UNIT 2:
DATABASE DESIGN

---

1. DEFINE FOLLOWING TERMS.

ER-Diagram

- Entity-Relationship diagram is graphical representation of database.
- An ER diagram shows the relationship among entity sets Entity.

Entity

- An entity is an object or component of data.
- It is anything in the enterprise that to be represented in our database.
- An entity is a person, a place or any object.
- E.g. Student , Bank Account

Attribute

- An attribute describes the property of an entity.
- A property or characteristic of an entity called attribute.
- E.g. Rollno, name, age etc

Relationship

- Relationship in ER Diagram shows an association between two or more entities.

Strong Entity

- A Strong entity is nothing but an entity set having a primary key attribute or a table that consists of a primary key column.

Weak Entity

- A weak entity is a type of entity, which does not have its key attribute.

Normalization

- Normalization is the process of removing redundant data from tables to improve data integrity, scalability and storage efficiency.

Ternary Relationship

- When there are, exactly three entity sets participating in a relationship then such type of relationship called ternary relationship.

Super class

- An entity type that represents a general concept at a high level, is called superclass.

Sub class

- An entity type that represents a specific concept at lower levels, is called subclass.
- The subclass is said to inherit from superclass.
- When a subclass inherits from one or more super classes, it inherits all their attributes

Generalization

- In generalization, two or more entities of lower level combine to form a higher-level entity if they have some attributes in common.

Specialization

- Specialization is a "Top-down approach" where higher level entity is specialized into two or more lower-level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.

2. EXPLAIN ER-MODEL HISTORY.

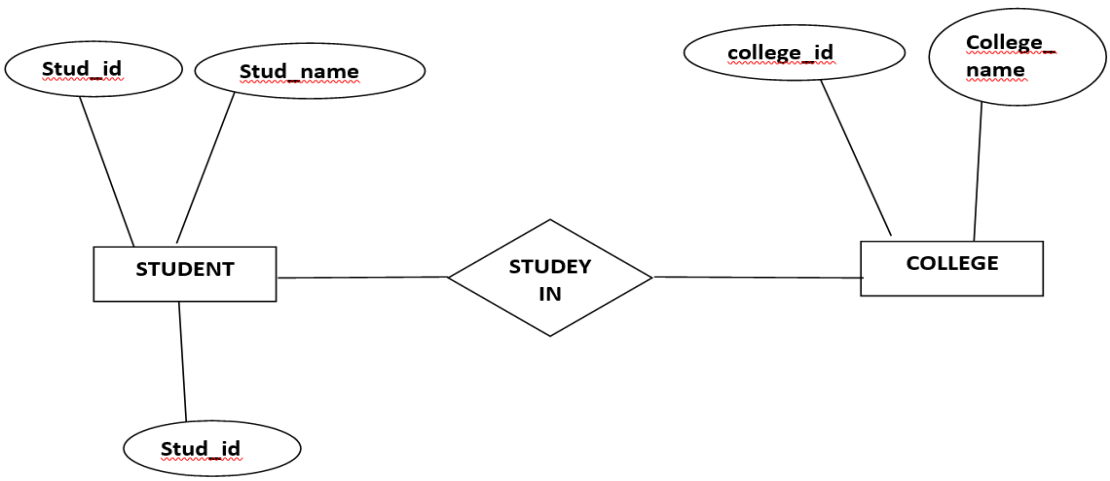
- Peter Chen developed ERDs in the 1970s .
- Peter Chen was a computer scientist who worked on improving database design.
- He published his proposal for entity relationship modeling in a 1976 paper titled "The Entity-Relationship Model: Toward a Unified View of Data".
- His entity relationship model was a way to visualize a database that unified other existing models into a single understanding.

3. WHAT IS ER-DIAGRAM? WRITE DOWN ITS USE.

- E-R diagram - (Entity-Relationship diagram) is graphical (pictorial) representation of database.
- ER diagram allows you to draw Database Design.
- It is widely used in Database Design.
- It is an easy-to-use graphical tool for modeling data
- It is a GUI representation of the logical structure of a Database.
- It uses different types of symbols to represent different objects of database.

ER-DIAGRAM USAGE:

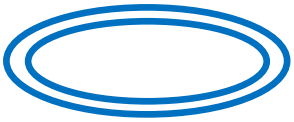
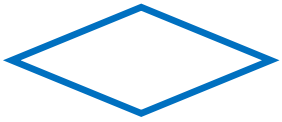
- Helps you to define terms related to entity relationship modeling.
- Provide a preview of how all your tables should connect
- Helps to describe entities, attributes, relationships
- Allows you to communicate with the logical structure of the database to users
- ER diagrams can be used by database designers as a blueprint
- ER diagrams are translatable into relational a table which allows you to build databases quickly.

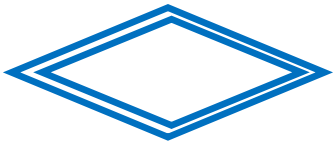


4. EXPLAIN ER-DIAGRAM WITH ITS SYMBOL.

- E-R diagram – Entity Relationship Diagram is graphical representation of database.
- ER model allows you to draw Database Design.
- It is a GUI representation of the logical structure of a Database.
- It uses different types of symbols to represent different objects of database.

