

SQL - Funkcje okna (Window functions)

Lab 1

Imiona i nazwiska:

Celem ćwiczenia jest przygotowanie środowiska pracy, wstępne zapoznanie się z działaniem funkcji okna (window functions) w SQL, analiza wydajności zapytań i porównanie z rozwiązaniami przy wykorzystaniu "tradycyjnych" konstrukcji SQL

Swoje odpowiedzi wpisuj w miejsca oznaczone jako:

Wyniki:

Ważne/wymagane są komentarze.

Zamieść kod rozwiązania oraz zrzuty ekranu pokazujące wyniki, (dołącz kod rozwiązania w formie tekstowej/źródłowej)

Zwróć uwagę na formatowanie kodu

Oprogramowanie - co jest potrzebne?

Do wykonania ćwiczenia potrzebne jest następujące oprogramowanie:

- MS SQL Server - wersja 2019, 2022
- PostgreSQL - wersja 15/16
- SQLite
- Narzędzia do komunikacji z bazą danych
 - SSMS - Microsoft SQL Managment Studio
 - DDataGrip lub DBeaver
- Przykładowa baza Northwind
 - W wersji dla każdego z wymienionych serwerów

Oprogramowanie dostępne jest na przygotowanej maszynie wirtualnej

Dokumentacja/Literatura

- Kathi Kellenberger, Clayton Groom, Ed Pollack, Expert T-SQL Window Functions in SQL Server 2019, Apres 2019
- Itzik Ben-Gan, T-SQL Window Functions: For Data Analysis and Beyond, Microsoft 2020
- Kilka linków do materiałów które mogą być pomocne - <https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver16>
 - <https://www.sqlservertutorial.net/sql-server-window-functions/>
 - <https://www.sqlshack.com/use-window-functions-sql-server/>
 - <https://www.postgresql.org/docs/current/tutorial-window.html>
 - <https://www.postgresqtutorial.com/postgresql-window-function/>
 - <https://www.sqlite.org/windowfunctions.html>
 - <https://www.sqlitetutorial.net/sqlite-window-functions/>
- Ikony używane w graficznej prezentacji planu zapytania w SSMS opisane są tutaj:
 - <https://docs.microsoft.com/en-us/sql/relational-databases/showplan-logical-and-physical-operators-reference>

Przygotowanie

Uruchom SSMS - Skonfiguruj połączenie z bazą Northwind na lokalnym serwerze MS SQL

Uruchom DataGrip (lub Dbeaver)

- Skonfiguruj połączenia z bazą Northwind
 - na lokalnym serwerze MS SQL
 - na lokalnym serwerze PostgreSQL
 - z lokalną bazą SQLite

Zadanie 1 - obserwacja

Wykonaj i porównaj wyniki następujących poleceń.

```
select avg(unitprice) avgprice
from products p;

select avg(unitprice) over () as avgprice
from products p;

select categoryid, avg(unitprice) avgprice
from products p
group by categoryid

select avg(unitprice) over (partition by categoryid) as avgprice
from products p;
```

Jaka jest są podobieństwa, jakie różnice pomiędzy grupowaniem danych a działaniem funkcji okna?

Wyniki:

Zadanie 2 - obserwacja

Wykonaj i porównaj wyniki następujących poleceń.

```
--1)
select p.productid, p.ProductName, p.unitprice,
       (select avg(unitprice) from products) as avgprice
from products p
where productid < 10

--2)
select p.productid, p.ProductName, p.unitprice,
       avg(unitprice) over () as avgprice
from products p
where productid < 10
```

Jaka jest różnica? Czego dotyczy warunek w każdym z przypadków? Napisz polecenie równoważne

- 1. z wykorzystaniem funkcji okna. Napisz polecenie równoważne
- 2. z wykorzystaniem podzapytania

Wyniki:

-- ...

