DBMS college Notes (2)

Topic → ER modeling, Enhanced ER Specialization Generalization

ER model

eg Company Database

- Requirements of the company

1) The company is organized into DEPARTMENTS. Each department has a name, number and an employee who manages the department. We keep track of start date of department manager.
2) Each department controls number of PROJECTS. Each project has a name, number & is located at single location.

ER model concepts

Entities & Attributes

- Entities are specific objects or things in mini-world that are represented in database. For eg the EMPLOYEE John Smith, Research DEPARTMENT, the product X PROJECT.
- Attributes are properties used to describe an entity. for eg an EMPLOYEE entity may have NAME, SSN, Address, Sex, Birth Date.
- A specific entity will have a value for each of its attributes for ex. a specific employee entity may have Name = 'John Smith', SSN = '12345', Address = '731, fondren, Houston', Sex = 'M'.

Each attribute has a value set (or data type) associated with it eg Integer, String, Subrange, enumerated type.

Types of Attributes

1) Simple
Each entity has a single atomic value for attribute eg SSN, Sex

2) Composite
The attribute may be composed of several components. for eg Name (FirstName, MiddleName, LastName), Address (Apt#, House#, Street, city, State, Zipcode, Country).

3) Multivalued
An entity may have multiple values for attribute. for eg COLOR of a CAR or Previous Degrees of a STUDENT Denoted as {COLOR} or {Previous Degrees}.

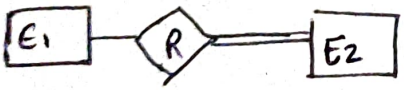
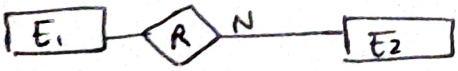
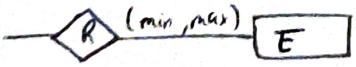
Nesting between them is possible

eg Previous Degrees of a STUDENT is composite multi-valued attribute denoted by {Previous Degrees (college, year, degree, field)}

