ABES Engineering College, Ghaziabad
Department of Computer Science & Engineering

Session: 2016-17 **Semester: Vth A** **Student Name: Abhishek Chopra**
Subject Code: NCS-502 **Subject Name: DBMS** **Uni. Roll No. 1403210009**

Assignment 3: Queries using Group by & Having Clause

SCHEMA: S_ORD (CUS_ID, DATE_ORDER, DATE_SHIPMENT, SP_NO, TOTAL, PAYMENT)

```
+--------+------------+---------------+-------+------------+---------+
| CUS_id | date_order | date_shipment | sp_no | total      | Payment |
+--------+------------+---------------+-------+------------+---------+
|     97 | 1992-08-28 | 1992-09-17    |    12 |   84000.00 | Credit  |
|     98 | 1992-08-31 | 1992-09-10    |    14 |     595.00 | Cash    |
|     99 | 1992-08-31 | 1992-09-18    |    14 |    7707.00 | Credit  |
|    100 | 1992-08-31 | 1992-09-10    |    11 |  601100.00 | Credit  |
|    101 | 1992-08-31 | 1992-09-15    |    14 |    8056.60 | Credit  |
|    102 | 1992-09-01 | 1992-09-08    |    15 |    8335.00 | Credit  |
|    103 | 1992-09-02 | 1992-09-22    |    15 |     377.00 | Cash    |
|    104 | 1992-09-03 | 1992-09-23    |    15 |   32430.00 | Credit  |
|    105 | 1992-09-04 | 1992-09-18    |    11 |    2722.24 | Credit  |
|    106 | 1992-09-07 | 1992-09-15    |    12 |   15634.00 | Credit  |
|    107 | 1992-09-07 | 1992-09-21    |    15 |  142171.00 | Credit  |
|    108 | 1992-09-07 | 1992-09-10    |    13 |  149730.00 | Credit  |
|    109 | 1992-09-08 | 1992-09-28    |    11 | 1020935.00 | Credit  |
|    110 | 1992-09-09 | 1992-09-21    |    11 |    1539.13 | Cash    |
|    111 | 1992-09-09 | 1992-09-21    |    11 |    2770.00 | Cash    |
|    112 | 1992-08-31 | 1992-09-10    |    12 |     550.00 | Credit  |
+--------+------------+---------------+-------+------------+---------+
```

QUESTIONS:

1. Write SQL query that displays the total and average payments of all the credit orders.

```
mysql> select SUM(total),Avg(total) from S_ord where Payment='Credit';
+------------+---------------+
| SUM(total) | Avg(total)    |
+------------+---------------+
| 2073370.84 | 172780.903341 |
+------------+---------------+
1 row in set (0.10 sec)
```

2. Write SQL query that displays the total and average payments grouped by type of payment.

```
mysql> select payment,count(*),SUM(total),Avg(total) from S_ord group by payment;
+---------+----------+------------+---------------+
| payment | count(*) | SUM(total) | Avg(total)    |
+---------+----------+------------+---------------+
| Cash    |        4 |    5281.13 |   1320.282501 |
| Credit  |       12 | 2073370.84 | 172780.903341 |
+---------+----------+------------+---------------+
2 rows in set (0.04 sec)
```

3. How many order dates are represented compared to the total number of orders?

```
mysql> select count(distinct date_order),count(*) as total_no_order from s_ord;
+----------------------------+----------------+
| count(distinct date_order) | total_no_order |
+----------------------------+----------------+
|                          9 |             16 |
+----------------------------+----------------+
1 row in set (0.06 sec)
```

4. How many customers and sales representative are represented compared to the total number of orders?

```
mysql> select count(distinct cus_id) as Customer,count(distinct sp_no) as Sales_rep,count(*)  as total_no_order from s_ord;
+----------+----------+----------------+
| Customer | Sales_rep | total_no_order |
+----------+----------+----------------+
|       16 |        5 |             16 |
+----------+----------+----------------+
1 row in set (0.00 sec)
```