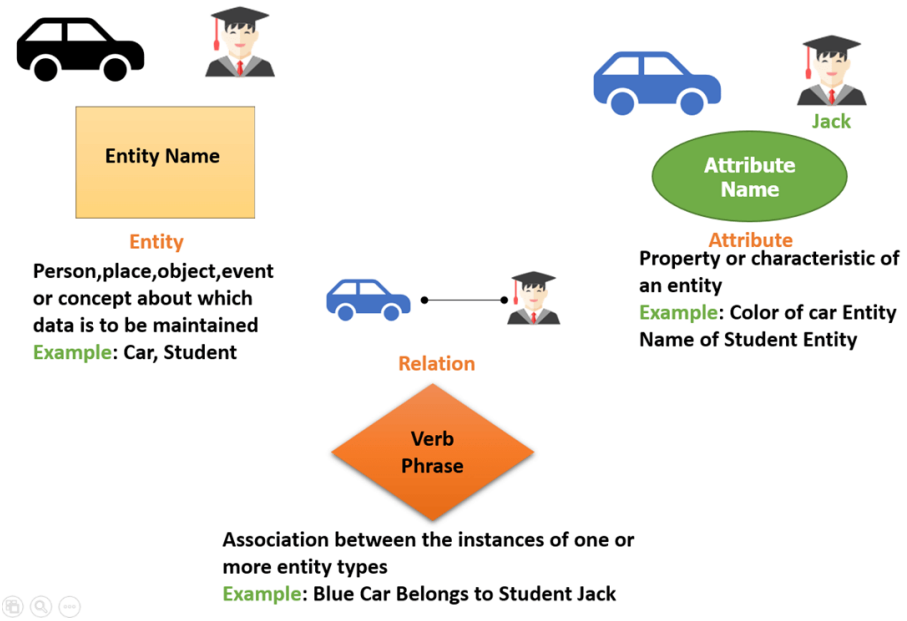
ER- Diagram Symbols:

Symbol	Shape	Name
	Rectangle	Entity / Entity Set
	Double Rectangle	Weak Entity
	Ellipse	Attribute
	Double Ellipse	Multi Valued Attribute
	Dashed Ellipse	Derived Attribute
	Diamond	Relationship

	Double Diamond	Weak Relationship
	Ellipse with Dashed line	Discriminating Attribute
	Line	Link to relate attributes & relationship
	Double Line	Total participation

5. EXPLAIN ER DIAGRAM COMPONENTS WITH EXAMPLE.

- E-R diagram – Entity Relationship Diagram is graphical representation of database.
- It helps you to define terms related to entity relationship modeling.
- It Provide a preview of how all your tables should connect, what fields are going to be on each table.
- This model is based on three basic concepts:
  - Entities
  - Attributes
  - Relationships



Entity

- An entity is an object or component of data.
- It cans either tangible or any non-tangible things also.
- An entity represented by a rectangle, which contains the name of an entity.

- It is a set (group) of entities of same type.
- Example:
  - **Person:** Employee, Student, Patient
  - **Place:** Store, Building
  - **Object:** Machine, product, and Car
  - **Event:** Sale, Registration, Renewal
  - **Concept:** Account, Course

Attribute

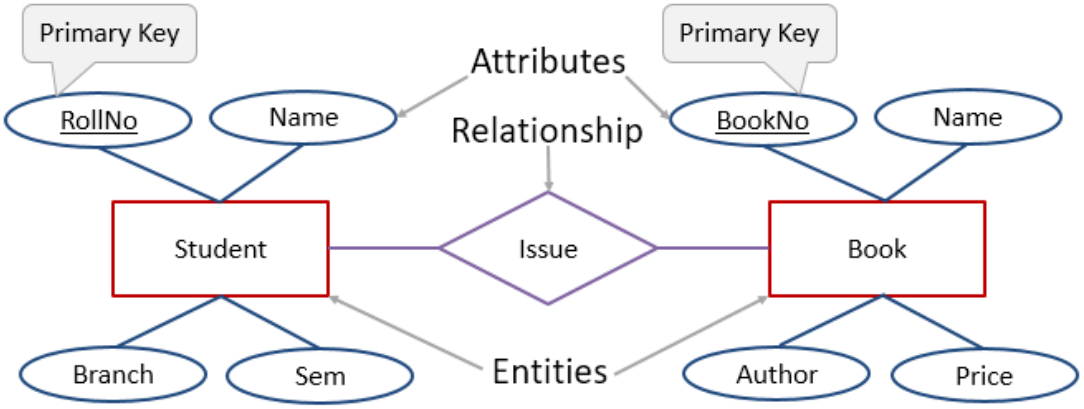
- Attribute is properties or details about an entity.
- An attribute represented by an oval containing name of an attribute.
- Example:
  - **Attributes of Student are:** Roll No, Student, Name, Branch, Semester, Address, Mobile No, Age
  - **Attributes Of A Lecture:** Time, Date, Duration, Place,

Relationship

- Relationship is an **association (connection)** between several entities.
- It should place between two entities and a line connecting it to an entity.
- A diamond containing relationship's name represents a relationship.
- Example:
  - Student & book have relation like student can use a book & book is issued to the student for reading.

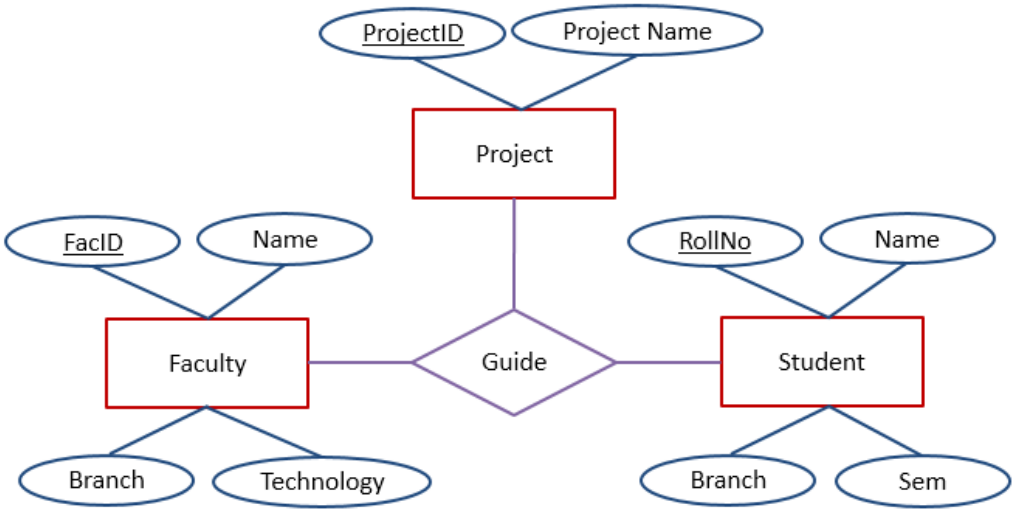
E-R Diagram of a Library System & Binary Relationship

- Each entity must have one primary key attribute.
- Relationship between 2 entities is called binary relationship.



