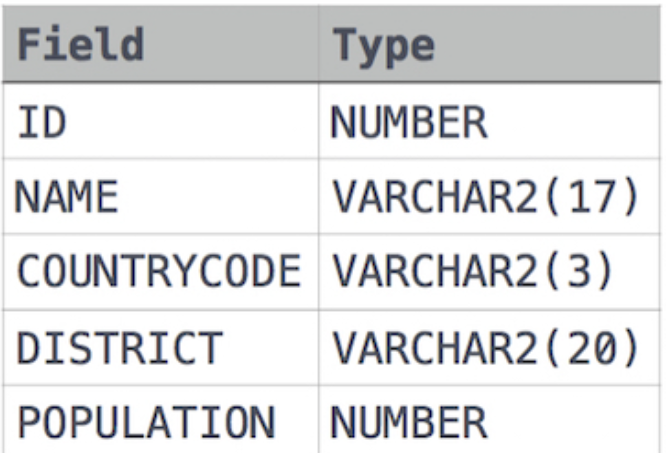
SQL TASK

10-01-2025

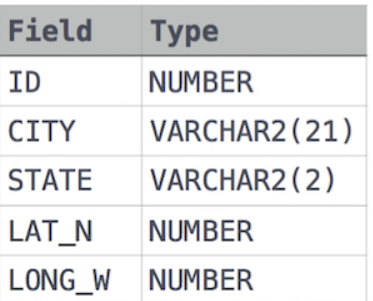
QUE 1) - Query all columns for all American cities in the CITY table with populations larger than 100000. The CountryCode for America is USA.

The CITY table is described as follows:

****

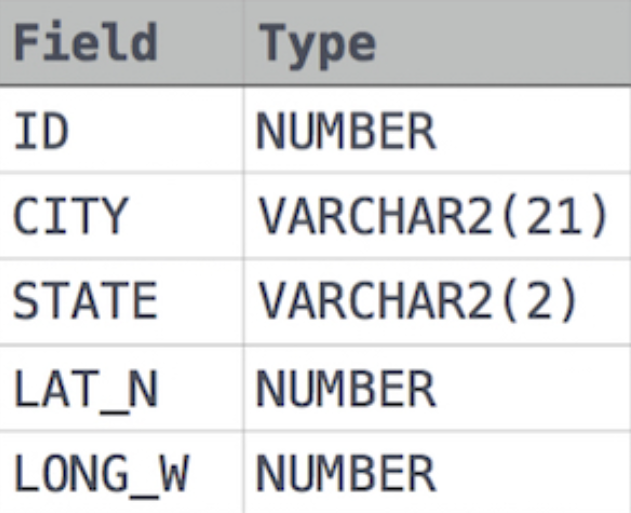
Solution - select \* from city where population > 100000 AND CountryCode = 'USA'

QUE 2) - 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:



Solution - select COUNT (city)- COUNT (DISTINCT city) from station;

QUE 3) - Query the list of CITY names ending with vowels (a, e, i, o, u) from **STATION**. Your result cannot contain duplicates.



Solution - SELECT DISTINCT CITY

FROM STATION

WHERE CITY LIKE '%a'

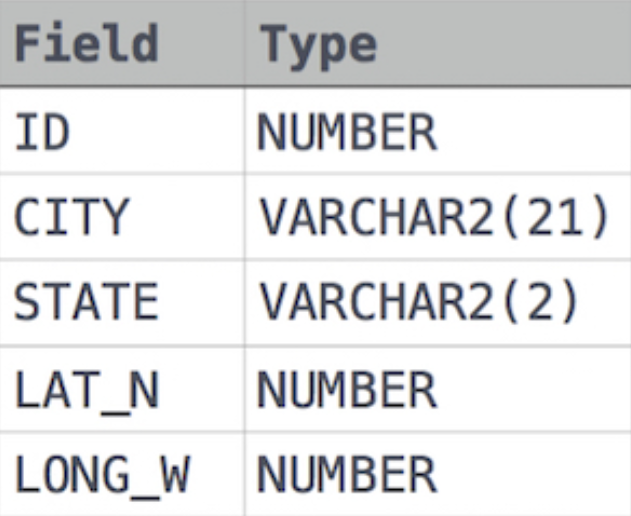
OR CITY LIKE '%e'

