Improving traveling habits using an OLAP cube

Development of a business intelligence system

Marcus Hellman
Abstract

The aim of this thesis is to improve the traveling habits of clients using the SpaceTime system when arranging their travels. The improvement of traveling habits refers to lowering costs and emissions generated by the travels.

To do this, a business intelligence system, including an OLAP cube, were created to provide the clients with feedback on how they travel. This to make it possible to see if they are improving and how much they have saved, both in money and emissions. Since these kind of systems often are quite complex, studies on best practices and how to keep such systems agile were performed to be able to provide a system of high quality. During this project, it was found that the pre-study and design phase were just as challenging as the creation of the designed components.

The result of this project was a business intelligence system, including ETL, a Data warehouse, and an OLAP cube that will be used in the SpaceTime system as well as mock-ups presenting how data from the OLAP cube could be presented in the SpaceTime web-application in the future.
Acknowledgements

First, I would like to thank SpaceTime and Dohi for giving me the opportunity to do this thesis. Also, I would like to thank my supervisors Anders Broberg and Kim Nilsson for all the help during this thesis project, providing valuable feedback on my thesis report throughout the project as well as guiding me through the introduction and development of the system. Last, but not least, I would like to thank my lovely future wife Ivana for all the support during this project, pushing me to do my best.
1 Introduction 1
  1.1 Purpose 1
  1.2 Thesis collaboration 2
  1.3 Outline 2

2 Problem description and approach 3
  2.1 Problem statement 3
  2.2 Method 4
    2.2.1 Theoretical part 4
    2.2.2 Practical part 4

3 Background 7
  3.1 Existing SpaceTime system 7
  3.2 Area of interest 9
  3.3 Terminology 9
    3.3.1 OLAP cube structure 10
    3.3.2 Operations 12
    3.3.3 Multidimensional query language 14
    3.3.4 OLAP - Storage modes 15
  3.4 Background summary 15

4 In-depth study 17
  4.1 OLAP cube - Design 17
    4.1.1 Fact table 17
    4.1.2 Dimensions 18
    4.1.3 Data source 19
    4.1.4 Design summary 21
  4.2 Agile Business intelligence 21
    4.2.1 Agile software development 21
1 Introduction

SpaceTime provides a service that give their clients the possibility to arrange their travels in the form of a booking system. The purpose of the system is to optimize travels, in order to decrease costs and emissions, by carpooling or using local transportation. This can only be achieved by changing the behavior of the people in the client organizations. But only providing a system for booking is not enough to make people change, they need to be able to see the change and the possible effects of it. This is why SpaceTime needs to be able to give feedback on the way their clients travel. A good way of doing this is to give them the possibility to see statistics on their traveling habits, more on this in chapter 2. One way to analyze data is by using an OLAP cube which is a tool for analyzing data from different angles, in OLAP called dimensions.

In On-Line Analytical Processing (OLAP) an OLAP cube, which is a pre-calculated multidimensional data set, is used to answer multidimensional queries quickly. This provides scalability as the data set becomes large since it contains answers to all queries and does not need to calculate them before returning the result of the query [2]. This kind of tool could be a part of the solution to providing the clients with a way of analyzing their travels.

In order to extend the system, studies in the area were necessary as well as studies on how well this kind of system responds to change, to keep them agile. To get additional knowledge regarding these kind of systems an extensive tutorial were performed during this period. To create a system that can provide the clients with relevant statistical data, it was necessary to find out what data that were interesting to them and how they would like it to be presented. Investigating this, an interview was performed with a selection of the clients as well as a survey, based on the information gathered from the interviews, clarifying their priorities. During the studies it became clear that a big part of creating these systems in a good way depends on the design, hence, a big effort was made on designing the system in order to be able to get it right the first time. Realizing the designed components, the major part were creating the data warehouse through ETL and then setting up the cube. More information on how the work was executed will be presented in chapter 2.

1.1 Purpose

The purpose of this thesis is to create a BI system, providing SpaceTime clients with statistics, that can help them improve the way they travel in order to reduce cost and emissions.
1.2 Thesis collaboration

This thesis has been done in collaboration with Dohi and SpaceTime. Dohi is a digital innovation agency and has staff involved in the SpaceTime service. They have provided a supervisor, named Kim Nilsson, that has good knowledge of the SpaceTime software. At SpaceTime, the collaboration has been with Anders Broberg regarding technical aspects and Gunnar Granberg regarding clients and how statistics should be presented.

1.3 Outline

This is a brief description of the chapters in the thesis.

1. Introduction
   A short introduction to the subject of the thesis and the purpose of the thesis. Also, a short introduction to Dohi and SpaceTime is included with whom this thesis was done in collaboration with.

2. Problem description and approach
   This chapter describes the problem in more detail as well as the method that were used during this project.

3. Background
   In this chapter the existing SpaceTime system is presented and a description of its components. It also gives an introduction to terms that are frequently used in this kind of systems.

4. In depth study
   This chapter presents information to understand how OLAP systems are structured describing the components in more detail. This involves best practices and things to have in mind when designing and creating such systems. It also describes how these systems can be approached in an agile way.

5. Results
   The results from the pre-study, design phase, and the final mock-ups are presented.

6. Discussion
   Decisions that were made are discussed and why the design ended up as it did are explained.

7. Conclusions
   Conclusions drawn during the project are presented.

8. Future work
   A description of what could be done in the future are presented. Both regarding things to do in the immediate future to integrate the new components as well as things that might be necessary to consider as the system grows.
2 Problem description and approach

The problem at hand and approach that was used during this project will be introduced in order to understand why and how things were done.

As SpaceTime plans to extend their service it is necessary to find out the best way to reach the goal of this extension. This concerns both investigating what is expected from the service and also the best way to realize it, in terms of best practices as well as design.

2.1 Problem statement

Today every developer has used, or at least heard of, agile software development. A set of principles used in software development that aims to rapidly and flexibly respond to change. This sound like a great approach to business intelligence (BI) applications as managers would have the possibility to ask developers to implement new functionality to meet changing demands in an organization. However, in large BI systems, the sheer number of components such as data sources, ETL, a data warehouse, and data cubes make it very time-consuming and difficult to meet changing BI demands. This applies even for slight changes as there often are tight dependencies between them and there might be even more components involved. In the Forrester BI maturity survey found that 49 percent of the respondents thought that their BI applications did not keep up with the changing requirements [7].

In a more technical aspect some research needs to be done in order to find out more about the different components of a BI system, the structure of the components, how to work with them, and some best practices that can be used when building this kind of system.

With these things in mind, the problem statement of this thesis project is

- What is the best way to create a Business Intelligence system that can help SpaceTime clients improve their travel habits?

It will be necessary to investigate to what extent it is possible to work agile with BI applications. Further, what different approaches exists to work agilely in BI application development and how well do they perform on making the application flexible to changes. Also, an interesting aspect is to find out whether extensive extra work is needed, during the development process, in order to make it possible to work agile and keep the application adaptable to change. Technical best practices will also need to be investigated in order to create a well-designed system with high performance and that is easy to maintain.
2.2 Method

The way this project was approached is described in this section. The approach can be divided into two parts, a theoretic in-depth study on the area to acquire necessary knowledge followed by a practical part addressing the matter at hand.