Zadanie 3

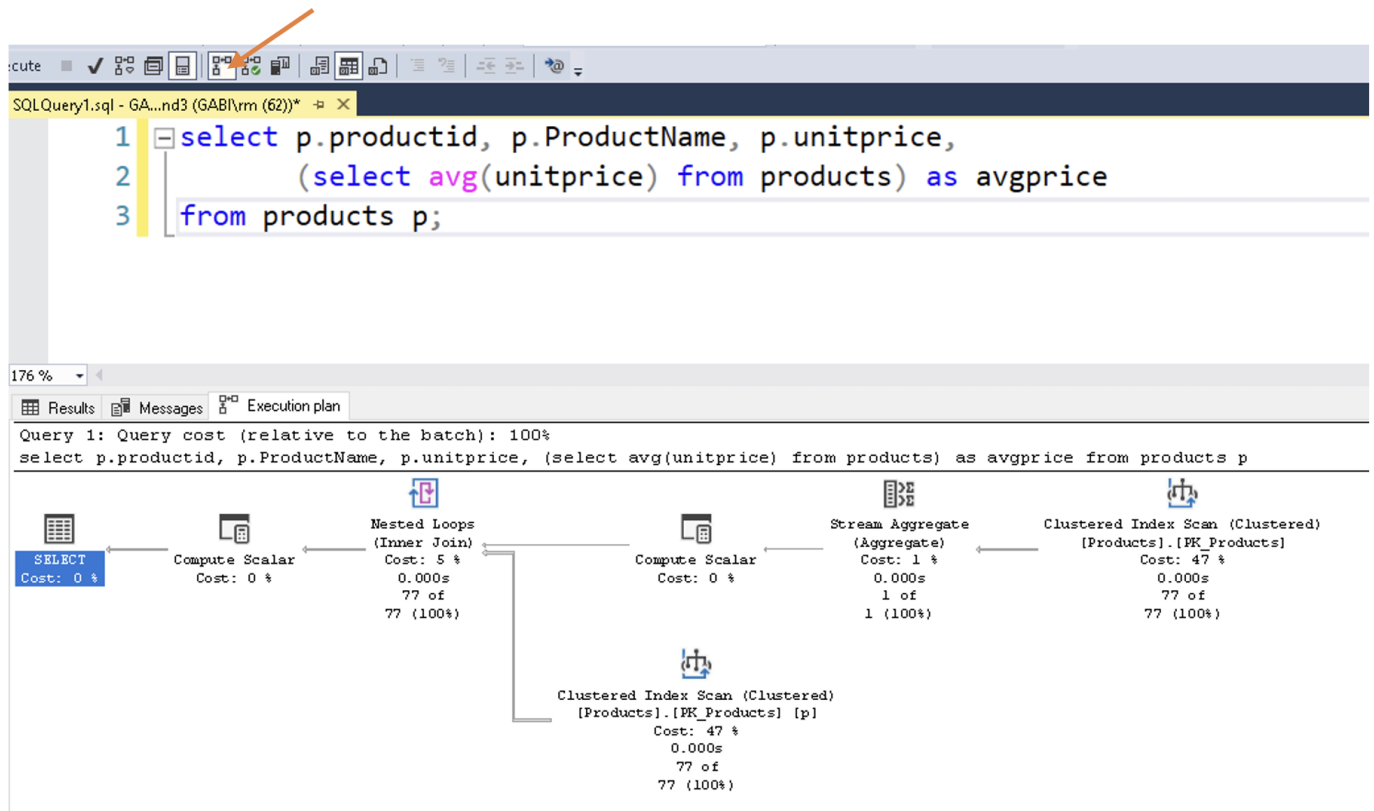
Baza: Northwind, tabela: products

Napisz polecenie, które zwraca: id produktu, nazwę produktu, cenę produktu, średnią cenę wszystkich produktów.

Napisz polecenie z wykorzystaniem z wykorzystaniem podzapytania, join'a oraz funkcji okna. Porównaj czasy oraz plany wykonania zapytań.

Przetestuj działanie w różnych SZBD (MS SQL Server, PostgreSQL, SQLite)

W SSMS włącz dwie opcje: Include Actual Execution Plan oraz Include Live Query Statistics



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query in the 'SQLQuery1.sql' file:

```
1 select p.productid, p.ProductName, p.unitprice,
2       (select avg(unitprice) from products) as avgprice
3 from products p;
```

The bottom pane shows the 'Execution plan' for 'Query 1: Query cost (relative to the batch): 100%'. The query text is repeated: `select p.productid, p.ProductName, p.unitprice, (select avg(unitprice) from products) as avgprice from products p`.

