DBMS Lec 3:

The video explains the Entity-Relationship (ER) model, a way to design and organize data in databases using entities (real-world objects), their attributes (properties), and relationships (connections between entities). It also covers types of attributes, constraints, and how to represent these elements in diagrams to make database design easier and more consistent.

Introduction to Data Models and ER Model

Data models help define and organize data in databases. The Entity-Relationship (ER) model is a high-level data model that visually represents real-world objects (entities), their attributes, and the relationships between them.

What is an Entity and Attribute?

An entity is a real-world object, such as a student or customer, that is distinguishable from others. Each entity has attributes—properties like name, ID, or address—that describe it.

Entity Uniqueness and Primary Key

Every entity must be uniquely identifiable, usually by a primary key (like student ID or customer ID). This ensures each record is distinct and can be referenced easily.

Entity Sets

An entity set is a collection of similar entities sharing the same attributes, such as all students in a university. Each member is an entity.

Relationships Between Entities

Relationships connect different entities, like a customer having a loan or a student enrolling in courses. These associations are essential for representing how data items interact in real life.

ER Diagrams as Blueprints

ER diagrams graphically represent entities (as rectangles), attributes (as ovals), and relationships (as diamonds or lines). These diagrams act as blueprints for building database tables and structures.

Attribute Types and Domains

Attributes can have domains, which are allowed values or data types (like numbers or text). Consistency constraints ensure only valid data is entered (e.g., names can’t be numbers).

Types of Attributes

- Attributes can be: - Simple (cannot be divided, like account number)
- Composite (can be split into parts, like address into street, city, etc.)
- Single-valued (only one value, like student ID)
- Multi-valued (multiple values, like phone numbers)
- Derived (calculated from other attributes, like age from date of birth)

The Meaning of Null Values

Null values in attributes mean the value is either not applicable, unknown, or not assigned yet. Handling nulls properly ensures data accuracy and consistency.

Relationships and Their Types

Relationships can be between two (binary), three (ternary), or more entities, but most are binary. The

types of entities involved can affect how relationships are managed.

Strong and Weak Entities

Strong entities have their own primary key and exist independently. Weak entities depend on a strong entity for identification and do not have a primary key on their own.

Degree of Relationships

The degree of a relationship is the number of entity sets involved (unary, binary, ternary, etc.). Most database designs use binary relationships for simplicity.

Mapping Cardinality (Relationship Constraints)

Mapping cardinality defines how many entities in one set can be associated with entities in another set: one-to-one, one-to-many, many-to-one, or many-to-many.

One to One Example: Person ↔ Passport

- One person has **only one passport**.
- One passport belongs to **only one person**.

One to Many Example: Department ⇔ Employees

- One department has **many employees**.
- Each employee belongs to **only one department**.

Many to One Example: Same as above (Employee → Department).

- Multiple employees (N) → One department (1).

Many to Many Example: Students ⇔ Courses

- One student can enroll in **many courses**.
- One course can have **many students**.

1:1 → Person–Passport

1:N → Department–Employees

N:1 → Employees–Department

M:N → Students–Courses

Participation Constraints

Participation constraints specify if all or only some entities in an entity set must participate in a relationship. Total participation means all must be involved; partial means only some are.

ER Notation Recap

ER diagrams use specific symbols for entities, attributes, relationships, and keys. Understanding these notations helps in clearly communicating database designs.--- This summary provides a structured explanation of the ER model, its components, and practical considerations for designing effective databases.