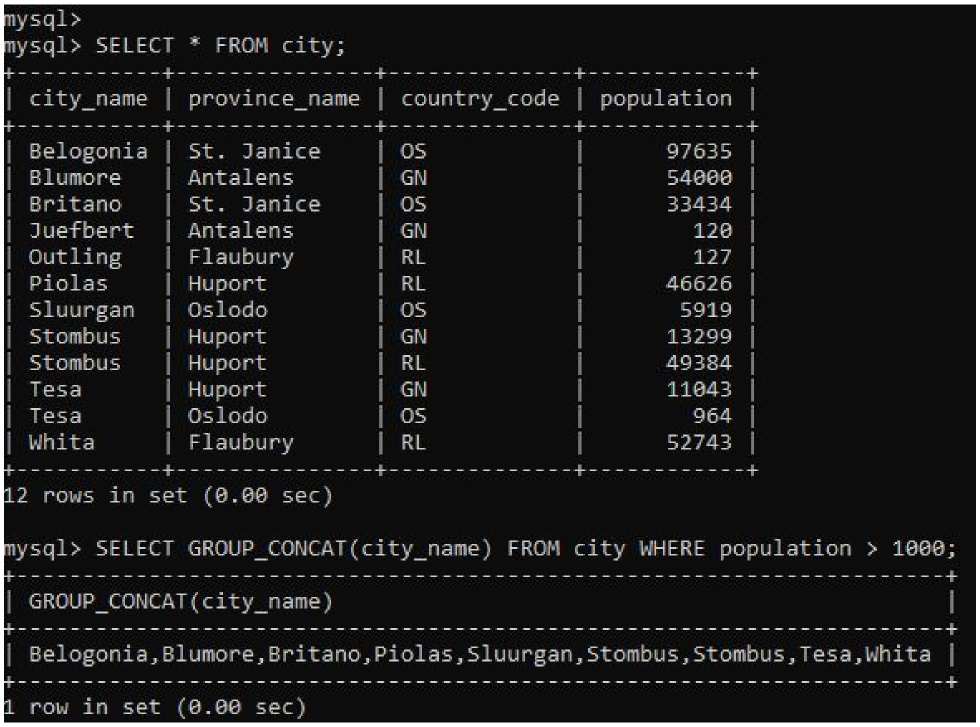
Kevin Hance
DBMS (CPSC 321)
10/17/2019
HW6

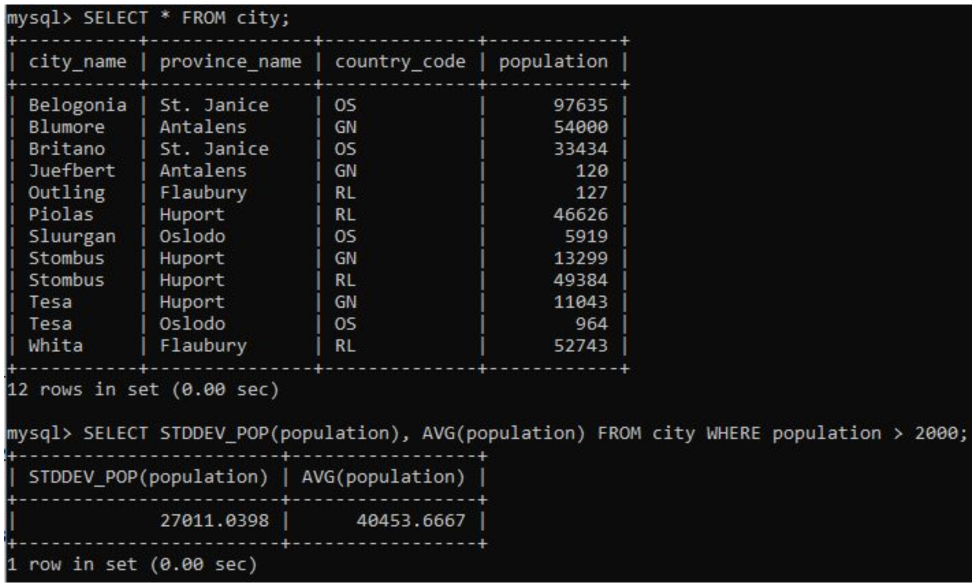
READING ASSIGNMENT:

1. 

```
mysql>
mysql> SELECT * FROM city;
+-----------+---------------+--------------+------------+
| city_name | province_name | country_code | population |
+-----------+---------------+--------------+------------+
| Belogonia | St. Janice    | OS           |      97635 |
| Blumore   | Antalens      | GN           |      54000 |
| Britano   | St. Janice    | OS           |      33434 |
| Juefbert  | Antalens      | GN           |        120 |
| Outling   | Flaubury      | RL           |        127 |
| Piolas    | Huport        | RL           |      46626 |
| Sluurgan  | Oslodo        | OS           |       5919 |
| Stombus   | Huport        | GN           |      13299 |
| Stombus   | Huport        | RL           |      49384 |
| Tesa      | Huport        | GN           |      11043 |
| Tesa      | Oslodo        | OS           |        964 |
| Whita     | Flaubury      | RL           |      52743 |
+-----------+---------------+--------------+------------+
12 rows in set (0.00 sec)

mysql> SELECT GROUP_CONCAT(city_name) FROM city WHERE population > 1000;
+-------------------------------------------------------------------+
| GROUP_CONCAT(city_name)                                           |
+-------------------------------------------------------------------+
| Belogonia,Blumore,Britano,Piolas,Sluurgan,Stombus,Stombus,Tesa,Whita |
+-------------------------------------------------------------------+
1 row in set (0.00 sec)
```

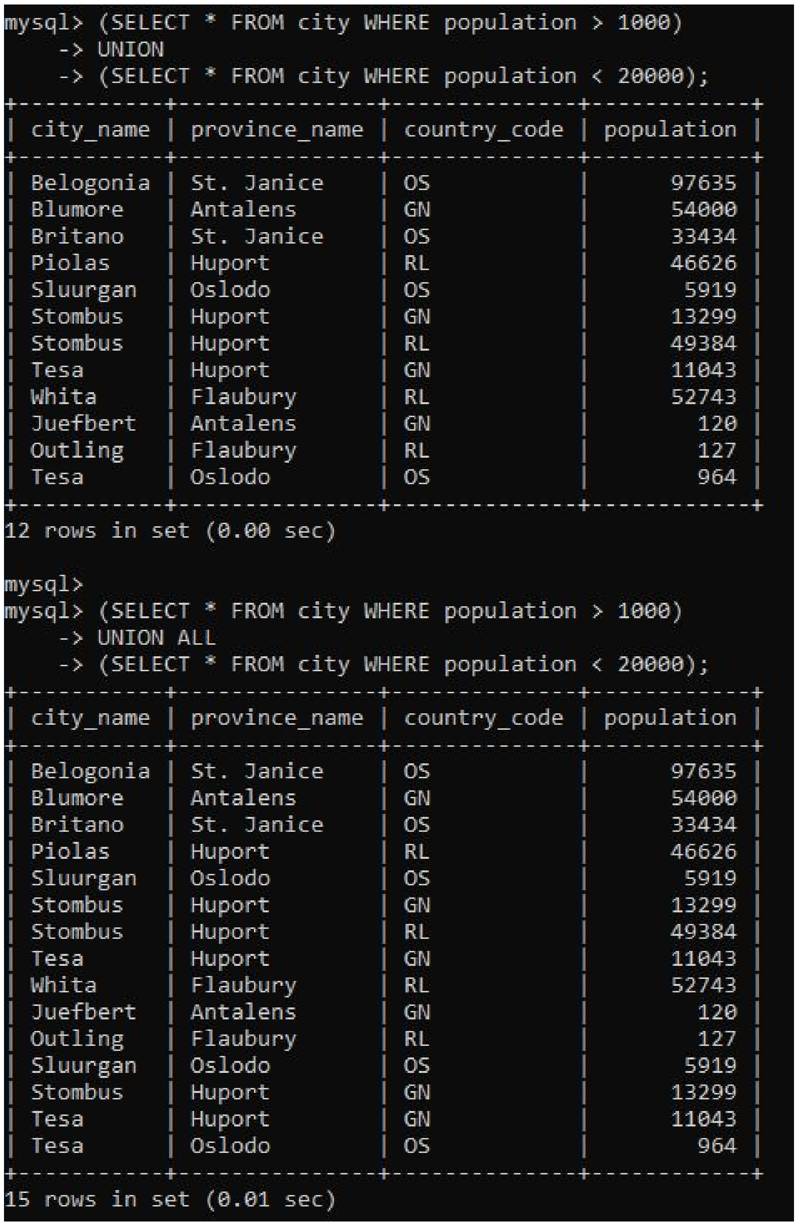
```
SELECT * FROM city;
SELECT GROUP_CONCAT(city_name) FROM city WHERE population > 1000;
```

