Chapter 8

Multimedia IR:
Models and Languages

Introduction

- Complex structures instead of simple data types
  - Different media types (images, sounds, text, graphs)
  - Mix of structured and unstructured data
    - Metadata
    - Semi-structured data
      - Data whose structure may not match, or only partially match, the structure prescribed by the data schema
    - The system must typically extract some features from the multimedia objects
- Multimedia IR system should handle Metadata
- Architecture of system depends on
  - Characteristics of data
  - Operations performed on data

Introduction

- Data modeling
  - Store multimedia object
  - Different kinds of medi
  - Semistructured dat
- Data retrieval
  - Data attributes and content
  - Query specification

    Fuzzy predicates, content-based predicates, object attributes, structural predicates
  - Query processing and optimization

    Query is parsed and compiled into an internal form
  - Query iteration

    The query execution until the user is satisfied

Find all images similar to a car

Find multimedia objects containing an apple

Find all red images

Find multimedia objects containing a video clip

Goal

- Multimedia IR systems should combine DBMS and IR technology
  - Data modeling capabilities of DBMSs
  - Similarity-based query capabilities of IR systems
- System should support **attribute-based** as well as **content-based** queries

Data modeling

- **Main tasks**
  - **A data model should be defined by which the user can specify the data to be stored into the system**
    - Support conventional and multimedia data types
    - Provide methods to analyze, retrieve, and query such data
  - **Provide a model for the internal representation of multimedia data**

Data modeling

- **RDBMSs may help**
  - But integration of multimedia data in traditional DBMSs is difficult
    - Unstructured data which requires methods to identify and represent content features and semantic structures
    - Large storage requirements
- **OODBMSs may help**
  - But performance stays behind
    - Storage, query processing, transaction management
- **Object-relational technology may help**
  - Extending the relational model with ability to represent complex data types (SQL3)

Data modeling
Object-oriented DBMS

- **Provide rich data model**
  - More suitable for modeling both multimedia data types and their semantic relationships
- **Class**
  - Attributes +operations
  - Inheritance
- **Drawback**
  - the performances of storage techniques, query processing, and transaction management is not comparable to that of relational DBMSs
  - Highly non-standard

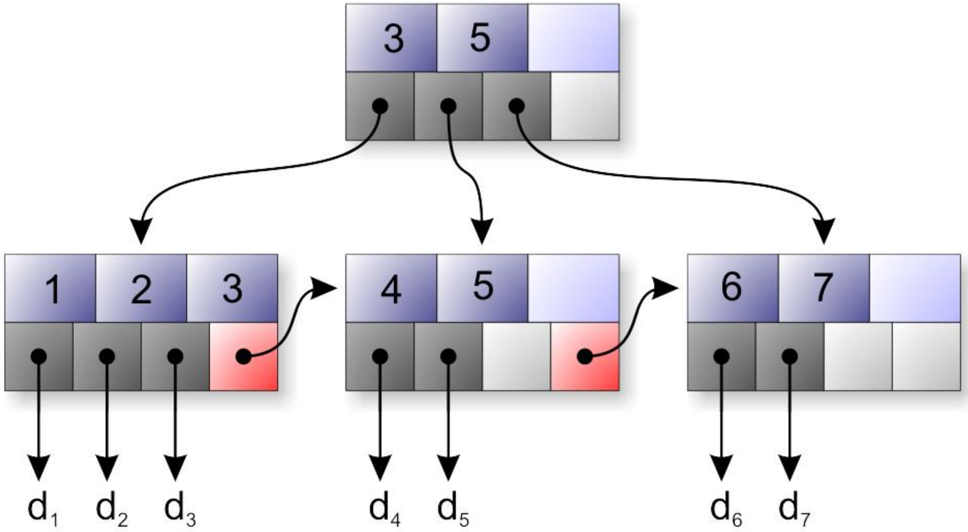
Data modeling
Object-relational DBMS

- **Extend the relational model**
  - Represent complex data types
  - Maintain the performance and the simplicity of relational DBMSs and related query languages
  - Define abstract data types
    - Allows one to define ad hoc data types for multimedia data

