CHAPTER-6

Entity – Relationship (ER) Model

❖ Introduction

- To design and develop database for any organization, it is necessary to understand data requirements of that organization.
- The E-R model is the most widely used high-level conceptual data model which represents data requirements in a concise manner.
- This chapter deals with various aspects of the E-R model.

❖ Basic E-R Concepts

- The E-R model employs three basic notions:
  - **Entities**
  - **Attributes**
  - **Relationships**
- These basic notions are described in following sub-sections in detail.

➢ Entity Sets

- **Entity**
- **Concrete v/s Abstract Entities**
- **Entity Set**
- **Disjoint ness**

▪ Entity

- "An entity is a 'thing' or 'object' existing in the real world that is identifiable from all other objects."
- For example, any customer of a bank is an entity. A customer, say 'ravina', exists in real world, as she is living thing.
- She can be distinguishable from other customers based on her customer id, name, address and other data.
- Also, she is different from an employee, an account, a loan, a transaction, or a branch.
- Similarly, each employee, account, loan, transaction and branch is also an entity.

▪ Concrete v/s Abstract Entities

- A concrete entity is a physically existing entity, such as a customer, an employee, or a book.
- An abstract entity has no physical existence, such as an account, a loan, or a subject.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- **Entity Set**

  - "An entity set is a set of entities of the same type."
  - All entities in an entity set share the same properties.
  - In a banking system, the set of all individual customers can be referred as an entity set *Customer.*
  - Similarly, an entity set *Employee* represents all individual employees of a bank.
  - So, a database for a banking application includes entity sets like *Customer, Employee, Account, Loan,* and *Branch.*
  - The individual entities that constitute an entity set are called *extension* or *member* of the entity set.

➢ Attributes

- **Definition**

  - "Attributes are descriptive properties of an entity set,"
  - For example, an entity set *Customer* has attributes like customer id, name, address and contact number denoted by *cid, cname, address, contact_no* respectively.

- **Value of an Attribute**

  - Each entity has a **value** for each of its attributes.
  - For example, a particular *customer* entity may have value 'C01' for *cid,* 'rahul' 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.
  - For example, a particular *customer* entity can be described by the set {*(cid,* 'C01'), *(cname,* 'rahul'), *(address,* 'unjha'), *(contact_no,9999999999*)}.

- **Domain of an Attribute**

  - "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.
  - For example, a bank is organized in three branches named 'unjha', 'mehsana' and 'kherva'.
  - So, a set of permitted values for an attribute *bname* is { 'unjha', 'mehsana', 'kherva'.}.
  - A value for a branch name must be one of these three values.
  - 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', or 'not known'.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- Not applicable means that value does not exist for the entity.
- Missing means that value does exist but it is unknown.
- Not known means that it is unknown whether or not the value actually exists.
- For example, a *null* value for customer name means that the value is missing, because, every customer must have a name.

➢ **Types of Attributes**

- Composite v/s Simple Attribute
- Single v/s Multi-valued Attribute
- Stored v/s Derived Attribute

- **Composite v/s Simple Attribute** :

  - Composite Attribute

    - **Composite attributes can be divided into smaller sub-parts.**
    - For example, the ***address*** attribute can be divided into sub parts such as *society, city, and pin-code.* Also, *society* can be further divided into *street number* and *society name.*
    - Composite attributes form a hierarchical structure as shown in figure 4.1

  - Simple Attribute

    - **Simple attributes cannot be divided into sub-parts.** Simple attributes are also referred as atomic attributes.
    - For example, *cid, ano, balance, bname* are simple attributes, as they cannot be divided into sub-parts.

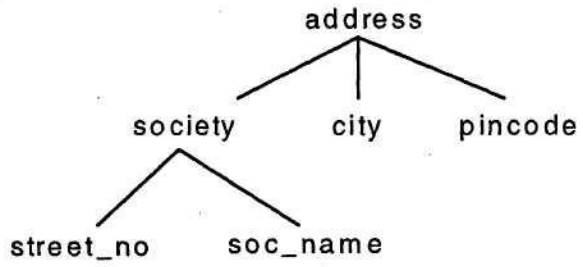


Figure 4.1 : Hierarchical structure of a composite attribute *address*

- **Single v/s Multi-valued Attribute** :

- Single valued Attribute

  - **Single valued attributes have single data value for a entity**.
  - For example, ***cid*** and ***ano*** are single valued attributes.
  - For any customer, there will be only one customer id.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- Similarly, for any account, there will be only one account number.

▪ Multi-value Attribute

- **Multi-valued attributes have multiple data values for a entity.**
- For example, ***contact no*** is a multi-valued attribute, any particular customer can have zero, one or more contact numbers.
- Multi-valued attributes may have lower and upper bound on the number of values.

