SQL Assignments

1. Display UnitPrice which is the maximum in the store

```
mysql> select max(unitprice) from item;
+----------------+
| max(unitprice) |
+----------------+
|       48000.00 |
+----------------+
1 row in set (0.01 sec)
```

2. Display the total number of items in the store.

```
mysql> select count(*) from item;
+----------+
| count(*) |
+----------+
|       24 |
+----------+
1 row in set (0.00 sec)
```

3. Display customerId and total number of orders placed by each customer.

```
mysql> select  customerid, count(orderid) from ordermaster group by customerid;
+------------+----------------+
| customerid | count(orderid) |
+------------+----------------+
|       1001 |              1 |
|       1005 |              3 |
|       1006 |              3 |
+------------+----------------+
3 rows in set (0.02 sec)
```

4. Display ItemCategory and average UnitPrice in each item category.

```
mysql> select itemcategory, avg(unitprice) from item group by itemcategory;
+-----------------+----------------+
| itemcategory    | avg(unitprice) |
+-----------------+----------------+
| Mobile          |   10666.666667 |
| Refrigerator    |   18714.285714 |
| Stationery      |       5.500000 |
| Television      |   27000.000000 |
| Washing Machine |   19500.000000 |
+-----------------+----------------+
5 rows in set (0.00 sec)
```

5. Display customerId of customers who have placed more than 1 order.

```
mysql> select customerid from ordermaster group by customerid having count(orderid) > 1;
+------------+
| customerid |
+------------+
|       1005 |
|       1006 |
+------------+
2 rows in set (0.00 sec)
```

6. Display ItemCategory of items which has the minimum unit price more than INR 10,000.

```
mysql> select itemcategory, min(unitprice) from item group by itemcategory having min(unitprice) > 10000;
+-----------------+----------------+
| itemcategory    | min(unitprice) |
+-----------------+----------------+
| Refrigerator    |       12000.00 |
| Television      |       15000.00 |
| Washing Machine |       11000.00 |
+-----------------+----------------+
3 rows in set (0.00 sec)
```

7. Display ItemCategory, total number of items for "Television" and "Refrigerator" if the total number of items exceeds 5. Display the results in the descending order of total number of items.

```
mysql> select itemcategory, count(itemcode) from item where itemcategory = 'Television' or itemcategory = 'Refrigerator' group by itemcategory having count(itemcode) >
5 order by count(itemcode) desc;
+-------------+----------------+
| itemcategory | count(itemcode) |
+-------------+----------------+
| Refrigerator |              7 |
| Television   |              6 |
+-------------+----------------+
2 rows in set (0.00 sec)
```

1. Display CustomerName and PhoneNumer of customers who have placed a single order of more than INR 45,000.

```
mysql> select customername, phonenumber from customer inner join ordermaster on customer.customerid = ordermaster.customerid where totalorderamount > 45000 group by ord
ermaster.customerid having count(orderid) = 1;
+-------------+-------------+
| customername | phonenumber |
+-------------+-------------+
| Mario        | 1234567890  |
+-------------+-------------+
1 row in set (0.00 sec)

mysql>
```

2. Display CustomerName and PhoneNumber of Customers who have placed orders in the month of June in year 2016.

```
mysql> select customername, phonenumber from customer inner join ordermaster on customer.customerid = ordermaster.customerid where orderdate like '2016-06-__';
+-------------+-------------+
| customername | phonenumber |
+-------------+-------------+
| Jacob        | 1234567895  |
| Phil         | 1234567894  |
+-------------+-------------+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

3. Display ItemCode, ItemName and UnitPrice of items which were not ordered by any customer.

