Fundamentals of Database Systems

[Object-oriented and Multimedia Databases]

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

November, 2019

Basics

Object-oriented databases use some of the concepts of object-oriented languages in defining data types.

Basics

Object-oriented databases use some of the concepts of object-oriented languages in defining data types.

An object typically has two components: state (value) and behavior (operations).

Basics

Object-oriented databases use some of the concepts of object-oriented languages in defining data types.

An object typically has two components: state (value) and behavior (operations).

Object-oriented databases are built around persistent programming languages but object-relational databases (which is a type of object-oriented databases) are built on top of the relation model.

Features of object-relational data models

- Extending the relational data model by including object orientation and constructs to deal with added data types.
- Allowing attributes of tuples to have complex types, including non-atomic values such as nested relations.
- Preserving relational foundations, in particular the declarative access to data, while extending modeling power.
- Getting upward compatibility with existing relational languages.

Complex data types

Motivation:

- To permit non-atomic (divisible) domains
- To allow more intuitive modeling for applications with complex data

Complex data types

Motivation:

- To permit non-atomic (divisible) domains
- To allow more intuitive modeling for applications with complex data

It violates the 1NF!

Object structure

In object-oriented databases, the state of a complex object can be constructed from other objects by using type constructors.

Object structure

In object-oriented databases, the state of a complex object can be constructed from other objects by using type constructors.

An object can be represented as a triplet (i, c, v), where i is the unique object identifier, c is a type constructor and v is the value.

Object structure

In object-oriented databases, the state of a complex object can be constructed from other objects by using type constructors.

An object can be represented as a triplet (i, c, v), where i is the unique object identifier, c is a type constructor and v is the value.

Example of some valid object representations are provided below:
$(i_1, atom, 1729)$
$(i_2, atom, \text{`Ramanujan'})$
$(i_3, set, \{i_1, i_2\})$

Nested relation and decomposition

Course	Subject	Faculty list
PGDBA	Fundamentals of Database Systems	{Debapriyo, Malay}
PGDBA	Inference	{Amitava}

Course	Subject
PGDBA	Fundamentals of Database Systems
PGDBA	Inference

Subject	Faculty
Fundamentals of Database Systems	Debapriyo
Fundamentals of Database Systems	Malay
Inference	Amitava

Structured types in SQL

```
 create type Name as
    (FirstName varchar(20),
    LastName varchar(20)
    )
final
create type Address as
    (City varchar(20),
    District varchar(20),
    PIN varchar(20)
    )
not final
```

Structured types in SQL

```
 create type Name as
    (FirstName varchar(20),
    LastName varchar(20)
    )
final
create type Address as
    (City varchar(20),
    District varchar(20),
    PIN varchar(20)
    )
not final
```

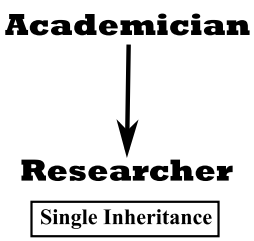
Note: `final` indicates whether subtypes can be created.

