



Database Application Development



Information System

- Information system
 - Collection
 - Management
 - Control, and
 - Dissemination of information throughout an organization.
- Database is fundamental component of IS
- Database development and usage should be viewed from perspective of the wider requirements of the organization.

System Development Life cycle

- The core aspect of software engineering is the **subdivision** of the **development process** into a series of phases, or steps
- Each phase focuses on one aspect of the development
- Each phase in the life cycle can be checked for correctness before moving on to the next phase
- The collection of these steps is referred to as the software development life cycle (SDLC).
- Why CYCLE?
- The software product moves through a life cycle
 - repeatedly as it is refined or redeveloped
 - until it is finally retired from use.



System Development – Phases 1

- Requirements
- Analysis
- Design
- Implementation
- Testing
- Maintenance

System Development – Phases 2

■ Requirements

- ☐ consultation with stakeholders about what they want from a system
- ☐ Produces a statement of requirements.

■ Analysis

- ☐ starts with the statement of requirements
- ☐ produces a system specification.
- ☐ The specification is a formal representation of what a system should do
- ☐ Is independent of how it may be realized.

■ Design

- ☐ begins with a system specification,
- ☐ provides a detailed description of how a system should be constructed.



System Development - Phases 3

■ Implementation

- ☐ the construction of a computer system according to a given design document
- ☐ Coding, Database construction, ...

■ Testing

- ☐ compares the implemented system against the design documents and requirements
- ☐ produces an acceptance report

■ Maintenance

- ☐ involves dealing with changes in the requirements or the implementation environment
- ☐ bug fixing

DB System Development 1

■ Requirements Gathering

- Understand the proposed system to obtain and document the **data** and **functional requirements**.
- The result of this step is a document that includes the detailed requirements provided by the users.
- It will include
 - what persistent data that users want to store
 - the description of the meaning and interpretation of the data elements
- The data description should answer
 - what are the data items
 - what are the properties of the data items
 - what relationships hold between the data items.
 - what constraints apply
- Different Methods are used
 - Interviewing end users individually and in a group
 - Questionnaire survey
 - Direct observation

DB System Development 2

■ Analysis

- produces a conceptual data model.
- a detailed description of the data that will suit user requirements
- The conceptual data model provides a shared, formal representation of what is being communicated between clients and developers
- Conceptual data model is concerned with the meaning and structure of data
- The conceptual data model is a formal representation of what data a database should contain and the constraints the data must satisfy.
- Expressed in terms that are independent of how the model may be implemented.
 - Focuses on the questions, “What is required?” not “How is it achieved?”

DB System Development 3

■ Logical Design

- Database design starts with a conceptual data model and produces a specification of a logical schema
- The output of this stage is a detailed **relational specification**
 - the logical schema, of all the tables and constraints needed to satisfy the description of the data in the conceptual data model.
 - which tables are most appropriate for representing the data in a database?

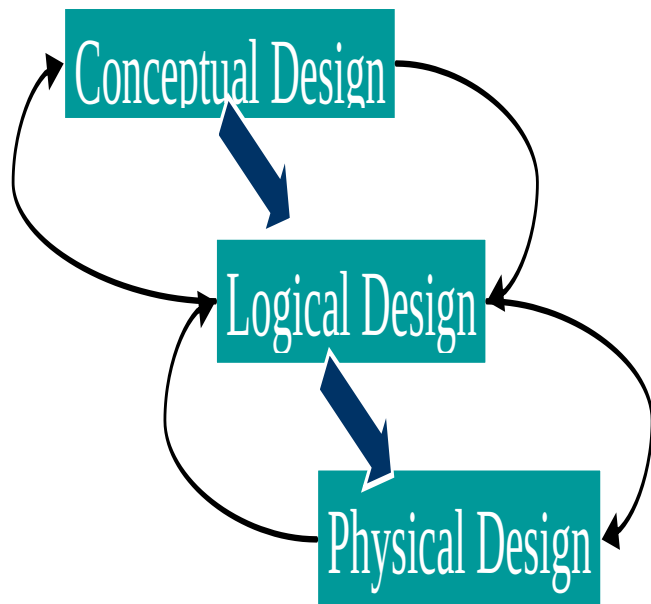
■ Implementation

- the construction of a database according to the specification of a logical schema.
- Is dependent on the choice of available DBMSs, database tools and operating environment.

Database Design Methodology

■ Three main phases

- Conceptual database design
- Logical database design
- Physical database design





Phases in Database Design

■ Conceptual Database Design

- The process of constructing a model of the data used in an enterprise, independent of all physical considerations.
- Discovering and analyzing organizational and user data requirements

■ Logical Database Design

- The process of constructing a model of the data used in an enterprise
- It is independent of all physical considerations.

■ Physical Database Design

- The process of producing a description of the implementation of the database on secondary storage
- It describes the base relations, file organizations, and indexes design used to achieve efficient access to the data
- Any associated integrity constraints and security measures is

Developing an E-R Diagram 1

- The important activities are to identify
 - Entities
 - Attributes
 - Relationships
 - Constraints
- ER model is developed based on these components
- Conceptual Data Model produces
 - ER diagrams
 - Description about the Diagram



Developing an E-R Diagram 2

1. Designing conceptual model

- ☐ Identify the entities, attributes, relationships, and constraints

- Analysis of requirements gathered
 - ☐ Nouns --- prospective entities
 - ☐ Adjectives --- prospective attributes
 - ☐ Verbs/verb phrases --- prospective relationships
 - ☐ Commanding Words --- prospective constraints

Developing an E-R Diagram 3

2. Develop the E-R Model

- Identify the
 - entities and their attributes
 - Relationships between entities
- Validate an E-R Model against requirement of the user.
- The basic E-R model presented for review.
- Collecting Feedback from the users
- The process is repeated(Iterative)
 - until the end users and designers agree that the E-R diagram is a representation of the organization's activities and functions.



The ER Model

- Components of the ER Model
 - Entity
 - Relationship
 - Attributes

The ER Model 2

- Entity type

- ☐ Group of objects with same properties, identified by enterprise as having an independent existence.
- ☐ Entity type or entity set is a collection of similar entities.

