Experiment No: 3

Title: Implementation of different types of functions with suitable examples.

- ***Number Function***
- ***Aggregate Function***
- ***Character Function***
- ***Conversion Function***
- ***Date Function***

Objective:

- To understand the different issues involved in the design and implementation of a database system
- To understand and use functions

Theory

NUMBER FUNCTION:

Abs(n) :Select abs(-15);

```
mysql> select abs(-14);
+----------+
| abs(-14) |
+----------+
|       14 |
+----------+
1 row in set (0.00 sec)
```

Exp(n): Select exp(4) from dual;

```
mysql> select exp(4);
+--------------------+
| exp(4)             |
+--------------------+
| 54.598150033144236 |
+--------------------+
1 row in set (0.00 sec)
```

Power(m,n):Select power(4,2) from dual;

```
mysql> select power(4,2);
+------------+
| power(4,2) |
+------------+
|         16 |
+------------+
1 row in set (0.00 sec)
```

Mod(m,n): Select mod(10,3) from dual;

```
mysql> select mod(10,3);
+-----------+
| mod(10,3) |
+-----------+
|         1 |
+-----------+
1 row in set (0.00 sec)
```

Round(m,n): Select round(100.256,2) from dual;

```
mysql> select round(100.256,2);
+-------------------+
| round(100.256,2) |
+-------------------+
|           100.26 |
+-------------------+
1 row in set (0.00 sec)
```

Sqrt(m,n);Select sqrt(16) from dual;

```
mysql> select sqrt(16);
+----------+
| sqrt(16) |
+----------+
|        4 |
+----------+
1 row in set (0.00 sec)
```

Aggregative operators: In addition to simply retrieving data, we often want to perform some computation or summarization. SQL allows the use of arithmetic expressions. We now consider a powerful class of constructs for computing aggregate values such as MIN and SUM.

1. **Count:** COUNT following by a column name returns the count of tuple in that column. If DISTINCT keyword is used then it will return only the count of unique tuple in the column. Otherwise, it will return count of all the tuples (including duplicates) count (*) indicates all the tuples of the column.

   Syntax: COUNT (Column name)

   Example:

```
mysql> SELECT COUNT(*) AS Total_Employees FROM EMPLOYEE;
+----------------+
| Total_Employees |
+----------------+
|             10 |
+----------------+
1 row in set (0.00 sec)
```

2. SUM: SUM followed by a column name returns the sum of all the values in that column.

Syntax: SUM (Column name)

Example:

```
mysql> SELECT SUM(Salary) AS Total_Salary FROM EMPLOYEE;
+--------------+
| Total_Salary |
+--------------+
|    518000.00 |
+--------------+
1 row in set (0.00 sec)
```

3. AVG: AVG followed by a column name returns the average value of that column values.

Syntax: AVG (n1, n2...)

Example:

```
mysql> SELECT AVG(Salary) AS Average_Salary FROM EMPLOYEE;
+----------------+
| Average_Salary |
+----------------+
|   51800.000000 |
+----------------+
1 row in set (0.00 sec)
```

4. MAX: MAX followed by a column name returns the maximum value of that column.

Syntax: MAX (Column name)

Example:

```
mysql> SELECT MAX(Salary) AS Highest_Salary FROM EMPLOYEE;
+----------------+
| Highest_Salary |
+----------------+
|       62000.00 |
+----------------+
1 row in set (0.00 sec)
```

5. MIN: MIN followed by column name returns the minimum value of that column.

Syntax: MIN (Column name)

Example:

```
mysql> SELECT MIN(Salary) AS Lowest_Salary FROM EMPLOYEE;
+---------------+
| Lowest_Salary |
+---------------+
|      40000.00 |
+---------------+
1 row in set (0.00 sec)
```