Support in commercial DBMSs

- Variable-length data types
- Details are platform specific
  - RAW, LONGRAW, LOB, BLOB, CLOB
  - IMAGE, TEXT
- No interpretation of data content
- SQL3 brings an extensible type system
  - OO like manner
- Proprietary implementations abound
  - Data cartridges in Oracle (ConText)
  - Data blades in Illustra (2D/3D spatial, text, images)

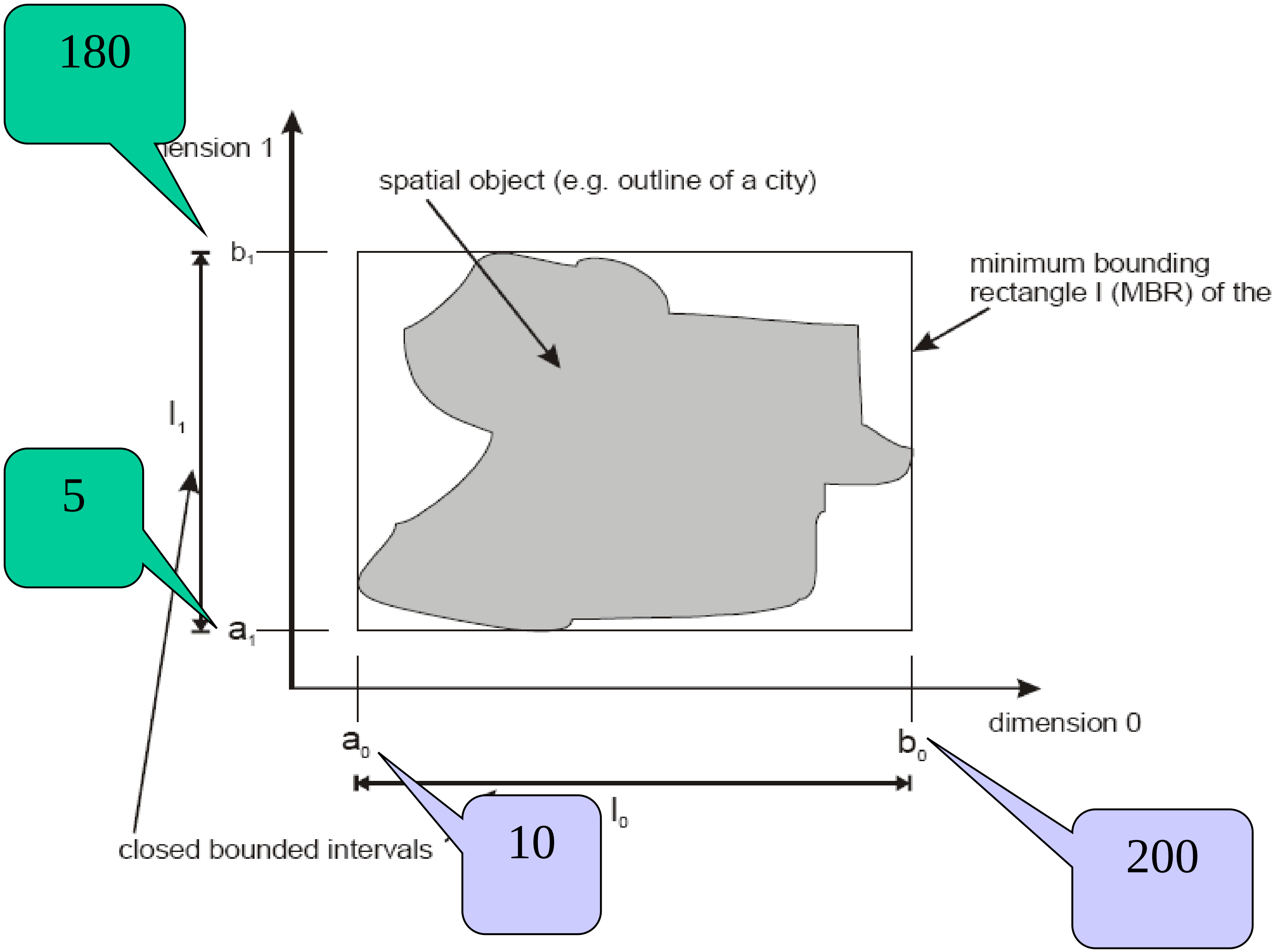
B-Tree
Traditional DBMS



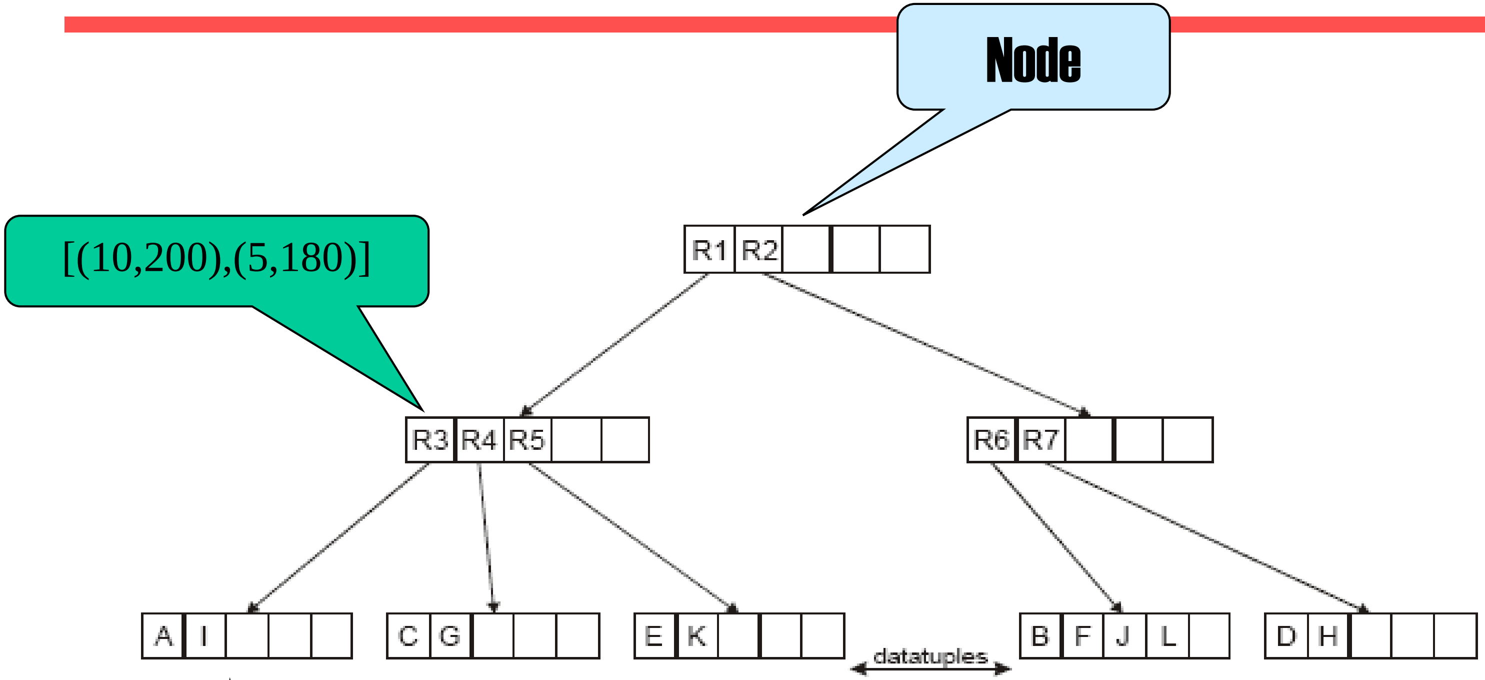
R-tree (Region Tree)

