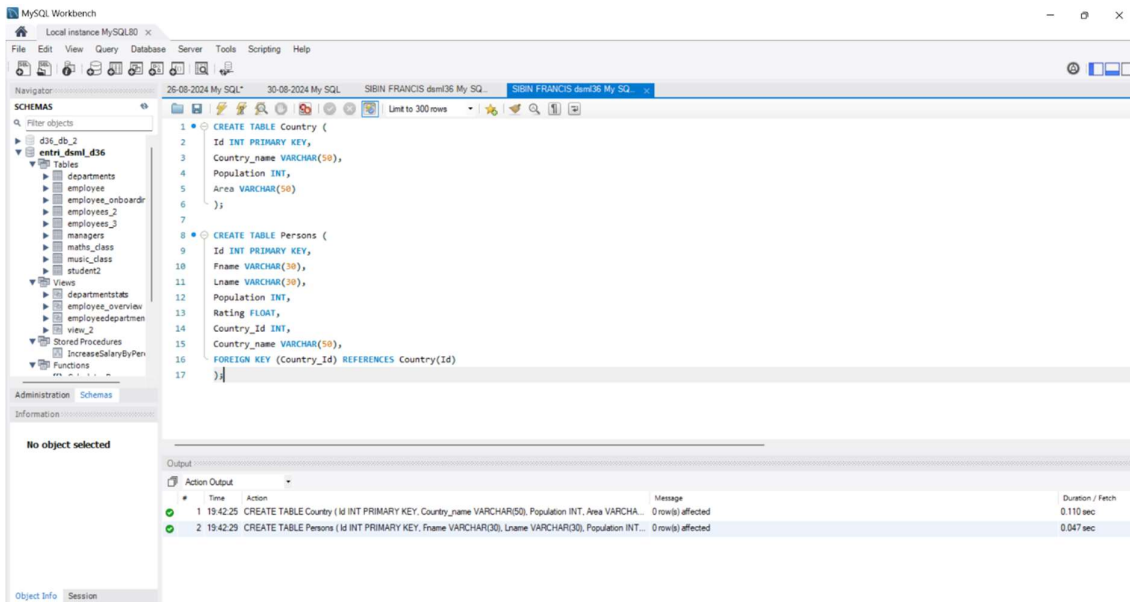


# SORTING AND GROUPING DATA

## ➤ CREATED TABLE **COUNTRY** AND **PERSONS**

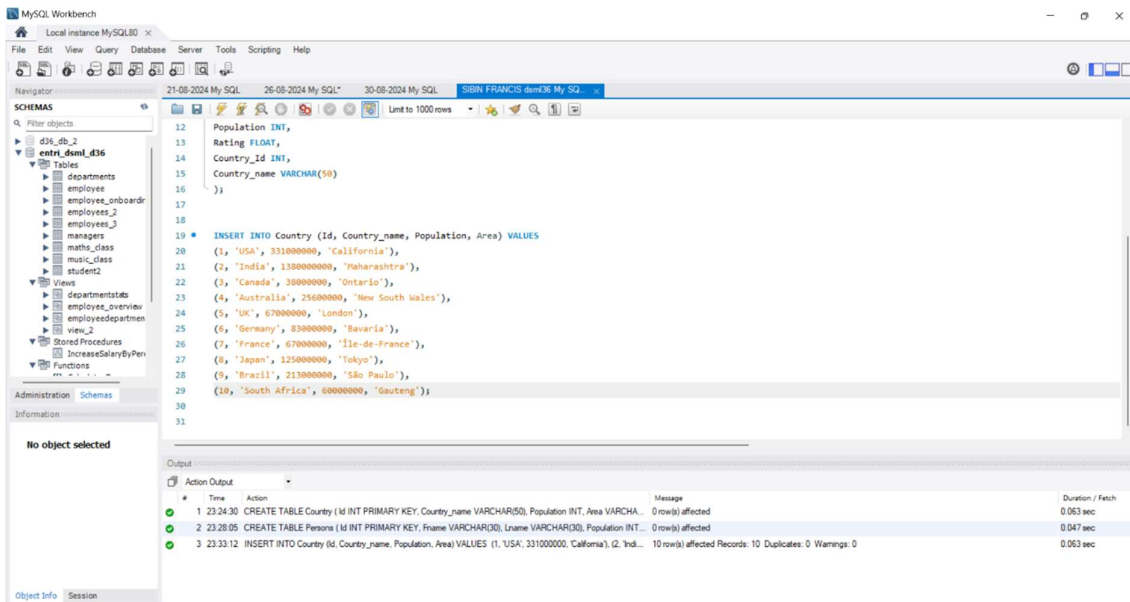


The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd36\_db\_2' selected. The main editor shows the SQL script for creating two tables: 'Country' and 'Persons'. The 'Country' table has columns: Id (INT PRIMARY KEY), Country\_name (VARCHAR(50)), Population (INT), and Area (VARCHAR(50)). The 'Persons' table has columns: Id (INT PRIMARY KEY), Fname (VARCHAR(30)), Lname (VARCHAR(30)), Population (INT), Rating (FLOAT), Country\_Id (INT), and Country\_name (VARCHAR(50)). A foreign key constraint is defined between Country\_Id in Persons and Id in Country. The 'Output' pane at the bottom shows the execution results: 'CREATE TABLE Country' and 'CREATE TABLE Persons' both succeeded, each affecting 0 rows.

```
1 CREATE TABLE Country (  
2   Id INT PRIMARY KEY,  
3   Country_name VARCHAR(50),  
4   Population INT,  
5   Area VARCHAR(50)  
6 );  
7  
8 CREATE TABLE Persons (  
9   Id INT PRIMARY KEY,  
10  Fname VARCHAR(30),  
11  Lname VARCHAR(30),  
12  Population INT,  
13  Rating FLOAT,  
14  Country_Id INT,  
15  Country_name VARCHAR(50),  
16  FOREIGN KEY (Country_Id) REFERENCES Country(Id)  
17 );
```

#	Time	Action	Message	Duration / Fetch
1	19:42:25	CREATE TABLE Country (Id INT PRIMARY KEY, Country_name VARCHAR(50), Population INT, Area VARCHAR(50))	0 row(s) affected	0.110 sec
