CSLR 51 : DBMS LAB-2

Roll no. : **106119100**

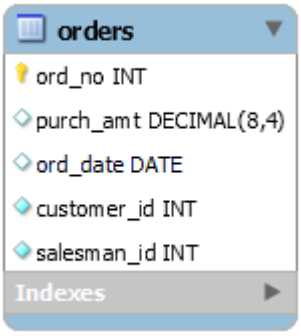
Name : **Rajneesh Pandey**

Section : **CSE-B**

PROBLEM 1

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

ER Diagram :




```
CREATE DATABASE purchaseDB;


CREATE TABLE orders(
  ord_no int NOT NULL,
  purch_amt DECIMAL(8,4),
  ord_date DATE,
  customer_id int NOT NULL,
  salesman_id int NOT NULL,
  PRIMARY KEY(ord_no)
  );
```

```
INSERT INTO orders(ord_no,purch_amt,ord_date, customer_id,salesman_id)
VALUES
(70001, 150.5, '2012-10-05', 3005, 5002),
(70009, 270.65, '2012-09-10', 3001, 5005),
(70002, 65.26, '2012-10-05', 3002, 5001),
(70004, 110.5, '2012-08-17', 3009, 5003),
(70007, 948.5, '2012-09-10', 3005, 5002),
(70005, 2400.6, '2012-07-27', 3007, 5001),
(70008, 5760, '2012-09-10', 3002, 5001),
(70010, 1983.43, '2012-10-10', 3004, 5006),
(70003, 2480.4, '2012-10-10', 3009, 5003),
(70012, 250.45, '2012-06-27', 3008, 5002),
(70011, 75.29, '2012-08-17', 3003, 5007),
(70013, 3045.6, '2012-04-25', 3002, 5001);
```

```
/* Queries */

SELECT * FROM orders;
```

```
MySQL localhost:3306 ssl SQL > use purchaseDB
Default schema set to `purchaseDB`.
Fetching table and column names from `purchasedb` for auto-completion... Press ^C to stop.
MySQL localhost:3306 ssl purchasedb SQL > SELECT * FROM orders;
+--------+-----------+------------+-------------+-------------+
| ord_no | purch_amt | ord_date   | customer_id | salesman_id |
+--------+-----------+------------+-------------+-------------+
|  70001 |  150.5000 | 2012-10-05 |        3005 |        5002 |
|  70002 |   65.2600 | 2012-10-05 |        3002 |        5001 |
|  70003 | 2480.4000 | 2012-10-10 |        3009 |        5003 |
|  70004 |  110.5000 | 2012-08-17 |        3009 |        5003 |
|  70005 | 2400.6000 | 2012-07-27 |        3007 |        5001 |
|  70007 |  948.5000 | 2012-09-10 |        3005 |        5002 |
|  70008 | 5760.0000 | 2012-09-10 |        3002 |        5001 |
|  70009 |  270.6500 | 2012-09-10 |        3001 |        5005 |
|  70010 | 1983.4300 | 2012-10-10 |        3004 |        5006 |
|  70011 |   75.2900 | 2012-08-17 |        3003 |        5007 |
|  70012 |  250.4500 | 2012-06-27 |        3008 |        5002 |
|  70013 | 3045.6000 | 2012-04-25 |        3002 |        5001 |
+--------+-----------+------------+-------------+-------------+
12 rows in set (0.0030 sec)
```

```
/* 1. From the given table, Write a SQL statement to get the minimum
purchase amount of all the orders. */
```

```sql
SELECT MIN(purch_amt)
FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT MIN(purch_amt)
                                           -> FROM orders;
+----------------+
| MIN(purch_amt) |
+----------------+
|        65.2600 |
+----------------+
1 row in set (0.0017 sec)
```

```sql
/* 2. From the given table, Write a SQL statement to get the maximum
purchase amount of all the orders. */

SELECT MAX(purch_amt)
FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT MAX(purch_amt)
                                           -> FROM orders;
+----------------+
| MAX(purch_amt) |
+----------------+
|      5760.0000 |
+----------------+
1 row in set (0.0008 sec)
```

```sql
/* 3. From the given table, Write a SQL statement to get the total purchase
amount of all the orders.*/

