College of Science and Technology

Royal University of Bhutan

DBS101 Database Systems Fundamentals

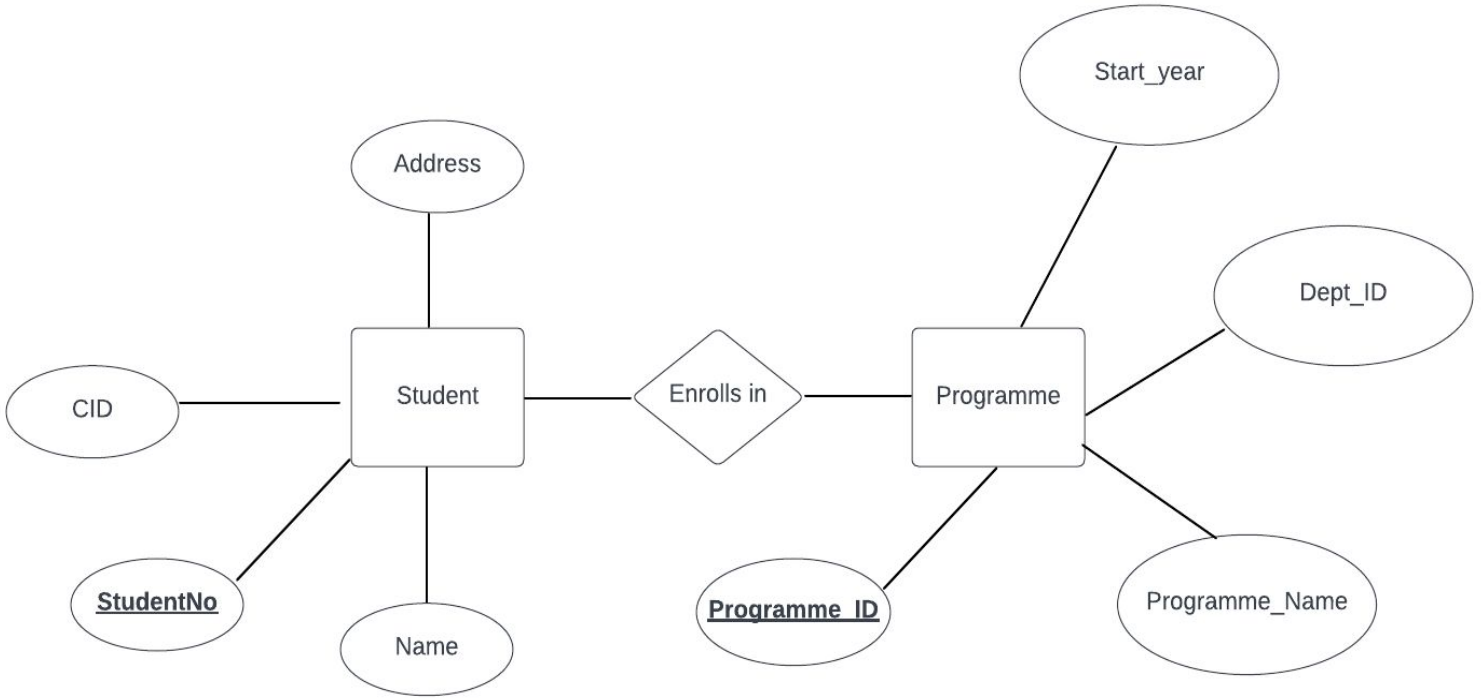
Lesson 3 Part 2

Learning Outcomes

1. Explain the database design process.
2. Understand relational data models.
3. Explain entity relationship diagrams
4. Identify complex attributes.
5. Map cardinalities between entities
6. Determine primary keys in entity sets
7. Remove redundant attributes

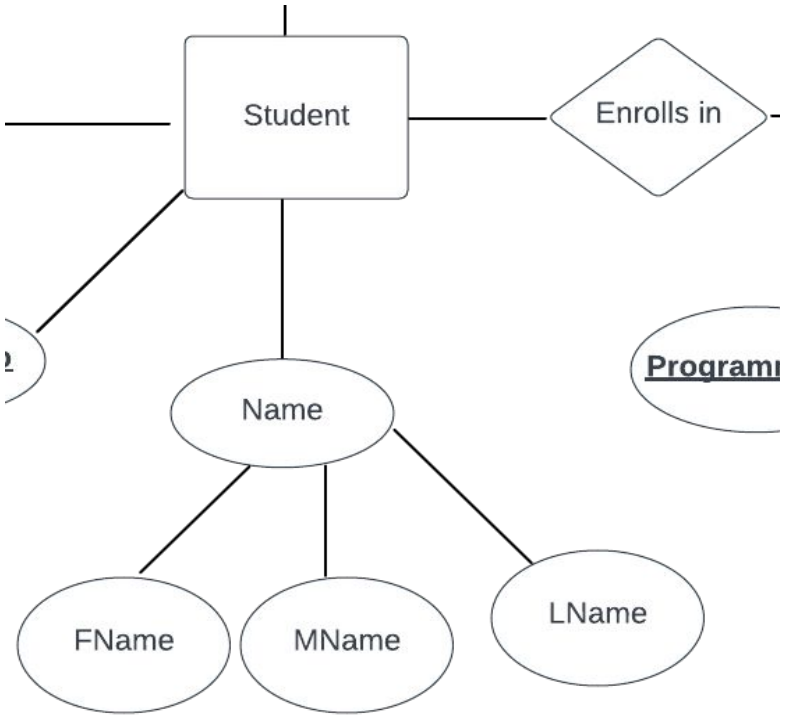
ERD: Complex Attributes

1. Simple Attributes: Attributes are not divided into subparts.



ERD: Complex Attributes

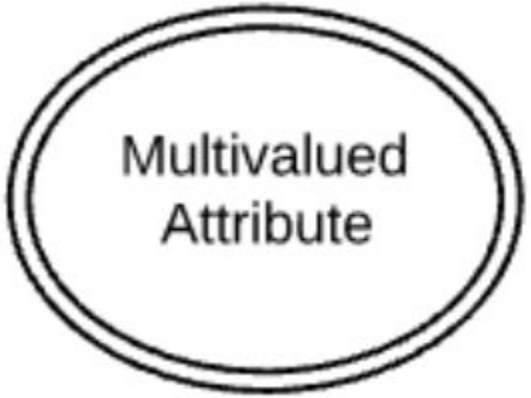
2. Composite Attributes: Attributes can be divided into subparts.



