UCS310
Database Management System

E-R Model

Lecture-13

Date: 10 Feb 2023

Dr. Sumit Sharma
Assistant Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

Recap

- Types of Attributes
- Relationship
  - 1:1
  - 1:M
  - M:1
  - N:M
- Entity
  - Strong vs Weak

Enhanced-ER (EER) Model Concepts

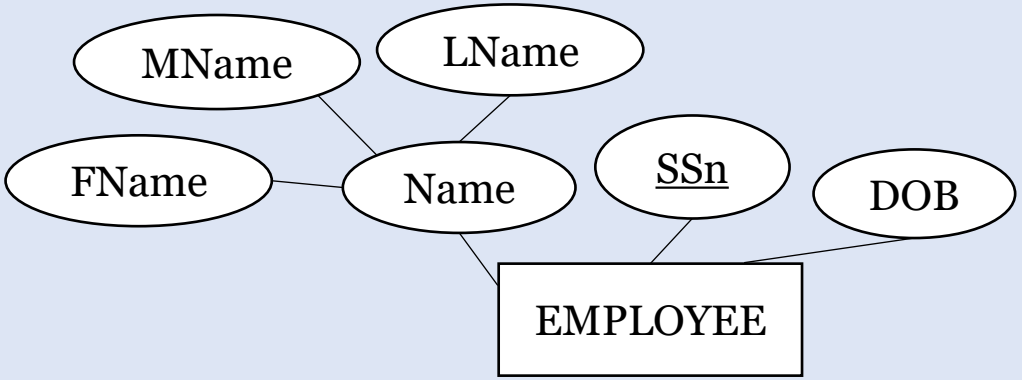
- Includes all modeling concepts of basic ER
- Additional concepts: subclasses/superclasses, specialization/generalization, categories, attribute inheritance
- The resulting model is called the enhanced-ER or Extended ER (E2R or EER) model
- It is used to model applications more completely and accurately if needed
- It includes some object-oriented concepts, such as inheritance

Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities

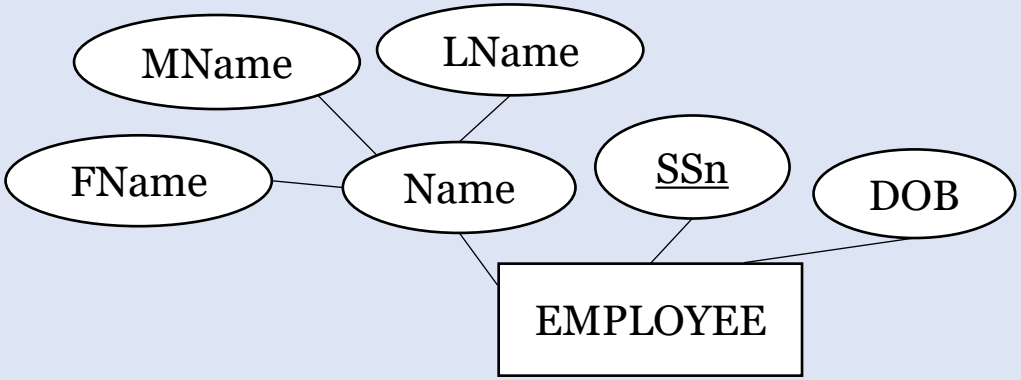
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



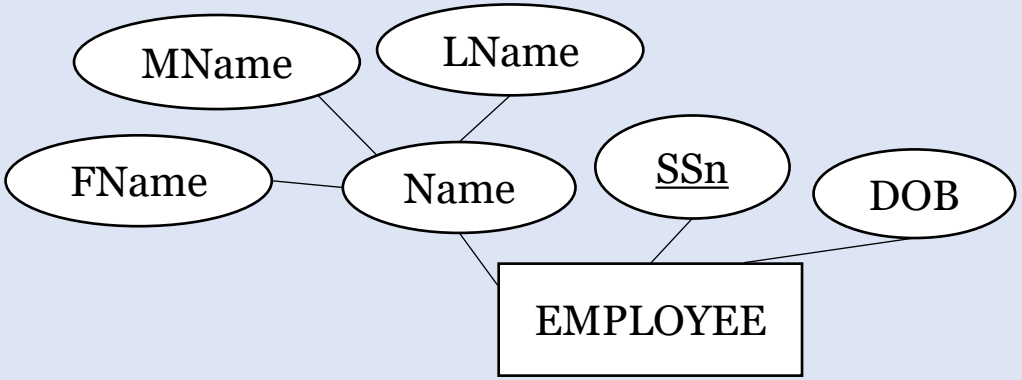
SECRETARY

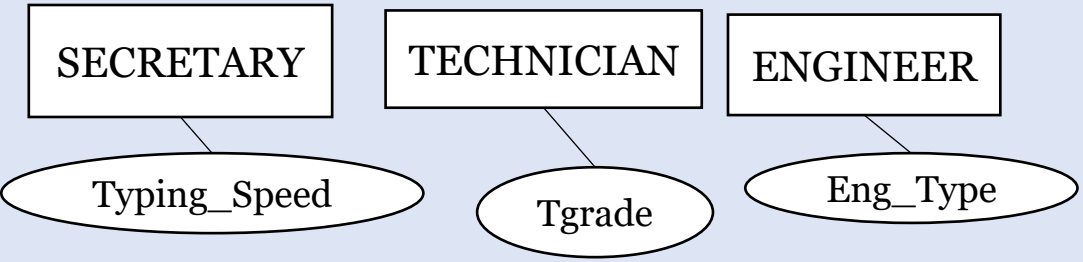
TECHNICIAN

ENGINEER

Subclasses and Superclasses

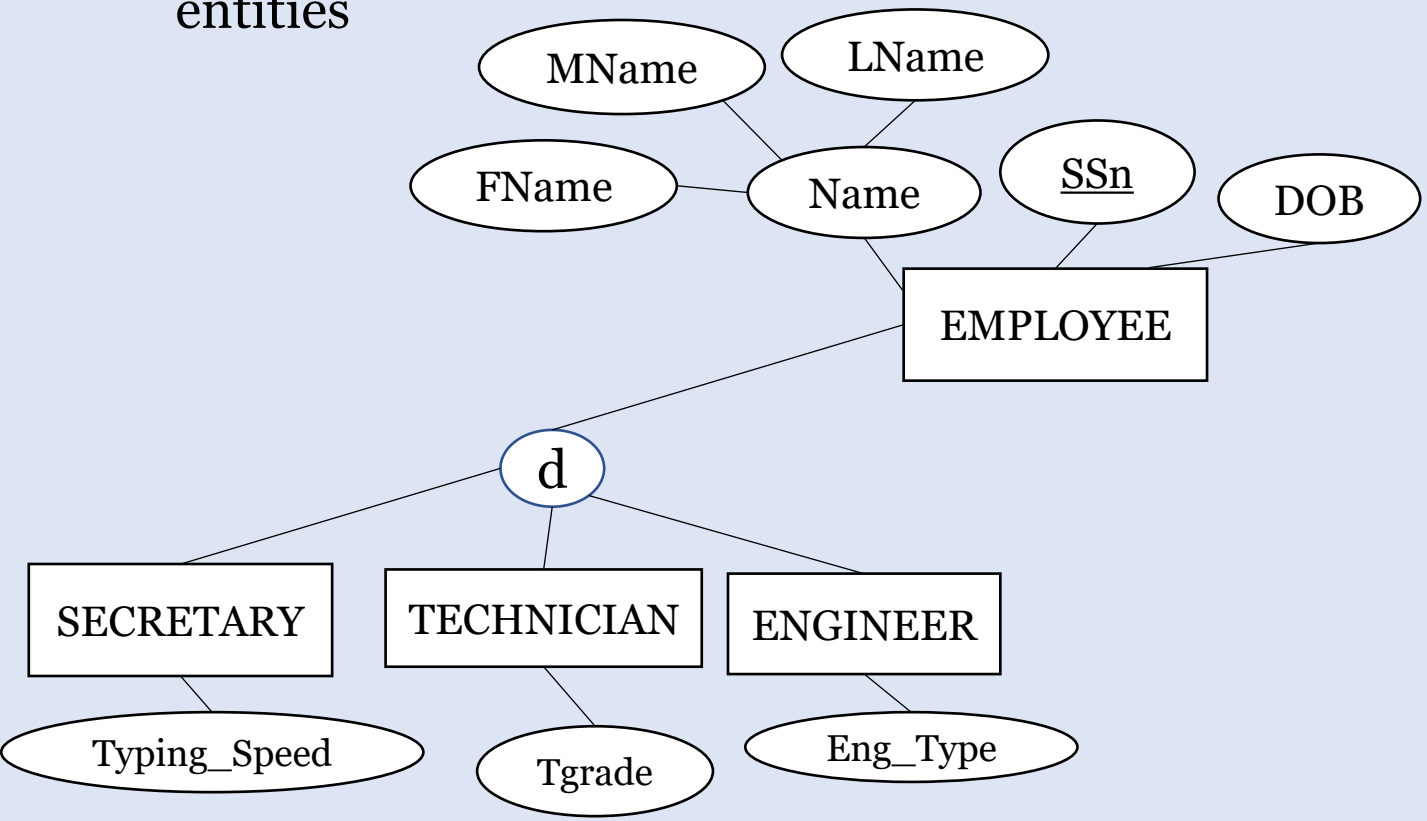
- An entity type may have additional meaningful subgroupings of its entities





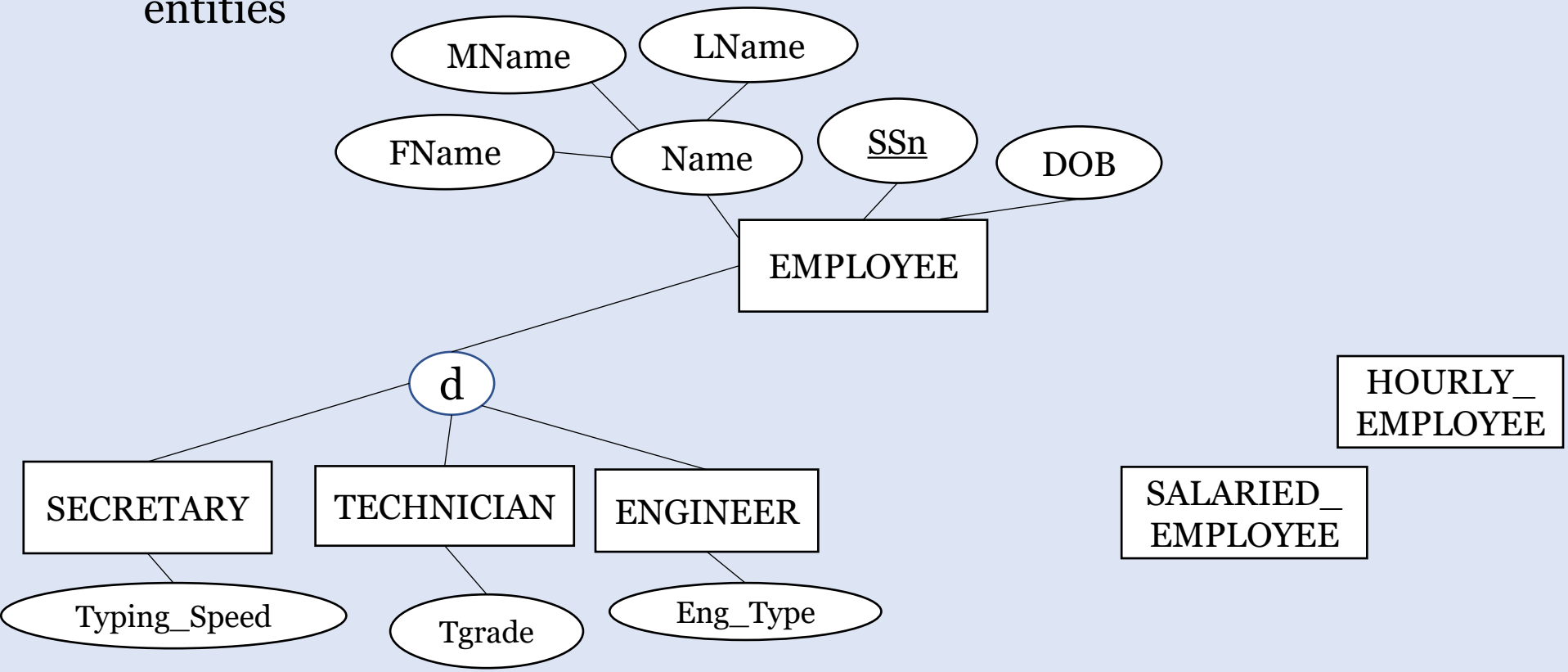
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