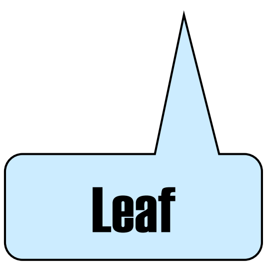
- R-tree
  - Represent a spatial object by its minimum bounding rectangle (**MBR**)
  - Data rectangles are grouped to form parent nodes (recursively grouped)
  - The MBR of a parent node completely contains the MBRs of its children
  - MBRs are allowed to overlap
  - Nodes of the tree correspond to disk pages

R-tree


180
5
10
200
dimension 1
spatial object (e.g. outline of a city)
minimum bounding
rectangle I (MBR) of the object
b_1
a_1
I_1
a_0
b_0
I_0
dimension 0
closed bounded intervals

R-Tree





m = minimum of entires in a node
M = maximum of entires in a node

m -Big good for small update

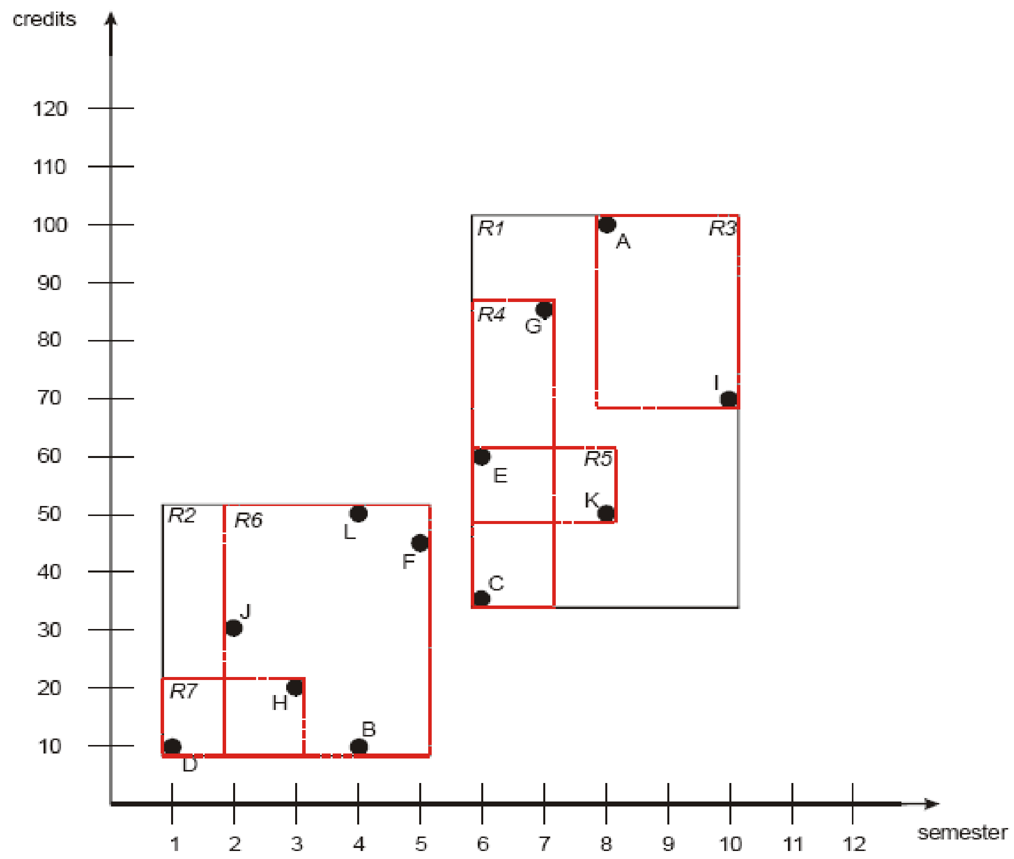
-Small good for high update

M depend on Harddisk

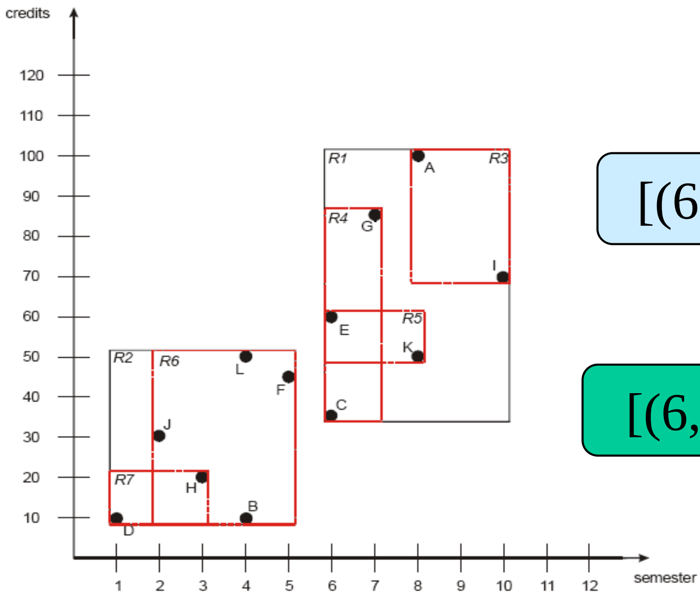
R-Tree Example

name	semester	credits
A	8	100
B	4	10
C	6	35
D	1	10
E	6	40
F	5	45
G	7	85
H	3	20
I	10	70
J	2	30
K	8	50
L	4	50

