



Chapter three

Conceptual Database Design

The Entity Relationship (E-R) Model

Cont...

- Entity-Relationship modeling is used to represent conceptual view of the database

The main components of ER Modeling are:

- **Entities**
 - Corresponds to entire table, not row
 - Represented by Rectangle
- **Attributes**
 - Represents the property used to describe an entity or a relationship
 - Represented by Oval
- **Relationships**
 - Represents the association that exist between entities
 - Represented by Diamond
- **Constraints**
 - Represent the constraint in the data



Cont...

- Before working on the conceptual design of the database, one has to know and answer the following basic questions.
 1. What are the entities and relationships in the enterprise?
 2. What information about these entities and relationships should we store in the database?



Cont...

3. What are the integrity constraints that hold? Constraints on each data with respect to update, retrieval and store.
4. Represent this information pictorially in ER diagrams, then map ER diagram into a relational schema.

Developing an E-R Diagram

- Designing conceptual model for the database is not a one linear process but an iterative activity where the design is refined again and again.
- To identify the **entities**, **attributes**, **relationships**, and **constraints** on the data, there are different set of methods used during the analysis phase.
- These include information gathered by...
 - Interviewing end users individually and in a group
 - Questionnaire survey
 - Direct observation
 - Examining different documents

Cont...

- The basic E-R model is graphically depicted and presented for review.
- The process is repeated until the end users and designers agree that the E-R diagram is a fair representation of the organization's activities and functions.
- Checking for Redundant Relationships in the ER Diagram.
 - Relationships between entities indicate access from one entity to another
- The last phase in ER modeling is validating an ER Model against requirement of the user

Graphical Representations in ER Diagramming

- Entity is represented by a RECTANGLE containing the name of the entity.

Strong Entity

Weak Entity

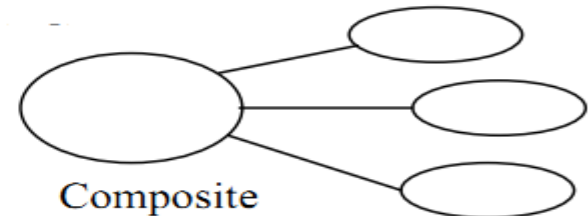
- Connected entities are called relationship participants
- Attributes are represented by OVALS and are connected to the entity by a line.



Attribute



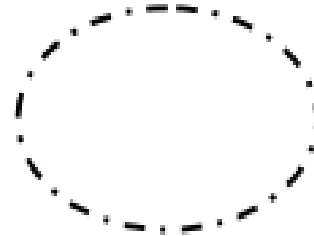
Multi-valued
Attribute



Composite
Attribute

- A derived attribute is indicated by a DOTTED LINE.

(.....)

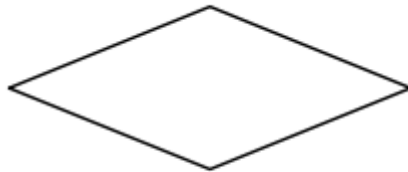


- PRIMARY KEYS are underlined.

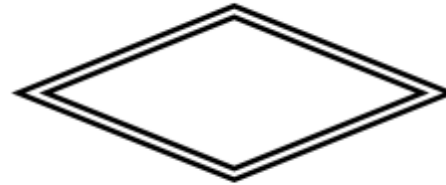


Cont...

