DATABASE DESIGN

Topics

- Introduction to database design
- Entities, Attributes And Entity Sets
- Relationships And Relationship Sets
- Additional features of the E-R model
- Conceptual design with the E-R model
- Conceptual design for large enterprises

1. DATABASE DESIGN

The database design process can be divided into six steps. The ER model is most relevant to the first three steps. Those six steps are-

i. Requirement Analysis
ii. Conceptual Database Design
iii. Logical Database Design
iv. Schema Refinement
v. Physical Database Design
vi. Application and Security Design

i. Requirements Analysis:

- This is the first step in designing any database application.
- This is an informal process that involves discussions and studies and analyzing the expectations of the users & the intended uses of the database.
- Under this, we have to understand the following.

a) What data is to be stored n a database?
b) What applications must be built?
c) What operations can be used?
d) What the users want from the database.

Example: For customer database, data is customer name, customer city, and customer no.

ii. Conceptual database design:

- The information gathered in the requirements analysis step is used to develop a higher-level description of the data.

- The goal of conceptual database design is a complete understanding of the database structure, meaning (semantics), inter-relationships and constraints. This step is often carried out using the ER model, or a similar high-level data model.

iii. Logical Database Design:

- Under this convert the conceptual database design into a database schema, we must choose a DBMS to implement our database design.

Beyond ER Design

- Once we have logical schema, we must consider performance criteria and design the physical schema. Finally, we must address security issues and ensure that users are able to access the data they need, but not data that we wish to hide from them.

iv. Schema Refinement:

- Under this, we have to analyze the collection of relations (tables) in our relational database schema to identify the potential problems and to refine (clear) it.

v. Physical Database Design:

- Physical database design is the process of choosing specific storage structures and access paths for the database files to achieve good performance for the various database applications.
- This step involves building indexes on some tables and clustering some tables.

vi. Security Design:

- In this step, we must identify different user groups and different roles played by various users.
- For each role, and user group, we must identify the parts of the database that they must be able to access, which are as below.

The ER model is widely used to develop an initial data base design. ER model describes the physical database model; it is basically useful in the design & communication of the logical database model.

2. Entities, Attributes And Entity Sets :

ENTITY

- An entity is an object that is distinguishable from other objects by a set of attributes.
- This is the basic object of E-R Model, which is a 'thing' in the real world with an independent existence.
- An entity may be an 'object' with a physical existence.
- Consists of a collection of objects.
- Entities can be represented by 'Ellipses'.

  Example: Customer, account etc.

ENTITY SET:-

- An entity set is a collection of similar entities (or) an entity set is a group of similar objects.
- **Example:** all customers, all account details

ATTRIBUTES

- Characteristics of an entity are called as an attribute.
- The properties of a particular entity are called as attributes of that specified entity.
- Example:
  - Name, street address, city --- customer database.
  - Acc-no, balance --- account database.

TYPES OF ATTRIBUTES:

These can be classified into following types.

- Simple Attributes.
- Descriptive attribute
- Composite Attributes.
- Single Valued Attributes.
- Mutivalued Attributes.
- Derived Attributes.

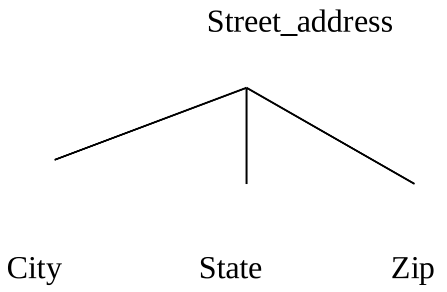
a) Simple Attributes:

- The attributes that are not divisible are called as 'simple or atomic attributes'.
- Example: cust_name, acc_no

b) Composite Attributes:

- The attributes that can be divided into smaller subparts, which represent more basic attributes with independent meaning.
- These are useful to model situations in which a user sometimes refers to the composite attribute as unit but at other times refers specifically to its components.

Example: Street_address can be divided into 3 simple attributes as Number, Street and Apartment_no.



c)Single Valued Attribute:

- The attributes having a single value for a particular entity are called as 'Single Valued Attributes'.
- Example: 'adaar number ' is a single valued attribute of 'Person'.

d) Muti Valued Attribute:

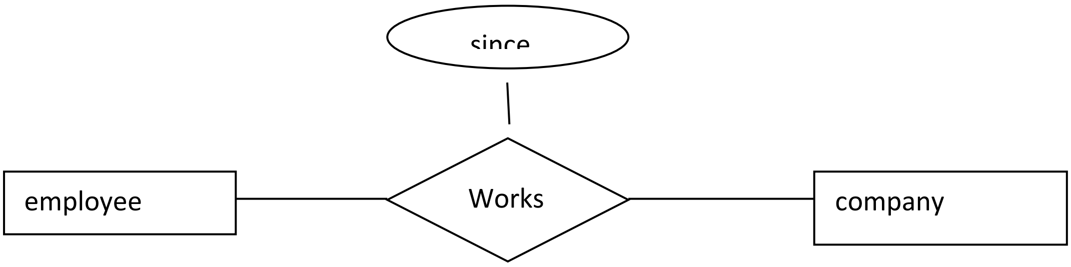
- The attributes, which are having a set of values for the same entity, are called as 'Multi Valued Attributes'.
- Example: A 'College Degree' attribute for a person.i.e, one person may not have a college degree, another person may have one and a third person may have 2 or more degrees.

e) Derived Attributes:

- An attribute which is derived from another attribute is called as a ‘derived attribute.
- Example: ‘Age’ attribute is derived from another attribute ‘Date’.

f) Descriptive attribute:

- Descriptive attributes are used to record information about relationship, rather than about any one of the participating entities.
- **Examples: since** is an attribute in works in relationship capture the information about participating entities employee & departments.



h) Null Attribute:-

- It is an attribute that uses a null value when an entity does not have a value for an attribute.

Relationship & Relationship Set

- A relationship is an association among two or more entities
- A **relationship set** is a set of relationships of the same type. A relationship set can be thought of as a set of *n*- tuples

f(*e*1*; : : : ; en*) *j e*1 *2 E*1*; : : : ; en 2 Eng*

- Each *n*-tuple denotes a relationship involving *n* entities *e*1 through *en*, where entity ***ei*** is in entity set *Ei*. we show the relationship set Works

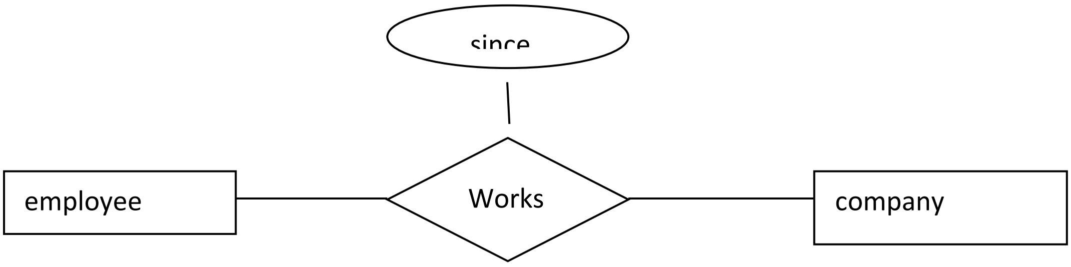


Fig:- Works in relation ship

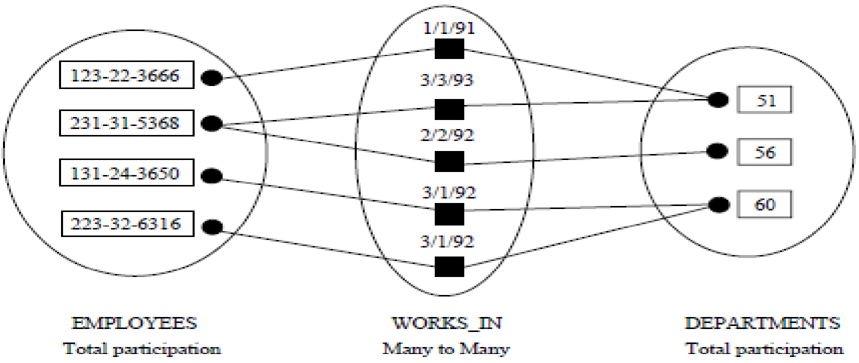
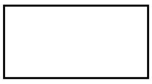
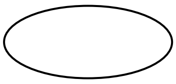
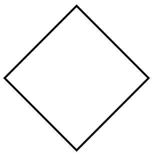
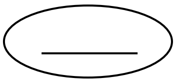
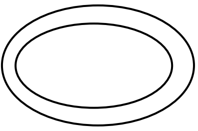
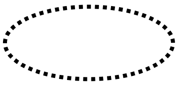


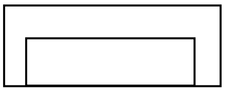
Fig:- An instance of works in relationship

