DATABASE MANAGEMENT SYSTEMS

Couse code:CSC403

Prof. Juhi Janjua

Module 2: Entity–Relationship Data Model

- The Entity-Relationship (ER) Model: Entity types: weak and strong entity sets, Entity sets, Types of attributes, Keys
- Relationship constraints: Cardinality and Participation
- Extended Entity-Relationship (EER) Model: Generalization, Specialization and Aggregation

Entity-Relationship (ER) Model

- It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

This model is based on three basic concepts:

1. Entities
2. Attributes
3. Relationships

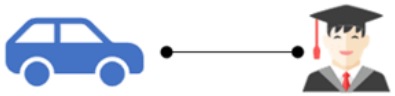
For example, in a University database, we might have entities for Students, Courses, and Lecturers. Student entity can have attributes like Rollno, Name, and DeptID. They might have relationships with Courses and Lecturers.

Entity Name

Entity

Person,place,object,event or concept about which data is to be maintained

Example: Car, Student



Relation

Verb Phrase

Association between the instances of one or more entity types

Example: Blue Car Belongs to Student Jack

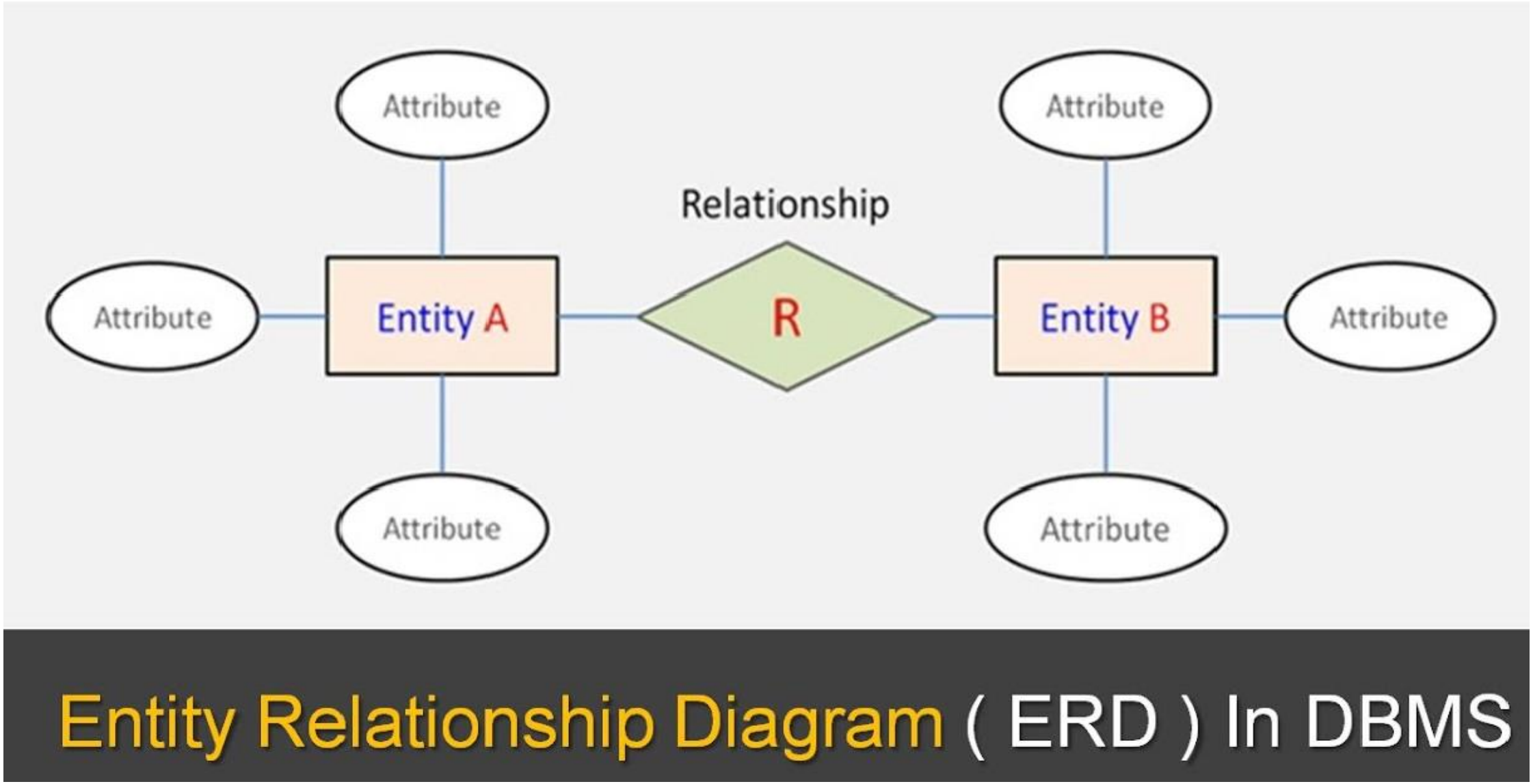
Jack

Attribute Name

Attribute

Property or characteristic of an entity

Example: Color of car Entity Name of Student Entity

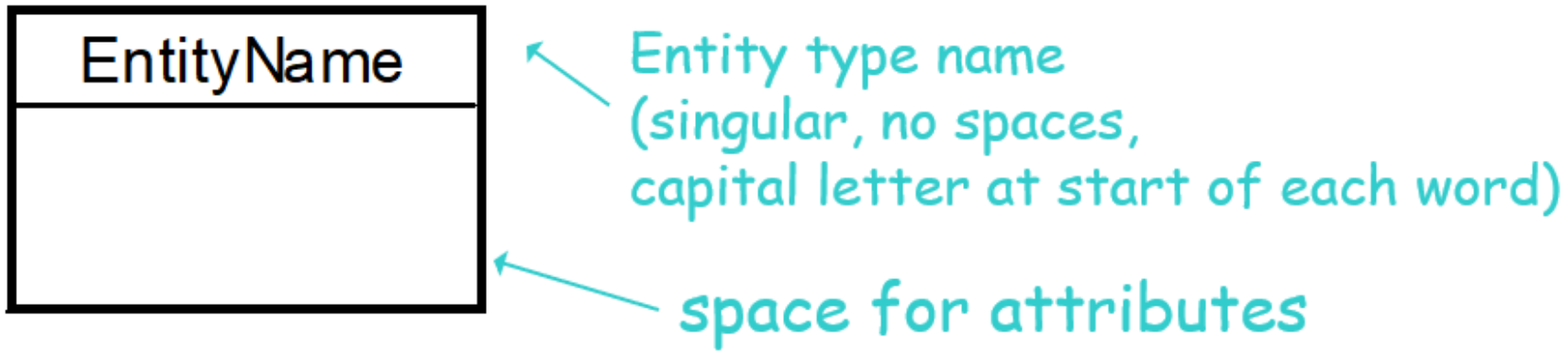


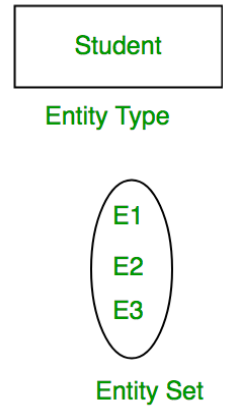
Entity Relationship Diagram (ERD) In DBMS

Entities

- It may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.
- Example:

Consider an organization - manager, product, employee, department etc. can be taken as an entity.

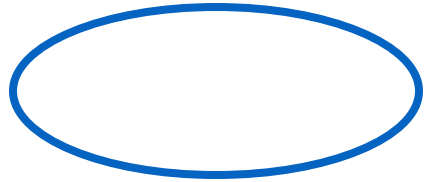




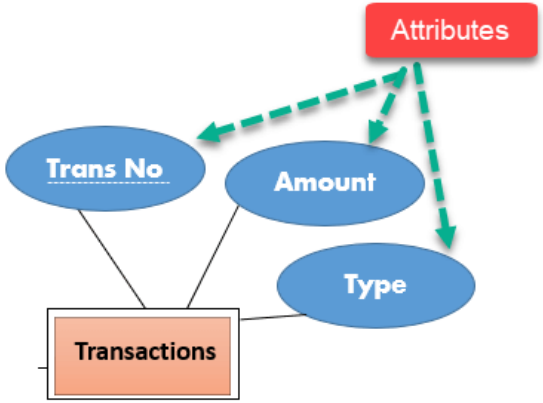
- An **entity set** is a set of entities of the same type (e.g., all persons having an account at a bank).

Attributes

- Each entity has a set of associated properties that describes the entity. These properties are known as ***attributes***.
- They are represented by an oval.