The execution plan diagram illustrates the following steps from right to left:

- Clustered Index Scan (Clustered)** [Products].[PK_Products]: Cost: 47 %, 0.000s, 77 of 77 (100%).
- Stream Aggregate (Aggregate)**: Cost: 1 %, 0.000s, 1 of 1 (100%).
- Compute Scalar**: Cost: 0 %.
- Nested Loops (Inner Join)**: Cost: 5 %, 0.000s, 77 of 77 (100%).
- Compute Scalar**: Cost: 0 %.
- SELECT**: Cost: 0 %.

An orange arrow points to the 'Include Actual Execution Plan' button in the top toolbar.

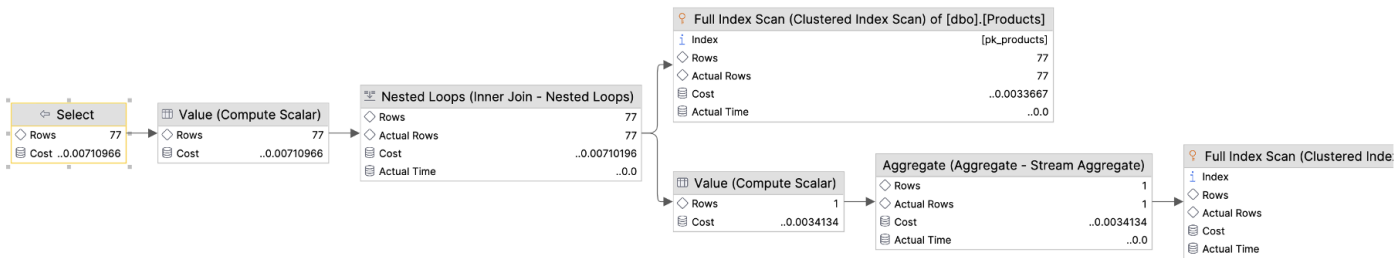
W DataGrip użyj opcji Explain Plan/Explain Analyze

The screenshot shows the SQL Server Enterprise Manager interface. In the top pane, a query is written:

```
1 select p.productid, p.ProductName, p.unitprice,
2      (select avg(unitprice) from products) as avgprice
3 from products p;
```

A context menu is open over the query, with the 'Explain Plan' option selected. The menu includes options like 'Show Context Actions', 'Paste', 'Copy / Paste Special', 'Column Selection Mode', 'Reformat', 'Folding', 'Save as Live Template...', 'Reformat Code', 'Go To', 'Generate...', 'Run !.sql!', 'More Run/Debug', 'Switch Session (w)', 'Explain Plan', 'Execute', 'Execute to File', and 'Open In'. The 'Explain Plan' submenu is also visible, showing 'Explain Plan', 'Explain Plan (Raw)', 'Explain Analyse', and 'Explain Analyse (Raw)'. The bottom pane shows a table with the following data:

	Rows	Actual Rows	Total Cost	Actual Total Ti
	77		0.00710966	
	77		0.00710966	
	77	77	0.00710196	0.0
	1		0.0034134	
	1	1	0.0034134	0.0
[PK_Products];	77	77	0.0033667	0.0
[PK_Products];	77	77	0.0033667	0.0



The screenshot shows the 'Show Diagram...' button in the context menu. The execution plan table is as follows:

Operation	Params	Rows	Actual Rows	Total Cost	Actual Total Time
Select		77		0.00710966	
Value (Compute Scalar)		77		0.00710966	
Value (Compute Scalar)	(Inner Join - N	77	77	0.00710196	0.0
Value (Compute Scalar)		1		0.0034134	
Aggregate (Aggregate - S		1	1	0.0034134	0.0
Full Index Scan (Clu: table: [dbo].[Products]; index: [PK_Products];		77	77	0.0033667	0.0
Full Index Scan (Clustered table: [dbo].[Products]; index: [PK_Products];		77	77	0.0033667	0.0

Wyniki:

Zadanie 4

Baza: Northwind, tabela products

Napisz polecenie, które zwraca: id produktu, nazwę produktu, cenę produktu, średnią cenę produktów w kategorii, do której należy dany produkt. Wyświetl tylko pozycje (produkty) których cena jest większa niż średnia cena.

Napisz polecenie z wykorzystaniem podzapytania, join'a oraz funkcji okna. Porównaj zapytania. Porównaj czasy oraz plany wykonania zapytań.

Przetestuj działanie w różnych SZBD (MS SQL Server, PostgreSQL, SQLite)

Wyniki:

zadanie	pkt
1	1
2	1
3	1
4	1
razem	4