Type inheritance

Inheritance is acquiring the properties of another type

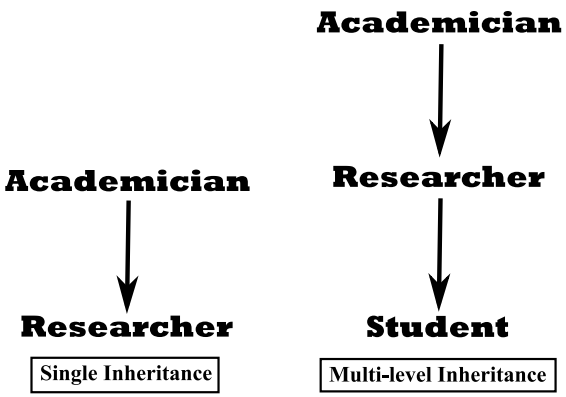
Type inheritance

Inheritance is acquiring the properties of another type



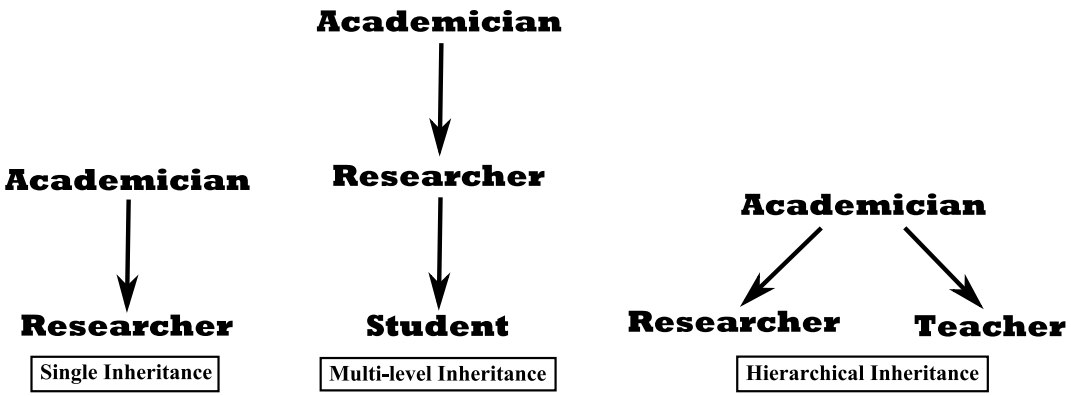
Type inheritance

Inheritance is acquiring the properties of another type



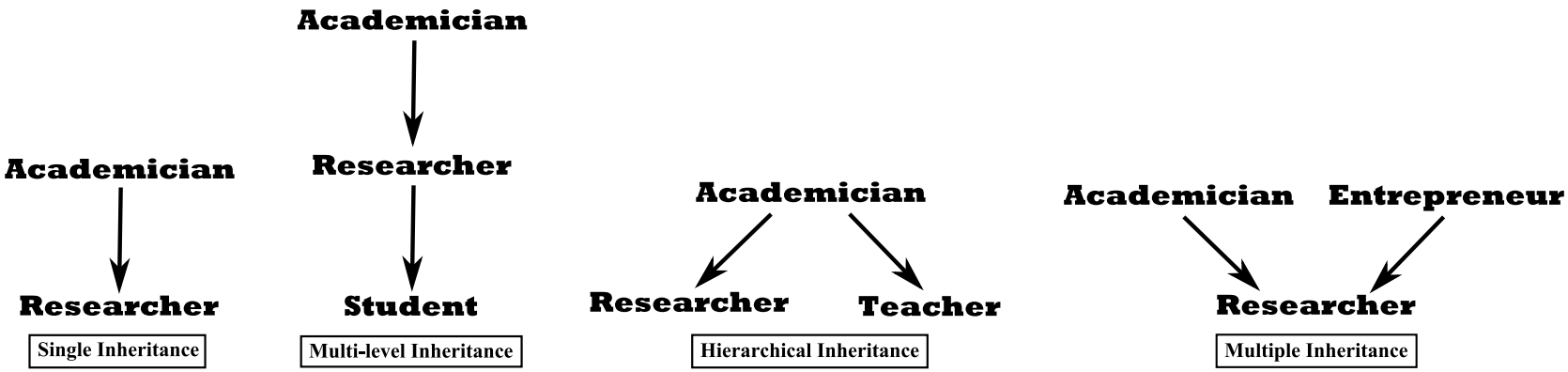
Type inheritance

Inheritance is acquiring the properties of another type



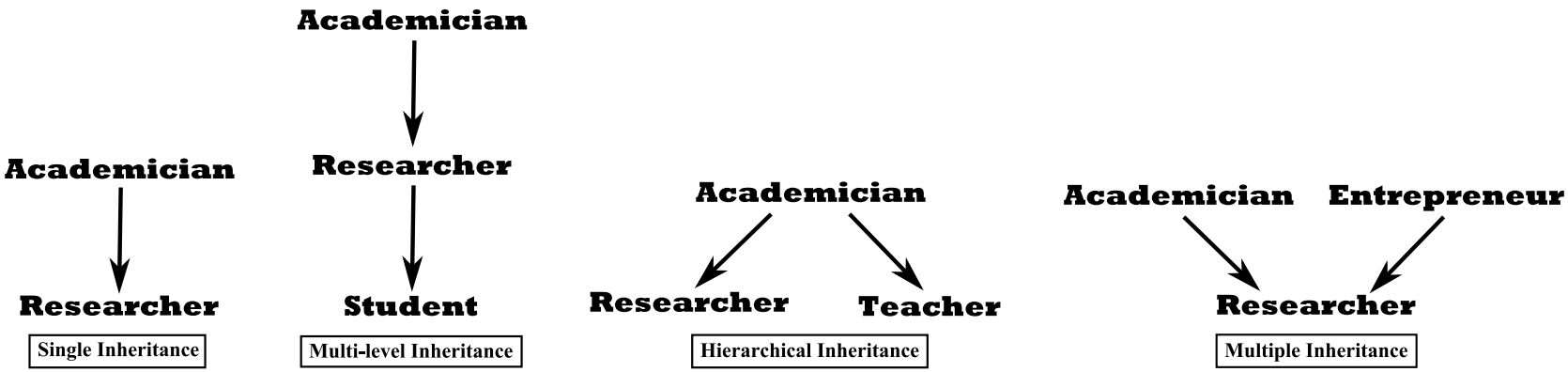
Type inheritance

Inheritance is acquiring the properties of another type



Type inheritance

Inheritance is acquiring the properties of another type



Note: subtype (derived type) inherits the properties of the supertype (base type).

Inheritance

Single Inheritance:

```
 create type Academician
     (Name varchar(20),
     Affiliation varchar(20)
     )
create type Researcher
     under Academician
          (Area varchar(20),
          Publication_count varchar(20)
          )
```

Inheritance

Multi-level Inheritance:

```
 create type Academician
     (Name varchar(20),
     Affiliation varchar(20)
     )
create type Researcher
     under Academician
          (Area varchar(20),
          Publication_count integer,
          Award_count integer
          )
create type Student
     under Researcher
          (Course varchar(20)
          )
```

Inheritance

Hierarchical Inheritance:

```
 create type Academician
     (Name varchar(20),
     Affiliation varchar(20)
     )
create type Researcher
     under Academician
          (Area varchar(20),
          Publication_count integer
          Award_count integer
          )
create type Teacher
     under Academician
          (Course varchar(20),
          Subject varchar(20)
          )
```

Inheritance

Multiple inheritance: Try at your own!!!

Basics

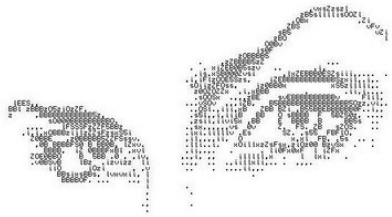
Multimedia comprises contents combining multiple forms

Basics

Multimedia comprises contents combining multiple forms

- Data depends on the capturing techniques.
- Queries might not return textual results.
- Size does matter.
- Automatic feature extraction and real-time processing.

