Jan Gniedziejko & Dominika Kiejdo

Data Warehouse Optimization - report

1. Aim of the laboratory

The aim of the task is to show issues concerning various physical cube models and aggregation design

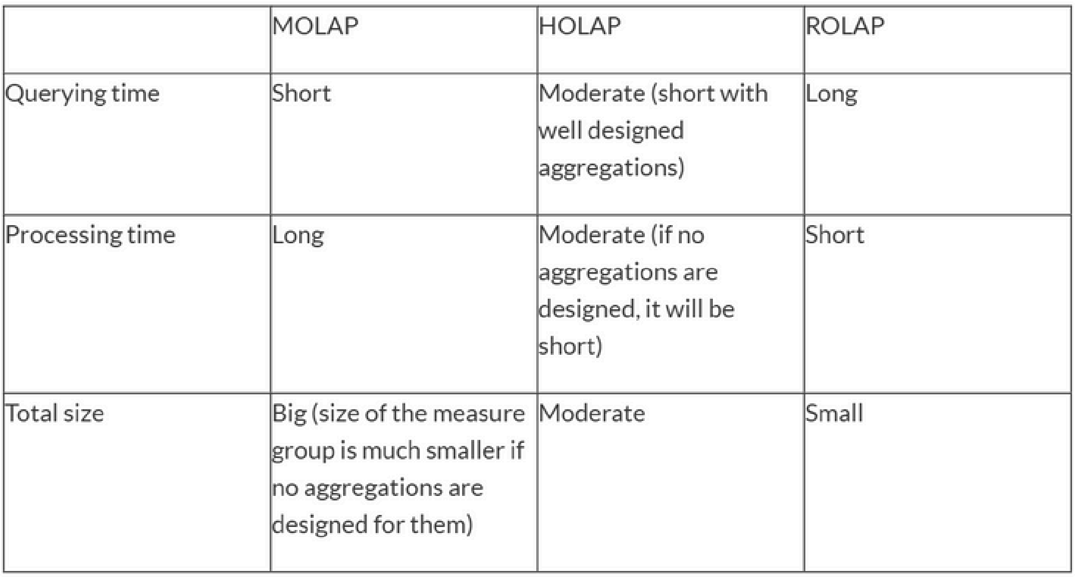
2. Preliminary Assumptions

Size of the database (Data Warehouse): 4176.00 MB
Number of rows in monitoring fact table (our main): 1307900

Testing environment:

Measurements were taken on MacBook Air equipped with an Intel Core i5-1030NG7 processor, 8GB of RAM and 256GB of Internal Storage. We used Windows 10 via Boot Camp Assistant (the specifications remained the same). To evaluate the processing time of a cube, we used SQL Server Management Studio with extension SQL Server Profiler. During the measurements, the only active applications on laptop were SSMS and a web browser with opened instructions, and Visual Studio 2019.

3. Theoretical Assumptions



	MOLAP	HOLAP	ROLAP
Querying time	Short	Moderate (short with well designed aggregations)	Long
Processing time	Long	Moderate (if no aggregations are designed, it will be short)	Short
Total size	Big (size of the measure group is much smaller if no aggregations are designed for them)	Moderate	Small

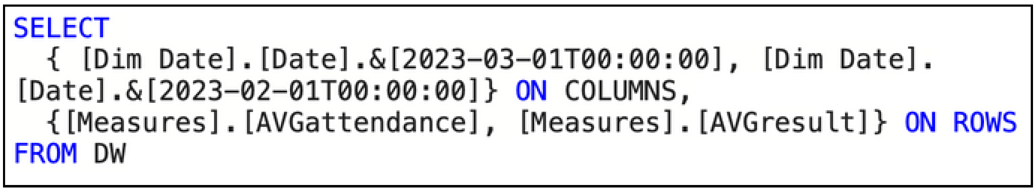
4. Testing

Testing query execution times for different models, with and without defined aggregations.
Testing cube processing times in the same testing settings

Brief description of the queries:

1. [Dates]

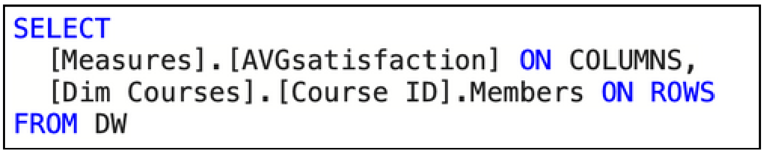
Query that shows what is the correlation in average attendance and average result in march 2023



```
SELECT
  { [Dim Date].[Date].&[2023-03-01T00:00:00], [Dim Date].
[Date].&[2023-02-01T00:00:00]} ON COLUMNS,
  {[Measures].[AVGattendance], [Measures].[AVGresult]} ON ROWS
FROM DW
```

