

Data:

Data ek nyu kachu tatva chhe — facts, numbers, observations vagere je by itself meaning na ape. Pan jyaare tene process kariye — organize, interpret kariye — ane ek context ma rakhie, to te **information** banay chhe, je useful ane meaningful bane chhe.

Data Base:

Database ek sangathit (organized) data-sangrah (collection) chhe, jemā data ne aapan asānī thi access, manage ane update kari sakiye chhīe. Data sadharan rite **table** (rows ane columns) ma store thāy chhe agar relational database hoy, ane non-relational database ma document, object, ke graph jaise structure ma pan raakhe chhe.

DBMS:

DBMS (Database Management System) ek software chhe je data ne database ma store, organize ane manage karva ma madad kare chhe. E user (ke applications) ane database vachche ek bridge nu kaam kare chhe — data ne securely store kare chhe, efficiently access karva ane update karva ni suvidha aape chhe. DBMS pramukh rite 2 prakar na hoy chhe:
Relational DBMS (RDBMS) — jaise MySQL, PostgreSQL, Oracle, SQL Server; ane **NoSQL DBMS** — jaise MongoDB, Cassandra, CouchDB.

DBMS Function:

- **Data Storage & Retrieval** — DBMS data ne structured form (tables etc) ma store kare chhe ane queries thi access karva de chhe. [TechTarget+2GeeksforGeeks+2](#)
- **Data Security** — Unauthorized users ne rokva mate access control, authentication ane permissions manage kare chhe. [Enozom+1](#)
- **Data Integrity** — Data ni accuracy ane consistency banavi rahe, constraints (jaise primary key, foreign key) no upyog kare chhe. [Enozom+1](#)
- **Backup & Recovery** — System crash ya failure ni situation ma data ne restore karva mate backup ane recovery mechanism provide kare chhe. [AtulHost+1](#)
- **Concurrency Control** — Ekaj samay ma ane badha users database ma kam kare shake, DBMS conflict ne avoid kare chhe ane data consistent rahe chhe.

ફીચર	File System	DBMS (Database Management System)
Definition / Nature	સ્ટોરેજ ડિવાઇસ પર ફાઇલોને સ્ટોર અને ઓર્ગનાઇઝ કરવાની રીત છે. (GeeksforGeeks)	ડેટાબેઝ મેનેજમેન્ટ માટેનું સોફ્ટવેર છે, જે સ્ટ્રક્ચર્ડ ડેટાને મેનેજ કરે છે. (GeeksforGeeks)
Data Handling / Structure	ડેટા ફાઇલો (ફોલ્ડર્સ, ડિરેક્ટરીઓ)માં સ્વતંત્ર રૂપે સ્ટોર થાય છે. (Testbook)	ડેટા ટેબલમાં (rows-columns) કે અન્ય relational structure માં હોય છે, જેમાં સંબંધ (relations) પણ હોઈ શકે છે. (Testbook)
Access / Data Retrieval	ફાઇલોને સીધા OS-level calls દ્વારા access કરવી પડે છે (file path, file operations). (Unstop)	SQL જેવી ક્વેરી ભાષા અને DBMS tools દ્વારા data ને ચુનંદગીથી access / retrieve કરી શકાય છે. (Difference Wiki)
Data Redundancy & Integrity	redundancy વધારે છે, કારણ કે દરેક ફાઇલ અલગ હોઈ શકે છે અને constraints અઓંઝોમેન્ટિક નથી. (GeeksforGeeks)	redundancy ઘટાડી શકાય છે (normalization), અને integrity constraints (PK, FK) enforce કરી શકાય છે. (GeeksforGeeks)
Security	ફાઇલ પર મૂળભૂત OS permission security હોય છે (read, write, etc.). (Unstop)	વધુ મજબૂત security: user authentication, authorization, access control, auditing possible. (GeeksforGeeks)
Backup & Recovery	બેકઅપ / રિકવરી સામાન્ય રીતે મેન્યુઅલ હોય છે, built-in mechanism બહુ ઓછી હોય છે. (GeeksforGeeks)	DBMS માં in-built backup & recovery mechanism હોય છે. (GeeksforGeeks)
Concurrency (એક્સાથે access)	concurrency નિયમ હોય શકે છે, એક્સાથે ધણાા users access માં ખૂબ જોહાં હોઈ શકે છે. (Unstop)	DBMS concurrency control આપે છે — transactions, locks, isolation levels દ્વારા data consistency બર્ફરાર રાપે છે. (Unstop)
Data Independence / Abstraction	ફાઇલ સ્ટોરેજની રાજીધરી માહિતી મળતી રહે છે — ડેટા કેવી રીતે સ્ટોર થયું છે તેના details જોવા પડે છે. (WikiTechy)	DBMS abstraction આપે છે — physical storage details છપાયેલી હોય છે; logical view users ને માંગ મુજબ અપાશે. (Annamalai University)
Examples	NTFS, EXT, FAT32, વગેરે. (GeeksforGeeks)	Oracle, MySQL, SQL Server, PostgreSQL, વગેરે. (GeeksforGeeks)
Cost & Complexity	સાંદું છે, ઓછું ખર્ચાળ અને સરળ સેવાપાત્ર છે. (Digital Notes Hub)	વધુ કોમ્પ્લેક્સ છે, સેટઅપ અને મેનેજમેન્ટ ખર્ચાળ હોઈ શકે છે. (Shiksha)

