

UNIT-V

Distributed object Database Management Systems

Object Oriented DataModel

Distributed Object DBMS – Introduction

- ▶ A Distributed Object Database Management System (DODBMS) manages objects distributed across multiple networked sites.

Fundamental Object Concepts

- ▶ Objects
- ▶ Classes
- ▶ Methods
- ▶ Attributes
- ▶ Encapsulation
- ▶ Abstraction

Object Models

- ▶ Object-Based Model
- ▶ Object-Oriented Model
- ▶ Complex Objects
- ▶ Relationships and Aggregation

Object Distributed Design

- ▶ Object Fragmentation
- ▶ Object Allocation
- ▶ Object Replication
- ▶ Distributed Access Strategies

Architectural Issues in DODBMS

- ▶ Transparency
- ▶ Scalability
- ▶ Reliability
- ▶ Security
- ▶ Interoperability

Object Management

- ▶ Object Creation
- ▶ Object Deletion
- ▶ Object Versioning
- ▶ Object Sharing
- ▶ Transaction Management

Distributed Object Storage

- ▶ Object Placement
- ▶ Storage Allocation
- ▶ Object Replication
- ▶ Distributed Storage Management

Object Query Processing

- ▶ Object Query Languages
- ▶ Path Expressions
- ▶ Query Optimization
- ▶ Distributed Query Execution

Object-Oriented Data Model

- ▶ Represents data as objects with attributes and methods.
- ▶ Supports inheritance, encapsulation, and polymorphism.

Inheritance

- ▶ Single Inheritance
- ▶ Multiple Inheritance
- ▶ Hierarchical Inheritance
- ▶ Code Reusability

Object Identity

- ▶ Unique identifier assigned to each object.
- ▶ Independent of object state and values.

Persistent Programming Languages

- ▶ Languages supporting persistent objects.
- ▶ Examples: Java Persistence, ObjectStore, db4o.

Persistence of Objects

- ▶ Object Lifetime Beyond Program Execution.
- ▶ Persistent Storage and Retrieval.
- ▶ Object Serialization.

Comparison: OODBMS vs ORDBMS

- ▶ OODBMS:
 - ▶ – Pure object model
 - ▶ – Complex object support
- ▶ ORDBMS:
 - ▶ – Relational + Object features
 - ▶ – SQL support

Advantages of OODBMS

- ▶ Better support for complex data
- ▶ Object reusability
- ▶ Closer integration with object-oriented programming

Challenges of OODBMS

- ▶ Lack of standardization
- ▶ Complex implementation
- ▶ Limited adoption compared to relational systems

Conclusion

- ▶ DODBMS and Object-Oriented Data Models provide efficient management of complex distributed objects and applications.