Functional Dependencies

1. Introduction:

- In a relational database, functional dependencies (FDs) are crucial for understanding the relationships between attributes within a table.

- A functional dependency between two attribute sets, X and Y, denoted as X → Y, indicates that the values of X uniquely determine the values of Y.

2. Armstrong's Axioms:

Armstrong's axioms are fundamental rules used for reasoning about functional dependencies. These axioms guide us in deriving and understanding the relationships between attributes.

a. Reflexivity Axiom:

- If Y is a subset of X, then X → Y.

- This axiom reflects the idea that any subset of attributes is functionally dependent on the whole set of attributes.

Example:

Consider a relation R with attributes A, B, and C. If A → B holds, then it's also true that AC → B.

b. Augmentation Axiom:

- If X → Y, then XZ → YZ for any attribute set Z.

- This axiom shows that adding attributes to both sides of a functional dependency maintains its validity.

Example:

If Name → Age, then NameAddress → AgeAddress holds true.

c. Transitivity Axiom:

- If X → Y and Y → Z, then X → Z.

- This axiom implies that if a functional dependency can be derived indirectly, it can be inferred directly.

Example:

If Course → Department and Department → Faculty, then Course → Faculty can be inferred.

3. Example Scenarios:

a. Student Table:

```
| Roll No | Name    | Age | Course |
|---------|---------|-----|--------|
| 101     | Alice   | 20  | CS     |
| 102     | Bob     | 22  | ECE    |
| 103     | Carol   | 21  | CS     |
```

- In the above table, Roll No → Name because each Roll No corresponds to a unique student's name.

- Roll No → Age because each student's Roll No uniquely determines their age.

- Course → Roll No because each course maps to multiple Roll Nos.

b. Course Enrollment Table:

```
| Roll No | Course |
|---------|--------|
| 101     | DBMS   |
| 102     | OS     |
| 103     | DBMS   |
```

- In this table, Roll No → Course because each Roll No maps to a specific course.

4. Inference Rules:

a. Union Rule:

- If X → Y and X → Z, then X → YZ.

Example:

If Roll No → Name and Roll No → Age, then Roll No → NameAge.

b. Decomposition Rule:

- If X → YZ, then X → Y and X → Z.

Example:

If Roll No → NameAge, then Roll No → Name and Roll No → Age.

c. Pseudo-Transitivity Rule:

- If X → Y and WY → Z, then WX → Z.

Example:

If Course → Department and CourseFaculty → Office, then CourseFaculty → Office can be inferred as Course → DepartmentOffice.

5. Conclusion:

- Functional dependencies play a pivotal role in maintaining the accuracy and integrity of relational databases.

- Armstrong's axioms provide a systematic approach to understanding and deriving functional dependencies.

- These concepts are essential for database normalization and the efficient design of relational databases.

Types of functional dependency

1. Trivial Functional Dependency

In 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

Example:

roll_no name age

42	abc	17
43	pqr	18
44	xyz	18

Here, {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.

2. Non-trivial Functional Dependency

In 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:

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

Here, 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}

3. Multivalued Functional Dependency

In 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.

For example,

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

Here, 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 !)

4. Transitive Functional Dependency

In 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.

For example,

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

Here, 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.

5. Fully Functional Dependency

In full functional dependency an attribute or a set of attributes uniquely determines another attribute or set of attributes. If a relation R has attributes X, Y, Z with the dependencies X->Y and X->Z which states that those dependencies are fully functional.

6. Partial Functional Dependency

In partial functional dependency a non key attribute depends on a part of the composite key, rather than the whole key. If a relation R has attributes X, Y, Z where X and Y are the composite key and Z is non key attribute. Then X->Z is a partial functional dependency in RBDMS.