ERD: Complex Attributes

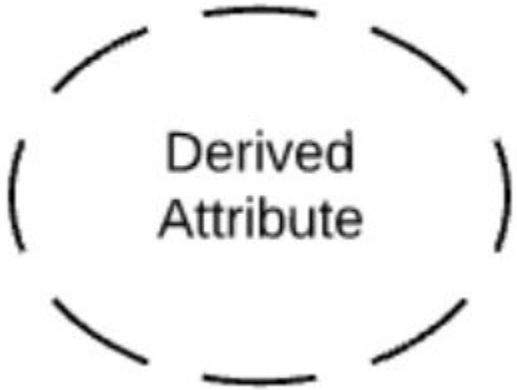
3. Single-valued Attributes: Attributes have single value for a particular entity.

4. Multi-valued Attributes: Attributes have more than one value for a particular entity



ERD: Complex Attributes

5. Derived Attributes: The value for the attribute can be derived from the values of other related attributes or entities.



ERD: Types of Attributes

Link

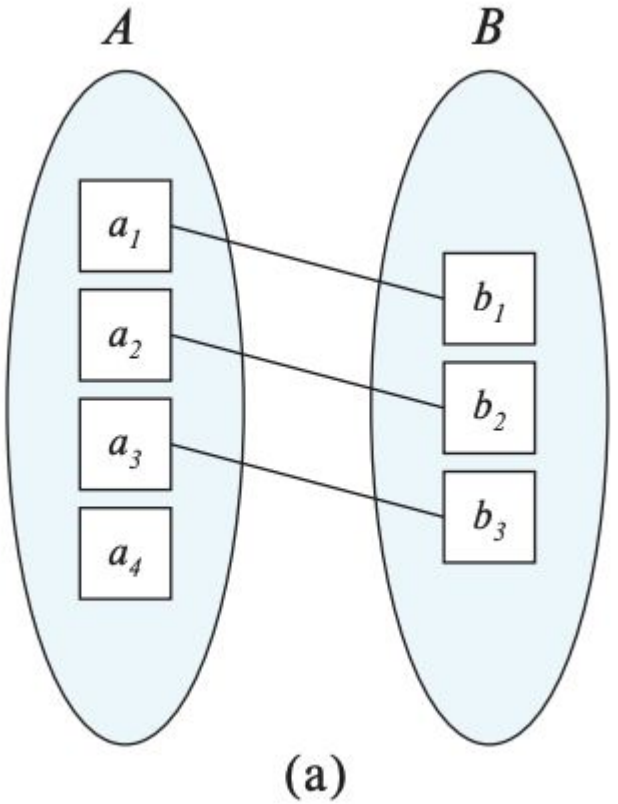
Note: An attribute takes a null value when an entity does not have a value for it. The null value may indicate “not applicable”—that is, the value does not exist for the entity. For example, a person who has no middle name may have the middle initial attribute set to null.

Mapping Cardinalities

Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set.

Mapping Cardinalities

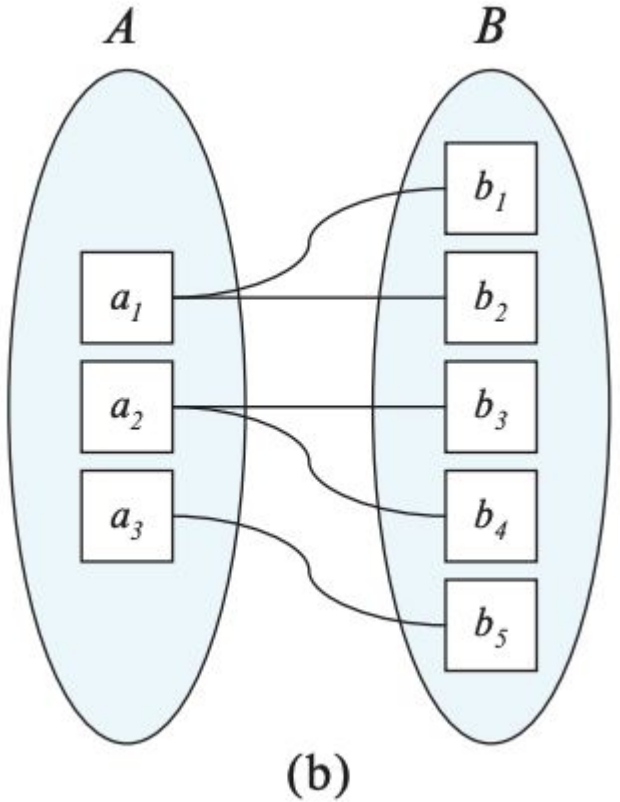
1. One to One: An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.



(a)

Mapping Cardinalities

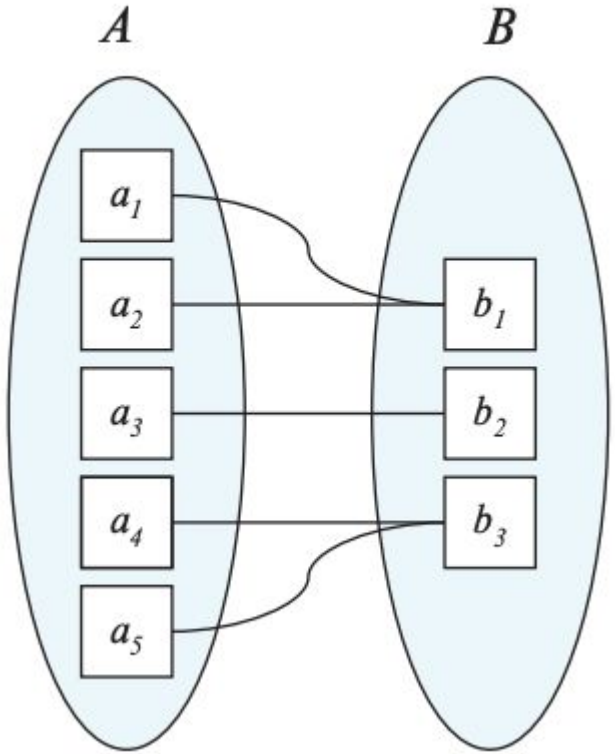
2. One-to-many. An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A.