Different multimedia types

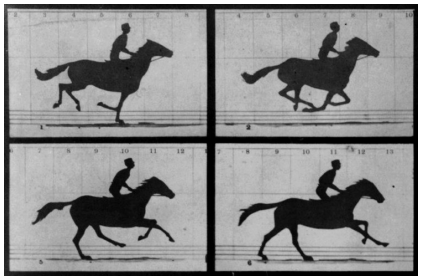


Text

Image

Audio

Animation



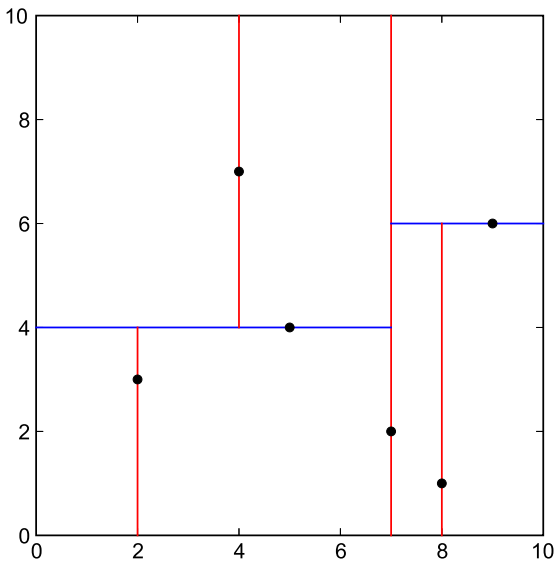
Video

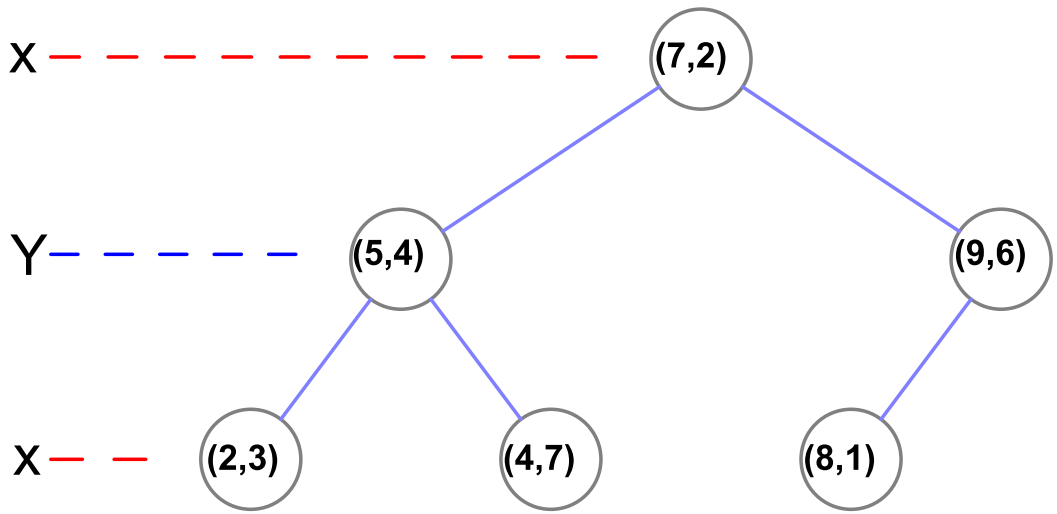
Interaction

Some data structures for multimedia

K-dimensional trees:

A k-dimensional tree (or k-d tree) is a space-partitioning data structure for organizing points in a k-dimensional space.





Some data structures for multimedia

Adding elements to a K-d tree:

1. Traverse the entire tree, starting from the root and moving to either the left or the right child depending on whether the element to be inserted is on the 'left'/'right' side of the splitting plane.
2. On getting a leaf node, add the new point as either the left or right child of the leaf node, again depending on which side of the node's splitting plane contains the new element.

Some data structures for multimedia

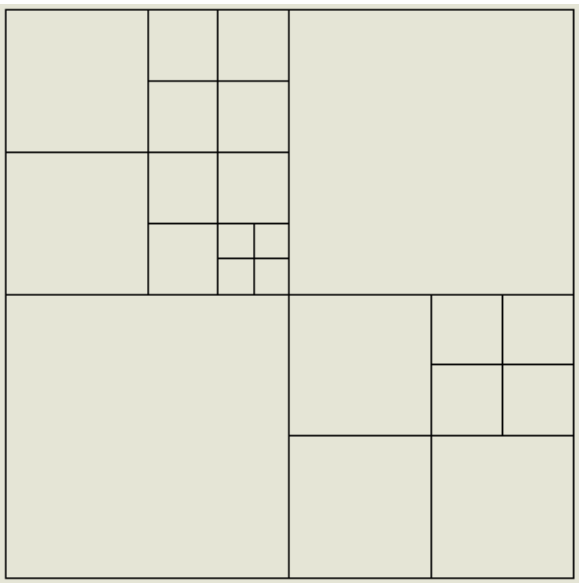
Deleting elements from a K-d tree:

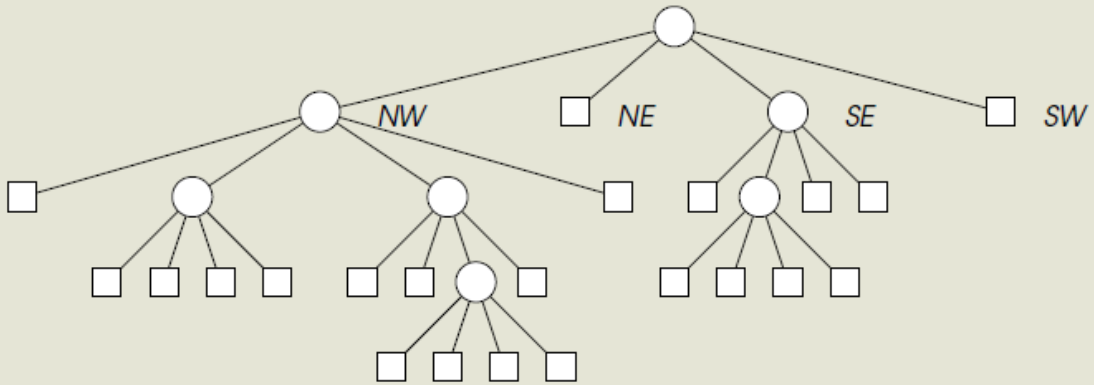
1. Find the node to be deleted using breadth-first search technique.
2. Now handle the following two cases:
   - No children – replace ptr to node by NULL.
   - Has children – replace node by minimum node in right subtree. If no right subtree exists, then first move left subtree to become right subtree.

