1. Write a query to find the name (first_name, last_name) and the salary of the employees who have a higher salary than the employee whose last_name='Bull'.

Query :

select first_name,last_name,salary from employees where salary > (select salary from employees where last_name='BULL');

Output :

```
mysql> select first_name,last_name,salary from employees where salary > (select salary from employees where last_name='BULL');
+------------+-----------+--------+
| first_name | last_name | salary |
+------------+-----------+--------+
| Steven     | King      |  24000 |
| Neena      | Kochhar   |  17000 |
| Lex        | De Haan   |  17000 |
+------------+-----------+--------+
3 rows in set (0.00 sec)
```

2. Write a query to find the name (first_name, last_name) of all employees who works in the IT department.

Query :

SELECT first_name, last_name FROM employees WHERE department_id IN (SELECT department_id FROM department WHERE department_name='IT');

Output :

```
mysql> SELECT first_name, last_name FROM employees WHERE department_id IN (SELECT department_id FROM department WHERE department_name='IT');
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Alexander  | Hunold    |
| Bruce      | Ernst     |
| David      | Austin    |
+------------+-----------+
3 rows in set (0.00 sec)
```

3. Write a query to find the name (first_name, last_name) of the employees who have a manager and worked in a USA based department.

Please refer employees and department table and locations

Query :

SELECT first_name, last_name FROM employees

WHERE manager_id in (select employee_id

FROM employees WHERE department_id

IN (SELECT department_id FROM department WHERE location_id

IN (select location_id from location where country_id='US')));

Output :

```
mysql> SELECT first_name, last_name FROM employees
    -> WHERE manager_id in (select employee_id
    -> FROM employees WHERE department_id
    -> IN (SELECT department_id FROM department WHERE location_id
    -> IN (select location_id from location where country_id='US')));
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Bruce      | Ernst     |
| David      | Austin    |
+------------+-----------+
2 rows in set (0.00 sec)
```

Department table

```
DEPARTMENT_ID | DEPARTMENT_NAME      | MANAGER_ID | LOCATION_ID |
+--------------+--------------------+------------+-------------+
|           10 | Administration     |        200 |        1700 |
|           20 | Marketing          |        201 |        1800 |
|           30 | Purchasing         |        114 |        1700 |
|           40 | Human Resources    |        203 |        2400 |
|           50 | Shipping           |        121 |        1500 |
|           60 | IT                 |        103 |        1400 |
|           70 | Public Relations   |        204 |        2700 |
|           80 | Sales              |        145 |        2500 |
|           90 | Executive          |        100 |        1700 |
|          100 | Finance            |        108 |        1700 |
|          110 | Accounting         |        205 |        1700 |
|          120 | Treasury           |          0 |        1700 |
|          130 | Corporate Tax      |          0 |        1700 |
|          140 | Control And Credit |          0 |        1700 |
|          150 | Shareholder Services |        0 |        1700 |
|          160 | Benefits           |          0 |        1700 |
|          170 | Manufacturing      |          0 |        1700 |
|          180 | Construction       |          0 |        1700 |
|          190 | Contracting        |          0 |        1700 |
|          200 | Operations         |          0 |        1700 |
|          210 | IT Support         |          0 |        1700 |
|          220 | NOC                |          0 |        1700 |
|          230 | IT Helpdesk        |          0 |        1700 |
|          240 | Government Sales   |          0 |        1700 |
|          250 | Retail Sales       |          0 |        1700 |
|          260 | Recruiting         |          0 |        1700 |
|          270 | Payroll            |          0 |        1700 |
+--------------+--------------------+------------+-------------+
```

Table locations

```
location_id   street_address       postal_code   city        state_province
country_id
-----------   ------------------   -----------   ----------   --------------
----------
1000          1297 Via Cola di Rie  989           Roma
IT
1100          93091 Calle della Te  10934         Venice
IT
1200          2017 Shinjuku-ku      1689          Tokyo        Tokyo Prefectu
JP
1300          9450 Kamiya-cho       6823          Hiroshima
JP
```

