Database Management Systems

MR22-1CS0102

Dr. Santhosh Manikonda
Department of CSE
School of Engineering

Malla Reddy University, Hyderabad

Database Management Systems
II Yr – I Sem
CSE

COURSE OBJECTIVES

1. To learn the evolution of DBMS versus File Systems, Data Models, and Layers of Abstraction
2. To understand the relational database design principles.
3. To learn Formal and Commercial Query Language Specifications.
4. To become familiar with the basic issues of transaction processing and concurrency control.
5. To become familiar with database storage structures and access techniques

REFERENCE BOOKS

1. Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill 3rd Edition
2. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
3. Data base System Concepts, Silberschatz, Korth, McGraw hill, Sixth Edition
4. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson, Eight Edition
5. MySQL Introduction - Antun Peicevic, First edition, Technical editor: Marko Maslac Copyright@2016 Geek University Press

Syllabus
UNIT - I

Overview of Database Systems

- Managing Data
- File system vs DBMS
- Advantages of DBMS
- Describing and Storage Data in a DBMS
- Queries in DBMS
- Transaction Management
- Structure of DBMS
- Role based access
- Data Models.

Introduction to Database Design

- E-R Diagrams
- Entities, Attributes and Entity Sets, Relationships and Relationship Sets
- Additional Features of the ER Model
- Conceptual Database Design With the ER Model

Database Management Systems – Data, Information…

- What is Data?
- What is Information?

Database Management Systems – Data, Information…

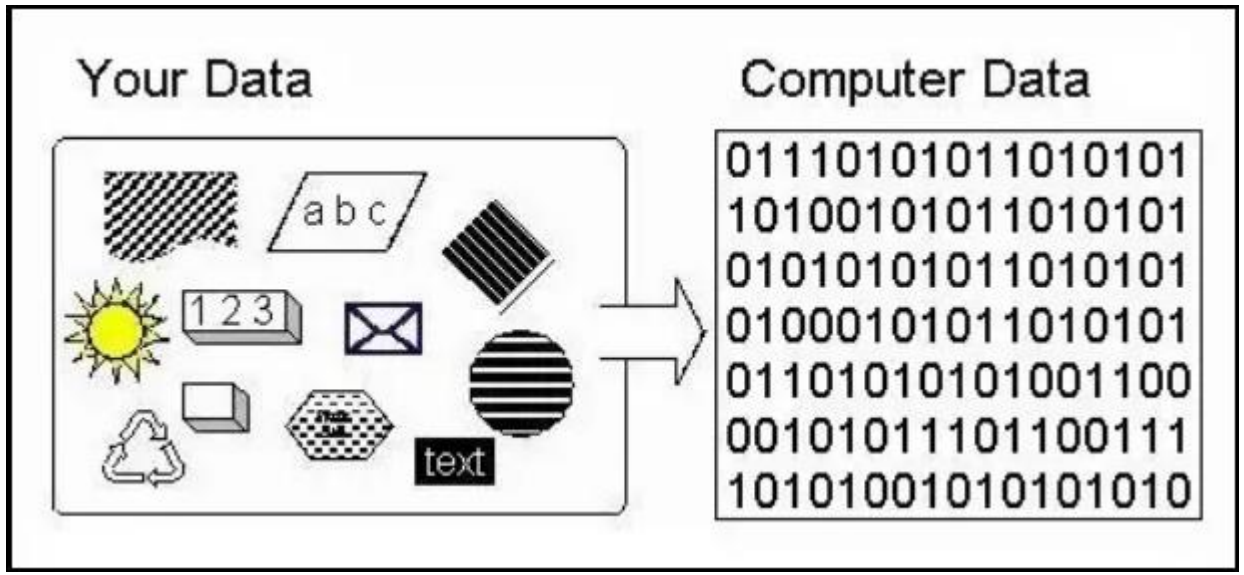
- Data: Raw or unprocessed facts or figures
  - *Quantitative – numeric*
  - *Qualitative – descriptive*
  - *Alone is not useful*
- Information: Processed / Interpreted / Organized / Structured / Presented in a meaningful way
  - Information is when data is given context - more specific
  - Knowledge is developed when information has been aggregated and analyzed to make decisions, set policies, and spark innovation
  - Wisdom is the combination of knowledge and experience
    - May take years to develop

Database Management Systems - Data, Information…

Data	Information
Data is used as input for the computer system	Information is the output of data
Data is unprocessed facts/figures	Information is processed data
Data is not specific	Information is specific
Data is a single unit	A group/collection of related data is called Information
Data doesn’t carry a meaning	Information must carry a logical meaning
Data is the raw material	Information is the product

Database Management Systems – Data Storage in computer

- All data in computers is stores as a sequence of 1's and 0's
  - 0 or 1 – 1 bit
  - 4 bits – 1 Nibble
  - 8 bits – 1 Byte
  - 16 bits – 1 Word



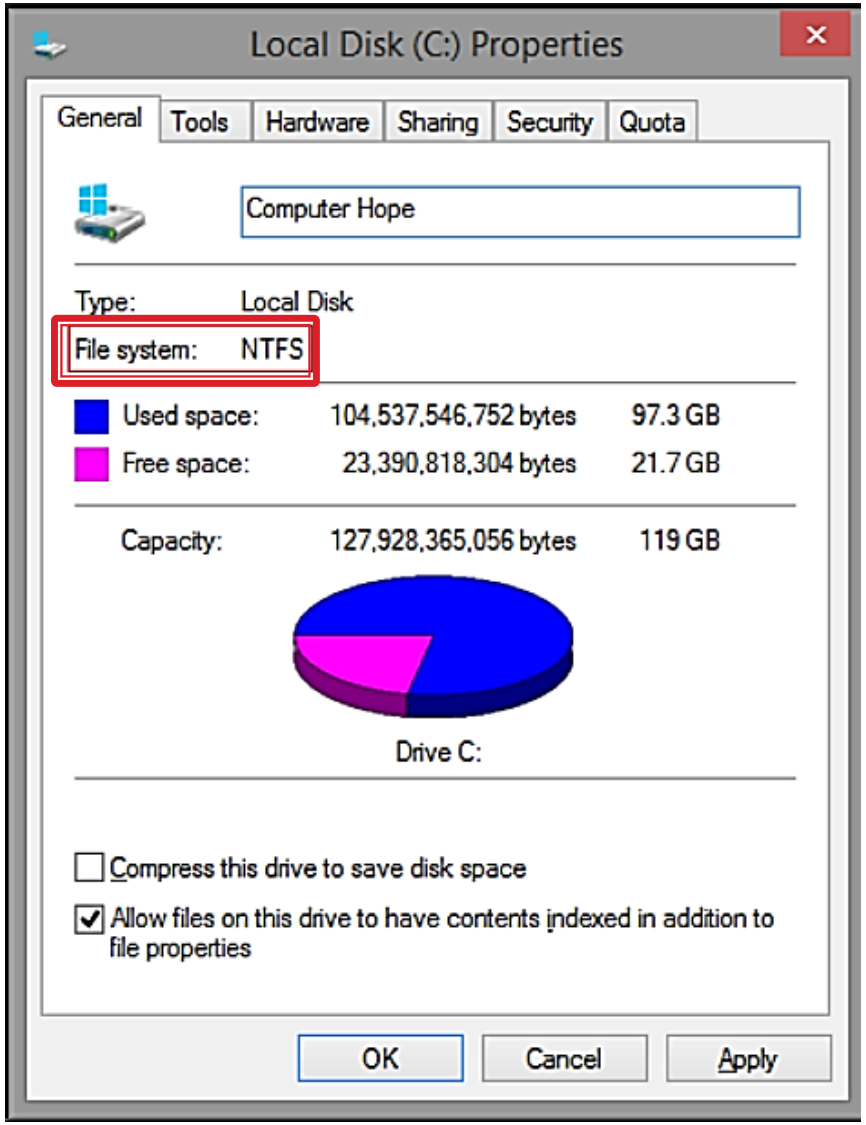
Database Management Systems – Managing Data

