BROOKS
I.C.T MULTIVERSE

DATABASE TUTORIAL

Ms. Access For Dummies/Beginners
Page 1-10
MS. Access Advanced/PowerUsers
Page 11-42.

http://brooksict.orgfree.com/

Definition: Microsoft Access is an application software that is classified under databases. It is a program that is used to manage data in the form of tables.

MS-Access is a RDBMS (*Relational Database Management System*) application developed by Microsoft Inc. that runs on Windows operating System.

Other examples of database application software's include *Oracle, SQL, Sybase, MS Access, MS SQL Server, Paradox, DB/2, dBase, and FoxPro*.
Ms. Access/paradox/FoxPro/dBase are **PC-Based databases systems** while MS SQL Server/oracle/ingress/Sybase are **Co-operate based systems**

A *database* is a collection of related Files and records.
It is a collection of information related to a particular subject or purpose.

Examples of databases.

You can create a database for;

- Customers' details. - Library records.
- Personal records. - Flight schedules.
- Employees' records. - A music collection.
- An Address book (or Telephone directory), where each person has the Name, Address, City & Telephone no.

Database Concepts

A Database system consists of *four* major components

- **Data**: refers to the raw values and facts
- **Hardware**: some of the hardware components used include secondary storage devices such as hard disks and tapes.
- **Software**: a special program known as *Database management system* (D.B.M.S) is used for creation, manipulation and organization of data in a database.
- **User**: refers to the person who uses the DBMS, there are three categories of users -
  Database Administrator: manages and ensures data security of the DBMS.
  Application Programmer: uses the DBMS to come up with software that can solve problems for specific groups of people/companies
  End User: uses the DBMS to solve individual problems

DATABASE MANAGEMENT SYSTEM.

The computer Software used to manage, maintain database as well as view update and retrieve data is called database management system. A DBMS is a complex software, which creates, expands & maintains the database, and it also provides the interface between the user and the data in the database. Through the DBMS, users communicate their requirements to the database using Data Description Languages (DDL's) & Data Manipulation Languages (DML's).

In fact, the DBMS provide an interface between the user's programs and the contents of the database.

Functions of DBMS.

○ Provides an *interface* for data entry

○ Allocates *storage* space in the memory for data held

○ Provides *security* to data and has control features to limit access to data

○ Places data in an organized manner for *easy* manipulation

○ Creates or constructs the database contents through the Data Manipulation Languages.

○ Interfaces (links) the user to the database contents through Data Manipulation Languages.

○ It maintains a dictionary of the data within the database & manages the data descriptions in the dictionary.

○ Monitors the usage of the database contents to determine the rarely used data and those that are frequently used, so that they can be made readily available, whenever need arises.

Types of database Models

1. *Flat file/File Management System Model*: These refer to the traditional methods of storing files, i.e., the use of paper files and cabinets
2. *Hierarchical Model*: data arranged in tree format, highest level is called root It is a data structure where the data is organized like a family tree or an organization chart. In a Hierarchical database, the records are stored in multiple levels. Units further down the system are subordinate to the ones above. In other words, the database has branches made up of parent and child records. Each parent record can have multiple child records, but each child can have only one parent.
3. *Network Model*: data can be accessed through multiple paths; items are linked here. A Network database model represents many-to-many relationships between data. It allows a data element or record to be related to more than one other data element or record. For example, an employee can be associated with more than one department
4. ***Relational Model***

   A database with tables related to each other on a common field to facilitate the data retrieval from multiple tables. A Relational database system, has the ability to quickly find & bring information stored in separate tables together using queries, forms, & reports. This means that, a data element in any one table can be related to any piece of data in another table as long as both tables share common data elements.
5. *Object oriented Model* [still under development]

N.B: We shall use the relational model in our study.

The three key components of relational database design are

An ***entity*** defines any person such as student, teacher;
An ***attribute*** defines the additional characteristics or information for an entity.
For example, name, gender
A ***relationship*** is a logical linkage between two entities that describe how the entities are associated with each other.

Features of a DBMS/Access.

- *Tables* (They contain records where your data is stored, each **row** represents a ***record*** while each ***column*** represents a ***field***)

  What is a field?

  A field in a database is *a piece* of information about a subject. Each field is arranged as a column in table.

  What is a record?

  A record is *complete information* about a subject. A record is a collection of fields and presented as a row in a table of database.
- *Queries* (it is a database tool used to extract data that meet given **criteria** from a table and displays the results, also used to design relationships between tables)
- *Forms* (used as a window to your tables and queries to allow editing and adding new data)
- *Reports* (provides summaries of your data that can be displayed or **printed**)
- *Macros* (stores a group of instructions to **automate** tasks within access)

NOTE:

What is a module?

A module object in Access is a program written using VBA (Visual Basic for Application) to automate and customize database function.