2.2.1 Theoretical part

The in-depth study that was performed mainly consisted of reading a lot of papers regarding OLAP, data cubes, Business Intelligence, Agile, and Agile Business Intelligence. In many of the papers found, only the introduction was read since they either were off topic, too similar to other papers or focused on a very narrow topic that with too much detail. The knowledge acquired from reading relevant papers were used when writing the terminology in chapter 3 and chapter 4. During the theoretical part, a Multidimensional Modeling tutorial [15] was done to get some practical experience on how to create and modify an OLAP cube.

The knowledge and experience from the theoretical part where very useful in the later parts of the practical part of the project when designing, creating, and making changes to the extension of the SpaceTime system.

2.2.2 Practical part

The practical part of this thesis was as mentioned in the introduction to extend the SpaceTime system in order to provide their clients with feedback on their traveling habits. In order to create such an extension, there were three phases that took part.

First of all, it was necessary to understand what statistics their clients value. From the people at SpaceTime, some information about this were acquired, to clarify this even more, interviews were held with a selection of the clients selected by SpaceTime. Since the clients are spread all over the country these interviews were performed via telephone. The length of the interviews varied between 10 to 20 minutes depending on how much they had used the system and how specific details they came up with during the interview. During the interviews, notes were written which later were compiled to sum up the areas that the clients found important. After the interviews had been summarized a survey were formed to help prioritize how important the areas of interest were. Before sending it to the clients the survey was reviewed by the SpaceTime crew and changed according to their feedback. Appendix A.1 shows the survey that was sent to the clients.

The next step was to design both the overall extension of the system and also the components that would be used. When deciding on what components that would be used, the acquired knowledge on OLAP systems and the existing system had to be considered. As the components that would be used in the system came clear, these needed to be designed correctly representing the underlying data and providing the possibility to answer the questions that the clients had. In order to achieve this, the database with information about the clients’ travels required extensive research. A big challenge was to design the system so that it would be easy to maintain and easy to change.

Lastly, with the design finished it was time to implement the system before showing it to the clients to see if it met their expectations. This was done with mock-ups created with data from the data cube and resulted in some last additions and changes based on their feedback.
This chapter has described the problem at hand as well as the method that was used when solving it, providing the clients with a system to improve the way they travel. Now it is necessary to have a closer look at the existing system and some terms that are frequently used in OLAP systems, this to get an understanding of why decisions were made as they were.
3 Background

The existing SpaceTime system, its components, communication channels, and purpose are to be described further. This followed by the areas that the clients are interested in being able to analyze and also terminology necessary to introduce to get a better understanding of OLAP systems.

3.1 Existing SpaceTime system

The existing SpaceTime system components, their communication channels, and purpose are described as a look at the existing SpaceTime is presented. In figure 1 an example is presented of how the result of a search for a trip looks in the booking system.

As shown in figure 1, different alternatives of how the trip could be performed are presented showing different transportation means and how good they are from an environmental perspective as well as the cost of the trip. The first four alternatives are connected to different resources which can be booked through the system. The existing SpaceTime system also provides the clients with feedback on how their resources are used. An example of how this feedback is delivered is shown in figure 2.
Figure 2: An example of how the clients get feedback regarding the use of their resources.

These are two views in the web application but there are more components used in the system which are described below

- **Postgres database** - All information about the clients and their travels are stored in the postgres database.

- **Data warehouse** - The data warehouse is an SQL server and is the data source used in the data cube. The information in the data warehouse is retrieved through ETL, the ETL is also where the structure and relations of the data warehouse are defined.

- **OLAP cube** - The OLAP cube runs on an SQL server and uses the data warehouse as its underlying data source. The cube contains information about the clients resources such as vehicles, bikes, rooms, and such.

- **Web application** - The SpaceTime web application is where the clients make their bookings which are stored in the postgres database. The web application also provides them statistic information about their resources. This information is retrieved from the cube through an ASP .NET API.

- **MVC application** - Information that cannot be accessed via the web application are extracted to Excel and Raindance and stored in a file system that can be reached from the MVC application.

In figure 3 an overview of the system is illustrated showing the different components, communication between components, and communication to the user.
To decide whether the existing components should be used in the extension of the system, it is necessary to look into the areas of interest and the context of the extension.

### 3.2 Area of interest

Based on the pre-study, the goal of the extension of the system is to provide the clients with a system where they can analyze their traveling habits. These involve being able to see costs, emissions, and degree of how much they carpool. All this to see whether their efforts to improve the way they travel pay off, one of these efforts being the use of the SpaceTime booking system.

Since the way the clients travel involves public transport it is not certain to what extent the components of the existing SpaceTime system can be used. Even though this is business intelligence and an OLAP cube is a good alternative for analyzing businesses, it may be necessary to take a different approach than to use the complete existing system. But before making these conclusions more information about OLAP systems, designing this kind of systems, and Agility in Business Intelligence is necessary to explain.

### 3.3 Terminology

This section gives an introduction to terms that are frequently used in the context of OLAP and business intelligence systems.
OLAP is an abbreviation for On-Line Analytical Processing and refers to analyzing data sets to look for useful information [17]. In the term OLAP cube, cube refers to a data cube which is a multidimensional data set. Even though most of us refer to a cube as a three-dimensional object, a data cube can have two dimensions, three dimensions, or even more. OLAP cubes make it possible to analyze data, from different dimensions or angles, in order to make decisions. In figure 4, a sample cube is shown with the dimensions Time, Location, and Products. Since big data sets can time consuming to query, OLAP cubes contain pre-calculated values that make querying the cube swift [2]. This is especially useful when analyzing different parts of the cube because the results of the query are returned without having to do calculations at that specific time since the answers are pre-calculated. Before engaging the opportunities and advantages of OLAP cubes, elements and operations need to be explained, to understand how a cube works.

Figure 4: A sample cube with the dimensions Time, Location and Product.

3.3.1 OLAP cube structure

Data in an OLAP cube is usually stored in a star- or snowflake schema. Either with one or several centralized fact tables each connected to one or more dimension tables [21]. This structure reminds of the structure of a star or a snowflake and hence the names [18].
Fact table

Fact tables consist of the measures of the data cube and foreign keys to the dimension tables surrounding it [21]. The measures are based on the content of the underlying data warehouse and the measures in a fact table are referred to as a measure group.

Measures

Measures can be thought of as labels on values [2] and are based on data from the data warehouse. For instance, a business may have measures such as in-store sales, internet sales, and unit price. The measures in a fact table or a dimension are grouped into a measure group.

Dimensions

The dimensions create the structure of the data cube and categorize measures so that the measure data may be analyzed. The different dimensions of a data cube provide the possibility to analyze the measure data from different perspectives. For instance, a business may have dimensions such as store, product, date, and customer. A business manager can then analyze their business based on those dimensions. The dimensions have attributes which are discrete identifiers and is what the dimension consists of. These are at different levels in the dimension hierarchy and at the month level, April 1985 is an example of a member in the date dimension.

Hierarchies

Dimension elements are typically ordered in hierarchies to provide structure [8] to the dimension. A commonly used dimension is Date which and may be organized in a Calendar
date hierarchy where day is grouped into month, month grouped into quarter, and quarter into year. This enables possibilities to analyze measures based on different levels in the dimensions. A dimension may have more than one hierarchy, the date dimension may for instance both have a Calendar date hierarchy and a Fiscal date hierarchy.