1400 US	2014 Jabberwocky Rd	26192	Southlake	Texas
1500 US	2011 Interiors Blvd	99236	South San	California
1600 US	2007 Zagora St	50090	South Brun	New Jersey
1700 US	2004 Charade Rd	98199	Seattle	Washington
1800 CA	147 Spadina Ave	M5V 2L7	Toronto	Ontario
1900 CA	6092 Boxwood St	YSW 9T2	Whitehorse	Yukon
2000 CN	40-5-12 Laogianggen	190518	Beijing	
2100 IN	1298 Vileparle (E)	490231	Bombay	Maharashtra
2200 AU	12-98 Victoria Stree	2901	Sydney	New South Wale
2300 SG	198 Clementi North	540198	Singapore	
2400 UK	8204 Arthur St		London	
2500 UK	Magdalen Centre, The	OX9 9ZB	Oxford	Oxford
2600 UK	9702 Chester Road	9629850293	Stretford	Manchester
2700 DE	Schwanthalerstr. 703	80925	Munich	Bavaria
2800 BR	Rua Frei Caneca 1360	01307-002	Sao Paulo	Sao Paulo
2900 CH	20 Rue des Corps-Sai	1730	Geneva	Geneve
3000 CH	Murtenstrasse 921	3095	Bern	BE
3100 NL	Pieter Breughelstraa	3029SK	Utrecht	Utrecht
3200 MX	Mariano Escobedo 999	11932	Mexico Cit	Distrito Feder

4. Write a query to find the name (first_name, last_name) of the employees who are managers

Query :

SELECT first_name, last_name from employees where employee_id in(select manager_id from employees);

Output :

```
mysql> SELECT first_name, last_name from employees where employee_id in(select manager_id from employees);
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Steven     | King      |
| Lex        | De Haan   |
| Alexander  | Hunold    |
+------------+-----------+
3 rows in set (0.00 sec)
```

5. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary.

Query :

SELECT first_name, last_name,salary from employees where salary >(select avg(salary) from employees);

Output :

```
mysql> SELECT first_name, last_name,salary from employees where salary >(select avg(salary) from employees);
+------------+------------+--------+
| first_name | last_name  | salary |
+------------+------------+--------+
| Steven     | King       |  24000 |
| Neena      | Kochhar    |  17000 |
| Lex        | De Haan    |  17000 |
| Alexander  | Hunold     |   9000 |
| Matthew    | Weiss      |   8000 |
| Adam       | Fripp      |   8200 |
| Payam      | Kaufling   |   7900 |
| Alex       | khoo       |   8000 |
| Jhonson    | lee        |   9000 |
| Kean       | Scout      |   9000 |
| Kean       | BULL       |   9000 |
+------------+------------+--------+
11 rows in set (0.01 sec)
```

6. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is equal to the minimum salary for their job grade

Query :

SELECT first_name, last_name FROM employees

WHERE salary = (SELECT MIN(salary) FROM employees);

Output :

```
mysql> SELECT  first_name, last_name FROM employees
    -> WHERE salary = (SELECT MIN(salary) FROM employees);
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Peter      | Vargas    |
| Peter A    | Vargas    |
| Peter B    | Vargas    |
| Peter C    | Vargas    |
| Peter D    | Vargas    |
| Peter E    | Vargas    |
| Peter F    | Vargas    |
| Peter ◆    | Vargas    |
+------------+-----------+
8 rows in set (0.00 sec)
```

7. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the average salary and works in any of the IT departments.

Query :

select first_name,last_name,salary from employees where department_id in (select department_id from department where department_name like 'IT_PROG%' and salary>(select avg(salary) from employees));

Output :

```
mysql> select first_name,last_name,salary from employees
    where department_id in (select department_id from department
    where department_name like 'IT_PROG%' and salary>(select avg(salary) from employees));
Empty set (0.00 sec)
```

8. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the earning of Luis.

Query:

SELECT first_name, last_name,salary from employees where salary>(select salary from employees where first_name='Luis');

Output :

```
mysql>  SELECT first_name, last_name,salary from employees
    where salary>(select salary from employees where first_name='Luis');
Empty set (0.00 sec)
```

9. Write a query to find the name (first_name, last_name), and salary of the employees who earn the same salary as the minimum salary for all departments.

