ENTITY RELATIONSHIP MODEL (ER MODEL)

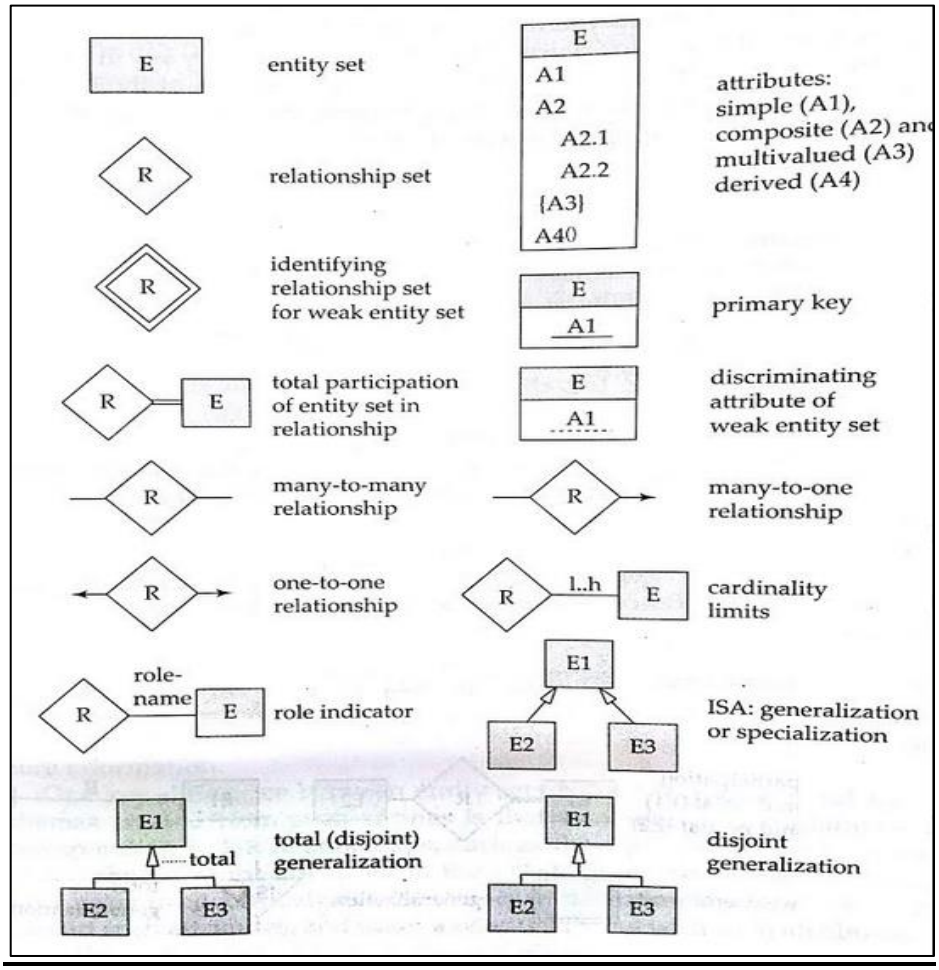
E-R Model:

- The Entity-Relationship(E-R) data model was developed to facilitate database design.
- The E-R model is very useful in mapping the meanings and interactions of real-world enterprises onto a conceptual schema.

Basic concepts employed in E-R Data model:

1. Entity sets
2. Relationship sets
3. Attributes
4. Degree of a Relationship set
5. Weak Entity Sets
6. Constraints
   - Mapping cardinalities
   - Participation constraints
   - Keys

Notations used in E-R Modeling



1. Entity Sets

- ***An entity*** is a 'thing' or 'object' in the real world that is distinguishable from all other objects. An entity has a set of properties.
  - **Example**: *Each person in a university is an entity.*
- ***An entity set*** is a set of entities of the same type that share the same properties.
  - **Example**:*The set of all students in the university is an entity set.*

2. Relationships sets

- A relationship is an association among several entities.
- A relationship set is a set of relationships of the same type.
  - **Example**: Consider two entity sets *student* and *section*. The relationship set

advisor denotes the association between instructors and students.

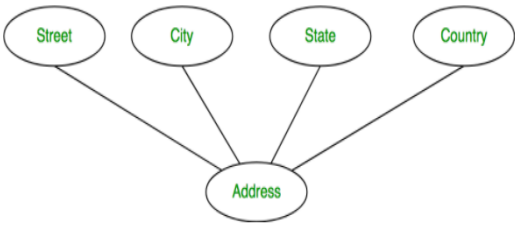
3. Attributes

- An attribute is defined as the properties that describe an entity.
- For each attribute, there is a set of permitted values, called the ***domain*** or ***value set*** of that attribute.
    - **Example:** An EMPLOYEE entity may be described by the employee's name, age, address, salary, and job.

Types of Attributes:

- **Simple Attribute:** An Attribute that is not divisible.
  - Example: Register number, Aadhar number
  - In E-R Diagram, it is represented as

- **Composite Attribute:** An Attribute that is divisible into smaller subparts
  - Example: Address. The address attribute can be sub-divided into street, address, city,state and pincode
  - In E-R Diagram, it is represented as



- **Single valued Attribute:** An Attribute that has a single value for a particular entity
  - Example: Date of birth, CGPA
- **Multi valued Attribute:** An Attribute that have a set of values for the same entity
  - Example: Phone number, email address
  - In E-R Diagram, it is represented as

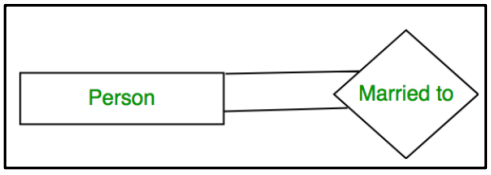
Stored Attribute: An Attribute that is already stored in the Database
  - Example: Date of birth, Date of Joining
- **Derived Attribute:** An Attribute that is derived from the stored Attribute
  - Example: Age, Experience
  - In E-R Diagram, it is represented as

4. Degree of a Relationship set

- The number of different entity sets participating in a relationship set is called as degree of a relationship set.