DBMS na Application (Use Cases) ane Kevu Madad Kare Chhe

1. Airline & Train Reservation System

- Udyog: IRCTC (railways), airline booking sites.
- DBMS madad kare chhe seat availability maintain karva, double-booking rokva, ane real-time update karva jevī booking ane cancellation na data ma. [GeeksforGeeks+2Scaler+2](#)

2. Education (Universities / Schools)

- Udyog: University portals, online exam systems.
- DBMS ma student records (admission, attendance, marks, fees) store thay chhe, ane students / teachers mate login-based access have. [GeeksforGeeks+1](#)

3. Banking System

- Udyog: ATM systems, online banking, UPI etc.
- DBMS customer info, account balance, transactions store kare chhe, ane transaction sathe sathe real-time update kare chhe.

[TechTarget+2shripada.com+2](#)

4. E-commerce Platforms

- Udyog: Amazon, Flipkart vagere.
- DBMS product details (price, stock), user cart data, orders vagere store kare chhe ane massive number of customers ne support kare chhe. [Scaler+1](#)

5. Telecommunication Systems

- Udyog: Telecom companies — Jio, Airtel, Vodafone.
- DBMS call records, billing info, data usage, customer profiles manage kare chhe. [BeginnersBook -](#)

6. Healthcare Systems

- Udyog: Hospitals, patient management systems.
- DBMS patient medical history, prescriptions, test-reports store kare chhe ane doctors ne jaldi access aape chhe. [GeeksforGeeks](#)

7. Social Media Platforms

- Udyog: Instagram, Facebook, WhatsApp.
- DBMS user profiles, posts, chats, followers, likes vagere data manage kare chhe ane real-time updates possible banave chhe. [GeeksforGeeks](#)

8. Government & Public Sector

- Udyog: Aadhaar, Voter ID, Passport, Driving License ane bijā public systems.
- DBMS citizens' personal details securely store kare chhe ane identity verification & other public-service operations ma help kare chhe. [upGrad+1](#)

9. Scientific / Weather Forecasting / Big Data (*thodi advanced use*)

- Udyog: Weather apps, research organizations.
- DBMS large volume data (jaise satellite weather data) store kare chhe ane pattern analysis karva mate use thay chhe

Database Schema su che?