```
mysql> select itemcode, itemname, unitprice from item where itemcode not in (select distinct itemcode from ordertransaction);
+----------+-----------------+-----------+
| itemcode | itemname        | unitprice |
+----------+-----------------+-----------+
| IT104    | Dual Door       |  12000.00 |
| IT106    | Pencil          |      2.00 |
| IT107    | LG G5 Silver    |  19000.00 |
| IT108    | Top Load        |  25000.00 |
| IT110    | Eraser          |      5.00 |
| IT112    | Sony Xperia Z5  |   9000.00 |
| IT113    | Fully Automatic |  22000.00 |
| IT116    | LED 32 Inch     |  20000.00 |
| IT118    | Semi Automatic  |  11000.00 |
| IT119    | Three DoorLux   |  18000.00 |
| IT120    | Sharpener       |      5.00 |
| IT122    | Three Door      |  21000.00 |
| IT123    | LED 50 Inch     |  38000.00 |
| IT124    | Three Door      |  22000.00 |
| IT125    | Four Door       |  21000.00 |
+----------+-----------------+-----------+
15 rows in set (0.02 sec)
```

4. Display itemCode and ItemName of items that are ordered exactly once.

```
mysql> select ot.itemcode, item.itemname from ordertransaction ot inner join item on ot.itemcode = item.itemcode group by ot.itemcode having count(ot.itemcode) = 1;
+----------+------------+
| itemcode | itemname   |
+----------+------------+
| IT102    | Galaxy S5  |
| IT103    | Front Load |
| IT109    | Four Door  |
| IT111    | LED 32 Inch |
| IT117    | Onida i505 |
| IT121    | LED 50 Inch |
+----------+------------+
6 rows in set (0.00 sec)
```

5. Display customerId of customers who have placed more than one order.

```
mysql> select customerid from ordermaster group by customerid having count(orderid) > 1;
+-------------+
| customerid |
+-------------+
|        1005 |
|        1006 |
+-------------+
2 rows in set (0.00 sec)
```

6. Display CustomerName and PhoneNumber of Customers who have placed order for most expensive item in the store.

```
mysql> select customername, phonenumber from customer where customerid = (select customerid from ordermaster where orderid in(select orderid from ordertransaction where
 itemcode in(select itemcode from item where unitprice = (select max(unitprice) from item))));
+-------------+------------+
| customername | phonenumber |
+-------------+------------+
| Phil        | 1234567894  |
+-------------+------------+
1 row in set (0.00 sec)
```

7. Display OrderId, CustomerId, CustomerName, Address and PhoneNumber for all the orders placed.

```
mysql> select om.orderid, om.customerid, c.customername, c.address, c.phonenumber from ordermaster om inner join customer c on om.customerid = c.customerid order by ord
erid;
+---------+------------+--------------+---------------------------------------+-------------+
| orderid | customerid | customername | address                               | phonenumber |
+---------+------------+--------------+---------------------------------------+-------------+
|   70001 |       1001 | Mario        | Street: 1, Cross: 2, Town: 3, Pin: 1231 | 1234567890  |
|   70002 |       1006 | Jacob        | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567895  |
|   70003 |       1005 | Phil         | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567894  |
|   70004 |       1006 | Jacob        | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567895  |
|   70005 |       1005 | Phil         | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567894  |
|   70006 |       1006 | Jacob        | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567895  |
|   70007 |       1005 | Phil         | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567894  |
+---------+------------+--------------+---------------------------------------+-------------+
7 rows in set (0.00 sec)
```

8. Display ItemCode, ItemName, QtyOrdered, UnitPrice for OrderId 70002.

```
mysql> select i.itemcode, i.itemname, ot.qtyordered, i.unitprice, ot.orderid from item i inner join ordertransaction ot on i.itemcode = ot.itemcode where ot.orderid = 7
0002;
+----------+------------+------------+-----------+---------+
| itemcode | itemname   | qtyordered | unitprice | orderid |
+----------+------------+------------+-----------+---------+
| IT103    | Front Load |          1 |  20000.00 |   70002 |
| IT109    | Four Door  |          1 |  18000.00 |   70002 |
| IT111    | LED 32 Inch |          2 |  22000.00 |   70002 |
+----------+------------+------------+-----------+---------+
3 rows in set (0.01 sec)
```

9. Display OrderId, CustomerId, CustomerName, Address and Phone for all the orders placed. Include the details about the customer even if there are no orders placed by the customer.

