

"Jai Shree Ram"

(Raw fact and fig. to give info.)

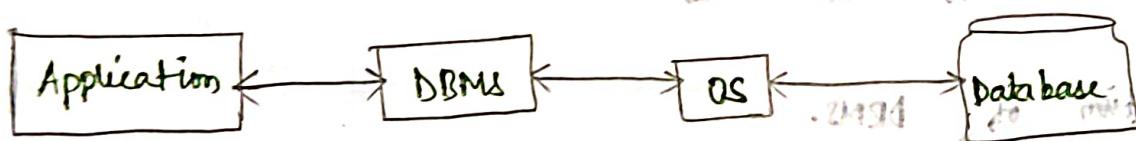
DBMS - Data Base Management system.

(Raw Fact) (Name, Roll) → fact resume

06/June (5:44)

- ↳ A DBMS is a collection of interrelated data and a set of programs to access those data.
- ↳ DBMS is used to organize the data in the form of a table, schema, views, reports, etc.
- ↳ Primary goal of DBMS is to provide a convenient and efficient way to store and retrieve database information that is both convenient and efficient.
- ↳ DBMS is a collection of related information stored so that it is available to many users for diff. purposes.
- ↳ DBMS is a collection of programs that enables users to create and maintain their database.

Fig



- ↳ DBMS can also be defined as an interface b/w the application program and OS to Access and Manipulate that database.

- ↳ DBMS is a software which is used to manage the database.

For eg; MySQL, Oracle, etc are very popular commercial database which is used in different application. When large amount of data is present, then DBMS comes into picture.

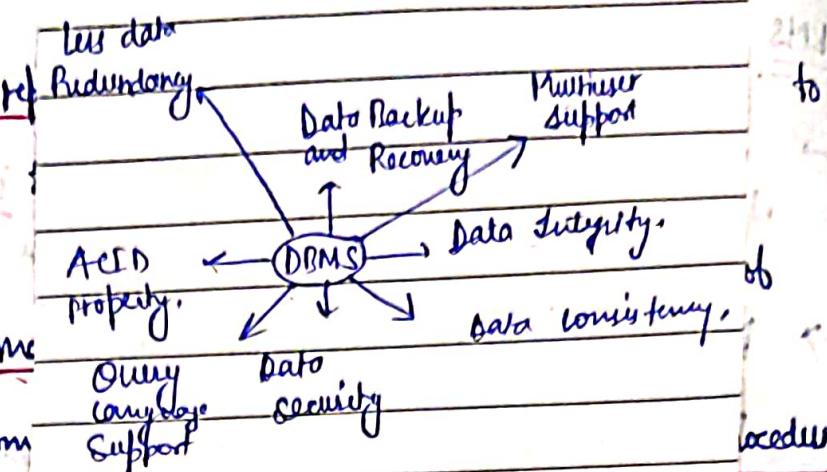
Characteristics of DBMS.

- It uses a digital store and manages data.
- It can provide the process that meets DBMS continues automatically.
- It contains ACID (Atomicity, Consistency, Isolation, Durability) properties which maintains state in case of system failure.
- It can reduce the complex relationship b/w data.
- It is used to support manipulation and processing of data.
- It provides security of data.
- It can view the DB from diff. viewpoint according to the requirement of the user.

* Application of DBMS.

- (i) Banking :- For maintaining customer information, accounts, wants and banking transaction.
- (ii) Universities :- For maintaining student records, course registration and grades.
- (iii) Railway Reservation :- For checking the availability of reservation in diff. trains, tickets, etc.
- (iv) Airlines :- For reservation and schedule information.
- (v) Telecommunication - call mode, monthly bills, etc.
- (vi) Finance - Holiday, sales, dates. → Purchase information.
- (vii)

Characteristics of DBMS.



Advantages of DBMS :-

- (i) control over redundancy :- It can control data redundancy because it stores all the data in one single database file and each record data is placed in the database.
- (ii) Data sharing :- In DBMS - the authorized user of an organization can share data among multiple users.
- (iii) Easily Maintained :- It can be easily maintained due to centralized nature of database system.
- (iv) Reduced time :- It reduces development time and maintenance time.
- (v) Backup :- It provides backup and recovery subsystem which makes automatic backup of data from hardware and software failure and recovery the data if required.
- (vi) Multiple user Interface :- It provides different types of user interface like GUI, Application program Interface (API).