- Relationships are represented by DIAMOND shaped symbols



Relationship

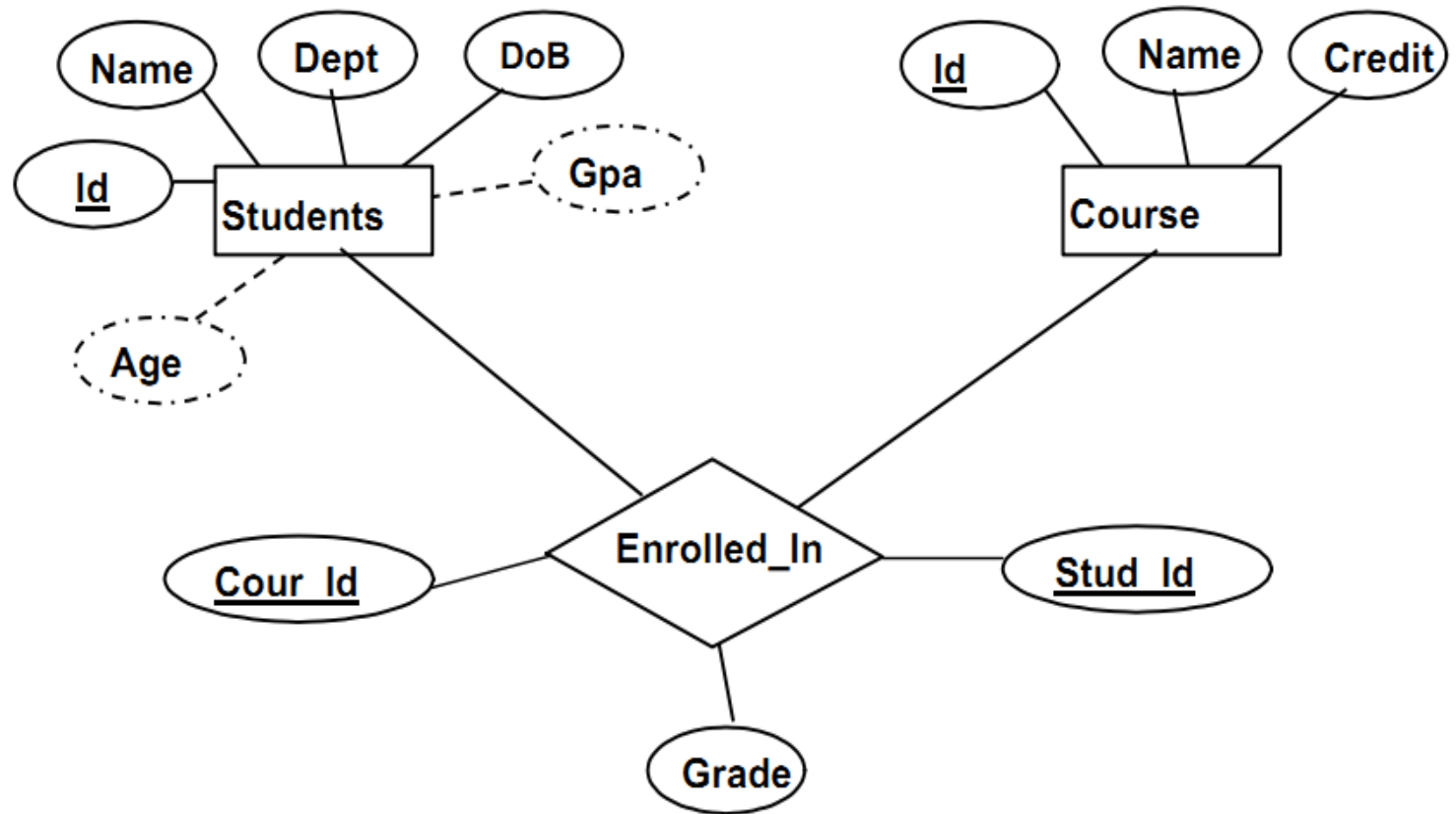


Weak Relationship

Example I

- Build an ER Diagram for the following information:
- Students
 - Have an Id, Name, Dept, Age, Gpa
- Courses
 - Have an Id, Name, Credit Hours
- Students enroll in courses and receive a grade

Cont...



Entity versus Attributes

- Consider designing a database of employees for an organization:
- Should **address** be an attribute of Employees or an entity (connected to Employees by a relationship)?
 - If we have several addresses per employee, address must be an entity (attributes cannot be multi valued)

Cont...

- If the structure (city, Woreda, Kebele, etc) is important, e.g. want to retrieve employees in a given city, address must be modeled as an **entity** (attribute values are atomic)
- Cardinality on Relationship expresses the number of entity occurrences/tuples associated with one occurrence/tuple of related entity.

Cont...

- Existence Dependency: the dependence of an entity on the existence of one or more entities.
- **Weak entity** : an entity that can not exist without the entity with which it has a relationship – it is indicated by a double rectangle
- Participating entity in a relationship is either optional or mandatory.

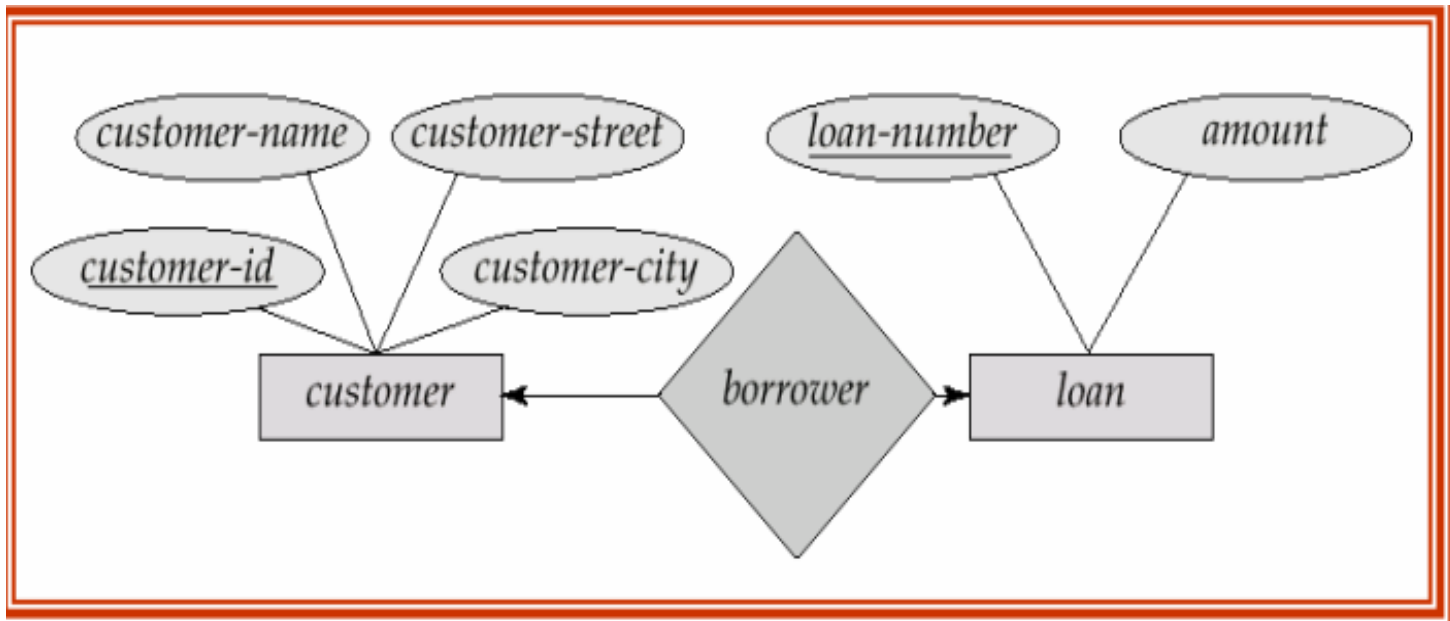


Structural Constraints on Relationship

- Constraints on Relationship / Multiplicity/ Cardinality Constraints
 - **Multiplicity constraint** is the number of or range of possible occurrence of an entity type/relation that may relate to a single occurrence/tuple of an entity type/relation through a particular relationship.