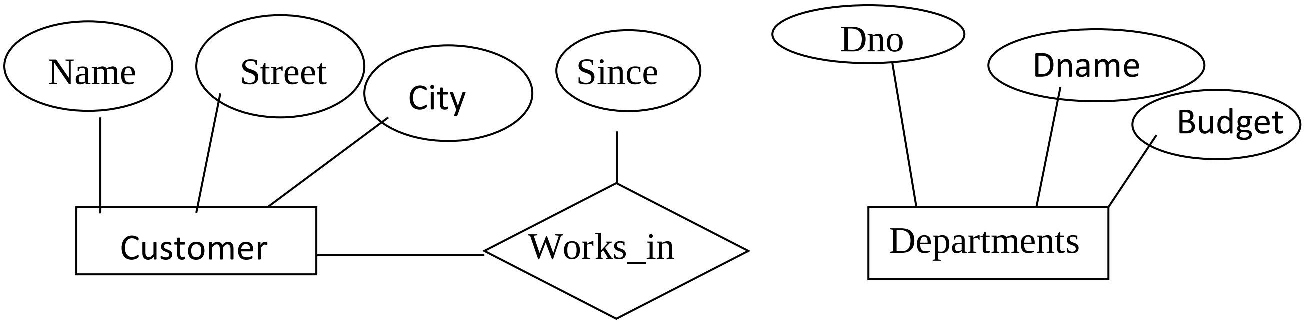
The symbols that can be used in this model are as follows.

- Rectangles  Entities.
-
- Ellipses  Attributes.
- Lines Links.
- Diamonds  Relationships.
- Under Lined Ellipse  Primary key.

Doubled Lined Ellipse  Multi Valued Attribute.

Dashed Ellipse  Derived Attributes.

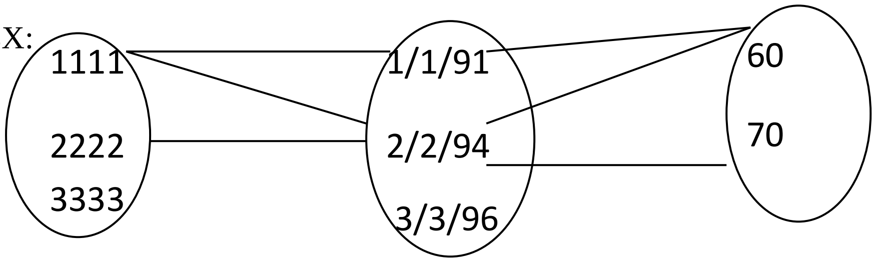
Double Lined Rectangle  Weak Entity Set.



Instance:

- An instance of a relationship set is a set of relationships.
- It is a snapshot of the relationship at some instant of time.

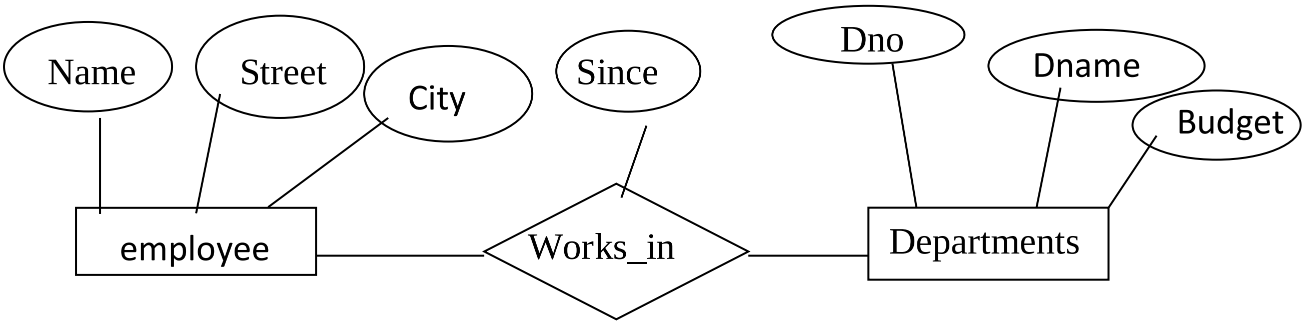
EX:



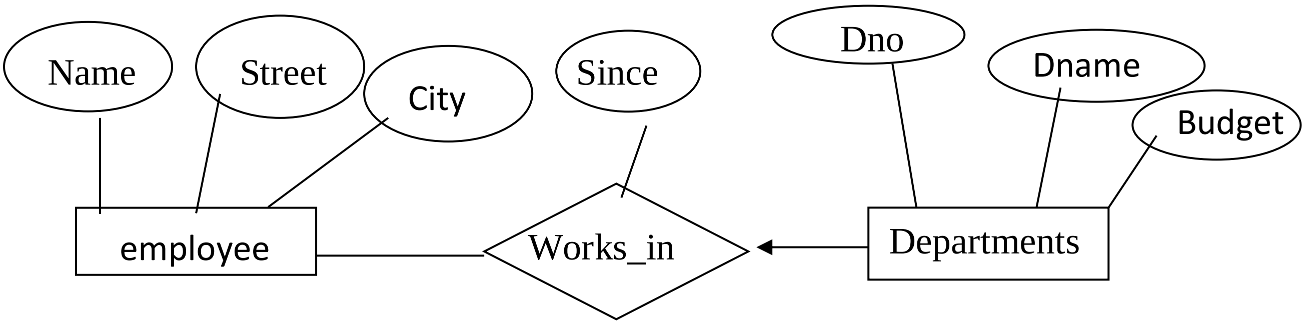
Mapping Cardinalities:- Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set. These are categorized into 4 types.

