DBMS LAB-4

NAME:- ABHISHIKTH BODA

ROLL NUMBER:- S20210010044

DATE:- 07-09-2022

Exercise 1:-

#Create table employee with the following constraints;

CREATE TABLE employee(emp_id, emp_name ,emp_dept emp_age, place, income);

Set emp_id as the primary key with auto increment starting from 2505.

ANSWER:-

```
mysql> CREATE TABLE employee(emp_id int, emp_name varchar(20), emp_dept varchar(20),
    -> emp_age int, place varchar(20), income int, doj date);
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER TABLE employee
    -> add primary key(emp_id)
    -> ;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE employee auto_increment=2505;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Table after construction:-

```
mysql> select * from employee;
+--------+----------+------------+---------+------------+--------+------------+
| emp_id | emp_name | emp_dept   | emp_age | place      | income | doj        |
+--------+----------+------------+---------+------------+--------+------------+
|   2505 | peter    | Finance    |      32 | Newyork    | 100000 | 2002-08-25 |
|   2506 | Mark     | HR         |      32 | California | 120000 | 1980-03-25 |
|   2507 | Donald   | Finance    |      28 | Arizona    | 100000 | 1995-12-26 |
|   2508 | Obama    | Management |      35 | Florida    | 500000 | 1990-10-30 |
|   2509 | Linklon  | HR         |      25 | Georgia    |  25000 | 2008-08-08 |
|   2510 | Kane     | Sales      |      29 | Alaska     |  30000 | 2000-01-01 |
|   2511 | Adam     | Management |      38 | California | 540000 | 2020-10-25 |
|   2512 | Mac      | Finance    |      40 | Florida    | 280000 | 1970-06-09 |
|   2513 | Manas    | Accounts   |      29 | India      | 600000 | 1990-12-11 |
|   2514 | Vasin    | Accounts   |      30 | India      | 800000 | 1989-10-10 |
|   2515 | peter    | Finance    |      32 | Newyork    | 100000 | 1989-10-10 |
|   2516 | Mark     | HR         |      32 | California | 120000 | 1990-12-11 |
|   2517 | Donald   | Finance    |      28 | Arizona    | 100000 | 1970-06-09 |
|   2518 | Obama    | Management |      35 | Florida    | 500000 | 2020-10-25 |
|   2519 | Linklon  | HR         |      25 | Georgia    |  25000 | 2000-01-01 |
|   2520 | Kane     | Sales      |      29 | Alaska     |  30000 | 2008-08-08 |
|   2521 | Adam     | Management |      38 | California | 540000 | 1990-10-30 |
|   2522 | Mac      | Finance    |      40 | Florida    | 280000 | 1995-12-26 |
|   2523 | Manas    | Accounts   |      29 | India      | 600000 | 1980-03-25 |
|   2524 | Vasin    | Accounts   |      30 | India      | 800000 | 2002-08-25 |
+--------+----------+------------+---------+------------+--------+------------+
20 rows in set (0.00 sec)
```

Questionnaire set:-

1. Calculate the total number of employees name available in the table.

ANSWER:-

```
mysql> select count(emp_name) as no_of_emp from employee;
+-----------+
| no_of_emp |
+-----------+
|        20 |
+-----------+
1 row in set (0.01 sec)

mysql>
```

2. Display the maximum salary of each department and also all departments put together .

ANSWER:-

```
mysql> select emp_dept,max(income) from employee group by emp_dept;
+------------+-------------+
| emp_dept   | max(income) |
+------------+-------------+
| Finance    |      280000 |
| HR         |      120000 |
| Management |      540000 |
| Sales      |       30000 |
| Accounts   |      800000 |
+------------+-------------+
5 rows in set (0.00 sec)

mysql>
```

3.Find the employees whose salary is between 100000 and 500000 but not exactly 120000.

ANSWER:-

