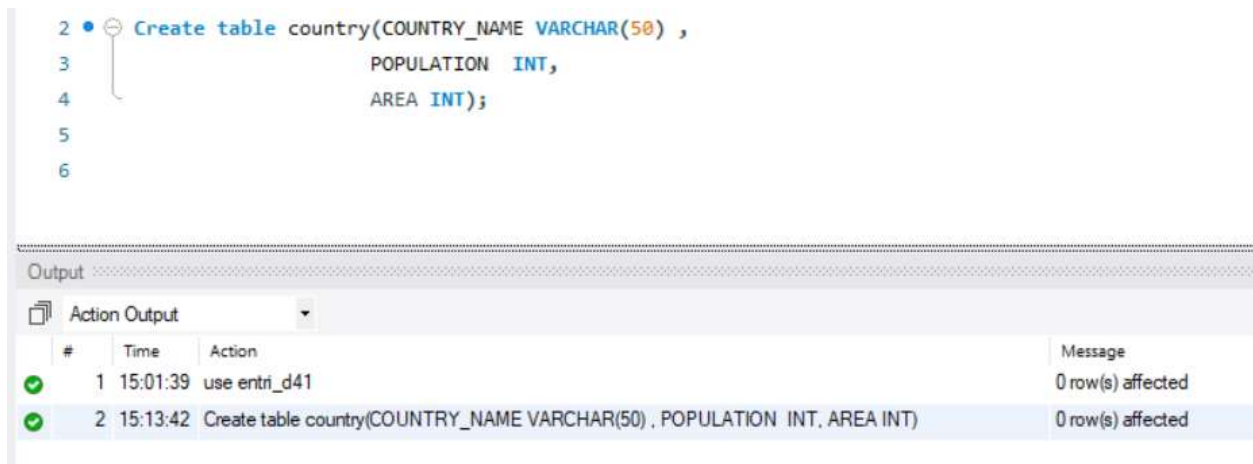


Create a table named Country with fields: Id Country\_name Population Area

```
CREATE TABLE COUNTRY(COUNTRY_NAME VARCHAR(50) ,  
                      POPULATION INT,  
                      AREA INT);
```



The screenshot shows a SQL IDE interface. The top pane displays the SQL code for creating a table named 'country'. The code is as follows:

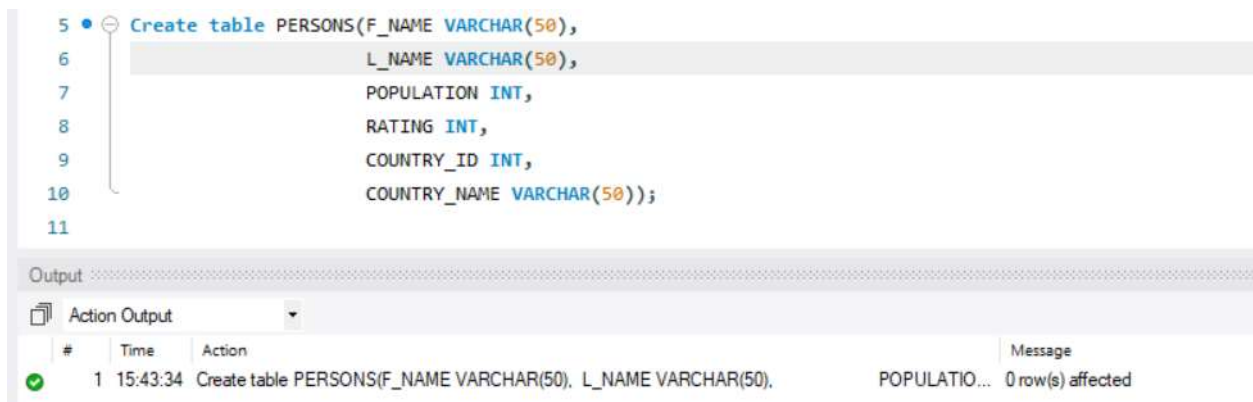
```
2 • Create table country(COUNTRY_NAME VARCHAR(50) ,  
3                          POPULATION INT,  
4                          AREA INT);  
5  
6
```

The bottom pane is titled 'Output' and shows the 'Action Output' tab. It contains a table with the following data:

| #   | Time     | Action  | Message           |
|-----|----------|---|-------------------|
| ✓ 1 | 15:01:39 | use entri_d41   | 0 row(s) affected |
| ✓ 2 | 15:13:42 | Create table country(COUNTRY_NAME VARCHAR(50) , POPULATION INT, AREA INT) | 0 row(s) affected |

