TASK-2

> Create a Database name entri_assignment.

```
mysql> create database entri_assignment;
Query OK, 1 row affected (0.12 sec)

mysql> use entri_assignment;
Database changed
```

> Create a Table with name departments (Department_id (pk), Department_name, Location_id).

```
mysql> create table departments
    -> (Department_id int primary key, Department_name varchar(50)not null, Location_id int);
Query OK, 0 rows affected (1.65 sec)

mysql> describe departments;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| Department_id   | int         | NO   | PRI | NULL    |       |
| Department_name | varchar(50) | NO   |     | NULL    |       |
| Location_id     | int         | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
3 rows in set (0.07 sec)
```

> Create a Table with name employees {Employee_id (pk) ,first_name,last_name ,email,phone_number,hire_date, job_id, salary, commission_pct, manager_id, department_id (fk reference to departments)}.

```
mysql> create table employees
    -> (Employee_id int primary key,first_name varchar(50),last_name varchar(50),email varchar(40),
    -> phone_number varchar(50),hire_date date,job_id varchar(40),salary int,
    -> commission_pct int ,manager_id int, department_id int, foreign key(department_id)references departments(Department_id));
Query OK, 0 rows affected (2.41 sec)

mysql> show tables;
+---------------------------+
| Tables_in_entri_assignment |
+---------------------------+
| departments               |
| employees                 |
+---------------------------+
2 rows in set (0.00 sec)
```

1. Select employees first name, last name, job_id and salary whose first name starts with alphabet S.

```
mysql> select first_name, last_name, job_id ,salary
    -> from employees
    -> where first_name like 's%' ;
+------------+-----------+----------+--------+
| first_name | last_name | job_id   | salary |
+------------+-----------+----------+--------+
| Steven     | King      | AD_PRES  |  24000 |
| Shelli     | Baida     | PU_CLERK |   2900 |
| Sigal      | Tobias    | PU_CLERK |   2800 |
| Shanta     | Vollman   | ST_MAN   |   6500 |
| Steven     | Markle    | ST_CLERK |   2200 |
+------------+-----------+----------+--------+
5 rows in set (0.06 sec)
```

2. Write a query to select employee with the highest salary (using an inner query).

```
mysql> select * from employees
    -> where salary = (select MAX(salary) from employees);
+-------------+------------+-----------+-------+--------------+------------+--------+--------+----------------+------------+---------------+
| Employee_id | first_name | last_name | email | phone_number | hire_date  | job_id | salary | commission_pct | manager_id | department_id |
+-------------+------------+-----------+-------+--------------+------------+--------+--------+----------------+------------+---------------+
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES |  24000 |           NULL |       NULL |            20 |
+-------------+------------+-----------+-------+--------------+------------+--------+--------+----------------+------------+---------------+
1 row in set (0.08 sec)
```

3. Select employee with the second highest salary.

```
mysql> select * from employees
    -> where salary = (select MAX(salary) from employees
    -> where salary < (select MAX(salary) from employees));
+-------------+------------+-----------+----------+--------------+------------+--------+--------+----------------+------------+---------------+
| Employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id | salary | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+--------+--------+----------------+------------+---------------+
|         101 | Neena      | Kochhar   | NKOCHHAR | 515.123.4568 | 1989-11-21 | AD_VP  |  17000 |           NULL |        100 |            20 |
|         102 | Lex        | De Haan   | LDEHAAN  | 515.123.4569 | 1993-09-12 | AD_VP  |  17000 |           NULL |        100 |            30 |
+-------------+------------+-----------+----------+--------------+------------+--------+--------+----------------+------------+---------------+
2 rows in set (0.11 sec)
```

4. Write a query to select employees and their corresponding managers and their salaries.

```
    -> concat(m.first_name,' ',m.last_name)as manager_name,m.salary manager_salary
    -> from employees e
    -> left join employees m on m.Employee_id=e.manager_id;
```

