**DBMS Functions'**

There are several functions that a DBMS performs to ensure data integrity and consistency of data in the database. The ten functions in the DBMS are: data dictionary management, data storage management, data transformation and presentation, security management, multiuser access control, backup and recovery management, data integrity management, database access languages and application programming interfaces, database communication interfaces, and transaction management.

1. **Data Dictionary Management**

The DBMS uses this function to look up the required data component structures and relationships. When programs access data in a database they are basically going through the DBMS. This function removes structural and data dependency and provides the user with data abstraction. In turn, this makes things a lot easier on the end user. The Data Dictionary is often hidden from the user and is used by Database Administrators and Programmers.

1. **Data Storage Management**

This particular function is used for the storage of data and any related data entry forms or screen definitions, report definitions, data validation rules, procedural code, and structures that can handle video and picture formats. Users do not need to know how data is stored or manipulated. Also involved with this structure is a term called performance tuning that relates to a database’s efficiency in relation to storage and access speed.

1. **Data Transformation and Presentation**

This function exists to transform any data entered into required data structures. By using the data transformation and presentation function the DBMS can determine the difference between logical and physical data formats.

1. **Security Management**

This is one of the most important functions in the DBMS. Security management sets rules that determine specific users that are allowed to access the database. Users are given a username and password or sometimes through biometric authentication (such as a fingerprint or retina scan) but these types of authentication tend to be more costly. This function also sets restraints on what specific data any user can see or manage.

1. **Multiuser Access Control**

Data integrity and data consistency are the basis of this function. Multiuser access control is a very useful tool in a DBMS, it enables multiple users to access the database simultaneously without affecting the integrity of the database.

1. **Backup and Recovery Management**

Backup and recovery is brought to mind whenever there is potential outside threats to a database. For example if there is a power outage, recovery management is how long it takes to recover the database after the outage. Backup management refers to the data safety and integrity; for example backing up all your mp3 files on a disk.

1. **Data Integrity Management**

The DBMS enforces these rules to reduce things such as data redundancy, which is when data is stored in more than one place unnecessarily, and maximizing data consistency, making sure database is returning correct/same answer each time for same question asked.

1. **Database Access Languages and Application Programming Interfaces**

A query language is a nonprocedural language. An example of this is SQL (structured query language). SQL is the most common query language supported by the majority of DBMS vendors. The use of this language makes it easy for user to specify what they want done without the headache of explaining how to specifically do it.

1. **Database Communication Interfaces**

This refers to how a DBMS can accept different end user requests through different network environments. An example of this can be easily related to the internet.  A DBMS can provide access to the database using the Internet through Web Browsers (Mozilla Firefox, Internet Explorer, Netscape).

1. **Transaction Management**

This refers to how a DBMS must supply a method that will guarantee that all the updates in a given transaction are made or not made. All transactions must follow what is called the ACID properties.

**A – Atomicity: states a transaction is an indivisible unit that is either performed as a whole and not by its parts, or not performed at all. It is the responsibility of recovery management to make sure this takes place.**

**C – Consistency: A transaction must alter the database from one constant state to another constant state.**

**I – Isolation: Transactions must be executed independently of one another. Part of a transaction in progress should not be able to be seen by another transaction.**

**D – Durability: A successfully completed transaction is recorded permanently in the database and must not be lost due to failures.**

## DATABASE USERS

**Database administrators** – DBA is responsible for authorizing access to the database, for coordinating and monitoring its use, and acquiring software and hardware resources as needed.

**Database designers** – identify data to be stored in the database and choosing appropriate structures to represent and store the data. Most of these functions are done before the database is implemented and populated with the data. It is the responsibility of the database designers to communicate with all prospective users to understand their requirements and come up with a design that meets these requirements. Database designers interact with all potential users and develop views of the database that meet the data and processing requirements of these groups. The final database must support the requirements of all user groups.

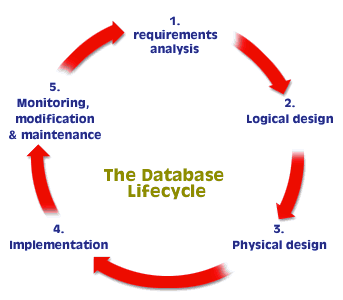
## End Users

* **Casual End Users** – occasionally access, may need different information each time. Use query language to specify requests.
* **Naïve or parametric end users** – main job is to query and update the database using standard queries and updates. These canned transactions have been carefully programmed and tested. Examples?
* **Sophisticated end users** – engineers, scientists, analysts who implement applications to meet their requirements.
* **Stand alone users** – maintain personal databases using ready made packages.

## System Analysts and Programmers

* Determine the end user requirements (especially naïve and parametric end users) and develop specifications for canned transactions that meet the requirements.
* Application programmers implement the specifications as programs, then test and debug the programs.

**The database life cycle (DBLC)**

The database life cycle (DBLC) defines the stages involved in getting any type of database off the drawing board and up and running.   
In fact, the DBLC never ends because database monitoring, modification, and maintenance are part of the life cycle, and these activities continue long after a database has been implemented. Put simply, the DBLC encompasses the lifetime of the database.   
The five stages in the database life cycle are:

1. Requirements analysis
2. Logical design
3. Physical design
4. Implementation
5. Monitoring, modification, and maintenance

**I. Requirements analysis**

Requirements Analysis is the first and most important stage in the *Database Life Cycle*.   
It is the most labor-intensive for the database designer.  
This stage involves assessing the informational needs of an organization so that a database can be designed to meet those needs.

**II. Logical design**

During the first part of Logical Design, a *conceptual model* is created based on the needs assessment performed in stage one. A conceptual model is typically an *entity-relationship (ER) diagram* that shows the tables, fields, and primary keys of the database, and how tables are related (linked) to one another.   
The tables sketched in the ER diagram are then *normalized*. The *normalization* process resolves any problems associated with the database design, so that data can be accessed quickly and efficiently.

1. *conceptual model:* A description of the structure of a database.
2. *entity-relationship (ER) diagram:* A diagram used during the design phase of database development to illustrate the organization of and relationships between data during database design.
3. *normalization:* The process of applying increasingly stringent rules to a relational database to correct any problems associated with poor design.

**III. Physical design**

The Physical Design stage has only one purpose: to maximize database efficiency.  
This means finding ways to speed up the performance of the RDBMS. Manipulating certain database design elements can speed up the two slowest operations in an RDBMS: retrieving data from and writing data to a database.

**IV. Implementation**

During the implementation stage of the DBLC, the tables developed in the ER diagram (and subsequently normalized) are converted into SQL statements and “fed” into the RDBMS to create a database. By this stage in the DBLC, the *System Administrator* has installed and configured an RDBMS.

**V. Monitoring, Modification, and Maintenance**

A successfully implemented database must be carefully *monitored* to ensure that it’s functioning properly and that it’s secure from unauthorized access. The RDBMS usually provides utilities to help monitor database functionality and security.   
Database *modification* involves adding and deleting records, importing data from other systems (as needed), and creating additional tables, user views, and other objects and tools. As an organization grows, its *information system* must grow to remain useful.