

Name of Faculty: Dr. Virendra Kumar Tiwari

Designation: Professor

Department: LNCT-MCA

Subject: 202-DBMS

Unit: V

Topic: *Object oriented Databases-basic idea and the model.*

Object-Oriented Databases

Object-oriented databases (OODB) are databases that represent data in the form of objects and classes. In object-oriented terminology, an object is a real-world entity, and a class is a collection of objects. Object-oriented databases follow the fundamental principles of object-oriented programming (OOP). The combination of relational model features (concurrency, transaction, and recovery) with object-oriented principles results in an object-oriented database model.

The object-oriented database model (OODBM) is an alternative implementation to that of a relational model. An object-oriented database is similar in principle to an object-oriented programming language. An object-oriented database management system is a hybrid application that uses a combination of object-oriented and relational database principles to process data. That said, we can use the following formula to outline the OODBM:

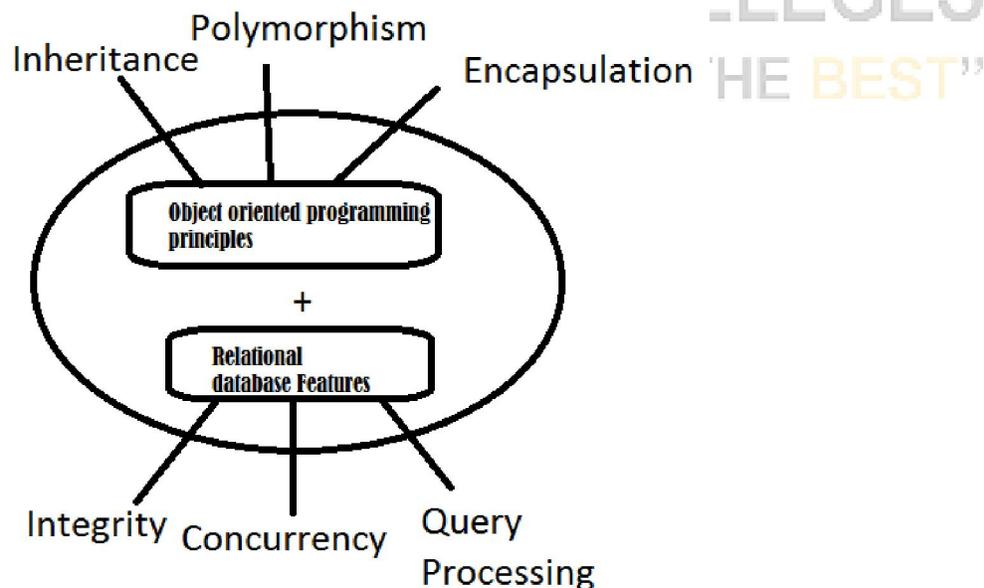


Figure 1: The Object-Oriented Database Model

Object-Oriented Programming + Relational Database Features = Object-Oriented Database Model

- Object oriented database systems are alternative to relational database and other database systems.
- In object oriented database, information is represented in the form of objects.
- Object oriented databases are exactly same as object oriented programming languages. If we can combine the features of relational model (transaction, concurrency, recovery) to object oriented databases, the resultant model is called as object oriented database model.

Fundamental Features

The object-oriented approach considers all entities as objects. An object has properties (state) and methods (behavior). Each object is identified using a unique object identifier. For example, let us consider a real-world entity called 'Student'. A student has states or properties such as a name, USN, date of birth, address, etc. Similarly, the student has behaviors or methods including attending classes, writing exams, paying fees, etc.

This next figure below shows how the 'Student' object can be represented.

