# SQL Basics

## Revisiting the select Query 1: 1.

https://www.hackerrank.com/challenges/revising-the-select-query/problem?isFullScreen=true

SELECT * FROM CITY → All columns.

WHERE COUNTRYCODE = 'USA' → filter american cities

AND POPULATION > 100 000 → population greater than 100000

## Revisiting the select Query II: 2.

https://www.hackerrank.com/challenges/revising-the-select-query-2/problem?isFullScreen=true

SELECT NAME FROM CITY → Query 'NAME' column

WHERE COUNTRYCODE = 'USA' → filter out American cities

AND POPULATION >120000 → population greater than 120000

## Select All 3.

https://www.hackerrank.com/challenges/select-all-sql/problem?isFullScreen=true

SELECT * FROM CITY

Query all columns from the table

## Select By Id 4.

https://www.hackerrank.com/challenges/select-by-id/problem?isFullScreen=true

SELECT * FROM CITY

WHERE ID = 1661 → specific ID

## Japanese City Attributes

5.

https://www.hackerrank.com/challenges/japanese-cities-attributes/problem?isFullScreen=true

SELECT * FROM CITY

WHERE COUNTRY CODE = 'JPN' → Speific for Japan country.

## Japanese city names

6.

https://www.hackerrank.com/challenges/japanese-cities-name/problem?isFullScreen=true

SELECT NAME FROM CITY → Querying `NAME' column on city.

WHERE COUNTRY CODE = 'JPN' → Speific for Japan country.

## Weather Observation Station 1

7.

https://www.hackerrank.com/challenges/weather-observation-station-1/problem?isFullScreen=true

from this table.

SELECT CITY, STATE FROM STATION

querying these two columns

## Weather Observation Station 3

8.

https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true

city column

SELECT DISTINCT CITY FROM STATION

exclude duplicates station table

WHERE ID % 2 = 0

ID is the even number.

## Weather Observation Station 4

9.

https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true

total number of city — total number of distinct city

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

FROM STATION
```

## Weather Observation Station 6

10.

https://www.hackerrank.com/challenges/weather-observation-station-6/problem?isFullScreen=true

```
SELECT CITY FROM STATION

WHERE SUBSTR( CITY, 1, 1) = 'a' OR

      SUBSTR (CUTY, 1, 1) = 'e' OR

      SUBSTR (CITY, 1, 1) = 'i' OR

      SUBSTR (CITY, 1, 1) = 'o' OR

      SUBSTR ( CITY, 1, 1) = 'u'
```

Starting with vowel's

Here 'SUBSTRING' can be used.

Note: SUBSTR ( string, start, length) ⇒ Extract a portion of string.

string FROM start for length

⇓

{-end} if -ve from backside.

## Weather Observation Station 7

11.

https://www.hackerrank.com/challenges/weather-observation-station-7/problem?isFullScreen=true

```
SELECT CITY FROM STATION

WHERE   SUBSTR( CITY,-1, 1) = 'a' OR

        SUBSTR (CUTY,-1,1) = 'e' OR

        SUBSTR (CITY,-1,1) = 'i' OR

        SUBSTR (CITY,-1,1) = 'O' OR

        SUBSTR ( CITY,-1, 1) = 'U'
```

ending with vowel's

Here 'SUBSTRING' can be used.

## Weather Observation Station 8

12.

https://www.hackerrank.com/challenges/weather-observation-station-8/problem?isFullScreen=true

result should not contain duplicates

```
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')
```

if the first characters are vowels

if the last characters are vowels

## Weather Observation Station 9

13.

https://www.hackerrank.com/challenges/weather-observation-station-9/problem?isFullScreen=true

avoid duplications

```
SELECT DISTINCT CITY FROM STATION

WHERE LOWER (SUBSTR ( CITY, 1,1)) NOT IN ('a', 'e', 'i', 'o','u')
```

LOWER ↑ so that capital and small both characters are covered

SUBSTR ( CITY, 1,1) ↑ First character of string

NOT IN ('a', 'e', 'i', 'o','u') are not vowels

## Weather Observation Station 10

14.

https://www.hackerrank.com/challenges/weather-observation-station-10/problem?isFullScreen=true

avoid duplications