OR CITY LIKE '%i'

OR CITY LIKE '%o'

OR CITY LIKE '%u';

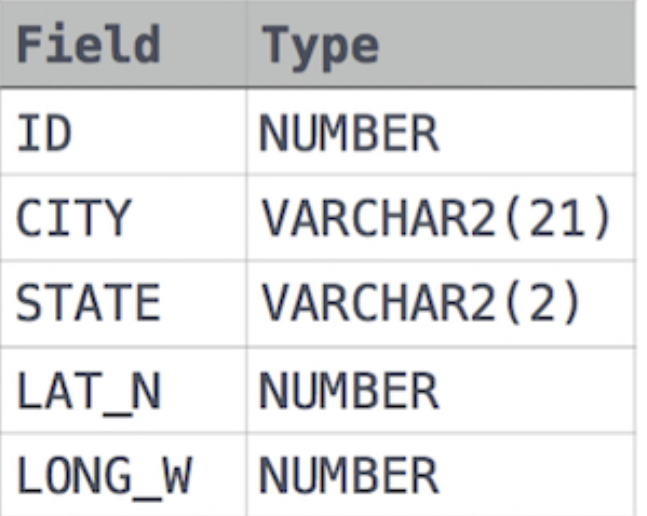
QUE 4) - Query the list of CITY names from **STATION** which have vowels (i.e., *a*, *e*, *i*, *o*, and *u*) as both their first and last characters. Your result cannot contain duplicates



Solution - SELECT DISTINCT CITY FROM STATION WHERE LOWER(LEFT(CITY, 1)) IN ('a', 'e', 'i', 'o', 'u') AND LOWER(RIGHT(CITY, 1)) IN ('a', 'e', 'i', 'o', 'u') ORDER BY CITY;

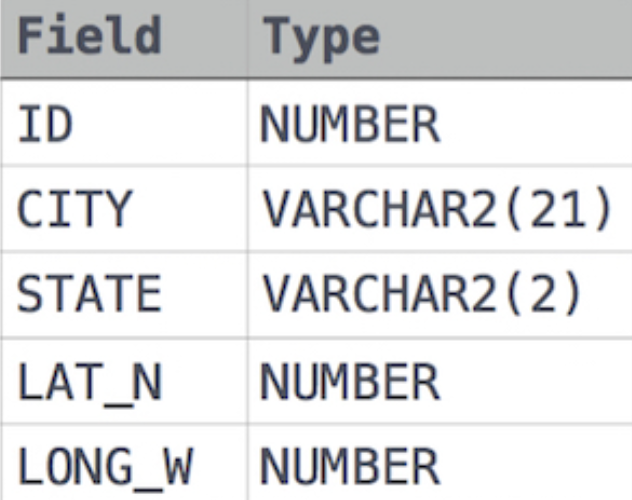
11-01-2025

QUE 5) - Query the list of CITY names from **STATION** that *do not start* with vowels. Your result cannot contain duplicates.



Solution - SELECT DISTINCT CITY FROM STATION WHERE CITY NOT LIKE 'A%' AND CITY NOT LIKE 'E%' AND CITY NOT LIKE 'I%' AND CITY NOT LIKE 'O%' AND CITY NOT LIKE 'U%';

QUE 6) - Query the list of CITY names from **STATION** that do not end with vowels. Your result cannot contain duplicates



Solution - SELECT DISTINCT CITY FROM STATION

WHERE CITY NOT LIKE '%A'

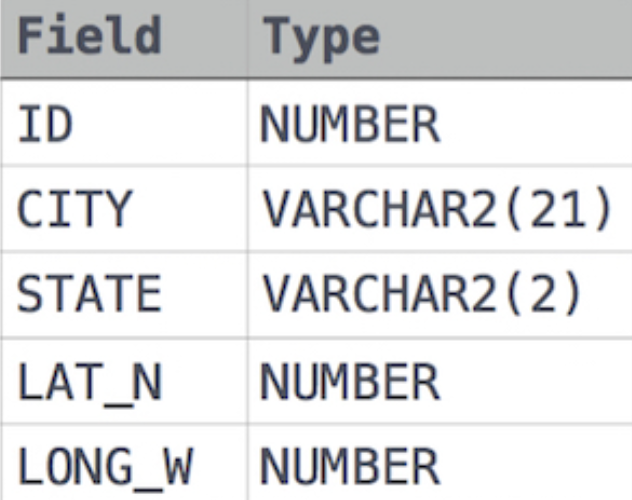
AND CITY NOT LIKE'%E'