Ternary Relationship:

- Relationship between 3 entities is called ternary relationship



6. EXPLAIN ENTITY TYPES.

- An entity is a person, a place or an object.
- An entity is represent by a rectangle, which contains the name of an entity.
- Entity can be Tangible or intangible.
- **Tangible Entity:**
  - Tangible Entities are those entities, which exist in the real world physically.
  - **Example:**
    - Person / student/ employee
    - Car
    - Product
    - Locker in bank
- **Intangible Entity:**
  - Intangible Entities are those entities, which exist only logically and have no physical existence.
  - **Example:**
    - Bank Account
    - Order
    - Location

Type of Entity:

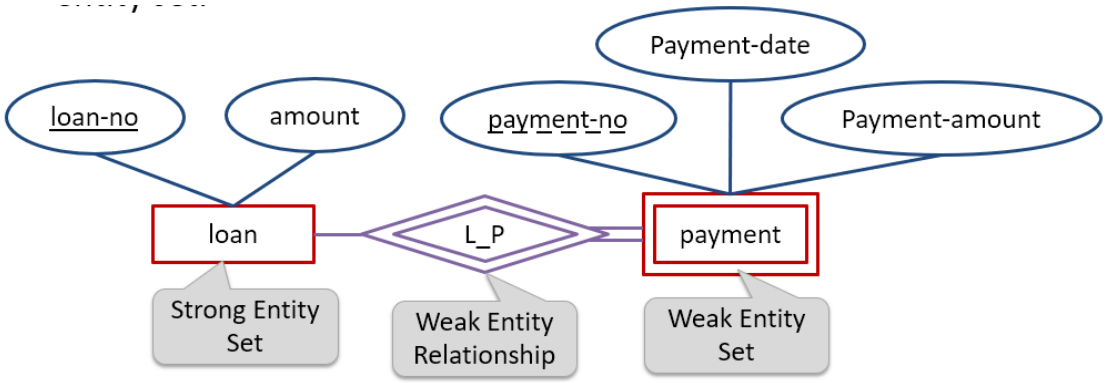
- There are two types of entity.
  - Strong entity
  - Weak entity

Strong Entity:

- Strong entity is those entity types, which has a key attribute.
- The primary key helps in identifying each entity uniquely.

Weak Entity:

- A weak entity is an entity set that do not have sufficient attributes for unique identification of its records.
- Weak entity do not have primary key.



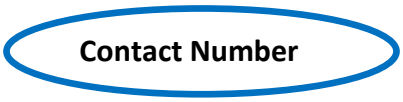
- The existence of a weak entity set depends on the existence of a strong entity set.
- The discriminator (partial key) of a weak entity set is the set of attributes that distinguishes all the entities of a weak entity set.
- The primary key of a weak entity set is created by combining the primary key of the strong entity set on which the weak entity set is existence dependent and the weak entity set's discriminator.
- We underline the discriminator attribute of a weak entity set with a dashed line.

7. EXPLAIN TYPES OF ATTRIBUTES.

- Attribute is properties or details about an entity.
- Attributes are divided into following types:
  - Simple attribute
  - Composite attribute
  - Stored attributes
  - Derived attribute
  - Single-valued Attribute
  - Multi valued attribute

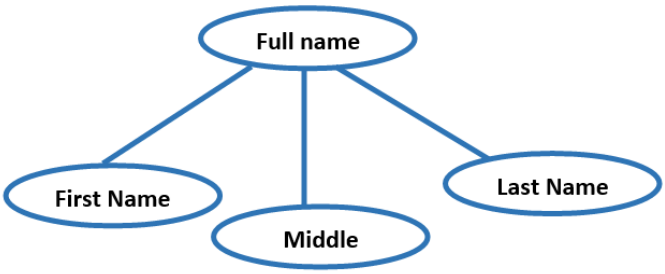
Atomic / Simple Attribute

- An attribute that is not divide further known as atomic/ simple attributes.
- For example, a student's contact number.
- **Symbol:**



Composite Attribute

- An attribute that is further divide into sub attribute called composite attribute.
- For example, a student's full name may further divided into first name, second name, and last name.
- **Symbol:**



Stored Attributes

- An attribute whose value is stored manually in database
- E.g. Birthrate
- **Symbol :**

Derived Attributes

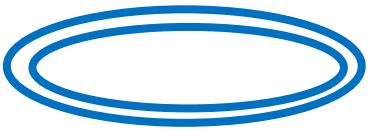
- This type of attribute does not include in the physical database.
- Its values derived from other attributes present in the database.
- For example, age
- **Symbol :**

Single Valued Attributes

- Attributes have a single value for a particular entity
- For example, a student can have more than one Email Id
- **Symbol :**

Multivalve Attributes

- Multivalve attributes can have more than one value for a particular entity.
- For example, a student can have more than one Email Id
- **Symbol :**