![Diagram of a date dimension showing its hierarchy and members.](image)

**Figure 6:** Illustration of a date dimension showing its hierarchy and members.

### 3.3.2 Operations

When OLAP cubes are used for analysis the basic and most used operations are Roll up, Drill down, Slice, and Dice [8, 2]. These are used to analyze the data on different levels or just to analyze a part of the cube, these are described in the following section.

**Roll-up**

The roll-up operation makes it possible to get an overview of data based on a specific dimension hierarchy [22]. If the dimensions location, date, and product are used in a business and the current level in their hierarchies are country, quarter, and product. We can roll up in the date dimension from quarter to year and see aggregated data for every year instead of every quarter.

**Drill-down**

Drill-down is the opposite of Roll-up and will show more detailed data [2]. In the previous example in the Roll-up section, the drill down operation could show aggregated data based on quarters instead of year if applied to the date dimension. In figure 7, this drill down example is illustrated.
Slice

A slice is a subset of a cube where one value corresponds to one or more members of a dimension [22]. For instance, if 2015 is selected from the time dimension, all other years will be omitted and the remaining subcube is called a slice of the original cube. In figure 8 below this example is illustrated.

![Slice example](image)

**Figure 8:** An example where the year 2015 is sliced out of a cube.

Dice

The dice operation is the equivalent of slicing in two or more dimensions [22]. If the subcube from the slice example above would be sliced again, this time selecting Sweden...
and Norway from the location dimension, the remaining subcube would have been the result of a dice operation. In figure 9, this dice operation is illustrated.

![Figure 9: A dicing example where Sweden and Norway from the location dimension and 2015 from the time dimension are diced out of the cube.](image)

### 3.3.3 Multidimensional query language

The standard query language for OLAP systems is called MDX (MultiDimensional eXpressions) and is used when querying multidimensional databases [10]. It has many similarities to SQL but is made for OLAP using dimensions, hierarchies, and OLAP specific operations [13]. Many MDX queries can be translated to SQL but even simple MDX queries may be really complex if translated to SQL. Below an example of an MDX query is shown.

```
SELECT
    { [Measures].[Internet Sales-Sales Amount],
        [Measures].[Reseller Sales-Sales Amount] } ON COLUMNS,
    { [Date].[Calendar Year].&[2007],
        [Date].[Calendar Year].&[2008] } ON ROWS
FROM [Adventure Works]
WHERE ( [Sales Territory].[Sales Territory Country].&[France] )
```

Let us break down this query and explain the parts of it. The SELECT statement sets the measures *Internet Sales-Sales amount* and *Reseller Sales-Sales Amount* from the Measures-dimension on the COLUMNS axis, and the members 2007 and 2008 from the Date-dimension on the ROWS axis. The FROM statement selects the *Adventure works* cube as the data source on which the query will be used. The WHERE statement selects the slice of the cube where *France* is the only member from the Sales Territory-dimension that will be used. When selecting *France*, we can see that in the Sales Territory-dimension there is a hierarchy where *Sales Territory Country is France* parent.
3.3.4 OLAP - Storage modes

The performance of OLAP cubes comes at a price since all aggregations are pre-calculated in order to be able to answer queries quickly. All those answers need to be stored which results in a huge cube, as relations with a large number of attributes are involved [19]. There are several approaches making the size of OLAP cubes reduced.

**MOLAP**

The traditional OLAP cube is MOLAP, which uses a multidimensional pre-calculated data cube, and provides excellent performance [2]. MOLAP cubes are aggregated from a data source, often a data warehouse, and the size needed for the cube both depends on the data source and the complexity of the cube. The complexity of the cube increases as there are more dimensions and relations defined, at the same time it provides more ways to analyze the data.

**ROLAP**

In the ROLAP methodology the data is stored in a relational database, often a data warehouse, and ROLAP provides the OLAP functionality of slicing and dicing, hence giving the appearance of an OLAP cube. But since the answers to queries are not pre-aggregated a ROLAP cube will need to aggregate the answer to the queries at run-time. As there are no pre-aggregated data the size of the ROLAP cube completely depends on the underlying data source [2]. ROLAP tools are often provided with a "Designer" feature where administrators can specify dimensions, attributes and hierarchies to make the OLAP functionality work [2].

**HOLAP**

HOLAP means Hybrid OLAP and combines MOLAP and ROLAP, using the advantages of both [2]. Often modern OLAP products provide HOLAP allowing the creator of a cube to define what parts that should be stored in which mode. A common solution is to use ROLAP to store large volumes of detailed data and MOLAP to store aggregations to be able to quickly answer queries of aggregated data which often is what analysts want to see.

3.4 Background summary

The existing system and the clients area of interest have been described and an OLAP cube seems like a good fit when analyzing the statistics of interest. Some frequently used terms were introduced and explained, briefly giving an idea of the components and structure of a cube, its operations, query language, and storage modes. This allows going more into detail about how to design an OLAP cube, its data source, and how to keep business intelligence systems agile.
4 In-depth study

The design of an OLAP cube, its data source, and how to use Agile software development in business intelligence systems are to be described, allowing to proceed to create a high-quality OLAP system.

4.1 OLAP cube - Design

In this section, the design of an OLAP cube, and its components are described in more detail and best practices that should be considered when creating a system of high-quality explained.

When designing an OLAP cube there are several important aspects to provide a functional and high performing cube. First to be mentioned is to use supported data sources and data providers. In a well designed OLAP cube the design of the dimensions is probably the most important part and it is very important to make sure that the attributes, relationships, and hierarchies fit the needs of the end user and correctly represents the underlying data [14].

The following sections describe some important things to have in mind when designing and configuring an OLAP cube. Some detailed setting names that are mentioned are based on Microsoft SQL Server Analysis Services but the idea of improving performance and avoiding confusion is the goal of these settings in any OLAP cube.

4.1.1 Fact table

The facts, or measures, in a fact table are referred to as a measure group. If multiple fact tables are in a cube but they are not usually used together, it may be a good idea to create more than one cube where each cube concerns a specific topic. This since multiple measure groups may have a negative effect on query performance even if only one measure group is queried. If a query that is used more seldom requires more than one cube, a special cube can be made by creating linked measure groups.

If more than one measure group is based on the same dimension and has the same granularity, they should possibly be in the same measure group containing several partitions. This to improve performance and make it more user-friendly. An exception to this is when a distinct count measure is used, to improve performance such groups should be in a measure group of its own. For instance, if a business has several customers that each buy several products, distinct count can be used to count the number of costumers. To minimize the data size of the cube one thing to have in mind is numeric data types differ in size, use the smallest numeric data type possible for each measure. To be remembered is that the data type needs to be large enough to hold the aggregated value, the ”all” value.
4.1.2 Dimensions

As mentioned earlier, dimension design is probably the most important aspect of a good design for an OLAP cube. To start with, a dimension should be independent and hold related business entities. For instance, a business Customers and Products should be in separate dimensions to avoid confusion and poor query performance.

In business intelligence, some dimensions are often duplicates with different names, a frequent case is the dimensions Order date and Ship date. These should not be created as two separate dimensions, instead, they should be based on the Date dimension who just plays the role of the other two. This will make it look like there are three dimensions but the complexity and size of the cube will be reduced.

Since designing the dimensions correctly can be quite a complex challenge, this section will point out important things to have in mind.

