

## **SORTING AND GROUPING DATA**

**-- Create Country table with fields: Id ,Country\_name ,Population & Area**

```
CREATE TABLE Country (  
    Id INT PRIMARY KEY,  
    Country_name VARCHAR(50),  
    Population BIGINT,  
    Area INT  
);  
  
DESC country;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	Id	int	NO	PRI	NULL	
	Country_name	varchar(50)	YES		NULL	
	Population	bigint	YES		NULL	
	Area	int	YES		NULL	

**-- Create persons table with fields: Id ,Fname, Lname, Population, Rating ,Country\_Id & Country\_name.**

```
CREATE TABLE Persons (  
    Id INT PRIMARY KEY,  
    Fname VARCHAR(50),  
    Lname VARCHAR(50),  
    Population BIGINT,  
    Rating DECIMAL(3, 2),  
    Country_Id INT,  
    Country_name VARCHAR(50),  
    FOREIGN KEY (Country_Id) REFERENCES Country(Id)
```

);

DESC persons;

Result Grid		Filter Rows:		Export:	Wrap Cell Content:	
	Field	Type	Null	Key	Default	Extra
►	Id	int	NO	PRI	<b>NULL</b>	
	Fname	varchar(50)	YES		<b>NULL</b>	
	Lname	varchar(50)	YES		<b>NULL</b>	
	Population	bigint	YES		<b>NULL</b>	
	Rating	decimal(3,2)	YES		<b>NULL</b>	
	Country_Id	int	YES	MUL	<b>NULL</b>	
	Country_name	varchar(50)	YES		<b>NULL</b>	

### -- Insert data into Country table

INSERT INTO Country (Id, Country\_name, Population, Area)

VALUES

(1, 'USA', 75000000, 9833520),

(2, 'India', 5000000, 3287263),

(3, 'Canada', 8000000, 9984670),

(4, 'UK', 150000000, 243610),

(5, 'Australia', 400000, 7692024),

(6, 'Germany', 2000000, 357022),

(7, 'France', 6900000, 551695),

(8, 'Brazil', 6000000, 8515767),

(9, 'Russia', 15500000, 17098242),

(10, 'Japan', 8650000, 377930);

select \* from country;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	Id	Country_name	Population	Area
▶	1	USA	75000000	9833520
	2	India	5000000	3287263
	3	Canada	8000000	9984670
	4	UK	150000000	243610
	5	Australia	400000	7692024
	6	Germany	2000000	357022
	7	France	6900000	551695
	8	Brazil	6000000	8515767
	9	Russia	15500000	17098242
	10	Japan	8650000	377930
✱	NULL	NULL	NULL	NULL

### -- Insert data into Persons table

```
INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name)
```

```
VALUES
```

```
(1, 'Jeeva', 'Joseph', 75000000, 4.5, 1, 'USA'),
(2, 'Arya', 'Shan', 5000000, 3.8, 2, 'India'),
(3, 'Emma', 'Emmi', 8000000, 4.9, 3, 'Canada'),
(4, 'Basim', 'Johny', 150000000, 4.2, 4, 'UK'),
(5, 'Aryan', 'John', 400000, 3.9, 5, 'Australia'),
(6, 'Maria', 'Grace', 2000000, 4.7, 6, 'Germany'),
(7, 'James', 'Wilson', 6900000, 2.1, 7, 'France'),
(8, 'Suzzy', 'Martin', 6000000, 4.0, 8, 'Brazil'),
(9, 'Robert', 'Williams', 15500000, 1.7, 9, 'Russia'),
(10, 'Jane', 'Taylor', 8650000, 4.6, 10, 'Japan');
```

```
select * from persons;
```

Result Grid		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	Id	Fname	Lname	Population	Rating	Country_Id	Country_name		
▶	1	Jeeva	Joseph	75000000	4.50	1	USA		
	2	Arya	Shan	5000000	3.80	2	India		
	3	Emma	Emmi	8000000	4.90	3	Canada		
	4	Basim	Johny	150000000	4.20	4	UK		
	5	Aryan	John	400000	3.90	5	Australia		
	6	Maria	Grace	2000000	4.70	6	Germany		
	7	James	Wilson	6900000	2.10	7	France		
	8	Suzzy	Martin	6000000	4.00	8	Brazil		
	9	Robert	Williams	15500000	1.70	9	Russia		
	10	Jane	Taylor	8650000	4.60	10	Japan		
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL		

-- (1) Write an SQL query to print the first three characters of Country\_name from the Country table.

