LABCYCLE-1
ASSIGNMENT-1

1.Display the details of employee names, salary, comm., hire date of employees.

Query:

```
mysql> select ename,sal,comm,hiredate from emp;
```

Output:

```
+--------+------+------+------------+
| ename  | sal  | comm | hiredate   |
+--------+------+------+------------+
| smith  |  800 | NULL | 1980-12-17 |
| allen  | 1600 |  300 | 1981-02-20 |
| ward   | 1250 |  500 | 1981-02-22 |
| jones  | 2975 | NULL | 1981-04-02 |
| martin | 1250 | 1400 | 1981-09-28 |
| blake  | 2850 | NULL | 1981-05-01 |
| clark  | 2450 | NULL | 1981-06-09 |
| scott  | 3000 | NULL | 1982-12-09 |
| king   | 5000 | NULL | 1981-11-17 |
| turner | 1500 |    0 | 1981-09-08 |
| adams  | 1100 | NULL | 1983-01-12 |
| james  |  950 | NULL | 1981-12-03 |
| ford   | 3000 | NULL | 1981-12-03 |
| miller | 1300 | NULL | 1982-01-23 |
+--------+------+------+------------+
14 rows in set (0.01 sec)
```

2.Display the details of employees who are working in the department whose number is 20.

Query:

```
mysql> select *from emp where deptno=20;
```

Output:

```
+-------+-------+---------+------+------------+------+------+--------+
| empno | ename | job     | mgr  | hiredate   | sal  | comm | deptno |
+-------+-------+---------+------+------------+------+------+--------+
|  7369 | smith | clerk   | 7902 | 1980-12-17 |  800 | NULL |     20 |
|  7566 | jones | manager | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7788 | scott | analyst | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7876 | adams | clerk   | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7902 | ford  | analyst | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
5 rows in set (0.13 sec)
```

3.Display the details of employees whose salary is between 1000 and 3000.

Query:

```
mysql> select *from emp where sal between 1000 and 3000;
```

Output:

```
+-------+--------+----------+------+------------+------+------+--------+
| empno | ename  | job      | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+----------+------+------------+------+------+--------+
```

```
|  7499 | allen  | salesman | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | ward   | salesman | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | jones  | manager  | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7654 | martin | salesman | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7698 | blake  | manager  | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | clark  | manager  | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7788 | scott  | analyst  | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7844 | turner | salesman | 7698 | 1981-09-08 | 1500 |    0 |     30 |
|  7876 | adams  | clerk    | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7902 | ford   | analyst  | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7934 | miller | clerk    | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+----------+------+------------+------+------+--------+
11 rows in set (0.14 sec)
```

4.Display the details of employees whose salary is less than 2000 or more than 3000.

Query:

```
mysql> select *from emp where sal<2000 or sal>3000;
```

Output:

```
+-------+--------+-----------+------+------------+------+------+--------+
| empno | ename  | job       | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+-----------+------+------------+------+------+--------+
|  7369 | smith  | clerk     | 7902 | 1980-12-17 |  800 | NULL |     20 |
|  7499 | allen  | salesman  | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | ward   | salesman  | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7654 | martin | salesman  | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7839 | king   | president | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7844 | turner | salesman  | 7698 | 1981-09-08 | 1500 |    0 |     30 |
|  7876 | adams  | clerk     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7900 | james  | clerk     | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7934 | miller | clerk     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
9 rows in set (0.00 sec)
```

5.Find all the employees who don’t have a Manager.

Query:

```
mysql> select *from emp where mgr is null;
```

Output:

```
+-------+-------+-----------+------+------------+------+------+--------+
| empno | ename | job       | mgr  | hiredate   | sal  | comm | deptno |
+-------+-------+-----------+------+------------+------+------+--------+
|  7839 | king  | president | NULL | 1981-11-17 | 5000 | NULL |     10 |
+-------+-------+-----------+------+------------+------+------+--------+
1 row in set (0.02 sec)
```

6.Display the employee names and yearly salary with header “Annual Salary”.

Query:

```
mysql> select ename,sal*12 as "Annual salary" from emp;
```

Output:

```
+--------+---------------+
| ename  | Annual salary |
+--------+---------------+
```

```
| smith  |          9600 |
| allen  |         19200 |
| ward   |         15000 |
| jones  |         35700 |
| martin |         15000 |
| blake  |         34200 |
| clark  |         29400 |
| scott  |         36000 |
| king   |         60000 |
| turner |         18000 |
| adams  |         13200 |
| james  |         11400 |
| ford   |         36000 |
| miller |         15600 |
+--------+---------------+
14 rows in set (0.10 sec)
```

7.Display the list of employees whose names starts with 'S' and contains exactly 5 characters.

Query:

```
mysql> select *from emp where ename like 's____';
```

Output:

```
+-------+-------+---------+------+------------+------+------+--------+
| empno | ename | job     | mgr  | hiredate   | sal  | comm | deptno |
+-------+-------+---------+------+------------+------+------+--------+
|  7369 | smith | clerk   | 7902 | 1980-12-17 |  800 | NULL |     20 |
|  7788 | scott | analyst | 7566 | 1982-12-09 | 3000 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
2 rows in set (0.00 sec)
```

9.Display the list of employees who are working in dept 20 or getting salary more than 3000

Query:

```
mysql> select * from emp where deptno=20 or sal>3000;
```

Output:

```
+-------+-------+-----------+------+------------+------+------+--------+
| empno | ename | job       | mgr  | hiredate   | sal  | comm | deptno |
+-------+-------+-----------+------+------------+------+------+--------+
|  7369 | smith | clerk     | 7902 | 1980-12-17 |  800 | NULL |     20 |
|  7566 | jones | manager   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7788 | scott | analyst   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | king  | president | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7876 | adams | clerk     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7902 | ford  | analyst   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+-----------+------+------------+------+------+--------+
6 rows in set (0.00 sec)
```

10.Find all clerks who earn between 1000 and 2000.

Query:

```
mysql> select * from emp where job='clerk' and sal between 1000 and
2000;
```

Output:

```
+-------+--------+-------+------+------------+------+------+--------+
```

```
| empno | ename  | job   | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+-------+------+------------+------+------+--------+
|  7876 | adams  | clerk | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7934 | miller | clerk | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+-------+------+------------+------+------+--------+
2 rows in set (0.00 sec)
```

11.Find all the employees who have comm.

Query:

```
mysql> select * from emp where comm is not NULL;
```

Output:

```
+-------+--------+----------+------+------------+------+------+--------+
| empno | ename  | job      | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+----------+------+------------+------+------+--------+
|  7499 | allen  | salesman | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | ward   | salesman | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7654 | martin | salesman | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7844 | turner | salesman | 7698 | 1981-09-08 | 1500 |    0 |     30 |
+-------+--------+----------+------+------------+------+------+--------+
4 rows in set (0.00 sec)
```

12.List dept no and name in alphabetical order of dept name.

Query:

```
mysql> select deptno,dename from dept order by dename;
```

Output:

```
+--------+------------+
| deptno | dename     |
+--------+------------+
|     10 | accounting |
|     40 | operations |
|     20 | research   |
|     30 | sales      |
+--------+------------+
4 rows in set (0.01 sec)
```

13.List the details of the employees in department 10 and 20 in alphabetical order of name.

Query:

```
mysql> select * from emp where deptno=10 or deptno=20 order by ename;
```

Output:

```
+-------+--------+-----------+------+------------+------+------+--------+
| empno | ename  | job       | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+-----------+------+------------+------+------+--------+
|  7876 | adams  | clerk     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7782 | clark  | manager   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7902 | ford   | analyst   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7566 | jones  | manager   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7839 | king   | president | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7934 | miller | clerk     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
|  7788 | scott  | analyst   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7369 | smith  | clerk     | 7902 | 1980-12-17 |  800 | NULL |     20 |
+-------+--------+-----------+------+------------+------+------+--------+
```

```
8 rows in set (0.00 sec)
```

14.Display name and total remuneration of all employees.

Query:

```
      mysql> select ename,sal + ifnull(comm,0) as 'Total remunaration' from
emp;
```

Output:

```
+--------+--------------------+
| ename  | Total remunaration |
+--------+--------------------+
| smith  |                800 |
| allen  |               1900 |
| ward   |               1750 |
| jones  |               2975 |
| martin |               2650 |
| blake  |               2850 |
| clark  |               2450 |
| scott  |               3000 |
| king   |               5000 |
| turner |               1500 |
| adams  |               1100 |
| james  |                950 |
| ford   |               3000 |
| miller |               1300 |
+--------+--------------------+
14 rows in set (0.02 sec)
```

15.Display all the employees who were hired during 1983.