- Managing Data
  - Process of efficiently and securely organizing, storing, retrieving, and manipulating data in a structured and systematic manner
  - Creating databases, defining data structures, enforcing data integrity rules, ensuring data security
  - Key aspects include
    - Data Organization
    - Data Storage
    - Data Retrieval
    - Data Manipulation
    - Data Integrity
    - Concurrency Control
    - Back Up and Recovery

Database Management Systems – File System

- What is a File System?
  - A file system is a method/structure used by OS to manage and organize the files in a storage medium
  - A file system is a technique of arranging the files in a storage medium like a hard disk, pen drive, DVD, etc.
  - It helps you to organize the data and allows easy retrieval of files when they are required.
  - It mostly consists of different types of files like mp3, mp4, txt, doc, etc. that are grouped into directories.
  - A file system enables you to handle the way of reading and writing data to the storage medium.
  - It is directly installed into the computer with the Operating systems such as Windows and Linux.

Database Management Systems – File System



File System - Disadvantages

1. Data redundancy and inconsistency
2. Difficulty in accessing data
3. Data Isolation
4. Integrity Problems
5. Atomicity
6. Concurrent Access
7. Security Problems

File System - Disadvantages

1. Data redundancy and inconsistency
   - The same information may be written in several files. This redundancy leads to higher storage and access cost.
   - It may lead data inconsistency that is the various copies of the same data may longer agree for example a changed customer address may be reflected in single file but not elsewhere in the system.

2. Difficulty in accessing data
   - The conventional file processing system does not allow data to retrieve in a convenient and efficient manner according to user choice.

3. Data isolation
   - Because data are scattered in various file and files may be in different formats with new application programs to retrieve the appropriate data is difficult.

File System - Disadvantages

4. Integrity Problems

- Developers enforce data validation in the system by adding appropriate code in the various application program.
- However when new constraints are added, it is difficult to change the programs to enforce them – *Integrity Constraints*

5. Atomicity

- It is difficult to ensure atomicity in a file processing system when transaction failure occurs due to power failure, networking problems etc.
- Atomicity: either all operations of the transaction are reflected properly in the database or non are

File System - Disadvantages

6. Concurrent access

- In the file processing system it is not possible to access a same file for transaction at same time.

7. Security problems

- There is no security provided in file processing system to secure the data from unauthorized user access.

Database Management System

- What is a Database Management System (DBMS)?
  - A collection of interrelated data is referred to as a *Database*
  - Database usually contains information of an organization/enterprise
  - A DBMS is a collection of interrelated data and a set of programs to access, manipulate and maintain the database
  - Accessing/retrieving of data from DBMS is done by using *Queries*

Database Management System

- Function of DBMS
  - *Defining database schema* - It must give facility for defining the database structure also specifies access rights to authorized users
  - *Manipulation of the database* - The DBMS must have functions like insertion of record into database updation of data, deletion of data, retrieval of data - *Querying*
  - *Sharing of database* - The DBMS must share data items for multiple users by maintaining consistency of data
  - *Protection of database* - It must protect the database against unauthorized users
  - *Database recovery* - If for any reason the system fails DBMS must facilitate database recovery

Database Management System

- Applications of DBMS
  - Railway Reservation System
  - Library Management System
  - Banking
  - Education Sector
  - Credit card exchanges
  - Social Media Sites
  - Broadcast communications
  - Online Shopping
  - Human Resource Management

File System vs DBMS

	File System	DBMS
Structure	It is a software that manages and organizes the files in a storage medium within a computer.	DBMS is a software for managing the database.
Data Redundancy	Redundant data may be present in a file system.	In DBMS there is no redundant data.
Query Processing	There is no efficient query processing in file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in file system.	There is more data consistency because of the process of normalization
Complexity	It is less complex	It has more complexity in handling as compared to file system
Security Constraints	File systems provide less security in comparison to DBMS	DBMS has more security mechanisms as compared to file system.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Back Up and Recovery	It doesn’t provide backup and recovery of data if it is lost	It provides backup and recovery of data even if it is lost

DBMS - Advantages

1. Data Independence
2. Reducing Data Redundancy
3. Sharing of Data
4. Data Security
5. Privacy
6. Backup and Recovery
7. Data Consistency

DBMS - Advantages

1. Data Independence

- Data independence refers to characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

2. Reducing Data Redundancy

- There is a single database and any change in it is reflected immediately.

3. Sharing of Data

- In a database, the users can share the data among themselves.
- There are various levels of authorization to access the data - data can only be shared based on the correct authorization protocols
- Remote users can also access the database

DBMS - Advantages

4. Data Security

- Only authorized users will be allowed to access the database - password authenticated
- Unauthorized users will not be allowed to access the database under any circumstances

5. Privacy

- Only the authorized users can access a database according to their privacy constraints
- There are levels of database access and a user can only view the data he is allowed to

DBMS - Advantages

6. Backup and Recovery

- Database Management System automatically takes care of backup and recovery
- The users don’t need to backup data periodically because this is taken care of by the DBMS
- Moreover, it also restores the database after a crash or system failure to its previous condition

7. Data Consistency

- Data consistency is ensured in a database because there is no data redundancy.
- All data appears consistently across the database - data is same for all the users viewing the database
- Moreover, any changes made to the database are immediately reflected to all the users

