**DATABASE MANAGEMENT SYSTEM**

* **What is database management system?**

A database management system (DBMS) is a software package with computer programs that control the creation, maintenance, and use of a database. It allows organizations to conveniently develop databases for various applications by database administrators (DBA’s) and other specialists. A database is an integrated collection of data records, files and other objects. A DBMS allows different user application programs too concurrently to access the same database. DBMSs may use a variety of database models, such as the relational model or object model, to conveniently describe and support applications. It typically supports query languages which are in fact high level programming languages, dedicated database languages that considerably simplify writing database application programs. A DBMS provides facilities for controlling data access, enforcing data integrity, managing concurrency control, and recovering the database after failures and restoring it from backup files, as well as maintaining database security.

**DEFINITION:**

A database management system (DBMS) is the software that allows a computer to perform database functions storing, retrieving, and adding, deleting, and modifying data. Relational database management system (RDBMS) implements the relational models of tables and relationships.

**EXAMPLES:**

Microsoft access, My SQL, Microsoft SQL, server, oracle and file maker pro are all examples of database management

**Objective of DBMS**

Database management system is very important. The main objectives of DBMS are:

* Understand why database are important to modern organizations.
* Understand how dataset work
* Understand how organizations can maximize their strategies potential with database.
* Keep track of things
* Store information that is more complicated than a simple list.

**Function of DBMS**

A DBMS makes it possible for users to create, edit and update data in database files. Once created, the DBMS makes it possible to store and retrieve data from those database files.

More specifically, a DBMS provides the following functions:

* Create database
* Create tables
* Create supporting structures
* Read database data
* Modify database data(insert, update, delete)
* Maintain database structure
* Enforce rules
* Control concurrency
* Provide security
* Perform backup and recovery

**Types of Database Management systems**

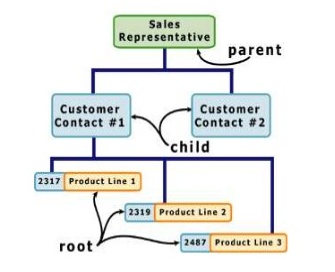
There are four structural types of DBMS:

* Hierarchical databases.
* Network databases.
* Relational databases.
* Object-oriented databases

**Hierarchical Databases (DBMS):**   
In the Hierarchical Database Model we have to learn about the databases. It is very fast and simple. In a hierarchical database, records contain information about their groups of parent/child relationships, just like as a tree structure. The structure implies that a record can have also repeating information. In this structure Data follows a series of records; it is a set of field values attached to it. It collects all records together as a record type. These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows. To create links between these record types, the hierarchical model uses these type Relationships.

**Structure Of A DBMS**

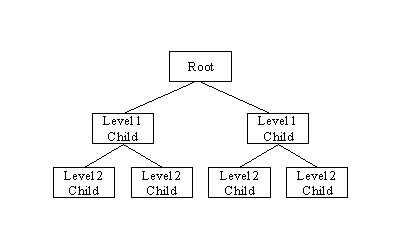
* A typical DBMS has a layered architecture.
* The figure does not show the concurrency control and recovery components.
* This is one of several possible architectures; each system has its own variation.



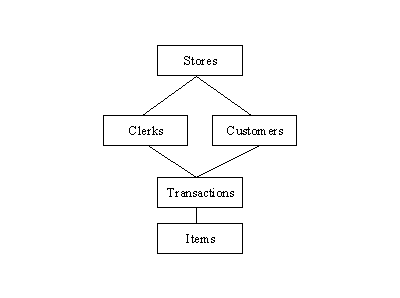
**Figure: Hierarchical database.**

**File System**

1. Data redundancy and inconistancy
2. Difficulty in accessing data
3. Data isolation
4. Integrity problems
5. Atomicity Problems
6. Con-Current access anomalies
7. Secuirity problems

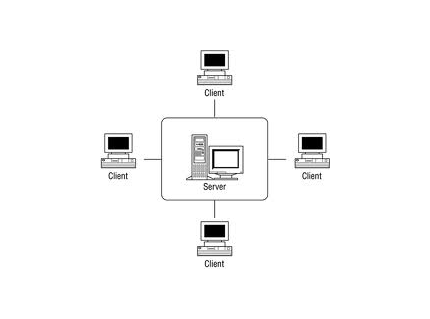


**Network Database:** A network databases are mainly used on large digital computers. It more connections can be made between different types of data, network databases are considered more efficiency it contains limitations must be considered when we have to use this kind of database. It is Similar to the hierarchical databases; network databases .**Network databases** are similar to hierarchical databases by also having a hierarchical structure. A network database looks more like a cobweb or interconnected network of records. In network databases, children are called members and parents are called occupier. The difference between each child and member can have more than one parent.



The approval of the network data model is similar with the esteem of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. The network model authorized the modeling of many-to-many relationships in data.  
**The network model** is very similar to **the hierarchical model** really. Actually the hierarchical model is a subset of the network model. However, instead of using a single-parent tree hierarchy, the network model uses set theory to provide a tree-like hierarchy with the exception that child tables were allowed to have more than one parent. It supports many-to-many relationships.

**Rational Database:**  
In relational databases, the relationship between data files is relational. Hierarchical and network databases require the user to pass a hierarchy in order to access needed data. These databases connect to the data in different files by using common data numbers or a key field. Data in relational databases is stored in different access control tables, each having a key field that mainly identifies each row. In the relational databases are more reliable than either the hierarchical or network database structures. In relational databases, tables or files filled up with data are called relations (tuples) designates a row or record, and columns are referred to as attributes or fields. Relational databases work on each table has a key field that uniquely indicates each row, and that these key fields can be used to connect one table of data to another.



