SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

Data Models in DBMS

Course Name : 23CAT603 - DATA BASE MANAGEMENT SYSTEM

Class : I Year / II Semester

Unit I – DBMS Architecture

Agenda

- **Data Models**
- Database Languages and Interfaces

Data Models

What is a Data Model in DBMS?

A data model can be defined as an integrated collection of concepts describing and manipulating data, as well as constraints on the data within an organization.

A data model is not just a set of tables; it is a conceptual representation of physical data stored in the database.

Data Models

Designing a Data Model

The first step in designing a data model is to define what kind of data you want to store in your database.

This usually involves defining the entities involved in your application. An entity is any object that exists independently of anything else. You might think about an **entity** as something like a **"person" or an "address."**

1. **Once you've identified all the entities, you need to decide how they relate to each other.**
2. **The next step is to figure out how those entities will interact with each other.**
3. **Once you know the basic structure of your data model, you can start thinking about how best to organize it into tables.**

Data Models

There are two main approaches: relational databases and hierarchical databases:

Relational databases are organized around a set of tables that share common fields.

Hierarchical databases are organized around a tree-like structure called a hierarchy.

Why use Data Models?

Data models are extremely useful for any organization as they let business users directly define core business rules.

Advantages of using data models

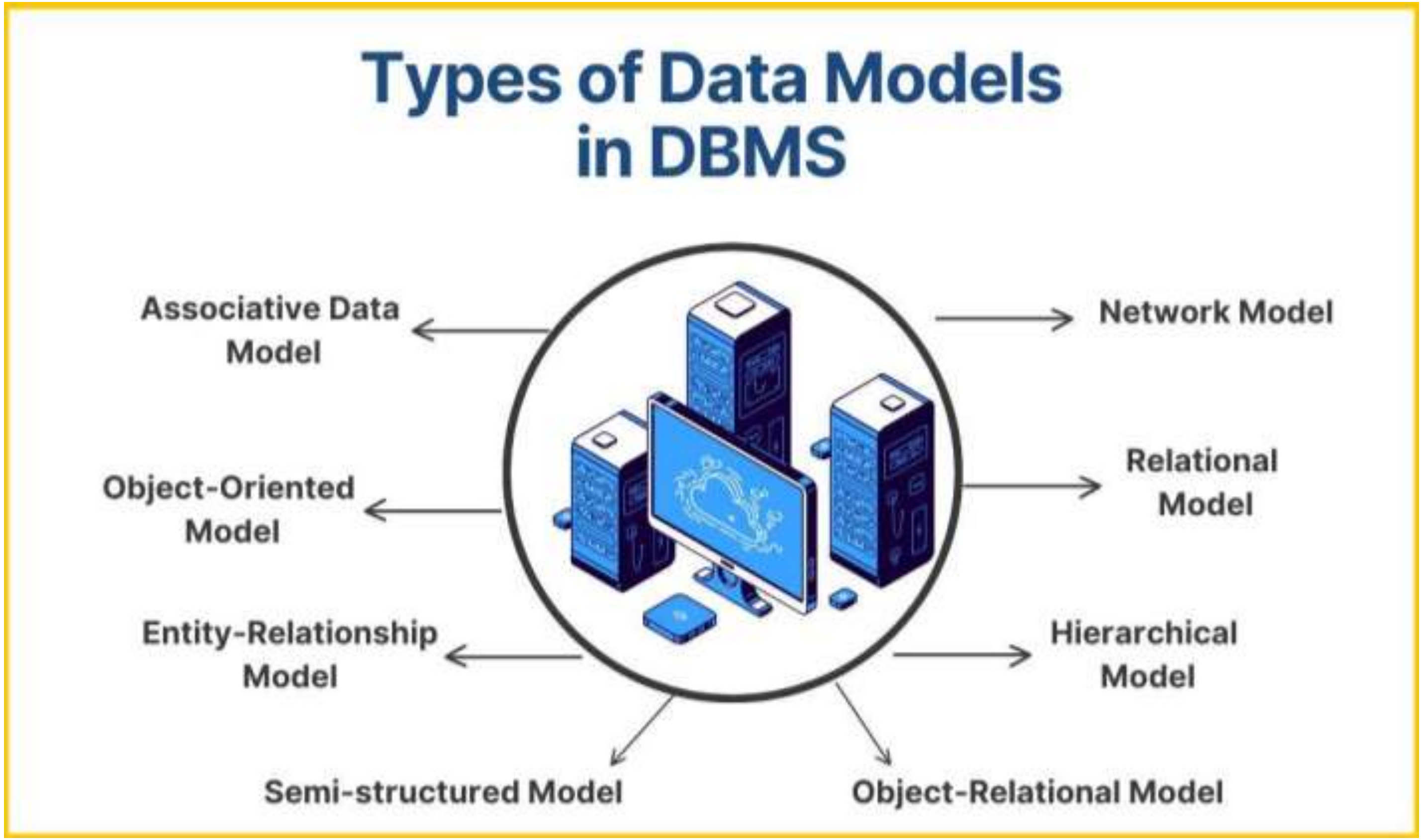
1. Helps to cut costs and deliver products faster
2. It improves business processes significantly
3. Reduces complexity and risks in data handling

Types of Data Model in DBMS

There are a variety of models for data. Some of the most well-known models include:

- A Hierarchical model
- Network model
- Model of Entity-Relationship
- Relational Model
- Object-Oriented Data Model
- Object-Relational Data Model
- Semi-Structured Data Model
- Associative Data Model
- Flat Data Model
- Context Data Model

Data Models in DBMS



1. Hierarchical Data Model

The hierarchical data model was the first DBMS-based model.

