Object Oriented Databases

OOAD Fall 2012

Arjun Gopalakrishna
Bhavya Udayashankar

Executive Summary

- The presentation on Object Oriented Databases gives a basic introduction to the concepts governing OODBs and looks at its details including its architecture, the query languages used etc. A contrast between OODBs and RDBs is also presented.
- The reader will gain insight into databases, data models, OODB architecture, Object Query Language, OODBMS.

Overview

- Databases
  - History
  - Types of Data Models
- Object Oriented Databases
  - Concepts
  - Architecture
  - ODL
  - OQL
  - OODB v/s Relational Database
  - OODBMS

Database I

- A database is a an organized collection of related data held in a computer or a data bank, which is designed to be accessible in various ways
  - The data within a database is structured so as to model a real world structures and hierarchies so as to enable conceptually convenient data storage, processing and retrieval mechanisms
  - Clients (Services or applications) interact with databases through queries (remote or otherwise) to Create, Retrieve, Update and Delete (CRUD) data within a database. This process is facilitated through a Database Management System (DBMS)

Database II

- Additionally, a DBMS also provides tools for maintenance such as running security checks, ensuring data integrity, backup and recovery.
- Although a database and its management system define different entities , they are inseparable and are crucial for business in all sectors of the modern world be it in technology oriented companies or hospitals and health care systems.

Database III

- Example: A database designed to store the location of all the class rooms in CU can be modeled to contain different data structures for each major building in CU and these data structures are used to store class room information

History I

- Computerized Databases evolved with DBMS in the 1960s with the availability of disks and drums to provide an easy alternative to maintaining large amount of diverse information.
- In the 1970s the main objective of database technology was to make the data independent of the logic of application programs to enable concurrent access to different application programs

History II

- The first generation of databases were navigational, where applications accessed data through record pointers moving from one record to another. This was a precursor to the IBMs hierarchical model (IMS System) and network model.
- This was followed by the relational model which placed the emphasis on content rather than links for data retrieval. This kind of database the most widely till date.

History III

- Relational models were limiting in the kind of data that could be held, the rigidity of the structure, and the lack of support for new data types such as graphics, xml, 2D and 3D data.
- In the 1980s With the advent of Object Oriented methodologies and languages, integration of database capabilities with object oriented programming language provided a unified programming environment. This led to the development of OODB and OODBMS where objects are stored in databases rather than data such as integers, strings or real numbers.

Data Models I

- The type of a database is decided by the data model used in he design of the database. Data models are data structures which describe how data are represented and accessed. Data models must be simple and intuitive to enable applications
- The major types of data models in the history of Databases are
  - **Hierarchical model** contains data organized into a tree-like structure.

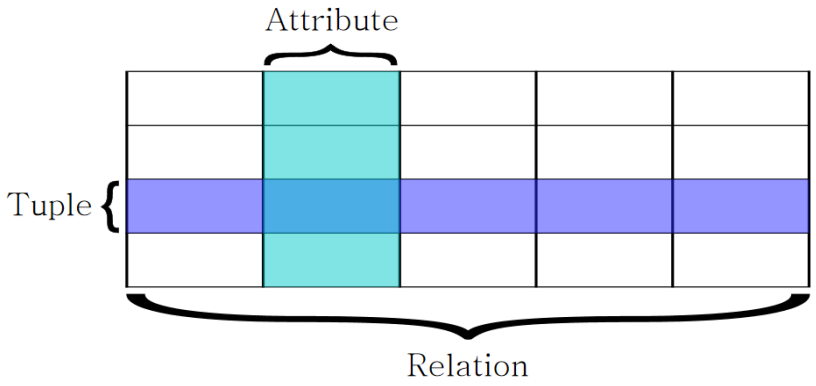
Data Models II

This supports parent-child relationships between data similar to a tree data structure where object types are represented by nodes and their relationships are represented by arcs. This model is restrictive in that it only allows one to many relationship (a parent can have many children but a child can only have one parent)

- **Network Model** is similar to the hierarchical model in representation of data but allows for greater flexibility in data access as it supports many to may relationships.