Data Organization in a Database.

In a database, data are organized in five ways, these are

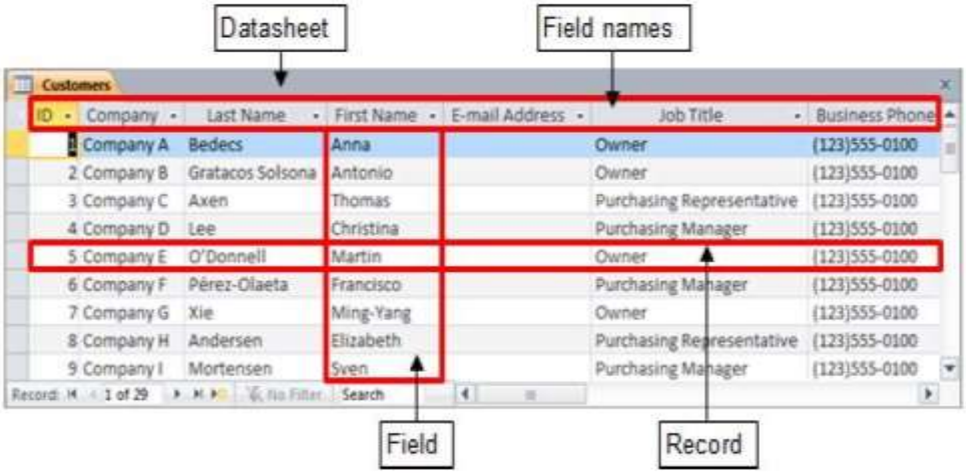
i) **Character types:** a character is the smallest element in a file, can be alphabetic, numeric or special.

ii) **Fields:** it is a combination of characters that represents information of the same data type (represented in a column in datasheet view of Ms. Access)

iii) **Records:** a record consists of a group of related fields that represents a single data item (represented as a row in datasheet view of Ms. Access)

iv) **Files:** it consists of a group of related records in a secondary storage media (a.k.a tables)

v) **Database:** it consists of a group of related files

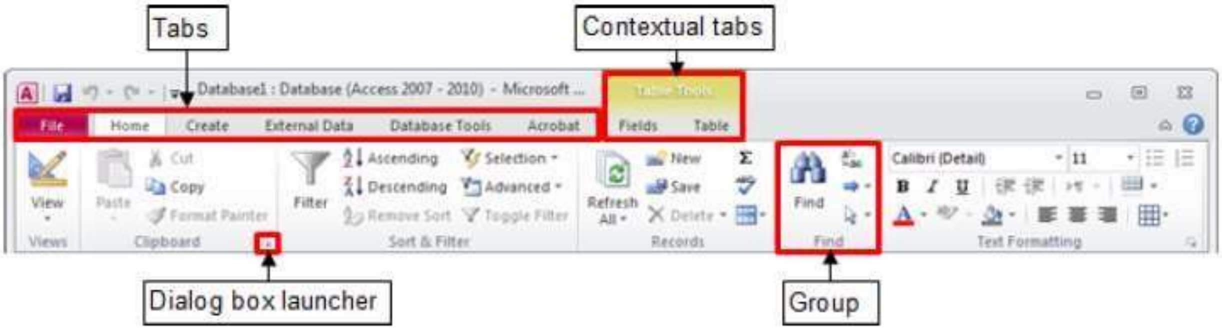


ID	Company	Last Name	First Name	E-mail Address	Job Title	Business Phone
1	Company A	Bedecs	Anna		Owner	(123)555-0100
2	Company B	Gratacos Solsona	Antonio		Owner	(123)555-0100
3	Company C	Axen	Thomas		Purchasing Representative	(123)555-0100
4	Company D	Lee	Christina		Purchasing Manager	(123)555-0100
5	Company E	O'Donnell	Martin		Owner	(123)555-0100
6	Company F	Pérez-Olaeta	Francisco		Purchasing Manager	(123)555-0100
7	Company G	Xie	Ming-Yang		Owner	(123)555-0100
8	Company H	Andersen	Elizabeth		Purchasing Representative	(123)555-0100
9	Company I	Mortensen	Sven		Purchasing Manager	(123)555-0100