```
mysql> select om.orderid, om.customerid, c.customername, c.address, c.phonenumber from customer c left outer join ordermaster om on om.customerid = c.customerid order b
y orderid;
+---------+------------+--------------+----------------------------------------+-------------+
| orderid | customerid | customername | address                                | phonenumber |
+---------+------------+--------------+----------------------------------------+-------------+
|    NULL |       NULL | Merlin       | Street: 12, Cross: 4, Town: 15, Pin: 1235 | 1234567903  |
|    NULL |       NULL | Megan        | Street: 1, Cross: 2, Town: 3, Pin: 1231   | 1234567891  |
|    NULL |       NULL | Morgan       | Street: 12, Cross: 4, Town: 15, Pin: 1235 | 1234567904  |
|    NULL |       NULL | Amy          | Street: 1, Cross: 2, Town: 3, Pin: 1231   | 1234567892  |
|    NULL |       NULL | Stuart       | Street: 1, Cross: 2, Town: 3, Pin: 1231   | 1234567893  |
|    NULL |       NULL | James        | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567896  |
|    NULL |       NULL | Dan          | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567897  |
|    NULL |       NULL | Henry        | Street: 12, Cross: 4, Town: 15, Pin: 1235 | 1234567898  |
|    NULL |       NULL | Eric         | Street: 12, Cross: 4, Town: 15, Pin: 1235 | 1234567899  |
|    NULL |       NULL | Ken          | Street: 12, Cross: 4, Town: 15, Pin: 1235 | 1234567901  |
|    NULL |       NULL | Mecon        | Street: 12, Cross: 4, Town: 15, Pin: 1235 | 1234567902  |
|   70001 |       1001 | Mario        | Street: 1, Cross: 2, Town: 3, Pin: 1231   | 1234567890  |
|   70002 |       1006 | Jacob        | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567895  |
|   70003 |       1005 | Phil         | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567894  |
|   70004 |       1006 | Jacob        | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567895  |
|   70005 |       1005 | Phil         | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567894  |
|   70006 |       1006 | Jacob        | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567895  |
|   70007 |       1005 | Phil         | Street: 2, Cross: 4, Town: 5, Pin: 1232   | 1234567894  |
+---------+------------+--------------+----------------------------------------+-------------+
18 rows in set (0.00 sec)
```

10. Display the ItemCode, UnitPrice of the all the items in each ItemCategory where the unitprice is less than the average unitprice for the itemcategory.

```
mysql> select itemcode, unitprice from item where unitprice < (select avg(unitprice) from item);
+----------+-----------+
| itemcode | unitprice |
+----------+-----------+
| IT101    |  15000.00 |
| IT102    |  13000.00 |
| IT104    |  12000.00 |
| IT106    |      2.00 |
| IT110    |      5.00 |
| IT112    |   9000.00 |
| IT115    |     10.00 |
| IT117    |  10000.00 |
| IT118    |  11000.00 |
| IT120    |      5.00 |
+----------+-----------+
10 rows in set (0.00 sec)
```

Aggregate Functions

1. Display the maximum of salary of the company.

```
mysql> select max(salary) from employee;
+-------------+
| max(salary) |
+-------------+
|   840000.00 |
+-------------+
1 row in set (0.01 sec)
```

2. Display the average salary of the company.

```
mysql> select avg(salary) from employee;
+---------------+
| avg(salary)   |
+---------------+
| 188350.657895 |
+---------------+
1 row in set (0.00 sec)
```

3. Display the maximum salary of employees who are TLs.

```
mysql> select max(salary) from employee where designation = 'TL';
+-------------+
| max(salary) |
+-------------+
|    92400.00 |
+-------------+
1 row in set (0.02 sec)
```

4. Display the total number of employees in the company.

