Multimedia Databases (MM DBMS)

Multimedia Databases

- To provide such database functions as indexing and consistency, it is desirable to store multimedia data in a database
  - rather than storing them outside the database, in a file system
- The database must handle large object representation.
- Similarity-based retrieval must be provided by special index structures.
- Must provide guaranteed steady retrieval rates for continuous-media data.

Types of multimedia data

- *Text:* using a standard language (SGML, HTML)
- *Graphics:* encoded in CGM, postscript
- *Images:* bitmap, JPEG, MPEG
- *Video:* sequenced image data at specified rates
- *Audio:* aural recordings in a string of bits in digitized form

Nature of Multimedia Applications

- *Repositories:* central location for data maintained by DBMS, organized in storage levels
- *Presentations:* delivery of audio and video data, temporarily stored, ‘VCR-like functionality’
- *Collaborative:* complex design, analyzing data

Management Issues

- *Modeling:* complex objects, wide range of types
- *Design:* still in research
- *Storage:* representation, compression, buffering during I/O, mapping
- *Queries:* techniques need to be modified
- *Performance:* physical limitations, parallel processing

Multimedia Data Formats

- Store and transmit multimedia data in compressed form
  - JPEG and GIF the most widely used formats for image data.
  - MPEG standard for video data use commonalties among a sequence of frames to achieve a greater degree of compression.
- MPEG-1 quality comparable to VHS video tape.
  - stores a minute of 30-frame-per-second video and audio in approximately 12.5 MB
- MPEG-2 designed for digital broadcast systems and digital video disks; negligible loss of video quality.
  - Compresses 1 minute of audio-video to approximately 17 MB.
- Several alternatives of audio encoding
  - MPEG-1 Layer 3 (MP3), RealAudio, Windows Media format, etc.

Continuous-Media Data

- Most important types are video and audio data.
- Characterized by high data volumes and real-time information-delivery requirements.
  - Data must be delivered sufficiently fast that there are no gaps in the audio or video.
  - Data must be delivered at a rate that does not cause overflow of system buffers.
  - Synchronization among distinct data streams must be maintained
    - Video of a person speaking must show lips moving synchronously with the audio

Video Servers

- **Video-on-demand** systems deliver video from central video servers, across a network, to terminals
  - Must guarantee end-to-end delivery rates
- Current video-on-demand servers are based on file systems; existing database systems do not meet real-time response requirements.
- Multimedia data are stored on several disks (RAID configuration), or on tertiary storage for less frequently accessed data.
- Head-end terminals - used to view multimedia data
  - PCs or TVs attached to a small, inexpensive computer called a set-top box.

Design Criteria for Multimedia Database

- Layout: Optimize layout of data blocks on secondary storage. If we can store the data optimally on disk, the access times for it will be much faster.
- Buffer requirements: Optimize buffer sizes for multiple data streams. Buffer refers to the "read ahead" data to keep user from waiting.

Cont… Design Criteria for Multimedia Database

- Admissibility criteria: Determine when it is safe to accept an additional client. If this isn't checked, MM database is vulnerable to buffer overflow attacks or DOS (denial of service) attacks.
- Scheduling: Ordering and prioritizing client requests such that client constraints are met and buffer resources are minimized.

What does a Multimedia DBMS need?

- Provide traditional database management concepts
  - Persistence of data
  - Consistent view of data (transactions, concurrency management, data integrity)
  - Recovery & Versioning
  - Security of Data
  - Query and retrieval of data
    (relational queries, information retrieval)

What does a Multimedia DBMS need?

However, a MM DBMS must provide additional features:

- Storage of multidimensional data
- Temporal relationships (relating to or limited by time)
- Descriptive (content oriented) search
- Device and format independence
- View-specific and simultaneous data access
- Management of large amounts of data
- Real time data transfer

Storing Multimedia Data

How can we store Multimedia Data?

- **ORDBMS** include object features within relational databases
  - Such as the facility for user defined types
- Implemented within Oracle using:
  - LOB
  - CLOB
  - BLOB
  - BFILE

Similarity-Based Retrieval

Examples of similarity based retrieval

- Pictorial data: Two pictures or images that are slightly different as represented in the database may be considered the same by a user.
  - E.g., identify similar designs for registering a new trademark.
- Audio data: Speech-based user interfaces allow the user to give a command or identify a data item by speaking.
  - E.g., test user input against stored commands.
- Handwritten data: Identify a handwritten data item or command stored in the database

Content Based Retrieval

- Multimedia databases need to support queries based on media content as well as traditional queries based on precise field values.
- MMDBs will also support both exact match and probabilistic retrieval.