2. 

```
mysql> SELECT * FROM city;
+-----------+---------------+--------------+------------+
| city_name | province_name | country_code | population |
+-----------+---------------+--------------+------------+
| Belogonia | St. Janice    | OS           |      97635 |
| Blumore   | Antalens      | GN           |      54000 |
| Britano   | St. Janice    | OS           |      33434 |
| Juefbert  | Antalens      | GN           |        120 |
| Outling   | Flaubury      | RL           |        127 |
| Piolas    | Huport        | RL           |      46626 |
| Sluurgan  | Oslodo        | OS           |       5919 |
| Stombus   | Huport        | GN           |      13299 |
| Stombus   | Huport        | RL           |      49384 |
| Tesa      | Huport        | GN           |      11043 |
| Tesa      | Oslodo        | OS           |        964 |
| Whita     | Flaubury      | RL           |      52743 |
+-----------+---------------+--------------+------------+
12 rows in set (0.00 sec)

mysql> SELECT STDDEV_POP(population), AVG(population) FROM city WHERE population > 2000;
+------------------------+-----------------+
| STDDEV_POP(population) | AVG(population) |
+------------------------+-----------------+
|             27011.0398 |      40453.6667 |
+------------------------+-----------------+
1 row in set (0.00 sec)
```

```
SELECT * FROM city;
SELECT STDDEV_POP(population), AVG(population) FROM city WHERE population > 2000;
```



```
mysql> (SELECT * FROM city WHERE population > 1000)
    -> UNION
    -> (SELECT * FROM city WHERE population < 20000);
+-----------+---------------+--------------+------------+
| city_name | province_name | country_code | population |
+-----------+---------------+--------------+------------+
| Belogonia | St. Janice    | OS           |      97635 |
| Blumore   | Antalens      | GN           |      54000 |
| Britano   | St. Janice    | OS           |      33434 |
| Piolas    | Huport        | RL           |      46626 |
| Sluurgan  | Oslodo        | OS           |       5919 |
| Stombus   | Huport        | GN           |      13299 |
| Stombus   | Huport        | RL           |      49384 |
| Tesa      | Huport        | GN           |      11043 |
| Whita     | Flaubury      | RL           |      52743 |
| Juefbert  | Antalens      | GN           |        120 |
| Outling   | Flaubury      | RL           |        127 |
| Tesa      | Oslodo        | OS           |        964 |
+-----------+---------------+--------------+------------+
12 rows in set (0.00 sec)

mysql>
mysql> (SELECT * FROM city WHERE population > 1000)
    -> UNION ALL
    -> (SELECT * FROM city WHERE population < 20000);
+-----------+---------------+--------------+------------+
| city_name | province_name | country_code | population |
+-----------+---------------+--------------+------------+
| Belogonia | St. Janice    | OS           |      97635 |
| Blumore   | Antalens      | GN           |      54000 |
| Britano   | St. Janice    | OS           |      33434 |
| Piolas    | Huport        | RL           |      46626 |
| Sluurgan  | Oslodo        | OS           |       5919 |
| Stombus   | Huport        | GN           |      13299 |
| Stombus   | Huport        | RL           |      49384 |
| Tesa      | Huport        | GN           |      11043 |
| Whita     | Flaubury      | RL           |      52743 |
| Juefbert  | Antalens      | GN           |        120 |
| Outling   | Flaubury      | RL           |        127 |
| Sluurgan  | Oslodo        | OS           |       5919 |
| Stombus   | Huport        | GN           |      13299 |
| Tesa      | Huport        | GN           |      11043 |
| Tesa      | Oslodo        | OS           |        964 |
+-----------+---------------+--------------+------------+
15 rows in set (0.01 sec)
```

3.