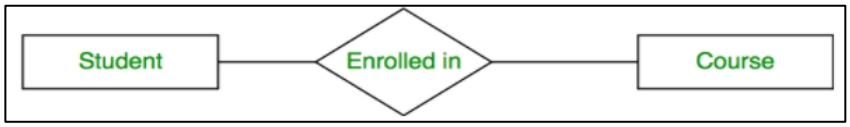
1. Unary Relationship

- ONE entity set participating in a relationship



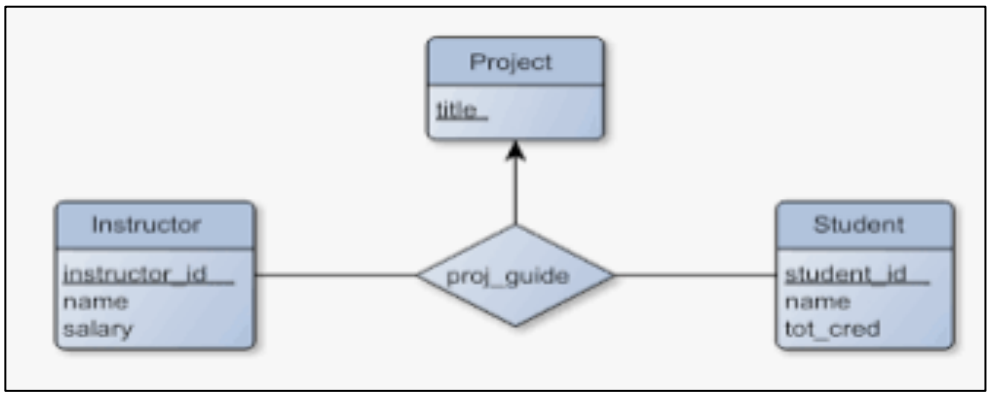
2. Binary Relationship

- TWO entity set participating in a relationship



3. n-ary Relationship

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



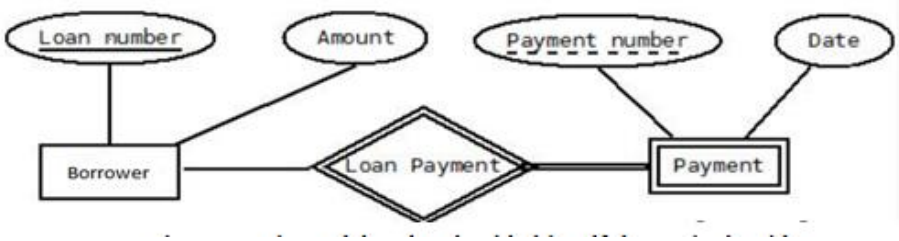
5. Weak Entity And Strong Entity Sets

Strong Entity Set:

- An entity set that has a primary key is termed as a strong entity set.

Weak Entity Set:

- An entity set that does not have sufficient attributes to form a primary key is termed as weak entity set.
- A weak entity set must be associated with another entity set, called ***identifying or owner relationship set***
- ***Weak entity set must have total participation in this identifying relationship set***.



The discriminator of the weak entity set is underlined with a dash, rather than a solid line

- The relationship set connecting the weak entity set to the identifying strong entity set is denoted by a double lined rectangle.

6. Constraints

- An E-R enterprise schema may define certain constraints to which the contents of the database must conform.

- The constraints in E-R model are:
  - Mapping cardinalities
  - Participation constraints.
  - Keys.

Mapping Cardinalities

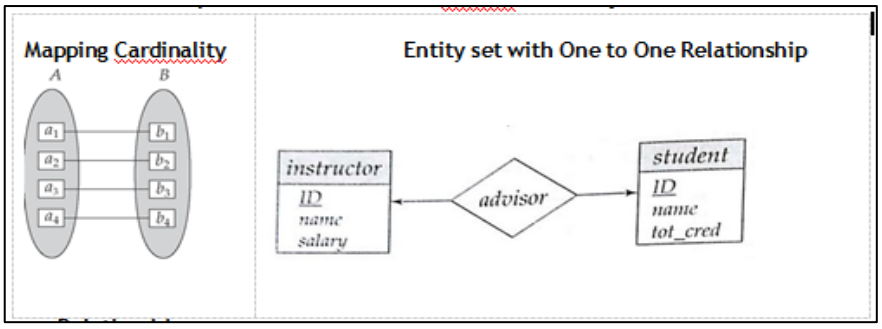
- Mapping cardinalities/ cardinality ratios express the number of entities to which another entity can be associated via a relationship set.
- For a ***binary relationship set R between entity sets A and B***, the mapping cardinality must be one of the following types:

i) One to one Relationship

- An entity in *A is associated with* ***atmost*** *one entity in B.*

One-to-one relationship between an *instructor* and a *student* :

- A student is associated with at most one *instructor* via the relationship *advisor*
- A *student* is associated with at most one instructor

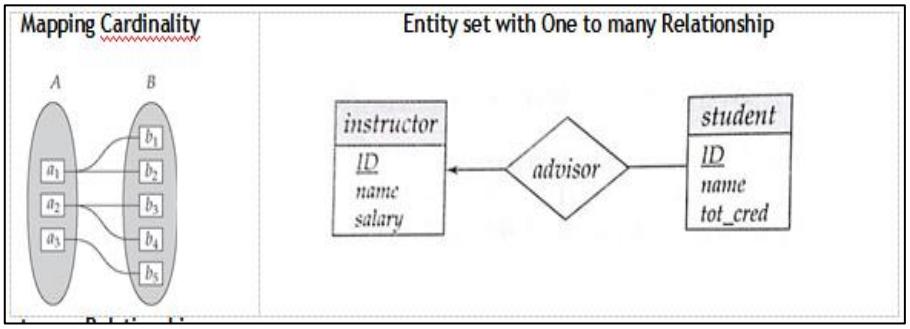


ii) One to many Relationship

- An entity in A is associated with any number entities in B.
- But, an entity in B can be associated with ***atmost*** one entity in A.

