Lecture no 3: Storage Structure

- After extraction of operational data, information is stored in databases.
- RDB & MDB are complementary and do not have to exclude each other.
  - In staging area (temporarily stored data) → RDBMS can be used, but it should not be available for user queries for performance reasons.
  - In presentation area → normalized databases (like RDBMS) are excluded from presentation area. This area used multidimensional database.
- Multidimensional DB →
  - MDB are optimized for OLAP & OLAP application
  - created by using input from staging area.
  - designed for efficient & convenient storage & retrieval of large volumes of data.
  - stored & viewed from different prespectives called dimensions.
  - Complexity grows quickly with no of dimensions & position. e.g., MDBMS (3 dimension with 10 values) RDBMS ($10^3 = 1000$)
- Performance advantages →
  - faster than RDB
  - Performance benefits are more for queries that generates cross-tab views of data

Lecture 4

RDB

- Obtaining same views as of MDB, RDB requires complex query.
- Complex queries like top-K can be tricky but are possible with SQL extentions.
- More maintenance due to need for indexes. Regular tuning is required.
- ad-hoc queries can be simplified or solved b/c data is stored in normalized tables.

MDB

- Ease of data presentation, maintenance & performance
- Data views are natural output of MDB
- Easy to maintain b/c data is organized
- Performance of MDB can be matched with RDB through database tuning.
- May not support all possible ad-hoc queries b/c designed for certain types of queries.

Feature	MDB	RDB
Data Structure	Multidimensional Cube	Normalized tables (rows and cols)
Optimization focus	OLAP, analytical queries	OLTP.
Performance	Fast for structured queries	Requires tuning for complex queries.
Ad-Hoc query	Limited flexibility	High flexibility.
Maintenance	Minimal	High
Transactional data handling	not well suited for transactional data	Well suited for transaction.

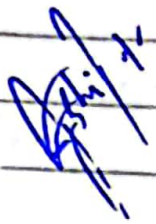
- MDB's are inappropriate → • dataset types are not related to each other results in sparse representation (empty space)
- MDB's are appropriate → when there are highly interrelated datatypes
  Example → • Budgeting
  • Product Profit

Tier Architecture (Structure framework for organizing software components in system into different layers or tiers)

DW architecture → generic two tier architecture
Three-tier architecture.
Web-based architecture.

Layered Architecture

- Data analysis comes in two flavours.



→ Thin client (HTTP / IIOP) → • Analytics executed on server
• client just display
• fits well for Internet DW access

→ Fat client (ODBC/NFS/JBDC) → • Server just deliver data
• Analytics executed on client

① - Generic Two-tier architecture → • data is not completely current in DW
• periodic extraction.

- Independent data mart → • mini warehouses
• Separate ETL for each independent data mart
• High data mart accen complexity.

- Dependent data mart (Operational data store) → • Single ETL for DW
• Data marts loaded from DW
• Simple data access as compared to Independent.

- Logical data mart (Active warehouse) → • ETL is near real-time
• Data marts are not separate databases but logical views of DW.

② - Generic Three-tier Architecture →

• Derived data → selected, formatted or aggregated for DSS support.

• Reconciled data → Detailed, current data intended to be single.