8. WHAT IS RELATIONSHIP? EXPLAIN ITS TYPE.

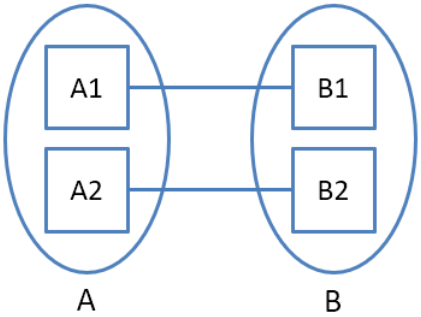
OR

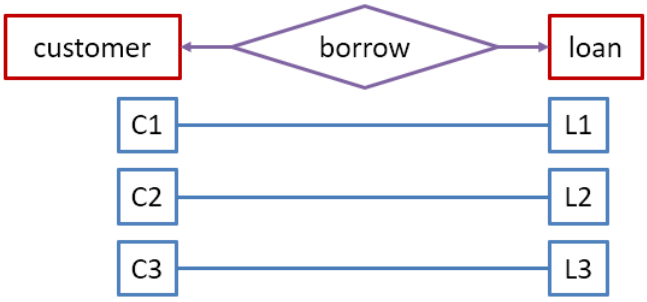
EXPLAIN MAPPING CARDINALITY

- A relationship type represents the association between entity types.
- It is most useful in describing binary relationship sets.
- In ER diagram, relationship type is represented by a diamond and connecting the entities with lines.
- There are four types of relationship.
  - One To One
  - One To Many
  - Many To One
  - Many To Many

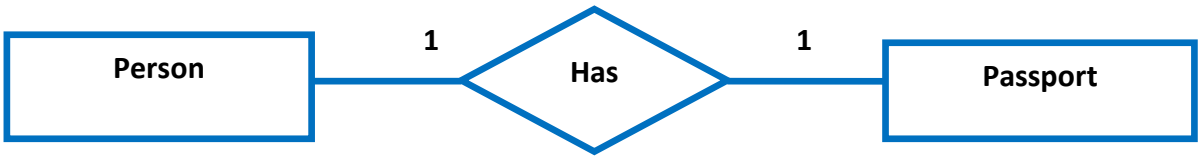
One To One *(1 – 1):*

- An entity in A is associated with only one entity in B and an entity in B is associated with only one entity in A.
- Example:
  - A customer is connected with only one loan using the relationship borrower and a loan is connected with only one customer using borrower.



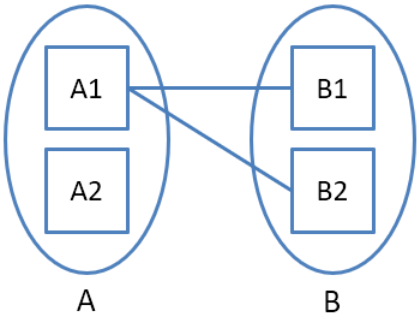


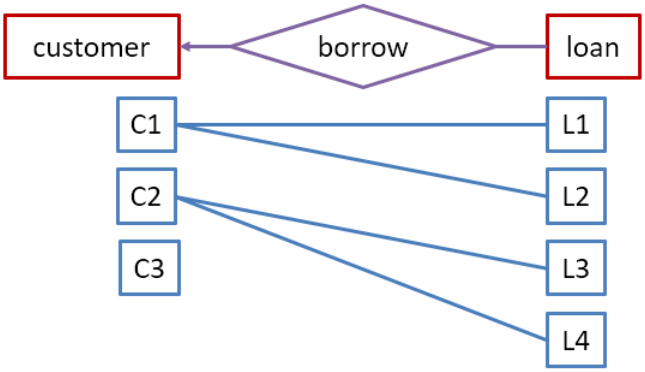
- Some real word example:
  - a person has only one passport and a passport
  - a department has only one department head



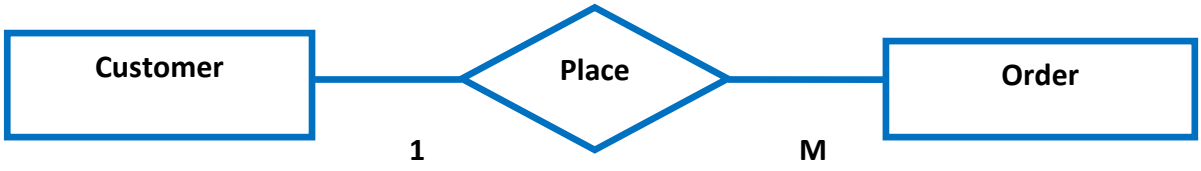
One to Many *(1 – M):*

- An entity in A is associated with more than one entity in B and an entity in B is associated with only one entity in A.
- Example:
  - A loan is connected with only one customer using borrower and a customer is connected with more than one loans using borrower.



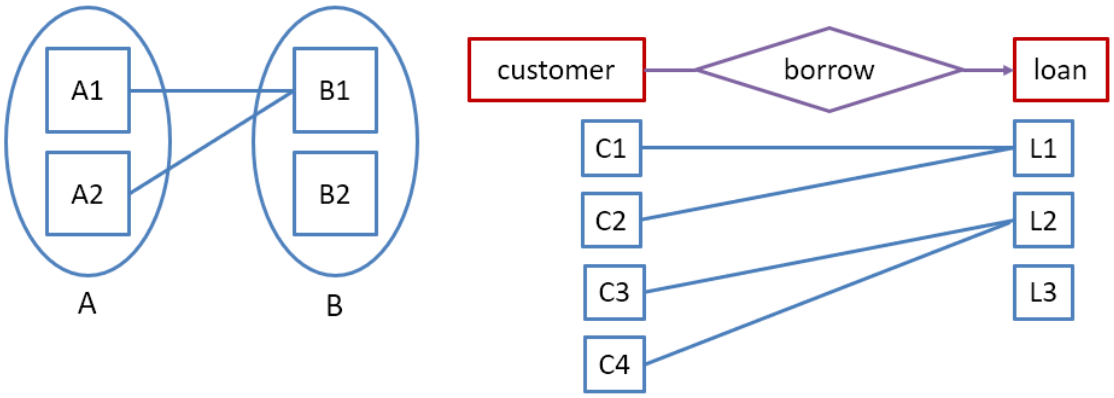


- Some other examples:
  - one customer can place many order
  - one doctor can treat many patients
  - one teacher can teach many subjects

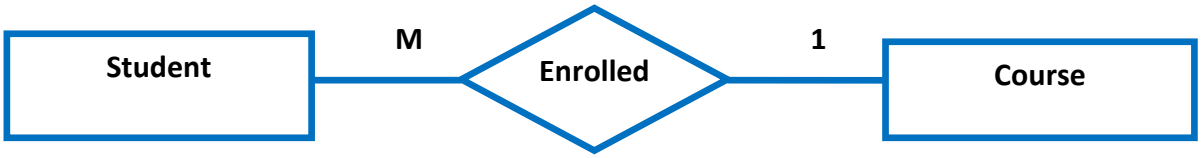


Many to One *(M – 1):*

- An entity in B is associated with more than one entity in A and an entity in A is associated with only one entity in B.
- Example:
  - A loan is connected with more than one customer using borrower and a customer is connected with only one loan using borrower.



