



Заявки с групиране и обобщаване на данни в SQL

Пример на запитвания за извличане на обобщени данни

- За всяка поръчка да се изведе номерът на поръчката, броят и общата стойност на поръчаните с нея стоки.
- Да се изведат имената на първите 5 града, от които има най-голям брой клиенти.
- За всеки клиент да се изведат данни за клиента, общия брой на направените от него поръчки и общата им стойност.
-

Пример: имената на първите 5 града, от които има най-голям брой клиенти.

```
SELECT TOP 5 City, COUNT(*) AS COUNT_Customers  
FROM Customers  
GROUP BY City  
ORDER BY 2 DESC
```

Резултат: City COUNT_Customers

London	6
México D.F.	5
Sao Paulo	4
Buenos Aires	3
Madrid	3

(5 row(s) affected)

Пример: Да се извлекат кодовете и имената на поръчаните стоки и за всяка поръчана стока сумарното поръчано количество.

```
USE Northwind
```

```
GO
```

```
SELECT P.ProductID, ProductName
```

```
, SUM(Quantity) AS Total_quantity
```

```
FROM Products P JOIN [Order Details] OD
```

```
ON P.ProductID = OD.ProductID
```

```
GROUP BY P.ProductID, ProductName
```

```
ORDER BY ProductID
```

SQLQuery1.sql - WIN-6C20A8VN2U0.Northwind (WIN-6C20A8VN2U0\MK (55))* - Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Help

Northwind Execute Debug

Object Explorer

- Connect
- WIN-6C20A8VN2U0 (SQL Server 12.0.2269 ->)
 - Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks2012
 - AdventureWorksDW2012
 - AdventureWorksDW2014
 - AdventureWorksLT2012
 - ContosoRetailDW
 - ContosoSales
 - ContosoSalesSQL
 - DQS_STAGING_DATA
 - itib
 - MasterData_AW
 - Northwind
 - Database Diagrams
 - Tables
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - primer
 - PRIMER2016
 - ReportServer
 - ReportServerTempDB
 - Security
 - Server Objects
 - Replication

SQLQuery1.sql - W...0A8VN2U0\MK (55))*

```
USE Northwind
GO
SELECT P.ProductID, ProductName
, SUM(Quantity) AS Total_quantity
FROM Products P JOIN [Order Details] OD
ON P.ProductID = OD.ProductID
GROUP BY P.ProductID, ProductName
ORDER BY ProductID
```

100 %

Results Messages

	ProductID	ProductName	Total_quantity
1	1	Chai	828
2	2	Chang	1057
3	3	Aniseed Syrup	328
4	4	Chef Anton's Cajun Seasoning	453
5	5	Chef Anton's Gumbo Mix	298
6	6	Grandma's Boysenberry Spread	301
7	7	Uncle Bob's Organic Dried Pears	763
8	8	Northwoods Cranberry Sauce	372
9	9	Mishi Kobe Niku	95
10	10	Ikura	742
11	11	Queso Cabrales	704

Query executed successfully. WIN-6C20A8VN2U0 (12.0 RTM) | WIN-6C20A8VN2U0\MK (55) | Northwind | 00:00:00 | 77 rows

Ready Ln 9 Col 1 Ch 1 INS

Частичен синтаксис на SELECT с клауза GROUP BY

```
SELECT <select_list>  
FROM {<table source>}[,.....n]  
[ WHERE <search_condition> ]  
[ GROUP BY <group by spec> ]  
[ HAVING <search_condition> ]  
[ ORDER BY order_expression  
          [ ASC | DESC ][,....] ]
```

Клауза **GROUP BY**

- **GROUP BY** <group by spec>
- Клаузата **GROUP BY** в оператора **SELECT** се използва за организиране на редовете в групи
- <group by spec> - списък с имената на колоните, по които се прави групирането.
- Не се препоръчва да се прави групиране по колони, които съдържат неопределени стойности (Null) - те се обработват като отделна група!!!

Особености списъка за избор при използване на клаузата **GROUP BY**

- В списъка за избор се задават
 - колоните, по които се прави групиране (колони, специфицирани в клаузата **GROUP BY**)
 - и нови колони, получавани чрез агрегатни функции.

Стъпки при изпълнение на **SELECT** с клауза **GROUP BY**

1.

Извлечените от таблиците редове се отделят в групи, така че всяка група съдържа редовете, за които колоните, зададе в клаузата **GROUP BY** имат равни стойности

2.

За всяка група се изчисляват обобщени стойности по зададените агрегатни функции

3.

