GLA UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with A Grade by NAAC

12-B Status from UGC

Database Management System (BCSC-1003)

Topic: Introduction to DBMS

Er. Priya Pandey

Assistant Professor, Dept. of CEA, GLA University,

Syllabus

GLA UNIVERSITY MATHURA

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	Introduction: An Overview of Database Management System, Database System Vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence, Database Language and Interfaces (DDL, DML, DCL), Database Development Life Cycle (DDLC) with Case Studies. Data Modeling Using the Entity-Relationship Model: ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Specialization, Generalization, Aggregation, Reduction of an ER Diagram to Tables, Extended ER Model. Relational Data Model and Language: Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra Database Design & Normalization I: Functional Dependencies, Primary Key, Foreign Key, Candidate Key, Super Key, Normal Forms, First, Second, Third Normal Forms, BCNF, Non-Redundant Cover, Canonical Cover	20

Syllabus

II	**Database Design & Normalization II:** 4th Normal Form, 5th Normal Form, Lossless Join Decompositions, MVD and JDs, Inclusion Dependence. **File Organization:** Indexing, Structure of Index files and Types, Dense and Sparse Indexing **Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Deadlock Handling. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, 2PL, Time Stamping Protocols for Concurrency Control, Validation Based Protocol. **Distributed Database:** Introduction of Distributed Database, Data Fragmentation and Replication.	20

Suggested Books

Text Books:

- Elmasri and Navathe, “Fundamentals of Database Systems”, 6th Edition, Addision Wesley, 2010.
- Sadalage, P. & Flower, “NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Presistence”, Pearson Education, 2012.

Reference Books:

- Date C J,” An Introduction to Database Systems”, 8th Edition, Addision Wesley.
- Korth, Silbertz and Sudarshan, “Database Concepts”, 5th Edition, TMH, 1998.
- Redmond, E. & Wilson, “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, 1st Edition.

Database

- A database is a collection of related data.
- Related data can be represented with the help of related tables as shown in figure below.
  - Consider the banking example with the following three tables: One show details of bank customers, the second shows accounts, and the third shows which accounts belong to which customers.

customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

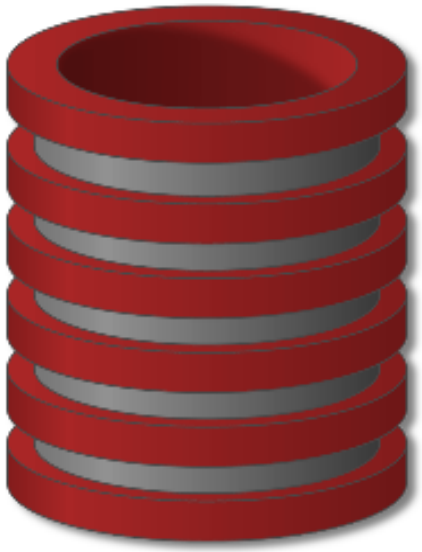
account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Whether you know it or not,
you're using a database every day



Database Management System (DBMS)

- A Database Management System (DBMS) is a software designed to store, retrieve, define, and manage data in a database.

- DBMS software primarily functions as an interface between the end user and the database, simultaneously managing the data, and the database schema in order to facilitate the organization and manipulation of data.

- A database management system functions through the use of system commands, first receiving instructions, then instructing the system accordingly, either to retrieve data, modify data, or load existing data from the system.

Applications of DBMS

- Railway Reservation System
- Library Management System
- Banking
- Universities and colleges
- Credit card transactions
- Social Media Sites
- Telecommunications
  - Finance
  - Military
  - Online Shopping

DBMS Functions

- *Defining*: a database involves specifying the data types, structures, and constraints for the data to be stored in the database.
- *Constructing*: the database the process of storing data(inserting data) itself on some storage medium that is controlled by the DBMS.
- *Manipulating*: a database includes querying (retrieving), updating (update/delete records) the database.
- *Sharing*: a database allows multiple users and programs to access the database concurrently.

