

# DATABASE MANAGEMENT SYSTEM

## REVISION NOTES

A database is an organized collection of data. It is considered as a container of information. In the manual system, you would maintain several files with different bits of information while in the computerized system you would use database programs such as Microsoft Access, OpenOffice.org Base, and MySQL, to organize the data as per your business need.

A database is basically a computerbased record/data/information keeping system. Data is raw, unorganised facts and entities relevant to the user need to be processed such as a digital representation of text, numbers, graphical images or sound. The data are stored in such a way that, they are independent of the programs used by the people for accessing the data.

Database can be created with the help of following structures



Database Structure

### DATABASE MANAGEMENT SYSTEM (DBMS):

DBMS is a collection of programs that enables users to create, maintain database and control all the access to the database. It is a computer based record keeping system.

The primary goal of the DBMS is to provide an environment that is convenient and efficient for user to retrieve and store information. It acts as an interface between the application program and the data stored in the database.

DBMS is a software package that manages database. e.g. MySQL, INGRES, MS-Access, etc.

DBMS is actually a tool that is used to perform any kind of operation on data in database. It also maintains data consistency in case of multiple users. The purpose of a DBMS is to bridge the gap between information and data.

Data can be organized into two types:

- **Flat File:** Data is stored in a single table. Usually suitable for less amount of data.
- **Relational:** Data is stored in multiple tables and the tables are linked using a common field. Relational is suitable for medium to large amount of data.

### DATABASE SERVERS

Database servers are dedicated computers that hold the actual databases and run only the DBMS and related software.

### ADVANTAGES OF A DATABASE/DBMS

The centralised nature of database system provides several advantages, which overcome the limitations of the conventional file processing system. These advantages are as follows

1. **Reduce Data Redundancy:** Redundancy means 'duplication of data'. This eliminates the replication of data item in different files, extra processing required to face the data item from a large database. This also ensures data consistency and saves the storage space.
2. **Enforcing Data Integrity:** It means that, the data contained in the database is accurate and consistent. Integrity constraints or consistency rules can be applied to database, so that the correct data can be entered into the database.
3. **Data Sharing:** The data stored in the database can be shared among multiple users or application programs.

4. **Data Security:** The DBMS ensures that the access of database is done only through an authorised user.
5. **Ease of Application Development:** The application programmer needs to develop the application programs according to the user's needs.
6. **Backup and Recovery:** The DBMS provides backup and recovery sub-system that is responsible to recover data from hardware and software failures.
7. **Multiple Views of Data:** A view may be the subset of database. Various users may have different views of the database itself.
8. **Enforced Standards:** It can ensure that, all the data follow the applicable standards.
9. **Data Independence:** System data descriptions are independent from the application programs.

### DISADVANTAGES OF A DATABASE/DBMS

There are many advantages of database, but database also have some minor disadvantages. These disadvantages are as follows

1. **Cost of Hardware and Software:** Through the use of a database system, new costs are generated due to additional hardware and software requirements.
2. **Complexity:** A database system creates additional complexity and requirements.
3. **Database Failures:** If database is corrupted due to power failure or it is corrupted on the storage media, then our valuable data may be lost or the system will stop working.
4. **Lower Efficiency:** A database system is a multi-user software, which is less efficient.

### SOME KEY FEATURES OF A DATABASE:

1. A database can have one or many tables. An address book example is a very simple one, in real world there are many more details involved.
2. Each table in a database contains information about one type of item. So, a database is a container that holds tables and other objects and manages how they can be used.
3. A record uniqueness in every table. It is important to be able to distinguish between different items having duplicate values. Uniqueness helps to avoid accidental duplication of records caused by user or computer error.
4. Every database table should have one or more fields designated as key. You can assign a unique value to this key for differentiating records that may have similar names or addresses.

### COMPONENTS OF A DATABASE

A database consists of several components. Each component plays an important role in the database system environment.

