571 392	22 603,4	100,00 %	100,00 %	6
25	Klusteroitu p_brandin pohjalta	7 200 571 392	79 934,8	100,00 %	353,64 %	7

Kuva 21. SSB Q1.1 jatkettut tulokset.

Tuloksista huomataan, että skaaloilla 0,1–10 d\_yearin pohjalta ositettu taulu on edelleen optimaalinen vaihtoehto ja tänne vuokaaviokin ohjaa. Skaalan 25 optimaalinen taulu on tulosten mukaan päivän pohjalta ositettu taulu. Vuokaavio ohjaisi tallentamaan kuukauden pohjalta ositettuun tauluun, jonka sijoitus on kolme. Suurimman skaalan taulun kanssa vuokaavio ei siis edelleenkään onnistu ohjaamaan optimaaliseen vaihtoehtoon. Vuokaavion ensimmäinen versio ohjasi vuoden pohjalta ositettuun tauluun ja tähän verrattuna kuukauden pohjalta ositettu taulu on tehokkaampi vaihtoehto, mutta ei tosiaan optimaalinen. Kehitystä kuitenkin tapahtui ensimmäiseen versioon verrattuna.

Vuokaavion kehitys päätettiin jättää tämän tutkimuksen osalta tähän, sillä tuloksiin ollaan riittävän tyytyväisiä. Vuokaavio ohjaa suurimmassa osassa tapauksista tässä tutkimuksessa määriteltyyn optimaaliseen vaihtoehtoon ja niissäkin tapauksissa, joissa valinta ei ole optimaalinen, onnistuu vuokaavio tarjoamaan normaalia taulua tehokkaamman ratkaisun. Vuokaavion raja-arvoja olisi mahdollista tarkentaa ja todetut ongelmatkohdat olisivat todennäköisesti ratkaistavissa laajemmilla suorituskykytestauksilla, mutta tämä jääköön mahdollisten jatkotutkimusten tehtäväksi.

## 5 Johtopäätökset

Tämän työn tutkimus sai alkunsa toimeksiantajaorganisaation tarpeesta pitää uuden pilvi-alustan päälle rakennettavan tietovaraston käytön kulut matalina. Tämän lisäksi organisaation raportointi- ja analytiikkatiimin tehtäväalue laajeni koskemaan datan muuntoa, joka kattaa raportointisovellusten käyttämien raportointitaulujen luonnin. Tätä uutta tehtäväaluetta varten tarvittiin tiimille tietoa, kuinka datan muunto-operaatiot tehdään tehokkaasti ja kuinka raportointitaulut tallennetaan optimaalisesti niille tehtyjen kyselyjen kannalta. Tiedon pohjalta organisaatio pystyy kasaamaan organisaation käyttötarkoitusta varten sopivan ohjeistuksen, kuinka kokonaisuutena data kannattaa muuntaa ja tallentaa.

Tutkimuksen tavoitteet määriteltiin seuraavalla tavalla: ”Tavoitteena on selvittää pilvitietovarastossa sijaitsevien raportointitaulujen optimointitavat kulutehokkuuden ja nopeuden osalta sekä muunto- että tallennusvaiheessa kyselyjen kannalta. Tämän lisäksi tavoitteena on luoda oikeaan tallennustapaan ohjaava ratkaisu, joka helpottaa käyttäjää optimoinnissa.”. Johdannossa tutkimusta jäsentelemään asetettiin kolme tutkimuskysymystä, joihin tutkimuksessa löydettiin vastaukset. Alla on tutkimuskysymykset lihavoituna ja näiden alla vastaukset kysymyksiin:

*Millä tavoin SQL-pohjaista datan muuntoa voidaan optimoida pilvitietovarastossa?*

Yleisesti tehokkain tapa on pyrkiä kirjoittamaan SQL-lauseet siten, että ne käsittelevät mahdollisimman vähän rivejä ja sarakkeita. Datan käsittelyn vähentyessä pienenevät myös käsittelyyn käytetyt resurssit ja siihen kuluva aika. SQL-kielessä on lukuisia eri tapoja vähentää käytettyjen resurssien määrää ja tästä syystä optimoinnin keinot ovat aina kysely- tai tietohakukohtaisia.

Tutkimuksessa kasattiin 19 kappaletta jo aiemmin tutkittua SQL-kielen optimointikeinoa taulukkoon. Optimointikeinot valittiin sillä näkökulmalla, että ne kaikki liittyvät datan muuntovaiheeseen ja myös se mielessä pitäen, että ne tehostavat SQL-lausetta, joko käytettyjen resurssien tai nopeuden kannalta. Tämän lisäksi pyrittiin painottamaan kolumnaarisessa tietovarastossa toimivia optimointikeinoja, sillä nämä saattavat erota rivipohjaisesti dataa käsittelevistä optimointikeinoista. Taulukkoon kasattujen optimointikeinojen lisäksi löydettiin yksi hajautettua tietojenkäsittelyä hyödyntävien järjestelmien (jota

pilvitietovarastot usein ovat) optimointikeino, broadcast join, joka vähentää datan vaihdantaa kyselyn käsittelijöiden välillä.

*Millä tavoin pilvitietovaraston raportointitaulujen tallennusta voidaan optimoida kyselyjen kannalta?*

Kirjallisuuskatsauksen pohjalta löytyi aluksi neljä eri tallennusvaiheen optimointikeinoa, jotka tehostavat raportointitauluihin tehtyjä kyselyitä. Nämä neljä ovat raportointitaulun indeksointi, materialisointi näkymäksi, ositus ja klusterointi. Tutkimuksen kolumnaarisii tietovarastoihin rajaaminen sulki näistä indeksoinnin pois. Materialisoitua näkymää ei myöskään pidetä tässä tutkimuksessa tallennuksen optimointina, sillä näihin perinteisesti turvautaan hätäratkaisuna, kun jotkut olemassa olevat näkymät aiheuttavat paljon siirräntää.

Näin ollen pilvitietovaraston raportointitaulujen tallennusta voidaan tehostaa osituksen, klusteroinnin ja näiden yhdistämisen keinoin. Molempien keinojen tehokkuushyödyt pohjautuvat siirräntän vähentämiseen, joka johtaa vähempien resurssien käytön kautta kulu- ja nopeushyötyihin.

Yllä mainittujen datan fyysisten tallennuskeinojen lisäksi kirjallisuuskatsauksessa selvisi, että denormalisoidun tietomallin käyttäminen ja mahdollisimman aggregoitu datan muoto tehostaa raportointitauluihin tehtäviä kyselyitä. Näiden mallien käyttö tallennuksen yhteydessä auttaa optimoinnissa.