Create another table named Persons with fields: Id Fname Lname Population Rating Country\_Id Country\_name

```
CREATE TABLE PERSONS(F_NAME VARCHAR(50),  
                      L_NAME VARCHAR(50),  
                      POPULATION INT,  
                      RATING INT,  
                      COUNTRY_ID INT,  
                      COUNTRY_NAME VARCHAR(50));
```



The screenshot displays a SQL editor with the following code:

```
5 • Create table PERSONS(F_NAME VARCHAR(50),  
6 L_NAME VARCHAR(50),  
7 POPULATION INT,  
8 RATING INT,  
9 COUNTRY_ID INT,  
10 COUNTRY_NAME VARCHAR(50));  
11
```

Below the editor is an "Output" section with a tab labeled "Action Output". It contains a table with the following data:

| #   | Time     | Action  | Message           |
|-----|----------|---|-------------------|
| ✓ 1 | 15:43:34 | Create table PERSONS(F_NAME VARCHAR(50), L_NAME VARCHAR(50), POPULATION INT, RATING INT, COUNTRY_ID INT, COUNTRY_NAME VARCHAR(50)); | 0 row(s) affected |

## INSERT INTO COUNTRY VALUES

('SINGAPORE',40000000,1000),('INDONESIA',100000000,2500),

INSERT INTO COUNTRY VALUES ('USA',6600000,4000),('ROME',200000,1000),  
('TANZANIA',5000000,1500),('KENYA',3000000,1000);

INSERT INTO COUNTRY VALUES ('UK',6500000,2000),('CANADA',450000000,5500),  
('AUSTRALIA',80000000,6500),('INDIA',500000000,3000);

```
12 * INSERT INTO COUNTRY VALUES('SINGAPORE',40000000,1000),('INDONESIA',100000000,2500);
13 * INSERT INTO COUNTRY VALUES ('USA',6600000,4000),('ROME',200000,1000),
14      ('TANZANIA',5000000,1500),('KENYA',3000000,1000);
15
16
17
18
19
```

| Output        |          |  |   |
|---------------|----------|--|---|
| Action Output |          |  |   |
| #             | Time     | Action   | Message   |
| ✓ 1           | 16:33:33 | INSERT INTO COUNTRY VALUES('USA',6600000,4000),('ROME',200000,1000), | (TA... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 |

| Output        |          |  |   |
|---------------|----------|--|---|
| Action Output |          |  |   |
| #             | Time     | Action   | Message   |
| ✓ 1           | 16:33:33 | INSERT INTO COUNTRY VALUES('USA',6600000,4000),('ROME',200000,1000),     | (TA... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 |
| ✓ 2           | 16:44:09 | INSERT INTO COUNTRY VALUES('UK',6500000,2000),('CANADA',450000000,5500), | ... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0    |

```

INSERT INTO PERSONS VALUES('KANNAN','RAVI',6600000,6,10001,'USA'),

('MANJU','NATH', 6600000,3,10001,''),

('SUKANYA','SURESH', 6500000,7,10002,'UK'),

('ANDREW','THOMPSON', 450000000,8,10003,'CANADA'),

('ANIL','CHOUDHARY', 500000000,4,10005,'INDIA'),

('SIMILY','MATHEW', 80000000,6,10004,'AUSTRALIA'),

('ABDUL','SALAM', 5000000,3,10225,'TANZANIA'),

('ROBERT','MOGAMBE', 5000000,4,10225,'TANZANIA'),

('SUKVINDER','SINGH', 6500000,6,10002,'UK');

```

```
19 • INSERT INTO PERSONS VALUE ('TINU','TOM',40000000,5,10010,'SINGAPORE');
```

```
20
```

Output



Action Output

| #   | Time     | Action  | Message                        |
|-----|----------|---|--------------------------------|
| ✖ 1 | 16:56:33 | Create table PERSONS(F_NAME VARCHAR(50), L_NAME VARCHAR(50).          | POPULATIO... Error Code: 1050. |
| ✔ 2 | 16:56:42 | INSERT INTO PERSONS VALUE ('TINU','TOM',40000000,5,10010,'SINGAPORE') | 1 row(s) affected              |

```

21 • INSERT INTO PERSONS VALUES('KANNAN','RAVI',6600000,6,10001,'USA'),
22 ('MANJU','NATH', 6600000,3,10001,''),
23 ('SUKANYA','SURESH', 6500000,7,10002,'UK'),
24 ('ANDREW','THOMPSON', 450000000,8,10003,'CANADA'),
25 ('ANIL','CHOUDHARY', 500000000,4,10005,'INDIA'),
26 ('SIMILY','MATHEW', 80000000,6,10004,'AUSTRALIA'),
27 ('ABDUL','SALAM', 5000000,3,10225,'TANZANIA'),
28 ('ROBERT','MOGAMBE', 5000000,4,10225,'TANZANIA'),
29 ('SUKVINDER','SINGH', 6500000,6,10002,'UK');
30