Attributes

An important part when designing dimensions are attribute relationships, all attribute relationships in the data need to be defined in the cube to reflect the data in a correct way. This helps optimizing data storage and MDX restrictions [14]. Unnecessary attributes should not be used since they have a negative effect on performance and usability for the end-user. It is often a better approach to add attributes that are found necessary than to first add all attributes found and remove the ones that are not needed [14]. Redundant relationships should be avoided since they only increase complexity and do not benefit the dimensions. Hence, such relationships shall be removed to avoid problems and increase performance [14].

If an attribute is below the granularity of the data source, day in the date dimension when the data source lowest granularity is week, it can be disabled from the hierarchy by setting AttributeHierarchyEnabled to false [14]. This to avoid confusion and unnecessary attribute members. This can also be used to increase to performance in other cases, for instance if an attribute has values that are close to unique they are seldom used to group by. For instance, if only customers phone numbers and Social Security Numbers are stored the AttributeHierarchyEnabled should be set to false to increase usability and performance.

The key columns of attributes need to define a unique member of an attribute. The Month attribute in a time dimension needs to have both Year and Month name as key columns while a Product ID most certainly may be enough as a unique identifier in a Product dimension.

If an attribute contains a lot of members (more than a million) a numeric key should be used to improve performance [14]. To make it easier for users to understand another string column still can be used as a name for the members.

If the relation between members of an attribute is not expected to change the Relationship-Type should be set to Rigid. If they are more prone to change it should be set to Flexible. A Rigid relationship may be the months of a year or location of a city while a Flexible relationship may be stores in a given country or the title of an employee. When these relationships are set correctly, changes and aggregations can be processed in an optimized way by the server.

If the natural ordering of an attribute is not alphabetical, the OrderBy and OrderByAttribute properties can be used to get the correct ordering. Time attributes such as months are a
good example of these, they are supposed to be ordered in chronological ordering instead of alphabetical ordering.

Hierarchies

Hierarchies are an important part of the dimensions and whenever there are attributes that are related in a dimension a hierarchy should be considered. All dimensions should hold at least one hierarchy, the exception is parent-child dimension since they, in a way, has a special type of hierarchy that will be described more in detail later. The order of the attributes in the hierarchies are important, if lower level attributes contain fewer members than the one above, the order of the attributes are often in the wrong order. Such a case may be State-Provoince above Country.

In hierarchies, all attributes should have a relationship with the level above to be able to optimize the server. Diamond-shape relationships are relationships that have the same start and end but no relationships in common. If a date dimension holds the hierarchies Day -> Month -> Year and Day -> Quarter -> Year these are called Diamond-shaped relationships and should be merged into one hierarchy holding all attributes.

In parent-child dimensions, an attribute can have the same attribute as its parent. In businesses a Employee may have another Employee as its parent, this is often the case with managers. These can be very powerful and flexible but should only be used when needed to avoid poor query performance. If used, the AttributeHierarchyVisible of the key attribute should be considered to be set to False since it is stored in the parent-child hierarchy in a more organized way.

4.1.3 Data source

OLAP cubes are aggregated from data which are stored in some kind of data source, most commonly in a star- or snowflake schema in a data warehouse[5]. Typically the data in a data warehouse are gathered from one or multiple data sources and the data sources may, most likely, not have the same structure.

Star- or Snowflake schema

When to choose a star- or snowflake schema depends on the size, the structure, and some other characteristics of the data. Below a comparison between when the two is suitable shown [6]

- **Maintenance / change** - The Snowflake schema has no redundant data which makes it easier to maintain and change than the Star schema that has redundant data.
- **Ease of use** - The Snowflake schema requires more complex queries which make it harder to understand and use.
- **Query performance** - With more foreign keys the Snowflake schema is more time consuming when it comes to executing queries than the Star schema.
- **Data warehouse** - The Snowflake schema is good for data warehouses with many-to-many relationships and Star schema with one-to-one or one-to-many relationships.

- **Joins** - The Snowflake schema has a higher number of joins than the Star schema.

- **Dimensions** - In a Snowflake schema there can be more than one dimension table in each dimension which is contrary to the Star schema with one dimension table in each dimension.

- **When** - With big dimension tables the Snowflake schema is better since it reduces space. With dimension tables with fewer rows, a Star schema can be used.

- **Normalization** - In a Snowflake schema the dimension tables are normalized with a de-normalized fact table. In a star schema both are de-normalized.

- **Data model** - Snowflake schema in a bottom-up approach and Star schema top down approach.

Whether a Snowflake- or a Star schema is used, the data needs to be transformed from the data source into the data warehouse. To be able to store the data in the data warehouse in a unified way it needs to be transformed.

**Extract, Transform, Load**

Extract, transform, load or ETL are used to gather data from different data sources into a data warehouse. The data sources are often relational databases but this may vary, Excel documents or text files can be used as data sources. On each data source, the procedure is performed as described below.

- **Extract** - Extract the data, that will be used in the data warehouse, from the data source.

- **Transform** - Transform the extracted data to fit the structure and format in the data warehouse. Depending on the requirements of the data warehouse some data may be deleted or altered in this process to meet the data warehouse purposes.

- **Load** - Load the transformed data into the data warehouse.
Before the ETL procedure can be implemented, the data warehouse needs to be designed to fit the purpose of the cube in order to know how to perform the transform operations.

**4.1.4 Design summary**

So the design is a big factor when creating this kind of systems and there are several components and settings that are necessary to have in mind. With these mentioned components, in some cases even more are used, agility may not seem easy to maintain. Since most developing projects these days work hard to achieve an agile way of working there should be no difference in this type of projects.

**4.2 Agile Business intelligence**

This section gives an overall idea of Agile software development and two of the most used methodologies in that area. Next up is the subject on how these methods work in business intelligence systems like a system using an OLAP cube.

**4.2.1 Agile software development**

Agile software development, often just called agile, is a set of twelve principles that can be used when developing software [12]. Agile is supposed to improve adaptability to change, customer satisfaction, and working code. In the *Manifesto for Agile Software Development* they state

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right,
we value the items on the left more.

Agile have not specified any methods on how to achieve this, but several methods have evolved based on the agile set of principles, among them Scrum and eXtreme Programming (XP), just to mention the two most popular methods [11]. Even though these methodologies point out certain ways to work, it is often recommended that agile methodologies are customized in order to work in each organization or team [9].

Scrum

Scrum is the most popular agile methodology that is introduced to software developing teams, much thanks to its simplicity and flexibility [1]. Scrum uses roles and artifact to increase the agility in software development project. The roles used in scrum are Product owner, Scrum master, and development team. These are briefly described below

- **Product owner** - The product owner is a business representative and speaks for the stakeholders and should not interfere in the technical aspect of the development. The product owner writes user stories and prioritize them.

- **Scrum master** - The scrum master works as a coach for the development team who makes sure the Scrum process is followed and serves to remove obstacles in the development teams way. The scrum master also communicates with the development team and the product owner to make sure progress keep on going.

- **Development team** - The development team consists of 3-9 members who have the required skills to complete each sprint and performs the actual work of analyzing, designing, implementing, testing, and developing the product. Each development team is self-organizing and responsible to deliver a complete increment of the product after each sprint.

Scrum uses elements to keep regularity and time-box intervals. Most of these have focus on the sprints and involve planning, daily meetings, review, and retrospective. If there are more than one scrum team working on different parts of a project, the scrum master of each team usually meet daily to keep the project coordinated. This since there may be interdependencies between the different teams.

