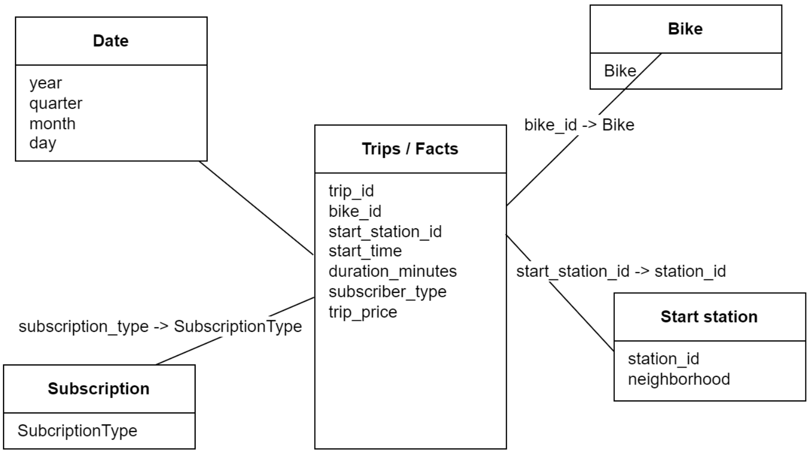
Assignment 1 - TDT4300

Hermann Owren Elton, Olaf Rosendahl

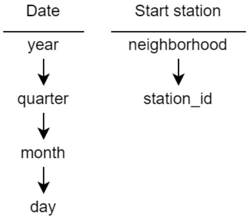
January 31, 2023

1 Modeling

When creating a star schema, one must find out which of the values/columns is needed for the requirements in order to for fill the wanted reports. Trip ids was necessary to count number of trips and date to separate in years. For total duration per bike, bike and duration must be known. Report 3 requires the month/year, trip price and duration, and report 4 requires the subscriber type and trip price. Finally, report 5 require year and the trip's relation to the neighborhood where it starts. With the information above, it's clear that the dimensions Date, Start station, Subscription and Bike is needed, in addition to the fact table. The star schema can be seen below:



(a) Star schema



(b) Concept hierarchy

Regarding concept hierarchies, there is two dimensions one can go into detail about here. For the Date dimension, we start with year as "All"-level, before we drill-down to quarter, month and day. The Start station dimension starts with neighborhood, before drilling down to station_id.

2 OLAP Operations

In the following section we have used different OLAP operation:

1. Drill-down: Allows for futher insight to a dataset, by going deeper into the hirarchy of a given result. This increases the granularity level.
2. Roll-up: Insted of increasing the dimention/granularity level as Drill-down does, Roll-up reduces the dimentions allowing for a wider less general view of the data.
3. Slice: The slice operation gives you a ”slice” of your result, for example if you only want a given year in your dataset by using the where clause you slice out the wanted result form the original query result.
4. Dice: This operation is similar to the slice operation but instead of using only one dimention you can pick multiple dimentions to create your new cube.
5. Pivot: This function transposes a given dataset into something else. You can for example use the * operator which is short for the crossjoin function this function gives to cross product of the two sets its preformed on.

4 Multi-Dimensional Expressions (MDX)

Report 1

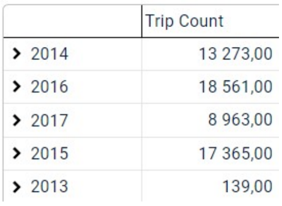
Task: *Trip counts in each year*

To solve this taks we created a mesure that counts the trip ids in the dataset with the COUNT function, when combined with the year level from the that dimention this gives us the trip count for each year.

Query:

```
SELECT
    {[Measures].[Trip Count]} ON COLUMNS,
    {[Date].[Date].[Year]} ON ROWS
FROM [Cube]
```

Results:



	Trip Count
› 2014	13 273,00
› 2016	18 561,00
› 2017	8 963,00
› 2015	17 365,00
› 2013	139,00

The results here allow for drill down to each quarter, and then each month for the given year.

