UNIT –III

RELATIONAL DATABASE DESIGN AND NORMALIZATION

1. Functional Dependency Inference Rules

Inference rules are a set of rules used to **derive all possible functional dependencies** from a given set of FDs.
These rules form the basis of **Armstrong's Axioms** and help in determining closure, keys, and normalization.

1.1 Armstrong's Axioms (Basic Inference Rules)

Rule 1: Reflexivity

If **Y is a subset of X**, then:

```
X → Y
```

Example:

```
{RollNo, Name} → Name
```

Rule 2: Augmentation

If **X → Y**, then adding the same attribute Z to both sides keeps it valid:

```
XZ → YZ
```

Example:

```
RollNo → Name  implies RollNo, CourseID → Name, CourseID
```

Rule 3: Transitivity

If **X → Y** and **Y → Z**, then:

```
X → Z
```

Example:

```
RollNo → Dept
```

```
Dept → HOD
Therefore: RollNo → HOD
```

1.2 Additional Inference Rules (Derived Rules)

Rule 4: Union

If

```
X → Y   and   X → Z
```

then

```
X → YZ
```

Rule 5: Decomposition

If

```
X → YZ
```

then

```
X → Y   and   X → Z
```

Rule 6: Pseudotransitivity

If

```
X → Y   and   WY → Z
```

then

```
WX → Z
```

These inference rules are essential for computing closure of attributes, finding keys, and simplifying FDs during normalization.

---

2. Minimal Cover (Canonical Cover)

A **minimal cover** is a simplified set of functional dependencies that is **equivalent** to the original set, but contains **no redundancy**.

It is useful for **decomposition**, **normalization**, and finding **keys**.

Properties of a Minimal Cover

A minimal cover must satisfy:

1. **Right side is single-valued**
   - Each FD must be of the form
   - `X → A`

     where **A** is a single attribute.

2. **No extraneous attributes on LHS**
   - Each attribute on left side is essential.
3. **No redundant dependencies**
   - Removing any FD should change the closure.

Steps to Compute Minimal Cover

Let **F** be the original set of FDs.

Step 1: Split RHS

Convert FDs so that the right-hand side has only one attribute.

Example:

```
A → BC  becomes
A → B
A → C
```

Step 2: Remove extraneous attributes from LHS

For each FD **X → A**:

- For each attribute **x in X**,
  check if **(X – x) → A** can still be derived using F.
- If yes, remove **x**.

Example:

```
AB → C
Check if B → C holds?
If yes → minimal LHS is B → C
```

Step 3: Remove redundant FDs

For each FD in F:

- Temporarily remove it.
- Compute closure of remaining FDs.
- If the removed FD can still be derived → it is redundant.

Example: Minimal Cover

Given:

```
F = { A → BC,  B → D,  A → D }
```

Step 1: Split RHS

```
A → B
A → C
B → D
A → D
```

Step 2: Remove extraneous attributes

None of the LHS have more than one attribute → nothing to remove.

Step 3: Remove redundant dependencies

Check redundancy:

- **Is A → D** redundant?
  Yes, because A → B and B → D gives A → D.
  Remove A → D.

Final minimal cover:

```
{ A → B, A → C, B → D }
```

Importance of Minimal Cover

- Helps in **3NF and BCNF decomposition**
- Removes redundancy in FDs
- Ensures minimal and efficient database schema
- Helps in computing attribute closure and candidate keys