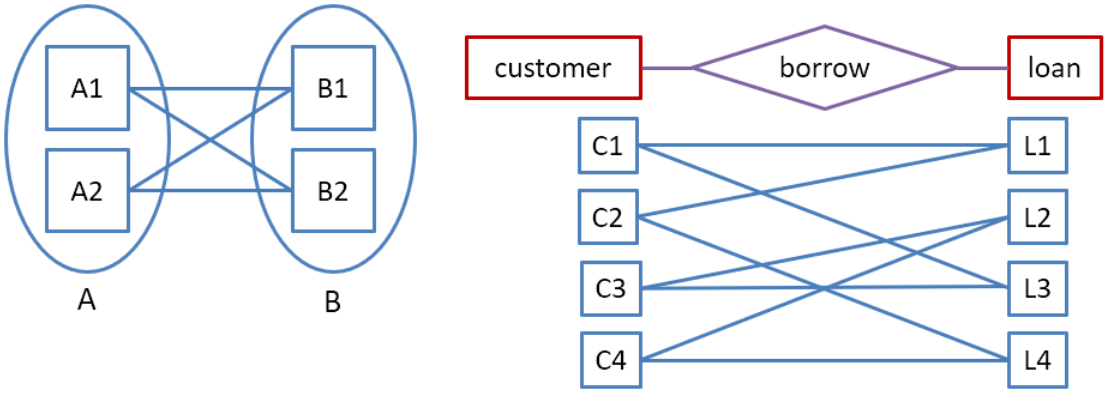
- Example : 2
  - one course there can be n students



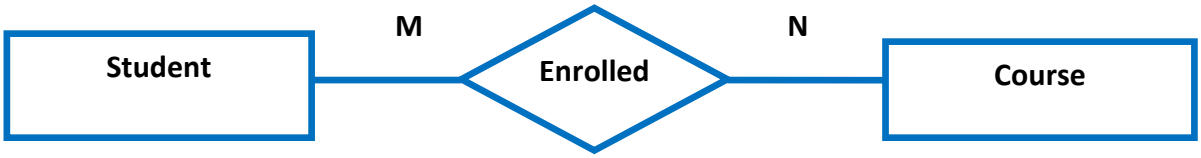
Many to Many *(N – N):*

- An entity in A is associated with more than one entity in B and an entity in B is associated with more than one entity in A.
- **Example:**

- A customer is connected with more than one loan using borrower and a loan is connected with more than one customer using borrower.



- For example.
  - student can take more than one course and one course can be taken by many students



9. WRITE A NOTE ON EER.

- EER stands for extended/enhanced entity relationship model.
- ER model is supported with the additional semantic concepts is called as Extended / Enhanced Entity Relationship Model .
- EER creates a design more accurate to database schemas.
- It reflects the data properties and constraints more precisely.
- It is used to represent a collection of objects that is union of objects of different of different entity types.
- Three new concepts added in ER:
  - Generalization
  - Specialization
  - Aggregation

Generalization

- It's a bottom-up approach
- Extract common properties from a set of entities and creates generalized entities from it.
- In generalization, two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- This approach creates a more generic entity, known as a superclass, by combining entities with similar features.
- sub-classes are combined to form a super-class.

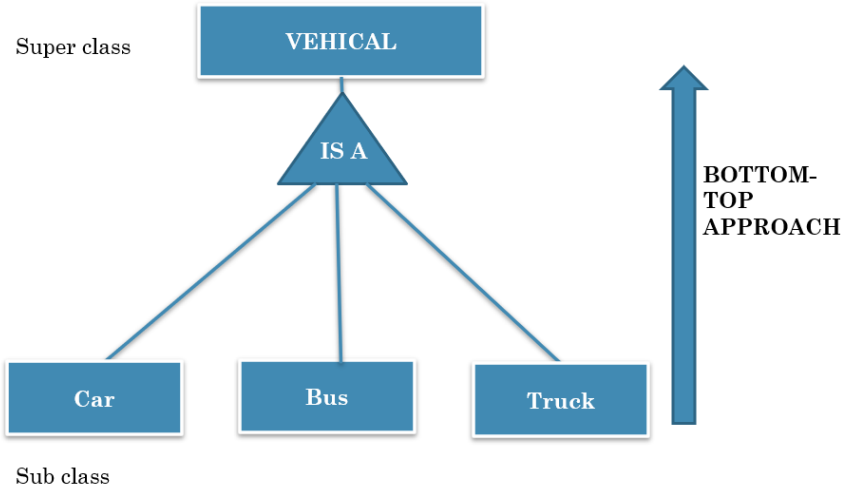
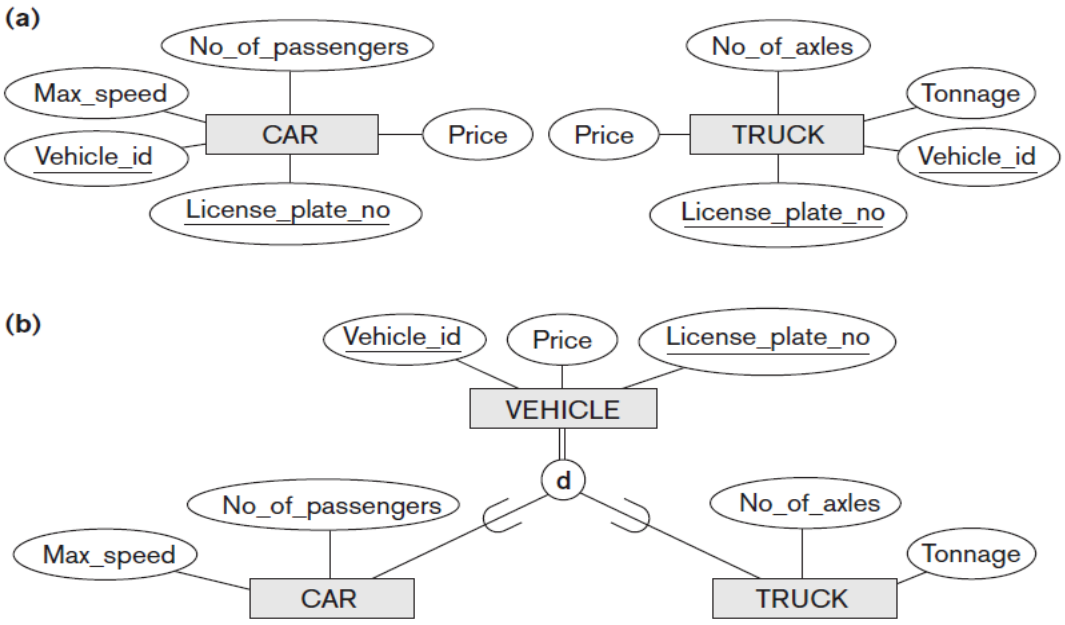


