SQL

- **Structured Query Language**

Grouping Data

- Listing the TOP *n* Values
- Using aggregate functions
- GROUP BY
- Generating aggregate values within result sets

TOP *n* Values

- **Lists only the first *n* rows of a result set**
- **Returns ties if WITH TIES is used**

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

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

Aggregate functions

Funkcja agregująca	Opis
AVG	Average of values in a numeric expression
COUNT	Number of values in an expression
COUNT (*)	Number of selected rows
MAX	Highest value in the expression
MIN	Lowest value in the expression
SUM	Total values in a numeric expression

Aggregate functions

- **Most aggregate functions ignore Null values**
- **COUNT(*) function counts all rows (including these with Null values)**

```
SELECT COUNT (*)
 FROM employees
```

```
SELECT COUNT(reportsto)
 FROM employees
```

Aggregate functions - examples

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

```
SELECT SUM(quantity)
 FROM [order details]
WHERE productid = 1
```

GROUP BY

- Using the GROUP BY Clause
- Using the GROUP BY Clause with the HAVING Clause

GROUP BY

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

productid	orderid	quantity
1	1	5
1	2	10
2	1	10
2	2	25
3	1	15
3	2	30

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

productid	total_quantity
1	15
2	35
3	45

Tylko wiersze spełniające klauzulę WHERE są grupowane

productid	total_quantity
2	35

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

GROUP BY - example

- This example returns information about orders from the orderhist table. The query groups and lists each product ID and calculates the total quantity ordered. The total quantity is calculated with the SUM aggregate function and displays one value for each product in the result set.

```
SELECT productid, SUM(quantity) AS total_quantity
 FROM [order details]
 GROUP BY productid
```

GROUP BY with the HAVING clause

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

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

productid	orderid	quantity
1	1	5
1	2	10
2	1	10
2	2	25
3	1	15
3	2	30

productid	total_quantity
2	35
3	45

GROUP BY with the HAVING clause - example

- **This example lists the product ID and quantity for products that have orders for more than 1,200 units.**

```
SELECT productid, SUM(quantity) AS total_quantity
 FROM [order details]
 GROUP BY productid
 HAVING SUM(quantity)>1200
```

ROLLUP and CUBE

- Using the GROUP BY Clause with the ROLLUP Operator
- Using the GROUP BY Clause with the CUBE Operator

GROUP BY with the ROLLUP operator

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

productid	orderid	total_quantity	
NULL	NULL	95	Grand total
1	NULL	15	Summarizes only rows for **productid 1**
1	1	5	Detail value for **productid 1**, **orderid 1**
1	2	10	Detail value for **productid 1**, **orderid 2**
2	NULL	35	Summarizes only rows for **productid 2**
2	1	10	Detail value for **productid 2**, **orderid 1**
2	2	25	Detail value for **productid 2**, **orderid 2**
3	NULL	45	Summarizes only rows for **productid 3**
3	1	15	Detail value for **productid 3**, **orderid 1**
3	2	30	Detail value for **productid 3**, **orderid 2**

Example

- This query contains a SELECT statement with a GROUP BY clause without the ROLLUP operator. The example returns a list of the total quantity that is ordered for each product on each order, for orders with an orderid less than 10250.

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

Example

- **This example adds the ROLLUP operator to the statement The result set includes the total quantity for:**
  - Each product for each order (also returned by the GROUP BY clause without the ROLLUP operator).
  - All products for each order.
    All products for all orders (grand total).

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

GROUP BY with the ROLLUP operator

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

productid	orderid	total_quantity
NULL	NULL	95
NULL	1	30
NULL	2	65
1	NULL	15
1	1	5
1	2	10
2	NULL	35
2	1	10
2	2	25
3	NULL	45
3	1	15
3	2	30