*Kuinka valitaan optimaalinen raportointitaulun tallennustapa?*

Optimaalisen raportointitaulun tallennustavan valintaan vaikuttavat raportointitaululle tehtävät suodatuksat, suodatuksen kohteena olevien attribuuttien kardinaliteetit sekä taulun ositusten fyysinen koko. Optimaalisen tallennustavan valintaan liittyy useita kysymyksiä ja päätöskohtia. Tätä prosessia helpottamaan luotiin tutkimuksessa vuokaavio, joka ohjaa tehostetumpaan raportointitaulun tallennustapaan.

Kokonaisuudessaan tutkimuksen tavoitteisiin päästiin, sillä tutkimuskysymyksiin löydettiin vastauksat ja oikeaan tallennustapaan ohjaava ratkaisu saatiin luotua vuokaavion muodossa. Tutkimuksen artefaktista eli vuokaaviosta ei kuitenkaan saatu tehtyä sellaista, että se ohjaisi joka tilanteessa optimaalisen tallennustavan vaihtoehtoon, josta jääkin seuraaville tutkijoille jatkotutkimusaiheita. Toimeksiantajaorganisaatiolle saatiin tutkimuksen myötä tarjottua

keinoja kulujen kurissa pitämiseen ja organisaatio pystyy tutkimuksen pohjalta muotoilemaan haluamansa ohjeistuksen raportointitaulujen muuntoon ja tallentamiseen.

### 5.1 Tutkimuksen luotettavuus ja rajoitteet

Tutkimuksen kokonaisuuden luotettavuutta heikentää osittain lyhyeksi jäänyt lähdeluettelo. Useampien lähteiden käyttö olisi tuonut tutkimuksen luotettavuuden tasoa korkeammalle. Toisaalta taas luotettavuutta nostaa kuitenkin suhteellisen tuoreet lähteet; yli puolet lähteenä käytetyistä kirjoista ja tieteellisistä artikkeleista on julkaistu viiden vuoden sisällä. Tutkimuksen merkittävimmissä osa-alueissa on myös käytetty useampia lähteitä, joka hieman paikkaa lähteiden vähyyttä.

Tutkimuksen empiirisen osan luotettavuutta pyrittiin parantamaan kuvaamalla testijärjestelyt mahdollisimman tarkasti, jonka lisäksi tutkimuksessa käytettiin sellaisia työkaluja ja aineistoa, jotka ovat kaikkien saatavilla. Tutkimuksen suorituskykytestaukseen liittyvä osio on tarpeen mukaan toistettavissa. Tämän lisäksi itse mittaustulosten luotettavuutta pyrittiin parantamaan toistamalla mittauksia ja ottamalla tuloksista keskiarvo, jotta yksi mahdollisesti kauemmin kestänyt tulos ei olisi vääristänyt tuloksia.

Tutkimusta rajoittavina tekijöinä voidaan pitää vuokaavion osalta sitä, että sen käyttöä on testattu ainoastaan yhdellä pilvitietovarastolla. Toiset pilvitietovarastot saattavat käsitellä dataa eri tavalla, johtaen erilaisiin suorituskykymittauksen tuloksiin ja tätä kautta rajoittaen vuokaavion toimivuutta ainoastaan GCP:n päällä toimivaan BigQueryyn. Tämän lisäksi ennen vuokaavion suurempaa käyttöönottoa se tarvitsee vielä jatkosuorituskykytestauksia sekä pienemmillä, että suuremmilla tietomäärillä. Tässä tutkimuksessa käytettiin vain neljän eri skaalan datasettejä, jota en koe riittäväksi määräksi sille, että vuokaavion voisi todeta yleisesti toimivaksi.

Vuokaavion osalta on myös hyvä muistaa empiirisessä osuudessa määritelty, tässä tutkimuksessa pidetty optimaalisuus. Optimaalisena tallennustapana pidetään tässä tutkimuksessa sellaista, joka prosessoi ensisijaisesti vähimmän määrän tavuja ja on näin ollen halvin käyttää. Mikäli optimaalisuus kulujen ja nopeuden pohjalta määritellään eri tavalla, eivät empiirisen osan tulokset ole enää päteviä. Osa näistä tutkimusta rajoittavista tekijöistä voidaan kääntää jatkotutkimusehdotuksiksi, joita on esitelty seuraavassa ja viimeisessä kappaleessa.

## 5.2 Jatkotutkimusehdotukset

Aiheen tutkimusta voisi jatkaa moneen eri suuntaan, sillä tutkimuksen aihepiiriin kuului pilvitietovaraston kulutehokkuus, datan muuntamisen optimointi sekä raportointitaulun tallennustavan optimointi kyselyjen kannalta. Pilvitietovaraston kulurakennetta voisi tutkia yleisemmällä otteella, sillä nyt keskityttiin vain datan käsittelyn loppupään optimointiin. Olisi kiinnostavaa analysoida, että mihin tietovarastoinnin vaiheeseen liittyy eniten kuluja ja etsiä optimointimahdollisuuksia muistakin tietovarastoinnin vaiheista.

Datan muuntamisen osalta tässä tutkimuksessa koottiin yhteen aiempien tutkimusten SQL:n optimoinnin keinoja, mutta näiden toimivuutta ei suorituskykymitattu. Vaikka osassa tutkimuksista, joiden pohjalta optimointikeinot koottiin, oli suorituskykymittauksia ja laskettu auki optimointikeinojen tehostusmittareita, olisi mielenkiintoista tietää kuinka listatut optimointikeinot vertautuvat keskenänsä. Jatkotutkimuksen myötä näistä voisi saada listan tehokkaimmista optimointikeinoista.

Vuokaavioon liittyviä jatkotutkimusehdotuksia on myös muutamia. Ensimmäiseksi, tässä tutkimuksessa luotua vuokaaviota testattiin ainoastaan yhden pilvitietovaraston kanssa; olisi kiinnostavaa tietää, että toimiiko vuokaavio yhtä hyvin muita pilvitietovarastoja tai kolumnaarisia tietovarastoja käyttäessä. Jatkotutkimusta voisi tehdä esimerkiksi siten, että toistaa vuokaavion käytön simuloinnin kappaleessa kuvatut suorituskykymittaukset toisilla pilvi-alustoilla tai tietovarastoissa.

Vuokaavioon liittyviä raja-arvoja voisi myös tarkentaa jatkotutkimusten avulla suorittamalla suorituskykymittaukset vielä pienemmillä ja suuremmilla taulujen ko'oilta. Tätä kautta voisi löytyä alaraja raportointitaulun koolle, jolloin yli päätänsä kannattaa harkita muita tallennusmuotoja, kuin normaali taulu. Nykyinen vuokaavion yläraja on myös todennäköisesti epätarkka, sillä se arvioitiin vain olemassa olevan, liian suurilla kokohypyillä olevien taulujen pohjalta.