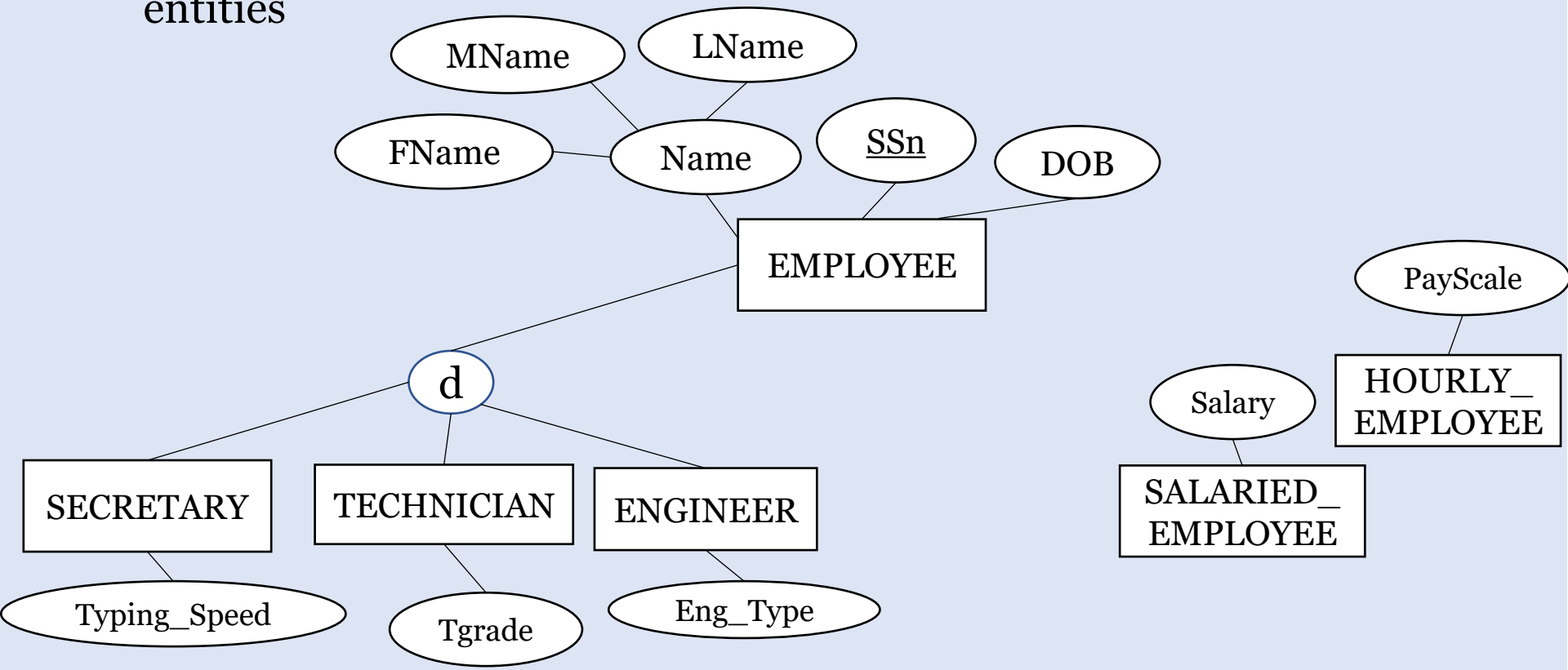
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



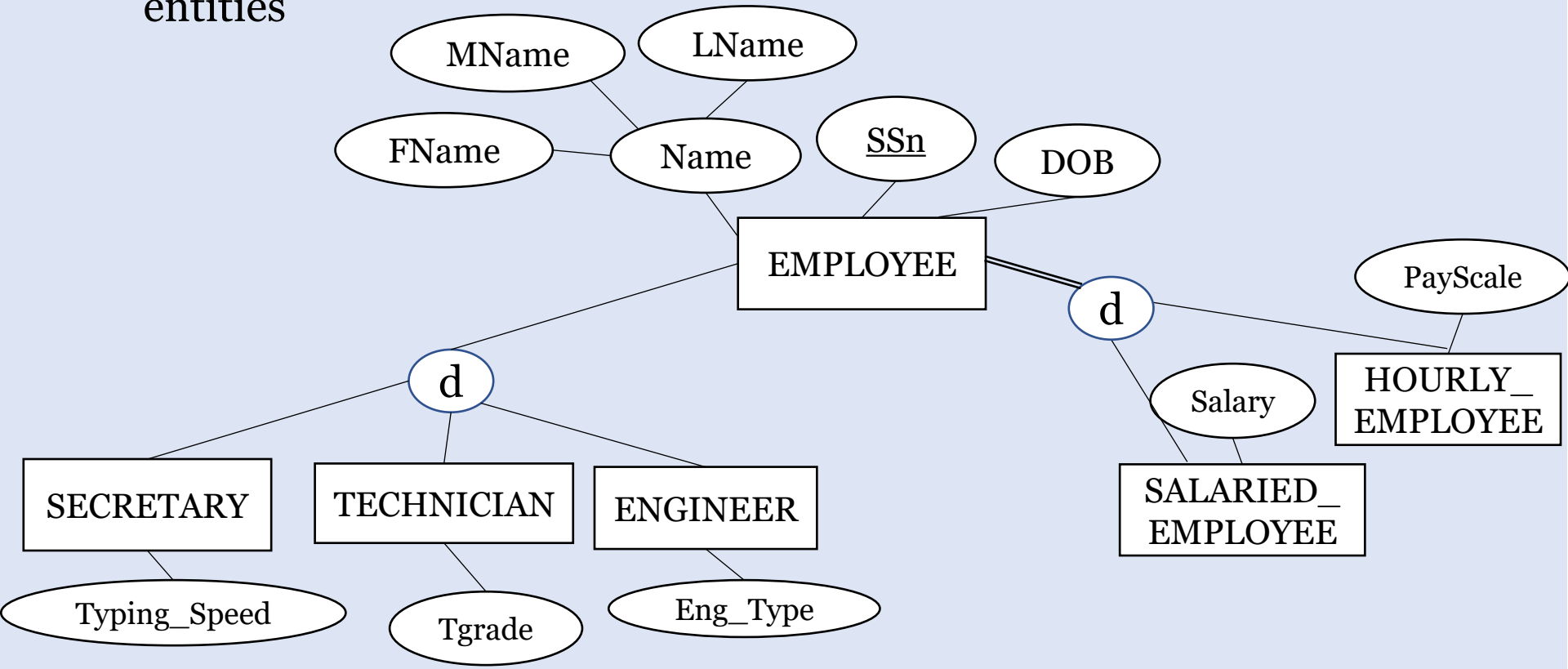
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



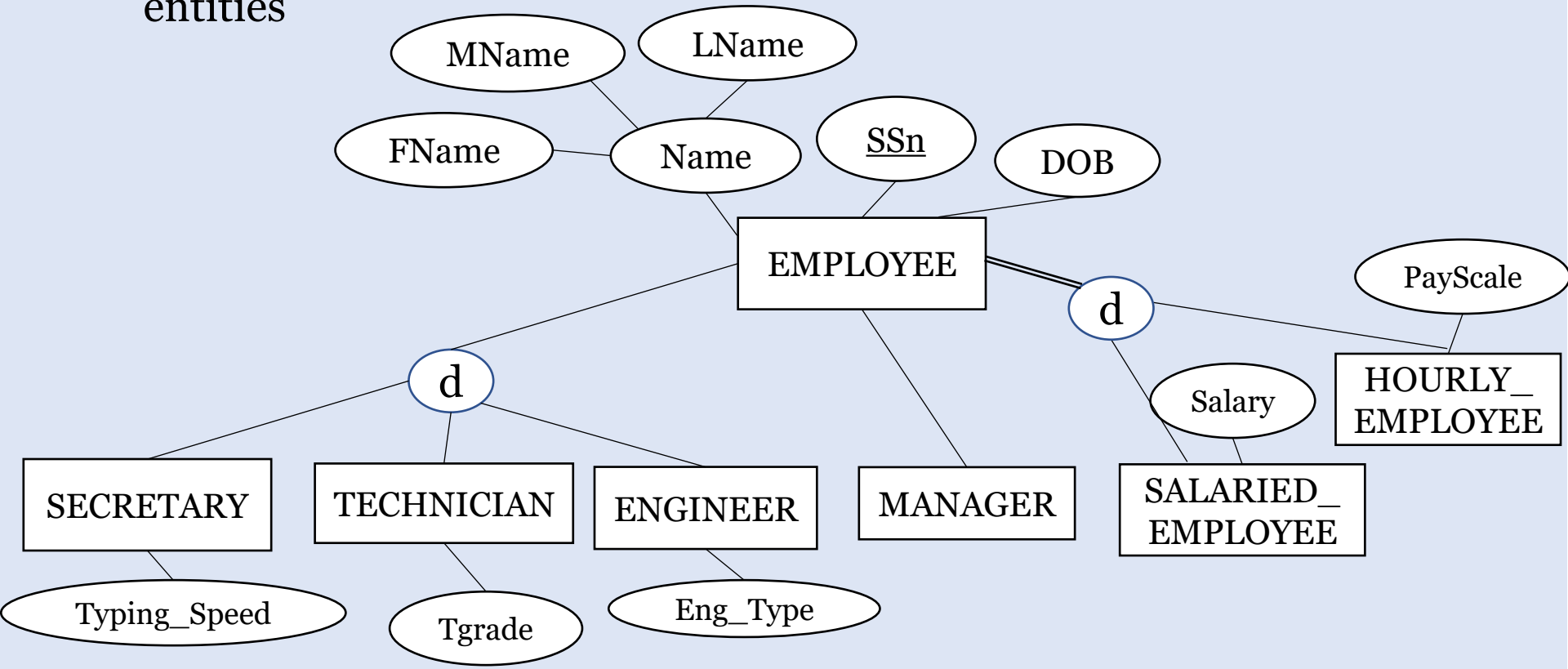
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