Query:

```
      mysql> select empno,ename,hiredate from emp where hiredate like
'1983%';
```

Output:

```
+-------+-------+------------+
| empno | ename | hiredate   |
+-------+-------+------------+
|  7876 | adams | 1983-01-12 |
+-------+-------+------------+
1 row in set, 1 warning (0.00 sec)
```

16.Display name, annual salary and commission of all sales persons whose monthly salary is greater than their commission. The output should be in the descending order of salary. If two or more employees have the same salary sort by ename, with in the highest salary order,

Query:

```
      mysql> select ename,sal*12 as 'Annual salary',comm from emp where
job='salesman' and sal>ifnull(comm,0) order by sal desc,ename;
```

Output:

```
+--------+---------------+------+
| ename  | Annual salary | comm |
+--------+---------------+------+
| allen  |         19200 |  300 |
| turner |         18000 |    0 |
```

```
| ward   |         15000 |  500 |
+--------+---------------+------+
3 rows in set (0.00 sec)
```

17.Increase the salary for all employees by 15%.

Query:

```
mysql> select ename,sal + (sal*0.15) from emp;
```

Output:

```
+--------+------------------+
| ename  | sal + (sal*0.15) |
+--------+------------------+
| smith  |           920.00 |
| allen  |          1840.00 |
| ward   |          1437.50 |
| jones  |          3421.25 |
| martin |          1437.50 |
| blake  |          3277.50 |
| clark  |          2817.50 |
| scott  |          3450.00 |
| king   |          5750.00 |
| turner |          1725.00 |
| adams  |          1265.00 |
| james  |          1092.50 |
| ford   |          3450.00 |
| miller |          1495.00 |
+--------+------------------+
14 rows in set (0.00 sec)
```

18.Display the list of employees in the descending order of job and salaries.

Query:

```
mysql> select empno,ename,job,sal from emp order by job desc,sal desc;
```

Output:

```
+-------+--------+-----------+------+
| empno | ename  | job       | sal  |
+-------+--------+-----------+------+
|  7499 | allen  | salesman  | 1600 |
|  7844 | turner | salesman  | 1500 |
|  7521 | ward   | salesman  | 1250 |
|  7654 | martin | salesman  | 1250 |
|  7839 | king   | president | 5000 |
|  7566 | jones  | manager   | 2975 |
|  7698 | blake  | manager   | 2850 |
|  7782 | clark  | manager   | 2450 |
|  7934 | miller | clerk     | 1300 |
|  7876 | adams  | clerk     | 1100 |
|  7900 | james  | clerk     |  950 |
|  7369 | smith  | clerk     |  800 |
|  7902 | ford   | analyst   | 3000 |
|  7788 | scott  | analyst   | 3000 |
+-------+--------+-----------+------+
14 rows in set (0.00 sec)
```

19.Select data as displayed Who, what and when
Allen has held the position of salesman in dept 30 since 20-feb-81
Ward has held the position of salesman in dept 30 since 22-feb-81.
Query:

```
      mysql> select concat(ename, 'has held the position of',job,'in
dept',deptno,'since',hiredate) as 'who,what and when' from emp;
```

Output:

```
+-------------------------------------------------------------+
| who,what and when                                           |
+-------------------------------------------------------------+
| smithhas held the position ofclerkin dept20since1980-12-17     |
| allenhas held the position ofsalesmanin dept30since1981-02-20  |
| wardhas held the position ofsalesmanin dept30since1981-02-22   |
| joneshas held the position ofmanagerin dept20since1981-04-02   |
| martinhas held the position ofsalesmanin dept30since1981-09-28 |
| blakehas held the position ofmanagerin dept30since1981-05-01   |
| clarkhas held the position ofmanagerin dept10since1981-06-09   |
| scotthas held the position ofanalystin dept20since1982-12-09   |
| kinghas held the position ofpresidentin dept10since1981-11-17  |
| turnerhas held the position ofsalesmanin dept30since1981-09-08 |
| adamshas held the position ofclerkin dept20since1983-01-12     |
| jameshas held the position ofclerkin dept30since1981-12-03     |
| fordhas held the position ofanalystin dept20since1981-12-03    |
| millerhas held the position ofclerkin dept10since1982-01-23    |
+-------------------------------------------------------------+
14 rows in set (0.00 sec)
```

ASSIGNMENT-2

1.Find out how many managers are there without listing them.

Query:

```
mysql> select count(*) from emp where job ='manager';
```

Output:

```
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)
```

2.Find the difference of highest and lowest salaries for each job type.

Query:

```
mysql> select empno,job,max(sal)-min(sal) from emp group by job;
```

Output :

```
+-------+-----------+-------------------+
| empno | job       | max(sal)-min(sal) |
+-------+-----------+-------------------+
|  7788 | analyst   |                 0 |
|  7369 | clerk     |               500 |
|  7566 | manager   |               525 |
|  7839 | president |                 0 |
|  7499 | salesman  |               350 |
+-------+-----------+-------------------+
5 rows in set (0.12 sec)
```

3.Find the Minimum, Maximum and average salary for each job.

Query:

```
mysql> select job,min(sal),max(sal),avg(sal) from emp group by job;
```

Output:

```
+-----------+----------+----------+-----------+
| job       | min(sal) | max(sal) | avg(sal)  |
+-----------+----------+----------+-----------+
| analyst   |     3000 |     3000 | 3000.0000 |
| clerk     |      800 |     1300 | 1037.5000 |
| manager   |     2450 |     2975 | 2758.3333 |
| president |     5000 |     5000 | 5000.0000 |
| salesman  |     1250 |     1600 | 1400.0000 |
+-----------+----------+----------+-----------+
5 rows in set (0.00 sec)
```

4.List the employee name, department name for each employee.

Query:

```
mysql> select ename,dename from emp, dept where emp.deptno =
dept.deptno;
```

Output:

```
+--------+------------+
| ename  | dename     |
+--------+------------+
```

```
| clark  | accounting |
| king   | accounting |
| miller | accounting |
| smith  | research   |
| jones  | research   |
| scott  | research   |
| adams  | research   |
| ford   | research   |
| allen  | sales      |
| ward   | sales      |
| martin | sales      |
| blake  | sales      |
| turner | sales      |
| james  | sales      |
+--------+------------+
14 rows in set (0.08 sec)
```

5.Display the list of employees who are working in dept no 10 and whose salary is >=5000.

Query:

```
      mysql> select empno,ename,deptno,sal from emp where deptno=10 and
sal>=5000;
```

Output:

```
+-------+-------+--------+------+
| empno | ename | deptno | sal  |
+-------+-------+--------+------+
|  7839 | king  |     10 | 5000 |
+-------+-------+--------+------+
1 row in set (0.00 sec)
```

6.Display the list of employees whose name end with letter say ‘s’.

Query:

```
      mysql> select empno,ename from emp where ename like '%s';
```

Output:

```
+-------+-------+
| empno | ename |
+-------+-------+
|  7566 | jones |
|  7876 | adams |
|  7900 | james |
+-------+-------+
3 rows in set (0.00 sec)
```

7.Display the list of employees whose salary is greater than 5000 and also have a comm

Query:

```
      mysql> select empno,ename,sal,comm from emp where sal>=500 and comm is
not NULL;
```

Output:

```
+-------+--------+------+------+
| empno | ename  | sal  | comm |
+-------+--------+------+------+
```

```
|  7499 | allen  | 1600 |  300 |
|  7521 | ward   | 1250 |  500 |
|  7654 | martin | 1250 | 1400 |
|  7844 | turner | 1500 |    0 |
+-------+--------+------+------+
4 rows in set (0.00 sec)
```

8.Display the list of employees whose salary is less than 1000 and don't have any comm.

Query:

```
      mysql> select empno,ename,sal,comm from emp where sal<=1000 and comm
is NULL;
```

Output:

```
+-------+-------+------+------+
| empno | ename | sal  | comm |
+-------+-------+------+------+
|  7369 | smith |  800 | NULL |
|  7900 | james |  950 | NULL |
+-------+-------+------+------+
2 rows in set (0.00 sec)
```

9.Display the list of employees who are not working in dept 10.

