UNIT 3: Key Constraints

Notebook:	Database Management		
Created:	2/17/2020 9:08 PM	**Updated:**	10/4/2023 1:23 PM
Author:	pagulearn.5334@gmail.com		

Key Constraints

- Each and every entity in any entity set is different from other one.
- From the database perspective , this difference can be represented by the values of attributes for each entity.
- So, No any two entities in an entity set will have exactly same values for all the attributes.

- In , Database there must be someway to uniquely identify an entity within an entity set.
- for , these purpose several key has been defined.

Super Key

- A super key is a set of one or more attributes that allows to identify each entity uniquely in the entity set.
- Example:
  - Entity set : Customer
  - Attribute **cid** can distinguish one customer from another.
  - **cid** is super key for customer entity set.
- If K is a super key then any super set of K is super key.
- Example:
  - For entity set customer - combination of cid with any other attribute (cid, contact-no) is a super key.

Candidate Key

- A super key for which no subset is a super key is called a candidate key.
- Candidate key is a minimal super key.
- A candidate key is also referred as "key" only.
- A candidate key is sufficient to identify each and every entity uniquely within an entity set.

- Example:
  - For entity set Customer : (cid,contact-no) is super key
  - But only **cid** alone can identify entities uniquely , so contact-no attribute is redundant in super key.
  - after removing such redundant attributes from a super key, the remaining attribute can be consider as a candidate key.
  - So, cid is a candidate key.
- Sometime , there can be more than one candidate keys for the same entity set .

Primary Key

- A Primary key is a candidate key that is chosen by the database designer to identify entities uniquely in an entity set.
- Example:
  - If database designer choose candidate key : cid to identify all customers uniquely then cid is a primary key for customer entity set.

Alternate Key

- An alternate key is a candidate key that is not chosen by the database designer to identify entities uniquely in an entity set.
- Example:
  - If database designer choose candidate key : cid to identify all customers uniquely then all other possible candidate keys can be consider as alternate key.

Primary Key for a Relationship Set

- Like as various entities of an entity set ,Various relationships of a relationship set should also be distinguished from each other.
- A primary key for relationship set can be defined in following manner.

- A primary key for a relationship set is a union of primary keys of all entity sets participating in that relationship set.

- The Primary key for a Relationship set R = Primary-key(E1) U Primary-key(E2) UPrimary-key(E3) U.....U Primary-key(En)
  - where E1,E2,E3.... En are entity sets participating in R.

Note :

- Primary key of a relationship set depends on the mapping cardinality of the relationship set.

Example :

Relationship Set : Account_Holder
Descriptive attribute : access_date
Two Entity set Participating in Relationship set : Customer , Account
Primary key for Account : ano
Primary key for Customer : cid

- If cardinality of relationship is **many to many from Customer-Account**
  - Primary key for Account_Holder = P.K(Customer) U P.K(Account)

- If cardinality of relationship is **many to one from Customer-Account**
  - Primary key for Account_Holder = P.K(Customer)

- If cardinality of relationship is **one to many from Customer-Account**
  - Primary key for Account_Holder = P.K(Account)

- If cardinality of relationship is **one to one from Customer-Account**
  - Primary key for Account_Holder = P.K(Customer) / P.K(Account)

Participation Constraints

- Association among two entity sets is referred to as a participation.
- Participation can be divided into two categories
  - Total Participation
  - Partial Participation

Total Participation

- The participation of an entity set E in a Relationship set R is said to be total , if every entity in E participates in at least one relationship in R.
- Example:
  - Relationship set: Borrower

- Entity set : Customer , Loan
  - Here , each and every loan belongs to at least one customer.
  - So the participation of Loan in the relationship set borrower is total.

Partial Participation

- The participation of an entity set E in a relationship set R is said to be partial , if not every entity in E participates in relationship in R.
- Example:
  - Relationship set: Borrower
  - Entity Set: Customer, Loan
    - it is not necessary that each and every customer in bank has taken a loan.
    - So ,there may be some customer in Customer entity set who does not have borrowed the loan.
    - So, Participation of Customer in relationship set Borrower is Partial.

KEY EXAMPLE:

Student(StudentID,email,fname,city)

Book(BookID,ISBN,Title,Author)

UNIT 3: Entity Relationship Model(E-R) Introduction

Notebook:	Database Management		
Created:	1/23/2020 10:35 PM	**Updated:**	9/26/2023 1:09 PM
Author:	pagulearn.5334@gmail.com		

Introduction

- Now a days Database applications have become an essential part of our day to day life.p
- But to design and develop database for any organization , it is necessary to understand **data requirements** of an organization or a system.
- The E-R model is the most widely used high level **conceptual data model** which represents data requirements in a concise manner.

Banking System as a Database Application

Data Requirements of the Banking System

The Bank keeps track on various branches, customers,accounts,loans and employees.The basic data requirements of Banking system is follows.

- The bank is organized into three branches -- [VVN,KSAD,ANAND]
  - each branch is uniquely identified by branch name.
  - each branch has an address
  - The bank monitors assets of each branch.
  - Each branch has its own branch manager.
- A Bank customer is a person who owns an account in the bank.
  - Each customer is given a unique identification no-customer_id
  - The bank also manages other data of customer such name,address,contact no.
- An Employee is a person who works in a bank.
  - Each employee is given a unique identification no-employee_id
  - other data about an employee -name, address,post,salary,branch name
  - one employee is considered as a manager for each branch and other employees of that branch works under this manager.
- Each account in a bank is given a unique identification number, -- account no.
  - Each account has its own balance
  - Each account belongs to some particular branch of a bank.
  - The bank maintains data about various transactions performed on these accounts.

    - These transaction data includes - transaction no, type,amount,date
    - A customer can have multiple accounts.
    - A single account can be jointly acquired by multiple customers.
- Each loan borrowed from the bank is given a unique identification number-loan no
    - for each loan ,bank keeps track of the loan amount and loan payments.
    - for loan payments , for specific loan contains data such as payment no,payment amount,date
    - one customer can have more than one loan.
    - A loan can be held by more than one customer.