One-to-one relationship:

- A customer is associated with at most one loan via the relationship borrower
- A loan is associated with at most one customer via borrower

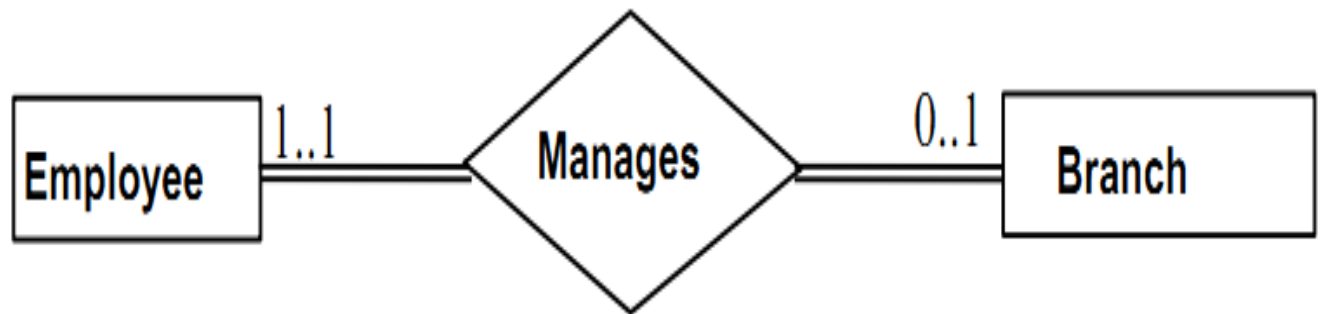


Cont...

E.g.: Relationship *Manages* between *STAFF* and *BRANCH*

The multiplicity of the relationship is:

- *One branch can only have one manager*
- *One employee could manage either one or no branches*

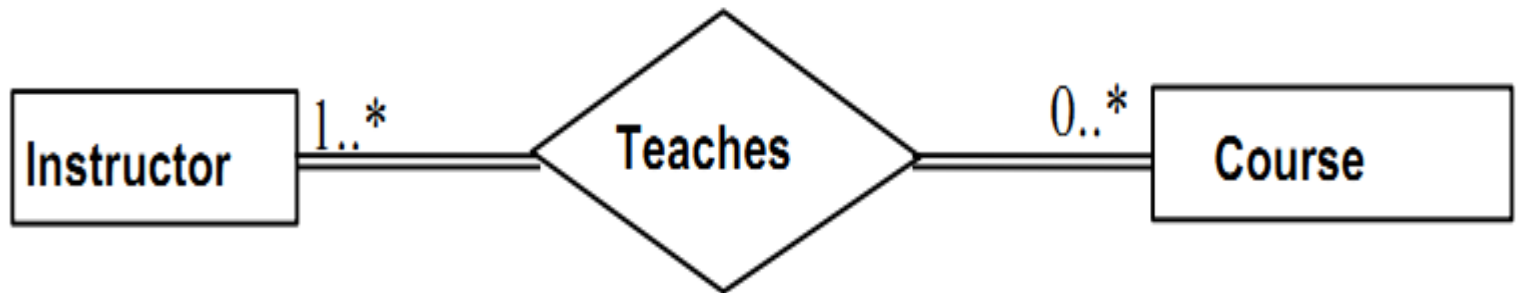




One-To-Many Relationships

Example:- Relationship Teaches between **INSTRUCTOR** and **COURSE** the multiplicity of the relationship

- One **Instructor** Teaches one or more course(s)
- One **Course** Thought by Zero or more **Instructor(s)**





Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set.
- The entity with total participation will be connected with the relationship using a double line.
- Example:- Participation of **EMPLOYEE** in “belongs to” relationship with **DEPARTMENT** is total since every employee should belong to a department.

Cont...



- Example 2:- Participation of **EMPLOYEE** in “manages” relationship with **DEPARTMENT**, DEPARTMENT will have total participation but not EMPLOYEE

Cont...

- **Partial participation:-** Some entities may not participate in any relationship in the relationship set
- Example Participation of EMPLOYEE in “manages” relationship with DEPARTMENT, EMPLOYEE will have partial participation since not all employees are managers.





Problem in ER Modeling

- The Entity-Relationship Model is a conceptual data model that views the real world as consisting of entities and relationships.
- The model visually represents these concepts by the Entity-Relationship diagram.
- The basic constructs of the ER model are **entities, relationships, and attributes**.
- Entities are concepts, real or abstract, about which information is collected.
- Relationships are associations between the entities.
- Attributes are properties which describe the entities.