Schema ek blueprint (naksa) chhe je batave chhe ke database ma data kem organized karvu chhe. [Webopedia+2](#)[GeeksforGeeks+2](#)

Logical Schema

- Logical (ke conceptual) schema ek high-level design chhe, je batave chhe **su data store karvu chhe** — tables, columns (attributes), ane temna relationships. [Sivo+1](#)
 - Ema data constraints pan hova chhe: primary keys, foreign keys, not-null vagere. [GeeksforGeeks+1](#)
 - Logical schema DB-agnostic hoy shake chhe — mane storage detail ni parvah nathi, sirf data structure joiye chhe. [Stack Overflow+1](#)
 - Aa level par “logical data independence” possible chhe — yani ke logical schema ma badlāv kari sakay chhe bina application programs ne kharab kare. [Engineering LibreTexts](#)
-

Physical Schema

- Physical schema batave chhe ke **data sacchi rite hardware par kem store thāvani chhe** — file structure, indexes, partitions, tablespaces vagere. [Wikipedia](#)
 - Ema data types, storage blocks, file paths define thay chhe. [GeeksforGeeks](#)
 - Physical schema performance ne optimize karva ma madad kare chhe — jaise index banāvānā decisions, partitioning etc. [Milvus](#)
 - “Physical data independence” ni khasiyat chhe — storage ma changes kariye, pan logical schema (ani structure) par effect na pade. [Wikipedia](#)
-

Example:

- Logical schema: *Customer* table — CustomerID, Name, Email; *Order* table — OrderID, CustomerID, Date.
- Physical schema: “Email” field ma data type VARCHAR (255) chhe, Customer.Email par index che, ane Order table ne date ne aadār par partition karo chho.

Codd's 12 Rules (Edgar F. Codd) — Relational DBMS matey

Dr. Edgar F. Codd ae relational database management system (RDBMS) mate 12 (actually 13, 0 thi 12 sudhi) rules propose karyaa hata je relational DBMS ne “sachu” relational manaay te ensure kare chhe. [Wikipedia+2](#)[database.guide+2](#)

Aa chhe tenu simple saar:

Rule 1 – Information Rule

Badhu data (user data ane metadata) ek table ma cells (rows & columns) ni value roope store thavu joiye. [Wikipedia+1](#)

1. Rule 2 – Guaranteed Access Rule

Pratyek datum (atomic value) logically access thai shake table name + primary key (row) + column name ni madad thi. [Redgate Software+1](#)

2. Rule 3 – Systematic Treatment of NULL Values

NULL values (khaali / missing / not applicable) ne uniform, system-wide tarike treat karvu joiye, ane e alag tarike handle thavu joiye zero ya blank string thi. [Redgate Software+1](#)

3. Rule 4 – Active Online Catalog

Database ni schema (metadata) ne ek catalog ma store karvu joiye, ane e catalog ma data access karva mate same relational language (jaise SQL) use thai shake. [Our Education Blog](#)

4. Rule 5 – Comprehensive Data Sublanguage

DBMS ek powerful, well-defined language support kare je data definition (DDL), data manipulation (DML), integrity constraints, transaction control vagere kare shake. [apgcm.edu.in+1](#)

5. Rule 6 – View Updating Rule

Je views (logical tables) theoretically update kari sakay chhe, te system ne practical pan update karva ni capacity joiye. [Our Education Blog](#)

6. Rule 7 – High-level Insert, Update, Delete

Multi-row operations supported hovavi joiye — set operations (jaise union, intersection) ane insert / delete / update ekaj query thi karva joiye. [GeeksforGeeks](#)

7. Rule 8 – Physical Data Independence

Physical storage change (indexing, partitioning, file layout) kariye to application programs ne koi farak na pade. [GeeksforGeeks](#)

8. Rule 9 – Logical Data Independence

Logical schema ma changes (jaise table structure change) karva par bhi users' applications ne effect na pade. [Our Education Blog](#)

9. Rule 10 – Integrity Independence

Integrity constraints (jaise primary key, foreign key) ne application code ni bahar DB level par define karvu joiye, ane DBMS automatically enforce kare chhe.

[GeeksforGeeks](#)

10. Rule 11 – Distribution Independence

Data agar alag-alag physical locations par hoi (distributed database), to user ne ae distribution lage nai — system background ma handle kare chhe. [SQLServer | PowerBI](#)

11. Rule 12 – Non-Subversion Rule

Jo low-level language (record-by-record) available hoy, to te relational integrity constraints ne bypass na kari shake — high-level rule vagar structure ne todva na de.

