Olga Paszkiewicz 187583

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
   a. Size of the database (data warehouse):
      i. Total size: 720.00 MB
      ii. Number of records:
         - performance table: 10000
         - performers table: 10929
         - animals table: 11980
         - ticket type table: 3
         - seating table: 200
         - times table: 24
         - places table: 10000
         - date table: 9497
         - is in table: 989814
         - taking part table: 989921
         - ticket acquiring table: 2000000
         - performance evaluation table: 1600381
         - attending the show table: 1000000
   b. Testing environment:
      i. Hardware:
         - Device: HP Laptop 14
         - Processor: AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz
         - GPU: AMD Radeon(TM) RX Vega 10 Graphics
         - Installed RAM: 8.00 GB (5.92 GB usable)
         - System type: 64-bit operating system, x64-based processor
      ii. Software:
         - Operating system: Windows 10 Home, version: 22H2
         - Visual studio 2022
         - SQL SSIS Project 2022 (version 1.3.2)

- SQL Server Managment Studio 19

3. Testing

Testing query execution times for different models, with and without defined aggregations. Testing cube processing times in the same testing settings. Aggregations were defined on the date dimension – total duration per month. Each task was executed and measured 10 times. The cache was cleared before every execution.

Brief description of the queries:

1. Query 1: Query with aggregations on dates

```
WITH
MEMBER [Measures].[TotalDuration] AS
    SUM([Measures].[Duration])

SELECT
{
    [Measures].[TotalDuration]
} ON COLUMNS,

NON EMPTY
{
    [Date].[Year].ALLMEMBERS
  *[Date].[Month].[Month].ALLMEMBERS
} ON ROWS
```

2. Query 2: Query for a particular dimension attribute

```
SELECT
    [Measures].[Percentage of participation] ON COLUMNS,
    [Performance].[Performance Name].Members ON ROWS
FROM
    [Circus DW]
WHERE
    [Performance].[Performance Type].&[acrobatics]
```

3. Query 3: General query

```
SELECT {[Measures].[Average Satisfaction]} ON COLUMNS,
    {[Performers].[Person Number].Members} ON ROWS
FROM [Circus DW]
```

Results:

	MOLAP		ROLAP
	Aggregates	No aggregates	No aggregates
Quering speed	6 milliseconds	32 milliseconds	132 milliseconds
	103 milliseconds	110 milliseconds	776 milliseconds
	719 milliseconds	736 milliseconds	3340 milliseconds (about 3 and a half seconds)
Processing time	35362 milliseconds (about 35 and a half seconds)	31017 milliseconds (about 31 seconds)	18749 milliseconds (about 18 and a half seconds)
Total size	366.41 MB	365.99 MB	335.50 MB

4. Discussion
    a. Querying Time:

    MOLAP with aggregates has the shortest querying time, clearly visible when executing the query with aggregations on dates, this time being shorter than MOLAP without aggregates and ROLAP (without aggregates). Which aligns with theoretical expectation, as it provides fast query performance, due to it having significantly reduced data retrieval time thanks to precomputed data. The difference in times between MOLAP with and without aggregates is not visible in other queries as the dimensions used to query them were not aggregated.

    MOLAP with and without aggregations has significantly faster querying time than ROLAP as it stores the data, while ROLAP queries data directly from the relational database.

    b. Processing Time:

MOLAP with aggregates has the longest processing time because it requires copying of the data to multidimensional structures (storing the multidimensional cubes and aggregates), which takes time to construct. Processing MOLAP without aggregates takes a little less time, thanks to less calculations needing to be

performed, while ROLAP has the shortest processing time as it does not involve creating and storing multidimensional structures.

c. Total Size:

In accordance with the theoretical assumptions the ROLAP has the smallest size and MOLAP with aggregates the biggest. ROLAP relies on the existing relational data, while others need additional storage for the precomputed aggregates and the multidimensional structures.

Theoretical assumptions:

	MOLAP	ROLAP
Querying time	Short	Long
Processing time	Long	Short
Total size	Big (size of the measure group is much Moderate Small smaller if no aggregations are designed for them)	Small

5. Results of tests
   a. Processing (in milliseconds)

	MOLAP		ROLAP
Nr	No aggr.	Aggr.	No aggr.
1	30798	30219	18497
2	27498	29865	18578
3	29858	27581	18973
4	26856	28336	19183
5	35877	43085	18485
6	34831	33376	18803
7	28248	40855	18607
8	34831	43326	18752
9	28248	41949	19015
10	33120	35023	18593
MEAN	31017	35362	18749

b. Quering speed (in milliseconds):

i. Query 1

	MOLAP		ROLAP
Nr	No aggr.	Aggr.	No aggr.
1	33	7	127
2	32	6	120
3	32	6	135
4	32	7	138
5	38	6	130
6	28	6	133
7	32	6	143
8	32	5	137
9	31	6	134
10	34	6	125
MEAN	32	6	132

ii. Query 2

	MOLAP		ROLAP
Nr	No aggr.	Aggr.	No aggr.
1	101	101	794
2	116	102	797
3	110	109	771
4	112	94	752
5	105	106	764
6	108	112	780
7	115	100	773
8	107	104	785
9	113	107	769
10	109	98	776
MEAN	110	103	776

iii. Query 3

	MOLAP		ROLAP
Nr	No aggr.	Aggr.	No aggr.

1	679	688	3157
2	706	711	3881
3	711	704	3288
4	720	688	3312
5	731	728	3410
6	753	757	3200
7	755	710	3336
8	798	735	3253
9	740	721	3258
10	765	744	3304
MEAN	736	719	3340