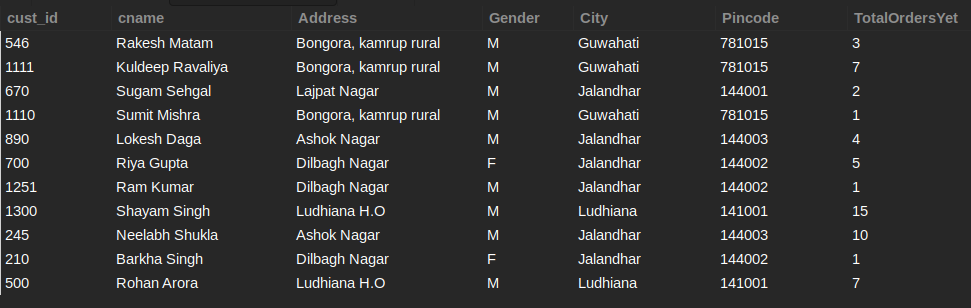
**Problem Statement:**  
List the number of customers from each city.

**Information about the table:**  
Table **Customer :**



**Output :**  
The result set should have 2 columns. The first column contains the city name and 2nd column should contain the corresponding number of customers. Rename the second column as Cust\_num.

Note: Write keywords and aggregate functions in uppercase alphabets.

select city, count(\*) as cust\_num from customer group by city

Console

 Download Dataset (db.sql)

+-----------+----------+

| city | cust\_num |

+-----------+----------+

| Guwahati | 3 |

| Jalandhar | 6 |

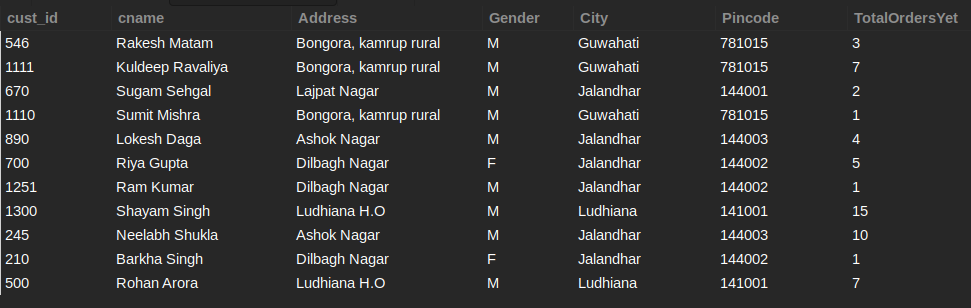
| Ludhiana | 2 |

+-----------+----------+

**SQL query - 5**

**Send Feedback**

**Problem Statement:**  
List out the total number of orders made to each address.

**Information about the table:**  
Table **Customer:**

**Output :**  
The result set should have the address and its corresponding sum of total orders made to that address.

Note: Write keywords and aggregate functions in uppercase alphabets.

select Address, sum(TotalOrdersYet) from customer group by Address

+-----------------------+---------------------+

| Address | sum(TotalOrdersYet) |

+-----------------------+---------------------+

| Bongora, kamrup rural | 11 |

| Lajpat Nagar | 2 |

| Ashok Nagar | 14 |

| Dilbagh Nagar | 7 |

| Ludhiana H.O | 22 |

+-----------------------+---------------------+

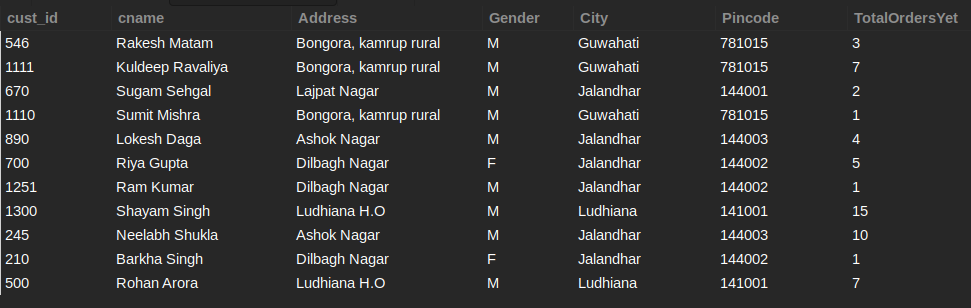
**SQL query - 9**

**Send Feedback**

**Problem Statement:**  
List the cities in descending order of the number of customers residing in them.

Note: Name the number of customers residing in them as "Number" using Alias Keyword.

