ASSIGNMENT-3

Q) Query all columns (attributes) for every row in the **CITY** table.

The **CITY** table is described as follows:  
A screenshot of a computer

AI-generated content may be incorrect.

SELECT \* FROM CITY;

Q) Query all columns for a city in **CITY** with the *ID* 1661.

SELECT \* FROM CITY WHERE ID = 1661;

Q) Query all attributes of every Japanese city in the **CITY** table. The **COUNTRYCODE** for Japan is JPN.

SELECT \* FROM CITY WHERE COUNTRYCODE = 'JPN';

Q) Write a query that prints a list of employee names (i.e.: the *name* attribute) from the **Employee** table in alphabetical order.

**Input Format**

The **Employee** table containing employee data for a company is described as follows:

A table of numbers with black text

AI-generated content may be incorrect.

where *employee\_id* is an employee's ID number, *name* is their name, *months* is the total number of months they've been working for the company, and *salary* is their monthly salary.

**Sample Input**

A table of numbers and letters

AI-generated content may be incorrect.

**Sample Output**

Angela

Bonnie

Frank

Joe

Kimberly

Lisa

Michael

Patrick

Rose

Todd

**SELECT name FROM Employee ORDER BY name;​**

Q) Query all attributes of every Japanese city in the CITY table. The COUNTRYCODE for Japan is JPN.

The CITY table is described as follows: ****

**SELECT \* FROM CITY WHERE COUNTRYCODE = 'JPN';**

Q) Query a list of CITY and STATE from the STATION table.  
The STATION table is described as follows:  
A table with text and numbers

AI-generated content may be incorrect.

where LAT\_N is the northern latitude and LONG\_W is the western longitude.

SELECT CITY, STATE FROM STATION;

Q) Query a list of **CITY** names from **STATION** for cities that have an even **ID** number. Print the results in any order, but exclude duplicates from the answer.

SELECT DISTINCT CITY FROM STATION WHERE MOD(ID, 2) = 0;

Q) Find the difference between the total number of **CITY** entries in the table and the number of distinct **CITY** entries in the table.  
The **STATION** table is described as follows:



where **LAT\_N** is the northern latitude and **LONG\_W** is the western longitude.

For example, if there are three records in the table with **CITY** values 'New York', 'New York', 'Bengalaru', there are 2 different city names: 'New York' and 'Bengalaru'. The query returns 1, because total number of records-number of cities=3-2=1.

SELECT COUNT(CITY) - COUNT(DISTINCT CITY) FROM STATION;

Q) Query the two cities in **STATION** with the shortest and longest *CITY* names, as well as their respective lengths (i.e.: number of characters in the name). If there is more than one smallest or largest city, choose the one that comes first when ordered alphabetically.  
The **STATION** table is described as follows:



where **LAT\_N** is the northern latitude and **LONG\_W** is the western longitude.

**Sample Input**

For example, **CITY** has four entries: **DEF, ABC, PQRS** and **WXY**.

**Sample Output**

ABC 3

PQRS 4

**Explanation**

When ordered alphabetically, the **CITY** names are listed as **ABC, DEF, PQRS,** and **WXY**, with lengths 3,3,4 and 3. The longest name is **PQRS**, but there are 3 options for shortest named city. Choose **ABC**, because it comes first alphabetically.

**Note**  
You can write two separate queries to get the desired output. It need not be a single query.

SELECT CITY, LENGTH(CITY)

FROM STATION

ORDER BY LENGTH(CITY), CITY

LIMIT 1;

SELECT CITY, LENGTH(CITY)

FROM STATION

ORDER BY LENGTH(CITY) DESC, CITY

LIMIT 1;

Q) Query the average population for all cities in **CITY**, rounded *down* to the nearest integer.

**Input Format**

The **CITY** table is described as follows: 

SELECT FLOOR(AVG(POPULATION)) FROM CITY;

Q) Given the **CITY** and **COUNTRY** tables, query the names of all the continents (*COUNTRY.Continent*) and their respective average city populations (*CITY.Population*) rounded *down* to the nearest integer.

**Note:** *CITY.CountryCode* and *COUNTRY.Code* are matching key columns.

**Input Format**

The **CITY** and **COUNTRY** tables are described as follows: A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

SELECT co.Continent, FLOOR(AVG(ci.Population))

FROM CITY ci, COUNTRY co

WHERE ci.CountryCode = co.Code

GROUP BY co.Continent;