SELECT SUM(purch_amt)
FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT SUM(purch_amt)
                                           -> FROM orders;
+----------------+
| SUM(purch_amt) |
+----------------+
|     17541.1800 |
+----------------+
1 row in set (0.0006 sec)
```

```sql
/* 4. From the given table,Write a SQL statement to get the average
purchase amount of all the orders. */
```

```sql
SELECT AVG(purch_amt)
FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT AVG(purch_amt)
                                           -> FROM orders;
+----------------+
| AVG(purch_amt) |
+----------------+
|  1461.76500000 |
+----------------+
1 row in set (0.0012 sec)
```

```sql
/* 5. From the given table, write a SQL query to count the number of
customers. Return number of customers. */

SELECT COUNT(*)
FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT COUNT(*)
                                           -> FROM orders;
+----------+
| COUNT(*) |
+----------+
|       12 |
+----------+
1 row in set (0.0015 sec)
```

```sql
/* 6. In the above table, add a new column “warranty” which is six months
from the date of order.*/

ALTER TABLE orders
ADD warranty DATE default (adddate(ord_date, interval 6 month));
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > ALTER TABLE orders
                                           -> ADD warranty DATE default (adddate(ord_date, interval 6 month));
Query OK, 12 rows affected (0.1864 sec)

Records: 12  Duplicates: 0  Warnings: 0
```

```sql
SELECT * FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT * FROM orders;
+--------+----------+------------+-------------+-------------+------------+
| ord_no | purch_amt | ord_date   | customer_id | salesman_id | warranty   |
+--------+----------+------------+-------------+-------------+------------+
|  70001 |  150.5000 | 2012-10-05 |        3005 |        5002 | 2013-04-05 |
|  70002 |   65.2600 | 2012-10-05 |        3002 |        5001 | 2013-04-05 |
|  70003 | 2480.4000 | 2012-10-10 |        3009 |        5003 | 2013-04-10 |
|  70004 |  110.5000 | 2012-08-17 |        3009 |        5003 | 2013-02-17 |
|  70005 | 2400.6000 | 2012-07-27 |        3007 |        5001 | 2013-01-27 |
|  70007 |  948.5000 | 2012-09-10 |        3005 |        5002 | 2013-03-10 |
|  70008 | 5760.0000 | 2012-09-10 |        3002 |        5001 | 2013-03-10 |
|  70009 |  270.6500 | 2012-09-10 |        3001 |        5005 | 2013-03-10 |
|  70010 | 1983.4300 | 2012-10-10 |        3004 |        5006 | 2013-04-10 |
|  70011 |   75.2900 | 2012-08-17 |        3003 |        5007 | 2013-02-17 |
|  70012 |  250.4500 | 2012-06-27 |        3008 |        5002 | 2012-12-27 |
|  70013 | 3045.6000 | 2012-04-25 |        3002 |        5001 | 2012-10-25 |
+--------+----------+------------+-------------+-------------+------------+
12 rows in set (0.0037 sec)
```

```
/* 7. From the given table, Find out the most recently purchased order and
least recently purchased order using the ord_date.*/

SELECT MIN(ord_date) AS LEAST_RECENTLY_USED, MAX(ord_date)
AS MOST_RECENTLY_USED from orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT MIN(ord_date) AS LEAST_RECENTLY_USED, MAX(ord_date)
                                         -> AS MOST_RECENTLY_USED from orders;
+--------------------+--------------------+
