Lecture 6

Database Management System

Er. Shiva Kunwar

Lecturer, GU

Lesson 2: Data Model (6hrs)

1. Different Data Model Concepts
2. E-R Model
3. Network/Hierarchical Model
4. Relational Model
5. **Entities, Relationships and Attributes**
6. **E-R Diagrams, Keys**
7. Generalization, Specialization and Aggregation

Topics

- Entities
- Entities Types and Entities Set
- Strong Entity Sets and Weak Entity Sets
- Relationship and Relationship Set
- Attributes and Keys; Primary Key, Candidate Key, Super Key
- E-R diagram
- Reducing E-R diagram to tables,

Entities

- An entity is a real-world object or concept that has a distinct existence and is distinguishable from other objects.
- In the context of a database, entities represent the main objects about which data is stored.
- Entities have attributes that describe their properties or characteristics.
- Each entity instance is uniquely identifiable, often by a primary key.
- Represented by rectangles in ER diagrams.
- *In a university database, "Student“, “Professor” and "Course" can be entities.*

Entities Types

- An entity type is a collection of entities that share common characteristics or properties.
- An entity type represents a category or class of objects in the real world.
- It defines a set of properties (attributes) applicable to all entities within that type.
- Represented by a rectangle in an ER diagram.
- *In a university database, "Student" is an entity type that includes individual students. Each student has attributes such as StudentID, Name, and DateOfBirth.*

Entities Set

- An entity set is a collection of all entities of a particular entity type at a given point in time.
- An entity set represents the current snapshot of all instances belonging to an entity type.
- The set evolves as entities are added or removed over time.
- Represented by the entire collection of rectangles of a specific entity type.
- *The set of all "Student" entities currently enrolled in a university forms the "Student" entity set.*

Strong Entity

- A strong entity is an entity that can be uniquely identified by its attributes.
- They can exist independently without relying on other entities for identification.
- Strong entities have a primary key that serves as a unique identifier.
- Represented by a rectangle in an ER diagram, with the entity name inside the rectangle.
- *In a university database, the "Student" entity is a strong entity. Each student can be uniquely identified by a StudentID, making it a strong entity.*

Weak Entity

- Weak entities rely on a strong entity, known as the "owner," for identification.
- They have a partial key, which is a set of attributes that, in combination with the owner entity's primary key, uniquely identifies instances.
- Represented by a double rectangle in an ER diagram, indicating its dependency on another entity.
- *In a database for a company's employees, a "Dependent" entity might be weak. It would depend on the "Employee" entity for identification, and its partial key could include attributes like "DependentName" and "Relationship" (relationship to the employee).*

Relationships

- A relationship is an association between two or more entities that signifies a real-world connection.
- It represents how instances of entities interact or are related to each other.
- Relationships have a name that describes the nature of the association (e.g., "Works for," "Enrolls in," "Manages").
- Represented by a diamond shape connecting related entities in an ER diagram.
- *In a university database, a "Teaches" relationship might exist between the "Professor" and "Course" entities. The relationship signifies which professors teach which courses.*

Degree of Relationship Set

1. Unary Relationship (Recursive Relationship)

- When there is only ONE entity set participating in a relation, the relationship is called a unary relationship.
- *"Manages" relationship where an employee manages another employee.*

2. Binary Relationship

- When there are TWO entities set participating in a relationship, the relationship is called a binary relationship.
- *"Works_In" relationship between "Employee" and "Department."*
- *"Enrolls_In" relationship between "Student" and "Course."*

Degree of Relationship Set

3. Ternary Relationship

- When there are THREE entities set participating in a relationship, the relationship is called a ternary relationship.
- *"Teaches" relationship involving "Professor," "Course," and "Department."*

4. n-ary Relationship

- When there are n entities set participating in a relation, the relationship is called an n-ary relationship.

Constraints

- E-R model has a capability to enforce constraints.
- Two most important type of constraints in ER model are-
- **Mapping Cardinalities (Cardinality ratio)**
- **Participation Constraints**

