2

Overview Of Data Modelling

Objectives

After completing this lesson, you should be able to:

- Overview of Data Model
- Stages of Data Model
- ACID in DBMS

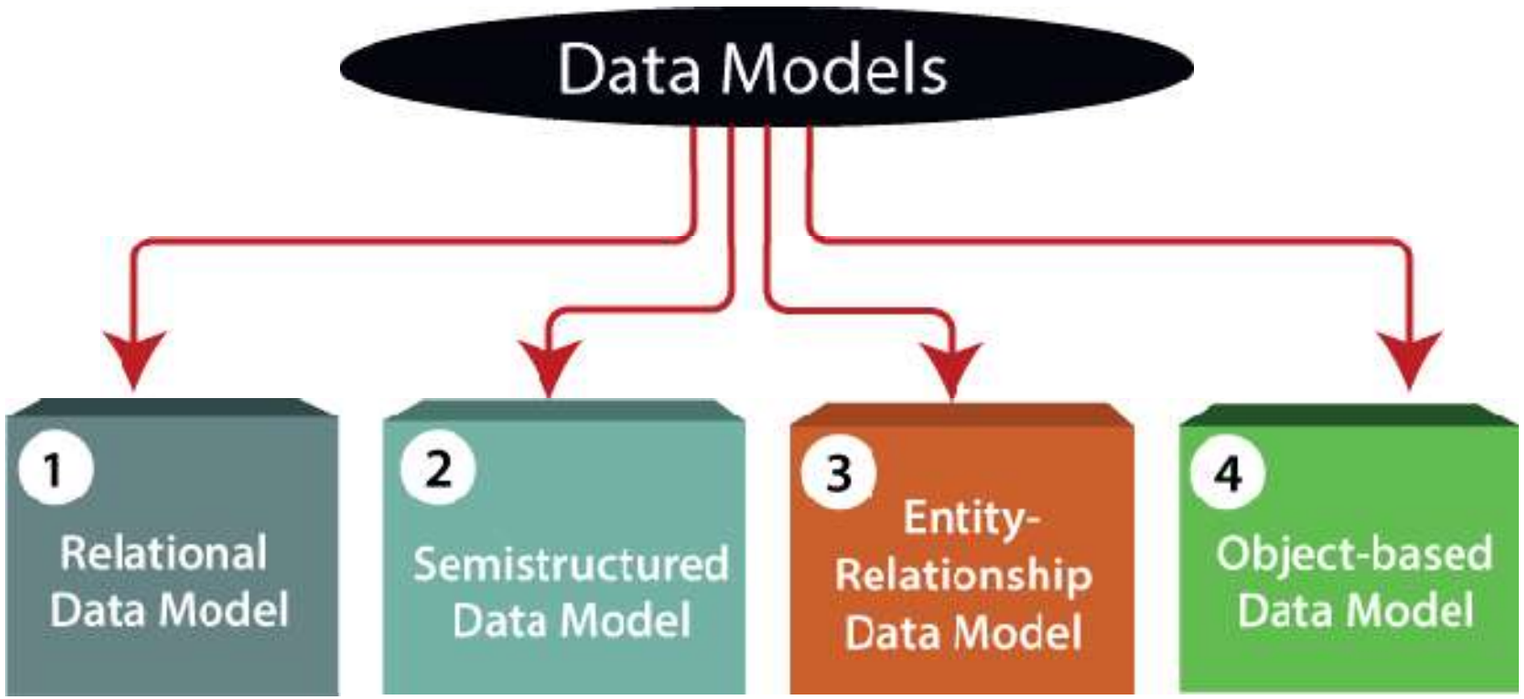
Course Roadmap

RDBMS Concepts

- Lesson 1: Introduction to Databases
- **Lesson 2: Overview of Data Model** — You are here!
- Lesson 3: Exploration of Entity Attribute Relationships and Notations
- Lesson 4: Normalization and De-Normalization

What is Data Modelling ?

- **Data modeling (data modelling)** is the process of creating a data model for the data to be stored in a database. This data model is a conceptual representation of Data objects, the associations between different data objects, and the rules.
- Data modeling helps in the visual representation of data and enforces business rules, regulatory compliances, and government policies on the data. Data Models ensure consistency in naming conventions, default values, semantics, security while ensuring quality of the data.
- It provides the conceptual tools for describing the design of a database at each level of data abstraction.


Data Models
1
Relational Data Model
2
Semistructured Data Model
3
Entity-Relationship Data Model
4
Object-based Data Model

1) Relational Data Model:

- This type of model designs the data in the form of rows and columns within a table.
- Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations.
- This model was initially described by Edgar F. Codd, in 1969.
- The relational data model is the widely used model which is primarily used by commercial data processing applications.

