• Object-Oriented Language Features: abstract data types inheritance object identity • Object-Oriented Database Features: persistence support of transactions simple querying of bulk data concurrent access resilience security

WHY OBJECT-ORIENTED DATABASE?

Vision Mission Problem

• Industry Trends: Integration and Sharing

• Seamless integration of operating systems, databases, languages, spreadsheets, word processors, AI expert system shells.

• Sharing of data, information, software components, products, computing environments.

• Referential sharing: Multiple applications, products, or objects share common sub-objects. (Hypermedia links are then used to navigate from one object to another) Object-oriented databases allows referential sharing through the support of object identity and inheritance.

Fundamentals of Object-Oriented Approach The object-oriented paradigm is illustrated below: Objects and Identity The following figure shows object with state and behavior. The state is represented by the values of the object's attributes, and the behavior is defined by the methods acting on the state of the object. There is a unique object identifier OID to identify the object. Complex Objects Complex objects are built by applying constructors to simpler objects including: sets, lists and tuples. An example is illustrated below: Encapsulation Encapsulation is derived from the notion of Abstract Data Type (ADT). It is motivated by the need to make a clear distinction between the specification and the implementation of an operation. It reinforces modularity and provides a form of logical data independence. Class A class object is an object which acts as a template. It specifies: A structure that is the set of attributes of the instances A set of operations A set of methods which implement the operations Instantiation means generating objects, Ex. 'new' operation in C++ Persistence of objects: Two approaches An implicit characteristic of all objects An orthogonal characteristic - insert the object into a persistent collection of objects Inheritance A mechanism of reusability, the most powerful concept of OO programming Association Association is a link between entities in an application In OODB, associations are represented by means of references between objects a representation of a binary association a representation of a ternary association reverse reference

ADVANTAGES OF OODB

• An integrated repository of information that is shared by multiple users, multiple products, multiple applications on multiple platforms.

• It also solves the following problems: 1. The semantic gap: The real world and the Conceptual model is very similar. 2. Impedance mismatch: Programming languages and database systems must be interfaced to solve application problems. But the language style, data structures, of a programming language (such as C) and the DBMS (such as Oracle) are different. The OODB supports general purpose programming in the OODB framework. 3. New application requirements: Especially in OA, CAD, CAM, CASE, object-orientation is the most natural and most convenient

Solution

Product / Service Features

Business Model

Competitor Analysis

Revenue

Income

Cash Flow

Team Composition

Mentor