

Retrieving all columns and rows from dataset.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PharmaInfo', including tables like 'Pharma_data_analysis'. The central query window contains the following SQL code:

```
--retrieving all columns and rows from dataset
select * from Pharma_data_analysis;
```

The Results pane at the bottom displays the query output as a table with 10 columns: Distributor, Customer_Name, City, Country, Latitude, Longitude, Channel, Sub_channel, Product_Name, and Product_Class. The table contains 9 rows of data, all from Poland.

Distributor	Customer_Name	City	Country	Latitude	Longitude	Channel	Sub_channel	Product_Name	Product_Class
Stehr-Champlin	Upton, Abemathy and Goodwin Pharmaceutical Ltd	Jaroslav	Poland	50.0162010192871	22.6777992248535	Hospital	Government	Palodizem	Mood Stabilizers
Stehr-Champlin	Senger-Klein Pharmacy	Kobylika	Poland	52.3395004272461	21.1958999633789	Hospital	Government	Andropoin	Antiseptics
Stehr-Champlin	Waelchi LLC Pharmacy	Katowice	Poland	50.25	19	Hospital	Government	Acynafine Microvate	Mood Stabilizers
Stehr-Champlin	Bernier, Murphy and Rau Pharmaceutical Ltd	Ruda Śląska	Poland	50.2627983093262	18.8535995483398	Pharmacy	Institution	Xenaminphen	Antimalarial
Stehr-Champlin	Plannerstill Group Pharmaceutical Ltd	Jelcz-Laskowice	Poland	51.0332984924316	17.3332996368408	Pharmacy	Retail	Vibratosine Alentrace	Antibiotics
Stehr-Champlin	Botford and Sons Pharmaceutical Ltd	Kolobrzeg	Poland	54.1666984558105	15.5666999816895	Pharmacy	Institution	Ceretosine	Antiseptics
Stehr-Champlin	Nolan, DuBuque and Schultz Pharm	Walbrzych	Poland	50.7667007446289	16.2833003997803	Hospital	Government	Pulmogen Altora	Antiseptics
Stehr-Champlin	Lemke Ltd Pharm	Tarnów	Poland	50.0125007629395	20.9883003234863	Hospital	Private	Ceftatana Serolamide	Antipretics
Stehr-Champlin	Stanton Cole	Skieriewice	Poland	51.9528007507324	20.1417007446289	Hospital	Private	Eproline	Antibiotics

The status bar at the bottom indicates 'Query executed successfully.' and '2,54,082 rows'.

Retrieving unique countries in dataset.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PharmaInfo'. The central query window contains the following SQL code:

```
--retrieving unique countries in dataset
select count(distinct(Country)) as No_of_countries from Pharma_data_analysis;
```

The Results pane at the bottom displays the query output as a table with 1 column: 'No_of_countries'. The table contains 1 row with the value 2.

No_of_countries
2

The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

Name of the customers from Retail Channel.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Pharmainfo'. The query editor in the center contains the following SQL query:

```
--Name of the customers from Retail Channel
select Customer_Name,Sub_channel from Pharma_data_analysis where Sub_channel='Retail';
```

The Results pane on the right displays the query output as a table with two columns: 'Customer_Name' and 'Sub_channel'. The table contains 17 rows of data, all with 'Retail' in the 'Sub_channel' column.

Customer_Name	Sub_channel
1 Pfannestill Group Pharmaceutical Ltd	Retail
2 Rohan PLC Pharmaceutical Limited	Retail
3 Will and Sons Pharma Plc	Retail
4 Steuber, Skiles and Kilback Pharma Plc	Retail
5 McCullough LLC Pharma Plc	Retail
6 Welch-Langworth	Retail
7 Moen, Murazik and Smith	Retail
8 Sauer and Sons Pharmaceutical Limited	Retail
9 Schowalter, Runolfsson and VonRueden Pharmaceuti...	Retail
10 Nader-Gaylord Pharmaceutical Ltd	Retail
11 Koss-Beier	Retail
12 Greenholt, Donnelly and O'Reilly Pharmaceutical Limited	Retail
13 Walsh-Brown Pharmacy	Retail
14 Nader-Borer Pharma Plc	Retail
15 McCullough LLC Pharma Plc	Retail
16 Kilback, Beer and Batz Pharma Plc	Retail
17 Lindgren-Simonis Pharm	Retail

The status bar at the bottom indicates 'Query executed successfully.' and '68,351 rows'.