2) Entity-Relationship Data Model:

- An ER model is the logical representation of data as objects and relationships among them.
- These objects are known as entities, and relationship is an association among these entities.
- This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing.
- A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity.
- A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.

3) Object-based Data Model:

- An extension of the ER model with notions of functions, encapsulation, and object identity, as well.
- This model supports a rich type system that includes structured and collection types.
- Thus, in 1980s, various database systems following the object-oriented approach were developed. Here, the objects are nothing but the data carrying its properties.

4) Semistructured Data Model:

- This type of data model is different from the other three data models
- The semistructured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets.
- The Extensible Markup Language, also known as XML, is widely used for representing the semistructured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

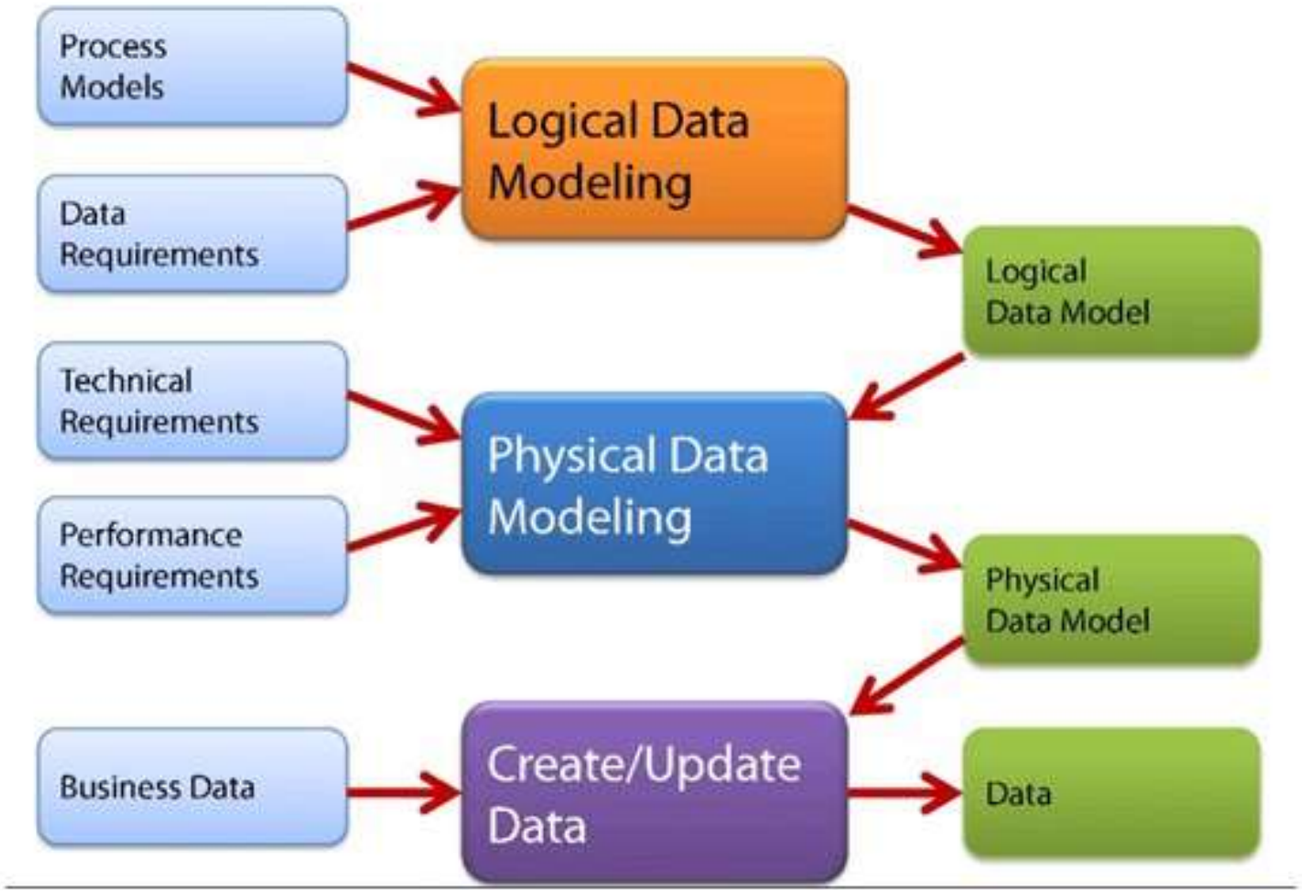
Why use Data Model?

