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:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

SELECT * FROM CITY WHERE COUNTRYCODE IN ('USA') AND POPULATION > 100000;

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:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

SELECT NAME FROM CITY WHERE COUNTRYCODE IN ('USA') AND POPULATION > 120000;

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

The CITY table is described as follows:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

SELECT * FROM CITY;

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

The CITY table is described as follows:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

SELECT * FROM CITY WHERE ID IN (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:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

SELECT * FROM CITY WHERE COUNTRYCODE IN ('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:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

SELECT NAME FROM CITY WHERE COUNTRYCODE IN ('JPN');

7. 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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

```
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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT CITY, LENGTH(CITY) FROM STATION ORDER BY LENGTH(CITY) ASC, CITY ASC LIMIT 1;

SELECT CITY, LENGTH(CITY) FROM STATION ORDER BY LENGTH(CITY) DESC, CITY ASC LIMIT 1;

11. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,1,1) IN ('A','E','I','O','U');

12. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,-1,1) IN ('A','E','I','O','U');

13. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,1,1) IN ('A','E','I','O','U') AND SUBSTRING(CITY,-1,1) IN ('A','E','I','O','U');

14. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,1,1) NOT IN ('A','E','I','O','U');

15. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,-1,1) NOT IN ('A','E','I','O','U');

16. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,1,1) NOT IN ('A','E','I','O','U') OR SUBSTRING(CITY,-1,1) NOT IN ('A','E','I','O','U');

17. 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:

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

SELECT DISTINCT CITY FROM STATION WHERE SUBSTRING(CITY,1,1) NOT IN ('A','E','I','O','U') AND SUBSTRING(CITY,-1,1) NOT IN ('A','E','I','O','U');

18. 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

Column	*Type*
ID	*Integer*
Name	*String*
Marks	*Integer*

The STUDENTS table is described as follows:

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

SELECT NAME FROM STUDENTS WHERE MARKS > 75 ORDER BY SUBSTRING(NAME,-3) ASC, ID ASC;

19. 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:

Column	Type
employee_id	Integer
name	String
months	Integer
salary	Integer

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.

SELECT NAME FROM EMPLOYEE ORDER BY NAME;

20. 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.

Input Format

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

Column	Type
employee_id	Integer
name	String
months	Integer
salary	Integer

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.

SELECT NAME FROM EMPLOYEE WHERE SALARY > 2000 AND MONTHS < 10 ORDER BY EMPLOYEE_ID;

21.