**Question 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';

**Question 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 COUNTRYCODE = 'USA' AND POPULATION > 120000;

**Question 3: SELECT ALL**

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

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


**Solution:**

select \* from CITY;

**Question 4: Select by ID**

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;

**Question 5: Japanese Cities’ Attribute**

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';

**Question 6: Japanese Cities’ Names**

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';

**Question 6: Weather Observation Station 1**

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


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

**Solution:**

SELECT CITY, STATE FROM STATION;

**Question 7: Weather Observation Station 3**

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:



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

**Solution:**

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

**Question 8: Weather Observation Station 4**

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 unique city names = 3-2 =1

**Solution:**

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

**Question 9: Weather Observation Station 5**

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  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 city, length(city) from station order by length(city) asc, city asc limit 1)

union

(select city, length(city) from station order by length(city) desc, city asc limit 1);

**Question 10: Weather Observation Station 6**

Query the list of CITY names starting with vowels (i.e., a, e, i, o, or u) from ****STATION****. Your result cannot contain duplicates.

****Input Format****

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



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

**Solution:**

select distinct(CITY) from STATION where CITY LIKE 'a%' OR CITY LIKE 'e%' OR CITY LIKE 'i%' OR CITY LIKE 'o%' OR CITY LIKE 'u%';

**Question 11: Weather Observation Station 7**

Query the list of CITY names ending with vowels (a, e, i, o, u) from ****STATION****. Your result cannot contain duplicates.

****Input Format****

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



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

**Solution:**

select distinct(CITY) from STATION where CITY LIKE '%a' OR CITY LIKE '%e' OR CITY LIKE '%i' OR CITY LIKE '%o' OR CITY LIKE '%u';

**Question 12: Weather Observation Station 8**

Query the list of CITY names from ****STATION**** which have vowels (i.e., a, e, i, o, and u) as both their first and last characters. Your result cannot contain duplicates.

****Input Format****

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



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

**Solution:**

select distinct(CITY) from STATION

where (CITY LIKE 'a%' OR CITY LIKE 'e%' OR CITY LIKE 'i%' OR CITY LIKE 'o%' OR CITY LIKE 'u%')

AND

(CITY LIKE '%a' OR CITY LIKE '%e' OR CITY LIKE '%i' OR CITY LIKE '%o' OR CITY LIKE '%u');

**Question 13: Weather Observation Station 9**

Query the list of CITY names from ****STATION**** that do not start with vowels. Your result cannot contain duplicates.

****Input Format****

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



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

**Solution:**

select distinct(CITY) from STATION where CITY NOT in(select CITY where CITY LIKE 'a%' OR CITY LIKE 'e%' OR CITY LIKE 'i%' OR CITY LIKE 'o%' OR CITY LIKE 'u%');

**Question 14: Weather Observation Station 10**

Query the list of CITY names from ****STATION**** that do not end with vowels. Your result cannot contain duplicates.

****Input Format****

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



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

**Solution:**

select distinct(CITY) from STATION where CITY NOT in(select CITY from STATION where CITY LIKE '%a' OR CITY LIKE '%e' OR CITY LIKE '%i' OR CITY LIKE '%o' OR CITY LIKE '%u');