Difference Between Data & Information in DBMS

- Data is raw, unprocessed, unorganized facts that are seemingly random and do not yet carry any significance or meaning.

- Information refers to data that has been organized, interpreted, and contextualized by a human or machine so that it possess relevance and purpose.

- Information is filtered data that has been made systematic and useful, and is considered to be more reliable and valuable to researchers as proper analysis and refinement has been conducted.

File System Vs DBMS

File System:

- The file system is basically a way of arranging the files in a storage medium like a hard disk.
- The file system organizes the files and helps in the retrieval of files when they are required.
- File systems consist of different files which are grouped into directories.
- The directories further contain other folders and files.
- The file system performs basic operations like management, file naming, giving access rules, etc.

File System Vs DBMS

DBMS:

- Database Management System is basically software that manages the collection of related data.
- It is used for storing data and retrieving the data effectively when it is needed.
- It also provides proper security measures for protecting the data from unauthorized access.
- In Database Management System the data can be fetched by queries and relational algebra.
- It also provides mechanisms for data recovery and data backup.

Database system vs. File System

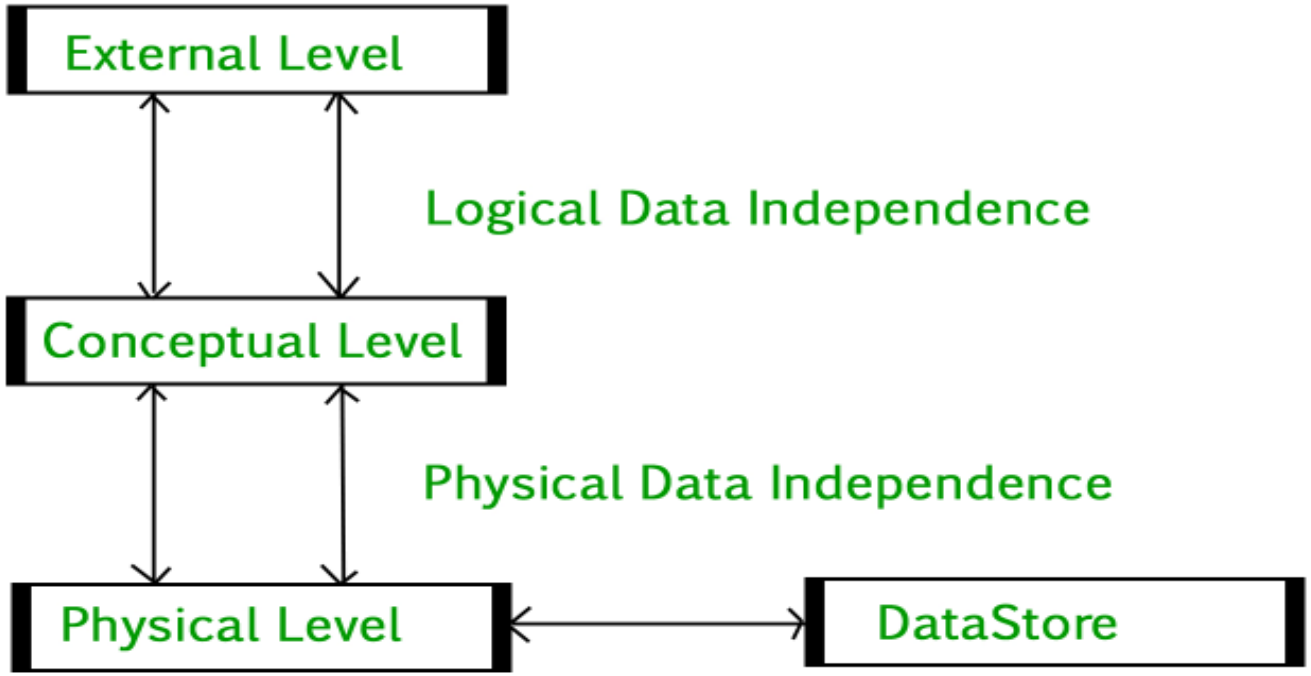
Key Point	Description	File System (*like data in text file*)	Database System
Data redundancy (*lead to higher storage and access cost***)**	**Same information is duplicated in many places**	This is found in File Systems. eg. Mobile and email id of student in cse_student_detail records and in Btech_student_detail records as well.	Data redundancy can be deal by **creating a separate table** having student#, mob & email.
Data Inconsistency	Two copies of duplicated data do not agree with each other.	This is **often a problem in file systems** as greater and unmanaged data redundancy is present.	Minimal data redundancy and that is also linked in the tables therefore **no inconsistency**.
Difficulty in Accessing data	Need to write a new program/ algorithm to carry out each new task.	**file system can't meet with changing needs:** we need to create program eg. • student with GPA>3.5 • student with GPA>3.5 and are from Delhi. Etc.	Searching required data is easy by **writing small queries** system get adapted to changing needs.