This model organizes data in a hierarchical tree structure.

The hierarchy begins at the root, which contains root data. It then expands into a tree, adding child nodes to the parent node. This model can easily represent real-world relationships, such as sitemaps, food recipes, or website navigation.

1. Hierarchical Data Model

Characteristics of Hierarchical Model:

One-to-many relationships: Here, the data is organized in a tree-like manner with a one-to-many relationship between data types. There can only be one path from any parent to any node.

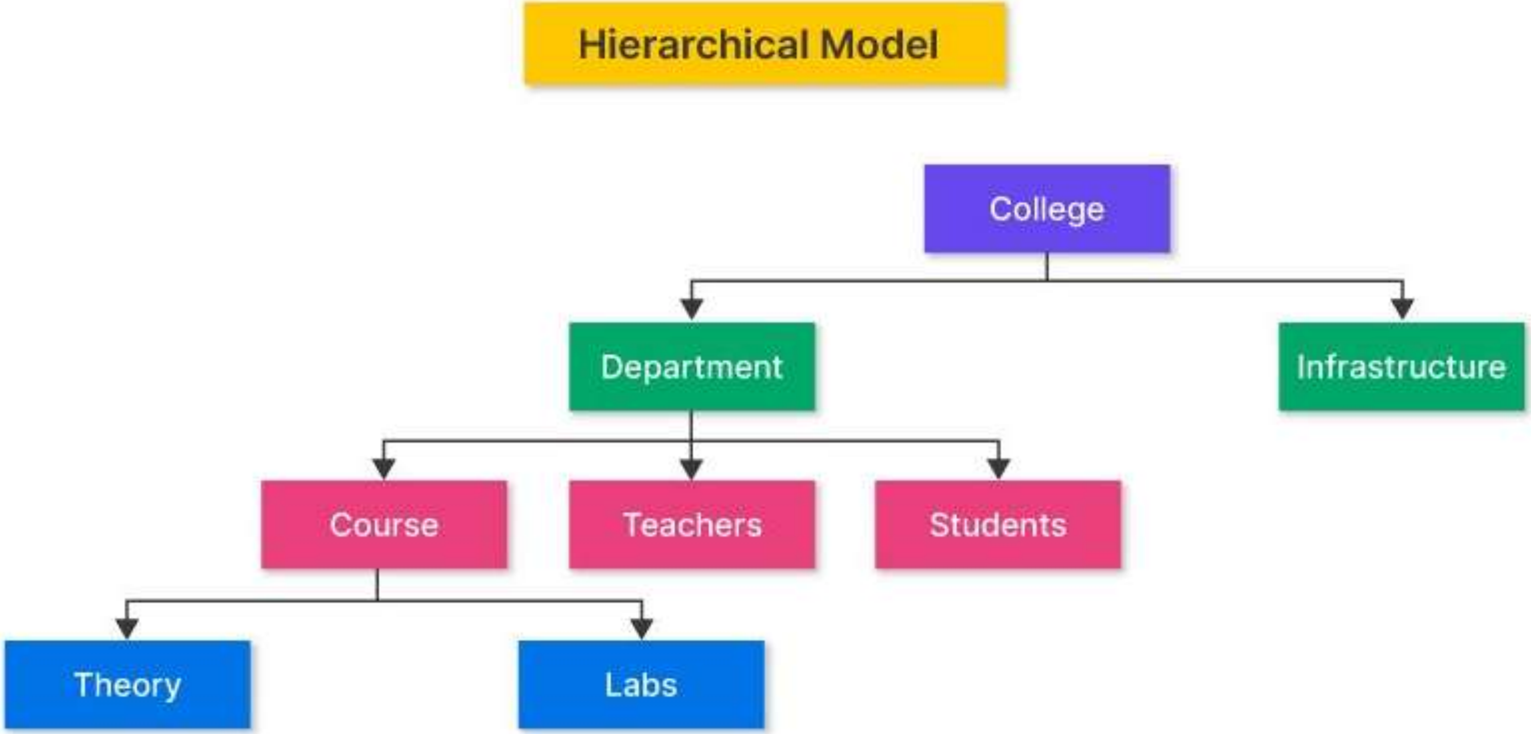
Parent-Child Relationship Every child node has its parent node, but a parent can have more than one child. Multiple parents are prohibited.

Deletion Issue: When a parent node has been deleted, the child node will also be deleted.

Pointers: Pointers link the parent and child nodes. They are used for navigation between stored data.

Example: The visual representation of the basic concepts of the database schema is shown below, where the college node points to two other nodes, the *department* and the *infrastructure*. This example is explained by taking real-world entities.

1. Hierarchical Data Model



1. Hierarchical Data Model

The Advantages of the Hierarchical Model

- It's very easy and quick to navigate through a tree-like structure.
- Changes in the parent node are automatically reflected in child nodes, so data integrity is maintained.
- Easy to find the difference.