- Some examples of entity types are:

- ☐ All students in HiLCoE, say STUDENT.
- ☐ All courses in HiLCoE, say COURSE

- Entity occurrence

- ☐ Uniquely identifiable object of an entity type.

Entity

- Entities are represented using a rectangle in the ER Model
- Different types of Entities

Physical existence	
Staff	Part
Property	Supplier
Customer	Product
Conceptual existence	
Viewing	Sale
Inspection	Work experience

The ER Model 3

■ Relationship

- A relationship is an association of entities
- The examples of relationship are:
 - Teaches is the relationship type between LECTURER and STUDENT.
 - Treatment is the relationship between DOCTOR and PATIENT.

■ Attributes

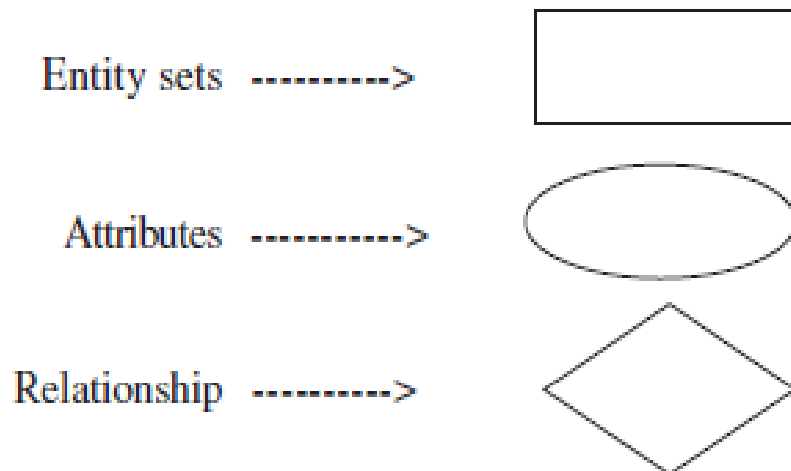
- Attributes are properties of entity types.
- Entities are described in a database by a set of attributes.
- The Building Blocks of an Entity–Relationship Diagram

■ Example of attributes:

- Brand, cost, and weight are the attributes of CELLPHONE.
- Roll number, name, and grade are the attributes of STUDENT.

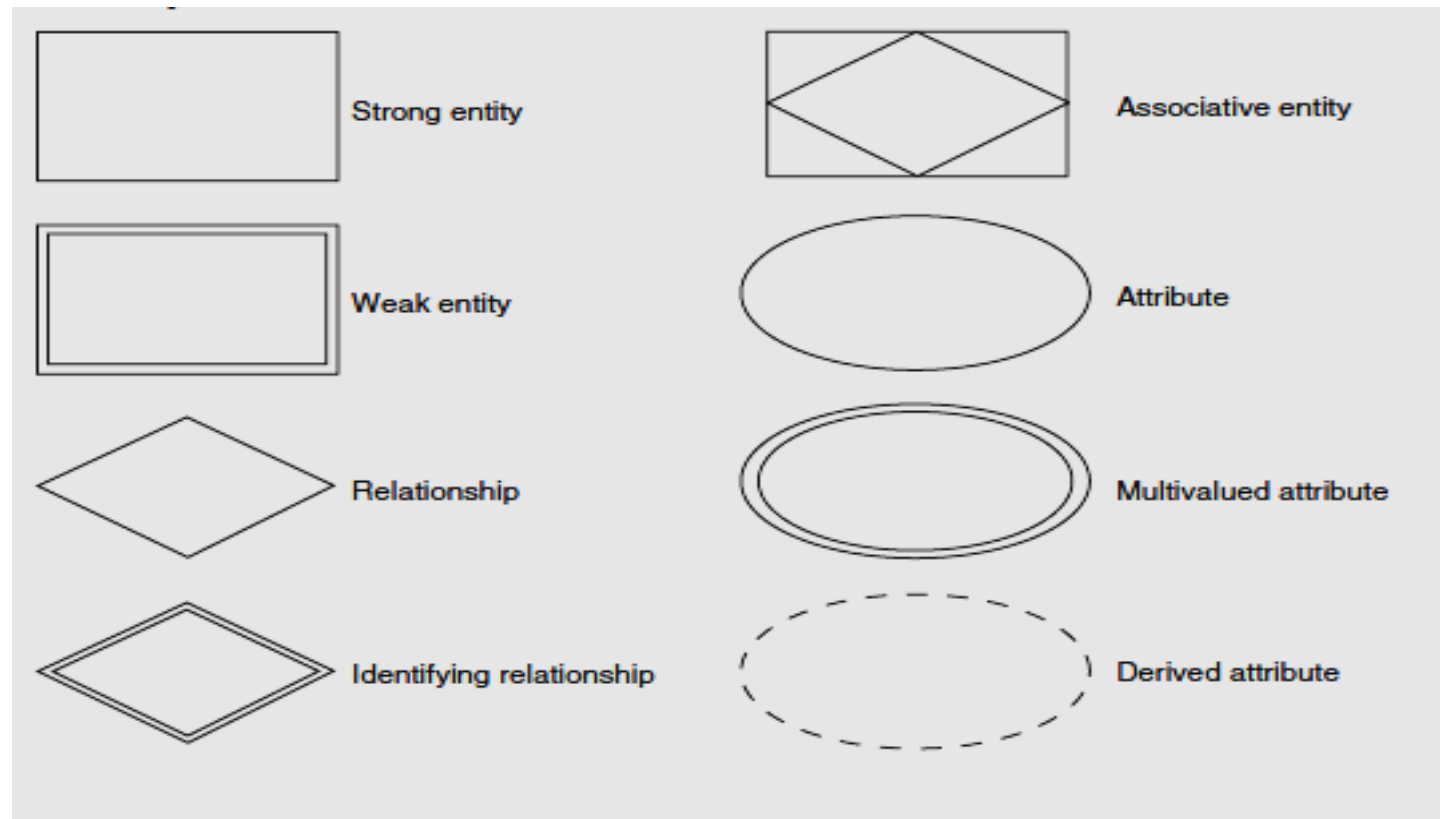
ER Diagram

- The ER diagram is used to represent database schema
- In ER diagram:
 - A rectangle represents an entity set.
 - An ellipse represents an attribute.
 - A diamond represents a relationship.
 - Lines represent linking of attributes to entity sets and of entity sets to relationship sets.