Basic E-R Concepts

- The concept of Entity relationship model was given by Chen in 1976.
- The Entity-Relationship model is based on the perception of a real world.
- A real world is consist of a collection of basic objects and relationship among these objects.
- This model is also referred as the E-R model.
- The E-R model is high level conceptual data model.
- It is widely used in database design to represent a conceptual schema.
- The conceptual schema is concise description of the data requirements of the organization and different kind of users associated with the organization.
- The conceptual schema represented by the E-R Model is called an **E-R Schema.**

- E-R model employs three basic notation
    - Entities
    - Attributes
    - Relationships

Entity set

- **Entity**
    - An Entity is a 'thing' or 'object' existing in the real world that is distinguishable from all other objects.
    - Each entity possesses certain properties. The values of these properties distinguish one entity from other entity.
    - Example:
        - any customer of a bank is an entity.

        - customer say 'Riya' exists in the real world, and she can be distinguishable from other customers based on her properties values.
- **Concrete Entities**
    - A Concrete entity is a physically existing entity, such as a customer,an employee,or a book.
- **Abstract Entities**
    - An abstract entity has no physical existence,--account ,loan , subject
- **Entity set**
    - An entity set is a set of entities of the same types.
    - All the entities in an entity set share the same properties.
    - The individual entities that constitute an entity set are called extension or member of the entity set.
    - **Example:**
        - In a Banking system , the set of all individual customers can be referred as an entity set **Customer.**
        - so all individual customer are called members of entity set **Customer.**
- **Disjoint ness**
    - Entity sets need not to be disjoint.
    - This means that, a single entity can be a member or extension of more than one entity set.
    - **Example:**
        - A single person can be a customer as well as an employee of a bank.
        - so , That person is an extension of both the entity sets customer as well as Employee.

Attributes

- Attributes are descriptive properties possessed by each member of an entity set.
- Example:
    - A member of entity set Customer has attributes
        - customer id
        - name
        - address
        - contact no

Value of an Attribute

- Each entity has a value for each of its attributes.
- Example:

- A customer entity may have value C01 for cid, 'Riya' for cname and so on.

Representation of an Entity

- In E-R model , an entity is represented by a set of (attribute,data value) pairs.
- Example:
  - A particular customer entity can be described by the set {(cid,'C01'),(name,'Riya'),(address,'Anand')}

Same Attributes ,but Different Values

- Each individual entity of an entity set shares same attributes , however each entity may have different values for each attributes.
- Example:
  - All individual customers of customer entity set shares same attributes like - cid, cname, address,contact_no , but each customer have different values for each of these attributes.

Domain of an Attributes

- A domain of an attribute is a set of permitted values for that attribute.
- A domain of an attribute is also known as value set of an attribute.
- Example:
  - A bank is organized in three branches named 'vvn','ksad','anand'.
  - so set of permitted values for an attribute bname is {'vvn','ksad','anand' }.
  - this set can be referred as a domain or value set for an attribute bname.

Null Value

- An attribute takes a null value when an entity does not have a value for that attribute.
- A null value indicates
  - not applicable
  - Missing
  - Not Known

Types Of Attributes

- Simple vs Composite Attribute
- Single vs Multiple Value Attribute
- Stored vs Derived Attribute

Composite vs Simple Attribute

- Composite attributes can be divided into smaller sub-parts.
- Composite attributes forms a hierarchical structures.
- Example:
    - Address attribute can be divided into sub parts such as
        - society
        - city
        - pin-code
    - Even society can be further divided into
        - street-no
        - street-name
- Simple attribute cannot be divided into sub-parts.
- Simple attributes are also referred as atomic attribute.
- Example:
    - cid , ano,balance,bname.

Single v/s Multi Valued Attribute

- Single value attributes have single data values for a particular entity.
- Example:
    - cid , ano
- In contrast to single valued attributes , multi valued attributes have multiple data values for a particular entity.
- Example:
    - contact no
- Multi valued attributes may have lower and upper bound on the number of values.
    - Contact no for any customer can be restricted to two.
    - So, any customer only up to two numbers will be stored and others will be ignored.
    - but it is not possible to put such kind of upper bound when data values cannot be ignored.
    - Example, **author_name** is a multi valued attributed for Book entity set.
        - it is necessary to store all the author names for any book.
- Multi value attribute are problematic for developing a database .

- They must be converted to single value attributes by some means.

Stored v/s Derived Attribute

- The value for derived attribute can be derived from the values of other related attributes.
- Example :
  - For Customer Entity set two attributes are : age , birthdate
  - Here, age can be calculated by using birth date and current date.
  - age can be considered as a derived attribute , while birthdate can be considered as stored or base attribute.
- The value of a derived attribute is not stored in a database .but it is computed from stored attribute whenever required.

UNIT 3 : Relationship and Constraints.

Notebook:	Database Management		
Created:	2/12/2020 9:10 PM	**Updated:**	4/20/2022 3:07 PM
Author:	pagulearn.5334@gmail.com		

Relationship

- A relationship is an association among several entities.
- Example:
  - in banking system ,each and every account is associated with particular customer.
  - Account "A01" is owned by a Customer named "Riya".
  - such kind of association is known as relationship.

Relationship Set

- A relationship set is a set of relationships of the same type.
- Example:
  - There is a relation between each account and particular customer.
  - The set of all such relationships ,named "Account_Holder" is known as relationship set.

Relationship Instance

- A relationship instance associates particular entities of entity sets.
- Example:
  - Account_holder relationship set contains different relationship instance like {A01,"Riya"} ,{A02,"RAJ"}.
  - above relationship instances represent the fact that , in a real world a bank customer "Riya" owns an account "A01"

Participation

- The association between two entity sets is referred as participation.
- Example:
  - Entity set Customer and Account **participates** in relationship set Account_Holder.

Entity's Role

- The function that an entity plays in a relationship is called that

entity's role.

- When entity sets participating in relationship set are distinct , roles are implicit and need not to be specified.
- But , in relationship sets , such as recursive relationship set it is necessary to specify entity's role.

Recursive Relationship Set

- A relationship set is recursive if participating entity sets are not distinct.
- If same entity set participates in a relationship set more than once.
- Example:
  - Entity set : Employee
  - Relationship set : Works_under
  - Employee as a worker works_under the another Employee as Manager.
- Here, same entity set Employee participates twice in a relationship set Works_under.
- So, this relationship set is a recursive one. where some entities are in the role of worker and some are in the role of managers.
- So , in such kind of relationship set it is necessary to specify entity's roles.

Descriptive Attributes

- Like as an entity , a relationship set may also have attributes. these attributes are called as descriptive attributes.
- Example:
  - Relationship set : Account_Holder
  - Entity set : Customer ,Account
- But it may be possible that Account_Holder can have an attribute : access_date .
- This attribute specifies the most recent date on which an account is accessed by a customer.