5. Write SQL query that displays the lowest and highest payments of all the orders.

```
mysql> select Max(total) as Highest_Payment,min(total) as Lowest_Payment from S_ord ;
+-----------------+----------------+
| Highest_Payment | Lowest_Payment |
+-----------------+----------------+
|      1020935.00 |         377.00 |
+-----------------+----------------+
1 row in set (0.06 sec)
```

6. What is the average amount of the order for each sales representative?

```
mysql> select sp_no,avg(total) from S_ord group by SP_no;
+-------+--------------+
| sp_no | avg(total)   |
+-------+--------------+
|    11 | 325813.273999 |
|    12 |  33394.666667 |
|    13 | 149730.000000 |
|    14 |   5452.866699 |
|    15 |  45828.250000 |
+-------+--------------+
5 rows in set (0.00 sec)
```

7. Write an SQL query to display the order dates and how many orders were on each date.

```
mysql> select Date_order,count(*) from s_ord group by Date_order;
+------------+----------+
| Date_order | count(*) |
+------------+----------+
| 1992-08-28 |        1 |
| 1992-08-31 |        5 |
| 1992-09-01 |        1 |
| 1992-09-02 |        1 |
| 1992-09-03 |        1 |
| 1992-09-04 |        1 |
| 1992-09-07 |        3 |
| 1992-09-08 |        1 |
| 1992-09-09 |        2 |
+------------+----------+
9 rows in set (0.00 sec)
```

8. Write SQL query to display the order amount by payment type for each sales representative.

```
mysql> Select Sp_no ,Payment, Total from s_ord group by SP_no, payment;
+-------+---------+-----------+
| Sp_no | Payment | Total     |
+-------+---------+-----------+
|    11 | Cash    |   1539.13 |
|    11 | Credit  | 601100.00 |
|    12 | Credit  |  84000.00 |
|    13 | Credit  | 149730.00 |
|    14 | Cash    |    595.00 |
|    14 | Credit  |   7707.00 |
|    15 | Cash    |    377.00 |
|    15 | Credit  |   8335.00 |
+-------+---------+-----------+
8 rows in set (0.00 sec)
```

9. Query to display the highest and lowest order for each order date where more than one order was placed.

```
mysql> select date_order,min(total),max(total) from s_ord group by date_order having count(date_order)>1;
+------------+------------+------------+
| date_order | min(total) | max(total) |
+------------+------------+------------+
| 1992-08-31 |     550.00 |  601100.00 |
| 1992-09-07 |   15634.00 |  149730.00 |
| 1992-09-09 |    1539.13 |    2770.00 |
+------------+------------+------------+
3 rows in set (0.05 sec)
```

10. SQL query to display the average order for each order date where more than one order was placed and the average order is greater than 1000. Display them in order of average order.

```
mysql> select date_order,avg(total) from s_ord group by date_order having count(date_order)>1 AND avg(total)>1000 order by avg(total);
+------------+--------------+
| date_order | avg(total)   |
+------------+--------------+
| 1992-09-09 |   2154.565002 |
| 1992-09-07 | 102511.666667 |
| 1992-08-31 | 123601.720020 |
+------------+--------------+
3 rows in set (0.04 sec)
```

11. Display the customer number with more than one order. Arrange alphabetically by customer id.

```
mysql> select cus_id from s_ord group by cus_id having count(*)>1 order by cus_id;
Empty set (0.04 sec)
```

```
mysql> select dname from dept1 where dnumber in (select dno from emp1 where ssn=7469);
+--------+
| dname |
+--------+
| CS    |
+--------+
1 row in set (0.00 sec)
```

```
mysql> select Fname,Lname,salary from emp1 where dno in (select distinct dno from emp1 where lname='goel');
+---------+-------+-----------+
| Fname   | Lname | salary    |
+---------+-------+-----------+
| Akansha | Goel  | 500000.00 |
+---------+-------+-----------+
1 row in set (0.00 sec)
```