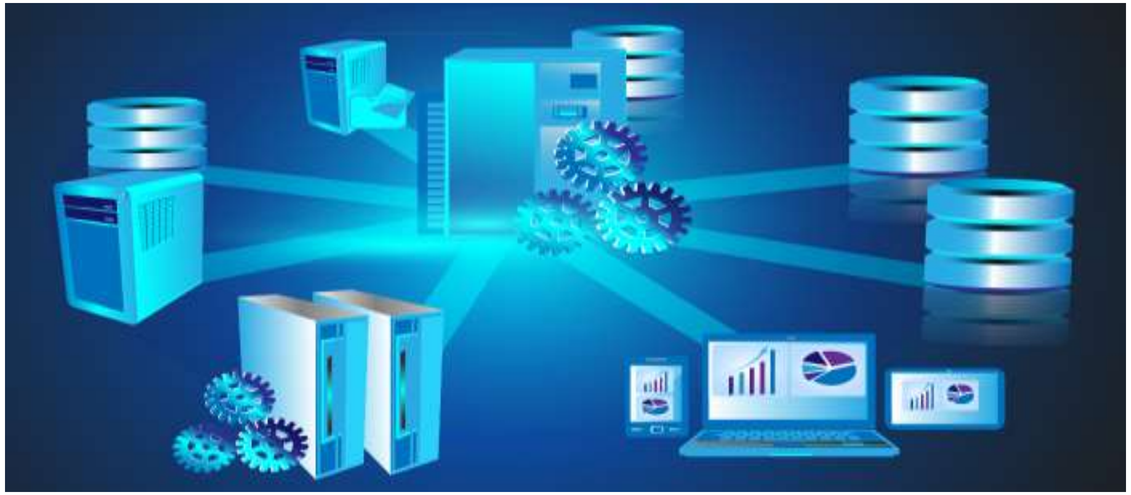
CSE716:Advanced Database

Lecture by A.M.Esfar-E-Alam



Course Outline

- Object-oriented Database
- Data Model, Design, Languages
- Object-Relational Database: Complex data types,
- Querying with complex data types,
- Design; Distributed Database: Levels of distribution transparency, Translation of global queries to fragment queries,
- Optimization of access strategies,
- Management of distributed transactions
- Concurrency control, reliability,
- Administration
- Parallel Database: Different types of parallelism,
- Design of parallel database;
- Multimedia Database Systems: Basic concepts,
- Design, Optimization of access strategies,
- Management of Multimedia Database Systems,
- Reliability; Database Wire-housing/Data mining:
- Basic concepts and algorithms.

BRAC UNIVERSITY
Inspiring Excellence

Object oriented database

- **Object**
- **Object Orieneted database**
- **ORDBMS**

Object

- An **object** is a real-world entity
- These entities have certain attributes that makes up the object structure.
- Also an object encapsulates the data code into a single unit which in turn provides data abstraction by hiding the implementation details from the user.
- There are generally two kinds of objects:
  - Transient object: that is creates and exists only during execution or runtime
  - Persistent Objects: An object that exists even after the program is completely executed (or terminated)
  - Object oriented database uses this kind of object to be able to store objects in secondary memory
- a **class** is a collection of objects.

Object oriented database

- 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:
- **Object-Oriented Programming + Relational Database Features = Object-Oriented Database Model**
- The figure below outlines the object-oriented database model along with its principles and features.

Object oriented database

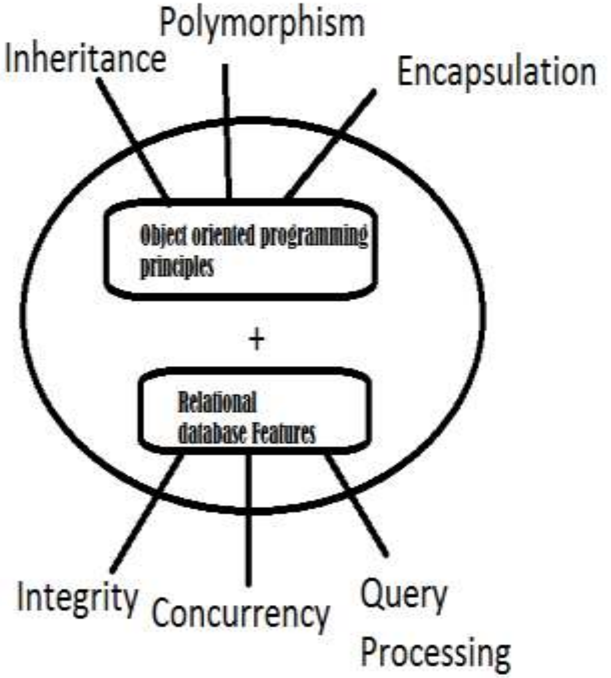
- **Some of the features of OODBMS are as follows:**

  1. Complexity
  OODBMS has the ability to represent the complex internal structure (of object) with multilevel complexity.

  2. Inheritance
  Creating a new object from an existing object in such a way that new object inherits all characteristics of an existing object.