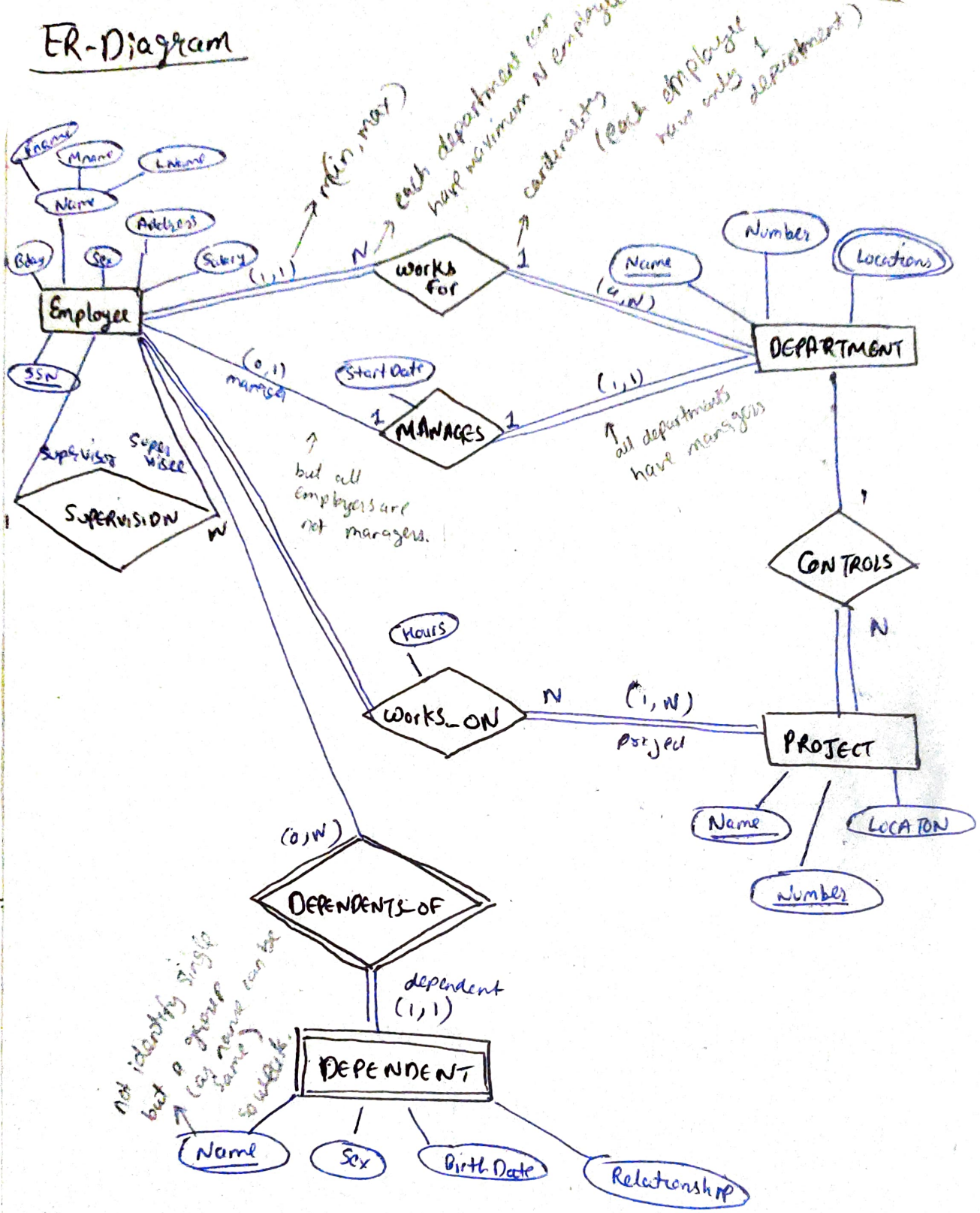
Relationships & Relationship types

- A relationship relates two or more distinct entities with a specific meaning. for eg EMPLOYEE JohnSmith works on ProductX PROJECT or EMPLOYEE Franklin Wong manges the Resarch Department.
- The degree of relationship type in number of participating entities type. Both MANAGES & WORK_ON are binary relationship.

Notations for ER Schemas

Symbol	Meaning
	Entity Type
	Weak Entity Type
	Relationship Type
	Identifying relationship type
	Attribute
	Key Attribute
	Multivalued attribute
	Composite attribute
	Derived attribute (like age can be derived from DoB.)
	Total participation of E_2 in R
	Cardinality ratio of 1:N for $E_1 : E_2$ in R
	Structural Constraint (min, max) on Participation of E in R.

ER-Diagram



- Relationship types are : works_for , Manages , Work_on , Controls , SUPERVISION , dependents_of
- weak entity type is : Dependent
  Identifying relationship is dependents_of
- Recursive Relationship type is : - SUPERVISION

Weak Entity types

→ A entity that does not have key attribute

- A weak entity must participate in identifying relationship with owner or identifying entity type.
- Entities are identified by combination of:
  - A partial key of weak entity type
  - The particular entity they are related to in identifying entity type.

eg DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via identifying relationship type DEPENDENT_OF.

Constraints on Relationships

Cardinality Ratio

Specifies maximum number of relationship instances that an entity can participate in.

- One-to-one (1:1)
- One-to-many (1:N) or Many-to-one (N:1)
- Many-to-Many

Participation

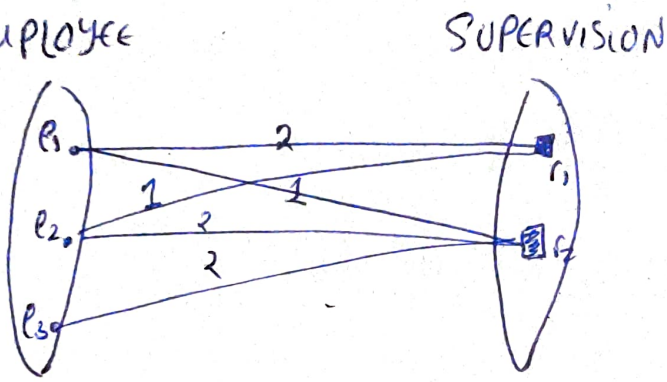
- Also called minimum cardinality constraint
- Specifies minimum number of relationship instances that each entity can participate in.

* Weak entity always comes with total participation

- Recursive relationship type

1) Both participations are same entity type in different roles.
2) Eg: SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) & (another) EMPLOYEE (in role of subordinate)
3) In ER diagram, need to display role names to distinguish participations.

In following figure, first role participation labeled with 1 and second role participation labeled with 2.



A relationship type can have attributes, for example, HoursPerWeek of WORKS_ON; its value for each relationship instance describes the number of hours per week that an EMPLOYEE works on PROJECT.

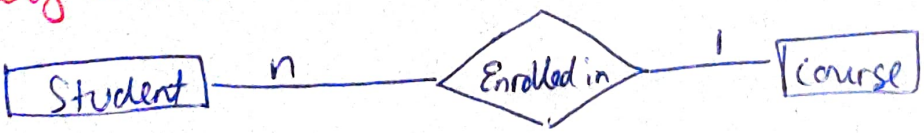
Alternative (min, max) notations for relationship structural constraints

- Specified on each participation of an entity type E in relationship type R.
- Specifies that each entity e in E participates in at least min & at most max relationship instances in R.