1. Many to Many:

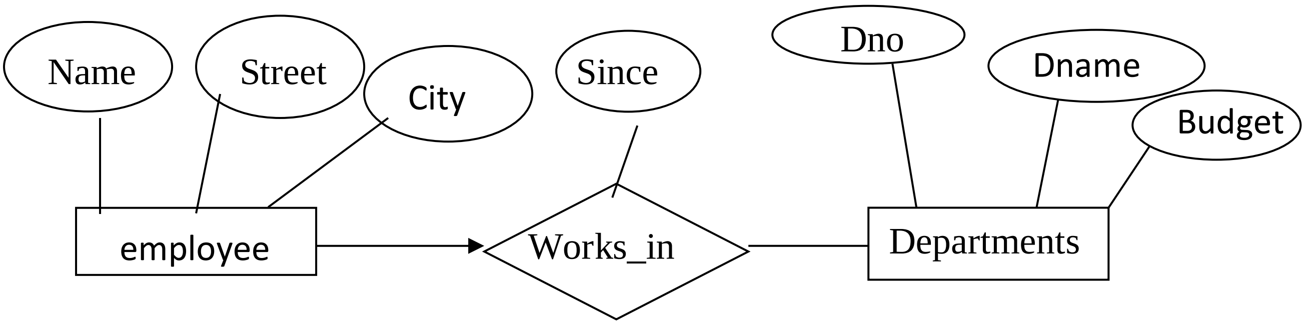
An employee is allowed to work in different departments and a department is allowed to have several employees.



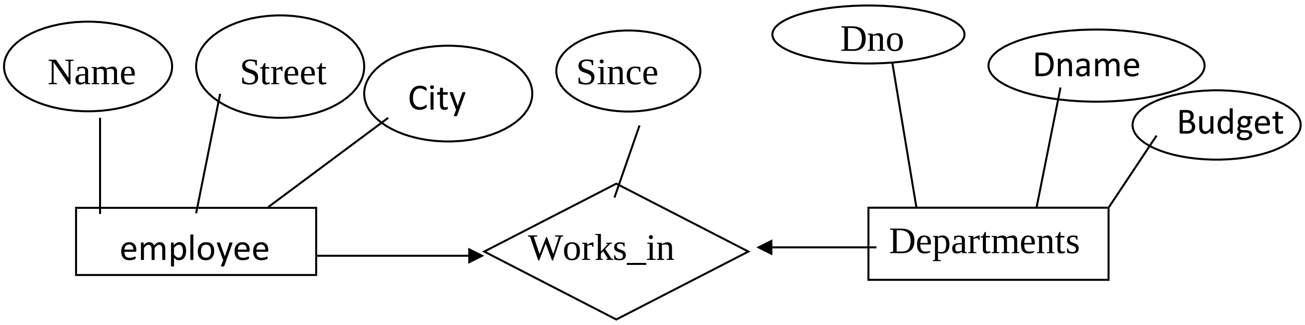
2. One to Many: 1 employee can be associated with many departments, where as each department can be associated with at most 1 employee as its manager.



3.Many to One: Each employee works in at most 1 department.i.e, many employees can work in same department.



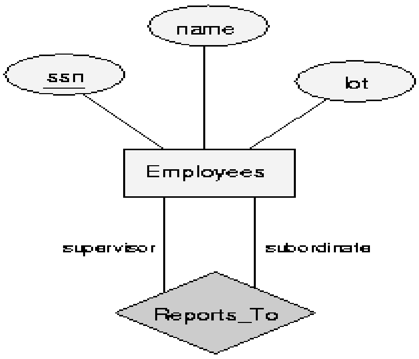
4.One to One: Each employee can manage at most 1 department.



Role: We indicate roles in E-R diagrams by labeling the lines that connect diamonds to rectangles.

Role indicator - If an entity set plays more than one role, role indicators describe the different purpose in the relationship.

Here an example is a single Employee entity set with a relation Reports-To that relates supervisors and subordinates. Here the emp1,emp2 are entities of employee .