```
mysql> select emp_name,income from employee
    -> where income > 100000 and income < 500000 and not income = 120000;
+----------+--------+
| emp_name | income |
+----------+--------+
| Mac      | 280000 |
| Mac      | 280000 |
+----------+--------+
2 rows in set (0.00 sec)

mysql>
```

4.Get the count of employees whose income is more than 1 lakh.

ANSWER:-

```
mysql> select count(emp_name) from employee where income>100000;
+-----------------+
| count(emp_name) |
+-----------------+
|              12 |
+-----------------+
1 row in set (0.00 sec)

mysql>
```

5. List the employees according to ascending order of salary.

ANSWER:-

```
mysql> select emp_name,income from employee order by income asc;
+----------+--------+
| emp_name | income |
+----------+--------+
| Linklon  |  25000 |
| Linklon  |  25000 |
| Kane     |  30000 |
| Kane     |  30000 |
| peter    | 100000 |
| Donald   | 100000 |
| peter    | 100000 |
| Donald   | 100000 |
| Mark     | 120000 |
| Mark     | 120000 |
| Mac      | 280000 |
| Mac      | 280000 |
| Obama    | 500000 |
| Obama    | 500000 |
| Adam     | 540000 |
| Adam     | 540000 |
| Manas    | 600000 |
| Manas    | 600000 |
| Vasin    | 800000 |
| Vasin    | 800000 |
+----------+--------+
20 rows in set (0.00 sec)

mysql>
```

6. For each department, retrieve the department name, the number of employees in the department, and Maximum income for the department.

ANSWER:-

```
mysql> select emp_dept,count(emp_id),max(income) from employee group by emp_dept;
+------------+---------------+-------------+
| emp_dept   | count(emp_id) | max(income) |
+------------+---------------+-------------+
| Finance    |             6 |      280000 |
| HR         |             4 |      120000 |
| Management |             4 |      540000 |
| Sales      |             2 |       30000 |
| Accounts   |             4 |      800000 |
+------------+---------------+-------------+
5 rows in set (0.00 sec)

mysql>
```

7. List the number of employees in each place.

ANSWER:-

```
mysql> select count(emp_id),place from employee group by place;
+---------------+------------+
| count(emp_id) | place      |
+---------------+------------+
|             2 | Newyork    |
|             4 | California |
|             2 | Arizona    |
|             4 | Florida    |
|             2 | Georgia    |
|             2 | Alaska     |
|             4 | India      |
+---------------+------------+
7 rows in set (0.00 sec)

mysql>
```

8. List the number of employee in each country sorted high to low.

ANSWER:-

```
mysql> select count(emp_id),place from employee group by place order by count(emp_id) desc;
+---------------+------------+
| count(emp_id) | place      |
+---------------+------------+
|             4 | California |
|             4 | Florida    |
|             4 | India      |
|             2 | Newyork    |
|             2 | Arizona    |
|             2 | Georgia    |
|             2 | Alaska     |
+---------------+------------+
7 rows in set (0.00 sec)

mysql>
```

9. List the number of employees in each place. (Only include places with more than 1 employee).

ANSWER:-

```
mysql> select count(emp_id),place from employee group by place having count(emp_id)>1;
+---------------+------------+
| count(emp_id) | place      |
+---------------+------------+
|             2 | Newyork    |
|             4 | California |
|             2 | Arizona    |
|             4 | Florida    |
|             2 | Georgia    |
|             2 | Alaska     |
|             4 | India      |
+---------------+------------+
7 rows in set (0.00 sec)

mysql>
```

10. List the number of employees in each place, except the California, sorted high to low. Only include places with 2 or more employees.

ANSWER:-

```
mysql> select count(emp_id),place from employee
    -> group by place
    -> having count(emp_id)>=2
    -> and not place = 'California'
    -> order by count(emp_id) desc;
+---------------+------------+
| count(emp_id) | place      |
+---------------+------------+
|             4 | Florida    |
|             4 | India      |
|             2 | Newyork    |
|             2 | Arizona    |
|             2 | Georgia    |
|             2 | Alaska     |
+---------------+------------+
6 rows in set (0.00 sec)

mysql>
```

Exercise2:

Tables for Exercise2:-

1. Create table customer (customer_name char(20),customer_street char(30),customer_city char(30),PRIMARY KEY(customer_name));

ANSWER:-