One-to-Many relationship between an *instructor* and a *student*:

- An instructor is associated with several (including 0) students via *advisor*
- A student is associated with at most one instructor via advisor

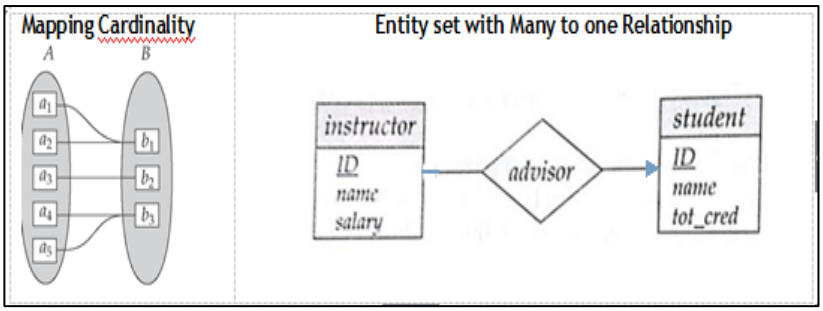


iii) Many to one Relationship

- An entity in A is associated with atmost one entity in B.
- But an entity in B can be associated with any number entities in A.
-

In a many-to-one relationship between an *instructor* and a *student*,

- an instructor is associated with at most one student via *advisor*,
- a student is associated with several (including 0) instructors via *advisor*

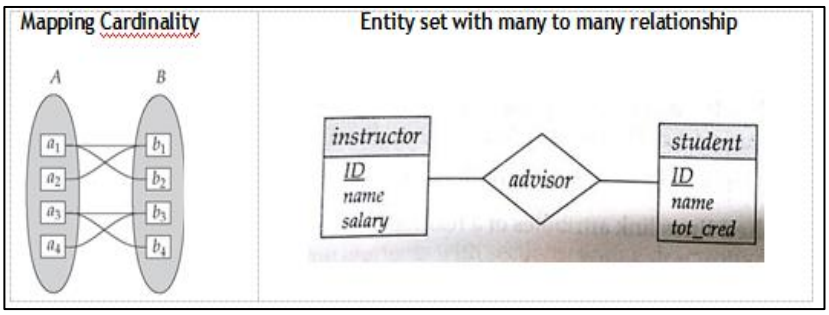


iv) Many to many Relationship

- An entity in A is associated with any number of entities in B.
- An entity in B can be associated with any number of entities in A.

In a many-to-one relationship between an *instructor* and a *student*

- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*



Participation Constraints:

1. Total Participation Constraint (indicated by double line)

- Every entity in the entity set participates in at least one relationship in the relationship set
- ***Example: participation of loan entity in borrower relationship is total.***
- Every loan must have a customer associated to it via borrower.

2. Partial Participation Constraint

- Some entities may not participate in any relationship in the relationship set.
- ***Example: participation of customer entity in borrower relationship is partial.***

Keys:

- Keys specify how entities within a given entity set are distinguished. A key attribute uniquely identifies the entity.
- The concepts of super key, candidate key and primary key are applicable to entity sets just as they are applicable to relation schemas.

Types of Keys:

- Super Key
- Candidate Key

- Primary Key
- Foreign Key

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A **candidate key** of an entity set is a minimal super key that uniquely identifies each occurrence of an entity type. Candidate key can never be NULL.
- A **Primary Key** is the candidate key that is selected to uniquely identify an entity type. Although several candidate keys may exist, one of the candidate keys is selected to be the primary key.
- A **foreign key** is an attribute of an entity set that refers to the primary key attribute of another entity set. It enforces referential integrity constraints.

E-R Model Advantages

- Conceptual simplicity
- Visual representation
- Effective communication tool
- Integrated with the relational data model

E-R Model Disadvantages

- Limited constraint representation
- Limited relationship representation
- Loss of information content

Extended E – R Model

The extended E-R features are:

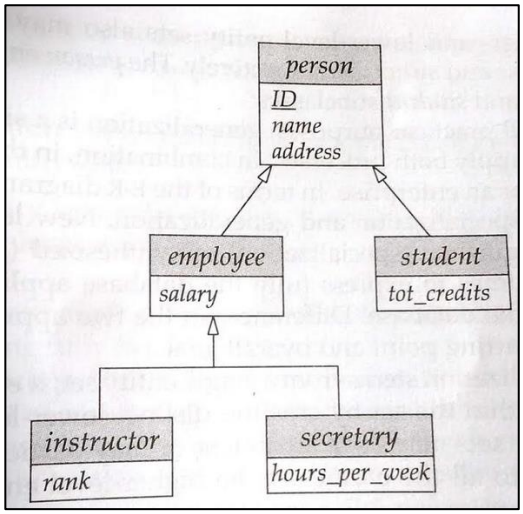
- **Specialization**
- **Generalization**
- **Attribute Inheritance**
- **Aggregation**

Specialization

- "Specialization is the process of designating sub groupings within an entityset."
- It is a top down design process.

Generalization

- "Generalization is the process in which multiple entity sets are synthesized into higher-level entity set on the basis of common features."
- It is a bottom – up design process.
- This approach results in the identification of a generalized super class from the original subclasses.



Generalization Constraints

1) "To determine which entities can be members of a given Lower-level entityset."

- **Condition Defined or predefined:**
  - In condition-defined lower-level entity sets, membership is evaluated on the basis of whether or not an entity satisfies an explicit condition or predicate.
- **User Defined:**
  - User defined lower-level entity sets are not constrained by a membership condition.
  - The database user assigns entities to a given entity set.

2) "To determine whether or not entities may belong to more than one lower-level entity set."

- **Disjoint:**
  - A disjoint generalization requires that an entity belong to no more than one lower –level entity set.
- **Overlapping:**
  - In Overlapping generalizations, the same entity may belong to more than one lower-level entity set.

