

Unit 1: Introduction to DBMS

1. DBMS की परिभाषा (Definition)

👉 DBMS (Database Management System) एक ऐसा software है जो data को **store, organize aur manage** करता है।

Ye ek **interface** है जो user aur database के बीच।

Definition:

“DBMS एक ऐसा system है जो data को structured form में store करता है और user को data insert, update, delete और retrieve करने की सुविधा देता है।”

2. DBMS की आवश्यकता (Need of DBMS)

1. **Data redundancy control** – बार-बार same data store होने से बचाता है।
 2. **Consistency maintain karta hai** – data हमेशा सही और updated रहता है।
 3. **Data security** – unauthorized access से data को बचाता है।
 4. **Data sharing** – एक ही data को multiple users access कर सकते हैं।
 5. **Backup & Recovery** – system failure होने पर भी data recover हो सकता है।
-

3. DBMS के मुख्य Components

1. **Hardware** – computer system, storage devices.
 2. **Software** – DBMS package (MySQL, Oracle, SQL Server).
 3. **Data** – वह actual जानकारी जो database में store होती है।
 4. **Users** –
 - DBA (Database Administrator)
 - Application Programmer
 - End User
 5. **Procedures** – वे rules और methods जो database operate करते हैं।
-

4. DBMS के Advantages (लाभ)

1. Data redundancy कम होती है।
 2. Data integrity और accuracy बनी रहती है।
 3. Multi-user access possible है।
 4. Data security high होती है।
 5. Backup और recovery आसानी से होता है।
 6. Large data को efficiently manage करता है।
-

5. DBMS के Disadvantages (हानि)

1. Cost ज्यादा आती है (software + hardware).
 2. Training की आवश्यकता होती है।
 3. System complexity ज्यादा है।
 4. Storage requirement high होती है।
 5. Agar DBMS fail ho jaye to pura system रुक जाता है।
-

6. DBMS के Applications (उपयोग)

- **Banking system** – account aur transaction management.
 - **Railway/Airline reservation** – ticket booking system.
 - **Library management** – books record.
 - **Education** – student records.
 - **Telecom** – call records, billing.
 - **E-commerce** – product aur customer data.
-

Exam Writing Tip:

Answer likhte waqt हमेशा “Definition → Need → Components → Advantages → Applications” ka sequence follow करो।

Diagram bhi add kar सकते हो:

Users → DBMS Software → Database

1. Database System Concepts (सिद्धांत)

Database system ek ऐसा system hai jisme **data ko structured form me store** किया जाता है aur **DBMS software** ke through us data ko efficiently manage kiya जाता है।

Main Concepts:

1. **Data** – actual information (जैसे student record).
2. **Database** – data का collection.
3. **DBMS** – software jo database ko manage करता है।
4. **User** – वो लोग jo DBMS ke through database access karte हैं।
5. **Query Language** – जैसे SQL, jo data retrieve/update ke liye use होती है।

Example:

- **Student Database:**
 - Data: नाम, Roll No, Course, Marks
 - Database: सभी students का collection
 - DBMS: MySQL/Oracle
 - User: Teacher, Student, Admin
-

2. Database Architecture (आर्किटेक्चर)

Database architecture बताता है कि **data** को कैसे **organize** और **manage** किया गया है।

(A) 1-Tier Architecture

- User directly database ke साथ interact करता है।
- Simple applications में use होता है।

📌 Example: MS-Access

(B) 2-Tier Architecture

- इसमें 2 layers होती हैं:
 1. **Client (User Interface)**
 2. **Server (Database)**
- User application के ज़रिए database से जुड़ता है।

📌 Example: Student marks enter karne के लिए एक desktop app जो directly MySQL database से जुड़ी हो।

(C) 3-Tier Architecture

- इसमें 3 layers होती हैं:
 1. **Presentation Layer (User Interface)**
 2. **Application Layer (Business Logic)**
 3. **Database Layer**
- सबसे ज्यादा secure aur scalable architecture।

📌 Example: Online Banking System

- User → Website/App (UI) → Application Server (Transaction Logic) → Database Server (Accounts & Balance data).
-

3. DBMS की Views (ANSI/SPARC 3-Level Architecture)

1. **External View (User View)**
 - हर user अपना अलग view देखता है।
 - Example: Student सिर्फ अपने marks देखेगा, teacher पूरे class ke marks देखेगा।

2. Conceptual View (Logical View)

- Pura database का logical structure.
- Example: Student, Teacher aur Courses ka relationship.

3. Internal View (Physical Storage)

- Data ka actual storage disk pe कैसे है।
- Example: Indexing, data blocks.

4. Advantages of DBMS Architecture

- Data security aur abstraction.
- Easy data sharing.
- Large systems के लिए scalable.
- Consistency aur integrity maintain होती है।

Exam Tip:

Diagram जरूर बनाना:

External View (Users)

↓

Conceptual View (Logical DB)

↓

Internal View (Physical DB)

-
- Example के साथ architecture explain करो तो marks ज्यादा मिलेंगे।

Components of DBMS (DBMS के मुख्य घटक)

DBMS कई parts (components) से मिलकर बना होता है, जो मिलकर database को **store, manage** और **access** करने में help करते हैं।

1. Hardware (हार्डवेयर)

- इसमें वो सारे physical devices आते हैं, जिन पर database चलता है।
- Example: Servers, Storage devices (HDD, SSD), Input/Output devices.