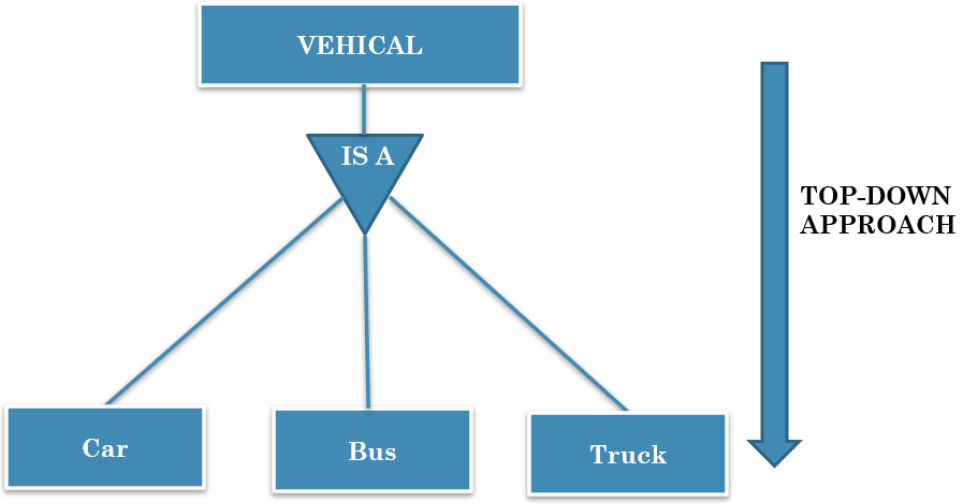
Figure 4.3
Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.



Specialization

- Specialization is opposite of Generalization.
- In Specialization, an entity is broken down into sub-entities based on their characteristics.
- Specialization is a "Top-down approach" where higher level entity is specialized into two or more lower level entities.

- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Specialization can be repeatedly applied to refine the design of schema.



10. EXPLAIN FUNCTIONAL DEPENDENCY IN DETAIL.

- The functional dependency is a relationship that exists between two attributes.
- Introduced by E. F. Codd, it helps in preventing data redundancy and gets to know about bad designs.
- Functional Dependency is represented by **->** (arrow sign)
- let us consider P is a relation with attributes **A** and **B**.

A - > B

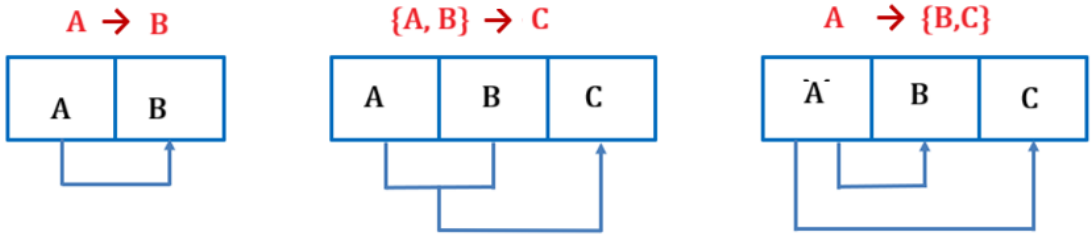
- The left side of this arrow is a Determinant.
- The right side of this arrow is a Dependent.
- In above example:
  - **A** is called **Determinant**, **B** is called the **Dependent** and B is functionally dependent on A
- Lets take one more example:
  - Here department table has two fields deptno and dept_name where deptno is primary key

Deptno	Dept_name
011	Marketing
022	HR
033	Finance
044	Accounting
055	Sales
066	Telecom

  - So if you want to know department name you need to know deptno.
  - So it is denotated by:

Deptno - > Dept_name

- Diagrammatic representation



Types of functional dependency:

- Following are the types of functional dependency
  - Full Functional Dependency
  - Partial Functional Dependency
  - Transitive Functional Dependency
  - Trivial Functional Dependency
  - Nontrivial Functional Dependency

Full Functional Dependency

- An attribute is fully functional dependent on another attribute, if it is Functionally Dependent on that attribute and not on any of its proper subset(attribute).
- Example:

supplier_id	item_id	price
1	1	540
2	1	545
1	2	200
2	2	201
1	1	540
2	2	201

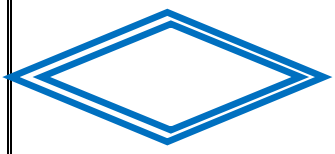
- Price is fully functionally dependent on {supplier_id, item_id}.

