

Introduction to DBMS

DBMS Stands for Data Base Management System

- It is the collection of interrelated data (Which is called as Database)
- It contains set of software tools/programs which access the data and also processes the data (this part if also called as QUERY & UPDATE mechanism)

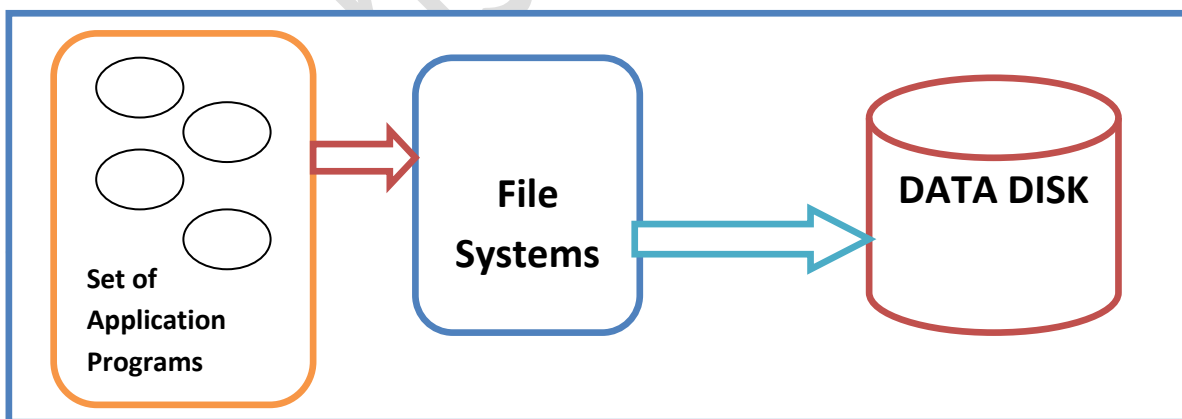
Formally, "database" refers to the data themselves and supporting data structures. Databases are created to operate large quantities of information by inputting, storing, retrieving and managing that information. Databases are set up so that one set of software programs provides all users with access to all the data.

A "database management system" is a suite of computer software providing the interface between users and a database or databases. Because they are so closely related, the term "database" when used casually often refers to both a DBMS and the data it manipulates.

Examples of Information Systems which require a Database:

- Banking System
- Library Management System
- Railway Reservation System
- Ecommerce