2. [Particular dimension attribute]

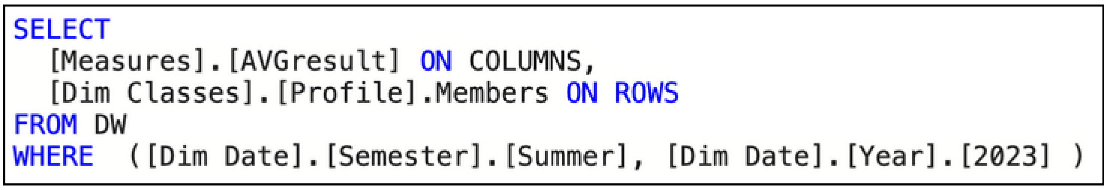
Query that shows what is the average satisfaction in every course



```
SELECT
  [Measures].[AVGsatisfaction] ON COLUMNS,
  [Dim Courses].[Course ID].Members ON ROWS
FROM DW
```

3. [General one]

Query that shows what were the average results in each profile in summer 2023



```
SELECT
  [Measures].[AVGresult] ON COLUMNS,
  [Dim Classes].[Profile].Members ON ROWS
FROM DW
WHERE  ([Dim Date].[Semester].[Summer], [Dim Date].[Year].[2023] )
```

To achieve optimal results of the processing time of a cube we decided to take approximately 10 samples for each modification. The obtained results are presented in the following table

Table 3.1. Processing time of cube and queries for MOLAP and ROLAP with and without aggregations

	Cube Processing		Query 1		Query 2		Query 3	
	Molap	Rolap	Molap	Rolap	Molap	Rolap	Molap	Rolap
Without aggregations	4896	1251	71	611	106	221	80	154
	4707	1623	40	92	73	125	44	116
	5093	1342	54	126	71	135	48	101
	4533	1601	53	104	69	146	52	102
	4766	1354	42	96	61	120	42	116
	5004	1419	44	102	66	122	42	111
	4494	1541	41	111	61	128	52	122
	4762	1566	51	103	66	127	45	129
	4962	1559	41	125	60	126	56	109
	6102	1648	65	108	65	124	46	110
With aggregations	4933	1900	37	100	56	126	40	107
	5125	1686	40	100	54	124	41	112
	6246	1779	40	98	58	127	41	108
	5181	1627	38	99	66	122	45	110
	5296	1769	35	97	61	132	46	107
	5460	1994	39	101	61	119	38	112
	5289	1725	37	106	61	123	37	108
	5271	1755	39	96	57	129	49	107
	5030	1835	39	98	60	121	42	109
	5055	1724	38	100	54	120	41	107

Afterwards, we decided to exclude outliers and calculate the mean and standard deviation for each column. The results are summarized in the following tables:

Tables 3.2. and 3.3. Mean and standard deviation of processing time of cube and queries for MOLAP and ROLAP with and without aggregations

	Cube Processing		Query 1		Query 2		Query 3	
Without agregation	Molap	Rolap	Molap	Rolap	Molap	Rolap	Molap	Rolap
Mean	4801,89	1517,00	45,75	107,44	65,78	128,11	47.44	117.00
SD	205,87	115,65	5,90	11,71	4,60	7,93	4.93	15.53

	Cube Processing		Query 1		Query 2		Query 3	
With agregation	Molap	Rolap	Molap	Rolap	Molap	Rolap	Molap	Rolap
Mean	5288.60	1779.40	38,20	99,50	58,80	124,30	42,00	108.70
SD	102,24	120,48	1,55	2,76	3,74	4,16	3,68	2.00

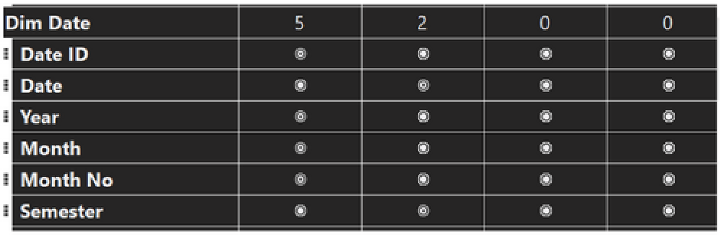
Table 3.4. Average time of processing cube and queries using MOLAP and ROLAP with and without aggregations

