NORMALIZATION

**DEFINITION :**

Normalization is the process to eliminate data redundancy and enhance data integrity in

the table. Normalization also helps to organize the data in the database. It is a multi-

step process that sets the data into tabular form and removes the duplicated data from

the relational tables.

Normalization organizes the columns and tables of a database to ensure that

database integrity constraints properly execute their dependencies. It is a

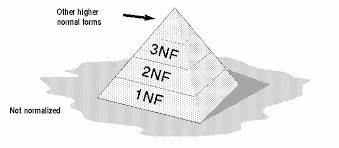
systematic technique of decomposing tables to eliminate data redundancy (repetition)

and undesirable characteristics like Insertion, Update, and Deletion anomalies.

**TYPES OF NORMALIZATION :**

The Normalization was classified into many forms . Some types of normalization

are explained:



1. **FIRST NORMAL FORM (1NF)**

* A table is referred to as being in its First Normal Form if atomicity of the table is 1.
* Here, atomicity states that a single cell cannot hold multiple values. It must hold only

a single-valued attribute.

* The First normal form disallows the multi-valued attribute, composite attribute, and
* their combinations.

**2 . SECOND NORMAL FORM (2NF)**

**2NF** builds on 1NF and requires that all non-key attributes are fully functionally depen-

dent on the entire primary key.

To achieve 2NF, you need to split the data into multiple tables to ensure that non-

key attributes are fully dependent on the primary key.

Now you will understand the second Normal Form with the help of an example.

**QUERY :**

select \* from organization

create table employee as

select emp\_id,emp\_name,dept\_id,manager\_id,job\_title,salary,location from organization

create table department as

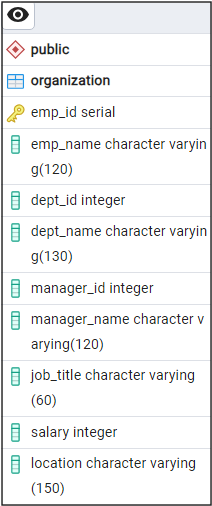
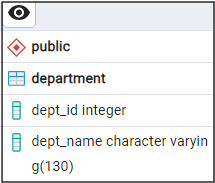
select dept\_id,dept\_name

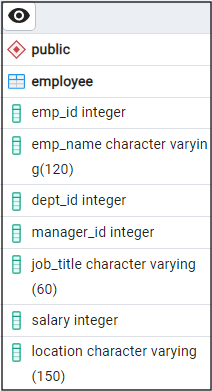
from organization

create table manager as

select manager\_id, manager\_name from organization

**OUTPUT**:





1. **THIRD NORMAL FORM (3NF)**

**3NF** requires that all attributes are not only fully functionally dependent on the

Primary key but also that there are no transitive dependencies (i.e., non-key attributes are

Dependent on other primary key).

**QUERY :**

Select e.emp\_id,e.emp\_name,e.dept\_id,e.manager\_id,e.job\_title,e.salary,e.location

from employee as e

inner join department as d

on e.dept\_id=d.dept\_id

inner join manager as m

on e.manager\_id=m.manager\_id

**OUTPUT :**

