- Consider $P_1(a, b)$ and $P_2(c, d)$ to be two points on a 2D plane.
  - a happens to equal the minimum value in Northern Latitude (LAT_N in **STATION**).
  - b happens to equal the minimum value in Western Longitude (LONG_W in **STATION**).
  - c happens to equal the maximum value in Northern Latitude (LAT_N in **STATION**).
  - d happens to equal the maximum value in Western Longitude (LONG_W in **STATION**).

  Query the Manhattan Distance between points P_1 and P_2 and round it to a scale of 4 decimal places.

```
SELECT ABS(ROUND((A-C)+(B-D),4))
FROM (
      SELECT MIN(LAT_N) AS A, MIN(LONG_W) AS B,MAX(LAT_N) AS C,
MAX(LONG_W) AS D
FROM STATION) AS T;
```

- Consider $P_1(a, c)$ and $P_2(b, d)$ to be two points on a 2D plane where (a, b) are the respective minimum and maximum values of Northern Latitude (LAT_N) and (c, d) are the respective minimum and maximum values of Western Longitude (LONG_W) in **STATION**.

  Query the Euclidean Distance between points P_1 and P_2 and format your answer to display 4 decimal digits.

```
SELECT ROUND(SQRT(POW((B-A),2)+POW((D-C),2)),4) AS S
FROM (
SELECT MIN(LAT_N) AS A, MIN(LONG_W) AS C,MAX(LAT_N) AS B,
MAX(LONG_W) AS D
FROM STATION) AS T;
```

- A median is defined as a number separating the higher half of a data set from the lower half. Query the median of the Northern Latitudes (LAT_N) from STATION and round your answer to 4 decimal places.

```
SELECT
ROUND(LAT_N,4) AS R

FROM(
    SELECT
    LAT_N
    FROM STATION
    ORDER BY LAT_N ASC
    LIMIT 250
    ) AS T
ORDER BY LAT_N DESC
LIMIT 1;
OR
SELECT ROUND(LAT_N,4)
FROM
(
SELECT
LAT_N,
ROW_NUMBER() OVER(ORDER BY LAT_N) AS R
FROM STATION) AS T
WHERE R=250.0000

;
```

- Query the total population of all cities in CITY where District is California.

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

- Query the average population of all cities in CITY where District is California.

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

- Query the sum of the populations for all Japanese cities in CITY. The COUNTRYCODE for Japan is JPN.

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

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

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

    - SELECT

```
NAME, AVG(SALARY) AS A
FROM EMPLOYEES
GROUP BY NAME ),
CT1 AS (
SELECT COUNT(A) AS A1
FROM CT),
CT2 AS (
SELECT COUNT(CT.A) AS A2
FROM CT
WHERE CT.A LIKE '%0%')
```

    - Samantha was tasked with calculating the average monthly salaries for all employees in the EMPLOYEES table, but did not realize her keyboard's key was broken until after completing the calculation. She wants your help finding the

difference between her miscalculation (using salaries with any zeros removed), and the actual average salary.Write a query calculating the amount of error (i.e.: average monthly salaries), and round it up to the next integer.

```
WITH ct AS (
    SELECT
        id,
        name,
        salary,
        CASE
            WHEN salary LIKE '%0%'
                THEN CAST(REPLACE(salary, '0', '') AS
DECIMAL(10,2))
            ELSE salary
        END AS flag
    FROM employees
    where salary between 1000 and 100000
)
SELECT round(CEILING(AVG(salary) - AVG(flag)),0) AS c
FROM ct;
```

- We define an employee's total earnings to be their monthly $salary \times months$ worked, and the maximum total earnings to be the maximum total earnings for any employee in the **Employee** table. Write a query to find the maximum total earnings for all employees as well as the total number of employees who have maximum total earnings. Then print these values as 2 space-separated integers.

```
SELECT top 1
concat(earnings,' ',count(earnings))
from(
select employee_id, name, months, salary, (months*salary) as
earnings, row_number() over(order by (months*salary) desc) as r
from Employee ) as t
group by earnings
order by count(earnings) desc
;
```

- Query the following two values from the STATION table:

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

```
select
round(sum(lat_n),2) as lat, round(sum(long_w),2)as long
from station;
```