Query:

SELECT first_name, last_name,salary from employees where salary=(select min(salary) from employees);

Output :

```
mysql> SELECT first_name, last_name,salary from employees where salary=(select min(salary) from employees);
+------------+-----------+--------+
| first_name | last_name | salary |
+------------+-----------+--------+
| Peter      | Vargas    |   2500 |
| Peter A    | Vargas    |   2500 |
| Peter B    | Vargas    |   2500 |
| Peter C    | Vargas    |   2500 |
| Peter D    | Vargas    |   2500 |
| Peter E    | Vargas    |   2500 |
| Peter F    | Vargas    |   2500 |
| Peter ◆    | Vargas    |   2500 |
+------------+-----------+--------+
8 rows in set (0.00 sec)
```

10. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary of all departments.
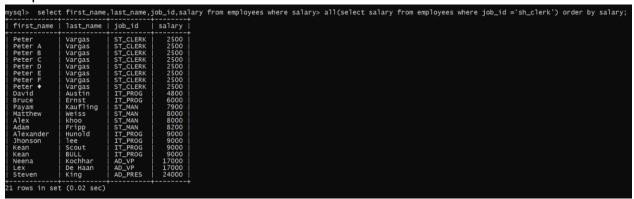
Query:

SELECT first_name, last_name, salary from employees where salary>(select avg(salary) from employees);

Output :

```
mysql> SELECT first_name, last_name, salary from employees where salary>(select avg(salary) from employees);
+------------+-----------+--------+
| first_name | last_name | salary |
+------------+-----------+--------+
| Steven     | King      |  24000 |
| Neena      | Kochhar   |  17000 |
| Lex        | De Haan   |  17000 |
| Alexander  | Hunold    |   9000 |
| Matthew    | Weiss     |   8000 |
| Adam       | Fripp     |   8200 |
| Payam      | Kaufling  |   7900 |
| Alex       | khoo      |   8000 |
| Jhonson    | lee       |   9000 |
| Kean       | Scout     |   9000 |
| Kean       | BULL      |   9000 |
+------------+-----------+--------+
11 rows in set (0.00 sec)
```

11. Write a query to find the name (first_name, last_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = 'SH_CLERK'). Sort the results of the salary of the lowest to highest.

Query :
 select first_name,last_name,job_id,salary from employees where salary> all(select salary from employees where job_id ='sh_clerk') order by salary;

Output :

```
mysql>  select first_name,last_name,job_id,salary from employees where salary> all(select salary from employees where job_id ='sh_clerk') order by salary;
+------------+-----------+---------+--------+
| first_name | last_name | job_id  | salary |
+------------+-----------+---------+--------+
| Peter      | Vargas    | ST_CLERK|   2500 |
| Peter A    | Vargas    | ST_CLERK|   2500 |
| Peter B    | Vargas    | ST_CLERK|   2500 |
| Peter C    | Vargas    | ST_CLERK|   2500 |
| Peter D    | Vargas    | ST_CLERK|   2500 |
| Peter E    | Vargas    | ST_CLERK|   2500 |
| Peter F    | Vargas    | ST_CLERK|   2500 |
| Peter ♦    | Vargas    | ST_CLERK|   2500 |
| David      | Austin    | IT_PROG |   4800 |
| Bruce      | Ernst     | IT_PROG |   6000 |
| Payam      | Kaufling  | ST_MAN  |   7900 |
| Matthew    | Weiss     | ST_MAN  |   8000 |
| Alex       | khoo      | ST_MAN  |   8000 |
| Adam       | Fripp     | ST_MAN  |   8200 |
| Alexander  | Hunold    | IT_PROG |   9000 |
| Jhonson    | lee       | IT_PROG |   9000 |
| Kean       | Scout     | IT_PROG |   9000 |
| Kean       | BULL      | IT_PROG |   9000 |
| Neena      | Kochhar   | AD_VP   |  17000 |
| Lex        | De Haan   | AD_VP   |  17000 |
| Steven     | King      | AD_PRES |  24000 |
+------------+-----------+---------+--------+
21 rows in set (0.02 sec)
```

12. Write a query to find the name (first_name, last_name) of the employees who are not supervisors

Query :
select b.first_name,b.last_name from employees b where not exists(select 'x' from employees a where a.manager_id=b.manager_id);

Output :

```
mysql> select b.first_name,b.last_name from employees b
    where not exists(select 'x' from employees a where a.manager_id=b.manager_id);
Empty set (0.00 sec)
```

14. Write a query to display the employee ID, first name, last name, salary of all employees whose salary is above average for their departments

Query :
 select employee_id,first_name,last_name,salary from employees a where salary>(select avg(salary) from employees where department_id=a.department_id);