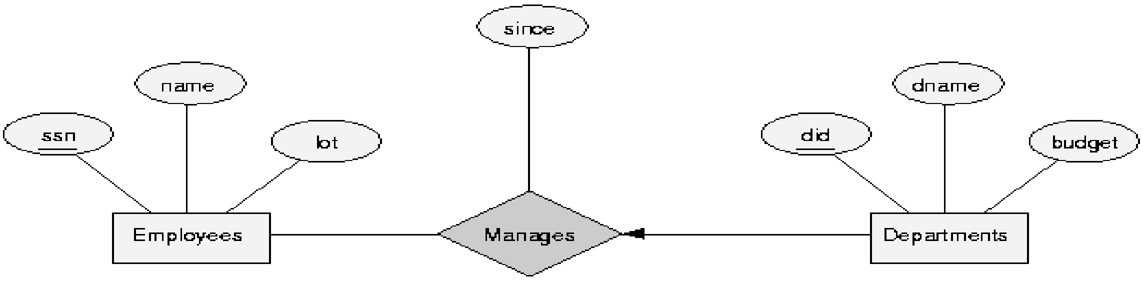


3. ADDITIONAL FEATURES OF THE E-R MODEL

3.1. Key Constraints:

- A key is a set of attributes for one entity set such that no two entities in this set agree on all the attributes of the key.

- Sometimes, an additional constraint exists for a given relationship set. For example, consider a relationship set "Manages" which associates departments with employees. If a department cannot have more than one manager, this is an example of a one-to-many relationship set. This type of constraint is called **a key constraint**.
- It is represented in the ER diagrams by drawing an arrow from an entity set E to a relationship set R when each entity in an instance of E appears in at most one relationship in (a corresponding instance of) R.



3.2 Participation Constraints:

- The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type.
- A department has at most one manager. This requirement is an example of participation constraints.
- There are 2 types of participation constraints, which are as below.

    1.Total.

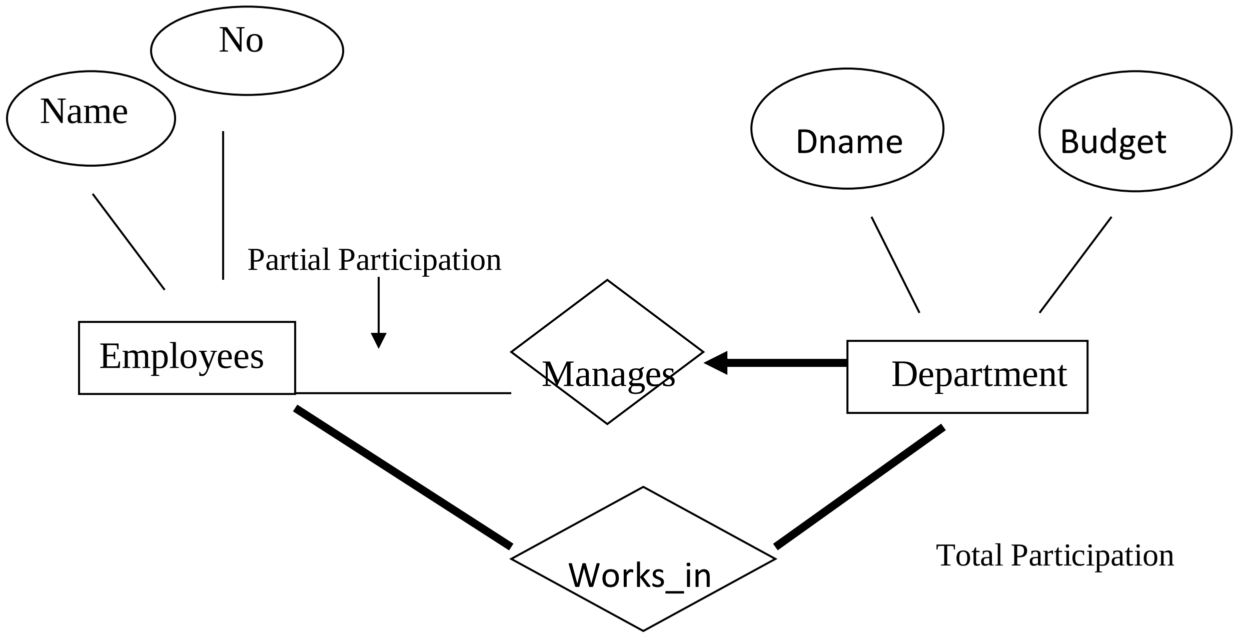
    2.Partial.

3.2.1. Total: An entity set dependent on a relationship set and having one to many relationships is said to be 'total'. The participation of the entity set 'departments' in the relationship set 'manages' is said to be total.

3.2.2.Partial: A participation that is not total is said to be partial.

Example: Participation of the entity set 'employees' in 'manages' is partial, since not every employee gets to manage a department.

- In E-R diagram, the total participation is displayed as a 'double line' connecting the participating entity type to the relationship, where as partial participation is represented by a single line.
- If the participation of an entity set in a relationship set is total, then a thick line connects the two.
- The presence of an arrow indicates a key constraint.