Total quantity sold for the 'antibiotics' product class.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Pharmainfo'. The query editor in the center contains the following SQL query:

```
--total quantity sold for the 'antibiotics' product class
select Product_Class,round(sum(Quantity),0) as Total_Quantity_Sold from Pharma_data_analysis
where Product_Class='Antibiotics' group by Product_Class;
```

The Results pane on the right displays the query output as a table with two columns: 'Product_Class' and 'Total_Quantity_Sold'. The table contains one row of data for the 'Antibiotics' product class.

Product_Class	Total_Quantity_Sold
1 Antibiotics	4154322

The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

Distinct months present in the dataset.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-DT4PIUD\SQLEXPRESS (S...)' under the 'PharmaInfo' database. The query editor in the center contains the following SQL code:

```
--distinct months present in the dataset
select distinct(Month) from Pharma_data_analysis;
```

The Results pane at the bottom shows the output of the query, displaying a list of months from February to March. The status bar at the bottom indicates 'Query executed successfully.' and '12 rows'.

Month
February
June
August
April
May
December
January
September
October
July
November
March

Total Sales for each year.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-DT4PIUD\SQLEXPRESS (S...)' under the 'PharmaInfo' database. The query editor in the center contains the following SQL code:

```
--Total Sales for each year
select Year,round(sum(Sales),0) as Total_Sales from Pharma_data_analysis
group by Year Order by Year asc;
```

The Results pane at the bottom shows the output of the query, displaying a table with columns 'Year' and 'Total_Sales'. The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

Year	Total_Sales
2017	2701480741
2018	3506897354
2019	2930937133
2020	2659672415

Customer with highest sales value.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PharmaInfo'. The main query window contains the following SQL code:

```
--Customer with highest sales value
select Top 1 Customer_Name,sum(Sales) as Highest_Sales_Value from Pharma_data_analysis
group by Customer_Name Order by Highest_Sales_Value desc;
```

The Results pane at the bottom shows the output of the query:

	Customer_Name	Highest_Sales_Value
1	Maz-Kutch Pharma Plc	\$3561780

The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

Names of all employees who are Sales Rep and are managed by 'James Goodwill'.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PharmaInfo'. The main query window contains the following SQL code:

```
--Names of all employees who are Sales Rep and are managed by 'James Goodwill'
select distinct(Name_of_Sales_Rep),Manager from Pharma_data_analysis
where Manager='James Goodwill';
```

The Results pane at the bottom shows the output of the query:

	Name_of_Sales_Rep	Manager
1	Alan Ray	James Goodwill
2	Erica Jones	James Goodwill
3	Thompson Crawford	James Goodwill

The status bar at the bottom indicates 'Query executed successfully.' and '3 rows'.

Retrieve the top 5 cities with the highest sales.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Pharmainfo'. The main query window contains the following SQL code:

```
--retrieve the top 5 cities with the highest sales.  
select Top 5 City,round(sum(Sales),0) as Highest_Sales from Pharma_data_analysis  
group by City order by Highest_Sales desc;
```

The Results pane at the bottom displays the following data:

	City	Highest_Sales
1	Butzbach	93561780
2	Baesweiler	64890501
3	Cuxhaven	56006680
4	Friedberg	52183635
5	Altenburg	50885320

The status bar at the bottom indicates 'Query executed successfully.' and '5 rows'.

Average price of products in each sub channel.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Pharmainfo'. The main query window contains the following SQL code:

```
--average price of products in each sub channel  
select Sub_channel,avg(Price) as Average_Price_of_Products from Pharma_data_analysis  
group by Sub_channel;
```

The Results pane at the bottom displays the following data:

	Sub_channel	Average_Price_of_Products
1	Institution	411
2	Government	413
3	Private	410
4	Retail	412

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

Name of the Sales rep and the corresponding sales records.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Pharmainfo'. The central query editor contains the following SQL code:

```
--Name of the Sales rep and the corresponding sales records
select distinct(Name_of_Sales_Rep),round(sum(Sales),0) as Total_Sales
from Pharma_data_analysis group by Name_of_Sales_Rep order by Total_Sales desc;
```

The Results pane at the bottom displays the output of the query as a table with 13 rows. The status bar at the bottom indicates 'Query executed successfully.' and '13 rows'.