Myös itse vuokaavion suorituskykymittaukseen liittyviä kysymyksiä voisi laajentaa jatkotutkimuksessa. Tässä tutkimuksessa valittiin neljä 13:sta SSB:ssä esitetystä kysymyksestä ja vuokaaviota optimoitiin näihin kysymyksiin liittyvien tulosten pohjalta. Jatkotutkimuksessa voisi ottaa mukaan myös loput kysymykset, joka toisi vuokaavion toimivuudelle lisää luotettavuutta tai nostaisi esille uusia ongelmakohtia vuokaavion toimivuudessa.

Viimeisenä ja mielenkiintoisimpana jatkotutkimusideana ehdotan, että jonkin yrityksen raportointitauluja optimoitaisiin case-tutkimuksen tyylisesti vuokaavion avulla ja mitattaisiin optimoinnin vaikutusta tietovaraston käytön kuluihin ja nopeuteen.

## Lähteet

- Abadi, D., Boncz, P. & Harizopoulos, S. (2009) Column-oriented database systems. *Proceedings of the VLDB Endowment* 2(2). pp. 1664–1665. DOI: 10.14778/1687553.1687625
- Amazon Web Services (2023) Amazon Redshift Pricing. Luettu 9.1.2024. Saatavissa: <https://aws.amazon.com/redshift/pricing/>.
- Beaulieu, A. (2020) Learning SQL: Generate, Manipulate, and Retrieve Data. O'Reilly Media, Inc.
- Bell, F., Chirumamilla, R., Joshi, B.B., Lindstrom, B., Soni, R. & Videkar, S. (2021) Snowflake Essentials : Getting Started with Big Data in the Cloud. Apress.
- Bhagat, V. & Gopal, A. (2012) Comparative Study of Row and Column Oriented Database. *Fifth International Conference on Emerging Trends in Engineering and Technology*. pp. 196-201. DOI: 10.1109/ICETET.2012.56.
- Braake, P. (2021) Data Modeling for Azure Data Services. Packt Publishing.
- Brocke, J., Hevner, A. & Maedche, A. (2020) Design Science Research : Cases. Springer.
- Celko, J. (2015) Joe Celko's Sql for Smarties : Advanced SQL Programming. Morgan Kaufmann.
- Celko, J. (2010) Joe Celko's SQL for Smarties: Advanced SQL Programming. Elsevier Science.
- Costa, E., Costa, C. & Santos, M.Y. (2019) Evaluating partitioning and bucketing strategies for Hive-based Big Data Warehousing systems. *Journal of Big Data* 6(1): 34. DOI: 10.1186/s40537-019-0196-1
- Debarros, A. (2022) Practical SQL: A Beginner's Guide to Storytelling with Data, 2nd Edition. No Starch Press.
- Densmore, J. (2021) Data Pipelines Pocket Reference. O'Reilly Media, Inc.
- Dibouliya, A. (2023). Modern data warehouse & how is it accelerating digital transformation. *International Journal of Advance Research, Ideas and Innovations in Technology*, 9(2). Luettu 27.5.2024. Saatavissa: <https://www.ijariit.com/manuscript/modern-data-warehouse-how-is-it-accelerating-digital-transformation/>
- Emergen Research (2023) Global Cloud Data Warehouse Market Size to Reach USD 43.55 Billion in 2032. Lehdistöiedote, 9.8.2023. Luettu 21.5.2024. Saatavissa: <https://www.emergenresearch.com/press-release/global-cloud-data-warehouse-market>
- Foxwell, H. (2020) Creating good data: a guide to dataset structure and data representation. Apress.
- Gartner (2023a) Gartner Forecasts Worldwide Banking and Investment Services IT Spending to Reach \$652 Billion in 2023. Luettu 30.8.2023. Saatavissa: <https://www.gartner.com/en/newsroom/press-releases/2023-06-21-gartner-forecasts-worldwide-banking-and-investment-services-it-spending-to-reach-652-billion-in-2023>.

Gartner (2023b) Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach nearly \$600 Billion in 2023. Luettu 30.8.2023. Saata-  
vissa: <https://www.gartner.com/en/newsroom/press-releases/2023-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023>.

Google (2024a) Information Schema Jobs. Luettu 16.1.2024. Saata-  
vissa: <https://cloud.google.com/bigquery/docs/information-schema-jobs>.

Google (2024b) BigQuery Pricing. Luettu 9.1.2024 Saata-  
vissa: <https://cloud.google.com/bigquery/pricing#storage>.

Harrington, J.L. (2009) Relational Database Design and Implementation : Clearly Explained. Morgan Kaufmann/Elsevier.

Hayath, T.M., Usman, K., Shafiulla, M. & Dadapeer (2023) An Overview of SQL Optimization Techniques for Enhanced Query Performance. *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* 1-5. DOI: 10.1109/ICDCECE57866.2023.10151265

Jukic, N., Jukic, B., Sharma, A., Nestorov, S. & Korallus, B.A. (2017) Expediting analytical databases with columnar approach. *Decision Support Systems* 95: 61-81. DOI: 10.1016/j.dss.2016.12.002

Kahn, M.G., Mui, J.Y., Ames, M.J., Yamsani, A.K., Pozdeyev, N., Rafaels, N. & Brooks, I.M., (2022) Migrating a research data warehouse to a public cloud: challenges and opportunities. *Journal of the American Medical Informatics Association*, 29(4), pp.592-600. DOI: 10.1093/jamia/ocab278

Lahtonen, T. (2002) Sql. Jyväskylä: Docendo.

Lightstone, S.S., Nadeau T. & Teorey, T.J. (2010) Physical Database Design. Morgan Kaufmann.

Luu, H. (2018) Beginning Apache Spark 2 with Resilient Distributed Datasets, Spark SQL, Structured Streaming and Spark Machine Learning Library. Apress.

Microsoft (2024) Azure Synapse Analytics Pricing. Luettu 9.1.2024. Saata-  
vissa: <https://azure.microsoft.com/en-us/pricing/details/synapse-analytics/#overview>.

Morales-Morales, M., Durán-Cazar, J.W., Tandazo-Gaona, E. & Santiago, M.C. (2019) Performance of columnar database. *Ingenius* (22): 47. DOI: 10.17163/ings.n22.2019.05

Morton, A. (2022) Mastering Snowflake Solutions : Supporting Analytics and Data Sharing. Apress.

Mucchetti, M. (2020) BigQuery for Data Warehousing: Managed Data Analysis in the Google Cloud. Apress.