Disadvantages of DBMS :-

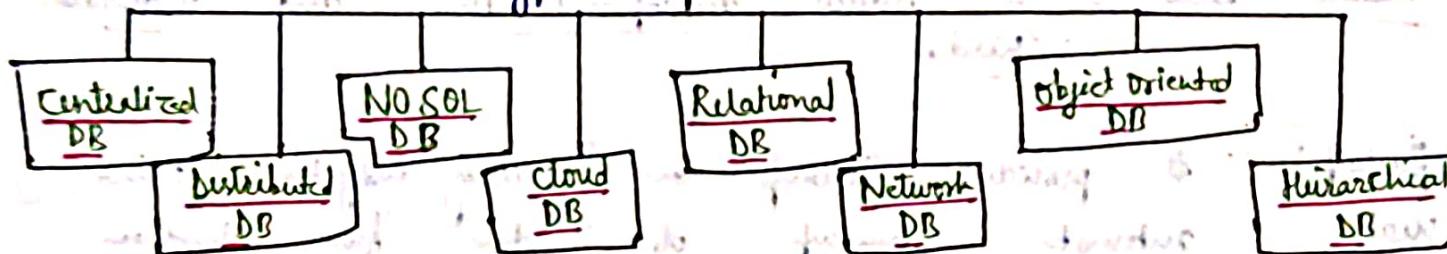
- (i) Cost of Hardware and software :- It requires high speed of data processor and large memory size to run DBMS.
- (ii) Size :- It occupies large space of disk and large memory to run them efficiently.
- (iii) Complexity :- DB system creates additional complexity and Requirements.

Higher impact of failure: Failure is highly impacted the data because in most of the organization all the data stored in a single DB and if the DB is damaged due to electric failure or DB corruption then the data may be lost forever.

* Types of database

↳ There are various types of DB used for storing different varieties of data.

Type of data Base



(i) Centralized DataBase

↳ It is a type of DB that stores data at one centralized database system.

↳ It enables the users to access the stored data from different location through several application.

↳ This App. contains the authentication process to let user access data security. (Security access karega data ko).

↳ An example of a centralized database can be central library that carries a central database of each library in a college / university.

↳ All data is stored in one central place.

- (iii) Distributed database \rightarrow data stored in Multi Physical location.
- \hookrightarrow In distributed database data is distributed among diff. db system of an organisation.
 - \hookrightarrow These db system are connected via communication links. Such links help the end user to access the data easily.

\hookrightarrow It is divided into two sub parts
it is one complete distributed database.

(i) Homogeneous database

(All nodes are same DBMS and are aware of each other.)

(ii) Heterogeneous database

(Different nodes may use diff. DBMS and may not be aware of each other.)

(iv) Relational Database

\hookrightarrow It stores data in the form of rows (Tuple) and columns (Attributes) and together forms a table (Relation).

\hookrightarrow A RDB uses SQL for storing, manipulating and maintaining the data.

\hookrightarrow Each table in the database contains a key that makes the data unique from others.

\hookrightarrow Eg. of RDB are MySQL, Microsoft SQL server, Oracle, etc.

(v) No SQL database \rightarrow it doesn't use a fixed schema. (fixed structure).

\hookrightarrow No SQL / Not only SQL is a type of database that is used for storing a wide range of data sets.

\hookrightarrow It is not a relational database as it stores data not only in tabular form but in several diff. ways.

- It is also divided into four sub-parts
- (i) Key Value Storage → Redis, Dynamo DB
 - (ii) Document oriented database → MongoDB, Couch DB
 - (iii) Graph database → Neo4j, Apache Cassandra
 - (iv) Wide-column store → HBase, Apache Hadoop

(v) Cloud database

- ↳ It is a type of database where data is stored in a virtual environment and executes over the cloud computing platform.
- ↳ It provides users with various cloud computing services (SaaS, PaaS, IaaS, etc.) for accessing the database.
- ↳ Some examples of cloud DB such as Amazon web services (AWS), microsoft Azure, Google cloud SQL, etc.

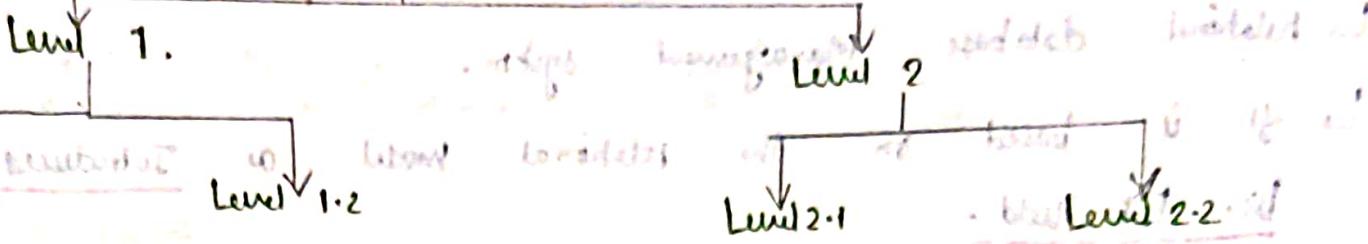
(vi) Object-oriented database

- ↳ This type of DB that uses the object base data model approach for storing data in the database system.
- ↳ The data is represented and stored as objects which are similar to the objects used in the object oriented Programming language (OOPS).
- ↳ Example :- DB4o (Database for object), object DB, versant object database

(vii) Hierarchical database

- ↳ It is a type of database that stores data in the form of parent - children relationship.
- ↳ It organizes data in tree like structure.