ER Diagram 2

■ Basic Symbols

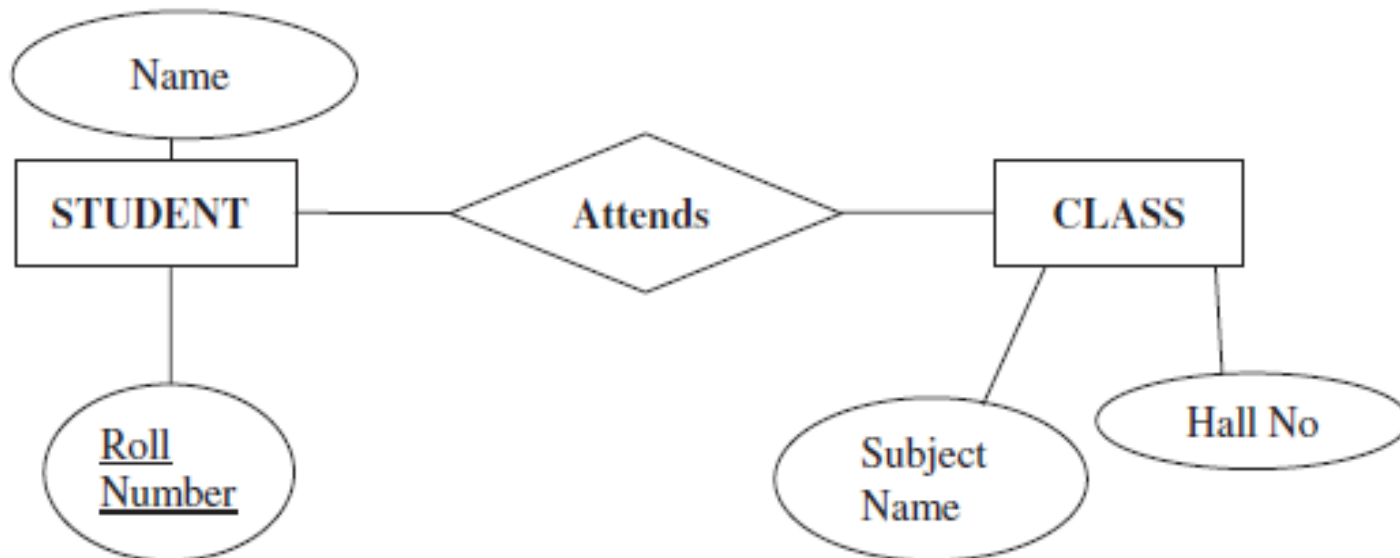


ER Diagram Symbols

- **Rectangles**, which represent entity sets
- **Double rectangles**: represent weak entity sets
- **Ellipses**, which represent attributes
- **Double ellipses**: represent multivalued attributes
- **Dashed ellipses**: denote derived attributes
- **Diamonds**, which represent relationship sets
- **Lines**: link attributes to entity sets and entity sets to relationship sets
- **Double lines**: indicate total participation of an entity in a relationship set

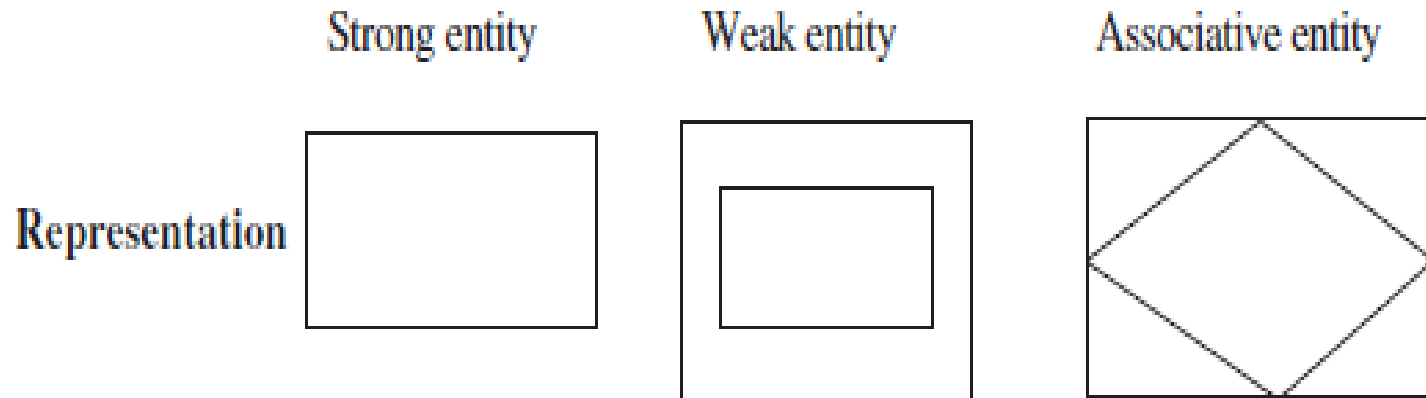
ER Diagram - Example

- STUDENT and CLASS are entities
- **Roll number** and the **Name** are attributes of STUDENT
- **Subject Name** and **Hall Number** are attributes of CLASS
- **Attends** is a relationship between entities STUDENT and CLASS



Classification of Entity Sets

- Entity sets can be broadly classified into:
 - Strong entity.
 - Weak entity.
 - Associative entity.