Myalapalli, V.K. & Chakravarthy, A.S.N. (2016) Revamping SQL Queries for Cost Based Optimization. *2016 International Conference on Circuits, Controls, Communications and Computing (I4C)* 1-6. DOI: 10.1109/CIMCA.2016.8053263

Myalapalli, V.K. & Savarapu, P.R. (2014) High Performance SQL. *2014 Annual IEEE India Conference (INDICON)* 1-6. DOI: 10.1109/INDICON.2014.7030467

- Peppers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee S. (2007) A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24(3): 45-77. DOI: 10.2753/MIS0742-1222240302
- Phillips, D. & Sundstrom, D. (2010) Star Schema Benchmark Dbgen. Luettu 22.4.2024. Saatavissa: <https://github.com/electrum/ssb-dbgen>
- Reis, J. & Housley, M. (2022) Fundamentals of Data Engineering. O'Reilly Media, Inc.
- Sanchez, J. (2016) A review of star schema benchmark. *arXiv Preprint* arXiv:1606.00295.
- Santoso, L. (2017) Data Warehouse with Big Data Technology for Higher Education. *Procedia Computer Science* (124): 93-99. DOI: 10.1016/j.procs.2017.12.134
- Serra, J. (2024) Deciphering Data Architectures. O'Reilly Media, Inc.
- Sherman, R. & Imhoff, C. (2015) Business Intelligence Guidebook : From Data Integration to Analytics. Morgan Kaufmann.
- Snowflake Inc. (2024) Snowflake Pricing Options. Luettu 9.1.2024. Saatavissa: <https://www.snowflake.com/en/data-cloud/pricing-options/>.
- Thallam (2020a) BigQuery Explained: Working with Joins, Nested & Repeated Data. Luettu 21.4.2024. Saatavissa: <https://cloud.google.com/blog/topics/developers-practitioners/bigquery-explained-working-joins-nested-repeated-data>.
- Thallam (2020b) BigQuery Explained: Storage Overview, and how to Partition and Cluster Your Data for Optimal Performance. Luettu 9.1.2024. Saatavissa: <https://cloud.google.com/blog/topics/developers-practitioners/bigquery-explained-storage-overview>.
- Vaisman, A. & Zimányi, E. (2022) Data Warehouse Systems : Design and Implementation. Springer.
- Vaisman, A. & Zimányi, E. (2014) Data Warehouse Systems Design and Implementation. Springer Berlin Heidelberg.
- Visweswara, S.P.D., Narechania, A. & Arulraj, J. (2020) SQLCheck: Automated Detection and Diagnosis of SQL Anti-Patterns. *arXiv.Org*. DOI: 10.1145/3318464.3389754
- Wu, Q. (2015) Research on Column-Store Databases Optimization Techniques. *2015 International Conference on Logistics, Informatics and Service Sciences (LISS)* 1-7. DOI: 10.1109/LISS.2015.7369708

## Liite 1. Star Schema Benchmarkin alkuperäiset ja tutkimusta varten uudelleenkirjoitetut kyselyt.

### Star Schema Benchmarkin Q1.1, alkuperäinen

Q1.1 YEAR = 1993, DISCOUNT = 2, QUANTITY = 25, so predicates are d\_year = 1993, lo\_quantity < 25, lo\_discount between 1 and 3.

```
select sum(lo_extendedprice*lo_discount) as revenue
from lineorder, date
where lo_orderdate = d_datekey
and d_year = 1993
and lo_discount between 1 and 3
and lo_quantity < 25;
```

### Star Schema Benchmarkin Q1.2, alkuperäinen

Q1.2 d\_yearmonthnum = 199401, lo\_quantity between 26 and 35, lo\_discount between 4 and 6.

```
select sum(lo_extendedprice*lo_discount) as revenue
from lineorder, date
where lo_orderdate = d_datekey
and d_yearmonthnum = 199401
and lo_discount between 4 and 6
and lo_quantity between 26 and 35;
```

### Star Schema Benchmarkin Q2.3, alkuperäinen

Q2.3 Change p\_category = 'MFGR#12' to p\_brand1 = 'MFGR#2339' and s\_region = 'EUROPE'.

```
select sum(lo_revenue), d_year, p_brand1
from lineorder, date, part, supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_brand1 = 'MFGR#2221'
and s_region = 'EUROPE'
group by d_year, p_brand1
order by d_year, p_brand1;
```

### Star Schema Benchmarkin Q3.2, alkuperäinen

Q3.2 Change restriction to a certain nation, and within that nation, revenue by customer city and supplier city, and year.

```
select c_city, s_city, d_year, sum(lo_revenue) as revenue
from customer, lineorder, supplier, date
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and c_nation = 'UNITED STATES'
and s_nation = 'UNITED STATES'
and d_year >= 1992 and d_year <= 1997
group by c_city, s_city, d_year
order by d_year asc, revenue desc;
```

### Star Schema Benchmarkin Q1.1, tutkimuksessa käytetty

```
SELECT
SUM(extendedprice*discount) AS revenue
FROM `reporting_table`
WHERE
year = 1993
AND discount BETWEEN 1 AND 3
AND quantity < 25
```

### Star Schema Benchmarkin Q1.2, tutkimuksessa käytetty

```
SELECT
SUM(extendedprice*discount) AS revenue
FROM `reporting_table`
WHERE
yearmonthnum = 199401
AND discount BETWEEN 4 AND 6
AND quantity BETWEEN 26 AND 35
```