Data Abstraction in DBMS — Samjhavanu

Definition:

DBMS मां *data abstraction* એ પ્રક્રિયા છે જેમાં બેક-એન્જની અટિલતાઓ (irrelevant details) યુઝર પાસેથી છુપાવવામાં આવે છે અને માત્ર જરૂરી માહિતી જ બતાવવામાં આવે છે. એનો ઉદ્દેશ એ છે કે યુઝરને અસરકારક અને સુરક્ષિત રિટે ડેટા ઉપલબ્ધ કરાવવો, તેના માટે એ જોઈએ કે તે અંદરની સ્ટોરેજ કે મૂલભૂત ડેટા માળખા વિશે જાણે નહીં.

મુખ્ય મુદ્દાઓ:

- ચુઝર્સ માત્ર એટલે સબંધિત અને જરૂરી માહિતી જ જોઈ શકે છે — સામગ્રી કેવી રીતે સ્ટોર છે, તે તેમને ન સમજાવવી પડે. [GeeksforGeeks+2Scaler+2](#)
- સિક્યુરિટી વધે છે, કારણ કે સંવેદનશીલ (sensitive) ડિટેલ્સ છુપાયેલી હોય છે. www.tpointtech.com
- એ વપરાશકર્તા માટે સરળ બનાવે છે — ડેટા એક્સેસ સરળ, ફાસ્ટ અને ચુઝર-ફેન્ડલી બને છે. [GeeksforGeeks](#)
- ઉદાહરણ તરીકે: ઓનલાઇન કપડાં ખરીદતી વખતે, તમે રંગ, સાઇઝ, બ્રાન્ડ જુઓ છો — પણ તમે manufacturing કે સ્ટોરેજની ‘જટિલ’ માહિતી ન જોઈ શકો. [Scaler+1](#)

Data Abstraction ના સ્તરો (Levels)

DBMS માં data abstraction મુખ્યત્વે 3 સ્તર હોય છે: [GeeksforGeeks+2WikiTechy+2](#)

1. Physical / Internal Level

- સૌથી નીચું abstraction સ્તર છે. [GeeksforGeeks](#)
- આ સ્તરે વર્ણવવામાં આવે છે કે ડેટા પોલીસ તરીકે કેવી રીતે સ્ટોર થાય છે — ફાઇલ્સ, access methods, disk allocation વગેરે. [GeeksforGeeks+1](#)
- બધું જ જટિલ હોય છે અને યુઝરને દેખાડવામાં આવતો ન હોય. [GeeksforGeeks](#)
- ડેટાબેઝ એડમિનિસ્ટ્રેટર (DBA) દ્વારા મેનેજ થાય છે. www.tpointtech.com

2. Logical / Conceptual Level

- મધ્યમ સ્તર છે. [GeeksforGeeks](#)
- અહીં બતાવવામાં આવે છે કે શું ડેટા સ્ટોર છે: ટેબલ, એટ્રિબ્યુટ્સ, સંબંધો (relationships), constraints. [GeeksforGeeks](#)
- આ સ્તર પર ડેવલપર્સ અથવા DBA કામ કરે છે. www.tpointtech.com
- physical changes (જેમ કે storage method) માટે logical level જળવાછ શકે છે — *physical data independence*. [GeeksforGeeks](#)

3. View / External Level

- સૌથી ઊંચું abstraction સ્તર છે. [GeeksforGeeks](#)
- દરેક યુઝર માટે જુદુ “view” હોય શકે છે — જેમ કે અભ્યાસ સંપૂર્ણ ડેટાબેસ ન joiye, માત્ર તે જ વિભાગ કે માહિતી જોઈ રહી છે. [WikiTechy](#)
- આ રીતે, users ને માત્ર તે જ માહિતી મળે છે જે તેમને જરૂરી છે, બાકી complexity છુપાયેલી રહે છે. [GeeksforGeeks](#)

- Security પણ અહીં વધે છે — કેટલાક યુઝર્સ મીહાજન થોડું કેટા જોઈ શકે છે. [LIS Academy](#)
-