- 
- While designing the ER model one could face a problem on the design which is called a **connection traps**.
 - Connection traps are problems arising from misinterpreting certain relationships
 - There are two types of connection traps
 - Fan trap:
 - Chasm Trap:

Fan trap:

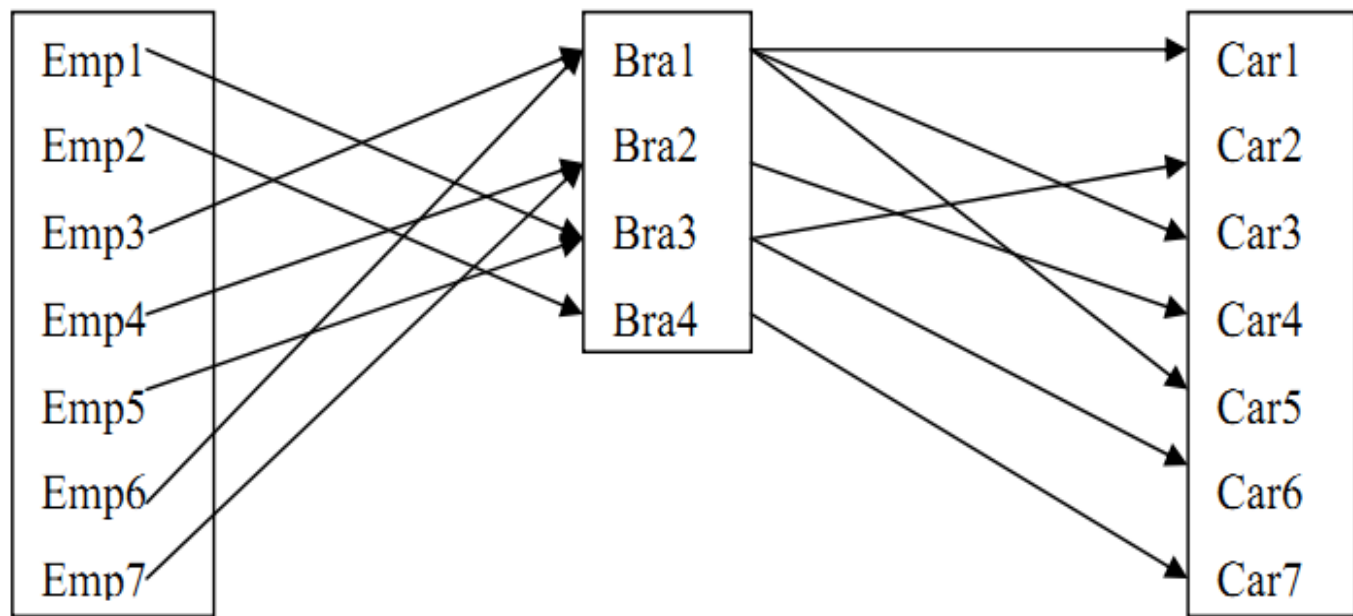
- Occurs where a model represents a relationship between entity types, but the **pathway** between certain entity occurrences is ambiguous.
- May exist where two or more **one-to-many** (1:M) relationships fan out from an entity.
- The problem could be avoided by restructuring the model so that there would be no 1:M relationships fanning out from a single entity and all the semantics of the relationship is preserved.

Cont...

Example:



Semantics description of the problem;



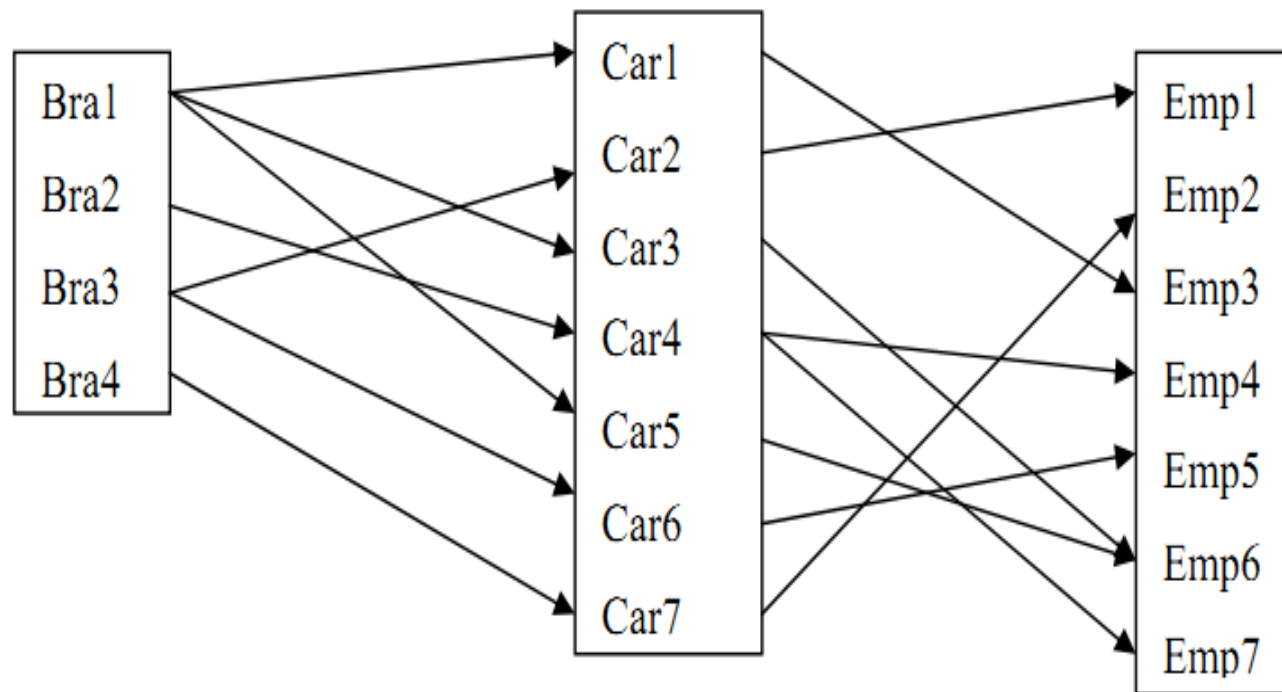
Cont...