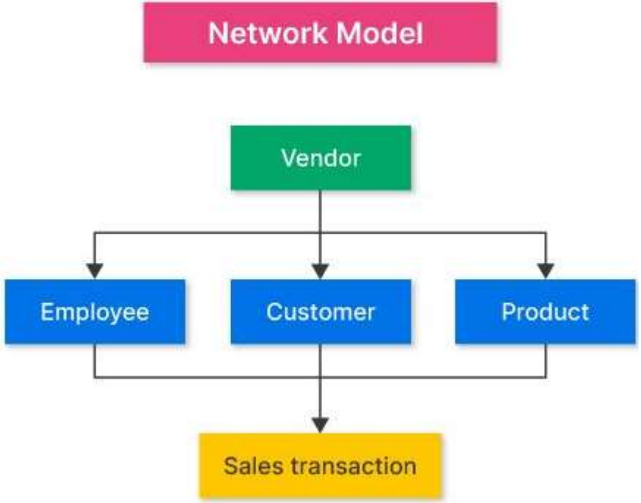
The Disadvantages of the Hierarchical Model

- Complex relationships cannot be supported.
- It does not support more than one parent of the child node. If there is a complex relationship where a child node requires two parent nodes, it cannot be represented with this model.
- The child node will be deleted automatically if a parent node has been deleted.

2. Network Model

This model is an extension of the hierarchical one - allowing for many-to-many relationships. Thus a record can have more than one parent in this model. This model was the most well-known before the relational one. This model replaces the hierarchical trees with a graph.

Example: In this example, we can see that all three child nodes (employee, customer and product) are connected to Vendor and Sales Transaction. In the hierarchical structure, this was not possible.



2. Network Model

Characteristics of Network Model

Ability To Merge More Relationships: In this model, data is more closely related to more relationships. This model can manage both one-to-one and many-to-many relationships.

Many routes: Because there are many relationships, multiple paths to the same record may exist. This allows data access to be quick and easy.

Circular Link List: The operations on the network model can be done using the circular linked list. A program maintains the current position and navigates through the records based on the relationship.

2. Network Model

The Advantages of the Network Model

- Data can be accessed more quickly than the hierarchical model. Because the data in the network model is more closely related, it is possible to take more than one route to reach a particular point. Thus, the data can be accessed faster and in many different ways.
- Data integrity is guaranteed because there is a parent/child relationship. Any changes in the parent record are reflected in a child record.

The Disadvantages of the Network Model

- The system can become more complicated as more relationships are needed to be managed. To use the model, the user must have a good understanding of it.
- Any change, such as updating, deletion, or insertion, is extremely complex.

3. Entity-Relationship (ER) Data Model

Entity-Relationship Data Model is a high-level diagram of a data model. **This model shows the real-world problem in a pictorial format to make it easier for stakeholders to understand.**

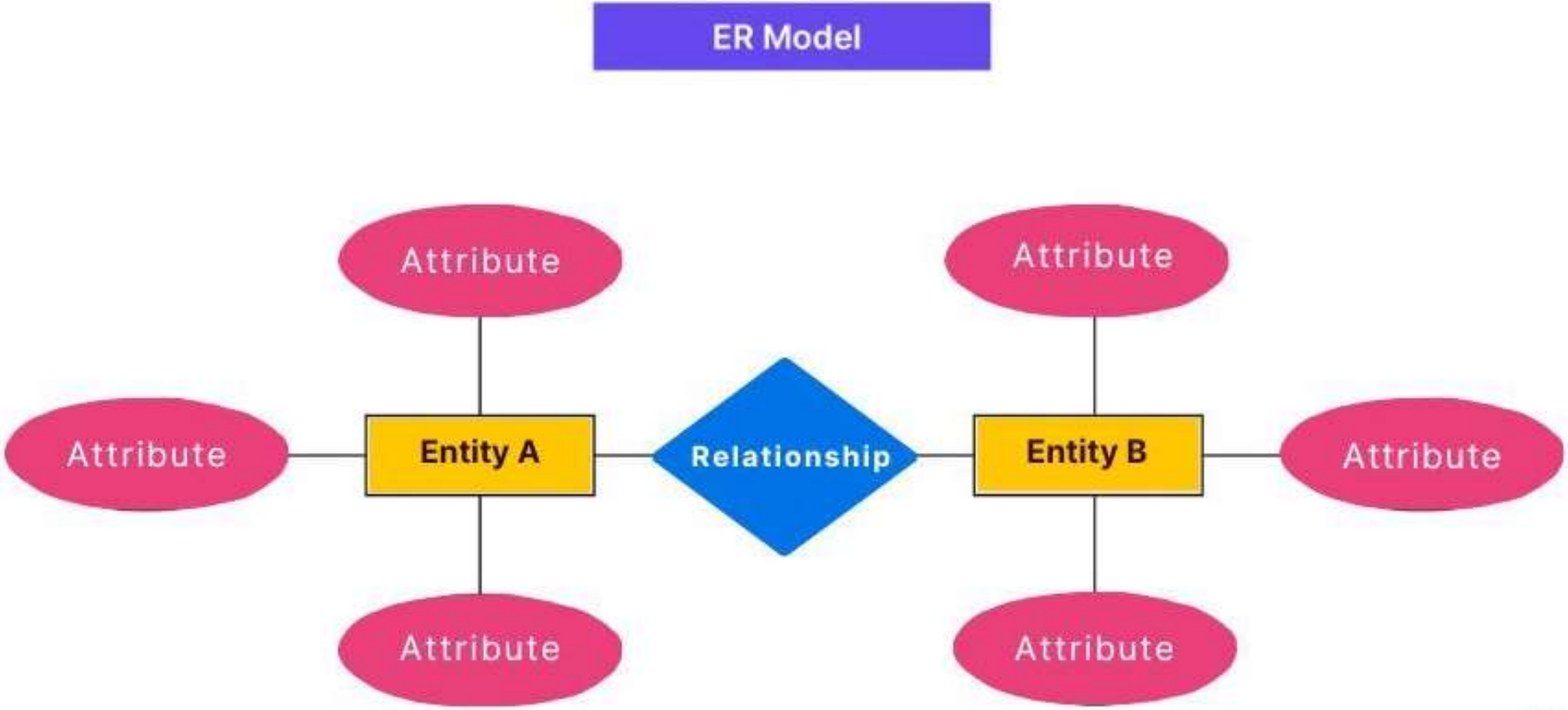
DBMS - ER Model diagram makes it easy for developers to see the system. The E-R model Diagram is used to show an ER Model or domain model. The following components make up the ER Diagram:

Entities: A real-world thing is an entity - it could be a person, a place, or even an idea. ***Examples:*** Students, Teachers, Courses, Buildings, Departments, etc., are all examples of entities that make up a School Management system.

Attributes: An entity can contain a real-world property called an attribute. ***Example:*** An entity student owns the property, student ID and age, etc. In the ER Model diagram, an ellipse shape is used to represent attributes.

Relationship: Relationship shows how two attributes are related. ***Example:*** A student working in a department. In the ER Model diagram, a diamond or rhombus shape represents the relationship.

3. Entity-Relationship (ER) Data Model



3. Entity-Relationship (ER) Data Model

Characteristics of the ER Model

Graphical representation for better understanding: It's very simple and easy to understand, so developers can use it to communicate with stakeholders.

ERD Diagram: The ERD diagram is used to represent the model visually.

Design of a Database: Database designers use this model to create a database.

Advantages of the ER Model

Easy: Conceptually, an ER Model can be very simple to create. The ER Diagram can be easily created if we know the relationships between the entities and attributes.

Effective Communication Tools: Database designers use this model to communicate their ideas.

Conversion to Any Model: This model is compatible with the relational model. It can easily be converted to a relational model by converting ER to the table. In fact, this model can be easily converted to any model, such as a hierarchical or network model.

3. Entity-Relationship (ER) Data Model

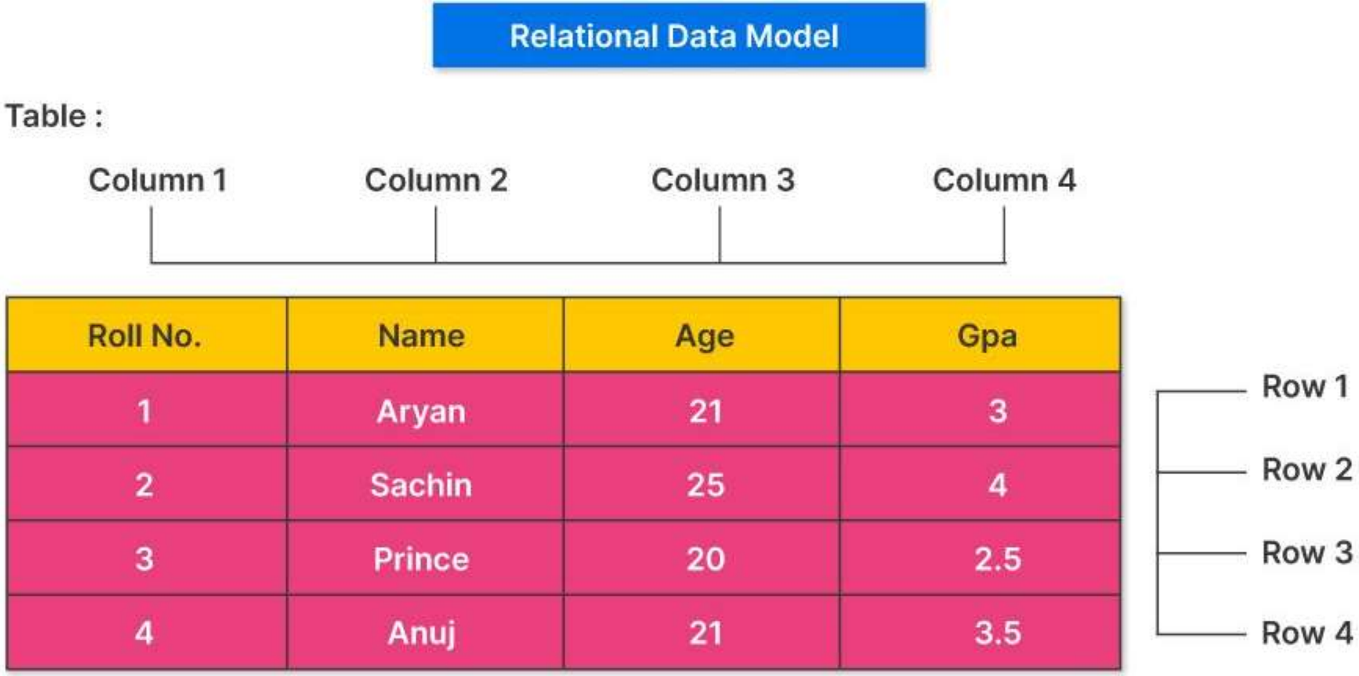
Disadvantages of the ER Model

There is no industry standard in notation: Developing an ER model according to industry standards is impossible. One developer may use notations that aren't understood by others.

Hidden Information: Some information might be lost or hidden within the ER model. Some information may be lost or hidden because it is a high-level view.

4. Relational Model

The relational model is the most popular model. **It is a model where the information is stored as two-dimensional tables.** All information is stored in rows and columns. The foundation of a relational model is tables. Therefore, the tables are known as *relations* within the model.