**The relational database has two major reasons:**

1. Relational databases can be used with little or no training.
2. Database entries can be modified without specify the entire body.

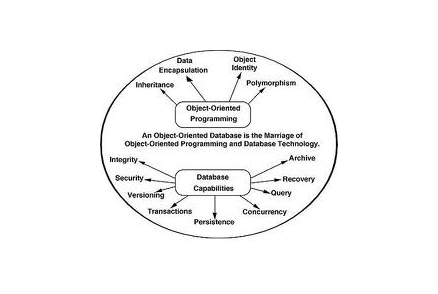
**Properties of Relational Tables:**

In the relational database we have to follow some properties which are given below.

* It’s Values are Atomic
* In Each Row is alone.
* Column Values are of the same thing.
* Columns are undistinguished.
* Sequence of Rows is Insignificant.
* Each Column has a common Name.

**Object- Oriented Model:**  
In this Model we have to discuss the functionality of the object oriented Programming .It takes more than storage of programming language objects. Object DBMS's increase the semantics of the C+ and Java .It provides full-featured database programming capability, while containing native language compatibility. It adds the database functionality to object programming languages. This approach is the analogical of the application and database development into a constant data model and language environment. Applications require less code, use more natural data modeling, and code bases are easier to maintain. Object developers can write complete database applications with a decent amount of additional effort.

The object-oriented database derivation is the integrity of object-oriented programming language systems and consistent systems. The power of the object-oriented databases comes from the cyclical treatment of both consistent data, as found in databases, and transient data, as found in executing programs.

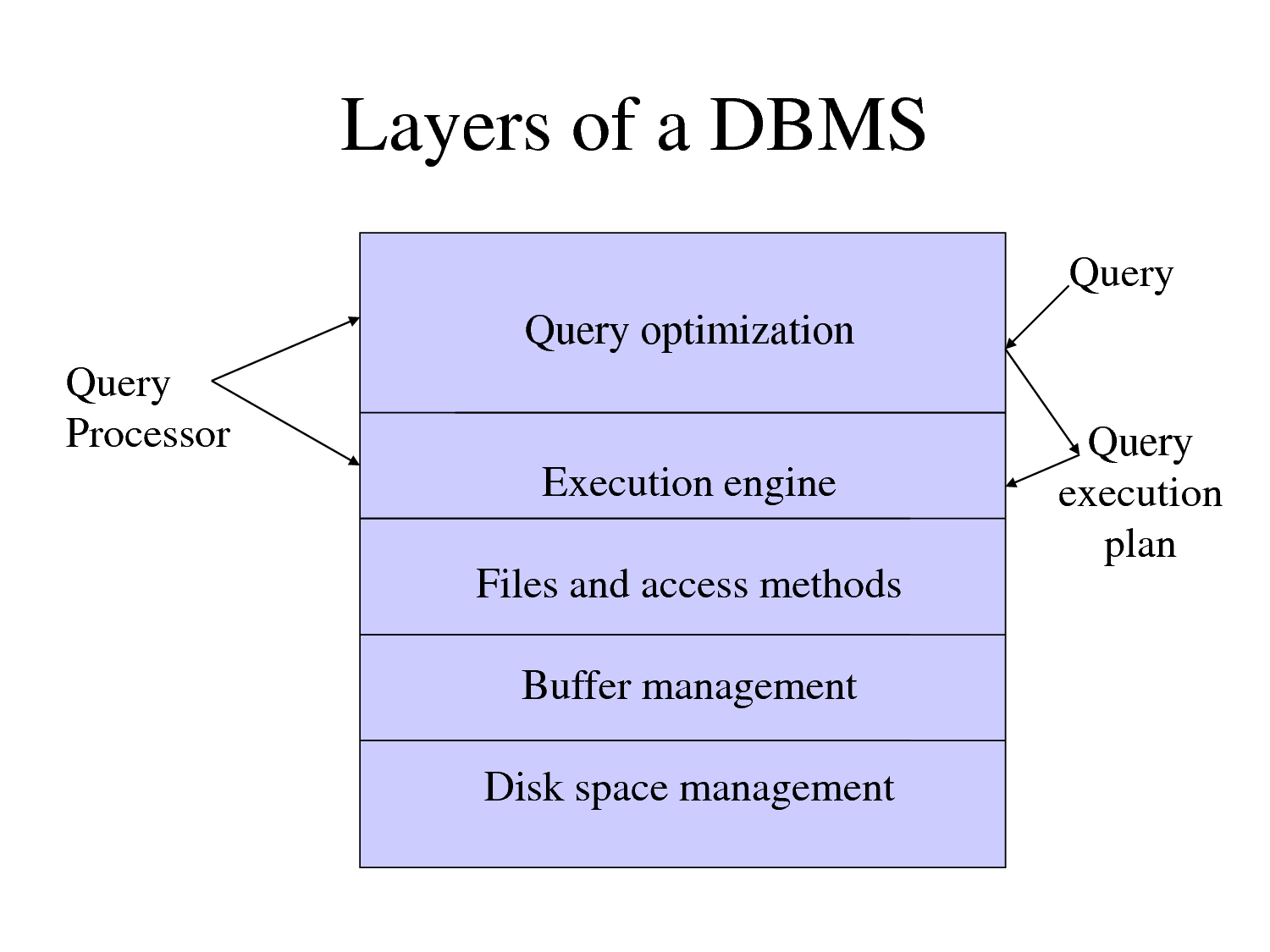


**Figure: object- oriented model**  
Object-oriented databases use small, recyclable separated of software called objects. The objects themselves are stored in the object-oriented database. Each object contains of two elements:

1. Piece of data (e.g., sound, video, text, or graphics).
2. Instructions or software programs called methods, for what to do with the data

**The Layers of Database Management Systems**

We have shown different types of aspects of DBMS. The layers of DBMS are also important for overall DBMS system. Including query processor and query execution plan, we can show the layers of DBMS:



**Data Languages**

The database provides a data definition language to specify the database schema and a data Manipulation language to express database queries and updates.

* **Data Definition Language**
* **Data Manipulation Language**

**Data-Definition Language (DDL)**

We specify a database schema by a set of definition expressed by a special language called a Data-Definition Language.

For instance the following statement in the SQL language defines the account table:

e. g. **Create table** account (account- number

**Chart (10)** balance **integer**)

Execution of the above DDL statement create the account table .in addition it updates a special set of tables called the DATA –DICTIONARY or DATA- DIRECTORY

**Data-Manipulation Language**

A data manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model. There are basically two types

* **Procedural DMLs**
* **Declarative DMLs**

The query in the SQL language finds the name of the customer whose customer –id is 1999

e.g. **Select** customer .customer –name **from** customer

**Where** customer-id=1999

**DATA MANIPULATION** is –

* The retrieval of the information stored in the database
* The insertion of new information into the database
* The deletion of information from the database
* The modification of information stored in the database

**TYPES OF DATABASE**

**Databases** can be classified according to:

* Number of users
* Database location(s)
* Expected type and extent of use

**Single user database:** supports only one user at a time.

* Desktop database: Single user; runs on pc

**Multiuser database:** supports multi user at a same time

* **Workgroup** and **Enterprise** databases

**Centralized database:** data located at a single site

**Distributed database:** data distributed across several different sites

**Operational database:** supports a company’s day to day operations

* Transactional or production database

**Data warehouse:** stores data for tactical or strategic decisions

**Unstructured data:** exist in their original state

**Structured data:** result from formatting

* Structure applied based on type of processing to be preformed

**Semi structured data:** have been processed to some extent

**Extensible Markup Language (XML):** represents data elements in textual formats

* **XML database** supports semi structured XML data

**Database Design process**

There are six stages in the design of a database:

1. Requirement analysis
2. Conceptual database design
3. Choice of the DBMS
4. Data model mapping
5. Physical design
6. Implementation

**Requirement Collection and Analysis:**

**Purpose:** to document the data requirements of the users. **Functional requirements** are the operations that will be applied to the database, including queries and update.

**Typical activities**:

* Identification of application areas and user groups.
* Analysis of existing documentation of application areas, e.g. policy documents, forms, reports, organization charts.
* Responses to user questionnaires are analyzed.

**Conceptual Database Design:** Two parallel activities:

1. **Schema design**

* Examines the data requirements of the database resulting from the analysis and produces a conceptual schema in DBMS-independent high level data model.

1. **Transaction design**

* Examines the database applications whose requirement were analyzed and produces high level specification for these transactions.jhe\/7

**Choice of a DBMS:**

**Purpose:** establish which the best framework for implementing the produced schema is:

* Type of DBMS (relational, network, deductive, object oriented)
* User and programmer interfaces.
* Types of query languages.
* Choice made on the basis of technical factors
* The DBMS has to support the required tasks.

**Data Model Mapping:**

**Purpose**: to transform the generic, DBMS independent conceptual schema in the data model of the chosen DBMS.

**Two stages:**

1. **System independent mapping**: no consideration of any specific characteristics that may apply to the specific DBMS package.
2. **Tailoring to DBMS**: different DBMSs may implement the same data model in slightly different ways.

**Physical Design:**

**Purpose**: to choose the specific storage structures and access paths for the database files.

* **Some relevant criteria :**
* **Response time**: may want to minimize database access time for data items referenced by frequently used transactions.
* **Space utilization**: less frequently used data and queries may be archived.

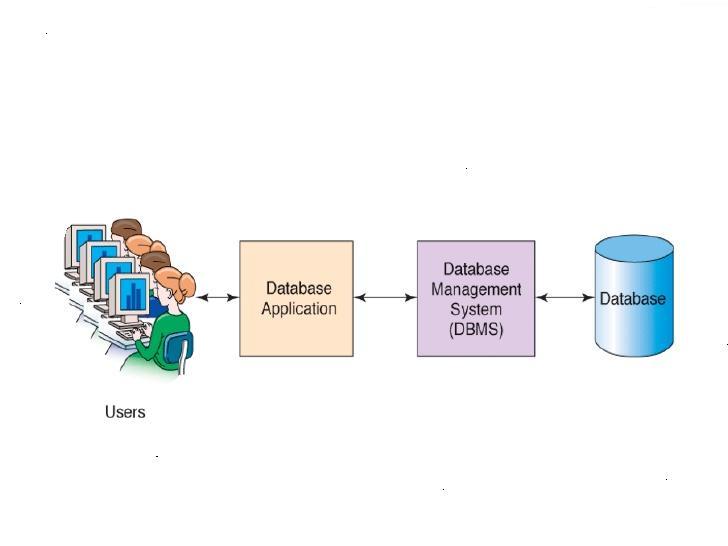
**Implementation:**

**Purpose:** Compile the database

* Populate the database
* Manually/ automatically (may need to convert data from a previous format.)
* Operational phase may begin

**Application of database management systems**

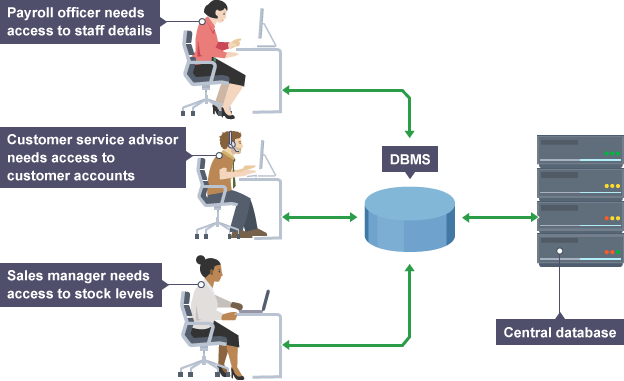
It is a set of one or more computer programs that serves as an intermediary between the users and DMBS. Application program that read or modify database data is sending by SQL statements to DMBS. Application program to present data to users is in the format of forms and reports.



