History of Databases

In 1960, Charles W. Bachman designed the integrated database system, the "first" DBMS.

- 1960s: Introduction of hierarchical and network models (e.g., IBM's IMS).
- 1970s: Emergence of the relational model proposed by Edgar F. Codd; creation of SQL.
- 1980s: Development of relational database management systems (RDBMS), e.g., Oracle, IBM DB2.
- 1990s: Rise of object-oriented databases and the beginning of web-based databases.
- 2000s - Present: Growth of NoSQL databases (e.g., MongoDB, Cassandra) to handle big data and unstructured data; cloud databases and distributed databases.

Database Management Systems (DBMS)

A database is a collection of related data stored in a standard format, designed to be shared by multiple users. A database is defined as "A collection of interrelated data items that can be processed by one or more application programs".

Example: A Bank, a Hospital, a University, a Manufacturing company

Data

Data is the raw material from which useful information is derived. The word data is the plural of Datum. Data is commonly used in both singular and plural forms. It is defined as raw facts or observations. It takes variety of forms, including numeric data, text and voice and images.

Data is a collection of facts, which is unorganized but can be made organized into useful information. The term Data and Information come across in our daily life and are often

interchanged.

Example: Weights, prices, costs, number of items sold etc.

Information

Data that have been processed in such a way as to increase the knowledge of the person who uses the data. The term data and information are closely related. Data are raw material resources that are processed into finished information products. The information as data that has been processed in such way that it can increase the knowledge of the person who uses it. In practice, the database today may contain either data or information.

Database System Applications

Databases are widely used. Here are some representative applications:

1. Banking: For customer information, accounts, and loans, and banking transactions.

2. Airlines: For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner - terminals situated around the world accessed the central database system through phone lines and other data networks.

3. Universities: For student information, course registrations, and grades.

4. Credit card transactions: For purchases on credit cards and generation of monthly statements.

5. Telecommunication: For keeping records of calls made, generating monthly bills,

maintaining balances on prepaid calling cards, and storing information about the communication networks.

6. Finance: For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds.

7. Sales: For customer, product, and purchase information.

8. Manufacturing: For management of supply chain and for tracking production of items in factories, inventories of items in warehouses / stores, and orders for items.

9. Human resources: For information about employees, salaries, payroll taxes and benefits, and for generation of paychecks.

FILE SYSTEM VS DBMS

File System	**DBMS**
1.The file system is a way of arranging the files in a storage medium within a computer.	DBMS is software for managing the database.
2.Redundant data can be present in a file system.	In DBMS there is no redundant data.

File System	DBMS
3.It doesn’t provide Inbuilt mechanism for backup and recovery of data if it is lost.	It provides tools for backup and recovery of data even if it is lost.
4.There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
5.There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
6.It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
7.File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
8. Only one user can access data at a time.	Multiple users can access data at a time.
9.It give details of storage and representation of data	It hides the internal details of Database
10.To access data in a file , user requires attributes such as file name, file location.	No such attributes are required.

File System	DBMS
11. Cobol, C++, XML	Oracle, SQL Server

Relational model:

➢ A relation, basically a table with rows and columns.

➢ Every relation has a schema, which describes the columns, or fields.

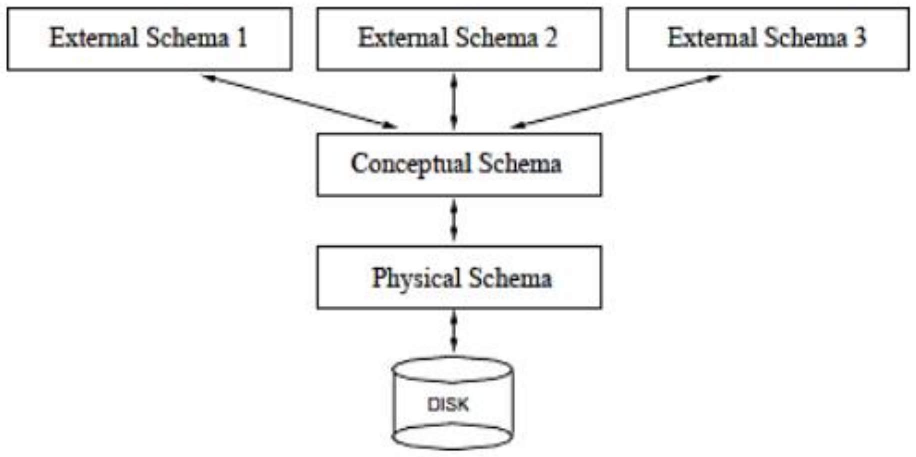
➢ Student information in a university database may be stored in a relation with the following

