TDT4300 - Exercise 1

Andreas B. Berg

2 – Modeling

Given the dataset create a star schema compatible with the requirements above. Also define the concept hierarchies for each dimension. Briefly explain any assumptions you have made. We are primarily looking for you to show modeling principles for data warehousing.

I have decided on a very simple star schema, to avoid splitting tables given in the assignment. I am very keen on getting feedback on whether this is a good idea or not!

There are more dimensions, hierarchies and attributes in the actual cube than mentioned here. These are necessary to answer the assignment, but I have included a couple others for practice. These include dimensions (trip duration, trip price) and hierarchies (start station -> postcode, end station -> postcode).

Note that due to updates to the assignment (see task 3), I could have avoided including end station as well.

Fact table

Trips	
start station	Foreign key to stations
end station	Foreign key to stations
start time	Foreign key to time
subscriber type	
trip count	
trip price sum	
trip price avg	// by mistake called [Trip revenue avg]
trip duration avg	

Dimension tables

Stations
station_id
name
city
region
neighborhood

Time
year
month
datetime

Concept hierarchies – Stations

ALL(station)
region
city
neighborhood
name, station id

Note: I assume that station id and name are 1-to-1, that they have the same granularity and can thus be used as key and value, respectively, for the most granular layer.

Concept hierarchies - Time

Note: All hierarchies are listed with increasing granularity

ALL(time)
year
month
datetime

3 – OLAP Operations

On top of your schema specify sequences of OLAP operations generating Report 1, Report 2 and Report 3. Keep in mind following assumption: For all concept hierarchies assume the "All"-level (level with zero granularity) as default.

NOTE: I am basing my answers on the following quote from the instructors (posted Feb. 26th around 1400 on https://piazza.com/class/kjijho7stlq6qm?cid=76): "[…] for simplifying the queries, whenever you have to consider both start and end station, it is just enough to consider the start station, so the query will be changed to find the most common start station, consider this for any other queries." This means that I will find most frequent start station (not both), etc. While I am very keen on learning how to combine different dimensions in order to see i.e., most likely path (start station -> end station), I do not have the time for it in this assignment.

***Report 1** The most common route in each year. (i.e. most frequent Start Station -> End Station).*

Updated: The most frequent start station in each year

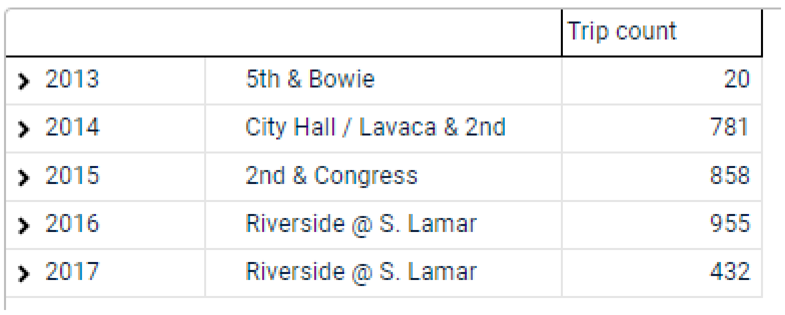
OLAP:

```
Drill-down on time (ALL -> year)
Drill-down on stations (ALL -> name)
(Order by trip count, decreasing)
Slice (first element for each year)
// These last two lines might be an illegal way of doing it, but I’m unsure
// how to order-and-select in OLAP
```

MDX:

```
SELECT
   GENERATE(
      [start time].[time].[year],
      TOPCOUNT(
         [start time].[time].CURRENTMEMBER
            * [start station].[station].[name].MEMBERS
         , 1
         , [Measures].[Trip count]
      )
   ) ON COLUMNS
FROM
   [TDT4300 - Øving 1]
```

Report:



		Trip count
› 2013	5th & Bowie	20
› 2014	City Hall / Lavaca & 2nd	781
› 2015	2nd & Congress	858
› 2016	Riverside @ S. Lamar	955
› 2017	Riverside @ S. Lamar	432

Report 2 *The most busy months in each neighborhood.*

I assume that this means the months with most trips out of a neighborhood.