Degree of the Relationship set

- The total no of entity set participating in the relationship set is known as the degree of the relationship set.
- Example:
  - Relationship set: Account_holder
  - its degree is 2 : two entity set : Account , Customer

    - A relationship set of degree 2 is called binary relationship set.
- A relationship set of degree 3 is called ternary relationship set.
- A relationship set of degree n is called n-ary relationship set.
- Example:
    - Relationship set : Works_in
    - Entity set : Employee , Job, Branch
    - Job entity set has attribute like post , duration
    - This Ternary relationship specifies which **employee works in** which **branch** at which **post** and since **how long.**

Constraints

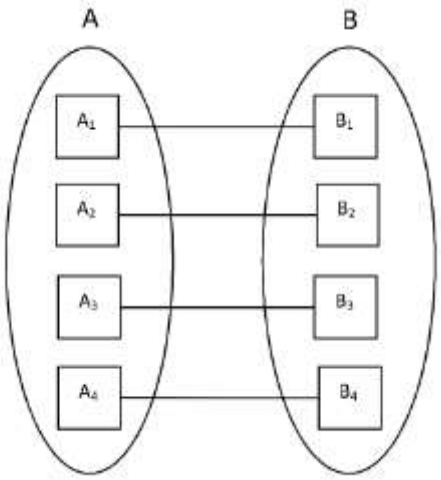
- The Constraints are restriction/rules to which the contents of the underlying database must confirm to.
- The E-R model specifies three types of constraints
    - Mapping Cardinalities
    - Key Constraints
    - Participation Constraints

Mapping Cardinalities

- Mapping Cardinalities express the number of entities to which another entity can be associated via a relationship set.
- Mapping cardinality is also reffered as cardinality ratio or cardinality constraint.
- It also represents how individual entities of an entity set associates in a relationship set.
- For a binary relationship set , mapping cardinality can be divided into four categories.
    - one to one
    - one to many
    - many to one
    - many to many
- Consider there are two entity set **E1** and **E2** participating in relationship set **R.**

One to One

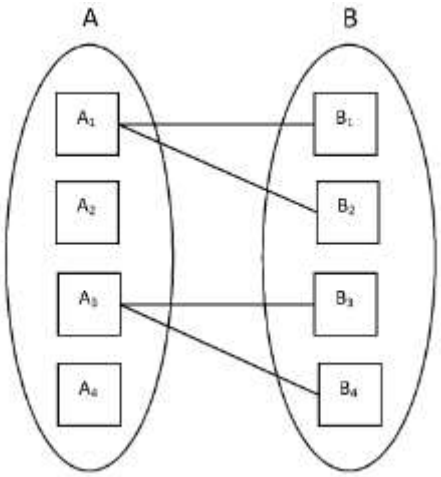
- An entity in E1 is associated with at most one entity of E2.
- and An Entity in E2 is associated with at most one entity of E1.



Example: In Banking System , no multiple accounts are allowed , no joints accounts are allowed.

One to Many

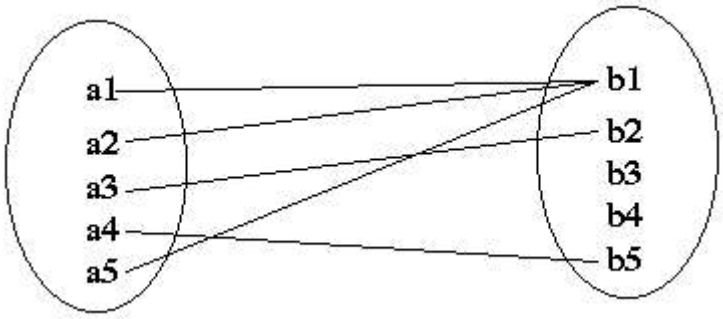
- An Entity in E1 is associated with zero or more entities of E2.
- and Entity in E2 is associated with at most one entity of E1.



Example: In Banking Multiple accounts are allowed but no join accounts are allowed.

Many to One

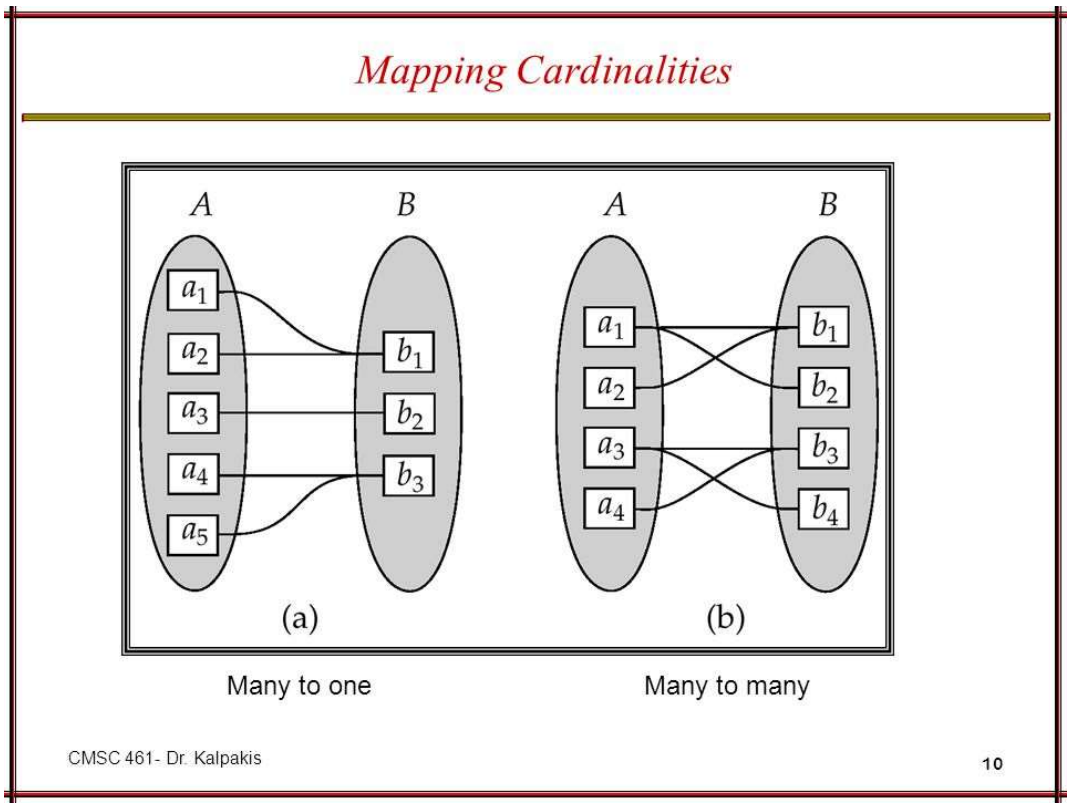
- An Entity in E1 is associated with at most one entity of E2.
- and Entity in E2 is associated with zero or more entities of E1.



