1.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'

# 2.Revising the Select Query I

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'

# **3.**Revising the Select Query II

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;

# 4.Select All

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

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


**Solution**:  
  
SELECT \* FROM CITY;

# 5.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;

# 6.Japanese Cities' Attributes

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 NAME FROM CITY WHERE COUNTRYCODE = 'JPN';

# 7.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;

**8.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 ORDER BY CITY ASC;

# 9.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'. Then 3-2= 1  
  
**Solution:**   
SELECT COUNT(CITY) - COUNT(DISTINCT(CITY)) FROM STATION;

# 10. 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.

**Solution:**   
  
Select city,length(city) from station order By length(city) asc, city asc limit 1;

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

# 11.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%' ;

# 12.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' ;

# 13. 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 '[AEIOU]%[AEIOU]';

# 14.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 NOT (CITY LIKE 'A%' OR

CITY LIKE 'E%' OR

CITY LIKE 'I%' OR

CITY LIKE 'O%' OR

CITY LIKE 'U%');

# 15. 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 NOT (CITY LIKE '%A' OR

CITY LIKE '%E' OR

CITY LIKE '%I' OR

CITY LIKE '%O' OR

CITY LIKE '%U');

16. WEATHER OBSERATION 11

Query the list of *CITY* names from **STATION** that either do not start with vowels or 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 NOT CITY LIKE '[AEIOU]%[AEIOU]';

# 17. Weather Observation Station 12

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

**Input Format**

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



**Solution:** SELECT DISTINCT CITY FROM STATION

WHERE NOT ((

CITY LIKE 'A%' OR

CITY LIKE 'e%' OR

CITY LIKE 'I%' OR

CITY LIKE 'O%' OR

CITY LIKE 'U%' )

OR

(CITY LIKE '%A' OR

CITY LIKE '%E' OR

CITY LIKE '%I' OR

CITY LIKE '%O' OR  
  
CITY LIKE '%U'));

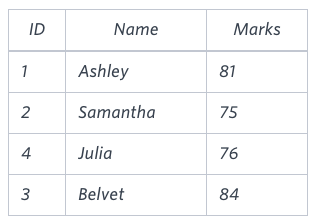
# 18. Higher Than 75 Marks

Query the Name of any student in **STUDENTS** who scored higher than  Marks. Order your output by the last three characters of each name. If two or more students both have names ending in the same last three characters (i.e.: Bobby, Robby, etc.), secondary sort them by ascending ID.

**Input Format**



**Sample Input**



**Solution:**

SELECT NAME FROM STUDENTS

WHERE MARKS > 75

ORDER BY SUBSTRING(NAME, LEN(NAME)-2, 3), id;

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



**Sample Output**

Angela

Bonnie

Frank

Joe

Kimberly

Lisa

Michael

Patrick

Rose

Todd

**Solution:**

SELECT NAME FROM EMPLOYEE

ORDER BY NAME;

# 20.Employee Salaries

Write a query that prints a list of employee names (i.e.: the *name* attribute) for employees in **Employee** having a salary greater than  per month who have been employees for less than  months. Sort your result by ascending *employee\_id*.



**Solution:** SELECT NAME

FROM EMPLOYEE

WHERE SALARY > 2000 AND

MONTHS < 10

ORDER BY EMPLOYEE\_ID ASC;