В резултата се разполага по един ред за всяка група - стойностите на полетата за групиране и изчислените обобщени стойности за групата

Пример: За всяка поръчка да се изведе
номер на поръчката, брой и обща
стойност на поръчаните с нея стоки

USE Northwind

GO

SELECT OrderID

, COUNT(ProductID) AS Ord_Count

, SUM(Quantity*UnitPrice) AS Total

FROM [Order Details]

GROUP BY OrderID

Ограничаване на редовете, включвани в групите

- Чрез клауза **WHERE** се задава условие, на което да отговарят редовете , които ще се включат в групата

USE Northwind

GO

```
SELECT OrderID, COUNT(ProductID) AS  
Ord_Count
```

```
, SUM(Quantity*UnitPrice) AS Total
```

```
FROM [Order Details]
```

```
WHERE OrderId BETWEEN 10200 AND 10500
```

```
GROUP BY OrderID
```

Стъпки при изпълнение на **SELECT** с клаузи **WHERE** и **GROUP BY**

1.

От таблиците се селектират редовете, които отговарят на зададеното чрез **WHERE** условие

2.

Върху селектираните редове се прави групиране

3.

Изчисляват се зададените обобщени стойности

4.

В резултата се разполага по един ред за всяка група - стойностите на полетата за групиране и изчислените обобщени стойности за групата

Задаване на условие за групите, които да се изведат в резултата

Клауза

HAVING <search_condition>

- В резултата се извеждат само групите, отговарящи на зададените в **HAVING** условия
- В **HAVING** могат да се задават до 128 условия, свързани с логическите оператори **AND**, **OR** или **NOT**

Пример на използване на клауза **HAVING**

```
SELECT ProductId  
, SUM(Quantity) AS Total_quantity  
FROM [Order Details]  
GROUP BY ProductId  
HAVING SUM(Quantity) > 1000  
ORDER BY ProductId
```

Резултат: Извеждат се редове само за тези продукти (групи), за които общото поръчано количество е по-голямо от 1000

Особености при задаване на **HAVING**

- Условия, на които трябва да отговарят групите, показвани в резултата могат да се задават върху
 - полетата за групиране
 - или върху изчисляваните обобщени стойности
- Клаузата **HAVING** може да се използва само с клауза **GROUP BY**
- ! Разполага се винаги след клаузата **GROUP BY**.

Пример на използване на клауза **HAVING**

USE Northwind

GO

SELECT ProductId, SUM(Quantity) AS Total_quantity

FROM [Order Details]

GROUP BY ProductId

HAVING (ProductID BETWEEN 1 AND 30)

AND (SUM(Quantity)>500)

ORDER BY ProductID

Резултат: Извеждат се редове само за групите, за които кодът на продукта е между 1 и 30 и сумарно поръчаното количество е по-голямо от 500.

Стъпки при изпълнение на **SELECT** с клауза **HAVING**

1.

Прави се групиране по зададените колони

2.

Изчисляват се обобщените стойности

3.

Върху създадените групи се прилагат зададените в **HAVING** условия

Пример на използване на клаузи **WHERE** и **HAVING** едновременно

USE Northwind

GO

SELECT ProductID

, SUM(Quantity) AS Total_quantity

FROM [Order Details]

WHERE OrderID BETWEEN 10200 AND 10500

GROUP BY ProductID

HAVING SUM(Quantity) > 1000

ORDER BY ProductId

Групиране и обобщаване на данни от свързани таблици

- При свързване на таблици е възможно да се зададе групиране по колони от едната таблица и обобщаване на данни върху колони от другата (свързаната с нея) таблица.

Пример:

- Справка, съдържаща код и име на продукта, брой поръчки, сумарно поръчано количество, средна цена и обща стойност на всички поръчки за всеки поръчан продукт

```
USE Northwind
GO
SELECT P.ProductID, P.ProductName
, COUNT(OD.ProductID) AS Count_Orders
, SUM(Quantity) SUM_Quantity
, CONVERT(decimal(10,2), AVG(OD.UnitPrice)) AS
        Average_Price
, SUM(OD.UnitPrice*Quantity) AS Total
FROM Products P INNER JOIN [Order Details] OD
        ON P.ProductID = OD.ProductID
GROUP BY P.ProductID, P.ProductName
ORDER BY Total
```

Пример:

- Да се изведат данни (код и име на клиента, брой поръчки и обща стойност на поръчките) за 5-те клиента, които са донесли най-големи приходи за фирмата.



