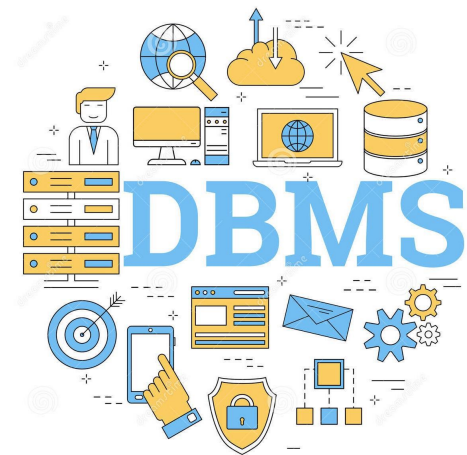# Unit III – Database Design

Dependencies and Normal forms - Functional Dependencies, Armstrong's axioms for FD's, closure of a set of FD's, minimal covers-Non- loss decomposition-First, Second, Third Normal Forms, Dependency Preservation-Boyce/Codd Normal Form-Multivalued Dependencies and Fourth Normal Form- Join Dependencies and Fifth Normal Form

Dependencies

Dependencies in DBMS is a relation between two or more attributes.

It has the following types in DBMS

- Functional Dependency

- Fully-Functional Dependency

- Transitive Dependency

- Multivalued Dependency

- Partial Dependency

Normal Forms

- **Normalization** is the process of minimizing **redundancy** from a relation or set of relations.

- Redundancy in relation may cause <mark>insertion, deletion, and update anomalies.</mark>

- So, it helps to minimize the redundancy in relations.

- **Normal forms** are used to eliminate or reduce redundancy in database tables.

First Normal Form

Second Normal Form

Third Normal Form

Boyce Codd Normal Form

Fourth Normal Form

Fifth Normal Form

	1NF	2NF	3NF	4NF	5NF
Decomposition of Relation	R	R_{11}	R_{21}	R_{31}	R_{41}
		R_{12}	R_{22}	R_{32}	R_{42}
			R_{23}	R_{33}	R_{43}
				R_{34}	R_{44}
					R_{45}
Conditions	Eliminate Repeating Groups	Eliminate Partial Functional Dependency	Eliminate Transitive Dependency	Eliminate Multi-values Dependency	Eliminate Join Dependency

Functional Dependencies

- A functional dependency is a <mark>constraint that specifies the relationship between two sets of attributes</mark>
  - where one set can accurately determine the value of other sets.

- It is denoted as **X → Y**,

- where X is a set of attributes that is capable of determining the value of Y.

- The attribute set on the left side of the arrow, **X** is called **Determinant**, while on the right side, **Y** is called the **Dependent**.

Example : R(A,B,C,D)
{
A -> B → Determinant
B -> C
C -> D → Dependent
D -> A
}
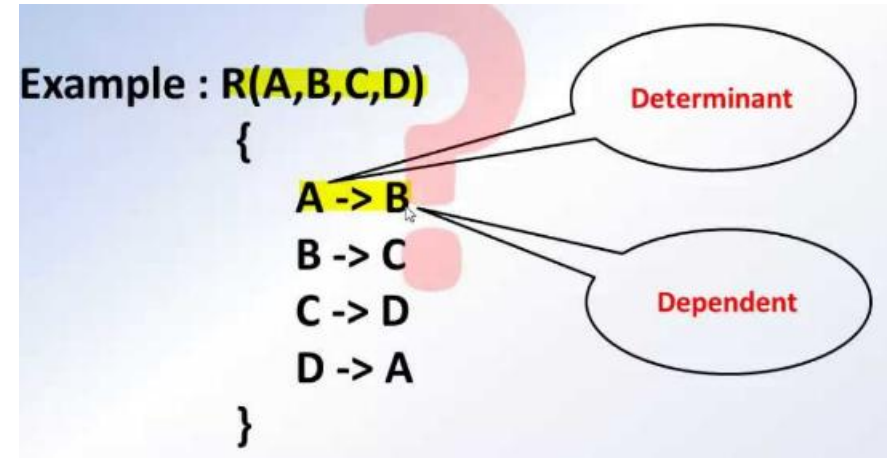
Example 1

Roll_no	Name	Marks	Dept	Course
1	A	78	CS	C1
2	B	60	EE	C1
3	A	78	CS	C2
4	B	60	EE	C3
5	C	80	IT	C3
6	d	80	EC	C2

Roll_no	Name	✓
Name	Roll_no	✗
Roll_no	marks	✓
Dept	Course	✗
Course	Dept	✗
Roll_no,Name	Marks	✓
Name	Marks	✗
Name, Marks	Dept	✓
Name, Marks	Dept, Course	✗
Roll_no	Name, marks	✓
Dept, Course	Name	✓
Roll_no,Marks	Dept	✓
Name	Course	✗
Name,Marks, Dept	Roll_no	✗

Example 2

roll_no	name	dept_name	dept_building
42	abc	CO	A4
43	pqr	IT	A3
44	xyz	CO	A4
45	xyz	IT	A3
46	mno	EC	B2
47	jkl	ME	B2

Valid Functional Dependencies

- roll_no → { name, dept_name, dept_building },→ Here, roll_no can determine values of fields name, dept_name and dept_building, hence a valid Functional dependency

- roll_no → dept_name , Since, roll_no can determine whole set of {name, dept_name, dept_building}, it can determine its subset dept_name also.

- dept_name → dept_building , Dept_name can identify the dept_building accurately, since departments with different dept_name will also have a different dept_building

- More valid functional dependencies: roll_no → name, {roll_no, name} ⋯→ {dept_name, dept_building}, etc.

invalid functional dependencies

- name → dept_name   Students with the same name can have different dept_name, hence this is not a valid functional dependency.

- dept_building → dept_name   There can be multiple departments in the same building, For example, in the above table departments ME and EC are in the same building B2, hence dept_building → dept_name is an invalid functional dependency.

- More invalid functional dependencies: name → roll_no, {name, dept_name} → roll_no, dept_building → roll_no, etc.

Types of Functional Dependencies

1. Trivial functional dependency

2. Non-Trivial functional dependency

3. Multivalued functional dependency

4. Transitive functional dependency

Trivial functional dependency

- A dependent is always a subset of the determinant. i.e. If X → Y and Y is the subset of X, then it is called trivial functional dependency.

- A→B is trivial functional dependency if B is a subset of A.

- The following dependencies are also trivial: A→A & B→B

Example 1 :

- ABC -> AB

- ABC -> A

- ABC -> ABC

Non - Trivial functional dependency

The dependent is strictly not a subset of the determinant. i.e. If **X →
Y** and **Y is not a subset of X**, then it is called Non-trivial functional
dependency.

Example 1 :

- Id -> Name

- Name -> DOB

Multivalued functional dependency

Entities of the dependent set are not dependent on each other. i.e. If a → {b, c} and there exists no functional dependency between b and c, then it is called a multivalued functional dependency.

bike_model	manuf_year	color
tu1001	2007	Black
tu1001	2007	Red
tu2012	2008	Black
tu2012	2008	Red
tu2222	2009	Black
tu2222	2009	Red

Transitive functional dependency

- Dependent is indirectly dependent on determinant. i.e. If a → b & b → c, then according to axiom of transitivity, a → c. This is a transitive functional dependency.

enrol_no → dept and dept → building_no. Hence, according to the axiom of transitivity, enrol_no → building_no is a valid functional dependency. This is an indirect functional dependency, hence called Transitive functional dependency.

enrol_no	name	dept	building_no
42	abc	CO	4
43	pqr	EC	2
44	xyz	IT	1
45	abc	EC	2