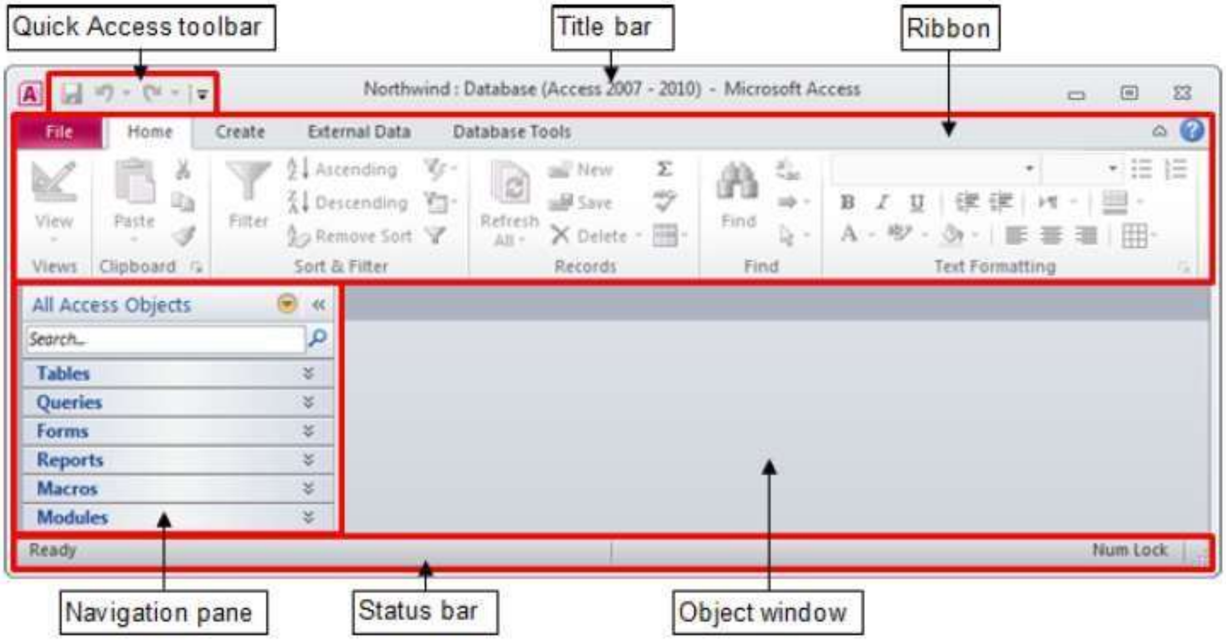
Microsoft Access.

Starting Ms. Access/Launching

- Click start button
- Select all programs, choose Microsoft office
- Click Ms. Access to open and create a new/**blank database**
- Under file name, state appropriate file name of your choice and finally click **create**



INTERFACE.



TABLES.

They are the basic unit of a database used for storage of data. They have two views, *design view* and *datasheet view.*

Datasheet view is a window that displays data from a table or query in row and column format where you can edit fields, add, delete data as well as search for data.

DATA TYPES.

A data type refers to the attribute of a field that determines the kind of data it can contain.

- Short Text – these are labels entries like names (ben, hellen etc)
- Number – are entries like 4,6,7,8,9 etc
- Long text/Memo – these are long text entries
- Date & time – it's for storing date and time
- Yes/No – are used when the entry is yes/no, true/false, on/off etc
- Hyperlink – connect the current object to another data source
- OLE object – is object linking, embedding and inserting object/graphic into the database
- Currency – used to hold currency values
- Auto number – for the automatic number entries of the computer e.g ID NO.
- Calculated – is a special data type used to display results of calculations or expressions based on other fields in the table. The results of a calculated field are read-only.
- *Lookup Wizard* is technically not a data type, but it does appear in the data type list because it easily enables you to create a drop-down list (combo box) of values

It is recommended that you use the new ***Attachment data type*** instead of the OLE Object data type.

N. B: Factors to Consider when designing a new database…. check out page 14

Definition of Terms

Primary key a.k.a. Key-Field.

It is a unique field which gives records their identity. In relationships, a primary key is used to refer to specific records in one table from another table.

Merits: Creates relationships between two or more tables for automatic association of data.

It is fast to find and open the data from different table with the primary key

Foreign key

The common field in child table that maintains relation with master table is foreign key.

What is Lookup Field?

A Field that stores a list from an existing table or query or a fixed set of values that is looked up.

TABLE RELATIONSHIPS.

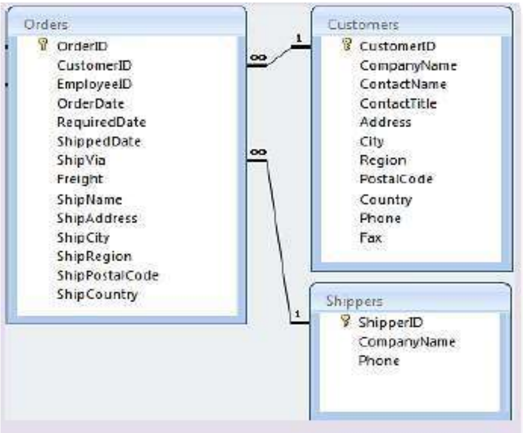
A relationship is a logical linkage between two entities that describe how the entities are associated with each other. There are three relationships possible between two tables namely: (***3 cardinalities***)