```
mysql> select * from customer;
+---------------+----------------+---------------+
| customer_name | customer_street | customer_city |
+---------------+----------------+---------------+
| Adams         | Spring          | Pittsfield    |
| Brooks        | Senator         | Brooklyn      |
| Curry         | North           | Rye           |
| Glenn         | Sand Hill       | Woodside      |
| Green         | Walnut          | Stamford      |
| Hayes         | Main            | Harrison      |
| Johnson       | Alma            | Palo Alto     |
| Jones         | Main            | Harrison      |
| Lindsay       | Park            | Pittsfield    |
| Smith         | North           | Rye           |
| Turner        | Putnam          | Stamford      |
| Williams      | Nassau          | Princeton     |
+---------------+----------------+---------------+
12 rows in set (0.00 sec)

mysql>
```

2. Create table branch (branch_name char(15),branch_city char(30),assets numeric(16,2),PRIMARY KEY(branch_name));

ANSWER:-

```
mysql> select * from branch;
+-------------+-------------+------------+
| branch_name | branch_city | assets     |
+-------------+-------------+------------+
| Brighton    | Brooklyn    | 7100000.00 |
| Downtown    | Brooklyn    | 9000000.00 |
| Mianus      | Horseneck   |  400000.00 |
| North Town  | Rye         | 3700000.00 |
| Perryridge  | Horseneck   | 1700000.00 |
| Pownal      | Bennington  |  300000.00 |
| Redwood     | Palo Alto   | 2100000.00 |
| Round Hill  | Horseneck   | 8000000.00 |
+-------------+-------------+------------+
8 rows in set (0.00 sec)

mysql>
```

3. Create table account (account_number char(15),branch_name char (15),balance numeric(12,2),PRIMARY KEY(account_number), FOREIGN KEY (branch_name) REFERENCES branch(branch_name));

ANSWER:-

```
mysql> select * from account;
+----------------+-------------+---------+
| account_number | branch_name | balance |
+----------------+-------------+---------+
| A-101          | Downtown    |  500.00 |
| A-102          | Perryridge  |  400.00 |
| A-201          | Brighton    |  900.00 |
| A-215          | Mianus      |  700.00 |
| A-217          | Brighton    |  750.00 |
| A-222          | Redwood     |  700.00 |
| A-305          | Round Hill  |  350.00 |
+----------------+-------------+---------+
7 rows in set (0.00 sec)

mysql>
```

4. Create table depositor(customer_name char(20),account_number char(10),PRIMARY KEY(customer_name,account_number),FOREIGN KEY (customer_name) REFERENCES customer(customer_name), FOREIGN KEY (account_number) REFERENCES account(account_number));

ANSWER:-

```
mysql> select * from depositor;
+---------------+----------------+
| customer_name | account_number |
+---------------+----------------+
| Johnson       | A-101          |
| Hayes         | A-102          |
| Johnson       | A-201          |
| Smith         | A-215          |
| Jones         | A-217          |
| Lindsay       | A-222          |
| Turner        | A-305          |
+---------------+----------------+
7 rows in set (0.00 sec)

mysql>
```

5. Create table loan(loan_number varchar(6),branch_name char(15),amount int,PRIMARY KEY(loan_number),FOREIGN KEY (branch_name) REFERENCES branch(branch_name));

ANSWER:-

