COMP3211 Advanced Databases

Data Types and Operations — Clean Notes

1 Big Picture

Advanced DBMSs must support **non-traditional data types** (temporal, spatial, multimedia) and provide suitable **operations** for querying them. A key theme: **not every operation makes sense for every type**, so systems must define meaningful operations (and avoid misleading ones).

2 Data Types in Databases

Common types covered:

- **Numeric** (integers, reals)
- **Character** (strings)
- **Temporal** (time-oriented data)
- **Spatial** (geometric data in 2D/3D)
- **Text** (documents, semi/unstructured text)
- **Image** (still images)
- **Audio & Video** (multimedia streams/files)

3 Operations on Data

Typical operation families:

- **Comparison:** equality, ordering, similarity
- **Arithmetic:** addition, multiplication, etc.
- **Fuzzy search / similarity:** approximate matching (especially for text/media)
- **Information retrieval queries:** e.g. documents containing a word; images containing a feature

Definition

Meaningful operation: An operation is meaningful when its result has a clear, consistent interpretation for that data type. If an operation is undefined or ambiguous, it should not be treated as a normal operator.

Operations: what's meaningful?

Units matter (weights vs numbers):

- $2\,\text{kg} + 2\,\text{kg}$ → meaningful (same unit, additive quantity)
- $2\,\text{kg} \times 2\,\text{kg}$ → generally **not meaningful** in normal DB semantics (unit becomes kg^2)
- $13 + 2\,\text{kg}$ → meaningless (incompatible kinds)
- $13 \times 2\,\text{kg}$ → meaningful (scale a quantity by a factor)

Media types:

- "Compare two images for equality" is usually not meaningful (tiny changes break equality).
- "Add two images" is ambiguous unless the system defines a specific image-processing meaning.

3.1 Ordering: Total vs Partial

A common question for any type is: **is it ordered**, and what kind of order?

Definition

Total order: any two values are comparable (for any a, b, either $a \leq b$ or $b \leq a$).
Partial order: some pairs may be incomparable (neither $a \leq b$ nor $b \leq a$ holds).

Note

Even if you *can* impose an order (e.g. by ID), the key question is whether the order has **semantic meaning** or is just a convenience.

4 Temporal Data

Temporal data adds the time dimension to support questions such as:

- Average price of product X during 1995
- Month with the most copies sold of video Y
- Treatment history of patient Z

4.1 Characteristics of Time

Time can differ by:

- **Structure:** linear; branching time (possible futures); directed acyclic graph; periodic/cyclic
- **Boundedness:** unbounded; bounded with an origin; bounded at both ends

4.2 Time Density (Discrete / Dense / Continuous)

Slides distinguish time models by how many time points exist between two points.

Model	Timeline resembles	Ordering property	Points between two points
Discrete	Integers ($\mathbb{Z}$)	Total order	Finite number of chronons
Dense	Rational numbers ($\mathbb{Q}$)	Partial order	Infinite number of chronons
Continuous	Real numbers ($\mathbb{R}$)	Total order	Infinite number of chronons

Definition

Chronon: the smallest representable time unit (a fixed period) used by a system (e.g. 1 second, 1 minute).

4.3 Granularity

Granularity = the resolution used when representing time.

Example / Intuition

Event A at 11:00 and Event B at 15:00 on the same day:

- If granularity = **1 day**, A and B occur in the same time unit ⇒ no precedence is visible.
- If granularity = **1 minute**, A precedes B clearly.

Note

The slides also highlight a distinction between:

- **Sequence:** order in which events are recorded/considered
- **Time:** actual temporal placement and distance

These can differ (e.g. logged later vs happened earlier).

4.4 Storing Time in a Database

A database fact/event can have multiple time notions:

- **Valid time:** when the fact is true in the real world
- **Transaction time:** when the fact is current/stored in the DB and retrievable
- **Bitemporal:** storing *both* valid and transaction time

Example / Intuition

A correction scenario (intuition):

- A fact could be valid from January, but only inserted into the DB in March.
- Valid time captures reality; transaction time captures DB history.