OLAP sequence / pseudocode:

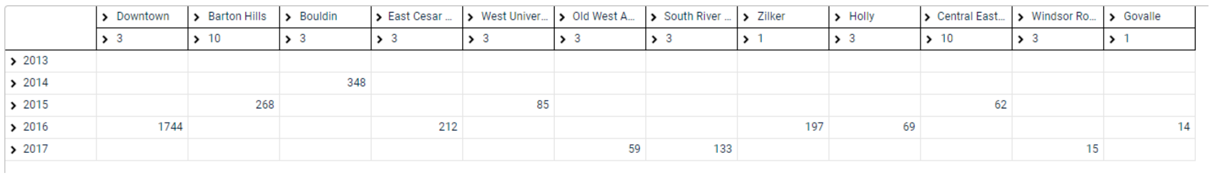
```
Drill-down time (ALL -> month)
Drill-down start station (ALL -> neighborhood)
(Order by trip count, decreasing)
Slice (first element for each station)
// These last two lines might be an illegal way of doing it, but I’m unsure
// how to order-and-select in OLAP
```

MDX:

The following gets the month with the most trips from a station, for all non-empty stations (meaning, stations not listed had no trips from the station). Note that I have opted to have years on rows, as the months display as only the month, without any information about what year it is (without hovering over it):

```
SELECT
   [start time].[time].[year] ON ROWS,
   GENERATE(
      NONEMPTY([start station].[station].[neighborhood].MEMBERS),
      TOPCOUNT(
      [start station].[station].CURRENTMEMBER
            * [start time].[time].[month].MEMBERS
      , 1
      , [Measures].[Trip count]
      )
   ) ON COLUMNS
FROM
   [TDT4300 - Øving 1]
```

Report:



	Downtown	Barton Hills	Bouldin	East Cesar ...	West Univer...	Old West A...	South River ...	Zilker	Holly	Central East...	Windsor Ro...	Govalle
	3	10	3	3	3	3	3	1	3	10	3	1
2013												
2014			348									
2015		268			85					62		
2016	1744			212				197	69			14
2017						59	133				15	

Report 3 *The average trip duration, average revenue throughout months from 2015 to 2017.*

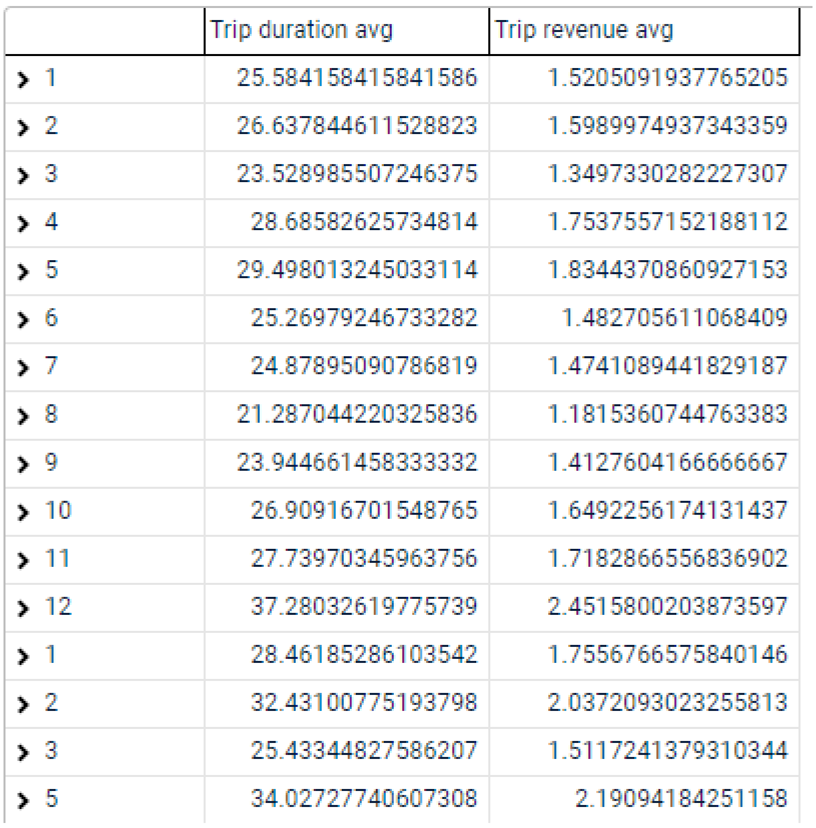
OLAP sequence / pseudocode:

```
Drill-down time (ALL -> months)
Slice time (range 2015 : 2017)
(Aggregate average trip duration and trip revenue)    // Previously calculated
```