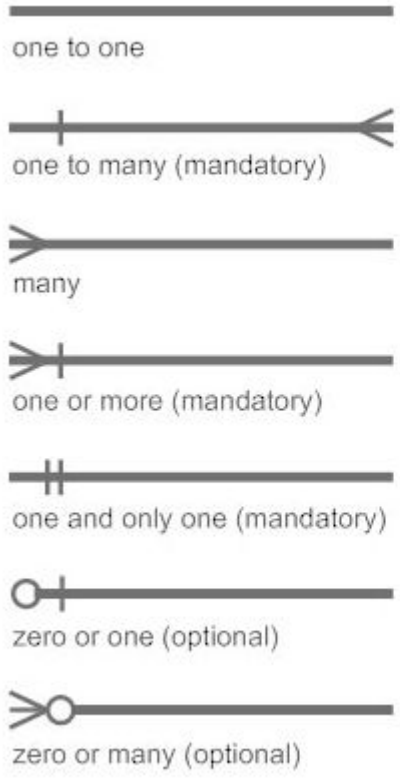
Mapping Cardinalities in Relationship

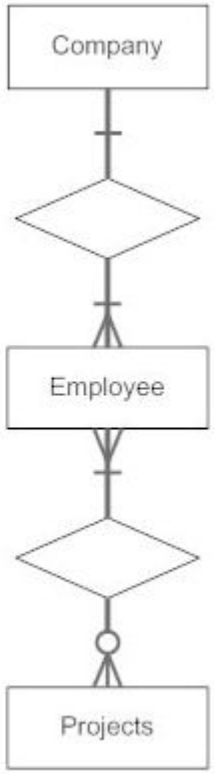
- Mapping Cardinalities describes no. of entities to which another entity can be associated via relationship set.
- Mapping cardinalities are most useful in describing binary relationship sets but it can also describe relationship sets that involve more than two entity sets.

Mapping Cardinalities in Relationship

- For binary relationship set between entity set A and B mapping cardinality must one of the following.
- **One to one**: An entity in A is associated with at most one entity in B and entity in B is associated with at most one entity in A.
- **One to many**: An entity in A is associated with zero or more entities in B but entity in B can be associated with at most one entity in A.
- **Many to one**: An entity in A is associated with at most one entity in B but an entity in B can be associated with zero or more entities in A.
- **Many to many**: An entity in A is associated with zero or more entities in B, and an entity in b is associated with zero or more entities in A.

Mapping Cardinalities in Relationship





Participation Constraints

- The participation of an entity set E in a relationship set R is said to be **total** if every entities in E participates in at least one relationship in R.
- If only some entities in E participate in relationship in R, then participation of entity set E in relationship set R is said to be **partial**.
- *The participation of loan in the relationship set borrower is total but customer entity set in borrower relationship set is partial since not all customers necessarily take loan from bank, customer may also those who are only account holder.*

Attributes

- An attribute is a property or characteristic of an entity that provides more information about the entity.
- Attributes describe the features, qualities, or traits associated with entities.
- Entities can have multiple attributes.
- Attributes have data types, such as string, number, or date, defining the kind of information they can store.
- Represented by ovals connected to the respective entity in ER diagrams.
- *Attributes for a "Student" entity might include "StudentID," "Name," and "DateOfBirth."*

Attributes Types

- **Key Attribute**: which uniquely identifies each entity *e.g.. Student_id*
- **Simple Attribute**: which cannot be divided further *e.g.. Phone_no*
- **Composite Attribute**: made of more than one simple attribute
  e.g.. name => first name, last name
- **Derived Attribute**: do not exist in physical database, and values derived from another attribute *e.g.. Age derived from DOB from database, here DOB is* ***Stored Attribute***
- **Single Valued Attribute**: having single value *e.g.. Student_id*
- **Multi Valued Attribute**: having multiple values *e.g.. Phone_no, email*
- **Descriptive Attributes**: attributes of the relationship

Keys

- A key is a set of one or more attributes that uniquely identifies an entity or a record within a database.
- Keys are essential for establishing relationships between tables and ensuring the accuracy and integrity of the data.
- In ER diagrams, primary keys are typically underlined.
- Foreign keys are represented as arrows pointing to the referenced table's primary key.
- *In a "Student" table, the "StudentID" could be the primary key, uniquely identifying each student. If there's a "Course" table, and we want to establish a relationship, the "StudentID" in the "Course" table becomes a foreign key, referencing the "Student" table's primary key.*

Keys Characteristics

Uniqueness

- Keys must ensure the uniqueness of values within a table.
- Each record should be uniquely identified by its key.

Uniqueness Across Tables

- In the case of a foreign key, the values should be unique in the referenced table's primary key.

Non-Null

- Keys should not contain null values.
- They should have valid and meaningful values to ensure reliable identification.

Keys Types

Primary Key

- A primary key is a unique identifier for each record in a table.
- No two records in the table can have the same primary key value.
- It ensures data integrity and serves as a reference for relationships with other tables.

Foreign Key

- A foreign key is a field in a table that refers to the primary key in another table.
- It establishes a link between the two tables, creating a relationship.
- Foreign keys maintain referential integrity.

