Normalization

It is the process of **organizing** the fields and tables of a relational database to reduce redundancy and dependency. Normalization usually involves **dividing large tables** into smaller ones and defining relationships between them.

The normal forms of a relational database theory provide the criteria for **determining** a table’s **degree of immunity** against logical inconsistencies and anomalies. Informally, a relational database is said to be “normalized” if it is in **the Third Normal Form**.

# Normal Forms

First Normal Form: Domain of each attribute contains only atomic values and the value of each attribute contains only a single value from that domain.

1) Eliminate duplicative columns from the same table. 2) Create separate tables for each group of related data.

Second Normal Form: No non-prime attribute in the table is functionally dependent on a subset of any candidate key.

1) Remove subsets of data that apply to multiple rows of the table and place them in a separate file.

Third Normal Form: Every non-prime attribute is non-transitively dependent on every candidate key in the table. No transitive dependency is allowed.

1) Remove columns that are not fully dependent on the primary key.

Elementary Key Normal Form: Every non trivial functional dependency in the table is either the dependency of an elementary key attribute or a dependency on a superkey.

Boyce-Codd Normal Form: Every non-trivial functional dependency in the table is a dependency on a superkey.

Fourth Normal Form: Every non-trivial multivalued dependency in the table is a dependency on a superkey.

Fifth Normal Form: Every non-trivial join dependency in the table is implied by the superkeys of the table.

Domain/Key Normal Form: Every constraint on the table is a logical consequence of the table’s domain constraints and key constraints.

Sixth Normal Form: Tables features no non-trivial dependencies at all.