3.3. Weak Entity Set:

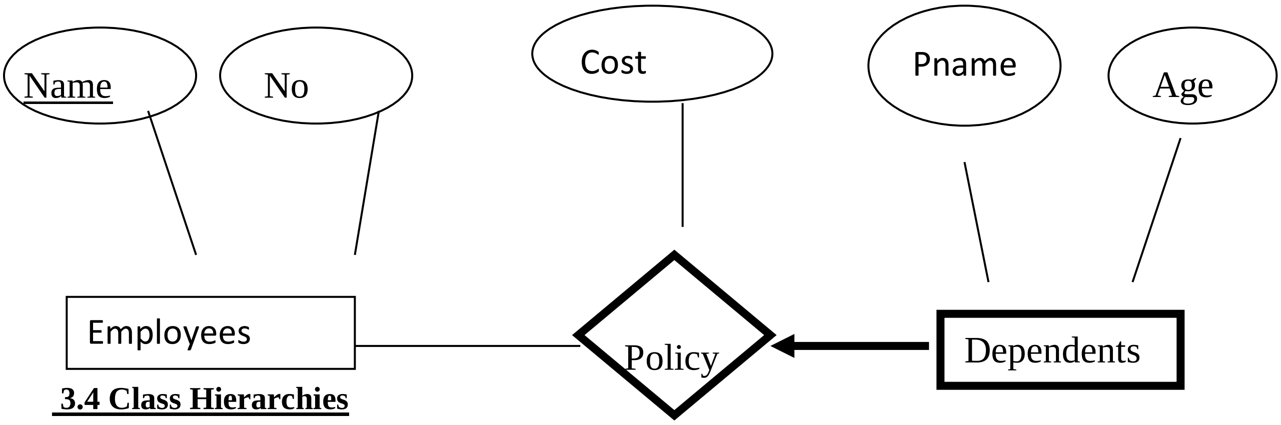
- An entity that cannot be identified uniquely without considering some primary key attributes of another identifying owner entity. **Weak Entity Type:** Entity types that do not have key attributes of their own are called as weak entity types. A weak entity type always has a 'total participation constraint'.
- A weak entity set can be identified uniquely only by considering some of its attributes in conjunction with the primary key of another entity (Identifying owner).
- For any weak entity set, following restrictions must hold.
  a. The owner entity set and the weak entity set must participate in a One-to-many relationship set, which is called as the 'Identifying Relationship Set' of the weak entity set.
  b. The weak entity set must have total participation in the identifying relationship set.

Example: 'Dependents' is an example of a weak entity set.

- **Partial key of the weak entity set:** The set of attributes of a weak entity set that uniquely identify a weak entity for a given owner entity is called as 'partial key of the weak entity set'.

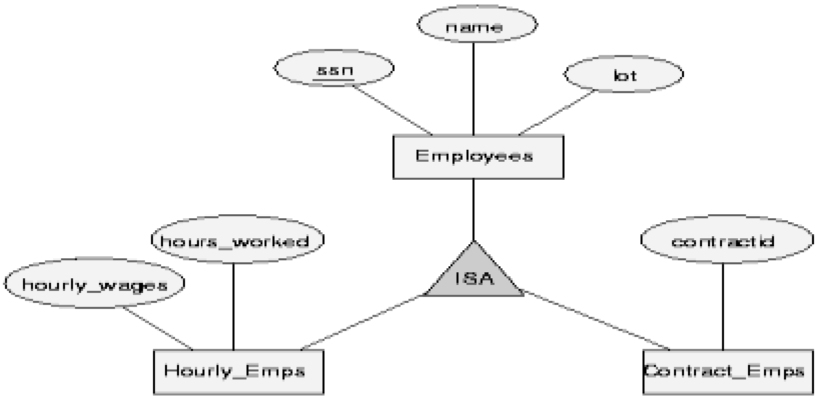
Example: 'Pname' is a partial key for dependents.

- The dependent weak entity set and its relationship to employees is shown in the following diagram.
- Linking them with a dark line indicates the total participation of dependents in policy.
- To understand the fact that dependents is a weak entity and policy is its identifying relationship, we draw both with dark lines.
- To indicate that 'pname' is a partial key for dependents, we underline it using a broken line.



3.4 Class Hierarchies

As with object-oriented programming, it is often convenient to classify an entity sets as a subclass of another. In this case, the child entity set inherits the attributes of the parent entity set. We will denote this scenario using an **"ISA"(**Read is a**)** triangle, as in the following ER diagram:



- Specialization is the process of identifying subsets of an entity set (the **superclass**) that share some distinguishing characteristic. Typically the superclass is defined first, the subclasses are definednext, and subclass-specific attributes and relationship sets are then added.

