

## 1. The maximum and minimum of population and population growth in all the countries

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'project1' expanded, showing tables 'facts' and 'sqlite\_sequence'. The main editor window contains the following SQL query:

```
1 SELECT MIN(population) AS min_pop,  
2        MAX(population) AS max_pop,  
3        MIN(population_growth) AS min_pop_growth,  
4        MAX(population_growth) AS max_pop_growth  
5 FROM project1.facts;
```

The 'Result Grid' at the bottom shows the query results:

min_pop	max_pop	min_pop_growth	max_pop_growth
5241	1367485388	0.01	3.32

## 2. Fetching all records from the 'facts' table where the population is the smallest among all the populations in that table.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'project1' expanded, showing tables 'facts' and 'sqlite\_sequence'. The main editor window contains the following SQL query:

```
1 SELECT * FROM project1.facts  
2 WHERE population = (SELECT MIN(population) FROM project1.facts);
```

The 'Result Grid' at the bottom shows the query results:

code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate
mh	Montserrat	102	102	0	5241	0.5	11.26	6.3	0

3. Fetching all records from the 'facts' table where the population is the biggest among all the populations in that table.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'project1' selected. The main editor window contains the following SQL query:

```
1 SELECT * FROM project1.facts
2 WHERE population = (SELECT MAX(population) FROM project1.facts);
```

The 'Result Grid' at the bottom shows the query results for the 'facts' table. The first row is highlighted.

code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migration_rate
ch	China	9596960	9326410	270550	1367485388	0.45	12.49	7.53	0.44

4. The average population and average area based on the data present in the facts table.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'project1' selected. The main editor window contains the following SQL query:

```
1 SELECT AVG(population) AS avg_population, AVG(area) AS avg_area
2 FROM project1.facts;
3
```

The 'Result Grid' at the bottom shows the query results. The first row is highlighted.

avg_population	avg_area
32890977.5349	592671.9070

5. Retrieving the names, populations, and areas of countries where the population is above the average population and the area is below the average area, and the results are sorted by population in descending order.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 SELECT name, population, area
2 FROM project1.facts
3 WHERE population > (SELECT AVG(population)
4 FROM project1.facts)
5 AND area < (SELECT AVG(area)
6 FROM project1.facts)
7 ORDER BY 2 DESC;
```

The Results Grid displays the following data:

name	population	area
Bangladesh	168957745	148460
Japan	126919659	377915
Philippines	100998376	300000
Vietnam	94348835	331210
Germany	80854408	357022
Thailand	67976405	513120
United Kingdom	64088222	243610
Italy	61855120	301340
Korea, South	49115196	99720
Spain	48146134	505370
Kenya	45925301	580367
Poland	38562189	312685
Uganda	37101745	241038

6. Retrieving information (name, population, area) about countries with populations higher than the average population across all countries in the "facts" table.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 SELECT name, population, area
2 FROM project1.facts
3 WHERE population > (SELECT AVG(population)
4 FROM project1.facts)
5
```

The Results Grid displays the following data:

name	population	area
Bangladesh	168957745	148460
Japan	126919659	377915
Philippines	100998376	300000
Vietnam	94348835	331210
Germany	80854408	357022
Thailand	67976405	513120
United Kingdom	64088222	243610
Italy	61855120	301340
Korea, South	49115196	99720
Spain	48146134	505370
Kenya	45925301	580367
Poland	38562189	312685
Uganda	37101745	241038
Iraq	37056169	438317
Moreocco	33322699	446550

7. **Selecting entities with above-average populations and below-average areas from the "facts" table and sorting them by population in descending order**

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT name, population, area
2 FROM project1.facts
3 WHERE population > (SELECT AVG(population)
4 FROM project1.facts)
5 AND area < (SELECT AVG(area)
6 FROM project1.facts)
7 ORDER BY population DESC;
```

The result grid displays the following data:

name	population	area
Bangladesh	168957745	148460
Japan	126919659	377915
Philippines	100998376	300000
Vietnam	94348835	331210
Germany	80854408	357022
Thailand	67976405	513120
United Kingdom	64088222	243610
Italy	61855120	301340
Korea, South	49115196	99720
Spain	48146134	505370
Kenya	45925301	580367
Poland	38562189	312685

8. **Retrieving information (name, population, area) from the facts table, but only for those entities whose population is above the average population of all entities**

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
1 SELECT name, population, area
2 FROM project1.facts
3 WHERE population > (SELECT AVG(population)
4 FROM project1.facts)
5
```

The result grid displays the following data:

name	population	area
Bangladesh	168957745	148460
Japan	126919659	377915
Philippines	100998376	300000
Vietnam	94348835	331210
Germany	80854408	357022
Thailand	67976405	513120
United Kingdom	64088222	243610
Italy	61855120	301340
Korea, South	49115196	99720
Spain	48146134	505370
Kenya	45925301	580367
Poland	38562189	312685
Uganda	37101745	241038
Iraq	37056169	438317
Morocco	33322699	446550

9. Retrieving the names and birth rates of the first 10 records from the 'facts' table ordered by 2<sup>nd</sup> column in descending order.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: project1.facts project1.sqlite\_sequence project1.sqlite\_sequence project1.facts project1.facts project1.facts facts facts

SCHEMAS

Filter objects

project1

- Tables
  - facts
  - Columns
  - Indexes
  - Foreign Keys
  - Triggers
  - sqlite\_sequence
  - Views
  - Stored Procedures
  - Functions

Administration Schemas

Information

Schema: project1

```
1 • SELECT name, birth_rate
2 FROM project1.facts
3 ORDER by 2 DESC
4 Limit 10;
```

Result Grid

name	birth_rate
Mali	44.99
Uganda	43.79
Zambia	42.13
Burkina Faso	42.03
Burundi	42.01
Malawi	41.56
Somalia	40.45
Angola	38.78
Mozambique	38.58
Afghanistan	38.57

Result Grid

Form Editor

Field Types

10. Retrieving the names and birth rates of the first 10 records from the 'facts' table ordered by birth rate in ascending order.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: project1.facts project1.sqlite\_sequence project1.sqlite\_sequence project1.facts project1.facts project1.facts facts facts

SCHEMAS

Filter objects

project1

- Tables
  - facts
  - Columns
  - Indexes
  - Foreign Keys
  - Triggers
  - sqlite\_sequence
  - Views
  - Stored Procedures
  - Functions

Administration Schemas

Information

Schema: project1

```
1 • SELECT name, birth_rate
2 FROM project1.facts
3 ORDER by birth_rate
4 Limit 10;
```

Result Grid

name	birth_rate
Monaco	6.65
Saint Pierre and Miquelon	7.42
Japan	7.93
Andorra	8.13
Korea, South	8.19
Singapore	8.27
Slovenia	8.42
Germany	8.47
Taiwan	8.47
San Marino	8.63

Result Grid

Form Editor

Field Types