Example:

Face recognition software. Content and fuzzy search capabilities.

Content Based Retrieval

- With relational databases, the result of a query is one set.
- With MMDBs, content queries that include weights and uncertainties are used.
- The results are probabilistic.
- Often, the returned results will be a ranked result set, indicating results found closest to user’s query.

Content Based Retrieval

Methods of doing retrieval:

- Text annotations (entered by people) describe contents
- Query by example compares media elements to specified graphical shapes or image characteristics
- Browsing enables users to quickly scan the data

Example: Think of a query-by-humming system for a music database

Multimedia Query Processing

1) Query on content of media information. Example: Show the details of the movie where an actor says: "Do you feel lucky, punk?".

2) Query by example. Example: What is the movie which contains this song.

3) Time indexed queries. Example: Show me the movie 30 minutes after its start. Based on temporal characteristics.

Multimedia Query Processing

4) Spatial queries: Show me an image where President Yelstin is seen to the left of President Clinton. Based on metadata.
5) Application specific queries: Show me the video where the river changes its course.

Problems / issues in MMDBS

1) Extremely high capacity storage
2) Efficiency requires compression techniques integrated into the DBMS
3) Retrieval of multimedia objects in a distributed system
4) Video on demand:
   - Server must contend with specialized temporal requests.
   - Video delivery rates must be synchronized at server and client
   - characteristics of communication network must be considered.
   - This is necessary to allow user to pause, rewind, fast forward VOD.

Multimedia Database Applications

- *Documentation and keeping Records*
- *Knowledge distribution*
- *Education and Training*
- *Marketing, Advertisement, Entertainment, Travel*
- *Real-time Control, Monitoring*

Research Problems

- *Information Retrieval in Queries:* Modeling the content of documents
- *Multimedia/Hypermedia Data Modeling and Retrieval:* Hyperlinks, Used in WWW
- *Text Retrieval*

Spatial & Geographic Data

Spatial and Geographic Databases

- Spatial databases store information related to spatial locations, and support efficient storage, indexing and querying of spatial data.
- Special purpose index structures are important for accessing spatial data, and for processing spatial join queries.
- **Computer Aided Design (CAD)** databases store design information about how objects are constructed E.g.: designs of buildings, aircraft, layouts of integrated-circuits
- Geographic databases store geographic information (e.g., maps): often called **geographic information systems or GIS**.

Representation of Geometric Information

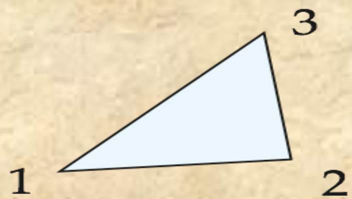
- Various geometric constructs can be represented in a database in a normalized fashion.
- Represent a line segment by the coordinates of its endpoints.
- Approximate a curve by partitioning it into a sequence of segments
  - Create a list of vertices in order, or
  - Represent each segment as a separate tuple that also carries with it the identifier of the curve (2D features such as roads).
- Closed polygons
  - List of vertices in order, starting vertex is the same as the ending vertex, or
  - Represent boundary edges as separate tuples, with each containing identifier of the polygon, or
  - Use **triangulation** — divide polygon into triangles
    - Note the polygon identifier with each of its triangles.

Representation of Geometric Constructs

line segment

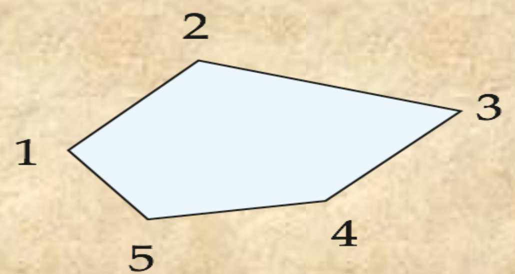
{(x1,y1), (x2,y2)}

triangle



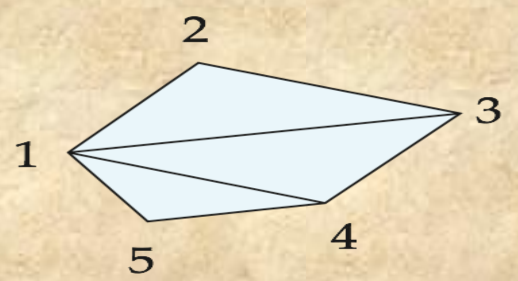
{(x1,y1), (x2,y2), (x3,y3)}

polygon



{(x1,y1), (x2,y2), (x3,y3), (x4,y4), (x5,y5)}

polygon



{(x1,y1), (x2,y2), (x3,y3), ID1}
{(x1,y1), (x3,y3), (x4,y4), ID1}
{(x1,y1), (x4,y4), (x5,y5), ID1}