➢ **Stored v/s Derived Attribute** :

▪ **The value for attribute can be derived from the values of other attributes is called derived attribute**.
▪ Consider two attributes ***age*** and brith_date for *Customer* entity set.
▪ Here, age can be calculated by using birth date and current date.
▪ So, *age* is considered as a derived attribute, **brith_date is considered as a stored attribute or base attribute.**

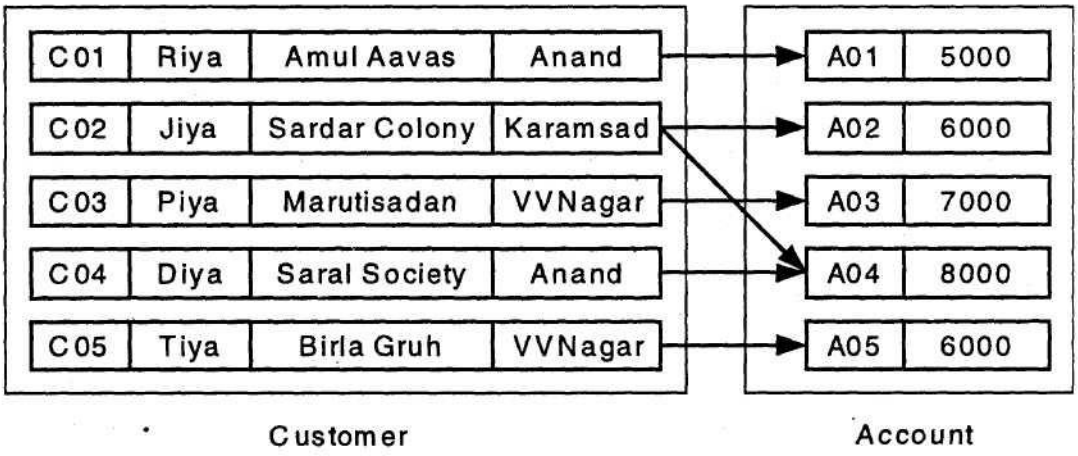
PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

➢ Relationships

- **Relationship**
  - **"A relationship is a connection between one or more entities."**
  - In banking application, each and every account is connected with some particular customer. This kind of connected is known as relationship.
  - For example, there is a relationship between a customer, say 'rahul', and some account, say account with account number 'A01'. This relationship indicates that 'A01' account is hold by a customer 'rahul'.

- **Relationship Set**

  - "A relationship set is a set of relationships of the same type."
  - For example, there is a relationship between each account and some particular customer.
  - The set of all such relationships, named *Account_Holder,* is known as a relationship set.
  - This relationship set is depicted in figure 4.2.



- **Participation**

  - The association between two entity sets is referred to as participation.
  - For example, entity sets *customer* and *account* participates in relationship *Account_Holder*

- **Relationship Instance**
  - While a relationship set associates two or more entity sets, a relationship instance associates particular entities.
  - For example, a particular *customer* entity named 'rahul' and *account* entity having number 'A01' participates in a relationship instance *Account_Holder*.
  - This relationship instance represents that, in real world, a bank customer named 'rahul' owns the account having number 'A01'.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- **Entity's Role**

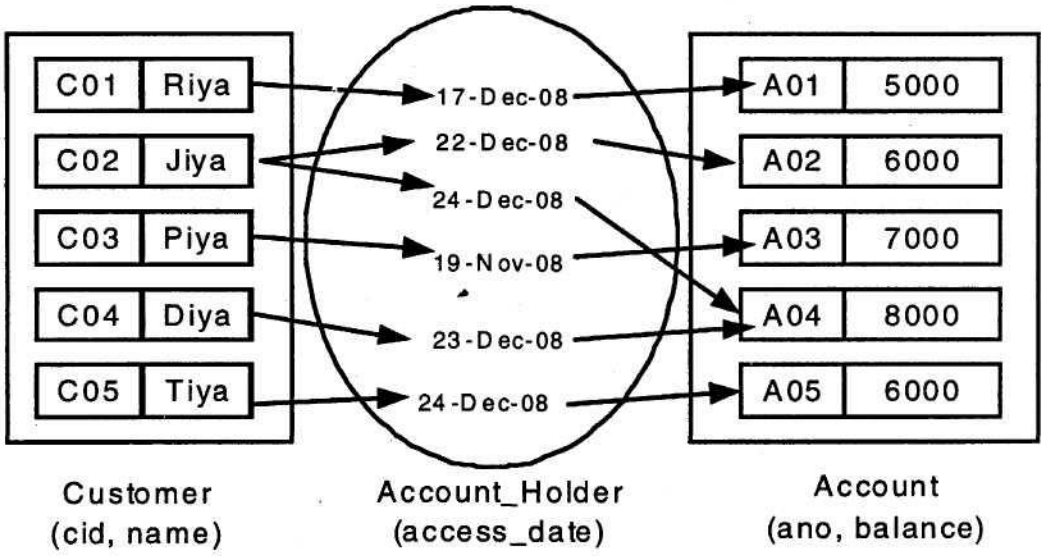
  - "The function that an entity plays in a relationship is called that entity's **role."**
  - When entity sets participating in relationship set are distinct, roles are implicit and need not to be specified.
  - But, in relationship sets, such as recursive relationship set, it is necessary to specify entity's role.