- **Problem:** Which car (Car1 or Car3 or Car5) is used by Employee 6 Emp6 working in Branch 1 (Bra1)?
- Thus from this ER Model one can not tell which car is used by which staff since a branch can have more than one car and also a branch is populated by more than one employee.
- Thus we need to restructure the model to avoid the connection trap.
- To avoid the Fan Trap problem we can go for restructuring of the E-R Model.
- This will result in the following E-R Model.

Cont...



Semantics description of the problem;





Chasm Trap:

- Occurs where a model suggests the existence of a relationship between entity types, but the **path way does not exist** between certain entity occurrences.
- May exist when there are one or more relationships with a minimum multiplicity on cardinality of **zero forming** part of the pathway between related entities.

Cont...



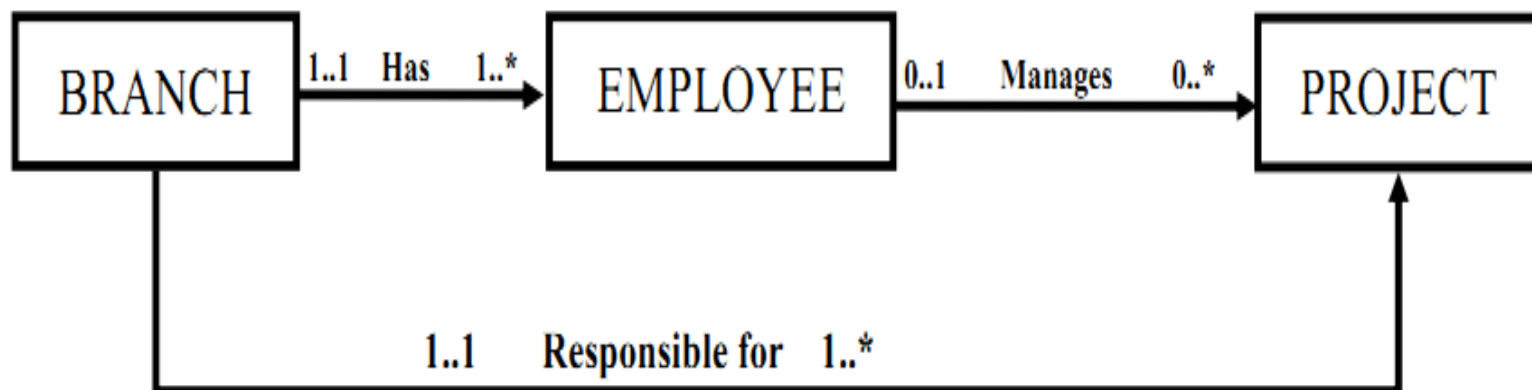
- If we have a set of projects that are not active currently then we can not assign a project manager for these projects.
- So there are project with no project manager making the participation to have a minimum value of zero.

Cont...

Problem:

- How can we identify which BRANCH is responsible for which PROJECT?
- We know that whether the PROJECT is active or not there is a responsible BRANCH.
- But which branch is a question to be answered, and since we have a minimum participation of zero between employee and PROJECT we can't identify the BRANCH responsible for each PROJECT.

The solution for this Chasm Trap problem is to add another relation ship between the extreme entities (BRANCH and PROJECT)



Enhanced E-R (EER) Models

- Object-oriented extensions to E-R model
- EER is important when we have a relationship between two **entities** and the **participation** is **partial** between entity occurrences.
- In such cases EER is used to reduce the complexity in participation and relationship complexity.
- **ER** diagrams consider entity types to be primitive objects
- **EER** diagrams allow refinements within the structures of entity types



Cont...

EER Concepts

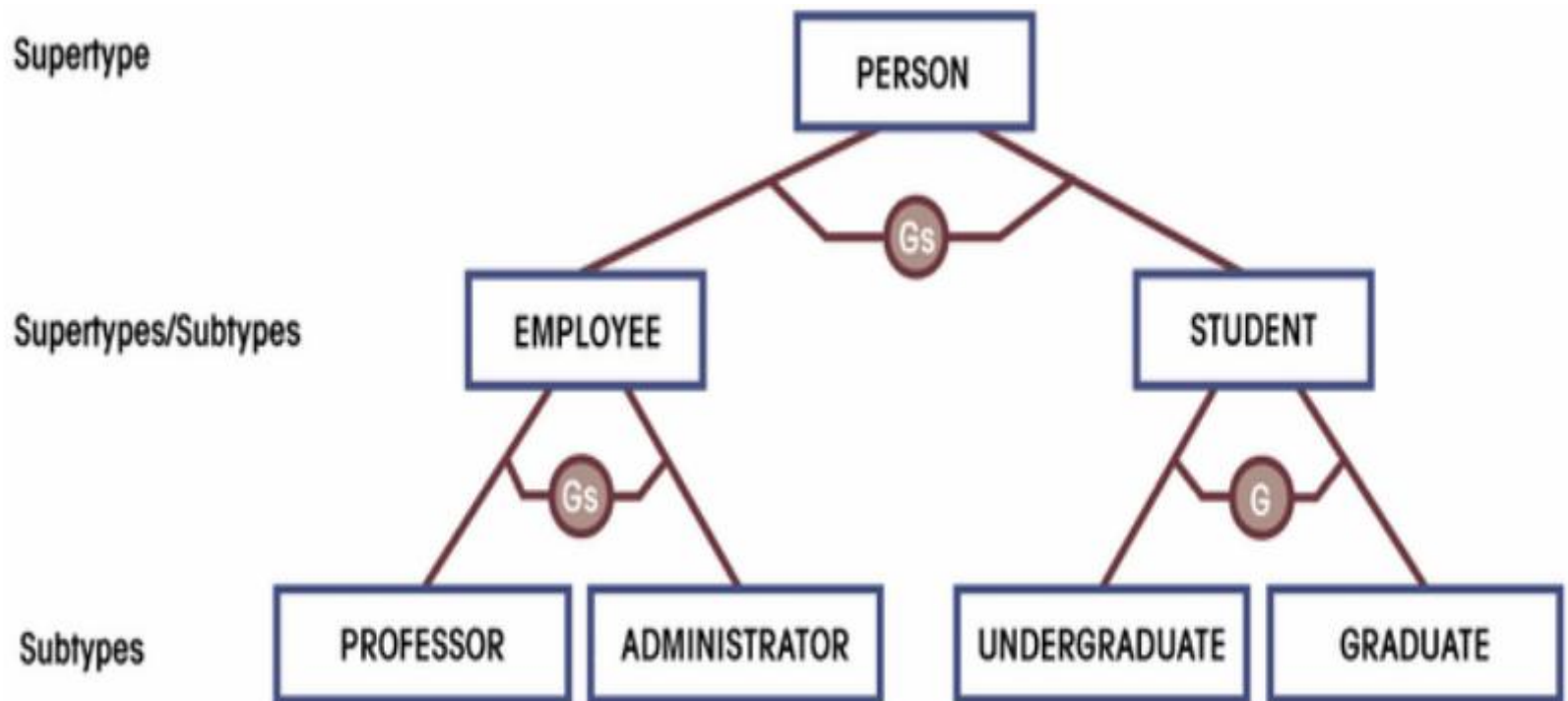
- **Generalization**
- **Specialization**
- **Sub classes**
- **Super classes**
- **Attribute Inheritance**
- **Constraints on specialization and generalization**

