Mariusz Godlewski - 197816 & Emir Cihan Polat - 207353

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): 176,80 MB
Number of rows in monitoring fact table (our main): 1799776
Second 397874

Testing environment:
Measurements were taken on a Lenovo laptop equipped with an Intel Core i5-12500H processor, 16GB of RAM and 512GBof Internal Storage. We used Windows 11 HOME 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 the laptop were SSMS and a web browser with opened instructions, and Visual Studio 2019.

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 shows summed sales in given Months

```
SELECT
  [Measures].[Total Sales] ON COLUMNS,
  [Sale Date].[Date Hierarchy].[Month].MEMBERS ON ROWS
FROM [DW MARKETING]
```

2. [Dim]

Query shows total cost of marketing for different property type

```
SELECT
  [Measures].[Total Cost] ON COLUMNS,
  [Property].[Type].MEMBERS ON ROWS
FROM [DW MARKETING]
```

3. (general one)
Query shows Top 3 performing campaigns out of all campaigns in terms of dollar spend to views generated

```
SELECT
    { [Measures].[ViewsPerDollar] } ON COLUMNS,
    TOPCOUNT(
        [Property].[Physical Property Id].MEMBERS,
        3,
        [Measures].[ViewsPerDollar]
    ) ON ROWS
FROM [DW MARKETING]
WHERE
    {
        [Campaign End date].[Month].LASTCHILD.LAG(11) :
        [Campaign End date].[Month].LASTCHILD
    }
```

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**, **MOLAP scheduled, MOLAP automatic** with and without aggregations

Without aggregation	Cube processing			With aggregations	Cube processing		
	MOLAP	Molap scheduled	Molap automatic		MOLAP	Molap scheduled	Molap automatic
	10050	10238	8512		15668	16296	15571
	9963	7765	8926		15743	14562	15498

	9960	10187	8797		14434	15519	15527
	8723	10312	8969		15322	15855	16410
	9543	10845	8834		15711	16543	15453
	9562	10756	8756		15893	15094	15589
	9923	10923	8984		15224	14321	15564
	10030	10246	8952		14678	15896	15198
	8862	10423	8876		14980	15678	15415
	9106	11032	8812		15432	15505	15589

Without aggregation	Query 1			Query 2			Query 3		
	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic
	264	331	284	477	469	471	3095	3310	3202
	261	243	333	372	552	518	3104	3261	3098
	262	306	357	414	558	486	2977	3258	2995
	288	265	275	375	452	547	2934	3229	3063
	182	356	334	355	495	423	3425	3333	3161
	202	304	234	521	413	499	3155	3234	3196
	252	297	354	486	456	520	3233	3342	3053
	356	342	361	480	489	483	2579	3378	3144
	195	332	387	550	523	501	3072	3212	2998
	263	286	276	436	554	539	2984	3298	3106

With aggregation	Query 1			Query 2			Query 3		
	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic
	339	298	322	534	473	366	3076	3092	3169
	301	323	269	378	519	503	3174	3149	3242
	294	221	217	480	331	519	3067	3224	3283
	271	236	299	427	441	391	3324	3321	3103
	343	195	271	502	412	502	3277	3273	3141
	300	274	256	445	438	410	3204	3240	3134
	309	245	243	498	470	457	3140	3209	3295
	295	290	218	525	502	487	3290	3099	3250
	275	299	312	512	448	445	3364	3154	3214
	333	156	298	409	406	471	3215	3287	3290

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**, **MOLAP scheduled, MOLAP automatic** with and without aggregations

	Cube processing				Cube processing		
Without aggregation	MOLAP	Molap scheduled	Molap automatic	With aggregations	MOLAP	Molap scheduled	Molap automatic
Mean	9572,2	10272,7	8841,8	Mean	15308,5	15526,9	15581,4
SD	506,39	935,86	139,80	SD	484,70	704,26	314,15

Without aggregation	Query 1			Query 2			Query 3		
	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic
Mean	252,5	306,2	319,5	446,6	496,1	498,7	3055,8	3285,5	3101,6
SD	50,78	35,33	49,11	66,73	49,75	36,04	219,90	55,14	74,76

With aggregation	Query 1			Query 2			Query 3		
	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic	MOLAP	Molap scheduled	Molap automatic
Mean	306	253,7	270,5	471	444	455,1	3213,1	3204,8	3212,1
SD	25,18	52,70	37,31	53,17	53,55	51,68	101,20	78,99	70,89

Table 3.4. Average time of processing cube and queries using **MOLAP**, **MOLAP scheduled, MOLAP automatic** with and without aggregations

	MOLAP		Molap scheduled		Molap automatic	
	Aggr.	No Aggr.	Aggr.	No Aggr.	Aggr.	No Aggr.
Querying speed (for 3 different queries)	306	252,5	253,7	306,2	270,5	319,5
	471	446,6	444	496,1	455,1	498,7
	3213,1	3055,8	3204,8	3285,5	3212,1	3101,6
Processing Time	15308,5	9572,2	15526,9	10272,7	15581,4	8841,8

5. Cache

```
<ClearCache
xmlns="http://schemas.microsoft.com/analysisservices/2003
```

```
/engine">
     <Object>
          <DatabaseID>Marketing</DatabaseID>
     </Object>
</ClearCache>
```

6. Discussion (comparison of the theory with the obtained results)

Cube Processing: Aggregations increased processing time across all MOLAP modes. The fastest processing was achieved using MOLAP automatic without aggregations (8841.8 ms on average), as this mode is optimized for minimal processing overhead.

Query Performance: Aggregations improved query speed, especially for more complex queries like Query 3. For instance, Query 1 in MOLAP scheduled dropped from 306,2 ms to 254.7 ms with aggregations. Yet not always acted as expected.

Mode Comparison:

> MOLAP automatic had the best processing times without aggregations.
>
> MOLAP scheduled provided the most consistent query times when using aggregations.

Query performance was generally better with aggregations, but at the cost of longer cube processing.

In summary, using aggregations improves query speed and stability, while avoiding them reduces processing time. The optimal configuration depends on whether the priority is faster cube updates or faster query execution.