1. Ensures that all data objects required by the database are accurately represented. Omission of data will lead to creation of faulty reports and produce incorrect results.
2. A data model helps design the database at the conceptual, physical and logical levels.
3. Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.
4. It provides a clear picture of the base data and can be used by database developers to create a physical database.
5. It is also helpful to identify missing and redundant data.
6. Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

Stages of Data Model

Stages of Data Model

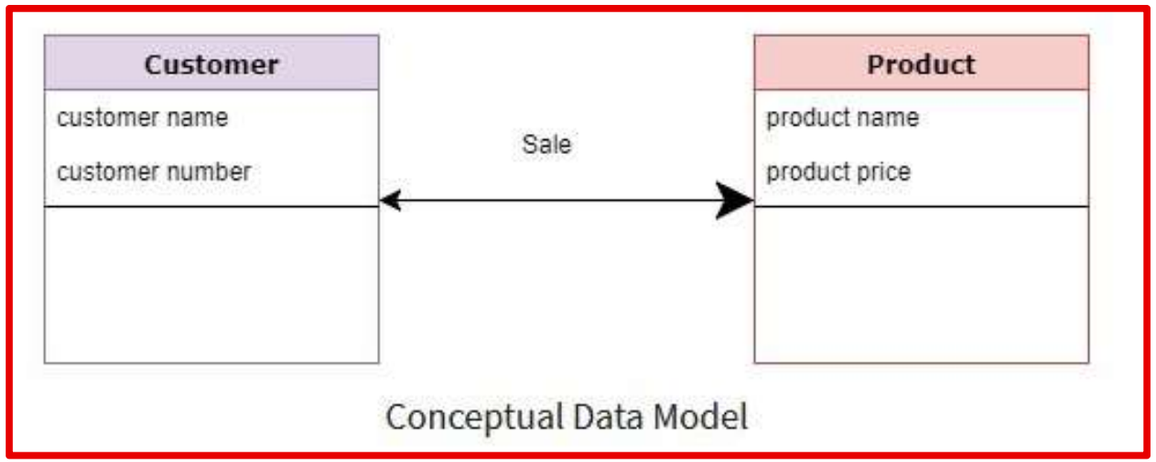
1. **Conceptual Data Model:** This Data Model defines **WHAT** the system contains. This model is typically created by Business stakeholders and Data Architects. The purpose is to organize, scope and define business concepts and rules.
2. **Logical Data Model:** Defines **HOW** the system should be implemented regardless of the DBMS. This model is typically created by Data Architects and Business Analysts. The purpose is to developed technical map of rules and data structures.
3. **Physical Data Model**: This Data Model describes **HOW** the system will be implemented using a specific DBMS system. This model is typically created by DBA and developers. The purpose is actual implementation of the database.


Process Models
Data Requirements
Technical Requirements
Performance Requirements
Business Data
Logical Data Modeling
Physical Data Modeling
Create/Update Data
Logical Data Model
Physical Data Model
Data

Conceptual Data Model

- A **Conceptual Data Model** is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships. In this data modeling level, there is hardly any detail available on the actual database structure. Business stakeholders and data architects typically create a conceptual data model.
- The 3 basic tenants of Conceptual Data Model are
  1. **Entity**: A real-world thing
  2. **Attribute**: Characteristics or properties of an entity
  3. **Relationship**: Dependency or association between two entities

Example and Usage

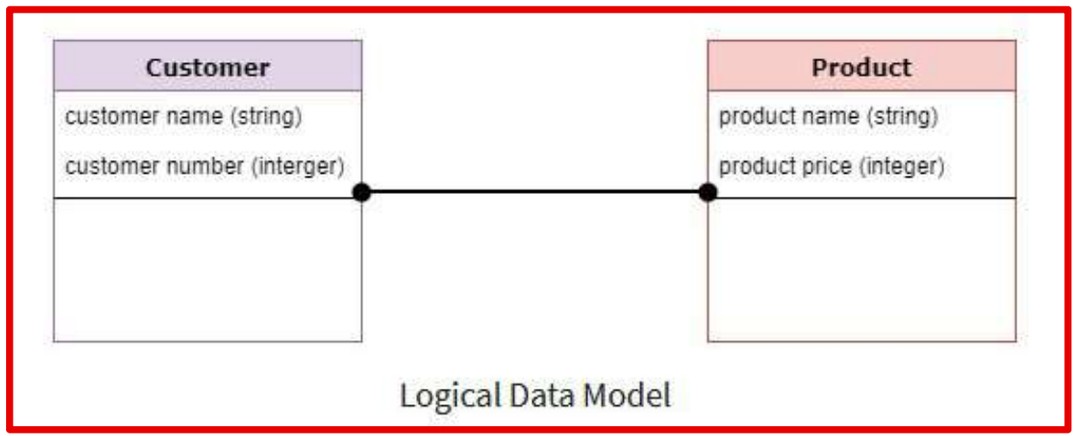


Characteristics of a conceptual data model

- Offers Organisation-wide coverage of the business concepts.
- This type of Data Models are designed and developed for a business audience.
- The conceptual model is developed independently of hardware specifications like data storage capacity, location or software specifications like DBMS vendor and technology. The focus is to represent data as a user will see it in the “real world.”