- Default (no constraint) : min = 0 , max = n
- Eg.
  - A department has exactly one manager & an employee can manage at most one department
    - Specify (0, 1) for participation of EMPLOYEE in MANAGES
    - Specify (1, 1) for participation of DEPARTMENT in MANAGES

- An employee can work for exactly one department but a department can have any no. of employee.
  - Specify (1,1) for participation of Employee in WORKS_FOR
  - Specify (0,N) for participation of DEPARTMENT in WORKS_FOR

* Many to one relationship



- n students enrolled in one course
- for one student there is one course

Enhanced - ER (EER) Model (E2R or EER)

- Includes all modeling concept of basic ER.
- Additional concepts : subclass / super classes, specialization generalization, categories, attribute inheritance
- It includes some object-oriented concepts like inheritance.

Subclasses or Superclasses

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

eg EMPLOYEE may be further grouped into SECRETARY, ENGINEER, MANAGER, SALARIED_EMPLOYEE, HOURLY_EMPLOYEE.

- Each is called subclass of EMPLOYEE
- EMPLOYEE is superclass for each of these subclasses.

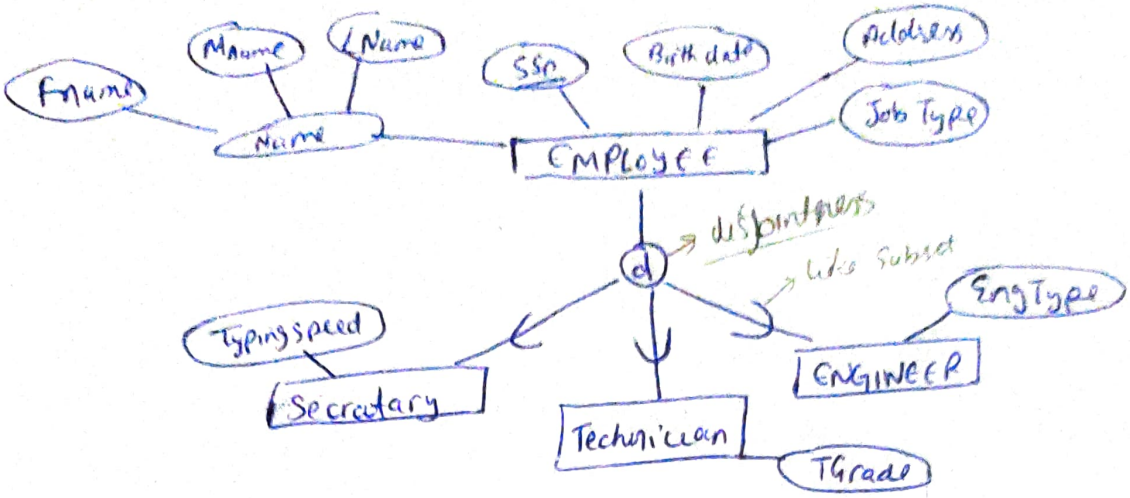
→ These are called superclass / subclass relationship
→ EMPLOYEE / SECRATARY, EMPLOYEE / TECHNICIAN

- These are also called IS-A relationship
  eg Secretary IS-A employee.
- An entity that is member of subclass inherits all attributes of entity as a member of superclass, it also inherits all relationships

Specialization

- Is process of defining a set of subclasses of superclass.
- Ex. {Secretary, Engineer, Technician} is specialization of EMPLOYEE based upon job type.
- Another specialization of EMPLOYEE based on Method of pay is {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}.

Attribute of Subclass are called specific attributes. For eg. TypingSpeed of SECRETARY.



Generalization

- Reverse of specialization process
- eg VEHICLE is generalization of CAR and TRUCK

- **Disjointness constraint**
  - Specifies that the subclasses of specialization must be disjointed (an entity can be member of at most one of the subclasses of of 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 specialization.
  - Specified by o in EER digram.
- **Completeness Constraint**
  - Total specifies that every entity in Superclass must be a member of some subclass in specialization / generalization
  - Shown in EER digram by double line
  - Partial allows an entity not to belong to any of subclass
  - Shown in EER diggrams by single line

Hence we have 4 types of specialization / generalization

1) Disjoint, total 2) Disjoint, partial 3) Overlapping, total
4) Overlapping, partial

→ Generalization usually is total because superclass is derived from subclasses.