one-to-one

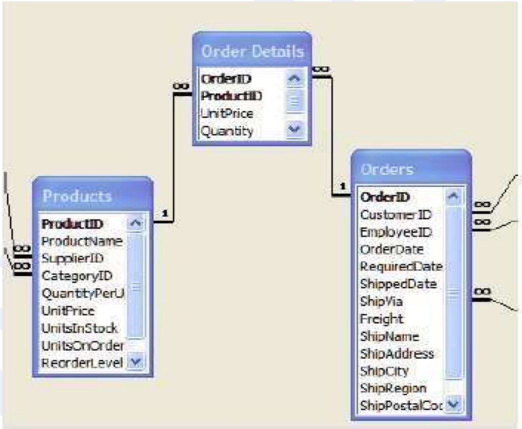
relation between two tables such that one primary key value corresponds to one foreign key value is known as one-to-one relationship, each record in parent table is related to one and only one record in child table.

one-to-many

relation between two tables such that one primary key value corresponds to many foreign key value is known as one-to-many relationship. In other words, each record in parent table is related to zero or many records in child table is one-to-many relationship.



1-many



many-many

many-to-many

relationship where many instances of an entity are associated with many instances of another entity is known as many-to-many relationship. Consider the enrollment relationship between a student and a class table. A single student may be enrolled in many classes and a single class may enroll many students. Also many students can do many subjects while many subjects can be done by many students

What is the meaning of following field properties?

Field Size:	Maximum number of character or value you can enter in field
Format:	Format for the field. It gives the layout for the field.
Input Mask:	creates a predefined structure into which the data for this field must be entered e.g. phone numbers
Caption:	The label for the field
Decimal Places:	The number of digits to the right of decimal place.
Default Value:	The value automatically entered for the new record.
Validation Rule:	An expression that limit the value to be entered in the field.
Validation Text:	The message to be displayed while user violates the rule when entering data.
Required:	The data should be entered the field can't be left blank.
Allow Zero Length:	Permits zero length of string
Indexed:	The field is to be indexed. Indexing speeds up the searching & sorting.

QUERIES

A query is a database feature that enables the user to select records, update tables and add new records to tables. A query is a question posed to tables and the return must meet a condition specified in the criteria section of the query

There are two types of queries

Select query: it is the default query, displays specific fields and data records from a table or tables based on the criteria set

Action query: are queries used to modify data in existing tables, types of action queries

Update query: is used to add data values to existing records

Append query: used to add new records to a table(s) from a dynaset

Delete query: it's used to remove records from a table (s) that meet some given criteria

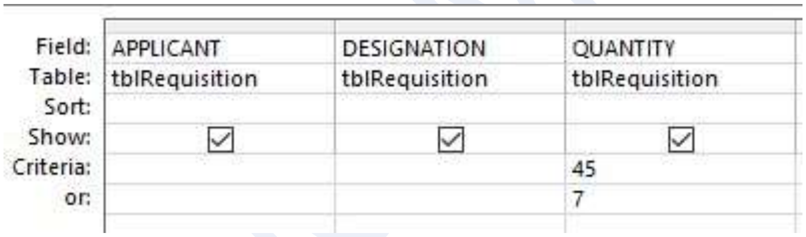
Make Table query: creates a new table and populates it with the results of the query.

Crosstab query: creates an interactive crosstab output based on the criteria set.

There are two categories of operators used in criteria expressions

1. ***Logical operators*** - AND, OR, NOT e.g. >60 AND <100, Like "kioko" OR "ben"
2. ***Relational operators*** - <, >, =, <=, <> (Not equal), e.g. <=150,

A **Wild card** is a placeholder for other characters, its used with text and date/time data type, e.g. **Like "J*"** ***Between** #4/7/08 **AND** #10/8/08* Practice use of **OR** under criteria e.g. 47 or 7



Field:	APPLICANT	DESIGNATION	QUANTITY
Table:	tblRequisition	tblRequisition	tblRequisition
Sort:			
Show:	☑	☑	☑
Criteria:			45
or:			7

You can add the clear grid command on the quick access toolbar to remove fields on the QBG

What is Dynaset?

A dynaset is a subset of the data stored in database that is displayed by the query.
The results returned by a query.