Logical Data Model

- The **Logical Data Model** is used to define the structure of data elements and to set relationships between them.
- The logical data model adds further information to the conceptual data model elements.
- The advantage of using a Logical data model is to provide a foundation to form the base for the Physical model. However, the modeling structure remains generic.



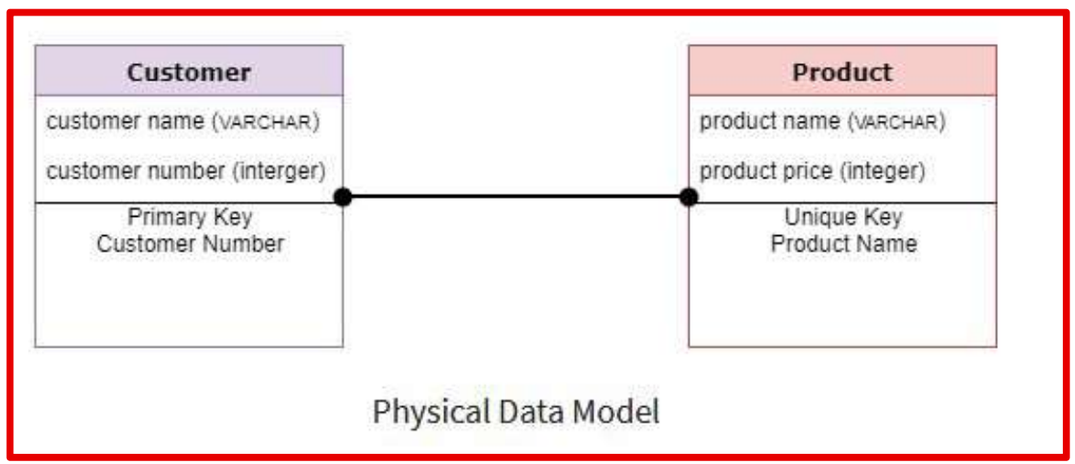
Logical Data Model

Characteristics of a Logical data model

- Describes data needs for a single project but could integrate with other logical data models based on the scope of the project.
- Designed and developed independently from the DBMS.
- Data attributes will have datatypes with exact precisions and length.
- Normalization processes to the model is applied typically till 3NF.

Physical Data Model

- A **Physical Data Model** describes a database-specific implementation of the data model.
- It offers database abstraction and helps generate the schema. This is because of the richness of meta-data offered by a Physical Data Model.
- The physical data model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers, and other RDBMS features.



Physical Data Model

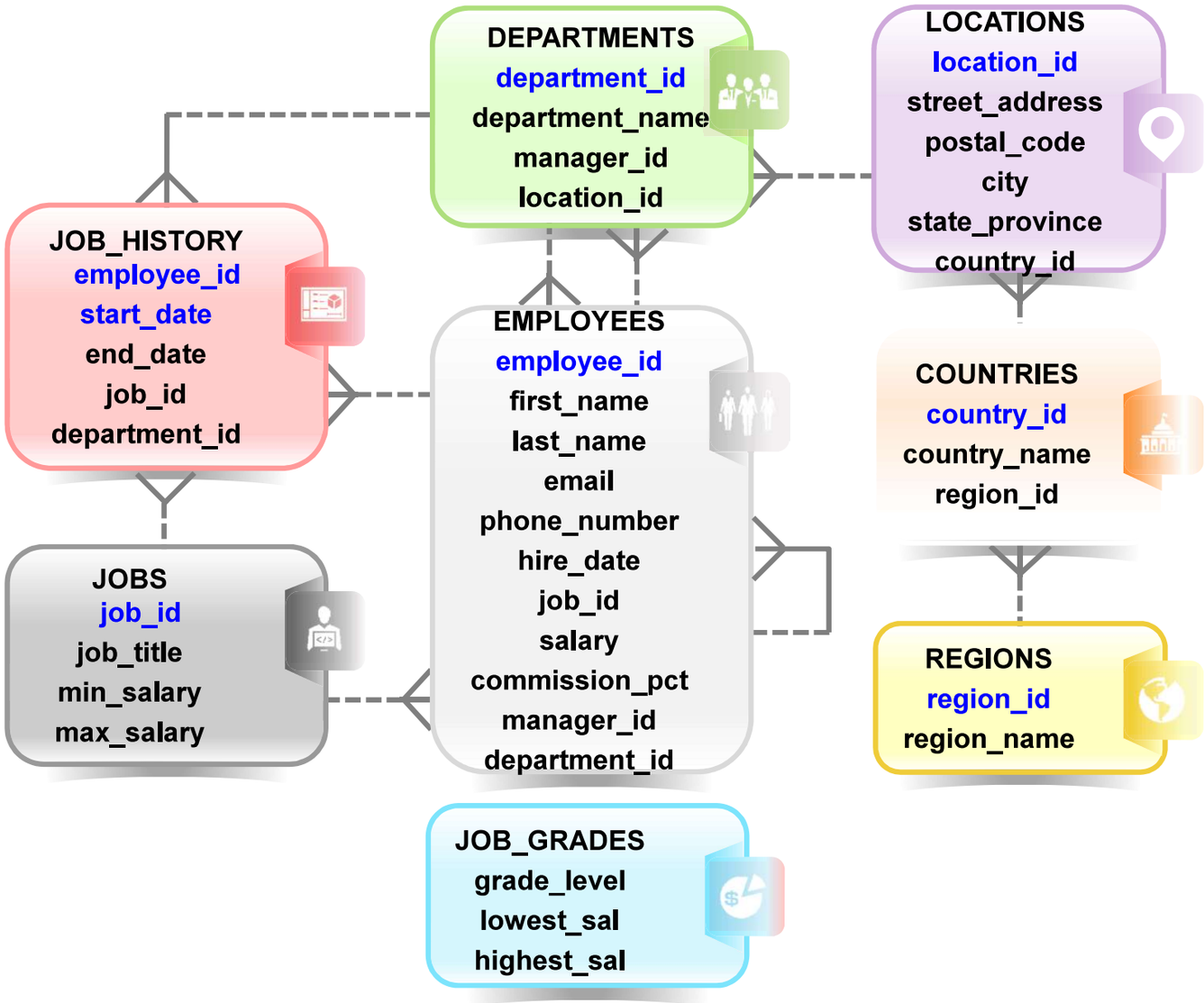
Characteristics of a physical data model:

- The physical data model describes data need for a single project or application though it maybe integrated with other physical data models based on project scope.
- Data Model contains relationships between tables that which addresses cardinality and nullability of the relationships.
- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project.
- Columns should have exact datatypes, lengths assigned and default values.
- Primary and Foreign keys, views, indexes, access profiles, and authorizations, etc. are defined.

Data model Schema and Instance

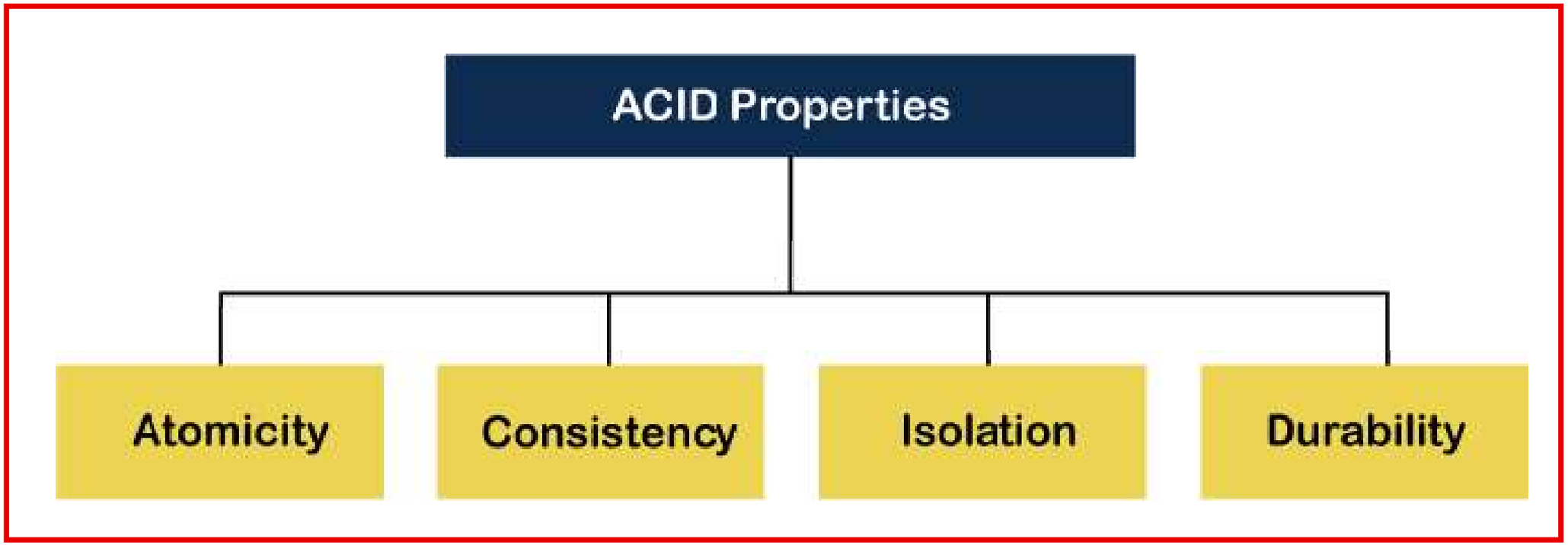
1. The data which is stored in the database at a particular moment of time is called an instance of the database.
2. The overall design of a database is called schema.
3. A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
4. A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
5. A database schema can be represented by using the visual diagram. That diagram shows the database objects and relationship with each other.
6. A database schema is designed by the database designers to help programmers whose software will interact with the database. The process of database creation is called data modeling.

