SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(IoT and Cybersecurity Including BCT)

COURSE NAME : 19SB504 DATABASE MANAGEMENT SYSTEMS

III YEAR / V SEMESTER

Unit III-**E-R Diagram models and NORMAL FORMS**

Topic : Conceptual design with the E-R model. E-R Issues-weak entity set

Conceptual design with the E-R model

- ✓ Conceptual design using the Entity-Relationship (E-R) model is the initial stage of database design, where you create an abstract representation of the database without considering specific database management system (DBMS) details.

- ✓ The goal is to define the high-level structure of the data, including entities, attributes, and their relationships.

Identify Entities:

- ✓ Start by identifying the main entities in the domain you are modeling.
- ✓ Entities are real-world objects, concepts, or subjects that you want to store information about. For example, in a university database, entities might include "Student," "Course," "Instructor," and "Department."

Define Attributes:

- ✓ For each entity, list the attributes or properties that describe the entity.
- ✓ These attributes represent the information you want to store. For a "Student" entity, attributes could include "StudentID," "Name," "Date of Birth," and "Email."

Identify Relationships:

- ✓ Determine how entities are related to each other.
- ✓ Relationships define how data from one entity is associated with data from another entity.
- ✓ For example, a "Student" may be related to a "Course" through an enrollment relationship.

Cardinality and Participation:

- ✓ Specify the cardinality of each relationship, indicating how many instances of one entity are related to another entity.
- ✓ Common notations include "1:1," "1:N" (one-to-many), and "M:N" (many-to-many).
- ✓ Indicate participation constraints, which specify whether entities are mandatory (total participation) or optional (partial participation) in a relationship.

Create an E-R Diagram:

- ✓ Represent your findings visually using an Entity-Relationship Diagram (ERD).
- ✓ In an ERD, entities are represented as rectangles, attributes as ovals, and relationships as diamond shapes connecting entities.

Add Verb Phrases:

- ✓ Use verb phrases to label the relationships in your diagram.
- ✓ This helps clarify the nature of the relationship.
- ✓ For example, "Enrolls In" for a relationship between "Student" and "Course."

Review and Refine:

- ✓ Review the E-R diagram with stakeholders to ensure that it accurately represents the requirements and relationships in the domain. Make adjustments as necessary.

Abstraction and Generalization:

- ✓ Consider abstract entities and generalization hierarchies, such as using inheritance to model common attributes and relationships among related entities.

Avoid Redundancy:

✓ Strive to eliminate data redundancy by ensuring that information is stored in the most appropriate entity and is not duplicated unnecessarily.

Documentation:

✓ Document the E-R model with descriptions of entities, attributes, and relationships, as well as explanations of any business rules or constraints.

E-R Issues

- ✓ During the process of designing a database using the Entity-Relationship (E-R) model, various issues and challenges can arise. It's essential to address these issues to create a well-structured and efficient database.

Ambiguity and Incompleteness:

Issue: Incomplete or vague requirements can lead to ambiguity in entity definitions and relationships.

Solution: Work closely with stakeholders to clarify requirements and ensure that the model accurately reflects the business domain.

Overcomplication:

Issue: Creating an overly complex E-R model with numerous entities, attributes, and relationships can make the model difficult to understand and maintain.

Solution: Simplify the model by consolidating entities, eliminating unnecessary attributes, and using generalization hierarchies when applicable.

Redundancy:

Issue: Redundant data can lead to data anomalies and increased storage requirements.

Solution: Normalize the model to eliminate data redundancy, and ensure that each piece of information is stored in a single, appropriate place.

Many-to-Many (M:N) Relationships:

Issue: Direct M:N relationships in the E-R model can be challenging to represent in a relational database.

Solution: Use junction tables (also known as association tables or bridge tables) to break down M:N relationships into two 1:N relationships.

Uniqueness and Identifiers:

Issue: Defining unique identifiers (keys) for entities can be complex, especially when composite keys or natural keys are involved.

Solution: Identify suitable keys for each entity, considering uniqueness and stability, and document them clearly.