(b)

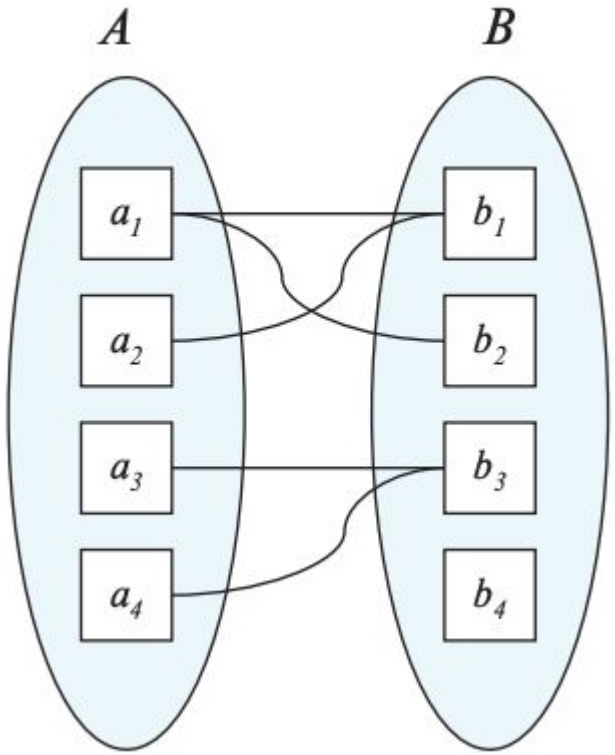
Mapping Cardinalities

3. Many-to-one. An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A.

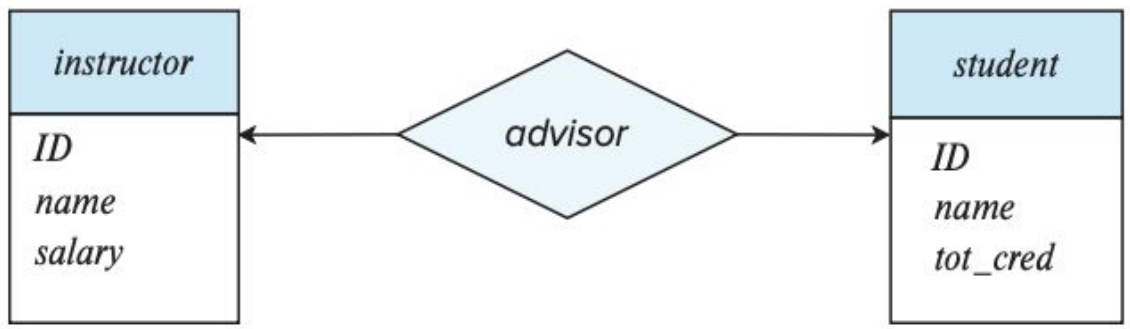


Mapping Cardinalities

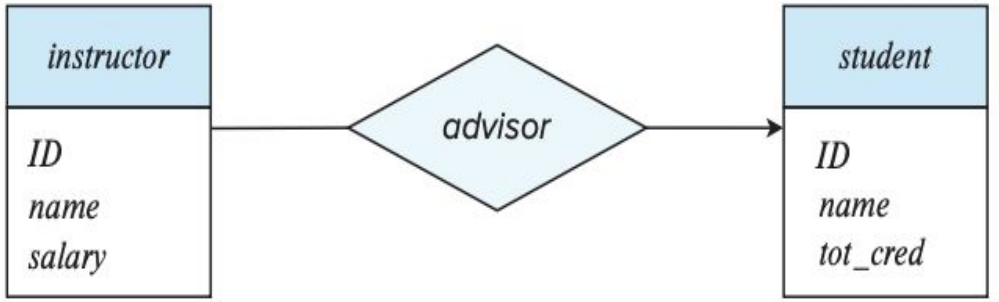
4. Many-to-many. An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A.



