

## Assignment-23

### Subqueries

Kaviya C

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:

```
mysql> SELECT first_name ,last_name ,salary FROM employees
-> WHERE salary> (SELECT salary from employees WHERE last_name='Bull');
```

| first_name  | last_name | salary   |
|-------------|-----------|----------|
| Alexander   | Hunold    | 9000.00  |
| Bruce       | ERNST     | 6000.00  |
| David       | Austin    | 4800.00  |
| Vali        | Pataballa | 4800.00  |
| Diana       | Lorentz   | 4200.00  |
| Daniel      | Faviet    | 9000.00  |
| John        | Chen      | 8200.00  |
| Ismael      | Sciarra   | 7700.00  |
| Jose Manuel | Urman     | 7800.00  |
| Luis        | Popp      | 6900.00  |
| Matthew     | Weiss     | 8000.00  |
| Adam        | Fripp     | 8200.00  |
| Payam       | Kaufling  | 7900.00  |
| Shanta      | Vollman   | 6500.00  |
| Kevin       | Mourgos   | 5800.00  |
| David       | Bernstein | 9500.00  |
| Peter       | Hall      | 90000.00 |
| Christopher | Olsen     | 8000.00  |
| Nanette     | Cambrault | 7500.00  |
| Oliver      | Tuvault   | 7000.00  |
| Patrick     | Sully     | 9500.00  |
| Allan       | McEwen    | 9000.00  |
| Lindsey     | Smith     | 8000.00  |
| Louise      | Doran     | 7500.00  |
| Sarath      | Sewall    | 7000.00  |
| Danielle    | Greene    | 9500.00  |
| Mattea      | Marvins   | 7200.00  |
| David       | Lee       | 6800.00  |
| Sundar      | Ande      | 6400.00  |
| Amit        | Banda     | 6200.00  |
| Tayler      | Fox       | 9600.00  |
| William     | Smith     | 7400.00  |
| Elizabeth   | Bates     | 7300.00  |
| Sundita     | Kumar     | 6100.00  |
| Alyssa      | hutton    | 8800.00  |

|           |            |         |
|-----------|------------|---------|
| Jonathon  | Taylor     | 8600.00 |
| Jack      | Livingston | 8400.00 |
| Kimberely | Grant      | 7000.00 |
| Charles   | Johnson    | 6200.00 |
| Nandita   | Sarchand   | 4200.00 |
| Jennifer  | Whalen     | 4400.00 |
| Pat       | Fay        | 6000.00 |
| Susan     | Mavris     | 6500.00 |
| William   | Gietz      | 8300.00 |

44 rows in set (0.00 sec)

- Write a query to find the name (first\_name, last\_name) of all employees who works in the IT department.

### Query:

```
mysql> SELECT first_name,last_name FROM employees WHERE department_id
-> IN (SELECT department_id FROM departments WHERE department_name='IT');
+-----+-----+
| first_name | last_name |
+-----+-----+
| Alexander | Hunold    |
| Bruce     | ERNST     |
| David     | Austin    |
| Vali      | Pataballa |
| Diana     | Lorentz   |
+-----+-----+
5 rows in set (0.10 sec)
```

- 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

### 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<br>country_id | street_address       | postal_code | city       | state_province |
|---------------------------|----------------------|-------------|------------|----------------|
| -----                     | -----                | -----       | -----      | -----          |
| 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  
MX

Mariano Escobedo 999 11932

Mexico Cit Distrito Feder

## Query:

```
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 departments WHERE location_id  
-> IN (SELECT location_id FROM locations WHERE country_id='US')));  
+-----+-----+
```

```
+-----+-----+  
| first_name | last_name |  
+-----+-----+  
| Neena      | Kochhar   |  
| Lex        | De Haan   |  
| Alexander  | Hunold    |  
| Bruce      | ERNST     |  
| David      | Austin    |  
| Vali       | Pataballa |  
| Diana      | Lorentz   |  
| Nancy      | GreenBerg |  
| Daniel     | Faviet    |  
| John       | Chen      |  
| Ismael     | Sciarra   |  
| Jose Manuel| Urman     |  
| Luis       | Popp      |  
| Den        | Raphaely  |  
| Alexander  | Khoo      |  
| Shelli     | Baida     |  
| Sigal      | Tobias    |  
| Guy        | Himuro    |  
| Karen      | Colmenares|  
| Matthew    | Weiss     |  
| Adam       | Fripp     |  
| Payam      | Kaufling  |  
| Shanta     | Vollman   |  
| Kevin      | Mourgos   |  
| Julia      | Nayer     |  
| Irene      | Mikilineni|  
| James      | Landry    |  
| Steven     | Markle    |  
| Laura      | Bissot    |  
| Mozhe      | Atkinson  |  
| James      | Marlow    |  
| TJ         | Olso      |  
| Jason      | Mallin    |  
| Michael     | Rogers    |  
| Ki         | Gee       |  
| Hazel      | Philtanker|  
+-----+-----+
```

