Name: Daniel Thommana SQL ASSGN 4

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='Peter'

```
mysql> select * from employees where last_name='Peter';
Empty set (0.00 sec)

mysql> select * from employees where first_name='Peter';
+-------------+------------+-----------+---------+--------------------+------------+---------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email   | phone_number       | hire_date  | job_id  | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+---------+--------------------+------------+---------+----------+----------------+------------+---------------+
|         144 | Peter      | Vargas    | PVARGAS | 650.121.2004       | 1998-07-09 | ST_CLERK |  2500.00 |           NULL |        124 |            50 |
|         150 | Peter      | Tucker    | PTUCKER | 011.44.1344.129268 | 1997-01-30 | SA_REP   | 10000.00 |           0.30 |        145 |            80 |
|         152 | Peter      | Hall      | PHALL   | 011.44.1344.478968 | 1997-08-20 | SA_REP   |  9000.00 |           0.25 |        145 |            80 |
+-------------+------------+-----------+---------+--------------------+------------+---------+----------+----------------+------------+---------------+
3 rows in set (0.00 sec)
```

```
mysql> select first_name,last_name,salary from employees where salary > (select max(salary) from employees where first_name="Peter");
+------------+-----------+----------+
| first_name | last_name | salary   |
+------------+-----------+----------+
| Steven     | King      | 24000.00 |
| Neena      | Kochhar   | 17000.00 |
| Lex        | De Haan   | 17000.00 |
| Nancy      | Greenberg | 12000.00 |
| Den        | Raphaely  | 11000.00 |
| John       | Russell   | 14000.00 |
| Karen      | Partners  | 13500.00 |
| Alberto    | Errazuriz | 12000.00 |
| Gerald     | Cambrault | 11000.00 |
| Eleni      | Zlotkey   | 10500.00 |
| Clara      | Vishney   | 10500.00 |
| Lisa       | Ozer      | 11500.00 |
| Ellen      | Abel      | 11000.00 |
| Michael    | Hartstein | 13000.00 |
| Shelley    | Higgins   | 12000.00 |
+------------+-----------+----------+
15 rows in set (0.00 sec)
```

2. Write a query to find the name (first_name, last_name) and the salary of the employees who have a higher salary than their manager.

```
mysql> select E.first_name EmpFName,E.last_name EmpLName,E.salary EmpSalary, M.first_name MngrFName,M.last_name MngrLName,M.salary MngrSalary from employees M join employees E on M.employee_id=E.manager_id
where E.salary > M.salary;
+---------+---------+-----------+-----------+-----------+------------+
| EmpFName | EmpLName | EmpSalary | MngrFName | MngrLName | MngrSalary |
+---------+---------+-----------+-----------+-----------+------------+
| Lisa     | Ozer     |  11500.00 | Gerald    | Cambrault |   11000.00 |
| Ellen    | Abel     |  11000.00 | Eleni     | Zlotkey   |   10500.00 |
+---------+---------+-----------+-----------+-----------+------------+
2 rows in set (0.00 sec)
```

3. Write a query to find the name (first_name, last_name) of all employees who works in the IT department along with their manager name;

```
mysql> select E.first_name EmpFName,E.last_name EmpLName,M.first_name MngrFName,M.last_name MngrLName,D.department_name from employees M join employees E on M.employee_id=E.manager_id join departments D
on D.department_id=E.department_id where department_name="IT";
+-----------+----------+-----------+-----------+-----------------+
| EmpFName  | EmpLName | MngrFName | MngrLName | department_name |
+-----------+----------+-----------+-----------+-----------------+
| Alexander | Hunold   | Lex       | De Haan   | IT              |
| Bruce     | Ernst    | Alexander | Hunold    | IT              |
| David     | Austin   | Alexander | Hunold    | IT              |
| Valli     | Pataballa | Alexander | Hunold    | IT              |
| Diana     | Lorentz  | Alexander | Hunold    | IT              |
+-----------+----------+-----------+-----------+-----------------+
5 rows in set (0.00 sec)
```

4. 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.