```
mysql> select count(*) from employee;
+----------+
| count(*) |
+----------+
|       19 |
+----------+
1 row in set (0.01 sec)
```

5. Display the total number of Managers in the company. (If an employee is playing the role of the supervisor for any other employee then the employee is considered as Manager).

```
mysql> select count(distinct managerEmployeeNumber) from employee;
+---------------------------------------+
| count(distinct managerEmployeeNumber) |
+---------------------------------------+
|                                    10 |
+---------------------------------------+
1 row in set (0.01 sec)
```

6. Display the total number SMs in the company.

```
mysql> select count(*) from employee where designation = 'SM';
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)
```

GROUP BY

1. Display designation and number of employees in each designation.

```
mysql> select designation, count(*) from employee group by designation;
+-------------+----------+
| designation | count(*) |
+-------------+----------+
| CEO         |        1 |
| MD          |        3 |
| SE          |        5 |
| SM          |        3 |
| SSE         |        3 |
| TL          |        4 |
+-------------+----------+
6 rows in set (0.00 sec)
```

2. Display designation and maximum salary for each designation.

```
mysql> select designation, max(salary) from employee group by designation;
+-------------+-------------+
| designation | max(salary) |
+-------------+-------------+
| CEO         |   840000.00 |
| MD          |   525000.00 |
| SE          |    36750.00 |
| SM          |   252000.00 |
| SSE         |    41895.00 |
| TL          |    92400.00 |
+-------------+-------------+
6 rows in set (0.00 sec)
```

3. Display Designation and maximum salary for each designation. Display the results in the decreasing order of maximum salary.

```
mysql> select designation, max(salary) from employee group by designation order by max(salary) desc;
+-------------+-------------+
| designation | max(salary) |
+-------------+-------------+
| CEO         |   840000.00 |
| MD          |   525000.00 |
| SM          |   252000.00 |
| TL          |    92400.00 |
| SSE         |    41895.00 |
| SE          |    36750.00 |
+-------------+-------------+
6 rows in set (0.00 sec)
```

4. Display DepartmentCode and number of employees working for each department.

```
mysql> select departmentcode, count(*) from employee group by departmentcode;
+-----------------+----------+
| departmentcode | count(*) |
+-----------------+----------+
| NULL           |        1 |
| .NETCap        |        7 |
| JavaCap        |        5 |
| LKM            |        6 |
+-----------------+----------+
4 rows in set (0.02 sec)
```

5. Display Designation and maximum salary for ‘TL’ and ‘SSE’

```
mysql> select designation, max(salary) from employee where designation = 'TL' or designation = 'SSE' group by designation;
+-------------+-------------+
| designation | max(salary) |
+-------------+-------------+
| SSE         |    41895.00 |
| TL          |    92400.00 |
+-------------+-------------+
2 rows in set (0.00 sec)
```

6. Display ManagerEmployeeNumber and Number of employees working under the Manager (Exclude Null from ManagerEmployeeNumber column).

```
mysql> select manageremployeenumber, count(*) from employee group by manageremployeenumber having manageremployeenumber is not null;
+----------------------+----------+
| manageremployeenumber | count(*) |
+----------------------+----------+
|                 7001 |        4 |
|                 7002 |        1 |
|                 7003 |        1 |
|                 7004 |        1 |
|                 7005 |        1 |
|                 7006 |        1 |
|                 7007 |        2 |
|                 7008 |        2 |
|                 7009 |        2 |
|                 7010 |        2 |
+----------------------+----------+
10 rows in set (0.00 sec)
```

7. Display DepartmentCode and NumberOfEmployees if the department has more than 5 employees.