employee_name	employee_salary	manager_name	manager_salary
Steven King	24000	NULL	NULL
Neena Kochhar	17000	Steven King	24000
Lex De Haan	17000	Steven King	24000
Aelxander Hunold	9000	Lex De Haan	17000
Bruce Ernst	6000	Aelxander Hunold	9000
David Austin	4800	Aelxander Hunold	9000
Valli Pataballa	4800	Aelxander Hunold	9000
Diana Lorentz	4200	Aelxander Hunold	9000
Nancy Greenberg	12000	Neena Kochhar	17000
Daniel Faviet	9000	Nancy Greenberg	12000
John Chen	8200	Nancy Greenberg	12000
Ismael Sciarra	7700	Nancy Greenberg	12000
Jose Manuel Urman	7800	Nancy Greenberg	12000
Luis Popp	6900	Nancy Greenberg	12000
Den Raphaely	11000	Steven King	24000
Alexander Khoo	3100	Den Raphaely	11000
Shelli Baida	2900	Den Raphaely	11000
Sigal Tobias	2800	Den Raphaely	11000
Guy Himuro	2600	Den Raphaely	11000
Karen Colmenares	2500	Den Raphaely	11000
Matthew Weiss	8000	Steven King	24000
Adam Fripp	8200	Steven King	24000
Payam Kaufling	7900	Steven King	24000
Shanta Vollman	6500	Steven King	24000
Kevin Mourgos	5800	Steven King	24000
Julia Nayer	3200	Matthew Weiss	8000
Irene Mikkilineni	2700	Matthew Weiss	8000
James Landry	2400	Matthew Weiss	8000
Steven Markle	2200	Matthew Weiss	8000
Laura Bissot	3300	Adam Fripp	8200
Mozhe Atkinson	2800	Adam Fripp	8200

```
31 rows in set (0.04 sec)
```

5. Write a query to select employees and their corresponding managers and their salaries (SELF Join).

```
mysql> select concat(e.first_name,' ',e.last_name)as employee_name, e.salary employee_salary,
    -> concat(m.first_name,' ',m.last_name)as manager_name,m.salary manager_salary
    -> from employees e
    -> inner join employees m on m.Employee_id=e.manager_id;
+-------------------+-----------------+------------------+----------------+
| employee_name     | employee_salary | manager_name     | manager_salary |
+-------------------+-----------------+------------------+----------------+
| Neena Kochhar     |           17000 | Steven King      |          24000 |
| Lex De Haan       |           17000 | Steven King      |          24000 |
| Aelxander Hunold  |            9000 | Lex De Haan      |          17000 |
| Bruce Ernst       |            6000 | Aelxander Hunold |           9000 |
| David Austin      |            4800 | Aelxander Hunold |           9000 |
| Valli Pataballa   |            4800 | Aelxander Hunold |           9000 |
| Diana Lorentz     |            4200 | Aelxander Hunold |           9000 |
| Nancy Greenberg   |           12000 | Neena Kochhar    |          17000 |
| Daniel Faviet     |            9000 | Nancy Greenberg  |          12000 |
| John Chen         |            8200 | Nancy Greenberg  |          12000 |
| Ismael Sciarra    |            7700 | Nancy Greenberg  |          12000 |
| Jose Manuel Urman |            7800 | Nancy Greenberg  |          12000 |
| Luis Popp         |            6900 | Nancy Greenberg  |          12000 |
| Den Raphaely      |           11000 | Steven King      |          24000 |
| Alexander Khoo    |            3100 | Den Raphaely     |          11000 |
| Shelli Baida      |            2900 | Den Raphaely     |          11000 |
| Sigal Tobias      |            2800 | Den Raphaely     |          11000 |
| Guy Himuro        |            2600 | Den Raphaely     |          11000 |
| Karen Colmenares  |            2500 | Den Raphaely     |          11000 |
| Matthew Weiss     |            8000 | Steven King      |          24000 |
| Adam Fripp        |            8200 | Steven King      |          24000 |
| Payam Kaufling    |            7900 | Steven King      |          24000 |
| Shanta Vollman    |            6500 | Steven King      |          24000 |
| Kevin Mourgos     |            5800 | Steven King      |          24000 |
| Julia Nayer       |            3200 | Matthew Weiss    |           8000 |
| Irene Mikkilineni |            2700 | Matthew Weiss    |           8000 |
| James Landry      |            2400 | Matthew Weiss    |           8000 |
| Steven Markle     |            2200 | Matthew Weiss    |           8000 |
| Laura Bissot      |            3300 | Adam Fripp       |           8200 |
| Mozhe Atkinson    |            2800 | Adam Fripp       |           8200 |
+-------------------+-----------------+------------------+----------------+
30 rows in set (0.04 sec)
```

6. Create a view for the above query.

```
mysql> create view emp_manager_dtls as
    -> select concat(e.first_name,' ',e.last_name)as employee_name, e.salary employee_salary,
    -> concat(m.first_name,' ',m.last_name)as manager_name,m.salary manager_salary
    -> from employees e
    -> inner join employees m on m.Employee_id=e.manager_id;
Query OK, 0 rows affected (0.15 sec)

mysql> show full tables;
+----------------------------+------------+
| Tables_in_entri_assignment | Table_type |
+----------------------------+------------+
| departments                | BASE TABLE |
| emp_manager_dtls           | VIEW       |
| employees                  | BASE TABLE |
+----------------------------+------------+
3 rows in set (0.09 sec)
```