- Generalization consists of identifying some common characteristics of a collection of entity sets and creating a new entity set that contains entities possessing these common characteristics. Typically the subclasses are defined first, the superclass is defined next, and any relationship sets that involve the superclass are then defined. Two entity sets Motorboats and Cars may be generalized into an entity set Motor Vehicles.

- We will assume that two subclasses are allowed to contain the same entity. However, if we wish to allow an entity to lie in more than one such subclass, we will specify an **overlap constraint**. (e.g. "Contract_Emps OVERLAPS Senior_Emps")

- **Covering constraints** determine whether the entities in the subclasses collectively include all entities in the super class. every Motor Vehicles entity have to be either a Motorboats entity or a Cars entity? Intuitively, yes; a characteristic property of generalization hierarchies is that every instance of a superclass is an instance of a subclass.

3.5. Aggregation:

- Aggregation is an abstraction for building composite objects from their component objects.
- Aggregation is used to represent a relationship between a whole object and its component parts.
- Aggregation allows us to indicate that a relationship set (identified through a dashed box) participates in another relationship set.
- This is illustrated with a dashed box around sponsors.
- If we need to express a relationship among relationships, then we should use aggregation.

3.5.1 Aggregation versus Ternary Relationship:

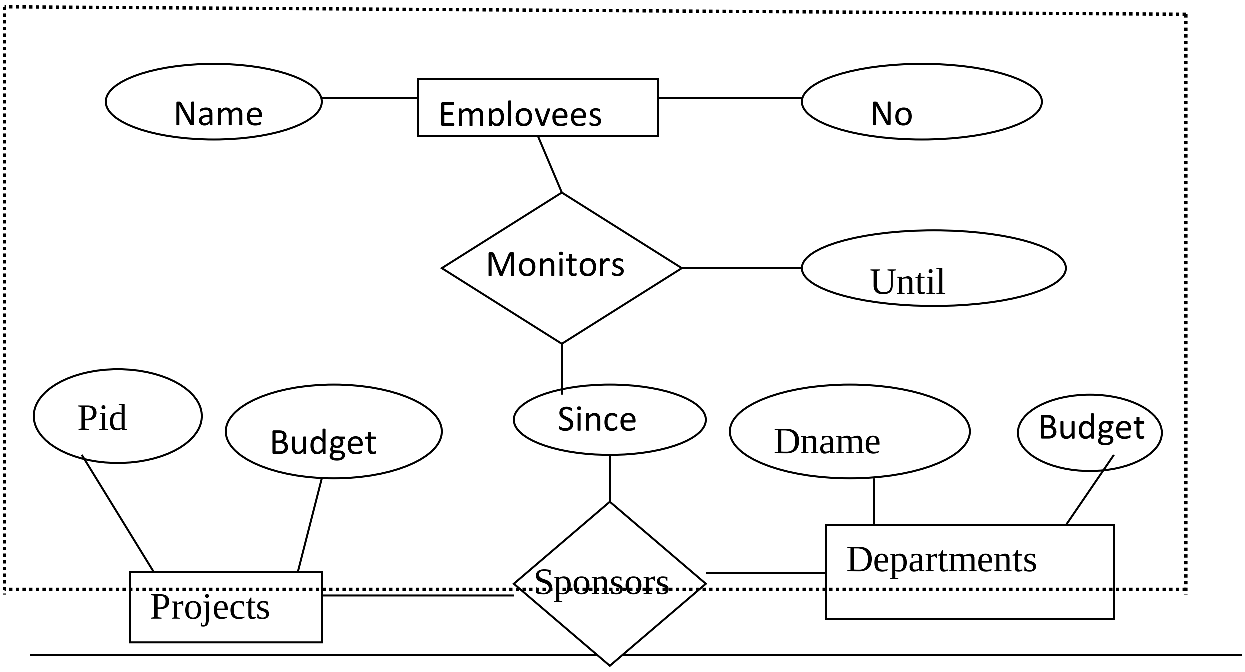
- We can use either aggregation or ternary relationship for 3 or more entity sets.
- The choice is mainly determined by The existence of a relationship that relates a relationship set to an entity set or second relationship set.
- The choice may also guided by certain integrity constraints that we want to express.