USE Northwind

GO

SELECT TOP 5 C.CustomerID, C.CompanyName
, COUNT(O.OrderID) AS Count_Orders

, SUM(OD.UnitPrice*Quantity) AS Total

FROM Customers C INNER JOIN Orders O

ON C.CustomerID = O.CustomerID

INNER JOIN [Order Details]OD

ON O.OrderID = OD.OrderID

GROUP BY C.CustomerID, C.CompanyName

ORDER BY 4 DESC

Пример:

- Да се изведат данни (код и име на клиента, брой поръчки и обща стойност на поръчките, дял в общите приходи) за 5-те клиента, които са донесли най-големи приходи за фирмата и какъв е тяхния дял в общите приходи.

USE Northwind

GO

SELECT TOP 5 C.CustomerID, C.CompanyName

, COUNT(O.OrderID) AS Count_Orders

, SUM(OD.UnitPrice*Quantity) AS Total

, **SUM(OD.UnitPrice*Quantity)/**

(SELECT SUM(UnitPrice*Quantity)

FROM [Order Details])*100 AS Total_Percent

FROM Customers C INNER JOIN Orders O

ON C.CustomerID = O.CustomerID

INNER JOIN [Order Details]OD

ON O.OrderID = OD.OrderID

GROUP BY C.CustomerID, C.CompanyName

ORDER BY 4 DESC

Справка за общата стойност на поръчаните стоки по години на поръчване

```
USE Northwind
```

```
GO
```

```
SELECT ProductName, YEAR(OrderDate) AS  
OrderDate
```

```
, SUM(Quantity*OD.UnitPrice) AS Total
```

```
FROM Products P JOIN [Order Details] OD
```

```
ON P.ProductID = OD.ProductId
```

```
JOIN Orders O
```

```
ON OD.OrderID = O.OrderID
```

```
GROUP BY ProductName, YEAR(OrderDate)
```

```
ORDER BY 1,2
```

Използване на клауза GROUP BY с клауза ROLLUP - пример

```
SELECT ProductName
      , YEAR(OrderDate) AS OrderDate
      , SUM(Quantity*OD.UnitPrice) AS Total
FROM Products P JOIN [Order Details] OD
      ON P.ProductID = OD.ProductId
      JOIN Orders O
      ON OD.OrderID = O.OrderID
GROUP BY
ROLLUP(ProductName, YEAR(OrderDate))
```



Results



Messages

	ProductName	OrderD...	Total
1	Alice Mutton	1996	7300,80
2	Alice Mutton	1997	19718,40
3	Alice Mutton	1998	7839,00
4	Alice Mutton	2012	624,00
5	Alice Mutton	NULL	35482,20
6	Aniseed Syrup	1996	240,00
7	Aniseed Syrup	1997	1760,00
8	Aniseed Syrup	1998	1080,00
9	Aniseed Syrup	NULL	3080,00

Пояснения - Резултатът съдържа:

- За всяка поръчана стока:
 - редове с данни за името на стоката, година на поръчките и обща стойност на поръчаното количество;
 - Обобщен ред за всички години; в него колоната за година ще съдържа NULL
- Обобщен ред за общата стойност на поръчаното количество за всички стоки и всички години; в него колоните име на стоката и година съдържат NULL

Пример на GROUP BY с клауза ROLLUP и ORDER BY

```
SELECT ProductName
      , YEAR(OrderDate) AS OrderDate
      , SUM(Quantity*OD.UnitPrice) AS Total
FROM Products P JOIN [Order Details] OD
  ON P.ProductID = OD.ProductId
JOIN Orders O
  ON OD.OrderID = O.OrderID
GROUP BY ROLLUP
      (ProductName, YEAR(OrderDate))
ORDER BY ProductName, 2
```



Results



Messages

	ProductName	OrderDate	Total
1	NULL	NULL	1355358,59
2	Alice Mutton	NULL	35482,20
3	Alice Mutton	1996	7300,80
4	Alice Mutton	1997	19718,40
5	Alice Mutton	1998	7839,00
6	Alice Mutton	2012	624,00
7	Aniseed Syrup	NULL	3080,00
8	Aniseed Syrup	1996	240,00
9	Aniseed Syrup	1997	1760,00