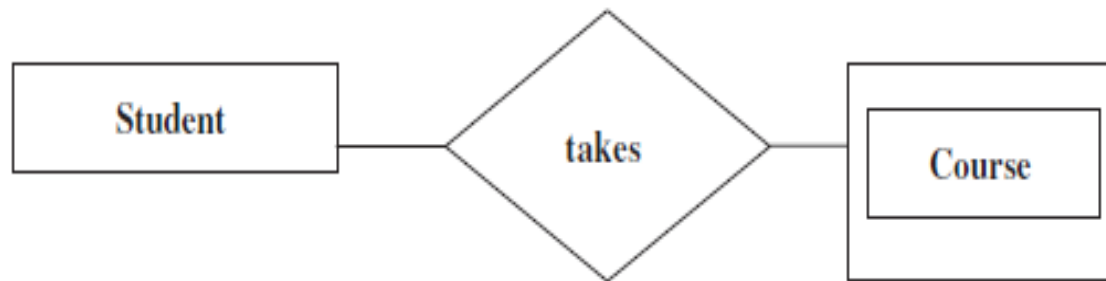
Classification of Entity Sets

■ Strong Entity

- Strong entity is one whose existence does not depend on other entity.

■ Example: Student takes course

- Here student is a strong entity.
- Course is a weak entity because, if there are no students to take a particular course, then that course cannot be offered.
 - The COURSE entity depends on the STUDENT entity



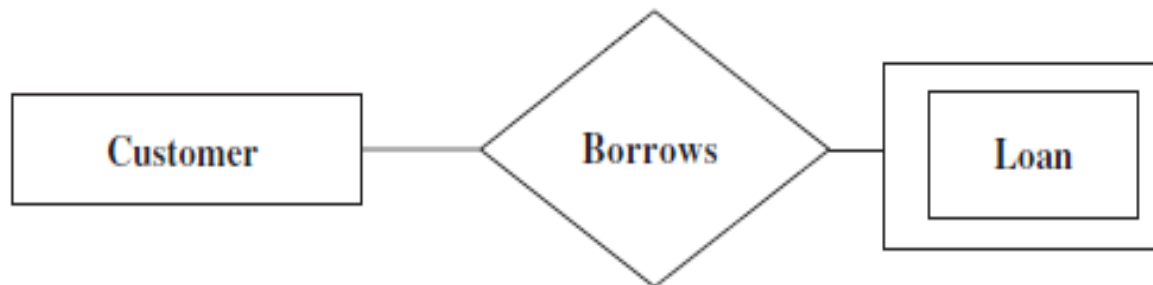
Classification of Entity Sets

■ Weak Entity

- Weak entity is one whose existence depends on other entity. In many cases,
- Weak entity does not have primary key.

■ Example: Customer borrows loan.

- Loan is a weak entity.
- For every loan, there should be at least one customer
- The entity loan depends on the entity customer



Weak Entity and Strong Entities

Strong Entity	Weak Entity
Exists independently of other entities	Dependent on a strong entity Cannot exist on its own
Strong entity has its own unique identifier	Does not have a unique identifier
Represented by a single line rectangle	Represented with a double-line rectangle

Attributes

- An Attribute is a property of an entity or a relationship
- The Domain of an Attribute is the set of allowable values for one or more attributes.
- Attribute (Structure based classification)
 - Simple Attribute
 - Composite Attribute
- Attribute (Value based classification)
 - Single Value
 - Multivalued Attribute
 - Derived Attribute

Attributes (Value based classification)

■ Single-valued Attribute

- Attribute that holds a single value for each occurrence of an entity type.

■ Multi-valued Attribute

- Attribute that holds multiple values for each occurrence of an entity type.
- Represented by double line ellipse in Traditional E-R

■ Derived Attribute

- Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.
- Represented by a dotted line ellipse in Traditional E-R

Single-valued Attribute

■ Examples

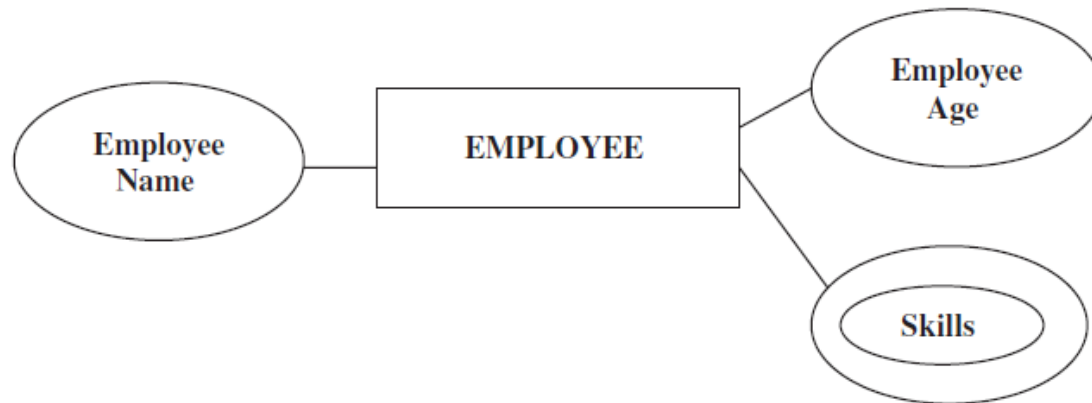
- ☐ age of a person
- ☐ Roll number of the student,
- ☐ Registration number of a car, etc

■ Representation of Single Value Attribute in ER Diagram



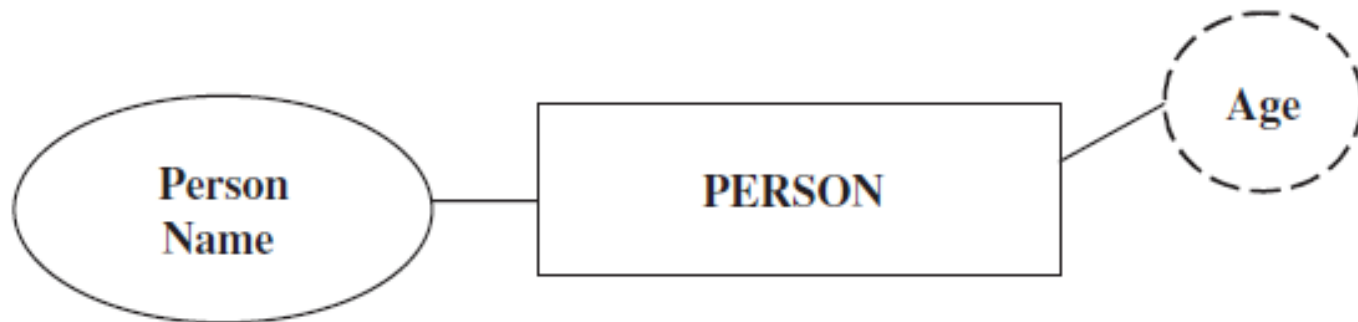
Multivalue Attribute

- More than one value will be associated with a multivalue attribute.
- The skills associated to an employee are a multivalue attribute.