MDX:

```
SELECT
   [start time].[time].[2015].FIRSTCHILD :
      [start time].[time].[2017].LASTCHILD on ROWS,
   {[Measures].[Trip duration avg], [Measures].[Trip revenue avg]} on COLUMNS
FROM
   [TDT4300 - Øving 1]
```

Report:



	Trip duration avg	Trip revenue avg
› 1	25.584158415841586	1.5205091937765205
› 2	26.637844611528823	1.5989974937343359
› 3	23.528985507246375	1.3497330282227307
› 4	28.68582625734814	1.7537557152188112
› 5	29.498013245033114	1.8344370860927153
› 6	25.26979246733282	1.482705611068409
› 7	24.87895090786819	1.4741089441829187
› 8	21.287044220325836	1.1815360744763383
› 9	23.944661458333332	1.4127604166666667
› 10	26.90916701548765	1.6492256174131437
› 11	27.73970345963756	1.7182866556836902
› 12	37.28032619775739	2.4515800203873597
› 1	28.46185286103542	1.7556766575840146
› 2	32.43100775193798	2.0372093023255813
› 3	25.43344827586207	1.5117241379310344
› 5	34.02727740607308	2.19094184251158

(continuing through, but I figured it is not necessary to include the whole thing here)

5 – Multi-Dimensional Expressions (MDX)

Deploy your schema and switch to the MDX tab. Finally, define and execute MDX queries generating the requested reports.

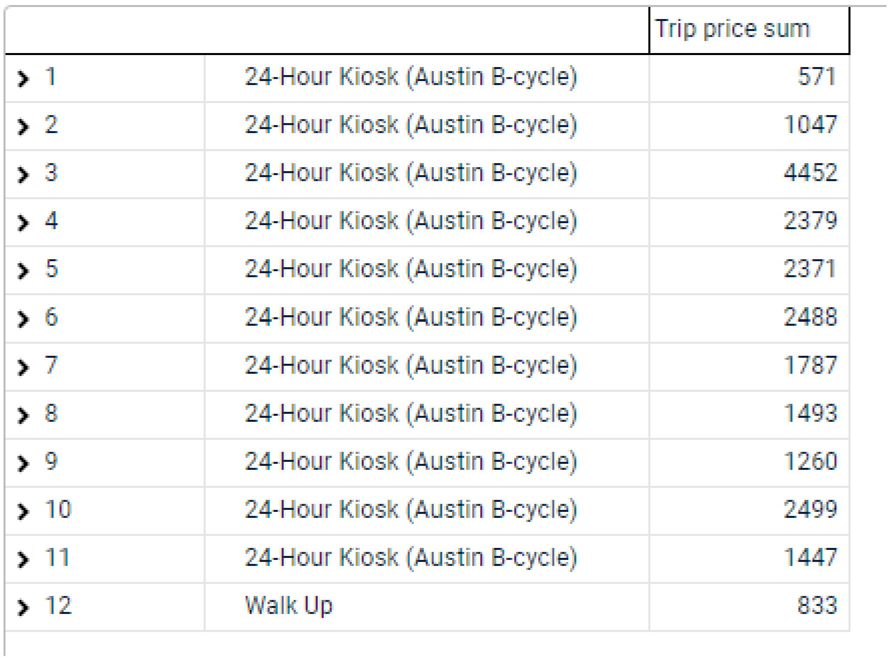
See above for MDX queries for report 1-3.