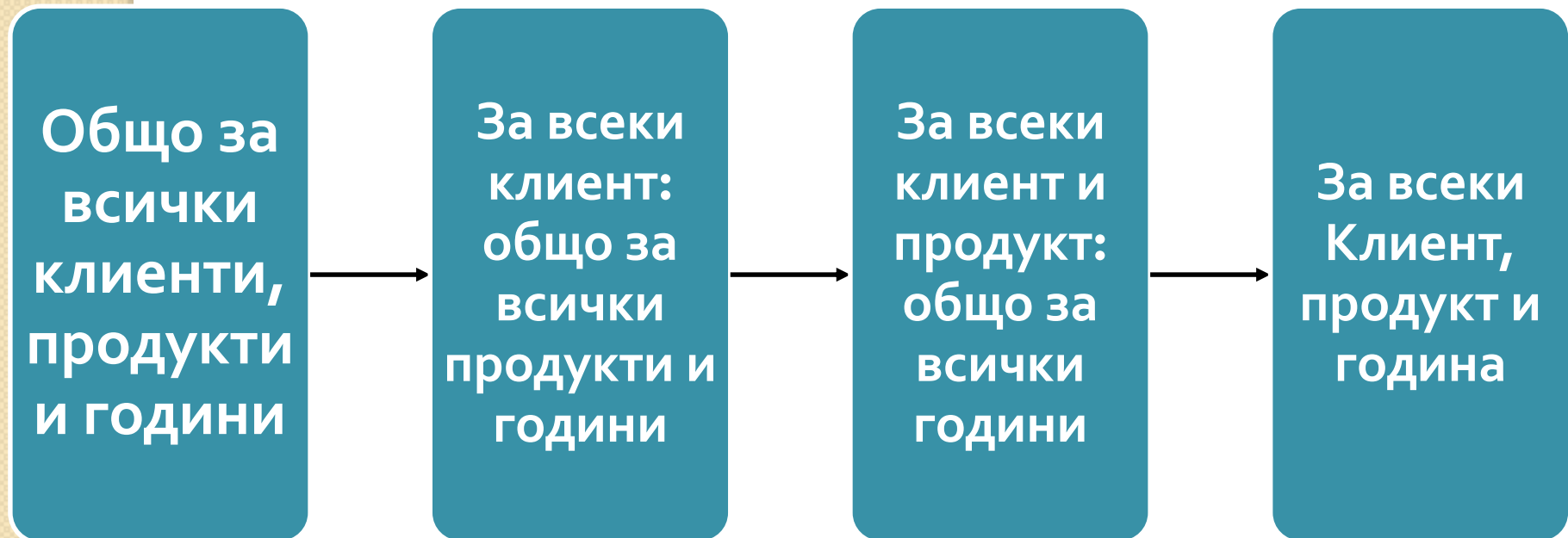
Използване на клауза **GROUP BY** с клауза **ROLLUP**


- Клаузата **ROLLUP** генерира резултат, който съдържа:
 - редове с обобщени данни за всяка група;
 - обобщени редове (subtotals) за всяка уникална комбинация от стойности на колоните за групиране т.е. за всяка подгрупа;
 - обобщен ред (grand totals) за целия резултатен набор.
- Обобщени редове се извеждат при смяна на групата.
- Резултатният набор е в релационен формат

Използване на клауза **GROUP BY** с клауза **ROLLUP**

- **ROLLUP** генерира резултатния набор, като показва обобщенията в йерархията на колоните, зададени за групиране.
- В обобщените редове **subtotals** и **grand total** колоните, по които се прави групиране съдържат стойността на групата или **NULL**.
- Ако се зададе подреждане на резултата по колоните за групите, обобщените редове предхождат детайлните редове.

Пример за използване на ROLLUP с повече от две колони за групиране:
извличане на обща стойност на поръчаните стоки по клиенти, стоки и година на поръчване





```
USE Northwind
GO
SELECT CompanyName, ProductName
, year(orderdate)
, SUM(Quantity*OD.UnitPrice) AS Total
FROM Customers C JOIN Orders O
ON C.CustomerID = O.CustomerID
    JOIN [Order Details] OD
    ON O.OrderID = OD.OrderID
        JOIN Products P
        ON P.ProductID = OD.ProductId
GROUP BY ROLLUP
(CompanyName, ProductName, year(orderdate))
```



Results



Messages

	CompanyName	ProductName	(No column na...	Total
33...	Trail's Head Go...	Rhonbrau Klosterbier	1997	155,00
33...	Trail's Head Go...	Rhonbrau Klosterbier	NULL	155,00
33...	Trail's Head Go...	Tarte au sucre	1997	493,00
33...	Trail's Head Go...	Tarte au sucre	1998	147,90
33...	Trail's Head Go...	Tarte au sucre	NULL	640,90
33...	Trail's Head Go...	Wimmers gute Semmelknodel	1997	199,50
33...	Trail's Head Go...	Wimmers gute Semmelknodel	NULL	199,50
33...	Trail's Head Go...	NULL	NULL	1571...
33...	Vaffeljernet	Aniseed Syrup	1997	140,00
33...	Vaffeljernet	Aniseed Syrup	NULL	140,00
33...	Vaffeljernet	Boston Crab Meat	1997	1214...
33...	Vaffeljernet	Boston Crab Meat	NULL	1214...
33...	Vaffeljernet	Filo Mix	1997	105,00
33...	Vaffeljernet	Filo Mix	NULL	105,00