	Name_of_Sales_Rep	Total_Sales
1	Jimmy Grey	985969994
2	Abigail Thompson	981056994
3	Shella Stones	958203898
4	Daniel Gates	950658635
5	Anne Wu	920168301
6	Moris Garcia	901195483
7	Stella Given	888340902
8	Jessica Smith	881698369
9	Steve Pepple	875449983
10	Mary Gerard	875270763
11	Erica Jones	871372192
12	Thompson Crawford	866964886
13	Alan Ray	842637242

Retrieve all sales made by employees from 'Rendsburg' in the year 2018.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Pharmainfo'. The central query editor contains the following SQL code:

```
--Retrieve all sales made by employees from 'Rendsburg' in the year 2018
select Year,City,sum(Sales) as Total_Sales from Pharma_data_analysis where Year=2018
and City='Rendsburg' group by Year,City;
```

The Results pane at the bottom displays the output of the query as a table with 1 row. The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

	Year	City	Total_Sales
1	2018	Rendsburg	9528627

Calculate total sales for each product class,for each month and order the results by year,month and product class.(months and product class are sort in alphabetical order from A-Z based on yearwise)

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--Calculate total sales for each product class,for each month and order the results by
--year,month and product class
select Year,Month,Product_Class,round(sum(Sales),0) as Total_Sales from Pharma_data_analysis
group by Year,Month,Product_Class
order by Year,Month,Product_Class asc;
(-- months and product class are sort in alphabetical order from A-Z based on yearwise)
```

The Results pane displays the following data:

Year	Month	Product_Class	Total_Sales
2017	April	Analgesics	32223716
2017	April	Antibiotics	40029228
2017	April	Antimalarial	17789675
2017	April	Antipretics	22868812
2017	April	Antiseptics	42712211
2017	April	Mood Stabilizers	33776544
2017	August	Analgesics	49744620
2017	August	Antibiotics	32449096
2017	August	Antimalarial	25887712
2017	August	Antipretics	39342305
2017	August	Antiseptics	45881555
2017	August	Mood Stabilizers	40529487
2017	December	Analgesics	64873444
2017	December	Antibiotics	28906649
2017	December	Antimalarial	20408987
2017	December	Antipretics	33815804

Query executed successfully. 288 rows.

Find Top 3 sales rep with the highest sales in 2019.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--Find Top 3 sales rep with the highest sales in 2019
select Top 3 Year,Name_of_Sales_Rep,round(sum(Sales),0) as Highest_Sales
from Pharma_data_analysis where Year=2019
group by Year,Name_of_Sales_Rep order by Highest_Sales desc;
```

The Results pane displays the following data:

Year	Name_of_Sales_Rep	Highest_Sales
2019	Jimmy Grey	310551051
2019	Sheila Stones	266924378
2019	Daniel Gates	245363929

Query executed successfully. 3 rows.

Calculate monthly total sales for each sub-channel ,and then calculate the average monthly sales for each sub-channel over the years.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--Calculate monthly total sales for each sub-channel and then calculate the average monthly
--sales for each sub-channel over the years.
select Year,Month,Sub_channel,round(sum(Sales),0) as Total_Sales from Pharma_data_analysis
group by Year,Month,Sub_channel
order by Year asc,Month asc>Total_Sales desc;

select Year,Month,Sub_channel,round(avg(Sales),0) as Avg_Sales from Pharma_data_analysis
group by Year,Month,Sub_channel
order by Year asc,Month asc,Avg_Sales desc;
```

The Results pane displays two tables. The first table shows monthly total sales, and the second table shows average monthly sales.

Year	Month	Sub_channel	Total_Sales
2017	April	Institution	50151370
2017	April	Retail	49076812
2017	April	Government	45892380
2017	April	Private	43680022
2017	August	Retail	67480099
2017	August	Government	61552965
2017	August	Institution	57379276
2017	August	Private	47422335

Year	Month	Sub_channel	Avg_Sales
2017	April	Institution	52847
2017	April	Private	50673
2017	April	Retail	44333
2017	April	Government	42651
2017	August	Retail	48442
2017	August	Government	45662
2017	August	Institution	45323
2017	August	Private	42379

Query executed successfully. DESKTOP-DT4PIUD\SQLEXPRESS ... DESKTOP-DT4PIUD\Himesh... PharmaInfo 00:00:00 192 rows

Create a summary report that includes the total sales,average price and total quantity sold for each product.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--Create a summary report that includes the total sales,average price and total quantity sold
--for each product
select Product_Name,round(sum(Sales),0) as Total_Sales,avg(Price) as Average_Price,sum(Quantity)
Total_Quantity from Pharma_data_analysis group by Product_Name;
```