	MOLAP		ROLAP	
	Aggr.	No aggr.	Aggr.	No aggr.
Querying speed (for 3 different queries)	38,20	45,75	99.50	107,44
	58,80	65,78	124.30	128,11
	42,00	47,44	108.70	117.00
Processing time	5182,22	4801,89	1779,40	1517,00

5. Cache and aggregation settings

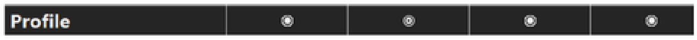
Aggregations:

aggregations made on monitoring table:
date
semester
profile

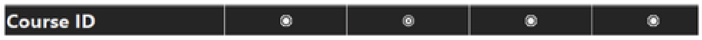


Dim Date	5	2	0	0
Date ID	◎	◉	◉	◉
Date	◉	◎	◉	◉
Year	◎	◉	◉	◉
Month	◎	◉	◉	◉
Month No	◎	◉	◉	◉
Semester	◉	◎	◉	◉

aggregations that were made on satisfaction table:
class id

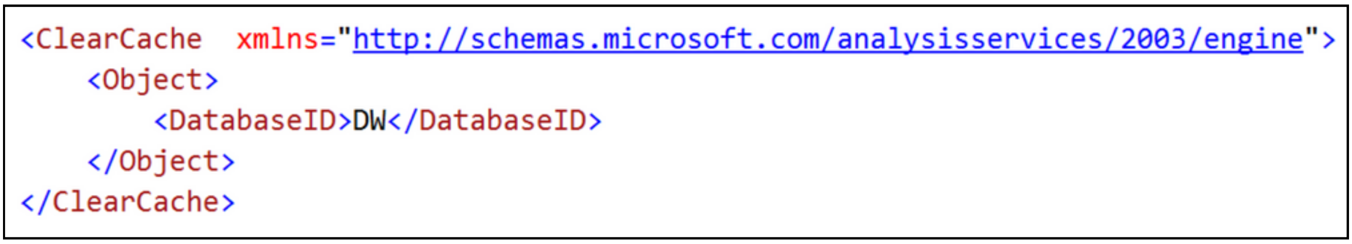


Profile	◉	◎	◉	◉



Course ID	◉	◎	◉	◉

Deleting cache memory:



```
<ClearCache  xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
    <Object>
        <DatabaseID>DW</DatabaseID>
    </Object>
</ClearCache>
```

6. Discussion

Processing the cube

Molap	Rolap
4801,89	1517,00

To sum it up, when it comes to processing our cube, the required time for MOLAP is roughly 3 times longer than for ROLAP (for both specifications: with and without aggregations). It aligns with the assumptions of both methods, as MOLAP transfer all the data from database into the cube, where in contrast ROLAP extracts only metadata indicating the location of specific information, leading to a shorter processing time.

Query execution

MOLAP	ROLAP
45,75	107,44
65,78	128,11
47,44	112,89

Since MOLAP takes more time to transfer all the data into the cube, it makes it up by accelerating the speed of query execution. This method doesnt need to go back to the database to take necessary (for query) information, since all necessary data is stored within the multidimensional cube, it does not need to access the database for query information. Therefore, by looking at our obtained results at the table above, we can clearly see the difference between Multidimensional and Relational OLAP.

Aggregations

	MOLAP	
	Without agregation	With agregation
Cube Processing	4801,89	5288.60
Query 1	45,75	38,20
Query 2	65,78	58,80
Query 3	47,44	42,00

	ROLAP	
	Without agregation	With agregation
Cube Processing	1517,00	1779,40
Query 1	107,44	99.50
Query 2	128,11	124.30
Query 3	117.00	108.70

At our first try, the time of query execution with aggregations (for both ROLAP and MOLAP) came out longer than for models without pre-defined aggregations. However, it turned out that we were defining the aggregations in a wrong way and after doing it over again, we obtained the results that are visible in the tables above. It can be noticed that for each query the average time of execution was shorter, when the models had defined aggregations. It is caused by the fact, that during query execution, the program didnt need to group measures by dimension members, because it was already done while processing the cube. That's why you can notice that average time of processing cube was longer, when with defined aggregations. The extent of query execution time reduction due to aggregations can vary. This variability is likely influenced by the complexity and nature of the queries. For example, queries that involve dividing measures into a large number of groups (e.g., 300 groups when dividing by students) may benefit more from aggregations compared to queries that involve fewer groups (e.g., 6 groups when dividing by profiles).

Conclusion

The tests conducted provided valuable insights into the performance characteristics of different physical cube models and the impact of aggregations. The results aligned well with theoretical assumptions:
MOLAP: Longer processing time but faster query execution.
ROLAP: Shorter processing time but slower query execution.
Aggregations: Increase processing time but reduce query execution time