 **Point:** Hardware provide करता है **platform** जहां database system चलता है।

2. Software (सॉफ्टवेयर)

- Actual **DBMS software** जो database को manage करता है।
- इसमें query language, data access tools और utilities होती हैं।
- Example: Oracle, MySQL, MS SQL Server, PostgreSQL.

👉 **Point:** Software ही वो medium है जो **user aur data ke** बीच **link** बनाता है।

3. Data (डेटा)

- Data ही सबसे important part है, क्योंकि पूरा system data को store और manage करने के लिए बनाया गया है।
- इसमें **raw facts** (जैसे नाम, roll no, salary) को meaningful information में organize किया जाता है।

👉 **Point:** "Data is the soul of DBMS."

4. Users (यूजर्स)

- वो लोग जो database use करते हैं। अलग-अलग users के अलग roles होते हैं:
 1. **Database Administrator (DBA):** Database ko maintain करता है, security aur backup ka dhyaan रखता है।
 2. **Database Designers:** Structure design करते हैं (tables, relationships).
 3. **End Users:** जो बस application के through data use करते हैं (जैसे students अपने marks देखते हैं)।
-

5. Procedures (प्रोसीजर/प्रक्रियाएं)

- Rules aur instructions jo बताती हैं कि database ko किस तरह से **design, maintain aur use** किया जाए।
- Example: Backup process, recovery steps, data access policy.

👉 **Point:** Procedures ensure करते हैं कि **DBMS** का सही और सुरक्षित **use** हो।

6. Query Processor (क्वेरी प्रोसेसर)

- ये वो component है जो **user queries ko interpret karke database tak** पहुँचाता है।
- SQL queries को समझकर उन्हें execution plan में बदलता है।

👉 **Point:** User bolta है "SELECT * FROM Student", और Query Processor DB ko कहता है data निकालने के लिए।

7. Database Engine (डेटाबेस इंजन)

- ये DBMS का **core part** है।
- Responsible for:
 - Data storage
 - Retrieval
 - Update
 - Transaction management

👉 **Point:** Engine ही वो दिमाग है जो data ko actually process करता है।

Diagram (Exam ke लिए important hai):

Users → Query Processor → DBMS Software → Database Engine → Data
(Stored in Hardware)

Summary (Quick याद रखने का तरीका):

- **H:** Hardware
- **S:** Software
- **D:** Data
- **U:** Users
- **P:** Procedures
- **Q:** Query Processor
- **E:** Engine

👉 **Trick:** "He Said Don't Use Poor Quality Engine" 😊

📌 Table Structure (टेबल संरचना)

Definition (परिभाषा):

A **table** in DBMS is a collection of rows and columns used to store data in a structured format.

DBMS में **table** एक ऐसा object है जिसमें data **rows (record)** और **columns (fields/attributes)** में store होता है।

🏗️ Structure of a Table

1. Table Name (टेबल का नाम)

- हर table का एक unique name होता है।
- Example: **Student**, **Employee**, **Product**

2. Columns / Fields / Attributes (कॉलम / फील्ड / गुणधर्म)

- ये table के vertical part होते हैं।
- हर column किसी एक type का data store करता है।
- Example: **Roll_No**, **Name**, **Age**, **Course**
- हर column का data type होता है जैसे: INT, VARCHAR, DATE

3. Rows / Records / Tuples (रो / रिकॉर्ड / ट्यूपल)

- ये table के horizontal part होते हैं।
- हर row ek complete record को represent करता है।
- Example: किसी student की पूरी जानकारी (Roll No + Name + Age + Course) एक row होगी।

4. Primary Key (प्राइमरी की)

- कोई ऐसा column (या set of columns) जो हर row को uniquely identify करे।
- Example: **Roll_No** in Student Table

5. Constraints (प्रतिबंध)

- Rules जो data पर लगाए जाते हैं।
- Example:
 - **NOT NULL** → column खाली नहीं रह सकता।
 - **UNIQUE** → value unique होनी चाहिए।
 - **CHECK** → condition लगाई जा सकती है।



Example Table Structure: **STUDENT**

Roll_No (INT)	Name (VARCHAR)	Age (INT)	Course (VARCHAR)
101	Raj	20	BCA
102	Priya	21	BBA
103	Aman	19	B.Tech



Summary (Quick Recall)

- **Table** = Rows + Columns
- **Columns** = Fields (data type defined)
- **Rows** = Records (data values)
- **Primary Key** = Uniquely पहचान करने वाला column

Schema (स्कीमा)

Definition (परिभाषा):

Schema is the **logical description of the entire database**.

