Name: Vishal Shashikant Salvi.
UID: 2019230069
Batch: C
Class: SE Comps

Experiment No. 4

Aim: To practise Data Retrieval and Date Commands on Constraints Table

---

EXERCISE ON COMPUTATIONS IN TABLE DATA

1)Find the names of all clients having 'a' as the second letter in their names.

```
mysql> select name from client_master where name like '_a%';
+-----------------+
| name            |
+-----------------+
| Vandana Saitwal |
| Basu Navindgi   |
| Ravi Sreedharan |
+-----------------+
     3    rows in set (0.00 sec)
```

2)Find out the clients who stay in a city whose second letter is 'a'.

```
mysql> select city from client_master where city like '_a%';
+--------+
| city   |
+--------+
| Madras |
+--------+
     1    row in set (0.00 sec)
```

3)Find the list of all clients who stay in 'Bombay' or 'Delhi'

```
mysql> select * from client_master where city='Bombay' or city='Delhi';
+----------+-----------------+--------+---------+-------------+----------+
| client_no | name           | city   | pincode | state       | bal_due  |
+----------+-----------------+--------+---------+-------------+----------+
| C00001   | Ivan Bayross    | Bombay | 400054  | Maharashtra | 15000.00 |
| C00003   | Pramada Jasgute | Bombay | 400057  | Maharashtra | 5000.00  |
| C00004   | Basu Navindgi   | Bombay | 400056  | Maharashtra |     0.00 |
| C00005   | Ravi Sreedharan | Delhi  | 100001  | Delhi       | 2000.00  |
| C00006   | Rukmini         | Bombay | 400050  | Maharashtra |     0.00 |
+----------+-----------------+--------+---------+-------------+----------+
     4    rows in set (0.00 sec)
```

4)Print the list of clients whose bat due is greater than value 10000.

mysql> select * from client_master where bal_due > 10000;

```
----------+-------------+-------+--------+-------------+---------+
| client_no | name        | city | pincode | state       | bal_due |
+---------- +-------------+-------+--------+-------------+ ---------+
| C00001   | Ivan Bayross | Bombay | 400054 | Maharashtra | 15000.00 |
+---------- +-------------+-------+--------+-------------+ ---------+
```

5)Print the information from sales order table for orders placed in the month of January.

mysql> select * from sales_order where month(order_date)=1;

```
+----------+------------+----------+-------------+-----------+-----------+------------+--------------+
| order_no | order_date | client_no | salesman_no | dely_type | billed_yn | dely_date | order_status |
+ ----------+ ------------+ ----------+------------ +----------- +---------- +----------- +------------- +
| O19001   | 1996-01-12 | C00001   | S00001   | F      | N      | 1996-01-20 | In Process |
| O19002   | 1996-01-25 | C00002   | S00002   | P      | N      | 1996-01-27 | Cancelled |
+ ----------+ ------------+ ----------+------------ +----------- +---------- +----------- +------------- +
2 rows in set (0.00 sec)
```

6)Display the order information for client_no 'CO0001' and 'C00002'.

mysql> select * from sales_order where client_no='C00001' or client_no='C00002';

```
+----------+------------+----------+-------------+-----------+----------+-----------+--------------+
| order_no | order_date | client_no | salesman_no | dely_type | billed_yn | dely_date | order_status |
+ ----------+ ------------+ ----------+------------ +----------- +---------- +----------- +------------- +
| O19001   | 1996-01-12 | C00001   | S00001   | F      | N      | 1996-01-20 | In Process |
| O19002   | 1996-01-25 | C00002   | S00002   | P      | N      | 1996-01-27 | Cancelled |
| O19003   | 1996-04-03 | C00001   | S00001   | F      | Y      | 1996-04-07 | Fulfilled |
+ ----------+ ------------+ ----------+------------ +----------- +---------- +----------- +------------- +
3 rows in set (0.01 sec)
```

7)Find products whose selling price is greater than 2000 and less than or equal to 5000.

mysql> select * from product_master where sell_price > 2000 and sell_price <= 5000;

```
+------------+-------------+---------------+--------------+-------------+-------------+-----------+------------+
| product_no | description | profit_percent | unit_measure | qty_on_hand | reorder_lvl | sell_price |
cost_price |
+------------ +-------------+--------------- +-------------- +------------ +------------- +----------+----------      +
| P07868    | Keyboards |    2.00 | Piece   |    10 |    3 |  3150.00 |  3050.00 |
+------------ +-------------+--------------- +-------------- +------------ +------------- +----------+----------      +
1 row in set (0.00 sec)
```

8)Find products whose selling price is more than 1500. Calculate a new selling price as, original selling price * .15. Rename the new column in the above query as new_price.

mysql> alter table product_master add column new_price int;
Query OK, 0 rows affected (0.07 sec) Records: 0 Duplicates: 0
Warnings: 0

mysql> update product_master set new_price=1.15*sell_price where sell_price>1500;
Query OK, 4 rows affected (0.01 sec)
Rows matched: 4 Changed: 4 Warnings: 0