{supplier_id, item_id}→ price

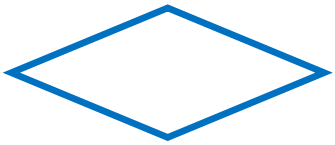
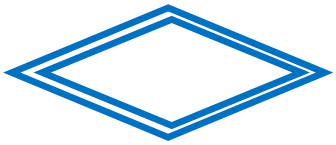
11. EXPLAIN WEAK ENTITY WITH EXAMPLE.

- A weak entity is an entity set that do not have sufficient attributes for unique Identification of its records.
- A weak entity does not have primary key.
- A weak entity is depending on strong entity to ensure its existence.
- A set of weak entity known as weak entity set.
- It **contains** a partial key called a discriminator, which helps in identifying a group of entities from the entity set.

Symbol/ representation

	Double rectangle	Weak Entity Set
	Double diamond	Weak Relationship
	Dashed line	Discriminating Attribute
	Line	Link to relate attributes & relationship
	Double line	Total participation

12. WRITE DOWN THE DIFFERENCE BETWEEN STRONG ENTITY & WEAK ENTITY.

Strong entity	Weak entity
Strong entity always has a primary key	Weak entity do not have a primary key but it has partial discriminator key
It is not dependent on any other entity	Which entity is dependent on the strong entity
Represented by a single rectangle 	Represented by a double rectangle 
A single diamond represents relationship between two strong entities. 	Represented by double rectangle relationship between a strong entity and the weak entity represented by double Diamond. 
A strong entity may or may not have total participation.	It has always total participation

13. WHAT IS NORMALIZATION? EXPLAIN 1NF, 2NF AND 3NF.

- Normalization is the process of organizing the data in the database.
- Normalization is use to minimize the redundancy from a relation or set of relations.
- Normalization is the process of organizing data in a database.
- Normalization helps to minimize data redundancy.
- The normal form is use to reduce redundancy from the database table.

Normal forms:

- Normal forms are:
  - **1NF (First Normal Form)**
  - **2NF (Second Normal Form)**
  - **3NF (Third Normal Form)**
  - **BCNF (Boyce–Codd normal form)**
  - **4NF (Forth normal form)**
  - **5NF (Fifth normal form)**

1NF (First Normal Form)

- A relation R is in first normal form (1NF) if and only if
  - Contains only atomic /single values
  - There are no **composite attribute** or multi-valued attributes

Customer ID	Name	Address
C01	Raj	Jamnagar Road, Rajkot
C02	Meet	Nehru Road, Jamnagar

- In above relation **address** is **composite attribute** which is further divided into sub-attributes as “Road” and “City”.
- So above relation is not in 1NF.
- Solution:
- **Divide composite attributes** into **number of sub-attributes** and insert value in proper sub-attribute

Customer ID	Name	Road	City
C01	Raj	Jamnagar Road	Rajkot
C02	Meet	Nehru Road	Jamnagar

2NF (Second Normal Form)

- The second normal form must satisfy 2 conditions:
  - The table should be in first normal form.
  - There should be no partial dependency.
- Example:

TEACHER_ID	SUBJECT	TEACHER_AGE
111	JAVA	38
111	C++	38
222	PHP	36
333	DBMS	40
333	PHP	40

- This table is in 1NF
- It not in 2NF because non prime attribute Teacher_Age is dependent on Teacher_id

- Solution: split down into two table based on prime attribute

Teacher details

TEACHER_ID	TEACHER_AGE
111	38
222	36
333	40

Teacher Subject

TEACHER_ID	SUBJECT
111	JAVA
111	C++
222	PHP
333	DBMS
333	PHP

3NF (Third Normal Form)

- The third normal form must satisfy 2 conditions:
  - It should be in the Second Normal form.
  - In addition, it should not have Transitive Dependency.
- Example:

RollNo	State	City
101	punjab	Maholi
102	gujarat	baroda
103	gujarat	Surat
104	punjab	Maholi
105	Nayan	Baroda
106	Bihar	Maholi

- Primary key: rollno
- Here the state →determine by city
- State is non prime so its called transitive dependency

14. WHAT IS ANOMALY IN DBMS? EXPLAIN WITH EXAMPLE.

- Anomalies are problems that can occur in poorly planned, un-normalized database where all the data are stored in one table.
- There are three types of anomalies that can arise in the database because of redundancy are
  - Insert anomaly
  - Delete anomaly
  - Update / Modification anomaly

Insert Anomaly:

- An insert anomaly occurs when certain attributes cannot be inserted into the database without the presence of another attribute.
- Consider a following relation:
  - emp_dept (E#, Ename, Address, D#, Dname, Dmgr#) E# as a primary key

E#	Ename	Address	D#	Dname	Dmgr#
1	Raj	Rajkot	1	CE	1
2	Meet	Surat	1	CE	1

Want to insert new department detail (IT)

- Suppose a new department (IT) has been started by the organization but initially there is no employee appointed for that department.
- We want to insert that department detail in emp_dept table.
- But the tuple for this department cannot be inserted into this table as the E# will have NULL value, which is not allowed because E# is primary key.
- This kind of problem in the relation where some tuple cannot be inserted is known as insert anomaly.

Delete Anomaly:

- A delete anomaly exists when certain attributes are lost because of the deletion of another attribute.
- Consider a following relation:

Want to delete Meet employee's detail

E#	Ename	Address	D#	Dname	Dmgr#
1	Raj	Rajkot	1	CE	1
2	Meet	Surat	1	IT	2
3	EKTA	RAJKOT	1	CE	1
4	NIHAR	SURAT	1	CE	1

- Now consider there is only one employee in some department (IT) and that employee leaves the organization.
- So we need to delete tuple of that employee (Meet).
- But in addition to that information about the department also deleted.
- This kind of problem in the relation where deletion of some tuples can lead to loss of some other data not intended to be removed is known as delete anomaly.