The Results pane displays a table with the following data:

Product_Name	Total_Sales	Average_Price	Total_Quantity
Zontoron	50508626	518	97507
Alpradipine	49463085	445	111153
Asparathasone Unipan	16242480	153	106160
Argalazine Abotstyl	50872553	628	81007.25
Enzastyl	2863458	26	110133
Andropoin	76947476	659	116764
Neuropogen Empibax	91751275	605	151655
Pazofenac	15022482	123	122134
Symbitrim	83613856	536	155996
Ionclotide	169083391	631	267961
Lovephilus	69439002	582	119311
Iburlinum	28558710	190	150309
Aggrakine	5667072	47	120576

Query executed successfully. DESKTOP-DT4PIUD\SQLEXPRESS ... DESKTOP-DT4PIUD\Himesh... PharmaInfo 00:00:00 240 rows

Find the top 5 customers with the highest sales for each year.

Using Windows function by applying ranking and partition of each years based on Highest sales,we will apply filter to top 5 customers.

The screenshot shows a SQL query in SQL Server Enterprise Manager. The query uses a window function to rank customers by sales for each year and filters for the top 5.

```
--Find the top 5 customers with the highest sales for each year.  
--Using Windows function by applying ranking and partition of each years based on Highest sales,  
--we will apply filter to top 5 customers  
select Year, Customer_name, Highest_Sales from  
(select Year, Customer_name, sum(Sales) as Highest_Sales,  
Rank() OVER (Partition By Year Order By sum(Sales) desc) as rankings from Pharma_data_analysis  
group by Year, Customer_name) Pharma_data_analysis where rankings<=5;
```

The results table shows the top 5 customers for each year from 2017 to 2020.

Year	Customer_name	Highest_Sales
2017	Wiegand, Jast and Yost Pharmaceutical Ltd	20947974
2017	Raynor-Graham	20691892
2017	Fadel-West Pharmaceutical Ltd	19381932
2017	Kuphal, Herzog and Purdy	16707639
2017	Leannon-West Pharmaceutical Limited	16639689
2018	Barrows, Zboncak and Reichert Pharm	22713841
2018	Zemlak Group Pharm	20691357
2018	Watsica, Larson and Labadie Pharmaceutical Ltd	20200981
2018	Senger-Klein Pharmaceutical Ltd	19949284
2018	McDemott Inc Pharmacy	19511107
2019	Mraz-Kutch Pharma Plc	76494324
2019	Zemlak-Witting	36611325
2019	Streich PLC	31116982
2019	Glechner, Bahringer and Morar Pharmaceutical...	27011286
2019	Prohaska, Bogisch and Gukowski Pharmaceut...	26786242
2020	Parker, Green and Emmerich Pharma Plc	51565996
2020	Torphy, Pfeffer and Jakubowski Pharm	27596295
2020	Stehr-Champlin Pharmacy	26688534
2020	Runolfsson, Swianiewski and Jaskolski Pharma...	25132379
2020	Goldner-Tilman Pharm	24981748

Calculate Year-over_Year growth in sales for each country.

By using LAG Function we will calculate previous year sales and create new table of YoYsales.

Based on that new table (YoY sales) we have calculate sales_difference and YoY% Growth by using CASE Expression.

The screenshot shows a SQL query in SQL Server Enterprise Manager. The query uses the LAG function to calculate previous year sales and then calculates the sales difference and year-over-year percentage growth using a CASE expression.