Suboptimal Cardinality:

Issue: Incorrectly defining the cardinality of relationships can lead to inaccurate data modeling.

Solution: Ensure that you accurately represent how many instances of one entity are related to another. Cardinality should reflect business rules.

Missing Entities or Relationships:

Issue: Failing to identify all relevant entities or relationships can lead to incomplete data models.

Solution: Continuously refine the model and collaborate with stakeholders to uncover missing elements.

Verbosity:

Issue: Overly verbose naming of entities, attributes, and relationships can make the model harder to read and maintain.

Solution: Use concise and descriptive names to improve clarity.

Lack of Documentation:

Issue: Inadequate documentation can result in confusion and difficulty understanding the model.

Solution: Provide comprehensive documentation for the E-R model, including entity and attribute descriptions, relationship explanations, and business rules.

Performance Considerations:

Issue: Failing to consider performance implications during the E-R design stage can lead to inefficient queries and slow database performance.

Solution: Think about query patterns, indexing, and other performance-related factors as you design the model. Denormalization may be necessary in some cases.

Consistency with Real-World Semantics:

Issue: Ensure that the E-R model accurately reflects the real-world semantics of the business domain.

Inconsistencies can lead to data integrity problems.

Solution: Validate the model with domain experts to ensure it aligns with the way the business operates.

WEAK ENTITY

- A weak entity set is an entity set that does not have a primary key attribute, which means it cannot be uniquely identified by its own attributes alone.

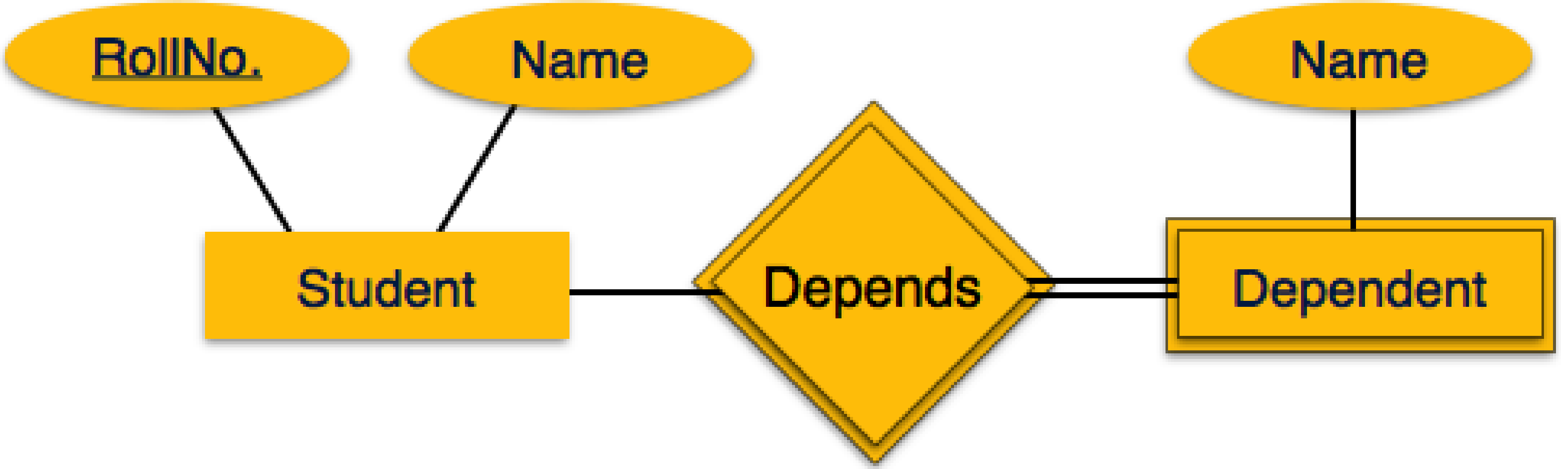
- Instead, a weak entity set depends on another entity set, known as the "owning" or "parent" entity set, to provide the means for identification.