Report 4 *Subscriber types generated the greatest revenue throughout the months of year 2014.*

MDX:

```
SELECT
   GENERATE(
      [start time].[time].[2014].CHILDREN,
      TOPCOUNT(
         [start time].[time].CURRENTMEMBER * [Subscriber type].[type].[type]
         , 1
         , [Measures].[Trip price sum]
      )
   ) on ROWS,
   [Measures].[Trip price sum] on COLUMNS
FROM
   [TDT4300 - Øving 1]
```

Report:



		Trip price sum
1	24-Hour Kiosk (Austin B-cycle)	571
2	24-Hour Kiosk (Austin B-cycle)	1047
3	24-Hour Kiosk (Austin B-cycle)	4452
4	24-Hour Kiosk (Austin B-cycle)	2379
5	24-Hour Kiosk (Austin B-cycle)	2371
6	24-Hour Kiosk (Austin B-cycle)	2488
7	24-Hour Kiosk (Austin B-cycle)	1787
8	24-Hour Kiosk (Austin B-cycle)	1493
9	24-Hour Kiosk (Austin B-cycle)	1260
10	24-Hour Kiosk (Austin B-cycle)	2499
11	24-Hour Kiosk (Austin B-cycle)	1447
12	Walk Up	833

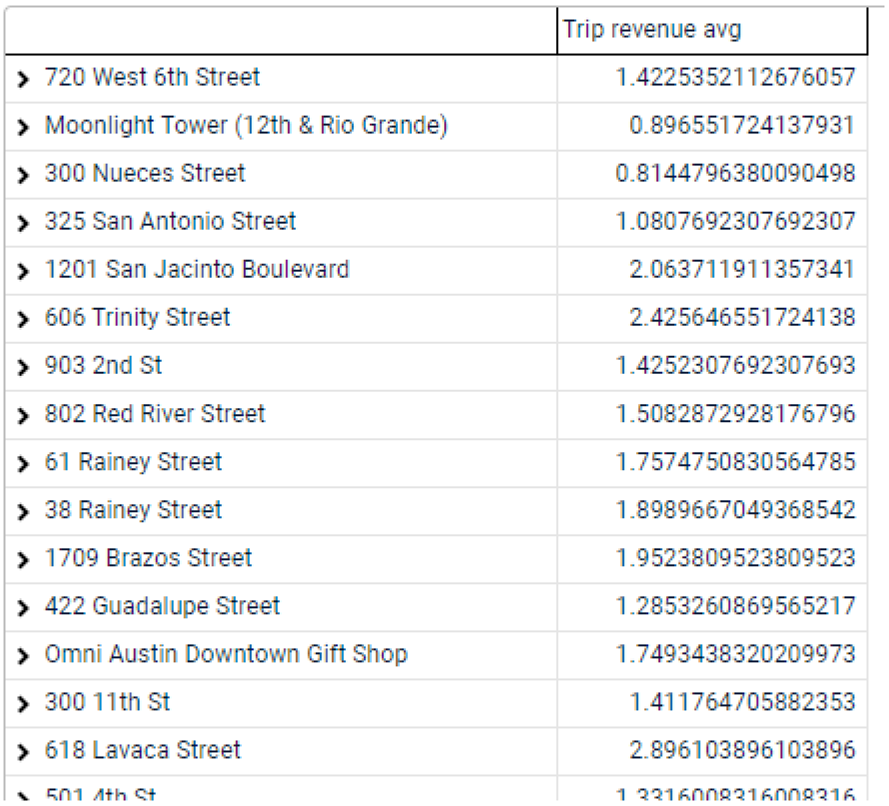
Report 5 *Average revenue generated from the stations(stations which are the start point or the end point of the trip) resides in "Downtown" neighborhood throughout years 2014 to 2016.*

I assume this wants us to find the average revenue (per trip) generated on trips starting on each station in Downtown. There are two ways to display this – as the combined average revenue for the whole period, or as the average revenue for each month. Below are both.