```

Output



Action Output

| #   | Time     | Action   | Message  |
|-----|----------|--|--|
| ✔ 1 | 17:40:40 | INSERT INTO PERSONS VALUES(KANNAN,'RAVI',6600000,6,10001,'USA'), (MANJU,'NATH', 660... | 5 row(s) affected Records: 9 Duplicates: 0 Warnings: 0 |

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

```
SELECT LEFT(COUNTRY_NAME, 3) AS COUNTRYCODE  
FROM COUNTRY;
```

70 • `SELECT LEFT(COUNTRY_NAME, 3) AS COUNTRYCODE`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

|   | COUNTRYCODE |
|---|-------------|
| ▶ | SIN         |
|   | IND         |
|   | USA         |
|   | ROM         |
|   | TAN         |
|   | KEN         |
|   | UK          |
|   | CAN         |
|   | AUS         |
|   | IND         |

Result 46 x

Output

Action Output

| #   | Time     | Action   | Message            |
|-----|----------|--|--------------------|
| ✓ 1 | 22:52:18 | SELECT LEFT(COUNTRY_NAME, 3) AS COUNTRYCODE FROM COUNTRY LIMIT 0, 1000 | 10 row(s) returned |

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

```
SELECT CONCAT(F_NAME , L_NAME)  
AS FULLNAME FROM PERSONS;
```

The screenshot displays a SQL IDE interface. At the top, a query editor shows the following SQL statement:

```
74 • SELECT CONCAT(F_NAME , L_NAME)  
75 AS FULLNAME FROM PERSONS;
```

Below the editor, the 'Result Grid' tab is active, showing a table with one column, 'FULLNAME', and ten rows of concatenated names:

| FULLNAME       |
|----------------|
| TINUTOM        |
| KANNANRAVI     |
| MANJUNATH      |
| SUKANYASURESH  |
| ANDREWTHOMPSON |
| ANILCHOUDHARY  |
| SIMILYMATHEW   |
| ABDULSALAM     |
| ROBERTMOGAMBE  |
| SUKVINDERSINGH |

At the bottom, the 'Output' tab is active, showing the 'Action Output' section. It contains a table with the following data:

| #   | Time     | Action  | Message            |
|-----|----------|---|--------------------|
| ✓ 1 | 22:56:32 | SELECT CONCAT(F_NAME , L_NAME) AS FULLNAME FROM PERSONS LIMIT 0, 1000 | 10 row(s) returned |

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

```
SELECT DISTINCT COUNT(COUNTRY_NAME),COUNTRY_NAME FROM PERSONS  
GROUP BY COUNTRY_NAME;
```

85 • `SELECT DISTINCT COUNT(COUNTRY_NAME) FROM PERSONS`  
86 `GROUP BY COUNTRY_NAME;`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

|     | COUNT(COUNTRY_NAME) |
|-----|---------------------|
| ▶ 1 |                     |
| 2   |                     |
| 4   |                     |

Result 89 ×

Output

Action Output

| # | Time     | Action   | Message           |
|---|----------|--|-------------------|
| 1 | 17:04:30 | SELECT DISTINCT COUNT(COUNTRY_NAME) FROM PERSONS GROUP BY COUNTRY_NAME ... | 3 row(s) returned |

85 • `SELECT DISTINCT COUNT(COUNTRY_NAME),COUNTRY_NAME FROM PERSONS`  
86 `GROUP BY COUNTRY_NAME;`  
87

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

|     | COUNT(COUNTRY_NAME) | COUNTRY_NAME |
|-----|---------------------|--------------|
| ▶ 1 |                     | SINGAPORE    |
| 4   |                     | USA          |
| 2   |                     | UK           |
| 1   |                     | CANADA       |
| 1   |                     | INDIA        |
| 1   |                     | AUSTRALIA    |
| 2   |                     | TANZANIA     |