3) Completeness constraint on Generalization or specialization:

"To determine whether or not an entity in the higher-level entity set must belong to atleast one of the lower-level entity set within the generalization or specialization."

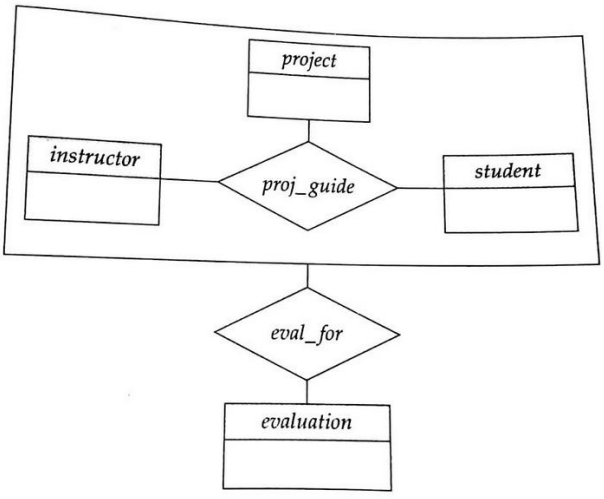
- **Total Generalization:**
  - Each higher-level entity must belong to a lower-level entity set.
- **Partial Generalization:**
  - Some higher-level entities may not belong to any lower-level entity set.

Attribute Inheritance:

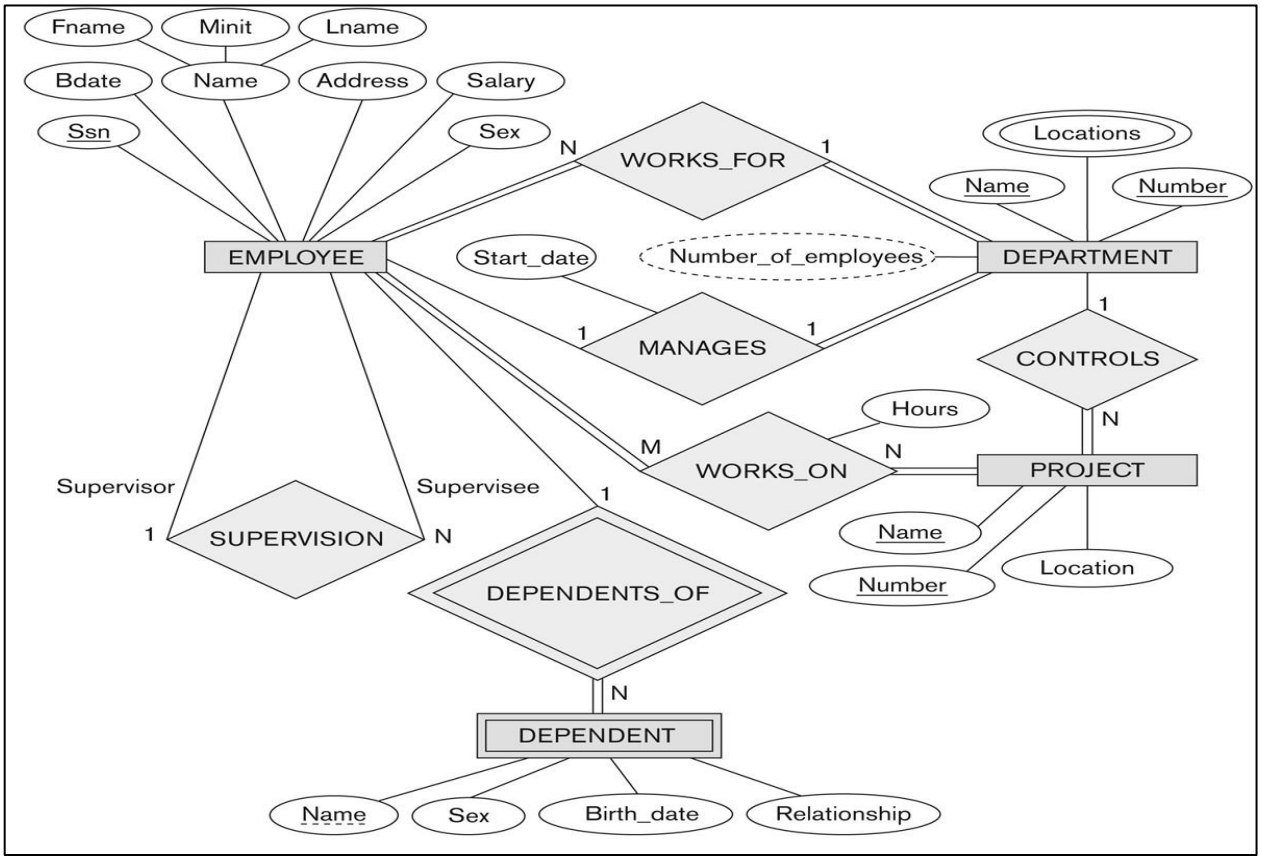
- A crucial property of the higher and lower level entities created by specialization and generalization is *attribute inheritance*.
- The attributes of the higher-level entity sets are said to be inherited to the lower-level entity sets.

Aggregation:

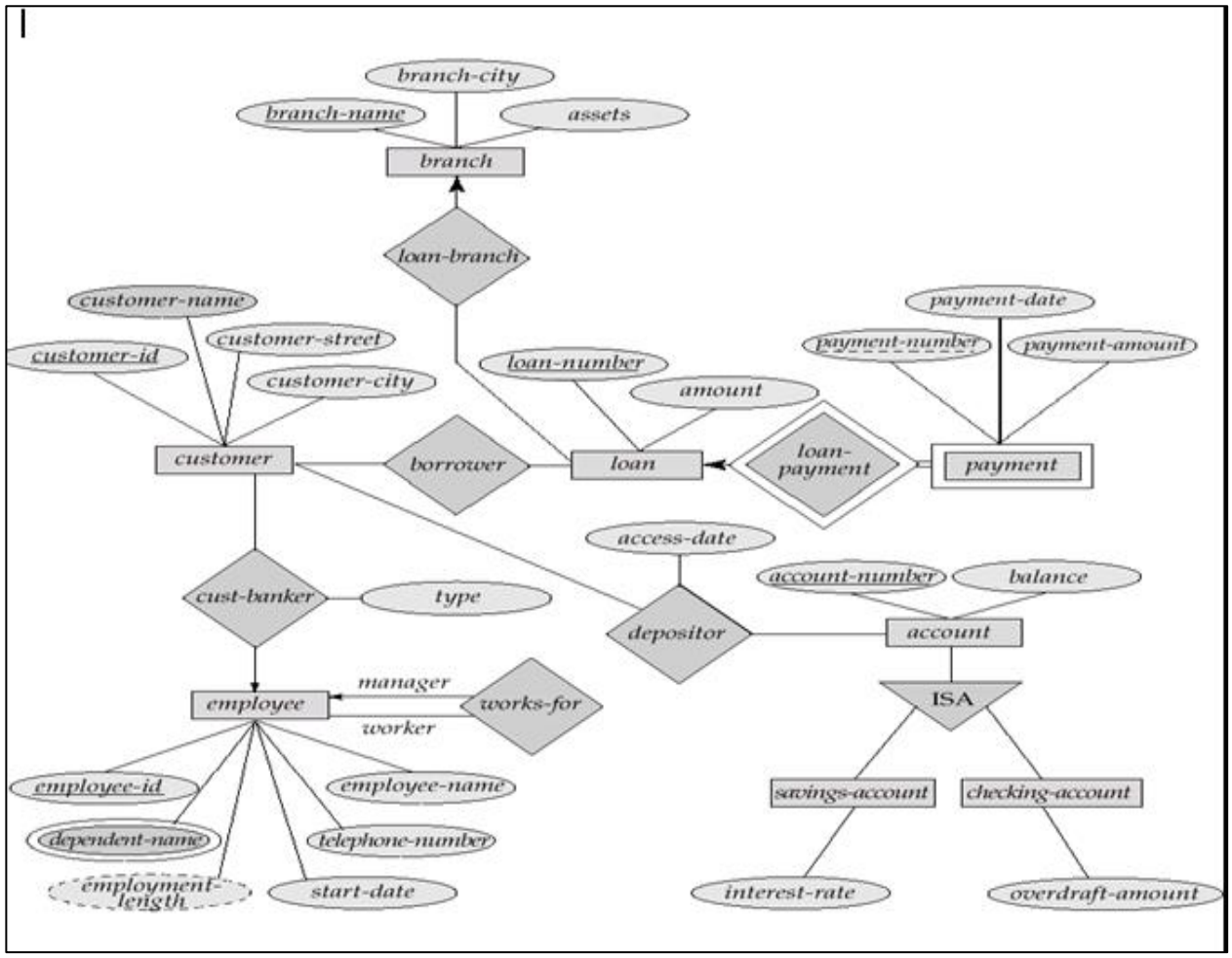
- Aggregation is an abstraction through which relationships are treated as higher level entities.
- It is used when we have to model a relationship involving (entity sets and) a *relationship set*.
- *Aggregation* allows us to **treat a relationship set as an entity set** for purposes of participation in (other)relationships.



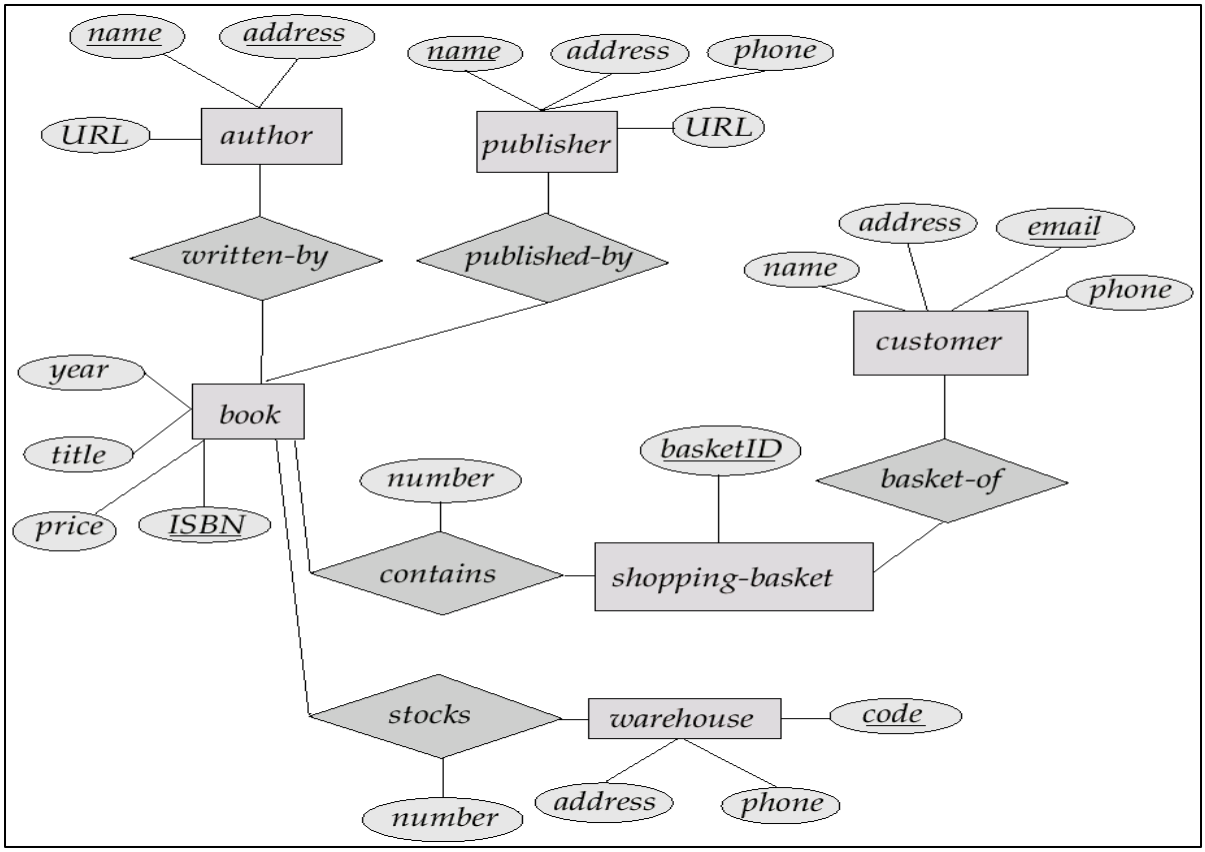
E-R Diagram for company database:



ER Diagram for Banking Enterprise



E-R Diagram for Library Database



==========

ER TO RELATIONAL MAPPING / REDUCING ER DIAGRAM TO RELATIONAL SCHEMAS

- Conversion of ER diagram to relational schemas is a SEVEN step process
    1. Mapping of Regular entity types
    2. Mapping of Weak entity types
    3. Mapping of binary 1:1 relationship types
    4. Mapping of binary 1:n and n:1 relationship types
    5. Mapping of binary m:n relationship types
    6. Mapping Multi-valued attributes
    7. Mapping n-ary relationship types

Step1: Mapping of Regular Entity Types

- For each strong/regular entity type, create a corresponding relation that includes all simple attributes.
- If it is a composite attribute, all simple attributes of the composite attribute are included in the relation.
- Choose a primary key attribute for the relation.

Step 2: Mapping of Weak Entity Types

- For each weak entity type, create a corresponding relation that includes all simple attributes.
- Add the primary key of owner entity type as a foreign key in the relation created for the weak entity type.
- The primary key for this corresponding relation is the combination of the primary key of the owner entity type and the partial key of the weak entity type.

Step 3: Mapping binary 1:1 Relationship types

- Choose one relation (table) as S and other as T (Better if S has total participation)
- Add to S all the simple attributes of the relationship
- Add the primary key attribute of T as a foreign key in S.

Step 4: Mapping binary 1:n and n:1 Relationship types

- Choose the relation (table) on the n-side of the relationship as S, Other as T.
- Add to S all the simple attributes of the relationship
- Add the primary key attribute of T as a foreign key in S.

Step 5: Mapping binary m:n Relationship types

- Create a new relation (table) S. Table S is termed as relationship relation.
- Add to S all simple attributes of the relationship.
- Add the primary keys of both the relations as the foreign keys in S. Their combination forms the primary key of S.

Step 6: Mapping multivalued attributes

- Create a new relation S.
- The primary key of the corresponding relation is added as a foreign key.
- Add the multivalued attribute to S.
- The combination of all attributes in S forms the primary key.

Step 7: Mapping of n-ary relationship types

- Create a new relation (table) S to represent the relationship.
- Add to S all simple attributes of the relationship.
- Add the primary keys of participating relations as the foreign keys in S.

ER model of University Database Application

Entity Types for University Database applications may be

- Instructor
- Department
- Student
- Course
- Dependent (Weak entity set)