**Information about the table:**  
Table **Customer:**



**Output :**  


Note: Write keywords and aggregate functions in uppercase alphabets.

select city, count(\*) as Number from Customer group by city order by count(city) desc

+-----------+--------+

| city | Number |

+-----------+--------+

| Jalandhar | 6 |

| Guwahati | 3 |

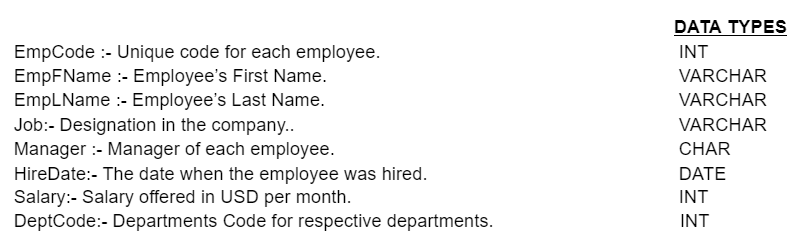
| Ludhiana | 2 |

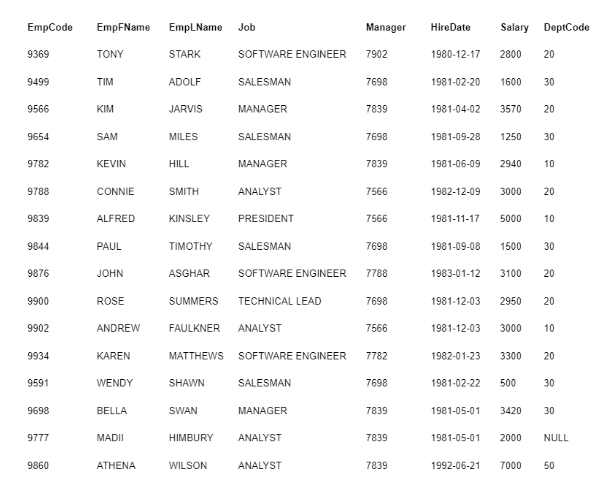
+-----------+--------+

**SQL query - 12**

**Send Feedback**

**Problem Statement:**  
List out the number of employees for each distinct role corresponding with their department code. Note: Name the number of employees as "number\_ of \_employees" using Alias Keyword.

**Information about the table:**  
Attributes list:   
Table **Employee\_data:**



**Output :**  


Note: Write keywords and aggregate functions in uppercase alphabets.

select job, DeptCode, count(\*) as number\_of\_employees from employee\_data group by job,DeptCode