Result 94 ×

Output

Action Output

| # | Time     | Action   | Message           |
|---|----------|--|-------------------|
| 1 | 17:16:06 | SELECT DISTINCT COUNT(COUNTRY_NAME),COUNTRY_NAME FROM PERSONS GROUP BY ... | 7 row(s) returned |

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

**SELECT MAX(POPULATION) FROM PERSONS;**

The screenshot displays a database query interface. At the top, a query editor shows the SQL statement: `SELECT MAX(POPULATION) FROM PERSONS;`. Below the editor, a toolbar includes options for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The 'Result Grid' is active, showing a single row with the column header 'MAX(POPULATION)' and the value '500000000'. At the bottom, a 'Result 71' tab is open, displaying an 'Output' section. This section shows a table with columns for '#', 'Time', 'Action', and 'Message'. A single entry is present, indicating the query was executed successfully at 23:43:30, returning 1 row(s).

| # | Time     | Action  | Message           |
|---|----------|---|-------------------|
| 1 | 23:43:30 | SELECT MAX(POPULATION) FROM PERSONS LIMIT 0, 1000 | 1 row(s) returned |



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

**SELECT MIN(POPULATION) FROM PERSONS;**

The screenshot displays a database query interface. At the top, a query editor shows the SQL statement: `SELECT MIN(POPULATION) FROM PERSONS;`. Below the editor, a toolbar includes options for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The 'Result Grid' is active, showing a single column header 'MIN(POPULATION)' and one data row with the value '5000000'. At the bottom, a 'Result 76' tab is open, displaying an 'Output' section. Below the output, an 'Action Output' table lists the execution details:

| # | Time     | Action  | Message           |
|---|----------|---|-------------------|
| 1 | 23:45:54 | SELECT MIN(POPULATION) FROM PERSONS LIMIT 0, 1000 | 1 row(s) returned |

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 VALUES('JALEEL','NULL',6600000,7,10001,'USA'),  
('MANOJ','NULL',6600000,8,10001,'USA');
```

```
91 * INSERT INTO PERSONS VALUES('JALEEL','NULL',6600000,7,10001,'USA'),  
92 ('MANOJ','NULL',6600000,8,10001,'USA');  
93  
94  
95  
96  
97
```

Output

Action Output





| # | Time     | Action   | Message  |
|---|----------|--|--|
| 1 | 23:38:40 | INSERT INTO PERSONS VALUES('JALEEL','NULL',6600000,7,10001,'USA'), ('MANOJ','NULL',6600000,8,10001,'USA'); | 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 |

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

**SELECT COUNT(\*) FROM PERSONS;**

84 • **SELECT COUNT(\*) FROM PERSONS;**


---

**Result Grid** |  |  Filter Rows:  | Export:  | Wrap Cell Content: 

|   | COUNT(*) |
|---|----------|
| ▶ | 12       |

Result 97 x

Output

 Action Output ▼

| #   | Time     | Action                                     |
|-----|----------|--|
| ✓ 1 | 17:29:00 | SELECT COUNT(*) FROM PERSONS LIMIT 0, 1000 |

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;**

96 • **SELECT POPULATION FROM COUNTRY**

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

|   | POPULATION |
|---|------------|
| ▶ | 40000000   |
|   | 100000000  |
|   | 6600000    |

COUNTRY 9 x

Output

Action Output

| #    | Time     | Action                                 | Message           |
|------|----------|--|-------------------|
| ✓ 11 | 11:21:16 | SELECT POPULATION FROM COUNTRY LIMIT 3 | 3 row(s) returned |

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 query editor with the following SQL query:

```
99 • SELECT * FROM COUNTRY
100 ORDER BY RAND()
101 LIMIT 3;
102
```

Below the query editor is a toolbar with options: Result Grid, Filter Rows, Export, Wrap Cell Content, and Fetch rows. The Result Grid displays the following data:

|   | COUNTRY_NAME | POPULATION | AREA |
|---|--------------|------------|------|
| ▶ | KENYA        | 3000000    | 1000 |
|   | INDIA        | 500000000  | 3000 |
|   | SINGAPORE    | 4000000    | 1000 |

Below the result grid is a tab labeled "COUNTRY 14" with a close button. The Output section shows the Action Output for the query:

| #    | Time     | Action  | Message           |
|------|----------|---|-------------------|
| ✓ 25 | 11:35:16 | SELECT * FROM COUNTRY ORDER BY RAND() LIMIT 3 | 3 row(s) returned |

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