5 Temporal SQL Extensions (TSQL)

Extensions mentioned:

- **WHEN clause** (temporal conditions)
- Timestamp retrieval
- Retrieval of temporally ordered information
- `TIME-SLICE` clause to specify a time domain
- Modified aggregate functions via `GROUP BY`

5.1 WHEN Clause

Format:

```
SELECT {select-list} FROM {relations} WHERE {conditions} WHEN {temporal clause}
```

Temporal comparison operators include:

- BEFORE / AFTER
- PRECEDES / FOLLOWS
- DURING
- EQUIVALENT
- ADJACENT
- OVERLAPS

Note

These operators relate to **interval reasoning** (as in Allen's Interval Calculus): rather than comparing single timestamps, you compare *interval relationships* (overlap, adjacency, containment, etc.).

6 Spatial Data

Spatial data represents objects in space.

6.1 Spatial Data Types

Types listed:

- Points
- Regions
- Boxes
- Quadrangles
- Polynomial surfaces
- Vectors

6.2 Common Spatial Operations

Operations listed:

- Length / distance (where defined)
- Intersection
- Containment
- Overlap
- Centre computation

6.3 Applications & Properties of Interest

Main application areas:

- Computer Aided Design (CAD)
- Computer generated graphics
- Geographic Information Systems (GIS)

Properties of interest:

- **Connectivity** (what is linked/connected?)
- **Adjacency** (what touches what?)
- **Order** (arrangement/sequence in space)
- **Metric relations** (distances, angles, areas)

6.4 Why Spatial DB Performance is Hard (from slides)

- Objects can be highly complex
- Data volumes can be very large
- Real-time constraints may apply
- Performance is not easy to achieve
- Often accessed via specialised graphical front-ends (operator skill matters)
- Query processing may not use standard SQL

7 Multimedia Data

7.1 Text Data

Text may be:

- Already machine-readable (word processors, spreadsheets, etc.)
- Extracted via OCR

Key issue: text is **essentially unstructured** ⇒ retrieval needs an index:

- Human-built index, or
- Automatically built **inverted list** (index of significant words → documents containing them)

Definition

Inverted index / inverted list: maps each word (term) to the set of documents that contain it, enabling fast queries like "find all documents containing word w".

Markup languages add structure:

- HTML (web)
- XML / SGML (portable documents with structured data; can define new markup languages)

DB support mentioned:

- **CLOBs** (Character Large Objects) for storing text documents
- Text search and retrieval facilities

7.2 Document-Style Queries (Motivation)

Typical useful queries:

- Legal documents concerning client "Jones"
- Suspects with false teeth who have been interviewed
- Articles on "databases"

7.3 Image Data

Examples:

- X-rays, maps, photographs

Storage:

- Stored as **BLOBs** (Binary Large Objects)
- No attached semantics by default (the DB stores bits, not meaning)

Image databases need support for:

- Image analysis and pattern recognition
- Image structuring and understanding
- Spatial reasoning and image information retrieval

Definition

QBIC (Query By Image Content): retrieve images using content features (e.g. colour/-texture/shape) rather than only filenames/labels.

7.4 Audio Data

Digitised audio:

- Formats: WAV, MP3
- Consumes large storage; compression commonly used

MIDI:

- More compact than digitised audio
- Stored as instruction sequences (e.g. `Note_On`, `Note_Off`, `Increase_Volume`)
- Interpreted by a synthesiser

7.5 Video Data

Video characteristics:

- Extremely space-hungry
- Stored as a sequence of frames (each frame can be $>$ 1MB)
- Playback typically 24–30 fps

Audio-video integration:

- Interleaved file structures coordinate time sequencing
- Examples: Microsoft AVI, Apple QuickTime

8 Quick Consolidation (What to Remember)

- Different data types ⇒ different meaningful operations.
- Ordering matters: total vs partial order; semantic vs convenience order.
- Temporal: structure, boundedness, density (discrete/dense/continuous), granularity, valid vs transaction time.
- Spatial: specialised types + geometry operations; large/complex data makes performance hard.
- Multimedia:
  - Text needs indexing (inverted index); markup adds structure (HTML/XML).
  - Images are BLOBs with no semantics unless analysed; QBIC = content-based retrieval.
  - Audio/video are storage-heavy; compression and timing/sync are key.