|          |           |
|----------|-----------|
| Renske   | Ladwig    |
| Stephen  | Stiles    |
| John     | Seo       |
| Joshua   | Patel     |
| Trenna   | Rajs      |
| Curtis   | Davies    |
| Randall  | Matos     |
| Peter    | Vargas    |
| John     | Russell   |
| Karen    | Partners  |
| Alberto  | Errazuriz |
| Gerald   | Cambraut  |
| Eleni    | Zlotkey   |
| Winston  | Taylor    |
| Jean     | Fleaur    |
| Martha   | Sullivan  |
| Girard   | Geoni     |
| Nandita  | Sarchand  |
| Alexis   | Bull      |
| Julia    | Delinger  |
| Anthony  | Carbo     |
| Kelly    | Chung     |
| Jennifer | Dilly     |
| Timothy  | Gates     |
| Randall  | Perkins   |
| Sarah    | Bell      |
| Britney  | Everet    |
| Samuel   | McCain    |
| Vance    | Jones     |
| Alana    | Walsh     |
| Kevin    | Feeney    |
| Donald   | Oconnell  |
| Douglas  | Grant     |
| Jennifer | Whalen    |
| Michael  | Hartstein |
| Susan    | Mavris    |
| Hermann  | Baer      |
| Shelley  | Higgins   |
| William  | Gietz     |

+-----+

75 rows in set (0.07 sec)

- 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      |
| Lex        | De Haan   |
| Alexander  | Hunold    |
| Neena      | Kochhar   |
| Nancy      | GreenBerg |
| Den        | Raphaely  |
| Matthew    | Weiss     |
| Adam       | Fripp     |
| Payam      | Kaufling  |
| Shanta     | Vollman   |
| Kevin      | Mourgos   |
| John       | Russell   |
| Karen      | Partners  |
| Alberto    | Errazuriz |
| Gerald     | Cambraut  |
| Eleni      | Zlotkey   |
| Michael    | Hartstein |
| Shelley    | Higgins   |

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

- 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       | 24000.00 |
| Neena      | Kochhar    | 17000.00 |
| Lex        | De Haan    | 17000.00 |
| Alexander  | Hunold     | 9000.00  |
| Nancy      | GreenBerg  | 12000.00 |
| Daniel     | Faviet     | 9000.00  |
| John       | Chen       | 8200.00  |
| Den        | Raphaely   | 11000.00 |
| Alexander  | Khoo       | 31000.00 |
| Shelli     | Baida      | 29000.00 |
| Sigal      | Tobias     | 28000.00 |
| Guy        | Himuro     | 26000.00 |
| Adam       | Fripp      | 8200.00  |
| John       | Russell    | 14000.00 |
| Karen      | Partners   | 13500.00 |
| Alberto    | Errazuriz  | 12000.00 |
| Gerald     | Cambraut   | 11000.00 |
| Eleni      | Zlotkey    | 10500.00 |
| Peter      | Tucker     | 10000.00 |
| David      | Bernstein  | 9500.00  |
| Peter      | Hall       | 90000.00 |
| Janette    | King       | 10000.00 |
| Patrick    | Sully      | 9500.00  |
| Allan      | McEwen     | 9000.00  |
| Clara      | Vishney    | 10500.00 |
| Danielle   | Greene     | 9500.00  |
| Lisa       | Ozer       | 11500.00 |
| Harrison   | Bloom      | 10000.00 |
| Tayler     | Fox        | 9600.00  |
| Ellen      | Abel       | 11000.00 |
| Alyssa     | hutton     | 8800.00  |
| Jonathon   | Taylor     | 8600.00  |
| Jack       | Livingston | 8400.00  |
| Hermann    | Baer       | 10000.00 |
| Shelley    | Higgins    | 12000.00 |
| William    | Gietz      | 8300.00  |

```
36 rows in set (0.01 sec)
```

- 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.