Update Anomaly:

- An update anomaly exists when one or more records (instance) of duplicated data is updated, but not all.
- Consider a following relation:

Want to update CE department's manager

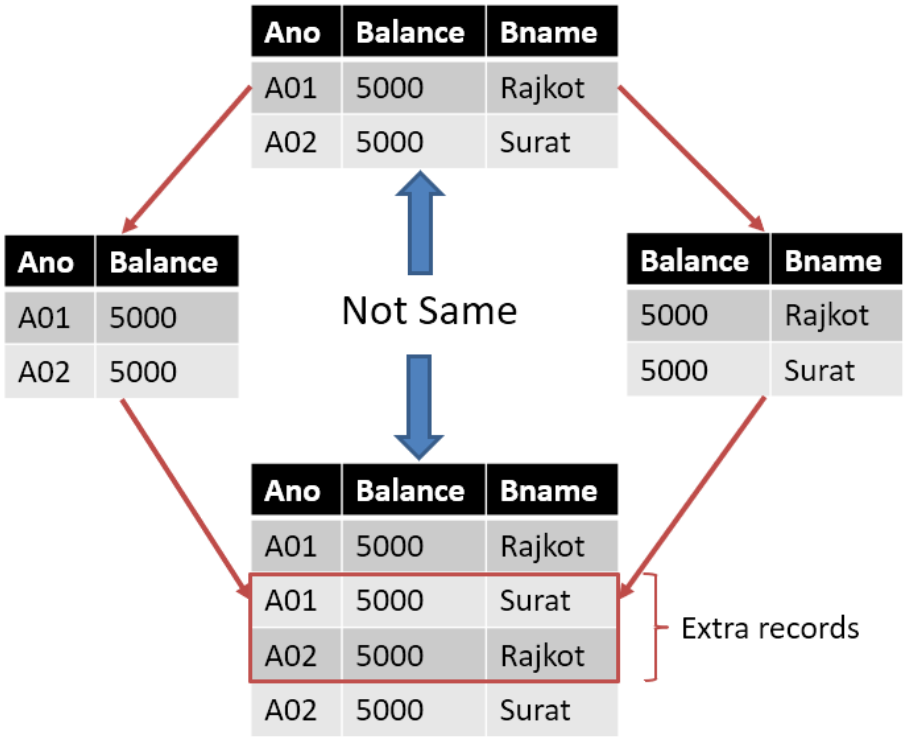
E#	Ename	Address	D#	Dname	Dmgr#
1	Raj	Rajkot	1	CE	M1
2	Meet	Surat	2	IT	M2
3	Jay	Rajkot	2	C.E	M2

- Suppose the manager of a (CE) department has changed, this requires that the Dmgr# in all the tuples corresponding to that department must be changed to reflect the new status.
- If we fail to update all the tuples of given department, then two different records of employee working in the same department might show different Dmgr# lead to inconsistency in the database.

15. WRITE A NOTE ON DECOMPOSITION PROCESS.

- Decomposition is the process of breaking down given relation into two or more relations.
- Relation R is replaced by two or more relations in such a way that:
  1. Each new relation contains a subset of the attributes of R
  2. Together, they all include all tuples and attributes of R
- Types of decomposition
  - Lossy decomposition
  - Lossless decomposition (non-loss decomposition)

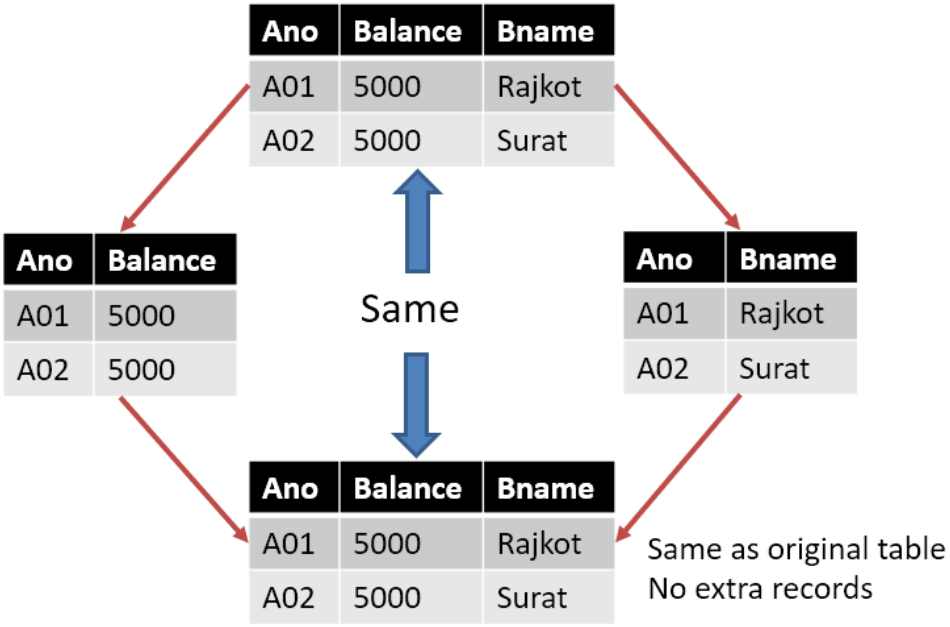
Lossy decomposition:



- The decomposition of relation R into R1 and R2 is lossy when the join of R1 and R2 does not yield the same relation as in R.
- This is also referred as lossy-join decomposition.
- The disadvantage of such kind of decomposition is that some information is lost during retrieval of original relation.

- From practical point of view, decomposition should not be lossy decomposition.

Lossless decomposition:



- The decomposition of relation R into R1 and R2 is lossless when the join of R1 and R2 produces the same relation as in R.
- This is also referred as a non-additive (non-loss) decomposition.
- All decompositions must be lossless.