Schema Diagram



ACID Properties In DB

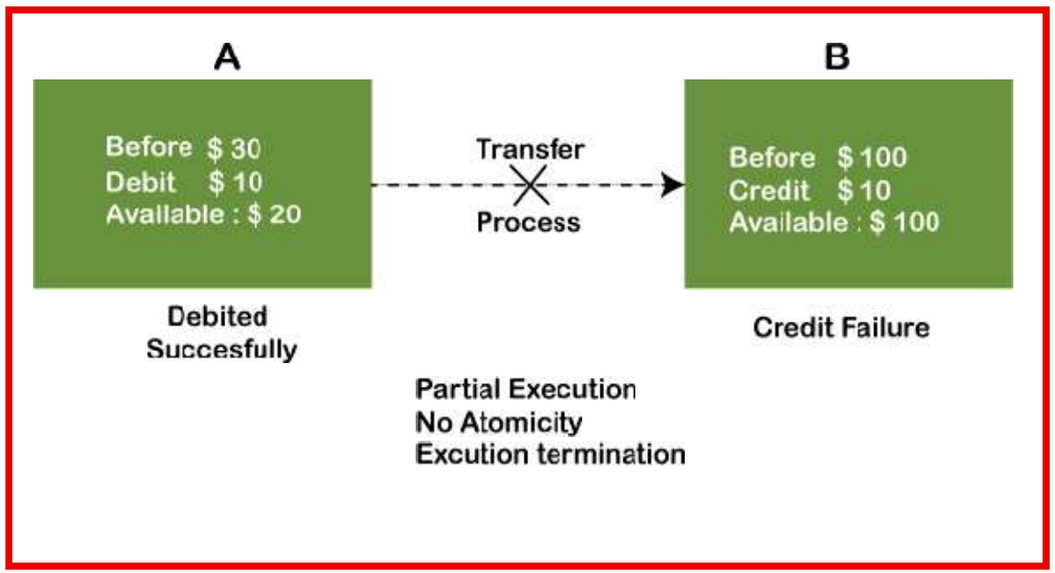
- DBMS is the management of data that should remain integrated when any changes are done in it.
- It is because if the integrity of the data is affected, whole data will get disturbed and corrupted. Therefore, to maintain the integrity of the data, there are four properties described in the database management system, which are known as the **ACID** properties.
- The ACID properties are meant for the transaction that goes through a different group of tasks, and there we come to see the role of the ACID properties.

