HACKERRANK Queries

1. **Employee Salaries**

   Link: https://www.hackerrank.com/challenges/salary-of-employees/problem

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:

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.

Sample Input

employee_id	name	months	salary
12228	Rose	15	1968
33645	Angela	1	3443
45692	Frank	17	1608
56118	Patrick	7	1345
59725	Lisa	11	2330
74197	Kimberly	16	4372
78454	Bonnie	8	1771
83565	Michael	6	2017
98607	Todd	5	3396
99989	Joe	9	3573

Sample Output

Angela
Michael
Todd
Joe

Explanation

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

Solution:

```
SELECT
  name
FROM
  EMPLOYEE
WHERE
  salary > 2000
and
  months < 10
ORDER BY
  employee_id;
```

2. **Higher Than 75 Marks**

   Link: https://www.hackerrank.com/challenges/more-than-75-marks/problem

Query the Name of any student in STUDENTS who scored higher than **75** 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:

Column	Type
ID	Integer
Name	String
Marks	Integer

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

Sample Input

ID	Name	Marks
1	Ashley	81
2	Samantha	75
4	Julia	76
3	Belvet	84

Sample Output

Ashley

Julia

Belvet

Explanation

Only Ashley, Julia, and Belvet have Marks > 75. If you look at the last three characters of each of their names, there are no duplicates and 'ley' < 'lia' < 'vet'.

Solution:

```
SELECT
  name
FROM
  Students
WHERE
  Marks > 75
ORDER BY
RIGHT(name, 3), ID ASC;
```

Using where clause you can retrieve data where marks of students is greater than 75.
By making use of order by clause you can sort the data in ascending order, here they have asked to sort using the last 3 characters of name and id.

3. **Weather Observation Station 5**
   Link:
   https://www.hackerrank.com/challenges/weather-observation-station-5/problem

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.

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 the 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.

Solution:

```
SELECT city, LENGTH(city) FROM STATION
ORDER BY LENGTH(city), city LIMIT 1;

SELECT city, LENGTH(city) FROM STATION
ORDER BY LENGTH(city) DESC, city DESC LIMIT 1;
```

First we need to print the shortest city name and its length for that sort of cities and length(this you can get using length()) in ascending order and using limit to retrieve first data.
Next we need to print the largest city name and its length for that sort of cities and length(this you can get using length()) in Descending order and using limit to retrieve first data.

4. Population Density Difference
   Link:
   https://www.hackerrank.com/challenges/population-density-difference/problem?h_r=internal-search

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

Input Format

The CITY table is described as follows:

CITY

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

Solution:

```
SELECT MAX(POPULATION) - MIN(POPULATION) FROM CITY;
```

5. Average Population
   Link:
   https://www.hackerrank.com/challenges/average-population/problem?isFullScreen=true

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

Input Format

The CITY table is described as follows:

CITY

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

Solution:

```
SELECT
  ROUND(AVG(POPULATION))
FROM
  CITY
```

6. Weather Observation Station 13
   Link:
   https://www.hackerrank.com/challenges/weather-observation-station-13/problem

Query the sum of Northern Latitudes (LAT_N) from STATION having values greater than 38.7880 and less than 137.2345. Truncate your answer to 4 decimal places.

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.

Solution:

```
SELECT
  TRUNCATE(SUM(LAT_N),4)
FROM
  STATION
WHERE
  LAT_N > 38.7880 AND LAT_N <137.2345;
```

7. Weather Observation Station 14
   Link: https://www.hackerrank.com/challenges/weather-observation-station-14/problem

Query the greatest value of the Northern Latitudes (LAT_N) from STATION that is less than 137.2345. Truncate your answer to 4 decimal places.

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.

Solution:

```
SELECT
  TRUNCATE(MAX(LAT_N),4)
FROM
  STATION
WHERE
  LAT_N < 137.2345;
```

8. Weather Observation Station 16
   Link:
   https://www.hackerrank.com/challenges/weather-observation-station-16/problem

Query the smallest Northern Latitude (LAT_N) from STATION that is greater than 38.7780. Round your answer to 4 decimal places.

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.

Solution:

```
SELECT
  ROUND(MIN(LAT_N),4)
FROM
  STATION
WHERE
  LAT_N > 38.7780;
```

9. Weather Observation Station 16
   Link:
   https://www.hackerrank.com/challenges/weather-observation-station-17/problem

Query the Western Longitude (LONG_W)where the smallest Northern Latitude (LAT_N) in STATION is greater than . Round your answer to decimal places.

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.

Solution:

```
SELECT
  ROUND(LONG_W, 4)
FROM
  STATION
WHERE
  LAT_N > 38.7780
ORDER BY
  LAT_N
LIMIT 1;
```

10. Weather Observation Station 15
    Link:
    https://www.hackerrank.com/challenges/weather-observation-station-15/problem

Query the Western Longitude (LONG_W) for the largest Northern Latitude (LAT_N) in STATION that is less than . Round your answer to decimal places.

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.

Solution:

```
SELECT
  ROUND(LONG_W, 4)
FROM
  STATION
WHERE
  LAT_N < 137.2345
ORDER BY
  LAT_N
DESC LIMIT 1 ;
```

11. Weather Observation Station 17

Link: https://www.hackerrank.com/challenges/weather-observation-station-17/problem

Query the Western Longitude (LONG_W)where the smallest Northern Latitude (LAT_N) in STATION is greater than 38.7780. Round your answer to 4 decimal places.

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.

Solution

```
SELECT
  ROUND(LONG_W, 4)
FROM
  STATION
WHERE
  LAT_N > 38.7780
ORDER BY
  LAT_N
LIMIT 1 ;
```

12. Type Of Triangle

Link: https://www.hackerrank.com/challenges/what-type-of-triangle/problem

Write a query identifying the type of each record in the TRIANGLES table using its three side lengths. Output one of the following statements for each record in the table:

- Equilateral: It's a triangle with sides of equal length.
- Isosceles: It's a triangle with sides of equal length.
- Scalene: It's a triangle with sides of differing lengths.
- Not A Triangle: The given values of A, B, and C don't form a triangle.

Input Format

The TRIANGLES table is described as follows:

Column	Type
A	Integer
B	Integer
C	Integer

Each row in the table denotes the lengths of each of a triangle's three sides.

Sample Input

A	B	C
20	20	23
20	20	20
20	21	22
13	14	30

Sample Output

Isosceles

Equilateral

Scalene

Not A Triangle

Explanation

Values in the tuple form an Isosceles triangle, because .

Values in the tuple form an Equilateral triangle, because . Values in the tuple form a Scalene triangle, because .

Values in the tuple cannot form a triangle because the combined value of sides and is not larger than that of side .

Solution:

```
SELECT
CASE
WHEN A+B > C THEN
CASE
WHEN A=B AND B=C THEN 'Equilateral'
WHEN A=B OR B=C OR C=A THEN 'Isosceles'
WHEN A!=B OR B!=C OR C!=A THEN 'Scalene'
END
ELSE 'Not A Triangle'
END
FROM TRIANGLES;
```