- **Recursive Relationship Set**

  - A relationship set is recursive if participating entity sets are not distinct. That is, the same entity set participates in a relationship set more than once.
  - Consider an entity set *employee* which represents individual employees of a bank.
  - Also, consider a relationship set *works_under* which represents relationships among employees, in a way that, which employee - as a *worker* - works under which another employee - as a *manager*.

- **Descriptive Attributes**

  - Like as an entity, a relationship may also have attributes. These attributes are called descriptive attributes.
  - Consider a relationship set *Account_Holder* with two entity sets *customer* and *account* participating in it.
  - This relationship associates each customer with its own account.
  - *Account_Holder* can have an attribute *access_date.*
  - This attribute specifies the most recent date on which an account is accessed by a customer.
  - The following figure 4.3 shows the relationship set *Account_Holder* with a descriptive attribute *access_date.* Limited attributes for entity set *Customer* is shown here.



PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- **Degree of the Relationship Set**

  - "The degree of the relationship set is the total number of entity sets participating in that relationship set."
  - For example, the degree of the relationship set *Account_Holder* is two, as, two entity sets - *customer* and *account* - participate in it.
  - A relationship set of degree 2 is called *binary* relationship set.

❖ Conversion of E-R Model into Relations

- The relational model uses a collection of tables to represent both *data* and the *relationships* among those data.
- In this model, entity sets and relationship sets of E-R schema are represented in form of tables.
- Each columns of a table corresponds to an attribute.
- Each row of a table represents individual entity or relationship of E-R model.
- This section describes mapping of E-R schema into a relational model
  - **Representation of String Entity Sets**
  - **Representation of Weak Entity Sets**
  - **Representation of Relationship Sets**

  - **Representation of String Entity Sets**

    - For each strong entity set, define a new table having columns same as attributes of an entity set.
    - The primary key for a table will be the same as that of an entity set.
    - For example, consider a relationship set *Loan_Payment* having two entity sets *Loan* and *Payment* participating in it.
    - Here, *Loan* is a strong entity set.
    - To represent this entity set, a new table called Loan is defined with columns *Ino* and *amount.*
    - The primary key for Loan table will be *Ino* as in entity set *Loan*

  - **Representation of Weak Entity Sets**

    - For each weak entity set, define a new table having columns same as attributes of an entity set.
    - The primary key for this table will be the combination of discriminating attribute of weak entity set and primary key of strong entity set.
    - For example, *Payment* is a weak entity set in relationship *Loan_Payment.*
    - A new table named Payment needs to be defined to represent this entity set having columns *Ino, pno, amount* and *pdate.*
    - This table contains one extra column *Ino.* And, primary key for this table will be combination of *Ino* and *pno* as shown in figure 4.21

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- **Representation of Relationship Sets:**

  - Relationship sets represent association (connection) among several entity sets.
  - To represent relationship sets in relational model, tables are used in a similar way to entity sets.
  - Formations of table for relationship set and its primary key depends upon the **mapping cardinality** of the relationship set.

  - This is described below
    - Consider a relationship set R having two entity sets E_1 and E_2 participating in R.
    - R has descriptive attributes a1, a_2 and a_3.
    - The primary keys of E, and E_2 are e1 and e_2 respectively.
    - There are two separate tables T_1 and T_2 corresponding to E_1 and E_2.

  - **If R is many-to-many relationship set...**
    - Define a separate table for R.
    - It will contain columns a_1 ,a_2 and a_3 corresponding, to each descriptive attribute of R. Along with this, it will have more columns corresponding to e_1 and e_2 - primary keys of entity sets E_1 and E_2.
    - The primary key for this table will be combination of e_1 and e_2.