Output :

```
mysql>  select employee_id,first_name,last_name,salary from employees a
    where salary>(select avg(salary) from employees where department_id=a.department_id);
+-------------+------------+-----------+--------+
| employee_id | first_name | last_name | salary |
+-------------+------------+-----------+--------+
|         100 | Steven     | King      |  24000 |
|         103 | Alexander  | Hunold    |   9000 |
|         120 | Matthew    | Weiss     |   8000 |
|         121 | Adam       | Fripp     |   8200 |
|         122 | Payam      | Kaufling  |   7900 |
+-------------+------------+-----------+--------+
5 rows in set (0.00 sec)
```
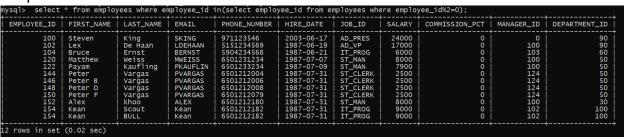
15. Write a query to fetch even numbered records from employees table.
Query :
mysql> select * from employees where employee_id in(select employee_id from employees where employee_id%2=0);
Output :

```
mysql>  select * from employees where employee_id in(select employee_id from employees where employee_id%2=0);
+-------------+------------+-----------+----------+--------------+------------+----------+--------+----------------+------------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL    | PHONE_NUMBER | HIRE_DATE  | JOB_ID   | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
+-------------+------------+-----------+----------+--------------+------------+----------+--------+----------------+------------+---------------+
|         100 | Steven     | King      | SKING    | 971123546    | 2003-06-17 | AD_PRES  |  24000 |              0 |          0 |            90 |
|         102 | Lex        | De Haan   | LDEHAAN  | 5151234569   | 1987-06-19 | AD_VP    |  17000 |              0 |        100 |            90 |
|         104 | Bruce      | Ernst     | BERNST   | 5904234568   | 1987-06-21 | IT_PROG  |   6000 |              0 |        103 |            60 |
|         120 | Matthew    | Weiss     | MWEISS   | 6501231234   | 1987-07-07 | ST_MAN   |   8000 |              0 |        100 |            50 |
|         122 | Payam      | Kaufling  | PKAUFLIN | 6501233234   | 1987-07-09 | ST_MAN   |   7900 |              0 |        100 |            50 |
|         144 | Peter      | Vargas    | PVARGAS  | 6501212004   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         146 | Peter B    | Vargas    | PVARGAS  | 6501212006   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         148 | Peter D    | Vargas    | PVARGAS  | 6501212008   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         150 | Peter F    | Vargas    | PVARGAS  | 6501212079   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         152 | Alex       | khoo      | ALEX     | 6501212180   | 1987-07-31 | ST_MAN   |   8000 |              0 |        100 |            30 |
|         154 | Kean       | Scout     | Kean     | 6501212182   | 1987-07-31 | IT_PROG  |   9000 |              0 |        102 |           100 |
|         154 | Kean       | BULL      | Kean     | 6501212182   | 1987-07-31 | IT_PROG  |   9000 |              0 |        102 |           100 |
+-------------+------------+-----------+----------+--------------+------------+----------+--------+----------------+------------+---------------+
12 rows in set (0.02 sec)
```

16. Write a query to find the 5th maximum salary in the employees table

Query :
 select distinct salary from employees a where 5=(select count(distinct salary) from employees b where b.salary>=a.salary);

Output :

```
mysql> select distinct salary from employees a
    where 5=(select count(distinct salary) from employees b where b.salary>=a.salary);
+--------+
| salary |
+--------+
|   8000 |
+--------+
1 row in set (0.00 sec)
```

17. Write a query to find the 4th minimum salary in the employees table

Query :
select distinct salary from employees a where 4=(select count(distinct salary) from employees b where b.salary<=a.salary);
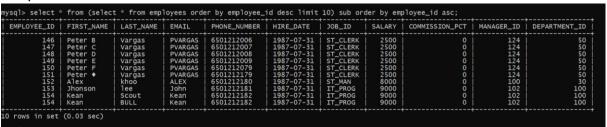
Output :

```
mysql> select distinct salary from employees a
    where 4=(select count(distinct salary) from employees b where b.salary<=a.salary);
+--------+
| salary |
+--------+
|   7900 |
+--------+
1 row in set (0.00 sec)
```

18. Write a query to select last 10 records from a table

Query :
 select * from (select * from employees order by employee_id desc limit 10) sub order by employee_id asc;