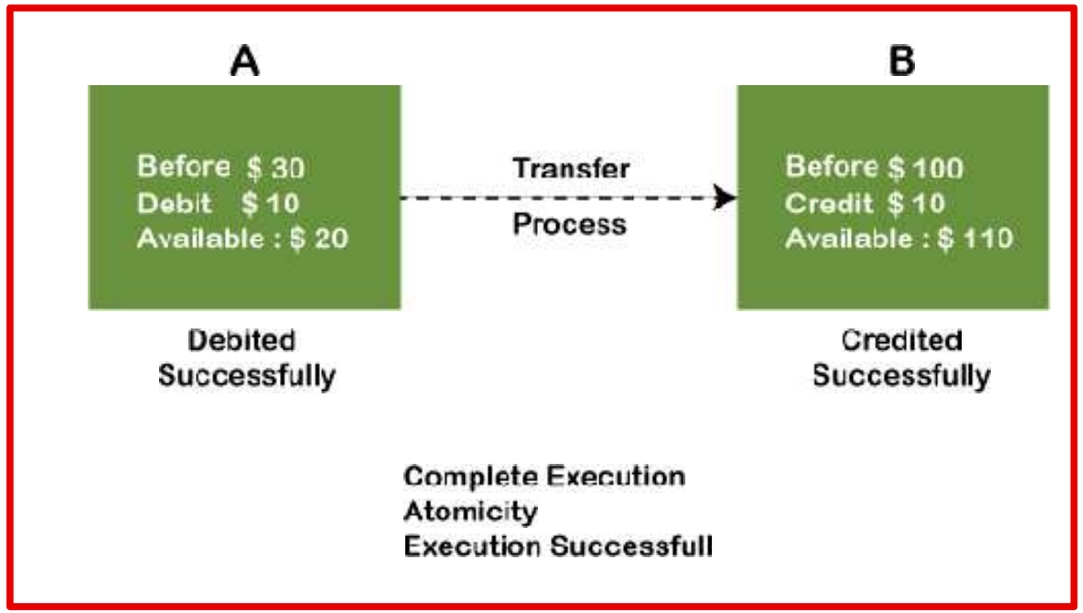

ACID Properties
Atomicity
Consistency
Isolation
Durability

1. Atomicity

- The term atomicity defines that the data remains atomic. It means if any operation is performed on the data, either it should be performed or executed completely or should not be executed at all.
- It further means that the operation should not break in between or execute partially. In the case of executing operations on the transaction, the operation should be completely executed and not partially.

Two Cases in Atomicity

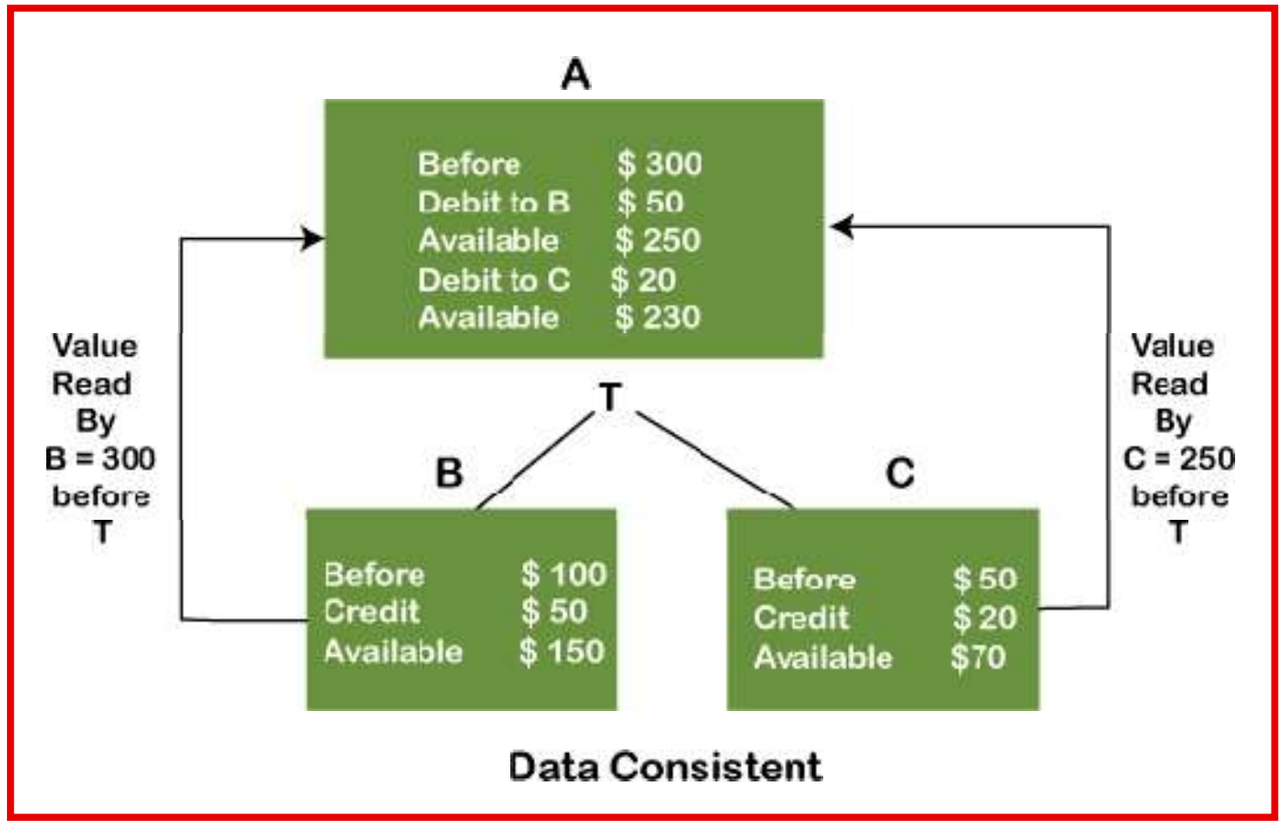




Thus, when the amount loses atomicity, then in the bank systems, this becomes a huge issue, and so the atomicity is the main focus in the bank systems.

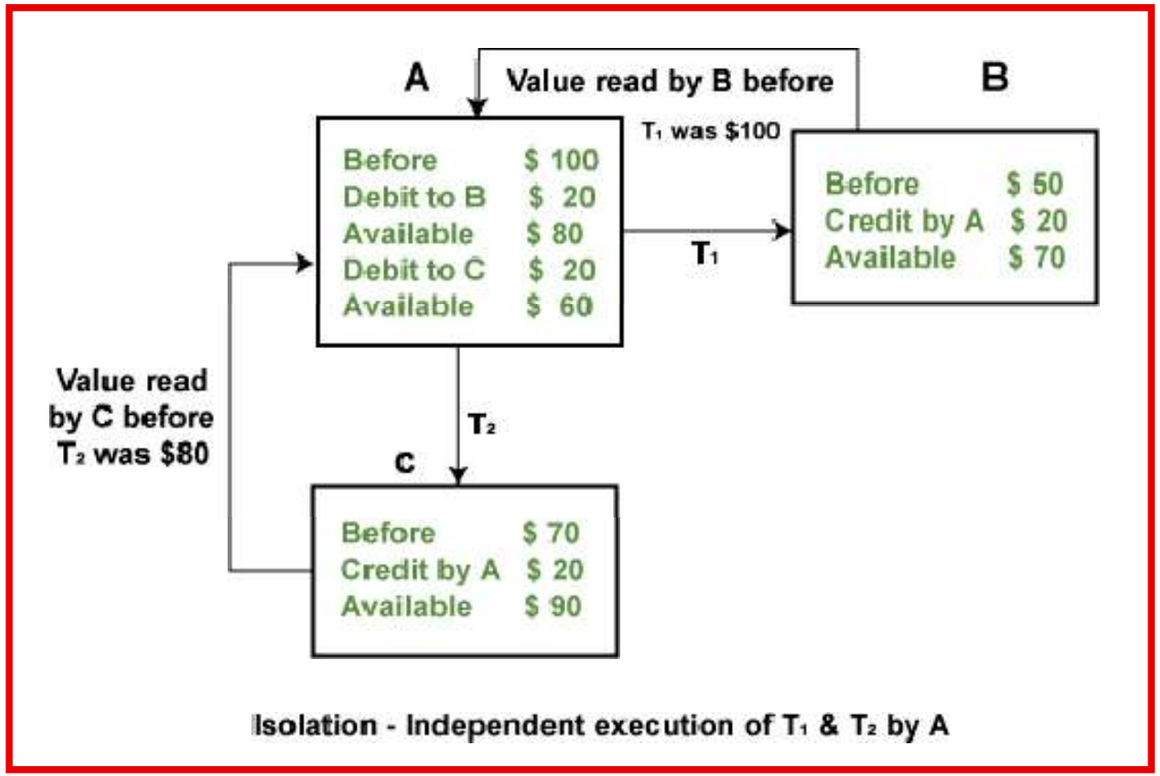
2. Consistency

- The word **consistency** means that the value should remain preserved always.
- In DBMS, the integrity of the data should be maintained, which means if a change in the database is made, it should remain preserved always.
- In the case of transactions, the integrity of the data is very essential so that the database remains consistent before and after the transaction. The data should always be correct.


A
Before $ 300
Debit to B $ 50
Available $ 250
Debit to C $ 20
Available $ 230
Value Read By B = 300 before T
T
B
C
Before $ 100
Credit $ 50
Available $ 150
Before $ 50
Credit $ 20
Available $70
Value Read By C = 250 before T
Data Consistent

3. Isolation

- The term 'isolation' means separation. In DBMS, Isolation is the property of a database where no data should affect the other one and may occur concurrently.
- In short, the operation on one database should begin when the operation on the first database gets complete.
- It means if two operations are being performed on two different databases, they may not affect the value of one another.
- In the case of transactions, when two or more transactions occur simultaneously, the consistency should remain maintained. Any changes that occur in any particular transaction will not be seen by other transactions until the change is not committed in the memory.


A
Value read by B before
B
T₁ was $100
Before $ 100
Debit to B $ 20
Available $ 80
Debit to C $ 20
Available $ 60
Before $ 50
Credit by A $ 20
Available $ 70
T₁
T₂
Value read by C before T₂ was $80
C
Before $ 70
Credit by A $ 20
Available $ 90
Isolation - Independent execution of T₁ & T₂ by A

4. Durability

- Durability ensures the permanency of something. In DBMS, the term durability ensures that the data after the successful execution of the operation becomes permanent in the database.
- The durability of the data should be so perfect that even if the system fails or leads to a crash, the database still survives. However, if gets lost, it becomes the responsibility of the recovery manager for ensuring the durability of the database.
- For committing the values, the COMMIT command must be used every time we make changes

Summary

In this lesson, you should have learned that:

- Overview of Data Model
- Stages of Data Model
- ACID in DBMS