Relational Data Model

Table :

Roll No.	Name	Age	Gpa
1	Aryan	21	3
2	Sachin	25	4
3	Prince	20	2.5
4	Anuj	21	3.5

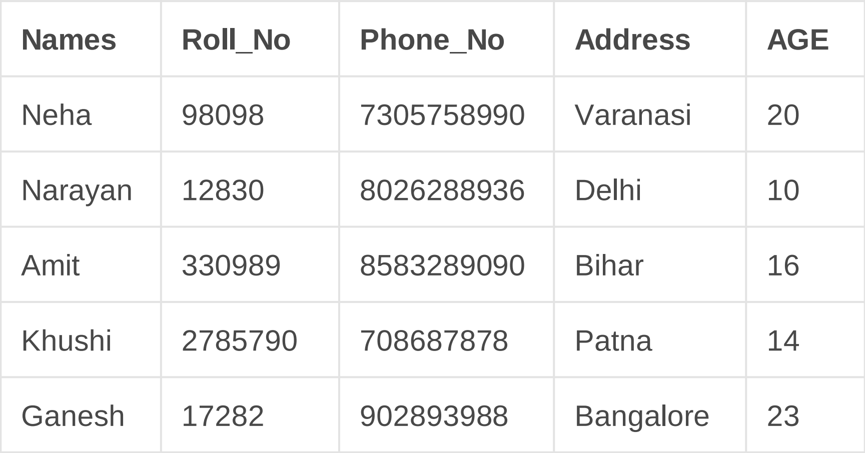
Column 1
Column 2
Column 3
Column 4
Row 1
Row 2
Row 3
Row 4

4. Relational Model

Characteristics of the Relational Model

Tuples: Each row within the table is known as the tuple. Each row is comprised of all the details regarding any particular specific instance. In the following example, each row is filled with all the information regarding a specific person - the first row contains information regarding Neha.

Field or attribute: Attributes are the property that defines the table or relationship. The attributes' values must be within that same domain. In the following example, attributes are the aspects of a *student* such as Roll_no, Age, Address, etc.



Names	Roll_No	Phone_No	Address	AGE
Neha	98098	7305758990	Varanasi	20
Narayan	12830	8026288936	Delhi	10
Amit	330989	8583289090	Bihar	16
Khushi	2785790	708687878	Patna	14
Ganesh	17282	902893988	Bangalore	23

4. Relational Model

Advantages of the Relational Model

Basic: This database model is less complex than the network and hierarchical models.

Scalable: This model can scale easily to add as many columns and rows as a user may like.

Structural Independence: It is possible to alter the structure of the database without altering the method for accessing the information. If we can modify the database structure without changing the ability of DBMS to access data, we can claim that structural independence is attained.

Disadvantages of the Relational Model

Hardware Overheads: This model calls for more powerful hardware computers and devices for storing data to conceal the complexity and make it easier for the user.

Poor Design: Because the model is extremely easy to create and utilize, users don't have to understand how data is stored for them to gain access to it. This simplicity of design could cause the development of a bad database, which could slow down as the database expands.

5. Object-Oriented Database Model

Object-oriented database models combine both data and the relationship in a single structure referred to as an object. It is possible to store images, audio, videos, and more in databases that were not possible with the relational model (although you could save video and audio in a relational database, it is advised not to store them in relational databases).

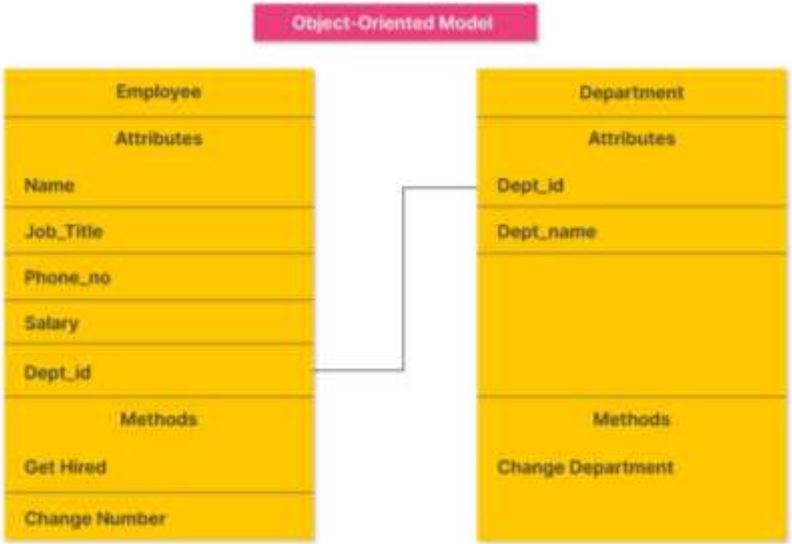
In this case, the model has additional objects linked by hyperlinks. This link is used to connect one object to another object. Let’s study the following example to understand the concept better.

In the example above, we have two objects (Employee and Department).

All the data is contained in a single unit (object), including their attributes like name, job_title, etc. for employees and dept_ID, dept_Name, for department.

These two objects are linked because of the one common attribute they share - the Department_Id.

The communication between the two objects is done through this common identification. This is the representation of the physical model of the Object-oriented data model.



4. Relational Model

Advantages of Object-Oriented Database Models (OODBs)

Natural Modeling: OODBs excel at representing real-world entities and their relationships.

Complex Object Handling: OODBs can handle complex data types beyond simple scalar values. They can store multimedia content, documents, and other intricate data structures within objects.

