HACKERRANK Questions

Query a list of **CITY** and **STATE** from the **STATION** 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 CITY, STATE FROM STATION;
```

Using Modulus:

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
ORDER BY CITY;
```

Find Difference:

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.

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

```
SELECT COUNT(CITY) - COUNT(DISTINCT CITY) AS difference
FROM STATION;
```

CREATE A DATABASE:

```
CREATE DATABASE IF NOT EXISTS school;

USE school;



CREATE TABLE Students (
student_id INT PRIMARY KEY,
first_name TEXT,
```

```
last_name TEXT
);

CREATE TABLE FavoriteSubjects (
student_id INT,
rank INT,
subject TEXT,
FOREIGN KEY (student_id) REFERENCES Students(student_id)
);

// Make sure that the csv are stored in var/lib/mysql/school

LOAD DATA INFILE 'Students.csv'
INTO TABLE Students
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
(student_id, first_name, last_name);

LOAD DATA INFILE 'FavoriteSubjects.csv'
INTO TABLE FavoriteSubjects
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
(student_id, rank, subject);

SELECT Students.first_name, Students.last_name, FavoriteSubjects
FROM Students
JOIN FavoriteSubjects ON Students.student_id = FavoriteSubjects
ORDER BY Students.student_id, FavoriteSubjects.rank;
```