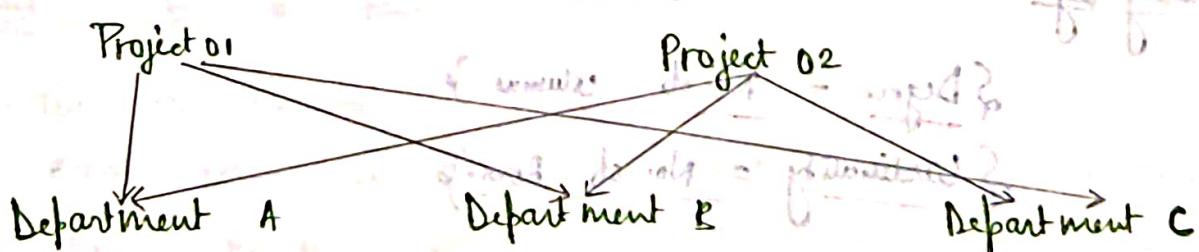
3. Hierarchical & Network



- ↳ Data gets stored in the form of records that are connected via link.
- ↳ Each child record in the tree will contain only one parent. On the other hand each parent record can have multiple child records.
- ↳ Examples - IBM Information Management System (IMS), XML DB, Windows Registry etc.

(viii) Network database

- ↳ It is the database that typically follows the network data model.
- ↳ In this representation of data it is in the form of node connected via links between them.
- ↳ It allows each record to have multiple children and parent node to form a generalized graph structure.



What is RDBMS?

tuple - Row

- ↳ Relational database Management system.
- ↳ It is based on the relational model as Introduced by Dr. E.F. Codd.
- ↳ It stores data in the form of Related tables.
- ↳ An important feature of RDBMS is that a single database can be spread across several tables.
- ↳ All modern DBMS like SOL, MS SOL Server, IBM DB2, ORACLE, My-SOL and microsoft Access are based on RDBMS.
- ↳ A relational database is the most commonly used DB. It contains several tables and each table has its primary key.
- ↳ Due to a collection of an organized set of tables data can be easily accessed in RDBMS.
- ↳ Everything is in a relational form of relations (Tables)