Derived Attribute

- The value of the derived attribute can be derived from the values of other related attributes or entities.
- Example: The Age of a person can be derived from the date of birth of the person
- Experience of an employee in an organization can be derived from date of joining of the employee



Attributes (Structure based classification)

■ Simple Attribute

- Attribute composed of a single component with an independent existence.
- Represented by an Ellipse in traditional E-R model

■ Composite Attribute

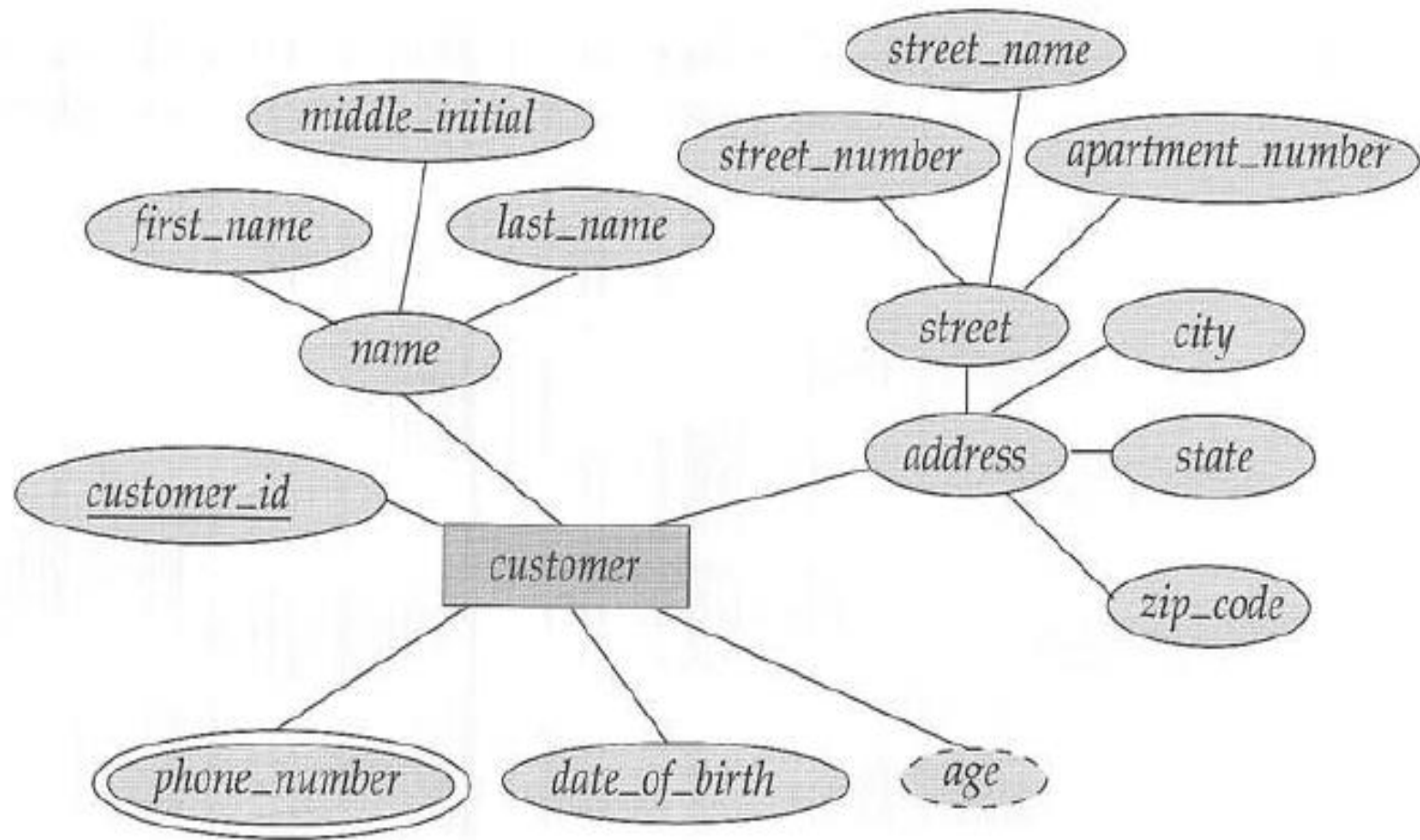
- Attribute composed of multiple components, each with an independent existence.
- Indicated by extension ellipses of an attribute ellipse

Composite Attribute

- Composite attribute is one which can be further subdivided into simple attributes.
- Example: Consider the attribute “address” which can be further subdivided into Street name, City, and State.