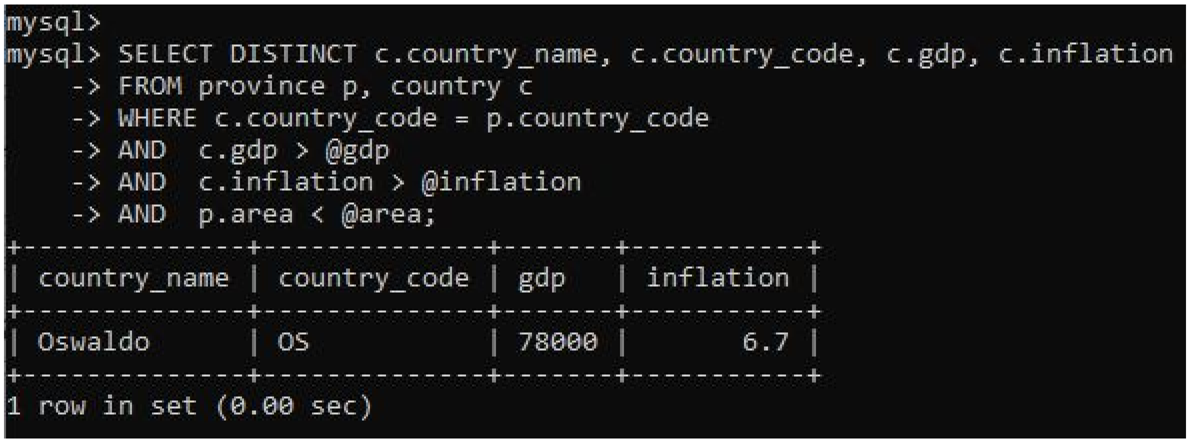
```
SELECT * FROM city;
(SELECT * FROM city WHERE population > 1000)
UNION
(SELECT * FROM city WHERE population < 20000);

(SELECT * FROM city WHERE population > 1000)
UNION ALL
(SELECT * FROM city WHERE population < 20000);
```

4. The main difference between INTERSECT and EXCEPT is that, if they were put into a Venn diagram, INTERSECT is the middle part where the circles overlap, and EXCEPT is the outer part where the circles do not overlap. INTERSECT could be used to find cities that have a population greater than x and less than y, where x < y, and EXCEPT could be used to find cities that have a population less than x and greater than y, where x < y.

TECHNICAL WORK:

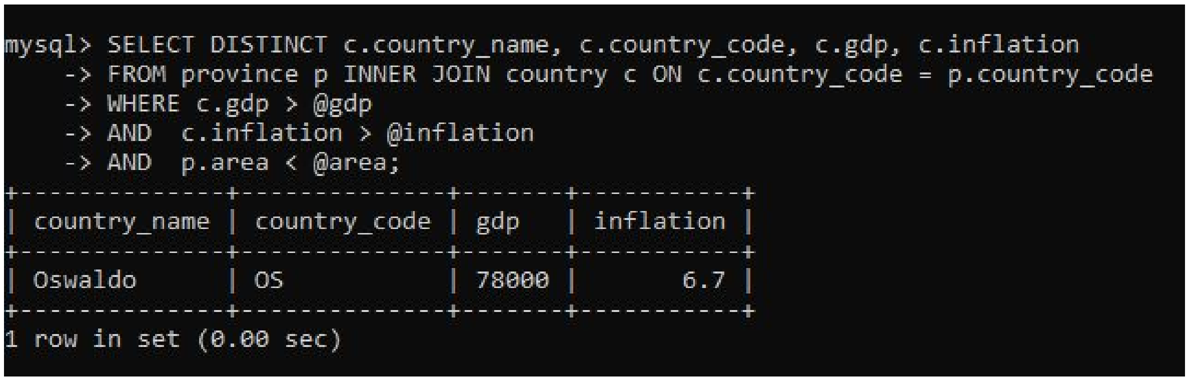
1. <insert images of “SELECT * FROM <each table>”



```
mysql>
mysql> SELECT DISTINCT c.country_name, c.country_code, c.gdp, c.inflation
    -> FROM province p, country c
    -> WHERE c.country_code = p.country_code
    -> AND  c.gdp > @gdp
    -> AND  c.inflation > @inflation
    -> AND  p.area < @area;
+--------------+--------------+-------+-----------+
| country_name | country_code | gdp   | inflation |
+--------------+--------------+-------+-----------+
| Oswaldo      | OS           | 78000 |       6.7 |
+--------------+--------------+-------+-----------+
1 row in set (0.00 sec)
```

