

Data Warehouse Optimization

1. Cel laboratorium

Celem zadania jest przedstawienie zagadnień związanych z różnymi modelami fizycznymi kostki i projektowaniem agregatów

2. Założenia wstępne

Hurtownia składa się z jednej tabeli faktów zawierającej 1 167 229 rzędów.

Środowisko testowe:

- Windows Home 11
- AMD Ryzen 5600H
- RAM 16GB
- Microsoft Visual Studio 2022
- Microsoft SQL Server Management Studio 2019
- Microsoft SQL Server Profiler 2018

3. Testowanie

Przeprowadzone zostały testy porównawcze czasów wykonania zapytań oraz przetwarzania kostek dla różnych modeli fizycznych kostek z i bez agregacji.

Opis zapytań:

1. Pokaż sumę opóźnień w tym i poprzednim miesiącu.
1. (agregacja na dacie)

```
WITH  
MEMBER [Measures].[Prev month amount] AS  
(  
    ParallelPeriod([Data].[Hierarchy].[Miesiac], 1, [Data].[Hierarchy].CurrentMember),  
    [Measures].[Opoznienie]  
)
```

```
SELECT NON EMPTY  
    {[Measures].[Opoznienie], [Measures].[Prev month amount]} ON COLUMNS  
FROM [KPL]  
WHERE  
(  
    [Data].[Hierarchy].[2023].[Listopad]  
)
```

2. Pokaż sumę opóźnień na każdej linii.
2. (zapytanie z wykorzystaniem atrybutu wymiaru)

```
SELECT NON EMPTY  
    {[Measures].[Opoznienie]} ON COLUMNS,  
    NON EMPTY [Junk].[Linia].MEMBERS ON ROWS  
FROM [KPL]
```

3. Pokaż w jakich latach zatrudnionych zostało 5 kierowców, którzy w ciągu swojego dotychczasowego zatrudnienia mieli najwięcej mandatów podczas swoich przejazdów.
(zapytanie złożone)

```
SELECT
  NON EMPTY {
    [Measures].[Ilosc_wystawionych_mandatow]
  } ON COLUMNS,
  NON EMPTY {
    TOPCOUNT(
      FILTER(
        [Kierowca].[Imie i Nazwisko].[Imie i Nazwisko].MEMBERS,
        NOT IsEmpty([Kierowca].[Imie i Nazwisko])
      ),
      5,
      [Measures].[Ilosc_wystawionych_mandatow]
    ) *
    [Kierowca].[Rok Zatrudnienia].[Rok Zatrudnienia].MEMBERS
  } ON ROWS
FROM [KPL]
WHERE (
  [Data].[Hierarchy].[2023].[Listopad]
)
```

	MOLAP		ROLAP		HOLAP	
	Agr	No agr	Agr	No agr	Agr	No agr
Query speed	1 – 5ms	1 – 20ms	1 – 78ms	1 – 121ms	1 – 6ms	1 - 84ms
	2 – 4ms	2 – 5ms	2 – 74ms	2 – 75ms	2 – 5ms	2 - 82ms
	3 - 28ms	3 – 33ms	3 - 141ms	3 - 142ms	3 - 132ms	3 - 136ms
Processing time	6598ms	6420ms	2581ms	1824ms	1515ms	1385ms
Total size	31.85MB	29,92MB	4.99MB	4.99MB	6.92MB	4.99MB

4. Analiza wyników - porównanie teorii z otrzymanymi rezultatami

Według teorii MOLAP powinien mieć długi czas przetwarzania kostki, duży rozmiar przez redundancje danych (hurtownia oraz analityczna baza danych mają te same dane), ROLAP posiadać mały rozmiar, być szybko przetwarzany, ale mieć dłuższy czas wykonywania zapytań, z kolei HOLAP z dobrze dobranymi agregacjami zachowywać się jak MOLAP, bez agregacji jak ROLAP.

Otrzymane rezultaty potwierdzają następujące elementy teorii:

MOLAP:

- Duży rozmiar
- Długi czas przetwarzania kostki
- Krótkie czasy wykonywania zapytań

ROLAP:

- Mały rozmiar
- Krótki czas przetwarzania kostki
- Długie czasy wykonywania zapytań

HOLAP:

- Przy dobrze dobranych agregacjach zachowuje się jak MOLAP
- Bez agregacji (lub ze źle dobranymi agregacjami) zachowuje się jak ROLAP

Agregacje:

Agregacje zostały dobrane w sposób, aby 2 pierwsze zapytania były dobrze zoptymalizowane. W przypadku HOLAP oraz MOLAP – zapytania przy agregacjach zachowują się zgodnie z założeniami, lecz analizując kolumnę przypisaną do ROLAP można zauważyć, że agregacja nie wpływa znacząco na drugie zapytanie (ponad to, w MOLAP przy drugim zapytaniu ciężko zauważyć różnicę przez krótki czas oryginalnego przetwarzania). W przypadku MOLAP i HOLAP agregacje powiększają całkowity rozmiar. We wszystkich przypadkach agregacje wydłużają czas przetwarzania kostki oraz pomniejszają czas wykonywania zapytań, nawet jeśli agregacja nie optymalizuje danego zapytania.