The major components of database are as follows

- **Data:** It is raw numbers, characters or facts represented by value. Most of the organisations generate, store and process large amount of data. The data acts as a bridge between the hardware and the software. Data may be of different types such as User data, Metadata and Application Metadata.
- **Software:** It is a set of programs that lies between the stored data and the users of database. It is used to control and manage the overall computerised database. It uses different types of software such as MySQL, Oracle, etc.
- **Hardware:** It is the physical aspect of computer, telecommunication and database, which consists of the secondary storage devices such as magnetic discs, optical discs, etc., on which data is stored.
- **Users:** It is the person, who needs information from the database to carry out its primary business responsibilities.

### TYPES OF USERS

The various types of users which can access the database system are as follows

- **Database Administrator (DBA):** A person, who is responsible for managing or establishing policies for the maintenance and handling the overall database management system is called DBA.
- **Application Programmer:** A person, who writes application programs in programming languages to interact and manipulate the database are called application programmer.
- **End-user:** A person, who interacts with the database system to perform different operations on the database like inserting, deleting, etc., through menus or forms is called end-user.

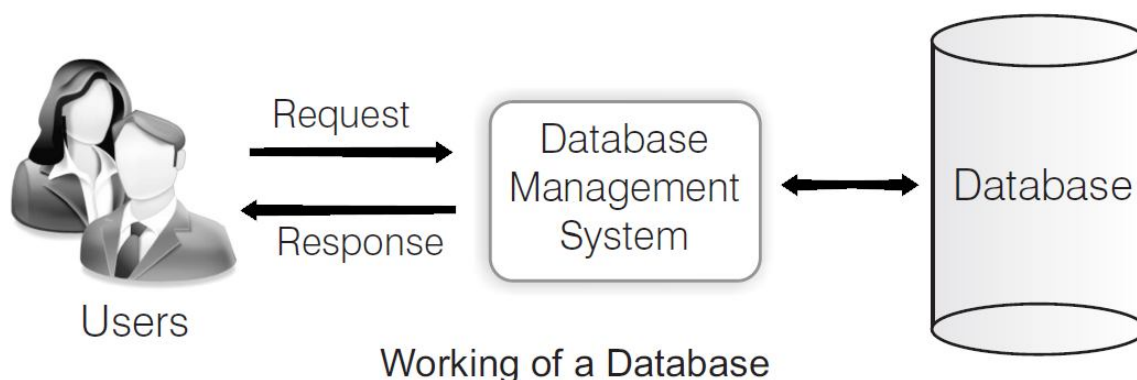
## RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

RDBMS is a type of DBMS that stores data in the form of relations (tables). Relational databases are powerful, so they require few assumptions about how data is related or how it will be extracted from the databases.

An important feature of relational database system is that a single database can be spread across several tables. Base, Oracle, DB2, SAP, Sybase, ASE, Informix, Access, etc., are the examples of RDBMS.

## WORKING OF A DATABASE

Database is created to operate large quantities of information by input, store, retrieve and manage the information. It is a centralised location which provides an easy way to access the data by several users. It does not keep the separate copies of a particular data file still a number of users can access the same data at the same time.



As the diagram shows, DBMS works as an interface between the user and the centralised database. First, a request or a query is forwarded to a DBMS which works (i.e. a searching process is started on the centralised database) on the received query with the available data and if the result is obtained, it is forwarded to the user.

If the output does not completely fulfill the requirements of the user, then a rollback (again search) is done and again search process is performed until the desired output is obtained.

## DATA INTEGRITY

Data Integrity ensures the accuracy, reliability and consistency of the data during any operation. Each type of data integrity are as follows

- **Entity Integrity:** It defines the primary key of a table. Entity integrity rule on a column does not allow duplicate and null values.
- **Domain Integrity:** It defines the type, range and format of data allowed in a column. Domain integrity states that all values in a column must be of same type.
- **Referential Integrity:** It defines the foreign key concepts. Referential integrity ensures that data in related tables remains accurate and consistent before and after changes.
- **User Defined Integrity:** If there is some business requirements which do not fit any above data integrity then user can create own integrity, which is called user defined integrity.