Generalization

- Generalization occurs when **two** or **more** entities represent categories of the same real-world object.
- Generalization is the process of defining a more **general entity type** from a set of more specialized entity types.
- A generalization hierarchy is a form of abstraction that specifies that two or more entities that share common attributes can be generalized into a higher level entity type.
- Is considered as **bottom-up** definition of entities.
- Generalization hierarchy depicts relationship between **higher level superclass** and **lower level subclass**

Cont...

- Example: Account is a generalized form for Saving and Current Accounts



Specialization

- Is the result of subset of a **higher level** entity set to form a **lower level entity set**.
- The specialized entities will have additional set of attributes (distinguishing characteristics) that distinguish them from the **generalized entity**.
- Is considered as **Top-Down** definition of entities.
- **Specialization** process is the inverse of the **Generalization** process



Cont...

- Reasons for Specialization
 - Attributes only partially applying to super classes
 - Relationship types only partially applicable to the superclass
- In many cases, an entity type has numerous sub-groupings of its entities that are meaningful and need to be represented explicitly.



Cont...

- Example: Saving Accounts and Current Accounts are Specialized entities for the generalized entity Accounts.
- **Manager, Sales, Secretary:** are specialized **employees**.

Subclass/Subtype

- An **entity type** whose **tuples** have attributes that distinguish its members from tuples of the **generalized or Superclass entities**.
- When one generalized Superclass has various subgroups with distinguishing features and these subgroups are represented by specialized form, the groups are called subclasses.
- **Subclasses** can be either mutually exclusive (disjoint) or overlapping (inclusive).
- A single **subclass** may inherit attributes from two distinct **superclasses**.

Cont...

- A mutually exclusive category/subclass is when an entity instance can be in only one of the subclasses.
- E.g.:An EMPLOYEE can either be SALARIED or PART-TIMER but not both.
- An overlapping category/subclass is when an entity instance may be in two or more subclasses.
- E.g.:A PERSON who works for a university can be both **EMPLOYEE** and a **STUDENT** at the same time.



Superclass /Supertype

- An entity type whose tuples share common attributes.
- Attributes that are shared by all entity occurrences (including the identifier) are associated with the supertype.
- Is the generalized entity



Relationship Between Superclass and Subclass

- The relationship between a superclass and any of its subclasses is called a superclass/subclass or class/subclass relationship
- An instance can not only be a member of a **subclass**. i.e. Every instance of a subclass is also an instance in the Superclass.
- A member of a subclass is represented as a distinct database object, a distinct record that is related via the key attribute to its super-class entity.

Cont...

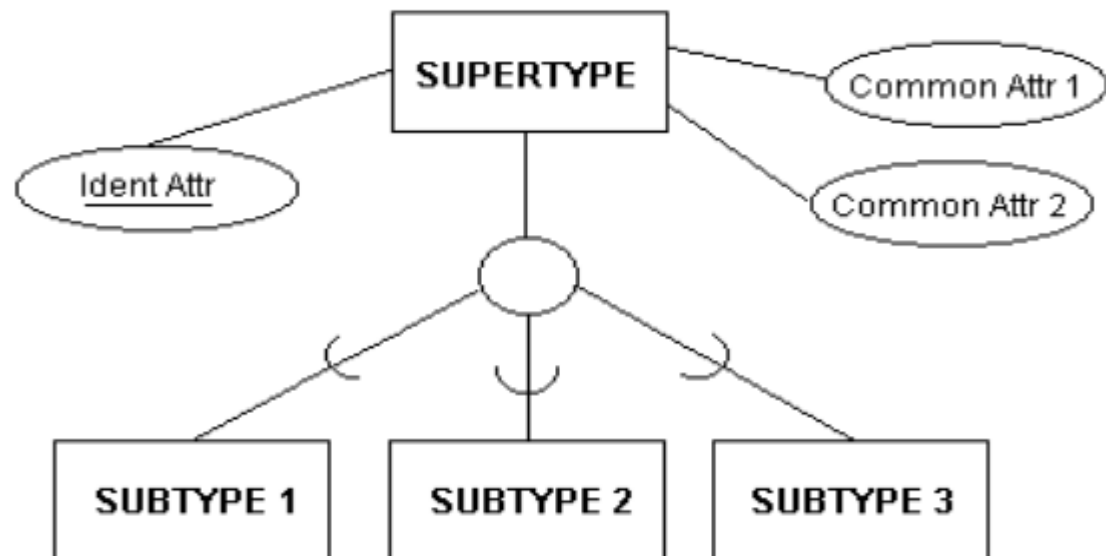
- An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the super-class.
- An entity occurrence of a sub class not necessarily should belong to any of the subclasses unless there is full participation in the specialization.
- A member of a subclass is represented as a distinct database object, a distinct record that is related via the key attribute to its super-class entity.