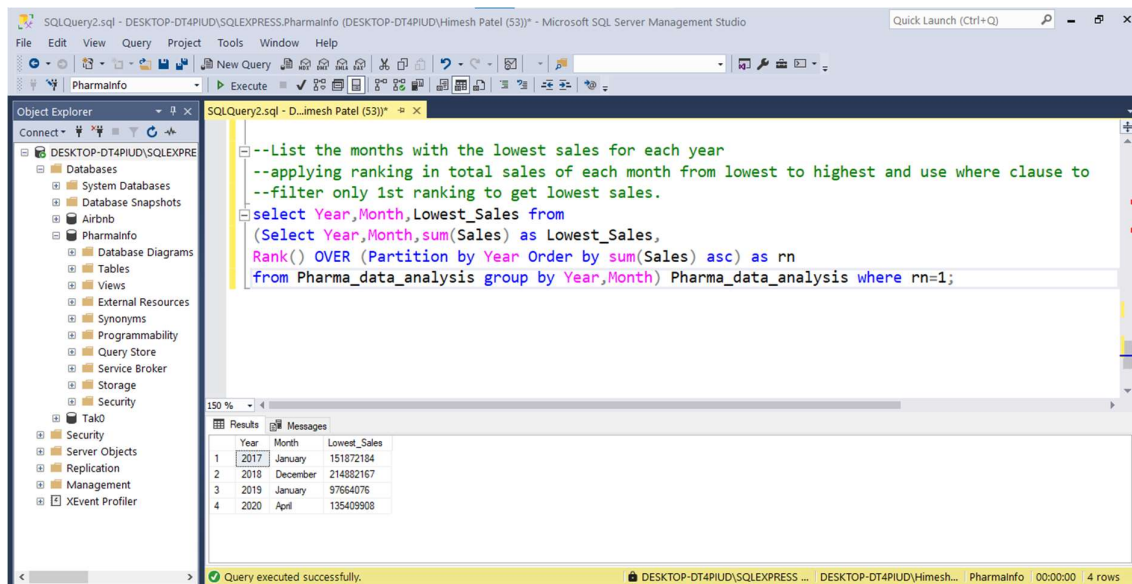
```
--Calculate Year-over_Year growth in sales for each country  
--by using LAG Function we will calculate previous year sales and create new table of YoYsales.  
select Country, Year, round(sum(Sales),0) as Actual_Sales, LAG(sum(Sales)) Over  
(Partition by Country Order By Year) as previous_year into YoYsales from Pharma_data_analysis  
group by Country, Year;  
  
-- Calculate sales_difference and YoY% Growth by using CASE Expression.  
select Country, Year, Actual_Sales, round(previous_year,0) as Previous_Year,  
round((Actual_Sales-previous_year),0) as sales_difference,  
CASE WHEN previous_year=0 THEN NULL  
ELSE round(((Actual_Sales-previous_year)/previous_year)*100,2) END as "YoY%" from YoYsales;
```

The results table shows the sales data for Germany and Poland from 2017 to 2020, including the previous year's sales, the sales difference, and the year-over-year percentage growth.

Country	Year	Actual_Sales	Previous_Year	sales_difference	YoY%
Germany	2017	2701480741	NULL	NULL	NULL
Germany	2018	2826017552	2701480741	124536811	4.61
Germany	2019	2930937133	2826017552	104919581	3.71
Germany	2020	2659672415	2930937133	-271264718	-9.26
Poland	2018	680879802	NULL	NULL	NULL

List the months with the lowest sales for each year.