  3. Encapsulation
  It is an data hiding concept in OOPL which binds the data and functions together which can manipulate data and not visible to outside world.

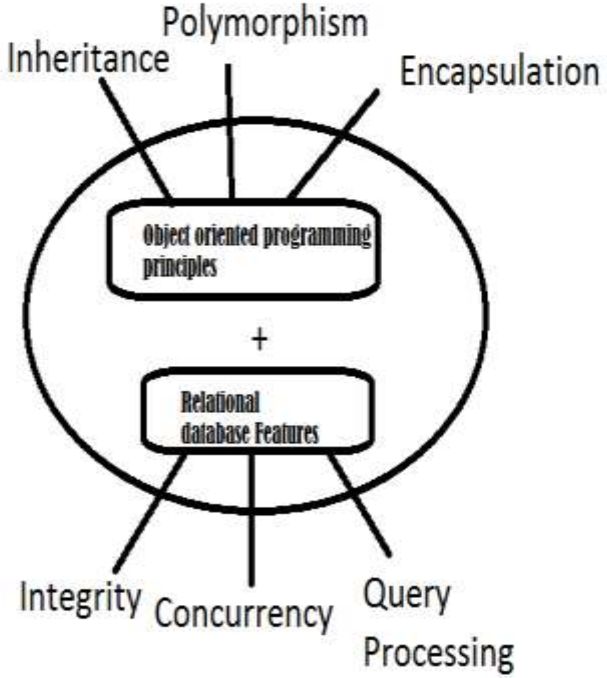
  4. Persistency
  OODBMS allows to create persistent object (Object remains in memory even after execution). This feature can automatically solve the problem of recovery and concurrency.



Object oriented database

5.User defined abstract data type(**ADT**):ADTs allow new data types with structures suited to particular applications to be defined

6.**Object Identity**: OIDs are used to identify objects



ORDBMS

- An Object relational model is a combination of a Object oriented database model and a Relational database model.
- So, it supports objects, classes, inheritance etc. just like Object Oriented models and has support for data types, tabular structures etc. like Relational data model.
- One of the major goals of Object relational data model is to close the gap between relational databases and the object oriented practises frequently used in many programming languages such as C++, C#, Java etc.

Advantages of Object Relational model

- The advantages of the Object Relational model are −
- **Inheritance**:The Object Relational data model allows its users to inherit objects, tables etc. so that they can extend their functionality. Inherited objects contains new attributes as well as the attributes that were inherited.
- **Complex Data Types**:Complex data types can be formed using existing data types. This is useful in Object relational data model as complex data types allow better manipulation of the data.
- **Extensibility**:The functionality of the system can be extended in Object relational data model. This can be achieved using complex data types as well as advanced concepts of object oriented model such as inheritance.

Object oriented model vs Relational Model

Object Oriented Model	Relational Model
Class	Relation/Table
Object/Object instance	Tuple/Record
Variable	Attribute/Coloumn
Method	Stored Procedure

Advantages of Object oriented database

- Enhanced modeling capabilities
- Extensibility
- Removal of impedance mismatch
- Expressive power
- Support for schema evolution
- Support for long duration transactions
- Applicability to advanced database applications
- Improved performance
- reusability

Disadvantages of Object oriented database

- Lack of universal data model and lack of standards
- Lack of experience
- Competition
- Query optimization compromises encapsulation
- Locking at object level may impact performance
- Complexity
- Lack of support for views
- Lack of support for security
- Poor performance
- Unable to support large scale system

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.
  Challenge : Storage of large ADTs and structured objects.
  Solution: As large ADTs need special storage, it is possible to store them on different locations on the disk from the tuples that contain them. **For e.g.** BLOBs (Binary Large Object like images, audio or any multimedia object.)
  Use of flexible disk layout mechanisms can solve the storage problem of structured objects.

  2. Query Processing

  Challenge: Efficient flow of Query Processing and optimization is a difficult task.
  Solution: By registering the user defined aggregation function, query processing becomes easier. It requires three implementation steps - **initialize, iterate and terminate.**

  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.
  Solution: While constructing a query plan, an optimizer must be familiar to the newly added index structures.

  For a given index structure, the optimizer must know:

  1. WHERE-clause conditions matched by that index.
  2. Cost of fetching a tuple for that index.

BRAC UNIVERSITY
Inspiring Excellence

Showcasing an Object oriented database

- **There are many object oriented database like realm, wakanda etc.**
- **I am going to demonstrate a object oriented database that I worked on**
- **Apple has practice to collect the best available features and practices to bring in their coding platform**
- **While object oriented database is not feasible in large scale system due to lack of experts and stability it is now widely used in mobile platforms**
- **Coredata is ios database system**

BRAC UNIVERSITY

Inspiring Excellence

Assignment 2

- **What is Impedance mismatch?Is it a advantage or disadvantage?Why so?**
- **What is Coredata stack?**
- **Why query optimization in ORDBMS is a challenge?**