+-------------------+----------+---------------------+

| job | DeptCode | number\_of\_employees |

+-------------------+----------+---------------------+

| SOFTWARE ENGINEER | 20 | 3 |

| SALESMAN | 30 | 4 |

| MANAGER | 20 | 1 |

| MANAGER | 10 | 1 |

| ANALYST | 20 | 1 |

| PRESIDENT | 10 | 1 |

| TECHNICAL LEAD | 20 | 1 |

| ANALYST | 10 | 1 |

| MANAGER | 30 | 1 |

| ANALYST | NULL | 1 |

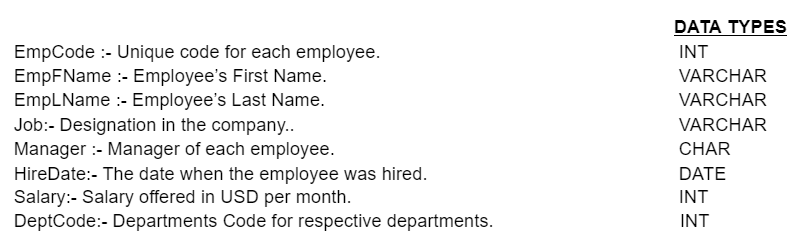
| ANALYST | 50 | 2 |

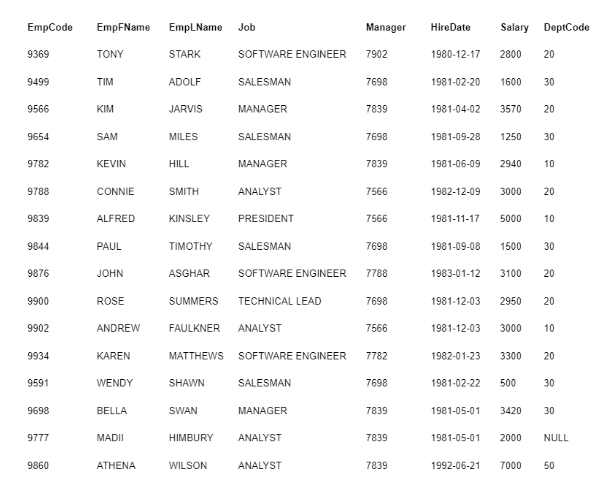
+-------------------+----------+---------------------+

**SQL query - 15**

**Send Feedback**

**Problem Statement:**  
List down the minimum salaries offered for each job role in each department, also list them in descending order based on the max salaries being offered for that role.

**Information about the table:**  
Attributes list:   
Table **Employee\_data:**



**Output :**  
The result set should have the job name, its department code and its corresponding minimum salary.

Note: Write keywords and aggregate functions in uppercase alphabets.

select job,deptcode,min(salary) from employee\_data group by deptcode,job order by max(salary) desc

| job | deptcode | min(salary) |

+-------------------+----------+-------------+

| ANALYST | 50 | 5000 |

| PRESIDENT | 10 | 5000 |

| MANAGER | 20 | 3570 |

| MANAGER | 30 | 3420 |

| SOFTWARE ENGINEER | 20 | 2800 |

| ANALYST | 20 | 3000 |

| ANALYST | 10 | 3000 |

| TECHNICAL LEAD | 20 | 2950 |

| MANAGER | 10 | 2940 |

| ANALYST | NULL | 2000 |

| SALESMAN | 30 | 500 |

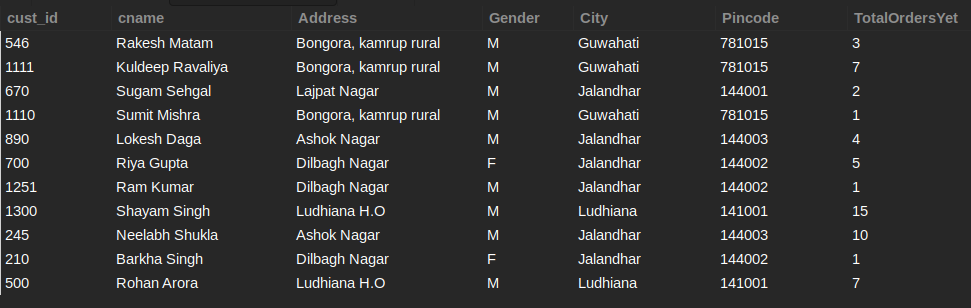
+-------------------+----------+-------------+

**SQL query - 17**

**Send Feedback**

**Problem Statement:**  
List down the addresses with the city and the pincode which appear more than twice in the table.

**Information about the table:**  
Table **Customer:**



**Output Table Structure:**  


Note: Write keywords and aggregate functions in uppercase alphabets.

+-----------------------+-----------+---------+

| address | city | pincode |

+-----------------------+-----------+---------+

| Bongora, kamrup rural | Guwahati | 781015 |

| Dilbagh Nagar | Jalandhar | 144002 |

+-----------------------+-----------+---------+

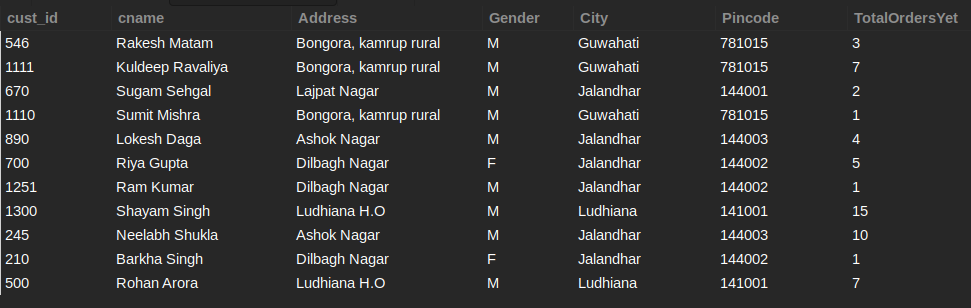
select address,city,pincode from customer group by pincode,address,city having count(\*)>2