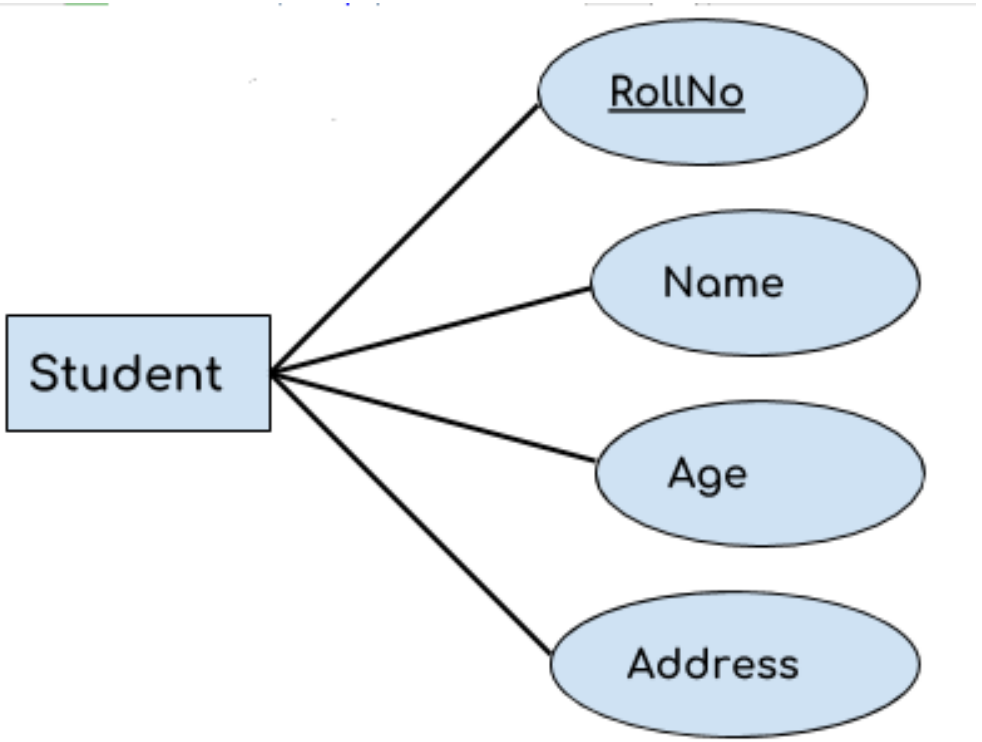


- Attributes can be:
  1. Key attribute
  2. Composite attribute
  3. Multivalued attribute
  4. Derived attribute



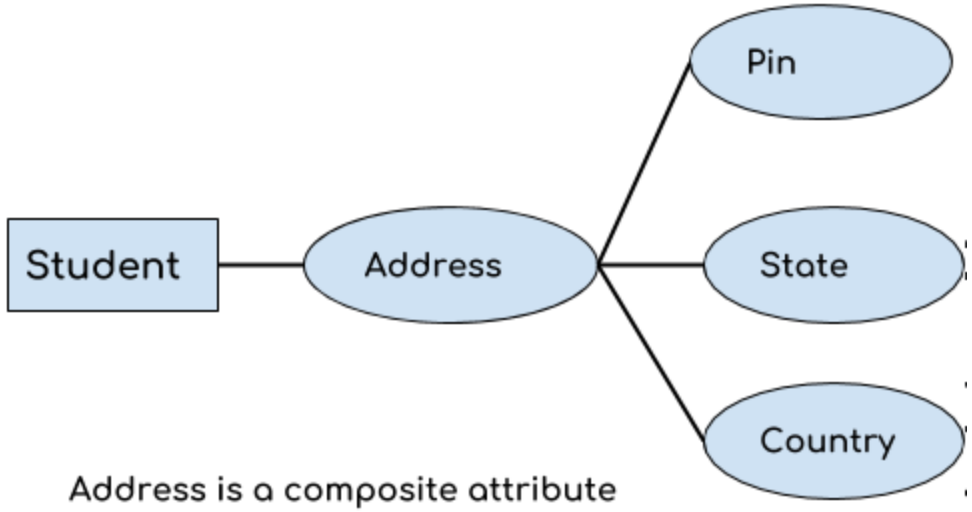
Key attribute

- It can uniquely identify an entity from an entity set.
- **Text of key attribute is underlined**.
- For example, student roll number can uniquely identify a student from a set of students



Composite attribute

- It is a combination of other attributes
- For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.



Address is a composite attribute

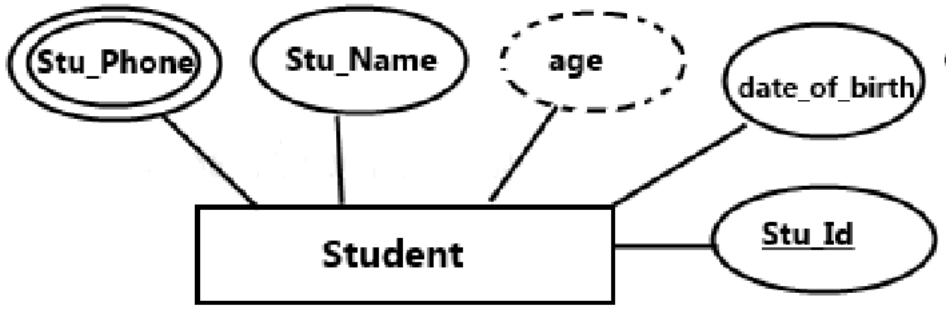
Multivalued attribute

- It can hold multiple values is known as multivalued attribute.
- It is represented with **double ovals** in an ER Diagram.
- For example – A person can have more than one phone numbers so the phone number attribute is multivalued.

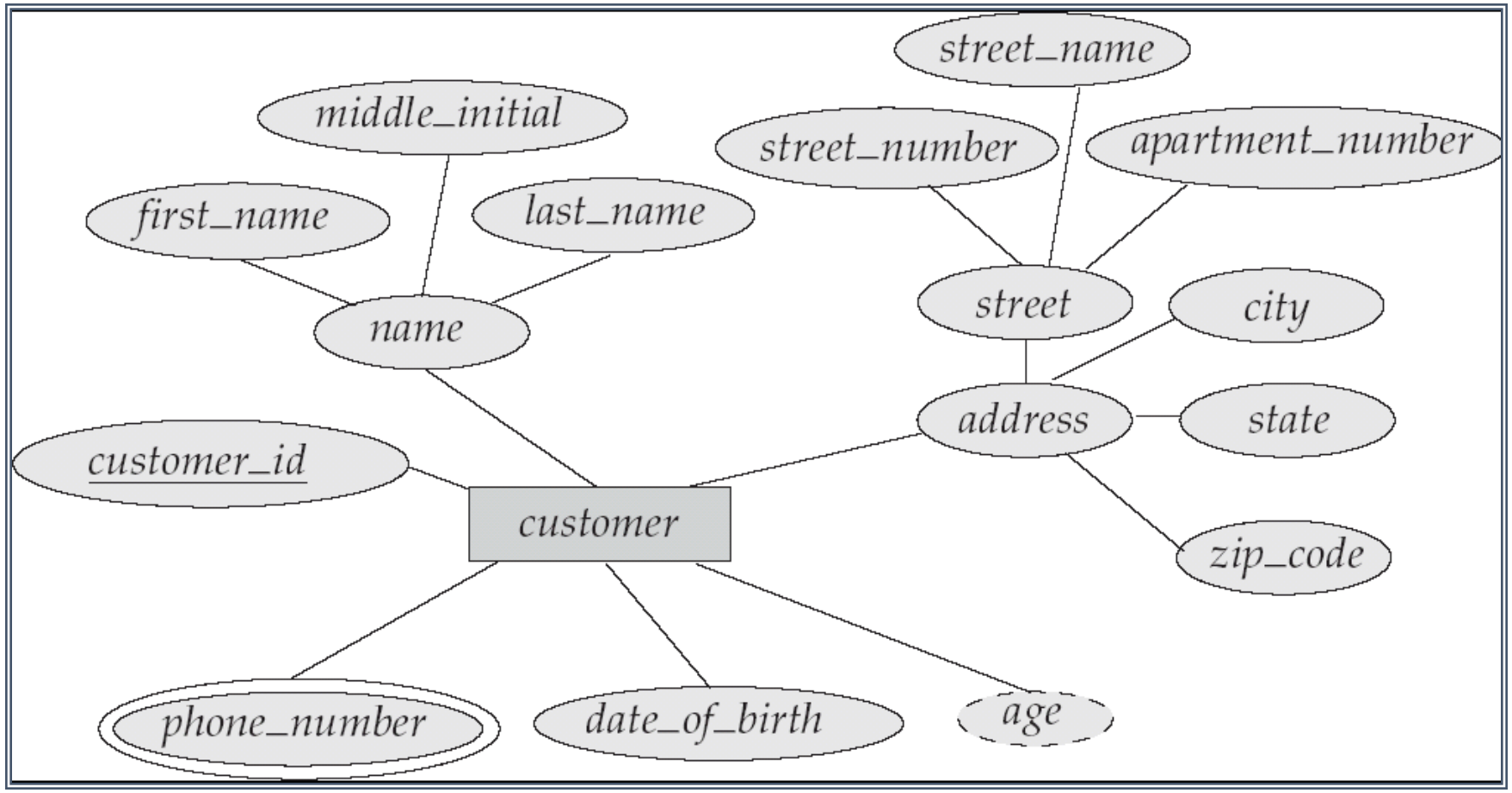
Derived attribute:

- It is one whose value is dynamic and derived from another attribute.
- It is represented by **dashed oval** in an ER Diagram.
- For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).