```
SELECT DISTINCT CITY FROM STATION

WHERE LOWER (SUBSTR ( CITY,-1,1)) NOT IN ('a', 'e', 'i', 'o','u')
```

LOWER ↑ so that capital and small both characters are covered

SUBSTR ( CITY,-1,1) ↑ Last character of string

NOT IN ('a', 'e', 'i', 'o','u') are not vowels

## Weather Observation Station 11

14.

https://www.hackerrank.com/challenges/weather-observation-station-11/problem?isFullScreen=true

duplicate

SELECT DISTINCT CITY FROM STATION

WHERE LOWER(SUBSTR(CITY,1,1)) NOT IN ('a','e','i','o','u')

first letter should not vowel

OR LOWER(SUBSTR(CITY,-1,1)) NOT IN ('a','e','i','o','u')

last letter should not characters

## Weather Observation 12

15.

https://www.hackerrank.com/challenges/weather-observation-station-12/problem?isFullScreen=true

duplicate

SELECT DISTINCT CITY FROM STATION

WHERE LOWER(SUBSTR(CITY,1,1)) NOT IN ('a','e','i','o','u')

first letter should not vowel

OR LOWER(SUBSTR(CITY,-1,1)) NOT IN ('a','e','i','o','u')

last letter should not characters

## Higher than 75 marks

16.

https://www.hackerrank.com/challenges/more-than-75-marks/problem?isFullScreen=true

SELECT NAME FROM STUDENTS

WHERE MARKS >75

ORDER BY SUBSTR(NAME,-3,3) ASC, ID ASC

last 3 characters in ascending order

id ascending order

## Employee Names

17.

https://www.hackerrank.com/challenges/name-of-employees/problem?isFullScreen=true

```
SELECT  NAME  FROM  EMPLOYEE

ORDER  BY  NAME  ASC
```

order name in ascending order

## Employee Salaries

18.

https://www.hackerrank.com/challenges/salary-of-employees/problem?isFullScreen=true

```
SELECT   NAME   FROM   EMPLOYEE

WHERE    SALARY > 2000   AND   MONTHS < 10

ORDER  BY  EMPLOYEE-ID   ASC
```

salary greater than$ 2000

less than 10 months

order by employee-id in ascending order

## Type of Triangle

19.

https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true

```
SELECT
  CASE
    WHEN A+B<=C  OR  A+C<=B  OR  B+C<=A   THEN
                                    'Not a Triangle'
    WHEN  (A=B)  OR(B=C)  THEN  'Equilateral'
    WHEN  (A=B)  OR  (A=C)  OR  (B=C)  THEN 'Isosceles'
    ELSE  'Scalene'
  END
FROM
    TRIANGLES
```

choices of triangles

## THE PADS.

20.

https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true

Name (A)/(D) / (P)/(S)

```
SELECT  CONCAT( NAME , '(', SUBSTRING(OCCUPATION, 1, 1), ')')

AS  OccupationName    FROM    OccupationName

ORDER  BY  NAME;

SELECT  CONCAT('There are total of' , COUNT(OCCUPATION), '',

        LOWER (OCCUPATION) , 's.')
FROM  OCCUPATION
GROUP BY  OCCUPATION

ORDER BY  COUNT(*) , OCCUPATION
```

↳ groupby the occupation and see the no. of rows.

⇝ sort according to number of occurreances and occupation in ascending order.

## Revising Aggregation. The Count function

21.

https://www.hackerrank.com/challenges/revising-aggregations-the-count-function/problem?isFullScreen=true

```
SELECT  COUNT(*)  FROM  CITY

WHERE  POPULATION > 100000
```

## Revising Aggregation- The Sum Function. 22.

https://www.hackerrank.com/challenges/revising-aggregations-sum/problem?isFullScreen=true

```
SELECT SUM (POPULATION) FROM CITY
WHERE DISTRICT = 'California'
```

sum of population.

## Revising Aggregation Function - Averages 23.

Average

```
SELECT AVG (POPULATION) FROM CITY
WHERE DISTRICT = 'California'
```

## Average Population 24.

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

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

↑ round down to nearest integer

average of population

## Japan Population 25.

https://www.hackerrank.com/challenges/japan-population/problem?isFullScreen=true

sum of population.