**SQL query - 18**

**Send Feedback**

**Problem Statement:**  
List down all the addresses which belong to Guwahati and have made more than 7 orders in total.

**Information about the table:**  
Table **Customer:**



**Output Table Structure:**  


Note: Write keywords and aggregate functions in uppercase alphabets.

+-----------------------+

| address |

+-----------------------+

| Bongora, kamrup rural |

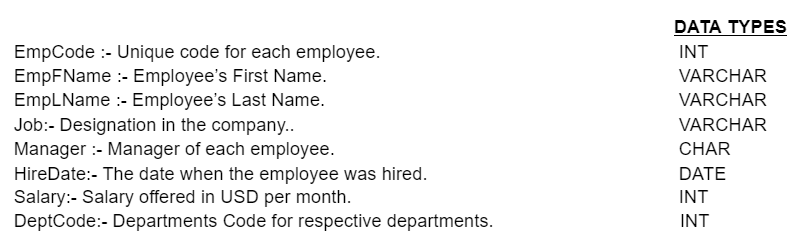
+-----------------------+

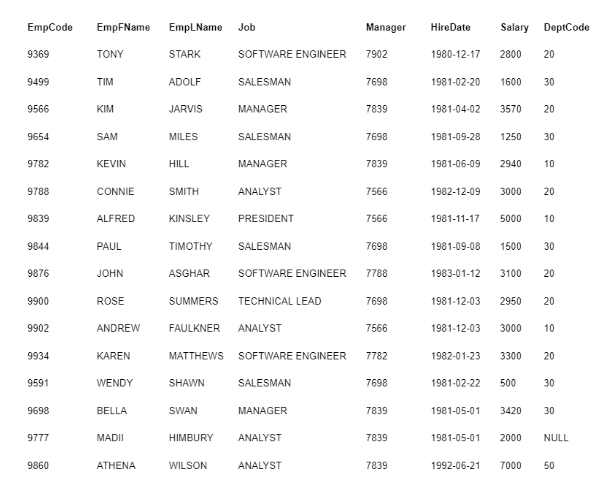
select address from customer where city = 'Guwahati' group by address having sum(TotalOrdersYet)>7

**SQL query - 21**

**Send Feedback**

**Problem Statement:**  
List down the managers handling more than 2 employees, and make sure those employees don’t belong to departments 10 and 20.

**Information about the table:**  
Attributes list:   
Table **Employee\_data:**



**Output :**  
The result set should have the manager's id and its corresponding number of employees.

Note: Write keywords and aggregate functions in uppercase alphabets.

+---------+----------------+

| manager | count(manager) |

+---------+----------------+

| 7698 | 4 |

| 7839 | 3 |

+---------+----------------+

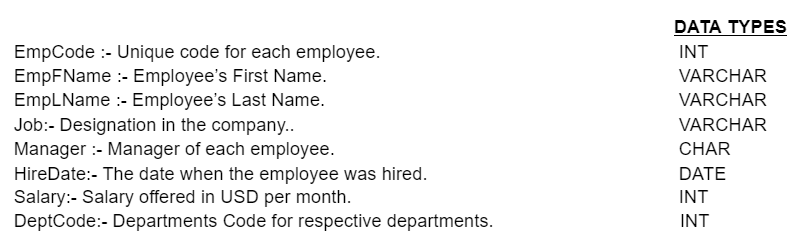
select manager,count(manager) from employee\_data