⇒ Hierarchy has a constraint that every subclass has only one superclass (single inheritance)

⇒ In a Lattice, a subclass can be subclass of more than one superclass (multiple inheritance)

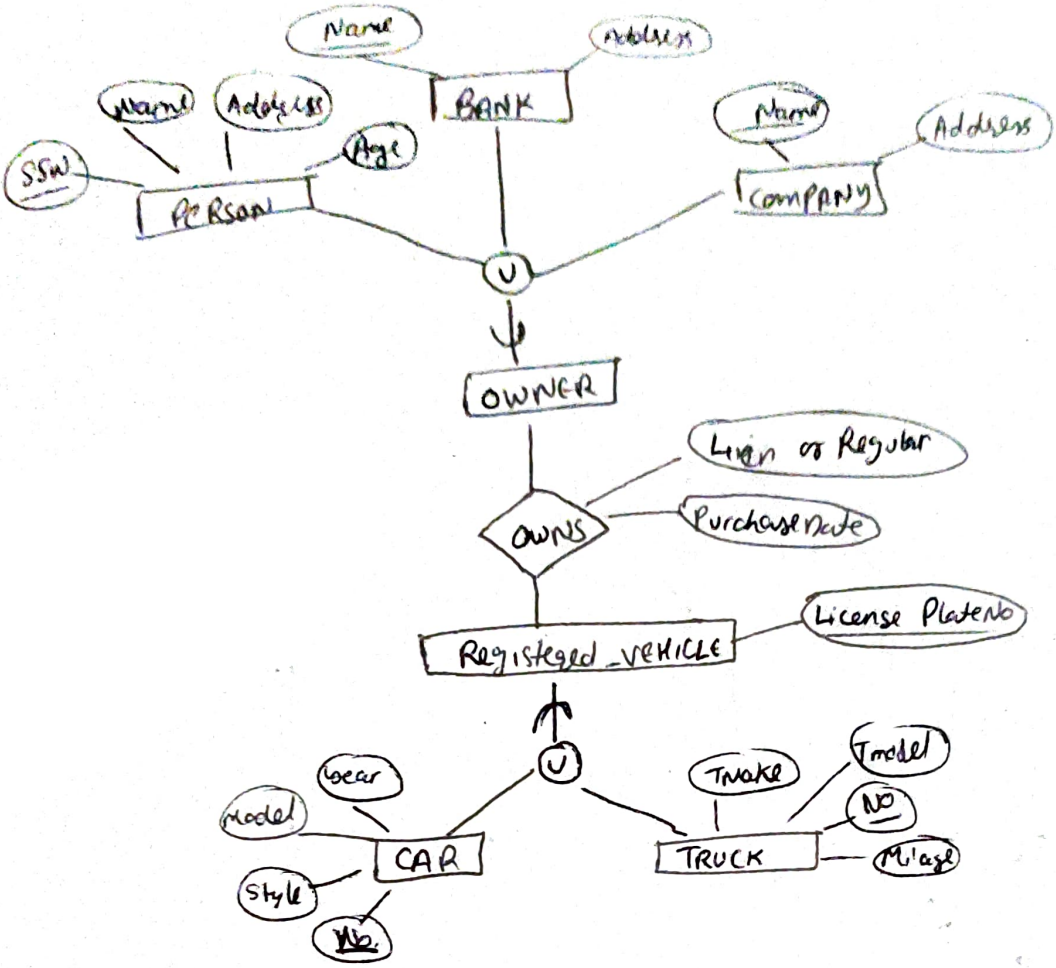
In lattice or hierarchy, a subclass inherits attributes not only of its direct superclass, but also of its predecessor superclass.

Categories (Union types)

- A shared subclass is subclass in more than one distinct superclass / subclass relationships where each relationships has a single superclass (multiple inheritance)
- Superclasses represent different entity type. Such a subclass is called a category or UNION Type.
  eg. Database for vehicle registration, vehicle owner can be person, a bank or a company.

→ Category (subclass) OWNER is subset of union of 3 superclasses COMPANY, BANK & PERSON

→ A category member must exist in at least one of its superclasses.