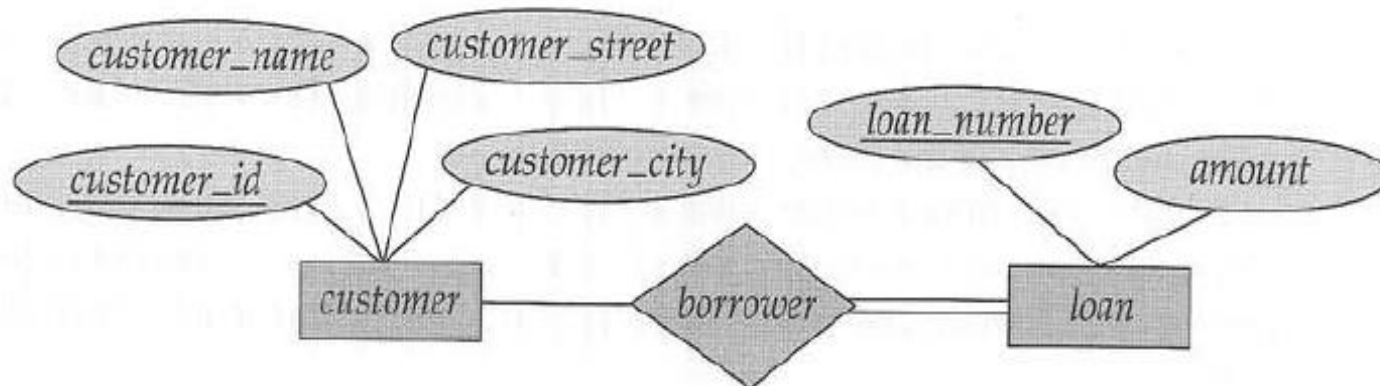


ER Diagram - Example



ER Diagram - Example

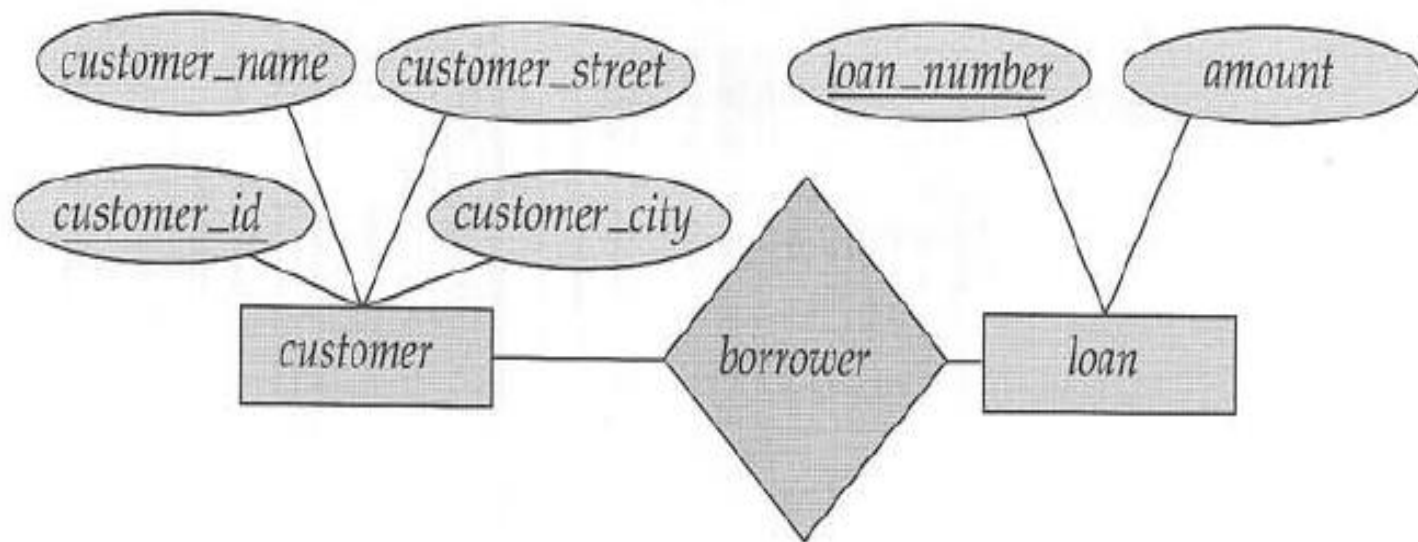
- Double lines are used in an E-R diagram to indicate that the participation of an entity set in a relationship set is total;
- Example: Relationship borrower between customers and loans.
 - A double line from loan to borrower indicates that each loan must have at least one associated customer.



Relationship

■ Relationship

- A set of meaningful associations among entity types.

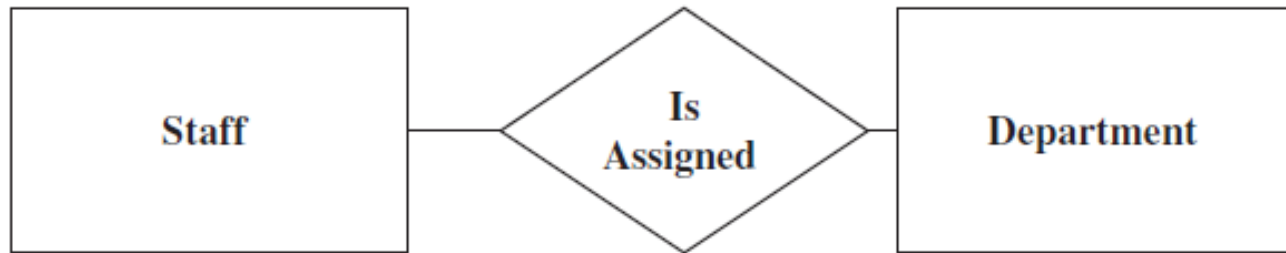


Relationship Degree

- Relationship degree refers to the number of associated entities.
- The relationship degree can be broadly classified into
 - unary,
 - binary, and
 - ternary relationship

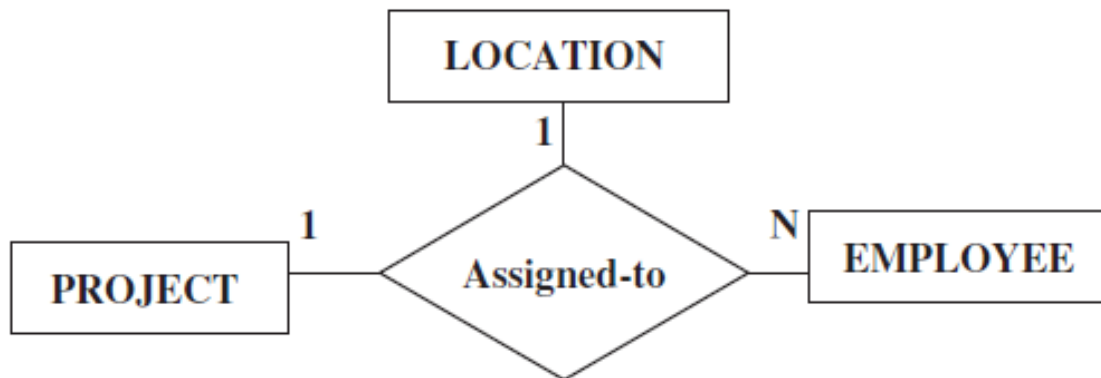
Binary Relationship

- In a binary relationship, two entities are involved.
- Example: STAFF is assigned be to a DEPARTMENT

