UNIT: 9 Fundamentals of Database

Contents:

- Introduction to DBMS,
- DBMS Models, Database Architecture,
- Database application.
- Database Design and Data Security,
- Data Warehouse and Data Mining, Big Data
- Database Administrator

What is a database?

- Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.
- A database is an organized collection of structured information, or data, typically stored electronically in a computer system.
- A database is usually controlled by a database management system (DBMS).
- A database is information that is set up for easy access, management and updating.
- Computer databases typically store aggregations of data records or files that contain information, such as sales transactions, customer data, financials and product information.
- Databases are used for storing, maintaining and accessing any sort of data.

What are databases used for?

- **Improve business processes.** Companies collect data about business processes, such sales, order processing and customer service.
- They analyze that data to improve these processes, expand their business and grow revenue.
- **Keep track of customers.** Databases often store information about people, such as customers or users.
- For example, social media platforms use databases to store user information, such as names, email addresses and user behavior. The data is used to recommend content to users and improve the user experience.
- **Secure personal health information.** Healthcare providers use databases to securely store personal health data to inform and improve patient care.
- **Store personal data.** Databases can also be used to store personal information. For example, personal cloud storage is available for individual users to store media, such as photos, in a managed cloud.

What is the database management system?

- Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data.
- A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.
- Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data.
- This helps provide data security, data integrity, concurrency, and uniform data administration procedures.
- DBMS optimizes the organization of data by following a database schema design technique called normalization, which splits a large table into smaller tables when any of its attributes have redundancy in values.
- DBMS offer many benefits over traditional file systems, including flexibility and a more complex backup system.

Characteristics of DBMS

Real-world entity:

- A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.

Relation-based tables:

- DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.

Isolation of data and application:

- A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes.
- DBMS also stores metadata, which is data about data, to ease its own process.

Less redundancy:

- DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values.
- Normalization is a mathematically rich and scientific process that reduces data redundancy.

Characteristics of DBMS Cont…

Consistency:

- Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.
- A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.

Query Language:

- DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used

ACID Properties:

- DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database.
- ACID properties help the database stay healthy in multi-transactional environments and in case of failure.

Characteristics of DBMS Cont…

Multiuser and Concurrent Access:

- DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.

Multiple views:

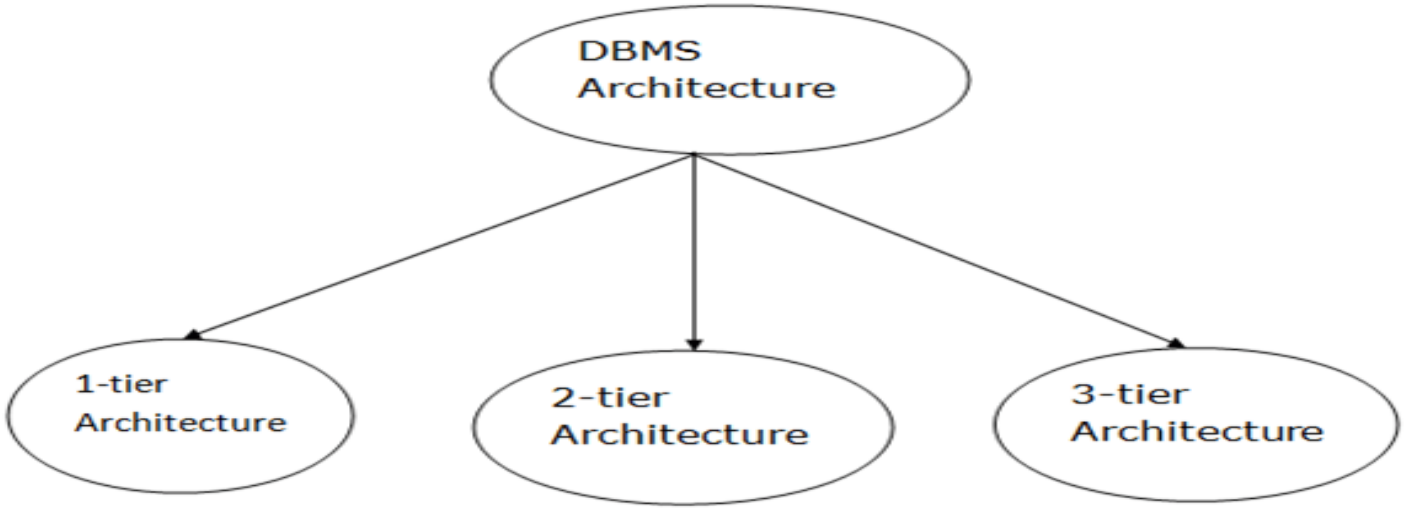
- DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department.
- This feature enables the users to have a concentrate view of the database according to their requirements.

Security:

- Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage.
- DBMS offers many different levels of security features, which enables multiple users to have different views with different features

DBMS Architecture

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.



DBMS Architecture Cont..

1-Tier Architecture

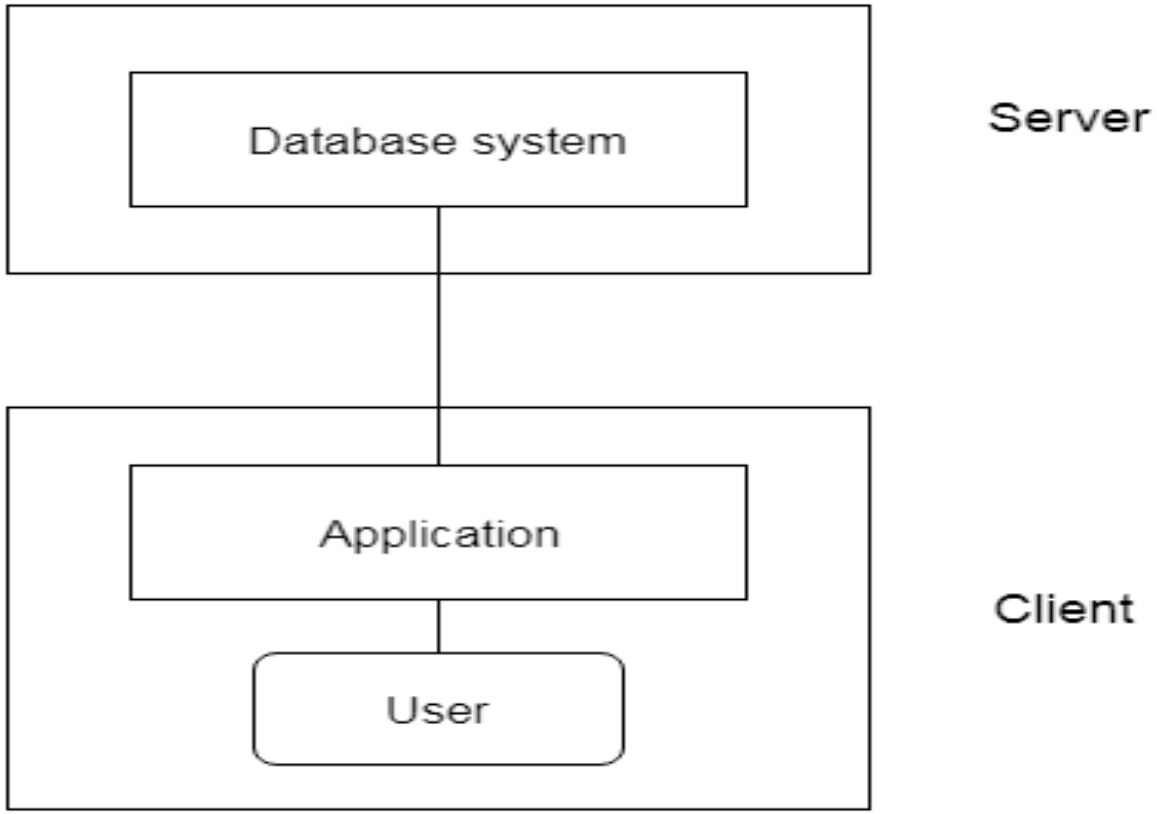
- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture

- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

DBMS Architecture Cont..

Fig: 2-tier Architecture



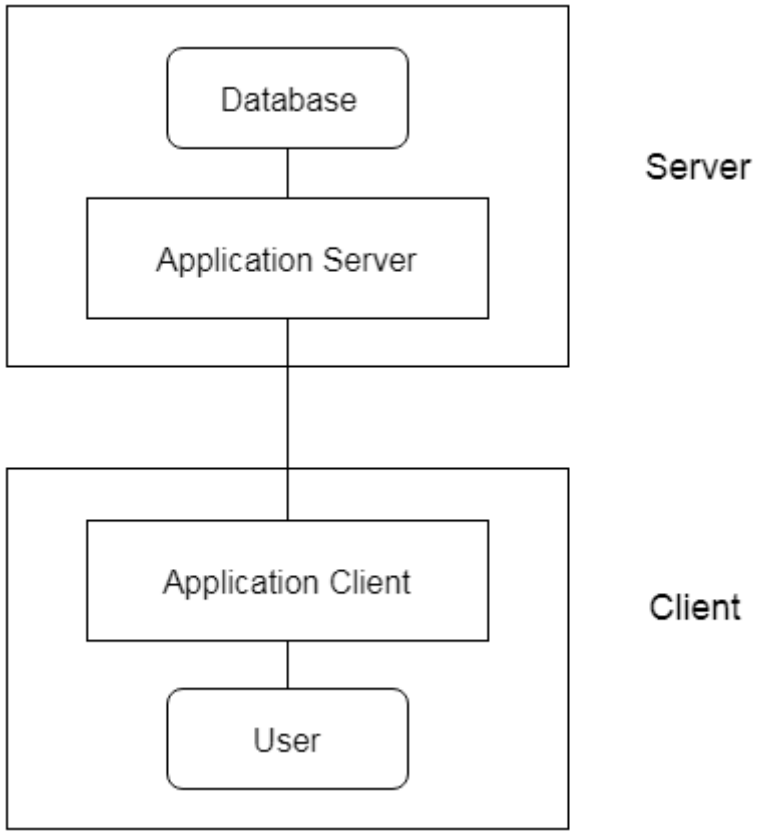
DBMS Architecture Cont..

3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of large web application.

DBMS Architecture Cont..

Fig: 3-tier Architecture



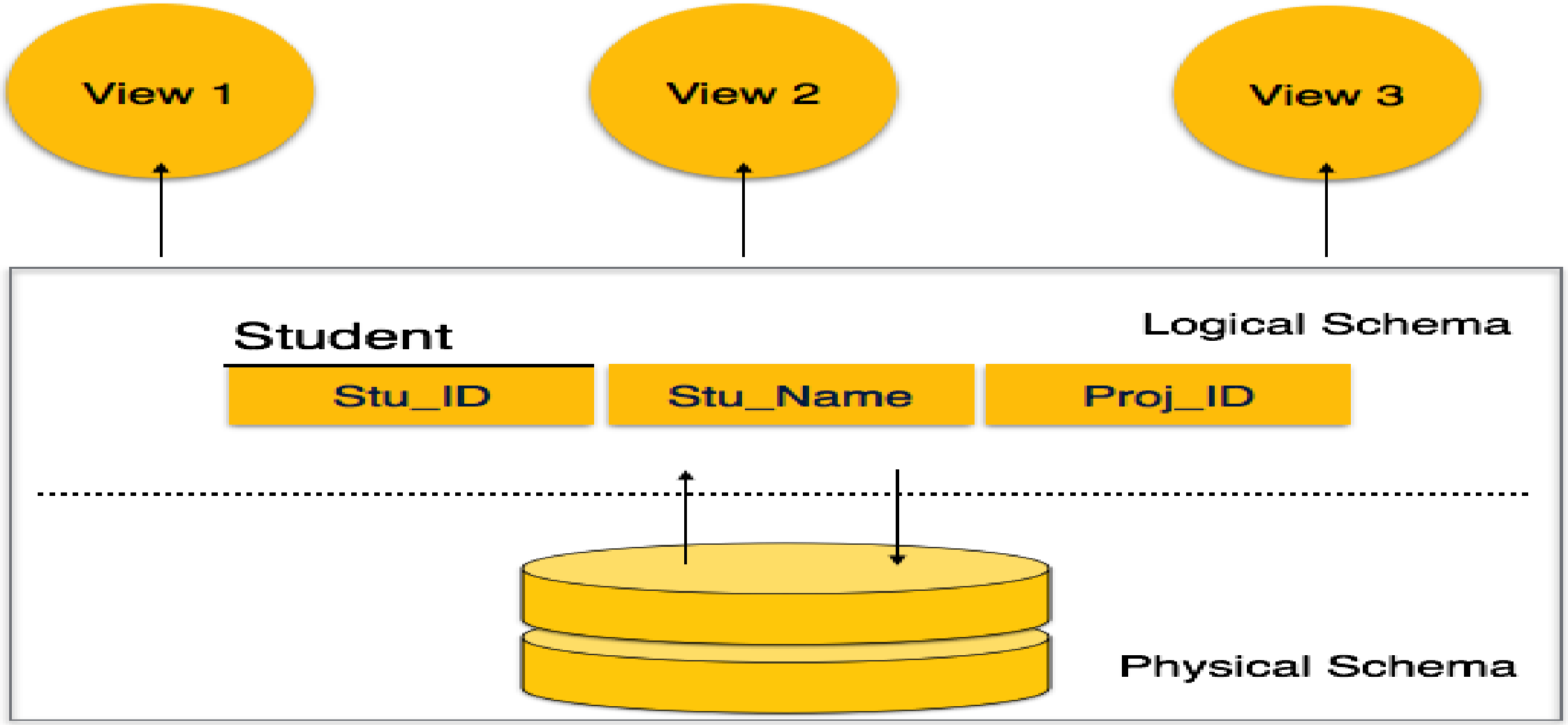
What is SQL?