| LEAST_RECENTLY_USED | MOST_RECENTLY_USED |
+--------------------+--------------------+
| 2012-04-25         | 2012-10-10         |
+--------------------+--------------------+
1 row in set (0.0005 sec)
```

```
/* 8. From the given table, find and display the next day of order.*/

SELECT ord_no,ADDDATE(ord_date,INTERVAL 1 DAY)
AS NEXT_DAY FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT ord_no,ADDDATE(ord_date,INTERVAL 1 DAY)
                                          -> AS NEXT_DAY FROM orders;
+--------+------------+
| ord_no | NEXT_DAY   |
+--------+------------+
|  70001 | 2012-10-06 |
|  70002 | 2012-10-06 |
|  70003 | 2012-10-11 |
|  70004 | 2012-08-18 |
|  70005 | 2012-07-28 |
|  70007 | 2012-09-11 |
|  70008 | 2012-09-11 |
|  70009 | 2012-09-11 |
|  70010 | 2012-10-11 |
|  70011 | 2012-08-18 |
|  70012 | 2012-06-28 |
|  70013 | 2012-04-26 |
+--------+------------+
12 rows in set (0.0005 sec)
```

```
/* 9. From the given table, print the smallest and largest integer just smaller
and larger than the purchase amount.*/

SELECT ord_no, CEIL(purch_amt) AS JUST_BIGGER FROM orders;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > SELECT ord_no, CEIL(purch_amt) AS JUST_BIGGER FROM orders;
+--------+-------------+
| ord_no | JUST_BIGGER |
+--------+-------------+
|  70001 |         151 |
|  70002 |          66 |
|  70003 |        2481 |
|  70004 |         111 |
|  70005 |        2401 |
|  70007 |         949 |
|  70008 |        5760 |
|  70009 |         271 |
|  70010 |        1984 |
|  70011 |          76 |
|  70012 |         251 |
|  70013 |        3046 |
+--------+-------------+
12 rows in set (0.0006 sec)
```

```
SELECT ord_no, FLOOR(purch_amt) AS JUST_SMALLER FROM orders;
```

```
MySQL localhost:3306 ssl purchasedb SQL > SELECT ord_no, FLOOR(purch_amt) AS JUST_SMALLER FROM orders;
+--------+--------------+
| ord_no | JUST_SMALLER |
+--------+--------------+
|  70001 |          150 |
|  70002 |           65 |
|  70003 |         2480 |
|  70004 |          110 |
|  70005 |         2400 |
|  70007 |          948 |
|  70008 |         5760 |
|  70009 |          270 |
|  70010 |         1983 |
|  70011 |           75 |
|  70012 |          250 |
|  70013 |         3045 |
+--------+--------------+
12 rows in set (0.0006 sec)
```

```
/*10. Explore the round function with various parameter settings on the
column purchase amount.*/
- The ROUND() function rounds a number to a specified number of decimal places.
SELECT ord_no,ROUND(purch_amt, 3) AS ROUNDED_UPTO_2_DECIMAL FROM orders;
```

```
MySQL localhost:3306 ssl purchasedb SQL > SELECT ord_no,ROUND(purch_amt, 3) AS ROUNDED_UPTO_2_DECIMAL FROM orders;
+--------+-----------------------+
| ord_no | ROUNDED_UPTO_2_DECIMAL |
+--------+-----------------------+
|  70001 |                150.500 |
|  70002 |                 65.260 |
|  70003 |               2480.400 |
|  70004 |                110.500 |
|  70005 |               2400.600 |
|  70007 |                948.500 |
|  70008 |               5760.000 |
|  70009 |                270.650 |
|  70010 |               1983.430 |
|  70011 |                 75.290 |
|  70012 |                250.450 |
|  70013 |               3045.600 |
+--------+-----------------------+
12 rows in set (0.0005 sec)
```