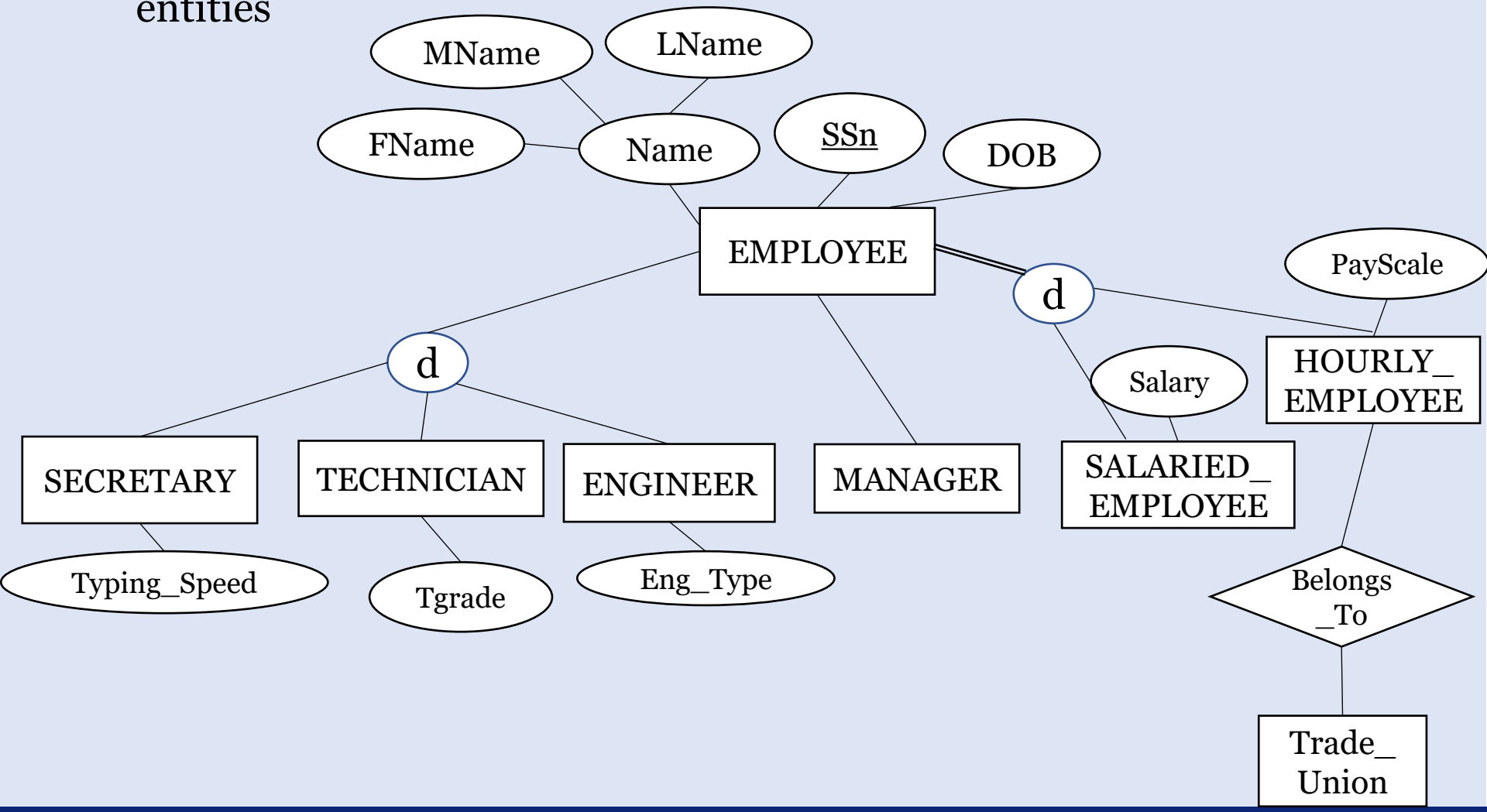
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



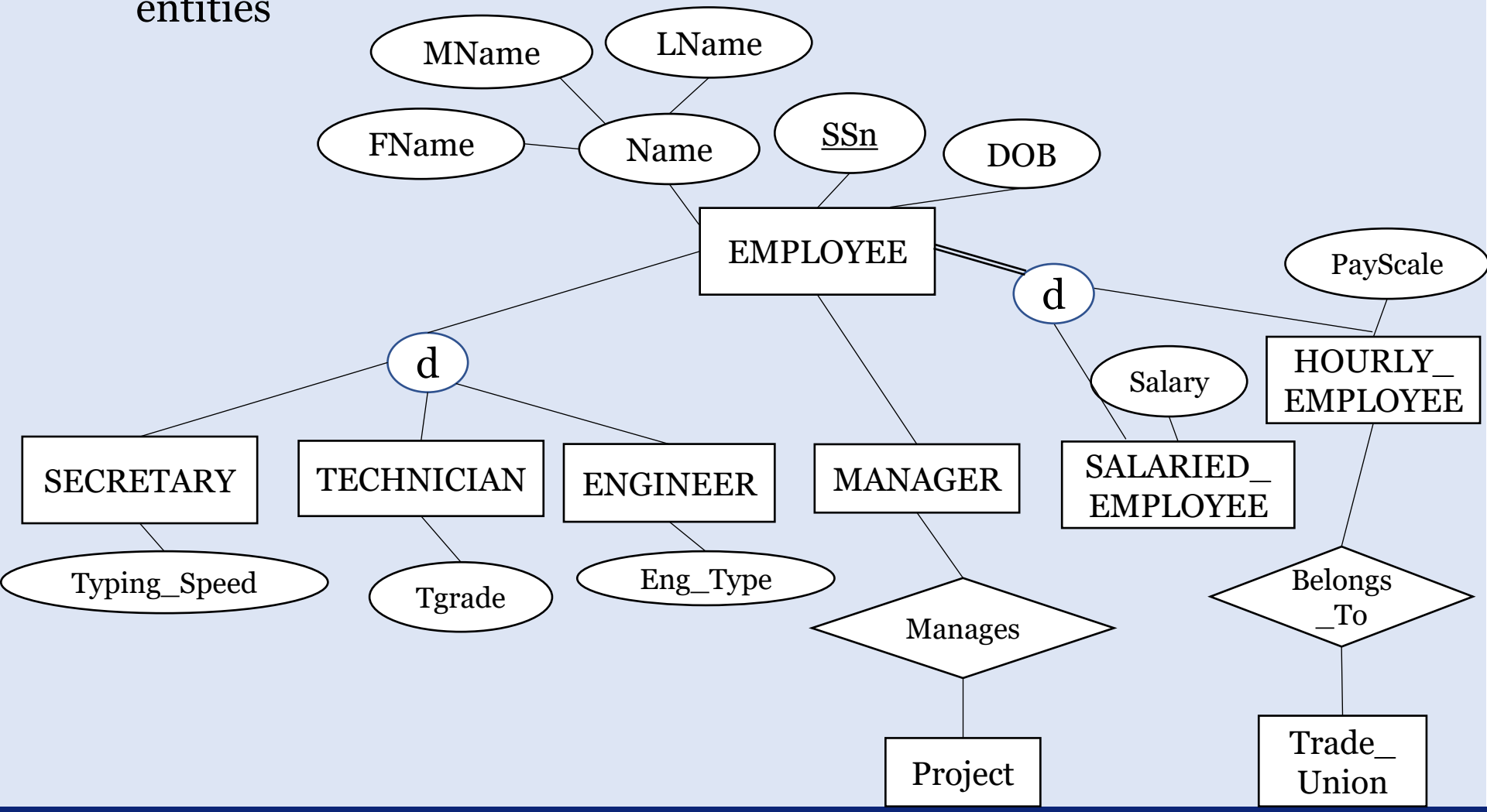
Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities



Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities

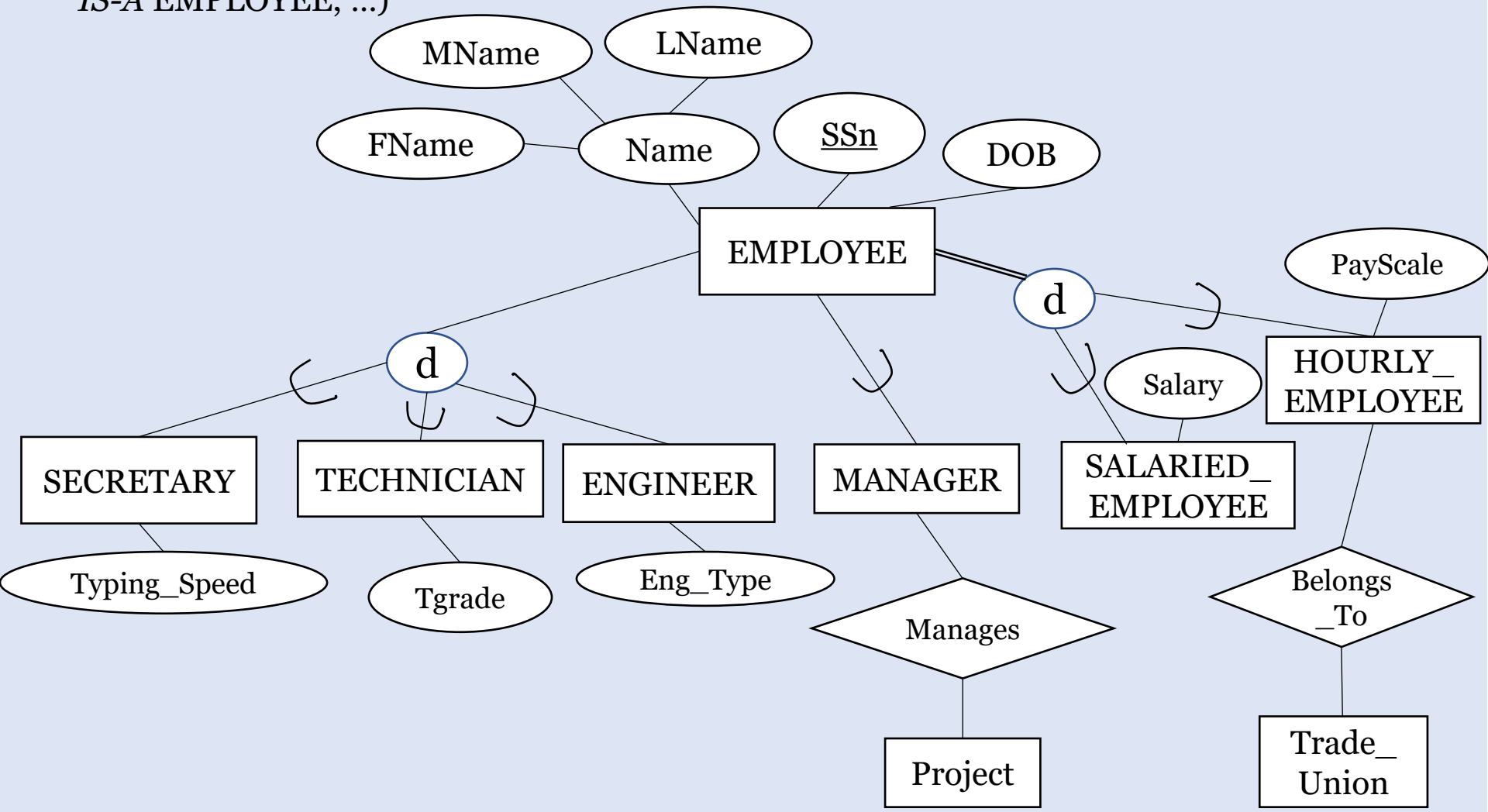
- Example: EMPLOYEE may be further grouped into SECRETARY, ENGINEER, MANAGER, TECHNICIAN, SALARIED_EMPLOYEE, HOURLY_EMPLOYEE,…
  - Each of these groupings is a **subset of EMPLOYEE** entities
  - Each is called a **subclass** of EMPLOYEE
  - EMPLOYEE is the **superclass** for each of these subclasses

- These are called **superclass/subclass** relationships.