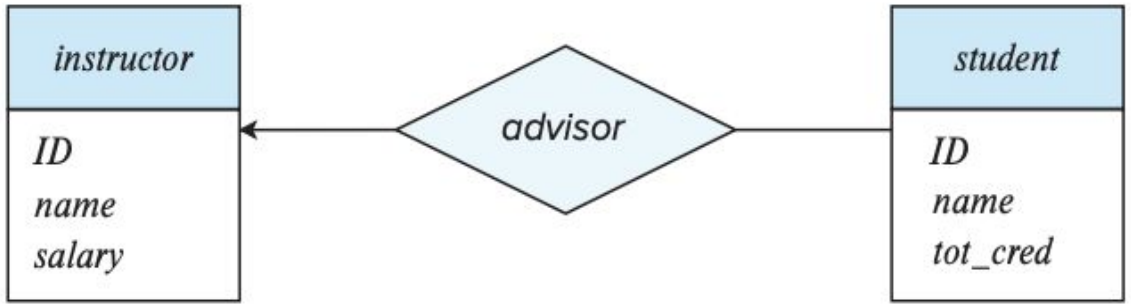
Mapping Cardinalities



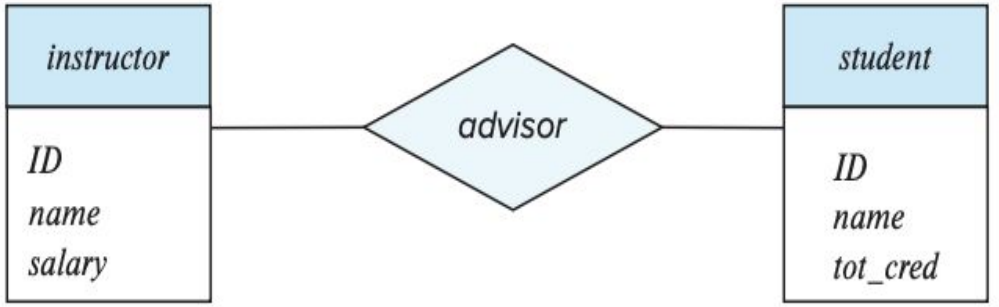
(a) One-to-one



(c) Many-to-one

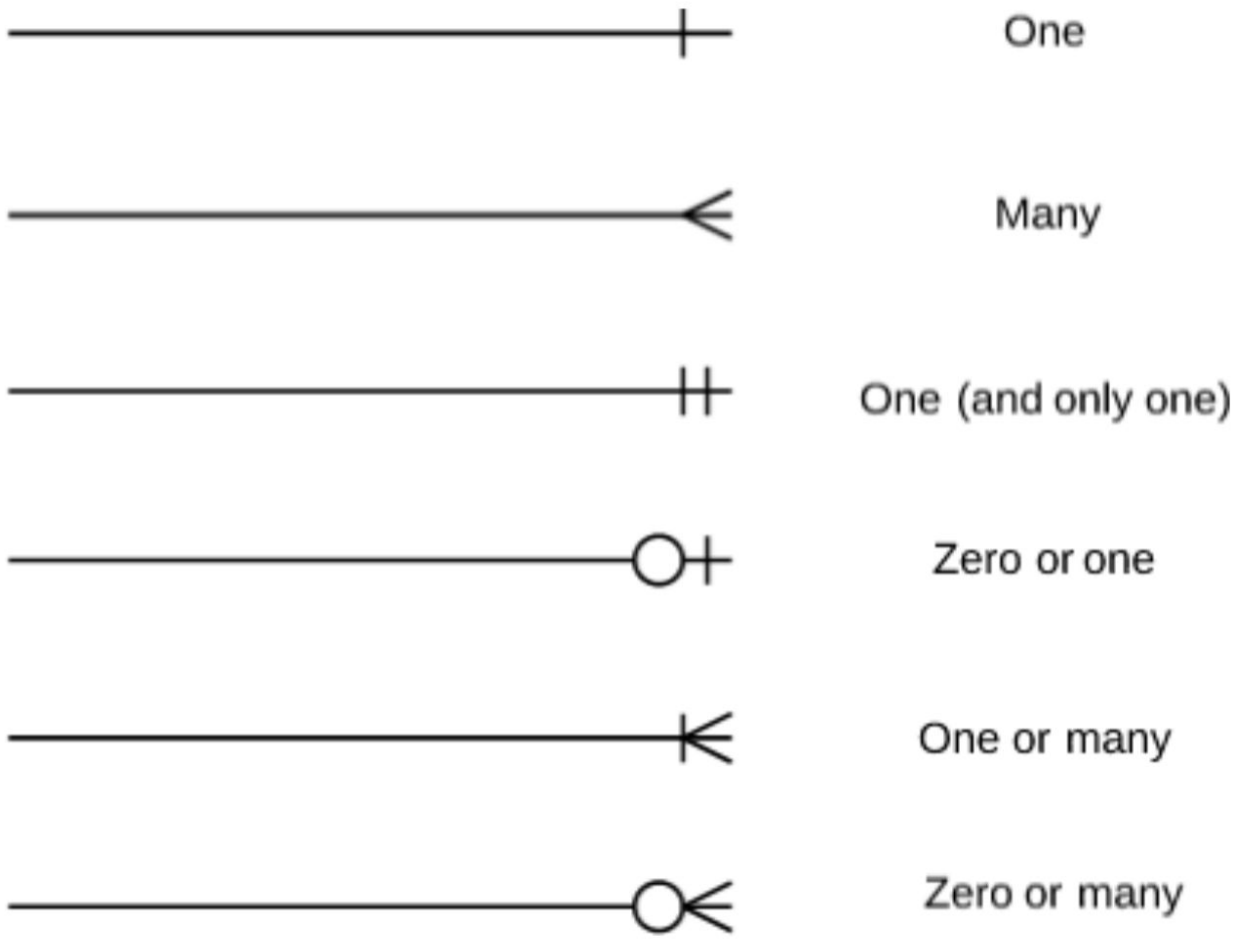


(b) One-to-many



(d) Many-to-many

Mapping Cardinalities: Crow's Foot Notation



Relationship Participation

There are two kinds of participation conditions: mandatory and optional. Most entities are involved in binary relationships, so it follows that there are four main types of membership relationships:

1. Mandatory for both entities

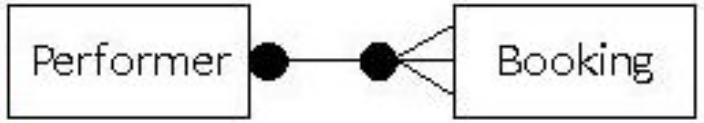


Figure 7.6

2. Mandatory for one entity, optional for the other

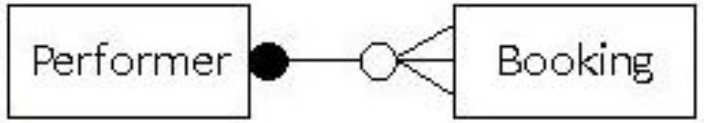


Figure 7.7

3. Optional for one entity, mandatory for the other

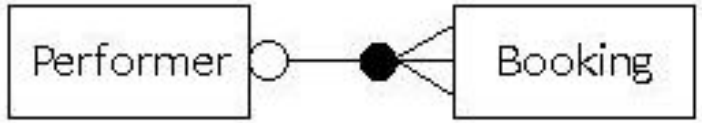


Figure 7.8

4. Optional for both entities

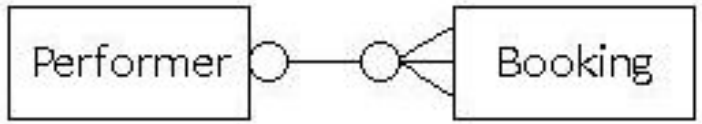


Figure 7.9

Weak Entities

Link to weak entities

Removing Redundant Attributes in Entity Sets

A good entity-relationship design does not contain redundant attributes.

5 minutes Break !!!!