Combined average revenue:

```
SELECT
      -- Remove NONEMPTY to see all stations:
   NONEMPTY([start station].[station].[Downtown].CHILDREN) on COLUMNS
FROM (
   SELECT [start time].[time].[2014].FIRSTCHILD
      : [start time].[time].[2016].LASTCHILD on COLUMNS
   FROM [TDT4300 - Øving 1]
)
WHERE
   [Measures].[Trip revenue avg]
```

Report:



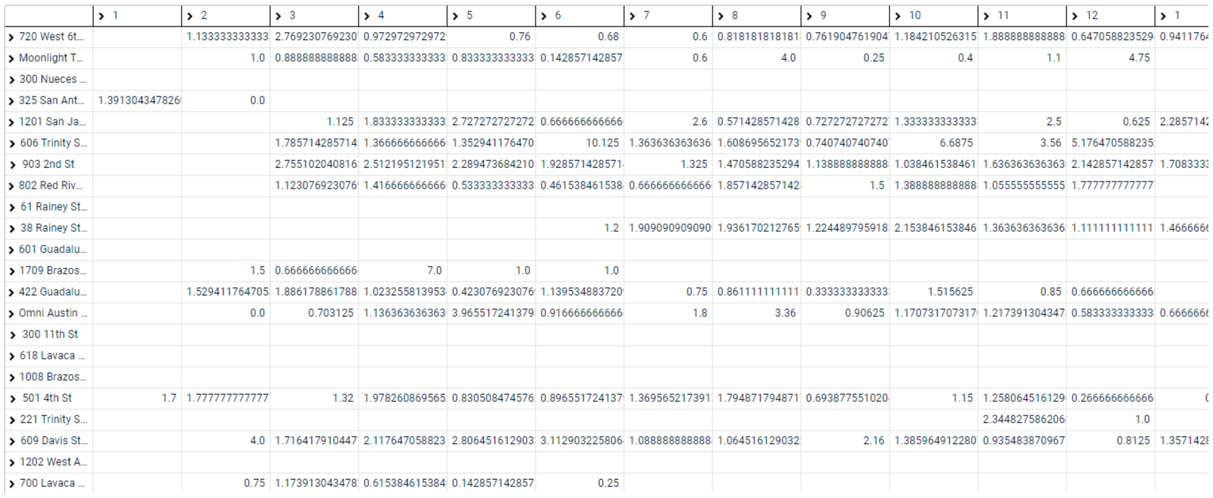
	Trip revenue avg
720 West 6th Street	1.4225352112676057
Moonlight Tower (12th & Rio Grande)	0.896551724137931
300 Nueces Street	0.8144796380090498
325 San Antonio Street	1.0807692307692307
1201 San Jacinto Boulevard	2.063711911357341
606 Trinity Street	2.425646551724138
903 2nd St	1.4252307692307693
802 Red River Street	1.5082872928176796
61 Rainey Street	1.7574750830564785
38 Rainey Street	1.8989667049368542
1709 Brazos Street	1.9523809523809523
422 Guadalupe Street	1.2853260869565217
Omni Austin Downtown Gift Shop	1.7493438320209973
300 11th St	1.411764705882353
618 Lavaca Street	2.896103896103896
501 4th St	1.3316008316008316

(it continues for all stations in Downtown)