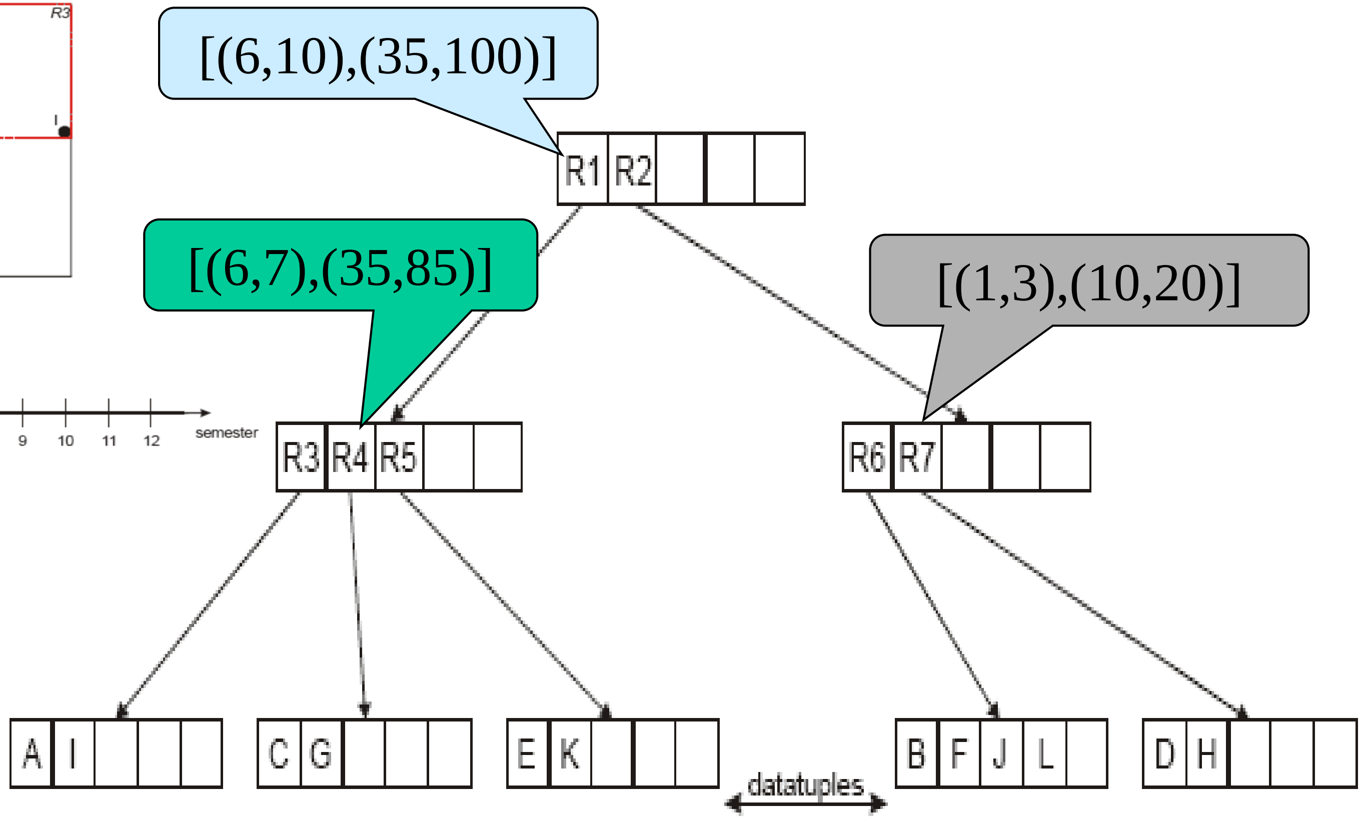
m = 2
M = 5



R-Tree Example

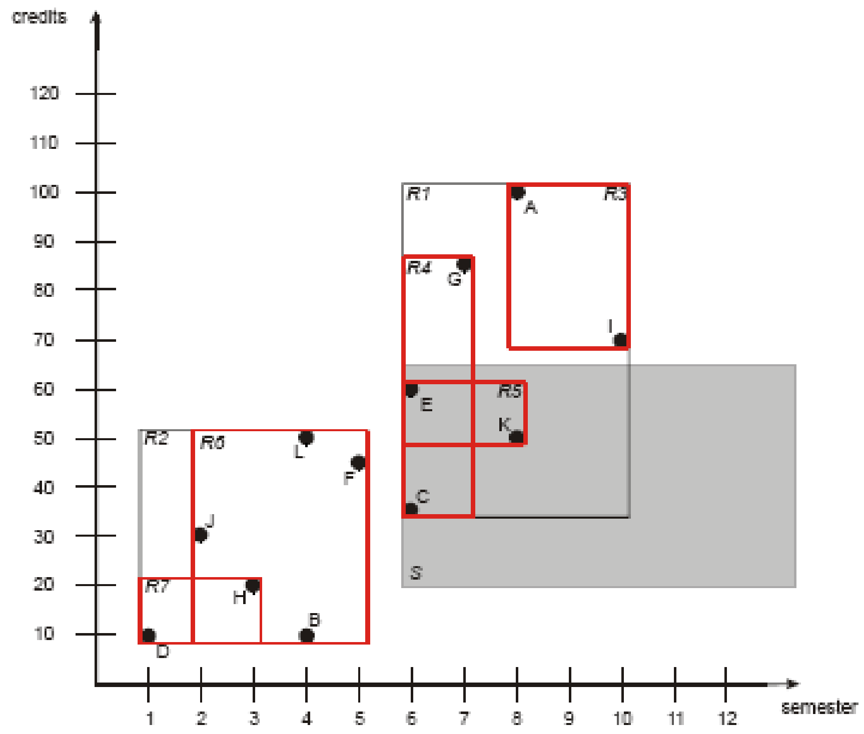


name	semester	credits
A	8	100
B	4	10
C	6	35
D	1	10
E	6	40
F	5	45
G	7	85
H	3	20
I	10	70
J	2	30
K	8	50
L	4	50