Query:

```
      mysql> select empno,ename,deptno from emp where deptno!= 10;
```

Output:

```
+-------+--------+--------+
| empno | ename  | deptno |
+-------+--------+--------+
|  7369 | smith  |     20 |
|  7499 | allen  |     30 |
|  7521 | ward   |     30 |
|  7566 | jones  |     20 |
|  7654 | martin |     30 |
|  7698 | blake  |     30 |
|  7788 | scott  |     20 |
|  7844 | turner |     30 |
|  7876 | adams  |     20 |
|  7900 | james  |     30 |
|  7902 | ford   |     20 |
+-------+--------+--------+
11 rows in set (0.00 sec)
```

ASSIGNMENT-3

1.Find the square root of 25

Query :

```
mysql> select sqrt(25);
```

Output:

```
+----------+
| sqrt(25) |
+----------+
|        5 |
+----------+
1 row in set (0.00 sec)
```

2.Find the value of 2 power 3.

Query:

```
mysql> select pow(2,3);
```

Output:

```
+----------+
| pow(2,3) |
+----------+
|        8 |
+----------+
1 row in set (0.02 sec)
```

3.Find the remainder of 11 divided by 3.

Query:

```
mysql> select mod(11,3);
```

Output:

```
+-----------+
| mod(11,3) |
+-----------+
|         2 |
+-----------+
1 row in set (0.00 sec)
```

4.Find the value 3/11(round the result up to 2 digits after the decimal).

Query:

```
mysql> select round(3/11,2);
```

Output:

```
+---------------+
| round(3/11,2) |
+---------------+
|          0.27 |
+---------------+
1 row in set (0.00 sec)
```

5.Round the value of 45.789 to the lowest integer, nearest largest integer.

Query:

```
mysql> select floor(45.789),ceil(45.789);
```

Output:

```
+---------------+--------------+
```

```
| floor(45.789) | ceil(45.789) |
+---------------+--------------+
|            45 |           46 |
+---------------+--------------+
1 row in set (0.11 sec)
```

6.Display the current date in the following format: 26th January 1998.

Query:

```
mysql> select date_format(current_date, '%D %M %Y');
```

Output:

```
+--------------------------------------+
| date_format(current_date, '%D %M %Y') |
+--------------------------------------+
| 13th May 2021                        |
+--------------------------------------+
1 row in set (0.12 sec)
```

7.Find the no of months between today and 12-jan-98.

Query:

```
mysql> select timestampdiff(month, '1998-01-12', Now());
```

Output:

```
+-----------------------------------------+
| timestampdiff(month, '1998-01-12', Now()) |
+-----------------------------------------+
|                                     280 |
+-----------------------------------------+
1 row in set (0.00 sec)
```

8.Find the experience of all the employees in terms of years.

Query:

```
mysql> select ename,timestampdiff(year,hiredate,now()) as experience
from emp;
```

Output:

```
+--------+------------+
| ename  | experience |
+--------+------------+
| smith  |         40 |
| allen  |         40 |
| ward   |         40 |
| jones  |         40 |
| martin |         39 |
| blake  |         40 |
| clark  |         39 |
| scott  |         38 |
| king   |         39 |
| turner |         39 |
| adams  |         38 |
| james  |         39 |
| ford   |         39 |
| miller |         39 |
+--------+------------+
```

```
14 rows in set (0.00 sec)
```

9.Find the people who joined before 87.

Query:

```
mysql> select ename,hiredate from emp where year(hiredate)<1987;
```

Output:

```
+--------+------------+
| ename  | hiredate   |
+--------+------------+
| smith  | 1980-12-17 |
| allen  | 1981-02-20 |
| ward   | 1981-02-22 |
| jones  | 1981-04-02 |
| martin | 1981-09-28 |
| blake  | 1981-05-01 |
| clark  | 1981-06-09 |
| scott  | 1982-12-09 |
| king   | 1981-11-17 |
| turner | 1981-09-08 |
| adams  | 1983-01-12 |
| james  | 1981-12-03 |
| ford   | 1981-12-03 |
| miller | 1982-01-23 |
+--------+------------+
14 rows in set (0.00 sec)
```

10.Find the people who joined in April.

Query:

```
mysql> select empno,ename,hiredate from emp where month(hiredate)=4;
```

Output:

```
+-------+-------+------------+
| empno | ename | hiredate   |
+-------+-------+------------+
|  7566 | jones | 1981-04-02 |
+-------+-------+------------+
1 row in set (0.00 sec)
```

11.Display the employee's names in sentence case, uppercase, lowercase.

Query:

```
mysql> select
concat(upper(substr(ename,1,1)),lower(substr(ename,2,6))) as "sentence
case", upper(ename) as "upper case", lower(ename) as "lower case" from emp;
```

Output:

```
+---------------+------------+------------+
| sentence case | upper case | lower case |
+---------------+------------+------------+
| Smith         | SMITH      | smith      |
| Allen         | ALLEN      | allen      |
| Ward          | WARD       | ward       |
| Jones         | JONES      | jones      |
| Martin        | MARTIN     | martin     |
```

```
| Blake          | BLAKE      | blake      |
| Clark          | CLARK      | clark      |
| Scott          | SCOTT      | scott      |
| King           | KING       | king       |
| Turner         | TURNER     | turner     |
| Adams          | ADAMS      | adams      |
| James          | JAMES      | james      |
| Ford           | FORD       | ford       |
| Miller         | MILLER     | miller     |
+----------------+------------+------------+
14 rows in set (0.10 sec)
```

12.Display first five characters of the string 'abcedfgh'.

Query:

```
mysql> select substr('abcdefgh',1,5);
```

Output:

```
+------------------------+
| substr('abcdefgh',1,5) |
+------------------------+
| abcde                  |
+------------------------+
1 row in set (0.00 sec)
```

13.Find the no of employees.

Query:

```
mysql> select count(*) from emp;
```

Output:

```
+----------+
| count(*) |
+----------+
|       14 |
+----------+
1 row in set (0.00 sec)
```

14.Find the no of departments in the organization.

Query:

```
mysql> select count(*) from dept;
```

Output:

```
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.00 sec)
```

15.Find the no of people who are working in dept no 10.

Query:

```
mysql> select deptno, count(*) from emp where deptno = 10;
```

Output:

```
+--------+----------+
| deptno | count(*) |
```

```
+--------+----------+
|     10 |        3 |
+--------+----------+
1 row in set (0.00 sec)
```

16.Find the no of employees who are drawing the salary more than 3000.

Query:

```
mysql> select count(*) from emp where sal>3000;
```

Output:

```
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)
```

17.Find the total salary, average salary, minimum, maximum salaries of all employee's salaries

Query:

```
mysql> select sum(sal),avg(sal),min(sal),max(sal) from emp;
```

Output:

```
+----------+-----------+----------+----------+
| sum(sal) | avg(sal)  | min(sal) | max(sal) |
+----------+-----------+----------+----------+
|    29025 | 2073.2143 |      800 |     5000 |
+----------+-----------+----------+----------+
1 row in set (0.00 sec)
```

18.Find the total salary of each employee (total sal=sal+comm. If comm is null consider min as 0).

Query:

```
mysql> select empno,ename,sal + ifnull(comm,0) as "total salary" from
emp;
```

Output:

```
+-------+--------+--------------+
| empno | ename  | total salary |
+-------+--------+--------------+
|  7369 | smith  |          800 |
|  7499 | allen  |         1900 |
|  7521 | ward   |         1750 |
|  7566 | jones  |         2975 |
|  7654 | martin |         2650 |
|  7698 | blake  |         2850 |
|  7782 | clark  |         2450 |
|  7788 | scott  |         3000 |
|  7839 | king   |         5000 |
|  7844 | turner |         1500 |
|  7876 | adams  |         1100 |
|  7900 | james  |          950 |
|  7902 | ford   |         3000 |
|  7934 | miller |         1300 |
+-------+--------+--------------+
14 rows in set (0.00 sec)
```

19.Display employee name, salary and the message depending on the salary of employee
=1500 'on target' <1500 'less' >1500 'more
Query:

```
mysql> select ename,sal, "on target" as message from emp where
sal=1500 union select ename,sal, "less" as message from emp where sal<1500
union select ename,sal,"more" as  message from emp where sal>1500;
```

Output:

```
+--------+------+-----------+
| ename  | sal  | message   |
+--------+------+-----------+
| turner | 1500 | on target |
| smith  |  800 | less      |
| ward   | 1250 | less      |
| martin | 1250 | less      |
| adams  | 1100 | less      |
| james  |  950 | less      |
| miller | 1300 | less      |
| allen  | 1600 | more      |
| jones  | 2975 | more      |
| blake  | 2850 | more      |
| clark  | 2450 | more      |
| scott  | 3000 | more      |
| king   | 5000 | more      |
| ford   | 3000 | more      |
+--------+------+-----------+
14 rows in set (0.13 sec)
```

20.Display the list of employees whose names contains 'a' irrespective of the case.
Query:

```
mysql> select ename from emp where ename like '%a%' or ename like
'%A%';
```

Output:

```
+--------+
| ename  |
+--------+
| allen  |
| ward   |
| martin |
| blake  |
| clark  |
| adams  |
| james  |
+--------+
7 rows in set (0.00 sec)
```

LAB CYCLE-2

ASSIGNMENT-1

1.Find the employees whose salary is more than the salary of `SMITH'.