2.

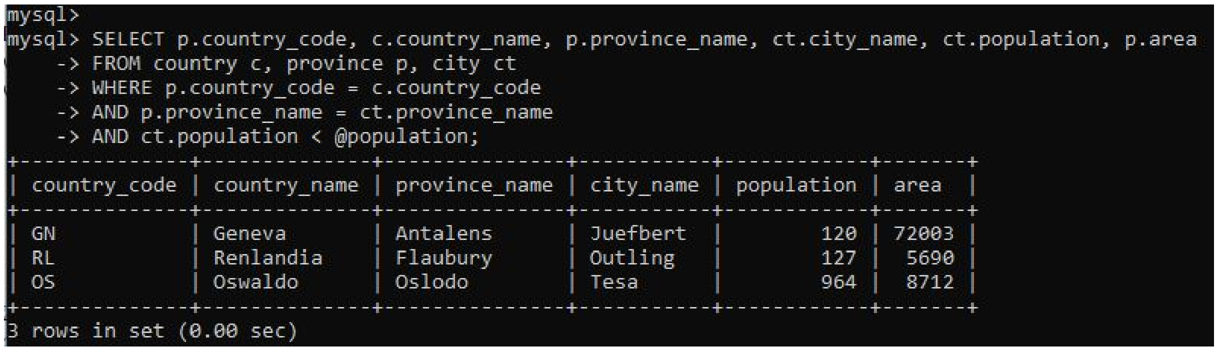
```
SET @gdp = 60000;
SET @inflation = 3.0;
SET @area = 10000;
SELECT DISTINCT c.country_name, c.country_code, c.gdp, c.inflation
FROM province p, country c
WHERE c.country_code = p.country_code
AND  c.gdp > @gdp
AND  c.inflation > @inflation
AND  p.area < @area;
```



```
mysql> SELECT DISTINCT c.country_name, c.country_code, c.gdp, c.inflation
    -> FROM province p INNER JOIN country c ON c.country_code = p.country_code
    -> WHERE c.gdp > @gdp
    -> AND  c.inflation > @inflation
    -> AND  p.area < @area;
+--------------+--------------+-------+-----------+
| country_name | country_code | gdp   | inflation |
+--------------+--------------+-------+-----------+
| Oswaldo      | OS           | 78000 |       6.7 |
+--------------+--------------+-------+-----------+
1 row in set (0.00 sec)
```

3.