by applying ranking in total sales of each month from lowest to highest and use 'where' clause to filter only 1st ranking to get lowest sales for each year.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
--List the months with the lowest sales for each year
--applying ranking in total sales of each month from lowest to highest and use where clause to
--filter only 1st ranking to get lowest sales.
select Year,Month,Lowest_Sales from
(select Year,Month,sum(Sales) as Lowest_Sales,
Rank() OVER (Partition by Year Order by sum(Sales) asc) as rn
from Pharma_data_analysis group by Year,Month) Pharma_data_analysis where rn=1;
```

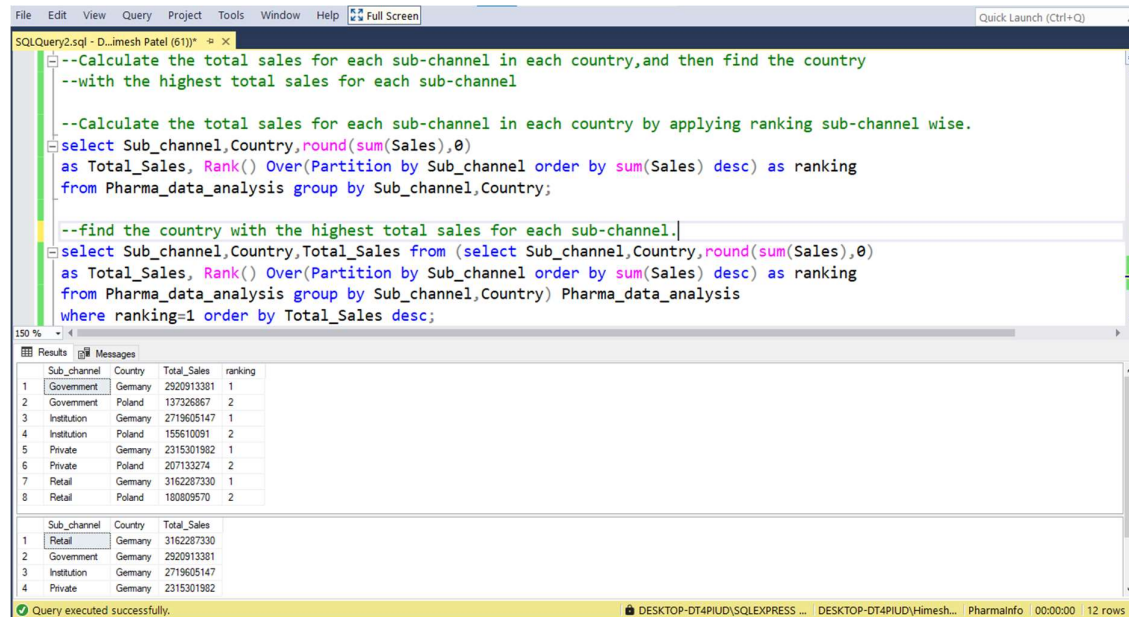
The results pane displays the following data:

	Year	Month	Lowest_Sales
1	2017	January	151872184
2	2018	December	214882167
3	2019	January	97664076
4	2020	April	135409908

The status bar at the bottom indicates: Query executed successfully. DESKTOP-DT4PIUD\SQLEXPRESS ... | DESKTOP-DT4PIUD\Himesh... PharmaInfo 00:00:00 4 rows

Calculate the total sales for each sub-channel in each country, and then find the country with the highest total sales for each sub-channel.

We have calculated the total sales for each sub-channel in each country by applying ranking in each sub-channel wise and then find the country with the highest total sales for each sub-channel.



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
--Calculate the total sales for each sub-channel in each country, and then find the country
--with the highest total sales for each sub-channel

--Calculate the total sales for each sub-channel in each country by applying ranking sub-channel wise.
select Sub_channel, Country, round(sum(Sales), 0)
as Total_Sales, Rank() Over(Partition by Sub_channel order by sum(Sales) desc) as ranking
from Pharma_data_analysis group by Sub_channel, Country;

--find the country with the highest total sales for each sub-channel.
select Sub_channel, Country, Total_Sales from (select Sub_channel, Country, round(sum(Sales), 0)
as Total_Sales, Rank() Over(Partition by Sub_channel order by sum(Sales) desc) as ranking
from Pharma_data_analysis group by Sub_channel, Country) Pharma_data_analysis
where ranking=1 order by Total_Sales desc;
```

The bottom pane shows the results of the query, which are displayed in two tables. The first table lists the top-ranked country for each sub-channel, and the second table lists the top-ranked country for each sub-channel.

Sub_channel	Country	Total_Sales	ranking
1 Government	Germany	2920913381	1
2 Government	Poland	137326867	2
3 Institution	Germany	2719605147	1
4 Institution	Poland	155610091	2
5 Private	Germany	2315301982	1
6 Private	Poland	207133274	2
7 Retail	Germany	3162287330	1
8 Retail	Poland	180809570	2

Sub_channel	Country	Total_Sales
1 Retail	Germany	3162287330
2 Government	Germany	2920913381
3 Institution	Germany	2719605147
4 Private	Germany	2315301982

The status bar at the bottom indicates that the query was executed successfully and that there are 12 rows in the results.