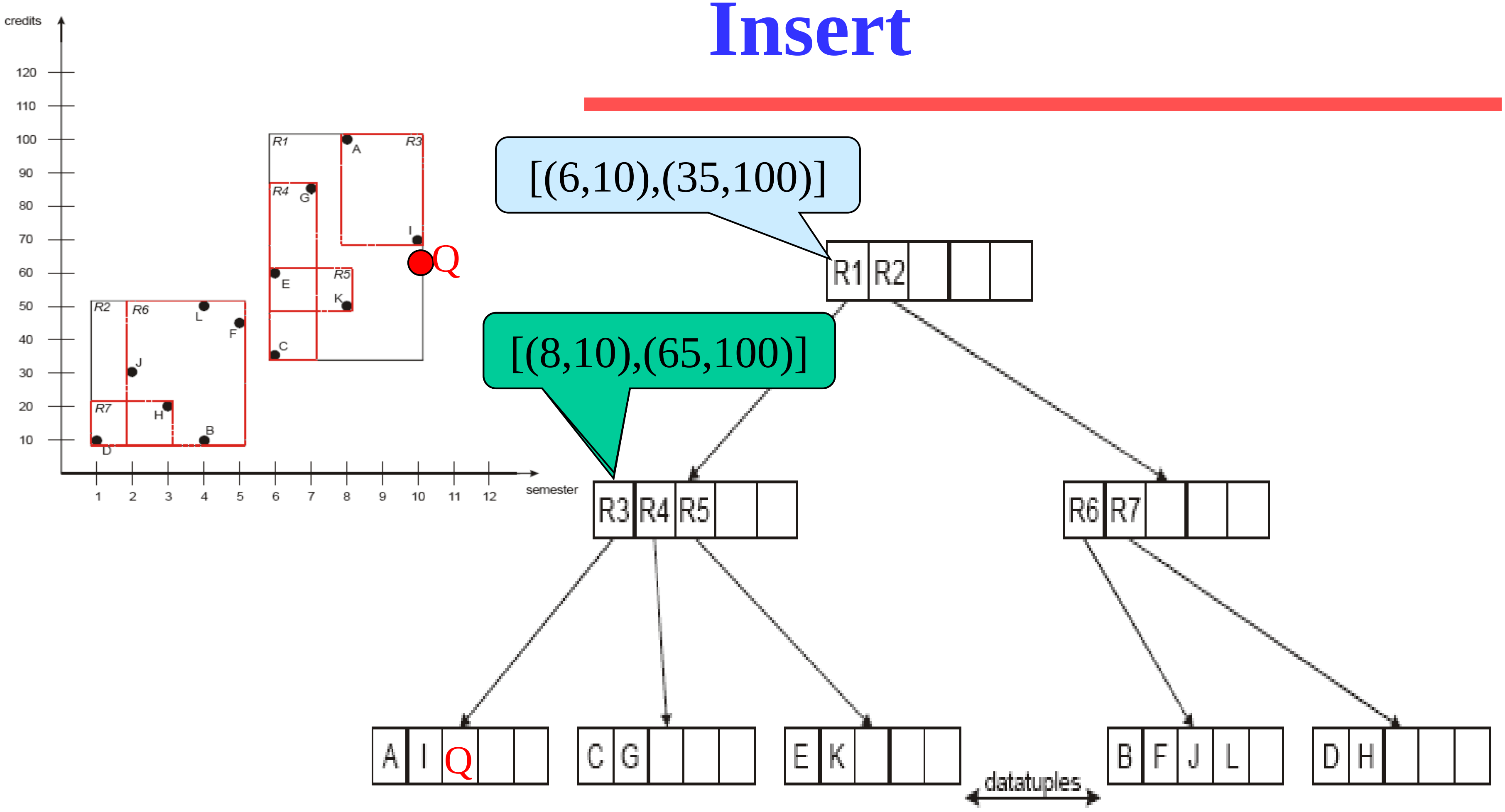
R-Tree Example Searching

name	semester	credits
A	8	100
B	4	10
C	6	35
D	1	10
E	6	40
F	5	45
G	7	85
H	3	20
I	10	70
J	2	30
K	8	50
L	4	50