- Example: EMPLOYEE/SECRETARY, EMPLOYEE/TECHNICIAN

Subclasses and Superclasses

- These are also called **IS-A relationships** (SECRETARY *IS-A* EMPLOYEE, TECHNICIAN *IS-A* EMPLOYEE, ...)



Subclasses and Superclasses

- Note: An entity that is member of a subclass represents the same real-world entity as some member of the superclass
  - The Subclass member is the same entity in a distinct specific role
  - An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the superclass
  - A member of the superclass can be optionally included as a member of any number of its subclasses

- Example: A salaried employee who is also an engineer belongs to the two subclasses ENGINEER and SALARIED_EMPLOYEE
- It is not necessary that every entity in a superclass be a member of some subclass

Attribute Inheritance in Superclass/ Subclass Relationship

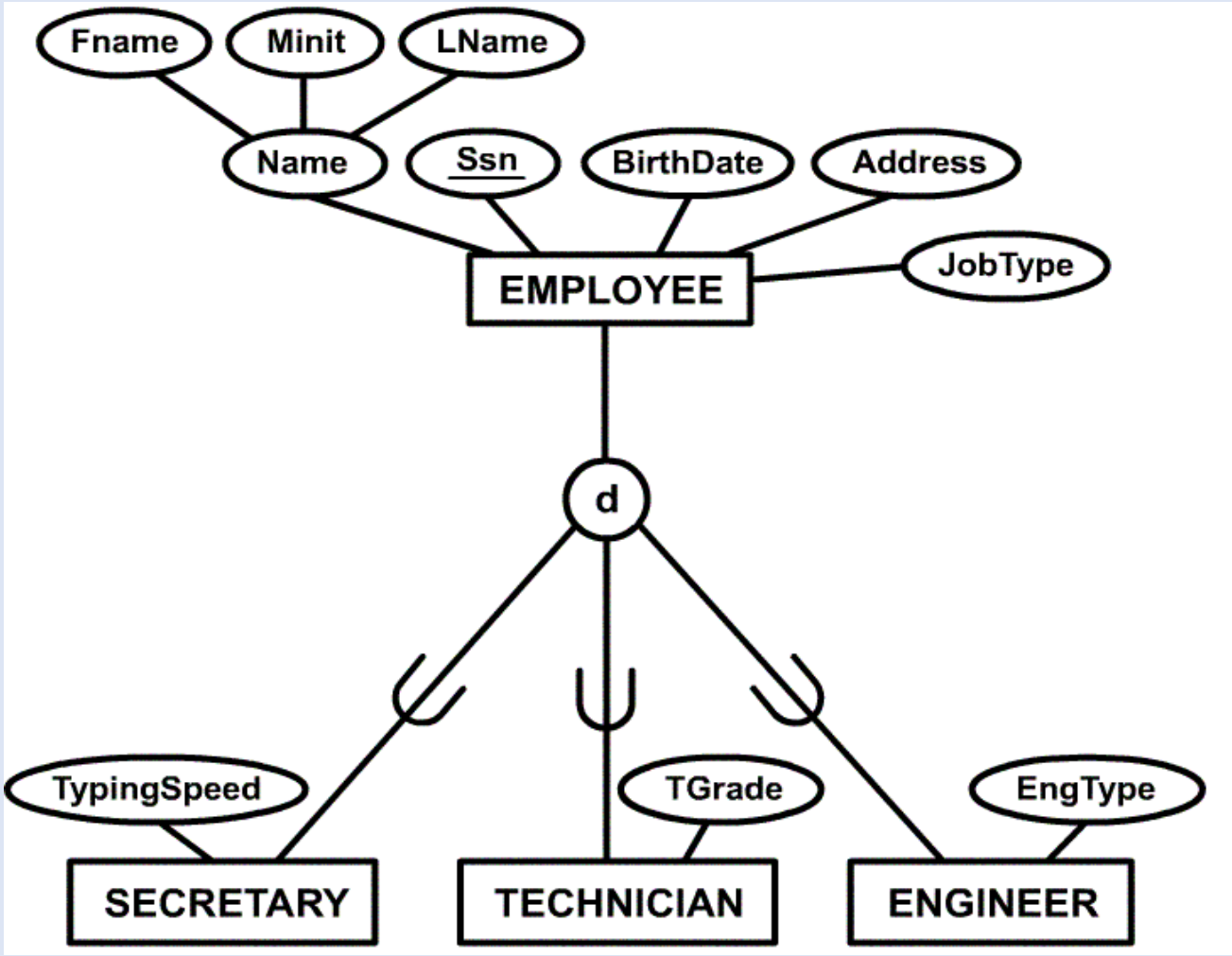
- An entity that is member of a subclass *inherits* all attributes of the entity as a member of the superclass
- It also inherits all relationships

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
- Top-Down Approach

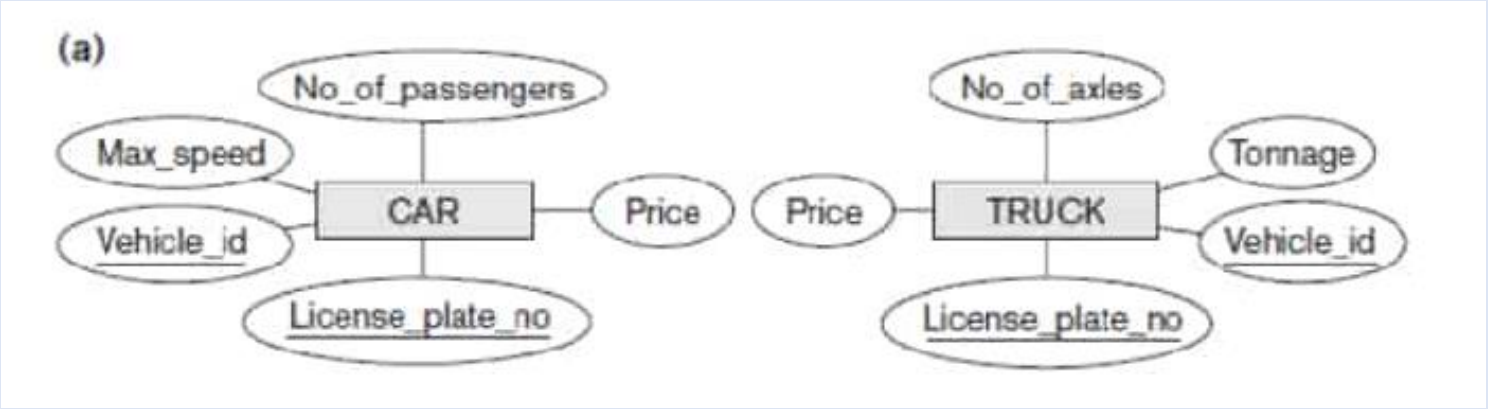
- Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon *job type*.
  - May have several specializations of the same superclass
- Example: Another specialization of EMPLOYEE *based in method of pay* is {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}.
  - Superclass/subclass relationships and specialization can be diagrammatically represented in EER diagrams
  - Attributes of a subclass are called specific/local attributes. For example, TypingSpeed of SECRETARY
  - The subclass can participate in specific relationship types. For example, BELONGS_TO of HOURLY_EMPLOYEE

Specialization Example

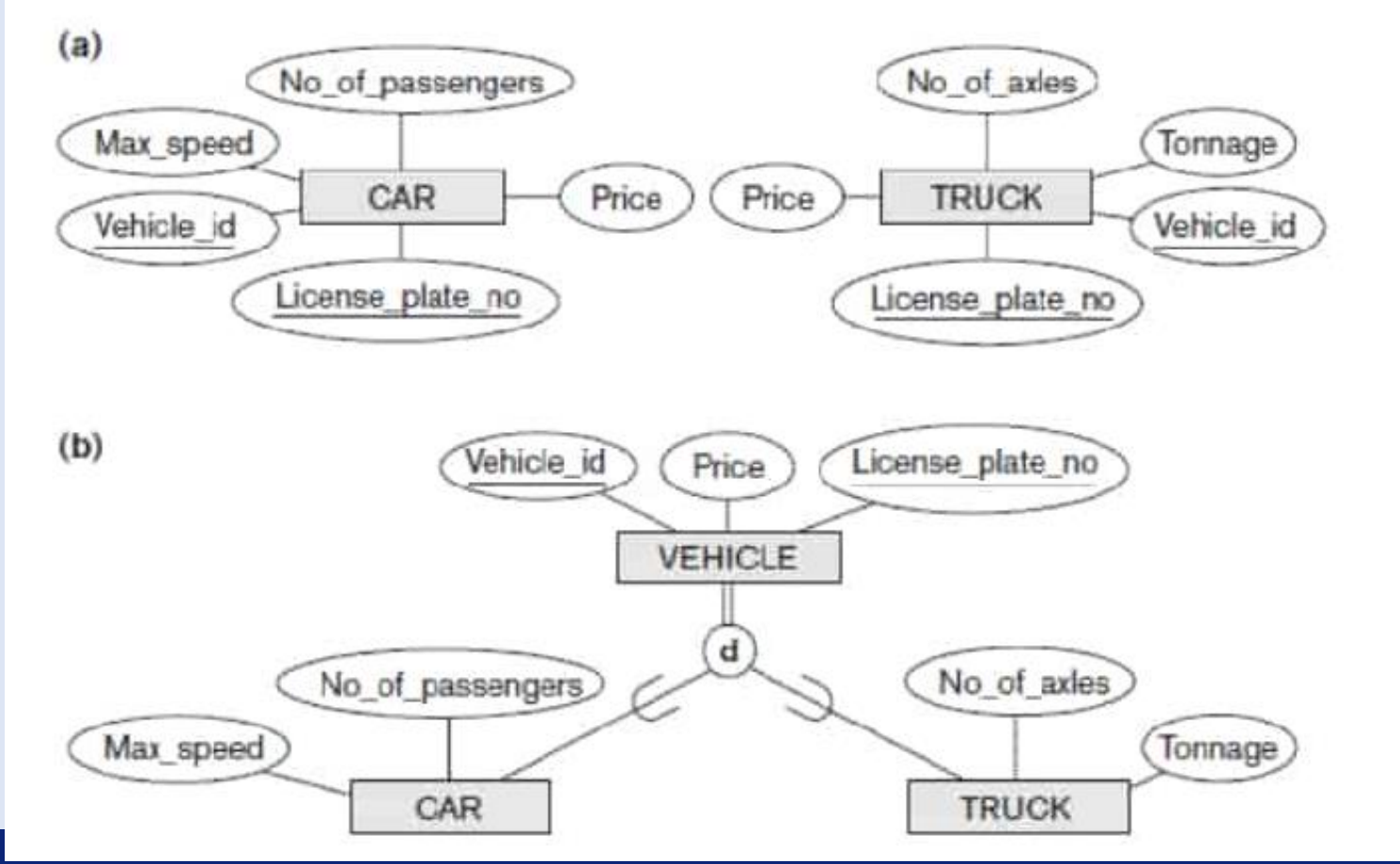


Generalization

- The reverse of the specialization process
- Several classes with common features are generalized into a superclass; original classes become its subclasses