**Figure: (System of DBMS Process)**

**Sectors of Applications:**

* **BANKING:** for customer, information, account, loans and transactions
* **AIRLINE:** for reservation and schedule information. Airlines were among first to use database in a geographically distributed manner terminals situated around the world accessed the central database system through phone lines and other data network.
* **UNIVERSITIES:** for students in formations, course registration and grades.



**Figure: DBMS could be viewed by lots of different people with different responsibilities.**

* **CREDITS AND TRANSECTIONS:** for purchases on credit cards and generation of monthly statements.
* **Real estate agency:** to share information on the properties.
* **SALES:** for customer product and purchase information.
* **HUMAN RESOURCES:** for information about employees, salaries, payroll taxes and benefits, and for generations of paybacks.
* **Manufacturing:** to keep track of supplies of component of or raw materials.
* **Marketing:** to track records of advertisers and prospective advertising outlets.
* **Shipping:** to keep track of orders and shipments.

**Functions of Database Application;**

* Create and process forms
* Process user queries
* Create and process reports
* Execute application logic
* Control database application

**The Database System Environment:**

Database system environment is very important to understand the modern organizations, understand how database works, and understand how organizations can maximize their strategies potential with database. The components of DBMS environment ensure an overall arrangement to the database management system.

**Database systems;** defines and regulates the collections, storage, management, use of data.

There are five major parts:

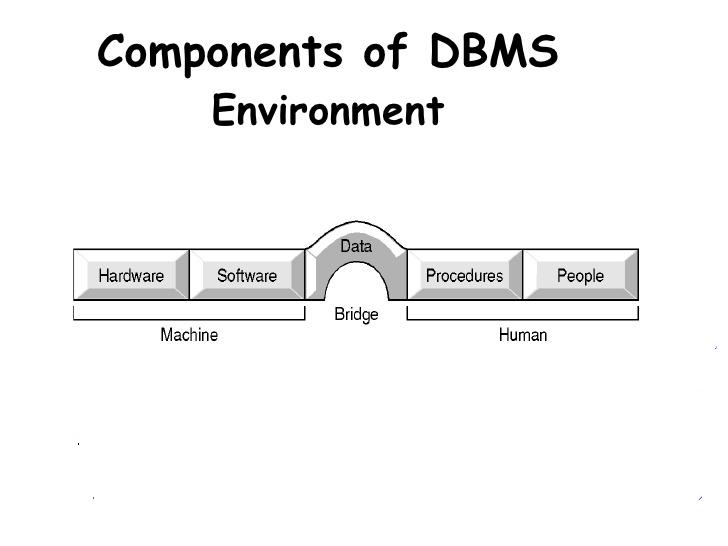
-hardware

-software

-people

-procedures

-data

**Hardware:** all the system’s physical devices.

**Software:** three types of software required:

-opening system software

- Software

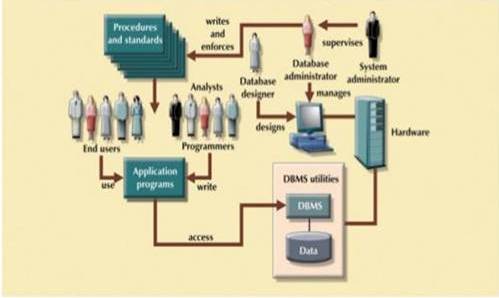
-application programs and utility software

**People:** all users of the database systems

* Systems and database administrators
* Database designers
* System analysis and programmers
* And users

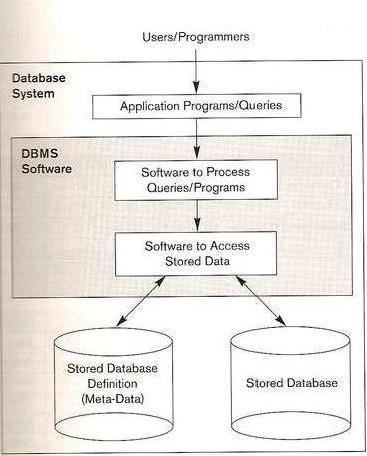
**Procedures:** instruction and rules that govern the design and use of the database system

**Data:** the collection of facts stored in the database



**Figure: The Database System Environment.**

Database systems are created and managed at different levels of complexity. Database solutions must be cost effective well as tactically and strategically effective. Database technology already in use affects selection of a database system.



**Figure: The Simplified Database System Environment**

**Eloquence DBMS Organization**

The following sections describe the structure and access methods employed by Eloquence DBMS.

**Eloquence DBMS Logical Structure**

Eloquence DBMS is organized into three sections: **database definition, database manipulation,** and **database maintenance.**

**Database definition** is accomplished using an editing program (for example, VI on UNIX, Notepad on Windows) and the schema program. These programs are used in conjunction with the database definition language (DBDL) to define the structure, size and security of a database.

**Database access and manipulation** is performed using the database manipulation statements. These statements are invoked from Eloquence programs, and serve as an interface between databases and application programs. The manipulation statements handle database access and structural housekeeping.