object **representation**

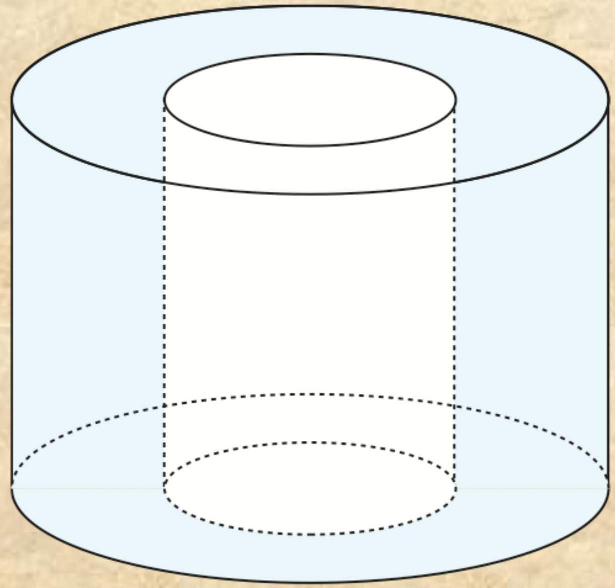
Representation of Geometric Information (Cont.)

- Representation of points and line segment in 3-D similar to 2-D, except that points have an extra z component
- Represent arbitrary polyhedra by dividing them into tetrahedrons, like triangulating polygons.
- Alternative: List their faces, each of which is a polygon, along with an indication of which side of the face is inside the polyhedron.

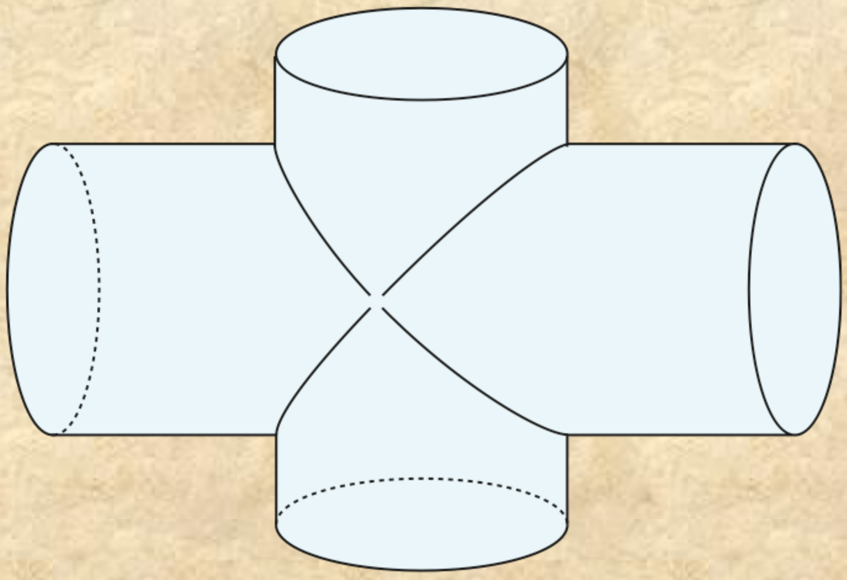
Design Database

- Represent design components as objects (generally geometric objects); the connections between the objects indicate how the design is structured.
- Simple two-dimensional objects: points, lines, triangles, rectangles, polygons.
- Complex two-dimensional objects: formed from simple objects via union, intersection, and difference operations.
- Complex three-dimensional objects: formed from simpler objects such as spheres, cylinders, and cuboids, by union, intersection, and difference operations.
- Wireframe models represent three-dimensional surfaces as a set of simpler objects.

Representation of Geometric Constructs



(a) Difference of cylinders



(b) Union of cylinders

- Design databases also store non-spatial information about objects (e.g., construction material, color, etc.)
- Spatial integrity constraints are important.
  - E.g., pipes should not intersect, wires should not be too close to each other, etc.

Geographic Data

Two types : Raster data & Vector data

- **Raster data** consist of bit maps or pixel maps, in two or more dimensions.
  - Example 2-D raster image: satellite image of cloud cover, where each pixel stores the cloud visibility in a particular area.
  - Additional dimensions might include the temperature at different altitudes at different regions, or measurements taken at different points in time.
- Design databases generally do not store raster data.