Average revenue for each month:

```
SELECT
      -- Remove NONEMPTY to see all stations:
   NONEMPTY([start station].[station].[Downtown].CHILDREN) on ROWS,
   [start time].[time].[2014].FIRSTCHILD
      : [start time].[time].[2016].LASTCHILD on COLUMNS
FROM
   [TDT4300 - Øving 1]
WHERE
   [Measures].[Trip revenue avg]
```

Report:



	› 1	› 2	› 3	› 4	› 5	› 6	› 7	› 8	› 9	› 10	› 11	› 12	› 1
› 720 West 6t...		1.133333333333	2.769230769230	0.972972972972	0.76	0.68	0.6	0.818181818181	0.761904761904	1.184210526315	1.888888888888	0.647058823529	0.9411764
› Moonlight T...		1.0	0.888888888888	0.583333333333	0.833333333333	0.142857142857	0.6	4.0	0.25	0.4	1.1	4.75	
› 300 Nueces ...													
› 325 San Ant...	1.391304347826	0.0											
› 1201 San Ja...			1.125	1.833333333333	2.727272727272	0.666666666666	2.6	0.571428571428	0.727272727272	1.333333333333	2.5	0.625	2.285714
› 606 Trinity S...			1.785714285714	1.366666666666	1.352941176470	10.125	1.363636363636	1.608695652173	0.740740740740	6.6875	3.56	5.176470588235	
› 903 2nd St			2.755102040816	2.512195121951	2.289473684210	1.928571428571	1.325	1.470588235294	1.138888888888	1.038461538461	1.636363636363	2.142857142857	1.708333
› 802 Red Riv...			1.123076923076	1.416666666666	0.533333333333	0.461538461538	0.666666666666	1.857142857142	1.5	1.388888888888	1.055555555555	1.777777777777	
› 61 Rainey St...													
› 38 Rainey St...						1.2	1.909090909090	1.936170212765	1.224489795918	2.153846153846	1.363636363636	1.111111111111	1.466666
› 601 Guadalu...													
› 1709 Brazos...		1.5	0.666666666666	7.0	1.0	1.0							
› 422 Guadalu...		1.529411764705	1.886178861788	1.023255813953	0.423076923076	1.139534883720	0.75	0.861111111111	0.333333333333	1.515625	0.85	0.666666666666	
› Omni Austin ...		0.0	0.703125	1.136363636363	3.965517241379	0.916666666666	1.8	3.36	0.90625	1.170731707317	1.217391304347	0.583333333333	0.666666
› 300 11th St													
› 618 Lavaca ...													
› 1008 Brazos...													
› 501 4th St	1.7	1.777777777777	1.32	1.978260869565	0.830508474576	0.896551724137	1.369565217391	1.794871794871	0.693877551020	1.15	1.258064516129	0.266666666666	(
› 221 Trinity S...											2.344827586206	1.0	
› 609 Davis St...		4.0	1.716417910447	2.117647058823	2.806451612903	3.112903225806	1.088888888888	1.064516129032	2.16	1.385964912280	0.935483870967	0.8125	1.357142
› 1202 West A...													
› 700 Lavaca ...		0.75	1.173913043478	0.615384615384	0.142857142857	0.25							

(it continues for all stations in Downtown and all months in the period)

