**SORTING AND GROUPING DATA**

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

CREATE TABLE Country (

Id INT PRIMARY KEY,

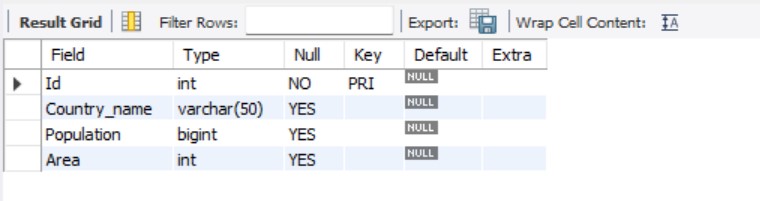
Country\_name VARCHAR(50),

Population BIGINT,

Area INT

);

DESC country;



**-- 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),

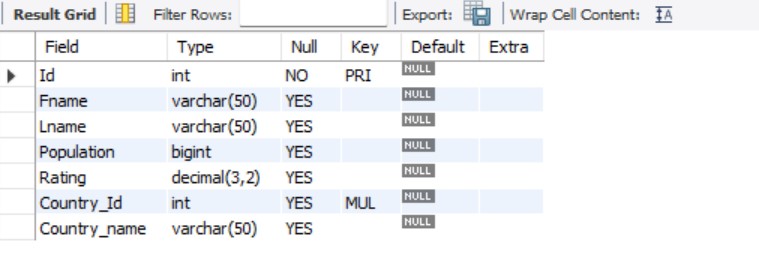
Country\_Id INT,

Country\_name VARCHAR(50),

FOREIGN KEY (Country\_Id) REFERENCES Country(Id)

);

DESC persons;



**-- 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),

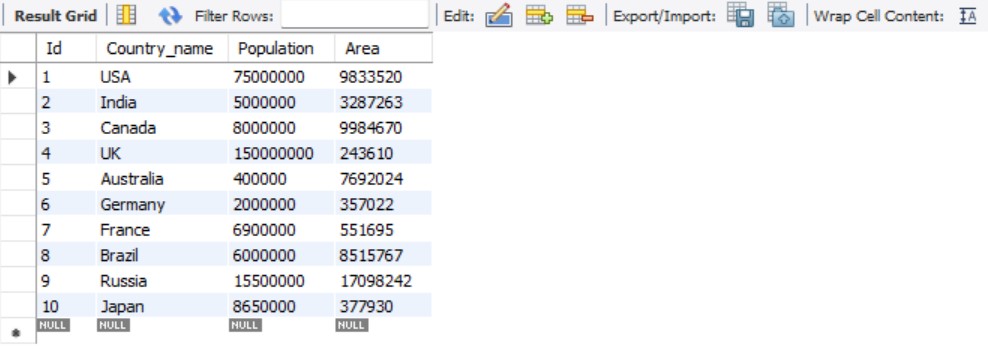
(7, 'France', 6900000, 551695),

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

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

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

select \* from country;



**-- 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'),

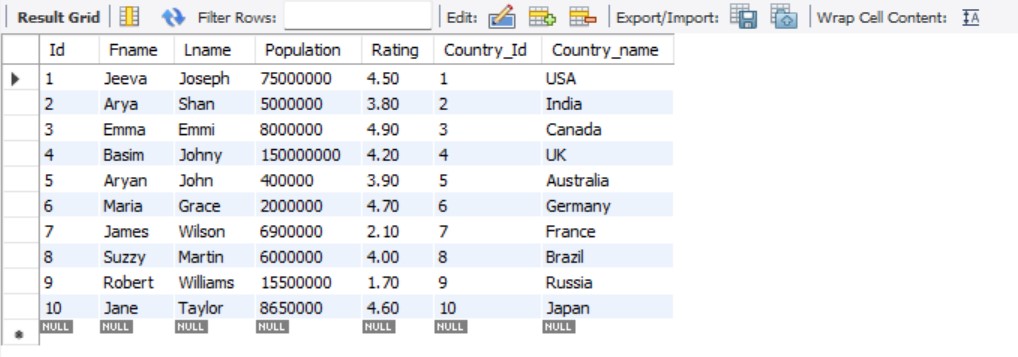
(7, 'James', 'Wilson', 6900000, 2.1, 7, 'France'),

