1. Write a query in MySQL to perform the following operations:

a. Display CNAME using the HAVING clause to filter the result set to only those records of CNAME column that have a count of greater than two (i.e. HAVING COUNT (\*) >2).

**ANS:** **SELECT CNAME FROM customer  
GROUP BY CITY  
HAVING COUNT (C\_NO) > 2;**

b. Select all customers from the Customer table and group them by the CNAME.

**Ans: SELECT \* FROM customer**

**GROUP BY CNAME**

c. Select all records from Customer table, then orders them by the CNAME field in ascending order.

**Ans: SELECT \* FROM customer  
ORDER BY CNAME**

2. Explain about having, order by, group by clause with example.

# ANS: HAVING clause

The HAVING clause allows you to specify conditions on the rows for each group - in other words, which rows should be selected will be based on the conditions you specify. The HAVING clause should follow the GROUP BY clause if you are going to use it.

**HAVING** clause syntax:

SELECT column1,   
SUM(column2)  
  
FROM "list-of-tables"  
  
GROUP BY "column-list"  
  
HAVING "condition";

HAVING can best be described by example. Let's say you have an employee table containing the employee's name, department, salary, and age. If you would like to select the average salary for each employee in each department, you could enter:

SELECT dept, avg(salary)  
  
FROM employee  
  
GROUP BY dept;

But, let's say that you want to ONLY calculate & display the average if their salary is over 20000:

SELECT dept, avg(salary)  
  
FROM employee  
  
GROUP BY dept  
  
HAVING avg(salary) > 20000;

# ORDER BY clause

ORDER BY is an optional clause which will allow you to display the results of your query in a sorted order (either ascending order or descending order) based on the columns that you specify to order by.

**ORDER BY** clause syntax:

SELECT column1, SUM(column2) FROM "list-of-tables" ORDER BY "column-list" [ASC | DESC];

[ ] = optional

This statement will select the employee\_id, dept, name, age, and salary from the employee\_info table where the dept equals 'Sales' and will list the results in Ascending (default) order based on their Salary.

ASC = Ascending Order - default  
  
DESC = Descending Order

For example:

SELECT employee\_id, dept, name, age, salary FROM employee\_info WHERE dept = 'Sales' ORDER BY salary;

If you would like to order based on multiple columns, you must seperate the columns with commas. For example:

SELECT employee\_id, dept, name, age, salary  
  
  
FROM employee\_info  
  
WHERE dept = 'Sales'  
  
ORDER BY salary, age DESC;

# GROUP BY clause

The GROUP BY clause will gather all of the rows together that contain data in the specified column(s) and will allow aggregate functions to be performed on the one or more columns. This can best be explained by an example:

GROUP BY clause syntax:

SELECT column1,   
SUM(column2)  
   
FROM "list-of-tables"  
   
GROUP BY "column-list";

Let's say you would like to retrieve a list of the highest paid salaries in each dept:

SELECT max(salary), dept  
   
FROM employee   
   
GROUP BY dept;

This statement will select the maximum salary for the people in each unique department. Basically, the salary for the person who makes the most in each department will be displayed. Their, salary and their department will be returned.

3. Write a query in MySQL to perform the following operations-

a. Truncate all the records from Customer table.

**ANS: TRUNCATE TABLE Customer**

b. Rename the table ‘Account’ to ‘Customer\_Account’.

**ANS: RENAME TABLE Account to Customer\_Account**

c. Drop the table ‘Customer’.

**ANS: DROP TABLE Customer**

d. Drop the table ‘Customer\_Account’.

**ANS: DROP TABLE Customer\_Account**