Geographic Data (Cont.)

- **Vector data** are constructed from basic geometric objects: points, line segments, triangles, and other polygons in two dimensions, and cylinders, spheres, cuboids, and other polyhedrons in three dimensions.
- Vector format often used to represent map data.
  - Roads can be considered as two-dimensional and represented by lines and curves.
  - Some features, such as rivers, may be represented either as complex curves or as complex polygons, depending on whether their width is relevant.
  - Features such as regions and lakes can be depicted as polygons.

Applications of Geographic Data

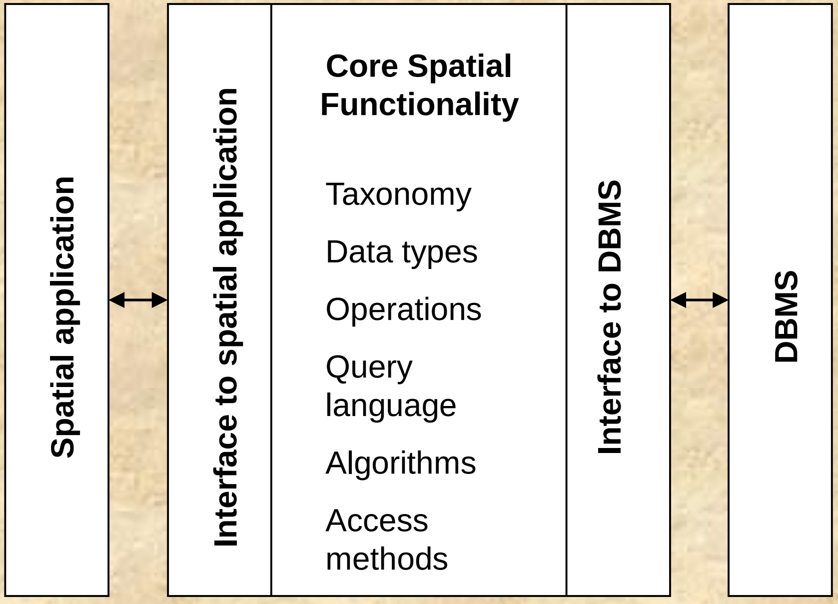
- Examples of geographic data
  - map data for vehicle navigation
  - distribution network information for power, telephones, water supply, and sewage
- Vehicle navigation systems store information about roads and services for the use of drivers:
  - **Spatial data**: e.g., road/restaurant/gas-station coordinates
  - **Non-spatial data**: e.g., one-way streets, speed limits, traffic congestion
- **Global Positioning System (GPS)** unit - utilizes information broadcast from GPS satellites to find the current location of user with an accuracy of tens of meters.
  - increasingly used in vehicle navigation systems as well as utility maintenance applications.

Spatial Database Management System

- Spatial Database Management System (SDBMS) provides the capabilities of a traditional database management system (DBMS) while allowing special storage and handling of spatial data.
- SDBMS:
  - Works with an underlying DBMS
  - Allows spatial data models and types
  - Supports querying language specific to spatial data types
  - Provides handling of spatial data and operations

SDBMS Three-layer Structure

- SDBMS works with a spatial application at the front end and a DBMS at the back end
- SDBMS has three layers:
  - Interface to spatial application
  - Core spatial functionality
  - Interface to DBMS



Spatial Queries

- **Nearness queries** request objects that lie near a specified location.
- **Nearest neighbor queries**, given a point or an object, find the nearest object that satisfies given conditions.
- **Region queries** deal with spatial regions. e.g., ask for objects that lie partially or fully inside a specified region.
- Queries that compute intersections or **unions** of regions.
- **Spatial join** of two spatial relations with the location playing the role of join attribute.

Spatial Queries (Cont.)

- Spatial data is typically queried using a graphical query language; results are also displayed in a graphical manner.
- Graphical interface constitutes the front-end
- Extensions of SQL with abstract data types, such as lines, polygons and bit maps, have been proposed to interface with back-end.
  – allows relational databases to store and retrieve spatial information
  – Queries can use spatial conditions (e.g., contains or overlaps).
  – queries can mix spatial and nonspatial conditions

Spatial Query Language

- Number of specialized adaptations of SQL
  - Spatial query language
  - Temporal query language (TSQL2)
  - Object query language (OQL)
  - Object oriented structured query language (O_2SQL)
- Spatial query language provides tools and structures specifically for working with spatial data
- SQL3 provides 2D geospatial types and functions

