# UNIT-2

# Object and Object Relational Databases

## #Object Database Concepts: [Imp.]

Object-oriented database (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 programming + Relational database feature = Object-oriented database model. |

The main goal of OODB is to combine the advantages of object-oriented technologies and traditional databases. ~~they combine~~ It helps improving the modeling of real-world concepts in the database.

### ⊛Characteristics: [Features]:

→ It keeps up a direct relation between real world and database objects.
→ It provides system generated object identifier for each objects.
→ OODBS are extensible, which identifies new datatypes and operations to be performed on them.
→ Provides encapsulation.
→ Provides inheritance.
→ Overloading, Overriding and late Binding.

### ⊛ OODBMS Advantages:

→ Enhanced modeling capabilities: since data are more closer to real world.
→ Capable of handling a large variety of data types such as pictures, audio, video etc.
→ Support for long-duration transaction because there will not be any expiration in the session.
→ Improved performance.
→ More expressive Query language.

## ⊛ OODBMS Disadvantages:

→ Lack of universal data model because there is no universally agreed data model.
→ Not easy for less experienced end-users.
→ Lack of standard.
→ Lack of support for security.
→ Implementation is complex.

## #Object Database Extensions to SQL: [Imp.]

SQL is a standard language for RDBMS. In SQL3 features of object-oriented databases were incorporated into SQL standards. Following are some of object database features that have been included in SQL.

i) Type constructor have been added. Basically two types of constructor:
Row type → tuple (or struct) constructor.
Collection type → set, list, and bag constructor.

ii) Mechanism of specifying object identifier through one of refrence type is included.

iii) Encapsulation of operation is provided through mechanism of user-defined types.

iv) Inheritence mechanisms are provided using keyword UNDER.

## #The ODMG Object Model:

ODMG stands for Object Data Management Group. It is a data model upon which the Object definition language (ODL) and Object query language (OQL) are based. It is meant to provide a standard data model for object database.

ODMG is made up of parts, some of which are as follows:
→ the object model.
→ object definition language (ODL)
→ object query language (OQL).
→ binding to object-oriented programming language.

ODMG Characteristics:

→ Easy to link with programming language.
→ No need for user defined keys.
→ Easy modeling.
→ Can store non-textual data.

Advantages:

i) Speed: Access to data can be faster because an object can be retrived directly.

ii) Improved performance: best for object oriented programming.

iii) Extensibility: any kinds of data structure are used to hold data.

iv) Capable of handling variety of data.

v) Avoids the impedance mismatch.

⊛ Differentiate objects and literals in ODMG:

| Objects | Literals |
|---|---|
| i) Objects have both object identifier and stats (current data). | i) Literals has a value (state) but no object identifier. |
| ii) Values can be modified. | ii) Constant value. |
| iii) Object may be transient or persistant. | iii) Literals may be atomic, structure, and collection. |

## # Object Definition Language (ODL):

Object Definition Language (ODL) is a language used in the context of object-oriented databases to define the structure and characteristics of objects within a database. In ODL, data is organized and stored as objects rather than in traditional relational tables. Some common elements found in an ODL include: Classes, Attributes, Relationships, Inheritance, Methods, Constraints.

ODL provides a powerful and expressive language for defining object-oriented database schemas. It enables the modeling and organization of data in a way that closely aligns with the principles of object-oriented programming, offering flexibility, modularity, and extensibility in database design.

Class Declarations:

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

Element Declarations:

attributes (<type>: <name>);
relationships (<rangetype>: <name>);

Example:

```
Type Data Tuple (Year, day, month)
Type Year, day, month integer
Interface Manager {
  Keys id;
  Attribute String name;
  Attribute String phone;
  relationship set <Employee> employees inverse Employee :: manager}
  interface Employee {
      keys id;
      attribute String name;
      attribute Date Start Date;
      relationship Manager inverse Manager :: employee
  }
```

## # Object Database Conceptual Design:

Object database conceptual design is the first stage in the object database design process. The goal at this stage is to design a database that is independent of database software and physical details. The conceptual design has four steps, which are as follows:

1) Data Analysis and Requirements: The first step in conceptual design is to discover the characteristics of the data elements for information.

↳ Information needs: What kind of information is needed?
↳ Information users: Who will use the information?
↳ Information sources: Where is the information to be found?
↳ Information constitution: What data elements are needed to produce the information?

## 2> Entity Relationship Modeling and Normalization:

For ER model, the designer must communicate and enforce appropriate standards to be used in the documentation of design.
↳ Identify, analyze, and refine business rules.
↳ Identify the main entities.
↳ Define relationships among entities.
↳ Define attributes, primary keys, and foreign keys for each of the entities.
↳ Normalize the entities.
↳ Complete the initial ER diagram.
↳ Validate ER model
↳ Modify ER model.

## 3> Data Model Verification:

In this step, ER model must be verified against the proposed system. Verification requires that the model be run through a series of tests against:
↳ End-user data views.
↳ All required transactions: SELECT, INSERT, UPDATE, and DELETE operations.
↳ Access rights and security.
↳ Business-imposed data requirements and constraints.

## 4> Distributed Database Design:

Although not a requirement for most databases, sometimes a database may need to be distributed among multiple geographically disperse locations. If the database data and processes are to be distributed across the system, portions of a database, known as database fragments, may reside in several physical locations.

## # Object Query Language (OQL): [Imp.]

Object Query Language (OQL) is an SQL like declarative language that provides a rich environment for efficient querying of database objects, including high-level primitives for object sets and structures.