DBMS - Disadvantages

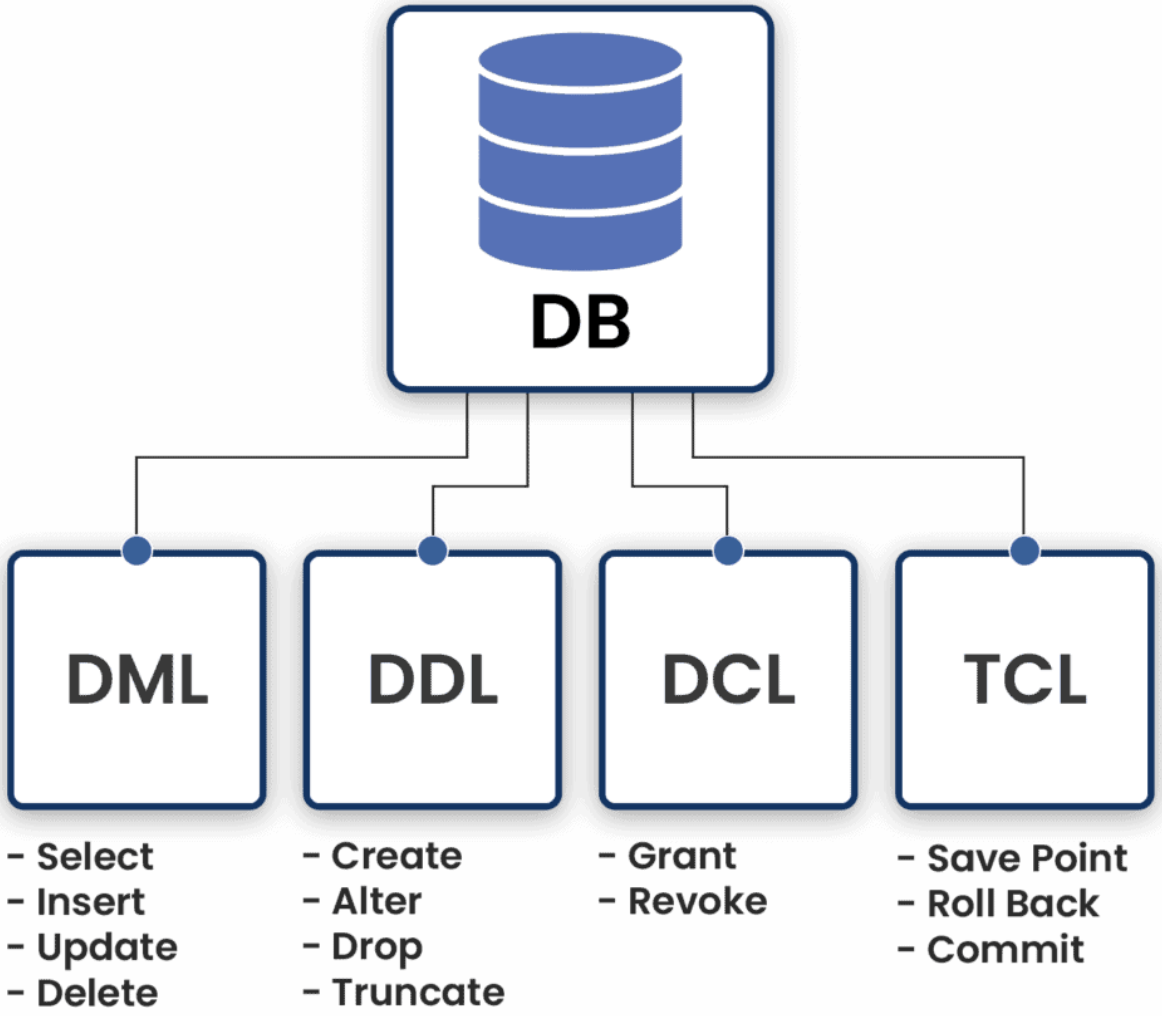
1. DBMS software and hardware (networking installation) cost is high
2. The processing overhead by the DBMS for implementation of security, integrity and sharing of the data.
3. Centralized database control
4. Setup of the database system requires more knowledge, money, skills, and time.
5. The complexity of the database may result in poor performance.

DBMS - Queries

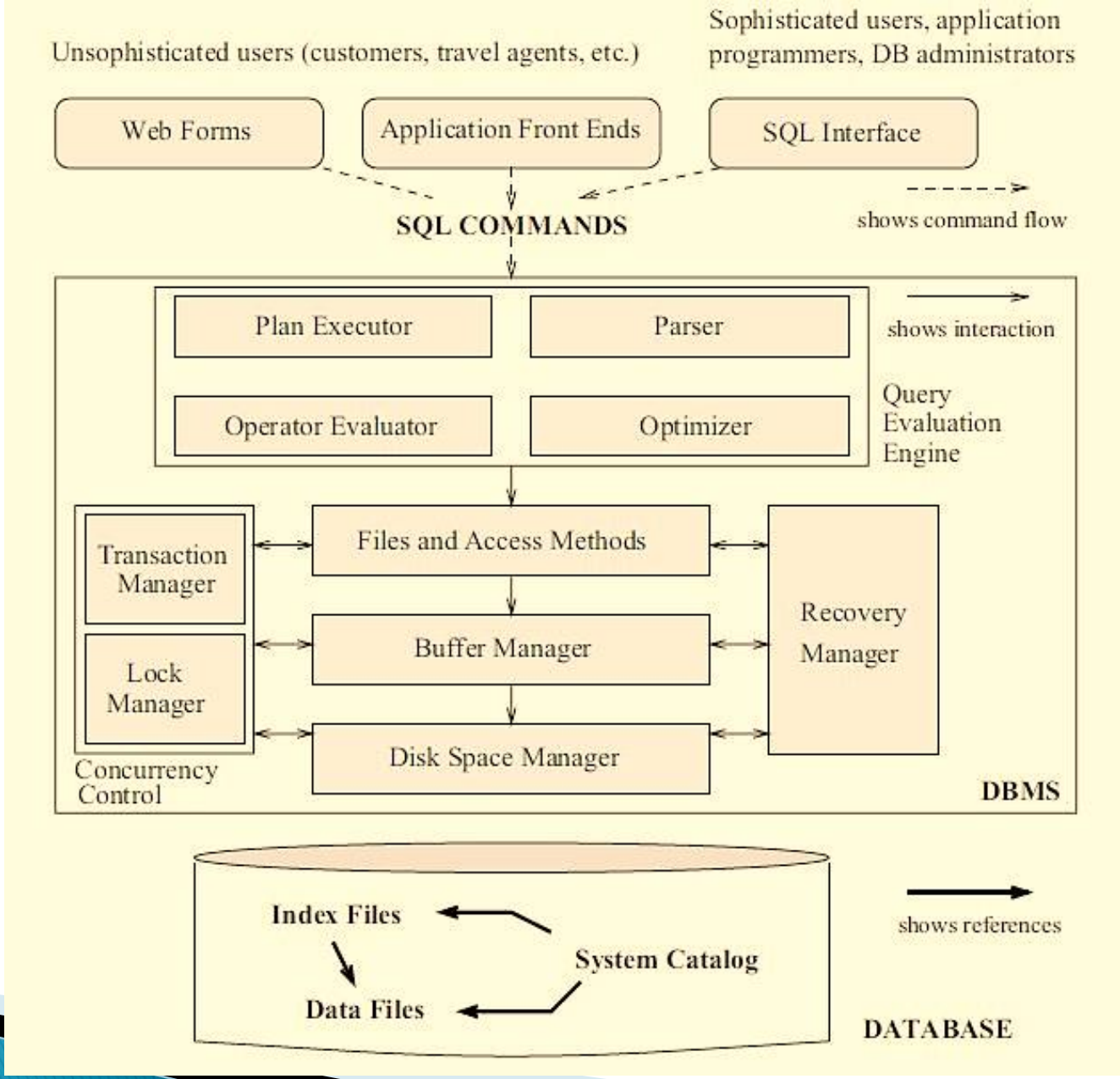
- Structured Query Language (SQL) is a database language that allows us to execute specific operations on existing databases as well as create new databases