Multivalued & Derived attribute



Customer entity with Composite, Multivalued, and Derived Attributes



Keys

- Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.
- There are different types of keys:

1. Super key
2. Candidate key
3. Primary key
4. Foreign key

Let's take an example to understand this:

Table: Employee

Emp_SSN	Emp_Number	Emp_Name
123456789	226	Steve
999999321	227	Ajeet
888997212	228	Chaitanya
777778888	229	Robert

Keys..

Super key:

- Set of one or more attributes (columns), which can uniquely identify a row in a table.
- All the following sets of super key can uniquely identify a row of the employee table.
- {Emp_SSN}
- {Emp_Number}
- {Emp_SSN, Emp_Number}
- {Emp_SSN, Emp_Name}
- {Emp_SSN, Emp_Number, Emp_Name}
- {Emp_Number, Emp_Name}

Candidate key:

- It is a minimal super key with no redundant attributes.
- The following two set of super keys are chosen as candidate key.
- {Emp_SSN}
- {Emp_Number}

Primary key:

- It is selected from a set of candidate keys.
- It uniquely identifies the entity.
- This is done by database admin or database designer.
- We can say that either {Emp_SSN} or {Emp_Number} can be chosen as a primary key for the table Employee.

Foreign key:

- Foreign keys are the columns of a table that points to the candidate key of another table.

Example:

Stu_Id column in Course_enrollment table is a foreign key as it points to the primary key of the Student table.

Course_enrollment table:

Course_Id	Stu_Id
C01	101
C02	102
C03	101
C05	102

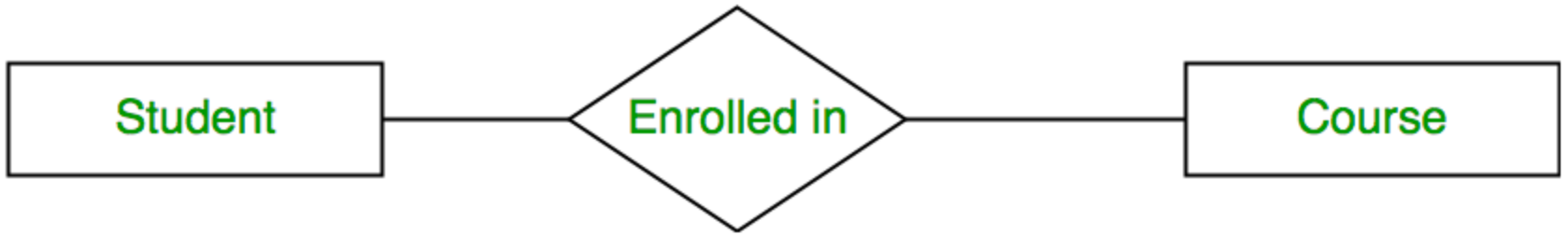
Student table:

Stu_Id	Stu_Name	Stu_Age
101	Chaitanya	22
102	Arya	26
103	Bran	25
104	Jon	21

Relationship Type & Relationship set

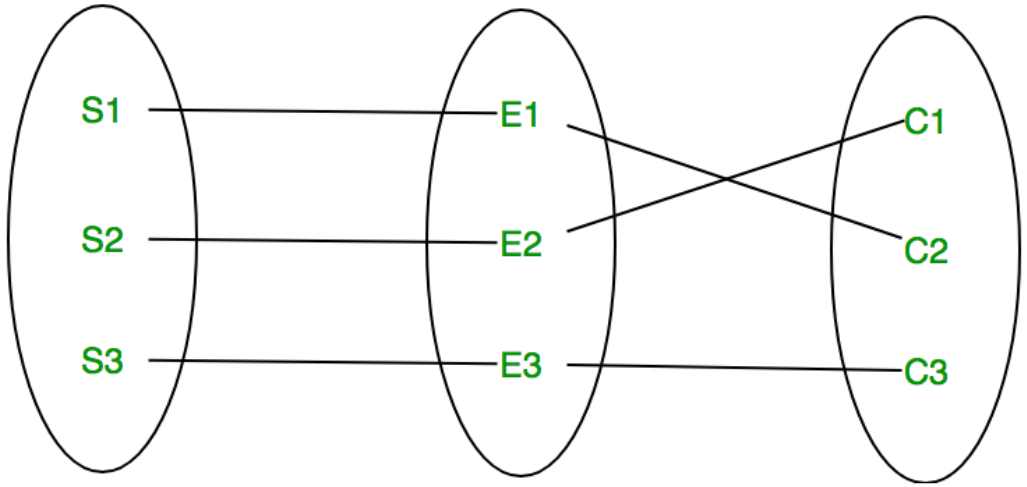
Relationship type:

- It represents the association between entity types.
- For example, ‘Enrolled in’ is a relationship type that exists between entity type Student and Course.
- In ER diagram, relationship type is represented by a **diamond** and connecting the entities with lines.



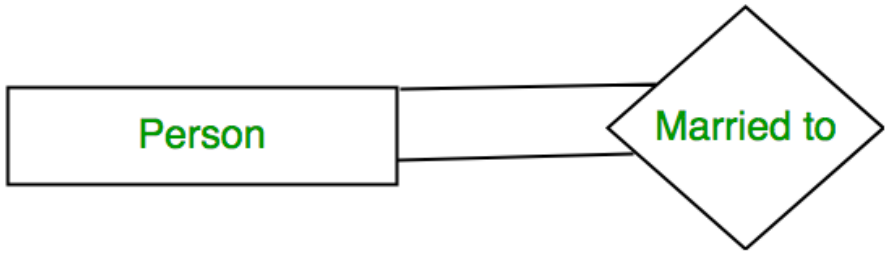
Relationship set:

A set of relationships of same type is known as relationship set. The following relationship set depicts S1 is enrolled in C2, S2 is enrolled in C1 and S3 is enrolled in C3.

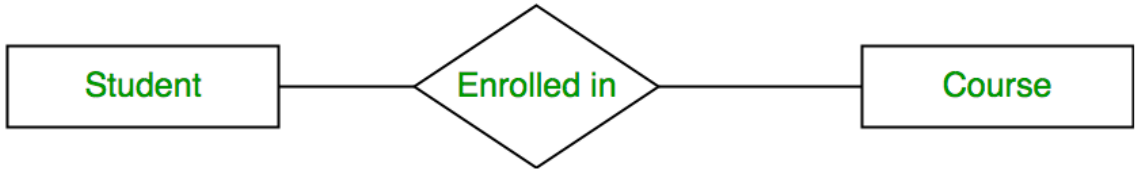


Degree of relationship set

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



Unary relationship - Only ONE entity set participating in a relation



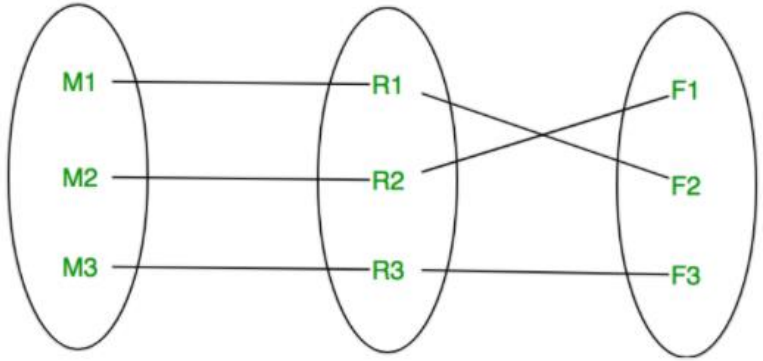
Binary relationship - TWO entities set participating in a relation

Cardinality

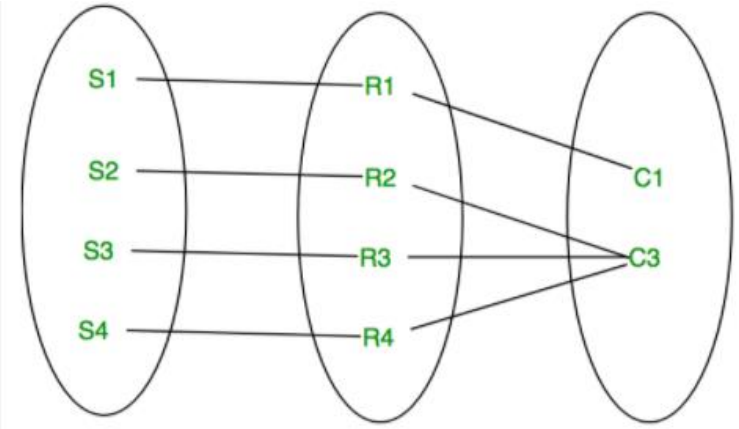
- The number of times an entity of an entity set participates in a relationship set is known as cardinality.
- Cardinality can be of different types:

1. One to one
2. Many to many
3. One to many
4. Many to one

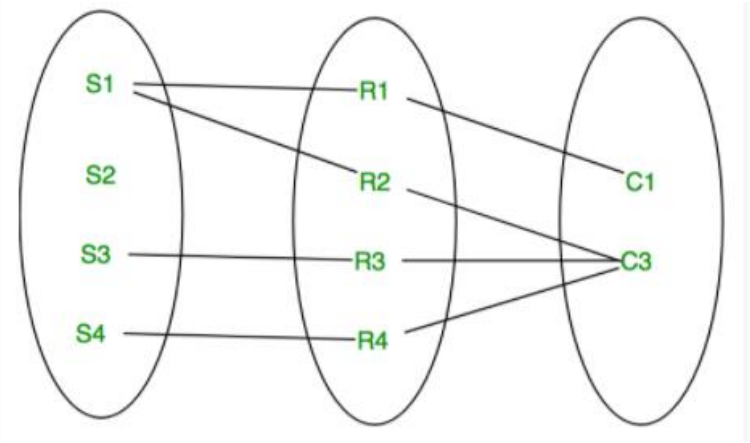
One to one:



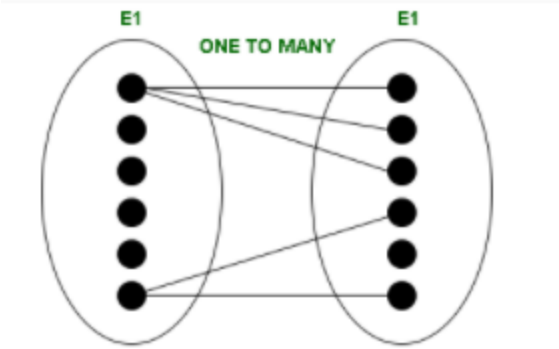
Many to one:



Many to many:



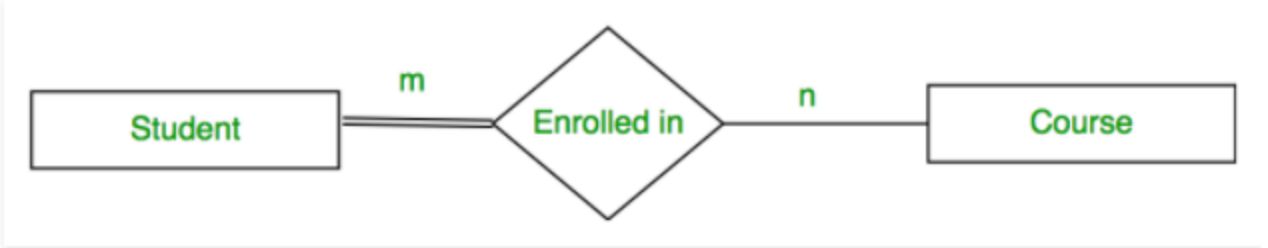
One to many:



Participation constraint

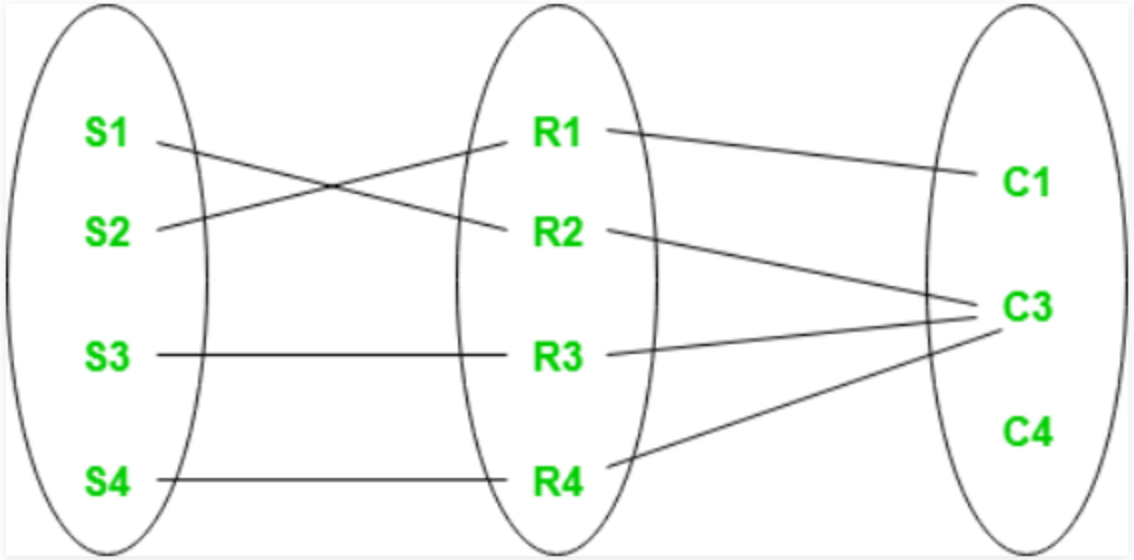
- It is applied on the entity participating in the relationship set.

1. **Total Participation –** Each entity in the entity set **must participate** in the relationship. If each student must enroll in a course, the participation of student will be total. Total participation is shown by **double line** in ER diagram.



2. **Partial Participation –** The entity in the entity set **may or may NOT participate** in the relationship. If some courses are not enrolled by any of the student, the participation of course will be partial.

The diagram depicts the ‘Enrolled in’ relationship set with Student Entity set having total participation and Course Entity set having partial participation



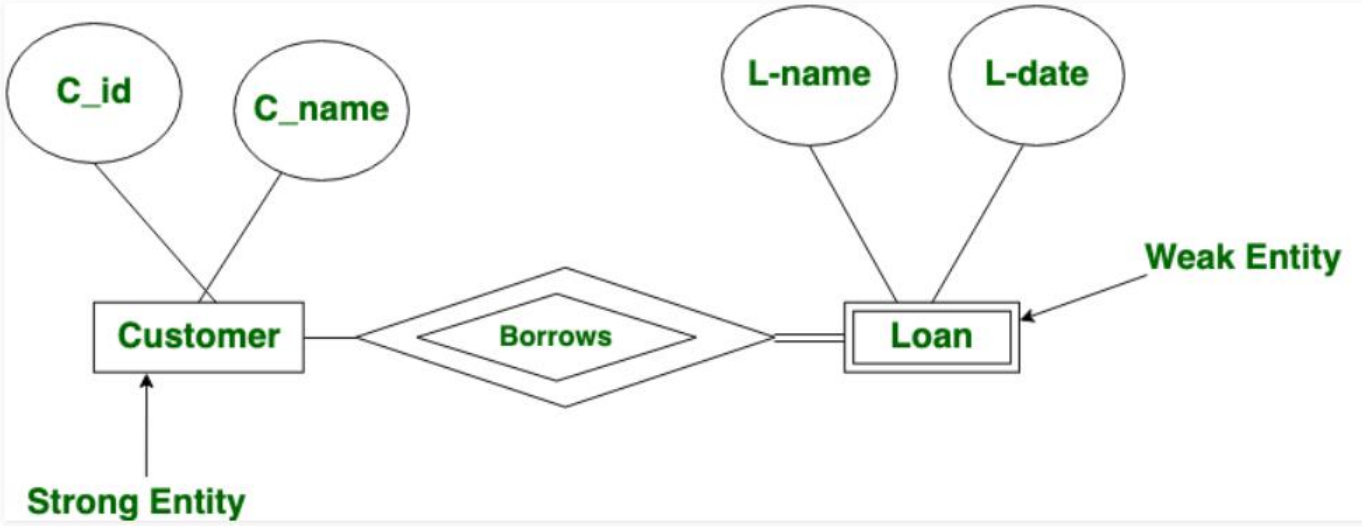
Weak & Strong entities

Strong entity:

- It is not dependent of any other entity in the schema.
- A strong entity will always have a primary key
- They are represented by a single rectangle.
- The relationship of two strong entities is represented by a single diamond.
- Various strong entities, when combined, create a strong entity set.