Early Information System: -



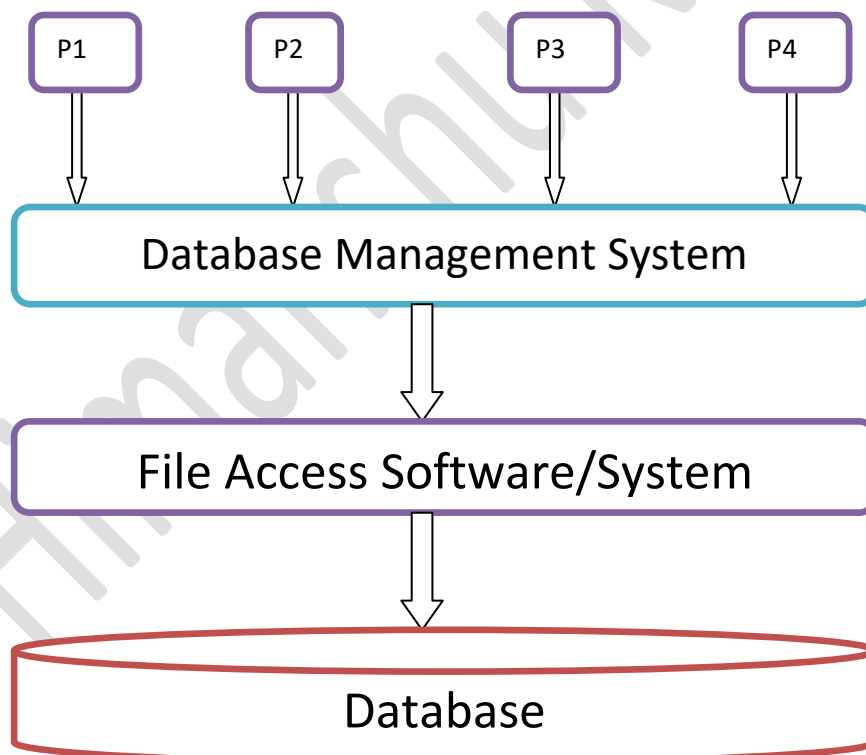
Problems with Early Information Systems

- **Disorganized Development**

- ✓ **Data Isolation:** multiple transactions occurring at the same time not impact each other's execution. For example, if Person1 issues a transaction against a database at the same time that Person2 issues a different transaction, both transactions should operate on the database in an isolated manner. The database should either perform Person1's entire transaction before executing Person2's or vice-versa. This prevents Person1's transaction from reading intermediate data produced as a side effect of part of Person2's transaction that will not eventually be committed to the database. Note that the isolation property does not ensure which transaction will execute first, merely that they will not interfere with each other.
- ✓ **Data Redundancy & Inconsistency:** Data redundancy occurs in database systems which have a field that is repeated in two or more tables. For instance, when customer data are duplicated and attached with each product bought, then redundancy of data is a known source of inconsistency since customer might appear with different values for given attribute. Data redundancy leads to data anomalies and corruption and generally should be avoided by design
- **Concurrency:** - When multiple users attempt to modify data at the same time, controls need to be established in order to prevent one user's modifications from adversely affecting modifications from simultaneous users. The system of handling what happens in this situation is called concurrency control.
- **Security:** Database security concerns the use of a broad range of information security controls to protect databases against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/administrative and physical. Database security is a specialist topic within the broader realms of computer security, information security and risk management.
 - Security risks to database systems include, for example:
 - ✓ Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations);
 - ✓ Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
 - ✓ Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended;
 - ✓ Physical damage to database servers caused by computer room fires or floods, overheating, lightning, accidental liquid spills, static discharge, electronic breakdowns/equipment failures and obsolescence;

- ✓ Design flaws and programming bugs in databases and the associated programs and systems, creating various security vulnerabilities (e.g. unauthorized privilege escalation), data loss/corruption, performance degradation etc.;
 - ✓ Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.
- **Integrity constraints:** refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle. The overall intent of any data integrity technique is the same: ensure data is recorded exactly as intended (such as a database correctly rejecting mutually exclusive possibilities,) and upon later retrieval, ensure the data is the same as when it was originally recorded. In short, data integrity aims to prevent unintentional changes to information. Data integrity is not to be confused with data security, the discipline of protecting data from unauthorized parties. Data integrity can be roughly divided into two overlapping categories:
 - ✓ Physical Integrity
 - ✓ Logical Integrity

Role of Database Management System



Overall Organization :- (Data Abstraction)

- Physical Level
- Conceptual Level (for Database Administrator)
- View Level (for Database Application Users)
- User Level (for Naïve users: Mostly Menu Driven)

Schema



A database schema of a database system is its structure described in a formal language supported by the database management system (DBMS) and refers to the organization of data as a blueprint of how a database is constructed (divided into database tables in the case of Relational Databases).

The formal definition of database schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real world entities are modeled in the database.

Database schema can be divided broadly in two categories:

1. **Physical Database Schema:** This schema pertains to the actual storage of data and its form of storage like files, indices etc. It defines the how data will be stored in secondary storage etc.
2. **Logical Database Schema:** This defines all logical constraints that need to be applied on data stored. It defines tables, views and integrity constraints etc.

Analogy

	Real World	Database																
Scheme	 Plan for a Standard-House	<table border="1"> <thead> <tr> <th>P-ID</th> <th>Name</th> <th>Prenome</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> Template for a Table	P-ID	Name	Prenome													
P-ID	Name	Prenome																
Instances	 Built Standard-Houses	<table border="1"> <thead> <tr> <th>P-ID</th> <th>Name</th> <th>Prenome</th> <th>Prenome</th> </tr> </thead> <tbody> <tr> <td>102356</td> <td>Smith</td> <td>John</td> <td>Prenome</td> </tr> <tr> <td>102357</td> <td>Potter</td> <td>Harry</td> <td>William</td> </tr> <tr> <td>523646</td> <td>Wood</td> <td>Lucinda</td> <td></td> </tr> </tbody> </table> Data-filled Tables	P-ID	Name	Prenome	Prenome	102356	Smith	John	Prenome	102357	Potter	Harry	William	523646	Wood	Lucinda	
P-ID	Name	Prenome	Prenome															
102356	Smith	John	Prenome															
102357	Potter	Harry	William															
523646	Wood	Lucinda																