Cont...

- The relationship between a subclass and a Superclass is an “IS A” or “IS PART OF” type.
-  Subclass IS PART OF Superclass
-  Manager IS AN Employee
- All subclasses or specialized entity sets should be connected with the superclass using a **line to a circle** where there is a subset symbol indicating the direction of subclass/superclass relationship.

Cont...

- We can also have subclasses of a subclass forming a hierarchy of specialization.
- Superclass attributes are shared by all subclasses of that superclass
- Subclass attributes are unique for the subclass.

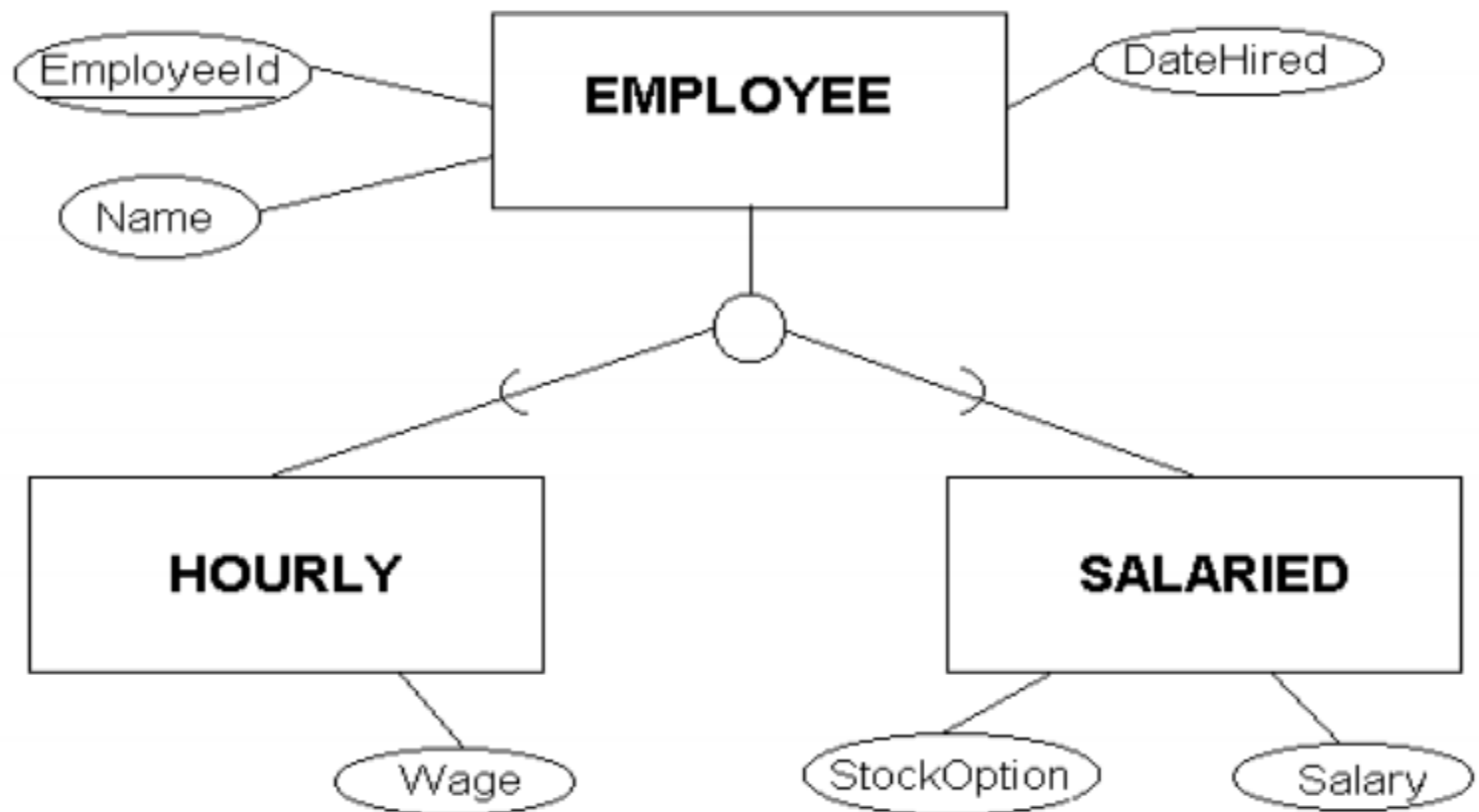




Attribute Inheritance

- An entity that is a member of a subclass inherits all the attributes of the entity as a member of the superclass.
- The entity also inherits all the relationships in which the superclass participates.
- An entity may have more than one subclass categories.
- All entities/subclasses of a generalized entity or superclass share a common unique identifier attribute (primary key). i.e.
- The primary key of the superclass and subclasses are always identical.

Cont...



Cont...