Key Point	Description	File System	Database System
Data Isolation	Format of files may or may not be same.	Data are scattered in **different files some in .doc or .txt or .xls** We require coding for each of them.	**Uniformity** in the way data is stored
Integrity problems	Data values must follow some **consistency constraint** such as • no account should have less than $25. • *account balance > 0*	we need to code it and if in future we want to change it we need to recode it! **Hard to add new constraints or change existing ones.**	**New constraint can easily be add, modify & drop.**
Atomicity Problem	Computer systems are **prone to failures**. Suppose a program transferring $50 from account A to B but in middle system crash then $50 was removed from but not credited to B. This lead to *inconsistent state*.	It **is difficult to ensure atomicity** in File system. *Transfer of funds from one account to another should either complete or not happen at all*	**Atomicity can easily be maintained**, these system have **recovery and back up** tools.

Key Point	Description	File System	Database System
Concurrent-access anomalies	Consider an account A holding $500, if two customer C1 & C2 withdraw $100 and $50 from A simultaneously then initially C1 & C2 see $500 now whichever write last it either show $400 or $450 while correct is $350.	File system **have greater challenge** as many application programs access same data simultaneously in multi-user system.	**Support multi-user system.**
Security problem	As faculty can upload and see attendance of student, whereas only *class advisor* can modify already uploaded attendance and *students* can only see their own attendance.	Enforcing security constraint is **difficult**.	Database **has internal procedures and commands** for this.

Introduction of 3-Tier Architecture in DBMS

- DBMS 3-tier architecture divides the complete system into three inter-related but independent modules as shown below:



Introduction of 3-Tier Architecture in DBMS

Physical Level:

- At the physical level, the information about the location of database objects in the data store is kept.
- Various users of DBMS are unaware of the locations of these objects.
- In simple term, physical level of a database describes how the data is being stored in secondary storage devices like disks and tapes.

Introduction of 3-Tier Architecture in DBMS

Conceptual Level:

- At conceptual level, data is represented in the form of various database tables.
- For Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware of their storage.
- Also referred as logical schema, it describes what kind of data is to be stored in the database.

Introduction of 3-Tier Architecture in DBMS

GLA UNIVERSITY MATHURA

External Level:

- An external level specifies a view of the data in terms of conceptual level tables.
- Each external level view is used to cater to the needs of a particular category of users.
- For Example, FACULTY of a university is interested in looking course details of students, STUDENTS are interested in looking at all details related to academics, accounts, courses and hostel details as well.
- So, different views can be generated for different users.
- The main focus of external level is data abstraction.

Data Independence

- ***Physical Data Independence*** – the ability to modify the physical schema without changing the logical schema
  - Applications depend on the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.
- ***Logical Data Independence-*** the ability to modify the logical schema without having to change external schema or application programs.

Data Independence

- Data independence can be explained using the three-schema architecture.

- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

- There are two types of data independence:
  (1) Logical Data Independence and
  (2) Physical Data Independence

Data Independence

Logical Data Independence:

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

Data Independence

Physical Data Independence:

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.


Thank
you