```
mysql> select departmentcode, count(*) from employee group by departmentcode having count(*)>5;
+-----------------+----------+
| departmentcode | count(*) |
+-----------------+----------+
| .NETCap        |        7 |
| LKM            |        6 |
+-----------------+----------+
2 rows in set (0.00 sec)
```

8. Display DepartmentCode and average salary if the average salary of the department is more than INR150,000 (Exclude Null under DepartmentCode column)

```
mysql> select departmentcode, avg(salary) from employee group by departmentcode having avg(salary) > 150000 and departmentcode is not null;
+----------------+--------------+
| departmentcode | avg(salary)  |
+----------------+--------------+
| JavaCap        | 186973.500000 |
| LKM            | 154857.500000 |
+----------------+--------------+
2 rows in set (0.00 sec)
```

9. Display Designation and average salary of each designation for “LKM” department if the average salary is more than INR 35,000. Display the results in the increasing order of average salary.

```
mysql> select designation, avg(salary) from employee where departmentcode = 'LKM' group by designation having avg(salary) > 35000 order by avg(salary);
+-------------+--------------+
| designation | avg(salary)  |
+-------------+--------------+
| SSE         |  41895.000000 |
| TL          |  90825.000000 |
| SM          | 231000.000000 |
| MD          | 441000.000000 |
+-------------+--------------+
4 rows in set (0.02 sec)
```

10. Display ProjectId, number of employees working in the project. Display the results in the decreasing order of number of employees (Exclude the results if the enddate is not null).

```
mysql> select projectid, count(employeenumber) from employeeprojects where enddate is null group by projectid order by count(employeenumber) desc;
+-----------+----------------------+
| projectid | count(employeenumber) |
+-----------+----------------------+
| P1        |                    3 |
| P3        |                    3 |
| P2        |                    1 |
+-----------+----------------------+
3 rows in set (0.00 sec)
```

Non-Correlated Subqueries

1. Display EmployeeName, Salary of employees whose salary is more than the average salary of the company.

```
mysql> select employeename, salary from employee where salary > (select avg(salary) from employee);
+--------------+-----------+
| employeename | salary    |
+--------------+-----------+
| Cynthya      | 840000.00 |
| Mario        | 525000.00 |
| Jacob        | 420000.00 |
| Lucy         | 441000.00 |
| Amy          | 252000.00 |
| Frank        | 231000.00 |
| Phil         | 231000.00 |
+--------------+-----------+
7 rows in set (0.00 sec)
```

2. Display EmployeeName, Salary of employee(s) who is getting the lowest salary in the company.

```
mysql> select employeename, salary from employee where salary = (select min(salary) from employee);
+--------------+----------+
| employeename | salary   |
+--------------+----------+
| Darwin       | 31500.00 |
| Dan          | 31500.00 |
| Joseph       | 31500.00 |
+--------------+----------+
3 rows in set (0.00 sec)
```

3. Display EmployeeName of employees who are working in project ‘P1’.

```
mysql> select employeename from employee where employeenumber in (select employeenumber from employeeprojects where projectid = 'p1');
+--------------+
| employeename |
+--------------+
| Amy          |
| Frank        |
| Stuart       |
+--------------+
3 rows in set (0.00 sec)
```

4. Display ProjectName of projects which has more than 2 employees (Exclude rows if end date is not null).

```
mysql> select projectname from project where projectid in (select projectid from employeeprojects where enddate is null group by projectid having count(employeenumber)
> 2);
+-------------+
| projectname |
+-------------+
| Retail      |
| Resources   |
+-------------+
2 rows in set (0.00 sec)
```

5. Display EmployeeName of Managers who have more than three team members.

```
mysql> select employeename from employee where employeenumber in (select manageremployeenumber from employee group by manageremployeenumber having count(employeenumber)
 > 3);
+--------------+
| employeename |
+--------------+
| Cynthya      |
+--------------+
1 row in set (0.00 sec)
```

6. Display the second maximum salary of the company.

```
mysql> select max(salary) from employee where salary < (select max(salary) from employee);
+--------------+
| max(salary) |
+--------------+
|   525000.00 |
+--------------+
1 row in set (0.00 sec)
```

7. Display the ProjectName of projects which currently does not have any employees.

