**Hurtownie Danych - laboratorium**

**Lista 3**

***Podstawy SQL: funkcje grupujące i okienkowe***

**Wstęp teoretyczny**

**Funkcje grupujące**

Klauzula instrukcji SELECT dzieli wynik zapytania na grupy wierszy, zwykle w celu wykonania jednej lub więcej agregacji dla każdej grupy. Polecenie SELECT – GROUP BY zwraca jeden wiersz wyniku na grupę.

Stosuje się następującą składnię polecenia GROUP BY:

|  |
| --- |
| SELECT ...  FROM ...  WHERE ...  **GROUP BY** {  column-expression  | **ROLLUP** ( <group\_by\_expression> [ ,...n ] )  | **CUBE** ( <group\_by\_expression> [ ,...n ] )  | **GROUPING SETS** ( <grouping\_set> [ ,...n ] )  | **()** --calculates the grand total  } [ ,...n ] |

Różnice pomiędzy ROLLUP, CUBE i GROUPING SETS:

GROUP BY ROLLUP

|  |
| --- |
| **GROUP BY ROLLUP (col1, col2, col3, col4, ..., coln)**  Wynik – podsumowania dla następujących zestawień kolumn:   * col1, col2, ..., coln * col1, col2, ..., coln-1, NULL * col1, ..., coln-2, NULL, NULL * ... * col1, col2, NULL, ..., NULL * col1, NULL, NULL, ..., NULL * NULL, NULL, ..., NULL |

GROUP BY CUBE

|  |
| --- |
| **GROUP BY CUBE (col1, col2, col3, col4, ..., coln)**  Wynik – podsumowania dla wszystkich możliwych kombinacji kolumn  Ile jest możliwych podzbiorów zbioru n-elementowego? |

GROUP BY GROUPING SETS

|  |
| --- |
| **GROUP BY GROUPING SETS (...)**  Wynik – podsumowania dla wszystkich wymienionych kombinacji kolumn |

**Funkcje okienkowe**

Klauzula OVER określa podział i kolejność zestawu wierszy przed zastosowaniem powiązanej funkcji okna. Klauzula OVER definiuje okno (określony przez użytkownika zestaw wierszy), a następnie wylicza wartość dla każdego wiersza w oknie.

|  |
| --- |
| OVER (  [ <PARTITION BY clause> ]  [ <ORDER BY clause> ]  [ <ROW or RANGE clause> ]  ) |

Klauzulę OVER stosuje się do obliczania wartości zagregowanych, np. średnie ruchome, sumy bieżące, wyniki dla ostatnich *n* wierszy w każdej grupie.

|  |
| --- |
| **PARTITION BY clause**  Dzieli zbiór wyników zapytania na określone podgrupy. Funkcja okna jest stosowana do każdej podgrupy osobno i obliczenia uruchamiane są ponownie dla każdej podgrupy. |

|  |
| --- |
| **ORDER BY clause**  Definiuje logiczną kolejność wierszy w każdej podgrupie wynikowego zbioru. Dla każdej podgrupy określa logiczną kolejność wykonywania obliczeń funkcji okna. |

|  |
| --- |
| **ROW or RANGE clause**  Ograniczają wiersze w partycji, określając granice:   * ROWS – fizyczne ograniczenie liczby wierszy, np.   + BETWEEN 2 PRECEEDING AND CURRENT ROW * RANGE – logiczne ograniczenie liczby wierszy, np.   + BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING   Obie klauzule wymagają klauzuli ORDER BY i na podstawie ustalonej kolejności determinują wynik. |

Funkcje szeregujące pozwalają tworzyć ranking, przypisując odpowiednie miejsca kolejnym wynikom. Najczęściej stosowane są następujące funkcje szeregujące:

|  |
| --- |
| * **ROW\_NUMBER** – nr pozycji * **RANK** – ranking (to samo miejsce dla tych samych wartości) * **DENSE\_RANK** – ranking, numerowanie ciągłe * **NTILE** – grupuje rekordy poprzez przypisanie tej samej wartości szeregującej członkom grupy |

Źródła:

<https://docs.microsoft.com/en-us/sql/t-sql/queries/select-group-by-transact-sql?view=sql-server-ver15>

<https://docs.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver15>

**Zadania do wykonania**

Baza danych: **AdventureWorks**

**Zad. 1. Wykorzystanie funkcji grupujących (rollup, cube, grouping sets)**

1. Przygotować zestawienie przedstawiające, ile pieniędzy wydali klienci na zamówienia na przestrzeni poszczególnych lat.

Wykonaj zestawienie przy użyciu poleceń rollup, cube, grouping sets.



*Przykładowe rozwiązanie*

1. Przygotować zestawienie przedstawiające łączną kwotę zniżek z podziałem na kategorie, produkty oraz lata.



*Przykładowe rozwiązanie*

**Zad. 2. Wykorzystanie funkcji okienkowych (over, over partition by, row\_number, rank, dense\_rank, ntile)**

1. Dla kategorii ‘Bikes’ przygotuj zestawienie prezentujące procentowy udział kwot sprzedaży produktów tej kategorii w poszczególnych latach w stosunku do łącznej kwoty sprzedaży dla tej kategorii. W zadaniu wykorzystaj funkcje okna.



*Przykładowe rozwiązanie*

Wykonaj podobne zestawienia dla pozostałych kategorii.

1. Przygotuj zestawienie prezentujące narastającą liczbę zamówień wykonanych przez klienta w poszczególnych latach. W zadaniu wykorzystaj funkcje okna.

Ogranicz wynik do 10 najlepszych klientów.

1. Przygotuj zestawienie dla sprzedawców z podziałem na lata i miesiące prezentujące liczbę obsłużonych przez nich zamówień w ciągu roku, w ciągu roku narastająco oraz sumarycznie w obecnym i poprzednim miesiącu. W zadaniu wykorzystaj funkcje okna.



*Przykładowe rozwiązanie*

1. Przygotuj zestawienie dla kategorii produktów prezentujące sumę maksymalnych cen produktów w poszczególnych podkategoriach należących danej tej kategorii. W zadaniu wykorzystaj funkcje okna.
2. Przygotuj ranking klientów w zależności od liczby zakupionych produktów. Porównaj rozwiązania uzyskane przez funkcje rank i dense\_rank.
3. Przygotuj ranking produktów w zależności od średniej liczby sprzedanych sztuk. Wyróżnij 3 (prawie równoliczne) grupy produktów: sprzedających się najlepiej, średnio i najsłabiej.

**Rozwiązania:**

**Wnioski:**

***Uwaga!***

* Sprawozdanie, bez wniosków podsumowujących aspekt zagadnień analizowanych na zajęciach laboratoryjnych i zawartych w sprawozdaniu, jest automatycznie oceniane negatywnie!