CHARACTER FUNCTION:

lower (char): select lower (‘HELLO’) from dual;

```
mysql> select lower("HELLO");
+----------------+
| lower("HELLO") |
+----------------+
| hello          |
+----------------+
1 row in set (0.01 sec)
```

upper (char) :select upper (‘hello’) from dual;

```
mysql> select upper("hello");
+----------------+
| upper("hello") |
+----------------+
| HELLO          |
+----------------+
1 row in set (0.01 sec)
```

ltrim (char):select ltrim (‘csbs’) from dual;

```
mysql> select ltrim('           csbs');
+--------------------------+
| ltrim('           csbs') |
+--------------------------+
| csbs                     |
+--------------------------+
1 row in set (0.00 sec)
```

rtrim (char): select rtrim (‘csbs’) from dual;

```
mysql> select rtrim('csbs                              ');
+----------------------------------------+
| rtrim('csbs                          ') |
+----------------------------------------+
| csbs                                   |
+----------------------------------------+
1 row in set (0.00 sec)
```

replace (char,search): select replace(‘jack and jue’,‘j’,‘bl’) from dual;

```
mysql> select replace('jack','j','bl');
+--------------------------+
| replace('jack','j','bl') |
+--------------------------+
| black                    |
+--------------------------+
1 row in set (0.00 sec)
```

CONVERSION FUNCTIONS:

The CONVERT() function converts a value into the specified datatype or character set.

Syntax: CONVERT(value, type)

Parameter Values

Parameter	Description
value	Required. The value to convert
type	Required. The datatype to convert to. Can be one of the following:

```
mysql> SELECT CONVERT(123, CHAR);
+--------------------+
| CONVERT(123, CHAR) |
+--------------------+
| 123                |
+--------------------+
1 row in set (0.00 sec)

mysql> SELECT CONVERT('15:05:21', TIME);
+--------------------------+
| CONVERT('15:05:21', TIME) |
+--------------------------+
| 15:05:21                 |
+--------------------------+
1 row in set (0.00 sec)

mysql> SELECT CONVERT('123.45', DECIMAL(5, 2));
+---------------------------------+
| CONVERT('123.45', DECIMAL(5, 2)) |
+---------------------------------+
|                          123.45 |
+---------------------------------+
1 row in set (0.00 sec)
```

LAB PRACTICE ASSIGNMENT:

Create a table EMPLOYEE with following schema:

(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation , Salary)

```
mysql> desc EMPLOYEE;
+-------------+----------+------+-----+---------+-------+
| Field       | Type     | Null | Key | Default | Extra |
+-------------+----------+------+-----+---------+-------+
| EMP_No      | int      | YES  |     | NULL    |       |
| E_Name      | char(10) | YES  |     | NULL    |       |
| E_address   | char(10) | YES  |     | NULL    |       |
| E_phnno     | int      | YES  |     | NULL    |       |
| DEPt_no     | char(10) | YES  |     | NULL    |       |
| DEPT_name   | char(10) | YES  |     | NULL    |       |
| Job_id      | int      | YES  |     | NULL    |       |
| Salary      | int      | YES  |     | NULL    |       |
| designation | char(25) | YES  |     | NULL    |       |
| join_date   | char(25) | YES  |     | NULL    |       |
+-------------+----------+------+-----+---------+-------+
10 rows in set (0.01 sec)
```

Write SQL statements for the following query.

1. List the E_no, E_name, Salary of all employees working for MANAGER.

```
mysql> select * from EMPLOYEE where designation='Manager';
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
|      1 | Riya   | Delhi     |   12345 | D_A     | python    |      1 |  20000 | Manager     |        5 | 2008-02-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
1 row in set (0.00 sec)
```

2. Display all the details of the employee whose salary is more than the Sal of any IT.

```
mysql> select * from EMPLOYEE where salary > 25000;
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst     |        3 | 2009-07-10 |
|      4 | Soniya | Kolkata   |   38467 | D_D     | Security  |      4 |  60000 | President   |        9 | 1982-07-10 |
|      5 | Hari   | Kolkata   |   38367 | D_E     | Finance   |      5 |  30000 | MGR         |        6 | 2000-07-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
3 rows in set (0.00 sec)
```

3. List the employees in the ascending order of Designations of those joined after 1982.

```
mysql> select * from EMPLOYEE where join_date > ('1982-12-12')order by designation;
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst     |        3 | 2009-07-10 |
|      1 | Riya   | Delhi     |   12345 | D_A     | python    |      1 |  20000 | Manager     |        5 | 2008-02-10 |
|      5 | Hari   | Kolkata   |   38367 | D_E     | Finance   |      5 |  30000 | MGR         |        6 | 2000-07-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
3 rows in set (0.00 sec)
```