Disadvantages of Object-Oriented Database Models (OODBs)

Limited Adoption: OODBs haven't achieved the same level of widespread adoption as relational models. This can lead to a smaller pool of developers experienced with OODB management systems and potentially limit the choice of tools and resources available.

6. Object-Relational Model

It is a mix of the relational model and object-oriented model. **This model was designed to bridge the gap between object-oriented and relational models**.

It has several advanced features, such as creating complicated data types based on our needs using available data types. **The issue with this particular model is that it is often complicated and challenging to manage.** Therefore, a thorough understanding of the model is necessary.

7. Semi-Structured Model

A semi-structured model is a modified version of the relational model, allowing for greater flexibility in representing data. It is impossible to distinguish between schema and data in this model because this model is based on the assumption that certain entities might have attributes that are missing. In contrast, other entities may include an additional attribute.

Example: An online news website stores information about news articles. The fixed attributes can be in the Authors Table (Article ID, Title, Author ID, Publication Date) and Articles Table (Author ID, Name, Bio). However, the news website might also want to store additional information about each article that doesn't fit neatly into predefined columns. This could include:

- A list of relevant keywords (tags) for better searchability.
- Links to related articles.
- Social media reactions (likes, shares) associated with the article (if applicable).
- Geolocation data if the article is about a specific location.
- This additional information can be represented in a semi-structured format.

8. Associative Data Model

An associative model or Associative Data Model (ADM) is a model with the data divided into two components - entity and association. Everything with an independent existence is termed an *entity,* and the relation between the entities is called a *connection or association*.

The data is further divided into items and links:

Items: An item contains a name and an identifier (specific numerical value).

Link: The link comprises the source, verb, subject, and unique identifier.

It's a more intuitive way of representing entities with an inherent relationship. Think of it as a way to model data that captures the "who connects to whom" directly within the entities themselves.

Example: A library keeps track of books and authors.

Items:

Item Name: Book
Identifier: book_id (unique identifier for each book, could be a number or code)
Name: title (title of the book)

Item Name: Author
Identifier: author_id (unique identifier for each author)
Name: full_name (author's full name)

8. Associative Data Model

Links:

Link Identifier: link_id (unique identifier for each link)

Source: book_id (references the book identifier)

Verb: written_by (describes the relationship)

Subject: author_id (references the author identifier)

Explanation:

We have two items: "Book" and "Author," each with an identifier and a name attribute.

Links connect these items. A link has its own identifier, specifies the source item (book), the verb describing the relationship ("written_by"), and the subject item (author).

8. Associative Data Model

Example Data:

Book:
book_id: 123
title: "The Lord of the Rings"
Author:
author_id: 456
full_name: "J.R.R. Tolkien"
Link:
link_id: 789
source: 123 (book_id)
verb: written_by
subject: 456 (author_id)

This example shows how the associative model captures the relationship between books and authors using items and links. By following the links, you can easily see which author wrote a particular book.

The associative data model focuses on representing real-world entities and their direct relationships, often in a single structure. Unlike relational models with separate tables, it emphasizes the connections between things.

9. Flat Data Model

A flat data model is a very basic way of storing data in a database management system (DBMS). It's like a simple spreadsheet with rows and columns. Imagine a single table with columns representing different data points (like name, age, etc.) and rows representing individual records (like entries for each customer).

Think of it as a flat list with no hierarchy or connections between entries. It's a good starting point for simple data storage, but for anything more complex, relational models (with tables linked through relationships) are more efficient.

10. Context Data Model

The context data model is a collection of a variety of models. It is comprised of models such as network models, object-oriented data models, relational models, etc. With this model, we can accomplish various tasks that aren't achievable using just one model.

By understanding the strengths and weaknesses of different data models, you can make informed decisions when designing and implementing your database, ensuring optimal data storage, retrieval, and manipulation. As the data landscape continues to evolve, the development of new data models and advancements in existing ones will be crucial for effectively managing and unlocking the potential of information in the digital age.

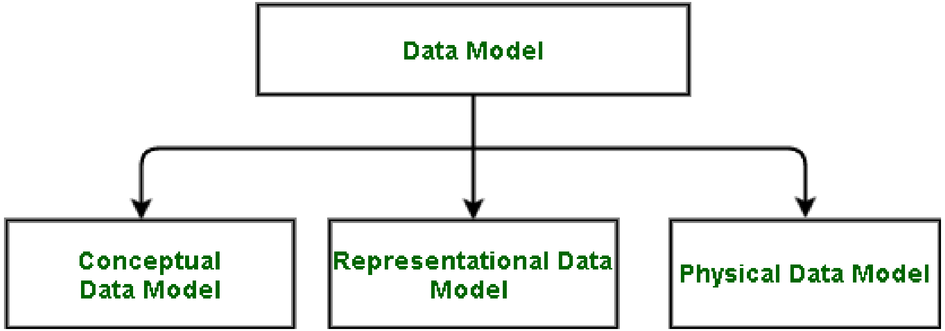
Data Models in DBMS

A Data Model in Database Management System (DBMS) is the concept of tools that are developed to summarize the description of the database. Data Models provide us with a transparent picture of data which helps us in creating an actual database. It shows us from the design of the data to its proper implementation of data.