WEAK ENTITY

- Mapping weak entity sets to relational mapping;
  - Create table for weak entity set.
  - Add all its attributes to table as field.
  - Add the primary key of identifying entity set.
  - Declare all foreign key constraints

ER to Relational Mapping

➢Mapping weak entity sets to relational mapping;

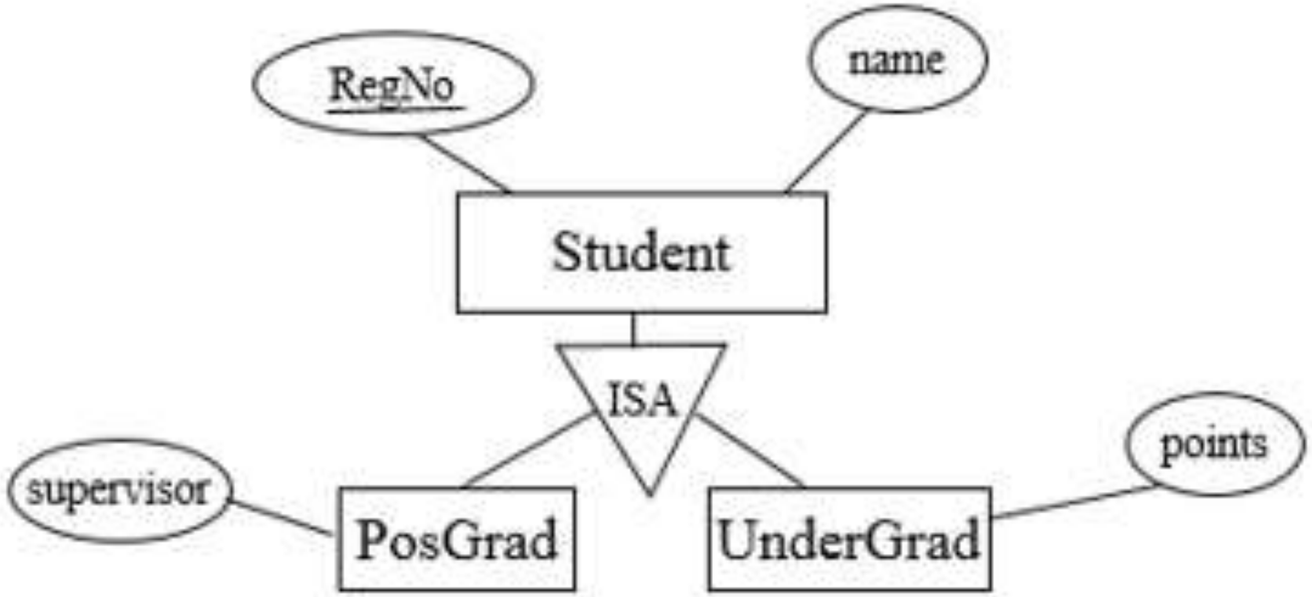


ER to Relational Mapping

- Mapping weak entity sets to relational mapping;
- create table Dependent(RollNo int, Name char(20),Dname char(20), primary key (RollNo), foreign key(RollNo) references Student)

ER to Relational Mapping

- Mapping specialization/generalization to relational tables :



ER to Relational Mapping

- Mapping specialization/generalization to relational tables – Method 1:
  - All the entities in the relationship are mapped to individual tables.
  - Student (Regno, name)
  - PosGrad (Regno, supervisor)
  - UnderGrad (Regno, points)

ER to Relational Mapping

- Mapping specialization/generalization to relational tables – Method 2:
  - Only subclasses are mapped to tables. The attributes in the superclass are duplicated in all subclasses.
  - PosGrad (Regno, name, supervisor)
  - UnderGrad (Regno, name, points)
  - This method is most preferred when inheritance is disjoint and complete

ER to Relational Mapping

- Mapping specialization/generalization to relational tables – Method 3:
  - Only the superclass is mapped to a table. The attributes in the subclasses are taken to the superclass.
  - Student (Regno, name, supervisor, points)
  - This method will introduce null values.

Thank You…….