```
mysql> select projectname from project where projectid not in(select projectid from employeeprojects);
+------------------+
| projectname      |
+------------------+
| Banking          |
| Internal Project |
+------------------+
2 rows in set (0.00 sec)
```

Correlated Subqueries

1. Display EmployeeName, Salary of employees whose salary is more than the average salary of the department they belong to.

```
mysql> select employeename, salary from employee e, (select departmentcode, avg(salary) av_sal from employee group by departmentcode) e1 where e.departmentcode = e1.dep
artmentcode and e.salary > e1.av_sal;
+--------------+-----------+
| employeename | salary    |
+--------------+-----------+
| Jacob        | 420000.00 |
| Frank        | 231000.00 |
| Mario        | 525000.00 |
| Amy          | 252000.00 |
| Lucy         | 441000.00 |
| Phil         | 231000.00 |
+--------------+-----------+
6 rows in set (0.00 sec)
```

2. Display EmployeeName of employees whose manager is younger than the employee.

```
mysql> select e.employeename, e.dob from employee e inner join employee m on e.manageremployeenumber = m.employeenumber where e.dob < m.dob;
+--------------+------------+
| employeename | dob        |
+--------------+------------+
| Phil         | 1974-12-11 |
| Stuart       | 1980-05-23 |
+--------------+------------+
2 rows in set (0.00 sec)
```

Inner Join

1. Display EmployeeName, DepartmentCode and DepartmentName of ALL employees.

```
mysql> select employeename, department1.departmentcode, departmentname from employee inner join department1 on employee.departmentcode = department1.departmentcode;
+--------------+---------------+----------------------------------+
| employeename | departmentcode | departmentname                   |
+--------------+---------------+----------------------------------+
| Jacob        | .NETCap        | Dotnet Capability                |
| Frank        | .NETCap        | Dotnet Capability                |
| Jack         | .NETCap        | Dotnet Capability                |
| Darwin       | .NETCap        | Dotnet Capability                |
| Dan          | .NETCap        | Dotnet Capability                |
| James        | .NETCap        | Dotnet Capability                |
| Joseph       | .NETCap        | Dotnet Capability                |
| Mario        | JavaCap        | Java Capability                  |
| Amy          | JavaCap        | Java Capability                  |
| Arnold       | JavaCap        | Java Capability                  |
| Stuart       | JavaCap        | Java Capability                  |
| Clarke       | JavaCap        | Java Capability                  |
| Lucy         | LKM            | Learning and Knowledge Management |
| Phil         | LKM            | Learning and Knowledge Management |
| Justin       | LKM            | Learning and Knowledge Management |
| Megan        | LKM            | Learning and Knowledge Management |
| Chelsea      | LKM            | Learning and Knowledge Management |
| Jimmy        | LKM            | Learning and Knowledge Management |
+--------------+---------------+----------------------------------+
18 rows in set (0.00 sec)
```

2. Display EmployeeName, ProjectName and StartDate of employees who are currently working on the project (include only if EndDate is NULL).

```
mysql> select e.employeename, p.projectname, ep.startdate from employee e inner join employeeprojects ep on ep.employeenumber = e.employeenumber inner join project p on
 p.projectid = ep.projectid where ep.enddate is null;
+--------------+-------------+------------+
| employeename | projectname | startdate  |
+--------------+-------------+------------+
| Amy          | Retail      | 2014-07-01 |
| Frank        | Retail      | 2016-06-01 |
| Stuart       | Retail      | 2015-04-01 |
| Dan          | Insurance   | 2014-07-16 |
| Phil         | Resources   | 2015-05-11 |
| Clarke       | Resources   | 2015-02-28 |
| Darwin       | Resources   | 2014-11-11 |
+--------------+-------------+------------+
7 rows in set (0.00 sec)
```

Outer Join

1. Display EmployeeName, ProjectId of ALL employees even if an employee is not assigned to any project (Include only if EndDate is Null).