AND CITY NOT LIKE'%I'

AND CITY NOT LIKE'%O'

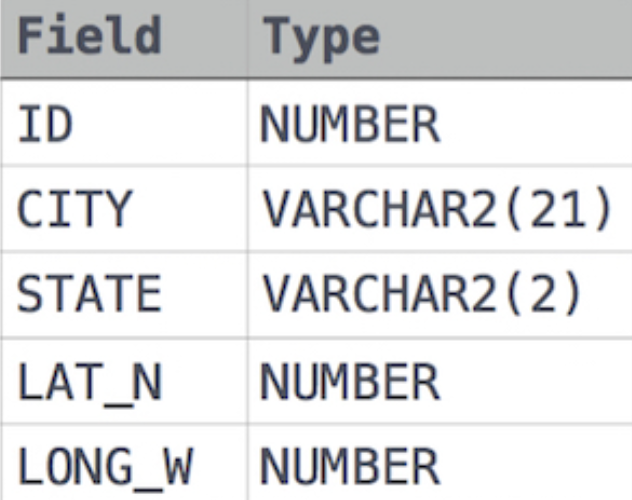
AND CITY NOT LIKE'%U';

QUE 7) - Query the list of CITY names from **STATION** that either do not start with vowels or do not end with vowels. Your result cannot contain duplicates



Solution - SELECT DISTINCT city FROM station WHERE RIGHT(LOWER(city), 1) NOT IN ('a', 'e', 'i', 'o', 'u') OR LEFT(LOWER(city), 1) NOT IN ('a', 'e', 'i', 'o', 'u');

QUE 8) - Query the list of *CITY* names from **STATION** that do not start with vowels and do not end with vowels. Your result cannot contain duplicates.



Solution - SELECT DISTINCT CITY from STATION WHERE RIGHT(LOWER(CITY),1) NOT IN ('a','e','i','o','u') AND LEFT(LOWER(CITY),1 ) NOT IN('a','e','i','o','u');

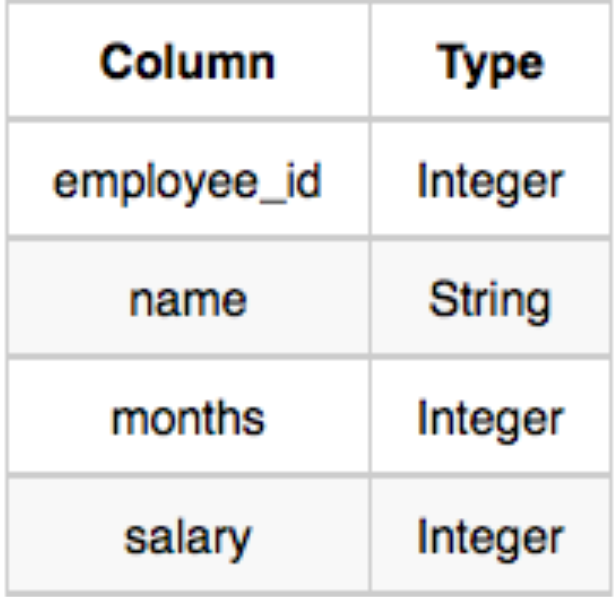
12-01-2025

QUE 9) - Query the Name of any student in **STUDENTS** who scored higher than  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