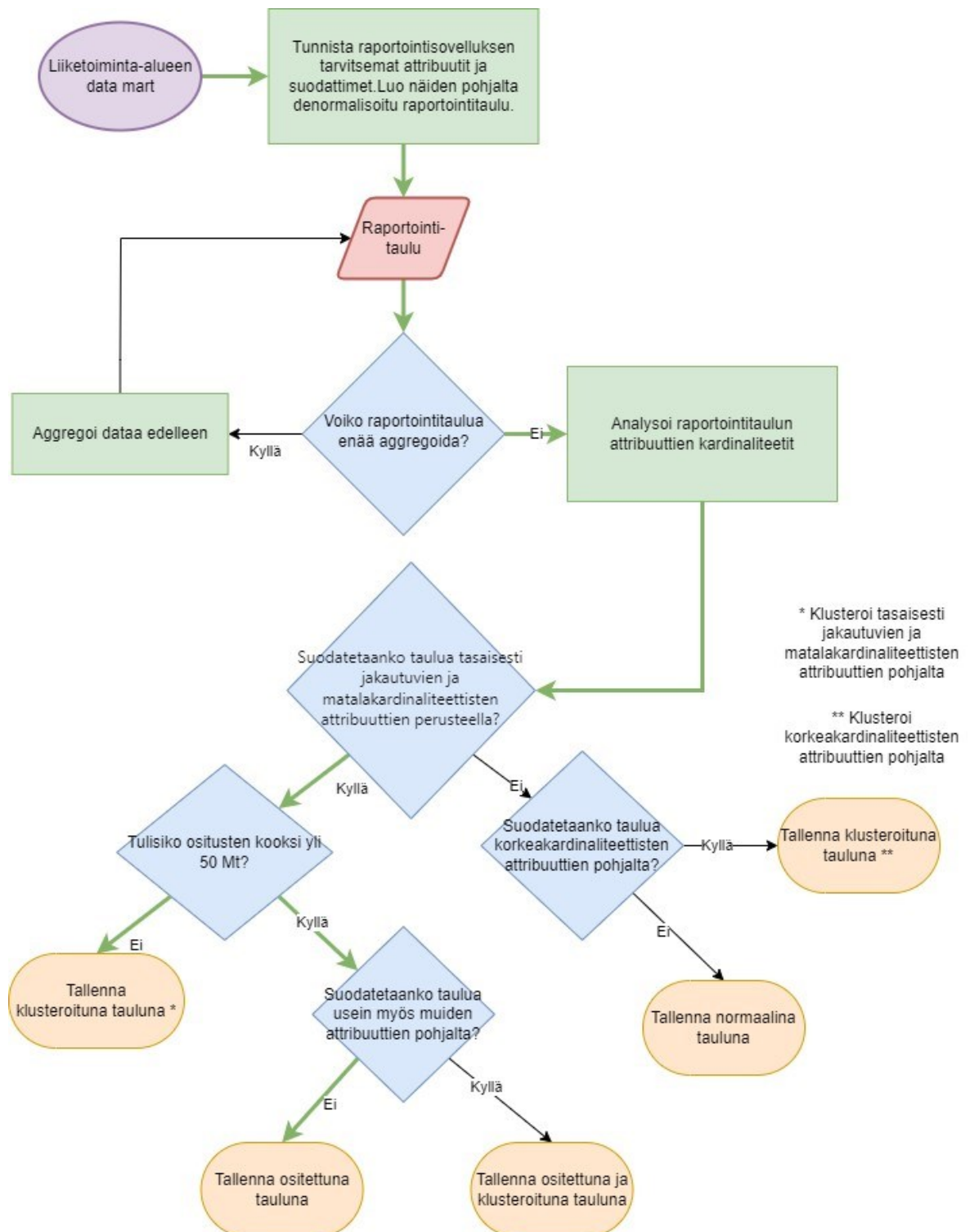
### Star Schema Benchmarkin Q2.3, tutkimuksessa käytetty

```
SELECT
SUM(revenue) AS revenue,
year,
brand
FROM `reporting_table`
WHERE brand = 'MFGR#2221'
AND region = 'EUROPE'
GROUP BY
year,
brand
ORDER BY
year,
brand
```

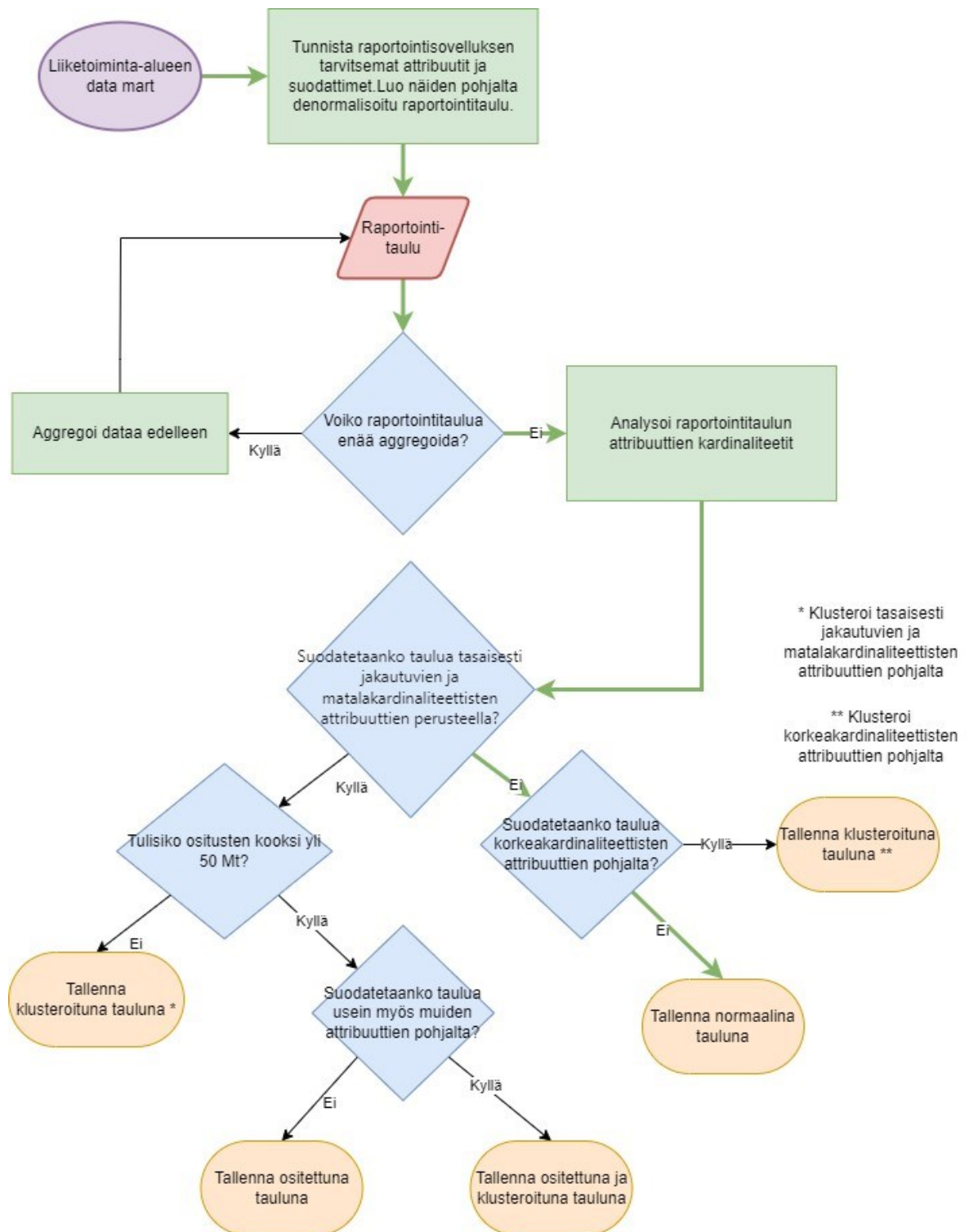
### Star Schema Benchmarkin muokattu Q3.2, tutkimuksessa käytetty

```
SELECT
cust_city,
city,
year,
SUM(revenue) AS revenue
FROM `reporting_table`
WHERE
cust_nation = 'UNITED STATES'
AND brand = 'MFGR#2221'
AND year BETWEEN 1992 AND 1997
GROUP BY
cust_city,
city,
year
```