(8, 'Suzzy', 'Martin', 6000000, 4.0, 8, 'Brazil'),

(9, 'Robert', 'Williams', 15500000, 1.7, 9, 'Russia'),

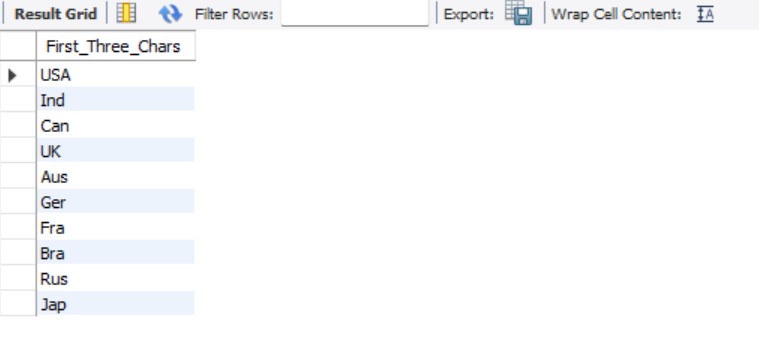
(10, 'Jane', 'Taylor', 8650000, 4.6, 10, 'Japan');

select \* from persons;



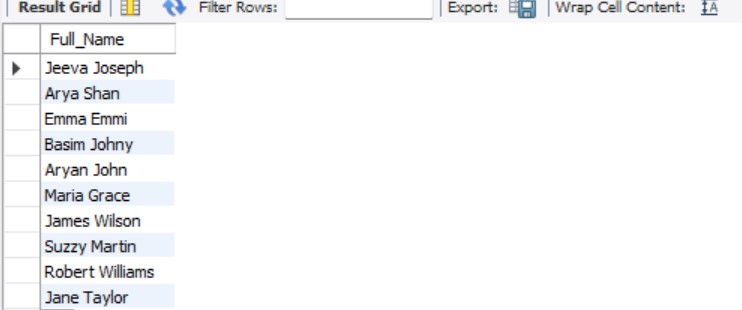
**-- (1) Write an SQL query to print the first three characters of Country\_name from the Country table.**

SELECT LEFT(Country\_name, 3) AS First\_Three\_Chars FROM Country;



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

SELECT CONCAT(Fname, ' ', Lname) AS Full\_Name FROM Persons;



**-- (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;



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

SELECT MAX(Population) AS Maximum\_Population FROM Country;



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

SELECT MIN(Population) AS Minimum\_Population FROM Country;



**-- (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, 'Daisyy', NULL, 15500000, 1.7, 9, 'Russia');

SELECT \* FROM persons;



SELECT COUNT(Lname) AS Count\_of\_Lnames FROM Persons;



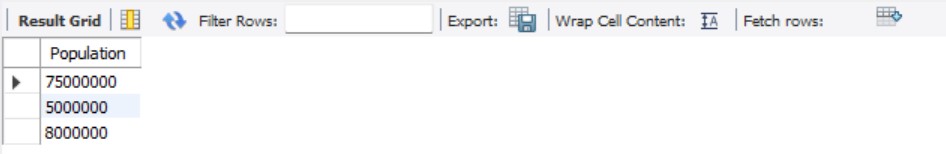
**-- (7) Write a query to find the number of rows in the Persons table.**

SELECT COUNT(\*) AS Total\_Rows FROM Persons;



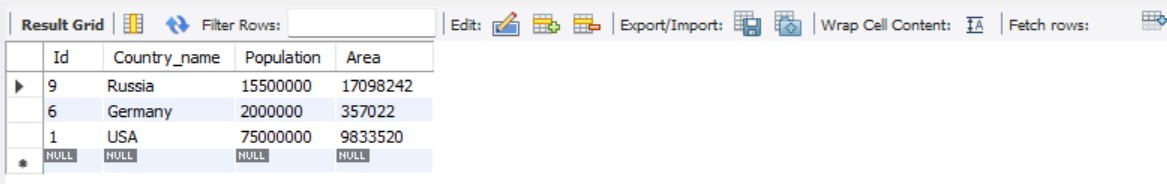
**-- (8) Write an SQL query to show the population of the Country table for the first 3 rows. (Hint: Use LIMIT)**

SELECT Population FROM Country LIMIT 3;



