

# CSE301 – DATABASE

## Entity-Relationship Model

# What is Data Modelling?

---

- Data modeling (data modelling) is the process of creating a **data model** for the data to be stored in a database.
- This **data model** is a **conceptual representation** of Data objects, the **associations** between different data objects, and the **rules**.
- Data modeling helps in the visual representation of data and enforces business rules, regulatory compliances, and government policies on the data.

# What is Data Model?

---

The data model is defined as an **abstract model** that organizes **data description**, **data semantics**, and consistency **constraints** of data.

Or

The data models are used to represent the data and how it is stored in the database and to set the relationship between data items

Or

The data model emphasizes on what data is needed and how it should be organized instead of what operations will be performed on data.

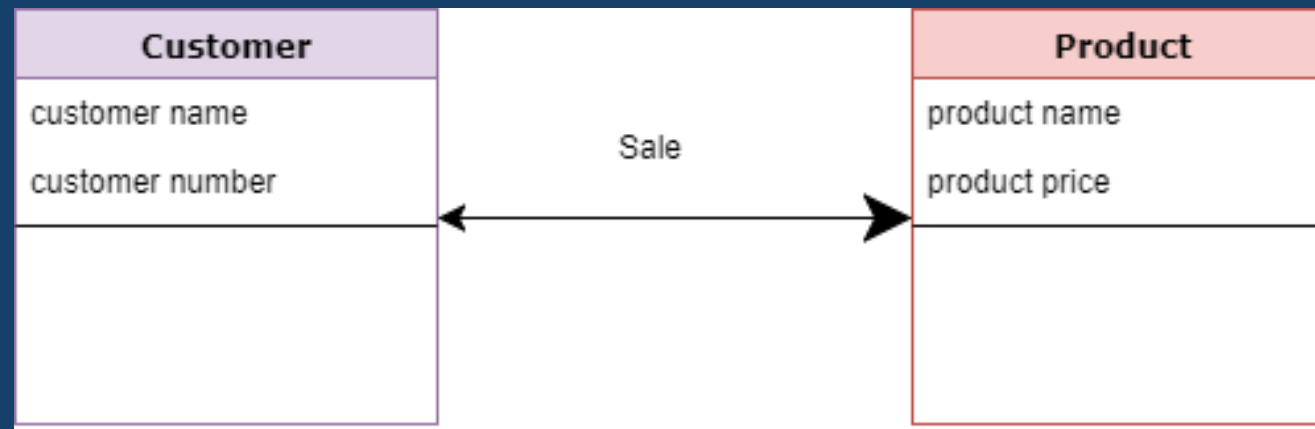
# Why use Data Model?

---

- Ensures that all data items required by the database are represented correctly. Data omissions will result in **erroneous reporting** and **wrong outcomes**.
- Helps design the database at the **conceptual**, **physical** and **logical** levels.
- Data Model structure helps to define the **relational tables**, **primary** and **foreign keys** and **stored procedures**.
- Provides a clear picture of the base data and can be used by database developers to create a physical database.
- Helps to **identify missing and redundant data**.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

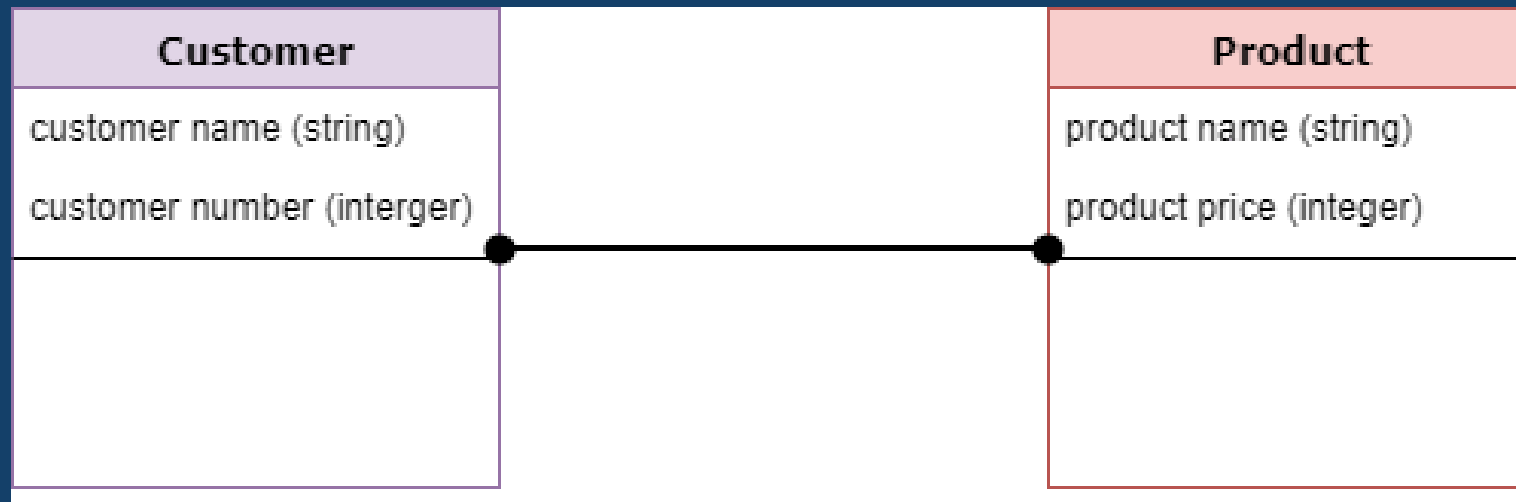
# Conceptual Data Model

- It is an organized **view of database concepts and their relationships**.
- The purpose of creating a conceptual data model is to establish **entities**, their **attributes**, and **relationships**.
  - **Entity**: A real-world thing or object.
  - **Attribute**: Characteristics or properties of an entity.
  - **Relationship**: Dependency or association between two entities.



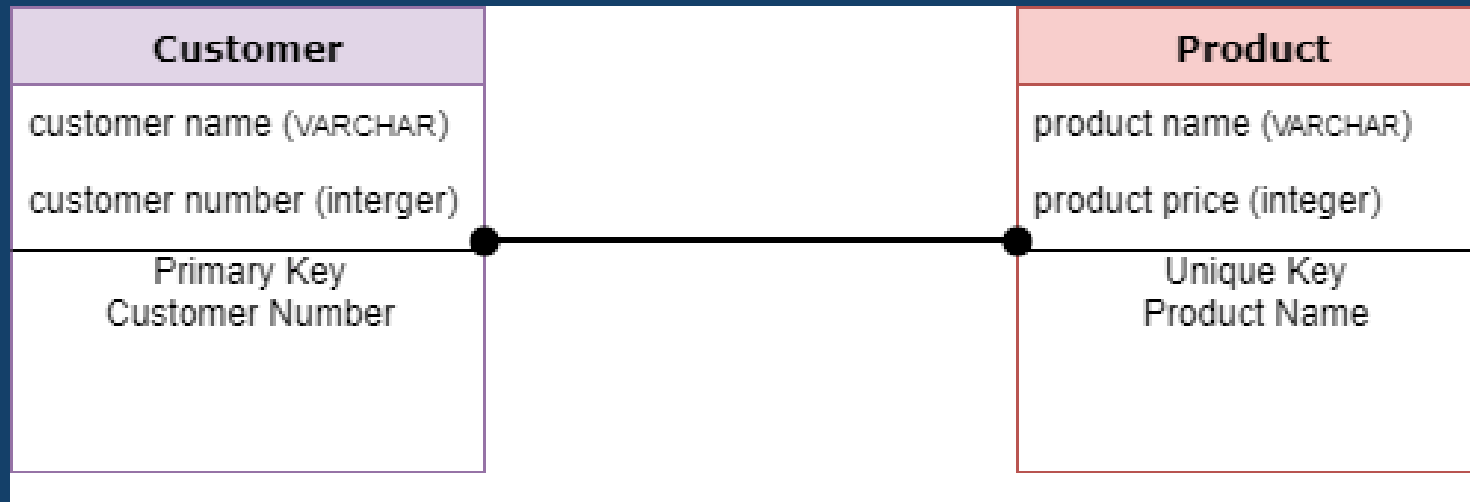
# Logical Data Model

- The Logical Data Model is used to define the structure of data elements and to set relationships between them.
- The logical data model adds further information to the conceptual data model elements.
- The advantage of using a Logical data model is to provide a foundation to form the base for the Physical model.



# Physical Data Model

- Describes a database-specific implementation of the data model.
- It offers database abstraction and helps generate the schema.
- The physical data model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers, and other RDBMS features.



# Types of Data Model

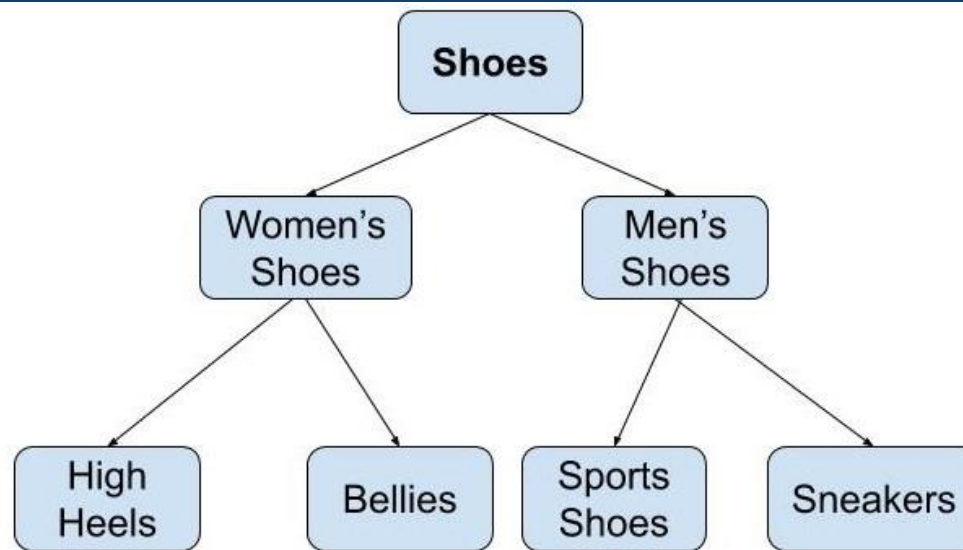
---

- Hierarchical Data Model
- Network Data Model
- Entity-Relationship Model
- Relational Data Model
- Object Data Model
- Object-relational Data Model
- NOSQL Data Model
  - Document-based Data Model
  - Graph-based Data Model
  - Column-based Data Model
  - Key-value Data Model
- Big Data Model
- and others



# Hierarchical Model

- It was the first DBMS model.
- This model organizes the data in the hierarchical tree structure.
- The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node.



*Hierarchical Model*

# Hierarchical Model

---

## – Advantages of Hierarchical Model

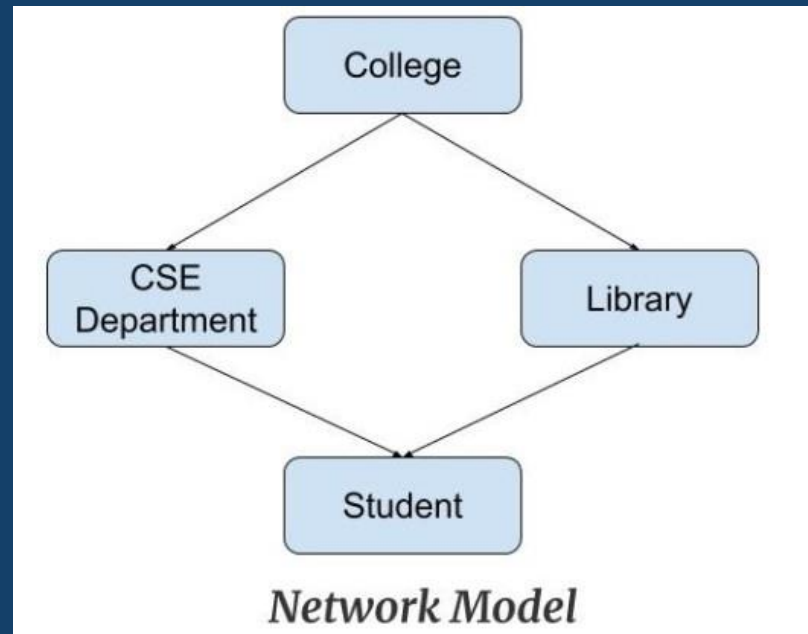
- It is very simple and fast to traverse through a tree-like structure.
- Any change in the parent node is automatically reflected in the child node so, the integrity of data is maintained.

## – Disadvantages of Hierarchical Model

- Complex relationships are not supported.
- As it does not support more than one parent of the child node so if we have some complex relationship where a child node needs to have two parent node then that can't be represented using this model.
- If a parent node is deleted, then the child node is automatically deleted.

# Network Model

- This model is an extension of the hierarchical model.
- It was the most popular model before the relational model. This model is the same as the hierarchical model, the only difference is that a record can have **more than one parent**.
- It replaces the hierarchical tree with a graph.



# Network Model

---

## – Advantages

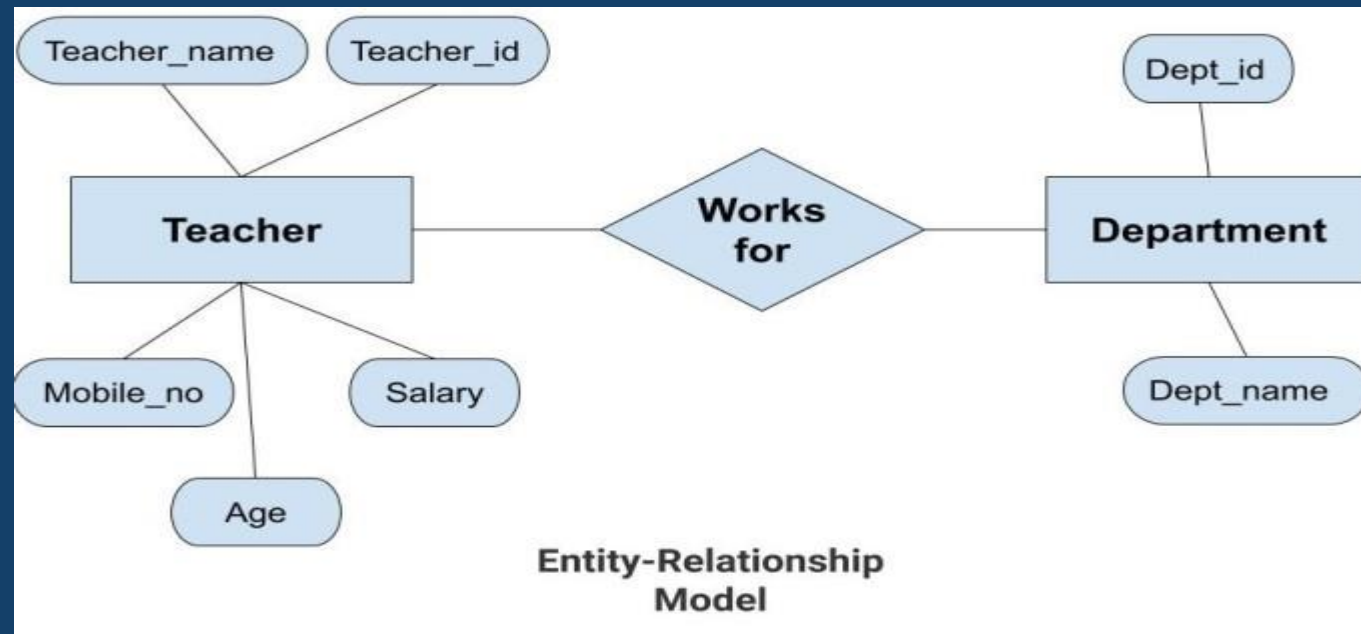
- The data can be accessed faster as compared to the hierarchical model.
- As there is a parent-child relationship so data integrity is present.
- Any change in parent record is reflected in the child record.

## – Disadvantages

- As more and more relationships need to **be handled the system might get complex**. So, a user must be having detailed knowledge of the model to work with the model.
- Any change like updating, deletion, insertion is **very complex**.

# Entity-Relationship Model

- Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model, we represent **the real-world problem** in the pictorial form to make it easy for **the stakeholders to understand**.



# Entity-Relationship Model

---

## – Advantages

- Simple: Conceptually ER Model is very easy to build.
- Effective Communication Tool: This model is used widely by the database designers for communicating their ideas.
- Easy Conversion to any Model: This model can be easily converted relational model by converting **the ER model to the table**.

## – Disadvantages

- There is no industry standard for developing an ER model. So one developer might use notations which are not understood by other developers.
- Some information might be **lost or hidden** in the ER model.

# Relational Model

- Relational Model is the most widely used model. In this model, the data is maintained in the form of a **two-dimensional table**.
- All the information is stored in the form of **row and columns**. The basic structure of a relational model is tables.
- So, the tables are also called relations in the relational model.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

**EMPLOYEE TABLE**

# Relational Model

---

## – Advantages

- Simple: This model is **simpler** as compared to the network and hierarchical model.
- Scalable: This model can be easily scaled as we can add as many rows and columns we want.
- Structural Independence: We can make changes in database structure without changing the way to access the data.

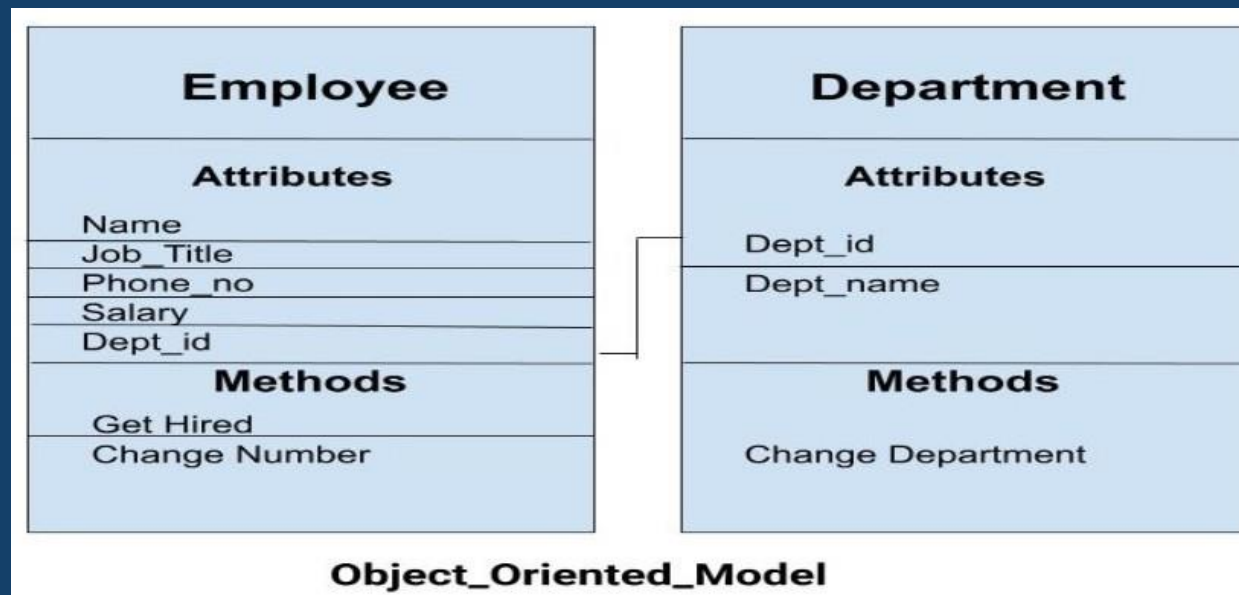
## – Disadvantages

- Hardware Overheads
- Bad Design

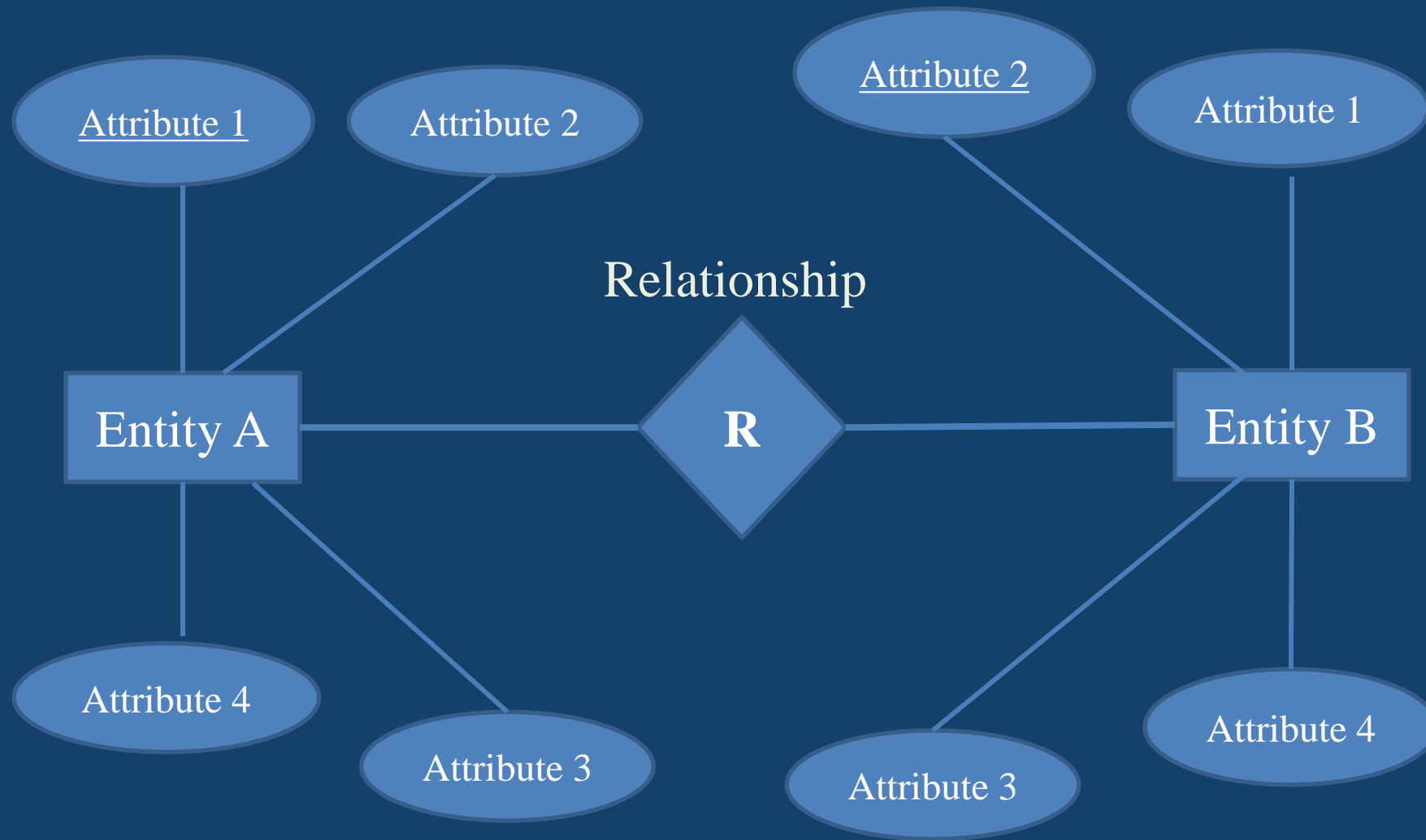


# Object-Oriented Data Model

- In this model, both the data and relationship are present in a single structure known as an object.
- We can store audio, video, images, etc. in the database which was not possible in the relational model(although you can store audio and video in relational database, it is advice not to store in the relational database).



# Entity-Relationship Model (ER)

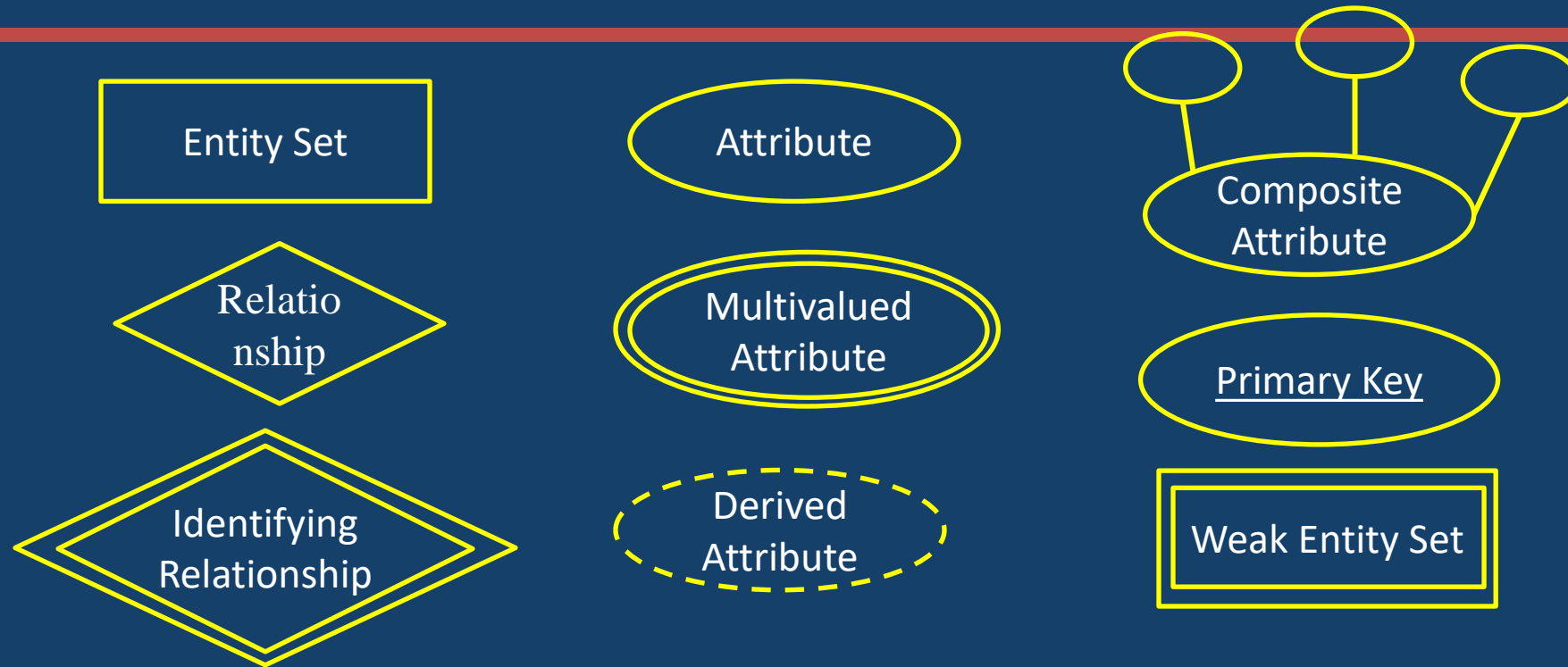


# What is ER Model?

---

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram.
- ER model helps to systematically analyze data requirements to produce a **well-designed database**.
- The ER Model represents real-world entities and the relationships between them.
- Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- ER diagrams are visual tools that are helpful to represent the ER model.

# ER Diagrams Symbols & Notations



Total participation of E2 in R



Cardinality

# Entity, Types of Entity, Entity Set

---

## – What is Entity?

- An entity is a real world “**thing**” or an “**object**” that is distinguishable from other objects based on the **values of the attributes** is process.

## – Classification of Entity

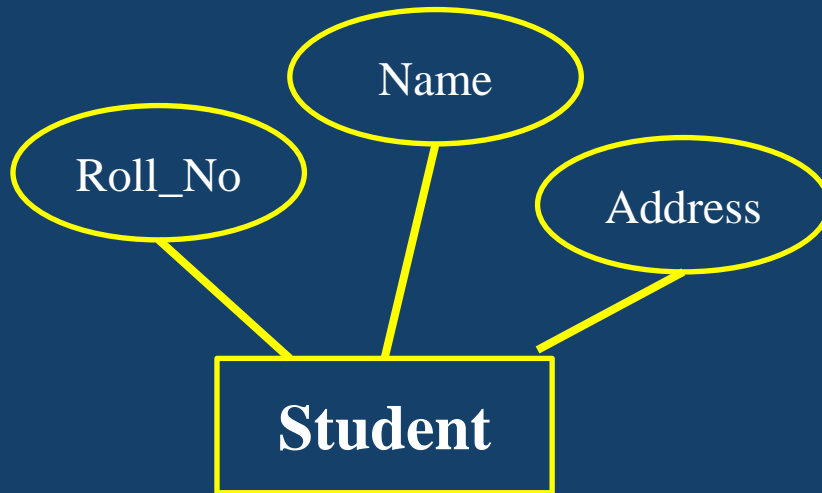
- **Tangible**: Entity which physically exist in real world.
  - E.g. Car, Customer\_Address, Bank\_Locker, etc.
- **Intangible**: Entity which exists logically.
  - E.g. Bank\_Account, Rate\_Of\_Interest, etc.

## – What is Entity Set?

- It is a **set of entities** of the **same type** that share the same properties, or attributes.

# Notation & Symbols: Entity, Entity Set

- Entity **can not be represented** in an **ER diagram** because it is an instance or data.
- Entity **can be represented** in a **Relational Model** by a row/tuple/record.
- Entity set is represented by a **rectangle** in an ER diagram .
- Entity set is represented by a **table** in a Relational Model.



ER Diagram

Student		
Roll_No	Name	Address
101	Jack	HCM
102	Tin	Hanoi
103	Phat	Dai An

Relational Model

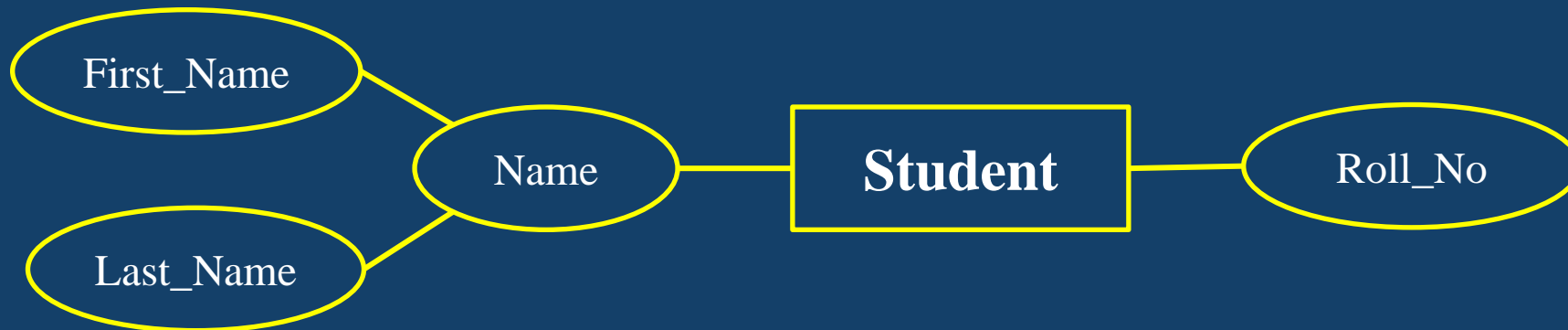
# Types of Attributes

## – Simple

- Attributes which can not divide further.
- Represented by simple oval
- E.g., **Roll\_No**.

## – Composite

- Attributes which can divide further.
- Represented by oval connected with another oval.
- E.g., **Name**. Name can further divide into first\_name and last\_name.



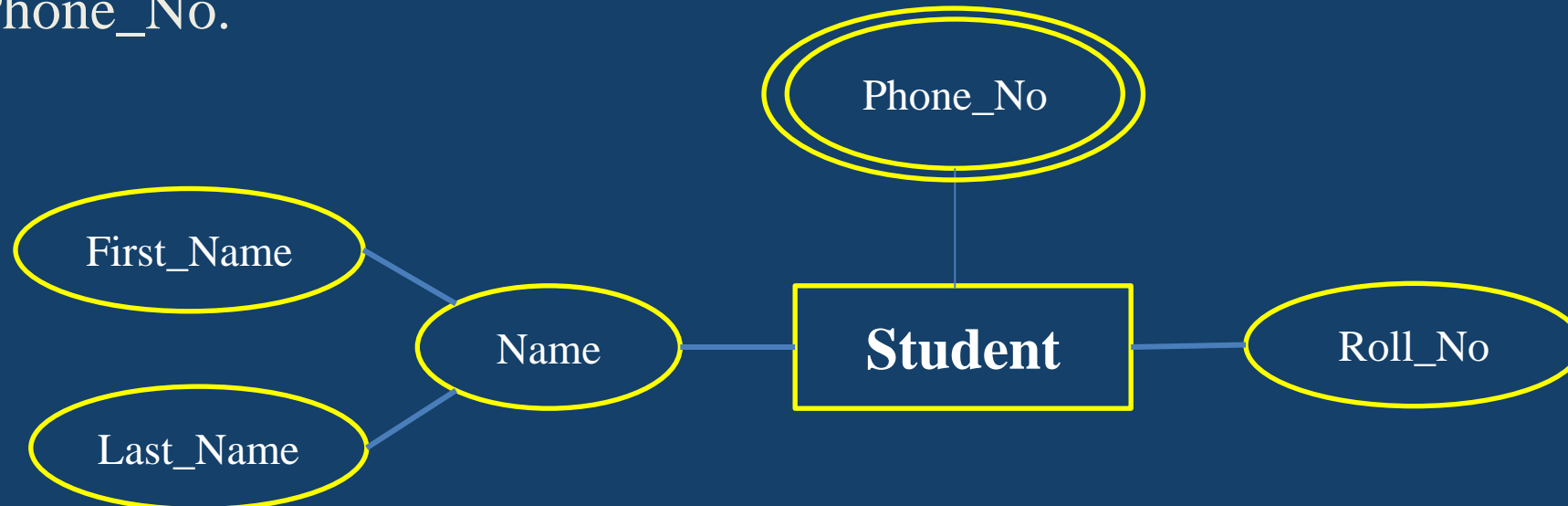
# Types of Attributes

## – Single value

- Attribute can have only one value at an instance of time.
- E.g. Date\_of\_Birth.

## – Multi-valued

- Attribute can have more than one value at an instance of time.
- E.g. Phone\_No.





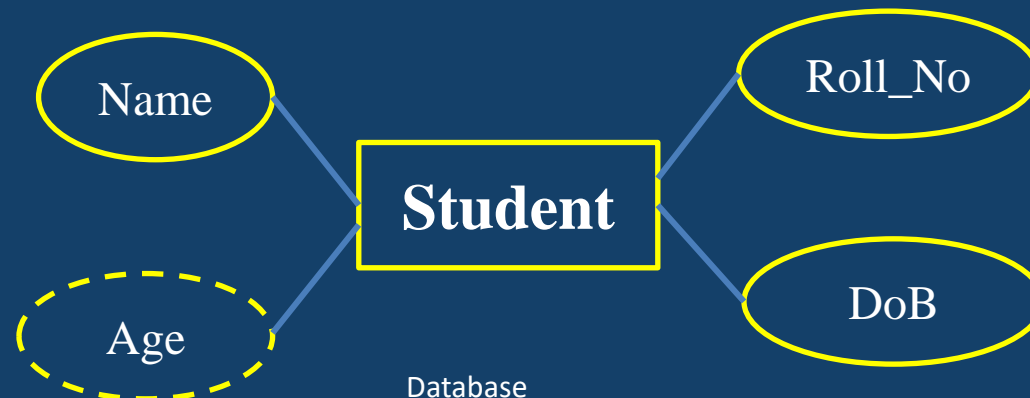
# Types of Attributes

## – Stored

- An attribute which are physically stored in the database.
- E.g., `student_id`, `name`, `roll_no`, `course_Id`.

## – Derived

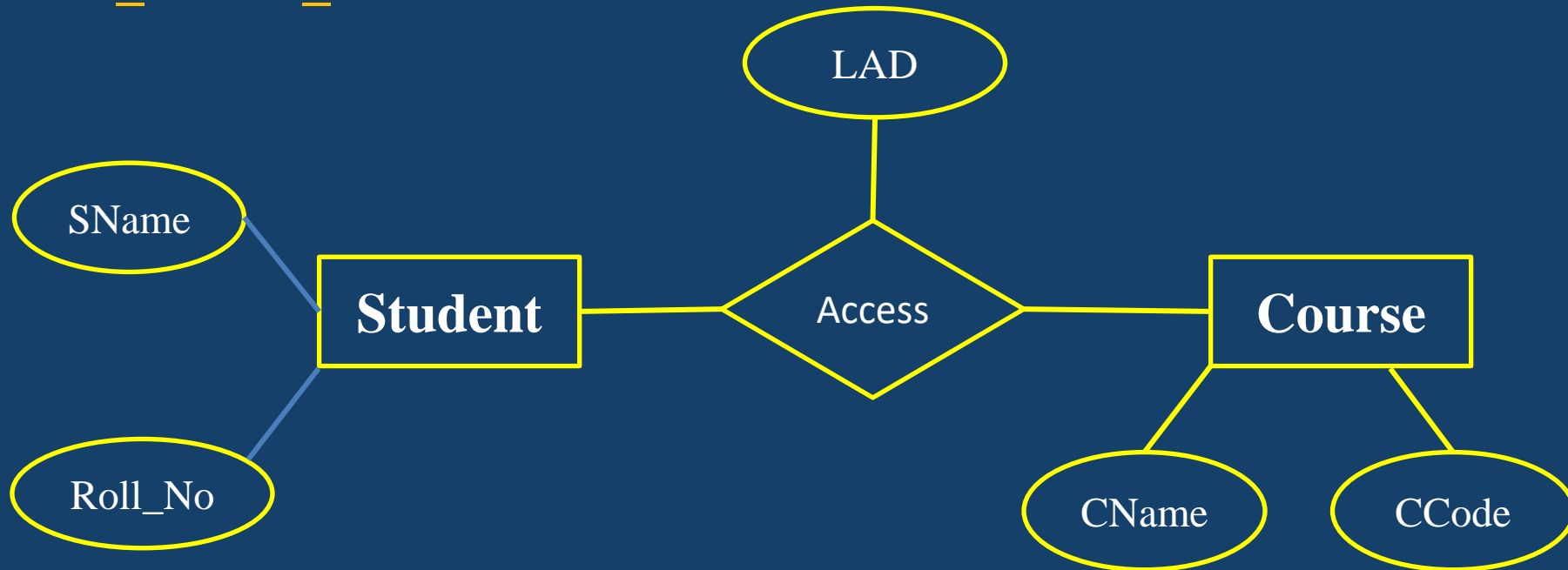
- An attribute whose values are calculated from other attributes.
- E.g., **Age**. In a student table, we can derive value of age with the help of `Date_of_Birth` attribute.
- Represented by dotted oval.



# Types of Attributes

## – Descriptive attribute

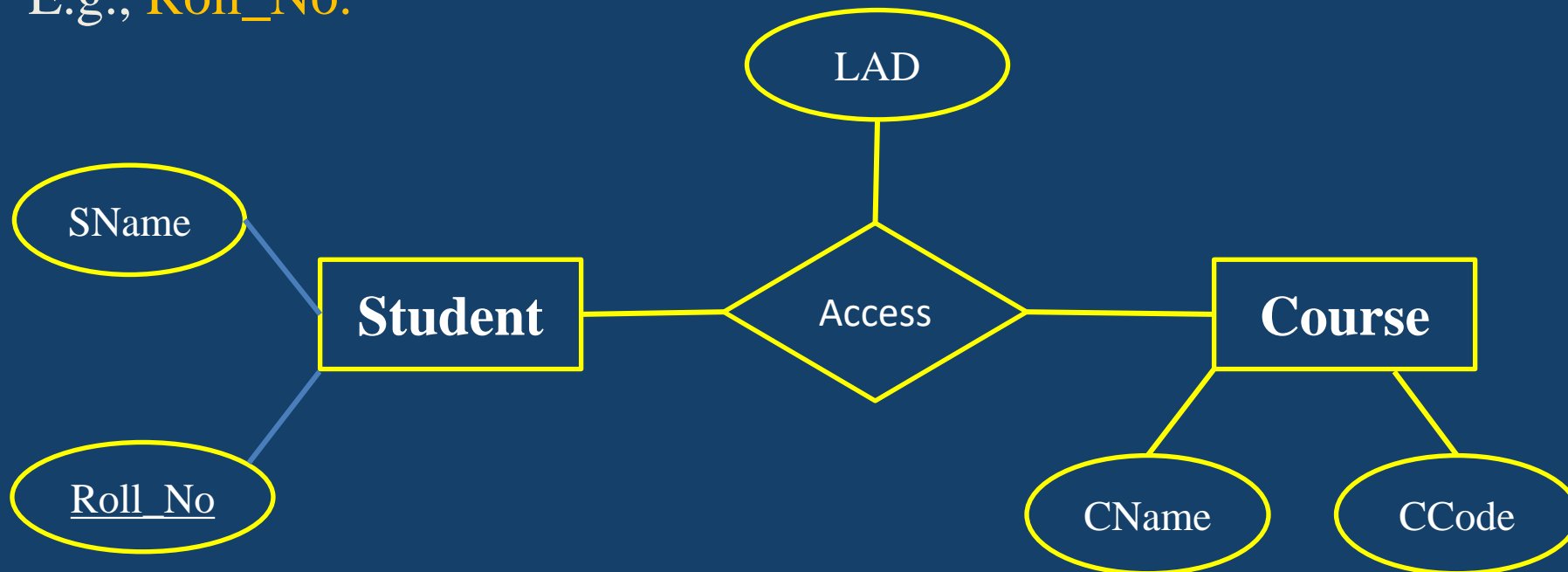
- The attribute(s) used for describing the relationship is called descriptive attributes, also referred as relationship attributes.
- They are actually used for storing information about the relationship. A relationship can have zero or more attributes.
- E.g. **Last\_Access\_Date**



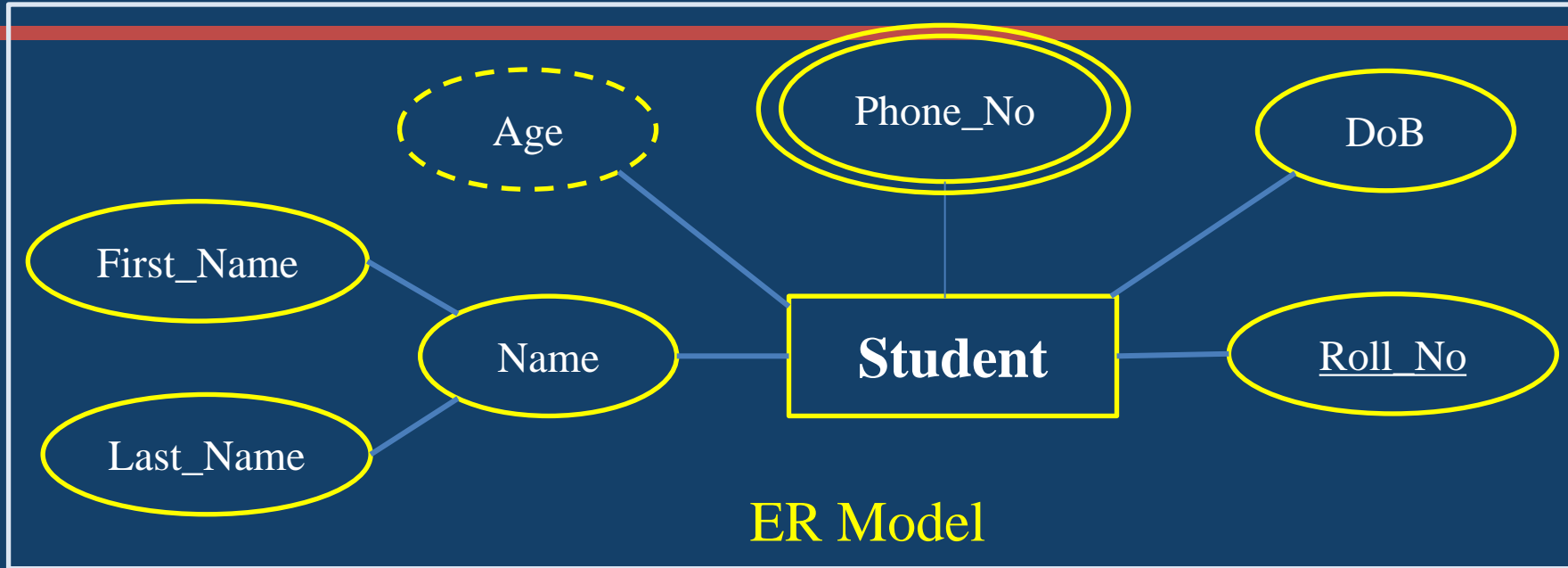
# Types of Attributes

## – Key Attribute

- The attribute or set of attribute uniquely identifies each entity set is called as key attribute.
- It represents a primary key.
- Its id denoted by an oval with underline lines.
- E.g., **Roll\_No**.



# Attributes



<u>Roll_No</u>	First_Name	Last_Name	DoB	Age

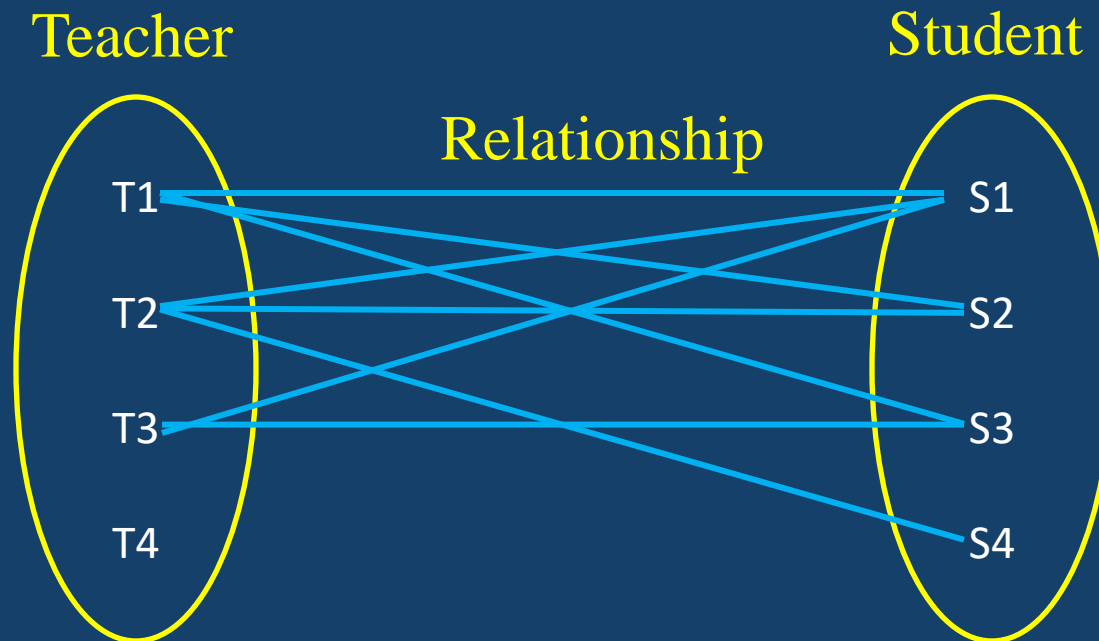
<u>Roll_No</u>	Phone_No

## Relational Model

# Relationship

## – What is Relationship?

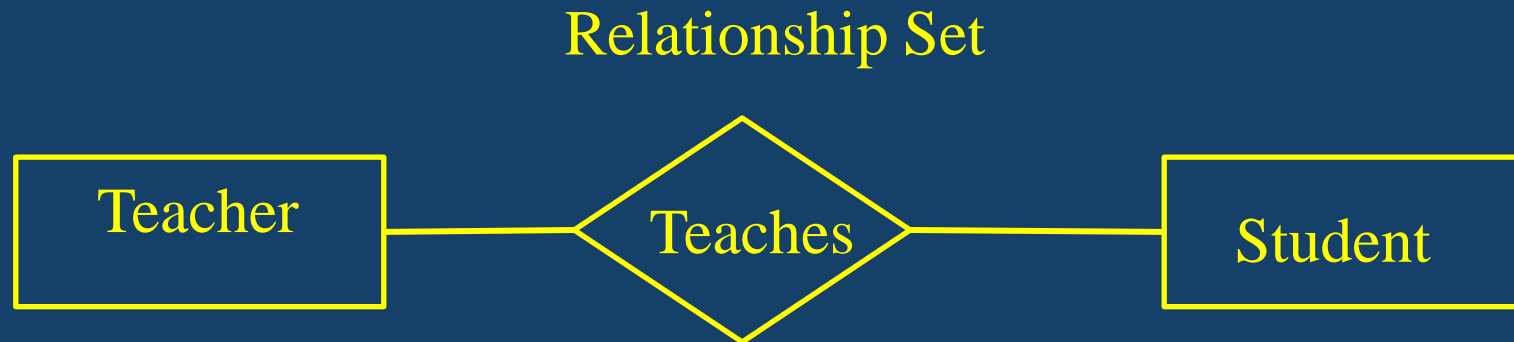
- It is an association between two or more entities of same or different entity set.
- **In relational model** represented either using a row in a **table**.



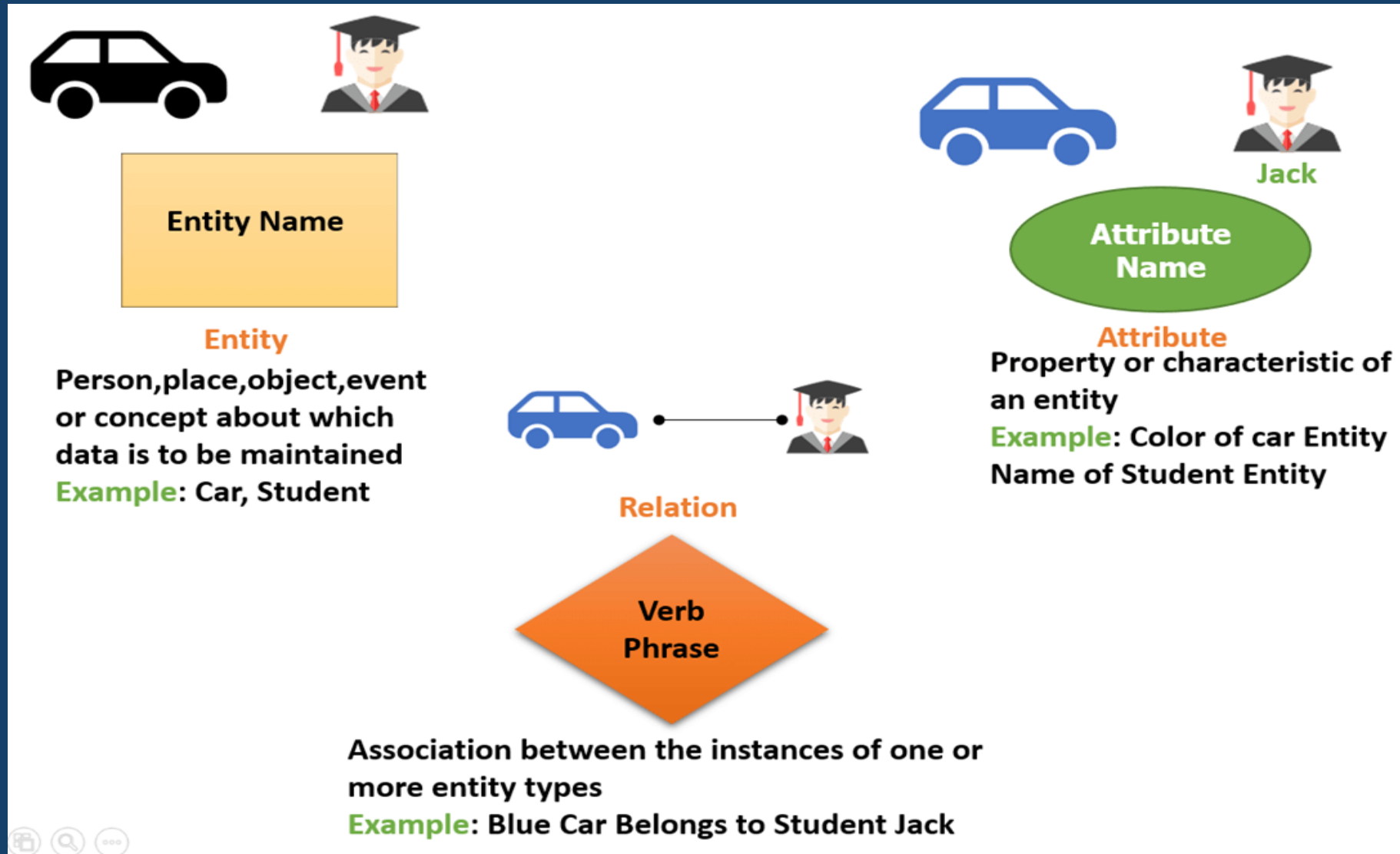
# Relationship Set/Type

## – What is Relationship Set/Type ?

- A relationship set is a **set of relationships** of the same type.
- Represented using **diamond** in ER diagram.
- In relational model either separated by table or by separate column (foreign key).



# ER Diagram



# Strong Entity Set

---

## – What is Strong Entity Set?

- It is an entity set which has primary key.
- E.g.
  - Entity set: Customer
  - Attributes: Customer\_Name, Customer\_AccNo, Phone



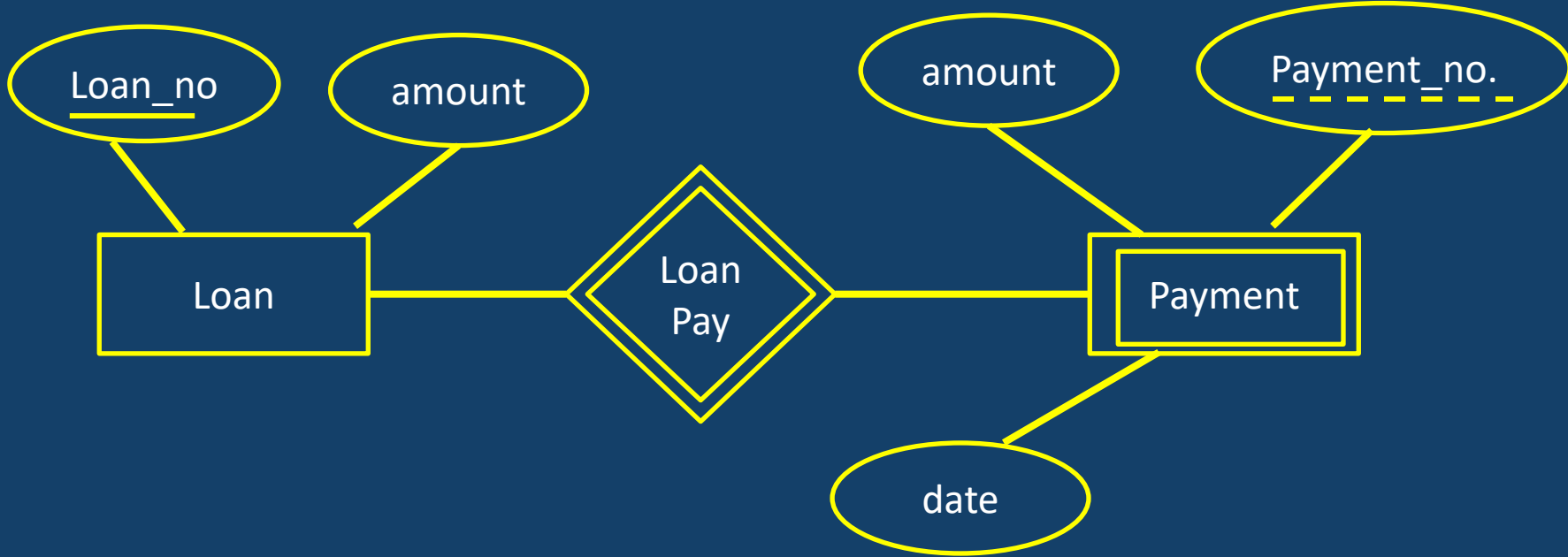
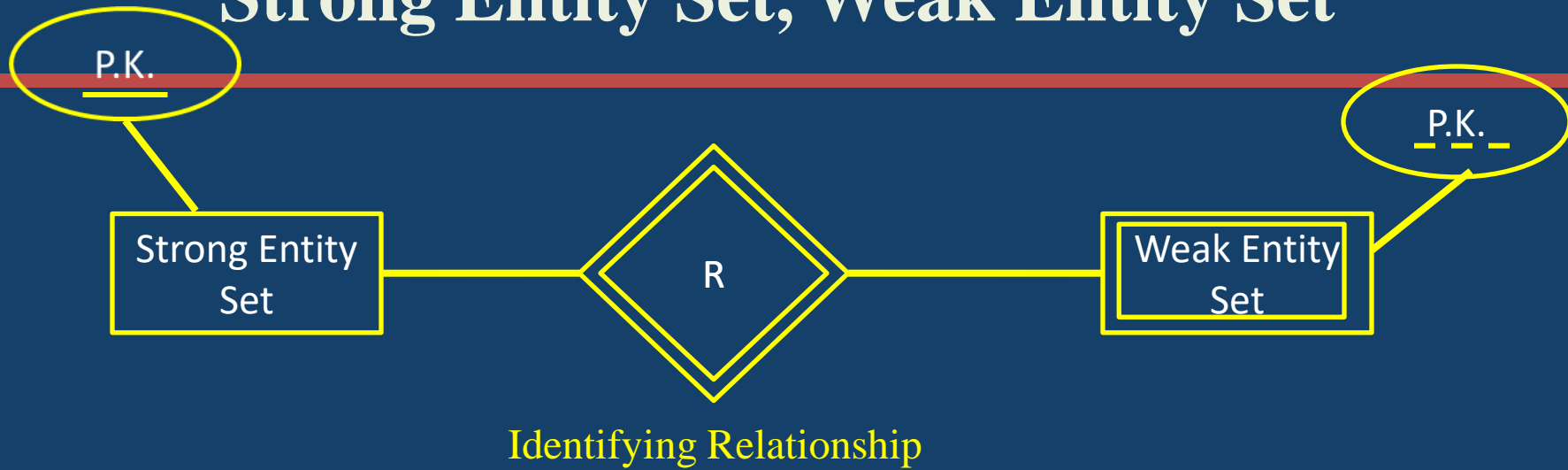
# Weak Entity Set

---

## – What is Weak Entity Set?

- It is an **entity set** which does not have sufficient attributes to form a primary key.
- E.g.
  - Entity set: Payment
  - Attributes: Payment\_Number, Payment\_Date, Payment\_Amount

# Strong Entity Set, Weak Entity Set



# Strong Entity Set, Weak Entity Set

Payment

Payment _Number	Payment _Date	Payment _Amount
P1	D1	10k
P2	D2	20k
P1	D1	10k
P2	D3	50k

Loan

User	<u>Loan Number</u>	Amount
U1	HSBCL0101	1000k
U2	HSBCL0202	1500k
U1	HSBCL0303	3200k

- In “Payment” table, there is no single attribute present that has unique values.
- So, it does not form primary key.

# Strong Entity Set, Weak Entity Set

- How to form the Primary Key of a Weak Entity Set?
- Primary key of a strong entity set on which weak entity is dependent plus the weak entity set discriminator.
- E.g., New Primary key in “Payment” table {Loan\_Number, Payment\_Number}

Payment

<u>Loan_Number</u>	<u>Payment_Number</u>	Payment_Date	Payment_Amount
HSBCL0101	P1	D1	10k
HSBCL0202	P2	D2	20k
HSBCL0202	P1	D1	10k
HSBCL0303	P2	D3	50k

Loan

User	<u>Loan_Number</u>	Amount
U1	HSBCL0101	1000k
U2	HSBCL0202	1500k
U1	HSBCL0303	3200k

# Weak Entity Set

---

- We can convert any weak entity set into strong entity set by adding an appropriate attributes. why, then, do we have weak entity sets?
  - Duplicating of key removed through weak entity set.
  - Maintains the dependency relationship.
  - Automatic deletion of weak entity set on deletion of corresponding strong entity set.
  - Fast data retrieval.

# Draw ER Diagram

---

A library management system tracks books, members, and loan transactions. The system includes three main entities: Book, Member, and Loan. The Book entity stores BookID, Title, Author, ISBN, and PublishedYear. The Member entity includes MemberID, Name, Email, and MembershipDate. The Loan entity records LoanID, LoanDate, ReturnDate, BookID, and MemberID. Each book can be loaned multiple times (one-to-many relationship with Loan), and each member can borrow multiple books (one-to-many relationship with Loan). This structure efficiently manages the library's resources and member activities.

# Draw ER Diagram

---

## Entities and Attributes:

Book – BookID, Title, Author, ISBN, PublishedYear

Member - MemberID, Name, Email, MembershipDate

Loan - LoanDate, ReturnDate, BookID, MemberID

## Relationships:

Book - Loan: A book can be loaned multiple times, but each loan transaction is linked to one book (One-to-Many relationship).

Member - Loan: A member can borrow multiple books, but each loan transaction is linked to one member (One-to-Many relationship).

# Draw ER Diagram?

---

- A publishing company produces books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more book publications. Every book require some items for publication. These items supplied by suppliers. One supplier can supply many items. Shop owner buys books from the publisher. Shop owner can buy many books, but one book can be bought by one shop owner only. Books are uniquely identified by book id.
- Identify the entities, attributes and relationship.
- Draw the ER diagram based on entities, attributes and relationship.