## KEY FIELDS

The key is defined as the column or the set of columns of the database table which is used to identify each record uniquely in a relation. If a table has id, name and address as the column names, then each one is known as the key for that table. The key field is a unique identifier for each record.

## TYPES OF KEY FIELDS

The following are the types of key fields available in the DBMS system

**Primary Key:** A field or a set of fields that uniquely identify each record in a table is known as a primary key. Each relation has atleast one column for which each row that must have a unique value. Only one column attribute can be defined as a primary key for each table.

A primary key must possess the following properties

- It does not allow null values.
- It has a unique index.
- It allows numbers and text both.

e.g. In the student's table, StudentId works as a primary key because it contains Ids which are unique for each student. Note: Data cannot be primary key.

**Candidate Key:** The set of all attributes which can uniquely identify each tuple of a relation are known as candidate keys. Each table may have one or more candidate keys and one of them will become the primary key. The candidate key of a relation is always a minimal key. e.g. Column StudentId and the combination of FirstName and LastName work as the candidate keys for the student table.

A candidate key must possess the following properties

- For each row, the value of the key must uniquely identify that row.
- No attribute in the key can be discarded without destroying the property of unique identification.

**Alternate Key:** From the set of candidate keys after selecting one of the keys as a primary key, all other remaining keys are known as alternate keys. e.g. From the candidate keys (StudentId, combination of FirstName and LastName), if StudentId is chosen as a primary key, then the combination of FirstName and LastName columns work as alternate keys.

**Foreign Key:** A field of a table (relation) that references the primary key of another table is referred to as foreign key. The relationship between two tables is established with the help of foreign key. A table may have multiple foreign keys and each foreign key can have a different referenced table. Foreign keys play an essential role in database design, when tables are broken apart, then foreign keys make it possible for them to be reconstructed. e.g. CourseId column of student table (reference table) works as a foreign key as well as a primary key for course table (referenced table).

Data in a relational database management system (RDBMS) is organized in the form of tables.

## DATABASE OBJECTS :

- 1) **Table:** A table is a set of data elements (values) that is organized using a model of vertical columns and horizontal rows. A table has a defined number of columns, but can have any number of rows. Each row is identified by the values appearing in a particular column identified as a unique key index or the key field.
- 2) **Columns or Fields or Attributes:** A column is a set of data values of a particular simple type, one for each row of the table. The columns provide the structure according to which the rows are composed. For example, cFirstName, or cLastName are fields in a row.
- 3) **Rows or Records or Tuples:** A row also called a Record or Tuple represents a single data item in a table. Each row in a table represents a set of related data, and every row in the table has the same structure.

### **CREATING DATABASE USING OPENOFFICE:**

- 1) Open the OpenOffice Base Application by Clicking on Start>Programs>OpenOffice.org 4>OpenOffice.org Base.
- 2) Create a new database by selecting the option Create a new database
- 3) Database wizard appear. Click Finish. The Save As dialog box appears.
- 4) Specify a name for the database in the File name: field and click Save.

### **CREATE TABLES:**

Tables are the basic building blocks of a database. You store the data in the database in the form of tables.

There are different ways to create a table:

#### **a) Use Wizard To Create Table:**

The following are the steps to create a table:

- 1) Click on Tables > Use Wizard to Create Table
- 2) Click the Select Fields > Choose Category > Select the table > Click on Next Button.
- 3) Select the fields as per the requirements. Click on Next button
- 4) Set the data types and properties of the selected fields. Click on Next Button
- 5) Set the Primary Key and Click on Next Button
- 6) Window to rename the table will open. A user can either go ahead with the same table name or can change it.
- 7) Click on Finish to insert the data in the table.