Types of Relational Models

- Conceptual Data Model
- Representational Data Model
- Physical Data Model

It is basically classified into 3 types:-



Data Models in DBMS

1. Conceptual Data Model

The conceptual data model describes the database at a very high level and is useful to understand the needs or requirements of the database. It is this model, that is used in the requirement-gathering process i.e. before the Database Designers start making a particular database. One such popular model is the entity/relationship model (ER model). The E/R model specializes in entities, relationships, and even attributes that are used by database designers. In terms of this concept, a discussion can be made even with non-computer science(non-technical) users and stakeholders, and their requirements can be understood.

Entity-Relationship Model(ER Model): It is a high-level data model which is used to define the data and the relationships between them. It is basically a conceptual design of any database which is easy to design the view of data.

Components of ER Model:

Entity: An entity is referred to as a real-world object. It can be a name, place, object, class, etc. These are represented by a rectangle in an ER Diagram.

Attributes: An attribute can be defined as the description of the entity. These are represented by Ellipse in an ER Diagram. It can be Age, Roll Number, or Marks for a Student.

Relationship: Relationships are used to define relations among different entities. Diamonds and Rhombus are used to show Relationships.

Data Models in DBMS

Characteristics of a conceptual data model

- Offers Organization-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.”
- Conceptual data models known as Domain models create a common vocabulary for all stakeholders by establishing basic concepts and scope

Data Models in DBMS

2. Representational Data Model

This type of data model is used to represent only the logical part of the database and does not represent the physical structure of the database. The representational data model allows us to focus primarily, on the design part of the database. A popular representational model is a Relational model. The relational Model consists of Relational Algebra and Relational Calculus. In the Relational Model, we basically use tables to represent our data and the relationships between them. It is a theoretical concept whose practical implementation is done in Physical Data Model.

The advantage of using a Representational data model is to provide a foundation to form the base for the Physical model

Data Models in DBMS

3. Physical Data Model

The physical Data Model is used to practically implement Relational Data Model. Ultimately, all data in a database is stored physically on a secondary storage device such as discs and tapes. This is stored in the form of files, records, and certain other data structures. It has all the information on the format in which the files are present and the structure of the databases, the presence of external data structures, and their relation to each other. Here, we basically save tables in memory so they can be accessed efficiently. In order to come up with a good physical model, we have to work on the relational model in a better way. Structured Query Language (SQL) is used to practically implement Relational Algebra.

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.

Data Models in DBMS

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 models provide a blueprint for organizing and structuring data within a database. From **hierarchical models to relational models**, each has its unique use case. To explore these models and their practical applications, the **GATE CS Self-Paced Course**provides detailed coverage on **DBMS concepts**, ensuring you're well-prepared for your database-related exams.

Data Models in DBMS

Some Other Data Models

1. Hierarchical Model

The hierarchical Model is one of the oldest models in the data model which was developed by IBM, in the 1950s. In a hierarchical model, data are viewed as a collection of tables, or we can say segments that form a hierarchical relation. In this, the data is organized into a tree-like structure where each record consists of one parent record and many children. Even if the segments are connected as a chain-like structure by logical associations, then the instant structure can be a fan structure with multiple branches. We call the illogical associations as directional associations.

2. Network Model

The Network Model was formalized by the Database Task group in the 1960s. This model is the generalization of the hierarchical model. This model can consist of multiple parent segments and these segments are grouped as levels but there exists a logical association between the segments belonging to any level. Mostly, there exists a many-to-many logical association between any of the two segments.

Data Models in DBMS

3. Object-Oriented Data Model

In the Object-Oriented Data Model, data and their relationships are contained in a single structure which is referred to as an object in this data model. In this, real-world problems are represented as objects with different attributes. All objects have multiple relationships between them. Basically, it is a combination of Object Oriented programming and a Relational Database Model.

4. Flat Data Model

The float data model basically consists of a two-dimensional array of data models that do not contain any duplicate elements in the array. This data model has one drawback it cannot store a large amount of data that is the tables can not be of large size.

Data Models in DBMS

5. Context Data Model

The Context data model is simply a data model which consists of more than one data model. For example, the Context data model consists of ER Model, Object-Oriented Data Model, etc. This model allows users to do more than one thing which each individual data model can do.

6. Semi-Structured Data Model

Semi-Structured data models deal with the data in a flexible way. Some entities may have extra attributes and some entities may have some missing attributes. Basically, you can represent data here in a flexible way.

Data Models in DBMS

Advantages of Data Models

Data Models help us in representing data accurately.

It helps us in finding the missing data and also in minimizing Data Redundancy.

Data Model provides data security in a better way.

The data model should be detailed enough to be used for building the physical database.

The information in the data model can be used for defining the relationship between tables, primary and foreign keys, and stored procedures.

Disadvantages of Data Models

In the case of a vast database, sometimes it becomes difficult to understand the data model.

You must have the proper knowledge of SQL to use physical models.

Even smaller change made in structure require modification in the entire application.