Output :

```
mysql> select * from (select * from employees order by employee_id desc limit 10) sub order by employee_id asc;
+-------------+------------+-----------+---------+--------------+------------+----------+--------+----------------+------------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL   | PHONE_NUMBER | HIRE_DATE  | JOB_ID   | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
+-------------+------------+-----------+---------+--------------+------------+----------+--------+----------------+------------+---------------+
|         146 | Peter B    | Vargas    | PVARGAS | 6501212006   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         147 | Peter C    | Vargas    | PVARGAS | 6501212007   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         148 | Peter D    | Vargas    | PVARGAS | 6501212008   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         149 | Peter E    | Vargas    | PVARGAS | 6501212009   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         150 | Peter F    | Vargas    | PVARGAS | 6501212079   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         151 | Peter ◆    | Vargas    | PVARGAS | 6501212179   | 1987-07-31 | ST_CLERK |   2500 |              0 |        124 |            50 |
|         152 | Alex       | khoo      | ALEX    | 6501212180   | 1987-07-31 | ST_MAN   |   8000 |              0 |        100 |            30 |
|         153 | Jhonson    | lee       | John    | 6501212181   | 1987-07-31 | IT_PROG  |   9000 |              0 |        102 |           100 |
|         154 | Kean       | Scout     | Kean    | 6501212182   | 1987-07-31 | IT_PROG  |   9000 |              0 |        102 |           100 |
|         154 | Kean       | BULL      | Kean    | 6501212182   | 1987-07-31 | IT_PROG  |   9000 |              0 |        102 |           100 |
+-------------+------------+-----------+---------+--------------+------------+----------+--------+----------------+------------+---------------+
10 rows in set (0.03 sec)
```

19. Write a query to list the department ID and name of all the departments where no employee is working.

Query :
   mysql> select * from department where department_id not in (select department_id from employees);

Output :

```
mysql>  select * from department where department_id not in (select department_id from employees);
+---------------+-------------------+------------+-------------+
| department_id | department_name   | manager_id | location_id |
+---------------+-------------------+------------+-------------+
|            10 | Administration    |        200 |        1700 |
|            20 | Marketing         |        201 |        1800 |
|            40 |  Human Resources  |        203 |        2400 |
+---------------+-------------------+------------+-------------+
3 rows in set (0.00 sec)
```

20. Write a query to get 3 maximum salaries

Query :

   SELECT DISTINCT salary FROM employees a WHERE 3 >= (SELECT COUNT(DISTINCT salary) FROM employees b WHERE b.salary >= a.salary) ORDER BY a.salary DESC;

Output :

```
mysql> SELECT DISTINCT salary FROM employees a
       WHERE 3 >= (SELECT COUNT(DISTINCT salary) FROM employees b
       WHERE b.salary >= a.salary) ORDER BY a.salary DESC;
+--------+
| salary |
+--------+
|  24000 |
|  17000 |
|   9000 |
+--------+
3 rows in set (0.00 sec)
```

21. Write a query to get 3 minimum salaries.

Query :
SELECT DISTINCT salary FROM employees a WHERE 3 >= (SELECT
COUNT(DISTINCT salary) FROM employees b WHERE b.salary <= a.salary)
ORDER BY a.salary DESC;

Output:

```
mysql> SELECT DISTINCT salary FROM employees a
    WHERE 3 >= (SELECT COUNT(DISTINCT salary) FROM employees b
    WHERE b.salary <= a.salary) ORDER BY a.salary DESC;
+--------+
| salary |
+--------+
|   6000 |
|   4800 |
|   2500 |
+--------+
3 rows in set (0.02 sec)
```

22. Write a query to get nth max salaries of employees

Query :
 select * from employees emp1 where(1)=(select count(distinct(emp2.salary))
from employees emp2 where emp2.salary>emp1.salary);

Output :

```
mysql>  select * from employees emp1 where(1)=(select count(distinct(emp2.salary)) from employees emp2 where emp2.salary>emp1.salary);
+-------------+------------+-----------+---------+--------------+------------+-------+--------+----------------+------------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL   | PHONE_NUMBER | HIRE_DATE  | JOB_ID| SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
+-------------+------------+-----------+---------+--------------+------------+-------+--------+----------------+------------+---------------+
|         101 | Neena      | Kochhar   | NKOCHHAR| 515123568    | 1987-06-18 | AD_VP | 17000  |              0 |        100 |            90 |
|         102 | Lex        | De Haan   | LDEHAAN | 5151234569   | 1987-06-19 | AD_VP | 17000  |              0 |        100 |            90 |
+-------------+------------+-----------+---------+--------------+------------+-------+--------+----------------+------------+---------------+
2 rows in set (0.02 sec)
```