there is no job grade table available in the database so used job_id

```
mysql> select concat( first_name," ", last_name) as EmpName, salary,job_id FROM employees e  WHERE e.salary = (SELECT min_salary FROM jobs j WHERE e.job_id = j.job_id);
+-------------------+---------+----------+
| EmpName           | salary  | job_id   |
+-------------------+---------+----------+
| Karen Colmenares | 2500.00 | PU_CLERK |
| Martha Sullivan  | 2500.00 | SH_CLERK |
| Randall Perkins  | 2500.00 | SH_CLERK |
+-------------------+---------+----------+
3 rows in set (0.00 sec)
```

Least salary per job

```
mysql> select concat(first_name," ",last_name) as EmpName,salary as Least_salary,job_id from (select first_name,last_name,salary,job_id,dense_rank() over (partition by job_id order by  salary asc) r
from employees ) as e  where r=1;
+-------------------+-------------+------------+
| EmpName           | Least_salary | job_id     |
+-------------------+-------------+------------+
| William Gietz     |      8300.00 | AC_ACCOUNT |
| Shelley Higgins   |     12000.00 | AC_MGR     |
| Jennifer Whalen   |      4400.00 | AD_ASST    |
| Steven King       |     24000.00 | AD_PRES    |
| Neena Kochhar     |     17000.00 | AD_VP      |
| Lex De Haan       |     17000.00 | AD_VP      |
| Luis Popp         |      6900.00 | FI_ACCOUNT |
| Nancy Greenberg   |     12000.00 | FI_MGR     |
| Susan Mavris      |      6500.00 | HR_REP     |
| Diana Lorentz     |      4200.00 | IT_PROG    |
| Michael Hartstein |     13000.00 | MK_MAN     |
| Pat Fay           |      6000.00 | MK_REP     |
| Hermann Baer      |     10000.00 | PR_REP     |
| Karen Colmenares  |      2500.00 | PU_CLERK   |
| Den Raphaely      |     11000.00 | PU_MAN     |
| Eleni Zlotkey     |     10500.00 | SA_MAN     |
| Sundita Kumar     |      6100.00 | SA_REP     |
| Martha Sullivan   |      2500.00 | SH_CLERK   |
| Randall Perkins   |      2500.00 | SH_CLERK   |
| TJ Olson          |      2100.00 | ST_CLERK   |
| Kevin Mourgos     |      5800.00 | ST_MAN     |
+-------------------+-------------+------------+
21 rows in set (0.00 sec)
```

5. 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.

```
mysql> select e.first_name,e.last_name,e.salary,d.department_name from employees e join departments d on d.department_id = e.department_id where e.salary > (select avg(salary) from employees)
AND d.department_name="IT";
+------------+-----------+---------+-----------------+
| first_name | last_name | salary  | department_name |
+------------+-----------+---------+-----------------+
| Alexander  | Hunold    | 9000.00 | IT              |
+------------+-----------+---------+-----------------+
1 row in set (0.00 sec)
```

6. 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.

```
mysql> select concat( first_name," ", last_name) as EmpName,salary,department_id,job_id from employees where salary = (select min(salary) from employees);
+----------+---------+---------------+----------+
| EmpName  | salary  | department_id | job_id   |
+----------+---------+---------------+----------+
| TJ Olson | 2100.00 |            50 | ST_CLERK |
+----------+---------+---------------+----------+
1 row in set (0.00 sec)
```

Least salary per department

```
mysql> select concat(first_name," ",last_name) as EmpName,salary as Least_salary,department_id from (select first_name,last_name,salary,department_id,dense_rank() over (partition by department_id
 order by  salary asc) r from employees ) as e  where r=1;
+-------------------+-------------+---------------+
| EmpName           | Least_salary | department_id |
+-------------------+-------------+---------------+
| Kimberely Grant   |      7000.00 |          NULL |
| Jennifer Whalen   |      4400.00 |            10 |
| Pat Fay           |      6000.00 |            20 |
| Karen Colmenares  |      2500.00 |            30 |
| Susan Mavris      |      6500.00 |            40 |
| TJ Olson          |      2100.00 |            50 |
| Diana Lorentz     |      4200.00 |            60 |
| Hermann Baer      |     10000.00 |            70 |
| Sundita Kumar     |      6100.00 |            80 |
| Neena Kochhar     |     17000.00 |            90 |
| Lex De Haan       |     17000.00 |            90 |
| Luis Popp         |      6900.00 |           100 |
| William Gietz     |      8300.00 |           110 |
+-------------------+-------------+---------------+
13 rows in set (0.00 sec)
```