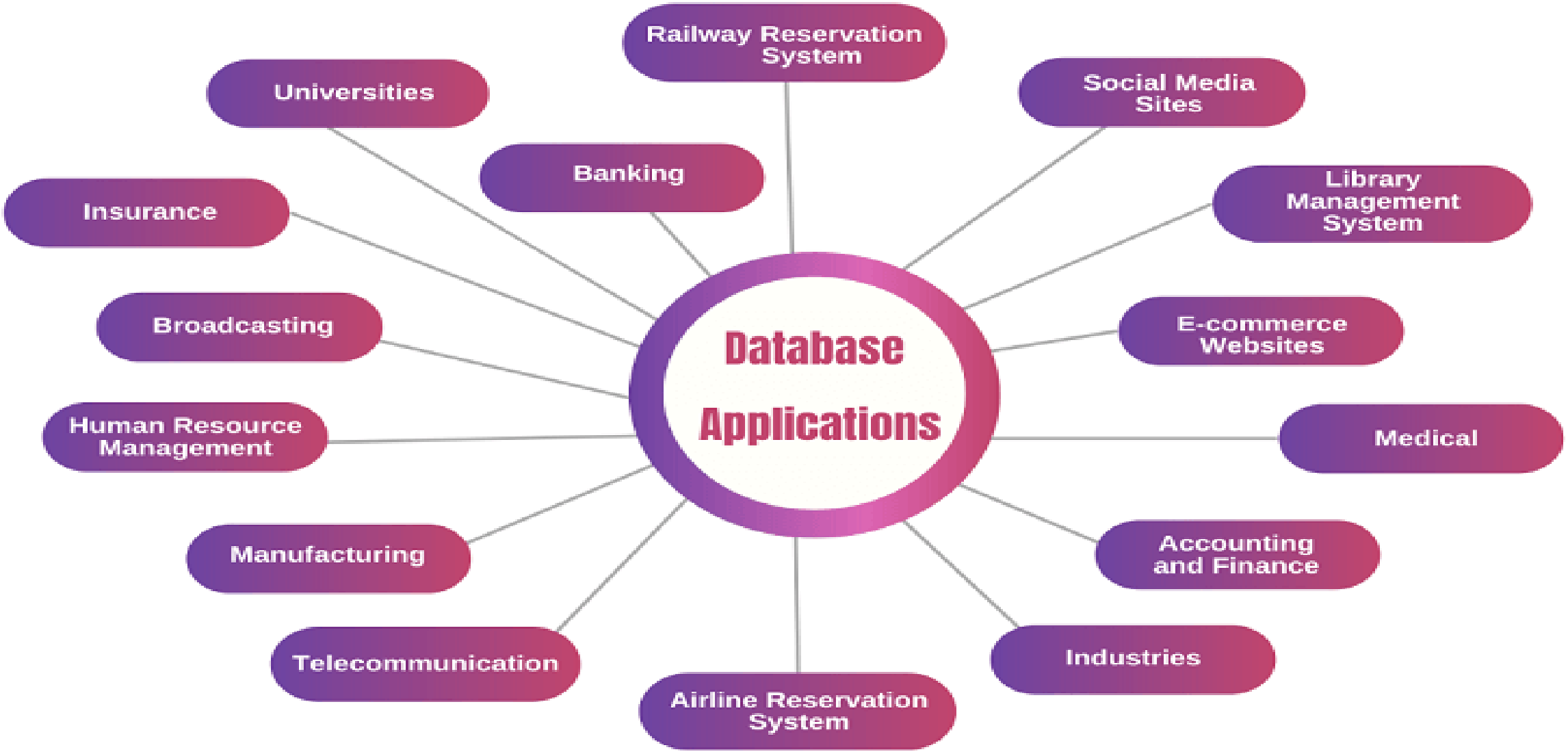
- Structured query language (SQL) is a programming language for storing and processing information in a relational database.
- A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values.
- SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database.
- SQL is the standard language for Relation Database System.
- You can use SQL statements to store, update, remove, search, and retrieve information from the database.
- **The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP.**

Database Schema

- A database schema refers to the logical and visual configuration of the entire relational database.
- ***A database schema can be divided broadly into three categories*** **−**
- **Conceptual schema-** It describes the structure of the whole database. The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data. In the conceptual level, internal details such as an implementation of the data structure are hidden.
- **Physical Database Schema** − This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- **Logical Database Schema** − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Database Schema Cont..




Database Applications
Railway Reservation System
Universities
Social Media Sites
Banking
Insurance
Library Management System
Broadcasting
E-commerce Websites
Human Resource Management
Medical
Manufacturing
Accounting and Finance
Telecommunication
Industries
Airline Reservation System

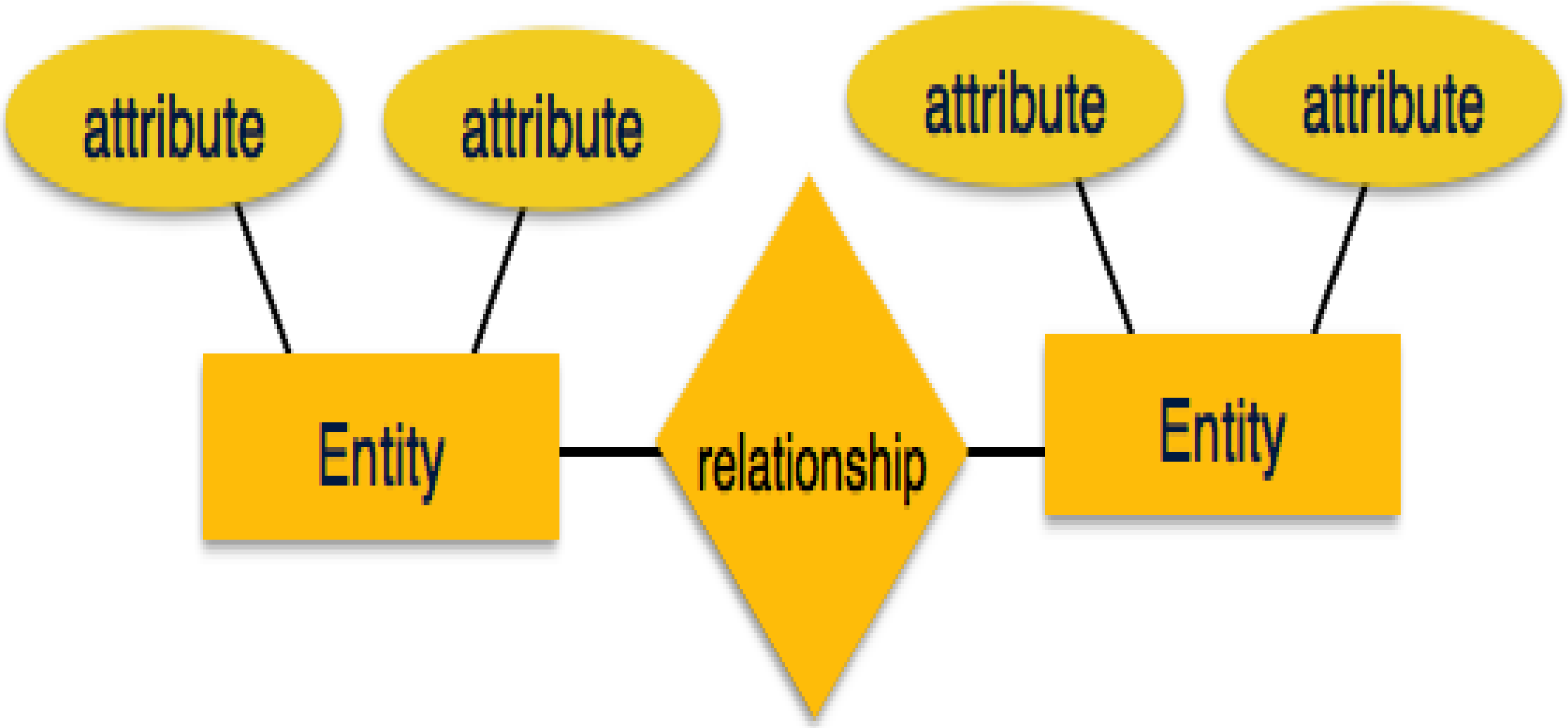
Database Models

- Data models define how the logical structure of a database is modeled.
- Data Models are fundamental entities to introduce abstraction in a DBMS.
- Data models define how data is connected to each other and how they are processed and stored inside the system.

1. Entity-Relationship Model

- Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them.
- While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes, and constraints.
- ER model stands for an Entity-Relationship model. It is a high-level data model.
- This model is used to define the data elements and relationship for a specified system.

The E-R Model Cont…



The E-R Model Cont…

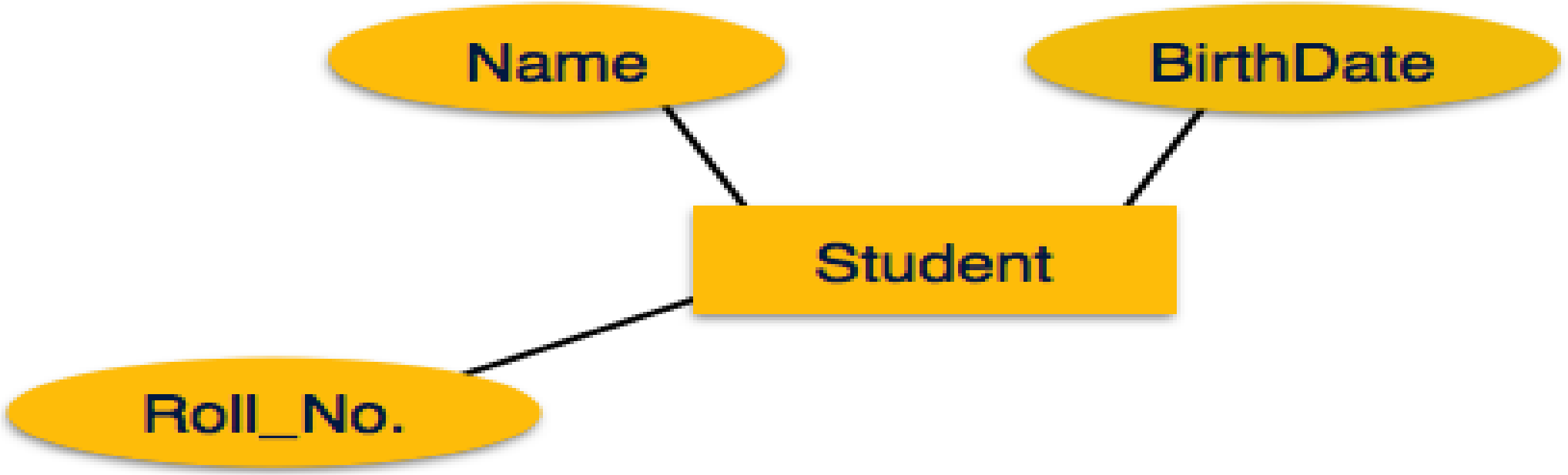
- ER Model is best used for the conceptual design of a database.
- ER Model is based on:
  - **Entities and their attributes**
- An entity in an ER Model is a real-world entity having properties called attributes.
- Every attribute is defined by its set of values called domain
- For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- Relationships among entities The logical association among entities is called relationship.