- Data definition language queries (DDL)
- Data manipulation language queries (DML)
- Data control language queries (DCL)
- Transaction Control language queries (TCL)

DBMS - Queries



Structure of DBMS



Structure of DBMS

Query Evaluation Engine

- DDL interpreter, which interprets DDL statements and records the definitions in the data dictionary.
- DML compiler, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.
- A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs query optimization, that is, it picks the lowest cost evaluation plan from among the alternatives.
- Query evaluation engine, which executes low- level instructions generated by the DML compiler

Structure of DBMS

Transaction manager

- A transaction is a collection of operations that performs a single logical function in a database application.
- Each transaction is a unit of both atomicity and consistency.
- Thus, we require that transactions do not violate any database-consistency constraints.
- That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates.
- Transaction manager ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

Structure of DBMS

Lock manager

- Ensures concurrency control together with Transaction Manager

File manager

- Manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

Buffer manager

- It is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory.
- The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

Types of Database users

1. Database Administrator (DBA)

DBA is responsible for:

- Deciding the instances for the database.
- Defining the Schema
- Liaising with Users
- Define Security
- Back-up and Recovery
- Monitoring the performance

2. Database Designers

- Database designers design the appropriate structure for the database, where we share data.

3. System Analyst

- System analyst analyses the requirements of end users, especially naïve and para- metric end users.

Types of Database users

4. Application Programmers

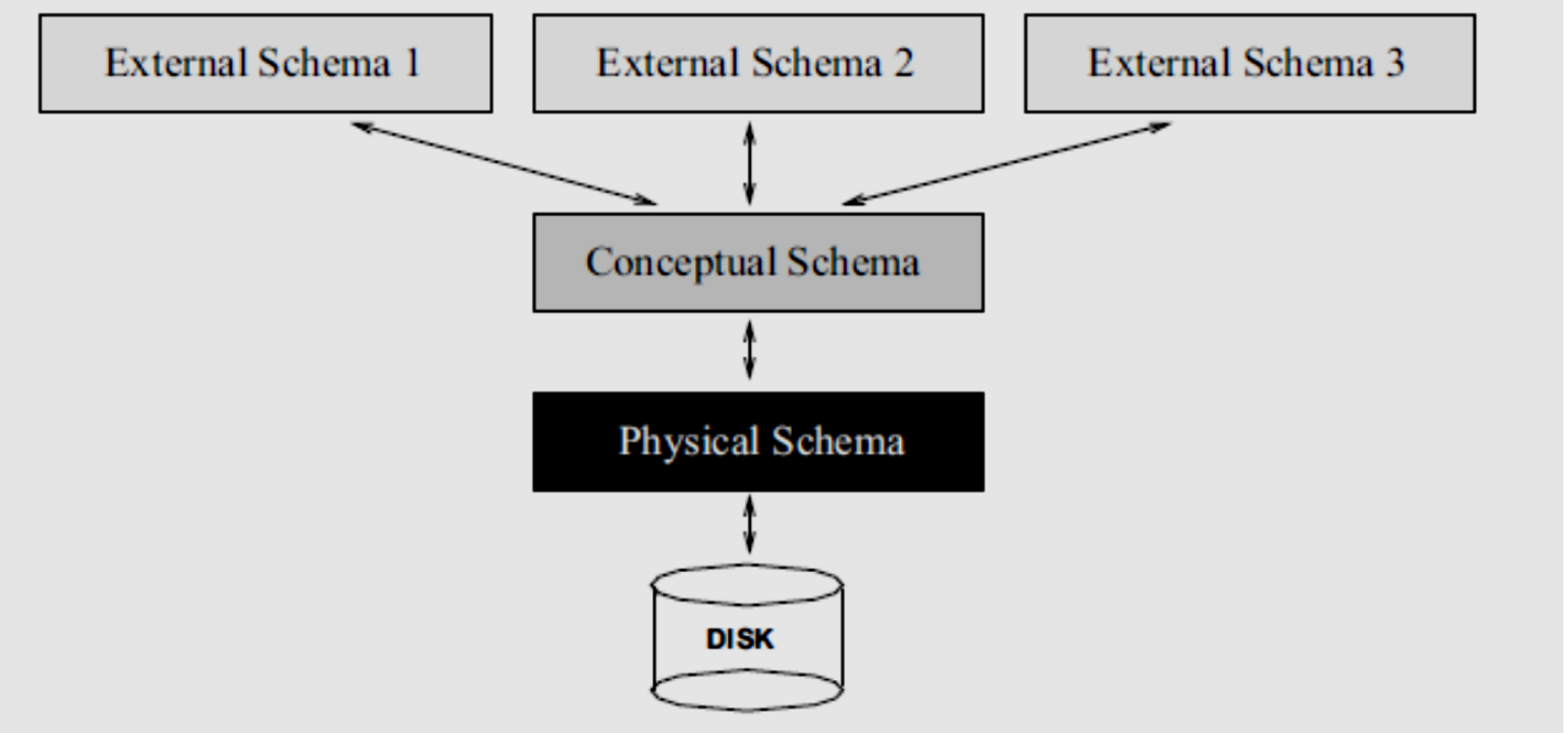
- Application programmers are computer professionals, who write application pro- grams.

5. Naïve Users / Parametric Users

- Has no knowledge of the database
- Use applications and get the desired result

Levels of Abstraction

- The data in a DBMS is described at three levels of abstraction
  - Conceptual or logical
  - Physical
  - External



Levels of Abstraction

- Conceptual Schema
- It describes the stored data in terms of the data model of the DBMS
- In a RDBMS, the conceptual schema describes all relations that are stored in the database
- Example: these relations contain information about entities, such as students and faculty, and about relationships, such as students' enrolment in courses.
- The choice of relations, and the choice of fields for each relation - **conceptual database design**

Students(*sid:* `string`, *name:* `string`, *login:* `string`,
age: `integer`, *gpa:* `real`)
Faculty(*fid:* `string`, *fname:* `string`, *sal:* `real`)
Courses(*cid:* `string`, *cname:* `string`, *credits:* `integer`)
Rooms(*rno:* `integer`, *address:* `string`, *capacity:* `integer`)
Enrolled(*sid:* `string`, *cid:* `string`, *grade:* `string`)
Teaches(*fid:* `string`, *cid:* `string`)
Meets_In(*cid:* `string`, *rno:* `integer`, *time:* `string`)

Levels of Abstraction

- Physical Schema
- Physical schema summarizes how the relations described in the conceptual schema are actually stored on storage devices
- Decide what file organizations to use to store the relations - Store all relations as unsorted files of records
- Create auxiliary data structures, such as, indexes – for speedy retrieval
- Physical schema are decided based on an understanding of how the data is typically accessed - physical database design

Levels of Abstraction

- External Schema
- External schemas are also in terms of the data model of the DBMS –
- It allows data access to be customized and authorized at the level of individual users or groups of users
- A database has only one conceptual and physical schema- only one set of relations
- A database may have many external schemas – role based access