Entity Type: Instructor

Attributes: id,name, dept, salary

Primary key: id

Entity Type: Department

Attributes: deptname, building, budget

Primary key : deptname

Entity Type: Student

Attributes: regno, phone, name, totalcredit, cgpa

Primary key: regno

Entity Type: course

Attributes: courseid, title, credits

Primary key: courseid

Weak Entity Type: Dependent

Attributes: name, relationship, location

Partial key: name

Relationship sets for University Database applications may be

Relationship set: worksfor

Participating entity types: instructor and department

Mapping cardinality: many to one

Relationship set: manages

Participating entity types: instructor and department

Mapping cardinality: one to one

Relationship set: teaches

Participating entity types: instructor and student

Mapping cardinality: many to many

Relationship set: enrolls

Participating entity types: student and course

Mapping cardinality: one to many

Relationship set: depends on

Participating entity types: student and dependent

Mapping cardinality: one to many

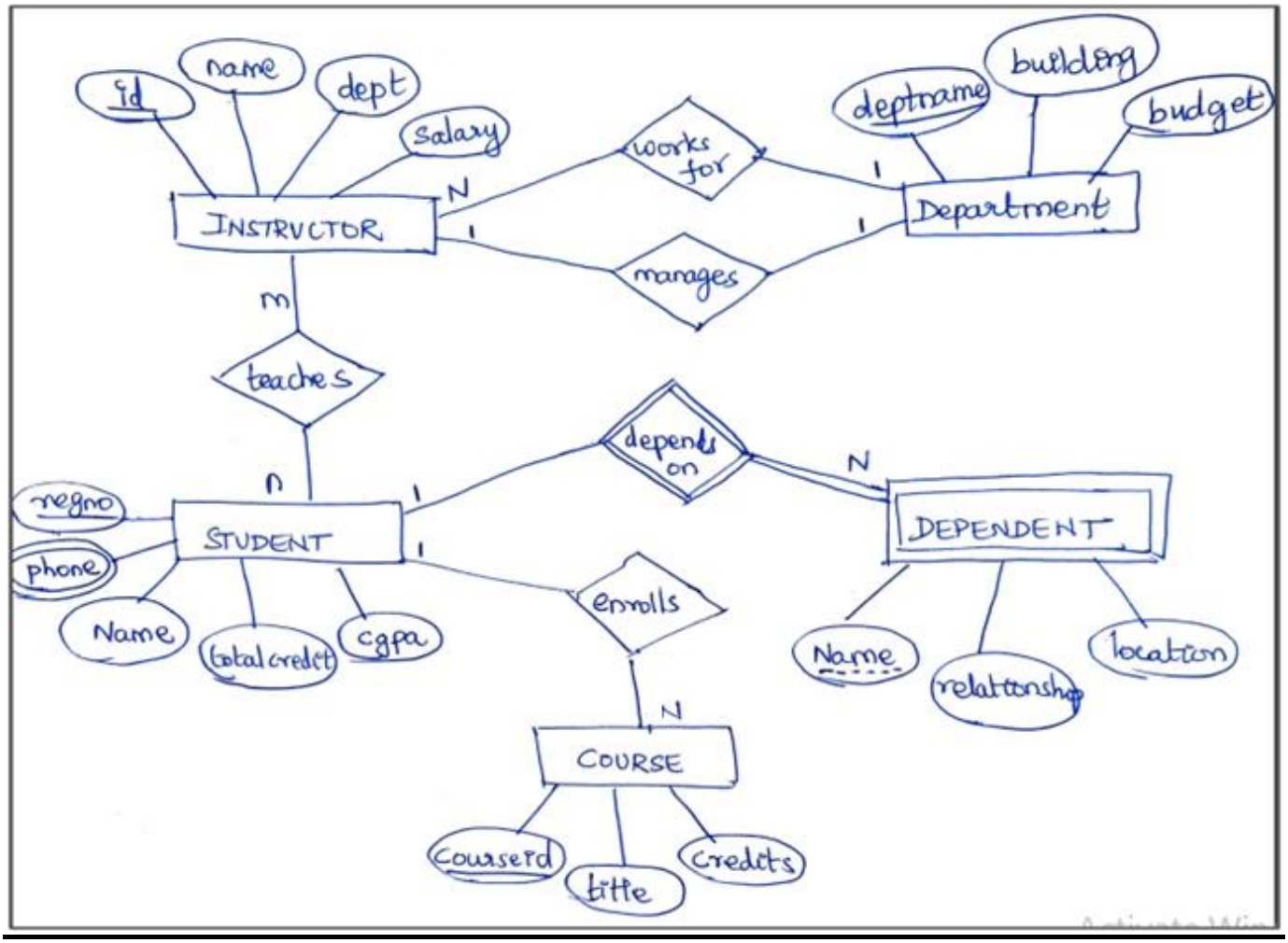


Fig: ER diagram for University Database Application

CONVERTING ER DIAGRAM FOR UNIVERSITY DATABASE INTO RELATIONAL SCHEMAS

Mapping of Regular Entity Types

- For each strong/regular entity type, create a corresponding relation that includes all simple attributes.
- If it is a composite attribute, all simple attributes of the composite attribute are included in the relation.
- Choose a primary key attribute for the relation.

INSTRUCTOR

id	Name	Dept	salary

DEPARTMENT

Deptname	Building	budget

STUDENT

Regno	Name	totalcredit	cgpa

COURSE

Courseid	Title	credits

Mapping of Weak Entity Types

- For each weak entity type, create a corresponding relation that includes all simple attributes.
- Add the primary key of owner entity type as a foreign key in the relation created for the weak entity type.
- The primary key for this corresponding relation is the combination of the primary key of the owner entity type and the partial key of the weak entity type.

INSTRUCTOR

id	Name	Dept	salary

DEPARTMENT

Deptname	Building	budget

STUDENT

Regno	Name	totalcredit	cgpa

COURSE

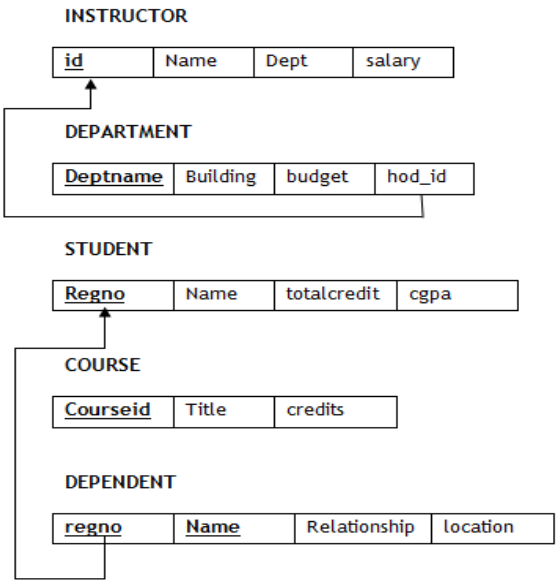
Courseid	Title	credits

DEPENDENT

regno	Name	Relationship	location

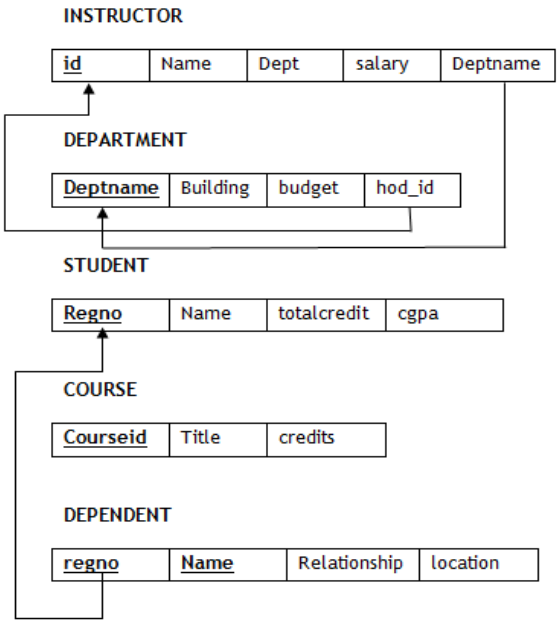
Mapping binary 1:1 Relationship types

- Choose one relation (table) as S and other as T (Better if S has total participation)
- Add to S all the simple attributes of the relationship
- Add the primary key attribute of T as a foreign key in S.