Report 6 *The most Busy hours in a day on a yearly basis in each neighborhood.*

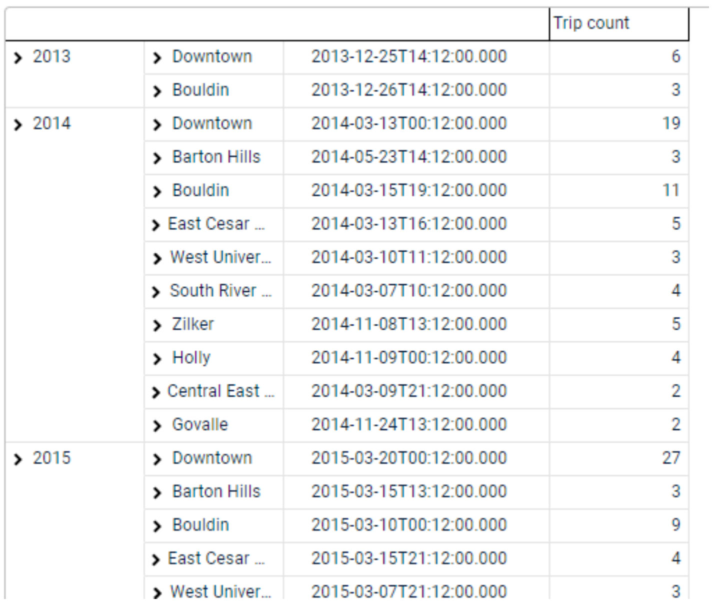
Updated: The hour with the most trips starting at a station, for each year and each neighborhood

I assume that, although start_time is listed as a DateTime, it is limited to hour-wise values (meaning, it is limited to one value per hour). This seems to be the case for this dataset, based on a quick overview.

I assume that the assignment wants us to find the busiest hour per year:

```
SELECT
   GENERATE(
      NONEMPTY([start time].[time].[year]
            * [start station].[station].[neighborhood]),
      TOPCOUNT(
         ([start time].[time].CURRENTMEMBER
            , [start station].[station].CURRENTMEMBER)
            * [start time].[time].[datetime]
         , 1
         , [Measures].[Trip count]
      )
   ) ON COLUMNS
FROM
   [TDT4300 - Øving 1]
```

Report:



			Trip count
2013	Downtown	2013-12-25T14:12:00.000	6
	Bouldin	2013-12-26T14:12:00.000	3
2014	Downtown	2014-03-13T00:12:00.000	19
	Barton Hills	2014-05-23T14:12:00.000	3
	Bouldin	2014-03-15T19:12:00.000	11
	East Cesar ...	2014-03-13T16:12:00.000	5
	West Univer...	2014-03-10T11:12:00.000	3
	South River ...	2014-03-07T10:12:00.000	4
	Zilker	2014-11-08T13:12:00.000	5
	Holly	2014-11-09T00:12:00.000	4
	Central East ...	2014-03-09T21:12:00.000	2
	Govalle	2014-11-24T13:12:00.000	2
2015	Downtown	2015-03-20T00:12:00.000	27
	Barton Hills	2015-03-15T13:12:00.000	3
	Bouldin	2015-03-10T00:12:00.000	9
	East Cesar ...	2015-03-15T21:12:00.000	4
	West Univer...	2015-03-07T21:12:00.000	3

(it continues for all non-empty tuples (year, neighborhood))

One could argue that this assignment is about finding the busiest hour in a day (i.e., 14-15), for each year and each neighborhood, and not – as I found above – the single busiest hour in a year. I agree with these findings. That being said, I simply did not find any way to compute this. I have tried generating, I have tried subqueries, I have tried converting keys and values to strings and integers and then back again. I have read probably every single Stack Overflow thread and blog post about MDX, and still no solution.

I have narrowed my problem down to one thing: Grouping the data based on hour in the day, and not the complete DateTime. The issue arises in that I don't know an effective way to handle DateTimes, and as such I do not know how to group by or summarize over the values. If this was, say, Python, I would solve this by looping and summing up the values in an array, but that does not – as far as I know – work with MDX. I simply could not find a way to do this, and greatly look forward to seeing the solution to this problem, to see how I should have done it.

This is my final attempt at this. I know it is wrong (it will not even run), but I figure it will at least show how I am thinking about and attacking this:

```
SELECT
   GENERATE(
      NONEMPTY([start time].[time].[datetime]),
      SUM(
         IIF(StrToValue(DateToString([start time].[time].[datetime].CHILDREN,
'HH')) = 0
            , [Measures].[Trip count]
            , 0
      )
   )
) ON COLUMNS
FROM
   [TDT4300 - Øving 1]
```