Data Models III

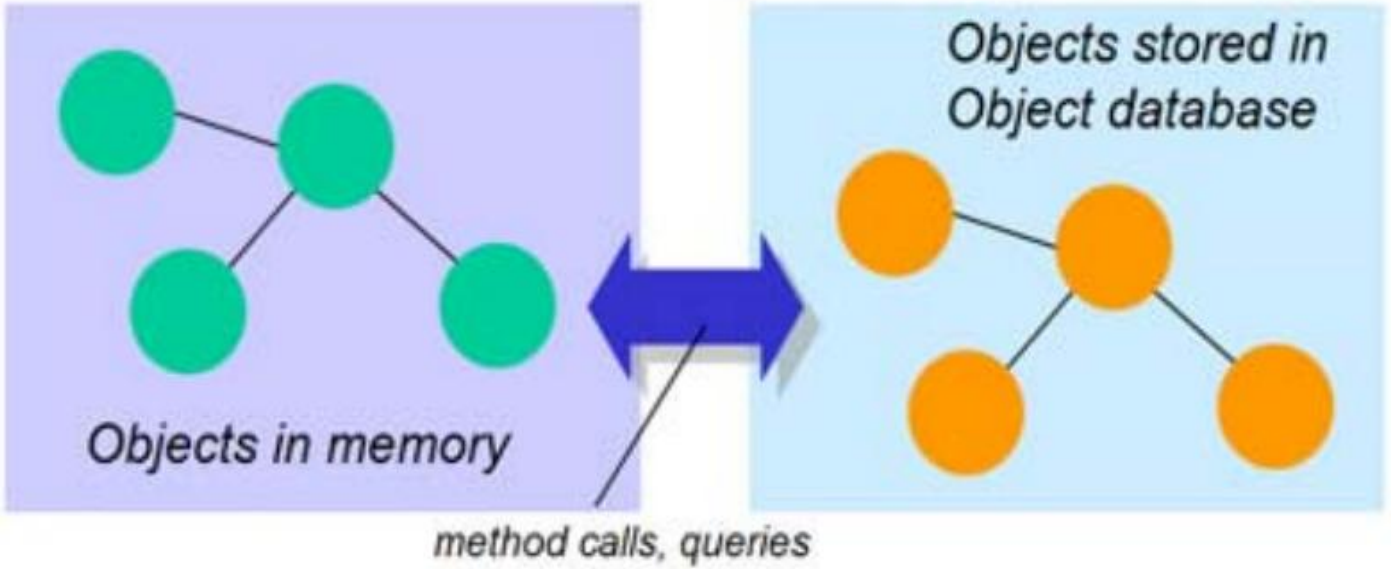
- **Relational Model** organizes data into two dimensional arrays known as relations(tables) and each relation consists of rows and columns. Another major characteristic of relational model is that of keys – designated columns in a relation used to order data or establish relations.



Data Models IV

- **Object Model** aims to reduce the overhead of converting information representation in the database to an application specific representation. Unlike a traditional database, an object model allows for data persistence and storage by storing objects in the databases. The relationships between various objects are inherent in the structure of the objects. This is mainly used for complex data structures such as 2D and 3D graphics which must otherwise be flattened before storage in a relational database.

Data Models V



Object Model

OO Database I

- Object oriented databases or object databases incorporate the object data model to define data structures on which database operations such as CRUD can be performed. They store objects rather than data such as integers and strings. The relationship between various data is implicit to the object and manifests as object attributes and methods
- Object database management systems extend the object programming language with transparently persistent data, concurrency control, data recovery, associative queries, and other database capabilities.

OO Database II

- **The Object-Oriented Database System Manifesto** by Malcolm Atkinson mandates that an object-oriented database system should satisfy two criteria: it should be a DBMS, and it should be an object-oriented system

- Thus OODB implements OO concepts such as object identity ,polymorphism, encapsulation and inheritance to provide access to persistent objects using any OO-programming language

OO Database III

- The tight integration between object orientation and databases provides programmers a unified environment when dealing with complex data such as 2D and 3D graphics.
- Object oriented databases are designed to work well with object oriented programming languages such as Python, Java, Objective-C.

OO DB Architecture I

- According to existing application requirements, OODB architecture can be divided into two categories:
  - Standalone OODB where all of the data available is stored in object data model. Thus there is no overhead in mapping objects to application objects. This is mostly useful for complex data.

OO DB Architecture II

- Object Relational DBMS: OODB acts as a staging layer for existing data in relational database. The data in relational database are mapped to object models and stored in object data database. Thus allowing application which require object models to tap into the object database and reduce overhead of mapping relational data to objects.
- In the first case, the database supports object inheritance similar to object oriented programming. This architecture is rare as the underlying design of the database is inefficient.

OO DB Architecture III

- Example: if class ‘A’ extends class ‘B’, they would have to be placed in different tables to allow for further inheritance of class ‘B’. This design then requires locks on all the classes involved and slow down transactions
- The second approach towards an OODB architecture is by enabling standard SQL database to support hierarchical data encapsulation.

Standards I

- Object Data Management Group (ODMG) put forward specifications with respect to data schema, programming language bindings, data manipulation and query languages required for the development of applications which work with object database.
  - Object Definition Language: ODL is the standardized language for defining the structure of database with respect to the object data model.

Standards II

- Example: Class declarations

  Interface < name > { elements = attributes, relationships, methods }

  Type Date Tuple {year, day, month}

Object Definition Language

- ODL creates a layer of abstraction making data language and database independent (Standalone OODB or Object Relational DB) to allow applications to move between compliant databases or different language implementations.
- ODL defines three components of the object oriented data model:
  - Abstraction
  - Inheritance
  - Encapsulation

Data Abstraction I

- In OODB, abstract data model is implemented as a graph, with vertices representing the objects and edges representing relations.
- Object instance: An entity in an object model is called an object instance.
- Object identification: Every object instance has a unique identity. This id used to reference object instances.
- Object Class: Similar object instances are grouped together into a class. The class defines the structure and the attributes are used to instantiate objects which conform to the classes specification.