```
mysql> select * from loan;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L-11        | Round Hill  |    900 |
| L-14        | Downtown    |   1500 |
| L-15        | Perryridge  |   1500 |
| L-16        | Perryridge  |   1300 |
| L-17        | Downtown    |   1000 |
| L-23        | Redwood     |   2000 |
| L-93        | Mianus      |    500 |
+-------------+-------------+--------+
7 rows in set (0.00 sec)

mysql>
```

6. Create table borrower(customer_name char(20),loan_number varchar(6),PRIMARY KEY(customer_name,loan_number),FOREIGN KEY (customer_name) REFERENCES customer(customer_name), FOREIGN KEY (loan_number) REFERENCES loan(loan_number));

ANSWER:-

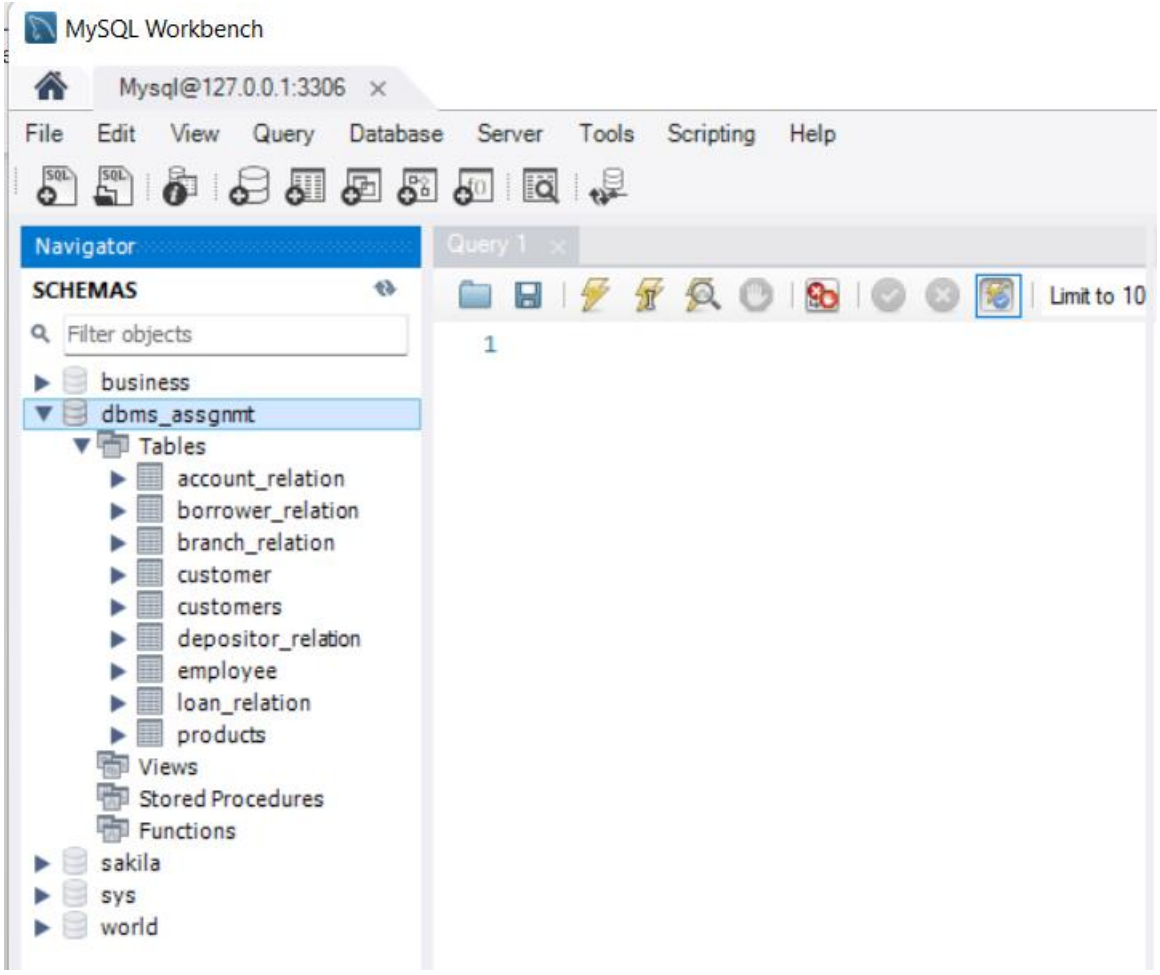
```
mysql> select * from borrower;
+---------------+-------------+
| customer_name | loan_number |
+---------------+-------------+
| Smith         | L-11        |
| Hayes         | L-15        |
| Adams         | L-16        |
| Jones         | L-17        |
| Williams      | L-17        |
| Smith         | L-23        |
| Curry         | L-93        |
+---------------+-------------+
7 rows in set (0.00 sec)

mysql>
```

Questionnaire set 2:-

1. Create the tables for above schema and load data from the respective .csv files.

ANSWER:-



2. For all customers who have loan from the bank,find their names,loan numbers and loan amount(with and without renaming tables).

ANSWER:-