Levels of Abstraction

- External Schema
- External schemas are also in terms of the data model of the DBMS –
- It allows data access to be customized and authorized at the level of individual users or groups of users
- A database has only one conceptual and physical schema- only one set of relations
- A database may have many external schemas – role based access

- Data independence is achieved by three levels of abstraction
  - Conceptual schema and External schema
  - Application programs are not affected by how data is structured and stored

Describing and Storing Data - Data models

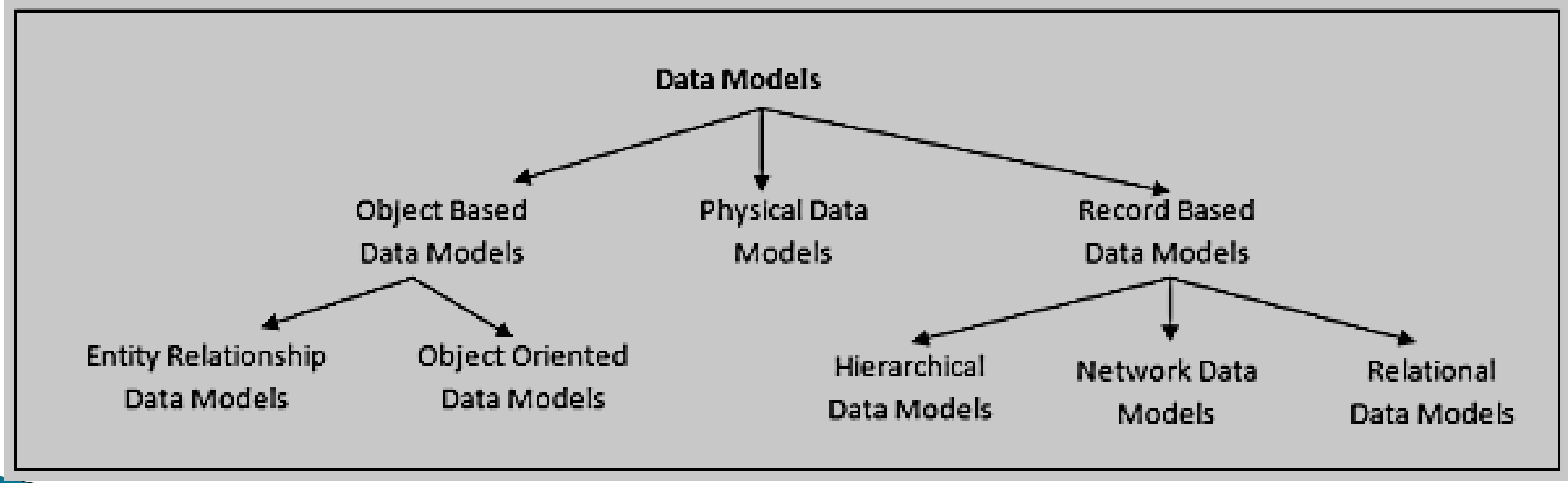
- DBMS allows a user to define the data to be represented in terms of a data model
- Data model is a *collection of high-level data description constructs that hide many low-level storage details*
- It is a collection of concepts that can be used to describe the structure of a database

- Data models according to the types of concepts they use to describe the database structure:

  - *High-level or conceptual data models* provide concepts that are close to the way many users perceive data.
  - *Low-level or physical data models* provide concepts that describe the details of how data is stored on the computer storage media

Describing and Storing Data - Data models

- Data models can be categorized according to the types of concepts they use to describe the database structure.



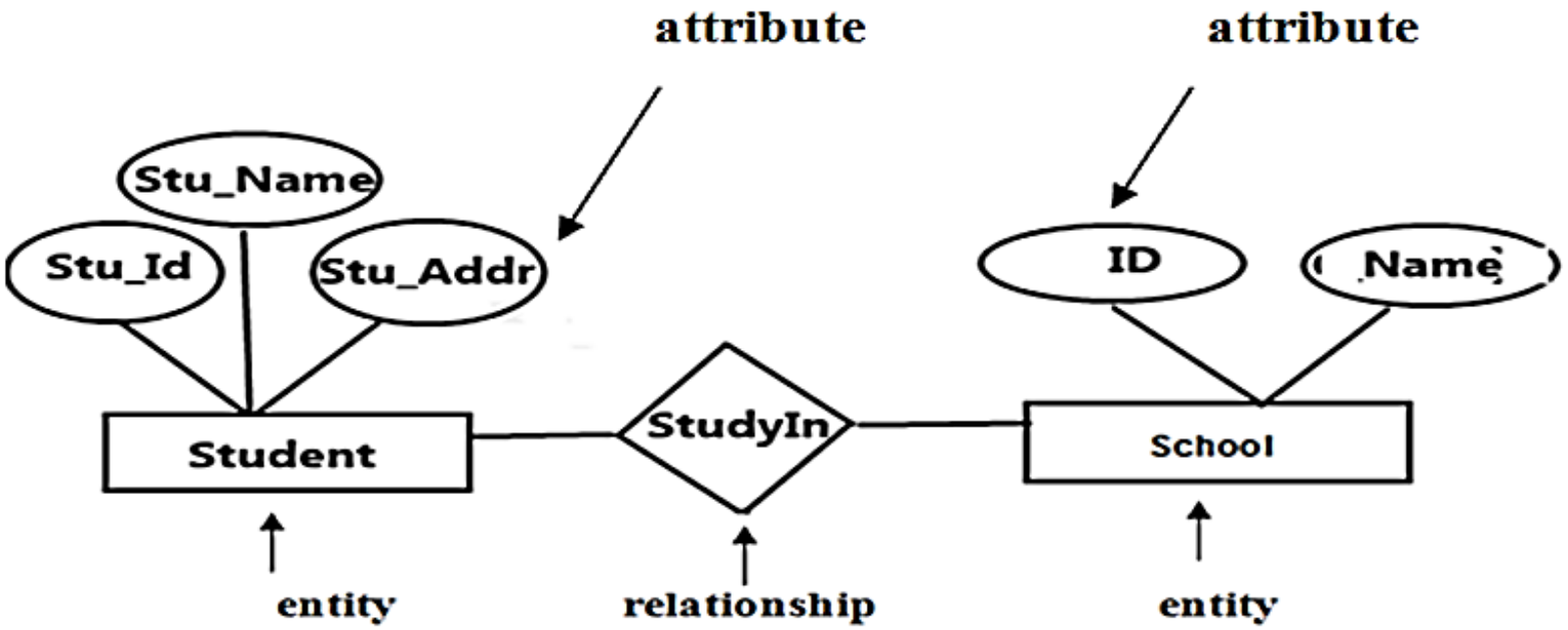
Describing and Storing Data - Data models

- A semantic data model is a more abstract, high-level data model that makes it easier for a user to come up with a good initial description of the data in an enterprise

- A widely used semantic data model called the entity-relationship (ER) model allows us to pictorially denote entities and the relationships among them.