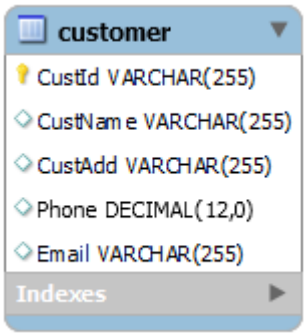
```
/* But If the 2nd Parameter is given Negative, then it rounds off upto Units Place,Tens.
SELECT ord_no,ROUND(purch_amt, -1) AS ROUNDED_UPTO_TENS FROM orders;
```

```
MySQL localhost:3306 ssl purchasedb SQL > SELECT ord_no,ROUND(purch_amt, -1) AS ROUNDED_UPTO_TENS FROM orders;
+--------+-------------------+
| ord_no | ROUNDED_UPTO_TENS |
+--------+-------------------+
|  70001 |               150 |
|  70002 |                70 |
|  70003 |              2480 |
|  70004 |               110 |
|  70005 |              2400 |
|  70007 |               950 |
|  70008 |              5760 |
|  70009 |               270 |
|  70010 |              1980 |
|  70011 |                80 |
|  70012 |               250 |
|  70013 |              3050 |
+--------+-------------------+
12 rows in set (0.0005 sec)
```

PROBLEM 2

CustId	CustName	CustAdd	Phone	Email
C0001	AmitSaha	L-10, Pitampura	4564587852	amitsaha2@gmail.com
C0002	Rehnuma	J-12, SAKET	5527688761	rehnuma@hotmail.com
C0003	CharviNayyar	10/9, FF, Rohini	6811635425	charvi123@yahoo.com
C0004	Gurpreet	A-10/2, SF, MayurVihar	3511056125	gur_singh@yahoo.com

ER Diagram :




```
CREATE DATABASE customerDB;

CREATE TABLE customer(
CustId varchar(255) NOT NULL,
CustName varchar(255),
CustAdd varchar(255),
Phone DECIMAL(12),
Email varchar(255),
PRIMARY KEY(CustId)
);

INSERT INTO customer (CustId,CustName ,CustAdd ,Phone ,Email )
 VALUES
( 'C0001' ,'AmitSaha' , 'L-10, Pitampura' , 4564587852 ,'amitsaha2@gmail.com'),
( 'C0002' ,'Rehnuma', 'J-12, SAKET' ,5527688761 ,'rehnuma@hotmail.com'),
( 'C0003' ,'CharviNayyar', '10/9, FF, Rohini' , 6811635425 ,'charvi123@yahoo.com'
),
( 'C0004' ,'Gurpreet' , 'A-
10/2, SF, MayurVihar', 3511056125 ,'gur_singh@yahoo.com');
```

```
/* Queries */

SELECT * FROM customer;
```

```
MySQL  localhost:3306 ssl  purchasedb  SQL > use customerDB
Default schema set to `customerDB`.
Fetching table and column names from `customerdb` for auto-completion... Press ^C to stop.
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT * FROM customer;
+--------+-------------+-----------------------+------------+--------------------+
| CustId | CustName    | CustAdd               | Phone      | Email              |
+--------+-------------+-----------------------+------------+--------------------+
| C0001  | AmitSaha    | L-10, Pitampura       | 4564587852 | amitsaha2@gmail.com |
| C0002  | Rehnuma     | J-12, SAKET           | 5527688761 | rehnuma@hotmail.com |
| C0003  | CharviNayyar | 10/9, FF, Rohini     | 6811635425 | charvi123@yahoo.com |
| C0004  | Gurpreet    | A-10/2, SF, MayurVihar | 3511056125 | gur_singh@yahoo.com |
+--------+-------------+-----------------------+------------+--------------------+
4 rows in set (0.0052 sec)
```

- 1. Display customer name in lower case and customer email in upper case from table CUSTOMER.

```
SELECT LOWER(CustName), UPPER(Email)
FROM customer;
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT LOWER(CustName), UPPER(Email)
                                          -> FROM customer;
+-----------------+---------------------+
| LOWER(CustName) | UPPER(Email)        |
+-----------------+---------------------+
| amitsaha        | AMITSAHA2@GMAIL.COM |
| rehnuma         | REHNUMA@HOTMAIL.COM |
| charvinayyar    | CHARVI123@YAHOO.COM |
| gurpreet        | GUR_SINGH@YAHOO.COM |
+-----------------+---------------------+
4 rows in set (0.0006 sec)
```