  - **If R is one-to-many relationship set from E_1 to E_2...**
    - No need to define a separate table.
    - Use common column to link tables T1 and T2 . As a common column, primary key of entity set at "one" side, is used.
    - As R is one-to-many from E1 to E_2, here, primary key of E1 will be used as a common column. So, table T_2 will have one more column in form of e1.

  - **If R is many-to-one relationship set from E_1 to E_2...**
    - Similar to above case, no need to define a separate table.
    - As R is many-to-one from E_1 to E_2, here, primary key of E_2 will be used as a common column. So, table T_1 will have one more column in form of e_2.

  - **If R is one-to-one relationship set...**
    - No need to define a separate table.
    - Use common column to link tables T_1 and T_2. As a common column, primary key of either entity set is used.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

No.	Table	Columns
1.	Branch	**bname**, asset, baddress
2.	Customer	**cid,** cname, address, contact_no
3.	Account	**ano,** balance, *bname*
4.	Transaction	**tno,** type, amount, tdate, *ano*
5.	Loan	**lno,** amount, *bname*
6.	Payment	***lno,*** **pno,** amount, pdate
7.	Employee	**eid,** ename, address, contact_no, post, salary, *mngr_id*, *bname*
8.	Account_Holder	***cid, ano,*** access_date
9.	Borrower	***cid, lno***

Figure 4.21 : Tables and related Columns of database for a banking system.

❖ Problems with E-R Models

- E-R diagrams provide the base for developing a database for any organization. So, enough care must be taken while constructing E-R diagrams.
- If the meanings of relationships are misinterpreted, they lead to problems. These problems are called **connection problems.**
- There are mainly two types of connection traps:
  - **Fan problems**
  - **Chasm problems**

  - **Fan problems (Traps)**

    - Fan traps occur when two-relationship sets, having mapping cardinality **one to many,** converge to single entity set.
    - **Problem** :
      - Consider an engineering college which operates different branches such as computer engineering, information technology etc.
      - Different kind of employees, such as lecturer, clerk, peon, etc. works in college.
      - Also, these employees belong to specific branches.
      - E-R diagram showing relationship-sets between Employee & College and College & Branch.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

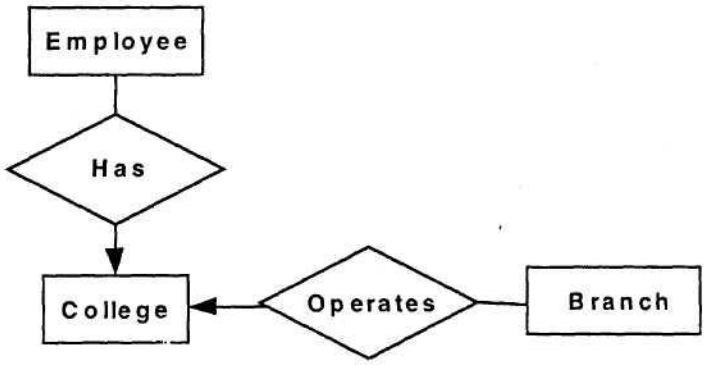


Figure 4.22 : An example of Fan Trap

- This diagram indicates that a college has more than one employees and it operates more than one branch.
- The two relationship sets - Has and Operates - are of one-to-many types from *College* to *Employee* and from *College* to *Branch* respectively.
- Also, these both relationship sets converge to single entity set *College.*
- **Now the problem is**:
  - This diagram cannot specify which employee works in which branch.
  - For example, it is difficult to answer to the question: "In which branch an employee having employee id 'E01' works?"
- Such kind of problem is known as Fan trap.

- **Solution** :

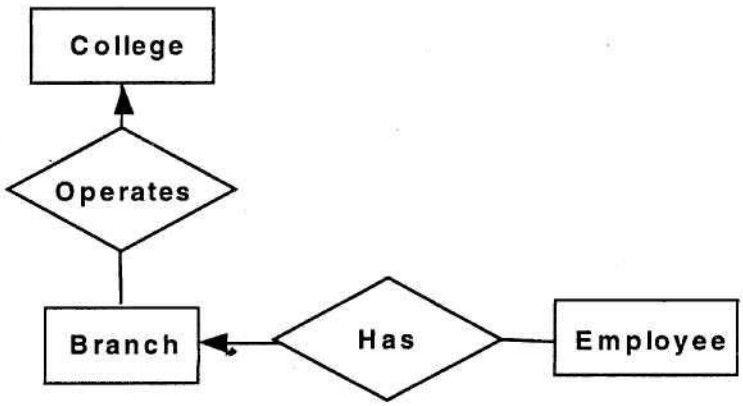


Figure 4.23 : Restructured E-R diagram to eliminate Fan trap.