Data models – E – R Model

- Conceptual or semantic data model is a more abstract, high-level data model
- It is easier for a user to come up with a good initial description of DB
- Subsequently translated into a database design in terms of the data model the DBMS actually supports
- A widely used semantic data model called the entity-relationship (ER) model allows us to pictorially denote entities and the relationships among them
  - *entities, attributes, and relationships*



Data models – E – R Model - Features

- Graphical Representation for Better Understanding – It is really straightforward and easy to comprehend, so developers can use it to interact with stakeholders
- Database Design – This approach is extensively used in database design and aids database designers in the creation of databases
- ER Diagram – The ER diagram is a visual representation of the model

Data models – E – R Model

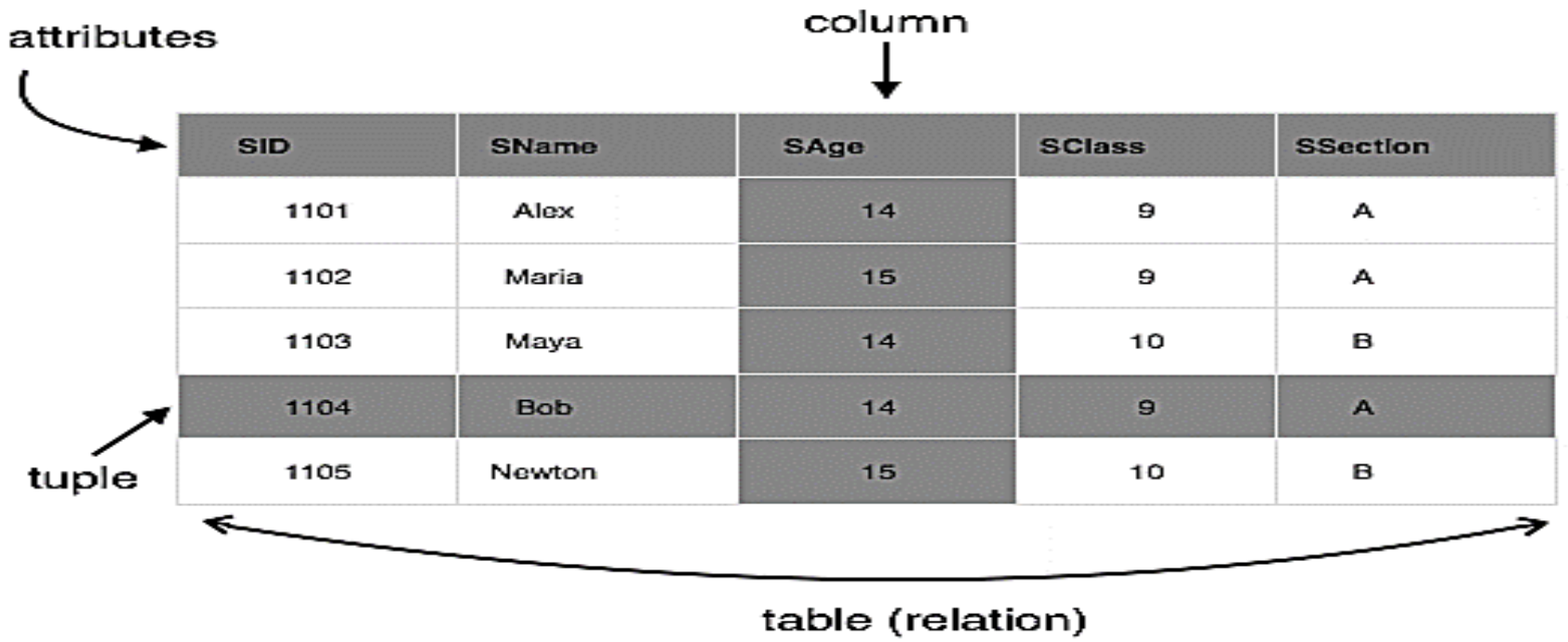
- Advantages
  - Simple – The ER Model is simple to construct conceptually if we know the relationship between the entities and the attributes.
  - Effective communication tool – Database designers frequently employ this model to communicate their thoughts.
  - Easy conversion to any model – This model translates neatly to the relational model, and it is simple to transform the ER model into a table.
    - This model can be transformed into a network model, a hierarchical model, and so forth.
- Disadvantages:
  - No industry standard for notation – When it comes to creating an ER model, there is no industry standard. As a result, one developer may utilize notations that are unfamiliar to other developers
  - Hidden information – In the ER model, certain information may be lost or hidden. Because it is a high-level view, there is a potential that some information specifics will be buried.

Data models – Relational Model

- The central data description construct in this model is a relation, which can be thought of as a set of records
- A description of data in terms of a data model is called a schema
- In the relational model, the schema for a relation specifies its name, the name of each field (or attribute or column), and the type of each field
- In relational model data is stored in the form of tables
- Data's physical storage is independent of its logical organization
- Example: University Database
  - Students(sid: string, name: string, login: string, age: integer, gpa: real)

Data models – Relational Model

- Example: University Database
  - Students(SID: integer, SName: string, SAge: integer, SClass: integer, SSection: string)
- Each record in the Students relation has five fields, with field names and types
- Every row follows the schema of the Students relation. The schema can therefore be regarded as a template for describing a student.



SID	SName	SAge	SClass	SSection
1101	Alex	14	9	A
1102	Maria	15	9	A
1103	Maya	14	10	B
1104	Bob	14	9	A
1105	Newton	15	10	B

Data models – Relational Model

- Relational Model Advantages
  - Structural independence is promoted using independent tables
  - Tabular view improves conceptual simplicity
  - Ad hoc query capability is based on SQL
  - Isolates the end user from physical-level details
  - Improves implementation and management simplicity

- Disadvantages:
  - Requires substantial hardware and system software overhead
  - Conceptual simplicity gives untrained people the tools to use a good system poorly
  - May promote information problems

Data models – Relational Model

- Properties of Relational Model
  - Every row is unique
  - All of the values present in a column hold the same data type
  - Values are atomic
  - The columns sequence is not significant
  - The rows sequence is not significant
  - The name of every column is unique