Example: No multiple accounts are allowed but join accounts are allowed.

Many to Many

- An Entity in E1 is associated with zero or more entities of E2.
- and Entity in E2 is associated with zero or more entities of E1.



- Example: Multiple accounts as well as join accounts are allowed.

UNIT 3: Enhanced Entity Relationship Model

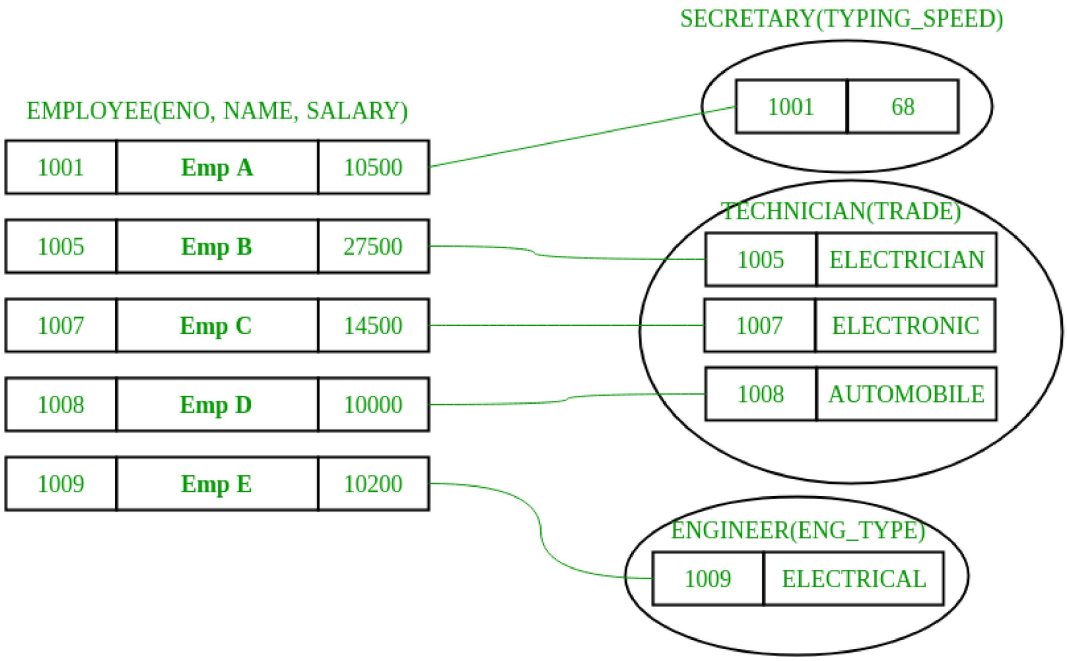
Notebook:	Database Management		
Created:	2/26/2020 9:50 PM	**Updated:**	5/6/2021 10:00 AM
Author:	pagulearn.5334@gmail.com		

Enhanced Entity Relationship Model

- The E-R model is adequate for representing database schema for old traditional system.
- It is not sufficient to represent modern day complex databases.

- The Enhanced Entity Relationship model provides certain extension to the basic E-R Model.
- The EER model includes all the basic concepts of E-R Model.
- It also provides some extended features to the E-R Model like
  - Concept of Super class and Subclass
  - Specialization
  - Generalization
  - Aggregation

- Sub Class:
  - A sub class is a subset of entities in the entity set , which has some distinct attributes from those in other subset.
- Super Class:
  - A super class is a generic entity set .



Specialization

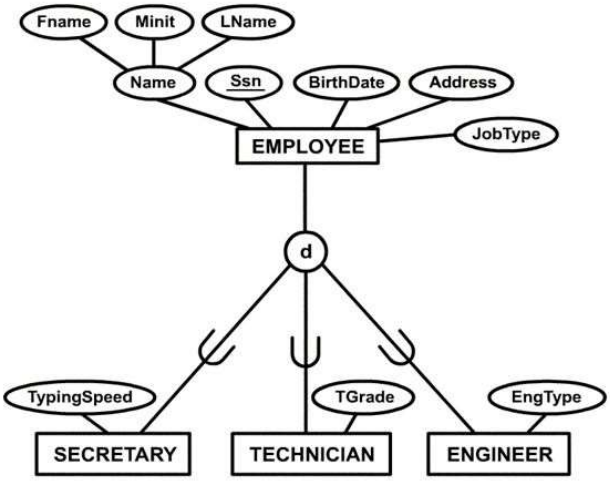
- Specialization is the process of defining sub classes of a super class based on some distinguishing characteristics.

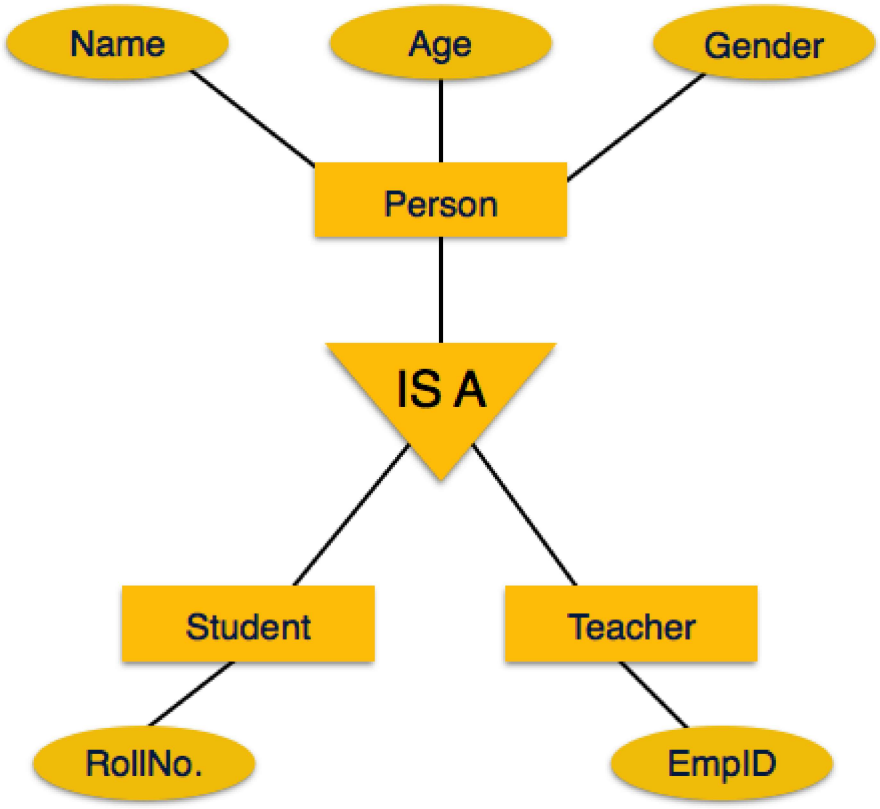
- Specialization identifies the sub sets of an entity set .
- Each of these sub sets posses some attributes which are unique to that subset.
- It is a top down process to define super class/sub class relationships.
- It maximize the difference between entities of an entity set and identifies unique characteristics for sub set of entities.