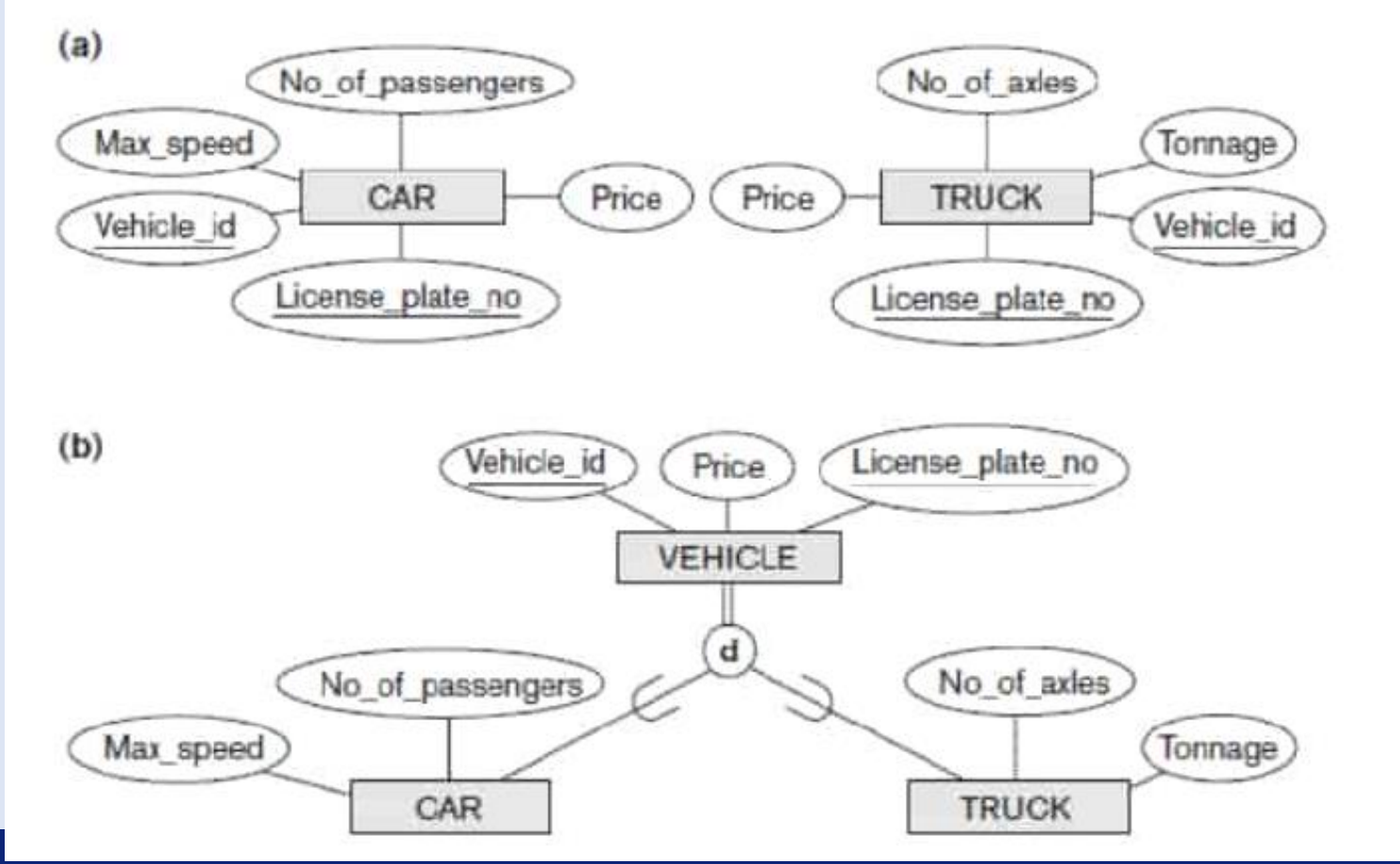


Generalization



Generalization

- We can view {CAR, TRUCK} as a specialization of VEHICLE
- Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK



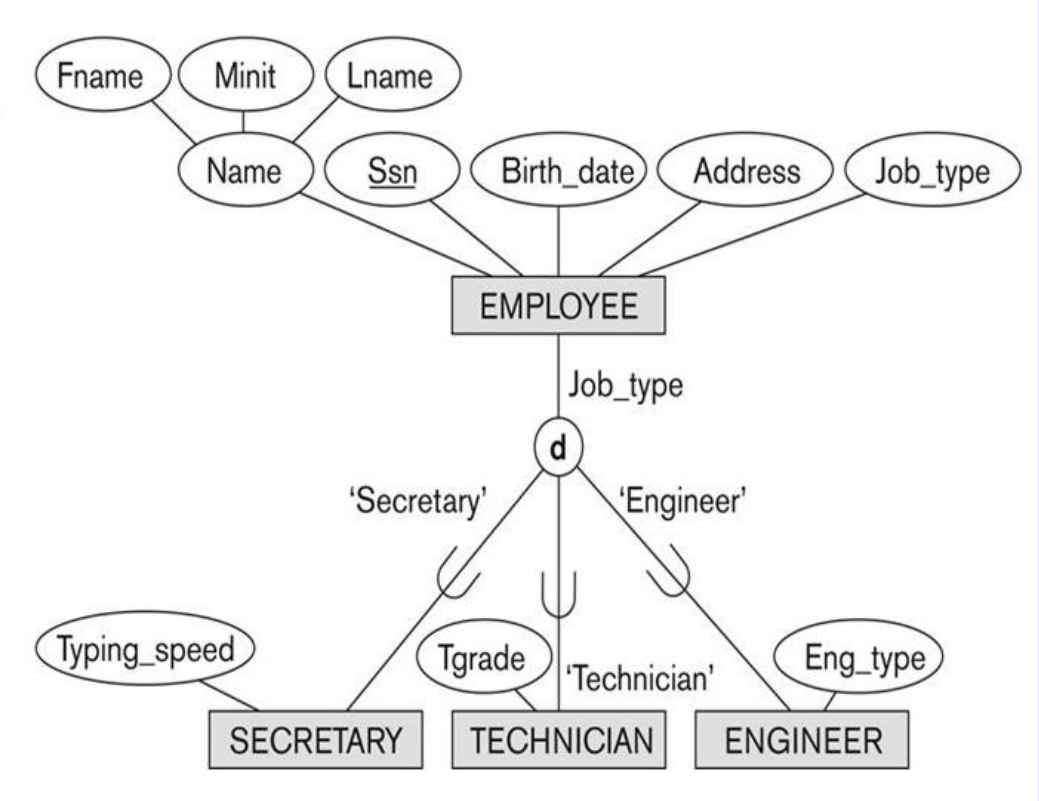
Generalization and Specialization

- Diagrammatic notation sometimes used to distinguish between generalization and specialization
  - Arrow pointing to the generalized superclass represents a generalization
  - Arrows pointing to the specialized subclasses represent a specialization

- Data Modeling with Specialization and Generalization
  - A superclass or subclass represents a set of entities
  - Shown in rectangles in EER diagrams (as are entity types)
  - Sometimes, all entity sets are simply called classes, whether they are entity types, superclasses, or subclasses

Constraints on Specialization and Generalization

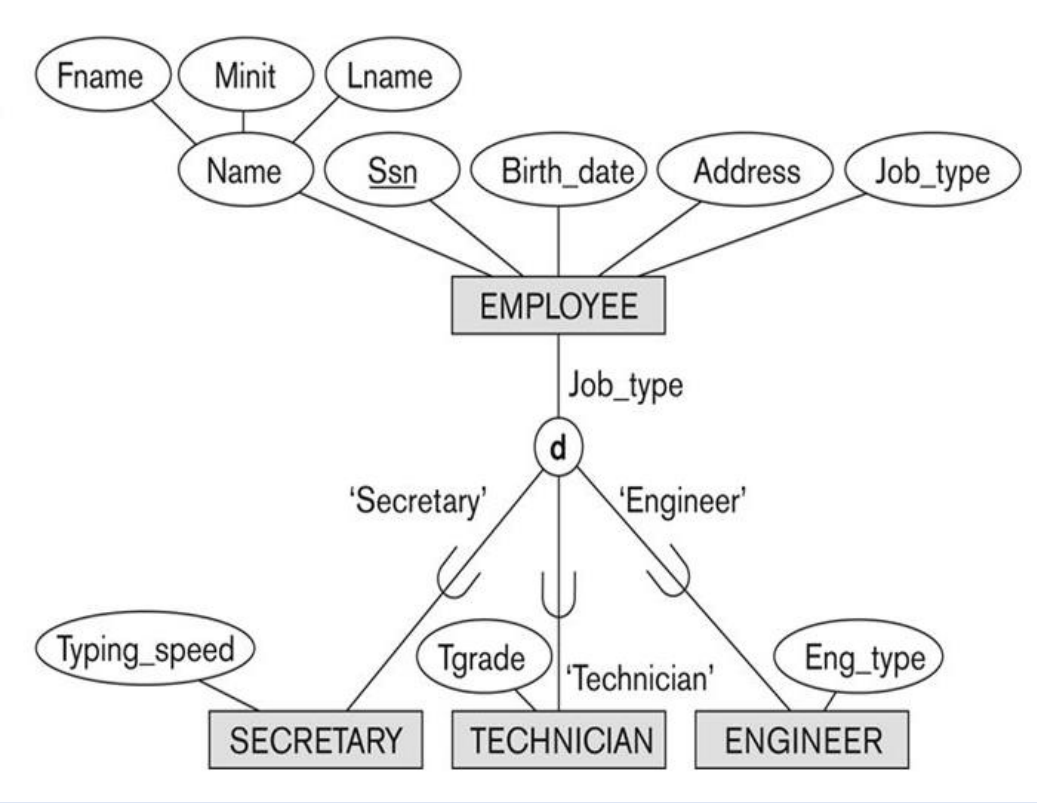
- If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called predicate-defined (or condition-defined) subclasses
  - Condition is a constraint that determines subclass members
  - Display a predicate-defined subclass by writing the predicate condition next to the line attaching the subclass to its superclass