👉 यह बताता है कि database में कौन-कौन सी tables होंगी, उनके fields (columns) क्या होंगे, data types क्या होंगे, और उनके आपस के relationships कैसे होंगे।

♦ Simple शब्दों में:

Schema = Structure + Organization of Database

(मतलब data कैसे store होगा और tables आपस में कैसे connected हैं)

Types of Schema (स्कीमा के प्रकार)

1. Physical Schema (भौतिक स्कीमा)

- Data actually **disk** पर कैसे **stored** है, ये बताता है।
- Example: data किस file format में save है, indexing कैसे हो रही है।

2. Logical Schema (तार्किक स्कीमा)

- Data logically कैसे organized है, tables और relationships क्या हैं।
- Example: Student और Course table का relation (one-to-many)।

3. View Schema / Subschema (दृश्य स्कीमा)

- यह बताता है कि किसी **specific user** को कौन सा **data** दिखेगा।
 - Example: Admin को पूरा student data दिखे, लेकिन teacher को सिर्फ Name और Marks दिखे।
-



Example:

Database: University

Schema description हो सकता है:

- **Table 1: Student** (Roll_No, Name, Age, Course_ID)
- **Table 2: Course** (Course_ID, Course_Name, Duration)
- **Relation:** Student.Course_ID → Course.Course_ID (Foreign Key)

👉 यह पूरा structure **University Database Schema** कहलाएगा।



Quick Summary

- **Schema = Blueprint of Database**
- इसमें **tables, fields, relationships, constraints** define होते हैं।
- Types → **Physical, Logical, View Schema**
- Example: University DB का पूरा design (tables + relation)

Three Views of Data (3-Tier Architecture in DBMS)

Database को समझने और manage करने के लिए तीन levels होते हैं:

1. **External View (User View)**
2. **Conceptual View (Logical View)**
3. **Internal View (Physical View)**

इन्हें हम **Three Schema Architecture** भी कहते हैं।

1 External View (बाहरी स्तर / User View)

👉 यह वो level है जहाँ **end users (students, teachers, customers, employees)** database को देखते और use करते हैं।

- हर user को उसके काम का **specific data** ही दिखता है, पूरा database नहीं।
- इसे **View Level** भी कहते हैं।

Features:

- Data का presentation user-friendly होता है।
- User को सिर्फ वो data दिखता है, जो उसे चाहिए।
- Security बढ़ती है क्योंकि हर user को पूरा DB access नहीं होता।

Example:

- Student portal में student को सिर्फ उसका नाम, roll no, subjects और marks दिखते हैं।
- Teacher को सिर्फ उसके पढ़ाए गए students के records दिखेंगे।

2 Conceptual View (वैचारिक स्तर / Logical View)

👉 यह पूरे database का **logical structure** है।

- इसमें सभी entities, attributes और उनके relationships define होते हैं।
- यह level बताता है कि “**Data logically कैसे organized है।**”

Features:

- पूरे DB का एक **global view** होता है।
- इसमें सभी tables, primary keys, foreign keys, और constraints define होते हैं।
- Users और Internal storage के बीच का bridge है।

Example:

- University Database:
 - Student (RollNo, Name, CourseID)
 - Course (CourseID, CourseName, Duration)
 - Relationship → Student.CourseID = Course.CourseID

👉 यह conceptual view database का **blueprint** है।

3 Internal View (आंतरिक स्तर / Physical View)

👉 यह level बताता है कि **data actually disk** पर कैसे **stored** है।

- इसमें file structure, indexes, hashing, compression techniques define होती हैं।
- ये storage optimization और efficiency पर focus करता है।

Features:

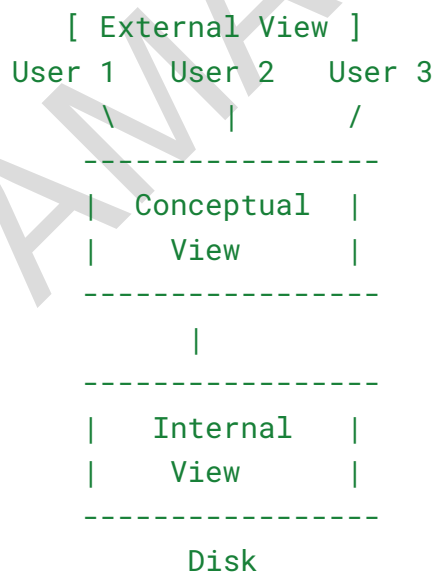
- Low-level details (storage path, blocks, records)।
- Performance improve करने के लिए indexing, clustering आदि का use।
- User को यह level दिखाई नहीं देता।

Example:

- Student table data **disk blocks, B-Tree index, hashing** के जरिए store हो रहा है।
 - RollNo column पर indexing की गई है ताकि search fast हो सके।
-



Diagram (Three Views of Data)





Quick Summary (Exam Point of View)

- **External View:** User-specific data view (security + simplicity)|
- **Conceptual View:** Global logical view (tables + relations)|
- **Internal View:** Physical storage details (indexes + file organization)|
- तीनों levels मिलकर DBMS में **Data Abstraction** और **Data Independence** provide करते हैं।

AMAN JAISWAL