- Example:
    - Entity set : Employee
        - Attributes :
            - eid , ename , address , bdate , salary .
    - Employee can be of two types:
        - **Permanent Employee :** having fixed salary
        - **Part time employee :** having variable salary - which depends on work_hr ,rate_per_hr.
    - Based on the above information an Entity set can be further classified into two sub category(entity set)
        - Permanent and Part-time Employee
    - In other words we can derive two sub classes from an existing super class -Employee.
    - This process is called as **Specialization.**
    - Here , both sub classes have common attributes - eid ,ename , address , bdate
    - but they also have some attributes which are unique to each subset -
        - Permanent Employee : [fixed salary]
        - Part time Employee: [variable salary , work_hr , rate_per_hr]
    - So , through the specialization we have two super class/sub class relationships
        - Employee/Permanent
        - Employee/Part time
    - Specialization provides a better way to represent such type of problems in a more meaningful way.

Extra Example:

Example of a Specialization





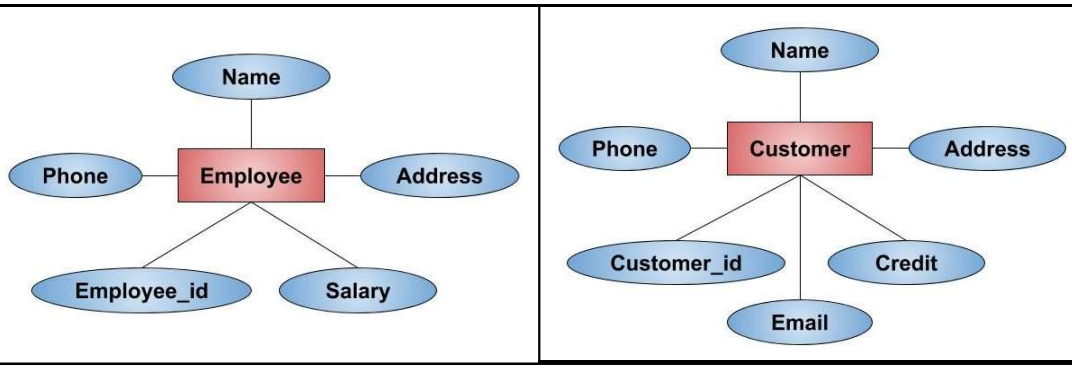
Generalization

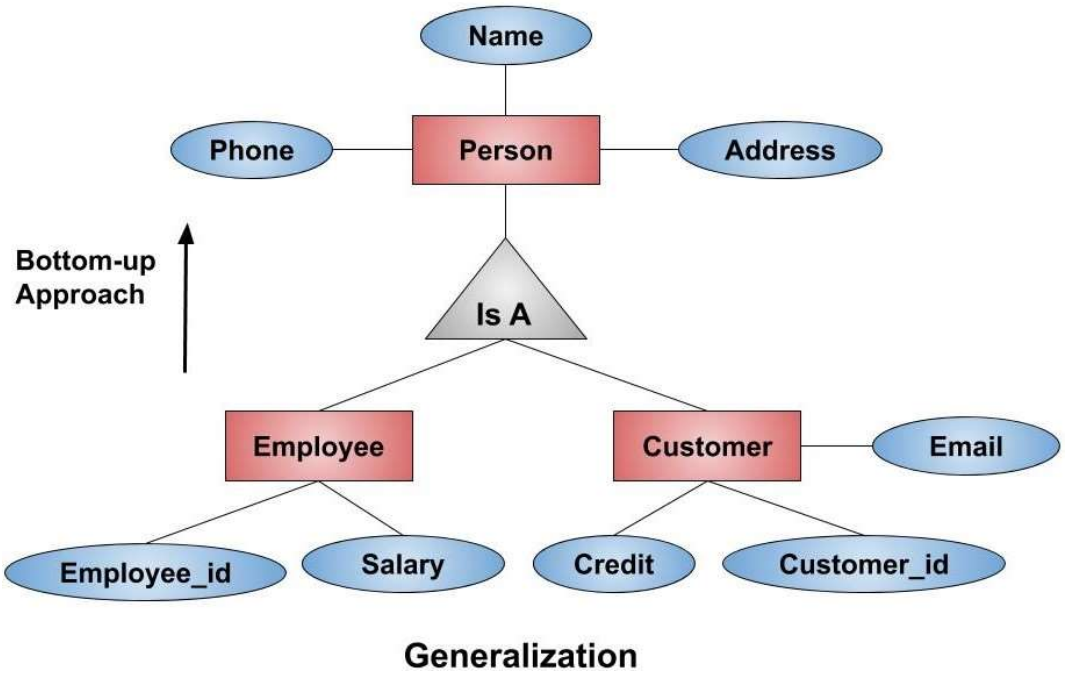
- Generalization is the process of defining a super class from sub classes based on some common characteristics.

- In other words , generalization identifies a super class from different sub classes.
- Super class possess some attributes which are common to the given sub classes.
- It is a bottom up process to define super class/sub class relationship.
- It minimizes the differences between entities of different entity sets by identifying common characteristics

- Example :
  - Entity set : Car
    - Attributes : company , model , price, vehicle_id , licence_no , no_of_seats
  - Entity set : Truck
    - Attributes: company , model ,price , vehicle_id, licence_no,capacity
  - Based on analysis we can derive a super class called **Vehicle** from the above two sub classes.
  - This super class contains the common attributes of the both sub classes.
  - Here , we can say that sub classes **Car** and **Truck** is generalized to a super class - **vehicle.**
- Generalization provides better way to represent such type of problems in more meaningful way.