#### **b) Creating table using Design View**

- 1) Click on Create Table in Design View.
- 2) Specify the field name and data type of the field.
- 3) Save the table by clicking on File > Save
- 4) Specify the table name. The default name is Table1. Click OK.
- 5) An alert appears, if there is no primary key in the table.
- 6) Click yes and set the primary key and finally save the table.

### **DATA TYPES:**

Datatypes are used to identify which type of data (value) we are going to store in the database. Data types in OpenOffice base are broadly classified into five categories listed below.

- Numeric Types
- Alphanumeric Types
- Binary Types
- Date time
- Other Variable types

**Numeric Types:** Numeric data types in a database can be used for storing information such as mobile number, roll number, door number etc. The different types of numeric data types available are listed here.

Name	Data type	Description
BOOLEAN	Yes / No	Values as 0 or 1. Example: True or False, Yes or No.
TINYINT	Tiny Integer	Store integer range between 0 to 255
SMALLINT	Small Integer	Store integer range between $-2^{15}$ to $+2^{15}-1$
INTEGER	Integer	Store integer range between $-2^{31}$ to $+2^{31}-1$
BIGINT	Big Integer	Range between $-2^{63}$ to $+2^{63}-1$
NUMERIC	Number	Maximum precision of $e^{(+/-)231}$
DECIMAL	Decimal	Maximum precision of $e^{(+/-)231}$
REAL	Real	$2^{-1074}$ to $(2-2^{-52}) * 2^{1023}$
FLOAT	Float	$2^{-1074}$ to $(2-2^{-52}) * 2^{1023}$
DOUBLE	Double	$2^{-1074}$ to $(2-2^{-52}) * 2^{1023}$

**Alphanumeric Types:** This data type is used to store information which has alphabets as well as numbers for example address, book summary field etc. The different types of Alphanumeric Types available are listed here.

Name	Data type	Description
LONGVARCHAR	Memo	Stores up to the max length or number indicated by user. It accepts any UTF 8 Character.
CHAR	Text (fix)	Stores exactly the length specified by user. Pads with trailing spaces for shorter strings. Accepts any UTF 8 Character.
VARCHAR	Text	Stores up to the specified length. No padding (Same as long var char)
VARCHAR_IGNORE CASE	Text	Stores up the specified length. Comparisons are not case sensitive but stores capitals as you type them.

**Binary Types:** Binary data types are used for storing data in binary formats. Binary data types in a database can be using for storing photos, music files, etc.

Name	Data type	Description
LONGVARBINARY	Image	Stores any array of bytes (images, sounds, etc.). No validation required.
BINARY	Binary (fix)	Stores any array of bytes. No validation required.
VARBINARY	Binary	Stores any array of bytes. No validation required.

**Date time:** Date time data types in a database can be used for storing information such as date of birth, date of admission, date of product sale, etc.



Name	Description	Format
Date	Stores month, day and year information	1/1/99 to 1/1/9999
Time	Stores hour, minute and second information	Seconds since 1/1/1970
Timestamp	Stores date and time information	

#### OTHER DATA TYPES:

Name	Description
Other/Object	Stores serialized Java objects “ user application must supply serialization routines

#### PERFORM OPERATIONS ON TABLE

In OpenOffice Base, data is stored in tables which can be inserted, modified and removed using appropriate options.

##### Inserting Data In The Table:

To insert the data in the table, follow the steps:

- 1) Select the table > Double click on it.
- 2) The table will open in Datasheet View.
- 3) Insert the required number of records in Datasheet View.

##### Editing Records In The Table:

To edit the data in the table, follow the steps:

- 1) Select the table > Double click on it.
- 2) The table will open in Datasheet View.
- 3) Edit the required record in Datasheet View.