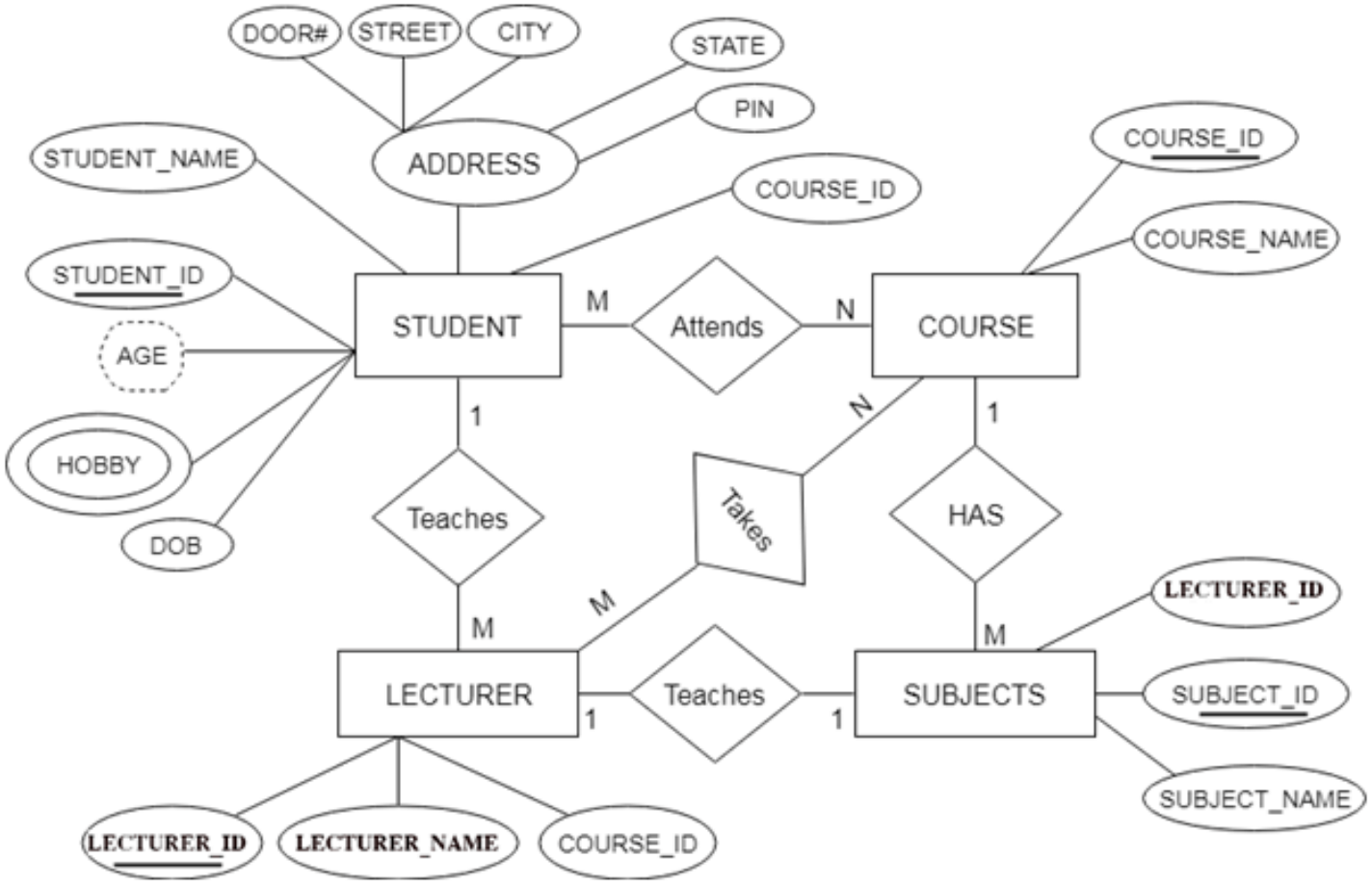
Weak entity:

- It is dependent on a strong entity to ensure its existence.
- Unlike a strong entity, a weak entity does not have any primary key.
- It instead has a partial discriminator key. A weak entity is represented by a **double rectangle**.
- The relation between one strong and one weak entity is represented by a **double diamond**.



S.NO	Strong Entity	Weak Entity
1.	Strong entity always has primary key.	While weak entity has partial discriminator key.
2.	Strong entity is not dependent of any other entity.	Weak entity is depend on strong entity.
3.	Strong entity is represented by single rectangle.	Weak entity is represented by double rectangle.
4.	Two strong entity's relationship is represented by single diamond.	While the relation between one strong and one weak entity is represented by double diamond.
5.	Strong entity have either total participation or not.	While weak entity always has total participation.

College ER diagram

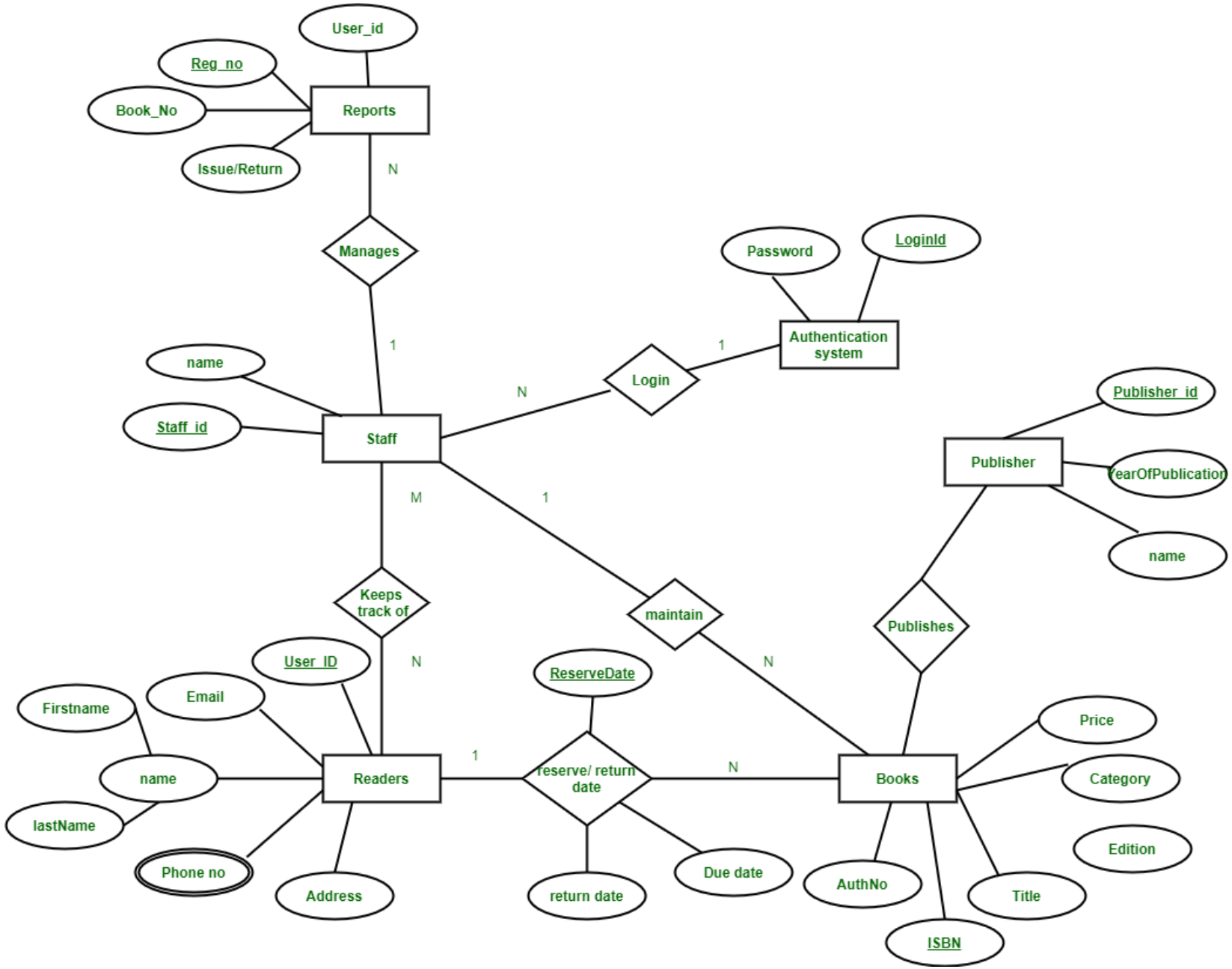


ER Diagram of Library Management System

Library Management System

The Library Management System database keeps track of readers with the following considerations –

- The system keeps track of the staff with a single point authentication system comprising login Id and password.
- Staff maintains the book catalog with its ISBN, Book title, price(in INR), category(novel, general, story), edition, author Number and details.
- A publisher has publisher Id, Year when the book was published, and name of the book.
- Readers are registered with their user_id, email, name (first name, last name), Phone no (multiple entries allowed), communication address. The staff keeps track of readers.
- Readers can return/reserve books that stamps with issue date and return date. If not returned within the prescribed time period, it may have a due date too.
- Staff also generate reports that has readers id, registration no of report, book no and return/issue info.