- This restructured E-R diagram is able to give the correct answer of the above question. And so, it doesn't have any Fan traps

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

➢ Chasm Traps

- Chasm traps occur when an E-R diagram suggests the existence of a relationship between entity sets, but, in reality, the connection does not exist.
- This is possible when a relationship set with partial participation forms the way of connection between entity sets.
- This way of connection is referred as *pathway*.
- **Problem** :
  - Consider the following E-R diagram of figure 4.24

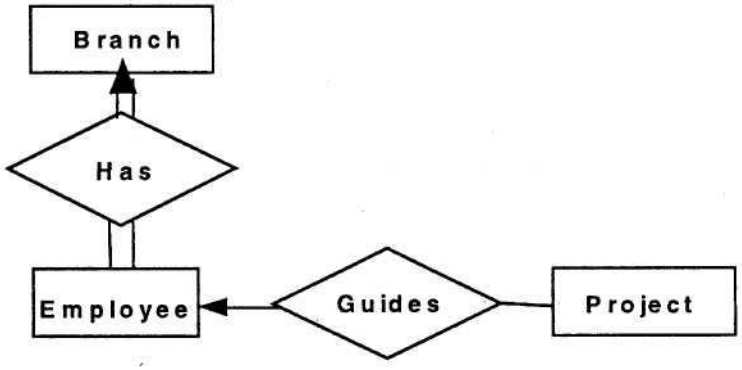


Figure 4.24 : An example of Chasm trap

- **Problem** :
  - In this E-R diagram, total and partial participations are represented explicitly.
  - The relationship set Guides represents that *Employee* (mostly, a lecturer) provides guidance to *Project*. But, all employees do not guide to project. So, participation of *Employee* into relationship set Guide is partial. Also, all projects do not need to be guided by some employee. This means, students can take projects as a self-guided. So, participation of *Project* into relationship set Guide is partial.
  - **Now the problem is**:
    - ♦ If any project is not associated with any employee, it cannot be associated with its branch.
    - ♦ For example, it is difficult to answer to the question: "In which branch a particular project, having project id 'P01' is carried out, if it is self-guided project ?"
    - ♦ Such kind of problem is known as Chasm trap.

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

- **Solution** :

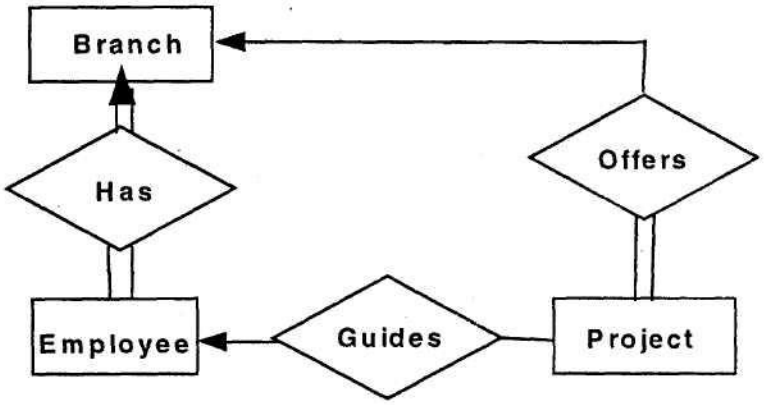
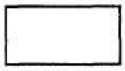
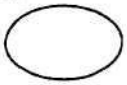
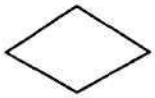


Figure 4.25 : Restructured E-R diagram to eliminate Chasm trap.

- Here, the missing link is 'Offers' relationship set between *Branch* and *Project* entity sets. This restructured E-R diagram is able to give the correct answer of the above question And so, it doesn't have any Fan traps

❖ E-R Diagram Symbols

Symbol	Notation	Represents...
	Rectangle	An entity set
	Ellipse	An attribute
	Diamond	A relationship set
	Line	Link between an entity set & attribute, and Link between an entity set & a relationship set

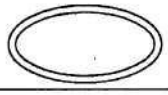
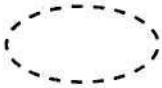
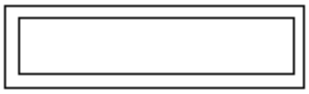
Symbol	Notation	Represents...
	Double Ellipse	A multi-valued attribute
	Dashed Ellipse	A derived attribute

Figure 4.8 : An E-R diagram: Notations for different types of attributes

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)

Symbol	Notation	Represents
	Double lines	**Indicate total participation of an entity in a relation ship set**
	**Double rectangles**	**Represent weak entity sets**

PREPARED BY: RAHUL PATEL
LECTURER IN COMPUTER DEPARTMENT (BSPP)