Some data structures for multimedia

Quad-trees:

A quad-tree is a tree data structure in which each internal node has exactly four children. Quad-trees are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. The regions may be square or rectangular, or may have arbitrary shapes.

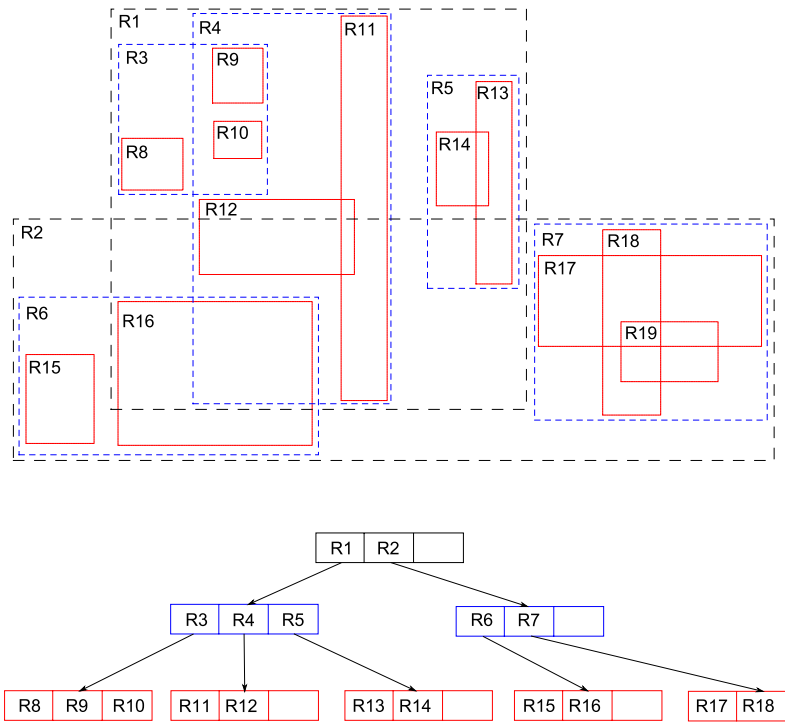




Some data structures for multimedia

R trees:

R-trees are a N-dimensional extension of B^+-trees, but are used for spatial access methods, i.e., for indexing multi-dimensional information; for example, the (X, Y) coordinates of geographical data.



The concept of metadata

Metadata refers to data about data

The concept of metadata

Metadata refers to data about data

- Content-dependent metadata: It depends on the content of the document. E.g., size of a document, maximum colors, number of rows and columns, etc. It can be further sub-divided as: direct content-based metadata (e.g., full-text indices, inverted tree and document vectors, etc.) and content descriptive metadata (e.g., textual annotations describing the contents of an image). Content descriptive metadata can be further subdivided into the types – domain-independent metadata and domain-specific metadata.
- Content-independent metadata: It does not depend on the content of the document. E.g., location, modification date and time, etc.

Multimedia database support by SQL

Large Object Domains: These are long unstructured sequences of data often of two kinds – Binary Large Objects (BLOBs, which are an unstructured sequence of bytes) and Character Large Objects (CLOBs, which are an unstructured sequence of characters).

Multimedia database support by SQL

Large Object Domains: These are long unstructured sequences of data often of two kinds – Binary Large Objects (BLOBs, which are an unstructured sequence of bytes) and Character Large Objects (CLOBs, which are an unstructured sequence of characters).

File reference: Instead of holding the data, a file reference contains a link to the data.

Multimedia database support by SQL

Large Object Domains: These are long unstructured sequences of data often of two kinds – Binary Large Objects (BLOBs, which are an unstructured sequence of bytes) and Character Large Objects (CLOBs, which are an unstructured sequence of characters).

File reference: Instead of holding the data, a file reference contains a link to the data.

```
CREATE TABLE Mango
    (Mango_Name VARCHAR2(30),
    Mango_Description CLOB DEFAULT EMPTY_CLOB(),
    Mango_Picture BLOB DEFAULT EMPTY_BLOB(),
    CONSTRAINT Prim_Mango PRIMARY KEY (Mango_Name)
)
```