2	19:42:25	CREATE TABLE Persons (Id INT PRIMARY KEY, Fname VARCHAR(30), Lname VARCHAR(30), Population INT, Rating FLOAT, Country_Id INT, Country_name VARCHAR(50), FOREIGN KEY (Country_Id) REFERENCES Country(Id))	0 row(s) affected	0.047 sec

## ➤ INSERTED VALUES FOR BOTH THE TABLES



The screenshot shows the MySQL Workbench interface with the SQL script for inserting data into the 'Country' and 'Persons' tables. The 'Country' table is populated with 10 rows of data, including USA, India, Canada, Australia, UK, Germany, France, Japan, Brazil, and South Africa. The 'Persons' table is populated with 10 rows of data, including employees from various countries. The 'Output' pane at the bottom shows the execution results: 'CREATE TABLE Country' and 'CREATE TABLE Persons' both succeeded, each affecting 0 rows. The 'INSERT INTO Country' statement affected 10 rows.

```
12 Population INT,  
13 Rating FLOAT,  
14 Country_Id INT,  
15 Country_name VARCHAR(50)  
16 );  
17  
18  
19 INSERT INTO Country (Id, Country_name, Population, Area) VALUES  
20 (1, 'USA', 331000000, 'California'),  
21 (2, 'India', 1380000000, 'Maharashtra'),  
22 (3, 'Canada', 380000000, 'Ontario'),  
23 (4, 'Australia', 25600000, 'New South Wales'),  
24 (5, 'UK', 67000000, 'London'),  
25 (6, 'Germany', 83000000, 'Bavaria'),  
26 (7, 'France', 67000000, 'Île-de-France'),  
27 (8, 'Japan', 125000000, 'Tokyo'),  
28 (9, 'Brazil', 213000000, 'São Paulo'),  
29 (10, 'South Africa', 60000000, 'Gauteng');  
30  
31
```