Id.	Name	Post	Salary
1	John	Manager	50000
2	David	Manager	50000
3	Mike	Manager	50000
4	Steve	Manager	50000
5	Mark	Manager	50000
6	Paul	Manager	50000
7	Tom	Manager	50000
8	Mike	Manager	50000
9	Steve	Manager	50000
10	Mark	Manager	50000
11	Paul	Manager	50000
12	Tom	Manager	50000
13	Mike	Manager	50000
14	Steve	Manager	50000
15	Mark	Manager	50000
16	Paul	Manager	50000
17	Tom	Manager	50000
18	Mike	Manager	50000
19	Steve	Manager	50000
20	Mark	Manager	50000
21	Paul	Manager	50000
22	Tom	Manager	50000
23	Mike	Manager	50000
24	Steve	Manager	50000
25	Mark	Manager	50000
26	Paul	Manager	50000
27	Tom	Manager	50000
28	Mike	Manager	50000
29	Steve	Manager	50000
30	Mark	Manager	50000
31	Paul	Manager	50000
32	Tom	Manager	50000
33	Mike	Manager	50000
34	Steve	Manager	50000
35	Mark	Manager	50000
36	Paul	Manager	50000
37	Tom	Manager	50000
38	Mike	Manager	50000
39	Steve	Manager	50000
40	Mark	Manager	50000
41	Paul	Manager	50000
42	Tom	Manager	50000
43	Mike	Manager	50000
44	Steve	Manager	50000
45	Mark	Manager	50000
46	Paul	Manager	50000
47	Tom	Manager	50000
48	Mike	Manager	50000
49	Steve	Manager	50000
50	Mark	Manager	50000
51	Paul	Manager	50000
52	Tom	Manager	50000
53	Mike	Manager	50000
54	Steve	Manager	50000
55	Mark	Manager	50000
56	Paul	Manager	50000
57	Tom	Manager	50000
58	Mike	Manager	50000
59	Steve	Manager	50000
60	Mark	Manager	50000
61	Paul	Manager	50000
62	Tom	Manager	50000
63	Mike	Manager	50000
64	Steve	Manager	50000
65	Mark	Manager	50000
66	Paul	Manager	50000
67	Tom	Manager	50000
68	Mike	Manager	50000
69	Steve	Manager	50000
70	Mark	Manager	50000
71	Paul	Manager	50000
72	Tom	Manager	50000
73	Mike	Manager	50000
74	Steve	Manager	50000
75	Mark	Manager	50000
76	Paul	Manager	50000
77	Tom	Manager	50000
78	Mike	Manager	50000
79	Steve	Manager	50000
80	Mark	Manager	50000
81	Paul	Manager	50000
82	Tom	Manager	50000
83	Mike	Manager	50000
84	Steve	Manager	50000
85	Mark	Manager	50000
86	Paul	Manager	50000
87	Tom	Manager	50000
88	Mike	Manager	50000
89	Steve	Manager	50000
90	Mark	Manager	50000
91	Paul	Manager	50000
92	Tom	Manager	50000
93	Mike	Manager	50000
94	Steve	Manager	50000
95	Mark	Manager	50000
96	Paul	Manager	50000
97	Tom	Manager	50000
98	Mike	Manager	50000
99	Steve	Manager	50000
100	Mark	Manager	50000
101	Paul	Manager	50000
102	Tom	Manager	50000
103	Mike	Manager	50000
104	Steve	Manager	50000
105	Mark	Manager	50000
106	Paul	Manager	50000
107	Tom	Manager	50000
108	Mike	Manager	50000
109	Steve	Manager	50000
110	Mark	Manager	50000
111	Paul	Manager	50000
112	Tom	Manager	50000
113	Mike	Manager	50000
114	Steve	Manager	50000
115	Mark	Manager	50000
116	Paul	Manager	50000
117	Tom	Manager	50000
118	Mike	Manager	50000
119	Steve	Manager	50000
120	Mark	Manager	50000
121	Paul	Manager	50000
122	Tom	Manager	50000
123	Mike	Manager	50000
124	Steve	Manager	50000
125	Mark	Manager	50000
126	Paul	Manager	50000
127	Tom	Manager	50000
128	Mike	Manager	50000
129	Steve	Manager	50000
130	Mark	Manager	50000
131	Paul	Manager	50000
132	Tom	Manager	50000
133	Mike	Manager	50000
134	Steve	Manager	50000
135	Mark	Manager	50000
136	Paul	Manager	50000
137	Tom	Manager	50000
138	Mike	Manager	50000
139	Steve	Manager	50000
140	Mark	Manager	50000
141	Paul	Manager	50000
142	Tom	Manager	50000
143	Mike	Manager	50000
144	Steve	Manager	50000
145	Mark	Manager	50000
146	Paul	Manager	50000
147	Tom	Manager	50000
148	Mike	Manager	50000
149	Steve	Manager	50000
150	Mark	Manager	50000
151	Paul	Manager	50000
152	Tom	Manager	50000
153	Mike	Manager	50000
154	Steve	Manager	50000
155	Mark	Manager	50000
156	Paul	Manager	50000
157	Tom	Manager	50000
158	Mike	Manager	50000
159	Steve	Manager	50000
160	Mark	Manager	50000
161	Paul	Manager	50000
162	Tom	Manager	50000
163	Mike	Manager	50000
164	Steve	Manager	50000
165	Mark	Manager	50000
166	Paul	Manager	50000
167	Tom	Manager	50000
168	Mike	Manager	50000
169	Steve	Manager	50000
170	Mark	Manager	50000
171	Paul	Manager	50000
172	Tom	Manager	50000
173	Mike	Manager	50000
174	Steve	Manager	50000
175	Mark	Manager	50000
176	Paul	Manager	50000
177	Tom	Manager	50000
178	Mike	Manager	50000
179	Steve	Manager	50000
180	Mark	Manager	50000
181	Paul	Manager	50000
182	Tom	Manager	50000
183	Mike	Manager	50000
184	Steve	Manager	50000
185	Mark	Manager	50000
186	Paul	Manager	50000
187	Tom	Manager	50000
188	Mike	Manager	50000
189	Steve	Manager	50000
190	Mark	Manager	50000
191	Paul	Manager	50000
192	Tom	Manager	50000
193	Mike	Manager	50000
194	Steve	Manager	50000
195	Mark	Manager	50000
196	Paul	Manager	50000
197	Tom	Manager	50000
198	Mike	Manager	50000
199	Steve	Manager	50000
200	Mark	Manager	50000
201	Paul	Manager	50000
202	Tom	Manager	50000
203	Mike	Manager	50000
204	Steve	Manager	50000
205	Mark	Manager	50000
206	Paul	Manager	50000
207	Tom	Manager	50000
208	Mike	Manager	50000
209	Steve	Manager	50000
210	Mark	Manager	50000
211	Paul	Manager	50000
212	Tom	Manager	50000
213	Mike	Manager	50000
214	Steve	Manager	50000
215	Mark	Manager	50000
216	Paul	Manager	50000
217	Tom	Manager	50000
218	Mike	Manager	50000
219	Steve	Manager	50000
220	Mark	Manager	50000
221	Paul	Manager	50000
222	Tom	Manager	50000
223	Mike	Manager	50000
224	Steve	Manager	50000
225	Mark	Manager	50000
226	Paul	Manager	50000
227	Tom	Manager	50000
228	Mike	Manager	50000
229	Steve	Manager	50000
230	Mark	Manager	50000
231	Paul	Manager	50000
232	Tom	Manager	50000
233	Mike	Manager	50000
234	Steve	Manager	50000
235	Mark	Manager	50000
236	Paul	Manager	50000
237	Tom	Manager	50000
238	Mike	Manager	50000
239	Steve	Manager	50000
240	Mark	Manager	50000
241	Paul	Manager	50000
242	Tom	Manager	50000
243	Mike	Manager	50000
244	Steve	Manager	50000
245	Mark	Manager	50000
246	Paul	Manager	50000
247	Tom	Manager	50000
248	Mike	Manager	50000
249	Steve	Manager	50000
250	Mark	Manager	50000
251	Paul	Manager	50000
252	Tom	Manager	50000
253	Mike	Manager	50000
254	Steve	Manager	50000
255	Mark	Manager	50000
256	Paul	Manager	50000
257	Tom	Manager	50000
258	Mike	Manager	50000
259	Steve	Manager	50000
260	Mark	Manager	50000
261	Paul	Manager	50000
262	Tom	Manager	50000
263	Mike	Manager	50000
264	Steve	Manager	50000
265	Mark	Manager	50000
266	Paul	Manager	50000
267	Tom	Manager	50000
268	Mike	Manager	50000
269	Steve	Manager	50000
270	Mark	Manager	50000
271	Paul	Manager	50000
272	Tom	Manager	50000
273	Mike	Manager	50000
274	Steve	Manager	50000
275	Mark	Manager	50000
276	Paul	Manager	50000
277	Tom	Manager	50000
278	Mike	Manager	50000
279	Steve	Manager	50000
280	Mark	Manager	50000
281	Paul	Manager	50000
282	Tom	Manager	50000
283	Mike	Manager	50000
284	Steve	Manager	50000
285	Mark	Manager	50000
286	Paul	Manager	50000
287	Tom	Manager	50000
288	Mike	Manager	50000
289	Steve	Manager	50000
290	Mark	Manager	50000
291	Paul	Manager	50000
292	Tom	Manager	50000
293	Mike	Manager	50000
294	Steve	Manager	50000
295	Mark	Manager	50000
296	Paul	Manager	50000
297	Tom	Manager	50000
298	Mike	Manager	50000
299	Steve	Manager	50000
300	Mark	Manager	50000
301	Paul	Manager	50000
302	Tom	Manager	50000
303	Mike	Manager	50000
304	Steve	Manager	50000
305	Mark	Manager	50000
306	Paul	Manager	50000
307	Tom	Manager	50000
308	Mike	Manager	50000
309	Steve	Manager	50000
310	Mark	Manager	50000
311	Paul	Manager	50000
312	Tom	Manager	50000
313	Mike	Manager	50000
314	Steve	Manager	50000
315	Mark	Manager	50000
316	Paul	Manager	50000
317	Tom	Manager	50000
318	Mike	Manager	50000
319	Steve	Manager	50000
320	Mark	Manager</td	