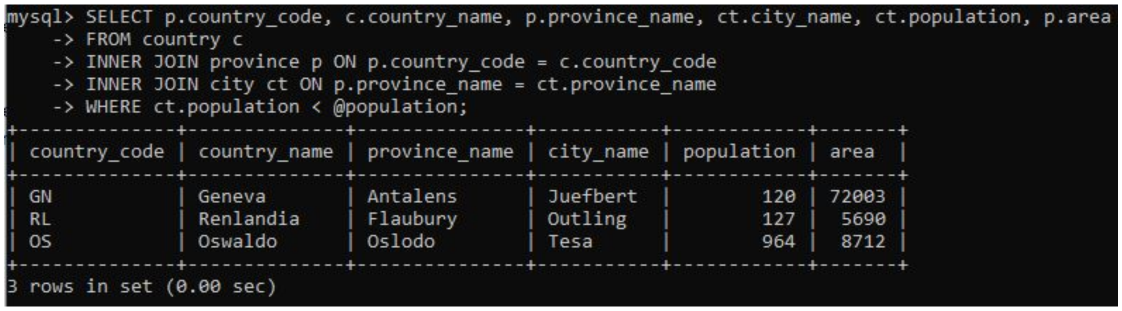
```
SET @gdp = 60000;
SET @inflation = 3.0;
SET @area = 10000;
SELECT DISTINCT c.country_name, c.country_code, c.gdp, c.inflation
FROM province p INNER JOIN country c ON c.country_code = p.country_code
WHERE c.gdp > @gdp
AND  c.inflation > @inflation
AND  p.area < @area;
```



```
mysql>
mysql> SELECT p.country_code, c.country_name, p.province_name, ct.city_name, ct.population, p.area
    -> FROM country c, province p, city ct
    -> WHERE p.country_code = c.country_code
    -> AND p.province_name = ct.province_name
    -> AND ct.population < @population;
+--------------+--------------+---------------+-----------+------------+-------+
| country_code | country_name | province_name | city_name | population | area  |
+--------------+--------------+---------------+-----------+------------+-------+
| GN           | Geneva       | Antalens      | Juefbert  |        120 | 72003 |
| RL           | Renlandia    | Flaubury      | Outling   |        127 |  5690 |
| OS           | Oswaldo      | Oslodo        | Tesa      |        964 |  8712 |
+--------------+--------------+---------------+-----------+------------+-------+
3 rows in set (0.00 sec)
```

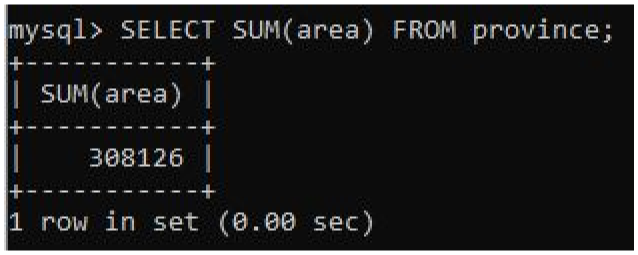
4.

```
SET @population = 1000;
SELECT p.country_code, c.country_name, p.province_name, ct.city_name,
ct.population, p.area
FROM country c, province p, city ct
WHERE p.country_code = c.country_code
AND p.province_name = ct.province_name
AND ct.population < @population;
```

5. 

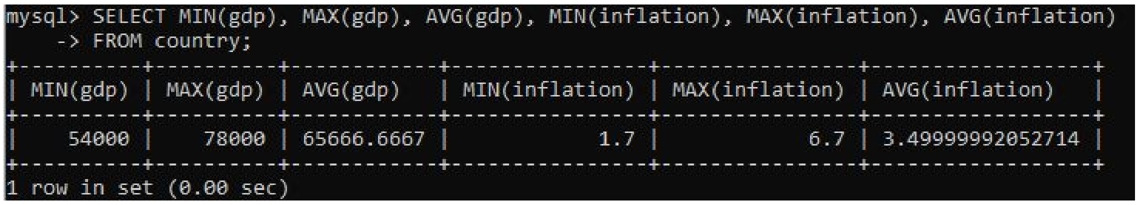
```
mysql> SELECT p.country_code, c.country_name, p.province_name, ct.city_name, ct.population, p.area
    -> FROM country c
    -> INNER JOIN province p ON p.country_code = c.country_code
    -> INNER JOIN city ct ON p.province_name = ct.province_name
    -> WHERE ct.population < @population;
+--------------+--------------+---------------+-----------+------------+-------+
| country_code | country_name | province_name | city_name | population | area  |
+--------------+--------------+---------------+-----------+------------+-------+
| GN           | Geneva       | Antalens      | Juefbert  |        120 | 72003 |
| RL           | Renlandia    | Flaubury      | Outling   |        127 |  5690 |
| OS           | Oswaldo      | Oslodo        | Tesa      |        964 |  8712 |
+--------------+--------------+---------------+-----------+------------+-------+
3 rows in set (0.00 sec)
```

```
SELECT p.country_code, c.country_name, p.province_name, ct.city_name,
ct.population, p.area
FROM country c
INNER JOIN province p ON p.country_code = c.country_code
INNER JOIN city ct ON p.province_name = ct.province_name
WHERE ct.population < @population;
```

6. 

