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='Chen'.

Query:

```
mysql> select first_name,last_name,salary from employees where salary > (select salary from employees where last_name='chen');
+------------+-------------+--------+
| first_name | last_name   | salary |
+------------+-------------+--------+
| Steven     | King        |  24000 |
|  Neena     |  Kochhar    |  17000 |
| Lex        | De Haan     |  17000 |
| Nancy      | Greenberg   |  12000 |
| Daniel     | Faviet      |   9000 |
| Den        | Raphaely    |  11000 |
|  Alexander |  Hunold     |   9000 |
+------------+-------------+--------+
7 rows in set (0.01 sec)
```

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

```
mysql> SELECT first_name, last_name FROM employees WHERE department_id IN (SELECT department_id FROM department WHERE department_name='IT');
Empty 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.

```
mysql> create table department(DEPARTMENT_ID int, DEPARTMENT_NAME varchar(40), MANAGER_ID int, LOCATION_ID int);
```

```
Query OK, 0 rows affected (0.03 sec)

mysql> insert into department values(10 ,' Administration ' , 200 , 1700
);
Query OK, 1 row affected (0.02 sec)

mysql> insert into department values(20 ,' Marketing ' , 201 , 1800 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(30 ,' Purchasing ' , 114 , 1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(40 ,' Human Resources ' , 203 ,
2400 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(50 ,' Shipping ' , 121 , 1500 );
Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(60 ,' IT ' , 103 , 1400 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(70 ,' Public Relations ' , 204 , 2700
);
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(80 ,' Sales ' , 145 , 2500 );
```

```
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(90 ,'Executive ' , 100 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(100 ,'Finance ' , 108 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(110 ,'Accounting ' , 205 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(120 ,'Treasury ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(130 ,'Corporate Tax ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(140 ,'Control and credit ' , 0 ,1700
);
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(150 ,'Shareholder services ' , 0
,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(160 ,'Benefits ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into department values(170 ,'manufacturing ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(180 ,'construction ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(190 ,'contracting ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(200,'operations ' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(210,'IT support' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(220,'NOC' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(230,'IT helpdesk' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(240,'Government sales' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(250,'Retails sales' , 0 ,1700 );
```

```
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(260,'Recruiting' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> insert into department values(270,'Payroll' , 0 ,1700 );
Query OK, 1 row affected (0.01 sec)

mysql> select * from department;
+---------------+----------------------+------------+-------------+
| 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 | Retails sales | 0 | 1700 |

| 260 | Recruiting | 0 | 1700 |

| 270 | Payroll | 0 | 1700 |

+---------------+----------------------+------------+-------------+

27 rows in set (0.01 sec)

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	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San	California	US
1600	2007 Zagora St	50090	South Brun	New Jersey	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
1800	147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA
1900	6092 Boxwood St	YSW 9T2	Whitehorse	Yukon	CA
2000	40-5-12 Laogianggen	190518	Beijing		CN
2100	1298 Vileparle (E)	490231	Bombay	Maharashtra	IN
2200	12-98 Victoria Stree	2901	Sydney	New South Wale	AU
2300	198 Clementi North	540198	Singapore		SG
2400	8204 Arthur St		London		UK
2500	Magdalen Centre, The	OX9 9ZB	Oxford	Oxford	UK
2600	9702 Chester Road	9629850293	Stretford	Manchester	UK
2700	Schwanthalerstr. 703	80925	Munich	Bavaria	DE

2800 Rua Frei Caneca 1360 01307-002 Sao Paulo Sao Paulo BR
2900 20 Rue des Corps-Sai 1730 Geneva Geneve CH
3000 Murtenstrasse 921 3095 Bern BE CH
3100 Pieter Breughelstraa 3029SK Utrecht Utrecht NL
3200 Mariano Escobedo 999 11932 Mexico Cit Distrito Feder MX

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

   Query:

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

   +------------+-------------+

   | first_name | last_name |

   +------------+-------------+

   | Steven | King |

   | Neena | Kochhar |

   | Lex | De Haan |

   | Nancy | Greenberg |

   | Den | Raphaely |

   | Alexander | Hunold |

   +------------+-------------+

   6 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:
   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 |
| Nancy     | Greenberg |  12000 |
| Den       | Raphaely  |  11000 |
+-----------+-----------+--------+
5 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.