Query:

```
mysql> select ename,sal from emp where sal>(select sal from emp where
ename='smith');
```

Output:

```
+--------+------+
| ename  | sal  |
+--------+------+
| allen  | 1600 |
| ward   | 1250 |
| jones  | 2975 |
| martin | 1250 |
| blake  | 2850 |
| clark  | 2450 |
| scott  | 3000 |
| king   | 5000 |
| turner | 1500 |
| adams  | 1100 |
| james  |  950 |
| ford   | 3000 |
| miller | 1300 |
+--------+------+
13 rows in set (0.01 sec)
```

2.Find the no of employees whose salary is more than their manager's salary.

Query:

```
mysql> select count(e1.ename)from emp e1 where e1.sal>(select e2.sal
from emp e2 where e1.mgr=e2.empno);
```

Output:

```
+-----------------+
| count(e1.ename) |
+-----------------+
|               2 |
+-----------------+
1 row in set (0.01 sec)
```

3.Find the employees whose salary is more than the average salary.

Query:

```
mysql> select e1.ename,e1.sal from emp e1 where e1.sal>(select
avg(e2.sal)from emp e2);
```

Output:

```
+-------+------+
| ename | sal  |
+-------+------+
| jones | 2975 |
| blake | 2850 |
| clark | 2450 |
| scott | 3000 |
| king  | 5000 |
| ford  | 3000 |
+-------+------+
6 rows in set (0.00 sec)
```

4.Find the list of `SALESMAN' whose salary is equal to the maximum salary among them.

Query :

```
      mysql> select e1.ename,e1.job,e1.sal from emp e1 where e1.sal=(select
max(e2.sal)from emp e2 where e2.job='salesman')and e1.job='salesman';
```

Output:

```
+-------+----------+------+
| ename | job      | sal  |
+-------+----------+------+
| allen | salesman | 1600 |
+-------+----------+------+
1 row in set (0.00 sec)
```

5.Find the employees whose experience is more than their managers.

Query:

```
      mysql> select e1.ename,e1.job,e1.hiredate from emp e1 where
e1.hiredate<(select e2.hiredate from emp e2 where e1.mgr=e2.empno);
```

Output:

```
+-------+----------+------------+
| ename | job      | hiredate   |
+-------+----------+------------+
| smith | clerk    | 1980-12-17 |
| allen | salesman | 1981-02-20 |
| ward  | salesman | 1981-02-22 |
| jones | manager  | 1981-04-02 |
| blake | manager  | 1981-05-01 |
| clark | manager  | 1981-06-09 |
+-------+----------+------------+
6 rows in set (0.00 sec)
```

6.Find the employees whose total salary (sal +comm.) is more than the total salary (sal+comm.) of `SMITH'.

Query :

```
mysql> select e.ename,e.job,e.sal+ifnull(e.comm,0)as 'totalsalary'
from emp e where e.sal+ifnull(e.comm,0)>(select e1.sal+ifnull(e1.comm,0)from
emp e1 where e1.ename='smith');
```

Output:

```
+--------+-----------+-------------+
| ename  | job       | totalsalary |
+--------+-----------+-------------+
| allen  | salesman  |        1900 |
| ward   | salesman  |        1750 |
| jones  | manager   |        2975 |
| martin | salesman  |        2650 |
| blake  | manager   |        2850 |
| clark  | manager   |        2450 |
| scott  | analyst   |        3000 |
| king   | president |        5000 |
| turner | salesman  |        1500 |
| adams  | clerk     |        1100 |
| james  | clerk     |         950 |
| ford   | analyst   |        3000 |
| miller | clerk     |        1300 |
+--------+-----------+-------------+
13 rows in set (0.01 sec)
```

7.Find the employees who earn minimum salary in each job and arrange them in descending order of salary.

Query:

```
mysql> select ename,job,sal from emp where sal in(select min(sal)from
emp group by job)order by sal desc;
```

Output:

```
+--------+-----------+------+
| ename  | job       | sal  |
+--------+-----------+------+
| king   | president | 5000 |
| scott  | analyst   | 3000 |
| ford   | analyst   | 3000 |
| clark  | manager   | 2450 |
| ward   | salesman  | 1250 |
| martin | salesman  | 1250 |
| smith  | clerk     |  800 |
+--------+-----------+------+
7 rows in set (0.01 sec)
```

8.Find the name of senior most employees in each department.

Query:

```
mysql> select ename from emp where hiredate in(select
min(hiredate)from emp group by deptno);
```

Output:

```
+--------+
| ename  |
+--------+
| smith  |
| allen  |
| clark  |
+--------+
3 rows in set (0.01 sec)
```

9.Display the departments that have no employees.

Query:

```
      mysql> select deptno,dename from dept where deptno not in(select
distinct deptno from emp);
```

Output:

```
+--------+------------+
| deptno | dename     |
+--------+------------+
|     40 | operations |
+--------+------------+
1 row in set (0.01 sec)
```

10.Write a query to display the message as `early joined' and `late joined' based on the hiredates of the employees comparing with 1- Jan - 82.

Query:

```
      mysql> select e1.ename,e1.job,e1.sal,e1.deptno from emp e1 where
e1.sal=(select max(e2.sal)from emp e2 where e2.deptno=30);
```

Output:

```
+-------+---------+------+--------+
| ename | job     | sal  | deptno |
+-------+---------+------+--------+
| blake | manager | 2850 |     30 |
+-------+---------+------+--------+
1 row in set (0.01 sec)
```

11.Display the list of employees who earn more than every employee's salary in Department 30.

Query:

```
      mysql> select ename from emp where sal>(select max(sal)from emp where
deptno=30);
```

Output:

```
+--------+
| ename  |
+--------+
| jones  |
```

```
| scott |
| king  |
| ford  |
+-------+
4 rows in set (0.00 sec)
```

ASSIGNMENT-2

Queries Using Group by and Having clauses

1.Find the no of employees in each department.

Query:

```
mysql> select deptno,count(*)'NO OF Employers'from emp group by
deptno;
```

Output:

```
+--------+-----------------+
| deptno | NO OF Employers |
+--------+-----------------+
|     10 |               3 |
|     20 |               5 |
|     30 |               6 |
+--------+-----------------+
3 rows in set (0.00 sec)
```

2.Display the lists of departments in which at least 4 employees are working.

Query:

```
mysql> select deptno,count(*)'No Of Employers'from emp group by deptno
having count(*)>=4;
```

Output:

```
+--------+-----------------+
| deptno | No Of Employers |
+--------+-----------------+
|     20 |               5 |
|     30 |               6 |
+--------+-----------------+
2 rows in set (0.00 sec)
```

3.Find the no of employees by deptno,job.

Query:

```
mysql> select deptno,job,count(*)'No of Employers' from emp group by
deptno,job;
```

Output:

```
+--------+-----------+-----------------+
| deptno | job       | No of Employers |
+--------+-----------+-----------------+
|     10 | clerk     |               1 |
|     10 | manager   |               1 |
|     10 | president |               1 |
|     20 | analyst   |               2 |
|     20 | clerk     |               2 |
|     20 | manager   |               1 |
|     30 | clerk     |               1 |
```

```
|     30 | manager   |               1 |
|     30 | salesman  |               4 |
+--------+-----------+-----------------+
9 rows in set (0.00 sec)
```

4.Find the list of employees who are getting the maximum salaries in their respective departments.

Query:

```
      mysql> select e.ename,e.deptno,e.sal from emp e
    -> where e.sal=(select max(e1.sal)from emp e1
    -> where e.deptno=e1.deptno group by e1.deptno);
```

Output:

```
+-------+--------+------+
| ename | deptno | sal  |
+-------+--------+------+
| blake |     30 | 2850 |
| scott |     20 | 3000 |
| king  |     10 | 5000 |
| ford  |     20 | 3000 |
+-------+--------+------+
4 rows in set (0.00 sec)
```

5.Find the employees who are getting more than the average salary of their respective departments.