**Database maintenance** operations are performed using the database utilities. These utilities provide the capability to create, purge, and erase data sets. You can also report on the structure of the database and, if desired, restructure the database.

**Database Organization**

There are three basic structures within an Eloquence database: **data items, data entries,** and **data****sets.**

**A data item** is the smallest data element. Each data item has a value and is referenced by a data item name. Data items correspond to program variables within an applications program.

|  |  |
| --- | --- |
| Some examples are: | |
| **Data Item Name** | **Data Item Values** |
| PRODUCT-NO | 50 |
|  | 100 |
|  | 1000 |
|  |  |
| PRODUCT-DESC | Tricycle |
|  | Standard Bicycle |
|  | 10-Speed Bicycle |

**A data entry,** or record, is an ordered collection of related data items. All data is transferred to and from a database on a record basis.

|  |  |  |
| --- | --- | --- |
| For example: | | |
| **Data-Entry Definition** | **PRODUCT-NO** | **PRODUCT-DESC** |
| Data Entry Values | 50 | Tricycle |
|  | 100 | Standard Bicycle |
|  | 1000 | 10-Speed Bicycle |

**A data set** is a collection of data entries sharing a common definition. All entries in a data set are stored as a separate file in a directory and are referred to by a data set name. Some examples are shown below:

**Data Set Name:**

PRODUCT

**Data Entry Definition:**

PRODUCT-NO

PROD-DESC

|  |  |  |
| --- | --- | --- |
| For examples | | |
| *Entry* |  |  |
| *Record No.* |  |  |
| 1 | 50 | Tricycle |
|  |  |  |
| 2 | 1000 | 10-Speed Bicycle |
|  |  |  |
| 3 | 100 | Standard Bicycle |

**Types of Data Sets**

There are two types of data sets in the Eloquence DBMS: **master data sets** and **detail data** **sets.**

**Detail data sets** are used to store "line item" information. **Master data sets** are generally used as indexes to information within detail data sets. For example, the CUSTOMER detail data set shown below contains information about a customer order, such as the customer's name and the product purchased.

**Data entries** contain pointer information used to link related entries. **Detail entries** contain pointers to other entries containing the same search item value. This linkage of related detail entries is known as a *data chain*. **Master entries** contain pointers to the beginning and end of data chains, along with the number of entries within the chain. This chain information is automatically maintained by the Eloquence DBMS.

**Database users**

Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

1. **Application Programmers -** They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal etc. These queries are converted into object code to communicate with the database. For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the C Program.
2. **Sophisticated Users -** They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirement. In short, we can say this category includes designers and developers of DBMS and SQL.
3. **Specialized Users -** These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.
4. **Stand-alone Users -** These users will have stand –alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.
5. **Native Users -** these are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc which has existing application and users use them to interact with the database to fulfill their requests.

**Database Administrator**

The life cycle of database starts from designing, implementing to administration of it. A database for any kind of requirement needs to be designed perfectly so that it should work without any issues. Once all the design is complete, it needs to be installed. Once this step is complete, users start using the database. The database grows as the data grows in the database. When the database becomes huge, its performance comes down. Also accessing the data from the database becomes challenge. There will be unused memory in database, making the memory inevitably huge. These administration and maintenance of database is taken care by database Administrator – DBA.  
A DBA has many responsibilities. A good performing database is in the hands of DBA.

**Installing and upgrading the DBMS Servers: -** DBA is responsible for installing a new DBMS server for the new projects. He is also responsible for upgrading these servers as there are new versions comes in the market or requirement. If there is any failure in up gradation of the existing servers, he should be able revert the new changes back to the older version, thus maintaining the DBMS working. He is also responsible for updating the service packs/ hot fixes/ patches to the DBMS servers.

**Design and implementation: -** Designing the database and implementing is also DBA’s responsibility. He should be able to decide proper memory management, file organizations, error handling, log maintenance etc for the database.

**Performance tuning: -** Since database is huge and it will have lots of tables, data, constraints and indices, there will be variations in the performance from time to time. Also, because of some designing issues or data growth, the database will not work as expected. It is responsibility of the DBA to tune the database performance. He is responsible to make sure all the queries and programs works in fraction of seconds.

**Migrate database servers: -** Sometimes, users using oracle would like to shift to SQL server or Netezza. It is the responsibility of DBA to make sure that migration happens without any failure, and there is no data loss.

**Backup and Recovery: -** Proper backup and recovery programs needs to be developed by DBA and has to be maintained him. This is one of the main responsibilities of DBA. Data/objects should be backed up regularly so that if there is any crash, it should be recovered without much effort and data loss.

**Security: -** DBA is responsible for creating various database users and roles, and giving them different levels of access rights.

**Documentation: -** DBA should be properly documenting all his activities so that if he quits or any new DBA comes in, he should be able to understand the database without any effort. He should basically maintain all his installation, backup, recovery, security methods. He should keep various reports about database performance.

In order to perform his entire task, he should have very good command over DBMS.

**Transaction Management:**

A transaction is an event which occurs on the database. Generally a transaction reads a value from the database or writes a value to the database. If you have any concept of Operating Systems, then we can say that a transaction is analogous to processes.

Although a transaction can both read and write on the database, there are some fundamental differences between these two classes of operations. A read operation does not change the image of the database in any way. But a write operation, whether performed with the intention of inserting, updating or deleting data from the database, changes the image of the database. That is, we may say that these transactions bring the database from an image which existed before the transaction occurred (called the **Before Image**or**BFIM**) to an image which exists after the transaction occurred (called the **After Image**or**AFIM**).  
  
