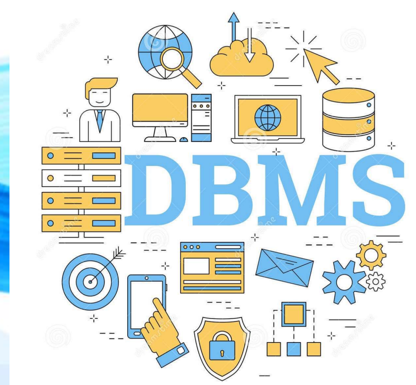
# Unit III – Database Design

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



# Types of Functional Dependencies

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

1.Trivial functional dependency

2.Non-Trivial functional dependency

3.Multivalued functional dependency

4.Transitive functional dependency

# Trivial functional dependency

- In **Trivial Functional Dependency**, a dependent is always a subset of the determinant.

  If **$X \rightarrow Y$** and **Y is the subset of X**, then it is called trivial functional dependency

# Trivial functional dependency

- Example

| roll_no | name | age |
|---|---|---|
| 42 | abc | 17 |
| 43 | pqr | 18 |
| 44 | xyz | 18 |

**{roll_no, name} → name** is a trivial functional dependency, since the dependent **name** is a subset of determinant set **{roll_no, name}**

Similarly, **roll_no → roll_no** is also an example of trivial functional dependency.

# Non Trivial functional dependency

- In **Non-trivial functional dependency**, the dependent is strictly not a subset of the determinant.

  If **X → Y** and **Y is not a subset of X**, then it is called Non-trivial functional dependency.

# Non Trivial functional dependency

- Example

| roll_no | name | age |
|---|---|---|
| 42 | abc | 17 |
| 43 | pqr | 18 |
| 44 | xyz | 18 |

**roll_no → name** is a non-trivial functional dependency, since the dependent **name** is **not a subset of** determinant **roll_no**

Similarly, **{roll_no, name} → age** is also a non-trivial functional dependency, since **age** is **not a subset of {roll_no, name}**

# Multivalued functional dependency

- **Multivalued functional dependency**, entities of the dependent set are **not dependent on each other.**

- If **a → {b, c}** and there exists **no functional dependency** between **b and c**, then it is called a **multivalued functional dependency.**

# Multivalued functional dependency

- Example

| roll_no | name | age |
|---|---|---|
| 42 | abc | 17 |
| 43 | pqr | 18 |
| 44 | xyz | 18 |
| 45 | abc | 19 |

**roll_no → {name, age}** is a multivalued functional dependency, since the dependents **name** & **age** are **not dependent** on each other(i.e. **name → age** or **age → name doesn't exist !**)

# Transitive Functional dependency

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

# Transitive functional dependency

Example

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

**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.

# Armstrong's axioms for FD's

**Primary**

- ✓Axiom of reflexivity
- ✓Axiom of augmentation
- ✓Axiom of transitivity

**Secondary**

- ✓Union
- ✓Composition
- ✓Decomposition
- ✓Pseudo Transitivity

# Axiom of reflexivity

- if Y is a subset of X, then X determines Y.
- If $X \supseteq Y$ then $X \rightarrow Y$
- $X \rightarrow X$

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |

# Axiom of Transitivity

- if X determines Y and Y determine Z, then X must also determine Z.

- **If X → Y and Y → Z**

  **then X → Z**

Ram is sibling of Sham
Sham is sibling of Mohan

↓

Ram is sibling of Mohan

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |

**Name → Marks and Marks → Dept then**

**name → Marks**

# Axiom of Augmentation

- The augmentation is also called as a partial dependency.
- In augmentation, if X determines Y, then XZ determines YZ for any Z.
- If X → Y then XZ → YZ

**R.No → Name then**

**R.No, Marks → Name, marks**

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |

# Axiom of Union

- if X determines Y and X determines Z, then X must also determine Y and Z.
- If X → Y and X → Z then X → YZ

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |

**R.No → Name and R.No → Marks**

**R.No → Name, marks**

# Axiom of Decomposition / Splitting

- Decomposition rule is also known as project rule. It is the reverse of union rule.
- if X determines Y and Z, then X determines Y and X determines Z separately.
- If X → YZ Then X → Y then X → Z

**Name, Marks→ Dept, Course then**

**Name, Marks→ Dept and**

**Name, Marks→ Dept, Course**

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |

# Axiom of Pseudo Transitivity

- if X determines Y and YZ determines W, then XZ determines W.
- If X → Y and YZ → W then XZ → W

**Roll No → Name, and**

**Name, Marks→ Dept**

**Then**

**Roll No, Marks→ Dept**

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |

# Axiom of Composition

- If X → Y and A → B then XA → YB

**Roll No → Name, and**

**Marks→ Dept**

**Then**

**Roll No, Marks→ Name, Dept**

| R.No | Name | Marks | Dept | Course |
|---|---|---|---|---|
| 1 | A | 78 | CS | C1 |
| 2 | B | 60 | EE | C1 |
| 3 | A | 78 | CE | C2 |
| 4 | B | 60 | EE | C3 |
| 5 | C | 80 | IT | C2 |


Thank You!