#	Time	Action	Message	Duration / Fetch
1	23:24:30	CREATE TABLE Country (Id INT PRIMARY KEY, Country_name VARCHAR(50), Population INT, Area VARCHAR(50))	0 row(s) affected	0.063 sec
2	23:28:05	CREATE TABLE Persons (Id INT PRIMARY KEY, Fname VARCHAR(30), Lname VARCHAR(30), Population INT, Rating FLOAT, Country_Id INT, Country_name VARCHAR(50), FOREIGN KEY (Country_Id) REFERENCES Country(Id))	0 row(s) affected	0.047 sec
3	23:33:12	INSERT INTO Country (Id, Country_name, Population, Area) VALUES (1, 'USA', 331000000, 'California'), (2, 'India', 1380000000, 'Maharashtra'), (3, 'Canada', 380000000, 'Ontario'), (4, 'Australia', 25600000, 'New South Wales'), (5, 'UK', 67000000, 'London'), (6, 'Germany', 83000000, 'Bavaria'), (7, 'France', 67000000, 'Île-de-France'), (8, 'Japan', 125000000, 'Tokyo'), (9, 'Brazil', 213000000, 'São Paulo'), (10, 'South Africa', 60000000, 'Gauteng')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.063 sec

## ➤ DISPLAYING BOTH THE TABLES

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd36\_db\_2' selected. The main editor contains a SQL query that inserts data into the 'Persons' table and then selects all data from it. The 'Result Grid' at the bottom shows the output of the SELECT statement, displaying a list of countries with their IDs, names, populations, and areas.

```
31 • INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name)
32 VALUES
33 (1, 'John', 'Doe', 1000000, 4.5, 1, 'USA'),
34 (2, 'Jane', 'Smith', 2000000, 4.8, 2, 'India'),
35 (3, 'Michael', 'Brown', 1500000, 4.2, 3, 'Canada'),
36 (4, 'Emily', 'Davis', 2500000, 3.9, 4, 'Australia'),
37 (5, 'James', 'Wilson', 1800000, 4.6, 5, 'UK'),
38 (6, 'Anna', 'Moore', 1300000, 4.6, 6, 'Germany'),
39 (7, 'Robert', 'Taylor', 1700000, 3.7, 7, 'France'),
40 (8, 'Linda', 'Anderson', 1400000, 4.9, 8, 'Japan'),
41 (9, 'David', 'Thomas', 2100000, 3.8, 9, 'Brazil'),
42 (10, 'Sarah', 'Jackson', 1600000, 4.3, 10, 'South Africa');
43
44 • SELECT * FROM Country;
45 • SELECT * FROM Persons;
```