Extended E-R Features

1. **Specialization**

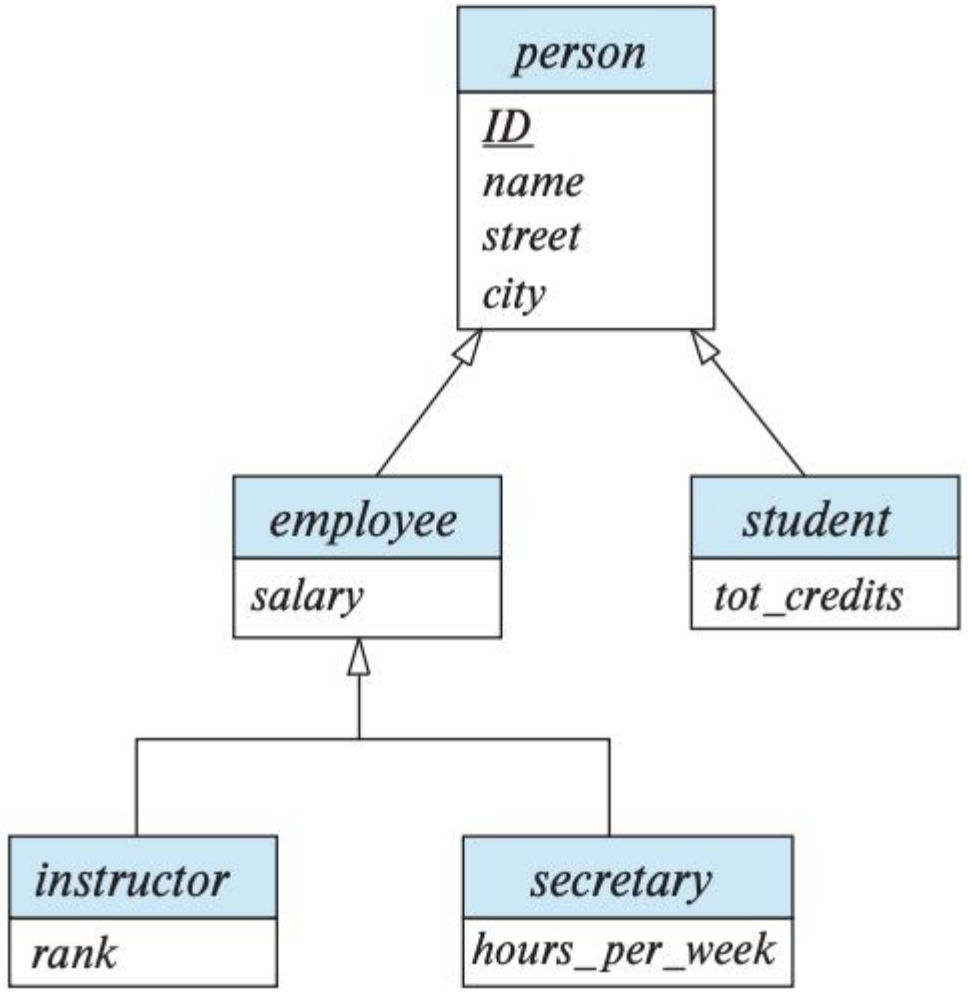
   An entity set may include subgroupings of entities that are distinct in some way from other entities in the set.

2. **Generalization**

   A containment relationship that exists between a higher-level entity set and one or more lower-level entity sets.

Extended E-R Features

Link to Example



Attribute Inheritance

The attributes of the higher-level entity sets are said to be inherited by the lower-level entity sets.

Whether a given portion of an E-R model was arrived at by specialization or generalization, the outcome is basically the same:

- A higher-level entity set with attributes and relationships that apply to all of its lower-level entity sets.
- Lower-level entity sets with distinctive features that apply only within a particular lower-level entity set.

Constraints on Specialisation/ Generalisation

1. **Membership constraints**
   Condition defined: Membership of a specialization/generalization relationship can be defined as a condition in the requirements e.g. tanker is a ship where cargo = ‘shoes’
   User defined: Sometimes the designer can define the superclass-subclass relationship. This can be done to simplify the design model or represent a complex relationship that exists between entities.

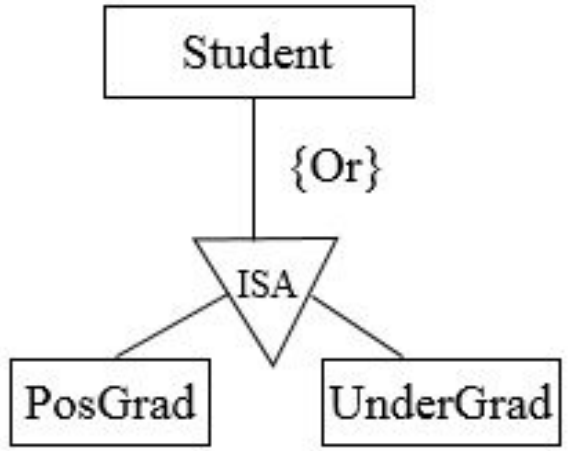
Constraints on Specialisation/ Generalisation

2. Disjoint constraints

Disjoint: The disjoint constraint only applies when a superclass has more than one subclass. If the subclasses are disjoint, then an entity occurrence can be a member of only one of the subclasses, e.g. postgrads or undergrads you cannot be both. To represent a disjoint superclass/subclass relationship.

Constraints on Specialisation/ Generalisation

2. Disjoint constraints

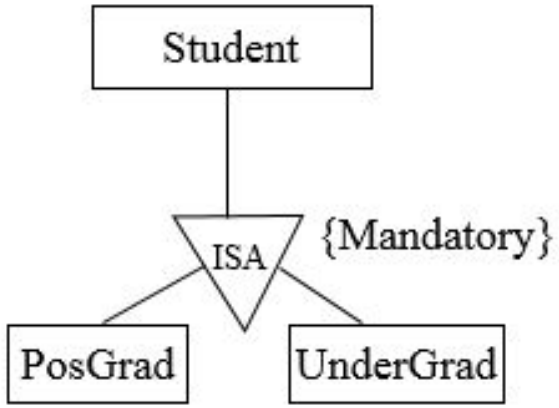


Overlapping: This applies when an entity occurrence may be a member of more than one subclass, e.g. student and staff,some people are both. {OR} is used to represent the overlapping specialization/generalization relationship in the ER diagram.