Spatial Query Language Operations

- Three types of queries:
  - Basic operations on all data types (e.g. IsEmpty, Envelope, Boundary)
  - Topological/set operators (e.g. Disjoint, Touch, Contains)
  - Spatial analysis (e.g. Distance, Intersection, SymmDiff)

Spatial Data Entity Creation

- Form an entity to hold county names, states, populations, and geographies

```
CREATE TABLE County(
    Name    varchar(30),
    State   varchar(30),
    Pop             Integer,
    Shape   Polygon);
```

- Form an entity to hold river names, sources, lengths, and geographies

```
CREATE TABLE River(
    Name    varchar(30),
    Source  varchar(30),
    Distance        Integer,
    Shape   LineString);
```

Example Spatial Query

- Find all the counties that border on Contra Costa county

```
SELECT  C1.Name
FROM    County C1, County C2
WHERE   Touch(C1.Shape, C2.Shape) = 1 AND C2.Name =
        'Contra Costa';
```

- Find all the counties through which the Merced river runs

```
SELECT  C.Name, R.Name
FROM    County C, River R
WHERE   Intersect(C.Shape, R.Shape) = 1 AND R.Name =
        'Merced';
```

```
CREATE TABLE County(            CREATE TABLE River(
   Name     varchar(30),           Name      varchar(30),
   State    varchar(30),           Source    varchar(30),
   Pop      Integer,               Distance  Integer,
   Shape    Polygon);              Shape     LineString);
```

Temporal Databases

Temporal DBs – Motivation

- Conventional databases represent the state of an enterprise at a single moment of time
- Many applications need information about the past
  - Financial (payroll)
  - Medical (patient history)
  - Government
- **Temporal DBs**

A system that manages ***time varying data***

Database that stores information about the state of real world across time.

Comparison

- Conventional DBs:
  - Evolve through transactions from one state to the next
  - Changes are viewed as modifications to the state
  - No information about the past
  - Snapshot of the enterprise
- Temporal DBs:
  - Maintain historical information
  - Changes are viewed as additions to the information stored in the database
  - Incorporate notion of time in the system
  - Efficient access to past states

Temporal Database Models

- Temporal Data Models: extension of relational model by adding temporal attributes to each relation
- Temporal Query Languages: TQUEL, SQL3
- Temporal Indexing Methods and Query Processing

Taxonomy of time

- Transaction time databases
  - Transaction time is the time when a fact is stored in the database
- Valid time databases:
  - Valid time is the time that a fact becomes effective in reality
- Bi-temporal databases:
  - Support both notions of time

Example

- Sales example: data about sales are stored at the end of the day
- Transaction time is different than valid time
- Valid time can refer to the future also!
  – Credit card: 03/01-04/06

Example : Queries

- Queries:
  - Timestamp (timeslice) queries: ex. “Give me all employees at 05/94”
  - Range-timeslice: “Find all employees with id between 100 and 200 that worked in the company on 05/94”
  - Interval (period) queries: “Find all employees with id in [100,200] from 05/94 to 06/96”

What is time varying data?

- You want a reprint of a customer's invoice of August 12, 1999.
- What was the stock value of the Oracle shares on June 15th, last year?
- What was the lowest stock quantity for every product last year? How much money will you save, if you keep the stocks at those levels?
- Where do you enter the new address of this customer as from the first of next month?
- What will your profits be next month, given the price list and cost prices by then?

What is time varying data?

Examples of application domains dealing with time varying data:

- Financial Apps (e.g. history of stock market data)
- Insurance Apps (e.g. when were the policies in effect)
- Reservation Systems (e.g. when is which room in a hotel booked)
- Medical Information Management Systems (e.g. patient records)
- Decision Support Systems (e.g. planning future contigencies)
- CRM applications (eg customer history / future)
- HR applications (e.g Date tracked positions in hierarchies)

What is time varying data?

In fact, time varying data has ***ALWAYS*** been in business requirements – but existing technology does ***not*** deal with it elegantly!

Temporal DB Design Approaches

Several implementation strategies are available

- Use a date type supplied in a non-temporal DBMS and build temporal support into applications (***traditional***)
- Implement an abstract data type for time (***object oriented***)
- Provide a program layer (***API***) above a non-temporal data model.

Certified for IBM DB2 e-business software

Implementation Approaches

- Generalise a non-temporal data model into a temporal data model (***Temporal Normal Form***)
- Re-design core database kernel (***pure Temporal Database***)

Certified for IBM DB2 e-business software