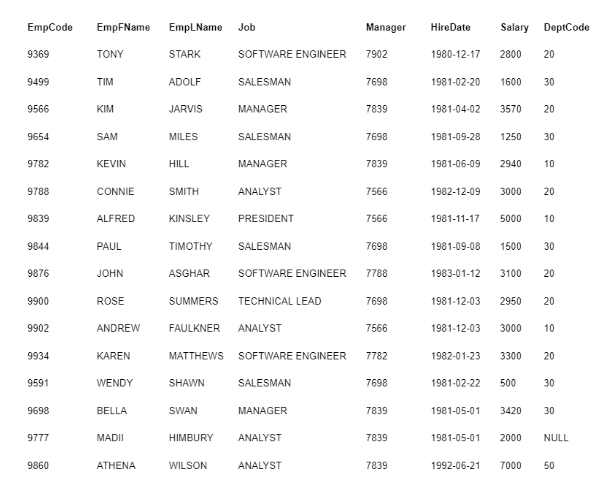
where deptcode not in (10,20) group by manager having count(manager)>2

**SQL query - 22**

**Send Feedback**

**Problem Statement:**  
For All the Analyst jobs list down the maximum salaries offered to them in different departments and under different managers, list all the details in ascending order based on the combined salary given out by that department.

**Information about the table:**  
Attributes list:   
Table **Employee\_data:**



**Output :**  
The result set should have the job name, its department code, its corresponding manager name and its corresponding maximum salary.

Note: Write keywords and aggregate functions in uppercase alphabets.

+---------+----------+---------+-------------+

| job | deptcode | manager | max(salary) |

+---------+----------+---------+-------------+

| ANALYST | NULL | 7839 | 2000 |

| ANALYST | 20 | 7566 | 3000 |

| ANALYST | 10 | 7566 | 3000 |

| ANALYST | 50 | 7839 | 7000 |

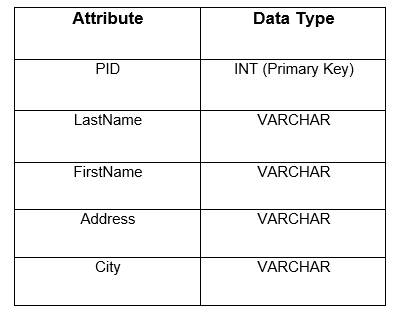
+---------+----------+---------+-------------+

select job,deptcode,manager, max(salary) from employee\_data where job = 'ANALYST'

group by job,deptcode,manager order by sum(salary)

**SQL Query - 1**

**Send Feedback**

**Problem Statement :**  
Write a query for creating a table named **People**, which contains information given in the table below: 

Print the Table schema once created as follows:

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'your\_table'

ORDER BY column\_name;

Note - 1: Replace 'your\_table' with the actual table name used in the query.

Note - 2: Write keywords of syntax in uppercase alphabets.

+------------+-------------+-----------+

| TABLE\_NAME | COLUMN\_NAME | DATA\_TYPE |

+------------+-------------+-----------+

| people | Address | varchar |

| people | City | varchar |

| people | FirstName | varchar |

| people | LastName | varchar |

| people | PID | int |

+------------+-------------+-----------+

CREATE TABLE People(

PID INT,

LastName VARCHAR(255),

FirstName VARCHAR(255),

Address VARCHAR(255),

City VARCHAR(255),

PRIMARY KEY(PID)

);

SELECT table\_name, column\_name, data\_type

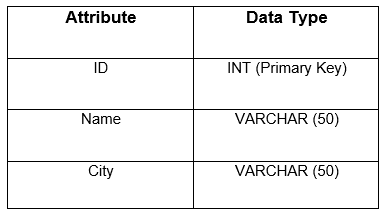
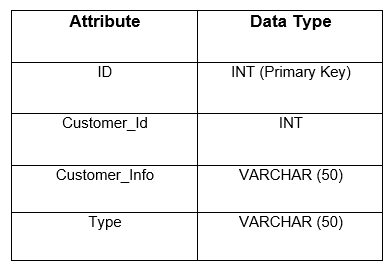
FROM information\_schema.columns

WHERE table\_name = 'People'

ORDER BY column\_name;

**SQL Query - 3**

**Send Feedback**

**Problem Statement:**  
  
Write multiple queries to create two tables named customer and contacts:  
  
1. The table **customer** contains features like ID, Name, City where all columns are never NULL. The ID here will be the primary key.  
  