The artifacts in Scrum used to keep track of what to focus on and how things are going. The Product backlog contains user stories that describe things that need to be done and are prioritized in the product backlog. The product backlog is never completed since newly discovered requirements may be added or old requirements removed. The Sprint backlog contains the user stories that the development team has decided to complete during the current sprint. A Product increment is the result of every sprint in terms of the product and
should be ready for release at the end of every sprint. However, the product owner does not necessarily choose to make a release at the end of every sprint. The Sprint burn-down chart is a visualization of the progress during the sprint and is updated by the development team after each day. The Release burn-down chart is similar to the sprint burn-down chart but is based on the sprints, showing the progress after each sprint.

eXtreme Programming

In software development, eXtreme Programming (XP) is another popular agile methodology. In XP, the code, its quality, and the adaptability to change during the project [20] are in focus. XP uses increments which are short cycles with possible releases similar to the sprints in Scrum. This to improve productivity and to make room for changes during the development process. As mentioned earlier, the code is in focus and the XP methodology means that code is the most important product of a software development project [20]. To provide high-quality code testing is necessary, in XP testing is taken to the extreme by saying that every piece of the code should be tested before moving to a new feature. Unit tests should be coded in advance of the code (Test-driven development or TDD) that will be tested and also acceptance tests are made to verify that the software meets the customer’s actual requirements [20].

XP has five values that say improves a software project, these are communication, simplicity, feedback, respect, and courage. Communication with the other developers and the customer is continuous to avoid misunderstandings. Simplicity means to keep the code and the design as clean and simple as possible, no implementations for the future that may be needed. Feedback comes from the testing, which if used as XP says, starts from day one. As delivery comes fast and suggested changes can be fixed, respect grows both within the team but also from the customer. With respect in the team, the courage comes to respond to changes both in requirements and technology [20].

4.2.2 Use of Agile in Business Intelligence

Agile seems like a great tool in order to be able to embrace change but at the same time articles about agile in BI most of the time reminds one to be cautious when trying to work agile with BI applications [4, 16, 7]. This section describes how to work agile in BI, advantages, risks, and special technologies that can make BI truly agile and flexible to change.

Using agile methodologies in BI

The purpose of using Agile in BI are the same as for any Agile development methodology, getting development done faster and being able to adapt to change[7]. This by working in increments that result in partially being able to deliver pieces that can be used during the development. Some concerns on using agile methodologies in BI are that BI systems most of the time has several components with tight dependencies. Hence, change results in making changes in all, or many of, the components making it difficult and time-consuming to make changes.
Advantages

There seem to quite a united point of view on where the gains are in using agile methodologies in BI. Even so, there is disagreement on whether some advantages truly are advantages or just perceived as such. But let us start where there is common ground.

- **Code quality** - Testing of code, preferably test automation, improves the quality of code and continuously makes sure that all parts work properly [16, 4].

- **Iterations** - Short iterations delivering working parts of the solution makes the project flexible and adaptable to change. It also opens up for users to start using the product to provide feedback in order make sure that the product meets the requirements and minimize the extra work due to change [16, 4].

- **Customer satisfaction** - Customer satisfaction is often improved using agile since they continuously can see improvement and has the possibility to affect the functionality of the product. In a survey, 78 percent reported improved stakeholder satisfaction using agile practices [3].

With these great advantages in mind, there are also some advantages that are perceived as an advantage but in reality does not differ from traditional methods.

- **Delivery** - Delivering the final product, with the same amount of functionality, often takes about the same time using agile in BI as traditional methods. Nevertheless, an agile approach often results in giving the appearance of faster delivery since the users can have access to the product during the development period [4].

- **Cost** - In short term, an agile approach often increases cost due in order to introduce agile to the team and train new members of the team to work agile. How much this will affect a project concerns the length of a project and if it will be able to use their skills during maintenance and future enhancements of the product [4]. Also, for agile to work well developer teams should be located close together which may increase travel costs and office space needed. In a survey on agile practices, it was reported that 23 percent of agile adopters saw an increase in cost while 37 percent reported lower costs [3].

Risk

Even though there are advantages using agile in BI, agile is not a panacea that automatically bring success. A Dr. Dobb’s survey reported that agile software development had an average success rate of 83 percent in a team where all members were located at the same place. In offshore situations, the number was 60 percent in the same survey [3]. Hence, offshore projects increase risk, reasons to this are decreased communication possibilities, both between team members but also between business and technological resources [4]. Having offshore projects may also involve being in different time zones, all this is contradictory to the agile principles.

Since data warehousing and BI often are quite complex and lacks support for automatic testing, a BI project should not be the first agile project a team takes on. An agile inexperienced team is often a big risk and should first gain agile experience before applying agile
to a BI project [4]. Another risk that often relates to inexperienced users is that one sees agile as a religion that should be followed at all cost. This is seldom a successful approach since all organizations have to customize an agile methodology to fit organizational needs. In a BI project this may mean that some parts of the project where requirements are vague, an agile approach could be used, the rest of the project where the requirements clear, a traditional waterfall model can be used [16]. This approach may also in itself be a risk. The ETL-design and the structure of the data model, which are core components in a BI project, may be vague and unstable. In this case, an agile approach, with short iterations where delivery is expected at the end, may result in that these core components continue to be vague which can lead to project failure. Hence, it is very important to have well-designed core components to avoid lots of rework or a failed project [4].

Many organizations traditionally work in a strict top-down governance. Agile teams are supposed to be self-organized and a top-down approach has no part in an agile workplace [4]. It is necessary to give the agile approach an optimal environment in order to succeed. It is necessary to be aware of these risks in order to avoid failure. If an agile methodology is to be implemented these need to be analyzed and counteractions made.

Criteria

When using agile in a BI project there are criteria that should be met in order to increase the chance of a successful project.

The team should mainly consist of experienced developers, preferably with experience within agile methods [4, 16]. All skills required during the project should exist within the development team and if there are gaps in the skill set, training is necessary to close those gaps [4]. As in any agile software development project, the teams should be small and self-organized and keep contact with the customers regularly.

Even though change should be expected during the project [16], the project scope and its objectives should be well defined. The planning in each iteration is crucial to ensure that all resources needed for the iteration are available, the development team stays motivated, and keeps focus on achieving the iteration goal [16, 4].

It is necessary to customize the agile methods to fit the organization’s rules on documentation since a part of agile is *working software over comprehensive documentation* [12] this may not go hand in hand. The organizational requirements ultimately trump the agile principles so one need to make sure on how to handle this.

Practices for success

In the extent of analyzing the risks and meeting the criteria described in the sections above, a BI project should make use of the following practices to introduce agile successfully.

Different vendors should be examined to find those who has support for the different components that will be used in the BI project. These then should be evaluated in order to try to make a decision regarding which one fits the organization best. It can be a good idea to do a minor pilot project when a vendor has been selected and evaluate it [4]. This to avoid mistakes in the beginning of the project.
To bring agility into a BI project it is important to not just work agile with the BI part of the project. To be able to adapt to change in requirements the database and ETL design has to be agile. This since they often are core components in a BI project [4].

Tools and technologies that facilitate design and automatic testing should be invested in. Such tools emerge continuously and may very well be a good idea to look at [4].