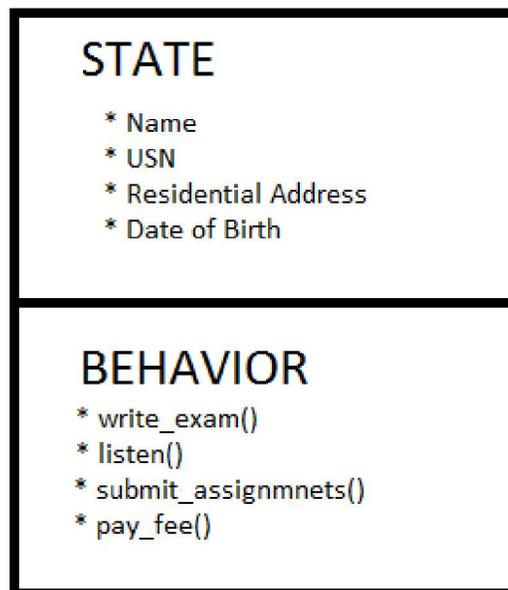
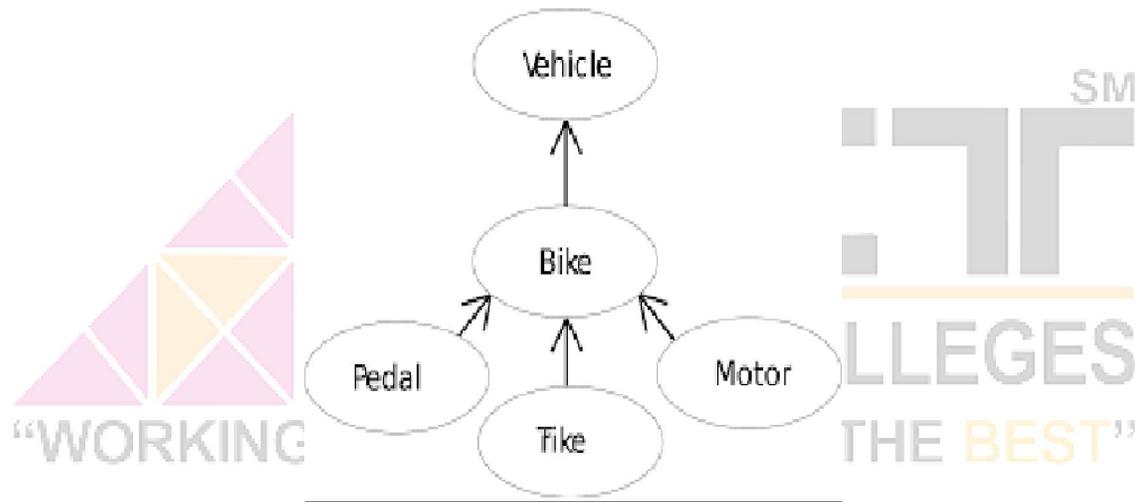


Figure 2: Representation of the Student Object

As you can see, the state and behavior cells are separate, but related. A class is a collection of similar objects. In this example, a collection of first year commerce students forms a class. In a class, the objects interact with each other using messages.

Encapsulation is an important object-oriented feature. This hides the implementation details from the end-users and displays only the needed descriptions. For example, imagine that you are buying a soap from a commercial store. You know its ingredients such as color and fragrance by looking on the box, but you aren't shown the actual method used to manufacture the soap.

Inheritance is considered important in object-oriented design because it enables re-usability. It is defined as the method of creating new classes from existing classes. The new classes not only inherit the properties of their parent class, but they also have their own unique properties. For example, when a child is born to parents, the child resembles either its mother or father, but it also has its own unique traits and personality. This figure below shows an additional example of how inheritance works.



Association refers to links between the various entities of an application. In an object-oriented database, association is denoted as references between various objects. For example, individuals from a 'Person' class can also be associated with the 'Committee' class. The association name here can be called 'Membership' as outlined in this figure below.