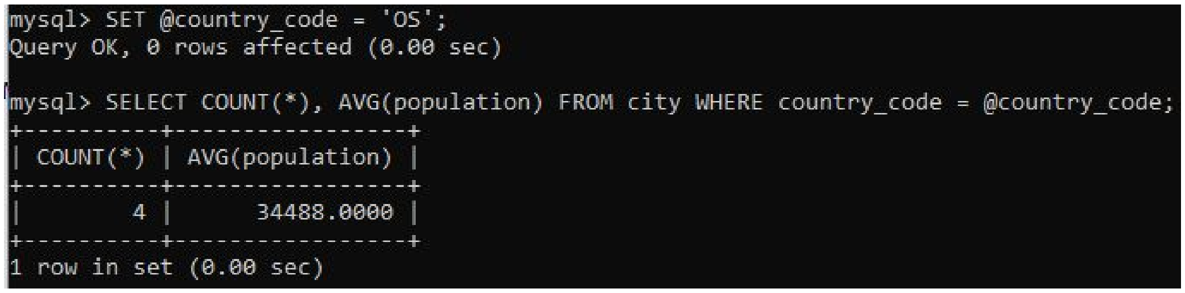
```
mysql> SELECT SUM(area) FROM province;
+-----------+
| SUM(area) |
+-----------+
|    308126 |
+-----------+
1 row in set (0.00 sec)
```

```
SELECT SUM(area) FROM province;
```

7. 

```
mysql> SELECT MIN(gdp), MAX(gdp), AVG(gdp), MIN(inflation), MAX(inflation), AVG(inflation)
    -> FROM country;
+----------+----------+------------+----------------+----------------+------------------+
| MIN(gdp) | MAX(gdp) | AVG(gdp)   | MIN(inflation) | MAX(inflation) | AVG(inflation)   |
+----------+----------+------------+----------------+----------------+------------------+
|    54000 |    78000 | 65666.6667 |            1.7 |            6.7 | 3.49999992052714 |
+----------+----------+------------+----------------+----------------+------------------+
1 row in set (0.00 sec)
```

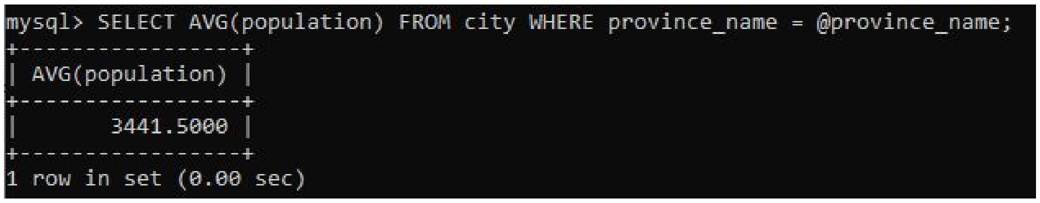
```
SELECT MIN(gdp), MAX(gdp), AVG(gdp), MIN(inflation), MAX(inflation), AVG(inflation)
FROM country;
```

8. 

```
mysql> SET @country_code = 'OS';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT COUNT(*), AVG(population) FROM city WHERE country_code = @country_code;
+----------+-----------------+
| COUNT(*) | AVG(population) |
+----------+-----------------+
|        4 |      34488.0000 |
+----------+-----------------+
1 row in set (0.00 sec)
```

```
SET @country_code = 'OS';
SELECT COUNT(*), AVG(population) FROM city WHERE country_code = @country_code;
```

9. 

```
mysql> SELECT AVG(population) FROM city WHERE province_name = @province_name;
+-----------------+
| AVG(population) |
+-----------------+
|       3441.5000 |
+-----------------+
1 row in set (0.00 sec)
```

```
SET @province_name = "Oslodo";
SELECT AVG(population) FROM city WHERE province_name = @province_name;
```

10. Question does not exist
11. and 12. I was unable to come up with a query that satisfied the requirements described in these questions.