Section - 8
Normalization for Refinement of Database

2 Approaches to design database :

1. E-R Modeling
2. Normalization

E-R Modeling	Normalization
1. Identifying entity and relations	1. Refinement of database
2. Conversion of ER diagram to tables	2. Design database by using tests or normal forms
3. Top - Down Approach	3. Bottom up approach

Normalization

→ We start with a larger table, we apply some tests, if tests are satisfied => Normalized table

→ If tests are not satisfied => unnormalized table and we decompose that larger table into smaller tables so that individual table clears that test.

→ It is a process of decomposing a larger table into smaller tables so that it satisfy series of tests. If the database satisfies the test, then database is considered normalized according to that test or rule or degree, otherwise database is considered unnormalized.

Section - 8
Normalization for Refinement of Database

2 Approaches to design database:

1. E-R Modeling
2. Normalization

E-R Modeling	Normalization
1. Identifying entity and relations	1. Refinement of database
2. Conversion of ER diagram to tables	2. Design database by using tests or normal forms
3. Top - Down Approach	3. Bottom up approach

Normalization

→ We start with a larger table, we apply some tests, if tests are satisfied ⇒ Normalized table

→ If tests are not satisfied ⇒ unnormalized table and we decompose that larger table into smaller tables so that individual table clears that test.

→ It is a process of decomposing a larger table into smaller tables so that it satisfy series of tests. If the database satisfies the test, then database is considered normalized according to that test or rule or degree, otherwise database is considered un-normalized.

When a test fails, the relation violating that test must be decomposed into relations so that it individually meet the normalization tests

There are 5 series of test that we apply on the database, so there are 5 degree or rules of normalization which are known as Normal forms

eg: first normal form, Second Normal form and so on

Objectives of Normalization

1. To create a formal framework to verify that designed database is accurate and it ensures that we can perform insertion, update, delete and retrieve operation without any anomalies or issues
2. To have a formal framework normalization process involves series of test to verify that our designed database is accurate
3. Here, we analyze the relations based on their keys & different kind of dependencies among their attributes
4. To reduce the need for restructing the relations as new data types are introduced
5. To obtain powerful relational retrieval algorithms

Main point: The objective of normalization is to assure that our designed database is correct & we can start building a system over it

Normal Forms

Fifth NF
Fourth NF
Third NF
Second NF
First NF

First NF	Second NF	Third NF	Fourth NF	Fifth NF
Eliminate repeating groups	Non-key attribute fully functional dependence on primary key	Eliminate transitive dependence on primary key	Remove multi value dependency	Project join dependency

Concept of Functional Dependence (FD)

→ In a relation R having 2 attributes X and Y. If for each value of X there is only one value of Y, then Y is called functionally dependent on X.

→ Represented as $X \longrightarrow Y$

→ ⇒ X is determinant, Y is determined.

→ X functionally determines Y

→ Y is functionally determined by X

☆ For each value of determinant there should be associated one and only one value of determined

Eg

Rno.	Name	Class
1	Reena	Btech
2	Meena	Mtech
3	Teena	BA
4	Reena	BSc
5	Sabina	Btech
6	Heena	Mtech

Rno → Name ✓ (For each value of Rno, there is only one name
Rno is unique
For Rno 1, there is only one name Reena)

Rno → Class ✓

Name → Rno ✗ for name Reena, we have 2 Roll no. 1 & 4

Class → Rno ✗ For class btech, 2 Roll no. are there 1 & 5
For Mtech, 2 Roll no. are there 2 & 6

Eg

A	B
1	1
2	4
3	9
4	16
2	4
7	9

Ans =

A → B ✓ for each value of A we are getting one value of B
Value 2 is repeated but B has same value 4, 4

B → A ✗ For value of 9, value of A = 3, 7 ? different

eg

A	B
1	1
2	4
3	9
4	16
3	10

$A \not\rightarrow B$ X

$B \rightarrow A$ ✓

unique

- If we have unique value of X, then Y will functionally determine Y - ie $X \rightarrow Y$
- If there is repeated value of X, then check for each value of Y correspondingly. If for repeated value of X, Y has same value, then $X \rightarrow Y$
- If Y has different values for same repeated value of X, then $X \not\rightarrow Y$

eg

Primary key

SNo.	SName	City	Status
S1	Raj	Delhi	10
S2	Mahesh	Bombay	20
S3	Ramesh	Chennai	20
S4	Raj	Bombay	10

Ans:

$SNo \rightarrow Sname$ $\quad Sname \not\rightarrow Sno$

$SNo \rightarrow City$ $\quad City \not\rightarrow Sno$

$Sno \rightarrow Status$ $\quad Status \not\rightarrow Sno$

- Every primary key of relation functionally determines all other non-key attributes of the relation

Concept of Fully functional Dependence (FFD)

→ Attribute Y is FFD on attribute X, if Y is functionally dependent on X but not functionally dependent on any proper subset of X

$(X_1, X_2) \longrightarrow Y$

X (composite attribute)

$X_1 \not\rightarrow Y$

$X_2 \not\rightarrow Y$

Primary key

eg #

SNo.	PNo.	Qty
S1	P1	100
S1	P2	120
S2	P1	100
S2	P2	130

SNo, PNo ⟶ Qty

Sno ↛ Qty

Pno ↛ Qty

⇒ Qty is fully functional dependent on combination of Sno and Pno.

Q) In eg on previous page of Sno, Sname - -
Is Sno, Sname ⟶ City FFD?

Ans =

Sno, Sname → City

Sno → City (For Raj, 2 cities: Delhi, Bombay)

Sname ↛ City

So Sname, Sno → city Not FFD