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

```
+-------------+-------------+--------+
| first_name  | last_name   | salary |
+-------------+-------------+--------+
| Steven      | King        |  24000 |
|  Neena      |  Kochhar    |  17000 |
| Lex         | De Haan     |  17000 |
| Nancy       | Greenberg   |  12000 |
| Daniel      | Faviet      |   9000 |
```

John	Chen	8200
Jose Manuel	Urman	7800
Ismael	Sciarra	7700
Den	Raphaely	11000
Alexander	Hunold	9000
+--------------+-------------+--------+
10 rows in 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:

```
mysql> SELECT first_name, last_name,salary from employees where salary=(select min(salary) from employees);
+------------+-----------+--------+
| first_name | last_name | salary |
+------------+-----------+--------+
| Alexander | Khoo | 3100 |
+------------+-----------+--------+
1 row in set (0.01 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:

```
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 |
| Nancy | Greenberg | 12000 |
| Den | Raphaely | 11000 |
+------------+-----------+--------+
5 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.

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 |
+--------------+-------------+--------------+--------+
| Alexander    | Khoo        | PU_CLERK     |   3100 |
| Diana        | Lorentz     | IT_PROG      |   4200 |
| David        | Austin      | IT_PROG      |   4800 |
| Valli        | Pataballa   | IT_PROG      |   4800 |
| Bruce        | Ernst       | IT_PROG      |   6000 |
| Luis         | Popp        | FI_ACCOUNT   |   6900 |
| Ismael       | Sciarra     | FI_ACCOUNT   |   7700 |
| Jose Manuel  | Urman       | FI_ACCOUNT   |   7800 |
| John         | Chen        | FI_ACCOUNT   |   8200 |
| Daniel       | Faviet      | FI_ACCOUNT   |   9000 |
| Alexander    | Hunold      | IT_PROG      |   9000 |
| Den          | Raphaely    | PU_MAN       |  11000 |
| Nancy        | Greenberg   | FI_MGR       |  12000 |
| Neena        | Kochhar     | AD_VP        |  17000 |
| Lex          | De Haan     | AD_VP        |  17000 |
| Steven       | King        | AD_PRES      |  24000 |
+--------------+-------------+--------------+--------+
16 rows in set (0.00 sec)
```

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

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 (0.01 sec)

13. Write a query to display the employee ID, first name, last name, and department names of all employees.
    Queries:

mysql> select employee_id,first_name,last_name,(select department_name from department d where a.department_id=d.department_id) department from employees a order by department;

```
+-------------+--------------+-------------+--------------+
| employee_id | first_name   | last_name   | department   |
+-------------+--------------+-------------+--------------+
|         104 | Bruce        | Ernst       |  IT          |
|         105 | David        | Austin      |  IT          |
|         106 | Valli        | Pataballa   |  IT          |
|         107 | Diana        | Lorentz     |  IT          |
|         103 |  Alexander   |  Hunold     |  IT          |
|         115 |  Alexander   |  Khoo       |  Purchasing  |
|         114 | Den          | Raphaely    |  Purchasing  |
|         100 | Steven       | King        | Executive    |
|         101 |  Neena       |  Kochhar    | Executive    |
|         102 | Lex          | De Haan     | Executive    |
|         108 | Nancy        | Greenberg   | Finance      |
|         109 | Daniel       | Faviet      | Finance      |
|         110 | John         | Chen        | Finance      |
|         112 | Jose Manuel  | Urman       | Finance      |
|         111 | Ismael       | Sciarra     | Finance      |
|         113 | Luis         | Popp        | Finance      |
+-------------+--------------+-------------+--------------+
16 rows in set (0.01 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

    Queries:

    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 |
    +-------------+------------+-------------+--------+
    | 104 | Bruce | Ernst | 6000 |
    | 100 | Steven | King | 24000 |
    | 108 | Nancy | Greenberg | 12000 |
    | 109 | Daniel | Faviet | 9000 |
    | 114 | Den | Raphaely | 11000 |
    | 103 | Alexander | Hunold | 9000 |
    +-------------+------------+-------------+--------+

    6 rows in set (0.00 sec)