Examples of calculations in queries

NAMES: [FIRST NAME] &" "& [SECOND NAME]

PROFIT*: [Income] – ([salary] + [allowances])*

Using the *Total function (∑)*

Number of Profit: *Count([PROFIT])*

To Delete all records of people's names starting with letter **B*** use delete query

To display All people whose third character name is E: *Type ??E** under NAME criteria

To display all students except *Form 4* students: *Type < >Form 4* under class

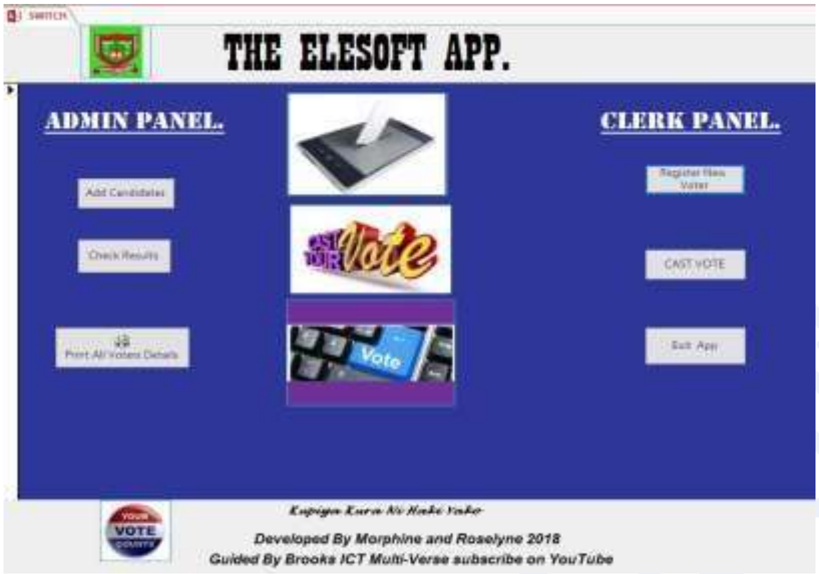
N.B: A **Parameter query** is a query that when run displays its own dialog box prompting you for information, such as criteria for retrieving records or a value you want to insert in a field. **(PAGE 16)**

FORMS

A form is a user interface used for viewing and making entries into a table more easily. It is designed using graphical objects called *controls*. They are found in the toolbox, some of the controls include the textbox and the label box. Controls are divided into *bound controls*, *unbound controls*, and *calculated controls* e.g. TOTAL BILL=SUM([BILL]), NUMBER OF PUPILS=COUNT([NAMES]).

Aa – Labels

Ab – Calculations



Kupiga Kura Ni Haki Yako

Developed By Morphine and Roselyne 2018

Guided By Brooks ICT Multi-Verse subscribe on YouTube

Ways of creating a new form.

- Using *AutoForm*: You can create a form based on a single table or query quickly using Autoform.
- Using Form *Wizard*: You can create a form based on multiple tables or queries using Form Wizard.
- *Design* View: You can create a custom form on your own in Design View.

Types of Forms

Basic forms, Datasheet forms, Navigation forms, Split forms, Modal dialog forms, and Multiple item forms

- **Basic Forms** display one record of data on a form at a time.
- **Multiple Item Forms** display multiple records of data at the same time. These are also referred to as Continuous Forms.
- **Datasheet Forms** display the data records in a datasheet view making the form look and function exactly like a table.
- **Split Forms/sub forms** are like a combination of a Basic Form and a Multiple Item Form all wrapped into one big form. On a Split Form, you can view a single record in its entirety and also view multiple detailed records of data at the same time. Split Forms are so named because you can view and interact with the data in two ways simultaneously.

What is a sub-form?

A sub form is a form contained within another form or a report. This feature is useful to represent one-to-many relationship between tables. In the form we can show the data of the Master table and at the same time in a second form we can show related data from the transaction table.

REPORTS

Reports are used for displaying records obtained from a query/table and to perform calculations. Types of reports include *tabular report, column report, mailing label report and totals report.* Create reports using the report wizard and print.

- **Report Header:** Controls in this section will display at the very top of the first page of a report.
- **Page Header:** Controls in this section will display at the very top of every page of a report. They will appear immediately after the Report Header on the first page of the report.
- **Group Header:** This section appears on a report only if you display the report data in a grouped fashion. Every grouping level of data will have a separate Group Header named after the control being grouped. Normally, this section will contain a textbox to display the field you are grouping the report data on. The data result will appear only once for each unique grouping section.
- **Detail section:** The detailed record-by-record data appear in this section of the report. If the report is being grouped, it will display every record within that group and then continue to the Group Footer section.
- **Group Footer:** This optional section appears on a report only if you display the report data in a grouped fashion. Every grouping level of data will have a separate Group Footer named after the control being grouped. Normally, this section will contain a calculated control to summarize the records in the Detail section. Often, records for each group will be counted, averaged, or totaled. After this section is displayed for one unique grouping, the report will return to the Group Header and then display the results for the next data grouping. The iteration over the Group Header, Detail section, and Group Footer continue until every unique data grouping is displayed.
- **Page Footer:** Controls in this section will display at the very bottom of every page of a report.
- **Report Footer:** Controls in this section will display on the last page of a report immediately after the last Detail section records and Group Footer results are displayed.

What are the different sections of a report?