Query:

```
      mysql> select e.ename,e.deptno,e.sal from emp e
    -> where e.sal>(select avg(e1.sal)from emp e1
    -> where e.deptno=e1.deptno group by e1.deptno);
```

Output:

```
+-------+--------+------+
| ename | deptno | sal  |
+-------+--------+------+
| allen |     30 | 1600 |
| jones |     20 | 2975 |
| blake |     30 | 2850 |
| scott |     20 | 3000 |
| king  |     10 | 5000 |
| ford  |     20 | 3000 |
+-------+--------+------+
6 rows in set (0.00 sec)
```

6.Display employee names, department name in which they are working.

Query:

```
      mysql> select ename,dename from emp join dept using(deptno);
```

Output:

```
+--------+------------+
| ename  | dename     |
```

```
+--------+------------+
| clark  | accounting |
| king   | accounting |
| miller | accounting |
| smith  | research   |
| jones  | research   |
| scott  | research   |
| adams  | research   |
| ford   | research   |
| allen  | sales      |
| ward   | sales      |
| martin | sales      |
| blake  | sales      |
| turner | sales      |
| james  | sales      |
+--------+------------+
14 rows in set (0.00 sec)
```

7.Display the list of employees who are working in the same city in which `King' is working.

Query:

```
      mysql> select empno,ename,location from emp natural join dept
    -> where location=(select location from emp natural join dept
    -> where ename='king');
```

Output:

```
+-------+--------+----------+
| empno | ename  | location |
+-------+--------+----------+
|  7782 | clark  | new york |
|  7839 | king   | new york |
|  7934 | miller | new york |
+-------+--------+----------+
3 rows in set (0.00 sec)
```

8.Display no of employees working in each city.

Output:

```
      mysql> select location,count(empno)'No Of Employers'from
    -> emp natural join dept
    -> group by location;
```

Output:

```
+----------+-----------------+
| location | No Of Employers |
+----------+-----------------+
| chicago  |               6 |
| dallas   |               5 |
```

```
| new york |               3 |
+----------+-----------------+
3 rows in set (0.00 sec)
```

9.Display the city with the highest no of employees.

Query:

```
      mysql> select location,count(*)'No of Employers'from emp natural join
dept
    -> group by location
    -> having count(empno)>=all(select count(empno)from emp natural join
dept
    -> group by location);
```

Output:

```
+----------+-----------------+
| location | No of Employers |
+----------+-----------------+
| chicago  |               6 |
+----------+-----------------+
1 row in set (0.01 sec)
```

10.Display the no of employees departments wise in each city.

Query:

```
      mysql> select deptno,dename,location,count(empno)'No Of Employers'from
emp natural join dept
    -> group by deptno,location;
```

Output:

```
+--------+------------+----------+-----------------+
| deptno | dename     | location | No Of Employers |
+--------+------------+----------+-----------------+
|     10 | accounting | new york |               3 |
|     20 | research   | dallas   |               5 |
|     30 | sales      | chicago  |               6 |
+--------+------------+----------+-----------------+
3 rows in set (0.00 sec)
```

11.Display the list of employees who are working in either `NEWYORK' or DELHI'.

Query:

```
      mysql> select empno,ename,location from emp natural join dept
    -> where location='new york' or location='delhi';
```

Output:

```
+-------+--------+----------+
| empno | ename  | location |
+-------+--------+----------+
|  7782 | clark  | new york |
```

```
|  7839 | king   | new york |
|  7934 | miller | new york |
+-------+--------+----------+
3 rows in set (0.00 sec)
```

12.Find the list employees who are not staying in the same city in which their manager stays.

Query:

```
      mysql> select empno,ename,location from emp e natural join dept
    -> where location!=(select location from emp e1 natural join dept
    -> where e.mgr=e1.empno);
```

Output:

```
+-------+-------+----------+
| empno | ename | location |
+-------+-------+----------+
|  7566 | jones | dallas   |
|  7698 | blake | chicago  |
+-------+-------+----------+
2 rows in set (0.00 sec)
```

VIEWS

13.Create views on emp table.

Query:

```
      mysql> create view v1 as(select *from emp where sal>=900);
Query OK, 0 rows affected (0.01 sec)
      mysql> select *from v1;
```

Output:

```
+-------+--------+-----------+------+------------+------+------+--------+
| empno | ename  | job       | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+-----------+------+------------+------+------+--------+
|  7499 | allen  | salesman  | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | ward   | salesman  | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | jones  | manager   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7654 | martin | salesman  | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7698 | blake  | manager   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | clark  | manager   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7788 | scott  | analyst   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | king   | president | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7844 | turner | salesman  | 7698 | 1981-09-08 | 1500 |    0 |     30 |
|  7876 | adams  | clerk     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7900 | james  | clerk     | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7902 | ford   | analyst   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7934 | miller | clerk     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
13 rows in set (0.01 sec)
```

14.Queries on views.

Query:

```
      mysql> create view v2 as(select empno,job,sal from emp where
sal<1500);
Query OK, 0 rows affected (0.01 sec)
      mysql> select *from v2;
```

Output:

```
+-------+----------+------+
| empno | job      | sal  |
+-------+----------+------+
|  7369 | clerk    |  800 |
|  7521 | salesman | 1250 |
|  7654 | salesman | 1250 |
|  7876 | clerk    | 1100 |
|  7900 | clerk    |  950 |
|  7934 | clerk    | 1300 |
+-------+----------+------+
6 rows in set (0.00 sec)
```

LAB CYCLE-3
FUNCTIONS & PROCEDURES

1.Write a program to print the multiplication table of a given number up to 10 terms.

Query:

```
mysql> delimiter //
mysql> create procedure mult(in n int)
    -> begin
    -> declare i int default 1;
    -> declare str varchar(500) default '';
    -> declare s varchar(50);
    -> while(i<=10) do
    -> set s=concat(n,"*",i,"=",n*i);
    -> set str=concat(str,'\n',s);
    -> set i=i+1;
    -> end while;
    -> select str;
    -> end;
    -> //
Query OK, 0 rows affected (0.02 sec)
```

Output:

```
mysql> call mult(5);//
+-------------------------------------------------------------------------+
| str                                                                     |
+-------------------------------------------------------------------------+
|
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50 |
+-------------------------------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```

2.Write a program to print all prime numbers between a range.

Query:

```
mysql> delimiter //
mysql> create procedure primeno(in a int,in b int)
    -> begin
    -> declare x int default 0;
    -> declare i int default 1;
    -> declare str varchar(500) default '';
    -> while(a<b) do
    -> set i=1;
```

```
          -> set x=0;
          -> while(i<=a) do
          -> if(a%i=0) then
          -> set x=x+1;
          -> end if;
          -> set i=i+1;
          -> end while;
          -> if(x=2) then
          -> set str=concat(str,"   ",a);
          -> end if;
          -> set a=a+1;
          -> end while;
          -> select str;
          -> end;
          -> //
Query OK, 0 rows affected (0.15 sec)
```

Output:

```
      mysql> call primeno(2,20)//
+-------------------------------------+
| str                                 |
+-------------------------------------+
|    2   3   5   7   11   13   17   19 |
+-------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

3.Write a program to print the reverse number of a given number.

Query:

```
      mysql> delimiter //
      mysql> create procedure rev(in n int)
          -> begin
          -> declare r int default 0;
          -> declare i int;
          -> while n!=0 do
          -> set i=n%10;
          -> set r=(r*10)+i;
          -> set n=floor(n/10);
          -> end while;
          -> select r;
          -> end;
          -> //
Query OK, 0 rows affected (0.01 sec)
```

Output:

```
mysql> call rev(6784)
    -> //
+------+
| r    |
+------+
| 4876 |
+------+
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

4. Write a program to update salary of the employee based on the following conditions
If salary is less than 2000 increment by 15%
>2000 & <3000 increase by 10% otherwise 10%.

Query:

```
mysql> delimiter //
mysql> create procedure updatesal(in eno int)
    -> begin
    -> declare sal1 int;
    -> select sal into sal1 from emp where empno=eno;
    -> if(sal1<2000) then
    -> set sal1=sal1+sal1*0.15;
    -> elseif(sal1>2000 && sal1<3000) then
    -> set sal1=sal1+sal1*0.10;
    -> else
    -> set sal1=sal1+sal1*0.05;
    -> end if;
    -> select eno,sal1;
    -> end;
    -> //
Query OK, 0 rows affected (0.00 sec)
```

Output:

```
mysql> call updatesal(7788)
    -> //
+------+------+
| eno  | sal1 |
+------+------+
| 7788 | 3150 |
+------+------+
1 row in set (0.06 sec)

Query OK, 0 rows affected (0.07 sec)
```

5. Write a program that returns the total gross salary of a given department.

Query:

```
mysql> delimiter //
mysql> create function totalgrosssal(dno int)
    -> returns int
    -> begin
    -> declare gsal int;
    -> select sum(sal) into gsal from emp where deptno=dno;
    -> return gsal;
    -> end;
    -> //
Query OK, 0 rows affected (0.00 sec)
```

Output:

```
mysql> select totalgrosssal(20)//
+------------------+
| totalgrosssal(20) |
```

```
+-------------------+
|             10875 |
+-------------------+
1 row in set (0.00 sec)
```

6.Write a program that returns no of employees, total salary of a given department.

Query:

```
mysql> delimiter //
mysql> create function employees(dno int)
    -> returns varchar(30)
    -> begin
    -> declare count int;
    -> declare grosssal1 int;
    -> select count(*),sum(sal) into count,grosssal1 from emp where
deptno=dno;
    -> return concat(count,"",grosssal1);
    -> end;
    -> //
Query OK, 0 rows affected (0.00 sec)
```

Output:

```
mysql> select employees(30);//
+---------------+
| employees(30) |
+---------------+
| 69400         |
+---------------+
1 row in set (0.00 sec)
```

CURSORS

1. Write a program to print employee's details in the following format.

Emp Name total salary

Calculate total salary as (sal* 12+comm.*12).

Query:

```
mysql> delimiter //
mysql> create procedure p01()
    ->       begin
    ->      declare done int default 0;
    ->      declare e_name varchar(10);
    ->      declare e_id,e_sal int;
    ->      declare res varchar(350) default ' ';
    ->      declare name_sal cursor for
    ->      select  empno,ename,(sal*12+ifnull(comm,0)*12) from emp;
    ->      declare continue handler for not found set done=1;
    ->      open name_sal;
    ->      get_sal:loop
    ->      fetch name_sal into e_id,e_name,e_sal;
    ->      if done=1 then
    ->     leave get_sal;
    ->     end if;
    ->     set res=concat(e_id,'\t',e_name,'\t',e_sal,'\n',res);
```

```
          ->     end loop get_sal;
          ->     select res;
          ->     close name_sal;
          ->     end//
      Query OK, 0 rows affected (0.24 sec)
```

Output:

```
mysql> call p01();//
+------------------------------------------------+
| res
|
+------------------------------------------------+
| 7934  miller  1430
7934    miller  15600
7902    ford    36000
7900    james   11400
7876    adams   13200
7844    turner  18000
7839    king    60000
7788    scott   36000
7782    clark   29400
7698    blake   34200
7654    martin  31800
7566    jones   35700
7521    ward    21000
7499    allen   22800
7369    smith   9600
+------------------------------------------------+

|
Query OK, 0 rows affected (0.02 sec)
```

2.Write a program to update the salary of the employee by 10% if the salary is less than 2000.

Query:

```
        mysql>  create procedure p02()
    ->        begin
    ->        declare done int default 0;
    ->       declare e_id,e_sal int;
    ->       declare e_name varchar(300);
    ->       declare res varchar(350) default ' ';
    ->       declare up_sal cursor for
    ->       select empno,ename,(sal+sal*0.1) from emp where sal<2000;
    ->       declare continue handler for not found set done=1;
    ->       open up_sal;
    ->       get_sal:loop
    ->       fetch up_sal into e_id,e_name,e_sal;
    ->       if done=1 then
    ->       leave get_sal;
    ->       end if;
    ->       set res=concat(e_id,'\t',e_name,'\t',e_sal,'\n',res);
    ->       end loop get_sal;
    ->        select res;
    ->        close up_sal;
```

```
    ->        end//
Query OK, 0 rows affected (0.15 sec)
```
Output:
```
mysql> call p02();
    -> //
+-------------------------------------------------+
| res
|
+-------------------------------------------------+
| 7934  miller  1430
7900    james   1045
7876    adams   1210
7844    turner  1650
7654    martin  1375
7521    ward    1375
7499    allen   1760
7369    smith   880
  |
+-------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```

TRIGGERS

1.write a program for insert trigger.

Query:
```
mysql> create table student(sid int primary key,sname varchar(20),deptno
int,dname varchar(20),percentage float,age int);
mysql> desc student;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| sid        | int(11)     | NO   | PRI | NULL    |       |
| sname      | varchar(20) | YES  |     | NULL    |       |
| deptno     | int(11)     | YES  |     | NULL    |       |
| dname      | varchar(20) | YES  |     | NULL    |       |
| percentage | float       | YES  |     | NULL    |       |
| age        | int(11)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.11 sec)
```
Insert trigger:
```
      mysql> delimiter //
      mysql> create trigger t1 before insert on student for each row
    -> begin
    -> if(new.age<18) then
    -> call invalidage();
    -> end if;
    -> end;
    -> //
Query OK, 0 rows affected (0.06 sec)
```
Output:

```
mysql> insert into student values(11,'sheetal',10,'CSE',84.5,24);//
Query OK, 1 row affected (0.11 sec)

mysql> insert into student values(12,'meghna',20,'IT',96.2,20);//
Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(13,'gayathriA',30,'ECE',86.4,19);//
Query OK, 1 row affected (0.01 sec)
```

2.write a program for delete trigger.

Query:

```
      mysql> create trigger t2 before delete on student
    -> for each row
    -> begin
    -> insert into student_log
values(old.sid,old.sname,old.deptno,old.dname,old.percentage,old.age);
    -> end;
    -> //
Query OK, 0 rows affected (0.01 sec)
```

Output:

```
mysql> select * from student;
    -> //
+-----+-----------+--------+-------+------------+------+
| sid | sname     | deptno | dname | percentage | age  |
+-----+-----------+--------+-------+------------+------+
|  11 | sheetal   |     10 | CSE   |       84.5 |   24 |
|  12 | meghna    |     20 | IT    |       96.2 |   20 |
|  13 | gayathriA |     30 | ECE   |       86.4 |   19 |
+-----+-----------+--------+-------+------------+------+
3 rows in set (0.00 sec)

mysql> delete from student where sid=12;
    -> //
Query OK, 1 row affected (0.02 sec)

mysql> select * from student;
    -> //
+-----+-----------+--------+-------+------------+------+
| sid | sname     | deptno | dname | percentage | age  |
+-----+-----------+--------+-------+------------+------+
|  11 | sheetal   |     10 | CSE   |       84.5 |   24 |
|  13 | gayathriA |     30 | ECE   |       86.4 |   19 |
+-----+-----------+--------+-------+------------+------+
2 rows in set (0.00 sec)

mysql> select * from student_log;
    -> //
+-----+--------+--------+-------+------------+------+
| sid | sname  | deptno | dname | percentage | age  |
+-----+--------+--------+-------+------------+------+
|  12 | meghna |     20 | IT    |       96.2 |   20 |
+-----+--------+--------+-------+------------+------+
```

```
1 row in set (0.00 sec)
```

3 . write a program for update trigger.

Query:

```
mysql>  create table student_upd(sid int primary key,sname
varchar(20),deptno int,dname varchar(20),percentage float,age int);
    -> //
Query OK, 0 rows affected (0.04 sec)

mysql> create trigger t3 before update on student
    -> for each row
    -> begin
    -> insert into student_upd
values(old.sid,old.sname,old.deptno,old.dname,old.percentage,old.age);
    -> end;
    -> //
Query OK, 0 rows affected (0.01 sec)
```

Output:

```
mysql> insert into student values(15,'jeevana',30,'ECE',86.3,20);
    -> //
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(16,'manvi',40,'EEE',80.4,21);
    -> //
Query OK, 1 row affected (0.01 sec)

mysql>  update student set age=21 where sname='mansi';
    -> //
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
    -> //
+-----+-----------+--------+-------+------------+------+
| sid | sname     | deptno | dname | percentage | age  |
+-----+-----------+--------+-------+------------+------+
|  11 | sheetal   |     10 | CSE   |       84.5 |   24 |
|  13 | gayathriA |     30 | ECE   |       86.4 |   19 |
|  15 | jeevana   |     30 | ECE   |       86.3 |   20 |
|  16 | manvi     |     40 | EEE   |       80.4 |   21 |
+-----+-----------+--------+-------+------------+------+
4 rows in set (0.00 sec)

mysql> select * from student_upd;
    -> //
+-----+-------+--------+-------+------------+------+
| sid | sname | deptno | dname | percentage | age  |
+-----+-------+--------+-------+------------+------+
|  16 | manvi |     40 | EEE   |       80.4 |   21 |
+-----+-------+--------+-------+------------+------+
1 row in set (0.00 sec)
```

LAB CYCLE - 4

Case Study

Bus Reservation System:

Develop an online reservation system Roadway Travels database by using MySQL.This system should allow the users to do reservations by using credit or debit card through internet and can also enquire about the availability of buses, timings and seats.