##### Deleting Records From The Table:

To remove the data from the table, follow the steps:

- 1) Select the table > Double click on it.
- 2) The table will open in Datasheet View.
- 3) Select the data > right click on selected data > select the Delete option

#### FIELD PROPERTIES:

To set the field properties: Select the table > Right click > Select the option Edit > the table will open in Design View.

In design view there are different properties of fields according to the data type set for each field.

The properties of numeric type data is given below :

- 1) **AutoValue** – if set to yes then field will get the auto numeric values.
- 2) **AutoValue** – if set to yes then field will get the auto numeric values.
- 3) **Length** – By default length of the field is 10 but the size of the field can be set to maximum length.
- 4) **Default Value** – A default value can be set for a field if user don't provide any value while entering the values in the table.
- 5) **Format** – This property helps to set the format of the data entered in the field such as 91-222-333.

The properties of character type data is shown below :

- 1) **Entry Required** – if set to yes then it will be must to insert the value in the field.

- 2) **Length** – By default length of the field is 10 but the size of the field can be set to maximum length.
- 3) **Default Value** – A default value can be set for a field if user don't provide any value while entering the values in the table.
- 4) **Format** – This property helps to set the format of the data entered in the field such as 91-222-333.

**SORTING DATA:** Sorting means to arrange the data in either ascending order or descending order.

### **REFERENTIAL INTEGRITY:**

Referential integrity is used to maintain accuracy and consistency of data in a relationship. In Base, data can be linked between two or more tables with the help of primary key and foreign key

Referential integrity helps to avoid:

- 1) Adding records to a related table if there is no associated record available in the primary key table.
- 2) Changing values in a primary if any dependent records are present in associated table(s).
- 3) Deleting records from a primary key table if there are any matching related records available in associated table(s).

### **CREATING AND EDITING RELATIONSHIPS BETWEEN TABLES:**

A relationship refers to an association or connection between two or more tables. Relationships between tables helps to:

- Save time as there is no need to enter the same data in separate tables.
- Reduce data-entry errors.
- Summarize data from related tables.

You can create a relationship between any two tables by selecting Relationships option from the Tools menu.

There are three types of relationships which can be created in tables:

- 1) **ONE to ONE Relationship:** In this relationship, both the tables must have primary key columns.
- 2) **One to Many Relationship:** In this relationship, one of the table must have primary key column.
- 3) **Many to Many Relationship:** In this relationship, no table has the primary key column.

### **REMOVE THE RELATIONSHIPS:**

The relationships applied on the tables can be removed also with the help of Delete option. Right Click on the relationship thread and select Delete option.

### **RETRIEVE DATA USING QUERY**

A query is to collect specific information from the pool of data. A query helps us join information from different tables and filter that information

OR

Queries are commands that are used to define the data structure and also to manipulate the data in the database.

There are two types of languages:-

DDL (Data Definition Language)

DML (Data Manipulation Language)

### **DATA DEFINITION LANGUAGE (DDL)**

DDL is used to define the structure of your tables and other objects in the database. In DBMS, it is used to specify a database schema as a set of definitions (expressed in DDL). In SQL, the Data Definition Language allows you to create, alter and destroy database objects. Basically, a data definition language is a computer language used to create and modify the structure of database objects in a database. These database objects include views, schemes, tables, indexes, etc.

This term is also known as data description language in some contexts, as it describes the fields and records in a database table.



Data definition language consists of various commands that lets you to perform some specified tasks as follows

- 1) **CREATE**: Uses to create objects in the database.
- 2) **ALTER**: Uses to alter the structure of the database table. This command can add up additional columns, drop existing columns and even change the data type of columns involved in a database table.
- 3) **DROP**: Uses to delete objects from the database.
- 4) **TRUNCATE**: Uses to remove all records from a table.
- 5) **RENAME**: Uses to rename an object.

### **DATA MANIPULATION LANGUAGE (DML)**

DML provides various commands used to access and manipulate data in existing database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data.