Използване на клауза **GROUP BY** с клауза **CUBE**

- **CUBE** генерира резултатен набор, показвайки обобщения за всички възможни комбинации на стойностите в зададените колони за групиране - $2^n - 1$
- **NULL** в резултатния набор индикира, че редовете са обобщения и са създадени от **CUBE**

Пример на използване на клауза **GROUP BY** с **CUBE**

```
SELECT ProductName
, YEAR(OrderDate) AS OrderDate
, SUM(Quantity*OD.UnitPrice) AS Total
FROM Products P JOIN [Order Details] OD
    ON P.ProductID = OD.ProductId
JOIN Orders O
    ON OD.OrderID = O.OrderID
GROUP BY
CUBE(ProductName, YEAR(OrderDate))
ORDER BY ProductName, 2
```




Results



Messages

	ProductName	OrderDate	Total
1	NULL	NULL	1355358,59
2	NULL	1996	222996,00
3	NULL	1997	645926,20
4	NULL	1998	462585,44
5	NULL	2012	23850,95
6	Alice Mutton	NULL	35482,20
7	Alice Mutton	1996	7300,80
8	Alice Mutton	1997	19718,40
9	Alice Mutton	1998	7839,00
10	Alice Mutton	2012	624,00
11	Aniseed Sy...	NULL	3080,00
12	Aniseed Sy...	1996	240,00
13	Aniseed Sy...	1997	1760,00

- 
- Ако се зададе **group by cube(a,b,c)**, то ще се генерира по един ред за всяка уникална комбинация:

(a,b,c) (a,b) (a,c) (b,c) (a) (b) (c)

- и един ред grand total

Пример: извличане на обща стойност на поръчаните стоки по клиенти, стоки и години на поръчване

USE Northwind

GO

```
SELECT CompanyName, ProductName  
, year(orderdate) AS Orders_Year  
, SUM(Quantity*OD.UnitPrice) AS Total  
FROM Customers C JOIN Orders O  
ON C.CustomerID = O.CustomerID  
JOIN [Order Details] OD  
ON O.OrderID = OD.OrderID  
JOIN Products P  
ON P.ProductID = OD.ProductId  
GROUP BY  
CUBE(CompanyName, ProductName, year(orderdate))  
ORDER BY 1,2,3
```

	CompanyNa...	ProductName	Orders_Y...	Total
1	NULL	NULL	NULL	1354458,59
2	NULL	NULL	1996	226298,50
3	NULL	NULL	1997	658388,75
4	NULL	NULL	1998	469771,34
5	NULL	Alice Mutton	NULL	35482,20
6	NULL	Alice Mutton	1996	7300,80
7	NULL	Alice Mutton	1997	19718,40
8	NULL	Alice Mutton	1998	8463,00
9	NULL	Aniseed Syrup	NULL	3080,00
10	NULL	Aniseed Syrup	1996	240,00
11	NULL	Aniseed Syrup	1997	1760,00
12	NULL	Aniseed Syrup	1998	1080,00
13	NULL	Boston Crab...	NULL	19048,30
14	NULL	Boston Crab...	1996	2998,80
15	NULL	Boston Crab...	1997	10474,30
16	NULL	Boston Crab...	1998	5575,20

Резултатът ще съдържа:

NULL -За всички клиенти	NULL -За всички стоки	NULL -За всички години
NULL - За всички клиенти	NULL -За всички стоки	За всяка година
NULL -За всички клиенти	За всяка стока	NULL – за всички години
За всеки клиент	NULL-За всички стоки	NULL – за всички години
За всеки клиент	NULL – за всички години	За всяка година
За всеки клиент	За всяка стока	NULL – за всички години
За всеки клиент	За всяка стока	За всяка година

Частичен синтаксис на SELECT с клауза GROUP BY

```
SELECT <select_list>  
FROM {<table source>}[,.....n]  
[ WHERE <search_condition> ]  
[GROUP BY <group by spec>  
|ROLLUP (<group by spec>)  
|CUBE(<group by spec>) ]  
[HAVING <search_condition> ]  
[ ORDER BY order_expression  
[ ASC | DESC ][,....] ]
```