

Data Base Management System

Welcome



Object Based Database



Atul Anand



BETN1CS15028



CSE-A(2nd Yr.)



Outline

- Object definitions
- Object Structures
- Object-oriented concepts
- OODBS
- OQL with an example
- SQL3 with examples



Definition of an object

Objects – User defined complex data types

- An object has structure or state (variables) and methods (behavior/operations)

An object is described by four characteristics

- Identifier: a system-wide unique id for an object
- Name: an object may also have a unique name in DB (optional)
- Lifetime: determines if the object is persistent or transient
- Structure: Construction of objects using type constructors

Object Oriented Database Systems

- Motivation: Reduce impedance mismatch, support for querying and indexing and addressing version management.

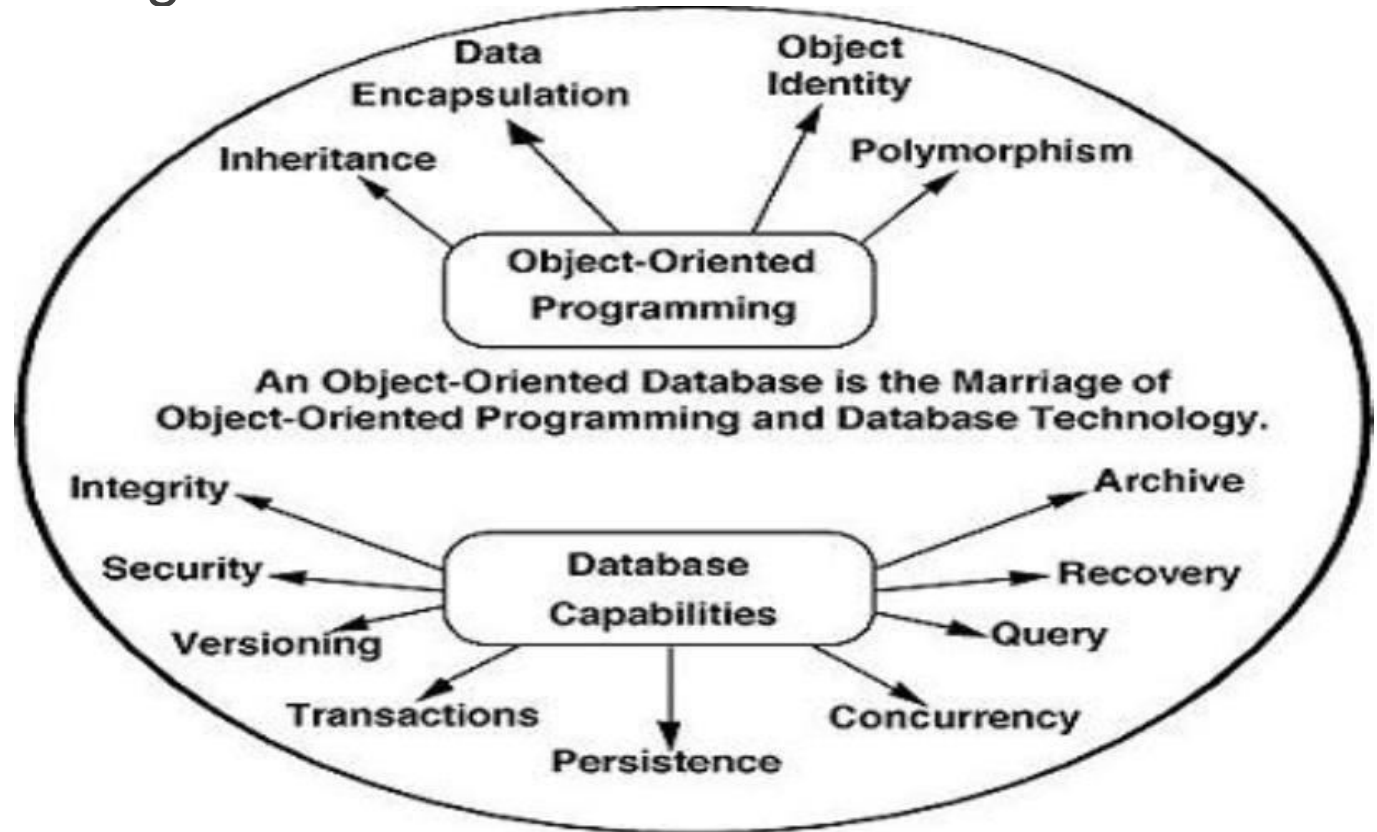


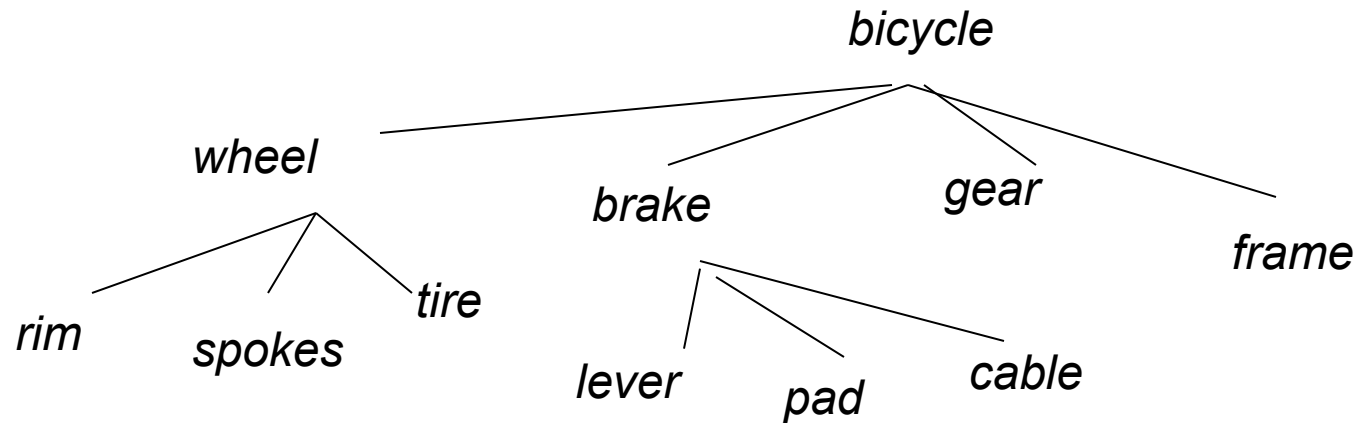
Figure 1. Makeup of an Object-Oriented Database



