

HACKER RANK SQL PRACTICE

I solve all by choosing the ORACLE SQL option.

BASIC SELECT

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

//Following questions are related by this table.//

//

'SELECT *' means that you select all columns in your table. Then, You must use the keyword 'FROM' to specify from which table to get the column names. With 'WHERE', you filter your table according to specified condition(s). If your column type is not a number, you use single quotes around text values. Otherwise, you should not be enclosed in quotes. //

Revising the Select Query 1

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

Solution:

```
SELECT * from CITY WHERE COUNTRYCODE='USA' and POPULATION>100000;
```

Revising the Select Query2

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

Solution:

```
SELECT NAME FROM CITY WHERE POPULATION>120000 AND COUNTRYCODE='USA';
```

Select All

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

Solution:

```
SELECT * FROM CITY;
```

Select By ID

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

Solution:

```
SELECT * FROM CITY WHERE ID=1661;
```

Japanese Cities' Attributes

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

Solution:

```
SELECT * FROM CITY WHERE COUNTRYCODE='JPN';
```

Japanese Cities' Names

Query the names of all the Japanese cities in the **CITY** table. The **COUNTRYCODE** for Japan is JPN.

Solution:

```
SELECT NAME FROM CITY WHERE COUNTRYCODE='JPN';
```

Weather Observation Station

There are some 12 questions that you will use **STATION** table.

STATION

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

- 1) Query a list of **CITY** and **STATE** from the **STATION** table.

Solution:

```
SELECT CITY, STATE FROM STATION;
```

2) Query the following two values from the **STATION** table:

1. The sum of all values in *LAT_N* rounded to a scale of decimal places.
2. The sum of all values in *LONG_W* rounded to a scale of decimal places.

Solution:

```
SELECT round(SUM(LAT_N),2), round(SUM(LONG_W),2) FROM 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.

Solution:

```
SELECT DISTINCT(CITY) FROM STATION WHERE MOD(ID,2)=0;
```

4) Find the difference between the total number of **CITY** entries in the table and the number of distinct **CITY** entries in the table.

Solution:

```
SELECT COUNT(CITY)-COUNT(DISTINCT(CITY)) FROM 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.

Solution:

```
SELECT CITY, LENGTH(CITY) FROM (select c1.* from (select CITY, LENGTH(CITY), row_number()  
over (order by LENGTH(CITY) ASC, CITY) as rn FROM STATION) c1 where c1.rn =1
```

union all

```
select c2.* from (select CITY, LENGTH(CITY), row_number() over (order by LENGTH(CITY) DESC,  
CITY) as rn FROM STATION) c2 where c2.rn =1);
```