4. List the employees along with their Experience and Daily Salary.

```
mysql> select * from EMPLOYEE;
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
|      1 | Riya   | Delhi     |   12345 | D_A     | python    |      1 |  20000 | Manager     |        5 | 2008-02-10 |
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst     |        3 | 2009-07-10 |
|      3 | hardik | mumbai    |   34567 | D_C     | data      |      3 |  15000 | Clerk       |        3 | 1981-07-10 |
|      4 | Soniya | Kolkata   |   38467 | D_D     | Security  |      4 |  60000 | President   |        9 | 1982-07-10 |
|      5 | Hari   | Kolkata   |   38367 | D_E     | Finance   |      5 |  30000 | MGR         |        6 | 2000-07-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+-------------+----------+------------+
5 rows in set (0.00 sec)
```

5. List the employees who are either ‘CLERK’ or ‘ANALYST’ .

```
mysql> select * from EMPLOYEE where designation='Clerk' or designation='Analyst';
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst    |        3 | 2009-07-10 |
|      3 | hardik | mumbai    |   34567 | D_C     | data      |      3 |  15000 | Clerk      |        3 | 1981-07-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
2 rows in set (0.00 sec)
```

6. List the employees who joined on 10-July-1981, 10-Feb-2008, 10-July-2009

```
mysql> select * from EMPLOYEE where join_date in ('1981-07-10','2008-02-10','2009-07-10');
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
|      1 | Riya   | Delhi     |   12345 | D_A     | python    |      1 |  20000 | Manager    |        5 | 2008-02-10 |
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst    |        3 | 2009-07-10 |
|      3 | hardik | mumbai    |   34567 | D_C     | data      |      3 |  15000 | Clerk      |        3 | 1981-07-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
3 rows in set (0.00 sec)
```

7. List the employees who are working for the department D_B or D_D.

```
mysql> select * from EMPLOYEE where DEPt_no='D_B' or DEPt_no='D_D';
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst    |        3 | 2009-07-10 |
|      4 | Soniya | Kolkata   |   38467 | D_D     | Security  |      4 |  60000 | President  |        9 | 1982-07-10 |
+--------+--------+-----------+---------+---------+-----------+--------+--------+------------+----------+------------+
2 rows in set (0.00 sec)
```

8. List the Enames those are starting with ‘S’ .

```
mysql> select E_Name from EMPLOYEE where E_name like 'S%';
+--------+
| E_Name |
+--------+
| Sneha  |
| Soniya |
+--------+
2 rows in set (0.00 sec)
```

9. Dislay the name as well as the first five characters of name(s) starting with ‘H’

```
2 rows in set (0.00 sec)

mysql> select E_Name, substring(E_Name,1,5) as F_F from EMPLOYEE where E_Name like 'H%';
+--------+-------+
| E_Name | F_F   |
+--------+-------+
| hardik | hardi |
| Hari   | Hari  |
+--------+-------+
2 rows in set (0.00 sec)
```

```
to use near '' at line 2
mysql> select E_Name from EMPLOYEE where E_name like 'H%';
+--------+
| E_Name |
+--------+
| hardik |
| Hari   |
+--------+
2 rows in set (0.00 sec)
```

10. List all the emps except ‘PRESIDENT’ & ‘MGR” in asc order of Salaries.

```
2 rows in set (0.00 sec)

mysql> select * from EMPLOYEE where designation NOT IN ('President', 'MGR') order by salary asc;
+--------+--------+-----------+---------+---------+-----------+-------+--------+-------------+----------+------------+
| EMP_No | E_Name | E_address | E_phnno | DEPt_no | DEPT_name | Job_id | Salary | designation | exp_year | join_date  |
+--------+--------+-----------+---------+---------+-----------+-------+--------+-------------+----------+------------+
|      3 | hardik | mumbai    |   34567 | D_C     | data      |      3 |  15000 | Clerk       |        3 | 1981-07-10 |
|      1 | Riya   | Delhi     |   12345 | D_A     | python    |      1 |  20000 | Manager     |        5 | 2008-02-10 |
|      2 | Sneha  | pune      |   12745 | D_B     | Java      |      2 |  30000 | Analyst     |        3 | 2009-07-10 |
+--------+--------+-----------+---------+---------+-----------+-------+--------+-------------+----------+------------+
```