  
2. The Table **contacts** contain features like ID, Customer\_Id, Customer\_info, Type. Customer\_ id being a foreign key that refers to ID(customer table). Also Customer\_info and Type are never NULL.  


Note: Describe both the tables as well. The syntax for describe is given below. Describe the Customer table first and then Contacts Table.

DESC <TABLE\_NAME>;

Note: Write keywords of syntax in uppercase alphabets.

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| Id | int | NO | PRI | NULL | |

| Name | varchar(50) | NO | | NULL | |

| City | varchar(50) | NO | | NULL | |

+-------+-------------+------+-----+---------+-------+

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| Id | int | NO | PRI | NULL | |

| Customer\_Id | int | YES | MUL | NULL | |

| Customer\_info | varchar(50) | NO | | NULL | |

| Type | varchar(50) | NO | | NULL | |

+---------------+-------------+------+-----+---------+-------+

CREATE TABLE customer(

Id INT,

Name VARCHAR(50) Not Null,

City VARCHAR(50) Not Null,

PRIMARY KEY(Id)

);

CREATE TABLE contacts(

Id INT,

Customer\_Id INT,

Customer\_info VARCHAR(50) Not Null,

Type VARCHAR(50) Not Null,

PRIMARY KEY(Id),

FOREIGN KEY(Customer\_Id) REFERENCES customer(Id)

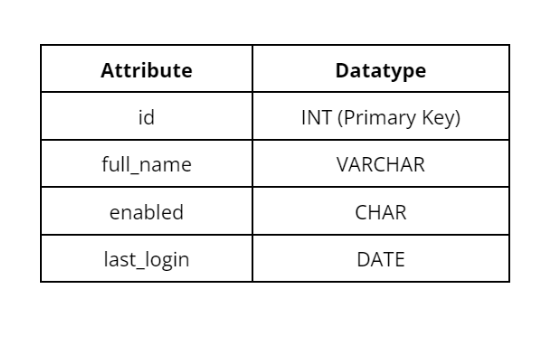
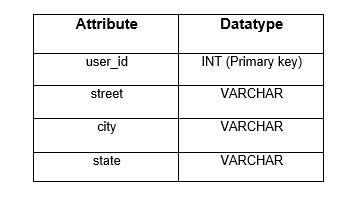
);

DESC customer;

DESC contacts;

**SQL Query - 4**

**Send Feedback**

**Problem Statement:**  
Consider the tables given below:  
  
1. The table **users** contains features like *id, full\_name, enabled, last\_login.*The attribute *id* here will be the *primary key*.  
Attribute List(s):   
  
2.The table **addresses** contains features like *user\_id, street, city, state*. The *user\_id* here will be the *primary key* as well as *foreign key* that refers to *id(users table)*. The attributes *street, city and state* should be declared *NOT NULL*.  
Attribute List(s):  
  
Formulate a SQL query to create these tables with all the constraints given along with the table. Also, add constraints to correlate these tables.  
  
Print the Table Schema for each table created as follows:

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'your\_table';

Note - 1: First print table **users** and then **addresses**. Position the above command just after the CREATE TABLE command for each table.

Note - 2: Replace 'your\_table' with the actual table name used in the query.

CREATE TABLE

table\_name | column\_name | data\_type

------------+-------------+-------------------

users | id | integer

users | full\_name | character varying

users | enabled | character

users | last\_login | date

(4 rows)

CREATE TABLE

table\_name | column\_name | data\_type

------------+-------------+-------------------

addresses | user\_id | integer

addresses | street | character varying

addresses | city | character varying

addresses | state | character varying

(4 rows)

CREATE TABLE users(

id INT,

full\_name VARCHAR(50),

enabled CHAR(50),

last\_login Date,

PRIMARY KEY(Id)

);

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'users';

CREATE TABLE addresses(

user\_id INT,

street VARCHAR(50) Not Null,

city VARCHAR(50) Not Null,

state VARCHAR(50) Not Null,

PRIMARY KEY(user\_id),

FOREIGN KEY(user\_id) REFERENCES users(id)

);

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'addresses';

**SQL Query - 6**

**Send Feedback**

**Problem Statement :**  
Given a table named **members**, write a query to add a column named cc\_number (Datatype - VARCHAR).