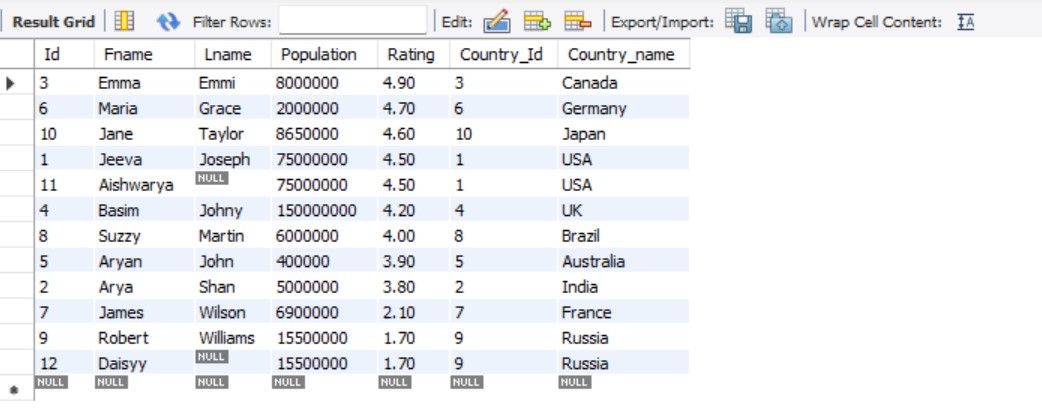
**-- (9) Write a query to print 3 random rows of countries. (Hint: Use rand() function and LIMIT)**

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



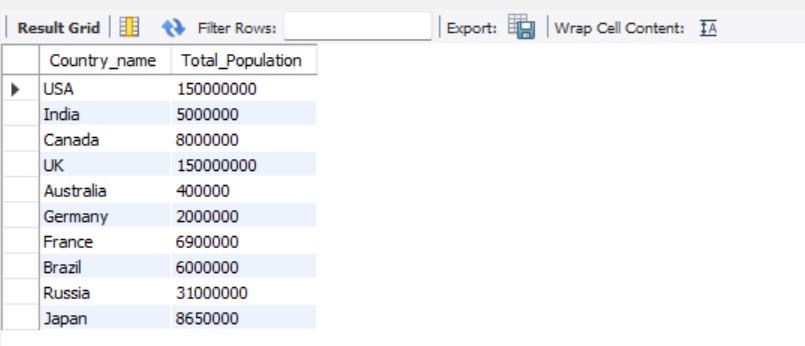
**-- (10) List all persons ordered by their rating in descending order.**

SELECT \* FROM Persons ORDER BY Rating DESC;



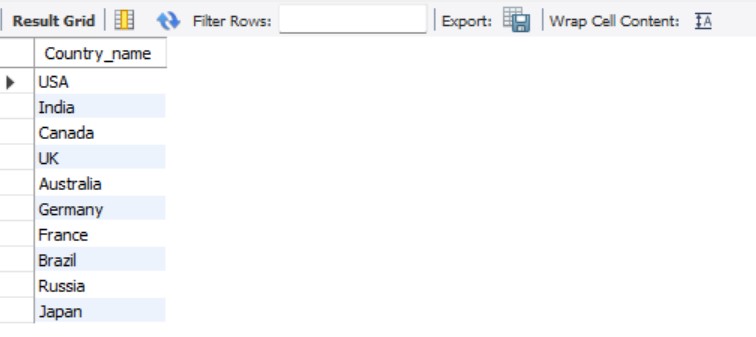
**-- (11)Find the total population for each country in the Persons table.**

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



**-- (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;



**-- (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**

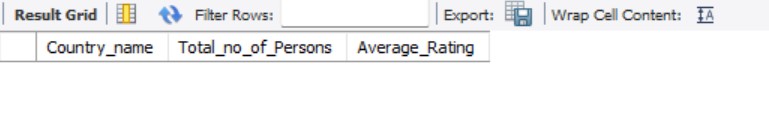
SELECT Country\_name, COUNT(Id) AS Total\_no\_of\_Persons, AVG(Rating) AS Average\_Rating

FROM Persons

GROUP BY Country\_name

HAVING COUNT(Id) >2

ORDER BY Average\_Rating ASC;



-- 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;