 **The Four Properties of Transactions:**

Every transaction, for whatever purpose it is being used, has the following four properties. Taking the initial letters of these four properties we collectively call them the **ACID Properties**. Here we try to describe them and explain them  
  
**Atomicity** − this property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.

**Consistency** − the database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.

**Durability** − the database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.

**Isolation** − In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.

**Storage Management:**

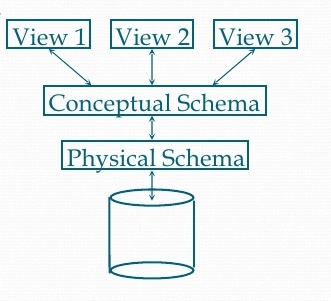
In relational Database Management Systems (DBMS1), the database is placed on secondary storage (disk). When the data in the database are required, they need to be read into memory and may be written back to disk if updated. The data must be organized on the disk for efficient accessing. In order to increase the efficiency of storage management, part of memory is used to cache some of these data. The management of data on disk and in memory belongs to the storage management of DBMS. The disk access is the bottleneck of the database system in many workloads, because the access time of disk is several magnitudes longer than that of memory. The objective of storage management is to improve the efficiency of disk access, therefore to improve the performance of the whole system. The storage management can be divided into three levels:

**1. Logical level** is the higher level. In this level, the logical information of the DBMS is used to optimize the data stored in memory. Not much research work is on this level and many DBMSs do not have this level in their storage management.   
  
**2. Physical in-memory level** is the middle level. In this level, the popular database pages are cached in the in-memory space buffer pool. The buffer pool can absorb most requests to the disk therefore reduce the number of disk accesses. This approach is used in almost all DBMSs and has important effects on their performance. There are many studies about the general buffer management and the buffer pool management in DBMS.  
  
**3. Physical on-disk level** is the lower level. The layout of database data and meta-data on the disk is studied to improve the disk I/O efficiency. It can be further divided into two sub-levels: software level and disk level. In the software level, the on-disk data structure and data placement strategy are designed based on the logical structure of data. This area has been studied intensively in the design of file systems. In the disk level, the logical structure of data is not considered. Instead, only the physical nature of disk is utilized to improve the disk I/O. This area has been studied in the design of storage systems (e.g. RAID systems).  
Other things that are considered are: typical workload types and different algorithms.

**Level of Abstraction**

There are three types of data abstraction:

* Physical schema: The lowest level of abstraction describes how the data are actually stored.
* Logical schema/ conceptual schema: The next higher level of abstraction describes what data are actually stored in the database and what relationship exists among those data.
* View level: The highest level of abstraction describes only part of the entire database.



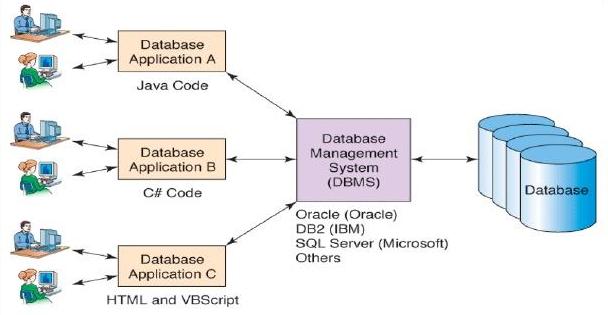
**Figure: level of Abstraction**

**Organizational DBMS**

**Organizational DMBS typically:**

* Support several users simultaneously
* Include more than one application
* Involve multiple computers
* Are complex in design
* Have many tables
* Have many database

In this figure below, we can see that, organization use DBMS with different code and different types of DBMS examples.



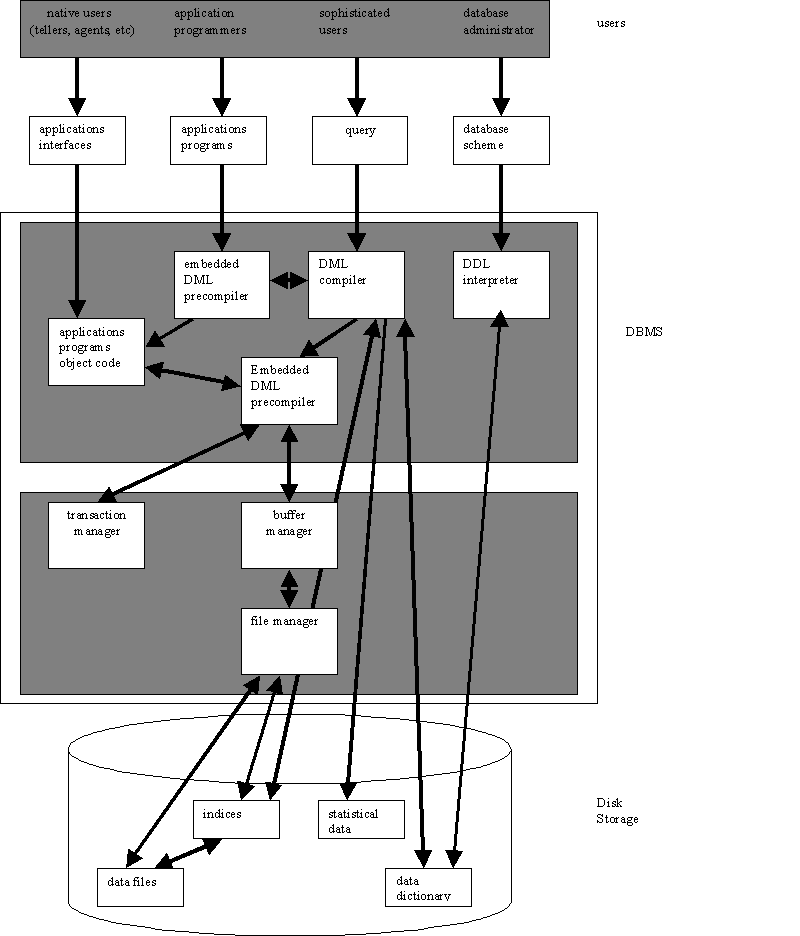
**Figure: the organizational DBMS**

**Overall Structure of DBMS**

DBMS are very large and typically divided into modules. Some of the services are also provided by the operating system of the host computer.   
The following is an example of what the structure might be:

* **Query processor**
  + **DML compiler** Translates the Data Manipulation Languages into query Engine instructions. It might also do optimization for query.
  + **Embedded DML precompiled** Converts the DML statements in an application program to normal procedure calls in the host language.
  + **DDL interpreter** Interprets DDL statements and records them in a set of tables containing *metadata*
* **Storage Manager**
  + **Authorization and integrity manager** Tests for the satisfaction of integrity constraints and Checks the authority of user to perform various actions.
  + **Transaction Manager** Ensures the database remains in a consistent (correct) state despite system failures.
  + **File manager** Responsible for the allocation of space on the disk storage system.
  + **Buffer manager** manages the data coming into and out of the system, including the caching of data.
* **Data structures**
  + **Data** files the database itself.
  + **Data dictionary** the metadata about the structure of the database. **Actually, this is a critical element in the DBMS!**
  + **Indices** Used to provide fast access to the data.
  + **Statistical data** The query processor uses this to optimize queries.

**Overall structure of DBMS is as:**



**Example of database management system**:

1. SQL
2. Oracle
3. FoxPro
4. MS access
5. My SQL

**SQL**

**Definition:** Structured Query Language makes it possible to obtain information fast from millions of records stored database though query commands.SQL software is extensively used in business, industry and government for designing custom databases that can be queried for hidden information.

**History and component**: Developed in 1970s by IBM, SQL was standardized by ANSL American National Standard Institute. SQL has two components namely data definition language( DDL) and data manipulation language(DML).A subcomponent of DDL is called data control language(DCL) controls what action a user is allowed to perform. Nowadays the standard is subject to continuous improvement by the joint technical committee ISO/IEC JTC.

**FEATURE:**

* Add/edit/delete databases ,table, view, roles, rules, defaults, function
* Register multiple servers and manage them all from one application
* Set primary keys, trigger, indexes, constraints
* Change column properties
* View/edit/export data in the data management section
* Backup and restore databases
* Attach, detach, and shrink databases
* Relationships management screen for foreign keys etc.
* Transfer database wizard to transfer databases from server to server

**2. ORACLE:**

Database commonly referred to as Oracleis a produced and marketed by Oracle Corporation.

“**Larry Ellison”** and his friend, former co workers **“Bob Miner”** and **“Ed oats”** started the consultancy Software Development Laboratories (SDL) in 1977. SDL developed the original version of the oracle software. The name *oracle* comes from the code name of a CIA funded project Ellison had worked on while previously employed by Ampex.

**FEATURES:**

* Concurrency
* Read consistency
* Self managing database
* Backup and recovery
* High availability
* Business intelligence
* Table compression
* Parallel execution
* Data mining
* Content management
* Security
* Improved query performance and scalability

**3. Fox Pro:**

This is a text-based procedurally-oriented programming language and DBMS, originally published by “Fox Software and later by Microsoft, for MS-DOS, MS Windows, Apple Macintosh, and UNIX. Visual FoxPro is a data-centric object-oriented and procedural programming language produced by Microsoft. It is derived from FoxPro which was developed by Fox software beginning in 1984. Fox Technologies merged with Microsoft in 1992, after which the software acquired further features and the prefix “Visual”. The last version of FoxPro (2.6) worked under Mac OS. DOS, Windows, and UNIX: Visual FoxPro 3.0, the first “Visual” version, reduced platform support to only Mac and Windows, and later versions were Windows-only. The current version of FoxPro is COM-based and Microsoft has stated that they do not intend to create a Microsoft Net version.

**Feature:**

* Create a database or format structure to store data.
* Add records in database
* Modify records, Edit records.
* Search particular records.
* Arrange records in particular order (ascending or descending order).
* Delete records.
* Delete Database or modify structure.
* Prepare Reports and Labels.

**4. Microsoft Access:**

Microsoft Access is a database management system. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions or sold separately. On May 12, 2010, the current version of Microsoft Access 2010 was released by Microsoft in Office 2010; Microsoft Office Access 2007 was the prior version. MS Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases. Software developers and architects can use Microsoft Access to develop application software, and “Power Users” can use it to build software applications. Like other Office applications, Access is supported by Visual basic for Applications, an object-oriented programming language that can reference a variety of objects including DAO (Data Access Objects), active Data Objects and many other ActiveX components.

**Features:**

* Users can create tables, queries, forms and reports, and connect them together with the help of this.
* Access also has report creation features that can work with any data source that Access can “access”.
* The original concept of Access was for end users to be able to “access” data from any source.
* Other features include: the import and export of data to many formats including Excel, Outlook, ASCII, dBase, Paradox, FoxPro, SQL Server, Oracle, ODBC, etc.
* It also has the ability to link to data in its existing location and use it for viewing, queering, editing, and reporting. This allows the existing data to change while ensuring that Access uses the latest data.
* Access is often used by people downloading data from enterprise level database for manipulation, analysis, and reporting locally.
* One of the benefits of Access from a programmer’s perspective is its relative compatibility with SQL (Structured query language) – queries can be viewed graphically Or edited as SQL statements and SQL statements can be used directly in Macros and VBA Modules to manipulate Access tables.
* Microsoft Access is designed to scale to support more data and users by linking to multiple Access database or using back-end database like Microsoft SQL Server.