Constraints on Specialization and Generalization

- If all subclasses in a specialization have membership condition on same attribute of the superclass, specialization is called an attribute defined-specialization

- Attribute is called the defining attribute of the specialization
- Example: JobType is the defining attribute of the specialization {SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE

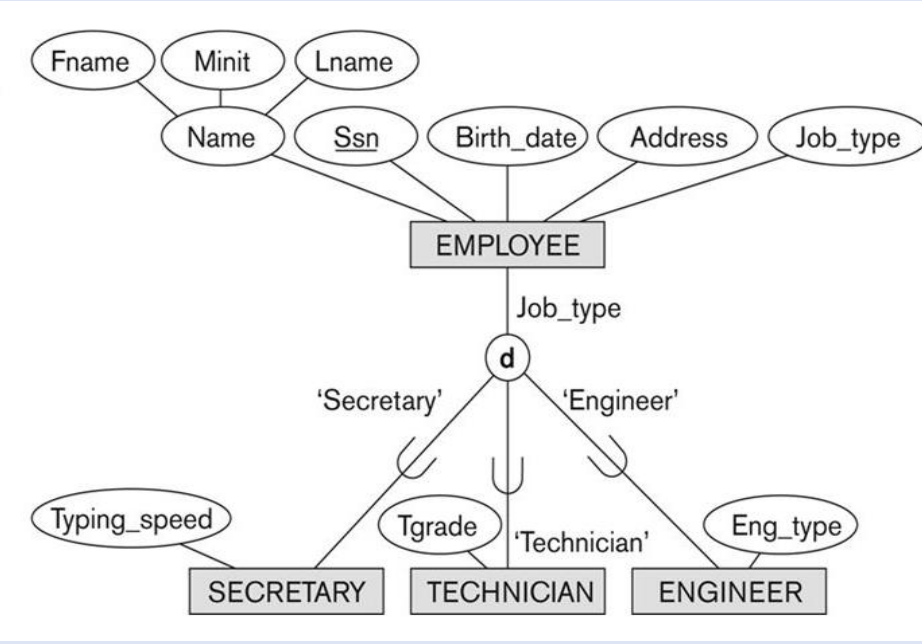


Constraints on Specialization and Generalization

- If no condition determines membership, the subclass is called user-defined
  - Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass
  - Membership in the subclass is specified individually for each entity in the superclass by the user

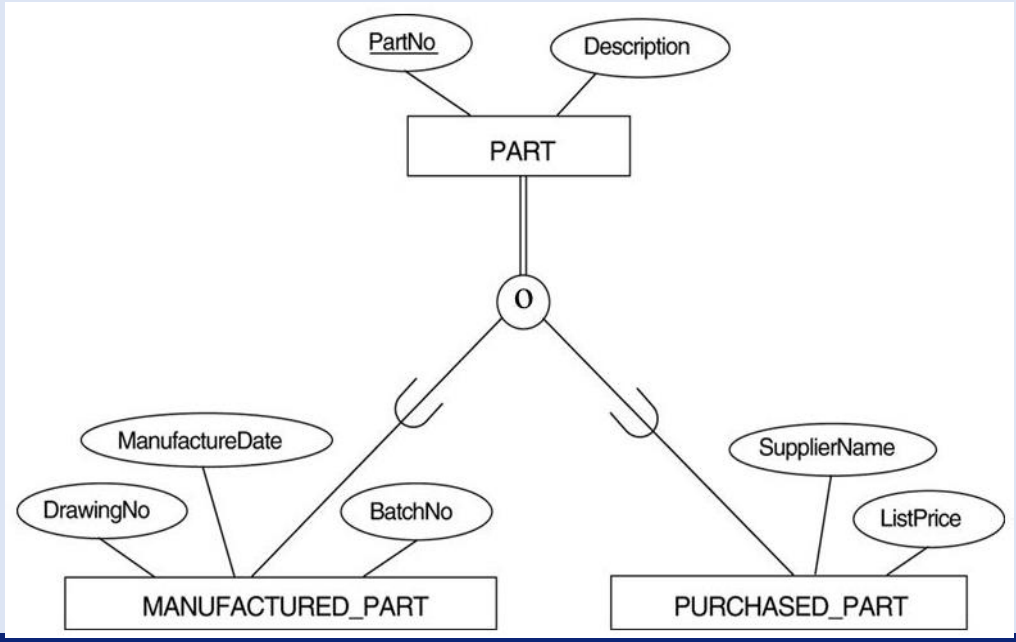
Constraints on Specialization and Generalization

- **Disjointness Constraint:**
  - Specifies that the subclasses of the specialization must be disjointed (an entity can be a member of at most one of the subclasses of the specialization)
  - Specified by **d** in EER diagram



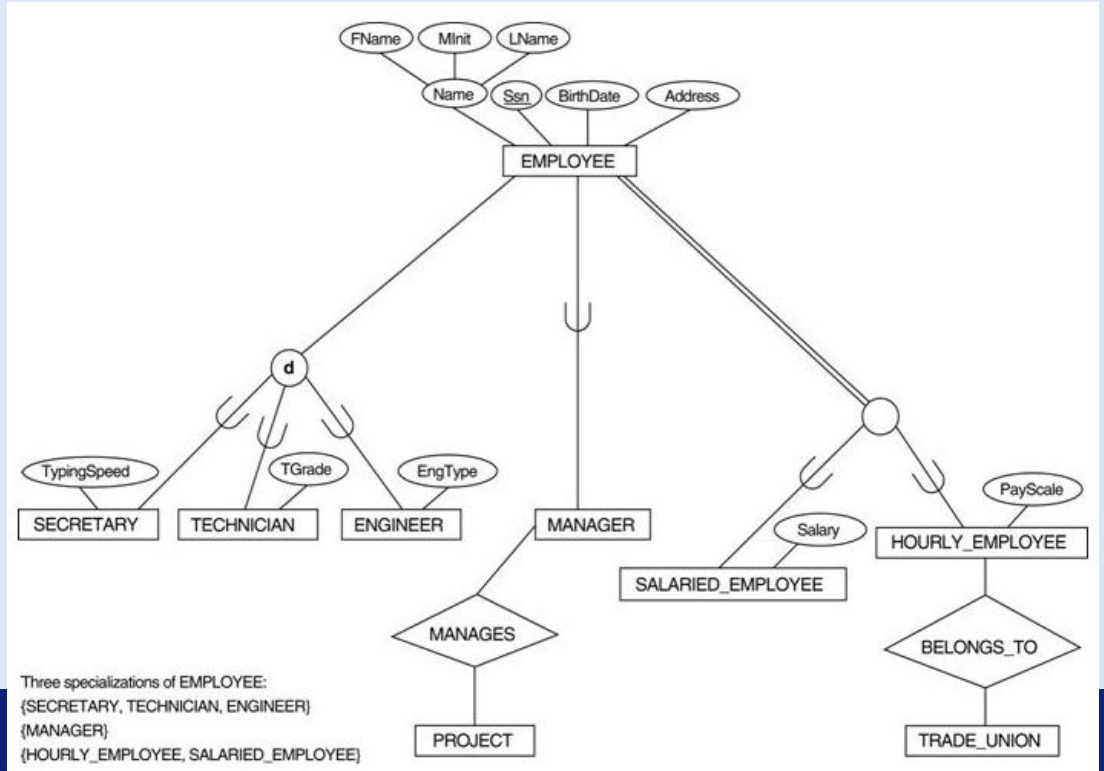
Constraints on Specialization and Generalization

- **Disjointness Constraint:**
  - Specifies that the subclasses of the specialization must be disjointed (an entity can be a member of at most one of the subclasses of the specialization)
  - Specified by **d** in EER diagram
  - If not disjointed, overlap; that is the same entity may be a member of more than one subclass of the specialization
  - Specified by o in EER diagram



Constraints on Specialization and Generalization

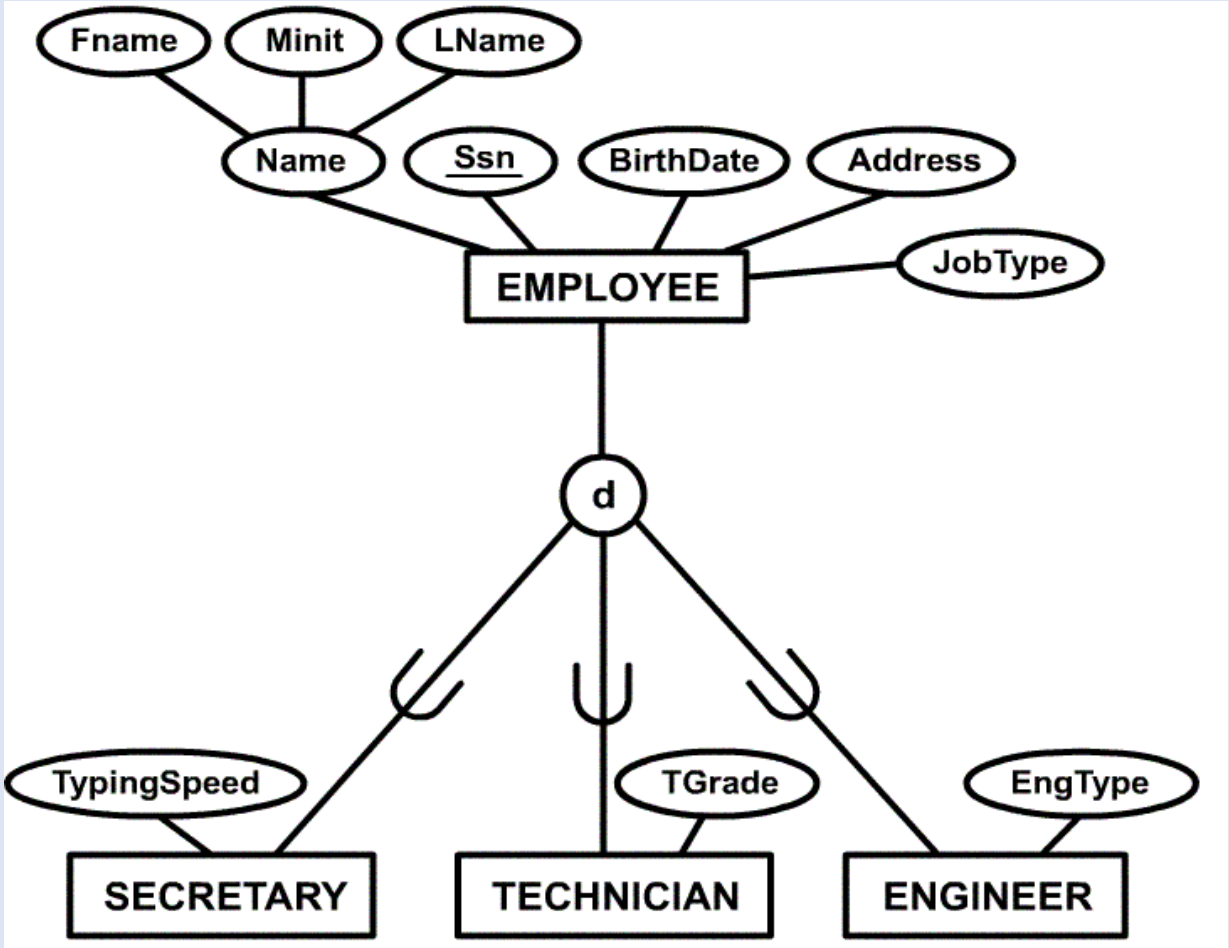
- **Completeness Constraint:**
  - Total specifies that every entity in the superclass must be a member of some subclass in the specialization/ generalization
  - Shown in EER diagrams by a double line
  - Partial allows an entity not to belong to any of the subclasses
  - Shown in EER diagrams by a single line