15. Write a query to fetch even numbered records from employees table.

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 |

+-------------+--------------+-----------+-----------+--------------+------------+-------------+----------------+------------+---------------+

| 104 | Bruce | Ernst | BERNST | 5904234568 | 1987-06-21 | IT_PROG | 6000 | 0 | 103 | 60 |

| 100 | Steven | King | SKING | 5151234567 | 1987-06-17 | AD_PRES | 24000 | 0 | 0 | 90 |

| 102 | Lex | De Haan | LDEHAAN | 5151234569 | 1987-06-19 | AD_VP | 17000 | 0 | 100 | 90 |

| 106 | Valli | Pataballa | VPATABAL | 5904234560 | 1987-06-23 | IT_PROG | 4800 | 0 | 103 | 60 |

| 108 | Nancy | Greenberg | NGREENBE | 5151244569 | 1987-06-25 | FI_MGR | 12000 | 0 | 101 | 100 |

| 110 | John | Chen | JCHEN | 5151244269 | 1987-06-27 | FI_ACCOUNT | 8200 | 0 | 108 | 100 |

| 112 | Jose Manuel | Urman | JMURMAN | 5151244469 | 1987-06-29 | FI_ACCOUNT | 7800 | 0 | 108 | 100 |

| 114 | Den | Raphaely | DRAPHEAL | 5151274561 | 1987-07-01 | PU_MAN | 11000 | 0 | 100 | 30 |

+-------------+--------------+-----------+-----------+--------------+------------+-------------+----------+----------------+------------+---------------+

8 rows in set (0.00 sec)

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

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

+--------+

| salary |

+--------+

| 9000 |

+--------+

1 row in set (0.00 sec)

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

Queries:

mysql> select distinct salary from employees a where 4=(select count(distinct salary) from employees b where b.salary<=a.salary);

+--------+

| salary |

```
+--------+
|   6000 |
+--------+
1 row in set (0.00 sec)
```

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

Queries:

```
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 |
+-------------+--------------+-----------+-----------+--------------+------------+--------------
+--------+---------------+------------+--------------+
|       106 | Valli       | Pataballa | VPATABAL  | 5904234560   | 1987-06-23
| IT_PROG     |   4800 |           0 |       103 |          60 |
|       107 | Diana       | Lorentz   | DLORENTZ  | 5904235567   | 1987-06-24
| IT_PROG     |   4200 |           0 |       103 |          60 |
|       108 | Nancy       | Greenberg | NGREENBE  | 5151244569   | 1987-06-
25 | FI_MGR      |  12000 |           0 |       101 |         100 |
|       109 | Daniel      | Faviet    | DFAVIET   | 5151244169   | 1987-06-26 |
FI_ACCOUNT   |   9000 |           0 |       108 |         100 |
|       110 | John        | Chen      | JCHEN     | 5151244269   | 1987-06-27 |
FI_ACCOUNT   |   8200 |           0 |       108 |         100 |
|       111 | Ismael      | Sciarra   | ISCIARRA  | 5151244369   | 1987-06-28 |
FI_ACCOUNT  |   7700 |           0 |       108 |         100 |
|       112 | Jose Manuel  | Urman     |  JMURMAN  | 5151244469   | 1987-
06-29 | FI_ACCOUNT   |   7800 |           0 |       108 |         100 |
|       113 | Luis        | Popp      | LPOPP     | 5151244567   | 1987-06-30 |
FI_ACCOUNT   |   6900 |           0 |       108 |         100 |
|       114 | Den         | Raphaely  | DRAPHEAL  | 5151274561   | 1987-07-
01 | PU_MAN      |  11000 |           0 |       100 |          30 |
|       115 |  Alexander   |  Khoo     |  AKHOO    | 5151274562   | 1987-07-02
|  PU_CLERK    |   3100 |           0 |       114 |          30 |
```

```
+------------+-------------+----------+----------+-------------+-----------+-------------
+--------+---------------+-----------+--------------+
10 rows in set (0.00 sec)
```

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

    Queries:

    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 |
|          50 |  Shipping            |       121 |       1500 |
|          70 |  Public Relations    |       204 |       2700 |
|          80 |  Sales               |       145 |       2500 |
|         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 | Retails sales        |         0 |       1700 |
|         260 | Recruiting           |         0 |       1700 |
|         270 | Payroll              |         0 |       1700 |
+---------------+----------------------+------------+-------------+
```

23 rows in set (0.02 sec)

20. Write a query to get 3 maximum salaries

Queries:

```
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 |
|  12000 |
+--------+
3 rows in set (0.00 sec)
```

21. Write a query to get 3 minimum salaries.

Queries:

```
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 |
+--------+
|   4800 |
|   4200 |
|   3100 |
+--------+
3 rows in set (0.00 sec)
```

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

```
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 | 5151234568 | 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 (0.00 sec)