Since small changes may be time-consuming in BI projects, it is necessary to dedicate time to refactor the design [4]. In the case of changed design, it is often necessary to recreate the data warehouse. Make sure that the ETL can do an initial load without manually changing the code in the ETL process to be able to set up and load the data warehouse.

At the start of a BI project, it is often vague what the requirements will be. To speed this up, there is an opportunity to extract data from sources into a temporary data environment that will not be used in the logical model. Neither the business people or the developers know what they want until they can see what exists, to facilitate the process may involve showing existing data to come up with requirements. As soon as the data warehouse forms it should be loaded with data to clarify business rules and logic that will have affect on the BI application [4]. This approach of loading data early may help the design and development to avoid starting over with the design from the beginning as requirements come clear [4].

Technologies for Agile BI

Even if Agile BI in most ways are the same as other agile software development methodologies it differs in that it requires special technologies and architectures as support to become truly agile [7].

Metadata-generated BI application

A BI system often includes a lot of different components a may include data sources, ETL, data warehouse, data cube, reports, queries, and more. In a large, global, multinational enterprise often consist of 20 to 30 components [7]. The idea with a metadata-generated BI application is to use metadata to define rules for how all the components depend on and communicate with each other. Metadata is also supposed to be used to auto-generate the components so that when there is a change of requirements, changes only has to be done in the metadata and not in all the different components.

This way of generating a BI application is present and a number of vendors provides this kind of BI application metadata model. All changes shall be done in the metadata model that handles all dependencies between components and automatically generates the application [7]. The different providers of these metadata models provide support for different BI platforms so it is necessary to choose the right one who fits the system.

Although this approach has benefits there still are risks. To provide the benefits the metadata-generated BI tools comes with great dependencies to each vendor’s technology and application. Since most of these vendors are relatively small and may be acquired by another company or cease to exist, their application may not be supported anymore [7]. They also often use specialized database management systems which may result in that some SQL and MDX-based tools cannot access or query the system. Due to risks, lack of SQL support, poor ETL features, and lack of IDE features most organizations use this kind of system in
coexistence with their traditional BI application [7].

To avoid failure when choosing such an application one must do some research to see what vendors support the necessary tools. Depending on what the organization will use their BI application for, it is also necessary to find out if the vendor supports automatic generation of the components that will be used [7].

4.2.3 Agile summary

So these agile ways of developing systems may be adopted and used when creating business intelligence system. Even so, some other aspects should be kept in mind and often a customized way of Agile need to be used with regard to techniques and technologies.

4.3 Summary of the in-depth study

The in-depth study has provided a good foundation describing best practices when creating a system using OLAP cubes and also an agile approach on business intelligence systems such as these. The results from this project are now to be revealed.
5 Results

As mentioned in section 2.2.2 the phases that took place were the pre-study, the design phase, implementation phase, and creation of the mock-ups of how the statistics should be presented to the clients. The results of the pre-study, design phase, and the mock-ups are to be presented. In chapter 6 choices and decisions that affected these results are discussed.

5.1 Pre-study

The result from the pre-study are both based on the interviews as well as the survey that the clients took. Below the result of the pre-study is presented. The priority level starts with one which is the highest priority.

1. **Carbon Dioxide emissions (CO\textsubscript{2})** - Should be able to see the amount of CO\textsubscript{2} that comes from their travels. Should be able to see based on transportation mean, time periods, and parts of the organization such as the entire organization, department or individual user.

2. **Transportation mean** - Should be able to see how much they use different transportation means. This involves walk, cycle, cars, buses, and public transportation. Should be able to see based on time as well as different levels in the organization.

3. **Costs** - Should be able to see the cost of their travels based on transportation mean, time and parts of the organization.

4. **Carpooling** - Should be able to see to what degree they carpool. This only concerns their own vehicles and should be able to see based on time as well as parts of the organization.

So these were they areas that had the highest priorities and the way they should be able to see and compare data. The full summary of the interviews and surveys can be found in appendix B.1.

5.2 Design

This section states the results of the design phase of this project. This includes both the overall system design as well as the design of the different components.

5.2.1 System design

The overall system design is presented in figure 11 below.
The new cube module will consist of the same components as the existing cube module, namely an ETL, a data warehouse, and an OLAP cube.

5.2.2 Data warehouse design

The design of the data warehouse is shown in figure 12 below showing the fact table surrounded by the dimension tables. The User and Transportation mean dimensions are connected to additional tables that will be used to create the hierarchies in the cube.
5.2.3 ETL design

The ETL design is illustrated in figure 13 below, showing the packages and their purpose.
The initial package will set up the data warehouse with tables and relations, create the Time dimension, and create the transportation mean dimension. These dimensions do not depend on the postgres database. The run package will then populate the user dimension and the fact table. In order to run the run package the initial package needs to be run before in order to set up the data warehouse.

### 5.2.4 Cube design

In figure 14 below the cube design is illustrated with the central fact table, including measures that will be used, and the dimensions surrounding it with their hierarchies.
5.3 Mock-ups

Below the final mock-ups are presented in figures 15-19.
Figure 15: Mock-up visualizing the carbon dioxide statistics.

Figure 15 shows a mock-up that shows the carbon dioxide emission generated by the clients travels. In the left part of the mock-up, a diagram is visualizing the amount of emissions based on transportation means over the years. To the right it is visualized how much of the emissions that come from different transportation means. It also presents some statistic numbers showing the amount of emissions from each transportation mean and how much emissions that have been saved from using environment-friendly options.
Figure 16: Mock-up visualizing the transportation mean statistics.

Figure 16 shows the mock-up presenting statistics on how much distance an organization has covered based on the different transportation means. The left diagram shows distance covered over the year and the diagram to the right how much of the total distance the different transportation means cover. The distance traveled by the different transportation means, and the total, can be seen in numbers on the top right.
Figure 17: Mock-up visualizing the costs statistics.

Figure 17 shows the mock-up presenting statistics on costs generated by the organizations travels. In the left diagram costs over the years are shown, based on the different transportation means. And to the right it is presented how much of the costs that are connected to each transportation mean, both in numbers and a diagram, as well as how much money the organization has saved by using cheaper transportation means.
Figure 18: Mock-up visualizing some carpool statistics.

Figure 18 shows the first of two mock-ups presenting statistics regarding carpooling in the organizations. The leftmost diagram shows the total number of travels, how many of them that had only one passenger, and how many that were done by carpooling. To the right the number of travels made by car and how many of them that carpooling was practiced both with numbers and a diagram.
Figure 19: Mock-up visualizing some more carpool statistics.

Figure 19 shows the second mock-up presenting statistics on carpooling. To the left the topmost diagram shows to what degree carpooling is used and in the other the degree to how many of the seats that have been used during travels by car. To the right the distance traveled and distance saved by carpooling is shown together with some numbers on the savings in emissions and costs that were the result of carpooling.
6 Discussion

Factors and decisions that led to the results in chapter 5 are discussed to point out why things were done as they were.

6.1 Pre study

The goal of the pre-study was to clarify what factors that were of interest regarding the clients' travels. As mentioned in chapter 2 the survey that was sent to the clients is in appendix A.1. After the survey had been answered it was summarized to point out the priorities among the clients. As seen in appendix B.1 there were two questions that scored really low, regarding which routes that often were traveled and what time of the day travel were performed, and therefore not included in the extension of the system. One more question, regarding how much of the carpooling happened through an application in SpaceTime, also were excluded from the system. Not because of low scores but because that there was no information in the postgres database that could show this.