```
SELECT LEFT(Country_name, 3) AS First_Three_Chars FROM Country;
```

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	First_Three_Chars						
▶	USA						
	Ind						
	Can						
	UK						
	Aus						
	Ger						
	Fra						
	Bra						
	Rus						
	Jap						

-- (2) Write an SQL query to concatenate first name and last name from Persons table.

```
SELECT CONCAT(Fname, ' ', Lname) AS Full_Name FROM Persons;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Full_Name			
▶	Jeeva Joseph			
	Arya Shan			
	Emma Emmi			
	Basim Johny			
	Aryan John			
	Maria Grace			
	James Wilson			
	Suzzy Martin			
	Robert Williams			
	Jane Taylor			

-- (3) Write an SQL query to count the number of unique country names from Persons table.

```
SELECT COUNT(DISTINCT Country_Id) AS Unique_Country_Names FROM Persons;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Unique_Country_Names			
▶	10			

-- (4) Write a query to print the maximum population from the Country table.

```
SELECT MAX(Population) AS Maximum_Population FROM Country;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Maximum_Population			
▶	150000000			

-- (5) Write a query to print the minimum population from Persons table.

```
SELECT MIN(Population) AS Minimum_Population FROM Country;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Minimum_Population			
▶	400000			

-- (6) Insert 2 new rows to the Persons table making the Lname NULL. Then write another query to count Lname from Persons table.

```
INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name)
VALUES
```

```
(11, 'Aishwarya', NULL, 75000000, 4.5, 1, 'USA'),
```

```
(12, 'Daisy', NULL, 15500000, 1.7, 9, 'Russia');
```

```
SELECT * FROM persons;
```

	Id	Fname	Lname	Population	Rating	Country_Id	Country_name
▶	1	Jeeva	Joseph	75000000	4.50	1	USA
	2	Arya	Shan	5000000	3.80	2	India
	3	Emma	Emmi	8000000	4.90	3	Canada
	4	Basim	Johny	150000000	4.20	4	UK
	5	Aryan	John	400000	3.90	5	Australia
	6	Maria	Grace	2000000	4.70	6	Germany
	7	James	Wilson	6900000	2.10	7	France
	8	Suzzy	Martin	6000000	4.00	8	Brazil
	9	Robert	Williams	15500000	1.70	9	Russia
	10	Jane	Taylor	8650000	4.60	10	Japan
	11	Aishw...	NULL	75000000	4.50	1	USA
	12	Daisy	NULL	15500000	1.70	9	Russia
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT COUNT(Lname) AS Count_of_Lnames FROM Persons;
```

	Count_of_Lnames
▶	10

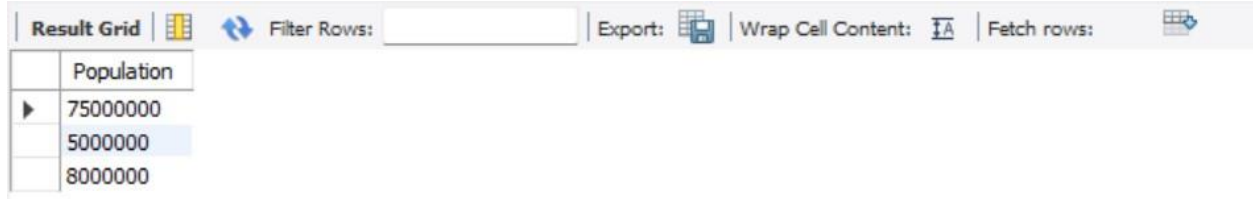
-- (7) Write a query to find the number of rows in the Persons table.

```
SELECT COUNT(*) AS Total_Rows FROM Persons;
```

	Total_Rows
▶	12

-- (8) Write an SQL query to show the population of the Country table for the first 3 rows. (Hint: Use LIMIT)

```
SELECT Population FROM Country LIMIT 3;
```

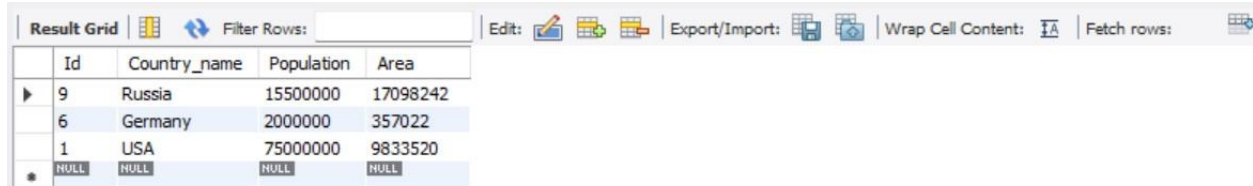


The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a table with a single column labeled 'Population'. The first three rows of data are displayed: 75000000, 5000000, and 8000000.

