Theory Part:

Task 1:

1. Describe the functional dependencies in the given tables.

**Answer: Functional Dependency** is a constraint that determines the relation of one attribute to another attribute in a Database Management System (DBMS). Functional Dependency helps to maintain the quality of data in the database. It plays a vital role to find the difference between good and bad database design.

| **Employee no** | **Employee Name** | **Salary** | **Department** |
| --- | --- | --- | --- |
| 1 | Amit | 50000 | OBIEE |
| 2 | Divya | 40000 | COGNOS |
| 3 | Rama | 57000 | Administrator |

1. For each of the given tables of this database determine they are in which normal form.

### Answer: Database is in 2nd Normal Form (2NF). Because The first condition in the 2nd NF is that the table has to be in 1st NF. The table also should not contain partial dependency. Here partial dependency means the proper subset of candidate key determines a non-prime attribute. To understand in a better way lets look at the below example.

1. Convert all the tables to 3rd normal forms.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Employee No | Employee Name | Department no | Department | Salary Slip No | Salary |
| 1 | Amit | OBIEE,ETL | 101 | 0001 | 50000 |
| 2 | Divya | COGNOS | 102 | 0002 | 40000 |
| 2 | Rama | Administrator | 103 | 0003 | 57000 |

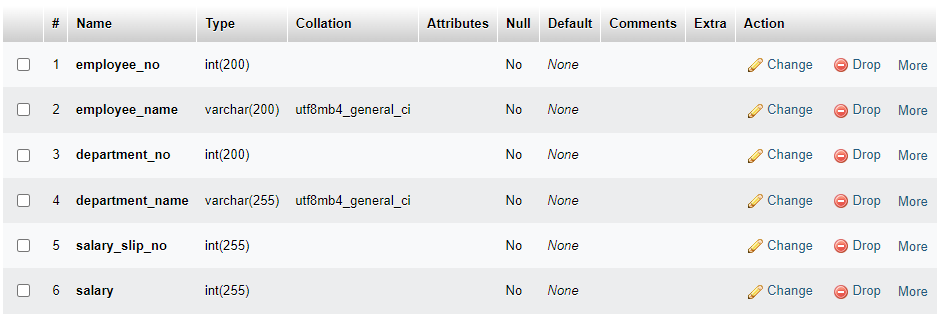
1. Design an enhanced entity relationship model based on normalized tables.

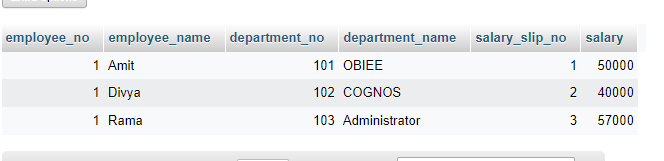
Answer: Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency. Normalized table has been created Screenshot has given below.

Lab Part:

Task 2:

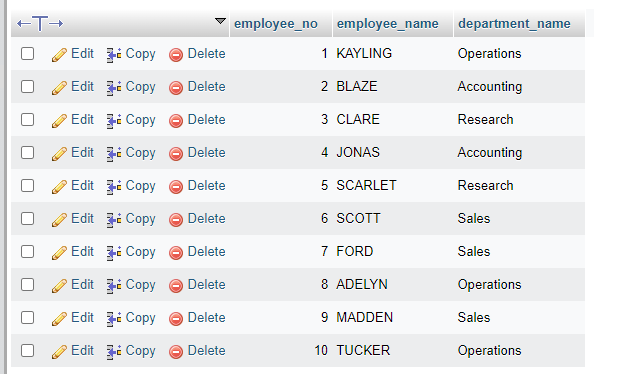
1. Develop a database on normalized tables and show the results through screenshots.



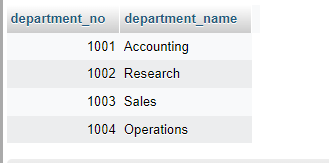


1. Insert at least 10 entity in each tables.

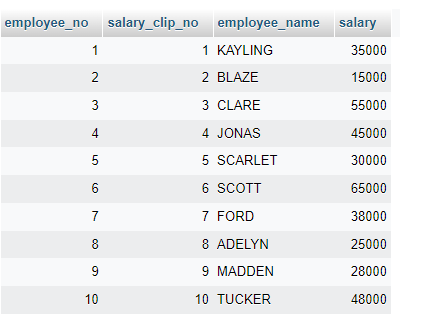
Employee Table:



Department Table:



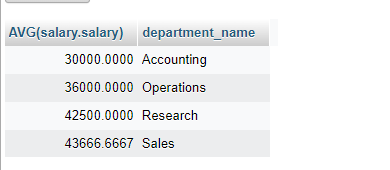
Salary Table:



1. Write a query to show average salary of employee each department.

Answer:

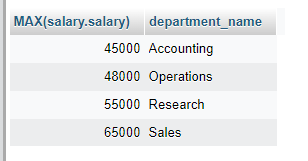
[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(salary.salary), employee.department\_name FROM salary INNER JOIN employee ON salary.employee\_name = employee.employee\_name GROUP BY employee.department\_name;



1. Write a query to show second maximum salary of employee in each department.

Answer:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [MAX](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_max)(salary.salary), employee.department\_name FROM salary INNER JOIN employee ON salary.employee\_name = employee.employee\_name GROUP BY employee.department\_name;



1. Write a query to show total number of employee in each department.

Answer:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(employee.department\_name) FROM salary INNER JOIN employee ON salary.employee\_name = employee.employee\_name GROUP BY employee.department\_name;