Refer employees and jobs table

**Query:**

```
mysql> SELECT first_name,last_name,salary FROM employees
-> WHERE employees.salary=(SELECT min_salary FROM jobs
-> WHERE employees.job_id=jobs.job_id);
```

| first_name | last_name  | salary  |
|------------|------------|---------|
| Karen      | Colmenares | 2500.00 |
| Martha     | Sullivan   | 2500.00 |
| Randall    | Perkins    | 2500.00 |

3 rows in set (0.08 sec)

- 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.

Refer employees and departments

### Query:

```
mysql> SELECT first_name,last_name ,salary FROM employees
-> WHERE department_id in(SELECT department_id FROM departments
-> WHERE department_name LIKE 'IT%')
-> AND salary >(SELECT AVG(salary) FROM employees);
```

| first_name | last_name | salary  |
|------------|-----------|---------|
| Alexander  | Hunold    | 9000.00 |

1 row in set (0.00 sec)

- Write a query to find the name (first\_name, last\_name), and salary of the employees who earns more than the earning of Mr. Bell.

### Query:



```
mysql> SELECT first_name,last_name,salary FROM employees
-> WHERE salary> (SELECT salary FROM employees
-> WHERE last_name='Bell') ORDER BY first_name;
```

| first_name  | last_name  | salary  |
|-------------|------------|---------|
| Adam        | Fripp      | 8200.00 |
| Alexander   | Hunold     | 9000.00 |
| Alexis      | Bull       | 4100.00 |
| Allan       | McEwen     | 9000.00 |
| Alyssa      | hutton     | 8800.00 |
| Amit        | Banda      | 6200.00 |
| Bruce       | ERNST      | 6000.00 |
| Charles     | Johnson    | 6200.00 |
| Christopher | Olsen      | 8000.00 |
| Daniel      | Faviet     | 9000.00 |
| Danielle    | Greene     | 9500.00 |
| David       | Austin     | 4800.00 |
| David       | Lee        | 6800.00 |
| David       | Bernstein  | 9500.00 |
| Diana       | Lorentz    | 4200.00 |
| Elizabeth   | Bates      | 7300.00 |
| Ismael      | Sciarra    | 7700.00 |
| Jack        | Livingston | 8400.00 |
| Jennifer    | Whalen     | 4400.00 |
| John        | Chen       | 8200.00 |
| Jonathon    | Taylor     | 8600.00 |
| Jose Manuel | Urman      | 7800.00 |
| Kevin       | Mourgos    | 5800.00 |
| Kimberely   | Grant      | 7000.00 |
| Lindsey     | Smith      | 8000.00 |
| Louise      | Doran      | 7500.00 |
| Luis        | Popp       | 6900.00 |
| Mattea      | Marvins    | 7200.00 |
| Matthew     | Weiss      | 8000.00 |
| Nandita     | Sarchand   | 4200.00 |
| Nanette     | Cambrault  | 7500.00 |
| Oliver      | Tuvault    | 7000.00 |
| Pat         | Fay        | 6000.00 |
| Patrick     | Sully      | 9500.00 |
| Payam       | Kaufling   | 7900.00 |
| Peter       | Hall       | 9000.00 |

|         |           |         |
|---------|-----------|---------|
| Peter   | Hall      | 9000.00 |
| Sarath  | Sewall    | 7000.00 |
| Shanta  | Vollman   | 6500.00 |
| Sundar  | Ande      | 6400.00 |
| Sundita | Kumar     | 6100.00 |
| Susan   | Mavris    | 6500.00 |
| Tayler  | Fox       | 9600.00 |
| Vali    | Pataballa | 4800.00 |
| William | Smith     | 7400.00 |
| William | Gietz     | 8300.00 |

45 rows in set (0.06 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   |
|------------|-----------|----------|
| Peter      | Tucker    | 10000.00 |
| Janette    | King      | 10000.00 |
| Harrison   | Bloom     | 10000.00 |
| Hermann    | Baer      | 10000.00 |

