Database Fundamentals

Cybersecurity Department

Course Code: ***CBS 213***

Lecture 2 : Data Models and the Relational Models

Halal Abdulrahman Ahmed

Lecture Outline

- Data and Information
- Data Models and Their Purpose
- Types of Data Models (Conceptual, Logical, Physical)
- Relational Model and Its Structure
- Key Terms in the Relational Model
- Integrity Rules
- Advantages of the Relational Model

Learning Outcomes

- By the end of this lecture, students will be able to:
- Distinguish between **data** and **information**.
- Explain the role and types of **data models**.
- Describe the **relational model** and its key terms.
- Understand the main **integrity rules** in databases.
- Recognize the **benefits** of using the relational model.

Data and Information

- Data refers to **raw and unprocessed facts** that by themselves do not carry any clear meaning. It can include **numbers, words, symbols, dates, or any measurable values** that are simply collected and stored. They exist independently but do not tell us anything useful until they are organized or interpreted.
- On the other hand, **information** is the **result of processing or organizing data** in a way that gives it **meaning and context**. It helps in **decision-making**, understanding patterns, and drawing conclusions.
- In simple terms, **data is the raw input**, while **information is the meaningful output** obtained after data has been processed.

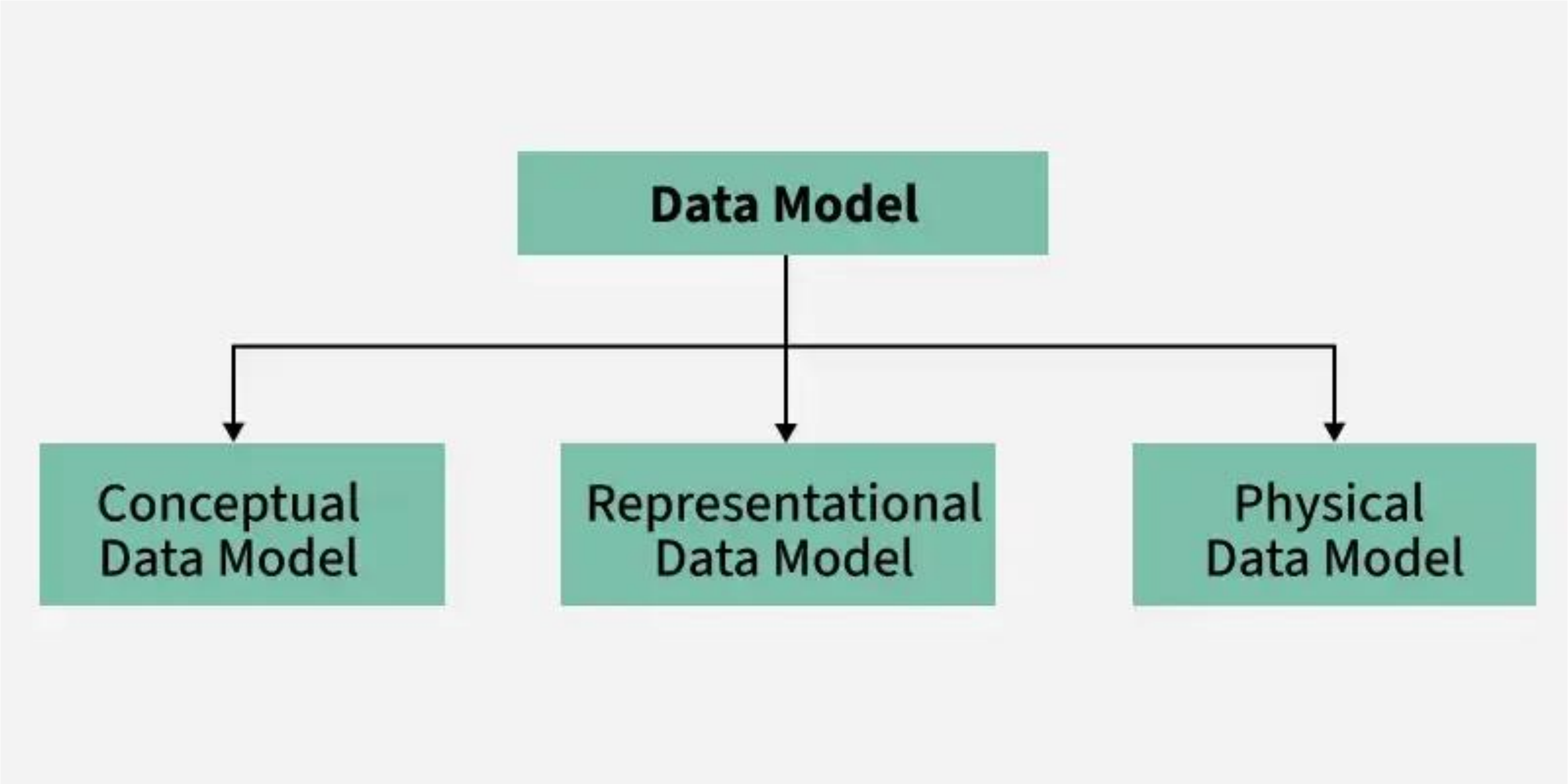
What is a Data Model?

- A **data model** is a **conceptual framework** that describes how data is structured, organized, and related inside a database.

Purpose

- Provides a **blueprint** before building the actual database.
- Helps designers and programmers understand the structure of data.
- Reduces duplication and inconsistency.
- Improves communication between database designers and users.

Types of Relational Models



Conceptual Data Model

- The **Conceptual Data Model** shows the database at a **very high level**.
- Used during **requirement gathering** to understand *what* data the system needs, not *how* it will be stored.
- Helps communicate between technical and non-technical users.
- Common example: **Entity-Relationship (ER) Model**.
- The ER model defines **entities**, **attributes**, and **relationships**.
- Useful for discussions with clients or stakeholders before actual design starts.

Its Components:

- **Entity** → real-world object (student, course, product).
- **Attribute** → property of an entity (name, ID, price).
- **Relationship** → association between entities (student enrolls in course).
- **ER diagram symbols:**
    - Entity = Rectangle   Attribute = Ellipse   Relationship = Diamond

Its Characteristics:

- Covers the whole organization’s concepts.
- Builds a **common vocabulary** for all stakeholders.
- Focuses on *what data means* in the real world.

Representational (Logical) Data Model

- Describes the logical structure of data rather than physical storage, and converts the conceptual design into a model ready for implementation.
- The most common representational model is the Relational Model. Data and relationships are shown using tables.
- Based on Relational Algebra and Relational Calculus for manipulation and queries. Provides the foundation for creating the physical model.

Key Features & Benefits:

- Represents **entities as tables** and **relationships through keys**.
- Each table has rows (tuples) and columns (attributes).
- Enforces **data integrity** through constraints and rules.
- Enables designers to focus on **database logic** (relations, constraints, joins).

Advantages:

  - Easy to understand and implement.
  - Provides a bridge between conceptual ideas and physical storage.
  - Supports independence from hardware details.

Physical Data Model

- The Physical Data Model defines how data is actually stored on hardware. It implements the logical model using a specific DBMS (e.g., MySQL, Oracle).
- Data is saved as files, records, indexes, and data structures. It shows how tables are organized in memory for fast access.
- SQL commands are used to build and manage the actual database. It is created by database administrators (DBAs) and developers.

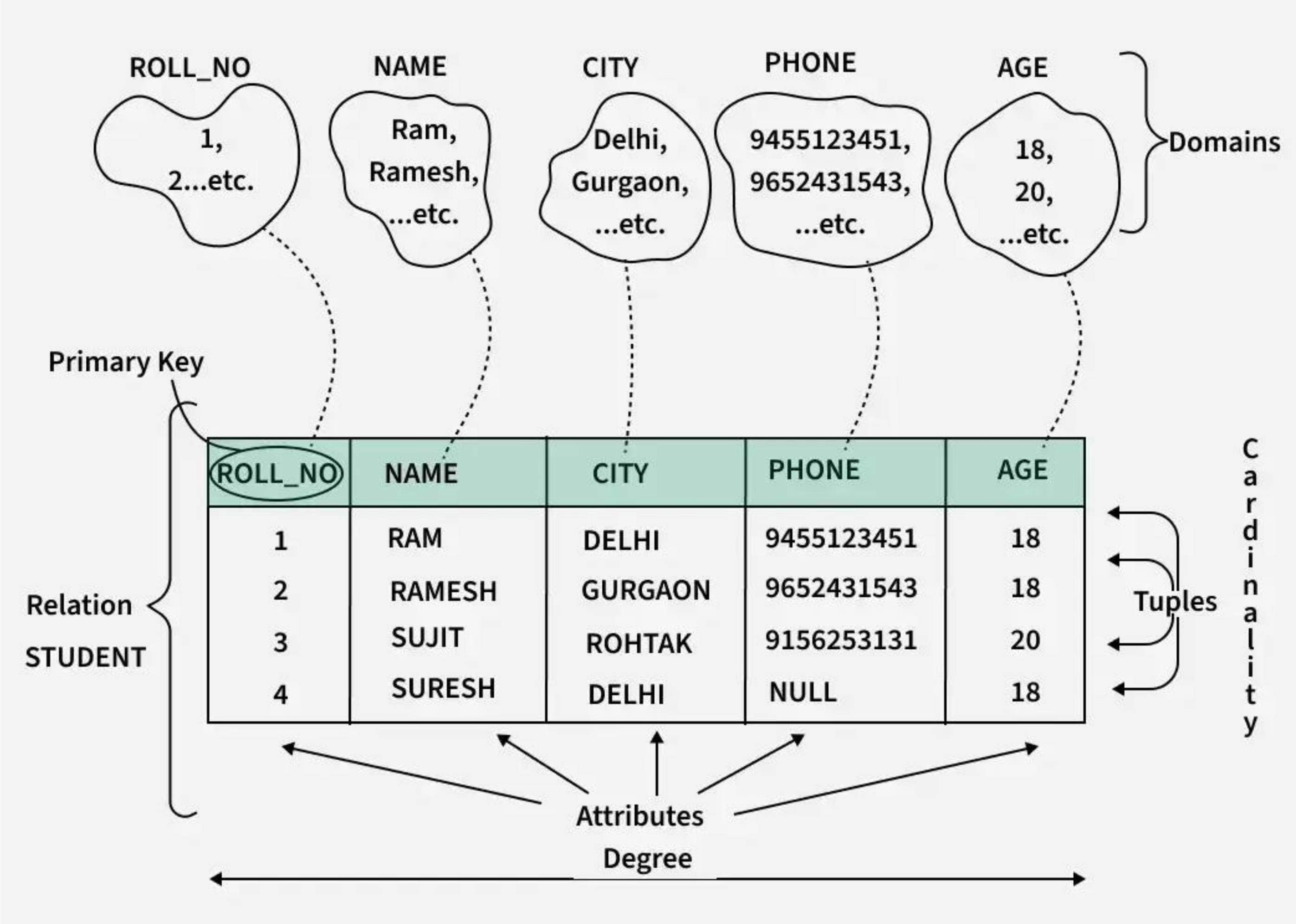
Its Characteristics & Details:

- Describes **how the system will be implemented**.
- Specifies **data types, field lengths, defaults**, and **storage locations**.
- Defines **primary / foreign keys, indexes, views, authorizations**.
- Focuses on **performance, storage efficiency, and security**.
- Often built for a **specific project or application**.
- Integrates with other physical models if needed for enterprise databases.

The Relational Model

- The Relational Model organizes data using tables (relations) consisting of rows and columns.
- The relational model represents how data is stored and managed in Relational Databases where data is organized into tables, each known as a relation.
- Each row of a table represents an entity or record and each column represents a particular attribute of that entity.
- The relational model transforms conceptual designs from ER diagrams into implementable structures. These structures are used in relational database systems like Oracle SQL and MySQL.

Example: Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in the table.



ROLL_NO	NAME	CITY	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI	NULL	18

Key Terms in the Relational Model

The relational model uses specific terms to describe how data is structured and organized.

- **Attribute:** Represents a column that describes one property or characteristic of an entity.
- **Relation Schema:** Defines the overall structure of a table, including its name and attributes. It describes what data is stored and how it is organized.
- **Tuple:** Represents a single record or row in a table containing values for each attribute.
- **Relation Instance:** Refers to all the rows present in a table at a specific time; it changes whenever data is added, modified, or deleted.
- **Degree:** The total number of attributes (columns) present in a relation.

Key Terms in the Relational Model (cont.)

- **Cardinality:** The total number of tuples (rows) present in a relation.
- **NULL Value:** Indicates missing, unknown, or inapplicable data. It represents the absence of a value, not zero or empty text.

Advantages of the Relational Model

Main Benefits:

- **Simplicity:** Easy to understand (tables = real-world objects).
- **Reduced Redundancy:** Data is not duplicated thanks to relationships.
- **Data Integrity:** Rules ensure correctness.
- **Flexibility:** You can join tables easily to get complex results.
- **Security & Multi-user Access:** Supports many users safely.
- **Independence:** Logical and physical data are separated.

Integrity Rules

Integrity rules ensure **accuracy, validity, and consistency** of data in the relational model.

Type	Meaning	Example
Entity Integrity	Every row must have a unique, non-null primary key.	Student_ID cannot be NULL
Referential Integrity	Foreign key must match a valid primary key.	A student’s enrollment must exist in Students table
Domain Integrity	Data values must match their defined type or rule.	GPA between 0–4
User-defined Integrity	Specific business rules applied by the designer.	Salary must be > 0

Question

- If you add a new student record, what changes schema or instance?

- **Answer: Instance**

  (Schema stays the same; data inside changes.)

Question

List two advantages you think are most important for cybersecurity.

Possible Answers:

- **Integrity:** Data accuracy and consistency are guaranteed.
- **Security:** Controlled access prevents unauthorized changes.

References

- Introduction to Database Systems: Modeling and Administration. (2023). *Introduction to Database Systems*. Cengage.
- GeeksforGeeks. (2025, July 12). *Relational model in DBMS*. Retrieved from https://www.geeksforgeeks.org/dbms/relational-model-in-dbms/
- GeeksforGeeks. (2025, July 15). *Data models in DBMS*. Retrieved from https://www.geeksforgeeks.org/dbms/data-models-in-dbms/

Any
Question