FORMAL DEFINATION of EER

- **Class C** : A set of entities ; could be entity type, subclass, superclass, category.
- **Subclass S** : A class whose entities must always be subset of entities in another class, called the superclass C of superclass / subclass (or IS-A) relationship S/C :

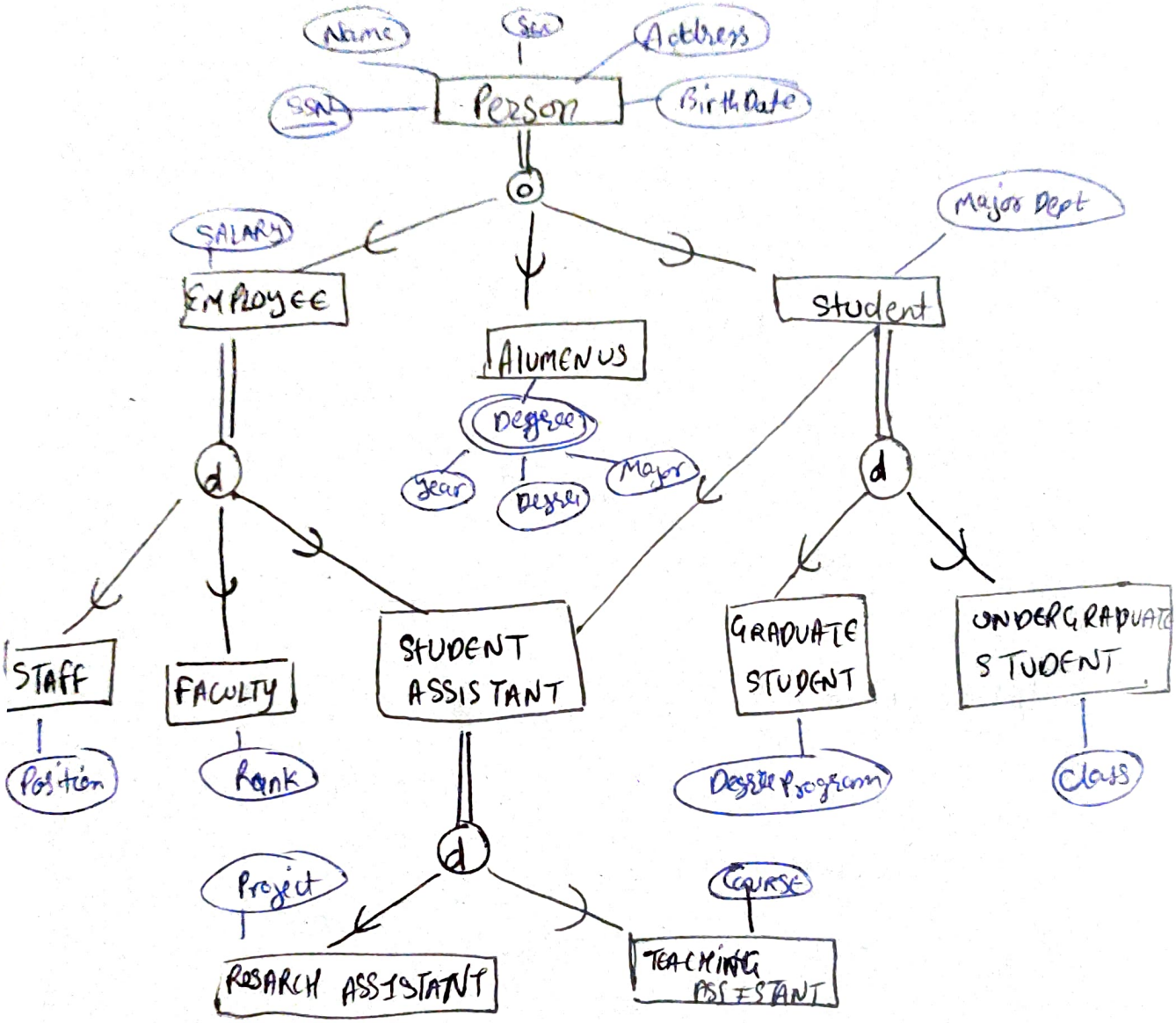
  $S \subseteq C$.

- **Specialization Z** : $Z = \{S_1, S_2 - - S_n\}$ a set of subclasses with same superclass G, hence G/S_i is superclass relationship for $i = 1, 2 - - n$
  - G is called generalization of subclasses $\{S_1, S_2 - - S_n\}$
  - Z is total if we always have :
    $S_1 \cup S_2 \cup - - S_n = G$ otherwise Z is partial
  - Z is disjoint if we always have :
    $S_i \cap S_j$ empty set for $i \neq j$ otherwise Z is overlapping

With Specialization / Generalization concept

UNIVERSITY

O - overlapping



difference betn o & d :-

- The disjoint rule states that an entity instance of a supertype can only be a member of one subtype. The overlap rule states that an entity instance of supertype can be member of multiple subtype.