	Population
▶	75000000
	5000000
	8000000

-- (9) Write a query to print 3 random rows of countries. (Hint: Use rand() function and LIMIT)

```
SELECT * FROM Country ORDER BY RAND() LIMIT 3;
```



The screenshot shows a database interface with a toolbar at the top. Below the toolbar is a table with four columns: 'Id', 'Country\_name', 'Population', and 'Area'. The first three rows of data are displayed, representing random selections: Russia (Id: 9, Population: 15500000, Area: 17098242), Germany (Id: 6, Population: 2000000, Area: 357022), and USA (Id: 1, Population: 75000000, Area: 9833520). A fourth row shows NULL values.

	Id	Country_name	Population	Area
▶	9	Russia	15500000	17098242
	6	Germany	2000000	357022
	1	USA	75000000	9833520
*	NULL	NULL	NULL	NULL

-- (10) List all persons ordered by their rating in descending order.

```
SELECT * FROM Persons ORDER BY Rating DESC;
```



The screenshot shows a database interface with a toolbar at the top. Below the toolbar is a table with eight columns: 'Id', 'Fname', 'Lname', 'Population', 'Rating', 'Country\_Id', and 'Country\_name'. The data is sorted by 'Rating' in descending order. The first three rows are: Emma Emmi (Rating: 4.90, Country: Canada), Maria Grace (Rating: 4.70, Country: Germany), and Jane Taylor (Rating: 4.60, Country: Japan). The table continues with more rows, including Jeeva Joseph (Rating: 4.50, Country: USA), Aishwarya (Rating: 4.50, Country: USA), Basim Johny (Rating: 4.20, Country: UK), Suzzy Martin (Rating: 4.00, Country: Brazil), Aryan John (Rating: 3.90, Country: Australia), Arya Shan (Rating: 3.80, Country: India), James Wilson (Rating: 2.10, Country: France), Robert Williams (Rating: 1.70, Country: Russia), and Daisy (Rating: 1.70, Country: Russia). A final row shows NULL values.

	Id	Fname	Lname	Population	Rating	Country_Id	Country_name
▶	3	Emma	Emmi	8000000	4.90	3	Canada
	6	Maria	Grace	2000000	4.70	6	Germany
	10	Jane	Taylor	8650000	4.60	10	Japan
	1	Jeeva	Joseph	75000000	4.50	1	USA
	11	Aishwarya	NULL	75000000	4.50	1	USA
	4	Basim	Johny	150000000	4.20	4	UK
	8	Suzzy	Martin	6000000	4.00	8	Brazil
	5	Aryan	John	400000	3.90	5	Australia
	2	Arya	Shan	5000000	3.80	2	India
	7	James	Wilson	6900000	2.10	7	France
	9	Robert	Williams	15500000	1.70	9	Russia
	12	Daisy	NULL	15500000	1.70	9	Russia
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- (11) Find the total population for each country in the Persons table.

SELECT Country\_name, SUM(Population) AS Total\_Population FROM Persons GROUP BY Country\_name;

Country_name	Total_Population
USA	150000000
India	5000000
Canada	8000000
UK	150000000
Australia	400000
Germany	2000000
France	6900000
Brazil	6000000
Russia	31000000
Japan	8650000

-- (12) Find countries in the Persons table with a total population greater than 50,000

SELECT Country\_name FROM Persons GROUP BY Country\_name HAVING SUM(Population) > 50000;

Country_name
USA
India
Canada
UK
Australia
Germany
France
Brazil
Russia
Japan

-- (13) List the total number of persons and average rating for each country, but only for countries with more than 2 persons, ordered by the average rating in ascending order



```

SELECT Country_name, COUNT(Id) AS Total_no_of_Persons, AVG(Rating) AS
Average_Rating


FROM Persons

GROUP BY Country_name

HAVING COUNT(Id) >2

ORDER BY Average_Rating ASC;

```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Country_name	Total_no_of_Persons	Average_Rating		

-- List the total number of persons and average rating for each country,  
but only for countries with 2 or more persons, ordered by the average rating in ascending  
order

```

SELECT Country_name, COUNT(Id) AS Total_no_of_Persons, AVG(Rating) AS
Average_Rating



FROM Persons

GROUP BY Country_name

HAVING COUNT(Id) >=2

ORDER BY Average_Rating asc;

```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Country_name	Total_no_of_Persons	Average_Rating		
▶	Russia	2	1.700000		
	USA	2	4.500000		