Constraints on Specialisation/ Generalisation

2. Completeness constraints:

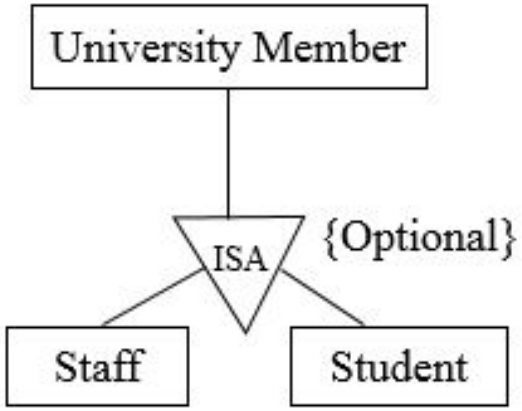
a.Total:Each superclass (higher-level entity) must belong to subclasses (lower-level entity sets), e.g. a student must be post grad or undergrad. To represent completeness in the specialization/generalization relationship, the keyword {Mandatory} is used.



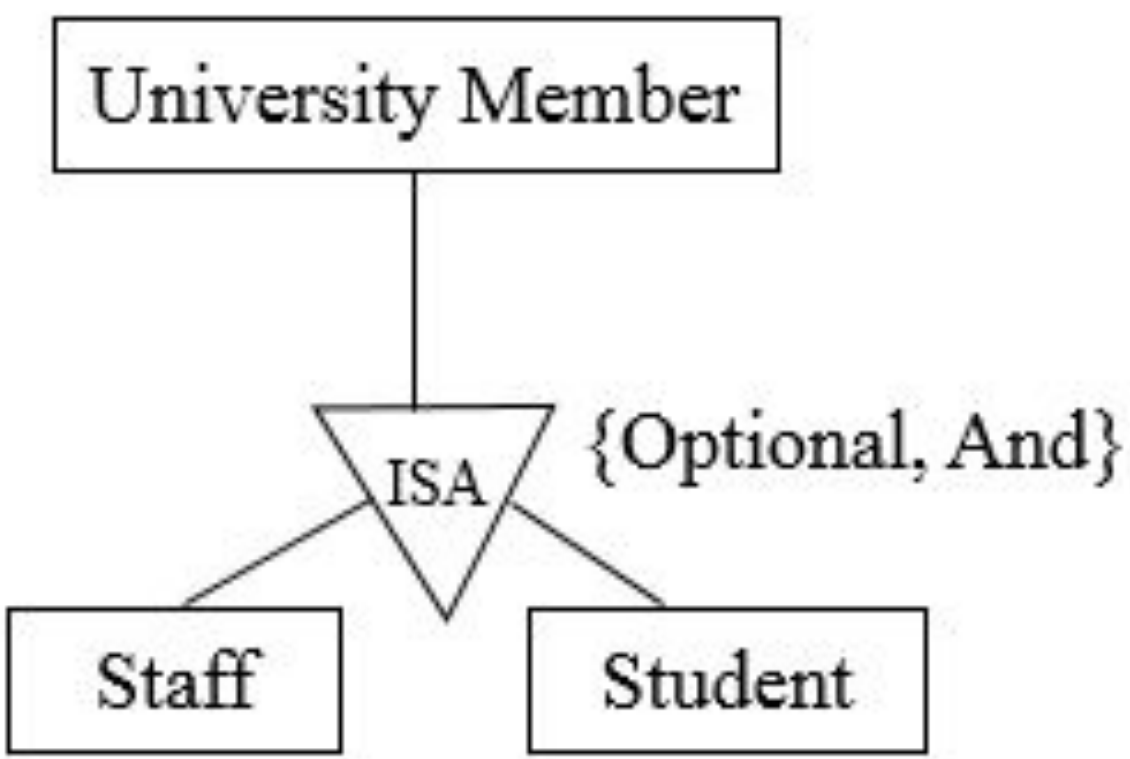
Constraints on Specialisation/ Generalisation

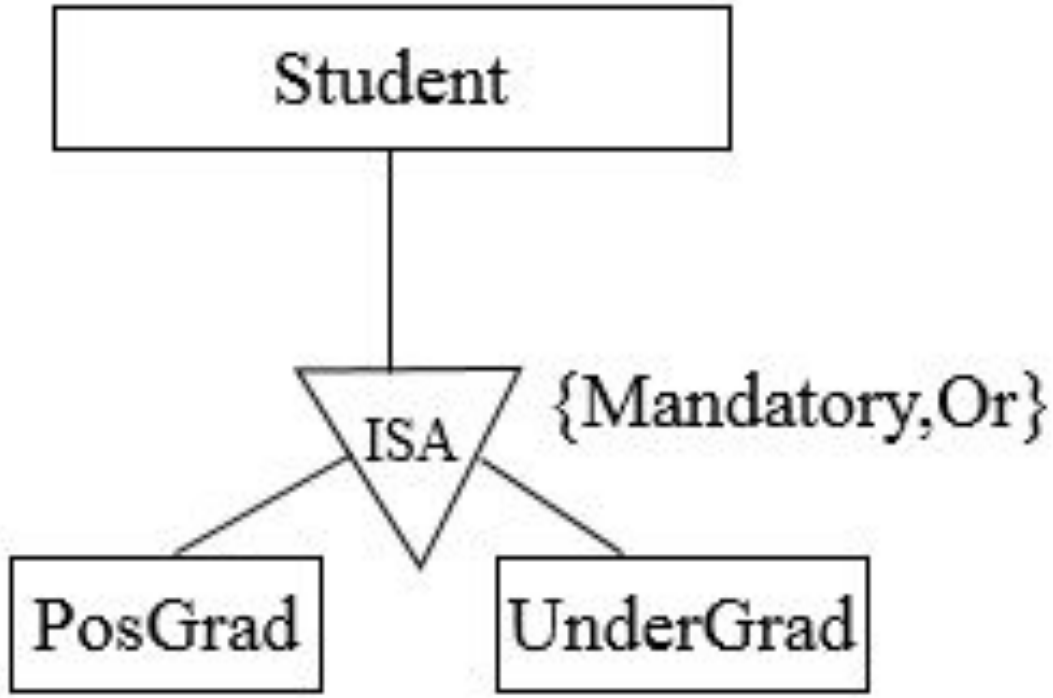
2. Completeness constraints:

b.Partial:Some superclasses may not belong to subclasses (lower-level entity sets), e.g. some people at UCT are neither student nor staff. The keyword {Optional} is used to represent a partial specialization/generalization relationship.