Report 2

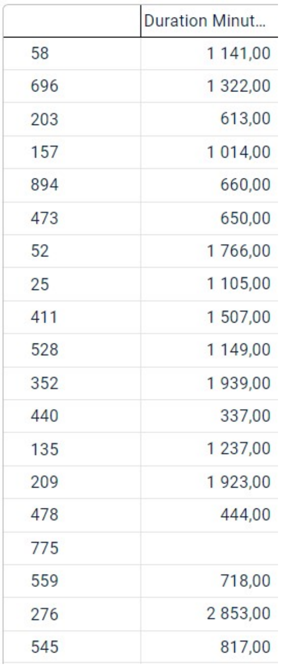
Task: *The total duration each bike was ridden in 2014*

For this task we created a measure that sums up the duration in minutes column, by using the SUM function. When combined with the Bike level on the trip dimention, and then filter the dataset to only contain data for the year with the use of the where clause, we get the total duration each bike was ridden in 2014

Query:

```
SELECT
    {[Measures].[Duration Minutes]} ON COLUMNS,
    {[Trip].[Bike].[Bike]} ON ROWS
FROM [Cube]
   WHERE ([Date].[Year].[2014])
```

Results (not all):



	Duration Minut...
58	1 141,00
696	1 322,00
203	613,00
157	1 014,00
894	660,00
473	650,00
52	1 766,00
25	1 105,00
411	1 507,00
528	1 149,00
352	1 939,00
440	337,00
135	1 237,00
209	1 923,00
478	444,00
775	
559	718,00
276	2 853,00
545	817,00

For this result there is no ability to drill down, because we wanted to look at single bikes.

Report 3

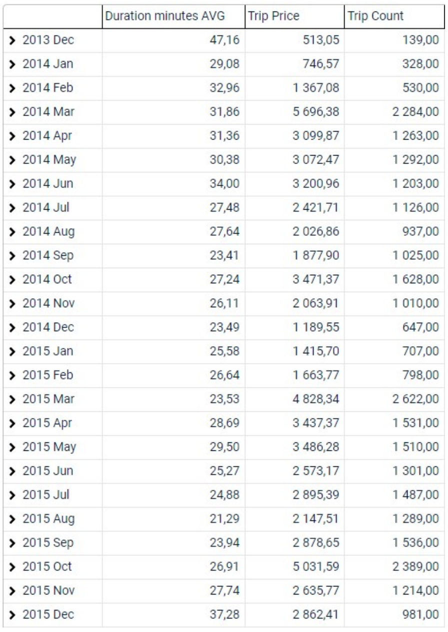
Task: *The trip count, revenue and average trip duration throughout the months from 2013 to 2015. Can you observer some patterns?*

To solve this task we needed three messures the Duration minutes AVG which uses the AVG function, Trip Price, which uses the SUM function to sum ut the trip price column, and lastly the same function from report 1 Trip Count. Combining this with the Month level from the Date hirarchy we get The trip count, revenue and average trip duration for all months in the data set. We then filter the results by using the Year hirarchy in the Date dimention and slice out anything that does not contain the years, 2013, 2014, 2015.

Query:

```
SELECT
   {[Measures].[Duration minutes AVG], [Measures].[Trip Price],
    ↪  [Measures].[Trip Count]} ON COLUMNS,
   {[Date].[Date].[Month]} ON ROWS
FROM [Cube]
   WHERE ({[Date].[Year].[2013], [Date].[Year].[2014], [Date].[Year].[2015]})
```

Results:



	Duration minutes AVG	Trip Price	Trip Count
› 2013 Dec	47,16	513,05	139,00
› 2014 Jan	29,08	746,57	328,00
› 2014 Feb	32,96	1 367,08	530,00
› 2014 Mar	31,86	5 696,38	2 284,00
› 2014 Apr	31,36	3 099,87	1 263,00
› 2014 May	30,38	3 072,47	1 292,00
› 2014 Jun	34,00	3 200,96	1 203,00
› 2014 Jul	27,48	2 421,71	1 126,00
› 2014 Aug	27,64	2 026,86	937,00
› 2014 Sep	23,41	1 877,90	1 025,00
› 2014 Oct	27,24	3 471,37	1 628,00
› 2014 Nov	26,11	2 063,91	1 010,00
› 2014 Dec	23,49	1 189,55	647,00
› 2015 Jan	25,58	1 415,70	707,00
› 2015 Feb	26,64	1 663,77	798,00
› 2015 Mar	23,53	4 828,34	2 622,00
› 2015 Apr	28,69	3 437,37	1 531,00
› 2015 May	29,50	3 486,28	1 510,00
› 2015 Jun	25,27	2 573,17	1 301,00
› 2015 Jul	24,88	2 895,39	1 487,00
› 2015 Aug	21,29	2 147,51	1 289,00
› 2015 Sep	23,94	2 878,65	1 536,00
› 2015 Oct	26,91	5 031,59	2 389,00
› 2015 Nov	27,74	2 635,77	1 214,00
› 2015 Dec	37,28	2 862,41	981,00