The above process involves many steps.

1. Analyzing the problem and identifying the entities and relationships.
2. ER Model
3. Creation of Tables /relations as per the case study.
4. Normalization
5. Inserting data into the tables.
6. Practice the following Queries:
   a. Display unique PPNO of all passengers.
   b. Display all the names of male passengers.
   c. Display the ticket numbers and names of all the passengers.
   d. Find the names of passengers whose age is between 30 and 45.
   e. Display the sorted list of passenger's names.
   f. Write a Query to display the Information present in the Passenger and cancellation tables.
   g. Find the total number of cancelled seats.

7. Design and develop the following:
   i. Trigger ii. Cursor

creating database

```
mysql> create database busreservation;
Query OK, 1 row affected (0.22 sec)

mysql> use busreservation;
Database changed
```

creating tables

bus table:

```
mysql>  create table bus(bid varchar(10) primary key,bname
varchar(15),source varchar(15),destination varchar(15),cost int,j_date
date,avail int,alloc int);
Query OK, 0 rows affected (1.88 sec)

mysql>  desc bus;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| bid         | varchar(10) | NO   | PRI | NULL    |       |
| bname       | varchar(15) | YES  |     | NULL    |       |
| source      | varchar(15) | YES  |     | NULL    |       |
| destination | varchar(15) | YES  |     | NULL    |       |
| cost        | int         | YES  |     | NULL    |       |
| j_date      | date        | YES  |     | NULL    |       |
| avail       | int         | YES  |     | NULL    |       |
| alloc       | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
8 rows in set (0.18 sec)
```

Passenger table:

```
mysql>  create table passenger(pid varchar(10) primary key,pname
varchar(10),gender varchar(6),age int,mailid varchar(25),mobileno long);
Query OK, 0 rows affected (0.97 sec)

mysql>  desc passenger;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| pid      | varchar(10) | NO   | PRI | NULL    |       |
| pname    | varchar(10) | YES  |     | NULL    |       |
| gender   | varchar(6)  | YES  |     | NULL    |       |
| age      | int         | YES  |     | NULL    |       |
| mailid   | varchar(25) | YES  |     | NULL    |       |
| mobileno | mediumtext  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Reservation table:

```
mysql>  create table reservation(rid int auto_increment primary key,bid
varchar(10),pid varchar(10),j_date1 date,res_date date,nosa int,source1
```

```
varchar(15),destination1 varchar(15),status varchar(10),foreign key(bid)
references bus(bid),foreign key(pid) references passenger(pid));
Query OK, 0 rows affected (1.04 sec)

mysql> desc reservation;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| rid          | int         | NO   | PRI | NULL    | auto_increment |
| bid          | varchar(10) | YES  | MUL | NULL    |                |
| pid          | varchar(10) | YES  | MUL | NULL    |                |
| j_date1      | date        | YES  |     | NULL    |                |
| res_date     | date        | YES  |     | NULL    |                |
| nosa         | int         | YES  |     | NULL    |                |
| source1      | varchar(15) | YES  |     | NULL    |                |
| destination1 | varchar(15) | YES  |     | NULL    |                |
| status       | varchar(10) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)
```

Ticket table:

```
mysql>  create table ticket(tid int auto_increment primary key,bid
varchar(10),pid varchar(10),j_date2 date,source2 varchar(15),destination2
varchar(15),nosa1 int,amount int,foreign key(bid) references
bus(bid),foreign key(pid) references passenger(pid));
Query OK, 0 rows affected (1.90 sec)

mysql> desc ticket;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| tid          | int         | NO   | PRI | NULL    | auto_increment |
| bid          | varchar(10) | YES  | MUL | NULL    |                |
| pid          | varchar(10) | YES  | MUL | NULL    |                |
| j_date2      | date        | YES  |     | NULL    |                |
| source2      | varchar(15) | YES  |     | NULL    |                |
| destination2 | varchar(15) | YES  |     | NULL    |                |
| nosa1        | int         | YES  |     | NULL    |                |
| amount       | int         | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
8 rows in set (0.04 sec)
```

```
mysql>  insert into bus values('B101','Fast
Express','Hyderabad','vijayawada',900,'2021-07-15',30,0);
Query OK, 1 row affected (0.20 sec)

mysql>  insert into bus values('B102','Sai
Express','Hyderabad','shiridi',2000,'2021-07-15',30,0);
Query OK, 1 row affected (0.10 sec)

mysql> insert into bus values('B103','Super
luxury','Hyderabad','bangalore',900,'2021-07-15',30,0);
Query OK, 1 row affected (1.15 sec)

mysql>  insert into bus values('B104','super
luxury','Hyderabad','srisailam',600,'2021-07-15',30,0);
Query OK, 1 row affected (0.11 sec)

mysql> insert into bus values('B105','venkat
express','Hyderabad','tirupathi',1000,'2021-07-15',30,0);
Query OK, 1 row affected (0.17 sec)

mysql> select * from bus;
+------+----------------+-----------+-------------+------+------------+--
-----+-------+
| bid  | bname          | source    | destination | cost | j_date     |
avail | alloc |
+------+----------------+-----------+-------------+------+------------+--
-----+-------+
| B101 | Fast Express   | Hyderabad | vijayawada  |  900 | 2021-07-15 |
30 |     0 |
| B102 | Sai Express    | Hyderabad | shiridi     | 2000 | 2021-07-15 |
30 |     0 |
| B103 | Super luxury   | Hyderabad | bangalore   |  900 | 2021-07-15 |
30 |     0 |
| B104 | super luxury   | Hyderabad | srisailam   |  600 | 2021-07-15 |
30 |     0 |
| B105 | venkat express | Hyderabad | tirupathi   | 1000 | 2021-07-15 |
30 |     0 |
+------+----------------+-----------+-------------+------+------------+--
-----+-------+
5 rows in set (0.00 sec)

mysql>  insert into passenger
values('P101','Ajay','M',28,'ajaysgr@gmail.com',9866346365);
```

```
Query OK, 1 row affected (0.92 sec)

mysql>  insert into passenger
values('P102','Arun','M',25,'arunsgr@gmail.com',9865678365);
Query OK, 1 row affected (0.08 sec)

mysql> insert into passenger
values('P103','Akhila','F',20,'akhi@gmail.com',98786554655);
Query OK, 1 row affected (0.07 sec)

mysql> insert into passenger
values('P104','chintu','F',21,'chintu@gmail.com',9875554655);
Query OK, 1 row affected (0.15 sec)

mysql> insert into passenger
values('P105','sai','M',19,'sai@gmail.com',9875554655);
Query OK, 1 row affected (0.21 sec)

mysql>  select * from passenger;
+------+--------+--------+------+-------------------+-------------+
| pid  | pname  | gender | age  | mailid            | mobileno    |
+------+--------+--------+------+-------------------+-------------+
| P101 | Ajay   | M      |   28 | ajaysgr@gmail.com | 9866346365  |
| P102 | Arun   | M      |   25 | arunsgr@gmail.com | 9865678365  |
| P103 | Akhila | F      |   20 | akhi@gmail.com    | 98786554655 |
| P104 | chintu | F      |   21 | chintu@gmail.com  | 9875554655  |
| P105 | sai    | M      |   19 | sai@gmail.com     | 9875554655  |
+------+--------+--------+------+-------------------+-------------+
5 rows in set (0.00 sec)

mysql> Create trigger bf_ins before insert on reservation
    -> For each row
    -> Begin
    -> Declare a,c int;
    -> Declare src,dest varchar(15);
    -> Select avail,alloc,source,destination into a,c,src,dest from bus
    -> Where bid=new.bid;
    -> If(a<new.nosa) then
    -> Call not_sufficient_seats();
    -> Else if(src!=new.source1 or dest!=new.destination1) then
    -> Call service_not_available_select_another_bus();
    -> Else
    -> Set a=a-new.nosa;
```

