



SQL

Operacije agregacj

- Wyświetlanie  $n$  początkowych wartości (TOP  $n$ )
- Użycie funkcji agregujących
- Grupowanie danych - klauzula GROUP BY
- Generowanie wartości zagregowanych
- Użycie klauzul COMPUTE i COMPUTE BY

# Wyświetlanie TOP $n$ wartości



- Wyświetlanie tylko pierwszych  $n$  wierszy zbioru wynikowego
- Określenie zakresu wartości w klauzuli ORDER BY
- fraza WITH TIES - zwracane są dodatkowo wszystkie elementy o wartościach takich jak element ostatni

```
USE northwind
SELECT TOP 5 orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
```

```
USE northwind
SELECT TOP 5 WITH TIES orderid, productid, quantity
FROM [order details]
ORDER BY quantity DESC
```

# Użycie funkcji agregujących



<i><b>Funkcja agregująca</b></i>	<i><b>Opis</b></i>
AVG	średnia wartości w wyrażeniu numerycznym
COUNT	liczba wartości w wyrażeniu
COUNT (*)	liczba wybranych wierszy
MAX	największa wartość w wyrażeniu
MIN	najmniejsza wartość w wyrażeniu
SUM	ogólna wartość w wyrażeniu numerycznym
STDEV	statystyczne odchylenie wszystkich wartości
STDEVP	statystyczne odchylenie dla populacji
VAR	statystyczna wariancja dla wszystkich wartości
VARP	statystyczna wariancja dla wszystkich wartości w populacji

# Użycie funkcji agregujących z wartościami Null



- Większość funkcji agregujących ignoruje wartości Null
- Funkcja COUNT(\*) zlicza wiersze z wartościami Null

```
USE northwind
```

```
SELECT COUNT (*)
```

```
FROM employees
```

```
USE northwind
```

```
SELECT COUNT(reportsto)
```

```
FROM employees
```

- Policz średnią cenę jednostkową dla wszystkich produktów:

```
USE northwind  
  
SELECT AVG(unitprice)  
  
FROM products
```

- Pokaż sumaryczną ilość sprzedanych produktów:

```
USE northwind  
  
SELECT SUM(quantity)  
  
FROM [order details]
```

1. Podaj liczbę produktów o cenach mniejszych niż 10\$ lub większych niż 20\$
2. Podaj maksymalną cenę produktu dla produktów o cenach poniżej 20\$
3. Podaj maksymalną, minimalną i średnią cenę produktu dla produktów sprzedawanych w butelkach ('bottle')
4. Wypisz informację o wszystkich produktach o cenie powyżej średniej
5. Podaj wartość zamówienia o numerze 10250



# Grupowanie danych – klauzula GROUP BY



- Użycie klauzuli GROUP BY
- Użycie klauzuli GROUP BY z klauzulą HAVING



# Użycie klauzuli GROUP BY



```
USE northwind
```

```
SELECT productid, orderid  
       ,quantity  
FROM orderhist
```

```
SELECT productid
```

```
       ,SUM(quantity) AS total_quantity  
FROM orderhist  
GROUP BY productid
```

```
SELECT productid
```

```
       ,SUM(quantity) AS total_quantity  
FROM orderhist  
WHERE productid = 2  
GROUP BY productid
```

- Napisz polecenie, które zwraca informacje o zamówieniach
  - zapytanie ma grupować i wyświetlać identyfikator każdego produktu, a następnie obliczać ogólną zamówioną ilość
  - ogólna ilość jest sumowana funkcją agregującą SUM i wyświetlana jako jedna wartość dla każdego produktu.

```
USE northwind
```

```
SELECT productid, SUM(quantity) AS total_quantity
```

```
FROM [order details]
```

```
GROUP BY productid
```

1. Podaj maksymalną cenę zamawianego produktu dla każdego zamówienia. Posortuj zamówienia wg maksymalnej ceny produktu
2. Podaj maksymalną i minimalną cenę zamawianego produktu dla każdego zamówienia
3. Podaj liczbę zamówień dostarczanych przez poszczególnych spedytorów
4. Który ze spedytorów był najaktywniejszy w 1997 roku?