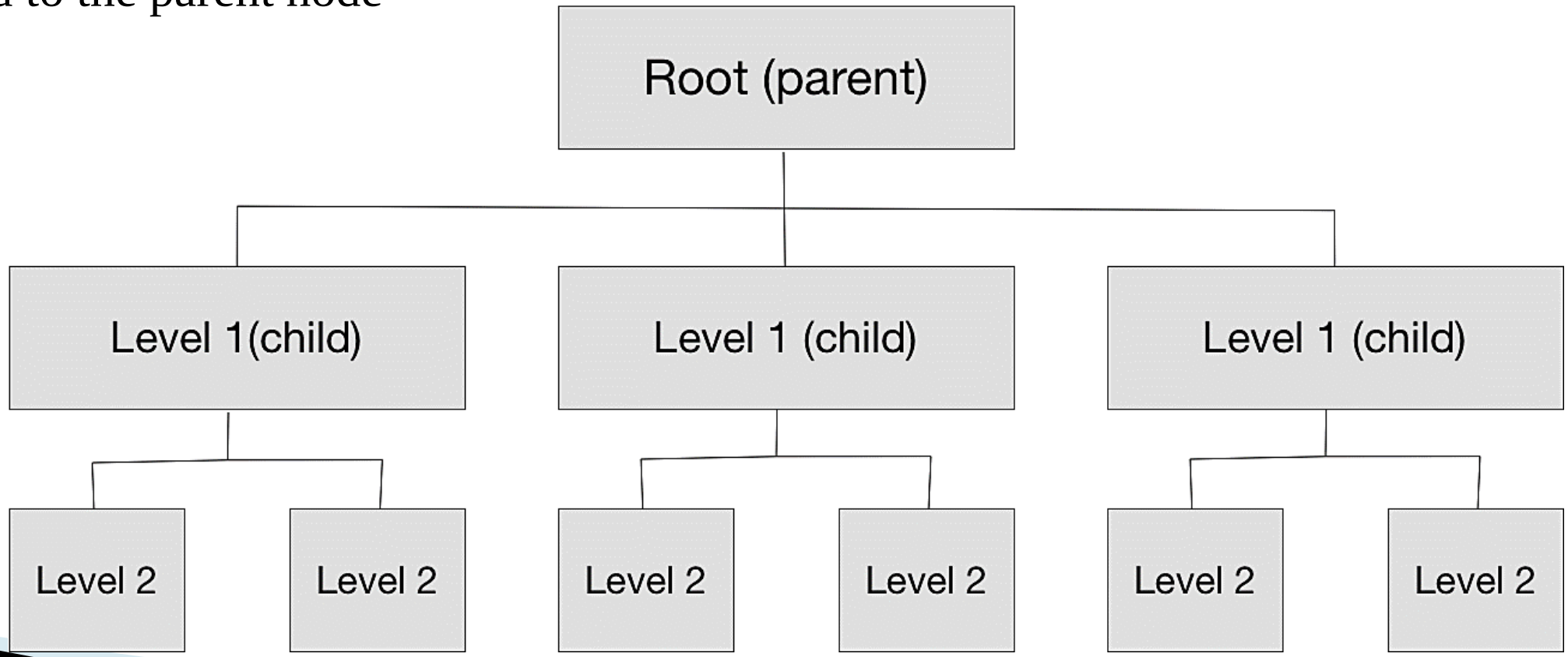
Data models – Relational Model

- In the relational model we can make the description of a collection of students more precise by specifying integrity constraints conditions that the records in a relation must satisfy
  - Example: Every student has a unique *sid* value

- Other Data Models
  - Hierarchical model
  - Network model
  - Object-oriented model
  - Object-relational model

Data models – Hierarchical Model

- Data is organized into a tree-like structure, implying a single parent for each record
- Hierarchy begins at the root, which contains root data, and then grows into a tree as child nodes are added to the parent node



Data models – Hierarchical Model

- Parent-Child Relationship
  - A parent node exists for each child node
  - A parent node might have several child nodes. It is not permitted to have more than one parent.
- One-to-many Relationship
  - The data is organized in a tree-like form, with the datatypes having a one-to-many relationship.
  - There can only be one path from any node to its parent.
- Deletion Problem
  - When a parent node is removed, the child node is removed as well.
- Pointers
  - Pointers are used to connect the parent and child nodes and to traverse and navigate between the stored data.

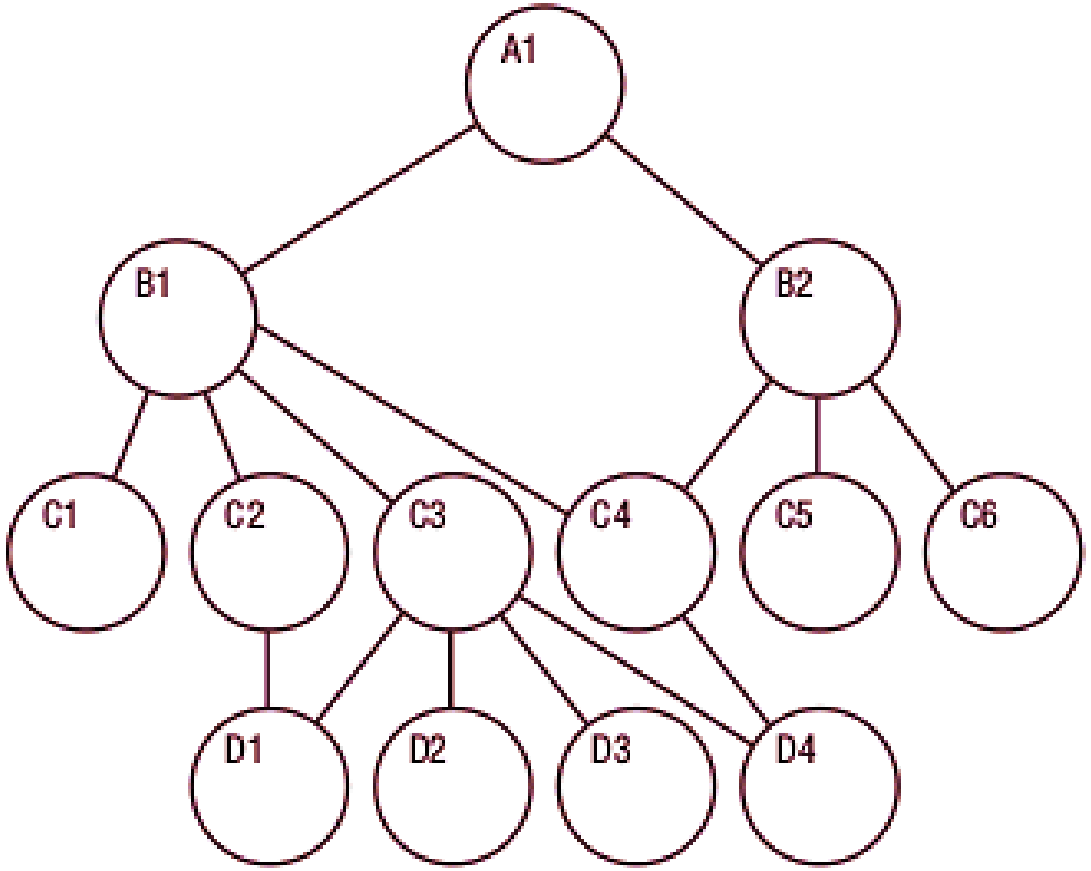
Data models – Hierarchical Model

- Advantages
  - A tree-like structure is incredibly straightforward and quick to navigate.
  - Any modification to the parent node is reflected automatically in the child node, ensuring data integrity.

- Disadvantages
  - Relationships that are complex are not supported
  - Because it only supports one parent per child node, if we have a complex relationship in which a child node needs to have two parents, we won't be able to describe it using this model
  - When a parent node is removed, the child node is removed as well.

Data models – Network Model

- The main difference between Network model and the hierarchical model is that any record can have several parents.
- It uses a graph instead of a hierarchical tree.