Sixth semester or higher and earned between 20 and 65 credits

R-Tree Example
Insert



Insert (Q,10,65)

Query languages

- Exact match is only one of the possible ways to query multimedia objects
  - Due to the semistructured nature of the data
- Content-based querying considers both structure and content of the data
  - Matching of features and degree of similarity
- Request specification
  - Browsing and navigation
  - Query by example
  - Specific query language

SQL3

- **Support extensible type system**
  - Provide constructs to define user-dependent abstract data types, in an object-oriented like manner
- **Collection data types**
  - Sets, multisets, and lists
  - The elements of a collection must have compatible types

SQL3 / SQL99

- Perspective from query language point of view
  - Functions and stored procedures
  - Active database facilities
- Perspective from multimedia point of view
  - Suitable for being used as an interface language for multimedia applications
    - External functions, user-defined data types
    - Triggers can enforce spatial and temporal constraints
    - A "standard"
  - However, no integrated IR techniques
    - Content-based search is application dependent
    - SQL/MM Full Text, SFQL

SQL3 Example

```
CREATE TABLE branch(
    Bno      VARCHAR(3),
    address ROW(
        street          VARCHAR(25),
        town            VARCHAR(15),
        pcode           ROW(  city_id          VARCHAR(4)
                              subpart        VARCHAR(4))));

INSERT INTO branch
VALUES(‘B5’, (‘22 Deer Rd’, ‘Sidcup’, (‘SW1’, ‘4EH’)));
```

SQL3 Example

```
CREATE TYPE person_type AS (
    PRIVATE
        date_of_birth DATE CHECK(date_of_birth > DATE ‘1990-01-01’);
    PUBLIC
        fname  VARCHAR(15) NOT NULL,
        lname  VARCHAR(15) NOT NULL,
        FUNCTION get_age (P person_type) RETURNS INTEGER
                RETURN /* code to calc age */
        END;

    END);
```

SQL3 Example

```
SELECT s.lname, s.get_age
FROM staff s
WHERE s.is_manager;

SELECT p.lname, p.address
FROM person p
WHERE p.get_age > 65;

SELECT p.lname, p.address
FROM ONLY (person) p
WHERE p.get_age > 65;
```

OQL (Object Query Language)

```
class Student
(extent students)
{
    attribute short id;
    attribute string name;
    attribute string address;
    attribute date birthdate;
    relationship set<Module> takes
            inverse Module takenby;
    short age();
};
```

OQL (Object Query Language)

```
class Module
   (extent modules)
{
   attribute string title;
   attribute short semester;
   relationship set<Student> takenby
         inverse Student takes;
};

class Postgrad extends Student
   (extent postgrads)
{
   attribute string thesis_title;
};
```

OQL (Object Query Language)

- select distinct x.age
  from Persons x
  where x.name = "Pat";

- Return literal of type set<struct>

  select distinct struct(a:x.age, s:x.sex)
  from Persons x
  where x.name = "Pat";

OQL Examples

- **Path Expressions**
  select c.address
  from Persons p, p.children c
  where p.address.street = "Main Street"
  and count(p.children) >= 2
  and c.address.city != p.address.city;

- **Methods**
  select max(select c.age from p.children c)
  from Persons p
  where p.name = "Paul";

Query languages

- Conditions on multimedia data
  - Multimedia query languages should provide predicates for expressin
    - Attribute predicates
    - Structural predicates
    - Semantic predicates
- Structural and se antic predicates can refer to spatial or t oral properties
  - Contain, intersect, be fter

Metadata,Database schema

Content of query data and feature data

Find all the objects containing the world OFFICE

Query languages
Query Expression

- Uncertainty, proximity and weigths
  - Imprecise terms and predicates
    - Normal, unacceptable, typical
  - Proximity predicates
    - Based on semantic distance, such as “nearest object”
  - Weights
    - Specifying degree of precision by which a condition must be verified
  - Allow the user to drive similarity-based selection of relevant objects