

ADBMS

UNIT 1

Motivation for complex data types
User defined abstract data types
Structured types

COMPILED BY: NIKHIL PAWANIKAR, UDIT

Motivation

- Support more complex kinds of data
- Two paths:
 - Object-Oriented Database Systems
 - Object-Relational Database Systems

Object-Oriented database system

are proposed as an alternative to relational system and aimed at application domains where complex objects play a central role.

Object-relational database systems

can be thought of as an attempt to extend relational database system with the functionality necessary to support a broader class of application...

- RDBMS
 - Relational Database Management Systems
- OODBMS
 - Object-Oriented Database Management Systems
- ORDBMS
 - Object-Relational Database Management Systems
- ODBMS
 - Object-Database Management Systems

Object-Oriented DBS

- Good for apps where complex objects play a central role
- Influenced by object-oriented PLs
- An attempt to add DBMS functionality to a PL environment
- Object Data Model & Object Query Language are developed

Object Relational DBS

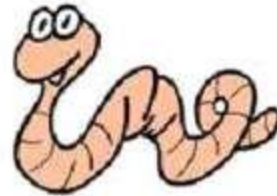
- Extend relational database systems with the functionality necessary to support a broader class of applications
- Bridge between relational and object-oriented paradigms
- RDBMS vendors are adding ORDBMS functionality

Advantages & Disadvantages

- ODBMS:
Navigational interfaces:
 - implemented efficiently
by pointers
 - optimized for specific
search routes or
viewpoints
- RDBMS:
Declarative interfaces:
 - better for general
purpose queries

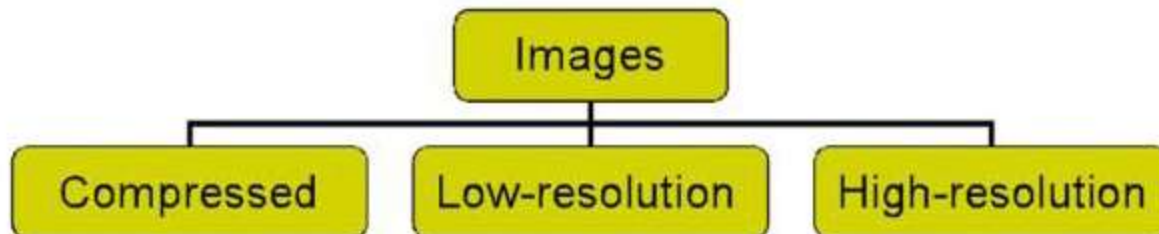
Motivating Example

- Dinky Entertainment Company
- Assets: collection of cartoon characters
- Especially: Herbert the Worm
- Also licensing Herbert's image, voice, video
- DB: products & films



New Data Types

- User-define abstract data type (ADT)
 - Dinky's assets include Herbert's image, voice, and video footage, and these must be stored in the database
 - Must write compress functions to support (lower resolution).



- **Structured types**

- In this application, as needed in many traditional business data processing applications, we need new types built up from atomic types using constructors for creating sets, records, arrays, sequences, and so on.. (see section 25.5)

- **Inheritance**

As the number data types grows, it is important to recognize the commonality between different types and to take advantage of it.

BLOB (A binary large object)

- ✓ is just a long stream of bytes, and DBSM's support consists of storing and retrieving BLOBs in such manner -
- ✓ Users don't worry the size of the BLOB
- ✓ BLOB can span several pages
- ✓ Unlike a traditional attributes
- ✓ Can be done in the host language in which the SQL code is embedded.

Manipulating the New Data

- Cereal Company: Clog wants to lease an image of Herbert in front of a sunrise



Frames(FrameNo: integer, image: jpeg_image, category: integer)

Categories(cid: integer, name: text, lease_price:float, comments: text)

```
SELECT  F.frameNo, thumbnail(F.image),C.lease_price
FROM    Frames F, Categories C
WHERE   F.categories = C.cid AND is_sunrise(F.image) AND
is_herbert(F.image)
```

Manipulating the New Data

- Dinky executives want to make sure that a number of Herbert films are playing at theaters near Andorra when the cereal hits the shelves

```
SELECT N.theater->name, N.theater->address, F.title
FROM Nowshowing N, Films F, Countries C
WHERE N.film = F.filmno      AND
overlaps(C.boundary,radius(N.theater->address,100))
AND C.name = 'Andora' AND 'Herbert the worm'
element F.stars
```

theater attribute in *Nowshowing* is a reference to an object in another table
stars attribute of the *Films* table is a set of names of each film's stars
Nowshowing and *Films* are joined by the equijoin clause
Nowshowing and *Countries* are joined by the spatial overlap clause

Additional Features

- User Defined Methods:
User defined types are manipulated via their own Methods, for example, *is_herbert*
- Operators for Structured Types:
For example, the ARRAY type supports standard array operation of accessing an element
- Operators for Reference types:
Reference types are dereferenced via an arrow (->) notation

Summary

- Traditional relational systems offer limited flexibility in the data types available
- Data is stored in tables and the type of each field value is limited to simple atomic types
- Limited type systems can be extended in three ways (complex types):
 - User-defined data types
 - Structured types
 - Reference types

User-Defined Abstract Data types

- JPEG standard
 - Jpeg_image
 - boundary (a column of type polygon)
- ORDBMS key feature is allowing users to define arbitrary new data type. Such as:
 - compress, rotate, shrink and crop
- **Combination of an atomic data type and its associated methods is called: Abstract data type (ADT).** Such object-relational systems, user allows to include ADT.

Defining Methods of n ADT

The user who creates a new atomic type must register the following methods:

- Size – returns size of the object
- Import – creates new items of this type from textual inputs like (insert statements)
- Export-Maps items for output like printing or for use in an application program

Structured Types – Type Constructors

- ROW
 - A type representing a row, or record, of n field with fields n of type n....etc
- Listof(base)
 - Type representing a sequence of base-type items
- Array(base)
 - A type representing an array of base-type items
- Setof(base)
 - A type representing a set of base-type items. Sets cannot contain duplicate elements.
- bagof(base)
 - A type resenting a bag or multiset of based-type items.

Manipulating Data of Structure Types

- Done by built in methods provided by DMBS

Built-in Operators for Structure types

- Rows(field extraction method (.))
 - ROW(n1 t1,...,) Dot notation, i.n of k.
- Sets and multisets
 - Set of objects can be compared using the traditional set methods.
 - An item of type setof(foo) can be compared with an item of type foo using the Element method.

- Lists

- Traditional list operations include head, which return the first element.

- Arrays

- Array types support an 'array index' method to allow users to access array items at a particular offset.

- Others operators

- Count, Sum, Avg, Max & Min.

EXAMPLE

```
SELECT F.filmno  
FROM Films F  
WHERE count(F.stars) > 2
```

```
SELECT F.filmno, F.title, S AS star  
FROM Films F, F.star AS S
```