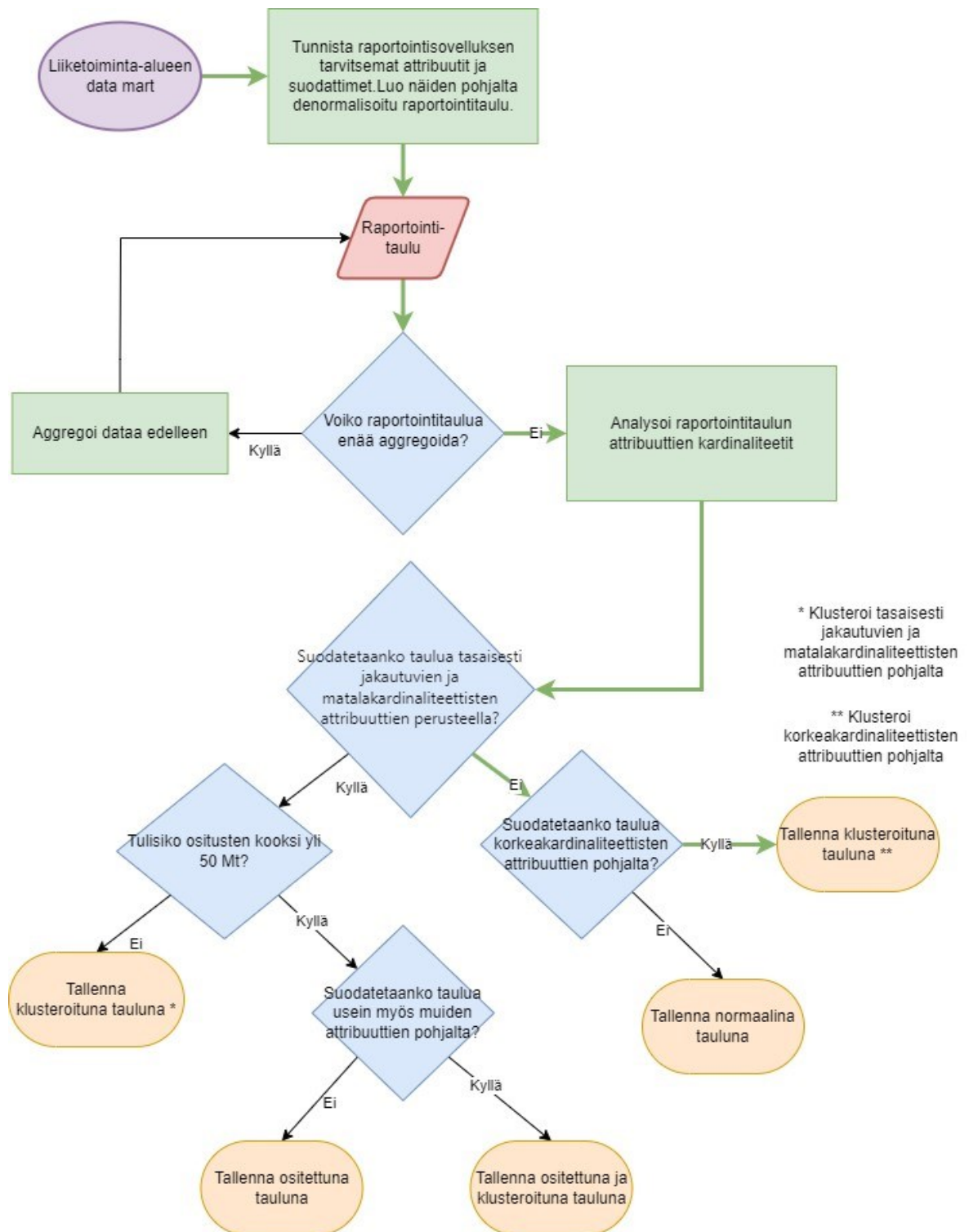
Liite 2. Vuokaavion ensimmäisen version ohjaamat tallennustavat Q1.1 osalta.



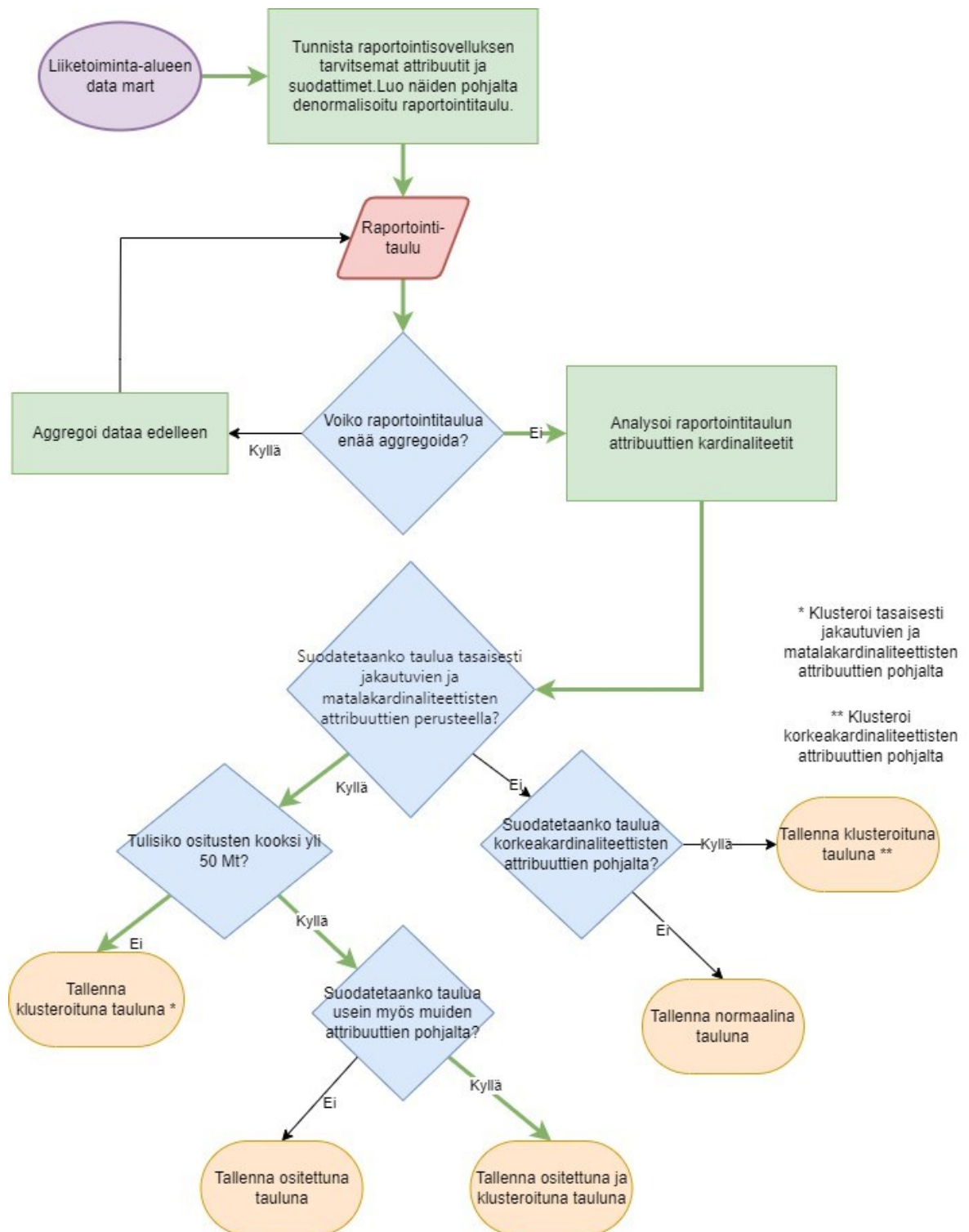
Liite 3. Vuokaavion ensimmäisen version ohjaama tallennustapa Q1.2 osalta.



