TDT4300 - Data Warehousing and Data Mining

Assignment 1

Martin Johannes Nilsen

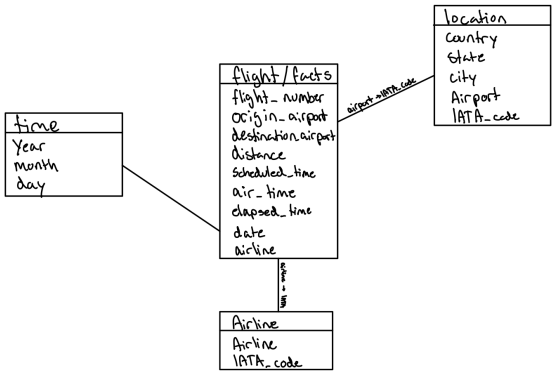
1st February 2022

1 Modeling

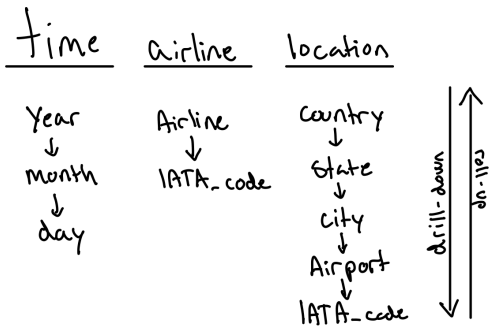
Star schema and concept hierarchies

For creating a star schema we have to find out which of the values/columns is important to know about the requirements for solving the 5 reported analytical tasks. For the longest duration in the air, we have to know the can use air_time, which also has the relation of wheels_on - wheels_off. For the next report question, we got to include both Elapsed_time and airline. Report 3 requires the field month, and report 4 requires the airport. Last, but not least, report 5 require distance flown each month. By summing up the information above, we seem to need the dimension location, time and airline, in addition to the fact table. The airline dimension can be described as a joint table.

This can be illustrated as done below:



Star schema



Concept hierarchy

When it comes to concept hierarchies, we have three dimensions we can go into detail about here. The star schema I drew above was intended to be in a concept hierarchy order based on granularity (depth of detail). I also have included the three concept hierarchies in a compact manner. For the time dimension, we start with the "All"-level at year, before we drill-down to month and day at last. For the location dimension, we start with country, before we can drill further down to state and city, ending up on the airport name and belonging IATA code. The last one, being a joint table for the airline information, gives ut the possibility to roll-up from the very granular IATA code to the airline name being a bit more readable.

2 OLAP Operations

I go into detail of the different operations during the next section, as I felt it was natural to discuss these while presenting the queries and the results. I have also added a little comment at the end of the document with a slight interpretation of the different operations in my own words.

4 Multi-Dimensional Expressions (MDX)

For solving this task, I made a couple of measures during the schema building. Each of them, in addition to the final MDX-queries used, will be described for each of the reports requested in sections below.

Report 1 - The longest duration of any flight in the air

For this one I created a measure taking the MAX() of the air_time-attribute. With this done, the MDX query could simply just select this measure as done below:

```
Select [Measures].[Longest duration] on columns
FROM [Cube]
```

Giving the resulting table:

	All
Longest duration	690

In this report I did not find it necessary to know more than the single measure, but if we wanted to, we could have expanded the query for drilling down into e.g. which airline having the flight with longest duration or the departure/arrival airport of this route.

Report 2 - Average elapsed time for each airline company

For this task, I created a measure taking the average of elapsed time. Combining this with a query which takes the average of elapsed time on columns and the airlines on rows will get the result we want. Below is the final query and result table.

```
Select [Measures].[Average elapsed time] on columns,
[Airline].[Airline] on rows
FROM [Cube]
```

	Average elapsed time
United Air Lines Inc.	191.0209334133974
American Airlines Inc.	171.6737130932138
US Airways Inc.	151.47222675533433
Frontier Airlines Inc.	154.0158112624071
JetBlue Airways	170.5659478117677
Skywest Airlines Inc.	99.74338521129704
Alaska Airlines Inc.	179.2018823068062
Spirit Air Lines	158.61287054409004
Southwest Airlines Co.	121.26467011651499
Delta Air Lines Inc.	142.70279663637837
Atlantic Southeast Airlines	98.60321864971402
Hawaiian Airlines Inc.	101.54818110442292
American Eagle Airlines Inc.	96.44852505328474
Virgin America	208.21361301369862

In the resulting table, we are able to drill down into the codes of each of the airport, ending up with a higher granularity. This is nice as the first level of granularity is more humanly readable, but if we want to use the information to compare to another set of data, we are going to need the higher granularity code.

Report 3 - The total number of flights flown in Februrary

For the third report I continued to experiment with the MDX language. For this task, a new concept of OLAP operations not used in the former tasks where introduced. As we only wanted the total number of flights flown in Februrary, it felt natural to use the slice-operation for just getting the column in the time dimension being February. This also meant that we drill down from the year granularity, to the finer month granularity.

```
Select [Measures].[Total number of flights] on 0,
FROM [Cube]
WHERE{([Time].[Time].[Month].[2015 Feb])}
```

Ending up with the following result table:

	2015 Feb
Total number of flights	970556804

Report 4 - Each Month and the airport with the highest amount of arrival flights

This task was a bigger challenge than the former. I have solved it in one way, by using 3 dimensions and introducing order and topcount (similar to the dice operation). This will limit the amount of rows to the single highest. With that being said, I am fully aware of the fact that I have made a risky assumption in the query. Even though it is a fact in the data we have that the same airport is having the max amount of arrival flights for every month, this can not be guaranteed (a higher chance is not a guarantee!). The solution to this would be to take the max for each column where the airport is the value. I could have done this separately for each month without too much trouble, but am not experienced enough with the tools to create this as one query. It should be mentioned that BDESC breaks the hierarchy.

```
Select [Measures].[Count arrival flights] on 0
[Time].[Time].[Month] on 1,
TOPCOUNT(
    Order(
        [Location].[Location].[Airport],[Measures].[Count arrival flights],
        BDESC)
    ,1)
On 2,
FROM [Cube]
```

	2015 Jan	2015 Feb	2015 Mar	2015 Apr
Hartsfield-Jackson Atlanta International Airport	29512	27366	32754	3236

Report 5 - Descending list of all months by the amount of total distance flown each month

This task was also dependent on order where we wanted it to be descending based on the flight distance. I have drilled down the time dimension from years to months, and ordered the row column descending based on the distance. This can be described in a query as done below:

```
Select [Measures].[Flight distance] on 0,
Order([Time].[Time].[Month], [Measures].[Flight distance], DESC) on 1
FROM [Cube]
```

	Flight distance
> 2015 Mar	411546494
> 2015 Jan	377507097
> 2015 Feb	343689908
> 2015 Apr	41744774

Last comment

In the end I want to briefly described the OLAP operations, even though I applied the theory behind them within each of the reports in subsection 4. Drill-down is to go further in taking a closer look at the insight. We go from a coarse to a finer level. Rolling up is the other way around, going from a month to a year for example. Simply being more generalizing. Dice is like a select operator in relational algebra, taking two or more attributes for getting a subset of the data. Slice is like project in relational algebra, slicing one dimension. Do a slice for time, you get a subset with only time. Lastly, we have the pivot operation which transposes the table to visualize it.