User_id
Reg_no
Book_No
Reports
Issue/Return
N
Manages
1
Password
LoginId
Authentication system
1
Login
N
name
Staff_id
Staff
M
1
Publisher_id
Publisher
YearOfPublication
name
Keeps track of
maintain
Publishes
N
N
User_ID
Email
Firstname
name
lastName
Phone no
Address
Readers
1
ReserveDate
reserve/ return date
N
return date
Due date
Books
Price
Category
Edition
Title
ISBN
AuthNo

Extended Entity-Relationship (EER) Model

- As the complexity of data increases, it became more and more difficult to use the traditional ER Model for database modelling.
- Hence some improvements were made to the existing ER Model to make it able to handle the complex applications better.
- Hence, as part of the **Extended ER Model**, along with other improvements, three new concepts were added to the existing ER Model, they were:

1. Specialization
2. Generalization
3. Aggregation

- Let's understand what they are, and why were they added to the existing ER Model

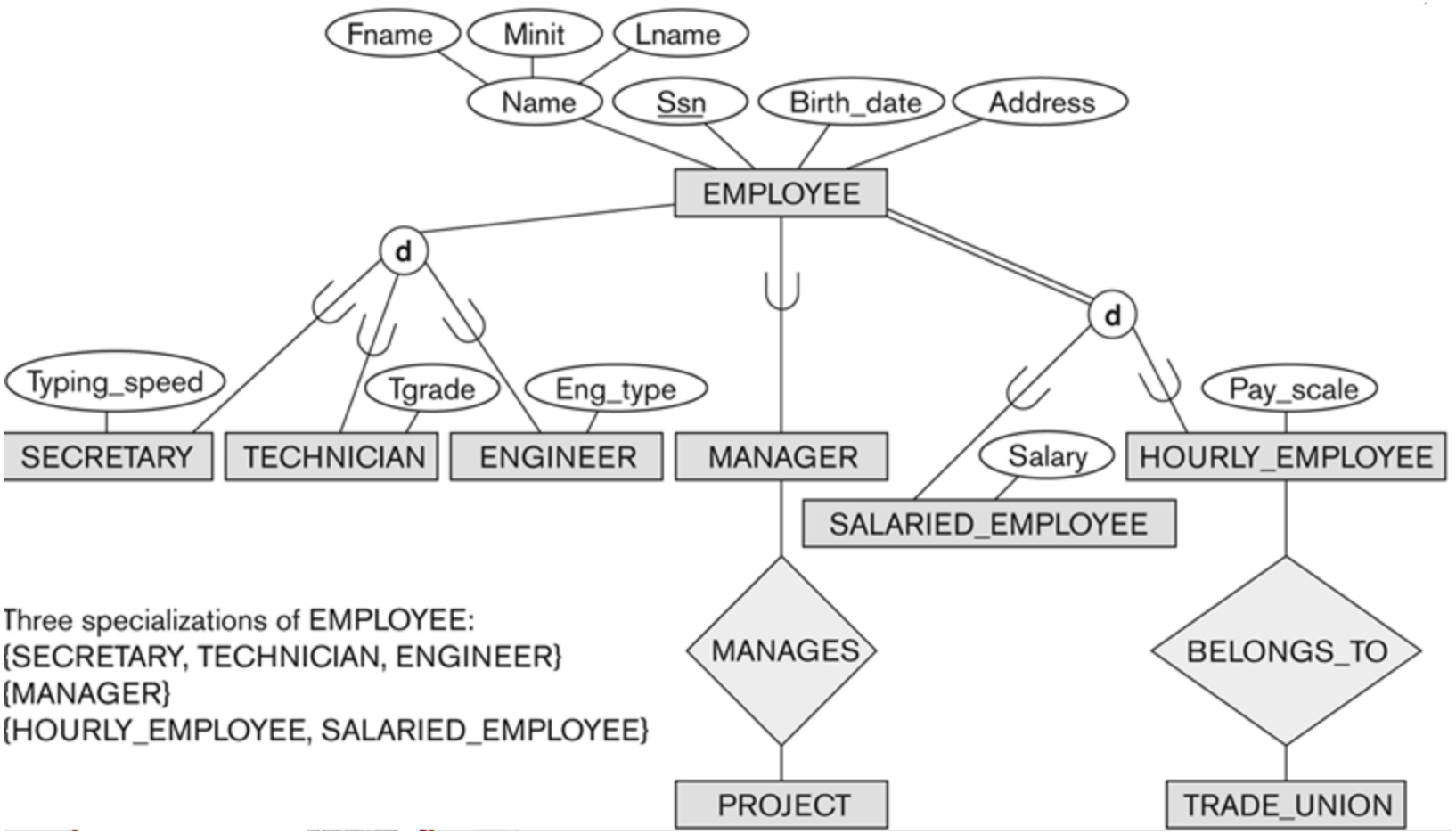
Specialization

- Specialization is the process of defining a set of subclasses of a superclass
- The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass
  - Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon *job type.*

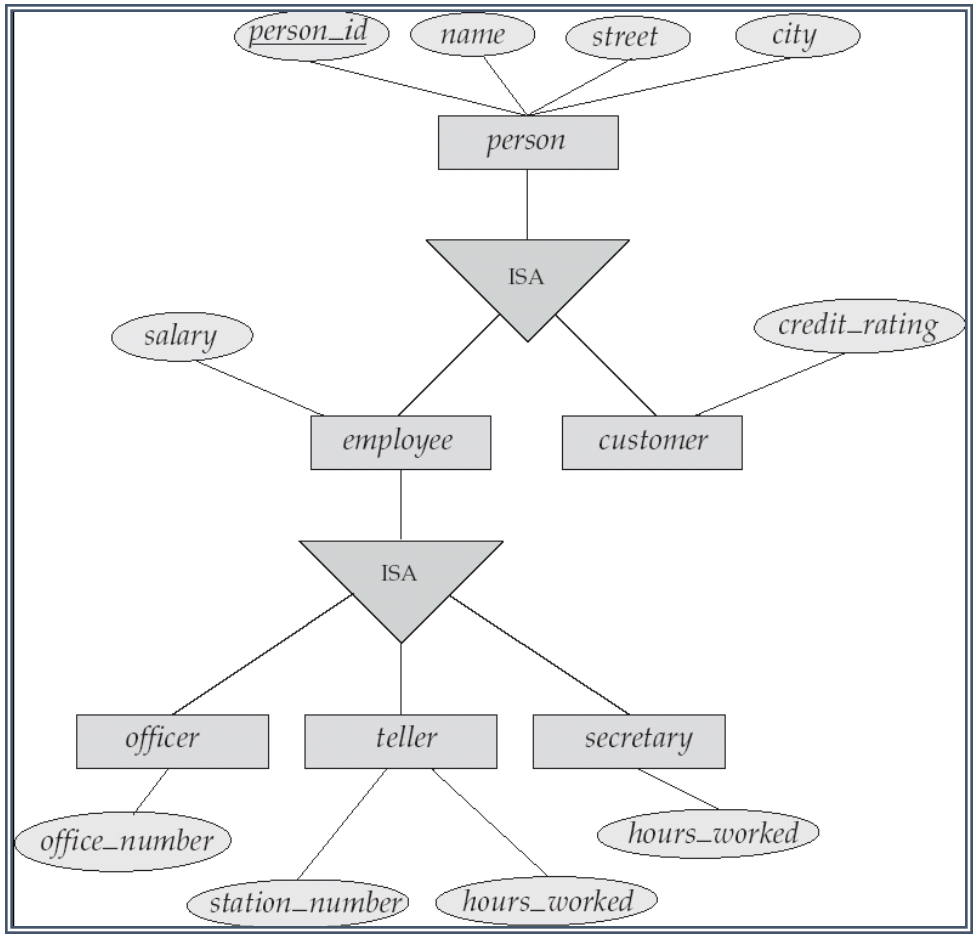
Specialization…

- Example: Another specialization of EMPLOYEE based on *method of pay* is {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}.
  - Superclass/subclass relationships and specialization can be diagrammatically represented in EER diagrams
  - Attributes of a subclass are called ***specific* or *local* attributes**.
    - For example, the attribute TypingSpeed of SECRETARY
  - The subclass can also participate in specific relationship types.
    - For example, a relationship BELONGS_TO of HOURLY_EMPLOYEE

Specialization…

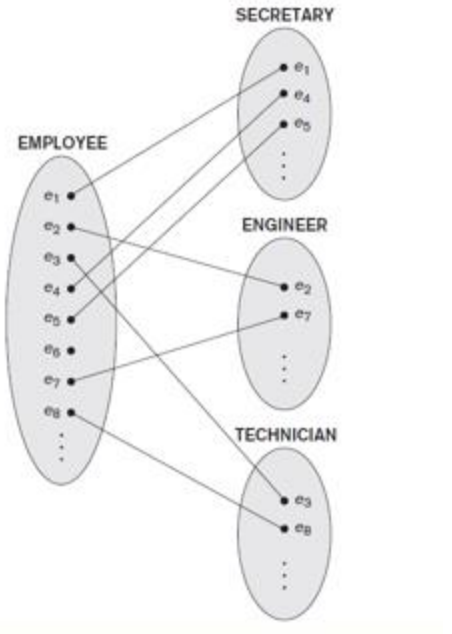


Specialization…



Specialization…

- **Specialization**
  - Process of defining a set of subclasses of an entity type
  - Defined based on some distinguishing characteristic of the entities in the superclass
- Subclass can define:
  - **Specific attributes**
  - **Specific relationship types**