4 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:**

```
mysql> SELECT * FROM employees WHERE salary>
-> ALL(SELECT AVG(salary)FROM employees GROUP BY department_id);
```

| 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  | 24000.00 | 0              | 0          | 90            |
| 115         | Alexander  | Khao      | AKHAO   | 515.123.4562       | 1987-07-02 | PU CLERK | 31000.00 | 0              | 114        | 30            |
| 116         | Shelli     | Baida     | SBAIDA  | 515.127.4563       | 1987-07-03 | PU CLERK | 29000.00 | 0              | 114        | 30            |
| 117         | Sigal      | Tobias    | STOBIAS | 515.127.4564       | 1987-07-04 | PU CLERK | 28000.00 | 0              | 114        | 30            |
| 118         | Guy        | Himuro    | GHIMURO | 515.127.4565       | 1987-07-05 | PU CLERK | 26000.00 | 0              | 114        | 30            |
| 152         | Peter      | Hall      | PHALL   | 011.44.1344.478968 | 1987-07-08 | SA REP   | 90000.00 | 0.25           | 145        | 80            |

6 rows in set (0.05 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:**

```
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  |
|-------------|------------|------------|---------|
| Jennifer    | Whalen     | AD_ASST    | 4400.00 |
| David       | Austin     | IT_PROG    | 4800.00 |
| Vali        | Pataballa  | IT_PROG    | 4800.00 |
| Kevin       | Mourgos    | ST_MAN     | 5800.00 |
| Bruce       | ERNST      | IT_PROG    | 6000.00 |
| Pat         | Fay        | MK_REP     | 6000.00 |
| Sundita     | Kumar      | SA_REP     | 6100.00 |
| Amit        | Banda      | SA_REP     | 6200.00 |
| Charles     | Johnson    | SA_REP     | 6200.00 |
| Sundar      | Ade        | SA_REP     | 6400.00 |
| Shanta      | Vollman    | ST_MAN     | 6500.00 |
| Susan       | Mavris     | HR_REP     | 6500.00 |
| David       | Lee        | SA_REP     | 6800.00 |
| Luis        | Popp       | FI_ACCOUNT | 6900.00 |
| Kimberely   | Grant      | SA_REP     | 7000.00 |
| Oliver      | Tuvault    | SA_REP     | 7000.00 |
| Sarath      | Sewall     | SA_REP     | 7000.00 |
| Mattea      | Marvins    | SA_REP     | 7200.00 |
| Elizabeth   | Bates      | SA_REP     | 7300.00 |
| William     | Smith      | SA_REP     | 7400.00 |
| Louise      | Doran      | SA_REP     | 7500.00 |
| Nanette     | Cambrault  | SA_REP     | 7500.00 |
| Ismael      | Sciarra    | FI_ACCOUNT | 7700.00 |
| Jose Manuel | Urman      | FI_ACCOUNT | 7800.00 |
| Payam       | Kaufling   | ST_MAN     | 7900.00 |
| Lindsey     | Smith      | SA_REP     | 8000.00 |
| Matthew     | Weiss      | ST_MAN     | 8000.00 |
| Christopher | Olsen      | SA_REP     | 8000.00 |
| John        | Chen       | FI_ACCOUNT | 8200.00 |
| Adam        | Fripp      | ST_MAN     | 8200.00 |
| William     | Gietz      | AC_ACCOUNT | 8300.00 |
| Jack        | Livingston | SA_REP     | 8400.00 |
| Jonathon    | Taylor     | SA_REP     | 8600.00 |
| Alyssa      | Hutton     | SA_REP     | 8800.00 |
| Alexander   | Hunold     | IT_PROG    | 9000.00 |
| Daniel      | Faviet     | FI_ACCOUNT | 9000.00 |
| Allan       | McEwen     | SA_REP     | 9000.00 |
| Peter       | Hall       | SA_REP     | 9000.00 |
| Patrick     | Sully      | SA_REP     | 9500.00 |
| Danielle    | Greene     | SA_REP     | 9500.00 |
| David       | Bernstein  | SA_REP     | 9500.00 |
| David       | Bernstein  | SA_REP     | 9500.00 |
| Taylor      | Fox        | SA_REP     | 9600.00 |

42 rows in set (0.00 sec)

- Write a query to find the name (first\_name, last\_name) of the employees who are not supervisors

## Query:

```
mysql> SELECT emp.first_name,emp.last_name FROM employees emp  
-> WHERE NOT EXISTS(SELECT 'X' FROM employees e  
-> WHERE e.manager_id=emp.employee_id);
```

| first_name  | last_name  |
|-------------|------------|
| Bruce       | ERNST      |
| David       | Austin     |
| Vali        | Pataballa  |
| Diana       | Lorentz    |
| Daniel      | Faviet     |
| John        | Chen       |
| Ismael      | Sciarra    |
| Jose Manuel | Urman      |
| Luis        | Popp       |
| Alexander   | Khoo       |
| Shelli      | Baida      |
| Sigal       | Tobias     |
| Guy         | Himuro     |
| Karen       | Colmenares |
| Julia       | Nayer      |
| Irene       | Mikilineni |
| James       | Landry     |
| Steven      | Markle     |
| Laura       | Bissot     |
| Mozhe       | Atkinson   |
| James       | Marlow     |
| TJ          | Olso       |
| Jason       | Mallin     |
| Michael     | Rogers     |
| Ki          | Gee        |
| Hazel       | Philtanker |
| Renske      | Ladwig     |
| Stephen     | Stiles     |
| John        | Seo        |
| Joshua      | Patel      |
| Trenna      | Rajs       |
| Curtis      | Davies     |
| Randall     | Matos      |
| Peter       | Vargas     |
| Peter       | Tucker     |

|             |            |
|-------------|------------|
| David       | Bernstein  |
| Peter       | Hall       |
| Christopher | Olsen      |
| Nanette     | Cambrault  |
| Oliver      | Tuvault    |
| Janette     | King       |
| Patrick     | Sully      |
| Allan       | McEwen     |
| Lindsey     | Smith      |
| Louise      | Doran      |
| Sarath      | Sewall     |
| Clara       | Vishney    |
| Danielle    | Greene     |
| Mattea      | Marvins    |
| David       | Lee        |
| Sundar      | Ande       |
| Amit        | Banda      |
| Lisa        | Ozer       |
| Harrison    | Bloom      |
| Tayler      | Fox        |
| William     | Smith      |
| Elizabeth   | Bates      |
| Sundita     | Kumar      |
| Ellen       | Abel       |
| Alyssa      | hutton     |
| Jonathon    | Taylor     |
| Jack        | Livingston |
| Kimberely   | Grant      |
| Charles     | Johnson    |
| Winston     | Taylor     |
| Jean        | Fleaur     |
| Martha      | Sullivan   |
| Girard      | Geoni      |
| Nandita     | Sarchand   |
| Alexis      | Bull       |
| Julia       | Delinger   |
| Anthony     | Carbo      |
| Kelly       | Chung      |
| Jennifer    | Dilly      |
| Timothy     | Gates      |
| Randall     | Perkins    |
| Sarah       | Bell       |
| Britney     | Everet     |

|          |          |
|----------|----------|
| Samuel   | McCain   |
| Vance    | Jones    |
| Alana    | Walsh    |
| Kevin    | Feeney   |
| Donald   | Oconnell |
| Douglas  | Grant    |
| Jennifer | Whalen   |
| Pat      | Fay      |
| Susan    | Mavris   |
| Hermann  | Baer     |
| William  | Gietz    |

89 rows in set (0.00 sec)

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

**Query:**