- ↳ Each column has a unique name.
- ↳ The sequence of rows is insignificant.
- ↳ Row or Record is a situation with its breaks.
- ↳ A row of a table is also called a record or tuple.
- ↳ Row contains the specific information of each entry.
- ↳ It is the horizontal entity in the table.

Properties of a Row

- ↳ No two rows are identical to each other in all their entries.
- ↳ All tuples/rows of the relation have the same format and same no. of entries.
- ↳ The order of the rows is irrelevant. They are identified by their content, not by their Position.

- ↳ columns / Attributes / subfields holding all the details of a relation.
- ↳ A column is a vertical entity in the table which contains all the information associated with a specific field in a table.

* Properties of columns

- ↳ Every column of a relation must have a name.
- ↳ Null values are permitted for the columns.
- ↳ Default value can be specified for an column automatically inserted if no other value is specified for an column.
- ↳ Column that uniquely identify each row of a relation are the Primary Key.

Data item / cell

- ↳ The smallest unit of data in the table is the individual data item:
- ↳ It is stored at the intersection of row and column.

Degree

- ↳ The total no. of columns that comprise a relation is known as the degree of the table.

Cardinality

- ↳ The total no. of rows at any one time in a relation is known as the cardinality.
- ↳ The relation whose cardinality is zero is called an empty table.

Domain

- ↳ The domain refers to the possible values each column can contain.
- ↳ It can be specified using standard datatypes such as Integer, floating numbers etc.