For this result you're able to drill down to each specific day of the month for a given month of a year. Giving you three more possible granularity levels.

Report 4

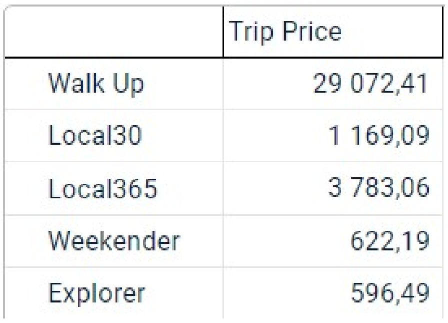
Task: *Subscriber types that generated revenue greater than 400 (USD) in the year 2015*

In this task we reuse the Trip price measure from the previous report, and combine this with the SubscriberType level in the Subscriber hirarchy which is filtered by to only contain relationships where the Trip price is above 400. We then filter the result to only contain the year 2015.

Query:

```
SELECT
   {[Measures].[Trip Price]} ON COLUMNS,
   FILTER([Trip].[Subscriber].[SubscriberType], [Measures].[Trip Price] > 400)
    ↪  ON ROWS
FROM [Cube]
   WHERE ([Date].[Year].[2015])
```

Results:



	Trip Price
Walk Up	29 072,41
Local30	1 169,09
Local365	3 783,06
Weekender	622,19
Explorer	596,49

For this result we only want to give data on the spesific subscriber types which does not have any drill down cabability.

Report 5

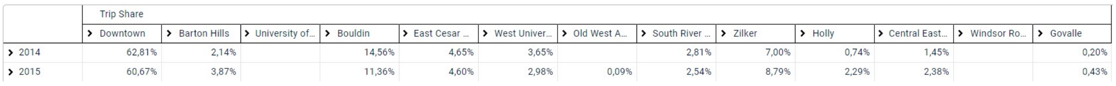
Task: *The share of trips per neighborhood and per year. Consider only years 2014 and 2015.*

In this task we used the WITH function to create advanced measures which are combinations of other measures or dimentions. We combine the Measure Trip Count from the previous reports devided by the total sum of Neighborhood this gives the Trip share pr neighborhood, this is stored in the new measure Trip Share. We then select the Trip Share and multiply them by the neighborhood to spread the trip share to each neighbourhood, we then combine this with the Year level in the Date hirarchy, and lastly filter the result to only contain the year 2015.

Query:

```
WITH
    MEMBER [Measures].[Trip Share] AS ([Measures].[Trip Count]) / SUM([Start
     ↪  station].[Start station].[Neighborhood].MEMBERS , [Measures].[Trip
     ↪  Count] ), FORMAT_STRING = "Percent"
SELECT
   {[Measures].[Trip Share] * [Start station].[Start station].[Neighborhood]}
    ↪  ON COLUMNS,
   NON EMPTY {[Date].[Date].[Year]} ON ROWS
FROM [Cube]
   WHERE ({[Date].[Year].[2014], [Date].[Year].[2015]})
```

Results:



	Trip Share												
	› Downtown	› Barton Hills	› University of...	› Bouldin	› East Cesar ...	› West Univer...	› Old West A...	› South River ...	› Zilker	› Holly	› Central East...	› Windsor Ro...	› Govalle
› 2014	62,81%	2,14%		14,56%	4,65%	3,65%		2,81%	7,00%	0,74%	1,45%		0,20%
› 2015	60,67%	3,87%		11,36%	4,60%	2,98%	0,09%	2,54%	8,79%	2,29%	2,38%		0,43%

For this result you can drill down and see results for each year for a given station id.