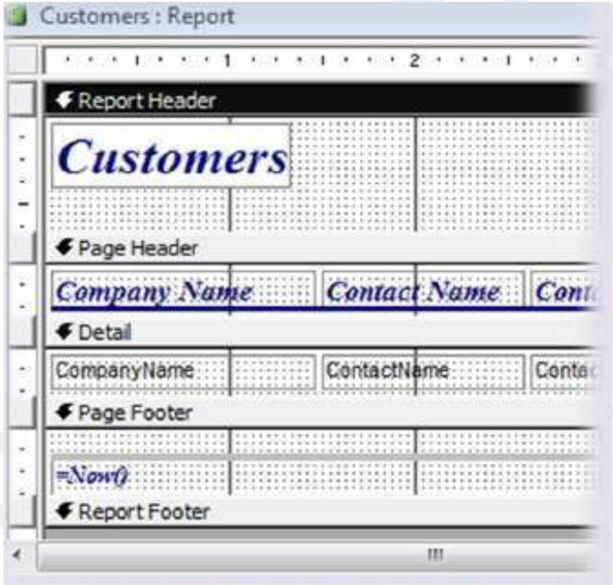


Figure 51: Different Sections in Report Design

N.B: To add calculations to reports or forms, add a *textbox* control in design view, type the expression on the control section written "unbound" starting with an equal sign on the footer

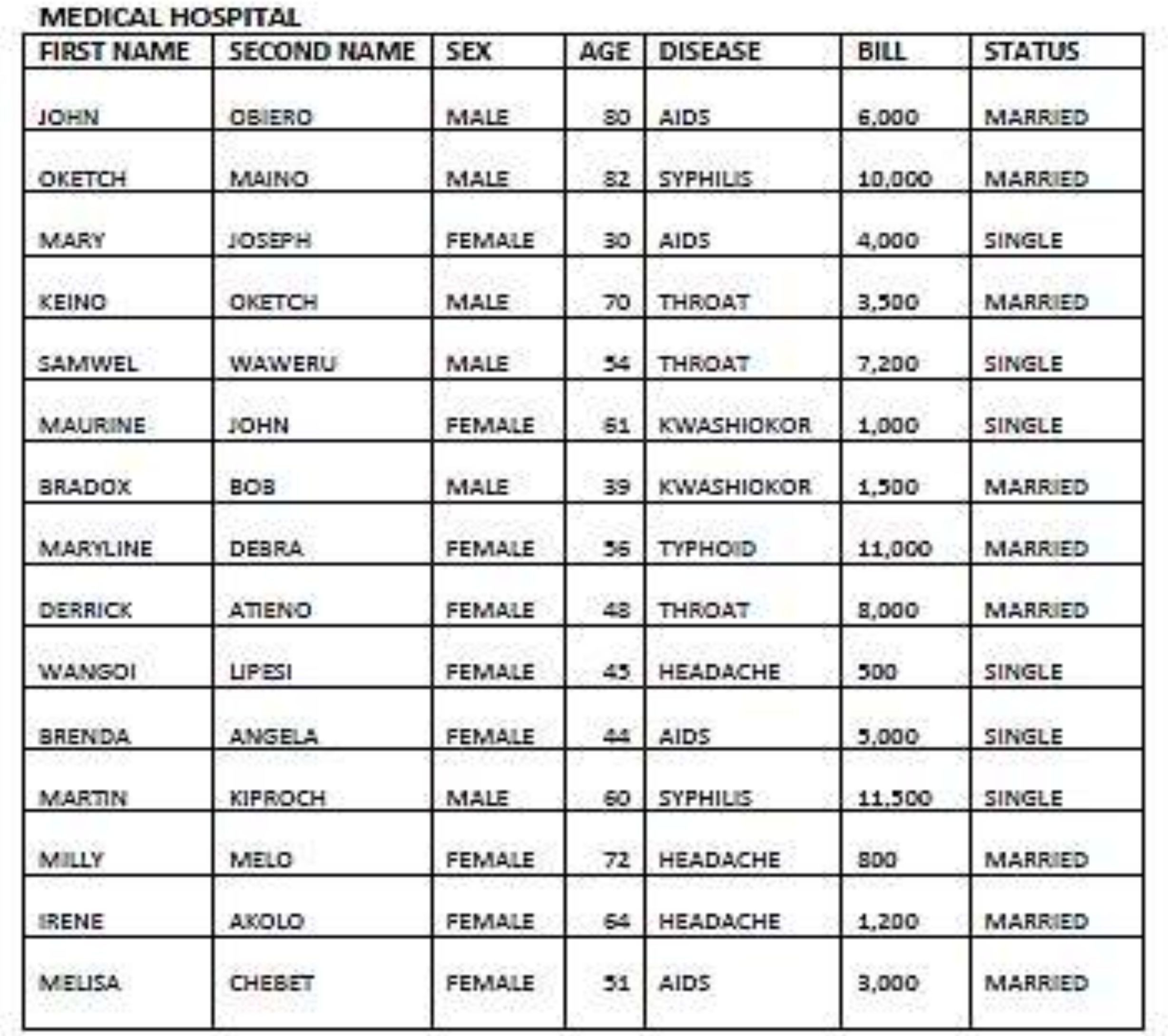
e.g. *Grand Total: Sum ([Amount Paid]),* *=Date( ),*