- * Dr. E.F Codd's Rules for RDBMS
 - ↳ Dr. E.F Codd is an IBM researcher who first developed the Relational data Model in 1970.
 - ↳ In 1985 Dr. Codd published a list of 12 rules that define an ideal RDB database and has also provided a guideline for the design of all Relational DB.
- Rule 1 The Information Rule
- ↳ This Rule simply requires that all data should be presented in table form. This is the basis of Relational Model.
- Rule 2 Guaranteed Access Rule
- ↳ Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, Primary-key (row-value) and Attribute-name (column-name).
- Rule 3 Systematic treatment of null value
- ↳ The null value in a DB must be given a systematic and uniform treatment. This is a very important rule because a null can be interpreted as one of the following - the data is not present, data is missing, data is not known or data is not applicable.

- Rule 4 Active online catalog
- ↳ The detailed description of the entire DB must be stored in an online Catalog, known as metatab.

- data dictionary, which can be accessed by authorized users
But user can use the same query language to access the catalog which they use to access the db itself.

Rule 5 Comprehensive data sublanguage Rule

↳ A db can only be accessed using a language having linear syntax that supports data definition, data manipulation and transaction management operations.

↳ All commercial RDB use form of SQL or their supported language.

Rule 6 View updating Rule

↳ Data can be presented in different logical combination of called views.

↳ Each view should support a the same range of data manipulation that has direct access to a table available.

Rule 7 High level Insert, update and Delete.

↳ A db must support high level insertion, updation and deletion. This must not be limited to a single row, that is it must also support several concurrent and mass operation to yield sets of data records.

Rule 8 Physical data Independence

↳ The data stored in db must be independent of the application that Access the database.

↳ Any change in the Physical structure of a DB must not have any impact on "How the data is being accessed by application".

Rule 9 Logical Data Independence.

↳ Is the logical data in a DB must be independent of its user's view (App.). Any change in logical data must not affect the App. use it.

Rule 10 Integrity Independence.

↳ The DB language (like SQL) should support constraints on user input that maintain DB integrity.

↳ No component of a Primary key can have a null value.

↳ If a foreign key is defined in one table, any value in it must exist as with a Primary key in another table.

Rule 11 Distribution Independence.

↳ The end users must not be able to see that the database is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed DBS.

It leads to rules like 1. no single site with the highest load, high with low volatility and with all the data in a single site.

Rule 12 Non-subversion Rule

In these should be no way to modify the DB structure other than through the multiple row DB language (SQL). Most DB today support Adminstration tools that allows some direct manipulation of data structures.

* DBMS

↳ Relationship b/w two tables or files are maintained Programmatically.

↳ Relationship b/w two tables and files can be specified at the time of table creation.

↳ DBMS App. stores data in file

↳ RDBMS App. stores data in tabular form.

↳ Normalization is not present in: DBMS

↳ Normalization is present

↳ It does not support client/server Architecture.

↳ Most of it supports client/server Architecture.

↳ It does not apply any security with regards to data manipulation.

↳ RDBMS defines the integrity constraints for the purpose of ACID.

↳ It uses file system to store data, so there will be no relation b/w the tables.

↳ Data values are stored in the form of tables so a relationship b/w these data values will be stored in the form of a table as well.

- ↳ It does not support distributed database.
 - ↳ It is meant to be for small organization and deal with small data. It supports single user.
 - ↳ It may satisfy less than 7 to 8 Rules of Dr. E.F. Codd.
 - ↳ Ex:- File systems, XML, etc.
- DBMS**
- ↳ It is a collection of data. In DBMS user is not required to write their Procedures.
 - ↳ Due to centralized approach, data sharing is easy.
 - ↳ It gives an abstract view of data that hides the details.
 - ↳ Data Inconsistency does not exist.
 - ↳ It Support distributed database.
 - ↳ It is designed to handle large amount of data, it supports multiple users.
 - ↳ It satisfies all 12 rules of Dr. E.F. Codd.
- Ex:- MySQL, Oracle, SQL Server, PostgreSQL, etc.
- File Processing System.**
- ↳ It is a collection of data. In User has to write their Procedures for managing the DB.
 - ↳ Data is distributed in many files and it may be in different format, so it's not easy to share data.
 - ↳ It provides the details of the data representation and storage of data. (Centralized and distributed).
 - ↳ Data Inconsistency exists.