6.2 System design

Since the existing system already has a data warehouse and a data cube, these were analyzed to decide whether to extend them or if a separate module would be implemented. The analysis of the existing cube and the pre-study showed that the extension of the system could not use the existing cube. This because to meet the need of the clients there would have to be completely different dimensions and a lot of other measures in the fact table.

6.3 Data warehouse design

The data warehouse is the foundation of the data cube and the design of it needs to have the expected functionality of the cube in mind, regarding measures and dimensions, but also factors in performance and maintainability.

From the pre-study it could be found that the dimensions that would be used were User, Time, and Transportation mean. This since these were the variables that the clients would like to be able to sort data on. During the process it was also found that the clients would like to be able to sort on internal and external vehicles, external being rental cars. To achieve this another dimension was created called External.

The user dimension needed a hierarchy to be able to see data aggregated into departments and the entire organization. In this phase of designing the dimension, there were some
issues. The users may be part of several departments, resulting in many-to-many relationships, and the departments may have another department as its parent, resulting in parent-child dimensions.

In the postgres database it existed a relationship between Department members and the Users that declared which users that belonged to what departments. Instead of using the User-table to create the User dimension, the Department member-table was used as the lowest granularity together with additional information stored in the User table. This resulted in that some data are stored more than one time, name of the user is one example, and that the data is not aggregated based on a user, but rather department members. The upside was that it made it possible to create a hierarchy where it would be possible to aggregate data on departments, which was a key functionality that the clients requested.

Due to the departments possibility to have another department as its parent, the user dimension was designed to have one table for the users, one for the departments, and one for the organizations. This makes it possible to create a parent-child dimension from the department table that can have a foreign key to another department. Ultimately this resulted in that the data warehouse would use a snowflake schema which makes the system easier to maintain as mentioned in section 4.1.3.

When designing the time dimension there were two alternatives considered, using Analysis Services to generate a time dimension or using the one used in the other cube. The Time dimension used in the other cube were re-used since it had custom features already implemented for sorting on certain time-spans.

The transportation means dimension used in the other cube were reused since it was customized to variables in the postgres database. The hierarchy in the transportation means dimension needed to be modified since the different types of transportation means were not supposed to be divided in the same way as the old cube.

The external dimension was used to provide the users with the functionality to sort on whether it was a vehicle of their own or a rental car.

The measures that were supposed to be used in the cube then were defined to be able to create the table that would be the fact table in the cube. The fact table also included foreign keys to the different dimension tables to be able to aggregate data based on the different dimensions.

### 6.4 ETL design

The ETL is the component that will create the data warehouse and then use the postgres database to populate the data warehouse with data. The ETL will use two packages, one for setting up and initializing the data warehouse with dimensions that are not depending on the postgres database, and one for populating the data warehouse with data from the postgres database. It was set up like this to be able to avoid running the initial package unless it is necessary to save time, this since creating especially the time dimension is quite time-consuming.
6.5 Cube design

The design of the cube is a big part of why the data warehouse was designed as it was. Hence, the cube design is an extension of the data warehouse providing the possibility to see aggregated data from the data warehouse based on the dimensions.

As the system had been designed and implemented it had to be tested to see that it could provide the clients with necessary statistics in the areas of interest. Before integrating it into the SpaceTime system it was necessary to decide how the statistics should be presented to the clients.

6.6 Mock-ups

To prepare for integration of the new components into the SpaceTime system, mock-ups were created to make sure that the way of presenting the statistic to the clients really has value to them. All the data in the mock-ups was taken from the data cube, except the cost- and emission mock-ups that used both data from the data cube but also multipliers representing the different cost models that the organizations should provide. When the layout of the mock-ups was created it was heavily influenced by the existing web application as it should fit the existing template. In figure 20 below a mock-up is compared with the existing web page for statistics showing the similarities.

![Figure 20: Comparison of a mock-up and the existing web page for statistics.](image)

6.7 Workflow

During the in-depth study approaches on how to create a system like this was learned. In this section, it is presented how the workflow turned out.

In order to create a system of high value to the clients, it is of the essence that the cube, and thereby the data warehouse, represents the data and relations that exist in the postgres database as mentioned in section 4.1. To make sure that this was the case the approach to creating the system were, as mentioned in section 4.1.2, to start off with an empty shell in the data warehouse and then populate it with necessary data. Since the data in the postgres
database were not structured as it would be in the data warehouse and the postgres database contain 93 tables, the relations and data needed to be studied in order to make sure that the data warehouse correctly represented the data in the postgres database.

6.7.1 ETL and data warehouse workflow

To make sure of that the data in the data warehouse correctly represents the data in the postgres database, each added data column in the data warehouse first were fetched and then compared to the data in the postgres database before choosing a new column to fetch. The workflow when creating the ETL, and by that populating the data warehouse, is illustrated in figure 21 below.

![Workflow during creation of ETL and data warehouse](image)

Figure 21: Workflow during creation of ETL and data warehouse.

These iterations were most definitely necessary in order to ensure that the data in the data warehouse matched the data in the postgres database. Several times it was found that the data was corrupt in some way and if it would not have been controlled during the development it most likely would be necessary to start all over again.

6.7.2 Cube workflow

When creating the cube first the data warehouse connections were made and the data source view. Then the cube was created by defining the fact table and the dimensions that would be used. When the dimensions had been created their hierarchies were defined in order to be able to aggregate data based on the dimension. The labels on the dimension attributes are originally set to the keys so they had to be modified to show the corresponding name. For example, the months should not be labeled 1, 2, 3, and so forth but rather January, February, March, and so on. Likewise, the label of users in the database should rather be their first and last name than the user key.
Creating the cube was less time-consuming than expected and resulted in a cube with the necessary functionality presented in an understandable way. Even so, some of the names of the measures could be confusing so a readme file explaining all the measures and dimensions was created.
7 Conclusions

In chapter 2 the problem of this thesis project was stated as

- What is the best way to create a Business Intelligence system that can help SpaceTime clients improve their travel habits?

To answer this, both a theoretical part, the in-depth study, and a practical part have taken place to be able to clarify how this should be done. Whether this way actually is the best way to execute such a project is hard to say. However, a functional and high-quality system has been created where decisions were based on best practices with the current circumstances in mind.

From an Agile perspective, this system was relatively small, having only one data source, and it was somewhat easy to adapt to change. Even so, it took noticeable effort and in the end one could understand that this becomes a big problem in systems with several data sources, requiring changes in several components.

From the in-depth study, it was learned that best practices in this kind of systems, always depend on circumstances as size, dependencies, update frequencies, performance, and change. With the circumstances of this project, the problem statement has been successfully answered through this thesis and the product created during this time. Whether it actually will improve the clients travel habits remain to be seen, but it definitely has the potential to do so.
8 Future work

The future work that remains to be done before the extension is complete is to be stated as well as some small notions on things that might have to be changed in the future.

As of now the extension need to be integrated into the system. The mock-ups has been shown to the clients but they only show examples of how to present the statistics. Filtering and sorting settings need to be able to be adjusted in the web application. Also, before starting the implementation the mock-ups should be discussed again with the clients for additional feedback to make sure nothing is overlooked.