# Użycie klauzuli GROUP BY z klauzulą HAVING



```
USE northwind  
  
SELECT productid, orderid  
       ,quantity  
  
FROM orderhist
```

```
USE northwind  
  
SELECT productid, SUM(quantity)  
       AS total_quantity  
  
FROM orderhist  
  
GROUP BY productid  
  
HAVING SUM(quantity)>=30
```

- Wyświetl listę identyfikatorów produktów i ilość dla tych produktów, których zamówiono ponad 1200 jednostek

```
USE northwind
```

```
SELECT productid, SUM(quantity) AS total_quantity
```

```
FROM [order details]
```

```
GROUP BY productid
```

```
HAVING SUM(quantity)>1200
```

1. Wyświetl zamówienia dla których liczba pozycji zamówienia jest większa niż 5
2. Wyświetl klientów, dla których w 1998 roku zrealizowano więcej niż 8 zamówień (wyniki posortuj malejąco wg łącznej kwoty za dostarczenie zamówień dla każdego z klientów)



# Generowanie wartości zagregowanych w zbiorach wynikowych



- Użycie klazuli GROUP BY z operatorem ROLLUP
- Użycie klauzuli GROUP BY z operatorem CUBE
- Użycie funkcji GROUPING



# Użycie klauzuli GROUP BY z operatorem ROLLUP



```
USE northwind
```

```
SELECT productid, orderid, SUM(quantity) AS total_quantity
```

```
FROM orderhist
```

```
GROUP BY productid, orderid
```

```
WITH ROLLUP
```

```
ORDER BY productid, orderid
```

- Przykład zwraca informacje o zamówieniach z tablicy *order details*.
  - Instrukcja *select* z klauzulą GROUP BY bez operatora ROLLUP.
  - Zwraca listę ogólnej ilości zamawianej dla każdego produktu w każdym zamówieniu, dla zamówień z *orderid* mniejszym niż 10250

```
USE northwind
```

```
SELECT orderid, productid, SUM(quantity) AS total_quantity
```

```
FROM [order details]
```

```
WHERE orderid < 10250
```

```
GROUP BY orderid, productid
```

```
ORDER BY orderid, productid
```

- Przykład dodaje operator ROLLUP do poprzedniego wyrażenia. Zbiór wynikowy zawiera ogólną ilość dla:
  - każdego produktu w każdym zamówieniu
  - wszystkich produktów dla każdego zamówienia
  - wszystkich produktów dla wszystkich zamówień

```
USE northwind
```

```
SELECT orderid, productid, SUM(quantity) AS total_quantity
```

```
FROM [order details]
```

```
WHERE orderid < 10250
```

```
GROUP BY orderid, productid
```

```
WITH ROLLUP
```

```
ORDER BY orderid, productid
```

# Użycie klauzuli GROUP BY z operatorem CUBE



```
USE northwind
```

```
SELECT productid, orderid, SUM(quantity) AS total_quantity
```

```
FROM orderhist
```

```
GROUP BY productid, orderid
```

```
WITH CUBE
```

```
ORDER BY productid, orderid
```

# Użycie funkcji GROUPING



```
SELECT productid, GROUPING (productid)
      ,orderid, GROUPING (orderid)
      ,SUM(quantity) AS total_quantity
FROM orderhist
GROUP BY productid, orderid
WITH CUBE
ORDER BY productid, orderid
```

# Przykład - GROUPING



- Zbiór wynikowy ma dodatkową kolumnę po *productid* i *orderid*
  - funkcja GROUPING zwraca 1, gdy wartości w określonej kolumnie zostały pogrupowane razem operatorem CUBE.
  - ogólna ilość dla każdego produktu dla każdego zamówienia, każdego produktu dla wszystkich zamówień, wszystkich produktów dla każdego zamówienia oraz totalną ilość dla wszystkich produktów i dla wszystkich zamówień.

```
SELECT orderid, GROUPING (orderid)  
      ,productid, GROUPING (productid)  
      ,SUM(quantity) AS total_quantity  
FROM [order details]  
WHERE orderid < 10250  
GROUP BY orderid, productid  
WITH CUBE  
ORDER BY orderid, productid
```



# Użycie klauzul COMPUTE i COMPUTE BY



## COMPUTE

```
USE northwind
SELECT productid, orderid
       ,quantity
FROM orderhist
ORDER BY productid, orderid
COMPUTE SUM(quantity)
```

## COMPUTE BY

```
USE northwind
SELECT productid, orderid, quantity
FROM orderhist
ORDER BY productid, orderid
COMPUTE SUM(quantity) BY productid
COMPUTE SUM(quantity)
```