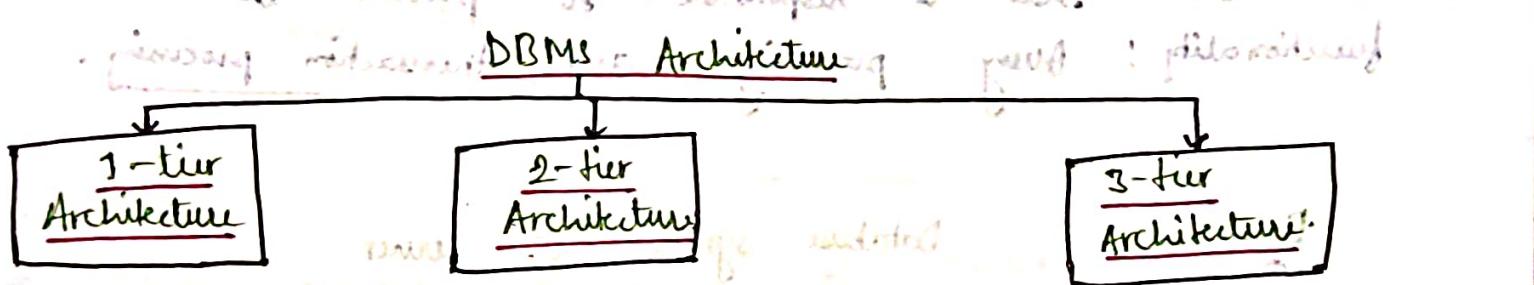
- ↳ Data Redundancy Problem is not found.
- ↳ database structure is very complex to design.
- ↳ Accessing db is easier
- ↳ Data independence exists and it can be of two types
 - (i) logical data Independence
 - (ii) Physical data Independence.

Agar logical level main kuch update karayे to uska effect physical level main hoga and vice versa.
- ↳ Integrity Constraints are easy to apply.
- ↳ In the db approach 3 types of data Model exists
 - Hierarchical data Model
 - Network data Model
 - Relational data Model
- ↳ Data Redundancy Problem is found.
- ↳ The file system has a simple structure.
- ↳ Accessing db is difficult.
- ↳ In file system approach there doesn't exist data Independence.
- ↳ Integrity Constraints are difficult to implement in File system.
- ↳ In the file system approach there is no concept of data Model exists.

Application Architecture of DBMS

- ↳ The DBMS design depends upon Architecture.
- ↳ The basic client / server Architecture is used to deal with a large no. of PCs, web servers, database servers and other components that are connected with networks.
- ↳ DBMS architecture depends upon how users are connected to the DB to get info. Their request done.

Type of DBMS.



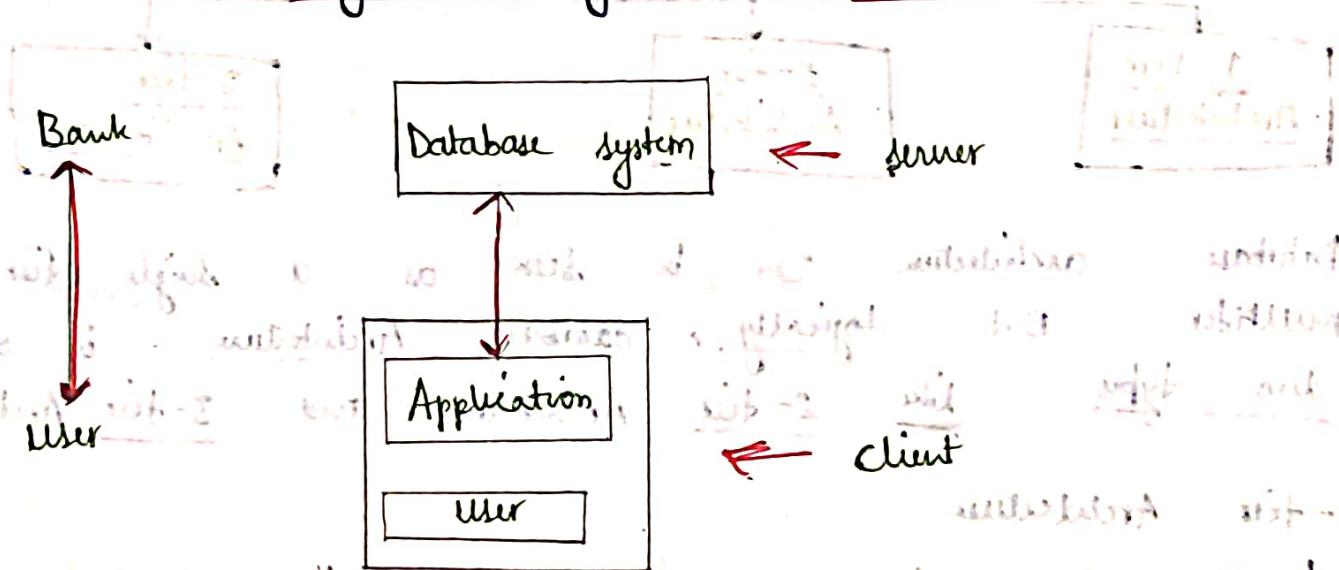
Database architecture can be seen as a single tier or multi-tier. But logically, database Architecture is of two types like 2-tier Architecture and 3-tier Architecture.