```
mysql> SELECT employee_id,first_name ,last_name,
-> (SELECT department_name FROM departments dept
-> WHERE emp.department_id=dept.department_id)
-> department FROM employees emp ORDER BY department;
```

| employee_id | first_name  | last_name  | department       |
|-------------|-------------|------------|------------------|
| 178         | Kimberely   | Grant      | NULL             |
| 205         | Shelley     | Higgins    | Accounting       |
| 206         | William     | Gietz      | Accounting       |
| 200         | Jennifer    | Whalen     | Administration   |
| 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          |
| 111         | Ismael      | Sciarra    | Finance          |
| 112         | Jose Manuel | Urman      | Finance          |
| 113         | Luis        | Popp       | Finance          |
| 203         | Susan       | Mavris     | Human Resources  |
| 103         | Alexander   | Hunold     | IT               |
| 104         | Bruce       | ERNST      | IT               |
| 105         | David       | Austin     | IT               |
| 106         | Vali        | Pataballa  | IT               |
| 107         | Diana       | Lorentz    | IT               |
| 202         | Pat         | Fay        | Marketing        |
| 201         | Michael     | Hartstein  | Marketing        |
| 204         | Hermann     | Baer       | Public Relations |
| 114         | Den         | Raphaely   | Purchasing       |
| 115         | Alexander   | Khoo       | Purchasing       |
| 116         | Shelli      | Baida      | Purchasing       |
| 117         | Sigal       | Tobias     | Purchasing       |
| 118         | Guy         | Himuro     | Purchasing       |
| 119         | Karen       | Colmenares | Purchasing       |
| 162         | Clara       | Vishney    | Sales            |
| 163         | Danielle    | Greene     | Sales            |
| 164         | Mattea      | Marvins    | Sales            |
| 165         | David       | Lee        | Sales            |
| 167         | Amit        | Banda      | Sales            |

|     |             |            |          |
|-----|-------------|------------|----------|
| 168 | Lisa        | Ozer       | Sales    |
| 169 | Harrison    | Bloom      | Sales    |
| 170 | Tayler      | Fox        | Sales    |
| 171 | William     | Smith      | Sales    |
| 172 | Elizabeth   | Bates      | Sales    |
| 173 | Sundita     | Kumar      | Sales    |
| 174 | Ellen       | Abel       | Sales    |
| 175 | Alyssa      | hutton     | Sales    |
| 176 | Jonathon    | Taylor     | Sales    |
| 177 | Jack        | Livingston | Sales    |
| 179 | Charles     | Johnson    | Sales    |
| 153 | Christopher | Olsen      | Sales    |
| 145 | John        | Russell    | Sales    |
| 146 | Karen       | Partners   | Sales    |
| 147 | Alberto     | Errazuriz  | Sales    |
| 148 | Gerald      | Cambraut   | Sales    |
| 149 | Eleni       | Zlotkey    | Sales    |
| 150 | Peter       | Tucker     | Sales    |
| 151 | David       | Bernstein  | Sales    |
| 152 | Peter       | Hall       | Sales    |
| 166 | Sundar      | Ande       | Sales    |
| 154 | Nanette     | Cambrault  | Sales    |
| 155 | Oliver      | Tuvault    | Sales    |
| 156 | Janette     | King       | Sales    |
| 157 | Patrick     | Sully      | Sales    |
| 158 | Allan       | McEwen     | Sales    |
| 159 | Lindsey     | Smith      | Sales    |
| 160 | Louise      | Doran      | Sales    |
| 161 | Sarath      | Sewall     | Sales    |
| 120 | Matthew     | Weiss      | Shipping |
| 121 | Adam        | Fripp      | Shipping |
| 122 | Payam       | Kaufling   | Shipping |
| 123 | Shanta      | Vollman    | Shipping |
| 124 | Kevin       | Mourgos    | Shipping |
| 125 | Julia       | Nayer      | Shipping |



|     |          |            |          |
|-----|----------|------------|----------|
| 126 | Irene    | Mikilineni | Shipping |
| 127 | James    | Landry     | Shipping |
| 128 | Steven   | Markle     | Shipping |
| 129 | Laura    | Bissot     | Shipping |
| 130 | Mozhe    | Atkinson   | Shipping |
| 131 | James    | Marlow     | Shipping |
| 132 | TJ       | Olso       | Shipping |
| 133 | Jason    | Mallin     | Shipping |
| 134 | Michael  | Rogers     | Shipping |
| 135 | Ki       | Gee        | Shipping |
| 144 | Peter    | Vargas     | Shipping |
| 140 | Joshua   | Patel      | Shipping |
| 143 | Randall  | Matos      | Shipping |
| 181 | Jean     | Fleaur     | Shipping |
| 182 | Martha   | Sullivan   | Shipping |
| 183 | Girard   | Geoni      | Shipping |
| 184 | Nandita  | Sarchand   | Shipping |
| 185 | Alexis   | Bull       | Shipping |
| 186 | Julia    | Delinger   | Shipping |
| 187 | Anthony  | Carbo      | Shipping |
| 188 | Kelly    | Chung      | Shipping |
| 189 | Jennifer | Dilly      | Shipping |
| 190 | Timothy  | Gates      | Shipping |
| 191 | Randall  | Perkins    | Shipping |
| 192 | Sarah    | Bell       | Shipping |
| 180 | Winston  | Taylor     | Shipping |
| 194 | Samuel   | McCain     | Shipping |
| 195 | Vance    | Jones      | Shipping |
| 196 | Alana    | Walsh      | Shipping |
| 197 | Kevin    | Feeney     | Shipping |
| 198 | Donald   | Oconnell   | Shipping |
| 199 | Douglas  | Grant      | Shipping |
| 142 | Curtis   | Davies     | Shipping |
| 141 | Trenna   | Rajs       | Shipping |
| 139 | John     | Seo        | Shipping |
| 138 | Stephen  | Stiles     | Shipping |
| 137 | Renske   | Ladwig     | Shipping |
| 136 | Hazel    | Philtanker | Shipping |
| 193 | Britney  | Everet     | Shipping |

107 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

**Query:**