**5. My SQL:**

It is named after developer Michael Wideners’ daughter, my. The SQL phrase stands for Structured Query Language.

The My SQL development project has made its source code available under the items of the GNU General Public License, as well as under a variety of projector agreements. My SQL was owned and sponsored by a single for-profit firm, the Swedish company My SQL AB, now owned by Oracle Corporation. My SQL is an open source database management system and is used in some of the most frequently visited websites on the Internet, including Flicker, Nokia.com, and You Tube and as previously mentioned, Wikipedia, Google, Face book and Twitter.

**Features:**

* Cross-platform support.
* Stored procedures.
* Triggers
* Cursors
* Updatable Views
* Replication support (i.e. Master-Master Replication & Master-slave Replication) with one master per slave, many slaves per master, no automatic support for multiple master per slave, many slaves per master, no automatic support for multiple masters per slave.
* Embedded database library.

**Advantages of Database Management System**

There are many advantages of database management system. Some of them are given below:

**Warehouse of Information:**

The database management systems are warehouse of information, where large amount of data can be stored. The common examples in commercial application are inventory data, personnel data. The best examples for the same would be the address book of a Cell phone, digital diaries etc.

**Defining Attributes:**

The unique data table field in a table is a primary key. The primary key helps in the identification of data. It also checks for duplicates to reduce data redundancy. There are tables which have a secondary key refers to the primary key of another table to establish a relationship between the two tables

**Systematic Storage:**

The data is stored in the form of tables which consists of rows and columns.

**Changes to schema:**

The table schema can be changed because it is not a permanent platform. The tables in the system can be edited to make any change without hampering the application that depends on that particular database.

**No Language Dependence:**

The database management systems are not language dependent. Therefore they can be used with various languages and on various platforms.

**Table joints:**

The data in two or more tables can be integrated into a single table. This enables to reduce the size of the database and helps in easy retrieval of data.

**Multiple Simultaneous Usages:**

The database can be used simultaneously by a number of users.

**Data security:**

Data management systems help to keep the data secured.

**Data consistency:** Data consistency ensures a consistent view of data to every user. It includes the accuracy, validity and integrity of related data.

**Disadvantages of Database Management**

Although the database system yields considerable advantages over previous data management approaches, database systems do carry significant disadvantages. For example:

**Increased cost :**

Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software and personnel required to operate and manage a database system can be substantial. Training, licensing and regulation compliance costs are often overlooked when database system are implemented.

**Management complexity:**

Database systems interface with many different technologies and have a significant on a company’s resources and culture. The changes introduced by the adoption of database system must be properly managed to ensure that they help advance the company’s objectives. Given the fact that database systems hold crucial company data that are accessed from multiple sources, security issues must be accessed constantly.

**Maintaining currency:**

To maximize the efficiency of the database system, you must keep your system current. Therefore, you must perform frequent updates and apply the latest patches and security measures to all components. Because database technology advances rapidly, personnel training costs tend to be significant. Vendor dependence given the heavy investment in technology and personnel training, companies might be reluctant to change database vendors. As a consequence, vendors are less likely to offer pricing point advantages to existing customers, and those customers might be limited in their choice of database system components.

**Frequent upgrade / replacement cycles:**

DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new feature

**Cost of Hardware and Software;**

A processor with high speed of data processing and memory of large size is required to run the DBMS software. It means that you have to upgrade the hardware used for file-based system. Similarly, DBMS software is also very costly.

**Cost of Data Conversion:**

When a computer file-based system is replaced with database system, the data stored into data file must be converted to database file. It is very difficult and costly method to convert data of data file into database. You have to hire database system designers along with application programmers. Alternatively, you have to take the services of some software house. So a lot of money has to be paid for developing software.

**Cost of Staff Training:**

Most database management system is often complex systems so the training for users to use the DBMS is required. Training is required to all levels, including programming, application development, and database administration. The organization has to be paid a lot of amount for the training of staff to run the DBMS.

**Appointing Technical Staff:**

The trained technical persons such as database administrator, application programmers, data entry operations etc. are required to handle the DBMS. You have to pay handsome salaries to these persons. Therefore, the system cost increases.

**Database Damage:**

In most of the organization, all data is integrated into a single database. If database is damaged due to electric failure or database is corrupted on the storage media, then your valuable data may be lost forever.

**Summary:**

A database management system (DBMS) is a software package with computer programs that control the creation, maintenance and use of a database. It allows organizations to conveniently develop databases for various applications by database administration (DBMS) and other specialists. A database is an integrated collection of data records, files and other objects. A DBMS process data stored in independent files and produce accurate, relevant and timely information which is a key to good decision making. It implements a database and manages its context properly.

DBMS allows different users’ application programs to concurrently access the same database. DBMSs may use a variety of database models such as the rational model or object model and so on to conveniently describe and support applications. Database design defines the database structure. Basically well designed database facilitates data management and generates valuable information.

On the other hand, poorly designed database leads to bad decision making and organization failure. It typically supports query languages which are in fact high level programming languages, dedicated database languages that considerably simplify writing database application programming languages.

A database provides facilities such as:

* controlling data access,
* enforcing data integrity,
* managing concurrency control
* recovering the database after failures and
* restoring it from backup files as well as
* Maintaining database security.

Besides some facilities, DBMS has also some limitation like:

* A complex conceptual design process,
* High DBMS acquisition costs
* Database systems are complex, difficult and time-consuming to design
* Requirement of extensive programming and face difficulty in changing structure
* Damage of whole structure affects all programs
* The need to hire database related employees and initial training for all the programmers and users.

