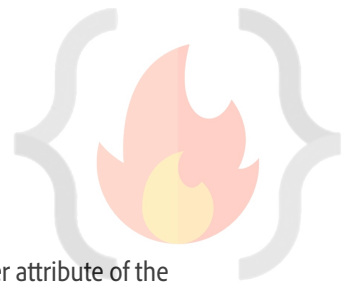


LEC-11: Normalisation



1. **Normalisation** is a step towards DB optimisation.
2. **Functional Dependency (FD)**
 1. It's a relationship between the primary key attribute (usually) of the relation to that of the other attribute of the relation.
 2. $X \rightarrow Y$, the left side of FD is known as a **Determinant**, the right side of the production is known as a **Dependent**.
 3. **Types of FD**
 1. **Trivial FD**
 1. $A \rightarrow B$ has trivial functional dependency if B is a subset of A. $A \rightarrow A$, $B \rightarrow B$ are also Trivial FD.
 2. **Non-trivial FD**
 1. $A \rightarrow B$ has a non-trivial functional dependency if B is not a subset of A. $[A \cap B \text{ is NULL}]$.
 4. **Rules of FD (Armstrong's axioms)**
 1. **Reflexive**
 1. If 'A' is a set of attributes and 'B' is a subset of 'A'. Then, $A \rightarrow B$ holds.
 2. If $A \supseteq B$ then $A \rightarrow B$.
 2. **Augmentation**
 1. If B can be determined from A, then adding an attribute to this functional dependency won't change anything.
 2. If $A \rightarrow B$ holds, then $AX \rightarrow BX$ holds too. 'X' being a set of attributes.
 3. **Transitivity**
 1. If A determines B and B determines C, we can say that A determines C.
 2. if $A \rightarrow B$ and $B \rightarrow C$ then $A \rightarrow C$.
3. **Why Normalisation?**
 1. To avoid redundancy in the DB, not to store redundant data.
4. **What happen if we have redundant data?**
 1. Insertion, deletion and updation anomalies arises.
5. **Anomalies**
 1. Anomalies means abnormalities, there are three types of anomalies introduced by data redundancy.
 2. **Insertion anomaly**
 1. When certain data (attribute) can not be inserted into the DB without the presence of other data.
 3. **Deletion anomaly**
 1. The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.
 4. **Updation anomaly** (or modification anomaly)
 1. The update anomaly is when an update of a single data value requires multiple rows of data to be updated.
 2. Due to updation to many places, may be **Data inconsistency** arises, if one forgets to update the data at all the intended places.
 5. Due to these anomalies, **DB size increases** and **DB performance become very slow**.
 6. To rectify these anomalies and the effect of these of DB, we use **Database optimisation technique** called **NORMALISATION**.
6. **What is Normalisation?**
 1. Normalisation is used to minimise the redundancy from a relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
 2. Normalisation divides the composite attributes into individual attributes OR larger table into smaller and links them using relationships.
 3. The normal form is used to reduce redundancy from the database table.
7. **Types of Normal forms**
 1. **1NF**
 1. Every relation cell must have atomic value.
 2. Relation must not have multi-valued attributes.



2. 2NF

1. Relation must be in 1NF.
2. There should not be any partial dependency.
 1. All non-prime attributes must be fully dependent on PK.
 2. Non prime attribute can not depend on the part of the PK.

3. 3NF

1. Relation must be in 2NF.
2. No transitivity dependency exists.
 1. Non-prime attribute should not find a non-prime attribute.

4. BCNF (Boyce-Codd normal form)

1. Relation must be in 3NF.
2. FD: $A \rightarrow B$, A must be a super key.
 1. We must not derive prime attribute from any prime or non-prime attribute.

8. Advantages of Normalisation

1. Normalisation helps to minimise data redundancy.
2. Greater overall database organisation.
3. Data consistency is maintained in DB.

CodeHelp