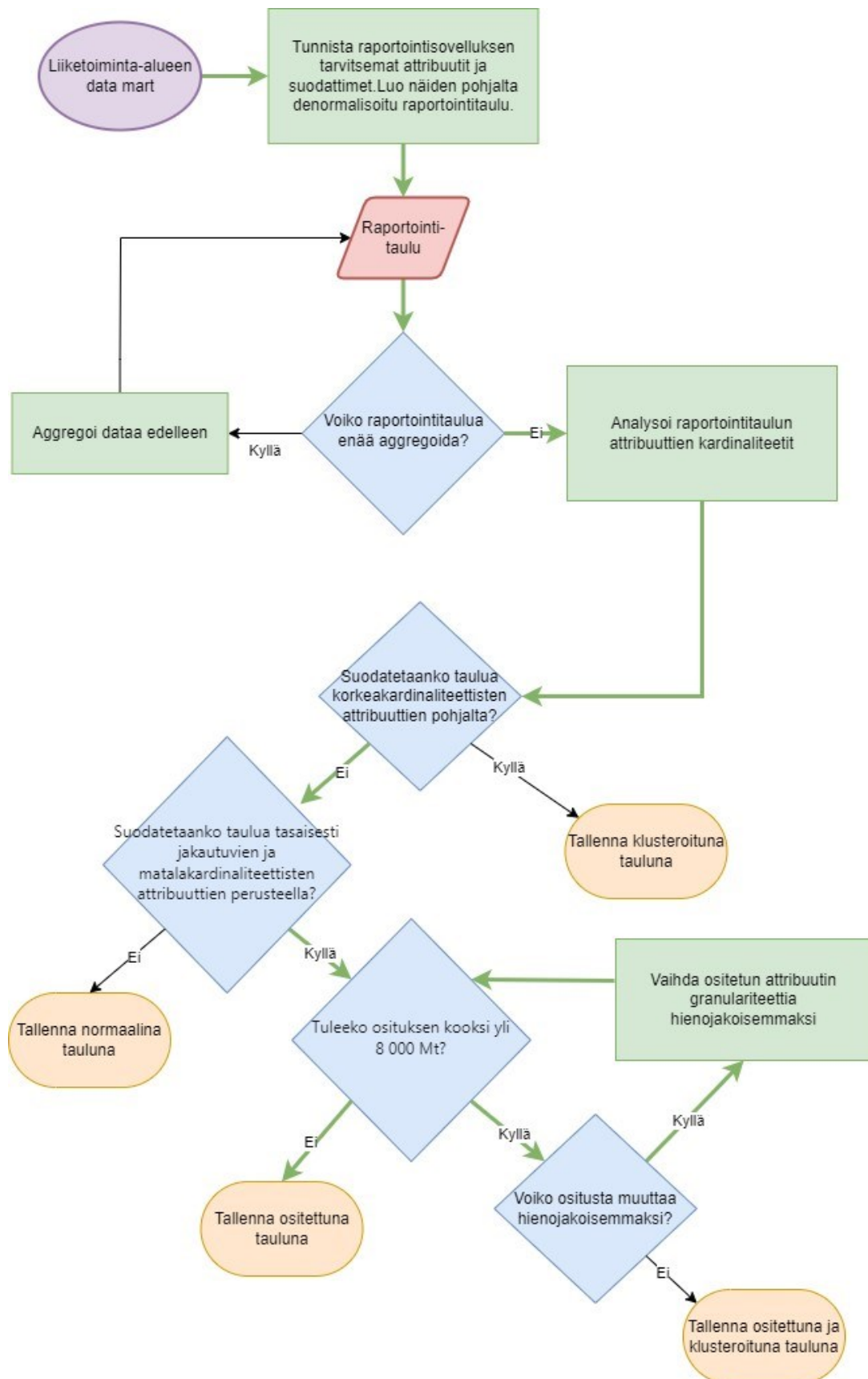
Liite 4. Vuokaavion ensimmäisen version ohjaama tallennustapa Q2.3 osalta.