Data models – Network Model

- Multiple Paths
  - There may be several paths to the same record due to the increased number of relationships
  - It allows for quick and easy data access

- The Ability to Merge More Relationships
  - Data is more connected in this model since there are more relationships
  - Network model can handle many-to-many as well as one-to-one relationships

- Graph
  - The Graph is used to perform operations on the network model
  - The present position is kept up to date with the help of a software, and it navigates through the records based on the relationship

Data models – Network Model

- Advantages
  - In comparison to the hierarchical model, data can be retrieved faster - because the data in the network model is more related, and there may be more than one path to a given node
  - Data integrity is present since there is a parent-child relationship. Any changes to the parent record are mirrored in the child record.

- Disadvantages
  - As the number of relationships to be managed grows, the system may get increasingly complicated.
  - To operate with the model, a user must have a thorough understanding of it
  - Any alteration, such as an update, deletion, or insertion, is extremely difficult.

Data models – Object Oriented Data Model

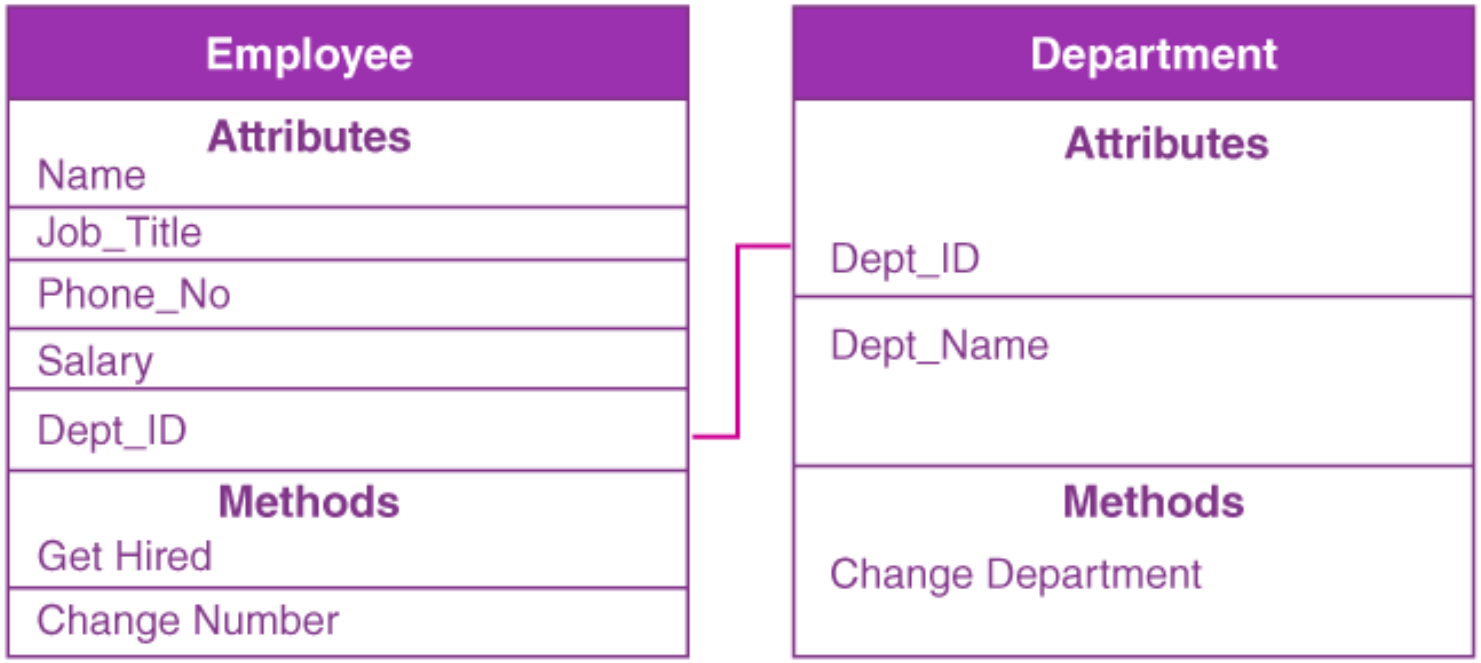
- Object Oriented Data Model (OODM) is a better representation of real-world challenges
- Both the data and the relationship are contained into a single structure - known as an object in this model
- We can now store audios, pictures, videos, and other types of data in databases
- An object is a representation of a physical entity – (entity of an ER model)
- An object's attributes are described by attributes
- A class is a group of objects that have similar structure (attributes) and behavior (methods)
- A class hierarchy is used to organize classes.
- The class hierarchy in some ways resembles an upside-down tree with only one parent for each class.
- The CUSTOMER and EMPLOYEE classes have the same parent PERSON class.

Data models – Object Oriented Data Model

- Inheritance refers to an object's capacity to inherit the characteristics and methods of those classes that are above it in the class hierarchy
- Two classes can be built as subclasses of the
  - PERSON class: CUSTOMER and EMPLOYEE.
  - EMPLOYEE and CUSTOMER will inherit all properties and methods from PERSON in this situation.

Data models – Object Oriented Data Model

- Employee and Department are the two objects in the example



Object Oriented Model

Data models – Object-Relational Data Model

- It combines the relational and Object-Oriented models
- This model was developed to bridge the gap between the Object-Oriented and relational models
- Many additional capabilities are available, such as the ability to create complicated data types based on our requirements utilizing existing data types
- The issue with ORDM is that it can become overly complicated and difficult to manage
- A thorough comprehension of the model is essential

Data models – Object-Relational Data Model

- Advantages
- Inheritance
  - Users of this data model can inherit objects, tables, and other data to expand their capability
  - Inherited objects have new properties in addition to those that were inherited

- Complex Data Types
  - Existing data types can be used to create complex data types
  - This is useful in the Object-Relational data model because complex data types allow for more effective data manipulation

- Extensibility
  - In the Object-Relational data model, the system's capability can be expanded
  - This can be accomplished through the use of complicated data types and advanced Object-Oriented concepts such as inheritance

Data models – Object-Relational Data Model

- Disadvantages
- Because it combines the features of both the Object-Oriented data model and the Relational data model, this data model can be rather complicated and hard to handle at times.

Evolution of Data Models

Evolution of Major Data Models

GENERATION	TIME	DATA MODEL	EXAMPLES	COMMENTS
First	1960s–1970s	File system	VMS/VSAM	Used mainly on IBM mainframe systems Managed records, not relationships
Second	1970s	Hierarchical and network	IMS, ADABAS, IDS-II	Early database systems Navigational access
Third	Mid-1970s	Relational	DB2 Oracle MS SQL Server MySQL	Conceptual simplicity Entity relationship (ER) modeling and support for relational data modeling
Fourth	Mid-1980s	Object-oriented Object/ relational (O/R)	Versant Objectivity/DB DB2 UDB Oracle 11g	Object/relational supports object data types Star Schema support for data warehousing Web databases become common
Fifth	Mid-1990s	XML Hybrid DBMS	dbXML Tamino DB2 UDB Oracle 11g MS SQL Server	Unstructured data support O/R model supports XML documents Hybrid DBMS adds object front end to relational databases Support large databases (terabyte size)
Emerging Models: NoSQL	Late 2000s to present	Key-value store Column store	SimpleDB (Amazon) BigTable (Google) Cassandra (Apache)	Distributed, highly scalable High performance, fault tolerant Very large storage (petabytes) Suited for sparse data Proprietary API