mysql> select * from product_master;

```
+------------+-------------+---------------+--------------+-------------+-------------+------------+-----------+----------+
| product_no | description | profit_percent | unit_measure | qty_on_hand | reorder_lvl | sell_price |
cost_price | new_price |
+------------+-------------+---------------+--------------+-------------+-------------+------------+-----------+----------+
```
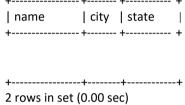
P00001	1.44 Floppies	5.00	Piece	100	20	525.00	500.00	NULL
P03453	Monitors	6.00	Piece	10	3	12000.00	11280.00	13800
P06734	Mouse	5.00	Piece	20	5	1050.00	1000.00	NULL
								NULL
P07865	1.22 Floppies	5.00	Piece	100	20	525.00	500.00	
P07868	Keyboards	2.00	Piece	10	3	3150.00	3050.00	3622
P07885	CD Drive	2.50	Piece	10	3	5250.00	5100.00	6037
P07965	540 HDD	4.00	Piece	10	3	8400.00	8000.00	9660
P07975	1.44 Drive	5.00	Piece	10	3	1050.00	1000.00	NULL
P08865	1.22 Drive	5.00	Piece	2	3	1050.00	1000.00	NULL
+----------- -------------+------------   +------------- +---------- +---------- ----------+------
-         +   --- -   -- ---   +-----   +-----------   +
     9   rows in set (0.00 sec)

9)List the names, city and state of clients who are not in the state of 'Maharashtra'.

mysql> select name,city,state from client_master where state!='Maharashtra';
+---------------- +------- +----------- +
| name | city | state |
+---------------- +------- +----------- +

+----------------+--------+-----------+
2 rows in set (0.00 sec)

10)Count the total number of orders.

mysql> select count(order_no) from sales_order;
+-----------------+
| count(order_no) |
+-----------------+
| 6 |
+-----------------+
     1   row in set (0.00 sec)

11)Calculate the average price of al! the products.

mysql> select avg(sell_price) from product_master;
+-----------------+
| avg(sell_price) |
+-----------------+
| 3666.666667 |
+-----------------+
     1   row in set (0.00 sec)

12)Determine the maximum and minimum product prices. Rename the output as max_price and min price respectively.
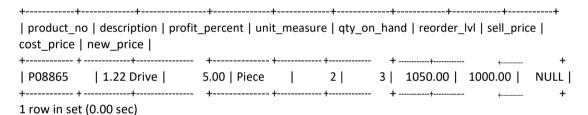
mysql> select max(sell_price) as max_price, min(sell_price) as min_price from product_master;
```
+-----------+-----------+
| max_price | min_price |
  ----------+--------
+              --    +
| 12000.00 |     525.00 |
  ----------+--------
+              --    +
    1    row in set (0.00 sec)
```
13)Count the number of products having price greater than or equal to 1500.

mysql> select count(product_no) from product_master where sell_price>=1500;
```
+-------------------+
| count(product_no) |
+-------------------+
|          4 |
+-------------------+
    1    row in set (0.00 sec)
```
14)Find all the products whose qty_on_hand is less than reorder level..

mysql> select * from product_master where qty_on_hand < reorder_lvl;

```
+------------+------------+---------------+--------------+-------------+------------+-----------+-----------+-----------+
| product_no | description | profit_percent | unit_measure | qty_on_hand | reorder_lvl | sell_price |
cost_price | new_price |
+------------ +------------+---------------  +-------------+------------+------------  +----------+----------     +--------         +
| P08865      | 1.22 Drive |          5.00 | Piece       |         2 |        3 |   1050.00 |   1000.00 |    NULL |
+------------ +------------+---------------  +-------------+------------+------------  +----------+----------     +--------         +
1 row in set (0.00 sec)
```

EXERCISE ON DATE MANIPULATION

1)Display the order number and day on which clients placed their order.

```
mysql> select order_no, order_date from sales_order;
+----------+------------+
| order_no | order_date |
+----------+------------+
| O19001   | 1996-01-12 |
| O19002   | 1996-01-25 |
| O19003   | 1996-04-03 |
| O19008   | 1996-05-24 |
| O46865   | 1996-02-18 |
| O46866   | 1996-05-20 |
+----------+------------+
     5   rows in set (0.00 sec)
```

2)Display the month (in alphabets) and date when the order must be delivered.

```
mysql> select monthname(order_date), order_date from sales_order;
+---------------------+------------+
| monthname(order_date) | order_date |
+---------------------
--                   +------------+
| January             | 1996-01-12 |
| January             | 1996-01-25 |
| April               | 1996-04-03 |
| May                 | 1996-05-24 |
| February            | 1996-02-18 |
| May                 | 1996-05-20 |
+---------------------   +------------+
     6   rows in set (0.00 sec)
```

3)Display the order_date in the format 'DD-Month-YY'. e.g. 12-February-96

```
mysql> select date_format(order_date,'%e-%M-%y') from sales_order;
+-----------------------------------+
| date_format(order_date,'%e-%M-%y') |
+-----------------------------------+
| 12-January-96           |
| 25-January-96           |
| 3-April-96              |
| 24-May-96               |
| 18-February-96          |
| 20-May-96               |
+-----------------------------------+
     7   rows in set (0.00 sec)
```

4)Find the date, 15 days after today's date.

```
mysql> select date_add(curdate(),interval 15 day);
+------------------------------------+
| date_add(curdate(),interval 15 day) |
+------------------------------------+
| 2020-04-02                 |
+------------------------------------+
     1   row in set (0.00 sec)
```

5)Find the number of days elapsed between today's date and the delivery date of the orders placed by the clients.

mysql> select datediff(curdate(), dely_date) from sales_order;
+--------------------------------+
| datediff(curdate(), dely_date) |

+   -+
| 8824 |
| 8817 |
| 8746 |
| 8697 |
| 8793 |
| 8701 |

+   -+
6 rows in set (0.00 sec)

Conclusion:

Thus from the given assignment we learn the various data computation and date functions in MySQl.