Data Independence (Abstraction સાથે સંબંધ)

- Data abstraction DBMS માં *data independence* લાવવા માટે મદદ કરે છે. [GeeksforGeeks](#)
- **Physical Data Independence:** તમે physical schema (જ્યાં કેમ સ્ટોર છે) બદલશો પણ logical schema પર અસર ન પડો. [GeeksforGeeks](#)
- **Logical Data Independence:** logical schema (ટેબલ, એટ્રિબ્યુટ) બદલાશો, પણ view level (ઉપયોગકર્તા દ્વારા) પર એ બદલાવનો અસર ન પડો.

Data Independence in DBMS — Samjhavani + Types

Data Independence ek DBMS ni khasiyat chhe ke tene ek level (schema) par badlava karva ni ajadi hoy etle ke biji level ne prabhavit na kare. [GeeksforGeeks+2](#)[GeeksforGeeks+2](#)

Types of Data Independence

1. Physical Data Independence

- Ee matlab ke tame internal (physical) schema ma badlav kari sako cho — jaise ke data ne store karva nu method (file structures, indexing), ya storage device (HDD thi SSD) badli sako cho — parantu logical schema (tables, relationships) ne effect na pade. [GeeksforGeeks+1](#)
- Example: Index add karvu, data files ek drive thi biji drive ma move karvu, ya compression method badlavu. [GeeksforGeeks](#)
- Benefit: Performance optimization kari sako cho bina existing applications ne change kare. [GeeksforGeeks](#)

2. Logical Data Independence

- Ee matlab ke tame logical / conceptual schema (jaise tables, columns, relationships) ma badlav kari sako cho — parantu users' view (external schema) ya applications ne j zaroorat na pade badlava ni. [GeeksforGeeks+1](#)
 - Example: Table ma navi column add karvu, ya 2 tables ne ek view ma merge karvu. [GeeksforGeeks](#)
 - Benefit: Database ne evolve kari sako cho business ya design ni requirement badlata hoy, ane pura system (apps) ne lakhvani zaroorat na pade. [GeeksforGeeks](#)
 - Logical Data Independence ma physical changes (storage) karta thodi mushkil padti hoi chhe — karan ke applications logical structure par depend kare chhe. [GeeksforGeeks](#)
-

DBMS Architecture (1-Tier, 2-Tier, 3-Tier) — Short Samjhavanu

DBMS architecture batave chhe ke DBMS ni design kem layered chhe, ane users, applications ane database ek biji sathe kem interact kare chhe. [GeeksforGeeks+1](#)

- **1-Tier Architecture**

- Ekaj machine par client + server + database chhe. [GeeksforGeeks](#)
- Udaaran: Microsoft Excel jema user interface ane data storage same device par hoy. [GeeksforGeeks](#)
- Faayda: Simple chhe, Setup sasto chhe. [GeeksforGeeks](#)
- Nuksan: Multi-user support limited, security ochi, data central control mushkil. [GeeksforGeeks](#)

- **2-Tier Architecture**

- Client-server model: User interface (client) direct database server sathe communicate kare chhe. [GeeksforGeeks](#)
- Udaaran: School library management system jya client app direct DB server access kare chhe. [GeeksforGeeks](#)
- Faayda: Access asaan chhe, low cost, easily deploy kari sakay. [GeeksforGeeks](#)
- Nuksan: Scalability limited, security issue ho sakay chhe, client-server tight coupling chhe. [GeeksforGeeks](#)

- **3-Tier Architecture**

- Tino layer chhe: Client layer, Application / middle layer (app server), ane Database layer. [GeeksforGeeks](#)
- Client application server ne vichare che pehla, pachhi application server DB server sathe interact kare chhe. [PrepBytes+1](#)
- Faayda: Scalability vadhe chhe, security better chhe (client direct DB sathe na batave), data integrity maintain rahe chhe. [GeeksforGeeks](#)
- Nuksan: Complex design, maintenance mushkil, cost vadhe chhe. [GeeksforGeeks](#)