Keys Types

Candidate Key

- A candidate key is a set of attributes that could serve as the primary key.
- It satisfies the uniqueness and irreducibility requirements for a key.
- *In the "Employee" table of a company database, both "EmployeeID" and "Email" could be candidate keys.*

Composite Key

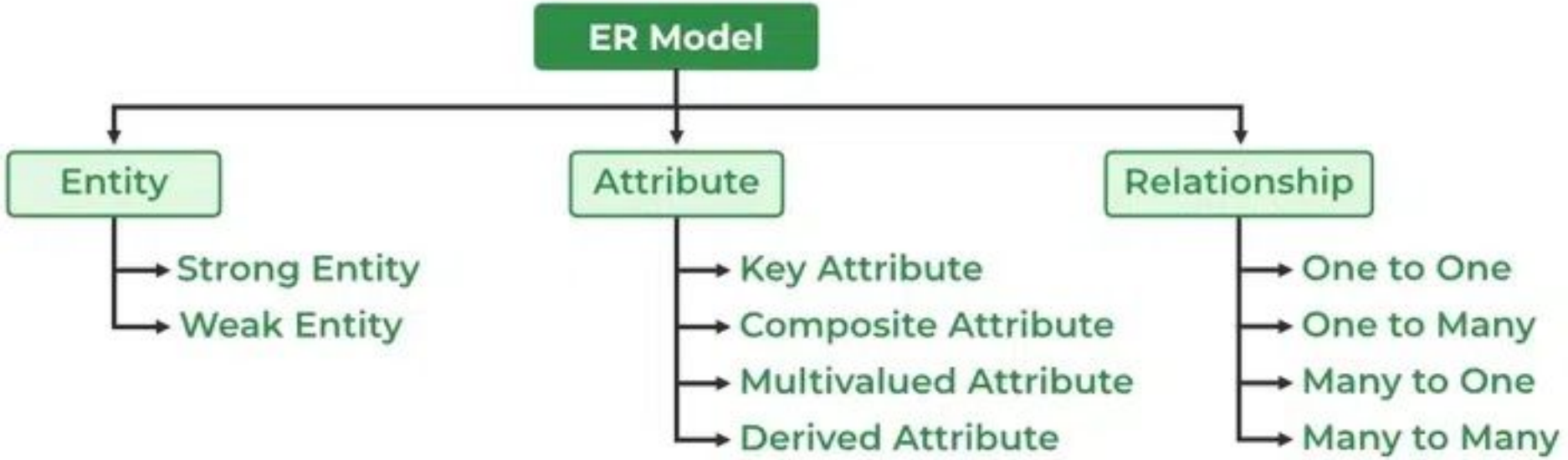
- A composite key is a key that consists of multiple attributes.
- It is used when a single attribute is not sufficient to uniquely identify a record.
- *In a "Purchase" table of an online store database, a composite key could be a combination of "CustomerID" and "ProductID" to uniquely identify each purchase.*

Keys Types

Super Key

- A super key is a set of one or more attributes (columns) that, taken together, can uniquely identify a tuple (row) in a relation (table) within a database.
- It is a broader concept than a primary key and can include more attributes than needed for uniqueness.
- A super key is essentially any combination of attributes that uniquely identifies a record.
- All primary keys are super keys, not all super keys are primary keys.
- *In the "Employee" table of a company database, both "EmployeeID" and "Email" could be candidate keys.*

Components of ER Model



ER Diagram

- Is a data modeling technique that creates a graphical representation of entities and their relationship within an information system.
- It illustrates the logical structure of the database.

Components of ER Diagram

- **Rectangles**: representing entity sets.
- **Ellipses**: representing attributes.
- **Diamonds**: representing relationship sets.
- **Lines**: linking attributes to entity sets and entity sets to relationship sets.
- **Double ellipses**: represents multivalued attributes.
- **Dashed ellipses**: represents derived attributes
- **Double lines**: represents total participation of entity in relationship sets
- **Double rectangles**: represents weak entity sets
- **Underline**: indicates primary key attributes

Important Tips Before Preparing an ER Diagram

- **Understand the Domain**
- **Identify Entities**
- **Define Relationships**
- **Attribute Identification**
- **Primary Keys**
- **Avoid Redundancy**
- **Use Consistent Naming Conventions**
- **Think About Cardinality**

Important Tips Before Preparing an ER Diagram

- **Understand the Domain**
- Before creating an ER diagram, it's crucial to thoroughly understand the domain or the application for which you are designing the database.
- This involves gaining insights into the business processes, user requirements, and the nature of the data involved.
- *Suppose you are designing a database for a library management system.*
- *Understanding the domain involves learning about library processes, user interactions, and the type of information stored.*
- *This could include knowledge about books, authors, borrowers, and library transactions.*