According to the diagram,

1. A project can be sponsored by any number of departments.

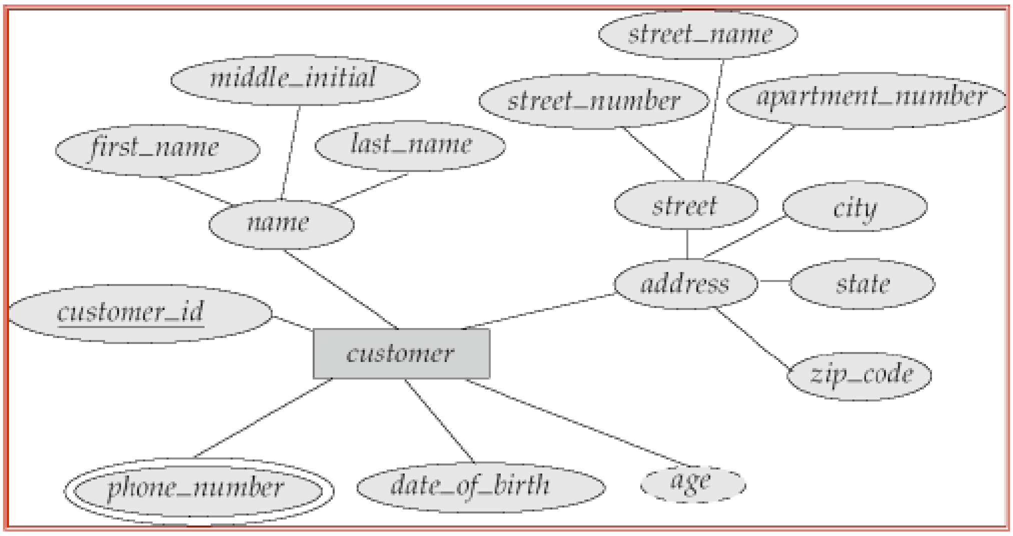
2. A department can sponsor 1 or more projects.

3. One Or more employees monitor each sponsor ship.

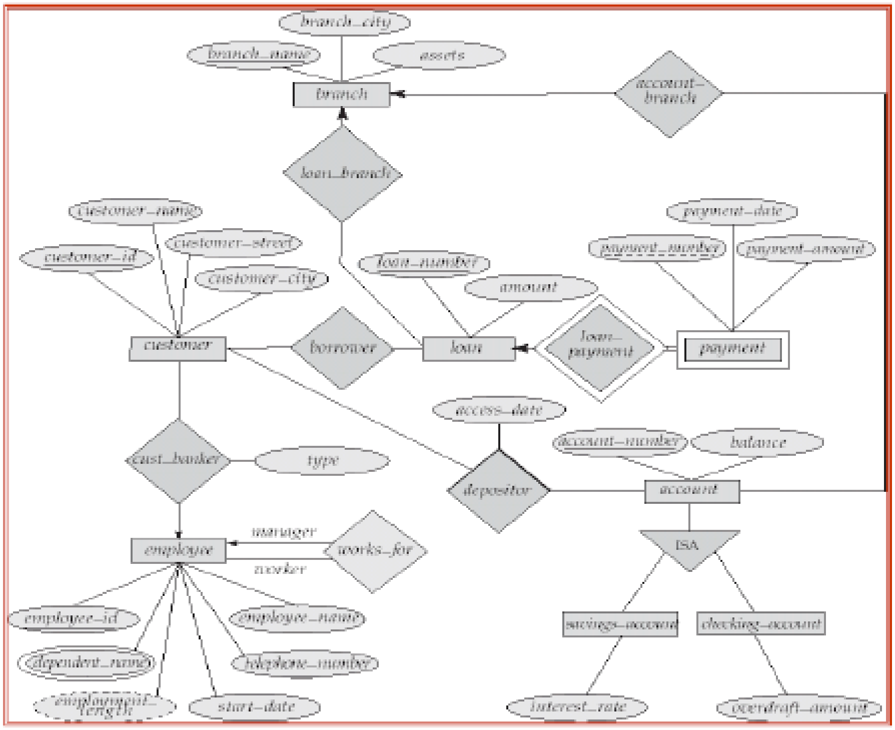


- Consider the constraint that each relationship be monitored by at most 1 employee.
- We cannot express this constraint in terms of the ternary relationship in the following diagram. In that we are using a ternary relationship instead of aggregation.
- Aggregation groups a part of an E-Are diagram into a single entity set allowing us to treat the aggregate entity set as a single unit without concern for the details of it's internal structure.
- Thus, the presence of such a constraint serves as another reason for using aggregation rather than a ternary relationship set.

E-R Diagram With Composite, Multivalued, and Derived Attributes



E-R Diagram for a Banking Enterprise


branch_city
branch_name
assets
branch
account–branch
loan_branch
customer_name
payment_date
customer_street
customer_id
payment_number
payment_amount
loan_number
customer_city
amount
customer
borrower
loan
loan–payment
payment
access–date
account–number
balance
cust_banker
type
depositor
account
manager
employee
works_for
worker
ISA
employee–id
employee_name
savings–account
checking–account
dependent_name
telephone_number
employment–length
start–date
interest_rate
overdraft_amount