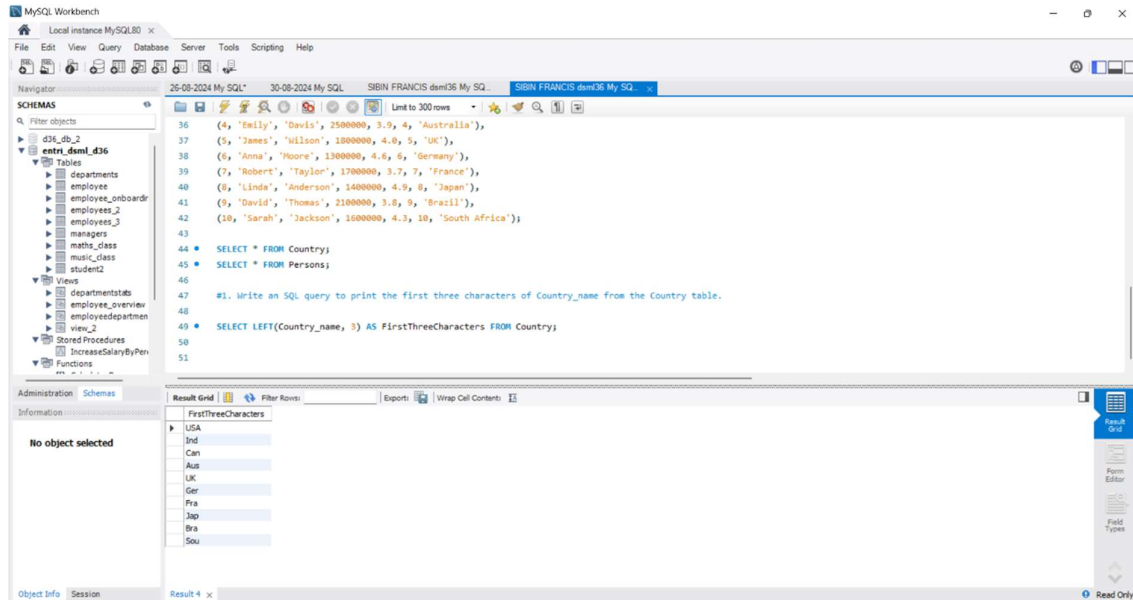
Id	Country_name	Population	Area
1	USA	331000000	California
2	India	1380000000	Maharashtra
3	Canada	38000000	Ontario
4	Australia	25600000	New South Wales
5	UK	67000000	London
6	Germany	83000000	Bavaria
7	France	67000000	Île-de-France
8	Japan	123000000	Tokyo
9	Brazil	213000000	São Paulo
10	South Africa	60000000	Gauteng

This screenshot shows the same MySQL Workbench interface, but the 'Result Grid' now displays the output of the SELECT \* FROM Persons query. The grid shows a list of 10 persons with their IDs, first names, last names, populations, ratings, country IDs, and country names.

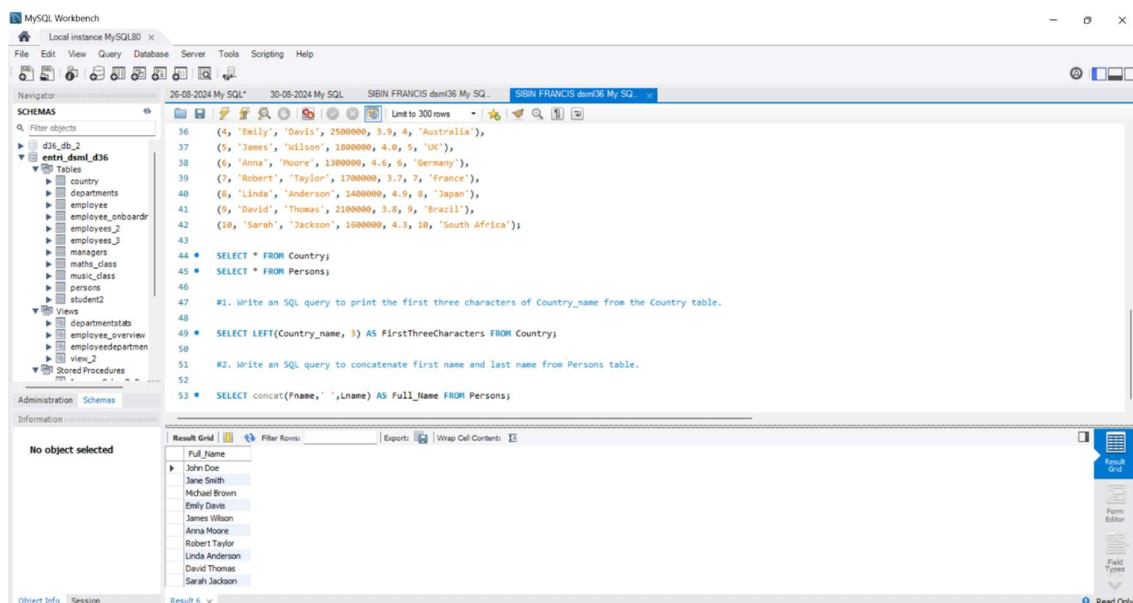
```
30
31 • INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name)
32 VALUES
33 (1, 'John', 'Doe', 1000000, 4.5, 1, 'USA'),
34 (2, 'Jane', 'Smith', 2000000, 4.8, 2, 'India'),
35 (3, 'Michael', 'Brown', 1500000, 4.2, 3, 'Canada'),
36 (4, 'Emily', 'Davis', 2500000, 3.9, 4, 'Australia'),
37 (5, 'James', 'Wilson', 1800000, 4.6, 5, 'UK'),
38 (6, 'Anna', 'Moore', 1300000, 4.6, 6, 'Germany'),
39 (7, 'Robert', 'Taylor', 1700000, 3.7, 7, 'France'),
40 (8, 'Linda', 'Anderson', 1400000, 4.9, 8, 'Japan'),
41 (9, 'David', 'Thomas', 2100000, 3.8, 9, 'Brazil'),
42 (10, 'Sarah', 'Jackson', 1600000, 4.3, 10, 'South Africa');
43
44 • SELECT * FROM Country;
45 • SELECT * FROM Persons;
```