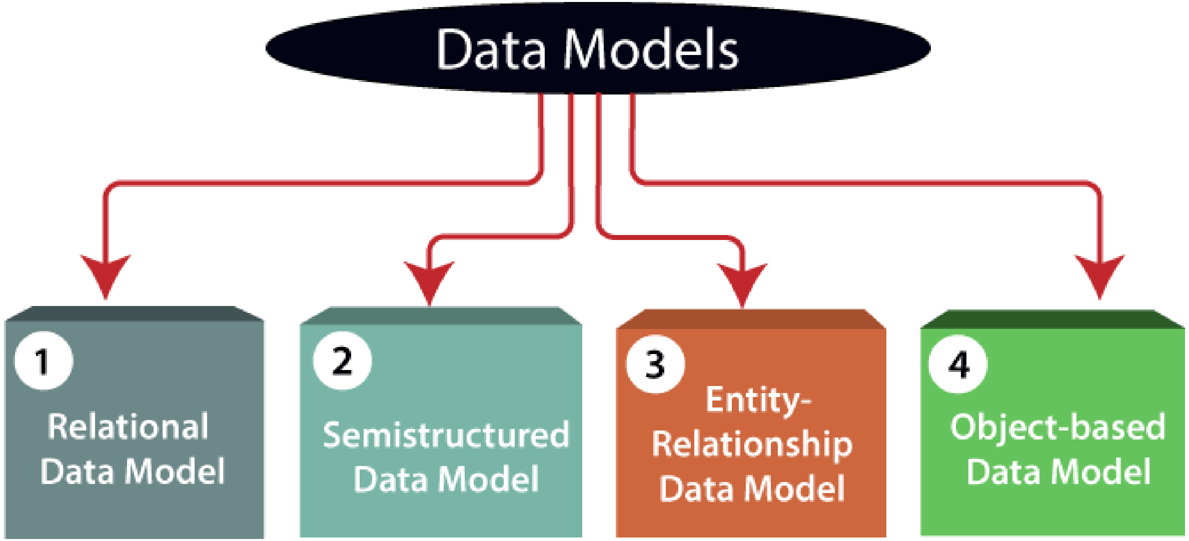
There is no set data manipulation language in DBMS.

To develop Data model one should know physical data stored characteristics.

Data Models in DBMS

Data Models

Data Model is the modeling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction. Therefore, there are following four data models used for understanding the structure of the database:

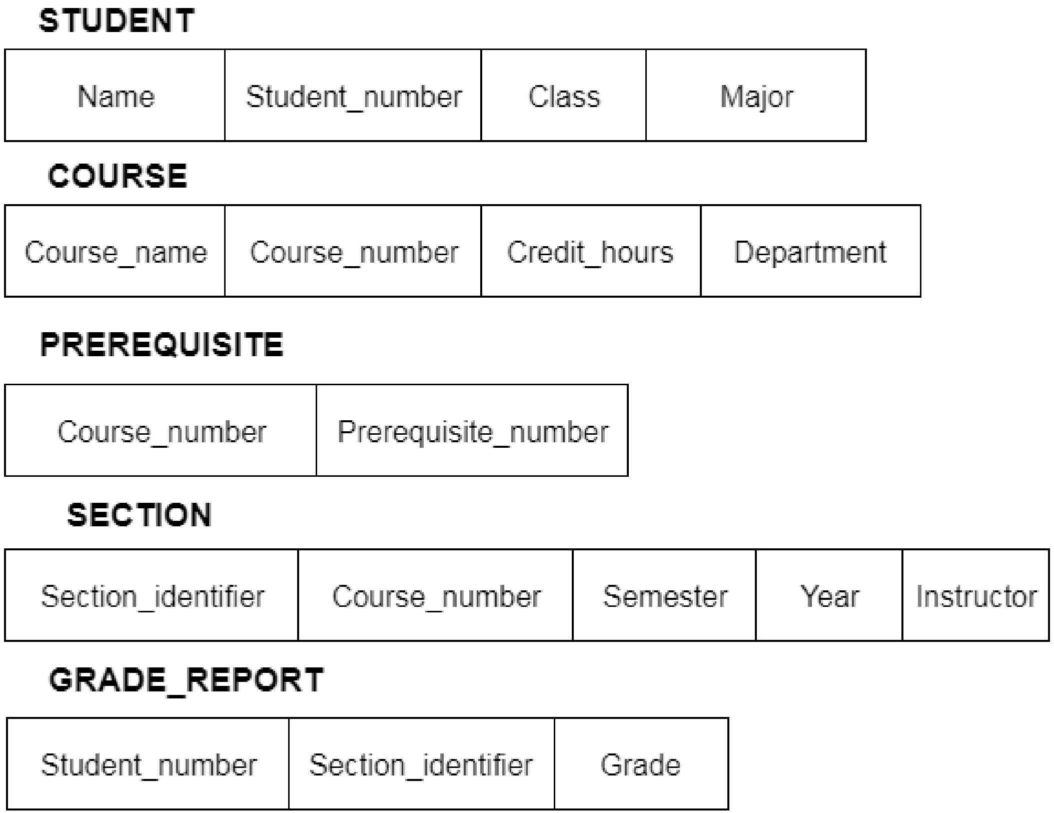

Data Models
1 Relational Data Model
2 Semistructured Data Model
3 Entity-Relationship Data Model
4 Object-based Data Model

Data Models in DBMS

Data model Schema and Instance

- The data which is stored in the database at a particular moment of time is called an instance of the database.
- The overall design of a database is called schema.
- A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
- A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
- A database schema can be represented by using the visual diagram. That diagram shows the database objects and relationship with each other.
- 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.
- A schema diagram can display only some aspects of a schema like the name of record type, data type, and constraints. Other aspects can't be specified through the schema diagram. For example, the given figure neither show the data type of each data item nor the relationship among various files.
- In the database, actual data changes quite frequently. For example, in the given figure, the database changes whenever we add a new grade or add a student. The data at a particular moment of time is called the instance of the database.

Data Models in DBMS



STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

References

https://www.geeksforgeeks.org/data-models-in-dbms/

https://www.javatpoint.com/dbms-data-model-schema-and-instance

https://unstop.com/blog/data-model-in-dbms