

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:

select \* from CITY where CountryCode='USA' and population >100000;

1. 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:

select Name from City where population > 120000 and CountryCode='USA';

1. Query all columns (attributes) for every row in the CITY table. The CITY table is described as follows:

select \* from CITY;

1. Query all columns for a city in CITY with the ID 1661. The CITY table is described as follows

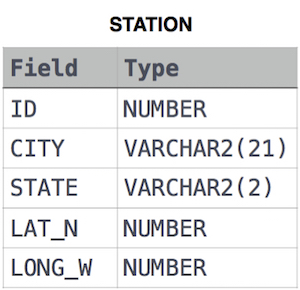
select \* from City where id= 1661;

1. 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';

1. 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:

select name from City where COUNTRYCODE ='JPN';



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

select CITY,STATE from STATION ;

1. 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:-

select Distinct CITY from STATION

where mod(id,2)=0;

1. 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 . Total number of records - number unique city name = 3-2=1;

select count(city)-count(distinct city)

from STATION;

1. 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.

??????????????????????????????????????????????????????????

1. 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.

select Distinct CITY from STATION

where Lower(substr(City,1,1)) in('a','e','i','o','u');

1. Query the list of CITY names ending 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.

select Distinct CITY from STATION

where Lower(substr(City,-1,1)) in('a','e','i','o','u');

1. 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:

select Distinct CITY from STATION

where Lower(substr(City,1,1)) in('a','e','i','o','u') and Lower(substr(City,-1,1)) in('a','e','i','o','u');

1. 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:

select Distinct CITY from STATION

where Lower(substr(City,1,1)) not in('a','e','i','o','u');

1. 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:

select Distinct CITY from STATION

where Lower(substr(City,-1,1)) not in('a','e','i','o','u');

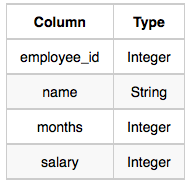
1. Query the list of CITY names from STATION that either do not start with vowels or do not end with vowels. Your result cannot contain duplicates.Input Format .The STATION table is described as follows:

select Distinct CITY from STATION

where Lower(substr(City,1,1)) not in('a','e','i','o','u') OR (Lower(substr(City,-1,1)) not in('a','e','i','o','u'));

1. Write a query that prints a list of employee names (i.e.: the name attribute) for employees in Employee having a salary greater than $2000 per month who have been employees for less than 10 months. Sort your result by ascending employee\_id. Input Format ...

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



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 the their monthly salary.

Sample Input



Sample Output : Angela Michael Todd Joe

Explanation :- Angela has been an employee for 1 month and earns $ 3443per month.

Michael has been an employee for 6 months and earns $ 2017 per month.

Todd has been an employee for 5 months and earns $ 3396 per month.

Joe has been an employee for 9 months and earns $ 3573 per month.

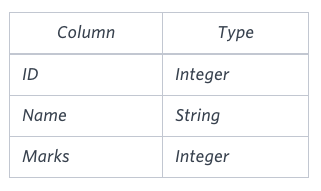
We order our output by ascending employee\_id.

select name from Employee

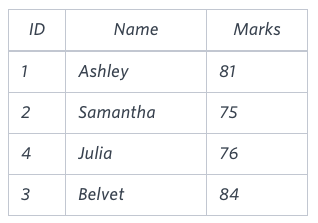
where months <10 and salary>2000

order by employee\_id asc;

1. 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.The STUDENTS table is described as follows: 

The Name column only contains uppercase (A-Z) and lowercase (a-z) letters.Sample Input



Sample Output

Ashley

Julia

Belvet

Explanation

Only Ashley, Julia, and Belvet have Marks > . If you look at the last three characters of each of their names, there are no duplicates and 'ley' < 'lia' < 'vet'.

select Name

from students

where marks>75

order by substr(name,-3,3),id;

1. 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:



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**

****

**Sample Output**

Angela

Bonnie

Frank

Joe

Kimberly

Lisa

Michael

Patrick

Rose

Todd

Joins



1. Given the CITY and COUNTRY tables, query the names of all cities where the CONTINENT is 'Africa'.Note: CITY.CountryCode and COUNTRY.Code are matching key columns. Input Format The CITY and COUNTRY tables are described as follows:

SELECT city.name

FROM city city JOIN country country

ON city.countrycode = country.code

WHERE country.continent = 'Africa';

1. Given the CITY and COUNTRY tables, que
2. ry 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.

SELECT COUNTRY.CONTINENT, round(avg(City.Population)-0.5)

FROM CITY INNER JOIN COUNTRY

ON CITY.COUNTRYCODE = COUNTRY.CODE

GROUP BY COUNTRY.CONTINENT;

(or)

SELECT COUNTRY.CONTINENT, FLOOR(AVG(CITY.POPULATION))

FROM CITY INNER JOIN COUNTRY

ON CITY.COUNTRYCODE = COUNTRY.CODE

GROUP BY COUNTRY.CONTINENT;

The CITY table is described as follows:



1. Query the total population of all cities in 22. CITY where District is California.

select sum(population) from city where district ='California';

1. Query the average population of all cities in CITY where District is California.

select avg(population) from city where district ='California';

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

select round (avg(population)) from city ;

1. Query the sum of the populations for all Japanese cities in CITY. The COUNTRYCODE for Japan is JPN.

select sum(population) from city where countrycode='JPN';

1. Query the difference between the maximum and minimum populations in CITY.

select max(population)-min(population) from city;