7. 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 8.0 Command Line Client

```
61 rows in set (0.00 sec)

mysql> select first_name,last_name,salary from employees where salary > (select max(salary) from employees where JOB_ID = 'SH_CLERK') order by salary;
+-------------+-----------+----------+
| first_name  | last_name | salary   |
+-------------+-----------+----------+
| Jennifer    | Whalen    |  4400.00 |
| David       | Austin    |  4800.00 |
| Valli       | Pataballa |  4800.00 |
| Kevin       | Mourgos   |  5800.00 |
| Bruce       | Ernst     |  6000.00 |
| Pat         | Fay       |  6000.00 |
| Sundita     | Kumar     |  6100.00 |
| Amit        | Banda     |  6200.00 |
| Charles     | Johnson   |  6200.00 |
| Sundar      | Ande      |  6400.00 |
| Shanta      | Vollman   |  6500.00 |
| Susan       | Mavris    |  6500.00 |
| David       | Lee       |  6800.00 |
| Luis        | Popp      |  6900.00 |
| Oliver      | Tuvault   |  7000.00 |
| Sarath      | Sewall    |  7000.00 |
| Kimberely   | Grant     |  7000.00 |
| Mattea      | Marvins   |  7200.00 |
| Elizabeth   | Bates     |  7300.00 |
| William     | Smith     |  7400.00 |
| Louise      | Doran     |  7500.00 |
| Nanette     | Cambrault |  7500.00 |
| Ismael      | Sciarra   |  7700.00 |
| Jose Manuel | Urman     |  7800.00 |
| Payam       | Kaufling  |  7900.00 |
| Christopher | Olsen     |  8000.00 |
| Matthew     | Weiss     |  8000.00 |
| Lindsey     | Smith     |  8000.00 |
| John        | Chen      |  8200.00 |
| Adam        | Fripp     |  8200.00 |
| William     | Gietz     |  8300.00 |
| Jack        | Livingston|  8400.00 |
| Jonathon    | Taylor    |  8600.00 |
| Alyssa      | Hutton    |  8800.00 |
| Daniel      | Faviet    |  9000.00 |
| Peter       | Hall      |  9000.00 |
| Allan       | McEwen    |  9000.00 |
| Alexander   | Hunold    |  9000.00 |
| David       | Bernstein |  9500.00 |
| Patrick     | Sully     |  9500.00 |
| Danielle    | Greene    |  9500.00 |
| Tayler      | Fox       |  9600.00 |
| Janette     | King      | 10000.00 |
| Peter       | Tucker    | 10000.00 |
| Hermann     | Baer      | 10000.00 |
| Harrison    | Bloom     | 10000.00 |
| Clara       | Vishney   | 10500.00 |
| Eleni       | Zlotkey   | 10500.00 |
| Ellen       | Abel      | 11000.00 |
| Den         | Raphaely  | 11000.00 |
| Gerald      | Cambrault | 11000.00 |
| Lisa        | Ozer      | 11500.00 |
| Nancy       | Greenberg | 12000.00 |
| Alberto     | Errazuriz | 12000.00 |
| Shelley     | Higgins   | 12000.00 |
| Michael     | Hartstein | 13000.00 |
| Karen       | Partners  | 13500.00 |
| John        | Russell   | 14000.00 |
| Lex         | De Haan   | 17000.00 |
| Neena       | Kochhar   | 17000.00 |
| Steven      | King      | 24000.00 |
+-------------+-----------+----------+
61 rows in set (0.00 sec)
```