```
mysql> select employeename, projectid from employee e left outer join employeeprojects ep on e.employeenumber = ep.employeenumber where ep.enddate is null;
+--------------+-----------+
| employeename | projectid |
+--------------+-----------+
| Cynthya      | NULL      |
| Mario        | NULL      |
| Jacob        | NULL      |
| Amy          | P1        |
| Frank        | P1        |
| Phil         | P3        |
| Arnold       | NULL      |
| Jack         | NULL      |
| Justin       | NULL      |
| Megan        | NULL      |
| Stuart       | P1        |
| Clarke       | P3        |
| Darwin       | P3        |
| Chelsea      | NULL      |
| Dan          | P2        |
| Jimmy        | NULL      |
| James        | NULL      |
| Joseph       | NULL      |
+--------------+-----------+
18 rows in set (0.00 sec)
```

2. Display EmployeeName, ProjectId of ALL employees even if an employee not assigned to any project (Include only if EndDate is Null). Display “Not Allocated” if ProjectId is null.

```
mysql> select employeename, ifnull(projectid, 'Not Allocated') from employee e left outer join employeeprojects ep on e.employeenumber = ep.employeenumber where ep.endd
ate is null;
+--------------+-----------------------------------+
| employeename | ifnull(projectid, 'Not Allocated') |
+--------------+-----------------------------------+
| Cynthya      | Not Allocated                     |
| Mario        | Not Allocated                     |
| Jacob        | Not Allocated                     |
| Amy          | P1                                |
| Frank        | P1                                |
| Phil         | P3                                |
| Arnold       | Not Allocated                     |
| Jack         | Not Allocated                     |
| Justin       | Not Allocated                     |
| Megan        | Not Allocated                     |
| Stuart       | P1                                |
| Clarke       | P3                                |
| Darwin       | P3                                |
| Chelsea      | Not Allocated                     |
| Dan          | P2                                |
| Jimmy        | Not Allocated                     |
| James        | Not Allocated                     |
| Joseph       | Not Allocated                     |
+--------------+-----------------------------------+
18 rows in set (0.01 sec)
```

SELF Join

1. Display EmployeeName, DateOfBirth of Employee, ManagerName and DateOfBirth of Manager.

```
mysql> select e.employeename, e.dob, m.employeename as 'Manager Name', m.dob as 'Manager DOB' from employee e inner join employee m on e.manageremployeenumber = m.emplo
yeenumber;
+--------------+------------+--------------+-------------+
| employeename | dob        | Manager Name | Manager DOB |
+--------------+------------+--------------+-------------+
| Cynthya      | 1975-05-12 | Cynthya      | 1975-05-12  |
| Mario        | 1976-02-14 | Cynthya      | 1975-05-12  |
| Jacob        | 1976-05-16 | Cynthya      | 1975-05-12  |
| Lucy         | 1978-05-15 | Cynthya      | 1975-05-12  |
| Amy          | 1978-09-16 | Mario        | 1976-02-14  |
| Frank        | 1978-09-17 | Jacob        | 1976-05-16  |
| Phil         | 1974-12-11 | Lucy         | 1978-05-15  |
| Arnold       | 1984-03-13 | Amy          | 1978-09-16  |
| Jack         | 1984-09-23 | Frank        | 1978-09-17  |
| Justin       | 1984-11-07 | Phil         | 1974-12-11  |
| Megan        | 1984-07-21 | Phil         | 1974-12-11  |
| Stuart       | 1980-05-23 | Arnold       | 1984-03-13  |
| Clarke       | 1994-02-24 | Arnold       | 1984-03-13  |
| Darwin       | 1992-05-03 | Jack         | 1984-09-23  |
| Chelsea      | 1994-01-19 | Justin       | 1984-11-07  |
| Dan          | 1991-05-27 | Jack         | 1984-09-23  |
| Jimmy        | 1993-08-11 | Justin       | 1984-11-07  |
+--------------+------------+--------------+-------------+
17 rows in set (0.00 sec)
```