7. Write a query to show the count of employees under each manager in descending order (from view).

```
mysql> select manager_name,count(employee_name) employee_count
    -> from emp_manager_dtls
    -> group by manager_name
    -> order by employee_count desc;
+------------------+----------------+
| manager_name     | employee_count |
+------------------+----------------+
| Steven King      |              8 |
| Nancy Greenberg  |              5 |
| Den Raphaely     |              5 |
| Aelxander Hunold |              4 |
| Matthew Weiss    |              4 |
| Adam Fripp       |              2 |
| Lex De Haan      |              1 |
| Neena Kochhar    |              1 |
+------------------+----------------+
8 rows in set (0.06 sec)
```

8. Find the count of employees in each department.

```
mysql> select Department_name,count(Employee_id) employee_count
    -> from departments d
    -> left join employees e on d.Department_id = e.Department_id
    -> group by d.Department_id,Department_name
    -> order by employee_count desc;
+----------------------+----------------+
| Department_name      | employee_count |
+----------------------+----------------+
| Shipping             |              7 |
| IT                   |              4 |
| Purchasing           |              3 |
| Human Resources      |              3 |
| Marketing            |              2 |
| Sales                |              2 |
| Payroll              |              2 |
| Public Relations     |              1 |
| Executive            |              1 |
| Finance              |              1 |
| Accounting           |              1 |
| Corporate Tax        |              1 |
| Control and Credit   |              1 |
| Shareholder Services |              1 |
| Benefits             |              1 |
| Treasury             |              0 |
+----------------------+----------------+
16 rows in set (0.29 sec)
```

9. Get the count of employees hired year wise .

```
mysql> select year(hire_date)year ,count(Employee_id) employee_count
    -> from employees
    -> group by year(hire_date)
    -> order by year(hire_date);
+------+----------------+
| year | employee_count |
+------+----------------+
| 1987 |              1 |
| 1989 |              1 |
| 1990 |              1 |
| 1991 |              1 |
| 1993 |              1 |
| 1994 |              3 |
| 1995 |              2 |
| 1996 |              1 |
| 1997 |             10 |
| 1998 |              4 |
| 1999 |              5 |
| 2000 |              1 |
+------+----------------+
12 rows in set (0.04 sec)
```

10 . create a stored procedure to get the " Get the count of employees hired in the input year"(IN year , OUT count).

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE GetEmployeeCountByYear
    -> (
    ->     IN inputYear INT,
    ->     OUT employeeCount INT
    -> )
    -> BEGIN
    ->     SELECT COUNT(*) INTO employeeCount
    ->     FROM employees
    ->     WHERE YEAR(hire_date)= inputYear;
    -> END //
Query OK, 0 rows affected (0.24 sec)

mysql> DELIMITER ;
```

11.Select the employees whose first_name contains "an" .

```
mysql> select first_name from employees
    -> where first_name like '%an%' ;
+-------------+
| first_name  |
+-------------+
| Aelxander   |
| Diana       |
| Nancy       |
| Daniel      |
| Jose Manuel |
| Alexander   |
| Shanta      |
+-------------+
7 rows in set (0.00 sec)
```

12. Select employee first name and the corresponding phone number in the format (_ _ _)-(_ _ _)-(_ _ _ _).

```
mysql> select first_name, concat
    -> ('(',SUBSTRING(phone_number,1,3),')-(',SUBSTRING(phone_number,5,3),')-(',SUBSTRING(phone_number,9,4),')')
    -> as Formatted_PhoneNumber
    -> from employees;
+-------------+----------------------+
| first_name  | Formatted_PhoneNumber |
+-------------+----------------------+
| Steven      | (515)-(123)-(4567)   |
| Neena       | (515)-(123)-(4568)   |
| Lex         | (515)-(123)-(4569)   |
| Aelxander   | (590)-(423)-(4567)   |
| Bruce       | (590)-(423)-(4568)   |
| David       | (590)-(423)-(4569)   |
| Valli       | (590)-(423)-(4560)   |
| Diana       | (590)-(423)-(5567)   |
| Nancy       | (515)-(124)-(4569)   |
| Daniel      | (515)-(124)-(4169)   |
| John        | (515)-(124)-(4269)   |
| Ismael      | (515)-(124)-(4369)   |
| Jose Manuel | (515)-(124)-(4469)   |
| Luis        | (515)-(124)-(4567)   |
| Den         | (515)-(127)-(4561)   |
| Alexander   | (515)-(127)-(4562)   |
| Shelli      | (515)-(127)-(4563)   |
| Sigal       | (515)-(127)-(4564)   |
| Guy         | (515)-(127)-(4565)   |
| Karen       | (515)-(127)-(4566)   |
| Matthew     | (650)-(123)-(1234)   |
| Adam        | (650)-(123)-(2234)   |
| Payam       | (650)-(123)-(3234)   |
| Shanta      | (650)-(123)-(4234)   |
| Kevin       | (650)-(123)-(5234)   |
| Julia       | (650)-(124)-(1214)   |
| Irene       | (650)-(124)-(1224)   |
| James       | (650)-(124)-(1334)   |
| Steven      | (650)-(124)-(1434)   |
| Laura       | (650)-(124)-(5234)   |
| Mozhe       | (650)-(124)-(6234)   |
+-------------+----------------------+
31 rows in set (0.04 sec)
```

13. Find the employees who joined in August, 1994.