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

```
mysql> SELECT employee_id, first_name,last_name,salary,department_id FROM employees AS A WHERE salary > (SELECT AVG(salary) FROM employees WHERE department_id = A.department_id) order by department_id;
+-------------+------------+-----------+----------+---------------+
| employee_id | first_name | last_name | salary   | department_id |
+-------------+------------+-----------+----------+---------------+
|         201 | Michael    | Hartstein | 13000.00 |            20 |
|         114 | Den        | Raphaely  | 11000.00 |            30 |
|         137 | Renske     | Ladwig    |  3600.00 |            50 |
|         141 | Trenna     | Rajs      |  3500.00 |            50 |
|         120 | Matthew    | Weiss     |  8000.00 |            50 |
|         121 | Adam       | Fripp     |  8200.00 |            50 |
|         122 | Payam      | Kaufling  |  7900.00 |            50 |
|         123 | Shanta     | Vollman   |  6500.00 |            50 |
|         124 | Kevin      | Mourgos   |  5800.00 |            50 |
|         184 | Nandita    | Sarchand  |  4200.00 |            50 |
|         185 | Alexis     | Bull      |  4100.00 |            50 |
|         188 | Kelly      | Chung     |  3800.00 |            50 |
|         189 | Jennifer   | Dilly     |  3600.00 |            50 |
|         192 | Sarah      | Bell      |  4000.00 |            50 |
|         193 | Britney    | Everett   |  3900.00 |            50 |
|         103 | Alexander  | Hunold    |  9000.00 |            60 |
|         104 | Bruce      | Ernst     |  6000.00 |            60 |
|         151 | David      | Bernstein |  9500.00 |            80 |
|         150 | Peter      | Tucker    | 10000.00 |            80 |
|         157 | Patrick    | Sully     |  9500.00 |            80 |
|         158 | Allan      | McEwen    |  9000.00 |            80 |
|         162 | Clara      | Vishney   | 10500.00 |            80 |
|         163 | Danielle   | Greene    |  9500.00 |            80 |
|         168 | Lisa       | Ozer      | 11500.00 |            80 |
|         169 | Harrison   | Bloom     | 10000.00 |            80 |
|         170 | Tayler     | Fox       |  9600.00 |            80 |
|         174 | Ellen      | Abel      | 11000.00 |            80 |
|         156 | Janette    | King      | 10000.00 |            80 |
|         152 | Peter      | Hall      |  9000.00 |            80 |
|         145 | John       | Russell   | 14000.00 |            80 |
|         146 | Karen      | Partners  | 13500.00 |            80 |
|         147 | Alberto    | Errazuriz | 12000.00 |            80 |
|         148 | Gerald     | Cambrault | 11000.00 |            80 |
|         149 | Eleni      | Zlotkey   | 10500.00 |            80 |
|         100 | Steven     | King      | 24000.00 |            90 |
|         108 | Nancy      | Greenberg | 12000.00 |           100 |
|         109 | Daniel     | Faviet    |  9000.00 |           100 |
|         205 | Shelley    | Higgins   | 12000.00 |           110 |
+-------------+------------+-----------+----------+---------------+
38 rows in set (0.01 sec)
```

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

```
mysql> select concat(first_name," ",last_name) as EmpName,salary from(select first_name, last_name,salary, dense_rank() over(order by salary desc)r from employees) as e where r=5;
+-------------------+----------+
| EmpName           | salary   |
+-------------------+----------+
| Michael Hartstein | 13000.00 |
+-------------------+----------+
1 row in set (0.00 sec)
```

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

```
mysql> select d.department_id,d.department_name from employees e right join departments d on d.department_id=e.department_id where employee_id is null;
+---------------+----------------------+
| department_id | department_name      |
+---------------+----------------------+
|           120 | Treasury             |
|           130 | Corporate Tax        |
|           140 | Control And Credit   |
|           150 | Shareholder Services |
|           160 | Benefits             |
|           170 | Manufacturing        |
|           180 | Construction         |
|           190 | Contracting          |
|           200 | Operations           |
|           210 | IT Support           |
|           220 | NOC                  |
|           230 | IT Helpdesk          |
|           240 | Government Sales     |
|           250 | Retail Sales         |
|           260 | Recruiting           |
|           270 | Payroll              |
+---------------+----------------------+
16 rows in set (0.00 sec)
```