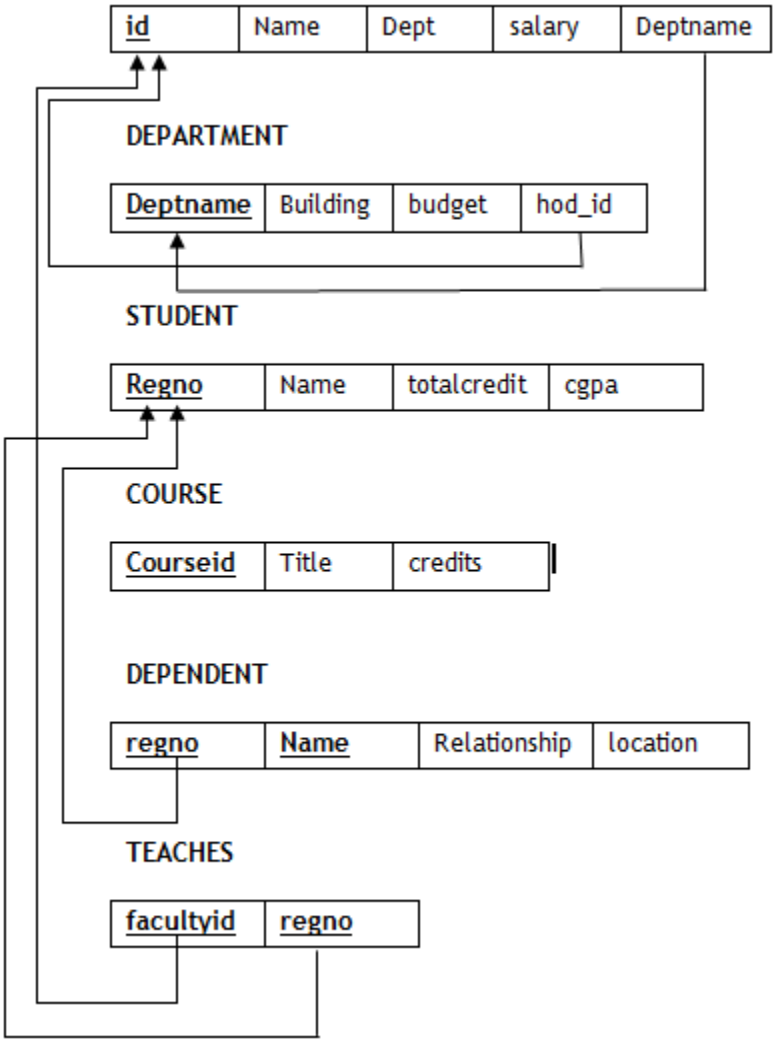
Mapping binary 1:n and n:1 Relationship types

- Choose the relation (table) on the n-side of the relationship as S, Other as T.
- Add to S all the simple attributes of the relationship
- Add the primary key attribute of T as a foreign key in S.



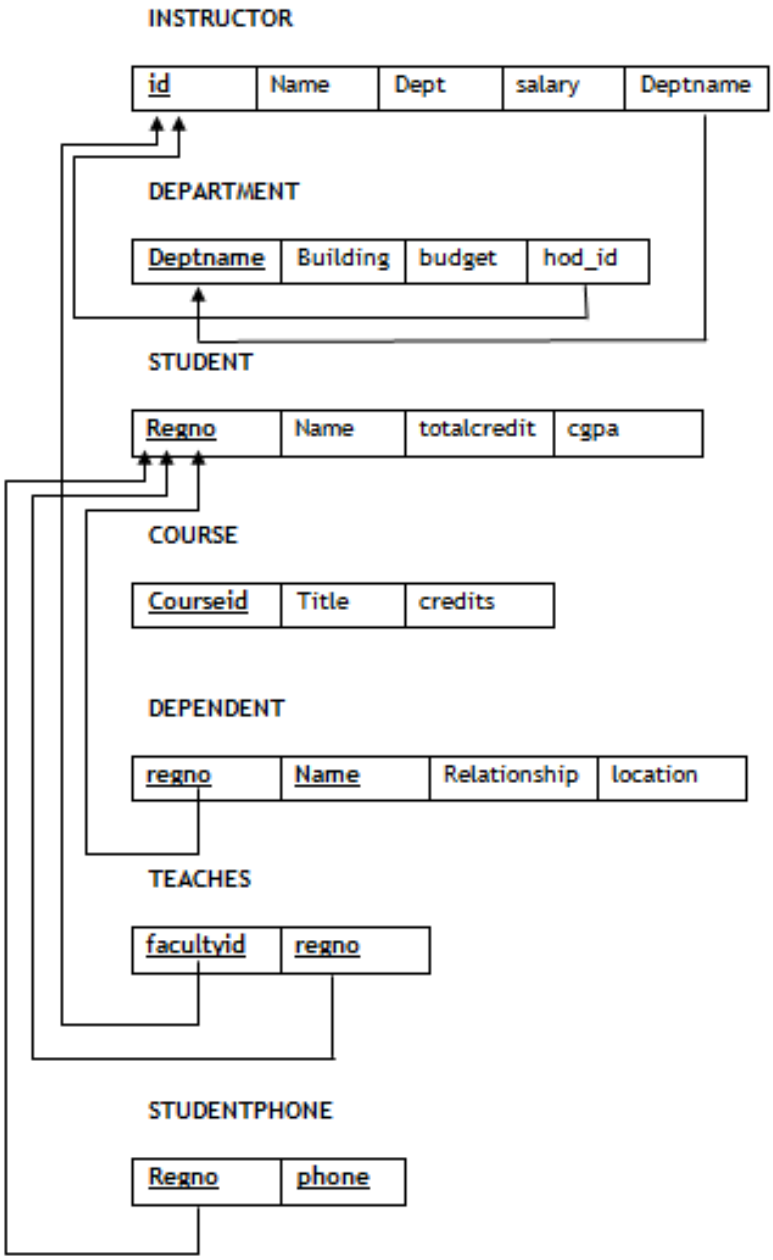
Mapping binary m:n Relationship types

- Create a new relation (table) S. Table S is termed as relationship relation.
- Add to S all simple attributes of the relationship.
- Add the primary keys of both the relations as the foreign keys in S. Their combination forms the primary key of S.



Mapping multivalued attributes

- Create a new relation S.
- The primary key of the corresponding relation is added as a foreign key.
- Add the multivalued attribute to S.
- The combination of all attributes in S forms the primary key.



The above is the equivalent relational schema for the ER diagram drawn for the university database