1-tier Architecture

- ↳ In this Architecture, the DB is directly available to the User. It means the user can directly sit on the DBMS and use it.
- ↳ Any changes done here will directly be done on the DB itself. It doesn't provide a handy tool for end users.
- ↳ The 1-tier Architecture is used for development of the local App, where programmers can directly communicate with the DB for the Quick response.

2-tier Architecture

- ↳ The 2-tier architecture is same as basic client/server.
In the two-tier architecture, app. on the client end can directly communicate with the DB at the server side. For this interaction, API's like ODBC, JDBC are used.
- ↳ The User Interface and application program are run on the client-side.
- ↳ The server side is responsible to provide the functionality: query processing and transaction processing.



3-tier Architecture

- ↳ The 3-tier architecture contains another layer b/w the client and server.
- ↳ In this Architecture, client can't directly communicate with the server.
- ↳ The app. on the client-end interacts with application server with db system. It further communicates with the DB system.

↳ The 3-tier architecture is used in case of large web application.

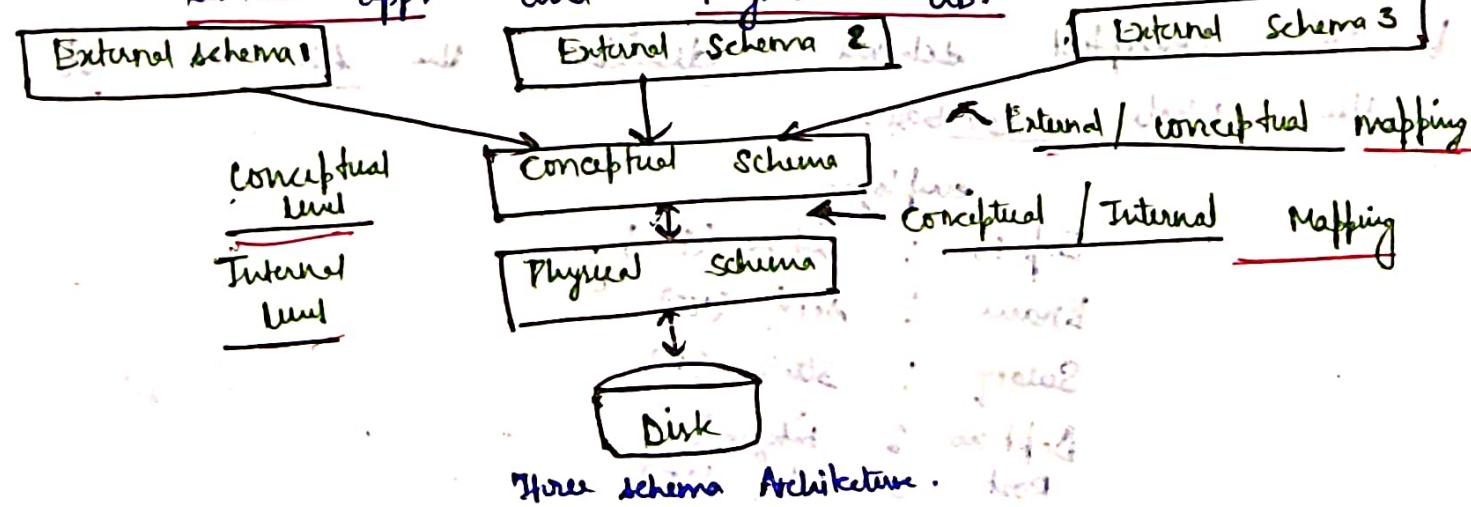


* Three schema Architecture of DBMS

↳ The overall design of the database is called the DB schema.

↳ The three schema Architecture is also called ANSI / SPARC (American National Standards Institute) (Standards for Planning and Requirements Committee) architecture or the three-level Architecture.

↳ The three schema Architecture is also used to separate the user app. and Physical db.



- 1 Internal level / Internal View
- ↳ The internal level has an static Internal Schema which describes the Physical storage of the db.

Internal view

Stored employee record length 60

Emp no. : 4 decimal offset 0 unique.

Ename : String length 15 offset 4.

Salary : 8, 2 decimal offset 19.

Dept no : 4 Decimal offset 27

Post : String length 15 offset 31

↳ The internal schema is also known as Physical schema.

↳ It is Physical data model. It is used to define that "how the data will be stored in a block".

↳ The Physical level is used to describe complex low-level data structure in detail.

- 2 conceptual level / logical level

↳ The conceptual schema describes the design of a database at the conceptual level.

↳ Conceptual level is also known as logical level.

↳ The conceptual schema describes the structure of the whole database.

employee

Emp no. : integer (4) key.

Ename : string (15)

Salary : String (8)

Dept no : Integer (4)

Post : String (15)