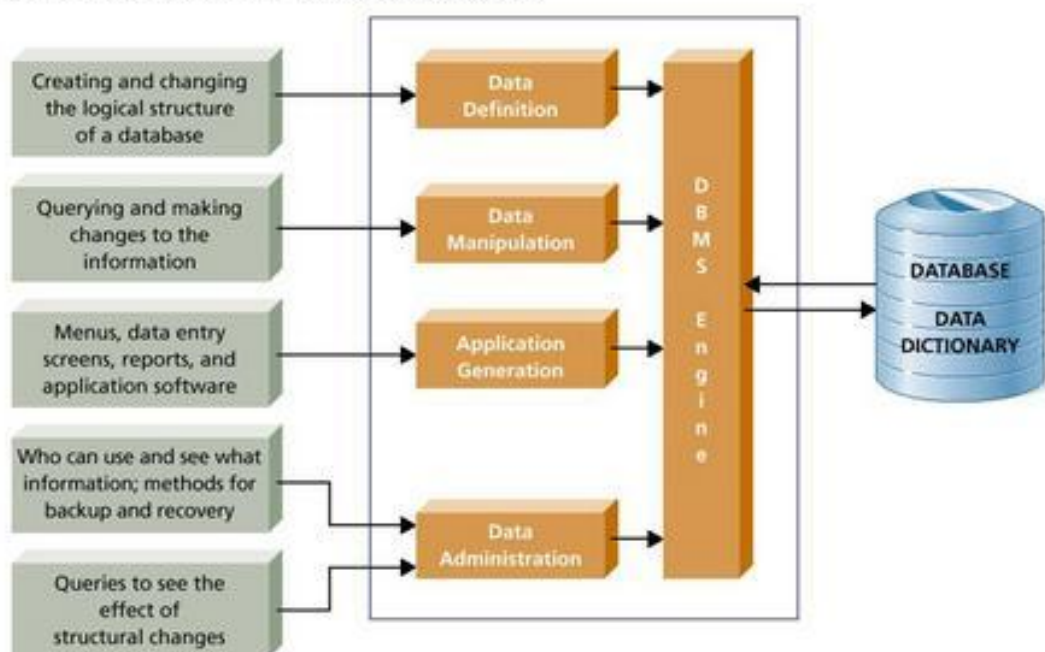
Instance

It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when database doesn't exist at all and very hard to do any changes once the database is operational. Database schema does not contain any data or information.

Database instances, is a state of operational database with data at any given time. This is a snapshot of database. Database instances tend to change with time. DBMS ensures that its every instance (state) must be a valid state by keeping up to all validation, constraints and condition that database designers has imposed or it is expected from DBMS itself.

Components of a Database

Software Subsystems of a Database Management System



Database Engine:

The Database Engine is the core service for storing, processing, and securing data. The Database Engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications within your enterprise. Use the Database Engine to create relational databases for online transaction processing or online analytical processing data. This includes creating tables for storing data, and database objects such as indexes, views, and stored procedures for viewing, managing, and securing data. You can use SQL Server Management Studio to manage the database objects, and SQL Server Profiler for capturing server events.

Data dictionary:

A data dictionary is a reserved space within a database which is used to store information about the database itself. A data dictionary is a set of table and views which can only be read and never altered.

Most data dictionaries contain different information about the data used in the enterprise. In terms of the database representation of the data, the data table defines all schema objects including views, tables, clusters, indexes, sequences, synonyms, procedures, packages, functions, triggers and many more. This will ensure that all these things follow one standard defined in the dictionary. The data dictionary also defines how much space has been allocated for and / or currently in used by all the schema objects. A data dictionary is used when finding information about users, objects, schema and storage structures. Every time a data definition language (DDL) statement is issued, the data dictionary becomes modified

A data dictionary may contain information such as:

- ✓ Database design information
- ✓ Stored SQL procedures
- ✓ User permissions
- ✓ User statistics
- ✓ Database process information
- ✓ Database growth statistics
- ✓ Database performance statistics

Query Processor:

A relational database consists of many parts, but at its heart are two major components: the storage engine and the query processor. The storage engine writes data to and reads data from the disk. It manages records, controls concurrency, and maintains log files. The query processor accepts SQL syntax, selects a plan for executing the syntax, and then executes the chosen plan. The user or program interacts with the query processor, and the query processor in turn interacts with the storage engine. The query processor isolates the user from the details of execution: The user specifies the result, and the query processor determines how this result is obtained.

The query processor components include

- ✓ DDL interpreter
- ✓ DML compiler
- ✓ Query evaluation engine

Report writer:

Also called a report generator, a program, usually part of a database management system, that extracts information from one or more files and presents the information in a specified format. Most report writers allow you to select records that meet certain conditions and to display selected fields in rows and columns. You can also format data into pie charts, bar charts, and other diagrams. Once you have created a format for a report, you can save the format specifications in a file and continue reusing it for new data.