

Notes

Lecture -1

Database

A **database** is:

A shared collection of logically related data, organized for easy access, management, and update.

Key characteristics:

- **Shared** among users.
- **Collection** of related data.
- **Self-describing** (contains metadata).
- **Logically related** entities, attributes, and relationships.

Relational Database (RDBMS)

- Introduced by **E.F. Codd (1970, IBM)**.
- Stores data in **tables (relations)** made of **rows (tuples)** and **columns (attributes)**.
- Uses **SQL (Structured Query Language)** for interaction.
- Easy to extend — new tables or columns can be added without breaking existing applications.

Data Model

Defines **how data is structured and manipulated**.

Components:

1. **Structural part** – rules for building the database.
2. **Integrity rules** – ensure data accuracy.
3. **Manipulative part** – defines operations (e.g., query, insert, delete).

Entities & Attributes

- **Entity:** A real-world object or concept (e.g., Student, Book).
- **Attribute:** A property of an entity (e.g., Name, ID, Address).
- A **table** represents a set of entities; each **row** is an instance.

Terminology Comparison

User Term	Relational Term	Programmer Term
Table	Relation	File / Array
Row	Tuple	Record
Column	Attribute	Field

Properties of Relations

- Each relation (table) has a **name**.
- Each cell holds a **single atomic value**.
- Each attribute has a **distinct name**.
- The **order of rows/columns** doesn't matter.
- Rows (tuples) are **unique**.
- **Degree** = number of columns, **Cardinality** = number of rows.

Keys

- **Primary Key:** Uniquely identifies each record (e.g., ISBN for Books).
- **Superkey:** Any set of attributes that uniquely identifies a row.
- **Candidate Key:** Minimal superkey — smallest unique identifier.
- **Foreign Key:** Attribute that references a key in another table, maintaining relationships.
- candidate key is the subset of super key

Relational Integrity

Ensures **data accuracy and consistency** through rules:

1. **Domain Constraints** – values must belong to a valid range or type.
2. **Entity Integrity Rule** – primary key values **cannot be null**.
3. **Referential Integrity Rule** – foreign keys must match a valid key in the parent table or be null.

Null Values

- Represent **unknown** or **non-applicable** data.
- Not the same as:
 - 0 (zero)
 - Empty string or space
- Some systems replace nulls with placeholders like 1 or "N/A".

Queries

Used to retrieve or modify data.

Example:

List animals that eat buns, weigh between 20–200 kg, ordered by age.

Two main methods:

- **QBE (Query By Example)**: Visual query method.
- **SQL (Structured Query Language)**: Text-based, uses English-like syntax (SELECT, WHERE, INSERT).

Lecture 2

What is ER Modelling?

Entity–Relationship (ER) Modelling is a **high-level conceptual data model** used to design a database before actually building it.

It helps you visualize **what data will be stored** and **how entities are related**.

Think of it as the **blueprint of a database**

ER Modelling → Database Design Stages

Stage	Description	Output
Requirement Collection	Interview users, identify data needs	List of requirements
Conceptual Design	Create ER diagram	Conceptual Schema
Logical Design	Convert ERD to database tables	Relational Schema
Physical Design	Decide storage, indexes, access paths	Optimized database

Example Flow:

Requirement: Store students and their courses

↓

Conceptual: ER Diagram (Student–Course)

↓

Logical: Create STUDENT and COURSE tables with foreign keys

↓

Physical: Add indexes for faster queries

Entity

A **real-world object** that you want to store information about.

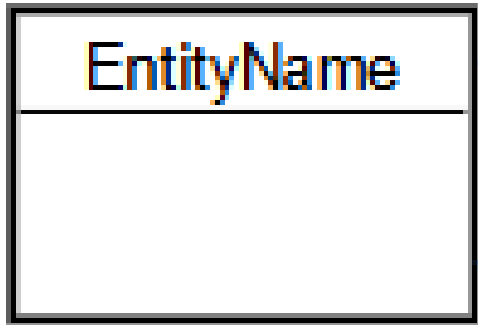
It can be a **person, place, object, or event**.

Examples:

- Student , Teacher , Course , Department

Types of Entities:

- **Strong Entity:** Exists independently (e.g., Student , Staff)
- **Weak Entity:** Depends on another entity for existence (e.g., Dependent , NextOfKin)



Attribute

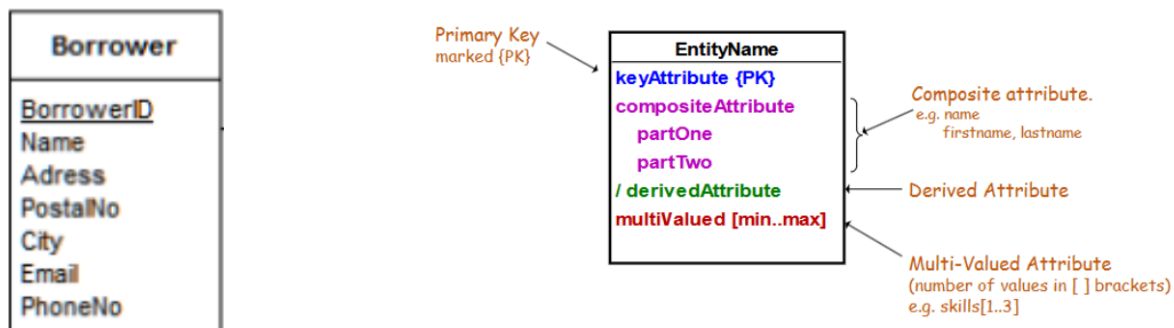
Properties or characteristics that describe an entity.