DML is mostly incorporated in SQL database. The basic goal of DML is to provide efficient human interaction with the system.

The DMLs are of two types

**Procedural DMLs** These require a user to specify what data is needed and how to get it.

**Non-Procedural DMLs** These require a user to specify what data is needed without specifying how to get it.

Various data manipulation language commands are as follows

- 1) **SELECT**: Used to retrieve data from a database.
- 2) **INSERT**: Used to insert data into a table.
- 3) **UPDATE**: Used to update existing data within a table.
- 4) **DELETE**: Used to delete all records from a table, the space of the records remains.
- 5) **LOCK TABLE**: Used to control concurrency.

### **SELECT STATEMENT :**

A **SELECT** statement retrieves zero or more rows from one or more tables. **SELECT** is the most commonly used Data Manipulation Language(DML) command. To retrieve all the columns in a table the syntax is: **SELECT \* FROM <Table Name>;**

For example, if you want to display all the data from table emp (short form of employee), the command is

Select \* from emp;

The **SELECT** statement has many optional clauses:

- **WHERE** specifies which rows to retrieve.
- **ORDER BY** specifies an order in which to return the rows.

for example : Select \* from emp where name = "Ravi";

The above query will show result of a particular employee named "Ravi".

Select \* from emp order by Salary;

The above query will show all the records of table emp according to ascending order of column Salary.

### **PERFORMING CALCULATIONS :**

In Base, simple calculations can be done on the data using arithmetic operators. Example:

1) To display the salary of all the employees after incrementing by 1000 then the following SQL command will be executed in Base. (Fields of table Employee are EmployeeID, FirstName, Salary)

Select "EmployeeID", "FirstName", "Salary" +1000 from "Employee";

2) To display the salary of all the employees after decreasing by 10000 then the SQL command will be:

Select "EmployeeID", "FirstName", "Salary" – 10000 from "Employee";

3) To display the salary of all the employees after incrementing it as twice the amount of present salary, then the SQL command will be .

Select "EmployeeID", "FirstName", "Salary" \* 2 from "Employee"

### **UPDATE STATEMENT :**

Update statement is used for modifying records in a database. The general syntax of the update statement is as follows:

UPDATE <table name> SET <Column name > = value [WHERE <Condition>];

for example :

1) To increase(update) the salary of employee "Ravi" by Rs 2000 (in table Employee) then the SQL command will be:

Update Employee set Salary = Salary + 2000 Where FirstName = "Ravi";

### **CREATE FORMS AND REPORTS USING WIZARD**

**FORM:** A form provides the user a systematic way of storing information into the database. It is an interface in a user specified layout that lets users to view, enter, and change data directly in database objects such as tables.

Creating Form Using Wizard : Steps To Create Form Using Wizard are :

- 1) Click Use Wizard to Create Form... option under Tasks group. The Form Wizard dialog box appears.
- 2) Select selective fields to be sent onto the form by selecting the field name and clicking >button and click Next.
- 3) Select the option Add Subform if you need to insert the contents in the table in a separate form and click Next.
- 4) Arrange selected fields in a form and click Next.
- 5) Select the data entry mode and click Next.
- 6) Specify the styles to be used in the form and click Next.
- 7) Specify the name of the form. Click Finish.

**REPORT:** A report helps to display the data in a summarized manner. It is used to generate the overall work outcome in a clear format.

Creating Reports using wizard : Steps To Create Report Using Wizard are :

- 1) Click on Use Wizard to Create Report... option available under Tasks.
- 2) Select all the table fields by selecting the >> button.
- 3) Redefine the label of the fields in the reports or else you can set the default name and click Next.
- 4) Define grouping for the fields of the table if required and click Next
- 5) Sort the field in the report by selecting the appropriate field name and sorting method(if required) and click Next.
- 6) Select the layout of the report and click Next.
- 7) Define a name for the report and click Finish.

.....