To find out number of all the records in the report place this on the footer in a text box

=**COUNT** (*)

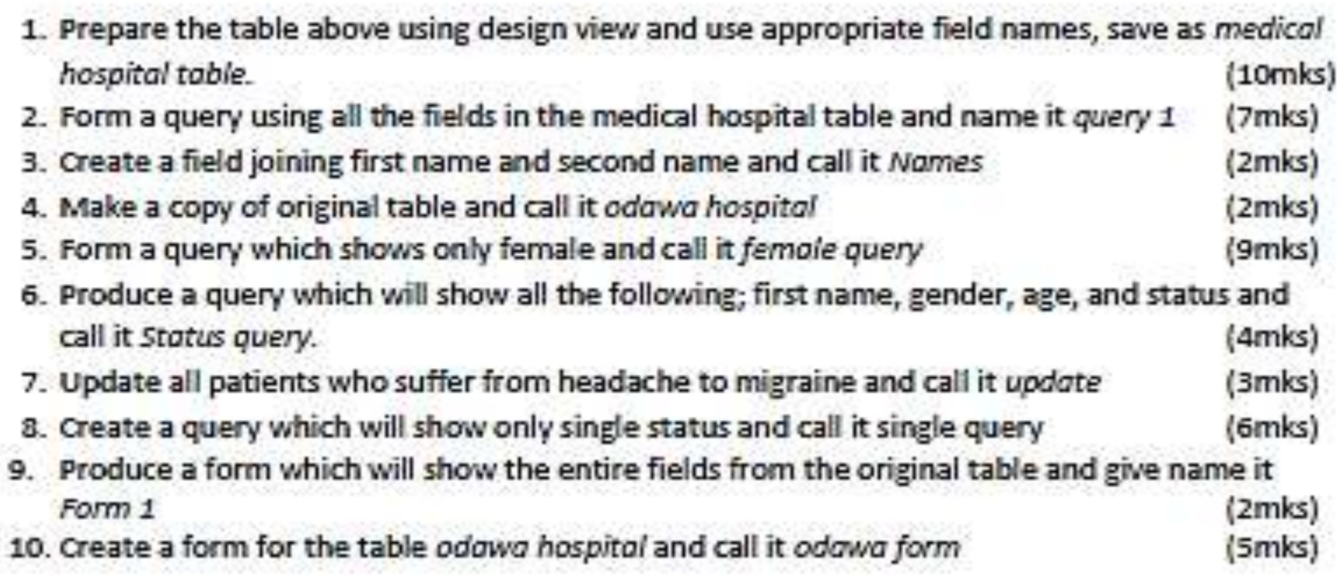
=**AVG**([AGE])

Practical Activity 1.



MEDICAL HOSPITAL

FIRST NAME	SECOND NAME	SEX	AGE	DISEASE	BILL	STATUS
JOHN	OBIERO	MALE	80	AIDS	6,000	MARRIED
OKETCH	MAINO	MALE	82	SYPHILIS	10,000	MARRIED
MARY	JOSEPH	FEMALE	30	AIDS	4,000	SINGLE
KEINO	OKETCH	MALE	70	THROAT	3,500	MARRIED
SAMWEL	WAWERU	MALE	54	THROAT	7,200	SINGLE
MAURINE	JOHN	FEMALE	61	KWASHIOKOR	1,000	SINGLE
BRADOX	BOB	MALE	39	KWASHIOKOR	1,500	MARRIED
MARYLINE	DEBRA	FEMALE	56	TYPHOID	11,000	MARRIED
DERRICK	ATIENO	FEMALE	48	THROAT	8,000	MARRIED
WANGOI	LIPESI	FEMALE	45	HEADACHE	500	SINGLE
BRENDA	ANGELA	FEMALE	44	AIDS	5,000	SINGLE
MARTIN	KIPROCH	MALE	60	SYPHILIS	11,500	SINGLE
MILLY	MELO	FEMALE	72	HEADACHE	800	MARRIED
IRENE	AKOLO	FEMALE	64	HEADACHE	1,200	MARRIED
MELISA	CHEBET	FEMALE	51	AIDS	3,000	MARRIED



1. Prepare the table above using design view and use appropriate field names, save as *medical hospital table.* (10mks)
2. Form a query using all the fields in the medical hospital table and name it *query 1* (7mks)
3. Create a field joining first name and second name and call it *Names* (2mks)
4. Make a copy of original table and call it *odawa hospital* (2mks)
5. Form a query which shows only female and call it *female query* (9mks)
6. Produce a query which will show all the following; first name, gender, age, and status and call it *Status query.* (4mks)
7. Update all patients who suffer from headache to migraine and call it *update* (3mks)
8. Create a query which will show only single status and call it single query (6mks)
9. Produce a form which will show the entire fields from the original table and give name it *Form 1* (2mks)
10. Create a form for the table *odawa hospital* and call it *odawa form* (5mks)

ADVANCED ACCESS - QUESTIONS.

