

Functional Dependencies

A functional dependency is an association between two attributes of the same relational database table. One of the attributes is called the determinant and the other attribute is called the determined.

Types of Functional Dependencies

- 1. Trivial Functional Dependency:** Some functional dependencies are said to be trivial because they are satisfied by all relations.
- 2. Non-Trivial Functional Dependency:** Non Trivial Functional Dependency can be categorized into –
 - Complete Non Trivial Functional Dependency – A Functional Dependency is completely non-trivial if none of the RHS attributes are part of the LHS attributes.
 - Semi Non-Trivial Functional Dependencies – A Functional Dependency is semi non-trivial if one of the RHS attributes are not part of the LHS attributes.
- 3. Multivalued dependency:** Multivalued dependency occurs when there are more than one independent multivalued attributes in a table.
- 4. Transitive dependency:** A functional dependency is said to be transitive if it is indirectly formed by two functional dependencies.

Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Normalization is divided into following normal forms:

- 1. First Normal Form (1NF):** A table is said to be in First Normal Form (1NF) if and only if each attribute of the relation is atomic.
- 2. Second Normal Form (2NF):** A table is said to be in 2NF if it is in 1NF and no non-prime attribute is dependent on the proper subset of any candidate key of table.
- 3. Third Normal Form (3NF):** A table is said to be in 3NF if it is in 2NF and no non-prime attribute is transitively dependent on prime key attribute.
- 4. Boyce-Codd Normal Form (3.5NF):** A table is said to be in BCNF, if it is in 3NF and its candidate keys do not have any partial dependency on the other attributes.
- 5. Forth Normal Form (4NF):** A table is said to be in 4NF, if it is in 3NF and cannot have multi-valued dependencies on a Primary Key.
- 6. Fifth Normal Form (5NF):** A table is said to be in 5NF, if it is in 4NF and composite key shouldn't have any cyclic dependencies.

Database Decomposition

- Decomposition is the process of breaking down in parts or elements.
- It replaces a relation with a collection of smaller relations.
- It breaks the table into multiple tables in a database.

Properties of Decomposition

Following are the properties of Decomposition:

- 1. Lossless Decomposition:** Decomposition must be lossless. It means that the information should not get lost from the relation that is decomposed.
- 2. Dependency Preservation:** This decomposition property can only be done by maintaining the functional dependency. When an update is made to the database, the system should be able to check that the update will not create an illegal relation.
- 3. Lack of Data Redundancy:** Lack of Data Redundancy is also known as a Repetition of Information. The proper decomposition should not suffer from any data redundancy.

Transaction

Transaction is an action, or series of actions that are being performed by a single user or application program, which reads or updates the contents of the database.

Types of transaction

Read only transaction: If the database operations in a transaction do not update the database but only retrieve data.

Read write transaction: If the database operation in a transaction retrieves as well as update the database.

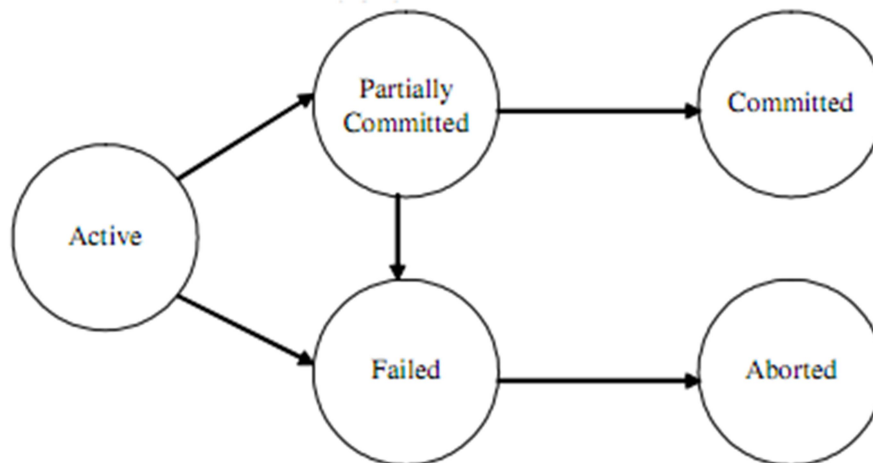
Transaction Properties

A transaction must have the following four properties, called ACID properties, to ensure that a database remains stable state after the transaction is executed:

- **Atomicity:** This property states that each transaction must be considered as a single unit and must be completed fully or not completed at all.
- **Consistency:** The database must remain in a consistent state after any transaction.

- **Isolation:** Isolation property of a transaction means that the data used during the execution of a transaction cannot be used by a second transaction until the first one is completed.
- **Durability:** The database should be strong enough to handle any system failure.

States of Transactions



Active: In this state, the transaction is being executed. This is the initial state of every transaction.

Partially-Committed: When a transaction executes its final operation, it is said to be in a partially committed state.

Committed: If a transaction executes all its operations successfully, it is said to be committed.

Failed: If a transaction cannot proceed to the execution state because of the failure of the system or database, then the transaction is said to be in failed state.

Aborted: If a transaction is failed to execute, then the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction.