Chapter 5

Object Oriented Database System

Chapter outline

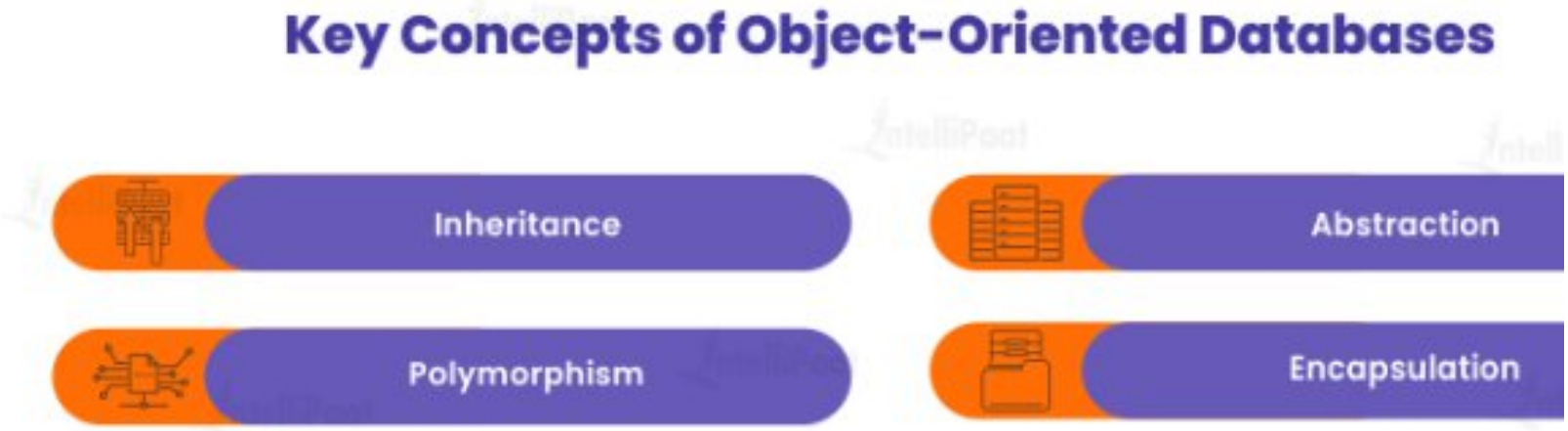
- Object Oriented Concepts/Abstraction, Encapsulation, Inheritance/
- Drawbacks of RDBMS
- OODBMS definition
- OODB design and implementation
- Object data modeling and E-R diagram
- Objects and attributes
- Object Identity
- Storing objects in RDBS

OODBMS definition

- ✓ **OODBMS** stands for **Object-Oriented Database Management System**. It is a database management system that integrates the principles of **object-oriented programming (OOP)** with database capabilities, allowing data to be stored, retrieved, and manipulated as **objects**.
- ✓ Instead of storing data in tables, rows, and columns (like in relational databases), an OODBMS stores data as objects, which are instances of classes.
- ✓ These objects contain both:
- ✓ Attributes (data members) – representing the state of the object.
- ✓ Methods (member functions) – defining the behavior or operations that can be performed on the object.
- ✓ This design approach allows for a more natural representation of real-world entities, as objects can model complex relationships and behaviors.

✓OODBs are particularly suitable for applications that require dynamic data structures and complex relationships, as they seamlessly integrate object-oriented concepts into the database realm, promoting efficient data management and retrieval.

Key Concepts of Object-Oriented Databases



Class

- ✓ A class represents a collection of objects having same characteristic properties that exhibit common behavior.
- ✓ It defines the **attributes (data)** and **methods (behaviors)** that the objects will have.
- ✓ It gives the blueprint or description of the objects that can be created from it.
- ✓ Creation of an object as a member of a class is called instantiation. Thus, object is an instance of a class.
- ✓ The constituents of a class are:
  - ➢ A set of attributes for the objects that are to be instantiated from the class.
  - ➢ Generally, different objects of a class have some difference in the values of the attributes.
  - ➢ A set of operations that portray the behavior of the objects of the class. Operations are also referred as functions or methods.

Example

Class: Teacher

Attributes:

- ✓ name
- ✓ subject
- ✓ salary

Methods:

- ✓ teach()
- ✓ giveExam()

Class: BankAccount

Attributes:

- ✓ accountNumber
- ✓ balance
- ✓ accountType

Methods:

- ✓ deposit()
- ✓ withdraw()

Class: Person

Attributes:

- ✓ name
- ✓ age
- ✓ Address

Methods:

- ✓ speak()
- ✓ walk()

Object

- An object is a real-world element in an object–oriented environment that may have a physical or a conceptual existence.
- It's an instance of a class and represents a real-world entity.
- An object is a fundamental unit in OOP.
- For example, in a banking system, "Account" could be an object, or in a car simulation, "Car" could be an object.
- Objects have:

State (Attributes/Properties): Data that describes the object.

- ✓ For a "Car" object, this might include color, make, model, speed.

Behavior (Methods/Functions): Actions that the object can perform.

- ✓ For a "Car" object, this might include start(), accelerate(), brake().

- Objects can be modeled according to the needs of the application.

Cont..

An object may have a physical existence, like a customer, a car, etc.; or an intangible conceptual existence, like a project, a process, etc.

Example:

An object person, class Student may have:

- ✓ Attributes: ID, Name, Age
- ✓ Methods: register(), updateProfile()

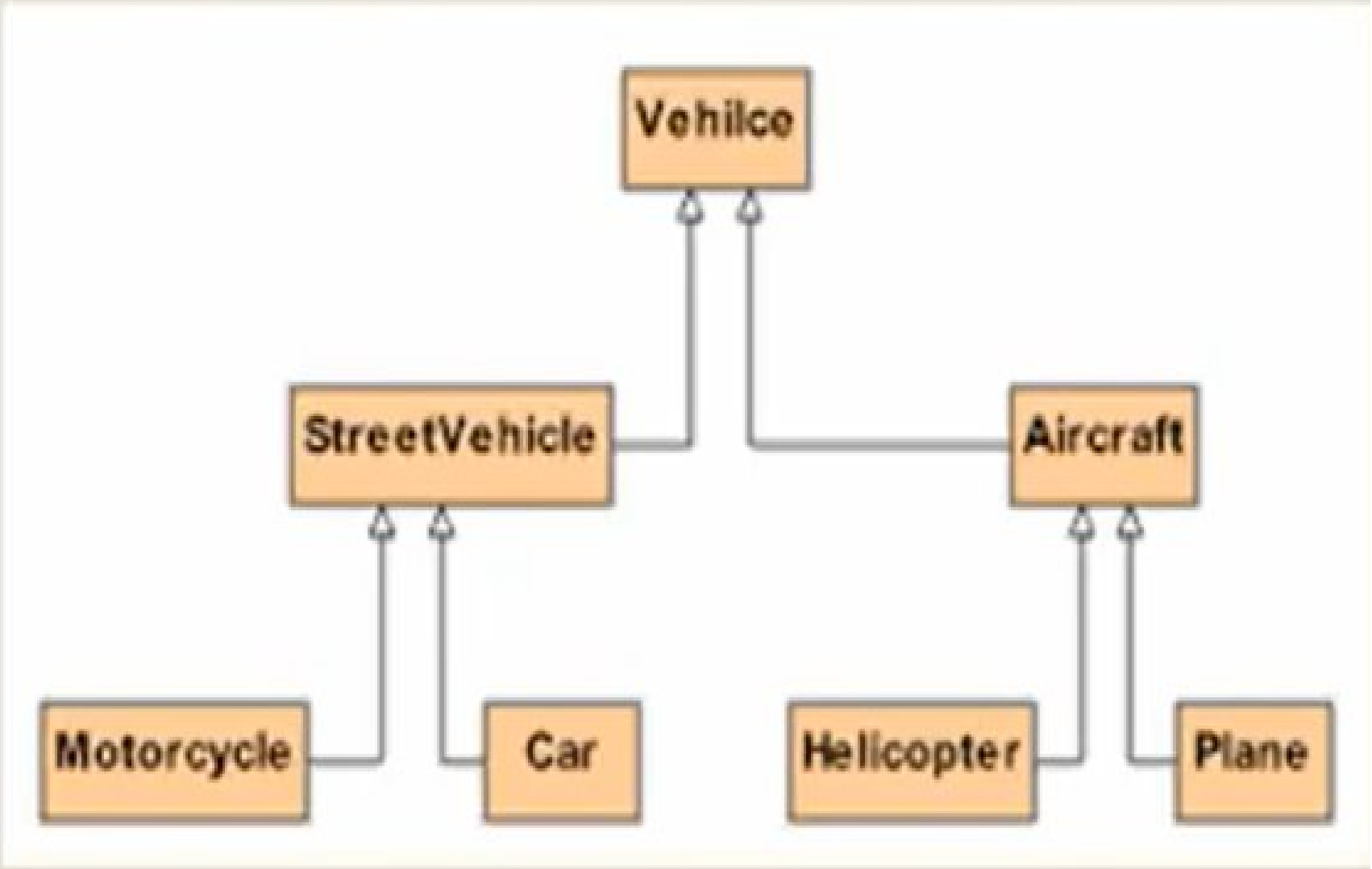
Object vs. Class

- ✓ A Class acts as a blueprint or a template. It defines the structure (attributes) and behavior (methods) that all objects of that type will have.
- ✓ An Object is a specific instance of a class. It has the structure and behavior defined by its class, but with unique values for its attributes.

Inheritance

- Inheritance is a mechanism in object-oriented programming that allows objects to inherit properties and methods from parent objects or classes.
- It promotes code reuse and the creation of hierarchical relationships between objects.
- In the context of OODBs, inheritance enables the organization and structuring of objects into a class hierarchy.
- It facilitates the sharing of common attributes and behaviors between objects, simplifying the modeling and manipulation of data.

Example


Vehilco
StreetVehicle
Aircraft
Motorcycle
Car
Helicopter
Plane

Abstraction

- Abstraction is a concept that focuses on representing complex systems or entities by emphasizing their essential characteristics while hiding the complex, underlying implementation details.
- It's the mechanism of **hiding complex implementation details** of the objects and the database while exposing only the essential and relevant information to the user or application.

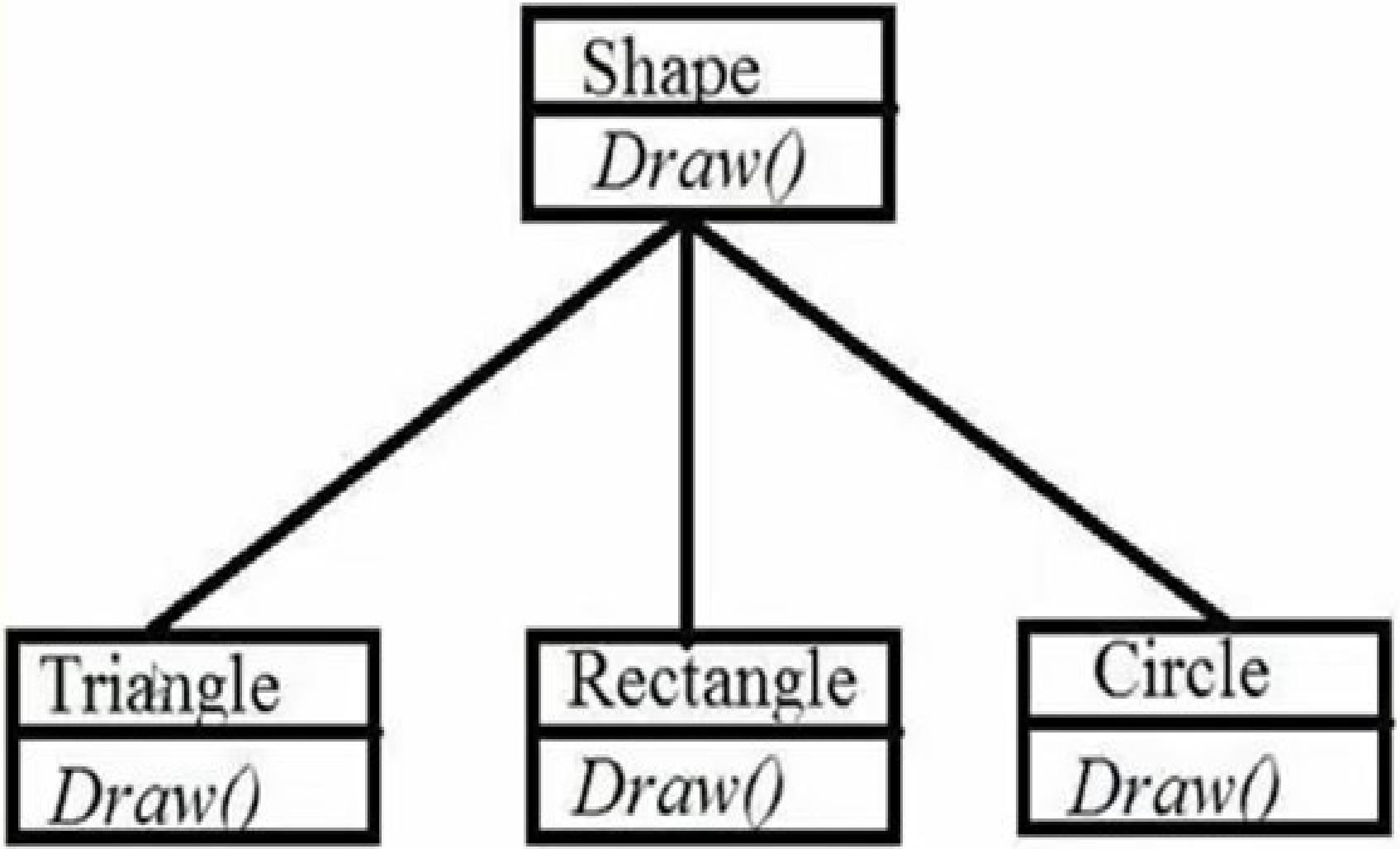
Eg:- Use ATM withdraw cash

CalculateGPA()

Polymorphism

- ✓ Polymorphism is the ability of objects to take on multiple forms or exhibit different behaviors based on their context.
- ✓ It allows objects of different classes to be treated as objects of a common superclass, promoting flexibility and extensibility in object-oriented systems.
- ✓ Polymorphism in OODBs enables objects to respond differently to the same message or method call, depending on their specific class or inheritance hierarchy.
- ✓ It simplifies the management of objects with varying behaviors and promotes code reuse and flexibility in modeling complex relationships between objects.

example



Encapsulation

- It is often described as the concept of **binding data and the methods that operate on that data into a single unit** (the class/object), while **restricting direct access** to the internal state.
- Think of it like a medicine capsule: it contains the ingredients (data) and is sealed, and you interact with it as a single, simple unit (the object's interface) without needing to know the complex internal formula.
- It provides data protection and information hiding, ensuring that an object's internal state is accessible and adaptable only through defined interfaces.
- Encapsulation in OODBs ensures that data integrity is maintained by controlling access to the object's data and enforcing the use of specified

Cont..

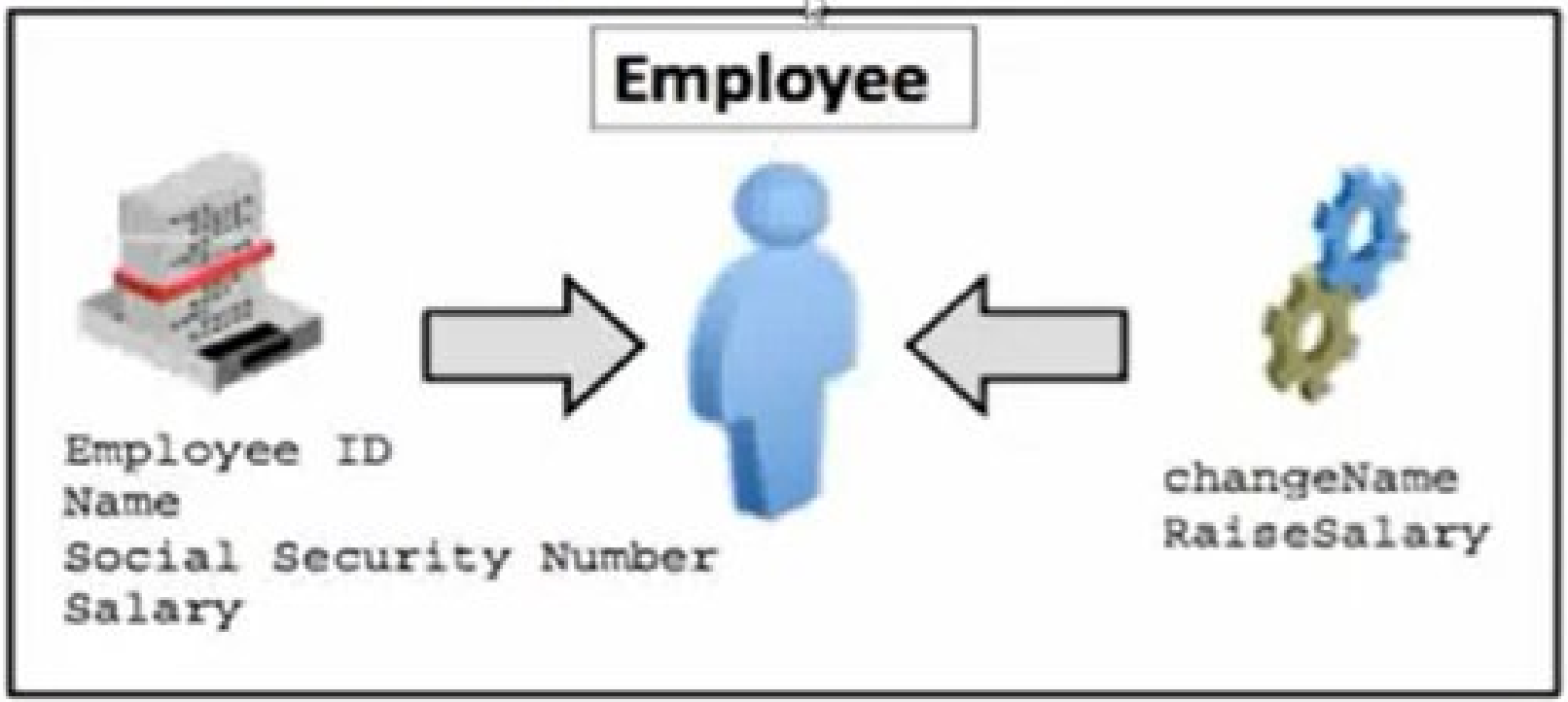
- Eg:

```
Class Student{
        private:
                string grade;
        public:
                void getGrade(){
                m
                }
                Void setGrade(){
                b
                }
}
```

Example

Encapsulation: Example

What data and operations would you encapsulate in an object that represents an employee?



OODB design and Implementation

- ✓ The design of an Object-Oriented Database Management System (OODBMS) is fundamentally different from a Relational Database Management System (RDBMS) because it aims to store real-world entities (objects) directly, along with their behaviors, rather than mapping them to a rigid set of tables.
- ✓ Object–Oriented Design *OOD* involves implementation of the conceptual model produced during object-oriented analysis.
- ✓ In OOD, concepts in the analysis model are mapped onto implementing classes, constraints are identified and interfaces are designed, resulting in a model for the solution domain, i.e., a detailed description of how the system is to be built on concrete technologies.

Cont..

1.Identify Real-World Entities and Define Classes

- **Identify Entities:** Analyze the application requirements and identify all relevant real-world entities (e.g., Customer, Product, Order, Department). These become your **Classes**.

2. Define Methods (Behavior)

- ✓ For each class, define the **methods** (operations/behavior) that act upon the object's data.
- ✓ Query Language: Determine how data will be queried, typically using an Object Query Language (OQL), which extends SQL with object-oriented concepts.

3. Define Structure (Attributes)

- In OODBMS, the object structure refers to how data is organized and represented as objects, embodying the principles of object-oriented programming

Cont..

- ✓ Unlike relational databases that store data in tables, OODBMSs store data as objects, which encapsulate both data (attributes) and behavior (methods) into a single unit.
- ✓ The structure of an object refers to the properties that an object is made up of.
- ✓ These properties of an object are referred to as an attribute.

Attributes: Define the properties or state of the class. Attributes can be simple data types (e.g., string, int), or they can be complex, containing collections (e.g., a Set or List) or even references to **other objects**.

- ✓ *Example:* The Student class has attributes like name (string) and dateOfBirth (date).

Cont..

4. Define Behavior (Methods)

- **Methods/Operations:** Define the actions or functions that can be performed on the object.
- Example: The Student class has methods like calculateGPA() or enrollInCourse(courseID).

5. Define Relationships and Concepts

- **Object Identity (OID):** Understand that every instance of a class (an object) will be assigned a unique, immutable Object ID, which is the system's way of managing relationships, avoiding the need for data-based primary and foreign keys.
- Every object in an OODB has a unique, system-generated identifier called an Object Identifier (OID).
- **Association (Relationships):** Relationships between classes are modeled using **reference attributes** (like pointers in programming languages) which store the OID of the related object(s).

OO Data modeling and E-R diagramming

❖OO Data Modeling, typically visualized using a **UML Class Diagram**, is the standard approach in **Object-Oriented software development**.

❖It models both the **structure** (data) and the **behavior** (operations) of a system.

Components of an OO Class Diagram

1. Class: Represented by a rectangle divided into three compartments: Class Name, Attributes, and Operations (Methods).

2. Attributes: The data fields (e.g., +name: String, -age: Integer).

3. Relationships:

- ✓**Association:** A general link between classes with a specified **multiplicity** (e.g., 1..*).
- ✓**Inheritance:** Shown as a solid line with a hollow triangle pointing from the subclass to the superclass.
- ✓**Composition/Aggregation:** Strong/weak "whole-part" relationships. eg A **Department** (whole) aggregates **Professors** (parts). If the Department is dissolved, the Professor objects still exist.

Cont.…

E-R Diagramming (Entity-Relationship Model)

- E-R Diagramming is a conceptual modeling technique specifically focused on **database design**.
- It models the **logical structure of data** and is the traditional used to designing a **Relational Database (RDBMS)** schema.

Components of an E-R Diagram

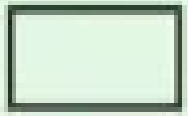
1. Entity: Represented by a rectangle.

2. Attribute: Represented by an oval (often simply listed within the entity rectangle in Crow's Foot notation).

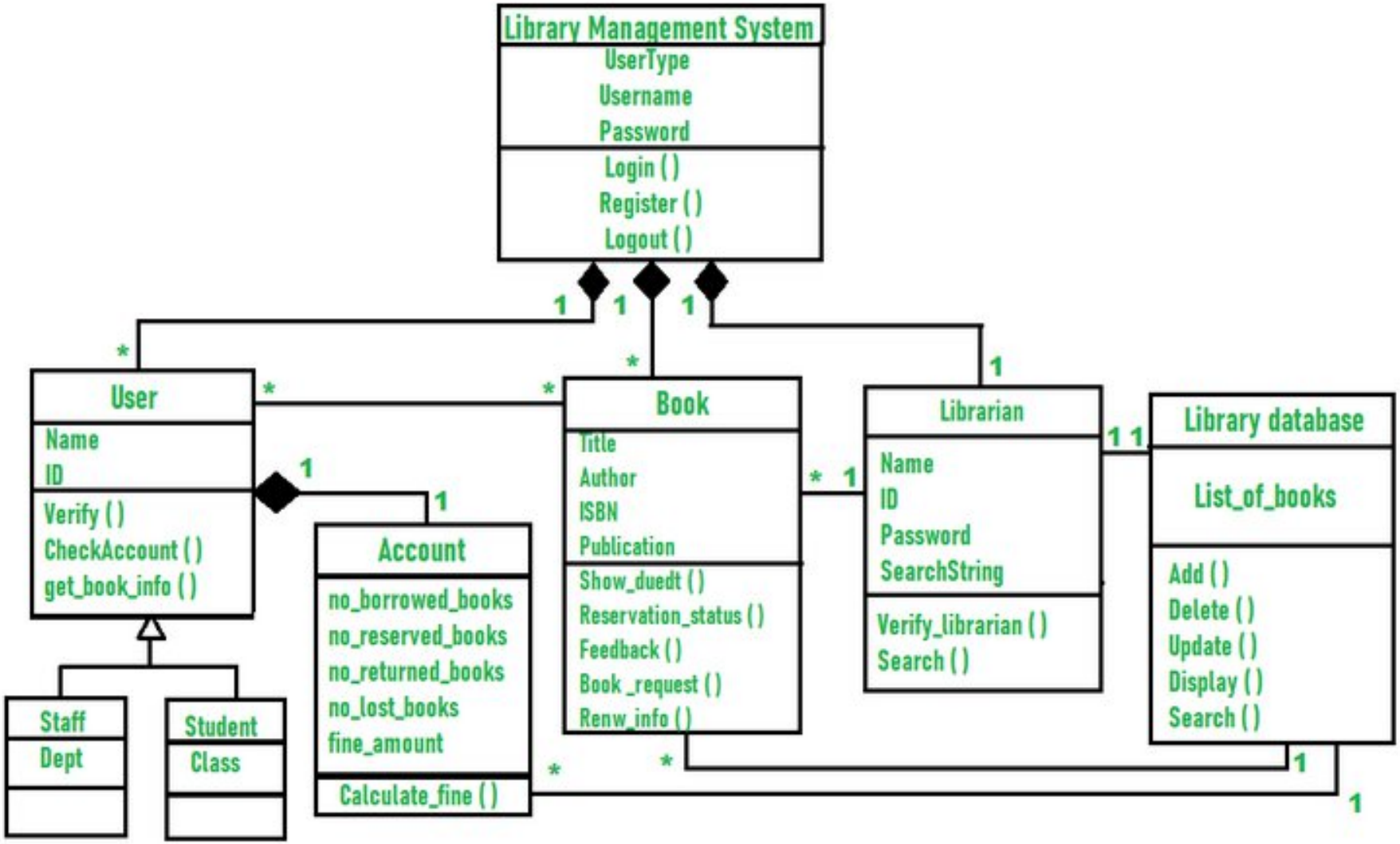
3. Relationship: Represented by a diamond connecting two or more entities.

4. Cardinality/Multiplicity: Constraints on the relationship (e.g., 1:1, 1:N, M:N).

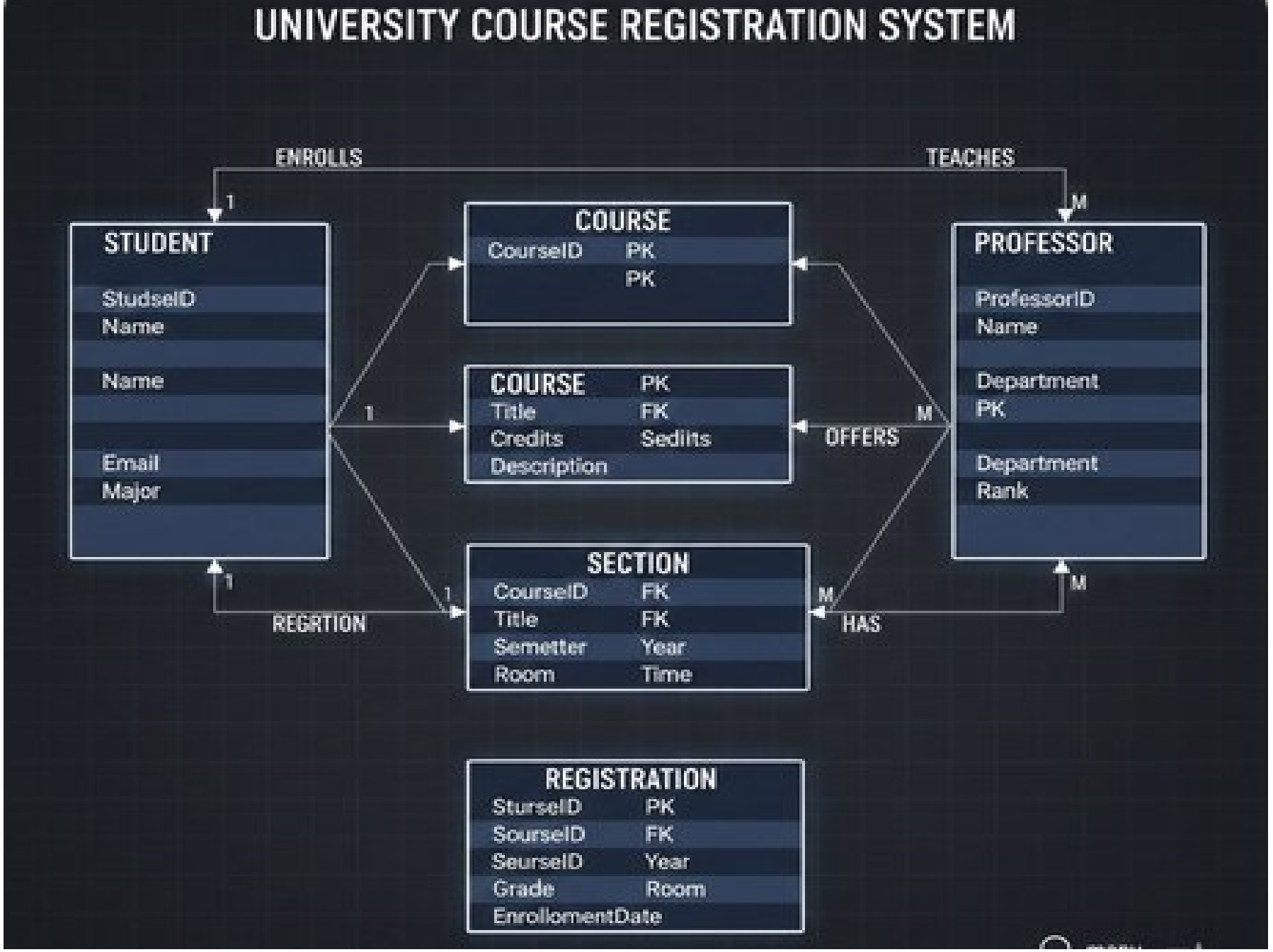
Symbols Used in ER Model

Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity

Class Diagram of Library Management System :



CLASS DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM


UNIVERSITY COURSE REGISTRATION SYSTEM
ENROLLS
TEACHES
1
M
STUDENT
StudsetID
Name
Name
Email
Major
COURSE
CourseID PK
PK
COURSE PK
Title FK
Credits Sedits
Description
PROFESSOR
ProfessorID
Name
Department
PK
Department
Rank
1
M
OFFERS
1
1
REGRTION
SECTION
CourseID FK
Title FK
Semetter Year
Room Time
M
M
HAS
REGISTRATION
SturseID PK
SourseID FK
SeurseID Year
Grade Room
EnrollomentDate

Objects and attributes

Objects

- The data object is actually a location or region of storage that contains a collection of attributes or groups of values that act as an aspect, characteristic, quality, or descriptor of the object.
- A vehicle is a data object which can be defined or described with the help of a set of attributes or data.

Attributes

- An attribute is a property or characteristic of an object – Examples: eye color of a person, temperature, etc. – Attribute is also known as variable, field, characteristic, or feature
- It can be seen as a data field that represents characteristics or features of a data object. For a customer object attributes can be customer Id, address etc.

Thank You!!!