

1 ACID Properties of Transactions

There are Four **ACID** properties of a Transaction :

Atomicity A transaction is an indivisible unit . The recovery subsystem is responsible for atomicity .

Consistency It is the responsibility of the DBMS and application developer so that each transaction must transform the database from one consistent state to another consistent state .

Isolation It is the responsibility of the concurrency control sub-system to ensure isolation where, each transaction can execute independently so that partial effects of each transaction should not be visible to other transactions .

Durability It is the responsibility of the recovery manager to record permanently the effects of a successfully completed transaction and must not be lost due to subsequent failure .

2 Entity Relationship Diagram

Entity Relationship Diagram (ER Diagram) can express the overall logical structure of a database graphically .

ER Diagrams consists of following major components :

Rectangles	Entities
Ellipse	Attributes
Diamonds	Relationships
Lines	Links
Double Ellipse	Multi-Valued Attributes
Dashed Ellipse	Derived Attributes
Double Lines	Total participations
Double Rectangles	Weak Entity-set

3 DBMS Keys

3.1 What are DBMS keys?

DBMS key is an attribute or a set of attributes which help you uniquely identify a record or a row of data in a relation (table)

3.2 Why we need DBMS keys?

- for identifying any row of data in a table uniquely
- we can **force identify of data** and **ensure integrity of data** is maintained
- to establish relationship between tables and identifying relationship between tables

3.3 Types of DBMS Keys :

Super Key A super key is a set of one or more columns (attributes) to uniquely identify rows in a table.

Candidate Key A super key with no redundant attribute is known as candidate key

Primary Key A primary is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.

Foreign Key Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

Composite / Compound Key A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.

Alternate Key Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.

Surrogate Key A surrogate key is a unique identifier used in databases for a modeled entity or an object. It is a unique key whose only significance is to act as the primary identifier of an object or entity and is not derived from any other data in the database and may or may not be used as the primary key, like :

- *user_id* in users table
- *product_id* in product table
- *basket_id* in basket table

A surrogate key has the following characteristics:

- the value is unique system-wide, hence never reused
- the value is system generated
- the value is not manipulable by the user or application
- the value contains no semantic meaning
- the value is not visible to the user or application
- the value is not composed of several values from different domains

4 Basic Concept of Database Normalization

4.1 What is Normalization ?

Normalization is a technique of organizing the data into multiple related tables and to minimize **Data Redundancy** .

4.2 What is Data Redundancy ?

Data Redundancy is repetition of similar data at multiple places .

4.3 Why Data Redundancy is Bad ?

- Repetition of data increases the size of database .
- Other issues like :
 - Insertion Problems
 - Deletion Problems
 - Updation Problems

4.4 Issues due to data redundancy :

- Insertion Anomaly
- Deletion Anomaly
- Updation Anomaly

4.5 What is Insertion Anomaly ?

Insert repeated data for each new row is called insertion anomaly .

4.6 What is Deletion Anomaly ?

Loss of related data when some other data is target to delete .

5 1st Normal Form

5.1 How to achieve 1st Normal Form ?

There are 4 basic rules that a table should follow, to be in 1st Normal Form :

Rule 1 : each column should contain atomic values .

Rule 2 : a column should contain values that are of the same type .

Rule 3 : each column should have a unique name .

Rule 4 : order in which data is saved doesn't matter .

6 2nd Normal Form

For a table to be in the second Normal Form :

- It should be in 1st Normal Form .
- And, It should not have any **Partial Dependency** .

7 3rd Normal Form

For a table to be in 3rd Normal Form :

- It should be in 2nd Normal Form .
- And it should not have **Transitive Dependency** .

8 Different Kind of Dependencies :

8.1 What is Partial Dependency ?

- A is part of the Candidate Key
- B is non prime attribute

$$A \rightarrow B$$

Or

$$part - of - candidate - key \rightarrow non - prime - attribute$$

* Means B depends on parts of Candidate Key rather than depending on the entire Candidate Key, This is **Partial Dependency** .

8.2 What is Transitive Dependency ?

- A is non prime attribute
- B is non prime attribute

$$A \rightarrow B$$

Or

$$non - prime - attribute \rightarrow non - prime - attribute$$

9 3.5 Normal Form

Boyce-Codd Normal Form (BCNF)

For a table to be in 3.5 Normal Form :

- It should be in 3rd Normal Form
- For any dependency $A \rightarrow B$, A should be a Super Key .

in other word :

For a table to be in 3.5 Normal Form :

- It should be in 3rd Normal Form
- For any dependency $A \rightarrow B$,
we can not have :

$$\begin{aligned} non - prime - attribute &\rightarrow non - prime - attribute \\ &And \\ non - prime - attribute &\rightarrow prime - attribute \end{aligned}$$

but only we can :

$$prime - attribute \rightarrow \begin{cases} prime - attribute \\ non - prime - attribute \end{cases}$$

10 Summary

10.1 Functional Dependency

$$prime - attribute \rightarrow non - prime - attribute$$

10.2 Partial Dependency

$$part - of - candidate - key \rightarrow non - prime - attribute$$

10.3 Transitive Dependency

$$non - prime - attribute \rightarrow non - prime - attribute$$

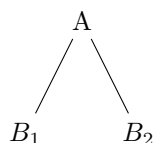
10.4 4th Normal Form

For a table to be in 4th Normal Form :

- It should satisfy BCNF .
- It should not have Multi-Valued Dependency .

10.5 What is Multi-Valued Dependency ?

$A \twoheadrightarrow B$, is Multi-Valued Dependency, if, for a single value of A we have more than one value of B :



10.6 We have Three Conditions for Multi-Valued Dependency

- $A \twoheadrightarrow B$, for a single value of A , more than one value of B exist .
- table should have at least 3 columns .
- for this table with A, B, C columns, B and C should be independent .