Example (Student):

StudentID , Name , Email , DOB

Types of Attributes:

Type	Description	Example
Simple (Atomic)	Single value	Name
Composite	Made up of parts	Address → Street, City, Postcode
Multi-Valued	Can have multiple values	PhoneNumber → Home, Work
Derived	Can be calculated from others	Age (from DOB)



Relationships

Relationships show **how entities are linked**.

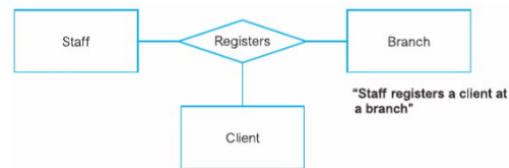
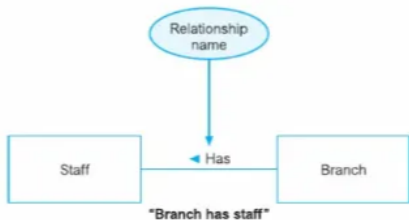
Example:

A Student enrolls in a Course .

Participants: Entities involved in the relationship.

Relationship Degree

Degree	Meaning	Example
Unary (1)	Relationship within the same entity	Employee supervises Employee
Binary (2)	Between two entities	Student enrolls in Course
Ternary (3)	Among three entities	Client registered by Staff at Branch



Types of Relationships

Type	Description	Example
1:1 (One-to-One)	One entity linked to one other	One employee manages one department
1:N (One-to-Many)	One entity linked to many others	One customer places many orders
M:N (Many-to-Many)	Many entities linked to many others	Students take many courses, and each course has many students





Cardinality (Constraints)

Defines **how many instances** of an entity can participate in a relationship.

Maximum Cardinality

- Max number of entities that can be related
- Examples: **1** (one), **N** (many)

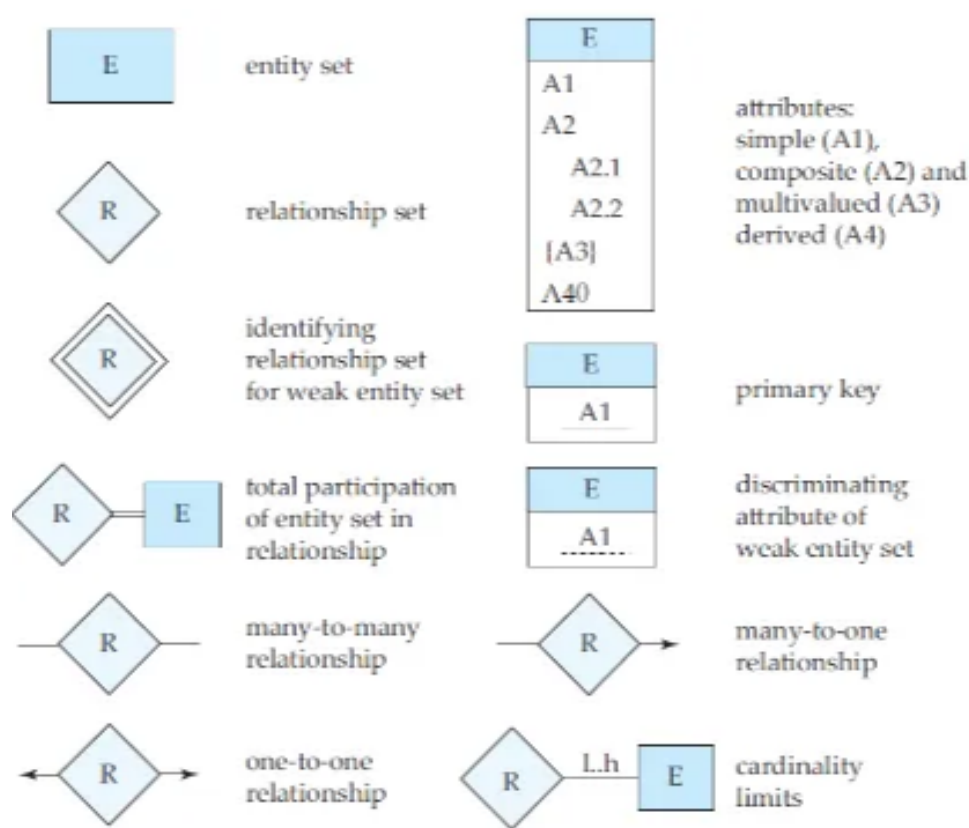
Minimum Cardinality

- Min number of entities that must participate
- **0** → **Optional**, **1** → **Mandatory**

Meaning:

- Each supplier can supply many items.
- An item **must** have one supplier (min 1).
- A supplier **may not** supply any item (min 0).

ER Diagram Symbols (Notation)



Entity Relationship Diagram (ERD)