schema

➢ Students (sid: string, name: string, address: string, age: integer)

Levels of Abstraction in a DBMS (Three tier schema architecture)

- ➢ The data in a DBMS is described at three levels of abstraction.
- ➢ The database description consists of a schema at each of these three levels of abstraction.
- ➢ External, Conceptual and Physical
- ➢ Views describe how users see the data.
- ➢ Conceptual schema defines logical structure.
- ➢ Physical schema describes the files and indexes used.



External Schema:

- This schema allows data access to be customized at the level of individual users or groups of users.
- A database has exactly one conceptual schema and one physical schema, but it may have

several external schemas.

- An external schema is a collection of one or more views and relations from the conceptual

schema.

- A view is conceptually a relation, but the records in a view are not stored in the DBMS.

Conceptual schema:

- The conceptual schema(also called as logical schema) describes the stored data in terms of the data model of the DBMS.

- In a relational DBMS, the conceptual schema describes all relations that are stored in the database.
- In our sample university database, these relations contain information about entities, such as students and faculty, and about relationships, such as students' enrollment in courses.

Students(sid: string, name: string, username: string, age: integer, gpa:real)

Faculty(fid: string, fname: string, salary : real)

Courses(cid: string, cname: string, credits: integer)

Rooms(nw: integer, address: string, capacity: integer)

Enrolled (sid: string, cid: string, grade: string)

Teaches (fid: string, cid: string)

Physical Schema:

- The physical schema specifies storage details.
- It summarizes how the relations described in the conceptual schema are actually stored on secondary storage devices such as disks and tapes.
- Decides what file organizations to use to store the relations and create auxiliary data structures, called indexes, to speed up data retrieval operations.

▪ A sample physical schema for the university database is to store all relations as unsorted files of records.

o Create indexes on the first column of the students, faculty and courses relations.

Database Architecture

Database architecture uses programming languages to design a particular type of software for businesses or organizations. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions.

The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows:

1-tier architecture

2-tier architecture

3-tier architecture

1-tier architecture:

In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.

Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.

The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-tier architecture:

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server.

The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: ODBC, JDBC are used.

The user interfaces and application programs are run on the client-side.

The server side is responsible to provide the functionalities like: query processing and transaction management.

To communicate with the DBMS, client-side application establishes a connection with the server side.

3-tier architecture:

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.

The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.

The application on the client-end interacts with an application server which further communicates with the database system.

End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.

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

Q) How to create database?

Syntax: create database<database_name>

Ex-   create database cimage

Q) How to select database?

Syntax: use <database name>

Ex- use cimage

Q) How to delete database?

Syntax: drop database <database name>

Ex- drop database cimage

Q) How to rename database?

Syntax: sp_renamedb <old database> <new database name>

Ex- sp_renamedb cimage cimage1

Q) How to display current database?

Syntax: select db_name()

Ex- select db_name()

Q) How to create table?

Syntax- create table <table name>(column name <data type>(size) constraint name,

```
                              column name <data type>(size) constraint name,
                              ………………);
```

Ex- create table customer(

```
            ID int primary key,
            cname varchar(30) not null,
            address varchar(100) null,
            city varchar(30) default 'Patna',
            email varchar(55) unique,
            age int check(age>18)
);
```

Q) How to display the structure of table?

Syntax: Sp_help <table name>

Ex-   Sp_help student

Q) How to delete table?

Syntax: drop table <table name>

Ex- drop table student

- ⇨ Rename table name

  Ex- sp_rename student , new_student