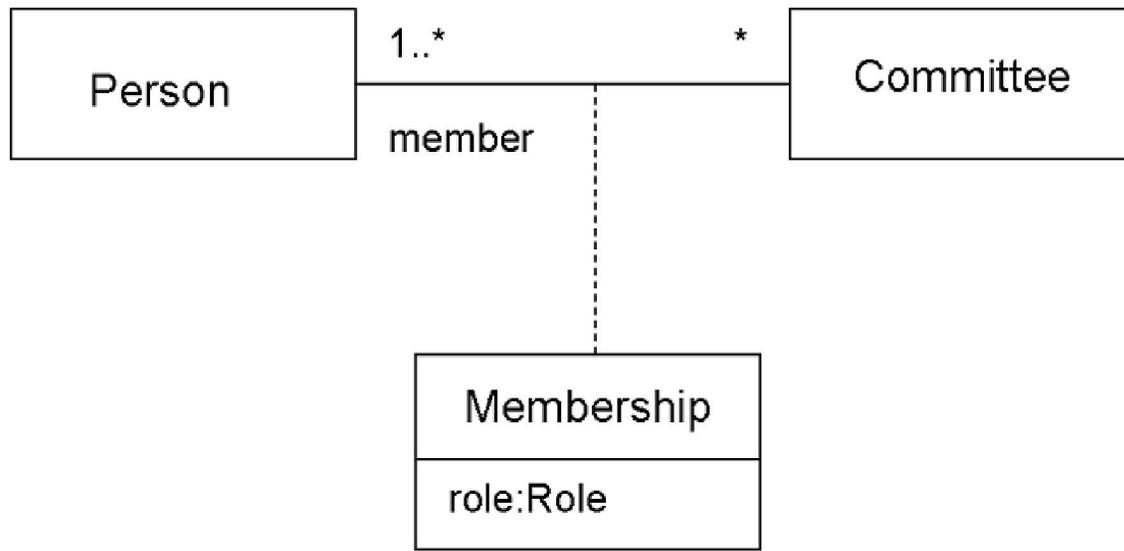
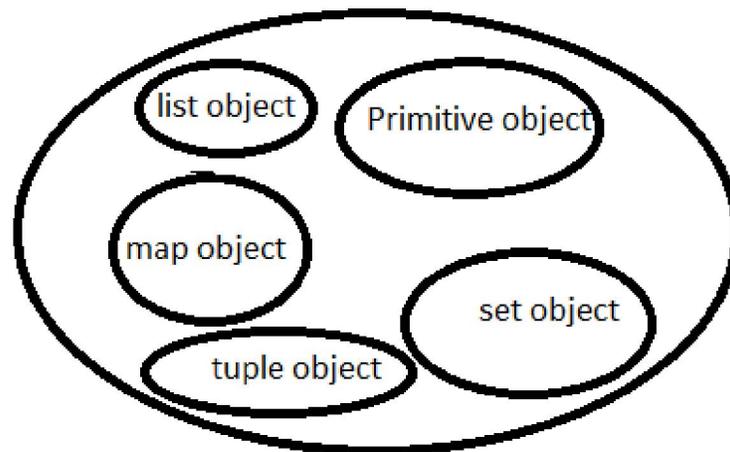


Figure 4: Association Example

The concept of complex objects is derived from applying constructors to simple objects. Simple objects are mainly items such as integers, byte strings, and characters. Complex objects are items such as maps, sets, lists, tuples, or collections of many primitive objects as shown in this figure here.

COMPLEX OBJECT



The object-oriented database allows for the creation of persistent objects. A persistent object is one that lives in computer memory even after completing its execution. This is different from the lifespan of normal objects, which expire after execution, are destroyed immediately, and freed from memory. Object persistence solves the database challenges of concurrency and recovery.

Challenges in ORDBMS implementation

During the implementation of ORDBMS, various challenges arise which need to be resolved. They are:

1. Storage and accessibility of data: It is possible to define new types with new access to structures with the help of OODBMS. Hence, it is important that the system must store ADT and structured objects efficiently along with the provision of indexed access.
2. Query Processing : Challenge: Efficient flow of Query Processing and optimization is a difficult task.
3. Query Optimization: Challenge: New indexes and query processing techniques increase the options for query optimization. But, the challenge is that the optimizer must know to handle and use the query processing functionality properly.
 1. WHERE-clause conditions matched by that index.
 2. Cost of fetching a tuple for that index.

Assignment:

Que-1. What is Object oriented Databases?

Que-2. Explain fundamental features of Object oriented databases.