Id	Fname	Lname	Population	Rating	Country_Id	Country_name
1	John	Doe	1000000	4.5	1	USA
2	Jane	Smith	2000000	4.8	2	India
3	Michael	Brown	1500000	4.2	3	Canada
4	Emily	Davis	2500000	3.9	4	Australia
5	James	Wilson	1800000	4	5	UK
6	Anna	Moore	1300000	4.6	6	Germany
7	Robert	Taylor	1700000	3.7	7	France
8	Linda	Anderson	1400000	4.9	8	Japan
9	David	Thomas	2100000	3.8	9	Brazil
10	Sarah	Jackson	1600000	4.3	10	South Africa

## ➤ SQL QUERY TO PRINT THE FIRST THREE CHARACTERS OF **COUNTRY\_NAME** FROM THE **COUNTRY** TABLE.



## ➤ SQL QUERY TO CONCATENATE **FIRST NAME** AND **LAST NAME** FROM **PERSONS** TABLE.



## ➤ SQL QUERY TO COUNT THE NUMBER OF UNIQUE COUNTRY NAMES FROM PERSONS TABLE.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd3s\_db\_2' and 'entri\_demo1\_d36' expanded. The 'entri\_demo1\_d36' table is selected. The main editor shows a SQL query with three numbered comments and three corresponding queries. The first query is a simple SELECT \* FROM Persons;. The second query uses LEFT to get the first three characters of Country\_name. The third query uses COUNT(DISTINCT Country\_name) to count unique country names. The 'Result Grid' at the bottom shows the output of the third query, which is a single row with the value 10.

```
45 SELECT * FROM Persons;
46
47 # 1. Write an SQL query to print the first three characters of Country_name from the Country table.
48
49 SELECT LEFT(Country_name, 3) AS FirstThreeCharacters FROM Country;
50
51 # 2. Write an SQL query to concatenate first name and last name from Persons table.
52
53 SELECT concat(Fname, ' ', Lname) AS Full_Name FROM Persons;
54
55 # 3. Write an SQL query to count the number of unique country names from Persons table.
56
57 SELECT COUNT(DISTINCT Country_name) AS UniqueCountryName FROM Persons;
```

