

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 data warehouse:

Data Warehouse contains:

- 26 280 tuples in Dim_date (dimension),
- 24 tuples in Dim_time (dimension),
- 1 500 tuples in Dim_mechanic (dimension),
- 200 000 tuples in Dim_part (dimension),
- 2000 tuples in Dim_plane (dimension),
- 200 000 tuples in Dim_test (dimension),
- 1 000 000 tuples in F_plane_repair (fact table),
- 2 999 745 tuples in F_making_repair (fact table),
- 3 000 064 tuples in F_parts_used_in_repair (fact table),

Testing environment:

- SQL Server Profiler 18
- Microsoft SQL Server Management Studio 18
- Visual Studio 2019 SSAS

3. 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. Zbadaj zależność skuteczności napraw od liczby mechaników uczestniczących w naprawach w poszczególnych miesiącach w ostatnim roku.
2. Która część ulegała awariom najczęściej podczas ostatniego miesiąca, a która najmniej?
3. Dla każdego samolotu wylicz ile procent jego napraw zakończyło się sukcesem?

	MOLAP		ROLAP		HOLAP	
	Aggr.	No aggr.	Aggr.	No aggr.	Aggr.	No aggr.
Querying speed (for 3 different queries)	14 ms	40 ms		212 ms	16 ms	212 ms
	751 ms	772 ms		2412 ms	2116 ms	2378 ms
	46 ms	97 ms		125 ms	48 ms	103 ms
Processing time	27120 ms	20200 ms		7710 ms	14881 ms	14063 ms
Total size	334,09 MB	332,86 MB		235,12 MB	236,28 MB	234,99 MB

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

Firstly, let's compare the total size. MOLAP has the biggest total size because it stores all data of measure group, copies of the fact tables and all aggregations calculated during the processing of the cube in the analytical database. HOLAP and ROLAP are approximately 30% smaller than MOLAP because there are no copies of fact tables and group of measures in their analytical database. However HOLAP stores aggregations in its analytical database, that's why its size with aggregations is a bit bigger than without them. The same situation happens in case of MOLAP.

When it comes to processing time, the longest was achieved by MOLAP. It is, because MOLAP has to duplicate data from data warehouse to the analytical database. With aggregations MOLAP has approximately 35% longer time of processing because it also stores all aggregations in the analytical database. HOLAP has 30% shorter processing time than MOLAP because it doesn't store copies of the same data in data warehouse and analytical database. The time is a bit longer with aggregations because HOLAP stores aggregations in the analytical database. The shortest time of processing has ROLAP (almost 50% shorter than HOLAP), because it stores all data in a data warehouse.

Next thing to consider is the querying time measured for 3 different MDX queries. For each query, MOLAP reached the shortest time. The reason why is that it stores all data and aggregations in the analytical database and doesn't have to connect to data warehouse for the information. That's also why the querying time with aggregations is much shorter than without them. ROLAP and HOLAP have comparable querying time but both has much longer than MOLAP. Moreover ROLAP has a bit longer time than HOLAP. This is, because they must read data from data warehouse. HOLAP is a bit faster than ROLAP because it stores at least aggregations in the analytical database. That's also the reason why the queries with aggregations defined are performing much faster than without aggregations. In the case of ROLAP, it has to generate and read aggregations which is much slower than when the OLAP server is doing this. In case of my second query, it performs similarly with defined aggregations and without, because one of the aggregations only minimally optimized data warehouse.

The last thing to say is that ROLAP model couldn't process the cube successfully with aggregations defined.