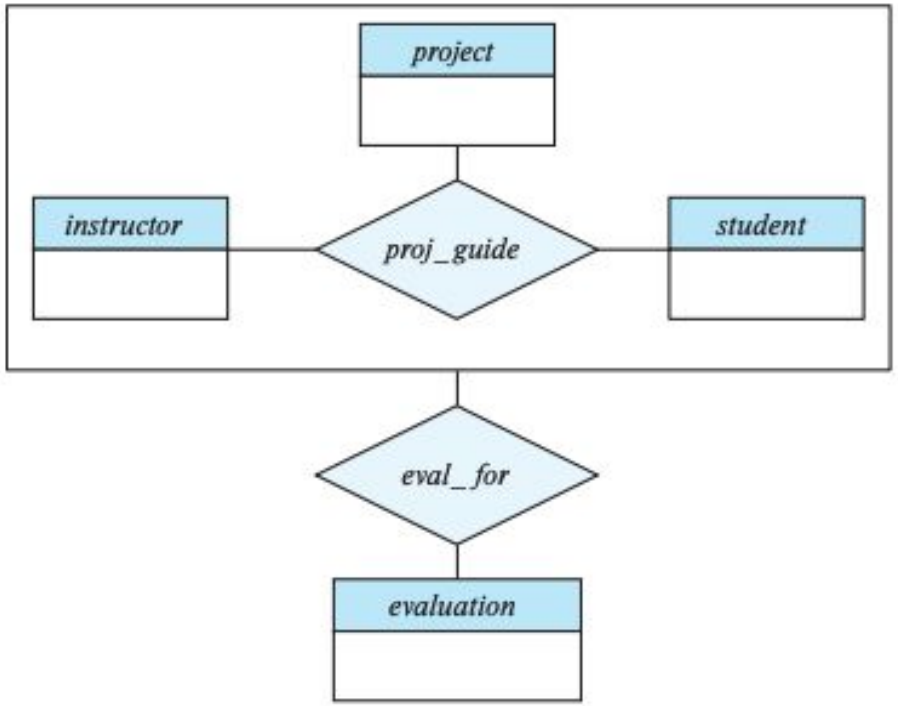
Constraints on Specialisation/ Generalisation





Aggregation

Limitation of E-R model: Cannot express relationships among relationships.



Aggregation

Aggregation is an abstraction through which relationships are treated as higherlevel entities.

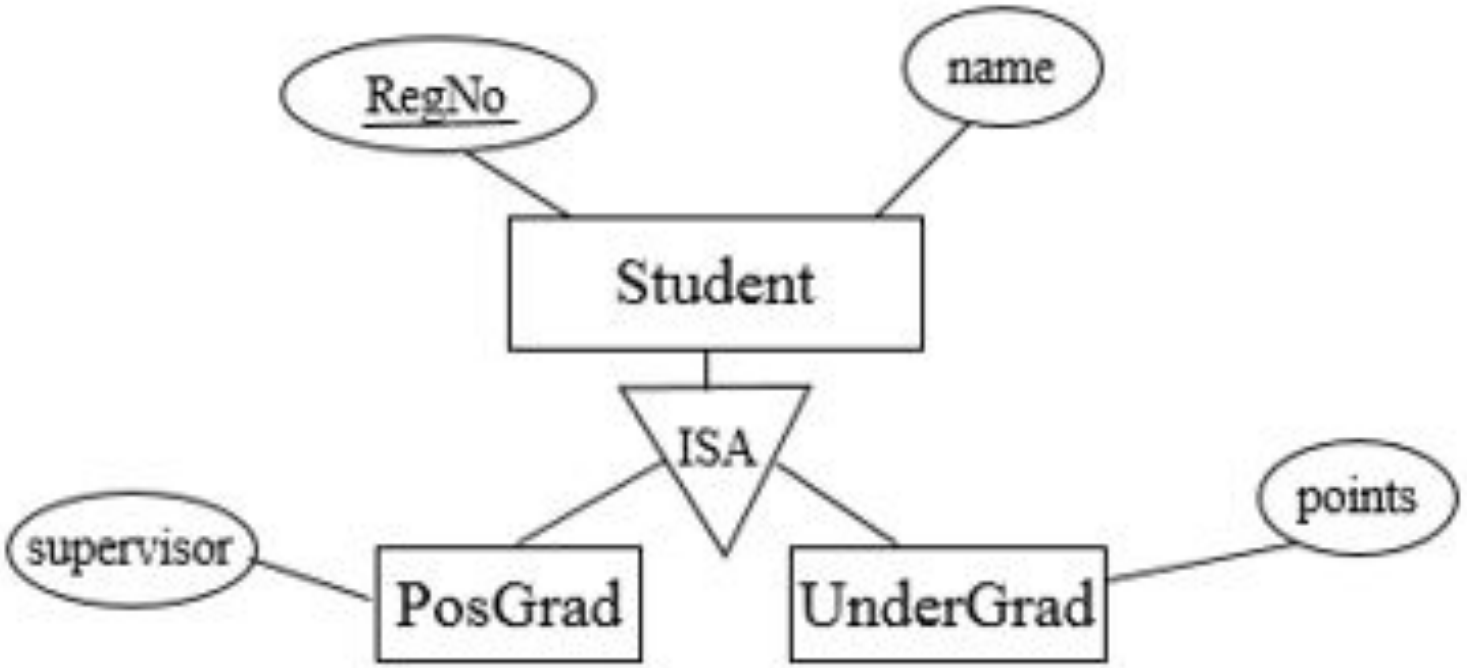
Reducing E-R Diagrams to Relational Schemas

Link to study guide

Link to Example

Mapping specialisation/generalisation to relational tables

3 methods



Mapping specialisation/generalisation 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)

Mapping specialisation/generalisation 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, e.g. every student is either PosGrad or UnderGrad and nobody is both.