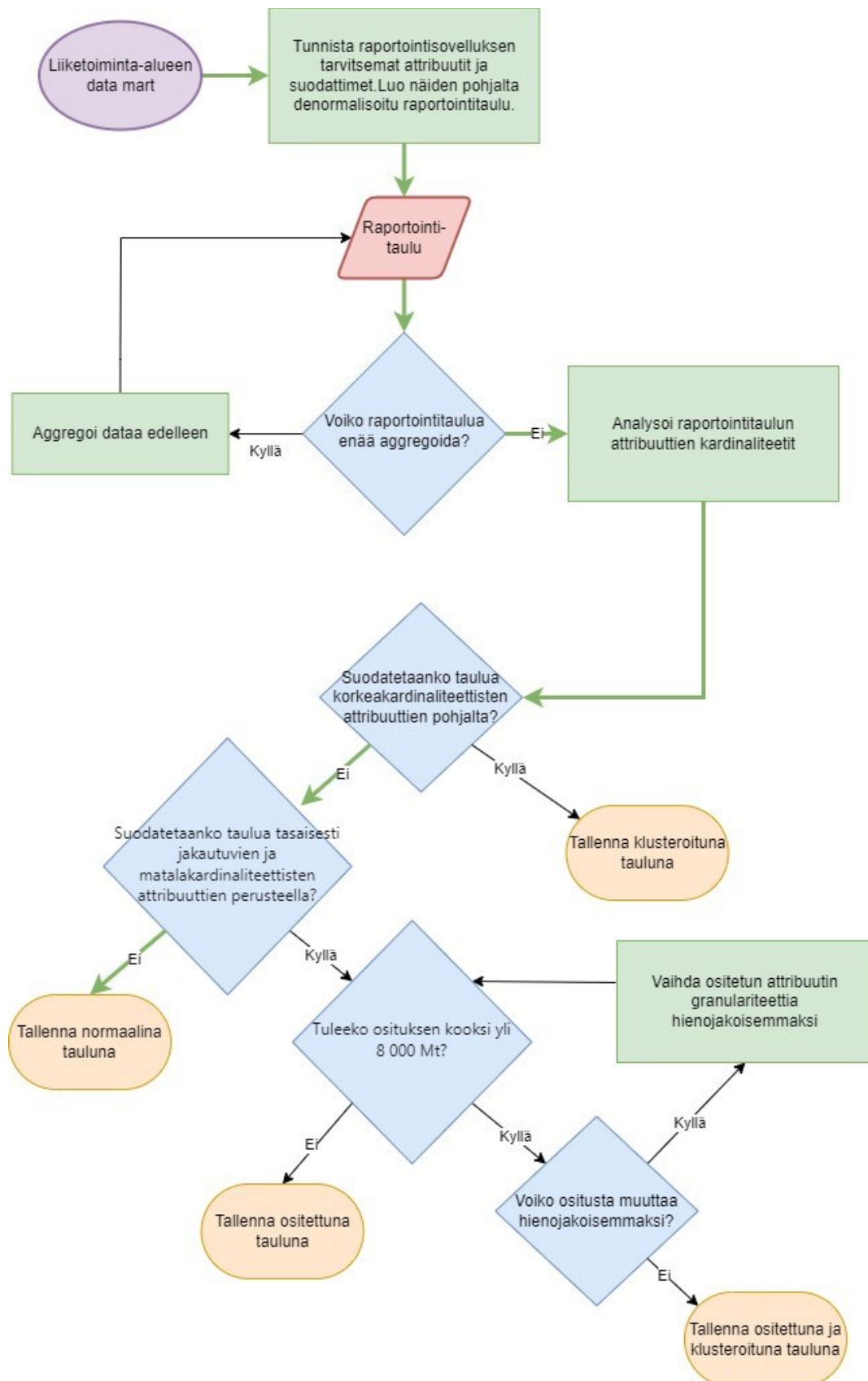
Liite 5. Vuokaavion ensimmäisen version ohjaamat tallennustavat Q3.2 osalta.



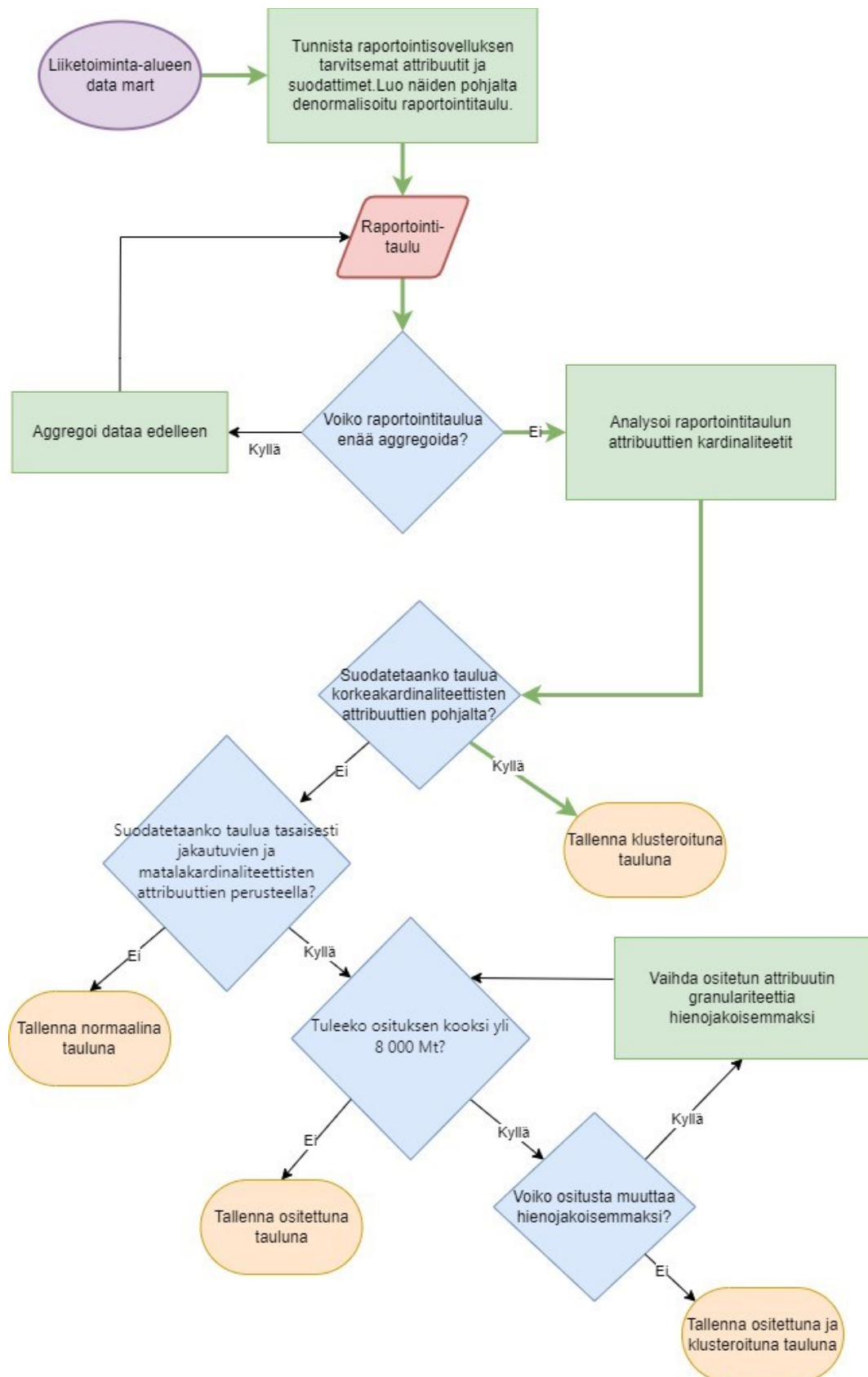
Liite 6. Vuokaavion toisen version ohjaamat tallennustavat Q1.1 osalta.



Liite 7. Vuokaavion toisen version ohjaama tallennustapa Q1.2 osalta.



Liite 8. Vuokaavion toisen version ohjaama tallennustapa Q2.3 osalta.



Liite 9. Vuokaavion toisen version ohjaama tallennustapa Q3.2 osalta.