The OQL syntax for queries is similar to the syntax of relational Standard Query Language (SQL), with additional features of ODMG concepts such as object identity, complex objects,

## 2> Entity Relationship Modeling and Normalization:

For ER model, the designer must communicate and enforce appropriate standards to be used in the documentation of design.

↳ Identify, analyze, and refine business rules.
↳ Identify the main entities.
↳ Define relationships among entities.
↳ Define attributes, primary keys, and foreign keys for each of the entities.
↳ Normalize the entities.
↳ Complete the initial ER diagram.
↳ Validate ER model
↳ Modify ER model.

## 3> Data Model Verification:

In this step, ER model must be verified against the proposed system. Verification requires that the model be run through a series of tests against:

↳ End-user data views.
↳ All required transactions: SELECT, INSERT, UPDATE, and DELETE operations.
↳ Access rights and security.
↳ Business-imposed data requirements and constraints.

## 4> Distributed Database Design:

Although not a requirement for most databases, sometimes a database may need to be distributed among multiple geographically disperse locations. If the database data and processes are to be distributed across the system, portions of a database, known as database fragments, may reside in several physical locations.

## # Object Query Language (OQL): [Imp.]

Object Query Language (OQL) is an SQL like declarative language that provides a rich environment for efficient querying of database objects, including high-level primitives for object sets and structures.

The OQL syntax for queries is similar to the syntax of relational Standard Query Language (SQL), with additional features of ODMG concepts such as object identity, complex objects, operations, inheritance, polymorphism and relationships. It is designed to work closely with programming languages for which an ODMG binding is defined such as C++, Java etc.

Example: Query to retrive the names of all departments in the college of 'Engineering' can be written as follows:

```
SELECT   D.Dname
FROM     D in Departments
WHERE    D.College = 'Engineering';
```

Here, Departments is entry point which refers to a persistent collection of objects. Whenever a collection is refrenced in as OQL query we should define a iterator variable D, that ranges over each object in collection.

⇒ A query does not have to follow SELECT.--- FROM--- WHERE structure in the simplest case:

Query 1 ⟷ Q1: DEPARTMENTS;
returns refence to the collection of all persistent DEPARTMENTS objects whose type is set <DEPARTMENT>

Q2: CS_DEPARTMENT;
returns refrence to the individual object of type DEPARTMENT

⇒ Names or attribute names are connected using dot notation.

```
Q3:  CS_DEPARTMENT.Chair
Q3A: CS_DEPARTMENT.chair.Rank
Q3B: CS_DEPARTMENT.Has-faculty.
```

## # Language Binding in the ODMG Standard:

Language binding in the ODMG standard enables the integration of programming languages with ODMG-compliant databases, enabling developers to work with objects using their preferred programming language through a standardized interface.
In the ODMG standard, language bindings are defined for several programming languages such as C++, Java, and Smalltalk.

i) C++ Language Binding: The ODMG C++ language binding provides language transparent extensions that provide facilities for object creation, naming, manipulation, deletion, transactions, and database operations. The C++ binding allows persistence capable classes created by inheritance.

ii) JAVA Language Binding: This binding uses established Java language practice and style to be natural to the Java environment and programmers. It adds classes and other constructs to the Java environment to support the ODMG object model including collections, transactions and databases without altering the semantics of the language.

iii) Smalltalk Language Binding: It provides ability to store, retrive and modify persistent objects in smalltalk. The binding includes a mechanism to invoke OQL and producers for transactions and operations on databases. This binding directly maps all the classes, ~~a~~ relationships, transactions, and database operations to standard collection classes and methods.

⊛ Differences between OODBMS and RDBMS: [Imp]

| OODBMS | RDBMS |
|---|---|
| i) Data in OODBMS is stored as objects. | i) Data in RDBMS in stored in tabular format. |
| ii) It can handle large and complex data | ii) It handles simple data. |
| iii) OODBMS stores data as well as methods to use it. | iii) RDBMS stores data only. |
| iv) OID is used to uniquely identify objects. | iv) Primary key is used to identify objects. |
| v) OODBMS provides powerful query languages such as OQL. | v) RDBMS typically use SQL for querying and manipulating data. |

## #What is object relational model?

Ans: The object-relational model (ORM) is a data model that combines features of both object-oriented programming (OOP) and relational database management systems (RDBMS). It aims to bridge the gap between the OOP and RDBMS.

In the object relational model, data is organized into objects that have attributes (properties) and methods (functions) just like in OOP. These objects can be directly mapped to relational database tables, where each object corresponds to a row in the table, and each attribute corresponds to a column.

Key Features:

i) Inheritance: (allows objects to inherit attributes and behaviours).
ii) Complex Data Types: (arrays, sets, and user-defined types).
iii) Object Identity: (have a unique identity).
iv) Methods and Functions:
v) Querying:

## #Define state of an object. Distinguish between persistent and transient objects. [Imp.]

Ans: The state of an object refers to the set of values stored in it's attributes or properties at any given time. The state defines the current condition or snapshot of an object, including the values of it's data and any other relevant information.

| Persistent Objects | Transient Objects |
|---|---|
| i) These are the objects with long-term existence. | i) These are the objects with short-term existence. |
| ii) State is stored in persistent medium (e.g. database). | ii) State is not automatically stored beyond program execution. |
| iii) Exists beyond the scope of a single program execution. | iii) Exist only during the execution of a program or specific operation. |
| iv) It enables data persistance and durability | iv) It is used for temporary data processing, immediate computations etc. |
| v) State is retained across different program invocations. | v) State is lost once the program or operation ends. |