```
mysql> SELECT employee_id ,first_name FROM employees as emp  
-> WHERE salary>(SELECT AVG(salary) FROM employees WHERE  
-> department_id=emp.department_id);
```

| employee_id | first_name |
|-------------|------------|
| 100         | Steven     |
| 103         | Alexander  |
| 104         | Bruce      |
| 108         | Nancy      |
| 109         | Daniel     |
| 115         | Alexander  |
| 116         | Shelli     |
| 117         | Sigal      |
| 118         | Guy        |
| 120         | Matthew    |
| 121         | Adam       |
| 122         | Payam      |
| 123         | Shanta     |
| 124         | Kevin      |
| 137         | Renske     |
| 141         | Trenna     |
| 145         | John       |
| 146         | Karen      |
| 147         | Alberto    |
| 152         | Peter      |
| 168         | Lisa       |
| 184         | Nandita    |
| 185         | Alexis     |
| 188         | Kelly      |
| 189         | Jennifer   |
| 192         | Sarah      |
| 193         | Britney    |
| 202         | Pat        |
| 205         | Shelley    |

29 rows in set (0.03 sec)

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

**Query:**

```
mysql> SET @i=0;
```

```
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> SELECT i,employee_id FROM (SELECT @i:=@i+1 AS i,employee_id  
-> FROM employees)a WHERE MOD(a.i,2)=0;
```

| +-----+-----+ |             |
|---------------|-------------|
| i             | employee_id |
| +-----+-----+ |             |
| 2             | 101         |
| 4             | 103         |
| 6             | 105         |
| 8             | 107         |
| 10            | 109         |
| 12            | 111         |
| 14            | 113         |
| 16            | 115         |
| 18            | 117         |
| 20            | 119         |
| 22            | 121         |
| 24            | 123         |
| 26            | 125         |
| 28            | 127         |
| 30            | 129         |
| 32            | 131         |
| 34            | 133         |
| 36            | 135         |
| 38            | 137         |
| 40            | 139         |
| 42            | 141         |
| 44            | 143         |
| 46            | 145         |
| 48            | 147         |
| 50            | 149         |
| 52            | 151         |

|     |     |
|-----|-----|
| 52  | 151 |
| 54  | 153 |
| 56  | 155 |
| 58  | 157 |
| 60  | 159 |
| 62  | 161 |
| 64  | 163 |
| 66  | 165 |
| 68  | 167 |
| 70  | 169 |
| 72  | 171 |
| 74  | 173 |
| 76  | 175 |
| 78  | 177 |
| 80  | 179 |
| 82  | 181 |
| 84  | 183 |
| 86  | 185 |
| 88  | 187 |
| 90  | 189 |
| 92  | 191 |
| 94  | 193 |
| 96  | 195 |
| 98  | 197 |
| 100 | 199 |
| 102 | 201 |
| 104 | 203 |
| 106 | 205 |

+-----+  
53 rows in set, 1 warning (0.07 sec)

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

**Query:**

```
mysql> SELECT DISTINCT salary FROM employees e1
-> WHERE 5=(SELECT COUNT(DISTINCT salary)FROM employees e2
-> WHERE e2.salary>=e1.salary);
```

| salary  |
|---------|
| 8800.00 |

+-----+  
1 row in set (0.05 sec)

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

**Query:**