- 2. Display emails after removing the domain name extension “.com” from emails of the customers.

```
SELECT TRIM('.com' from Email) 'Email without .com'
FROM customer;
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT TRIM('.com' from Email) 'Email without .com'
                                          -> FROM customer;
+--------------------+
| Email without .com |
+--------------------+
| amitsaha2@gmail    |
| rehnuma@hotmail    |
| charvi123@yahoo    |
| gur_singh@yahoo    |
+--------------------+
4 rows in set (0.0007 sec)
```

```sql
-- 3. Display the length of the customer name.

SELECT CustName, LENGTH(CustName)  'Length of Names'
FROM customer;
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT CustName, LENGTH(CustName)  'Length of Names'
                                           -> FROM customer;
+--------------+------------------+
| CustName     | Length of Names |
+--------------+------------------+
| AmitSaha     |                8 |
| Rehnuma      |                7 |
| CharviNayyar |               12 |
| Gurpreet     |                8 |
+--------------+------------------+
4 rows in set (0.0016 sec)
```

```sql
-
- 4. Display the custid and CustName joined together and the numeric position of the letter A in the Customer name.

SELECT CONCAT(CustID,CustName) AS ID_AND_NAMES, LOCATE ('a',CustName) 'Position'
FROM customer;
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT CONCAT(CustID,CustName) AS ID_AND_NAMES, LOCATE ('a',CustName) 'Position'
                                           -> FROM customer;
+------------------+----------+
| ID_AND_NAMES     | Position |
+------------------+----------+
| C0001AmitSaha    |        1 |
| C0002Rehnuma     |        7 |
| C0003CharviNayyar |        3 |
| C0004Gurpreet    |        0 |
+------------------+----------+
4 rows in set (0.0007 sec)
```

```sql
-
- 5. Write a SQL statement to display the data for those customers whose names end with ‘a’.

SELECT * FROM customer WHERE CustName LIKE '%a';
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT * FROM customer WHERE CustName LIKE '%a';
+--------+----------+-----------------+------------+---------------------+
| CustId | CustName | CustAdd         | Phone      | Email               |
+--------+----------+-----------------+------------+---------------------+
| C0001  | AmitSaha | L-10, Pitampura | 4564587852 | amitsaha2@gmail.com |
| C0002  | Rehnuma  | J-12, SAKET     | 5527688761 | rehnuma@hotmail.com |
+--------+----------+-----------------+------------+---------------------+
2 rows in set (0.0006 sec)
```

```
- 6. Extract a substring from email starting at position 2 and 3 characters long.

SELECT SUBSTRING(Email,2,3) 'Substring 3 len'
FROM customer;
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT SUBSTRING(Email,2,3) 'Substring 3 len'
                                          -> FROM customer;
+-----------------+
| Substring 3 len |
+-----------------+
| mit             |
| ehn             |
| har             |
| ur_             |
+-----------------+
4 rows in set (0.0005 sec)
```

```
-- 7. Extract 20 character string from Customer Address starting with position 1.

SELECT SUBSTRING(CustAdd,1,20) 'Address 20 chars'
FROM customer;
```

```
MySQL  localhost:3306 ssl  customerdb  SQL > SELECT SUBSTRING(CustAdd,1,20) 'Address 20 chars'
                                          -> FROM customer;
+----------------------+
| Address 20 chars     |
+----------------------+
| L-10, Pitampura      |
| J-12, SAKET          |
| 10/9, FF, Rohini     |
| A-10/2, SF, MayurVih |
+----------------------+
4 rows in set (0.0005 sec)
```