Normalization

Definition: Normalization is a process by which we can divide a large relation of table into smaller one in a non-loss way, i.e., we can get some or more information from the smaller relation than a one large relation.

The decomposition of relation depends on certain rule.

1NF: A relation R is said to be in the first normal form (1NF) if and only if all the attributes of the relation R are atomic in nature i.e., there does not exist any repeating group.

- Remove repeating groups.
- Disallow multivalued attributes.

Go through PPT of NF.

Ex1.1: We have consider another example:

Consider **DEPT** table as:

DEPT (Dno, Dname, Dmarg, Dlocation)

Dno	Dname	Dmarg	Dlocation
D04	Admin	334	Bangalore, New Delhi, Hydrabad
D05	Research	335	Kolkata, Chennai

The above table is not in 1NF. So we can represent the table in the following way

DEPT

Dno	Dname	Dmarg	Dlocation
D04	Admin	334	Bangalore
D04	Admin	334	New Delhi
D04	Admin	334	Hydrabad
D05	Research	335	Kolkata
D05	Research	335	Chennai

We may break the DEPT table into two tables in the following way:

DEPT

Dno	Dname	Dmarg
D04	Admin	334
D05	Research	335

LOC

Dno	Dlocation
D04	Bangalore
D04	New Delhi
D04	Hydrabad
D05	Kolkata
D05	Chennai

Hence this relation is in 1 NF.

2NF: It is in 1NF and all non-key attributes are fully functionally dependent on prime attribute.

- Remove all partial dependencies

Go through PPT of NF

Ex2.1- Consider the relation is defined as:

EMP (Eno, Pno, Hours, Ename, Pname, Plocation) and the dependencies are defined as

Eno → Ename
Pno → Pname, Plocation
Eno, Pno → Hours

Therefore, **EMP** is not in 2NF and hence break the table in the following way:

EMP1 (Eno, Ename)

EMP2 (Pno, Pname, Plocation)

EMP3 (Eno, Pno, Hours)

Now the above relation is in 2NF.

Ex2.2- Consider the relation is defined as:

Student (Reg_no, Name, Course_name, Course_Id, Fees, Duration, Voter_Id) such that

Reg_no → Name, Course_name, Course_Id, Fees, Duration
Voter_Id → Name, Course_name, Course_Id, Fees, Duration
Course_Id → Course_name
Course_name → Fees, Duration

Here, Reg_no and Voter_Id are primary key and composite key. Therefore, we can write Reg_no, Voter_Id → Name, Course_name, Course_Id, Fees, Duration

If we remove Reg_no or Voter_Id then dependency exist, so it is partial dependency. Hence it is not in 2NF.

Therefore, we break the relation into two relations as:

S1 (Reg_no, Name, Course_name, Course_Id, Fees, Duration) and **S2** (Reg_no, Voter_Id)

No partial dependency exist and hence relation is in 2NF.

Note: In **S1**, Reg_no is primary key since Reg_no is more closed than Voter_Id with respect to other keys in relation Student. This property is called *closeness* property.

Reg_no, Voter_Id → Voter_Id

Reg_no, Voter_Id → Reg_No

3NF: It is in 2NF and all non-key attributes are non-transitively dependent on key attribute.

- Remove all transitive dependencies

Go through PPT of NF

Ex3.1- Consider the relation is defined as:

EMP (Eno, Ename, DOB, Address, Dno, Dname, Dmargno) and the dependencies are defined as

Eno → Ename, DOB, Address
Dno → Dname, Dmargno
Eno → Dno

Therefore, **EMP** is not in 3NF and hence break the table in the following way:

EMP (Eno, Ename, DOB, Address)

DEPT (Dno, Dname, Dmargno)

EMP1(Eno, Dno)

Ex3.2- Consider the relation **Student** is defined in Ex2.2.

After removing partial dependency we have **S1** (Reg_no, Name, Course_name, Course_Id, Fees, Duration) and **S2** (Reg_no, Voter_Id)

Now from **S1** the dependencies are as follows:

Course_Id → Course_name
Course_name → Fees, Duration

Transitive exists on above relation. So it is not in 3NF and hence we break the table **S1** as:
S11 (Reg_no, Name, Course_Id)
S12 (Course_name, Course_Id)
S13 (Course_name, Fees, Duration)
S2 (Reg_no, Voter_Id)

Hence the above relation is in 3NF.

BCNF (Boyce-Codd Normal Form): A relation R is in BCNF iff every determinate is a candidate key.

Determinate: An attribute (simple or composite) is called a determinate on which some other attribute is fully functional dependent.

- It must be in 3NF
- All determinates are super key

Go through PPT of NF

Ex4.1- Go through PPT

Ex4.2- Consider the following relation:

T (A, B, C) such that A, B → C and C → B

The above relation is in 3NF but it is not in BCNF (C is not super key)

Ex4.3- The relation **S11, S12** and **S13** (from Ex3.2) are not in BCNF because Course_name consists of multiple courses such as M.Tech degree in CSE, IT, ECE and similarly B.Tech degree in CSE, CSBS, ECE, EE.

If course name is unique (university conducted only B.Tech in CSBS) then it will in BCNF.

Ex4.4- Consider a relation **R** (A, B, C, D) such that

A→B,C,D

B,C → A,D

D→A \\ D+={A}

Now discuss as:

A→B,C,D	BCNF	A is super key
B,C → A,D	BCNF	B,C are super key
D→A	Not BCNF	D is not super key

3NF vs BCNF

3NF	BCNF
1. It concentrate on primary key	1. It concentrate on candidate key
2. Redundancy is higher as compared to BCNF	2. 0% redundancy
3. It may preserve all the dependencies.	4. It may not preserve the dependencies.
5. X→Y allowed if X is a super key or y is a part of some key.	3. X→Y is allowed if X is a super key.

Exercise: Consider a relation **Report** is defined as follows:

Report (RID, Editor, Deptno, D_name, D_address, AID, A_name, A_address) and the dependencies is described as:

RID → Editor
Editor → Deptno
Deptno → D_name, D_address
AID → A_name, A_address