Example



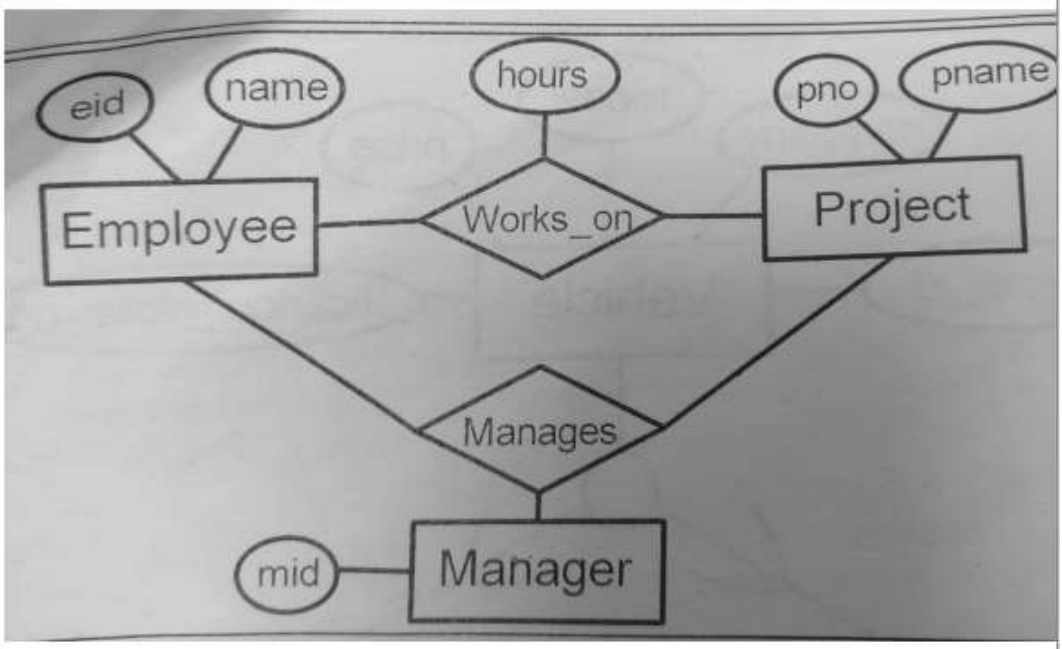


Generalization

Aggregation

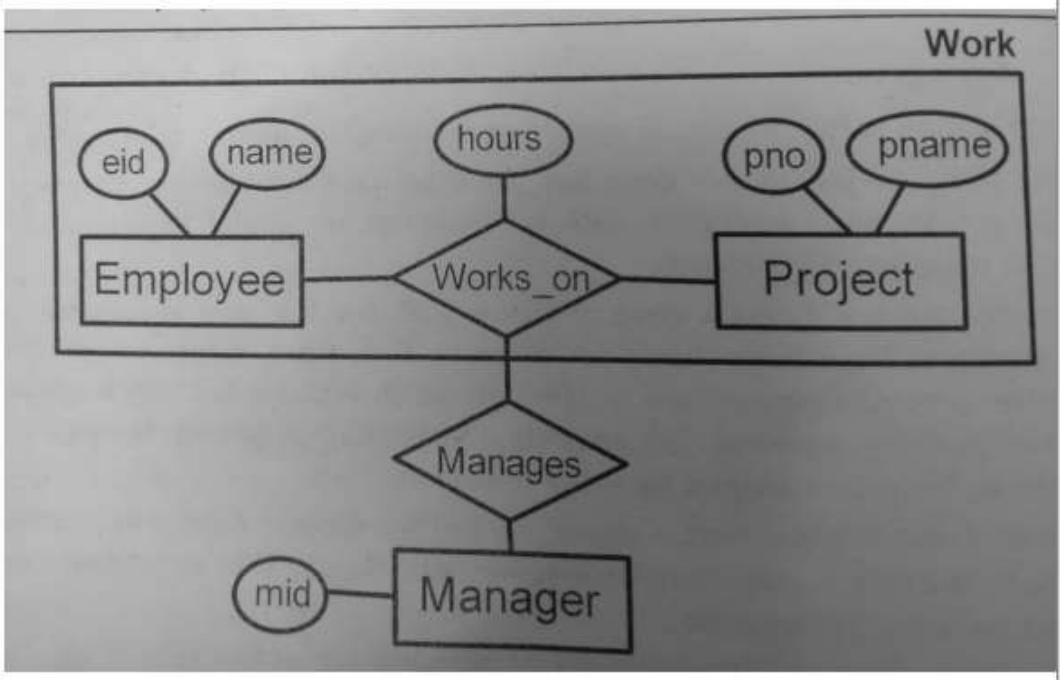
- Aggregation is the process of specifying relationship among relationship sets.
- E-R model does not allow to have relationship between two relationship sets. such relationship are useful to express E-R diagram in more meaningful way.

Example :



- Entity sets : Employee , Project , Manager
- Relationship set :
  - Works_on : it specifies which employees are working on which projects for how much hours.
  - Manages : it specifies relationship among employee and project managed by managers.
- Both these relationship set Works_on and Manages - represent overlapping of information.
  - Which means, they have some redundancy.

- Aggregation treats relationship set as an abstract entity set.
- It allows to define relationship between other relationship set.



- Through aggregation we can represent relationship set Works_on , Entity set : Employee and Project combined as an abstract entity set named **Work.**
- These abstract entity set **Work** is associated with relationship set **Manages** through relationship set **Works_on.**

- So , without introducing redundancy , these diagram represent
    - An Employee works on a project for some specific hours.
    - An Employee and Project may have an associated manager.

E-R Diagram and Symbols

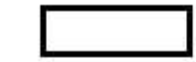
Notebook:	Database Management		
Created:	2/24/2020 9:09 PM	**Updated:**	4/3/2021 11:20 AM
Author:	pagulearn.5334@gmail.com		
URL:	https://whatisdbms.com/e-r-diagrams-in-dbms-components-symbols-and-notations/		

E-R Diagram

- An Entity relationship diagram is a graphical representation of the overall logical structure of a database.
- In E-R model , E-R diagram are used to represent an E-R schema.
- An E-R schema is a concise description of the data requirements of different kinds of users of organization.