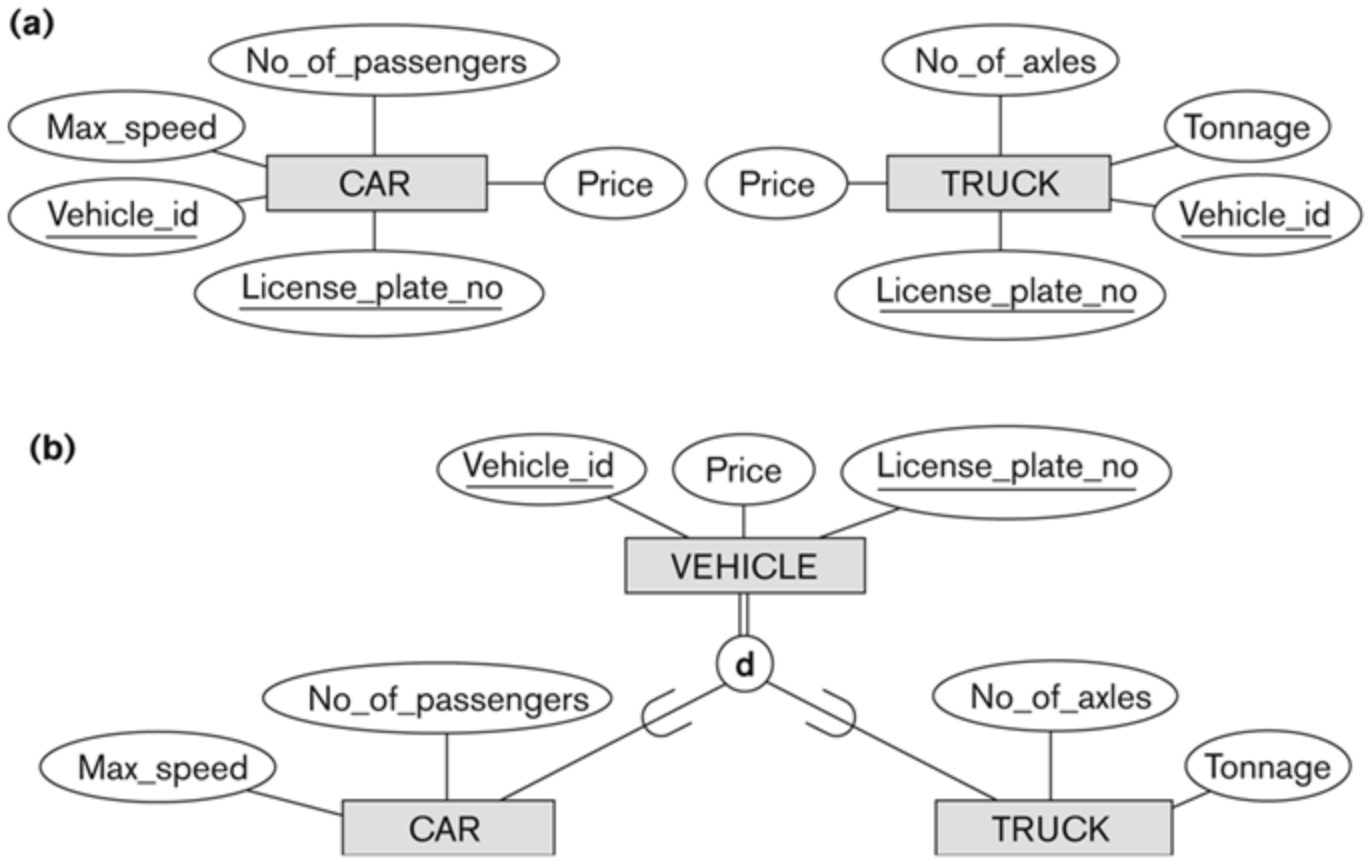


Instances of specialization

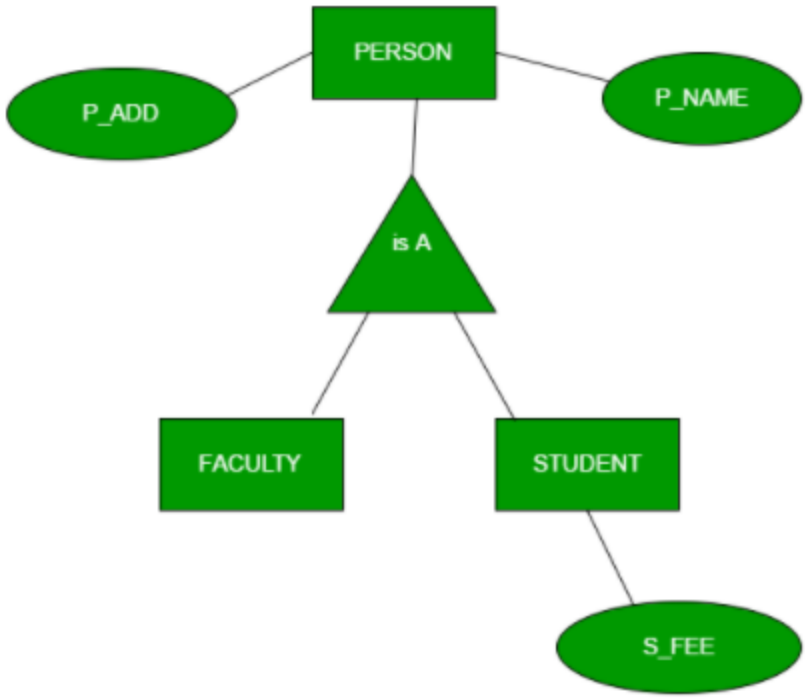
Generalization

- Generalization is the reverse of the specialization process
- Several classes with common features are generalized into a superclass;
  - original classes become its subclasses
- Example: CAR, TRUCK generalized into VEHICLE;
  - both CAR, TRUCK become subclasses of the superclass VEHICLE.
  - We can view {CAR, TRUCK} as a specialization of VEHICLE
  - Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK

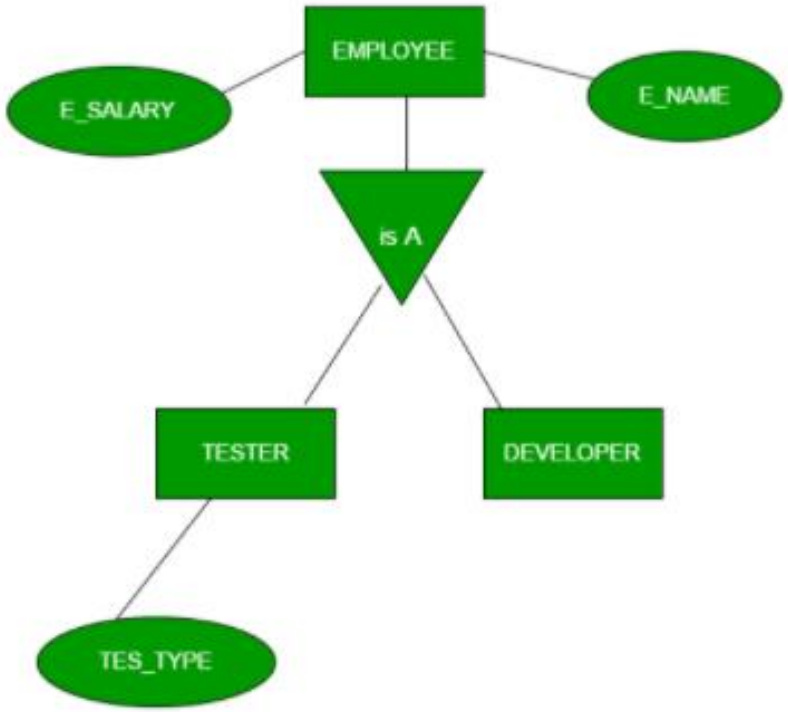
Generalization…



Generalization



Specialization



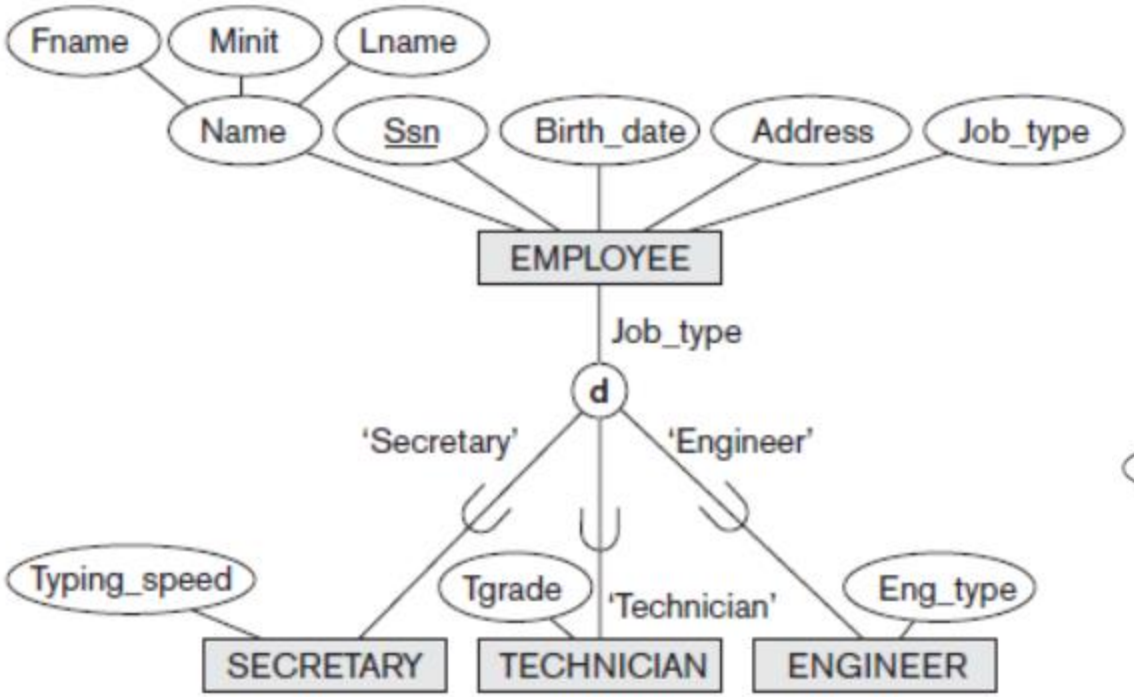
Design Constraints on a Specialization/Generalization

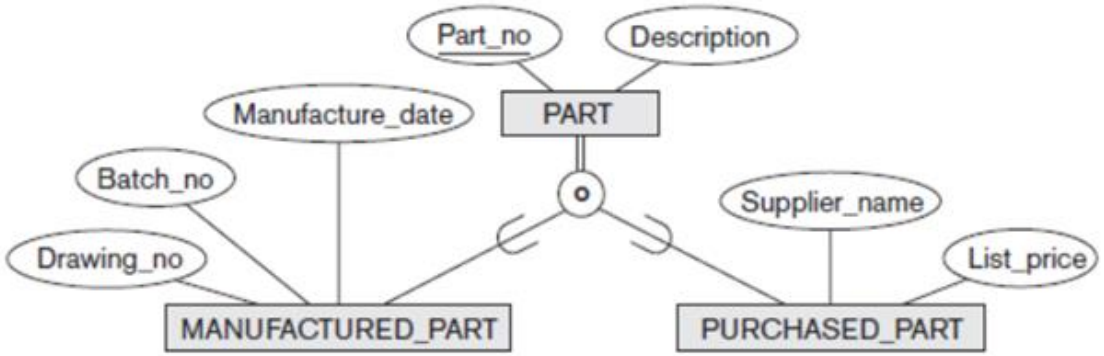
- Constraint on which entities can be members of a given lower-level entity set.
  - condition-defined
    - Example: all customers over 65 years are members of *senior-citizen* entity set; *senior-citizen* ISA *person*.
- Constraint on whether entities may belong to more than one lower-level entity set within a single generalization.
  - **Disjoint**
    - an entity can belong to only one lower-level entity set
    - Noted in E-R diagram by writing *disjoint* next to the ISA triangle
  - **Overlapping**
    - an entity can belong to more than one lower-level entity set

Design Constraints on a Specialization/Generalization…

- Completeness constraint -- specifies whether an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization.
  - **total** : an entity must belong to one of the lower-level entity sets
  - **partial**: an entity need not belong to one of the lower-level entity sets

Constraints example





Aggregation

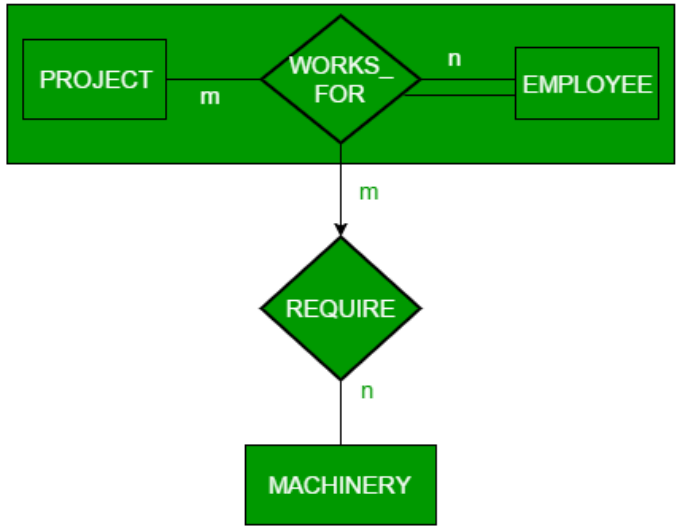
- An ER diagram is not capable of representing relationship between an **entity and a relationship** which may be required in some scenarios.
- In those cases, a relationship with its corresponding entities is aggregated into a higher-level entity.

Aggregation…

- Example:

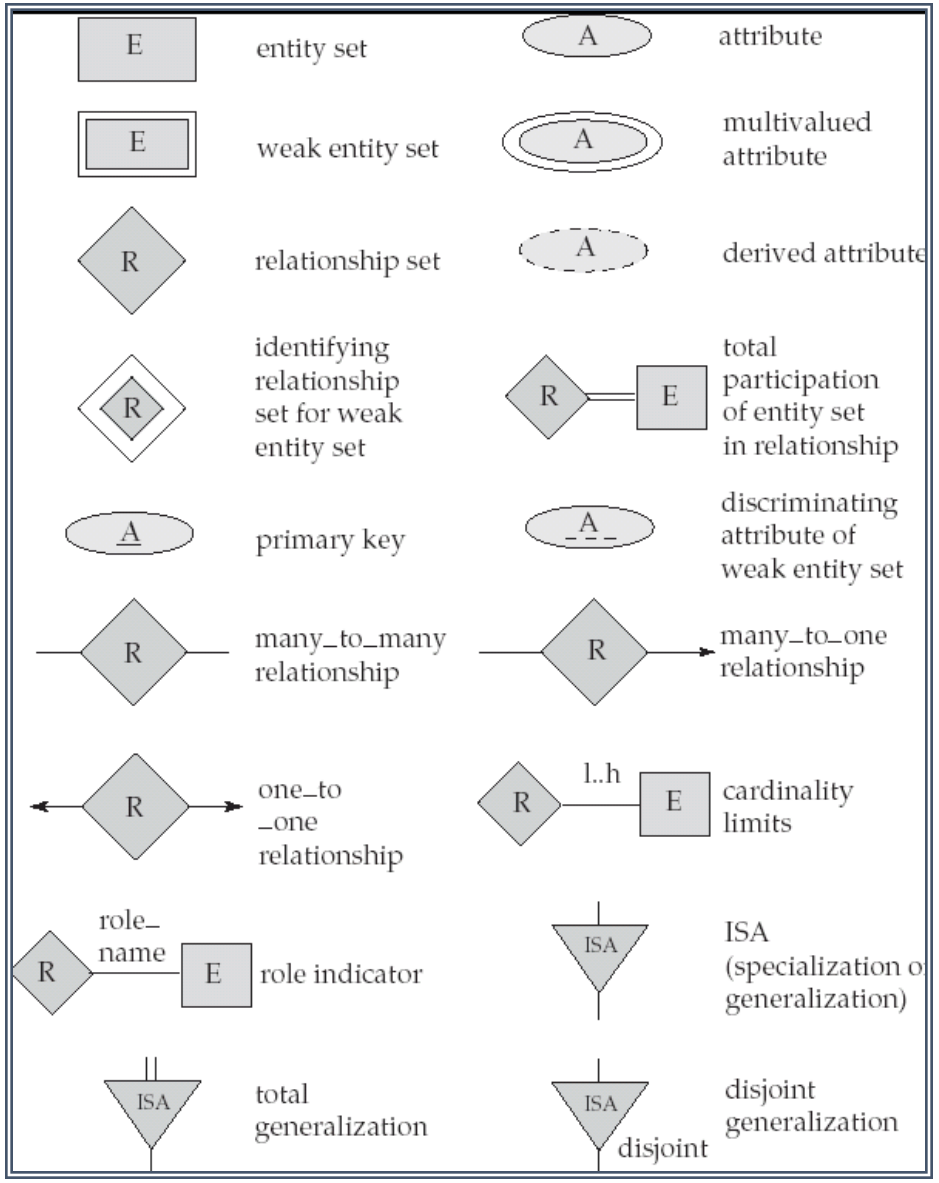
Employee working for a project may require some machinery. So, REQUIRE relationship is needed between relationship WORKS_FOR and entity MACHINERY.

- Using aggregation, WORKS_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into single entity and relationship REQUIRE is created between aggregated entity and MACHINERY.



Aggregation

Summary of Symbols Used in E-R Notation



ER Diagram for Banking Enterprise

E-R Diagram for a Banking Enterprise