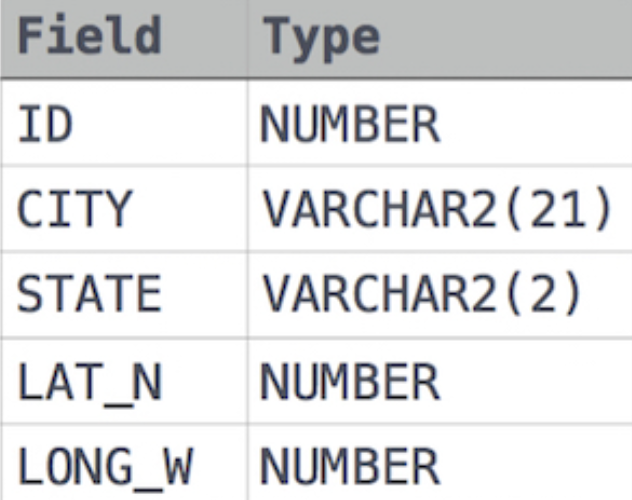
Solution - select name from students where marks > 75 order by RIGHT(NAME,3),ID ;

QUE 10) - Write a query that prints a list of employee names (i.e.: the *name* attribute) from the **Employee** table in alphabetical order.



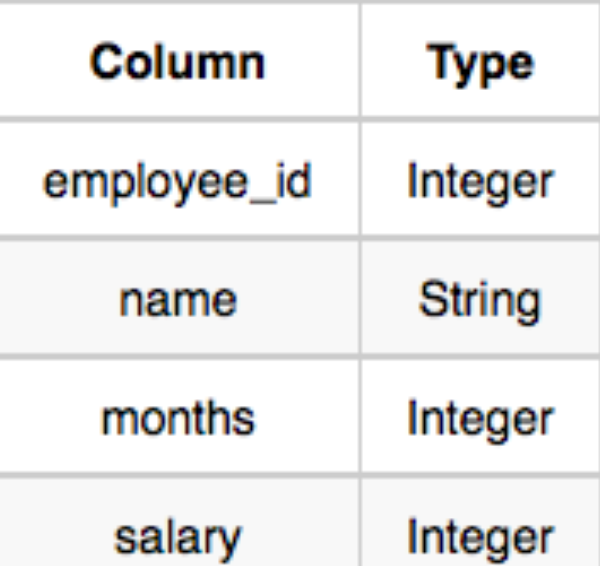
Solution - select name from Employee order by name;

QUE 11) - 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



Solution - SELECT CITY, LENGTH(CITY) AS NAME\_LENGTH FROM STATION ORDER BY LENGTH(CITY), CITY LIMIT 1; SELECT CITY, LENGTH(CITY) AS NAME\_LENGTH FROM STATION ORDER BY LENGTH(CITY) DESC, CITY LIMIT 1;

QUE 12) - Write a query that prints a list of employee names (i.e.: the name attribute) for employees in **Employee** having a salary greater than  per month who have been employees for less than  months. Sort your result by ascending employee\_id



Solution - SELECT NAME FROM EMPLOYEE WHERE SALARY > 2000 AND MONTHS <10 ORDER BY employee\_id;

13-01-2025

QUE NO 13 ) - Query the total population of all cities in CITY where *District* is California.

Input Format

The CITY table is described as follows:

****

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

QUE NO 14) - Query the average population of all cities in CITY where *District* is California.

Input Format

****

SOLUTION - select avg(population) from city where District = 'California';

QUE NO – 15) - Query the average population for all cities in CITY, rounded *down* to the nearest integer.

****

SOLUTION - select ROUND(avg(population)) from city;

QUE NO - 16) - Query the sum of the populations for all Japanese cities in CITY. The COUNTRYCODE for Japan is JPN.

****

SOLUTION - select sum(population) from city where COUNTRYCODE = 'JPN' ;

14-01-2025

QUE NO 17 ) - Query the difference between the maximum and minimum populations in CITY.

****

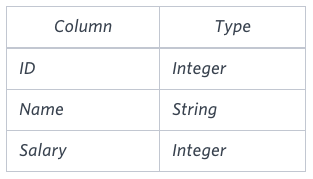
SOLUTION - select (MAX(population)-MIN(population)) from city;

QUE NO 18) - 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.