```
mysql> SELECT DISTINCT salary FROM employees e1
-> WHERE 4=(SELECT COUNT(DISTINCT salary) FROM employees e2
-> WHERE e2.salary<=e1.salary);
+-----+
| salary |
+-----+
| 11500.00 |
+-----+
1 row in set (0.05 sec)
```

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

**Query:**

```
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 |
|-------------|------------|-----------|----------|--------------|------------|------------|----------|----------------|------------|---------------|
| 197         | Kevin      | Feeney    | BEVERETT | 650.507.9822 | 1987-07-22 | SH_CLERK   | 3000.00  | 0              | 124        | 50            |
| 198         | Donald     | Oconnell  | DOCONNEL | 650.507.9833 | 1987-07-23 | SH_CLERK   | 2600.00  | 0              | 124        | 50            |
| 199         | Douglas    | Grant     | DGRANT   | 650.507.9844 | 1987-07-24 | SH_CLERK   | 2600.00  | 0              | 124        | 50            |
| 200         | Jennifer   | Whalen    | JWHALEN  | 515.213.4444 | 1987-07-25 | AD_ASST    | 4400.00  | 0              | 101        | 10            |
| 201         | Michael    | Hartstein | MHARTSE  | 650.507.5555 | 1987-07-26 | MK_MAN     | 3900.00  | 0              | 100        | 20            |
| 202         | Pat        | Fay       | PFAY     | 650.507.6666 | 1987-07-27 | MK_REP     | 6000.00  | 0              | 201        | 20            |
| 203         | Susan      | Mavris    | BEVERETT | 650.507.7777 | 1987-07-28 | HR_REP     | 6500.00  | 0              | 101        | 40            |
| 204         | Hermann    | Baer      | BEVERETT | 650.507.8888 | 1987-07-29 | PR_REP     | 10000.00 | 0              | 101        | 70            |
| 205         | Shelley    | Higgins   | SHIGGINS | 650.507.8080 | 1987-07-30 | AC_MGR     | 12000.00 | 0              | 101        | 110           |
| 206         | William    | Gietz     | WGIEZT   | 650.507.2876 | 1987-07-01 | AC_ACCOUNT | 8300.00  | 0              | 205        | 110           |

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.

**Query:**

```
mysql> SELECT * FROM departments WHERE department_id
-> NOT IN (SELECT department_id FROM employees);
```

| department_id | department_name        | manager_id | location_id |
|---------------|------------------------|------------|-------------|
| 120           | Treasury               | 0          | 1700        |
| 130           | Corporate Tax          | 0          | 1700        |
| 140           | Control and Credit     | 0          | 1700        |
| 150           | Sharehold and 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        |

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

20. Write a query to get 3 maximum salaries

**Query:**

```
mysql> SELECT DISTINCT salary FROM employees emp1
-> WHERE 3>=(SELECT COUNT(DISTINCT salary) FROM employees emp2
-> WHERE emp2.salary>=emp1.salary) ORDER BY emp1.salary DESC;
```

| salary   |
|----------|
| 9600.00  |
| 9500.00  |
| 90000.00 |

```
3 rows in set (0.05 sec)
```

21. Write a query to get 3 minimum salaries.

**Query:**

```
mysql> SELECT DISTINCT salary FROM employees emp1
-> WHERE 3>=(SELECT COUNT(DISTINCT salary) FROM employees emp2
-> WHERE emp2.salary<=emp1.salary) ORDER BY emp1.salary DESC;
```

| salary   |
|----------|
| 11000.00 |
| 10500.00 |
| 10000.00 |

```
3 rows in set (0.05 sec)
```

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

## Query:

```
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 |
|-------------|------------|-----------|----------|--------------------|------------|--------|---------|----------------|------------|---------------|
| 151         | David      | Bernstein | DBERNSTE | 011.44.1344.345268 | 1987-07-07 | SA_REP | 9500.00 | 0.25           | 145        | 80            |
| 157         | Patrick    | Sully     | PSULLY   | 011.44.1344.929268 | 1987-07-13 | SA_REP | 9500.00 | 0.35           | 146        | 80            |
| 163         | Danielle   | Greene    | DGREENE  | 011.44.1344.229278 | 1987-07-19 | SA_REP | 9500.00 | 0.15           | 147        | 80            |

```
3 rows in set (0.05 sec)
```