Data Abstraction II

- Object reference or relationships: The object model directly supports references. Object instances ”reference” each other using object identities.

Encapsulation I

- Encapsulation in the object model concept allows for including processing or behavior with the object instances defined by the class. This allows code and data to be packaged together.
- This includes an interface and an implementation for each object. The interface to an object is visible to an application. The implementation consists of attributes which represent its state and methods, which represent the operations which can be performed on the object.

Inheritance

- Derived directly from object oriented programming languages, object data models also allow for inheritance, polymorphism and overriding.

Object Query Language I

- Developed by ODMG, Object Query Language allows SQL-like queries to be performed on a OODB.
- Like SQL, it is a declarative language. Based loosely on SQL, OQL includes additional language constructs which allow for object oriented design such as operation invocation and inheritance.
- Query Structures look very similar in SQL and OQL but the results returned are different.

Object Query Language II

- Example: OQL query to obtain Voter names who are from the state of Colorado

  Select distinct v.name

  From voters v

  Where v.state = “Colorado”

Object Query Language III

Voter Id	Name	State
V1	George Love	Colorado
V2	Winnie the Pooh	Florida
V3	John Lewis Hall	Colorado

Result from SQL table with rows

Name
George Love
John Lewis Hall

Result from OQL collection of objects

String	String
George Love	John Lewis Hall

Object Query Language IV

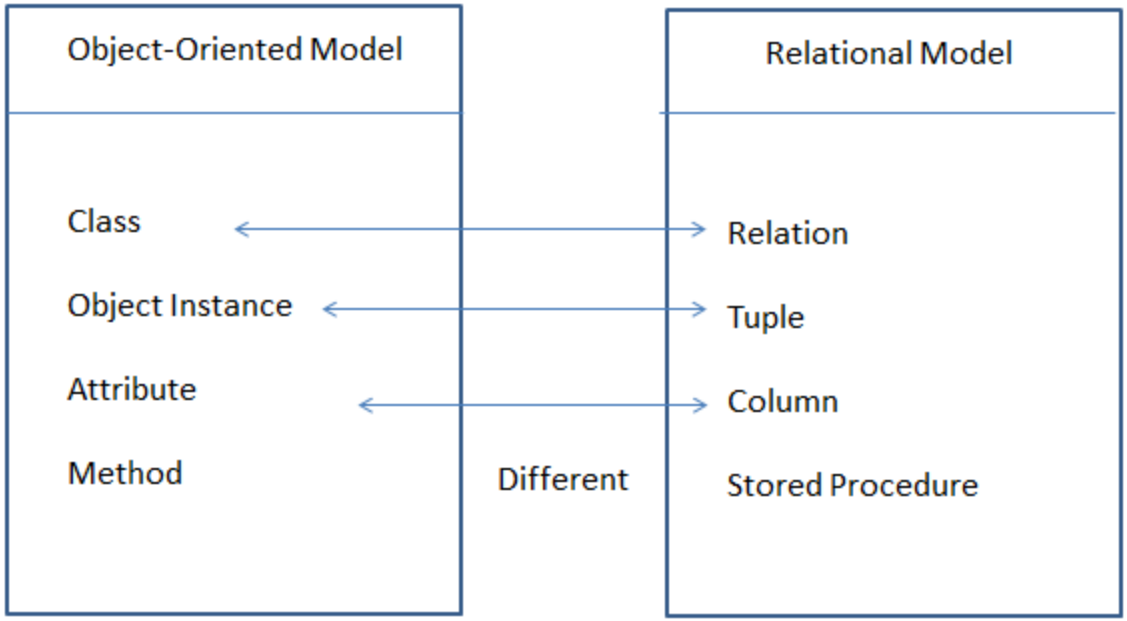
- More example of OQL with integration to OO Language:
  - Create objects as in OO languages and then make them persistent using the set() method on the database.

    ```
    Person p1 = new Person("Pikes Peak", 78);
    db.set(p1);
    ```

  - Retrieve by age (null default for string)

    ```
    Person p = new Person (null, 35);
    ObjectSet<Person> result = db.get(p);
    ```

Object v/s Relational Models



Object v/s Relational Models

- A method in an object model is defined in the class to which the object belongs.
- A stored procedure is a sub-routine available to applications and this is external to the database , defined in the data dictionary.
  - Example: Stored procedures for data validation
- OODB is OO language specific whereas Relational DB are language independent via SQL
- No impedance mismatch in applications using OODB where as object relational mapping must be performed in relational database for use in OO applications.

OODBMS

- Advantages
  - Can handle large collections of complex data including user defined data types. Thus support for aggregation, composition, reference etc.
  - Expressive data relationships
  - Version control for evolving classes and projects
  - Efficiently handles many-to-many relationships
- Real world OODBMS products:
  - db4o – database for objects
  - InterSystems
  - Objectivity.Inc
  - Versant

References

- Wikipedia
- The Object Oriented Database System Manifesto, Malcolm Atkinson et al
- Various other online sources