```
SELECT * FROM PERSONS  
ORDER BY RATING DESC;
```

106 • SELECT \* FROM PERSONS  
107 ORDER BY RATING DESC;

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

|   | F_NAME    | L_NAME    | POPULATION | RATING | COUNTRY_ID | COUNTRY_NAME |
|---|-----------|-----------|------------|--------|------------|--------------|
| ▶ | ANDREW    | THOMPSON  | 450000000  | 8      | 10003      | CANADA       |
|   | MANOJ     | NULL      | 6600000    | 8      | 10001      | USA          |
|   | SUKANYA   | SURESH    | 6500000    | 7      | 10002      | UK           |
|   | JALEEL    | NULL      | 6600000    | 7      | 10001      | USA          |
|   | KANNAN    | RAVI      | 6600000    | 6      | 10001      | USA          |
|   | SIMILY    | MATHEW    | 80000000   | 6      | 10004      | AUSTRALIA    |
|   | SUKVINDER | SINGH     | 6500000    | 6      | 10002      | UK           |
|   | TINUJ     | TOM       | 40000000   | 5      | 10010      | SINGAPORE    |
|   | ANIL      | CHOUDHARY | 500000000  | 4      | 10005      | INDIA        |
|   | ROBERT    | MOGAMBE   | 5000000    | 4      | 10225      | TANZANIA     |
|   | MANJU     | NATH      | 6600000    | 3      | 10001      |              |
|   | ABDUL     | SALAM     | 5000000    | 3      | 10225      | TANZANIA     |

PERSONS 17 x

Output

Action Output

| #   | Time     | Action   | Message            |
|-----|----------|--|--------------------|
| ✓ 1 | 12:21:50 | SELECT * FROM PERSONS ORDER BY RATING DESC LIMIT 0, 1000 | 12 row(s) returned |

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

**SELECT SUM(POPULATION) FROM PERSONS;**

109 • **SELECT SUM(POPULATION) FROM PERSONS;**  
110

---

**Result Grid** | | Filter Rows:  | Export: | Wrap Cell Content:

|   | SUM(POPULATION) |
|---|-----------------|
| ▶ | 1119400000      |

Result 23 x

Output

Action Output ▼

| #   | Time     | Action  | Message           |
|-----|----------|---|-------------------|
| ✓ 1 | 12:33:36 | SELECT SUM(POPULATION) FROM PERSONS LIMIT 0, 1000 | 1 row(s) returned |

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

```
SELECT COUNTRY_NAME ,POPULATION FROM PERSONS  
WHERE POPULATION >50000  
GROUP BY COUNTRY_NAME,POPULATION;
```

105 • **SELECT COUNTRY\_NAME ,POPULATION FROM PERSONS**  
106 **WHERE POPULATION >50000**  
107 **GROUP BY COUNTRY\_NAME,POPULATION;**  
108  
109  
110

Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

|   | COUNTRY_NAME | POPULATION |
|---|--------------|------------|
| ▶ | SINGAPORE    | 40000000   |
|   | USA          | 66000000   |
|   | UK           | 65000000   |
|   | CANADA       | 450000000  |
|   | INDIA        | 500000000  |
|   | AUSTRALIA    | 80000000   |
|   | TANZANIA     | 5000000    |

PERSONS 105 ×

Output

Action Output ▾

| #   | Time     | Action  | Message           |
|-----|----------|---|-------------------|
| ✓ 1 | 17:37:04 | SELECT COUNTRY_NAME ,POPULATION FROM PERSONS WHERE POPULATION >50000 GRO... | 7 row(s) returned |



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 COUNT(*), AVG(RATING), COUNTRY_NAME
FROM PERSONS
GROUP BY COUNTRY_NAME
HAVING COUNT(*) >= 2
```

118 • SELECT COUNT(\*), AVG(RATING), COUNTRY\_NAME  
119 FROM PERSONS  
120 GROUP BY COUNTRY\_NAME  
121 Having COUNT(\*) >= 2

Result Grid Filter Rows:  Export: Wrap Cell Content:

|   | COUNT(*) | AVG(RATING) | COUNTRY_NAME |
|---|----------|-------------|--------------|
| ▶ | 2        | 3.5000      | TANZANIA     |
|   | 4        | 6.0000      | USA          |
|   | 2        | 6.5000      | UK           |

Result 74 x

Output

Action Output ▼

| #   | Time     | Action   | Message           |
|-----|----------|--|-------------------|
| ✓ 1 | 16:10:28 | SELECT COUNT(*), AVG(RATING), COUNTRY_NAME FROM PERSONS GROUP BY COUNTRY_... | 3 row(s) returned |