Input Format

The EMPLOYEES table is described as follows:

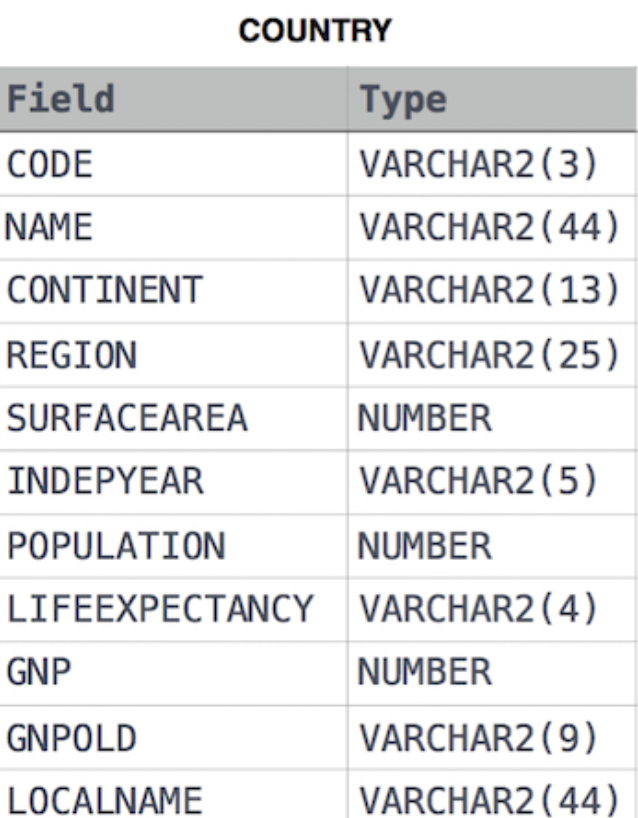
****

SOLUTION - SELECT CEIL( AVG(Salary) - AVG(REPLACE(Salary,'0','')) ) FROM EMPLOYEES;

QUE NO 19) - Given the CITY and COUNTRY tables, query the names of all cities where the CONTINENT is 'Africa'.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

The CITY and COUNTRY tables are described as follows:****

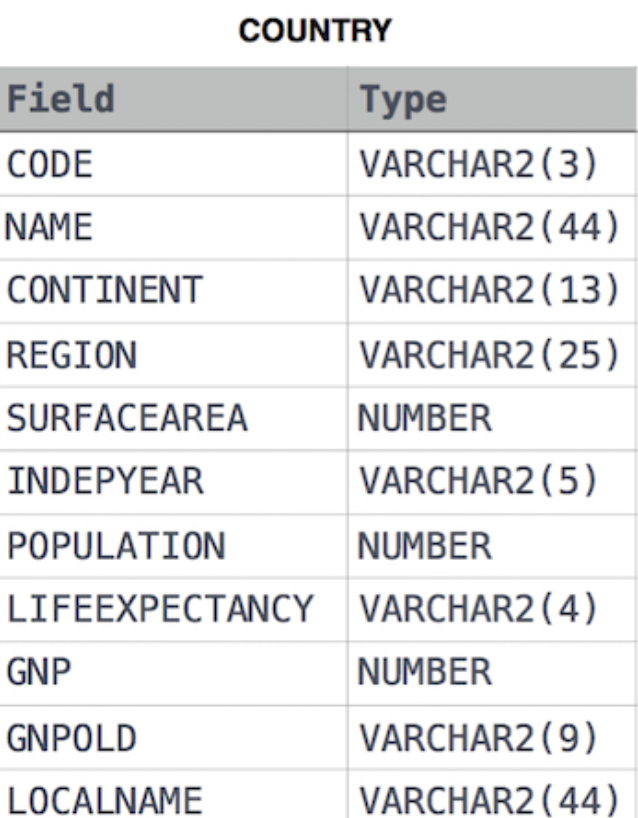
****

SOLUTION - select CITY.name from city as CITY INNER JOIN COUNTRY

ON CITY.CountryCode = COUNTRY.Code where Country.continent = 'Africa';

QUE NO – 20 ) - Given the CITY and COUNTRY tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

****

****

SOLUTION - SELECT COUNTRY.CONTINENT,FLOOR(AVG(CITY.POPULATION)) FROM CITY

INNER JOIN COUNTRY

ON CITY.CountryCode = COUNTRY.Code

GROUP BY COUNTRY.CONTINENT;