```
mysql>  select customer_name,loan_number,amount from borrower natural join loan;
+---------------+-------------+--------+
| customer_name | loan_number | amount |
+---------------+-------------+--------+
| Smith         | L-11        |    900 |
| Hayes         | L-15        |   1500 |
| Adams         | L-16        |   1300 |
| Jones         | L-17        |   1000 |
| Williams      | L-17        |   1000 |
| Smith         | L-23        |   2000 |
| Curry         | L-93        |    500 |
+---------------+-------------+--------+
7 rows in set (0.00 sec)

mysql>
```

3. Find the customer names,loan numbers and loan amounts for all loans at perryridge branch.

ANSWER:-

```
mysql> select customer_name,loan_number,amount from depositor
    -> natural join loan where loan.branch_name='Perryridge';
+---------------+-------------+--------+
| customer_name | loan_number | amount |
+---------------+-------------+--------+
| Johnson       | L-16        |   1300 |
| Johnson       | L-15        |   1500 |
| Hayes         | L-16        |   1300 |
| Hayes         | L-15        |   1500 |
| Johnson       | L-16        |   1300 |
| Johnson       | L-15        |   1500 |
| Smith         | L-16        |   1300 |
| Smith         | L-15        |   1500 |
| Jones         | L-16        |   1300 |
| Jones         | L-15        |   1500 |
| Lindsay       | L-16        |   1300 |
| Lindsay       | L-15        |   1500 |
| Turner        | L-16        |   1300 |
| Turner        | L-15        |   1500 |
+---------------+-------------+--------+
14 rows in set (0.01 sec)

mysql>
```

4. Find the names of all branches that have assets greater than at least one branch located at Brooklyn.

ANSWER:-

```
mysql> select branch_name from branch where assets > (select min(assets) from branch where branch_city = 'Brooklyn');
+-------------+
| branch_name |
+-------------+
| Downtown    |
| Round Hill  |
+-------------+
2 rows in set (0.01 sec)

mysql>
```

5. List in alphabetical order all customers who have loans at the perryridge branch.

ANSWER:-

```
mysql> select customer_name
    -> from borrower,loan
    -> where loan.branch_name = 'Perryridge' and borrower.loan_number = loan.loan_number
    -> order by customer_name;
+---------------+
| customer_name |
+---------------+
| Adams         |
| Hayes         |
+---------------+
2 rows in set (0.01 sec)
```

6. Print the entire Loan relation in descending order of amount.If several loans have the same amount,order them in ascending order by loan number.

ANSWER:-

```
mysql> SELECT * from loan
    -> order by amount DESC, loan_number ASC;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L-23        | Redwood     |   2000 |
| L-14        | Downtown    |   1500 |
| L-15        | Perryridge  |   1500 |
| L-16        | Perryridge  |   1300 |
| L-17        | Downtown    |   1000 |
| L-11        | Round Hill  |    900 |
| L-93        | Mianus      |    500 |
+-------------+-------------+--------+
7 rows in set (0.00 sec)
```

7. Find the average balance for all accounts.

ANSWER:-