- E-R diagram are simple and clear.
- They do not include any implementation details.
- Any non-technical user can also understand data requirements of an organization represented by E-R diagram.
- E-R schema represented by an E-R diagram can directly be mapped on to a relational model.

Basic notation used in E-R diagrams



Represents Entity

Represents Attribute

Represents Relationship

Links Attribute(s) to entity set(s) or
Entity set(s) to Relationship set(s)

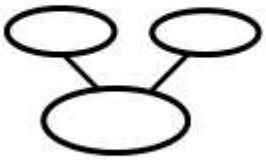
Represents Multivalued Attributes

Represents Derived Attributes

Represents Total Participation of Entity

Represents Weak Entity

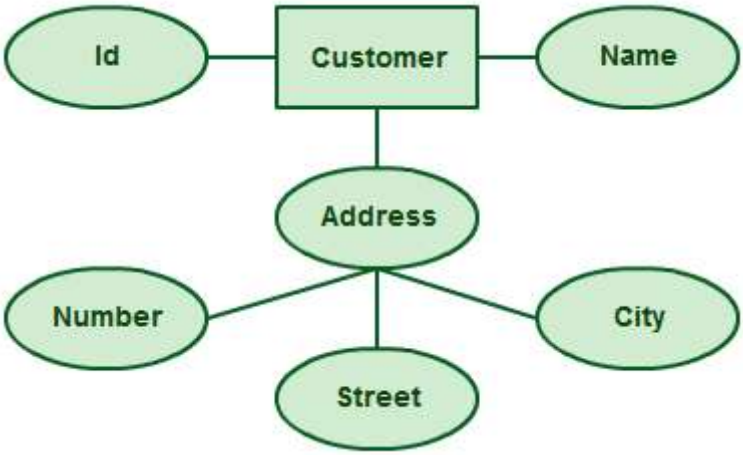
Represents Weak Relationships



Represents Composite Attributes

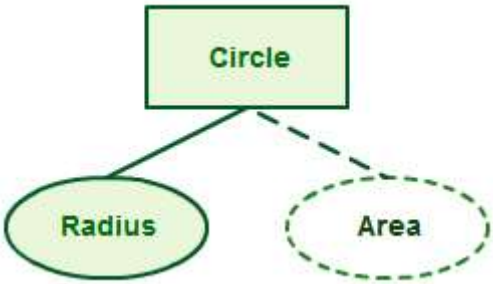
Represents Key Attributes / Single Valued
Attributes

Example of an Entity set and Attributes



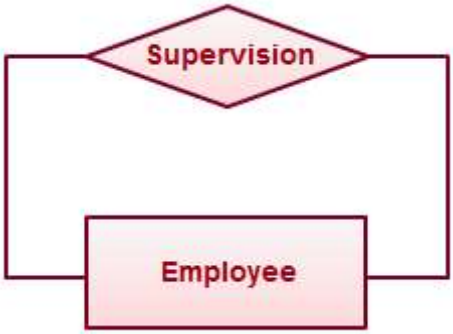
Example of Multi value attributes

Example of Derived attribute

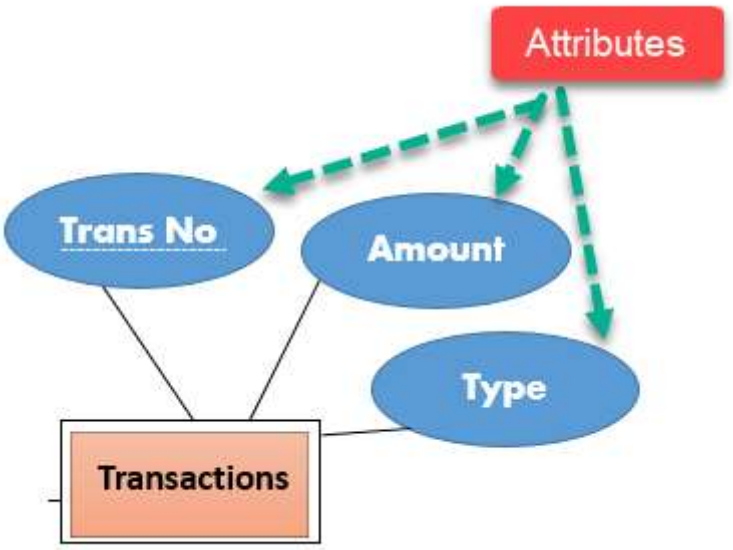


Example of Relationship Set

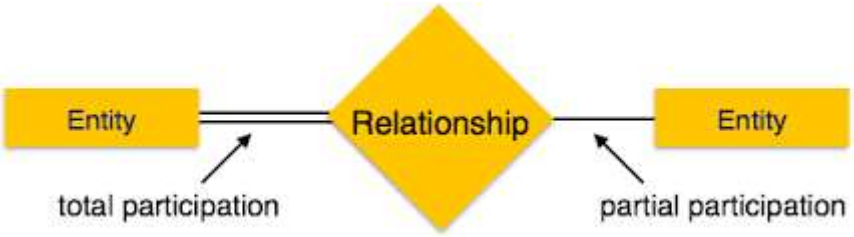
Example of Recursive Relationship set



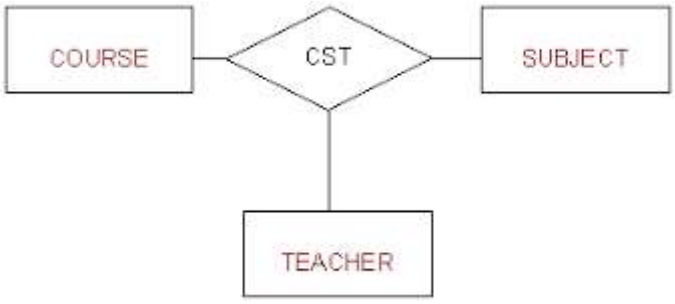
Example of Key Constraint



Example of Participation



Example of Ternary Relationship set



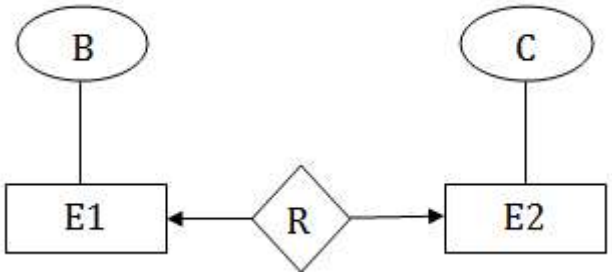
Weak Entity set And Strong Entity set

- An entity set that does not have sufficient attributes to form a primary key is called a weak entity set.