It was decided that in the first version of the new system the costs and carbon oxide values should be based on the clients own cost and emission models, multiplied with the distance measure. However, it would be nice in the future to change the system so that all these values could be handled in the back-end of the system. To do this further changes are needed in several of the system components.

The data warehouse may be trimmed down, removing some columns that have been used during the development process. These columns have been used to compare data in the data warehouse and the postgres database to make sure everything was transferred correctly. The reason why these are kept is that they may be used when adding extra functionality in the future and since the data warehouse is relatively small the performance is not a problem. If the data grows a lot and performance become an issue, these may be necessary to optimize.

Regarding the data cube, everything is at the moment stored in MOLAP mode to provide the highest performance. If the data in the future becomes really large modifications in the storage modes may be needed to meet the new conditions. In that case, other optimization settings may also be necessary to provide high performance, but for now, this is the best option.
References


A First appendix

A.1 SpaceTime Survey

SpaceTime

Enkätens syfte är att ta reda på hur SpaceTimes tjänster kan utvidgas för vara till nytta för ert företag. Vad är intressant för ert företag att veta om ert resande?

1. Hur intressant är det att veta fördelningen av hur ni reser baserat på transportmedel?
   Markera endast en oval.
   
   1 2 3 4 5
   Inte intressant Mycket intressant

2. Hur intressant är det att veta hur långt ni reser med respektive transportmedel?
   Markera endast en oval.
   
   1 2 3 4 5
   Inte intressant Mycket intressant

3. Hur intressant är det att veta vad ert resande kostar baserat på transportmedel?
   Markera endast en oval.
   
   1 2 3 4 5
   Inte intressant Mycket intressant

4. Hur intressant är det att kunna jämföra kostnader för ert resande under olika perioder?
   Markera endast en oval.
   
   1 2 3 4 5
   Inte intressant Mycket intressant

5. Hur intressant är det att veta hur mycket koldioxid ni släpper ut vid användandet av olika transportmedel?
   Markera endast en oval.
   
   1 2 3 4 5
   Inte intressant Mycket intressant
6. Hur intressant är det att kunna jämföra koldioxidutsläppet för ert resande under olika perioder?
Markera endast en oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>inte intressant</td>
<td></td>
<td></td>
<td></td>
<td>Mycket intressant</td>
</tr>
</tbody>
</table>

7. Hur intressant är det att veta hur mycket ni samåker?
Markera endast en oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>inte intressant</td>
<td></td>
<td></td>
<td></td>
<td>Mycket intressant</td>
</tr>
</tbody>
</table>

8. Hur intressant är det att veta hur stor del av samåkningen som sker tack vare SpaceTimes tjänster?
Markera endast en oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>inte intressant</td>
<td></td>
<td></td>
<td></td>
<td>Mycket intressant</td>
</tr>
</tbody>
</table>

9. Hur intressant är det att kunna se hur mycket ni reser under olika tider på dygnet?
Markera endast en oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>inte intressant</td>
<td></td>
<td></td>
<td></td>
<td>Mycket intressant</td>
</tr>
</tbody>
</table>

10. Hur intressant är det att kunna se vilka sträckor som ni reser mest?
Markera endast en oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>inte intressant</td>
<td></td>
<td></td>
<td></td>
<td>Mycket intressant</td>
</tr>
</tbody>
</table>

11. Vilket av nedanstående områden är viktigast för er att veta mer om?
Markera endast en oval.

- Våra resekostnader
- Våra koldioxidutsläpp
- Hur mycket vi samåker
- Restider på dygnet
- Hur SpaceTime bidragit till samåkande
- Sträckor vi reser mest
- Hur mycket vi använder olika transportmedel

https://docs.google.com/forms/d/1Ope9Y3AtUtsuxy_QidvDyLQxsnnUtAZQ9rQ0eRDZzo/edit?usp=forms_page
12. Vilket av nedanstående områden är näst viktigast för er att veta mer om?
   Markera endast en oval.
   
   □ Våra resekostrakter
   □ Våra koldioxidutsläpp
   □ Hur mycket vi samårker
   □ Restider på dygnet
   □ Hur SpaceTime bidragit till samåkande
   □ Sträckor vi reser mest
   □ Hur mycket vi använder olika transportmedel

13. Vad ska statistik kring ert resande användas till?

14. Finns det några krav på det er statistik skall användas till? Om ja, vidareutveckla gärna för att vi skall kunna ge er en så bra tjänst som möjligt.

15. Har du fler förslag på statistik som kan vara av värde för ert jobb med att optimera resandet?

Tillhandahålls av

Google Forms

https://docs.google.com/forms/d/1Ope9Y3AtUtsuxy_QidvD-yLQxsnnUtAZQ9rQ0eRDZzo/edit?usp=forms_home
B   Second appendix

B.1   Summary pre study

Sammanställning av enkät och intervjuer
Det vi behöver är koldioxidutsläpp kopplat till fordon och tid
Vilka olika transportmedel som används kopplat till tid
Kostnader för resor
Hur mycket folk samäker och hur mycket av samäkning som sker mha spacet ime, alltså boka in sig på resa
Nyttjandegraden på resurser, både fordon men även rum

Enkät
Svar på vad som är viktigast, i rangordning
1. Våra koldioxidutsläpp
2. Hur mycket vi använder olika transportmedel
3. Våra resekostnader
4. Hur mycket vi samäker

Snittpoäng per fråga (1-5)
Hur intressant är det att veta fördelningen av hur ni reser baserat på transportmedel?
4
Hur intressant är det att veta hur långt ni reser med respektive transportmedel?
5
Hur intressant är det att veta vad ert resande kostar baserat på transportmedel?
4
Hur intressant är det att kunna jämföra kostnader för ert resande under olika perioder?
4,5
Hur intressant är det att veta hur mycket koldioxid ni släpper ut vid användandet av olika transportmedel?
4,5
Hur intressant är det att kunna jämföra koldioxidutsläppet för ert resande under olika perioder?
4,75
Hur intressant är det att veta hur mycket ni samäker?
5,5 (ett svar var 5)
Hur intressant är det att veta hur stor del av samäkningen som sker tack vare SpaceTimes tjänster?
3,25 (ett svar var 5)
Hur intressant är det att kunna se hur mycket ni reser under olika tider på dygnet?
2,25
Hur intressant är det att kunna se vilka sträckor som ni reser mest?
Finns det några krav på det er statistik skall användas till? Om ja, vidareutveckla gärna för att vi skall kunna ge er en så bra tjänst som möjligt.

Krav finns ej, önskemål däremot finns att interaktivitet ska förbättras för att göra den enskilda användaren mer delaktig i statistiken och sin egna påverkan på resor.

Kunna mäta CO2 utsläpp och se hur vi väljer fossilfria transporter.

Sammanfattning enkät
Viktigast är koldioxidutsläpp, resekostnader och hur mycket man reser med olika transportmedel. För en kund var information kring samäkning viktigast men denne tyckte likt de andra att de tidigare nämnda punkterna var viktiga.

Minst intressant var det att kunna se när på dygnet man reser samt vilka sträckor man reser.

Vad ska statistik kring ert resande användas till?

- Utveckling av strategier, optimering av fordonssflotta, förbättrad miljöpåverkan med mera.
- Uppföljning av miljöplanen.
- Nytjandegraden på våra tre hyrbilar.
- Sammanställning för beräkning av utsläpp och körsträckor.