UniqueCountryName
10

## ➤ QUERY TO PRINT THE MAXIMUM POPULATION FROM THE COUNTRY TABLE

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd3s\_db\_2' and 'entri\_demo1\_d36' expanded. The 'entri\_demo1\_d36' table is selected. The main editor shows a SQL query with four numbered comments and four corresponding queries. The first query is a simple SELECT \* FROM Country;. The second query uses LEFT to get the first three characters of Country\_name. The third query uses COUNT(DISTINCT Country\_name) to count unique country names. The fourth query uses MAX(Population) to find the maximum population. The 'Result Grid' at the bottom shows the output of the fourth query, which is a single row with the value 138000000.

```
42 (10, 'Sarah', 'Jackson', 1600000, 4.3, 10, 'South Africa');
43
44 SELECT * FROM Country;
45 SELECT * FROM Persons;
46
47 # 1. Write an SQL query to print the first three characters of Country_name from the Country table.
48
49 SELECT LEFT(Country_name, 3) AS FirstThreeCharacters FROM Country;
50
51 # 2. Write an SQL query to concatenate first name and last name from Persons table.
52
53 SELECT concat(Fname, ' ', Lname) AS Full_Name FROM Persons;
54
55 # 3. Write an SQL query to count the number of unique country names from Persons table.
56
57 SELECT COUNT(DISTINCT Country_name) AS UniqueCountryName FROM Persons;
58
59 # 4. Write a query to print the maximum population from the Country table.
60
61 SELECT max(Population) AS MaximumPopulation FROM Country;
```

MaximumPopulation
138000000

## ➤ QUERY TO PRINT THE MINIMUM POPULATION FROM PERSONS TABLE.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd3s\_db\_2' selected. The main editor contains a SQL query with five tasks. The 'Result Grid' at the bottom shows the output of the fifth query, which is the minimum population from the 'Persons' table.

```
47 # 1. Write an SQL query to print the first three characters of Country_name from the Country table.
48
49 • SELECT LEFT(Country_name, 3) AS FirstThreeCharacters FROM Country;
50
51 # 2. Write an SQL query to concatenate first name and last name from Persons table.
52
53 • SELECT concat(Fname, ' ', Lname) AS Full_Name FROM Persons;
54
55 # 3. Write an SQL query to count the number of unique country names from Persons table.
56
57 • SELECT COUNT(DISTINCT Country_name) AS UniqueCountryName FROM Persons;
58
59 # 4. Write a query to print the maximum population from the Country table.
60
61 • SELECT max(Population) AS MaximumPopulation FROM Country;
62
63 # 5. Write a query to print the minimum population from Persons table.
64
65 • SELECT min(Population) AS MinimumPopulation FROM Persons;
```

MinimumPopulation
1000000

## ➤ INSERTED 2 NEW ROWS TO THE PERSONS TABLE MAKING THE LNAME NULL.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd3s\_db\_2' selected. The main editor contains a SQL query with five tasks. The 'Result Grid' at the bottom shows the output of the fifth query, which is the minimum population from the 'Persons' table.

```
63 # 5. Write a query to print the minimum population from Persons table.
64
65 • SELECT min(Population) AS MinimumPopulation FROM Persons;
66
67 # 6. Insert 2 new rows to the Persons table making the Lname NULL.
68 # Then write another query to count Lname from Persons table.
69
70 # Foreign key Constrains
71 • INSERT INTO Country (Id, Country_name, Population, Area) VALUES
72 (11, 'UAE', 331500000, 'Abu Dhabi'),
73 (12, 'Turkey', 1385000000, 'Istanbul');
74
75 • INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name) VALUES
76 (11, 'Janice', NULL, 1200000, 4.6, 11, 'UAE'),
77 (12, 'Richie', NULL, 2200000, 4.9, 12, 'Turkey');
78
79
80
81
82
```

Id	Fname	Lname	Population	Rating	Country_Id	Country_name
6	Anna	Moore	1300000	4.6	6	Germany
7	Robert	Taylor	1700000	3.7	7	France
8	Linda	Anderson	1400000	4.9	8	Japan
9	David	Thomas	2100000	3.8	9	Brazil
10	Sarah	Jackson	1600000	4.3	10	South Africa
11	Janice		1200000	4.6	11	UAE
12	Richie		2200000	4.9	12	Turkey