Constraints on Specialization and Generalization

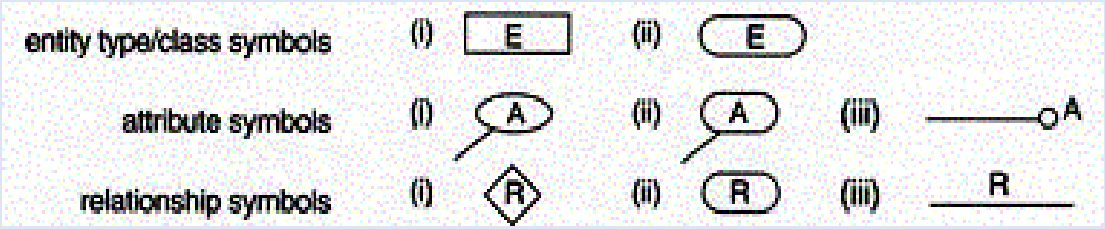
- Hence, we have four types of specialization/generalization:
  - Disjoint, total
  - Disjoint, partial
  - Overlapping, total
  - Overlapping, partial
- Note: Generalization usually is total because the superclass is derived from the subclasses.

Disjoint Partial Specialization Example

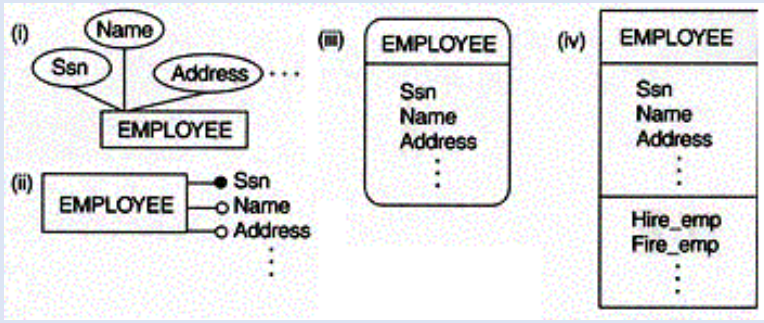


Alternative Diagrammatic Notation

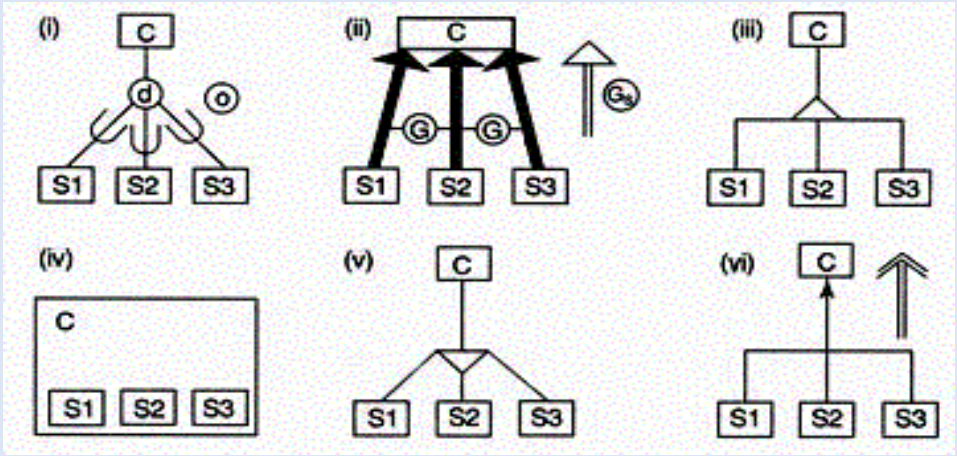
Symbols for entity type / class, attribute and relationship



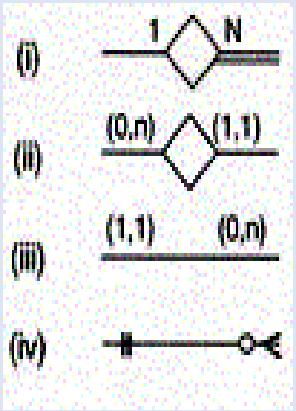
Displaying attributes



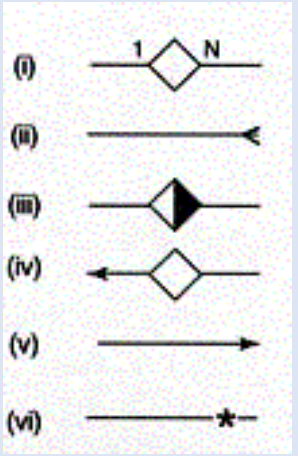
Notations for displaying specialization / generalization



Various (min, max) notations



Displaying cardinality ratios



Difference between Specialization and Generalization

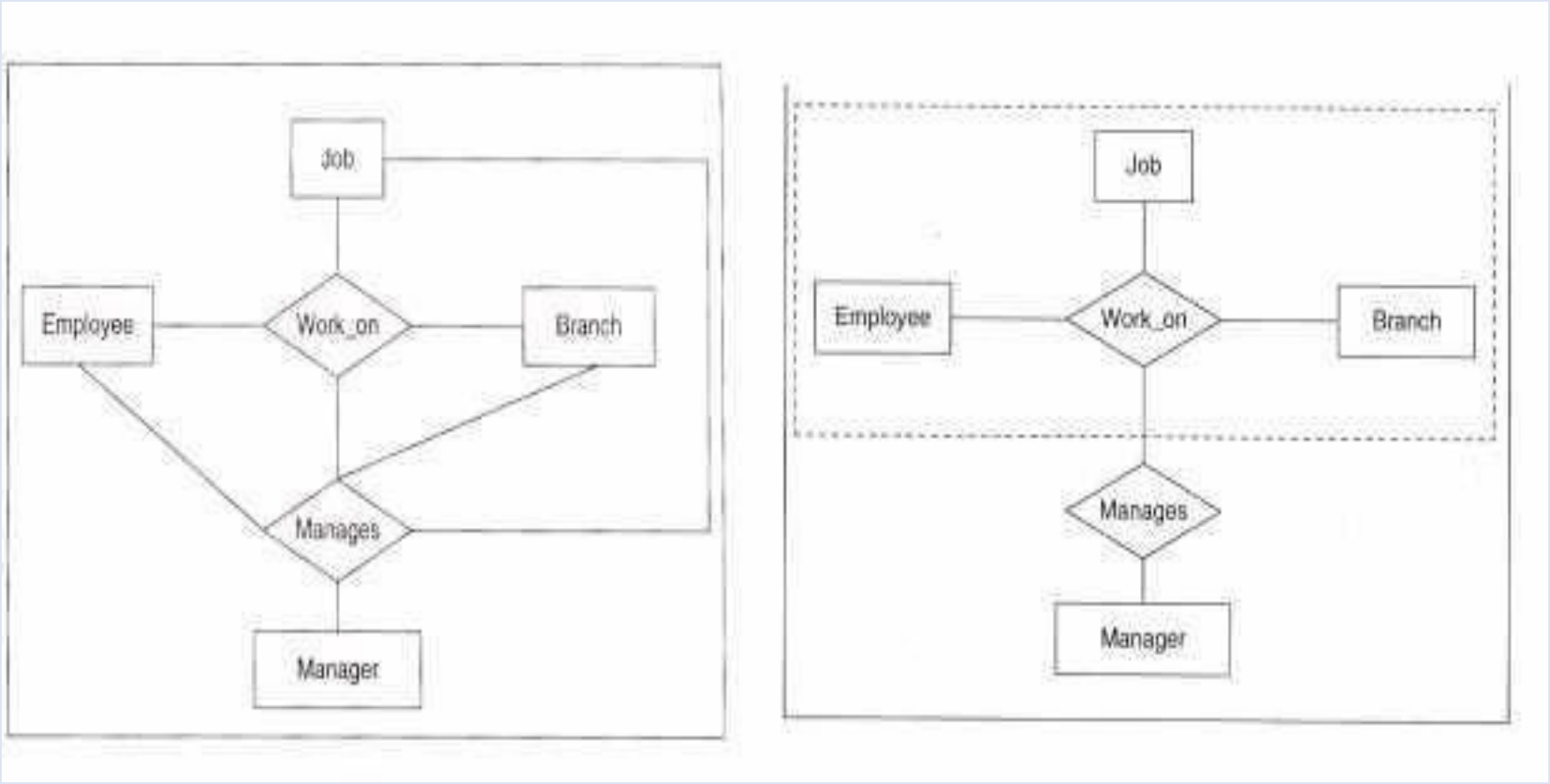
- Generalization is a bottom-up approach. However, specialization is a top-down approach.
- Generalization club all the entities that share some common properties to form a new entity. On the other hands, specialization spilt an entity to form multiple new entities that inherit some properties of the spiltted entity.
- Generalization helps in reducing the size of schema whereas, specialization is just opposite it increases the number of entities thereby increasing the size of a schema.
- Generalization is always applied to the group of entities whereas, specialization is always applied on a single entity
- Generalization results in a formation of a single entity whereas, Specialization results in the formation of multiple new

Aggregation

- One limitation of the E-R model is that it cannot express relationships among relationships.
- The best way to model a situation like this is by the use of aggregation