```
mysql> select avg(balance)
    -> from account;
+--------------+
| avg(balance) |
+--------------+
|   614.285714 |
+--------------+
1 row in set (0.00 sec)

mysql>
```

8. Find no.of tuples in customer relation.

ANSWER:-

```
mysql> select count(customer_name)
    -> from customer;
+-----------------------+
| count(customer_name) |
+-----------------------+
|                    12 |
+-----------------------+
1 row in set (0.03 sec)

mysql>
```

9. Find the total of all loan amounts.

ANSWER:-

```
mysql> select sum(amount)
    -> from loan;
+-------------+
| sum(amount) |
+-------------+
|        8700 |
+-------------+
1 row in set (0.00 sec)

mysql>
```

10. Find the average account balance at the Perryridge branch.

ANSWER:-

```
mysql> SELECT avg(balance)
    -> from account
    -> where branch_name = 'Perryridge';
+--------------+
| avg(balance) |
+--------------+
|   400.000000 |
+--------------+
1 row in set (0.00 sec)

mysql>
```

11. Find the average account balance at each branch.

ANSWER:-

```
mysql> SELECT branch_name, avg(balance)
    -> from account
    -> group by branch_name;
+-------------+--------------+
| branch_name | avg(balance) |
+-------------+--------------+
| Brighton    |   825.000000 |
| Downtown    |   500.000000 |
| Mianus      |   700.000000 |
| Perryridge  |   400.000000 |
| Redwood     |   700.000000 |
| Round Hill  |   350.000000 |
+-------------+--------------+
6 rows in set (0.00 sec)

mysql>
```

12. Find the average account balance at each branch ,where the account balance is more than 1200.

ANSWER:-

```
mysql> SELECT branch_name, avg(balance)
    -> from account
    -> where balance > 1200
    -> group by branch_name;
Empty set (0.00 sec)

mysql>
```

13. Find the number of depositors for each branch.

ANSWER:-

```
mysql> SELECT account.branch_name, count(depositor.customer_name)
    -> from account, depositor
    -> where depositor.account_number = account.account_number
    -> group by branch_name;
+-------------+--------------------------------+
| branch_name | count(depositor.customer_name) |
+-------------+--------------------------------+
| Brighton    |                              2 |
| Downtown    |                              1 |
| Mianus      |                              1 |
| Perryridge  |                              1 |
| Redwood     |                              1 |
| Round Hill  |                              1 |
+-------------+--------------------------------+
6 rows in set (0.00 sec)

mysql>
```

14. Find the average balance for each customer who lives in ‘’Harrison” and has at least 3 accounts.

ANSWER:-

```
mysql> SELECT depositor.customer_name, avg(balance)
    -> from depositor, account, customer
    -> where depositor.account_number = account.account_number and
    -> depositor.customer_name = customer.customer_name and
    -> customer_city = 'Harrison'
    -> group by depositor.customer_name
    -> having count(depositor.account_number) >= 3;
Empty set (0.01 sec)

mysql>
```

Exercise-3:-

1) Display your name with the first letter being capital, where the entered name is in lower case.

ANSWER:-

```
mysql> SELECT CONCAT(UPPER(SUBSTR("abhishikth",1,1)),LOWER(SUBSTR("abhishikth",2)));
+-----------------------------------------------------------------------+
| CONCAT(UPPER(SUBSTR("abhishikth",1,1)),LOWER(SUBSTR("abhishikth",2))) |
+-----------------------------------------------------------------------+
| Abhishikth                                                            |
+-----------------------------------------------------------------------+
1 row in set (0.00 sec)
```

2) Display 2nd- 6th characters of your name.

ANSWER:-

```
mysql> select substr("abhishikth",2,5);
+--------------------------+
| substr("abhishikth",2,5) |
+--------------------------+
| bhish                    |
+--------------------------+
1 row in set (0.00 sec)

mysql>
```

3) Find the length of your full institute name.

ANSWER:-

```
mysql> SELECT LENGTH("Indian Institute Of Information Technology Sri City");
+-------------------------------------------------------------+
| LENGTH("Indian Institute Of Information Technology Sri City") |
+-------------------------------------------------------------+
|                                                          51 |
+-------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

4) Display all the Employee names with its first letter in upper case.