Entity

- Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

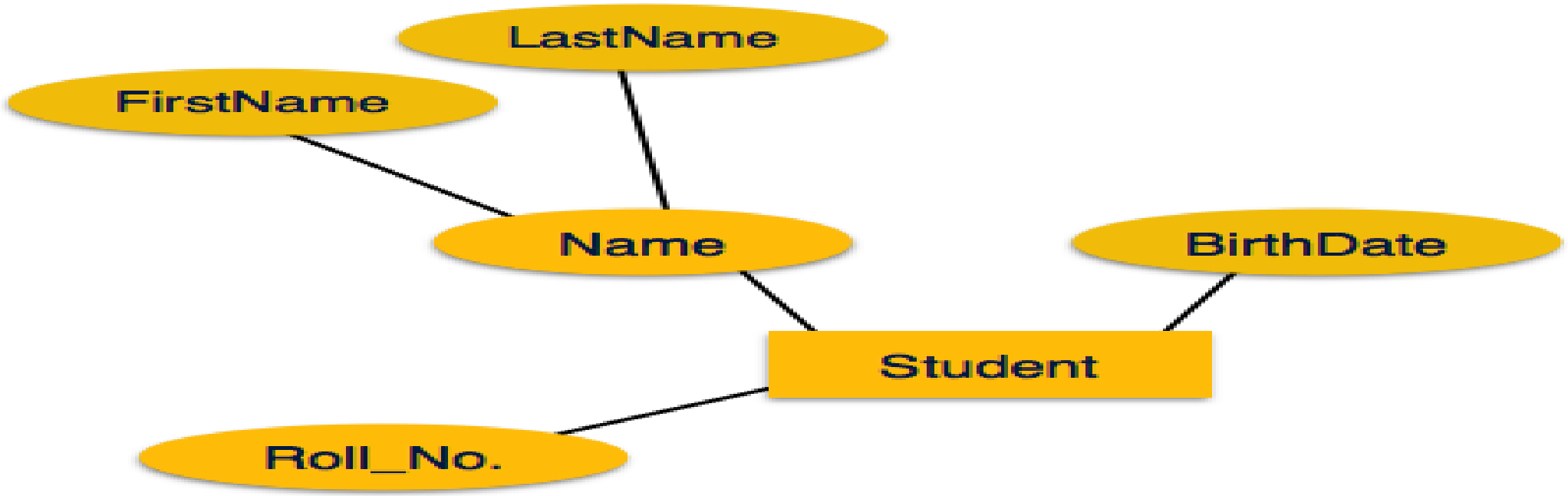
Attributes

- Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



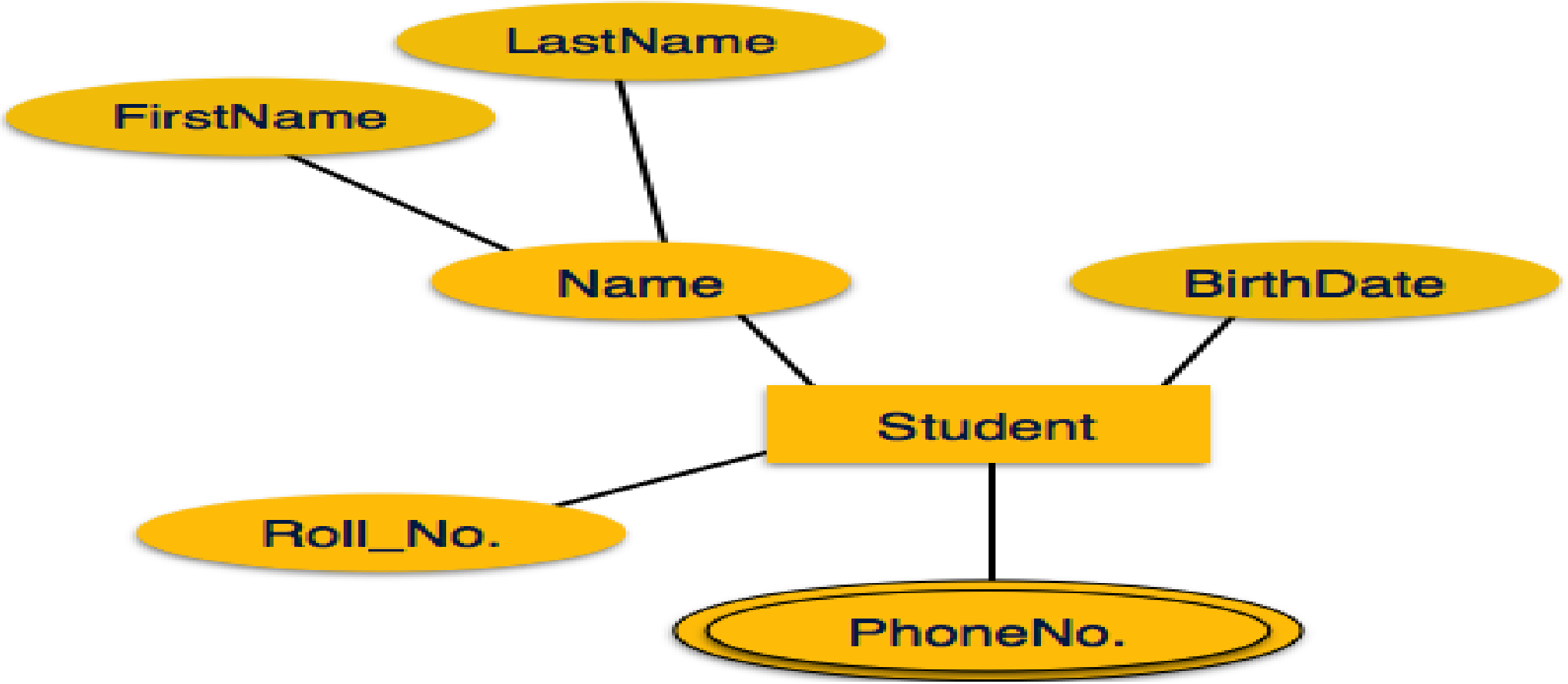
Attributes Cont..

- If the attributes are **composite**, they are further divided in a tree like structure.
- Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.



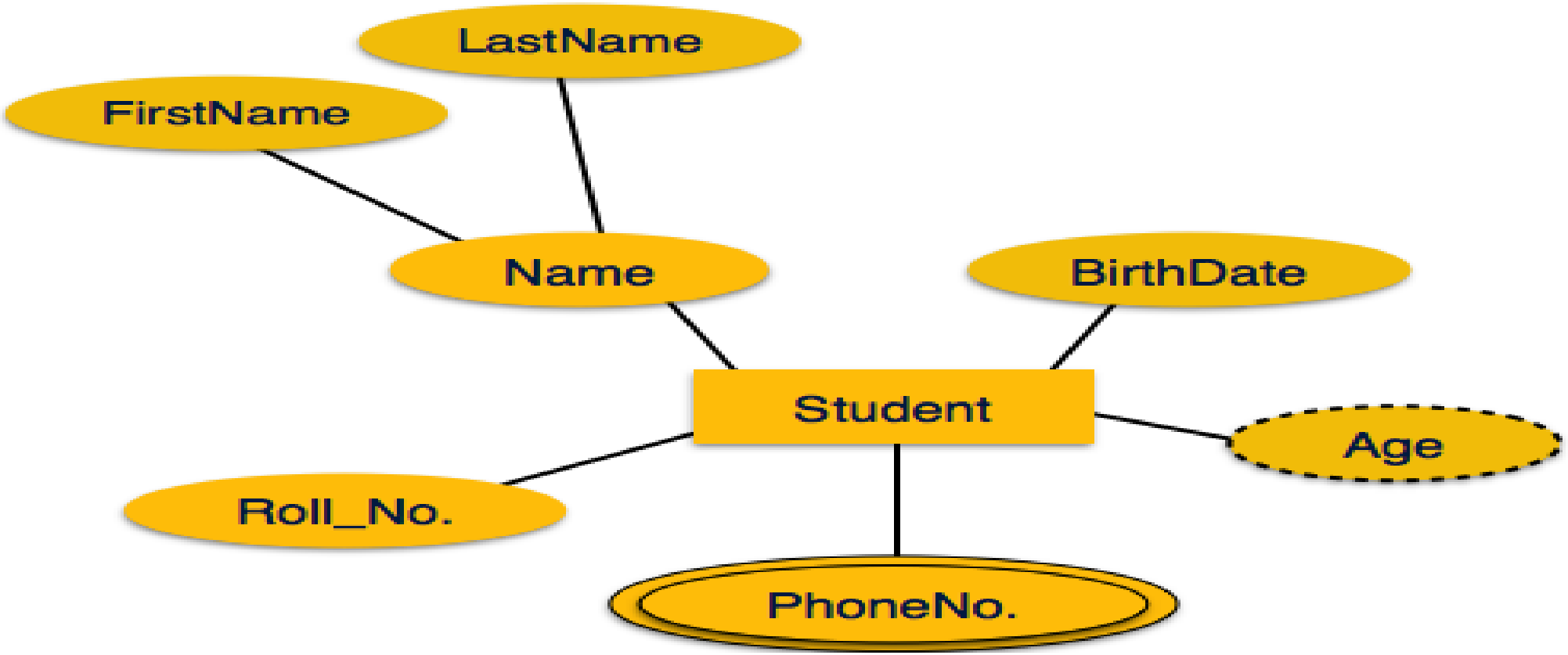
Attributes Cont..

- **Multivalued** attributes are depicted by double ellipse.



Attributes Cont..

- **Derived** attributes are depicted by dashed ellipse

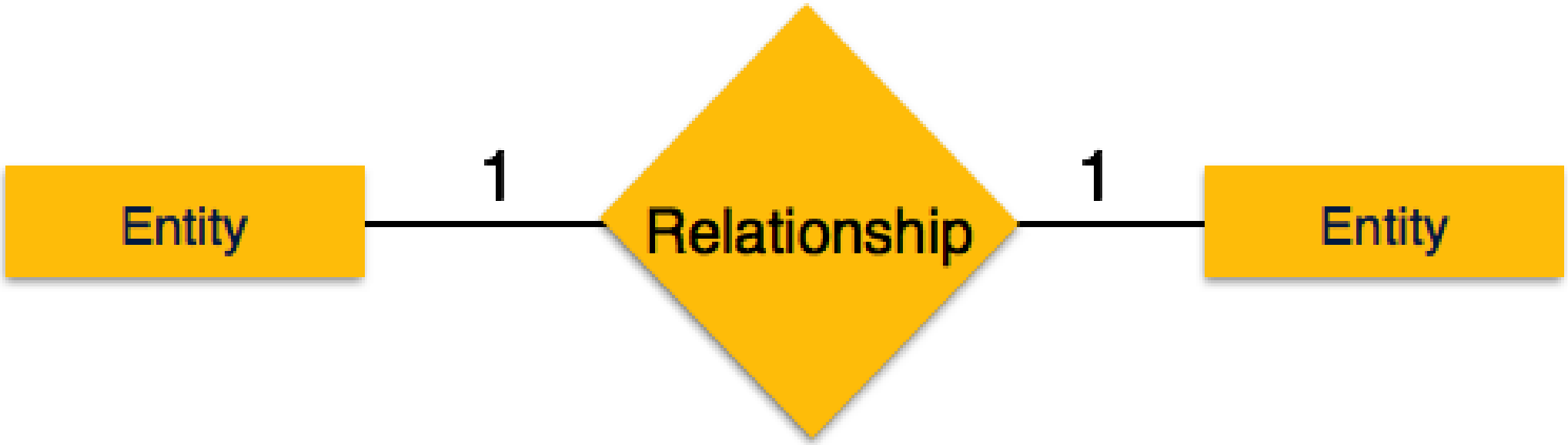


Relationship

- Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box.
- All the entities (rectangles) participating in a relationship, are connected to it by a line.
- Binary Relationship and Cardinality
- A relationship where two entities are participating is called a **binary relationship**.
- Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

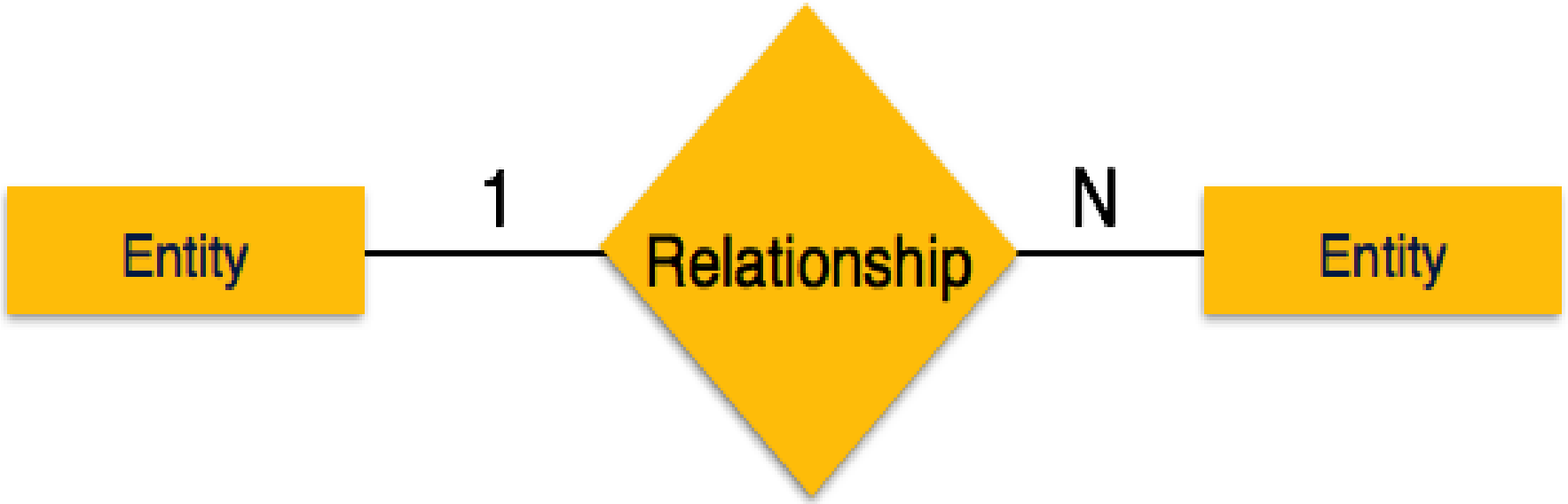
Relationship Cont…

One-to-one − When only one instance of an entity is associated with the relationship, it is marked as '1:1'.
The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



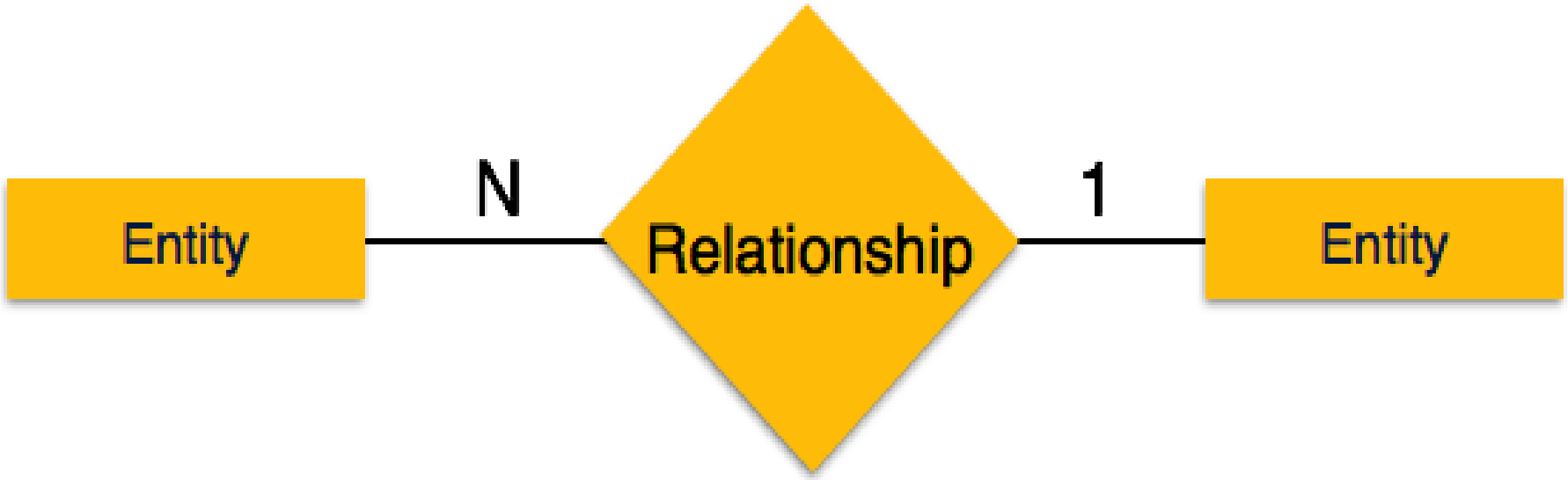
Relationship Cont…

- **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.
- The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship



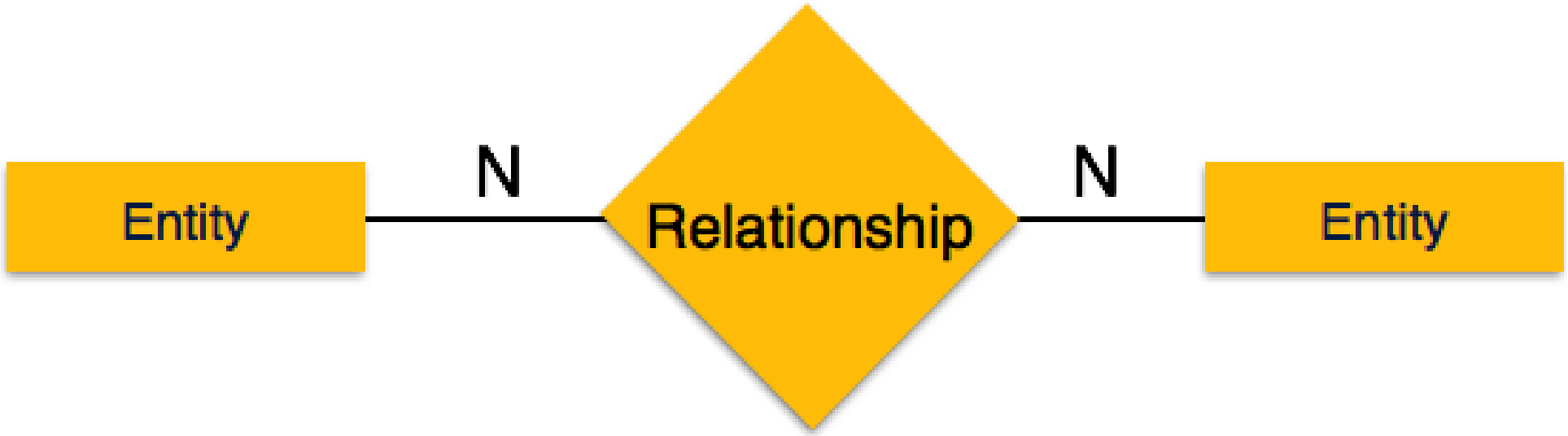
Relationship Cont…

- **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'.
- The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.



Relationship Cont…

- **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



Database Models Cont..

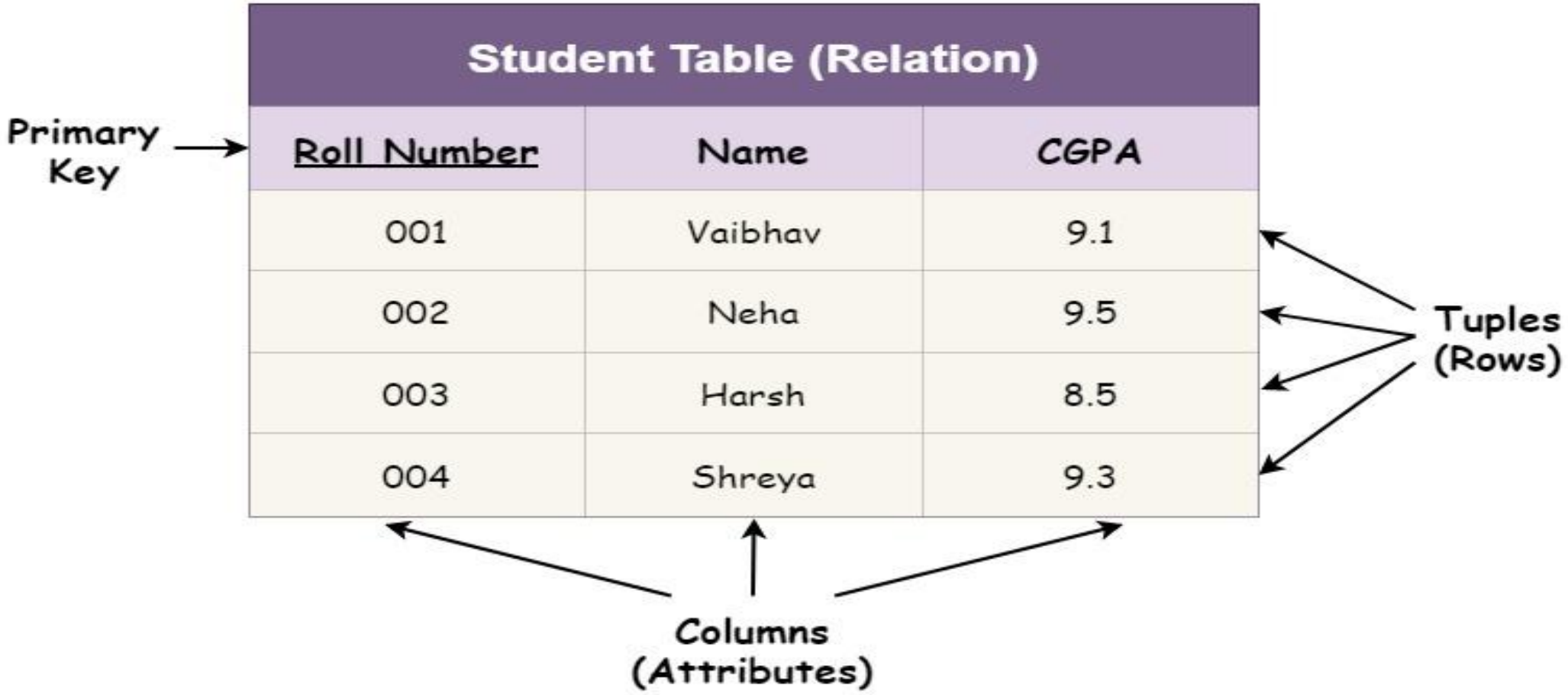
2. Relational Model

- What is the Relational Model? The relational model represents how data is stored in Relational Databases. A relational database consists of a collection of tables, each of which is assigned a unique name. Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE, and AGE shown in the table.
- The most popular data model in DBMS is the Relational Model. It is more scientific model than others.
- This model is based on first-order predicate logic and defines a table as an n-ary relation
- **Relational Model** (RM) represents the database as a collection of relations. A **relation** is nothing but a table of values.
- Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship. ... The data are represented as a set of relations
- In **relational model**, the **data** and relationships are represented by collection of inter-related tables.
- Each table is a group of column and rows, where column represents attribute of an entity and rows represents records

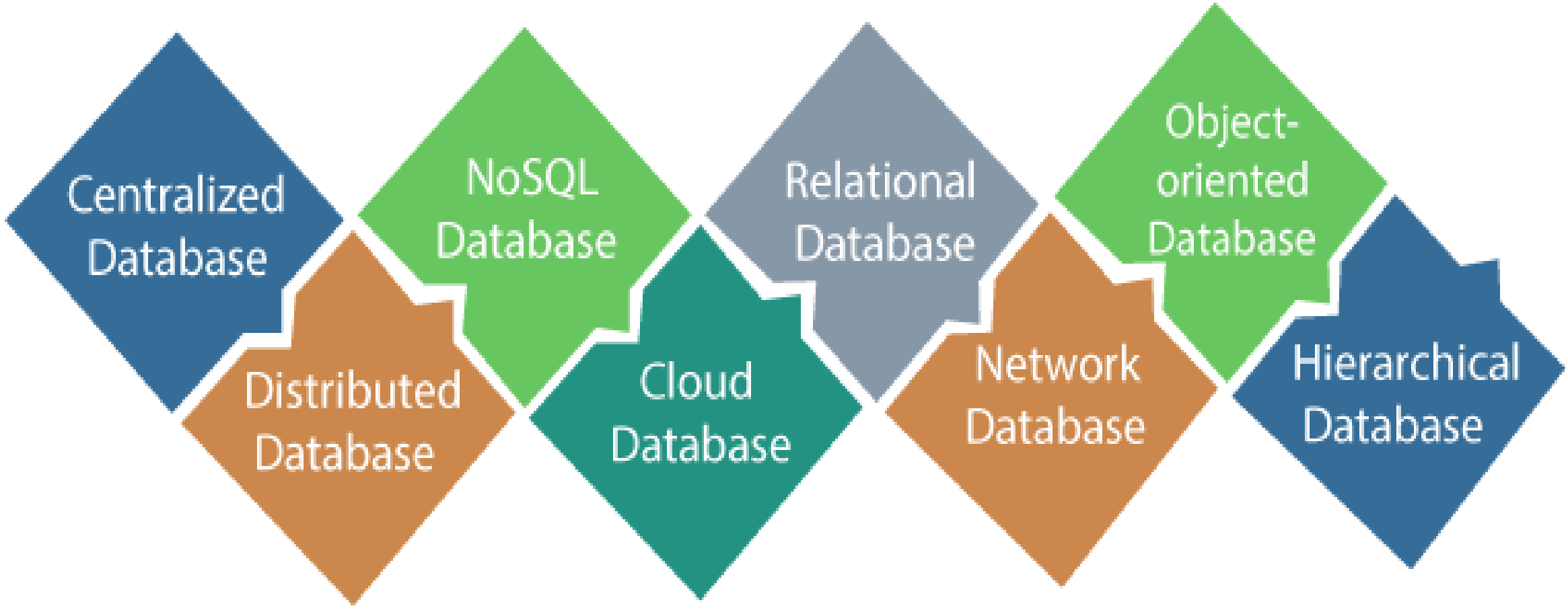
Database Concepts Cont…

Student				
Rol_No	Name	Address	Phone	Age
1	Ram	Delhi	9455123451	18
2	Ramesh	Gurgaon	9652431543	18
3	Sujit	Rohtak	9156253131	20
4	Suresh	Delhi	9156768971	18
3	Sujit	Rohtak	9156253131	20
2	Ramesh	Gurgaon	9652431543	18

Relational Model in DBMS



Types of Database



Types of Databases Cont..

1) Centralized Database

- It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. These applications contain the authentication process to let users access data securely. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.

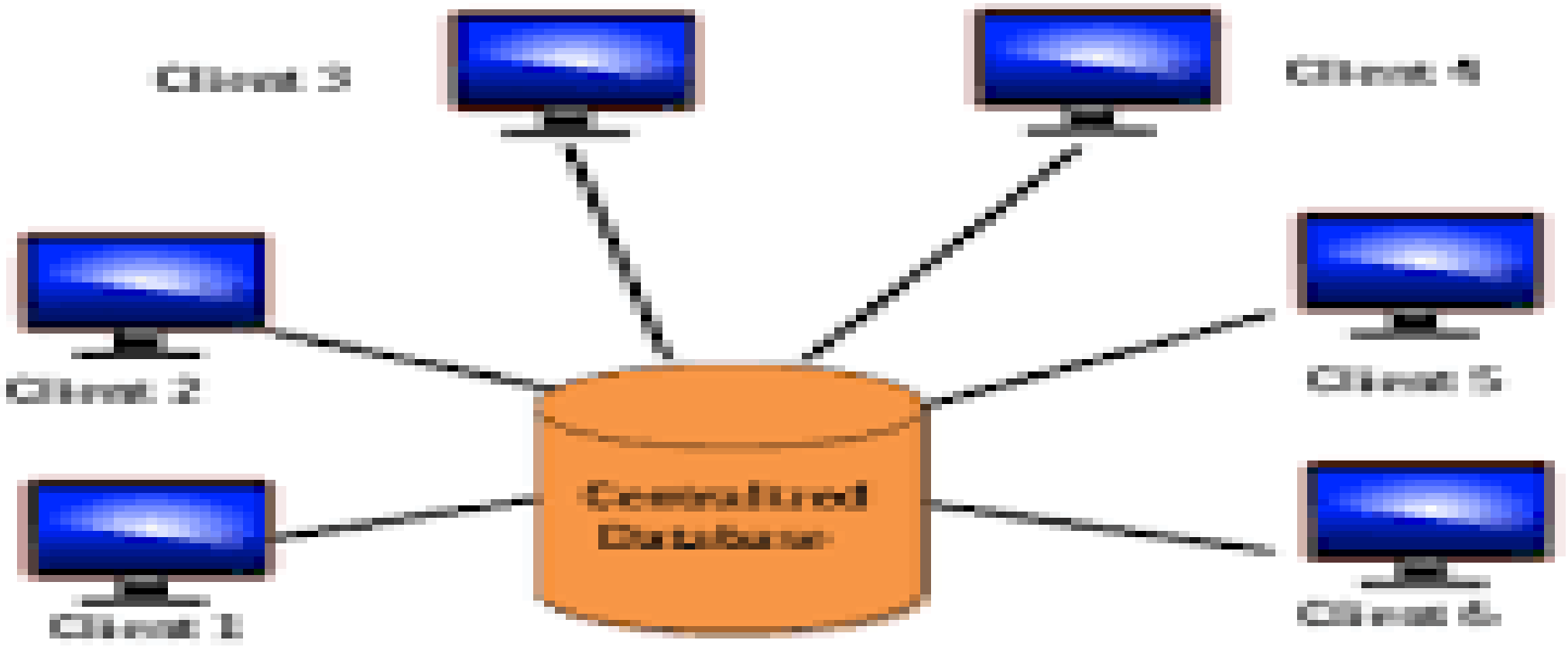
Advantages of Centralized Database

- It has decreased the risk of data management, i.e., manipulation of data will not affect the core data.
- Data consistency is maintained as it manages data in a central repository.
- It provides better data quality, which enables organizations to establish data standards.
- It is less costly because fewer vendors are required to handle the data sets.

Disadvantages of Centralized Database

- The size of the centralized database is large, which increases the response time for fetching the data.
- It is not easy to update such an extensive database system.
- If any server failure occurs, entire data will be lost, which could be a huge loss.

Centralized Database



Types of Databases Cont..

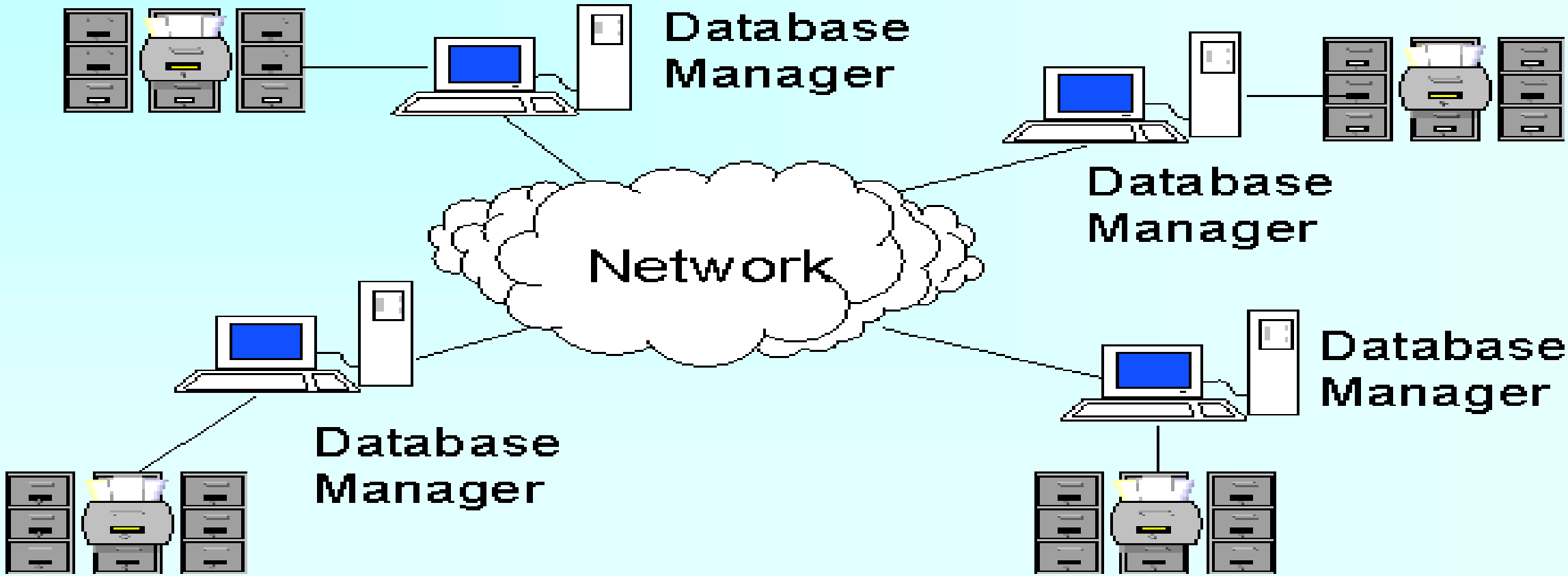
2) Distributed Database

- A distributed database is a database that runs and stores data across multiple computers, as opposed to doing everything on a single machine.
- Typically, distributed databases operate on two or more interconnected servers on a computer network.

Advantages of Distributed Database

- Modular development is possible in a distributed database, i.e., the system can be expanded by including new computers and connecting them to the distributed system.
- One server failure will not affect the entire data set.

A Distributed Database

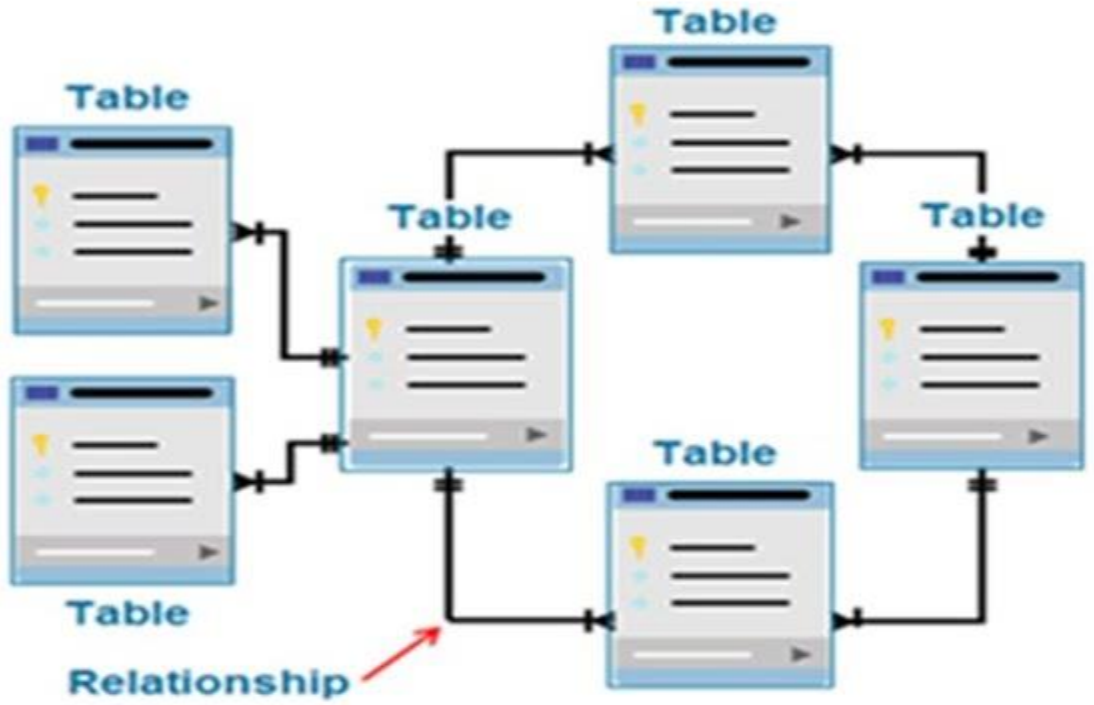


Yair Amir & Jonathan Stanton Fall 00/ Lecture 8 9

Types of Databases Cont..

3) Relational Database

- This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation).
- A relational database uses SQL for storing, manipulating, as well as maintaining the data.
- **Examples** of Relational databases are MySQL, Microsoft SQL Server, Oracle, etc.



RDBMS

Relational Databases

Relational Database Management System

www.learncomputerscienceonline.com

Types of Databases Cont..

Properties of Relational Database

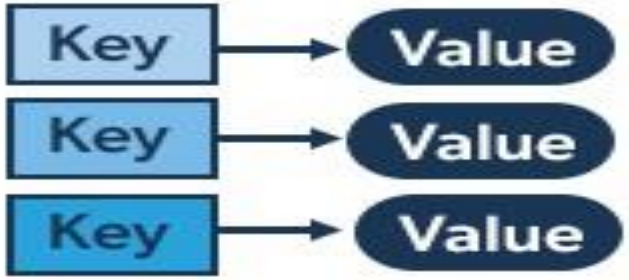
- There are following four commonly known properties of a relational model known as ACID properties, where:
- **A means Atomicity:** This ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy. For example, a transaction will either be committed or will abort.
- **C means Consistency:** If we perform any operation over the data, its value before and after the operation should be preserved. For example, the account balance before and after the transaction should be correct, i.e., it should remain conserved.
- **I means Isolation:** There can be concurrent users for accessing data at the same time from the database. Thus, isolation between the data should remain isolated. For example, when multiple transactions occur at the same time, one transaction effects should not be visible to the other transactions in the database.
- **D means Durability:** It ensures that once it completes the operation and commits the data, data changes should remain permanent.

Types of Databases Cont..

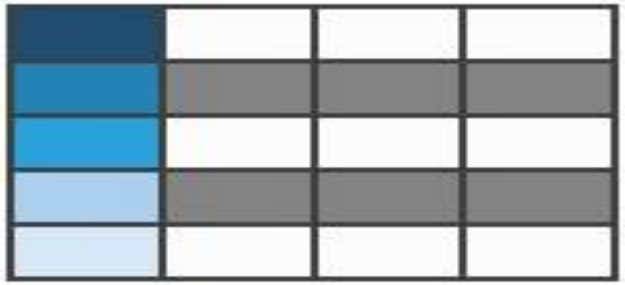
- **4) NoSQL Database**
- NoSQL databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables.
- NoSQL databases come in a variety of types based on their data model.
- The main types are document, key-value, wide-column, and graph.
- They provide flexible schemas and scale easily with large amounts of data and high user loads.

NoSQL

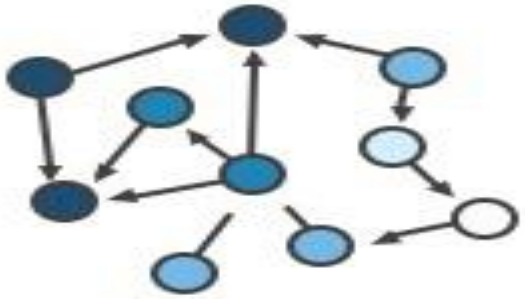
Key-Value



Column-Family



Graph



Document

Types of Databases Cont..

4) NoSQL Database Cont.

- Thus, NoSQL presented a wide variety of database technologies in response to the demands. We can further divide a NoSQL database into the following four types:

1. **Key-value storage:** It is the simplest type of database storage where it stores every single item as a key (or attribute name) holding its value, together.
2. **Document-oriented Database:** A type of database used to store data as JSON-like document. It helps developers in storing data by using the same document-model format as used in the application code.
3. **Graph Databases:** It is used for storing vast amounts of data in a graph-like structure. Most commonly, social networking websites use the graph database.
4. **Wide-column stores:** It is similar to the data represented in relational databases. Here, data is stored in large columns together, instead of storing in rows.

Types of Databases Cont..

- **Advantages of NoSQL Database**
- It enables good productivity in the application development as it is not required to store data in a structured format.
- It is a better option for managing and handling large data sets.
- It provides high scalability.
- Users can quickly access data from the database through key-value.

Types of Databases Cont..

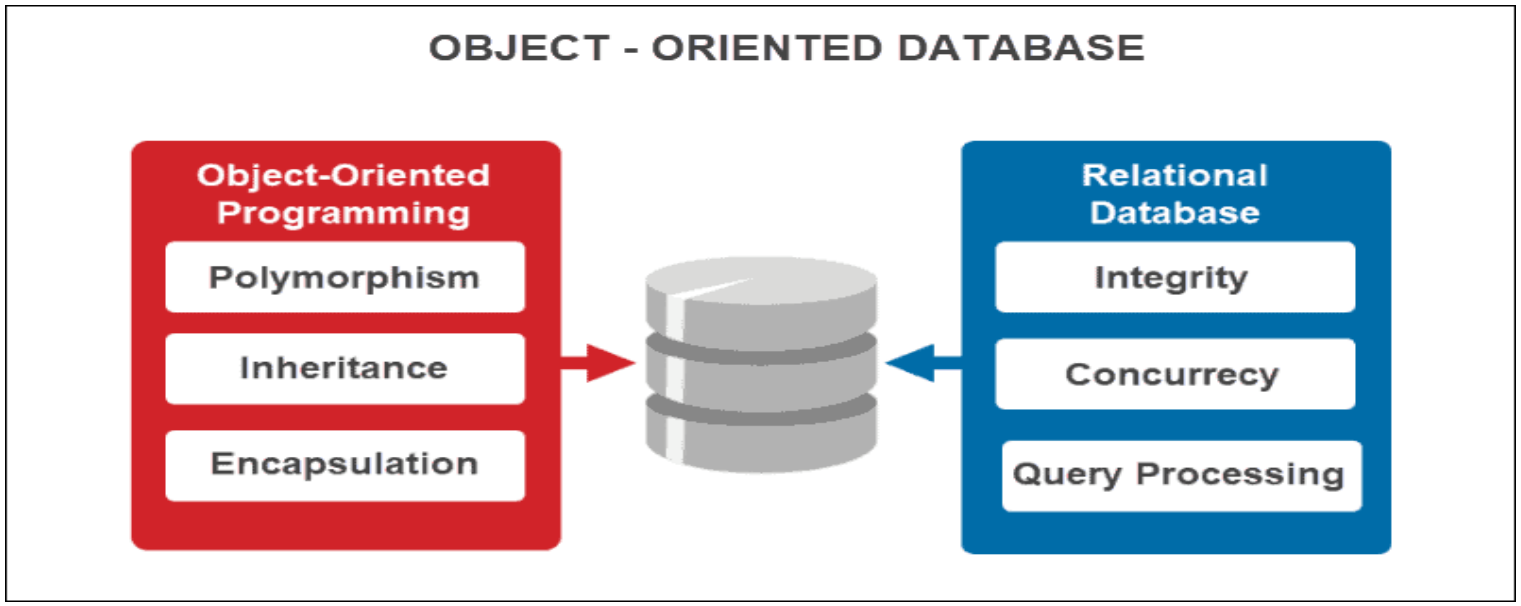
5) Cloud Database

- A type of database where data is stored in a virtual environment and executes over the cloud computing platform.
- It provides users with various cloud computing services for accessing the database.
- There are numerous cloud platforms, but the best options are:
- Amazon Web Services(AWS)
- Microsoft Azure
- Kamatera
- PhonixNAP
- ScienceSoft
- Google Cloud SQL, etc

Types of Databases Cont..

6) Object-oriented Databases

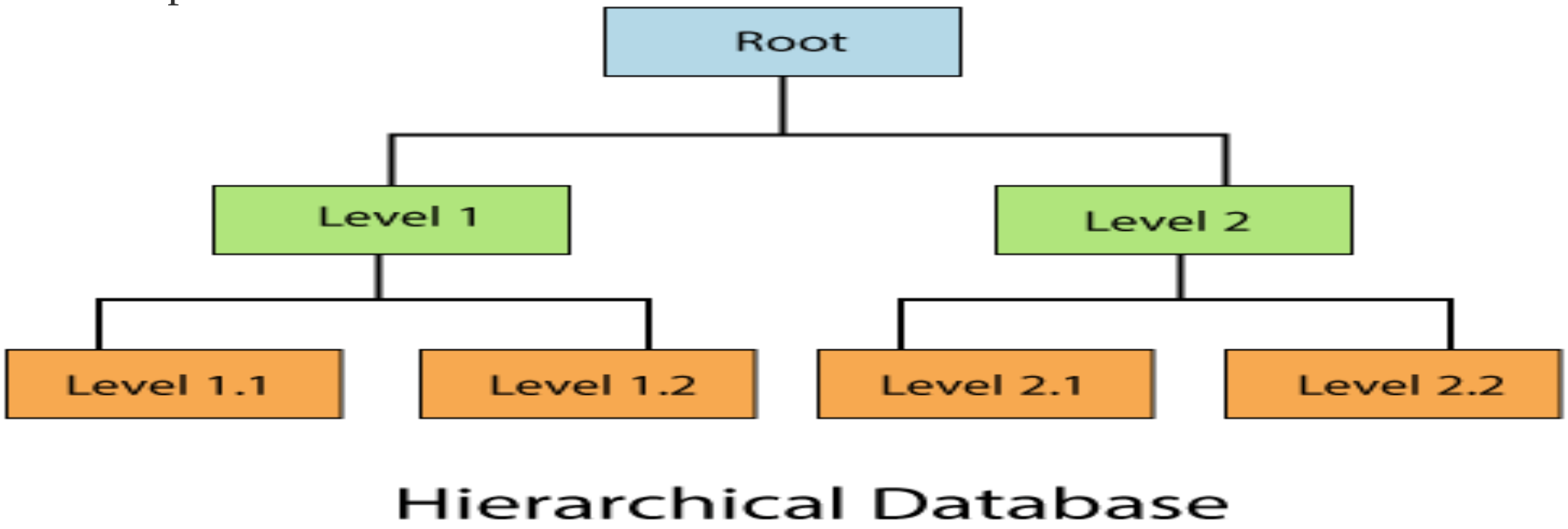
- The type of database that uses the object-based data model approach for storing data in the database system.
- The data is represented and stored as objects which are similar to the objects used in the object-oriented programming language.



Types of Databases Cont..

7) Hierarchical Databases

- It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure. Data get stored in the form of records that are connected via links.
- Each child record in the tree will contain only one parent. On the other hand, each parent record can have multiple child records.

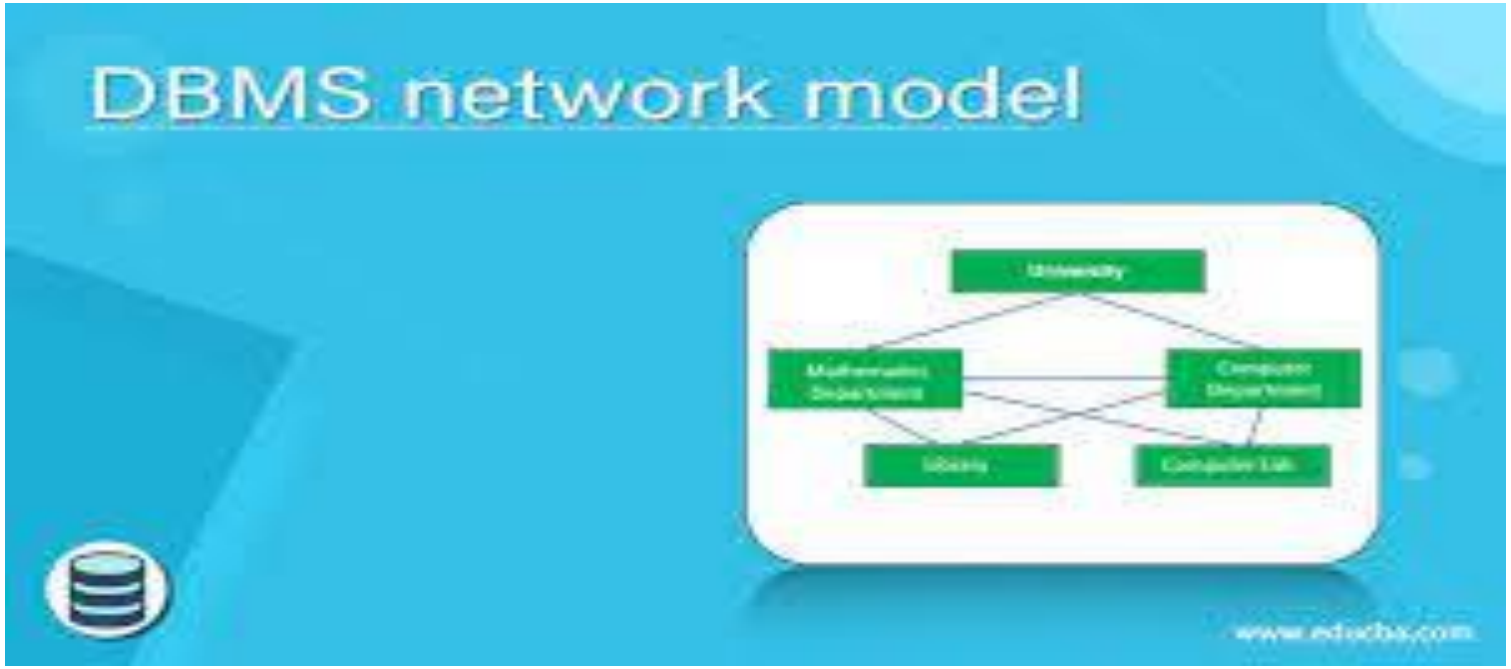


Hierarchical Database

Types of Databases Cont..

8) Network Databases

- It is the database that typically follows the network data model. Here, the representation of data is in the form of nodes connected via links between them.
- Unlike the hierarchical database, it allows each record to have multiple children and parent nodes to form a generalized graph structure.



Types of Databases Cont..

9) Personal Database

- Collecting and storing data on the user's system defines a Personal Database. This database is basically designed for a single user.

Advantage of Personal Database

- It is simple and easy to handle.
- It occupies less storage space as it is small in size.

10) Operational Database

- The type of database which creates and updates the database in real-time.
- It is basically designed for executing and handling the daily data operations in several businesses. F
- or example, An organization uses operational databases for managing per day transactions.

Types of Databases Cont..

11) Enterprise Database

- Large organizations or enterprises use this database for managing a massive amount of data. It helps organizations to increase and improve their efficiency. Such a database allows simultaneous access to users.
- **Advantages of Enterprise Database:**
- Multi processes are supportable over the Enterprise database.
- It allows executing parallel queries on the system.

Advantages of Database Management System

- **Better Data Transferring**
- **Better Data Security**
- **Better data integration.**
- **Minimized Data Inconsistency**
- **Faster data Access**
- **Better decision making**
- **Increased end-user**
- **Simple**
- **Data abstraction**
- **Reduction in data Redundancy**
- **Application development**
- **Data sharing**
- **Data organization**
- **The atomicity of data can be maintained**

Advantages of Database Management System Cont..

- *The DBMS allows concurrent access to multiple users by using the synchronization technique.*
- **Data consistency and accuracy**
- **Improved data security**
- **Efficient data access and retrieval**
- **Scalability and flexibility**
- **Improved productivity**

Advantages of Database Management System over Traditional File System

- **Data Integrity and Security**
- **Reduced Data Redundancy**
- **Improved Data Access and Availability**
- **Improved Data Sharing**
- **Improved Data Integration**
- **Improved Data Backup and Recovery**
- **Data sharing**
- **Data independence**
- **Data integrity**
- **Data security**
- **Data backup and recovery**
- **Decreased data redundancy**

What is Database Normalization?

- Normalization is the process of organizing data in a database.
- It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.
- **Normalization** is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization rules divides larger tables into smaller tables and links them using relationships.

Database Normalization Cont…

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)

Database Normalization With Examples Cont..

- Here you see **Movies Rented column has multiple values.** Now let's move into 1st Normal Forms:

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

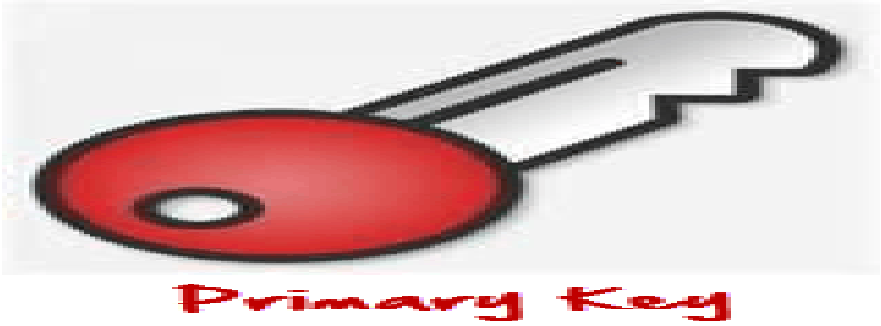
Database Normalization With Examples Cont..

- **1NF (First Normal Form) Rules**
- Each table cell should contain a single value.
- Each record needs to be unique.
- The above table in 1NF- ***Example of 1NF in DBMS***

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Database Normalization With Examples Cont..

- **What is a KEY in SQL**
- A **KEY in SQL** is a value used to identify records in a table uniquely. An SQL KEY is a single column or combination of multiple columns used to uniquely identify rows or tuples in the table. SQL Key is used to identify duplicate information, and it also helps establish a relationship between multiple tables in the database.
- Note: Columns in a table that are NOT used to identify a record uniquely are called non-key columns.
- **What is a Primary Key?**
- Primary Key in DBMSA primary is a single column value used to identify a database record uniquely.
- It has following attributes
- A primary key cannot be NULL
- A primary key value must be unique
- The primary key values should rarely be changed
- The primary key must be given a value when a new record is inserted.



Database Normalization With Examples Cont..

- **What is Composite Key?**
- A composite key is a primary key composed of multiple columns used to identify a record uniquely
- In our database, we have two people with the same name Robert Phil, but they live in different places. Composite key in Database
- Hence, we require both Full Name and Address to identify a record uniquely. That is a composite key.

Database Normalization With Examples Cont..

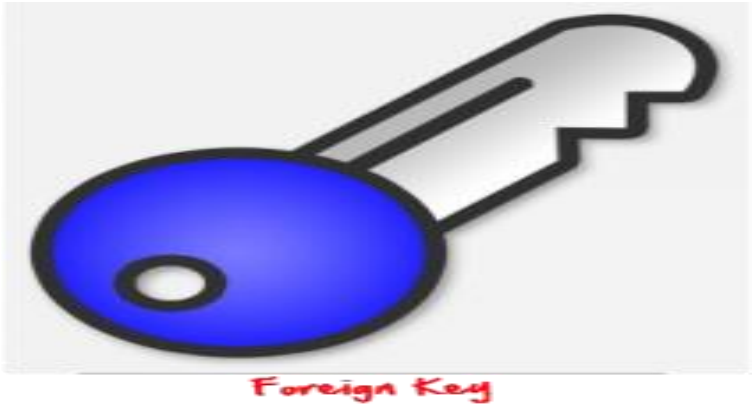
- **2NF (Second Normal Form) Rules**
- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key that does not functionally dependant on any subset of candidate key relation
- It is clear that we can't move forward to make our simple database in 2nd Normalization form unless we partition the table above.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Database Normalization With Examples Cont..

- We have divided our 1NF table into two tables viz. Table 1 and Table2. Table 1 contains member information. Table 2 contains information on movies rented.
- We have introduced a new column called Membership_id which is the primary key for table 1. Records can be uniquely identified in Table 1 using membership id
- **Database – Foreign Key**
- In Table 2, Membership_ID is the Foreign Key



MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Database Normalization With Examples Cont..

- Foreign Key references the primary key of another Table! It helps connect your Tables
- A foreign key can have a different name from its primary key
- It ensures rows in one table have corresponding rows in another
- Unlike the Primary key, they do not have to be unique. Most often they aren’t
- Foreign keys can be null even though primary keys can not

Database Normalization With Examples Cont..

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Foreign Key references Primary Key
Foreign Key can only have values present in primary key
It could have a name other than that of Primary Key

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Database Normalization With Examples Cont..

- **Why do you need a foreign key?**
- Suppose, a novice inserts a record in Table B such as

Insert a record in Table 2 where Member ID =101

MEMBERSHIP ID	MOVIES RENTED
101	Mission Impossible

But Membership ID 101 is not present in Table 1

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Database will throw an ERROR. This helps in referential integrity

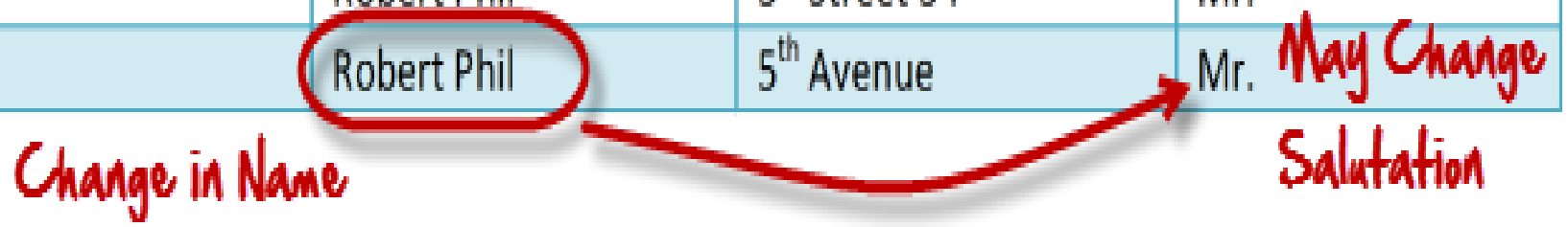
Database Normalization With Examples Cont..

- You will only be able to insert values into your foreign key that exist in the unique key in the parent table. This helps in referential integrity.
- The above problem can be overcome by declaring membership id from Table2 as foreign key of membership id from Table1
- Now, if somebody tries to insert a value in the membership id field that does not exist in the parent table, an error will be shown!

Database Normalization With Examples Cont..

- **What are transitive functional dependencies?**
- A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change
- Consider the table 1. Changing the non-key column Full Name may change Salutation.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.



Database Normalization With Examples Cont..

- **NF (Third Normal Form) Rules**
- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies
- To move our 2NF table into 3NF, we again need to again divide our table.
- **3NF Example**
- Below is a 3NF example in SQL database:

Database Normalization With Examples Cont..

Membership ID	Full names	Physical Address	Salutation ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1

Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Salutation ID	Salutation
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

Database Normalization With Examples Cont..

- We have again divided our tables and created a new table which stores Salutations.
- There are no transitive functional dependencies, and hence our table is in 3NF
- In Table 3 Salutation ID is primary key, and in Table 1 Salutation ID is foreign to primary key in Table 3
- Now our little example is at a level that cannot further be decomposed to attain higher normal form types of normalization in DBMS. In fact, it is already in higher normalization forms. Separate efforts for moving into next levels of normalizing data are normally needed in complex databases. However, we will be discussing next levels of normalization in DBMS in brief in the following.

Database Normalization With Examples Cont..

- **BCNF (Boyce-Codd Normal Form)**
- Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one **Candidate** Key.
- Sometimes is BCNF is also referred as **3.5 Normal Form.**
- **4NF (Fourth Normal Form) Rules**
- If no database table instance contains two or more, independent and multivalued data describing the relevant entity, then it is in 4th Normal Form.
- **5NF (Fifth Normal Form) Rules**
- A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.

What is database security ?

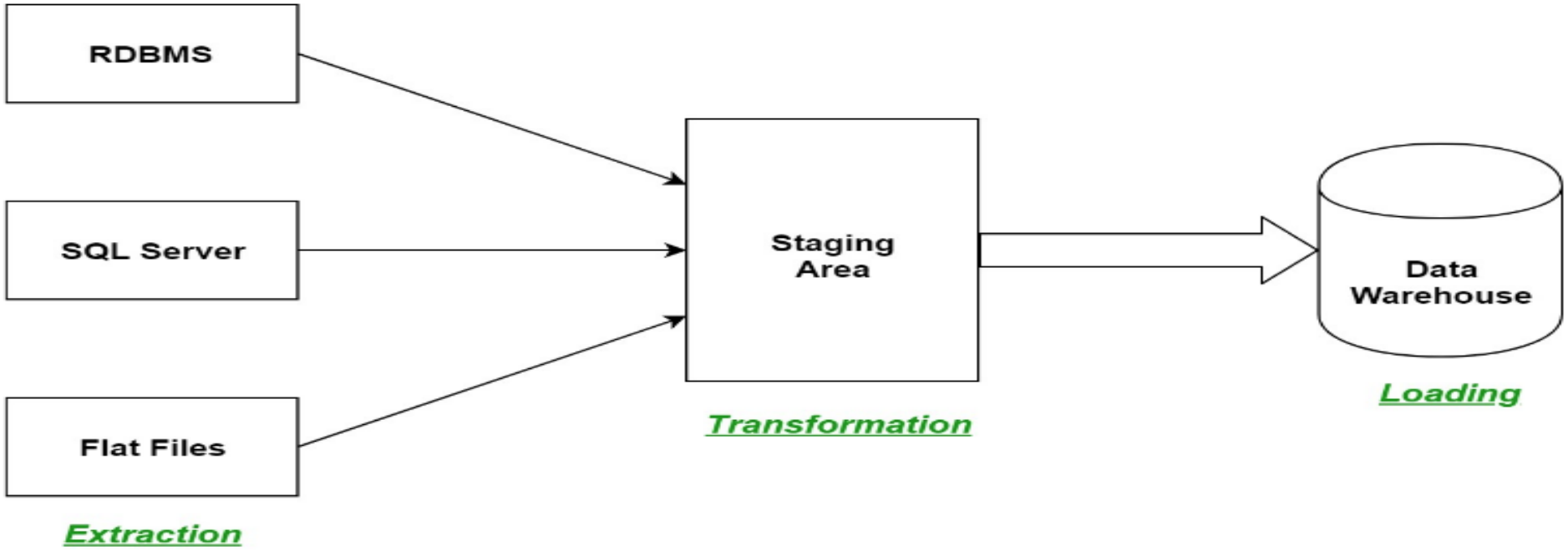
- Database security is the processes, tools, and controls that secure and protect databases against accidental and intentional threats.
- The objective of database security is to secure sensitive data and maintain the confidentiality, availability, and integrity of the database.
- Database security refers to the range of tools, controls, and measures designed to establish and preserve database confidentiality, integrity, and availability. This article will focus primarily on confidentiality since it's the element that's compromised in most data breaches.
- ***Database security must address and protect the following:***
- The data in the database
- The database management system (DBMS)
- Any associated applications
- The physical database server and/or the virtual database server and the underlying hardware
- The computing and/or network infrastructure used to access the database
- Database security is a complex and challenging endeavor that involves all aspects of information security technologies and practices.
- It's also naturally at odds with database usability.
- The more accessible and usable the database, the more vulnerable it is to security threats; the more invulnerable the database is to threats, the more difficult it is to access and use.

What are the types of database security?

- Access control.
- Auditing.
- Authentication.
- Encryption.
- Integrity controls.
- Backups.
- Application security.

Data Ware housing and Data mining

- Data warehousing is a method of organizing and compiling data into one database, whereas data mining deals with fetching important data from databases.
- Data mining attempts to depict meaningful patterns through a dependency on the data that is compiled in the data warehouse.



DATA WAREHOUSE

- A data warehouse is where data can be collected for mining purposes, usually with large storage capacity.
- Various organizations' systems are in the data warehouse, where it can be fetched as per usage.
- **Source Extract Transform Load Target.**
- (Data warehouse process)
- Data warehouses collaborate data from several sources and ensure data accuracy, quality, and consistency. System execution is boosted by differentiating the process of analytics from traditional databases.
- In a data warehouse, data is sorted into a formatted pattern by type and as needed. The data is examined by query tools using several patterns.
- Data warehouses store historical data and handle requests faster, helping in online analytical processing, whereas a database is used to store current transactions in a business process that is called online transaction processing.

FEATURES OF DATA WAREHOUSES

- **Subject Oriented:**
- It provides you with important data about a specific subject like suppliers, products, promotion, customers, etc. Data warehousing usually handles the analysis and modeling of data that assist any organization to make data-driven decisions.
- **Integrated:**
- Different heterogeneous sources are put together to build a data warehouse, such as level documents or social databases.
- **Time-Variant:**
- The data collected in a data warehouse is identified with a specific period.
- **Nonvolatile:**
- This means the earlier data is not deleted when new data is added to the data warehouse. The operational database and data warehouse are kept separate and thus continuous changes in the operational database are not shown in the data warehouse.

APPLICATIONS OF DATA WAREHOUSES

- Data warehouses help analysts or senior executives analyze, organize, and use data for decision making.
- *It is used in the following fields:*
- **Consumer goods**
- **Banking services**
- **Financial services**
- **Manufacturing**
- **Retail sectors**

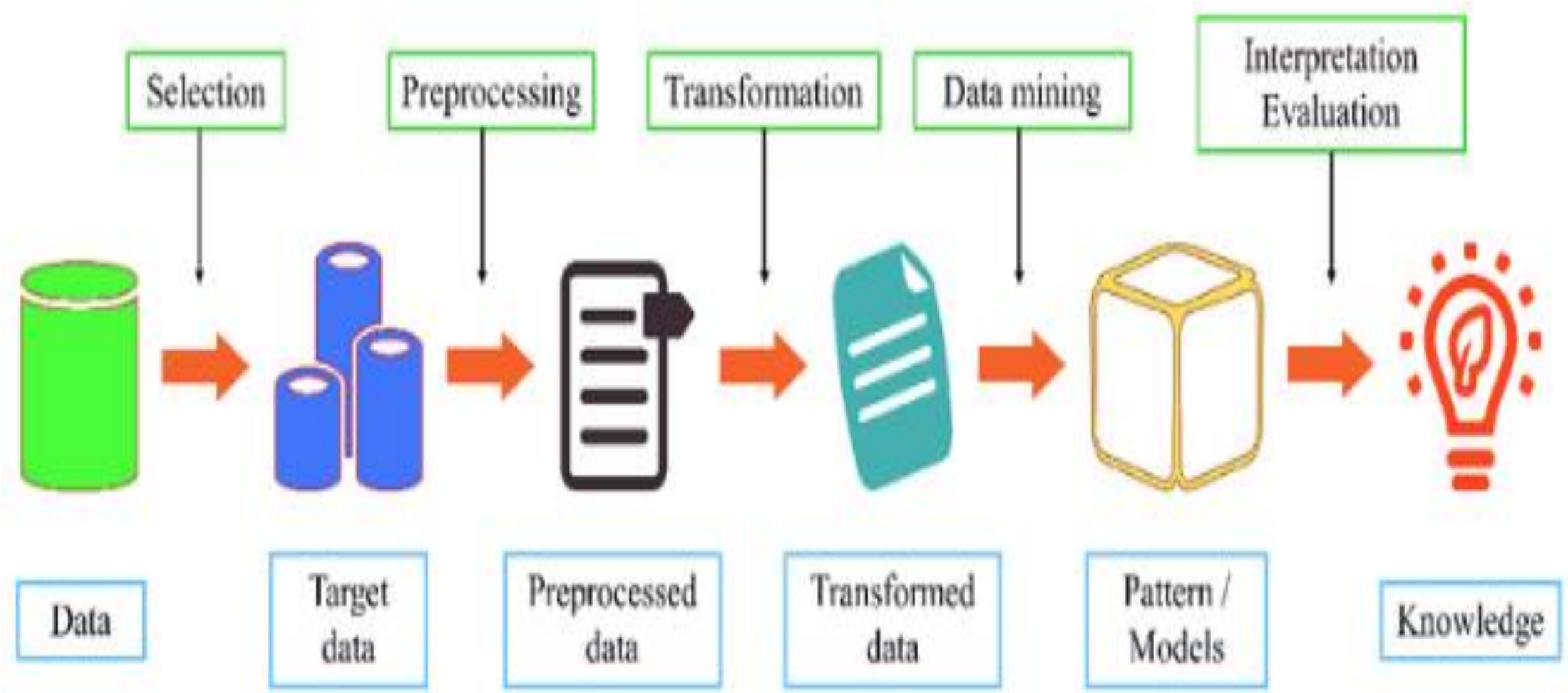
ADVANTAGES OF DATA WAREHOUSING

- Cost-efficient and provides quality of data
- Performance and productivity are improved
- Accurate data access and consistency

DATA MINING

- In this process, data is extracted and analyzed to fetch useful information. In data mining hidden patterns are researched from the dataset to predict future behavior.
- Data mining is used to indicate and discover relationships through the data.
- Data mining uses statistics, artificial intelligence, machine learning systems, and some databases to find hidden patterns in the data.
- It supports business-related queries that are time-consuming to resolve.

DATA MINING CONT..



FEATURES OF DATA MINING

- It is good with large databases and datasets
- It predicts future results
- It creates actionable insights
- It utilizes the automated discovery of patterns

ADVANTAGES OF DATA MINING

- **Fraud Detection:**
- It is used to find which insurance claims, phone calls, debit or credit purchases are fraud.
- **Trend Analysis:**
- Existing marketplace trends are analyzed, which provides a strategic benefit as it helps in reduction of costs, as in manufacturing per demand.
- **Market Analysis:**
- It can predict the market and therefore help to make business decisions. For example: it can identify a target market for a retailer, or certain types of products desired by types of customers.
-

What is big data?

- Big data is a combination of structured, semistructured and unstructured data collected by organizations that can be mined for information and used in machine learning projects, predictive modeling and other advanced analytics applications.
- Systems that process and store big data have become a common component of data management architectures in organizations, combined with tools that support big data analytics uses.
- **Big data is often characterized by the three V's:**
- ***the large volume of data in many environments;***
- ***the wide variety of data types frequently stored in big data systems; and***
- ***the velocity at which much of the data is generated, collected and processed***.
- Big data is data that contains greater variety, arriving in increasing volumes and with more velocity.
- This is also known as the three Vs. Put simply, big data is larger, more complex data sets, especially from new data sources.

Advantages of Big Data

- The increase in the amount of data available presents both opportunities and problems.
- In general, having more data on customers (and potential customers) should allow companies to better tailor products and marketing efforts in order to create the highest level of satisfaction and repeat business.
- Companies that collect a large amount of data are provided with the opportunity to conduct deeper and richer analysis for the benefit of all stakeholders.

What is a database administrator (DBA)?

- A database administrator, or DBA, is responsible for maintaining, securing, and operating databases and also ensures that data is correctly stored and retrieved.
- In addition, DBAs often work with developers to design and implement new features and troubleshoot any issues
- A database administrator, or DBA, is responsible for maintaining, securing, and operating databases and also ensures that data is correctly stored and retrieved.
- In addition, DBAs often work with developers to design and implement new features and troubleshoot any issues. A DBA must have a strong understanding of both technical and business needs.
- The role of DBA is becoming increasingly important in today's information-driven business environment.
- Thoroughout the world, more and more organizations depend on data to discover analytical insights on market conditions, new business models, and cost-cutting measures.
- The global cloud computing market is also expected to expand as companies move their business operations to the cloud.
- Consequently, the need for qualified DBAs will only continue to grow.

Different types of DBAs

- ***There are several types of database administrators, each with specific duties and responsibilities.***
- **The most common types of DBAs include system administrators, database architects, database analysts, data modelers, application DBAs, task-oriented DBAs, performance analysts, data warehouse administrators, and cloud DBAs.**
- System administrators are responsible for the overall management and upkeep of a computer system, including installing and configuring software, applying security patches, and monitoring system performance.
- Database architects design databases to meet the specific needs of an organization.
- Database analysts collect and analyze data to help improve database performance. They may also be responsible for developing reports and providing recommendations to database administrators.
- Data modelers create and maintain data models that depict the relationship between data elements. Data modeling is a critical component of effective database design.
- Application DBAs are responsible for administrating databases that support applications. Specific tasks include installing and configuring applications, ensuring that data is synchronized correctly between databases, and troubleshooting application-related issues.

Different types of DBAs

- Task-oriented DBAs focus on a particular area of database administration, such as backup and recovery, security, or performance tuning. They typically have in-depth knowledge of a specific database management system (DBMS).
- Performance analysts monitor database performance and identify areas where improvement is needed. They may also be responsible for creating performance reports and providing recommendations to database administrators.
- Data warehouse administrators manage databases that store data for business intelligence or decision-support applications. They are responsible for extracting data correctly, transforming the data, and loading it into the data warehouse.
- Cloud DBAs are responsible for administrating databases hosted in a cloud computing environment, provisioning and managing database instances, setting up replication and high availability, and monitoring database perform

Database Administrator Cont..

- ***Database administrator can be classified as***

1. System DBA

- Overview System DBAs typically have a background in system architecture and are responsible for the physical and technical aspects of a database.
- This can include installing upgrades and patches to fix program bugs and ensuring that the database works properly in a firm's computer system.

2. Application DBA

- Overview Application DBAs use complex programming languages to write or debug programs that work with the database.
- Usually this database has been designed for a specific application or a set of applications, such as customer service software


Thank you!