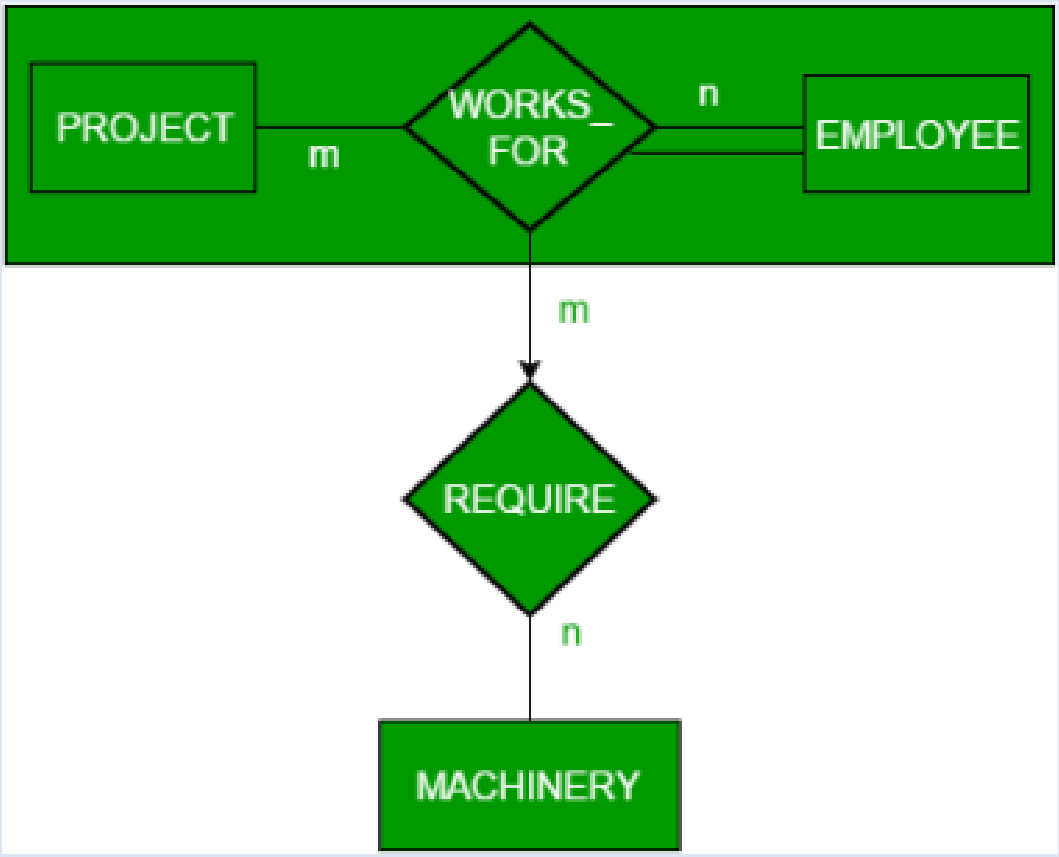
Aggregation

- One limitation of the E-R model is that it cannot express relationships among relationships



Aggregation

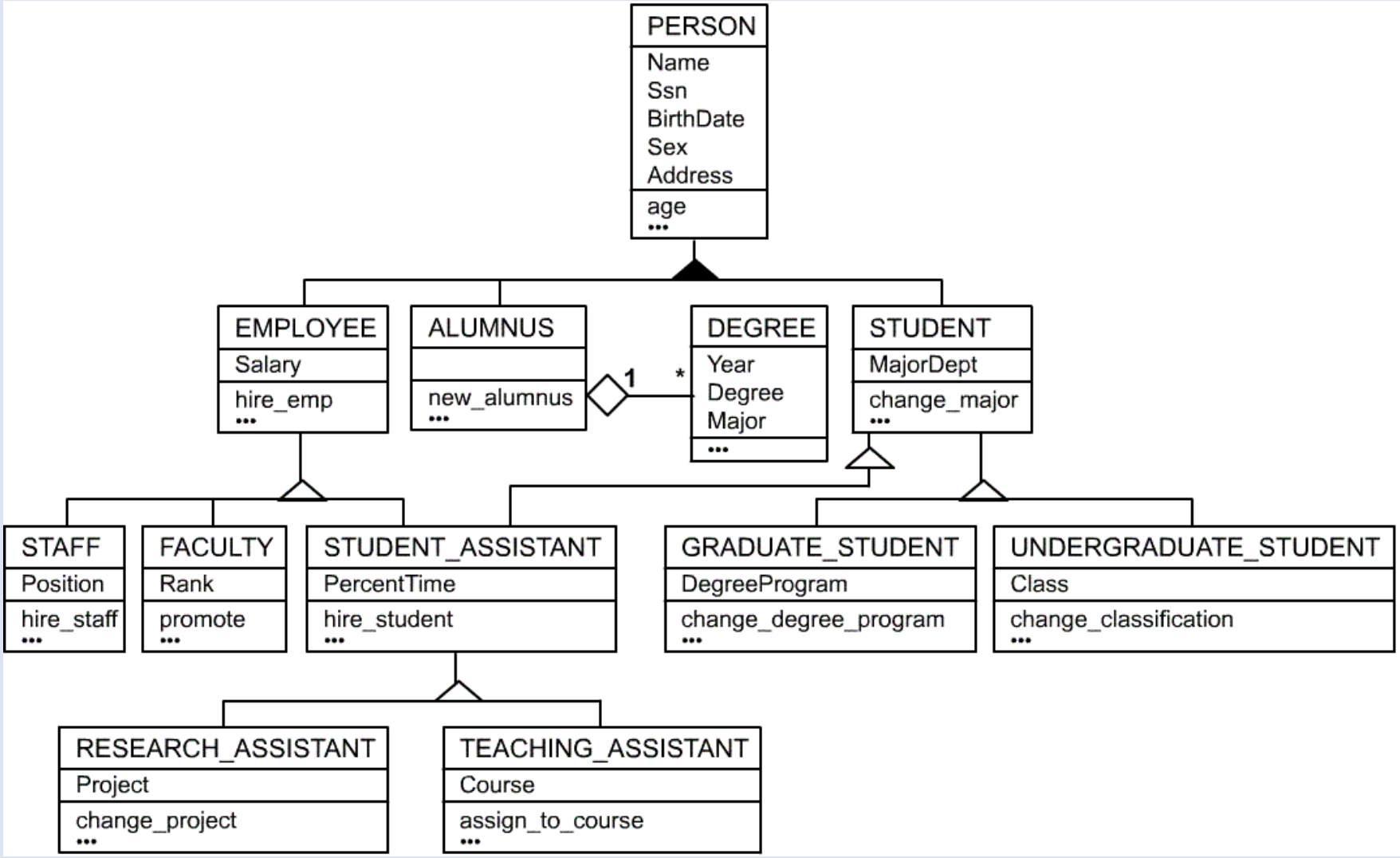
- One limitation of the E-R model is that it cannot express relationships among relationships.
- The best way to model a situation like this is by the use of aggregation



To represent aggregation, create a schema containing:

1. primary key of the aggregated relationship
2. primary key of the associated entity set
3. descriptive attribute, if exists.

UML Example of Specialization/ Generalization



Steps to Design E-R Diagram

- Identify the Entities
- To find relationships among these entities
- To identify the key attributes for every Entity.
- To identify other relevant attributes
- To draw the complete e-r diagram with all attributes including primary key

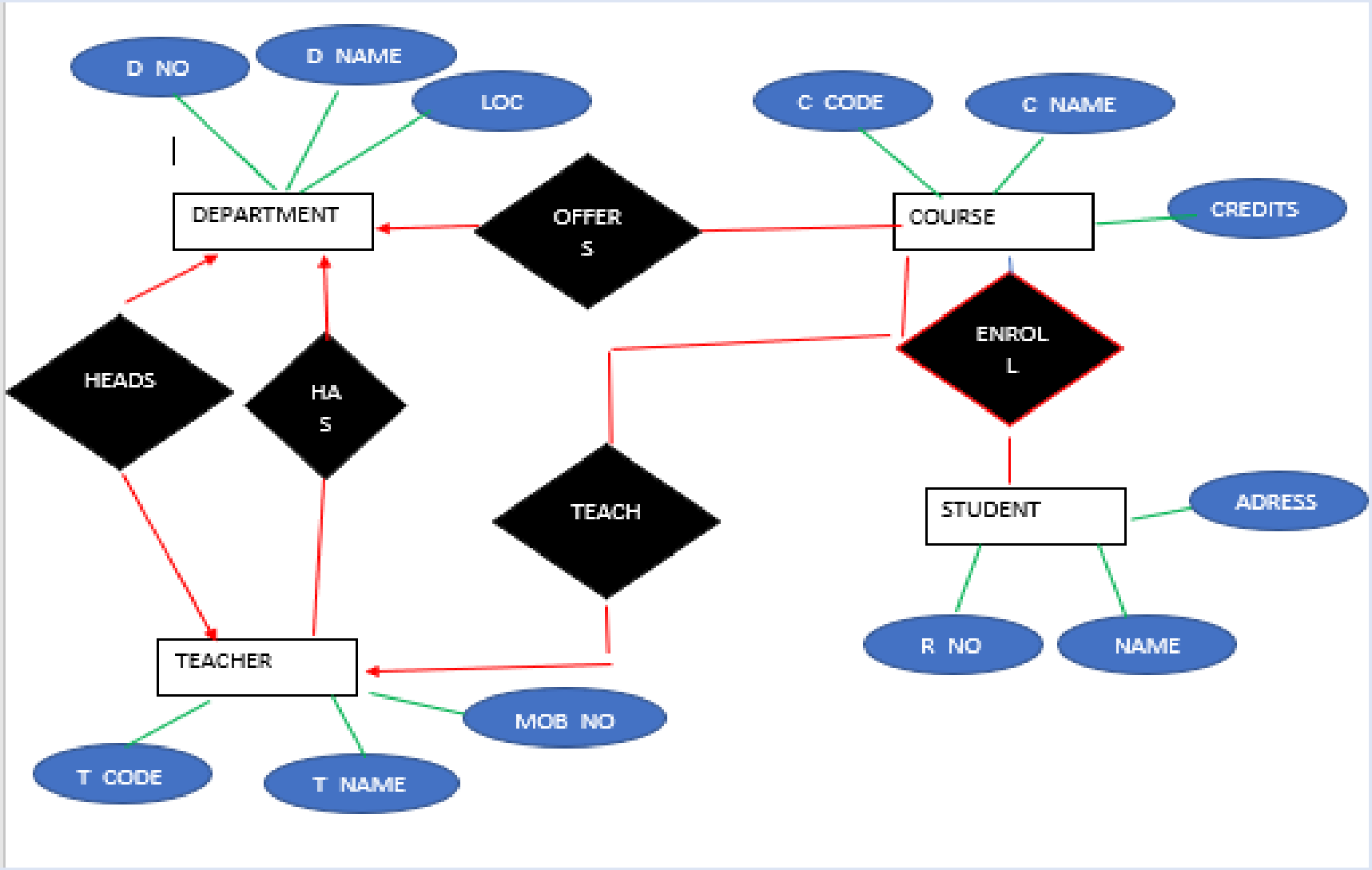
Case Study

- Consider, a university contains many departments.
  - Each department can offer any number of courses.
  - Many teachers can work in a department.
  - A teacher can work only in one department.
  - For each department there is a Head.
  - A teacher can be head of only one department.
  - Each teacher can take any number of courses.
  - A student can enroll for any number of courses.
  - Each course can have any number of students.

Case Study

- To identify the Entities
  - University, Department, Course, Teacher, Student Consider University as single instance
- Relationship:
  - Each course belong to one department(1:M) Department & teacher(1:M)
  - Department Head and teacher(1:1) Teacher and course(1:M)
  - Student & course(M:M)
- Key Attributes:
  - D_no,C_code,Roll_no,t_code
- Relevant Attributes:
  - d_name,loc,c_name,credits,t_name,mo b_no,name,address

Case Study


D NO
D NAME
LOC
DEPARTMENT
OFFER
S
C CODE
C NAME
COURSE
CREDITS
HEADS
HA
S
ENROL
L
TEACH
STUDENT
ADRESS
R NO
NAME
TEACHER
MOB NO
T CODE
T NAME

Thanks!