1.Query all columns for all American cities in the **CITY** table with populations larger than 100000. The **CountryCode** for America is USA.

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



**Solution:**

**select \* from city where population > 100000 and countrycode='USA'**

**2.** Query the **NAME** field for all American cities in the **CITY** table with populations larger than 120000. The CountryCode for America is USA.

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



**Solution:**

select name from city where population>=120000 and countrycode='USA'

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

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



**Solution:**

**select \* from city**

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

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



**Solution:**

**select \* from city where ID=1661**

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

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



**Solution:**

**select \* from city where Countrycode='JPN'**

**6.** Query the names of all the Japanese cities in the **CITY** table. The **COUNTRYCODE** for Japan is JPN.  
The **CITY** table is described as follows:



**Solution:**

**select name from city where countrycode='JPN'**

**7.** Query a list of **CITY** and **STATE** from the **STATION** table.  
The **STATION** table is described as follows:

****

**Solution:**

**select city,state from station**

**8.**

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.  
The **STATION** table is described as follows:

****

**Solution:**

**select distinct city from station**

**where id % 2 = 0**

**9.**

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 , because

# Example:Total number of records – number of unique city names =3-2 =1

**Solution:**

**select count(city)-count(distinct city) from station**

**10.**

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.

**Solution:**

**SELECT TOP 1 CITY, LEN(CITY) FROM STATION ORDER BY LEN(CITY),CITY**

**SELECT TOP 1 CITY, LEN(CITY) FROM STATION ORDER BY LEN(CITY) DESC,CITY**

**(there are two queries)**