Object-Oriented Concepts

- **Abstract Data Types**
 - Class definition, provides extension to complex attribute types
- **Encapsulation**
 - Implementation of operations and object structure hidden
- **Inheritance**
 - Sharing of data within hierarchy scope, supports code reusability
- **Polymorphism**
 - Operator overloading

Object Containment



- Each component in a design may contain other components
- Can be modeled as containment of objects. Objects containing other objects are called *complex* or *composite* objects.
- Multiple levels of containment create a *containment hierarchy*: links interpreted as **is-part-of**, not **is-a**.
- Allows data to be viewed at different granularities by different users.



What is Object Oriented Database? (OODB)

- A database system that incorporates all the important object-oriented concepts
- Some additional features
 - Unique Object identifiers
 - Persistent object handling





Advantages of OODBS

- Designer can specify the structure of objects and their behavior (methods)
- Better interaction with object-oriented languages such as Java and C++
- Definition of complex and user-defined types
- Encapsulation of operations and user-defined methods

Object Query Language (OQL)



- Declarative query language
 - Not computationally complete
- Syntax based on SQL (select, from, where)
- Additional flexibility (queries with user defined operators and types)



Example of OQL query

The following is a sample query

“what are the names of the black product?”

Select distinct p.name

From products p

Where p.color = “black”

⇒ Valid in both SQL and OQL, but results are different.




Result of the query (SQL)

Original table

Product no	Name	Color
P1	Ford Mustang	Black
P2	Toyota Celica	Green
P3	Mercedes SLK	Black

Result



Name
Ford Mustang
Mercedes SLK

- The statement queries a relational database.

=> Returns a table with rows.



Result of the query (OQL)

Original table

Product no	Name	Color
P1	Ford Mustang	Black
P2	Toyota Celica	Green
P3	Mercedes SLK	Black

Result

String	String
Ford Mustang	Mercedes SLK

- The statement queries a
object-oriented database

=> Returns a collection
of objects.



Comparison

- Queries look very similar in SQL and OQL, sometimes they are the same
- In fact, the results they give are very different

Query returns:

OQL	SQL
Object	Tuple
Collection of objects	Table

SQL3 “Object-oriented SQL”



- Foundation for several OO database management systems – ORACLE8, DB2, etc
- New features – “relational” & “Object oriented”
- Relational Features – new data types, new predicates, enhanced semantics, additional security and an active database
- Object Oriented Features – support for functions and procedures



User defined Data Types

Creating a “row type”

Example:

```
create row type AddressType(  
    street      char(50),  
    city        char(20));
```

```
create row type StarType(  
    name        char(30),  
    address     AddressType);
```


Creating Data Types (contd.)

Creating “Table”

```
create table Address of type  
AddressType;
```

```
create table MovieStar of type  
StarType;
```

Instances of Row types are tuples in
tables



Sample Query

Find the names and street addresses of those MovieStars who stay in the city “Columbus”:

```
select MovieStar.name,  
  
        MovieStar.address.street  
from   MovieStar  
where MovieStar.address.city =  
        “Columbus”;
```



Complex Data and Queries

A Water Resource Management example

- A database of state wide water projects
- Includes a library of picture slides
- Indexing according to predefined concepts – prohibitively expensive
- Type of queries
 - Geographic locations
 - Reservoir levels during droughts
 - Recent flood conditions, etc

Complex Data and Queries (contd.)

- Addressing these queries
 - Linking this database to landmarks on a topographic map
 - Examining the captions for each slide
 - Implementing image-understanding programs
 - Inspecting images and ascertaining attributes
- These type of queries necessitate dedicated “methods”



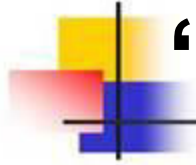
Creating Functions

**create function one() returns int4
as 'select 1 as RESULT'
language 'sql';**

select one() as answer;

answer
1

Creating “tables” with “methods”




Implementation

create table slides (

```
id int,  
date date,  
caption document,  
picture CD_image,  
method containsName  
    (name varchar)  
returns boolean  
as external name 'matching'  
language 'C'      );
```



Creating Tables (Contd.)



```
create table landmarks(  
           name      varchar (30),  
           location   point);
```



Implementation (contd.)

Sample query – find a picture of a reservoir with low water level which is in “Sacramento”

select P.id

from slides P, landmarks L

where IsLowWaterLevel (P.picture)
and

P.containsName (L.name) and
L.name = “Sacramento”;



Atul Anand



BETN1CS15028



CSE-A(2nd Yr.)

Thank You

