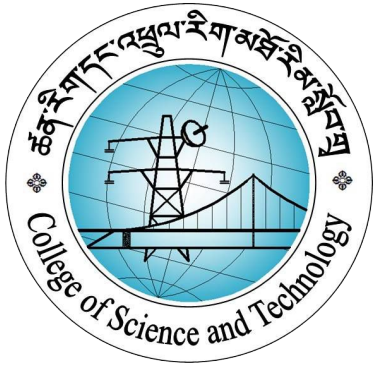

Royal University of Bhutan

DBS101 Database Systems Fundamentals

Lesson 20

Learning Outcomes

1. Explain the concept of column-oriented storage and its advantages.
2. Understand the storage organization in main-memory databases.

Column Oriented Storage

What is Column-Oriented Storage?

- Stores each attribute of a relation separately in contrast to row-oriented storage where all attributes stored together

Example: instructor relation stored column-wise

Column Oriented Storage

10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 13.14 Columnar representation of the *instructor* relation.

Column Oriented Storage

Benefits of Column-Oriented Storage

- **Reduced I/O when only some attributes accessed**
  Example: Query on just instructor names
- **Improved CPU cache performance**
  Example: Fetch cache lines for one column only
- **Improved compression effectiveness**
  Example: Compress numeric salary values together
- **Enables vector processing operations**
  Example: Compare salary against constant in parallel

Column Oriented Storage

Benefits of Column-Oriented Storage

Good for Data Analysis Queries

- Access many rows but few attributes Example: Computing averages on a few columns
- Not suited for transaction processing queries Example: Fetching entire instructor records

Column Oriented Storage

Column Storage Formats

- Simple: Each column in a separate file Example: instructor_id, instructor_name files
- ORC: Columnar representation in stripes with indices Example: 250MB stripes, 10K value indices
- Column groups: Store related columns together Example: (id, name) and (dept, salary) groups

Main Memory Databases

- Entire database resides in main memory
- Designed to optimize in-memory data access
- No disk I/O for reading data (only for persistence)

Main Memory Databases

Benefits of Main-Memory Storage

- **No buffer manager or block operations needed**
  Example: Direct pointer from (id, name) to (salary)
- **Direct pointers to records possible**
  Example: Pointer to specific instructor record
- **Columnar data in consecutive memory locations**
  Example: Entire id column is one memory segment

Main Memory Databases

Main-Memory Storage Optimizations

- **No slotted page structure, direct record allocation**
  Example: Simple pointer-based record data structure
- **Logical column arrays split into physical arrays**
  Example: Multiple id column segments with indirection table
- **Other optimizations possible without disk access overhead**

Main Memory Databases

Memory Fragmentation Issues

- **Direct record allocation can lead to fragmentation**
  Example: Inserting variable length names
- **Need compaction or special memory managers**
  Example: Periodic compaction to defragment
- **Appending to column arrays requires data movement**
  Example: Relocate existing id values when inserting new ids

Next class:
Unit VI : Non Relational Databases
6.1 Data Management with Distributed Databases
6.2 BASE : Basically Available, Soft State, Eventually Consistent
6.3 Types of Eventual Consistency
6.4 Types of non relational Databases
6.5 Key-Value Pair Databases
6.6 Document Databases
6.7 Column Family Databases