ANSWER:-

```
mysql> SELECT CONCAT(UPPER(SUBSTR(emp_name,1,1)),LOWER(SUBSTR(emp_name,2))) from
+-------------------------------------------------------------+
| CONCAT(UPPER(SUBSTR(emp_name,1,1)),LOWER(SUBSTR(emp_name,2))) |
+-------------------------------------------------------------+
| Peter                                                       |
| Mark                                                        |
| Donald                                                      |
| Obama                                                       |
| Linklon                                                     |
| Kane                                                        |
| Adam                                                        |
| Mac                                                         |
| Manas                                                       |
| Vasin                                                       |
| Peter                                                       |
| Mark                                                        |
| Donald                                                      |
| Obama                                                       |
| Linklon                                                     |
| Kane                                                        |
| Adam                                                        |
| Mac                                                         |
| Manas                                                       |
| Vasin                                                       |
+-------------------------------------------------------------+
20 rows in set (0.00 sec)
```

5) List the department name of each employee as a three letter code.

ANSWER:-

```
mysql> SELECT SUBSTR(emp_dept,1,3) from employee;
+----------------------+
| SUBSTR(emp_dept,1,3) |
+----------------------+
| Fin                  |
| HR                   |
| Fin                  |
| Man                  |
| HR                   |
| Sal                  |
| Man                  |
| Fin                  |
| Acc                  |
| Acc                  |
| Fin                  |
| HR                   |
| Fin                  |
| Man                  |
| HR                   |
| Sal                  |
| Man                  |
| Fin                  |
| Acc                  |
| Acc                  |
+----------------------+
20 rows in set (0.00 sec)
```

6) Display the month of the joining of each employee.

ANSWER:-

```
mysql> SELECT Date_Format(doj,"%M") from employee
+-----------------------+
| Date_Format(doj,"%M") |
+-----------------------+
| August                |
| March                 |
| December              |
| October               |
| August                |
| January               |
| October               |
| June                  |
| December              |
| October               |
| October               |
| December              |
| June                  |
| October               |
| January               |
| August                |
| October               |
| December              |
| March                 |
| August                |
+-----------------------+
20 rows in set (0.01 sec)
```

7) Display the date of joining of each employee in dd/mm/yy format.

ANSWER:-

```
mysql> SELECT DATE_FORMAT(doj, "%d/%c/%y") from employee;
+------------------------------+
| DATE_FORMAT(doj, "%d/%c/%y") |
+------------------------------+
| 25/8/02                      |
| 25/3/80                      |
| 26/12/95                     |
| 30/10/90                     |
| 08/8/08                      |
| 01/1/00                      |
| 25/10/20                     |
| 09/6/70                      |
| 11/12/90                     |
| 10/10/89                     |
| 10/10/89                     |
| 11/12/90                     |
| 09/6/70                      |
| 25/10/20                     |
| 01/1/00                      |
| 08/8/08                      |
| 30/10/90                     |
| 26/12/95                     |
| 25/3/80                      |
| 25/8/02                      |
+------------------------------+
20 rows in set (0.00 sec)

mysql>
```

8) Display the experience of each employee in terms of months.

ANSWER:-

```
mysql> SELECT emp_name, TIMESTAMPDIFF(MONTH, doj, curdate()) from employee;
+----------+--------------------------------------+
| emp_name | TIMESTAMPDIFF(MONTH, doj, curdate()) |
+----------+--------------------------------------+
| peter    |                                  240 |
| Mark     |                                  509 |
| Donald   |                                  320 |
| Obama    |                                  382 |
| Linklon  |                                  168 |
| Kane     |                                  272 |
| Adam     |                                   22 |
| Mac      |                                  626 |
| Manas    |                                  380 |
| Vasin    |                                  394 |
| peter    |                                  394 |
| Mark     |                                  380 |
| Donald   |                                  626 |
| Obama    |                                   22 |
| Linklon  |                                  272 |
| Kane     |                                  168 |
| Adam     |                                  382 |
| Mac      |                                  320 |
| Manas    |                                  509 |
| Vasin    |                                  240 |
+----------+--------------------------------------+
20 rows in set (0.00 sec)

mysql>
```

9) Display the experience of each employee in terms of years and months.