- Consider the EMPLOYEE supertype entity shown above.
- This entity can have several different subtype entities (for example: HOURLY and SALARIED), each with distinct properties not shared by other subtypes.
- But whether the employee is HOURLY or SALARIED, same attributes (EmployeeId, Name, and DateHired) are shared.
- The Supertype EMPLOYEE stores all properties that subclasses have in common.
- And HOURLY employees have the unique attribute Wage (hourly wage rate), while SALARIED employees have two unique attributes, StockOption and Salary.

Constraints on specialization and generalization

Completeness Constraint

- The Completeness Constraint addresses the issue of whether or not an occurrence of a Superclass must also have a corresponding Subclass occurrence.
- The completeness constraint requires that **all instances** of the subtype be represented in the supertype.
- The **Total Specialization** Rule specifies that an entity occurrence should at least be a member of one of the subclasses.
- Total Participation of superclass instances on subclasses is diagrammed with a double line from the Supertype to the circle as shown below.

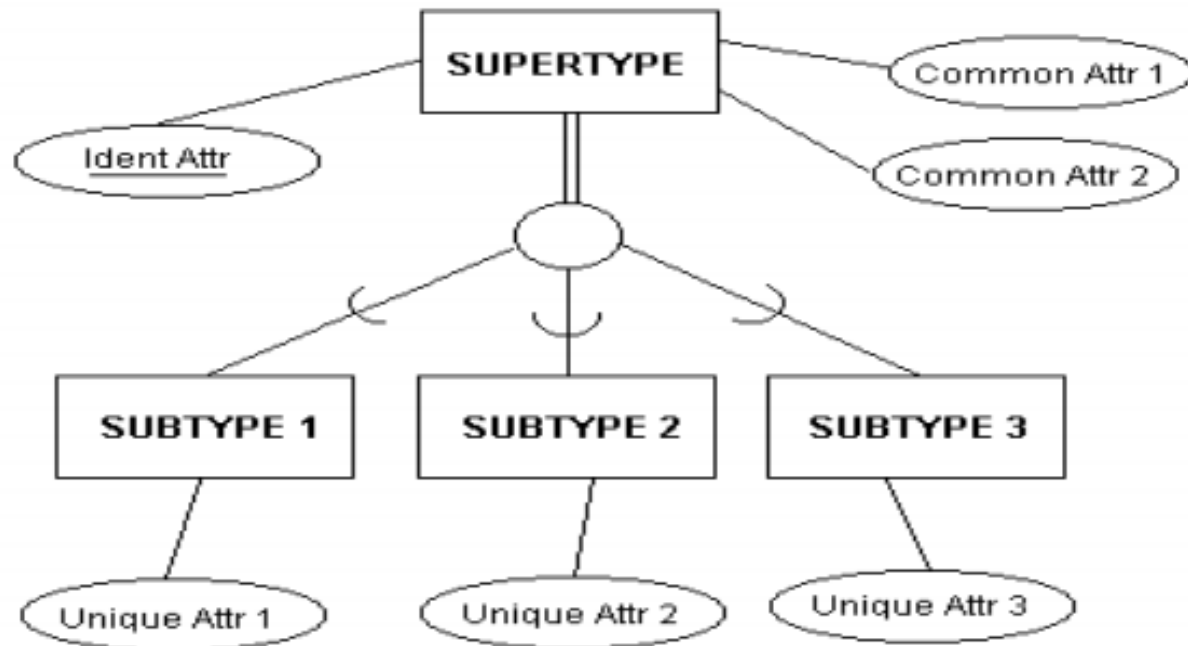


Cont...

Example

- If we have **EXTENTION** and **REGULAR** as subclasses of a superclass **STUDENT**, then it is mandatory that each student to be either **EXTENTION** or **REGULAR** student.
- Thus the participation of instances of **STUDENT** in **EXTENTION** and **REGULAR** subclasses will be total.

Total Specialization Rule



Cont...

- The **Partial Specialization Rule** specifies that it is not necessary for all entity occurrences in the superclass to be a member of one of the subclasses.
- Here we have an optional participation on the specialization.
- Partial Participation of superclass instances on subclasses is diagrammed with a single line from the Supertype to the circle.

Cont...

- If we have **MANAGER** and **SECRETARY** as subclasses of a superclass **EMPLOYEE**, then it is not the case that all employees are either manager or secretary.
- Thus the participation of instances of employee in **MANAGER** and **SECRETARY** subclasses will be partial.

Disjointness Constraints.

- Specifies the rule whether one entity occurrence can be a member of more than one subclasses. i.e. it is a type of business rule that deals with the situation where an entity occurrence of a Superclass may also have more than one Subclass occurrence.
- The **Disjoint Rule restricts** one entity occurrence of a superclass to be a member of **only one of the subclasses**.
- Example: a EMPLOYEE can either be SALARIED or PART-TIMER, but not the both at the same time.
- The Overlap Rule allows one entity occurrence to be a member of more than one subclass.
- Example: EMPLOYEE working at the university can be both a STUDENT and an EMPLOYEE at the same time.

Cont...

- The two types of constraints on generalization and specialization (Disjointness and Completeness constraints) are not dependent on one another.
- That is, being disjoint will not favour whether the tuples in the superclass should have Total or Partial participation for that specific specialization.

Cont...

- This is diagrammed by placing either the letter "d" for disjoint or "o" for overlapping inside the circle on the Generalization Hierarchy portion of the E-R diagram.
- From the two types of constraints we can have four possible constraints
 -  Disjoint AND Total
 -  Disjoint AND Partial
 -  Overlapping AND Total
 -  Overlapping AND Partial