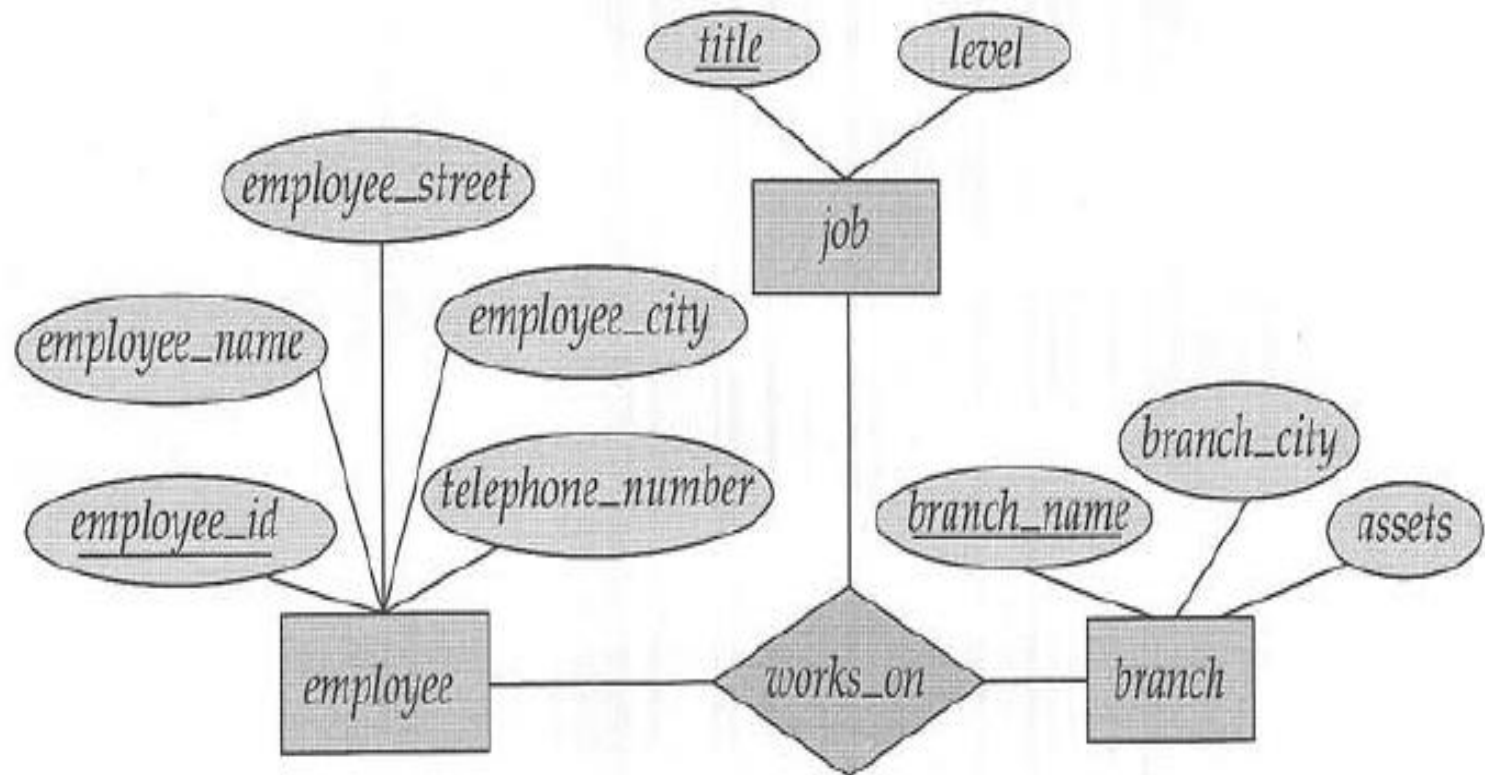


Ternary Relationship

- Three entities are involved
- Example: Employee is assigned to a project and location
 - Here we are considering three entities EMPLOYEE, PROJECT, and LOCATION.

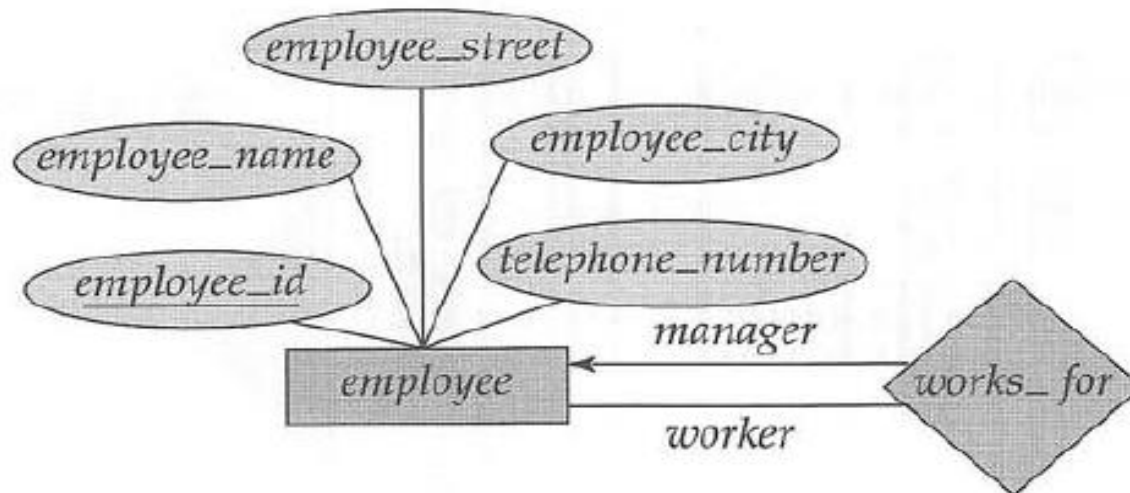


Ternary Relationship - Example



Unary Relationship

- Also known as Recursive relationship.
- In the unary relationship the number of associated entity is one.
 - An entity is related to itself
- Example: an Employee works for another employee (a supervisor)