```
SELECT SUM (POPULATION) FROM CITY
WHERE COUNTRYCODE = 'JPN'
```

countrycode is JPN

## Population Density Difference

26.

https://www.hackerrank.com/challenges/population-density-difference/problem?isFullScreen=true

```
SELECT   MAX( POPULATION)  -  MIN( POPULATION)

FROM     CITY
```

maximum

minimum of population of city

## The Blunder

27.

https://www.hackerrank.com/challenges/the-blunder/problem?isFullScreen=true

round upto next integer

absolute difference

Replace salary string '0' with ''

```
SELECT   CEILING  (ABS (AVG( CAST( REPLACE (Salary, '0', '')

         AS DECIMAL))  -  AVG (SALARY))    FROM  EMPLOYEES
```

average

CAST AS DECIMAL

## Top Earners

28

https://www.hackerrank.com/challenges/earnings-of-employees/problem?isFullScreen=true

```
SELECT  (months * salary ) as  earnings , COUNT (*)  FROM Employee

GROUP  BY    earnings

ORDER  BY    earnings    DESC

LIMIT 1  →  TOP
```

total monthly earnings

count of employee according to earnings

## weather Observation Station - 2

29

https://www.hackerrank.com/challenges/weather-observation-station-2/problem?isFullScreen=true

```
SELECT   ROUND (SUM (LAT-N) ,2)
         ROUND ( SUM (LONG.W), 2)
         FROM   STATION
```

## weather Station Observation - 13

30

https://www.hackerrank.com/challenges/weather-observation-station-13/problem?isFullScreen=true

round of to 4 digit

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

greater

less than

## weather observation station 14

31

https://www.hackerrank.com/challenges/weather-observation-station-14/problem?isFullScreen=true

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

## Weather Observation Station 15

32

https://www.hackerrank.com/challenges/weather-observation-station-15/problem?isFullScreen=true

round up to 4 decimal points.

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

ordered in desunding order limited 1

## weather Observation Station 16

33

https://www.hackerrank.com/challenges/weather-observation-station-16/problem?isFullScreen=true

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

## Weather Observation Station 17

34

https://www.hackerrank.com/challenges/weather-observation-station-17/problem?isFullScreen=true

```
SELECT ROUND (LONG-W, 4) FROM STATION
WHERE LAT_N > 38.7780
ORDER BY LAT-N ASC LIMIT 1
```

Ordering by the latitude limited to1 .

## Weather Observation Station 18

35

https://www.hackerrank.com/challenges/weather-observation-station-17/problem?isFullScreen=true

max

min → longitude

```
SELECT  ROUND (ABS (MAX(LONG-W) -  MIN( LONG_W))
                    + ABS (MAX (LAT-N) -  MIN( LAT_N)),4)
FROM  STATION
```

→ Rouded 4

max

min → latitude

## Weather Observation Station 19

36

https://www.hackerrank.com/challenges/weather-observation-station-19/problem?isFullScreen=true

```
SELECT  ROUND ( SQRT ((POWER ((MAX(LONG_W) - MIN (LONG_W)),2)
                    + POWER ((MAX(LAT-N) - MIN (LAT_N)),2) , 4)
FROM  STATION
```

(Euclidean distance)

## Population Census

37

https://www.hackerrank.com/challenges/asian-population/problem?isFullScreen=true

```
SELECT  SUM( CITY. POPULATION)  FROM  CITY
INNER JOIN   COUNTRY   ON   CITY. COUNTRYCODE =  COUTRY. CODE
WHERE  COUNTRY. CONTINENT=  'Asia'
```

population sum

table

joined

countrycode

continent → 'Asia'

## African Cities

38

https://www.hackerrank.com/challenges/african-cities/problem?isFullScreen=true

```
SELECT   CITY. NAME   FROM   CITY  ← Table

INNER JOIN   COUNTRY   ON   CITY. COUNTRYCODE = COUNTRY. CODE
```

joined on country

code.

```
WHERE   CONTINENT = 'Africa'
```

Africa

## Average Population of Each Continent

3

https://www.hackerrank.com/challenges/average-population-of-each-continent/problem?isFullScreen=true

floor, average city popl$^n$

```
SELECT   COUNTRY. CONTINENT, FLOOR(AVG( CITY. POPULATION))
 FROM    CITY ← Table
INNER JOIN   COUNTRY   ON   CITY.   COUTRYCODE =   COUNTRY. CODE
```

joined on country table

```
GROUP BY   CONTINENT
```

grouped by continent