Note : Print the Table Schema once created as follows:

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'your\_table'

ORDER BY column\_name;

Note - 1: Replace 'your\_table' with the actual table name used in the query.

Note - 2: Write keywords of syntax in uppercase alphabets.

+------------+-------------+-----------+

| TABLE\_NAME | COLUMN\_NAME | DATA\_TYPE |

+------------+-------------+-----------+

| members | cc\_number | varchar |

| members | City | varchar |

| members | ID | int |

| members | mem\_Name | varchar |

| members | member\_dob | date |

| members | money | int |

+------------+-------------+-----------+

ALTER TABLE members ADD COLUMN cc\_number VARCHAR(50);

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'members'

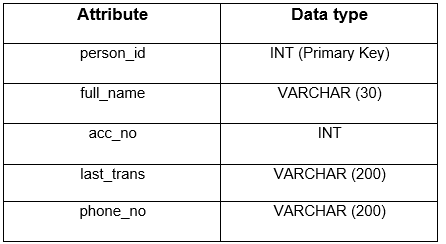
ORDER BY column\_name;

**SQL Query - 7**

**Send Feedback**

**Problem Statement :**  
Given a table named Bank, write a query to change the existing column person\_id to Pid VARCHAR(50).

table **Bank**



Syntax to describe the above table.

DESC <TABLE\_NAME>;

Note: Write keywords of syntax in uppercase alphabets.

+------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| Pid | varchar(50) | YES | | NULL | |

| full\_name | varchar(30) | YES | | NULL | |

| acc\_no | int | YES | | NULL | |

| last\_trans | varchar(200) | YES | | NULL | |

| phone\_no | varchar(200) | YES | | NULL | |

+------------+--------------+------+-----+---------+-------+

ALTER TABLE Bank

CHANGE COLUMN person\_id Pid VARCHAR(50);

ALTER TABLE members

DROP COLUMN member\_dob;

DESC Bank;

**SQL Query - 9**

**Send Feedback**

**Problem Statement :**  
Write a query to rename the Table consumers to Consumer\_Data.  
  
Print the Table Schema once created as follows:

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'your\_table'

ORDER BY column\_name;

Note - 1: Replace 'your\_table' with the actual table name used in the query.

Note - 2: Write keywords of syntax in uppercase alphabets.

+---------------+-------------+-----------+

| TABLE\_NAME | COLUMN\_NAME | DATA\_TYPE |

+---------------+-------------+-----------+

| consumer\_data | full\_name | varchar |

| consumer\_data | id | int |

| consumer\_data | last\_login | date |

| consumer\_data | phone\_no | varchar |

| consumer\_data | totalOrders | int |

+---------------+-------------+-----------+

ALTER TABLE Consumers

RENAME TO Consumer\_Data;

SELECT table\_name, column\_name, data\_type

FROM information\_schema.columns

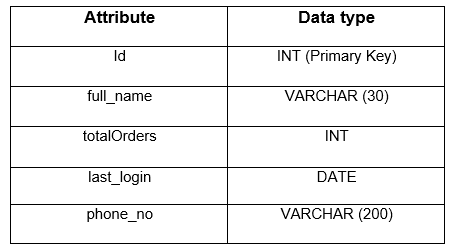
WHERE table\_name = 'Consumer\_Data'

ORDER BY column\_name;

**SQL Query - 10**

**Send Feedback**

**Problem Statement :**  
Write a SQL query to remove the attribute 'last\_login' and rename the 'full\_name' to 'customer\_name' in the customer table.  
  
Table **customer**



Syntax to describe the table.

DESC <TABLE\_NAME>;

Note: Write keywords of syntax in uppercase alphabets.

+---------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+--------------+------+-----+---------+-------+

| id | int | YES | | NULL | |

| customer\_name | varchar(30) | YES | | NULL | |

| totalOrders | int | YES | | NULL | |

| phone\_no | varchar(200) | YES | | NULL | |

+---------------+--------------+------+-----+---------+-------+

ALTER TABLE customer

drop COLUMN last\_login;

ALTER TABLE customer

CHANGE COLUMN full\_name customer\_name VARCHAR(30);

DESC customer;