Important Tips Before Preparing an ER Diagram

- **Identify Entities**
- Start by identifying the main entities in the system.
- Entities are the fundamental building blocks that represent real-world objects or concepts.
- They could be objects like "Customer," "Product," or "Order" in a retail management system.
- *In the library management system, entities could include "Book," "Author," "Borrower," and "Transaction.“*
- *Each of these represents a distinct object or concept in the library domain.*

Important Tips Before Preparing an ER Diagram

- **Define Relationships**
- Determine how entities are related to each other.
- Relationships define the associations between entities.
- It differs with types like one-to-one, one-to-many, or many-to-many.
- *In the library management system, there could be a "Written by" relationship between "Author" and "Book," indicating that an author writes one or more books.*
- *This relationship is crucial for understanding how authors and books are connected.*

Important Tips Before Preparing an ER Diagram

- **Attribute Identification**
- List the attributes for each entity.
- Attributes describe the properties or characteristics of entities.
- Be comprehensive in identifying attributes to capture all relevant information.
- *In the library management system, for the "Book" entity, attributes could include "ISBN," "Title," "Genre," and "Publication Year.“*
- *For the "Author" entity, attributes might include "AuthorID," "Name," and "Nationality."*

Important Tips Before Preparing an ER Diagram

- **Primary Keys:**
- Clearly define primary keys for each entity.
- The primary key uniquely identifies each instance of an entity.
- Ensure that primary keys are unique and essential for data integrity.
- *In the library management system, in the "Book" entity, the "ISBN" could be the primary key.*
- *In the "Author" entity, the "AuthorID" could serve as the primary key.*
- *This ensures unique identification of each book and author.*

Important Tips Before Preparing an ER Diagram

- **Avoid Redundancy:**
- Minimize redundancy in the ER diagram by normalizing the model
- Normalization involves organizing data to reduce duplication and dependency.
- This ensures that data is stored efficiently and consistently, contributing to overall database integrity.
- *In the library management system, for example, there is information about the author's nationality in both the "Author" and "Book" entities, it might be redundant.*
- *Normalization would involve creating a separate "AuthorDetails" entity to store non-repetitive information about authors.*

Important Tips Before Preparing an ER Diagram

- **Use Consistent Naming Conventions:**
- Adopt consistent and meaningful names for entities, attributes, and relationships.
- Consistent naming conventions improve the clarity and maintainability of the ER diagram.
- Ensure that names are easily understandable and follow a standardized format.
- *In the library management system, use "BookTitle" in one entity and "AuthorName" in another, rather than mixing terms like "Title" and "Name."*
- *Consistency in naming enhances readability.*

Important Tips Before Preparing an ER Diagram

- **Think About Cardinality**
- Determine the cardinality of relationships.
- Cardinality specifies the number of instances of one entity associated with another.
- *In the library management system, in the "Borrower-Transaction" relationship, a borrower may have multiple transactions (one-to-many), but each transaction is associated with one borrower (one-to-one).*
- *Understanding cardinality is crucial for designing the database schema accurately.*

Example

- In a university, a Student enrolls in Courses. A student must be assigned to at least one or more Courses. Each course is taught by a single Professor. To maintain instruction quality, a Professor can deliver only one course.

Example

- Step 1) Entity Identification
- We have three entities

- Student
- Course
- Professor

Example

- Step 2) Relationship Identification
- We have the following two relationships

- The student is assigned a course
- Professor delivers a course

Example

- Step 4) Identify Attributes
- Once, you have a list of Attributes, you need to map them to the identified entities. Ensure an attribute is to be paired with exactly one entity. If you think an attribute should belong to more than one entity, use a modifier to make it unique.

- Once the mapping is done, identify the primary Keys. If a unique key is not readily available, create one.

Example

- Step 4) Identify Attributes
- Entity Primary Key Attribute
- Student Student_ID StudentName
- Professor Employee_ID ProfessorName
- Course Course_ID CourseName

Example

- Step 4) Cardinality Identification
- For them problem statement we know that,

- A student can be assigned multiple courses
- A Professor can deliver only one course

Practical

- Construct an ER diagram for a car insurance company with a registration id and an address and have name and address of its customers. Each customer own one or more cars which has its model, color and engine number. Each car is associated with zero or any number of recorded accidents.

Practical

- UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system.
- Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center.
- Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute.
- Prepare an ER Diagram that captures this information.

END OF LECTURE 6

- SHIVA.KUNWAR@HOTMAIL.COM
- +977-9819123654

Google classroom code : drbzdcf

PREVIEW FOR LECTURE 7

ER DIAGRAM Contd.