```
mysql> select concat(first_name,' ',last_name)as employee_name ,year(hire_date) hire_date
    -> from employees
    -> where year(hire_date)=1994 and month(hire_date)='08' ;
+-----------------+-----------+
| employee_name   | hire_date |
+-----------------+-----------+
| Nancy Greenberg |      1994 |
| Daniel Faviet   |      1994 |
+-----------------+-----------+
2 rows in set (0.00 sec)
```

14. Find the maximum salary from each department.

```
mysql> select Department_name, max(salary) max_salary
    -> from departments d
    -> inner join employees e on d.Department_id = e.Department_id
    -> group by Department_name;
+----------------------+------------+
| Department_name      | max_salary |
+----------------------+------------+
| Marketing            |      24000 |
| Purchasing           |      17000 |
| IT                   |       9000 |
| Human Resources      |       7900 |
| Finance              |      12000 |
| Payroll              |       9000 |
| Benefits             |       7700 |
| Shareholder Services |       7800 |
| Control and Credit   |       6900 |
| Sales                |       5800 |
| Public Relations     |       2900 |
| Corporate Tax        |       2500 |
| Shipping             |       8200 |
| Executive            |       2400 |
| Accounting           |       2800 |
+----------------------+------------+
15 rows in set (0.03 sec)
```

15.Write a SQL query to display the 5 least earning employees.

```
mysql> select * from employees
    -> order by salary asc
    -> limit 5;
+-------------+------------+-------------+----------+--------------+------------+----------+--------+----------------+------------+---------------+
| Employee_id | first_name | last_name   | email    | phone_number | hire_date  | job_id   | salary | commission_pct | manager_id | department_id |
+-------------+------------+-------------+----------+--------------+------------+----------+--------+----------------+------------+---------------+
|         128 | Steven     | Markle      | SMARKLE  | 650.124.1434 | 2000-03-04 | ST_CLERK |   2200 |           NULL |        120 |            50 |
|         127 | James      | Landry      | JLANDRY  | 650.124.1334 | 1999-01-02 | ST_CLERK |   2400 |           NULL |        120 |            90 |
|         119 | Karen      | Colmenares  | KCOLMENA | 515.127.4566 | 1999-04-08 | PU_CLERK |   2500 |           NULL |        114 |           130 |
|         118 | Guy        | Himuro      | GHIMURO  | 515.127.4565 | 1998-01-02 | PU_CLERK |   2600 |           NULL |        114 |            60 |
|         126 | Irene      | Mikkilineni | IMIKKILI | 650.124.1224 | 1998-11-12 | ST_CLERK |   2700 |           NULL |        120 |            50 |
+-------------+------------+-------------+----------+--------------+------------+----------+--------+----------------+------------+---------------+
5 rows in set (0.00 sec)
```

16. Find the employees hired in the 80s.

```
mysql> select concat(first_name,' ',last_name)as employee_name ,year(hire_date) hire_year
    -> from employees
    -> where year(hire_date) between 1980 and 1989 ;
+---------------+-----------+
| employee_name | hire_year |
+---------------+-----------+
| Steven King   |      1987 |
| Neena Kochhar |      1989 |
+---------------+-----------+
2 rows in set (0.00 sec)
```

17. Find the employees who joined the company after 15th of the month.

```
mysql> select * from employees
    -> where DAY(hire_date)>15 ;
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
| Employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id   | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
|         100 | Steven     | King      | SKING    | 515.123.4567 | 1987-06-17 | AD_PRES  | 24000    |           NULL |       NULL |            20 |
|         101 | Neena      | Kochhar   | NKOCHHAR | 515.123.4568 | 1989-11-21 | AD_VP    | 17000    |           NULL |        100 |            20 |
|         103 | Aelxander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-09-30 | IT_PROG  |  9000    |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG  |  6000    |           NULL |        103 |            60 |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG  |  4800    |           NULL |        103 |            60 |
|         108 | Nancy      | Greenberg | NGREENBE | 515.124.4569 | 1994-08-17 | FI_MGR   | 12000    |           NULL |        101 |           100 |
|         120 | Matthew    | Weiss     | MWEISS   | 650.123.1234 | 1996-07-18 | ST_MAN   |  8000    |           NULL |        100 |            50 |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
7 rows in set (0.03 sec)
```