Mapping specialisation/generalisation 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. When we insert an undergraduate record in the table, the supervisor column value will be null. In the same way, when we insert a postgraduate record in the table, the points value will be null.

Mapping aggregations to relational tables

No separate relation is required to represent the aggregation; the relation created from the defining relationship is used instead.

Guidelines for deciding between Entities and Attributes

- Attributes directly describe the entity
- Entities can have their own attributes and relationships with other entities
- Generally, treat something as an entity if you may want to store additional information about it

Guidelines for deciding between Entities and Relationships

- Use relationships to describe actions between entities
- Can transform attributes into relationships if appropriate

Extra:

Usually simpler to use binary relationships, but n-ary relationships can sometimes better model complex interconnections

Can transform n-ary relationships into binary by creating new entities, but this can increase complexity

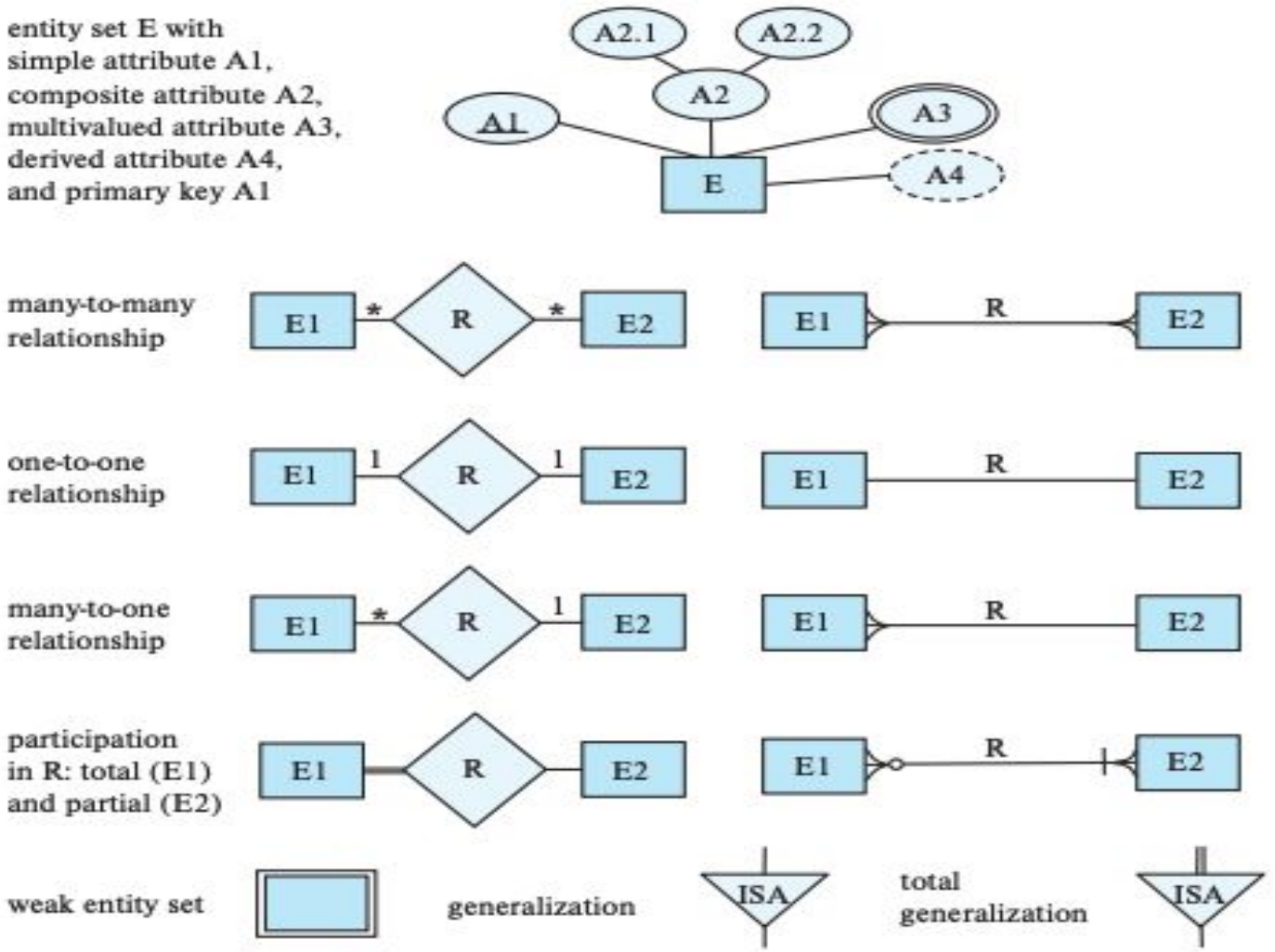


Figure 6.27 Alternative E-R notations.

Unified Modeling Language UML

Entity-relationship diagrams help model the data representation component of a software system.

Other components include models of user interactions with the system, specification of functional modules of the system and their interaction, etc.

The **Unified Modeling Language (UML)** is a standard developed under the auspices of the Object Management Group (OMG) for creating specifications of various components of a software system.

Unified Modeling Language UML

Entity-relationship diagrams help model the data representation component of a software system.

Other components include models of user interactions with the system, specification of functional modules of the system and their interaction, etc.

The **Unified Modeling Language (UML)** is a standard developed under the auspices of the Object Management Group (OMG) for creating specifications of various components of a software system.

Unified Modeling Language

Some of the parts of UML are:

- Class diagram
- Use case diagram
- Activity diagram
- Implementation diagram

Topic Submission for Practical Assessment: 27/2/24

Flipped Class 2: Will Begin at 3:30 pm

References

Entity-Relationship Diagram Symbols and notation. (n.d.). Lucidchart. https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning

GfG. (2023, November 6). *Constraints on relational database model*. GeeksforGeeks. https://www.geeksforgeeks.org/constraints-on-relational-database-model/