- In contrast , an entity set having sufficient attributes to form a primary key is called a Strong entity set.
- Example :
    - Entity set : Employee , Job, Branch
    - Relationship set : Works_in
    - Here , Job entity set is a weak entity set . It's existence depends on Entity set Employee(Strong Entity set)
    - Here, **post** is a discriminating attribute among each job of an employee.
    - This attribute , along with the primary key of strong entity set---Employee, forms a primary key for a Job entity set.

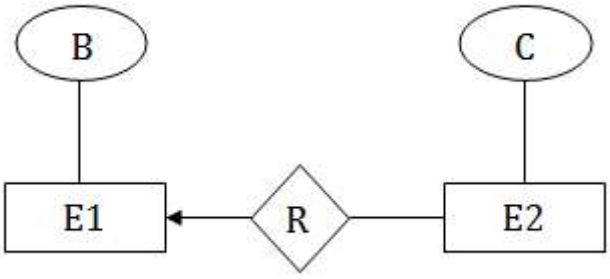
Cardinality Constraint notations

One to one



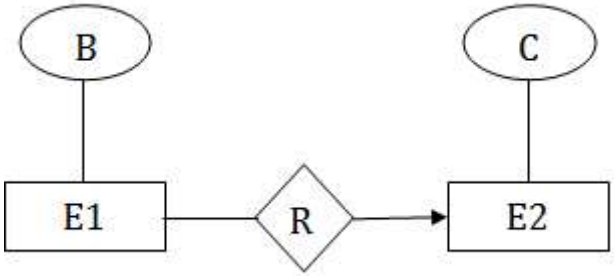
Example: " Student allotted a project"

one to many:



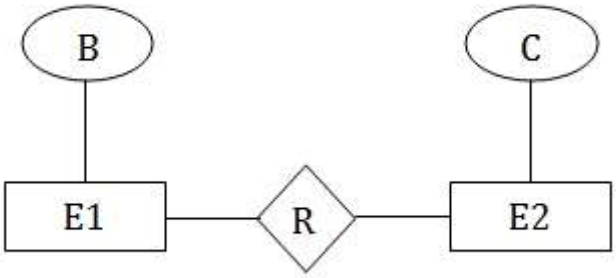
Example:" A department recruits faculty"

many to one



Example:" Many houses are owned by a person".

many to many



Example:" Author writes books"

Procedure to Construct an E-R Diagram

- List out the "things" or "objects" from the description of collected organizational data requirements and also check whether this thing can be consider as an entity or not.
- Taking one entity set at a time , identify whether a connection exists between this entity set and another entity set or not.

- Whenever a connection exits , define a relationship set between that pair of entity sets.
- Identify attributes for each entity set and relationship set if required. analyze these attributes to get primary key as well as classify types off attributes.
- For each relationship set , explore the mapping cardinality and participation constraints .
- Repeat above steps untill all the relationship set has been defined.
- form the E-R diagram using all entity sets and relationship sets.
- Review this E-R diagram to make sure that it represents all the data requirements of an organization.
- Repeat all above steps untill complete schema is formed.

Banking System E-R diagram

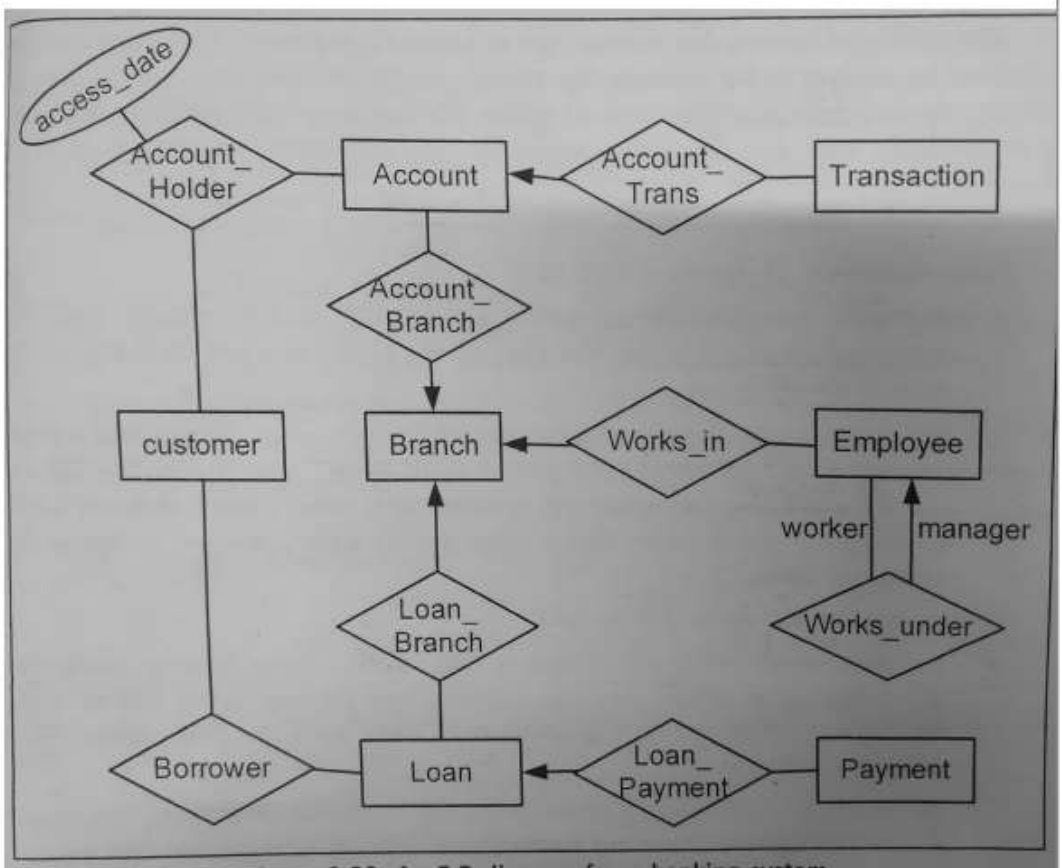


Figure 3.20: An E-R diagram for a banking system