Cardinality

- E-R diagrams also provide a way to indicate more complex constraints on the number of times each entity participates in relationships
- A relationship set can have an associated minimum and maximum cardinality
 - A minimum value of 1 indicates total participation of the entity set in the relationship set.
 - A maximum value of 1 indicates that the entity participates in at most one relationship,
 - a maximum value * indicates no-limit
- Note that a label 1..x on an edge is equivalent to a double line.

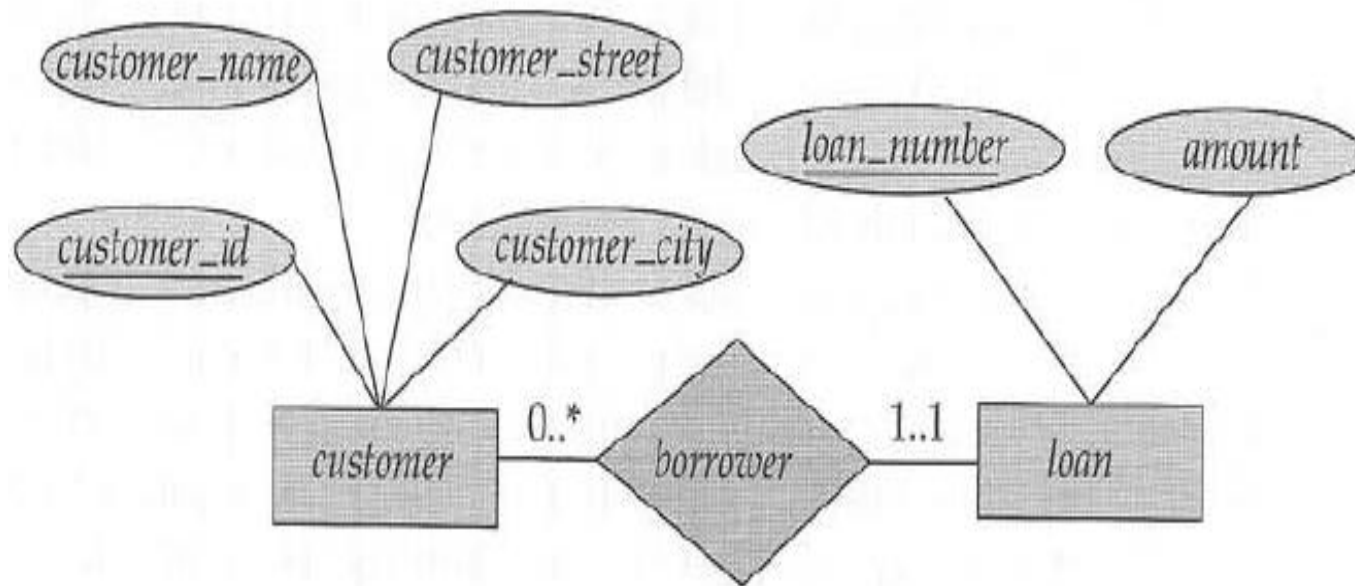
Cardinality (multiplicity constraints)

Alternative ways to represent
multiplicity constraints

Meaning

0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero or three or six, seven, or eight entity occurrences

Cardinality - Example





Relationship Classification

- Relationship is an association among one or more entities.
- This relationship can be broadly classified into:
 - one-to-one relation
 - one-to-many relation
 - many-to-many relation
- One-to-Many Relationship Type
 - associates one entity to more than one entity
- Examples:
 - Country has States: For one country there can be more than one state hence it
 - Parent–child Relationship: A parent can have more than one child.



Relationship Types

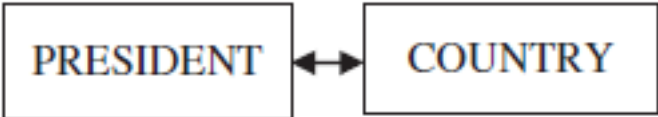
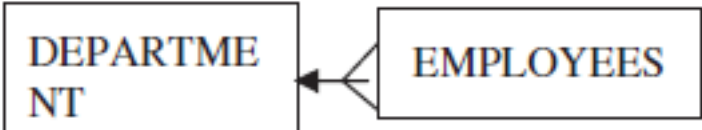
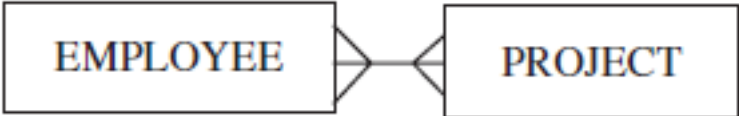
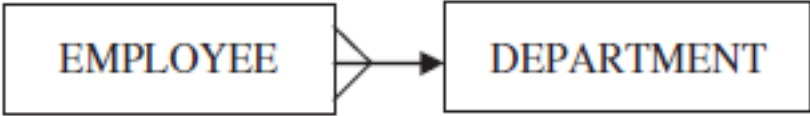
- One-to-One Relationship Type

- A special case of one-to-many relationship
 - Example: The relationship between the President and the country

- Many-to-Many Relationship Type

- Example: Relationship between EMPLOYEE and PROJECT
 - Many employees will be working in many projects

Relationship Types

Relationship type	Representation	Example
One-to-one		
One-to-many		
Many-to-many		
Many-to-one		



Practice Exercise

- Design an E-R diagram for keeping track of your favorite sports competition.
 - You should store the teams, the matches played and the scores in each match.