```
    -> Set c=c+new.nosa;
    -> Update bus set avail=a,alloc=c
    -> Where bid=new.bid;
    -> End if;
    -> End if;
    -> End;
    -> //
Query OK, 0 rows affected (0.24 sec)

mysql> Create trigger af_ins after insert on reservation
    -> For each row
    -> Begin
    -> Declare c1 int;
    -> Select cost into c1 from bus
    -> Where bid=new.bid;
    -> Set c1=c1*new.nosa;
    -> Insert into
ticket(bid,pid,j_date2,source2,destination2,nosa1,amount)
    ->
Values(new.bid,new.pid,new.j_date1,new.source1,new.destination1,new.nosa,
c1);
    -> End //
Query OK, 0 rows affected (0.14 sec)

mysql> insert into
reservation(bid,pid,j_date1,res_date,nosa,source1,destination1,status)val
ues('B101','P101','2021-07-
15',sysdate(),5,'Hyderabad','vijayawada','confirmed');//
Query OK, 1 row affected, 1 warning (0.40 sec)

mysql> select * from reservation;
    -> //
+-----+------+------+------------+------------+------+-----------+-------------+-----------+
| rid | bid  | pid  | j_date1    | res_date   | nosa | source1   | destination1 | status    |
+-----+------+------+------------+------------+------+-----------+-------------+-----------+
|   1 | B101 | P101 | 2021-07-15 | 2021-07-15 |    5 | Hyderabad | vijayawada   | confirmed |
+-----+------+------+------------+------------+------+-----------+-------------+-----------+
1 row in set (0.00 sec)

mysql> select * from ticket;//
+-----+------+------+------------+-----------+--------------+-------+--------+
| tid | bid  | pid  | j_date2    | source2   | destination2 | nosa1 | amount |
+-----+------+------+------------+-----------+--------------+-------+--------+
|   1 | B101 | P101 | 2021-07-15 | Hyderabad | vijayawada   |     5 |   4500 |
+-----+------+------+------------+-----------+--------------+-------+--------+
```

```
1 row in set (0.00 sec)

mysql> create table cancellation(cid int auto_increment primary key,tid
int,bid varchar(30),pid varchar(10),rid int,c_date date,nosc int,charge
int,foreign key(bid) references bus(bid),foreign key(rid) references
reservation(rid),foreign key(pid) references passenger(pid),foreign
key(tid) references ticket(tid));//
Query OK, 0 rows affected (2.83 sec)

mysql> desc cancellation;//
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| cid    | int         | NO   | PRI | NULL    | auto_increment |
| tid    | int         | YES  | MUL | NULL    |                |
| bid    | varchar(30) | YES  | MUL | NULL    |                |
| pid    | varchar(10) | YES  | MUL | NULL    |                |
| rid    | int         | YES  | MUL | NULL    |                |
| c_date | date        | YES  |     | NULL    |                |
| nosc   | int         | YES  |     | NULL    |                |
| charge | int         | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
8 rows in set (0.12 sec)

mysql> Create trigger bf_ins1 before insert on cancellation
    -> For each row
    -> Begin
    -> Declare n,a1,a2,c,m int;
    -> Declare jd date;
    -> Select nosa1,j_date2,amount into n,jd,m from ticket where
tid=new.tid;
    -> Select cost,avail,alloc into c,a1,a2 from bus Where bid=new.bid;
    -> If(jd<new.c_date) then
    -> Call cancellation_not_possible();
    -> Else
    -> Set n=n-new.nosc;
    -> Set m=m-c*new.nosc;
    -> If(n<=0) then
    -> Call cancellation_not_possible();
    -> Update reservation set nosa=n where pid=new.pid and bid=new.bid;
    -> Update ticket set nosa1=n,amount=m where tid=new.tid;
    -> Set a1=a1+new.nosc;
    -> Set a2=a2-new.nosc;
```

```
    -> Update bus set avail=a1,alloc=a2 where bid=new.bid;
    -> End if;
    -> End if;
    -> End;
    -> //
Query OK, 0 rows affected (0.16 sec)

mysql> insert into
cancellation(tid,bid,pid,rid,c_date,nosc,charge)values(1,'B101','P101',1,
sysdate(),3,nosc*20);//
Query OK, 1 row affected, 1 warning (0.13 sec)

mysql> select * from cancellation;//
+-----+------+------+------+------+------------+------+--------+
| cid | tid  | bid  | pid  | rid  | c_date     | nosc | charge |
+-----+------+------+------+------+------------+------+--------+
|   1 |    1 | B101 | P101 |    1 | 2021-07-15 |    3 |     60 |
+-----+------+------+------+------+------------+------+--------+
1 row in set (0.02 sec)

mysql> select * from reservation;//
+-----+------+------+------------+------------+------+----------+-------------+-----------+
| rid | bid  | pid  | j_date1    | res_date   | nosa | source1  | destination1 | status    |
+-----+------+------+------------+------------+------+----------+-------------+-----------+
|   1 | B101 | P101 | 2021-07-15 | 2021-07-15 |    5 | Hyderabad | vijayawada   | confirmed |
+-----+------+------+------------+------------+------+----------+-------------+-----------+
1 row in set (0.00 sec)

mysql> select * from ticket;//
+-----+------+------+------------+-----------+-------------+-------+--------+
| tid | bid  | pid  | j_date2    | source2   | destination2 | nosa1 | amount |
+-----+------+------+------------+-----------+-------------+-------+--------+
|   1 | B101 | P101 | 2021-07-15 | Hyderabad | vijayawada   |     5 |   4500 |
+-----+------+------+------------+-----------+-------------+-------+--------+
1 row in set (0.00 sec)

mysql> select * from bus;//
+------+----------------+-----------+-------------+------+------------+-------+------
-+
| bid  | bname          | source    | destination | cost | j_date     | avail | alloc
|
+------+----------------+-----------+-------------+------+------------+-------+------
-+
| B101 | Fast Express   | Hyderabad | vijayawada  |  900 | 2021-07-15 |    25 |     5
|
| B102 | Sai Express    | Hyderabad | shiridi     | 2000 | 2021-07-15 |    30 |     0
|
```

```
| B103 | Super luxury   | Hyderabad | bangalore   |  900 | 2021-07-15 |    30 |     0
|
| B104 | super luxury   | Hyderabad | srisailam   |  600 | 2021-07-15 |    30 |     0
|
| B105 | venkat express | Hyderabad | tirupathi   | 1000 | 2021-07-15 |    30 |     0
|
+------+----------------+-----------+-------------+------+------------+-------+------
-+
   5  rows in set (0.00 sec)
```

a.Display unique PPNO of all passengers.

Query:

```
mysql> select pid from passenger;
```

Output:

```
+------+
| pid  |
+------+
| P101 |
| P102 |
| P103 |
| P104 |
| P105 |
+------+
5 rows in set (0.00 sec)
```

b. Display all the names of male passengers.

Query:

```
mysql> select pid,pname,gender from passenger where gender='M';
```

Output:

```
+------+-------+--------+
| pid  | pname | gender |
+------+-------+--------+
| P101 | Ajay  | M      |
| P102 | Arun  | M      |
| P105 | sai   | M      |
+------+-------+--------+
   3  rows in set (0.33 sec)
```

c. Display the ticket numbers and names of all the passengers

Query:

```
mysql> select pid,pname,tid from passenger natural join ticket;
+------+-------+-----+
| pid  | pname | tid |
+------+-------+-----+
| P101 | Ajay  |   1 |
+------+-------+-----+
1 row in set (0.04 sec)
```

d. Find the names of passengers whose age is between 20 and 25.

Query:

```
mysql> select pid,pname,age from passenger where age between 20 and 25;
```

Output:

```
+------+--------+------+
| pid  | pname  | age  |
+------+--------+------+
| P102 | Arun   |   25 |
| P103 | Akhila |   20 |
| P104 | chintu |   21 |
+------+--------+------+
3 rows in set (0.12 sec)
```

e. Display the sorted list of passenger's names.

Query:

```
mysql> select pid,pname from passenger order by pname;
```

Output:

```
+------+--------+
| pid  | pname  |
+------+--------+
| P101 | Ajay   |
| P103 | Akhila |
| P102 | Arun   |
| P104 | chintu |
| P105 | sai    |
+------+--------+
5 rows in set (0.10 sec)
```

f. Write a Query to display the Information present in the Passenger and cancellation tables.

Query:

```
mysql> select * from passenger natural join cancellation;
```

Output:

```
+------+-------+--------+------+-------------------+------------+-----+------+------
+------+------------+------+--------+
| pid  | pname | gender | age  | mailid            | mobileno   | cid | tid  | bid  |
rid  | c_date     | nosc | charge |
+------+-------+--------+------+-------------------+------------+-----+------+------
+------+------------+------+--------+
| P101 | Ajay  | M      |   28 | ajaysgr@gmail.com | 9866346365 |   1 |    1 | B101 |
1 | 2021-07-15 |    3 |     60 |
+------+-------+--------+------+-------------------+------------+-----+------+------
+------+------------+------+--------+
1 row in set (0.18 sec)
```

g. Find the total number of cancelled seats.

Query:

```
mysql> select sum(nosc) from cancellation;
```

Output:

```
+------------+
| sum(nosc) |
+------------+
|          3 |
+------------+
1 row in set (0.01 sec)
```