## ➤ QUERY TO COUNT LNAME FROM PERSONS TABLE.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd3s\_db\_2' selected. The main editor contains a SQL script with the following queries:

```
67 # 6. Insert 2 new rows to the Persons table making the Lname NULL.
68 # Then write another query to count Lname from Persons table.
69
70 # Foreign key Constrains
71 INSERT INTO Country (Id, Country_name, Population, Area) VALUES
72 (11, 'Janice', 331500000, 'Abu Dhabi'),
73 (12, 'Turkey', 1355000000, 'Istanbul');
74
75 INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name) VALUES
76 (11, 'Janice', NULL, 1200000, 4.6, 11, 'UAE'),
77 (12, 'Richie', NULL, 2200000, 4.9, 12, 'Turkey');
78
79 SELECT count(Lname) FROM Persons WHERE Lname IS NOT NULL;
```

The 'Result Grid' shows the output of the last query:

count(Lname)
10

The 'Action Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:34:46	INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name) VALUES (11, Janice...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.031 sec
2	18:35:00	SELECT * FROM Persons LIMIT 0.300	12 row(s) returned	0.000 sec / 0.000 sec
3	18:37:24	SELECT count(Lname) FROM Persons WHERE Lname IS NOT NULL LIMIT 0.300	1 row(s) returned	0.000 sec / 0.000 sec

## ➤ QUERY TO FIND THE NUMBER OF ROWS IN THE PERSONS TABLE

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'd3s\_db\_2' selected. The main editor contains a SQL script with the following queries:

```
71 INSERT INTO Country (Id, Country_name, Population, Area) VALUES
72 (11, 'Janice', 331500000, 'Abu Dhabi'),
73 (12, 'Turkey', 1355000000, 'Istanbul');
74
75 INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name) VALUES
76 (11, 'Janice', NULL, 1200000, 4.6, 11, 'UAE'),
77 (12, 'Richie', NULL, 2200000, 4.9, 12, 'Turkey');
78
79 SELECT count(Lname) FROM Persons WHERE Lname IS NOT NULL;
80
81 # 7. Write a query to find the number of rows in the Persons table.
82
83 SELECT COUNT(*) AS Rowcount FROM Persons;
```

The 'Result Grid' shows the output of the last query:

Rowcount
12

The 'Action Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:34:46	INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name) VALUES (11, Janice...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.031 sec
2	18:35:00	SELECT * FROM Persons LIMIT 0.300	12 row(s) returned	0.000 sec / 0.000 sec
3	18:37:24	SELECT count(Lname) FROM Persons WHERE Lname IS NOT NULL LIMIT 0.300	1 row(s) returned	0.000 sec / 0.000 sec
4	18:40:25	SELECT COUNT(*) AS Rowcount FROM Persons LIMIT 0.300	1 row(s) returned	0.000 sec / 0.000 sec



## ➤ QUERY TO SHOW THE POPULATION OF THE COUNTRY TABLE FOR THE FIRST 3 ROWS

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'country' selected under 'd36\_db\_2'. The main editor shows a SQL query with the following content:

```
72 (11, 'UAE', 331500000, 'Abu Dhabi');
73 (12, 'Turkey', 138000000, 'Istanbul');
74
75 • INSERT INTO Persons (Id, Fname, Lname, Population, Rating, Country_Id, Country_name) VALUES
76 (11, 'Janice', NULL, 1200000, 4.6, 11, 'UAE'),
77 (12, 'Richie', NULL, 2200000, 4.9, 12, 'Turkey');
78
79 • SELECT count(Lname) FROM Persons WHERE Lname IS NOT NULL;
80
81 # 7. Write a query to find the number of rows in the Persons table.
82
83 • SELECT COUNT(*) AS Rowcount FROM Persons;
84
85 # 8. Write an SQL query to show the population of the Country table for the first 3 rows. (Hint: Use LIMIT)
86
87 • SELECT Population FROM Country LIMIT 3;
88
89
90
```

The 'Result Grid' at the bottom shows the output of the query:

Population
331000000
138000000
38000000

## ➤ QUERY TO PRINT 3 RANDOM ROWS OF COUNTRIES.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'country' selected under 'd36\_db\_2'. The main editor shows a SQL query with the following content:

```
76 (11, 'Janice', NULL, 1200000, 4.6, 11, 'UAE'),
77 (12, 'Richie', NULL, 2200000, 4.9, 12, 'Turkey');
78
79 • SELECT count(Lname) FROM Persons WHERE Lname IS NOT NULL;
80
81 # 7. Write a query to find the number of rows in the Persons table.
82
83 • SELECT COUNT(*) AS Rowcount FROM Persons;
84
85 # 8. Write an SQL query to show the population of the Country table for the first 3 rows. (Hint: Use LIMIT)
86
87 • SELECT Population FROM Country LIMIT 3;
88
89 # 9. Write a query to print 3 random rows of countries. (Hint: Use rand() function and LIMIT)
90
91 • SELECT * FROM Country
92 ORDER BY rand() LIMIT 3;
93
94
95
```

The 'Result Grid' at the bottom shows the output of the query:

Id	Country_name	Population	Area
11	UAE	331500000	Abu Dhabi
8	Japan	125000000	Tokyo
3	Japan	380000000	Ontario

## ➤ ALL PERSONS ORDERED BY THEIR RATING IN DESCENDING ORDER

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema, including tables like country, departments, employee, and persons. The main editor window contains a SQL query: `SELECT Fname, Lname, Rating FROM Persons ORDER BY Rating DESC ;`. The bottom pane shows the result grid with 16 rows of data, ordered by rating in descending order.

Fname	Lname	Rating
Linda	Anderson	4.9
Richie	Moore	4.9
Jane	Smith	4.8
Anna	Moore	4.6
Janice	Moore	4.6
John	Doe	4.5
Sarah	Jackson	4.3
Michael	Brown	4.2
James	Wilson	4
Emily	Davis	3.9
David	Thomas	3.8
Robert	Taylor	3.7

## ➤ TOTAL POPULATION FOR EACH COUNTRY IN THE PERSONS TABLE

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema. The main editor window contains a SQL query: `SELECT Country_name, sum(Population) AS Total_Population FROM Persons GROUP BY Country_name;`. The bottom pane shows the result grid with 18 rows of data, grouped by country name.

Country_name	Total_Population
USA	2000000
India	2000000
Canada	1500000
Australia	2500000
UK	1800000
Germany	1300000
France	1700000
Japan	1400000
Brazil	2100000
South Africa	1600000
UAE	1200000
Turkey	2200000



## ➤ COUNTRIES IN THE PERSONS TABLE WITH A TOTAL POPULATION GREATER THAN 50,000

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'country', 'departments', 'employee', and 'persons'. The main editor window contains a SQL query: `SELECT Country_name, sum(Population) AS Total_Population FROM Persons GROUP BY Country_name;`. The 'Result Grid' at the bottom shows the results of the query, listing countries and their total populations. The results are as follows:

Country_Id	Country_name	Population
1	USA	3000000
2	India	2000000
3	Canada	1500000
4	Australia	2500000
5	UK	1800000
6	Germany	1300000
7	France	1700000
8	Japan	1400000
9	Brazil	2100000
10	South Africa	1600000
11	UAE	1200000
12	Turkey	2200000

## ➤ TOTAL NUMBER OF PERSONS AND AVERAGE RATING FOR EACH COUNTRY, COUNTRIES WITH MORE THAN 2 PERSONS, ORDERED BY THE AVERAGE RATING IN ASCENDING ORDER.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the database structure. The main editor window contains a SQL query: `SELECT Country_name, sum(Population) AS Total_Population FROM Persons GROUP BY Country_name;`. The 'Result Grid' at the bottom shows the results of the query, listing countries and their total populations. The results are as follows:

Country_Id	Country_name	Population
1	USA	3000000
2	India	2000000
3	Canada	1500000
4	Australia	2500000
5	UK	1800000
6	Germany	1300000
7	France	1700000
8	Japan	1400000
9	Brazil	2100000
10	South Africa	1600000
11	UAE	1200000
12	Turkey	2200000