ANSWER:-

```
mysql> select emp_name, concat(floor(datediff(curdate(),doj)/365) , 'Years '
,floor(datediff(curdate(),doj)/365%12) , 'Months') as experience from employe
e;^Z
+----------+------------------+
| emp_name | experience       |
+----------+------------------+
| peter    | 20Years 8Months  |
| Mark     | 42Years 6Months  |
| Donald   | 26Years 2Months  |
| Obama    | 31Years 7Months  |
| Linklon  | 14Years 2Months  |
| Kane     | 22Years 10Months |
| Adam     | 1Years 1Months   |
| Mac      | 52Years 4Months  |
| Manas    | 31Years 7Months  |
| Vasin    | 32Years 8Months  |
| peter    | 32Years 8Months  |
| Mark     | 31Years 7Months  |
| Donald   | 52Years 4Months  |
| Obama    | 1Years 1Months   |
| Linklon  | 22Years 10Months |
| Kane     | 14Years 2Months  |
| Adam     | 31Years 7Months  |
| Mac      | 26Years 2Months  |
| Manas    | 42Years 6Months  |
| Vasin    | 20Years 8Months  |
+----------+------------------+
20 rows in set (0.00 sec)

    ->
```

10) Display the date of the next Friday after today's date.

ANSWER:-

```
mysql> SELECT if(5-Date_Format(curdate(),"%w")>0, curdate()+ interval 5-Date_
Format(curdate(),"%w") day, curdate()+ interval 12-Date_Format(curdate(),"%w"
) day);
+-------------------------------------------------------------------------------
-------------------------------------------------------------------+
| if(5-Date_Format(curdate(),"%w")>0, curdate()+ interval 5-Date_Format(curda
te(),"%w") day, curdate()+ interval 12-Date_Format(curdate(),"%w") day) |
+-------------------------------------------------------------------------------
-------------------------------------------------------------------+
| 2022-09-09
                                                                   |
+-------------------------------------------------------------------------------
-------------------------------------------------------------------+
1 row in set (0.01 sec)

mysql>
```

11) Display the day of joining of each employee.

ANSWER:-

```
mysql> SELECT Date_Format(doj,"%W") from employee;
+-----------------------+
| Date_Format(doj,"%W") |
+-----------------------+
| Sunday                |
| Tuesday               |
| Tuesday               |
| Tuesday               |
| Friday                |
| Saturday              |
| Sunday                |
| Tuesday               |
| Tuesday               |
| Tuesday               |
| Tuesday               |
| Tuesday               |
| Tuesday               |
| Sunday                |
| Saturday              |
| Friday                |
| Tuesday               |
| Tuesday               |
| Tuesday               |
| Sunday                |
+-----------------------+
20 rows in set (0.00 sec)

mysql>
```

12) Display the date corresponding to 15 days after today's date.

ANSWER:-

```
mysql> select curdate() + interval 15 DAY;
+-----------------------------+
| curdate() + interval 15 DAY |
+-----------------------------+
| 2022-09-22                  |
+-----------------------------+
1 row in set (0.00 sec)

mysql>
```

13) Display the value 94204.27348 truncated up to 2 digits after the decimal point.

ANSWER:-

```
mysql> select Truncate(94204.27348,2);
+-------------------------+
| Truncate(94204.27348,2) |
+-------------------------+
|                94204.27 |
+-------------------------+
1 row in set (0.00 sec)

mysql>
```

14) Display the value of the expression 5 + 89.

ANSWER:-

```
mysql> select 5+89;
+------+
| 5+89 |
+------+
|   94 |
+------+
1 row in set (0.00 sec)

mysql>
```

15) Find out the square root of 6464312.

ANSWER:-

```
mysql> select sqrt(664312);
+-------------------+
| sqrt(664312)      |
+-------------------+
| 815.0533724855078 |
+-------------------+
1 row in set (0.00 sec)

mysql>
```

←THE END→