How is "no nulls" constraint implemented for an attribute?

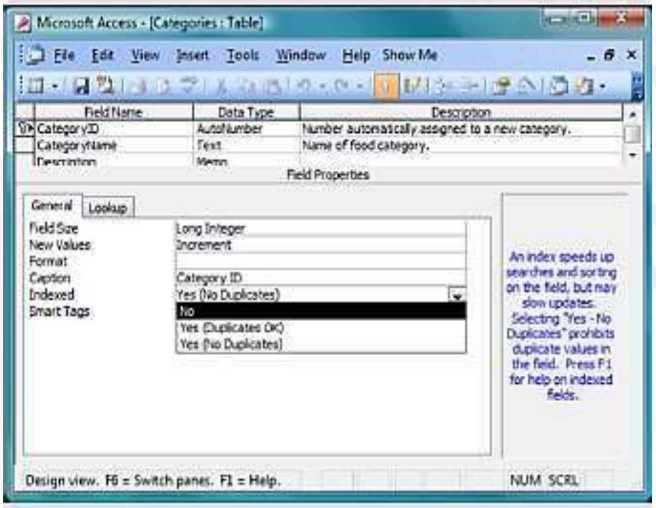
The not null (NN) concept allows the database designer to force the user to enter a value in a particular column. The not null concept is used for columns involved in a primary key. The option must be set when table is created, and set by using the NULL or NOT NULL keywords.

How is a "no changes" constraint implemented for an attribute?

The no change (NC) concept allows the database designer to prohibit changes to the values in a column. This concept is used mainly for columns participating in primary keys. Preventing changes to a primary key is recommended because primary keys are used to create relationship between tables, and changes to a primary key could result in a referential integrity violation.

How is a "no duplicates" constraint implemented for an attribute?

The no duplicate (ND) concept specifies that the values in a column must be unique. This concept is used in primary keys and alternate keys.



Microsoft Access - [Categories : Table]

Field Name	Data Type	Description
CategoryID	AutoNumber	Number automatically assigned to a new category.
CategoryName	Text	Name of food category.
Description	Memo	

Field Properties

Field Size	Long Integer
New Values	Increment
Format	
Caption	Category ID
Indexed	Yes (No Duplicates)
Smart Tags	

No
Yes (Duplicates OK)
Yes (No Duplicates)

An index speeds up searches and sorting on the field, but may slow updates. Selecting "Yes - No Duplicates" prohibits duplicate values in the field. Press F1 for help on indexed fields.

Design view. F6 = Switch panes. F1 = Help.

Figure 22: 'No Duplicates' constraints is implemented in Indexed properties in Access

What do the following formatting symbols mean in Format properties of fields?

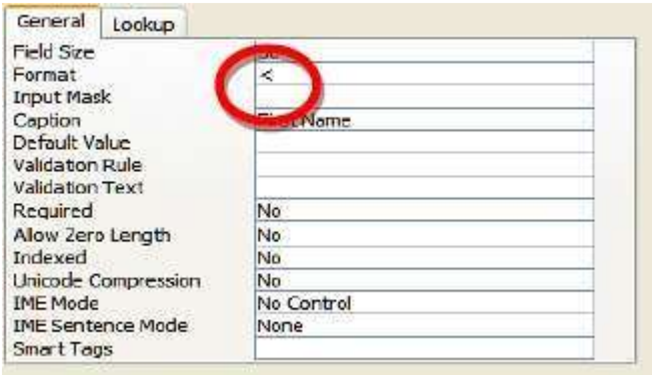
< Display contents in lower case

> Display contents in upper case

@"NULL" Displays NULL if left empty

@"Not entered" Displays 'Not entered' if field is empty

$ Right aligns text in column



Field Size	
Format	<
Input Mask	
Caption	First Name
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

How can you view the entire data of a memo field?

In datasheet view you can directly enter text but if you wish to view entire data of memo field you need to open the zoom window. To open zoom window press Shift+F2.

What is the effect of enabling 'Cascade Delete Related Fields' when defining relationship?

If cascade delete related fields option is enabled, Access automatically deletes related records in the child tables if the record from master table is deleted. For example, if a student is removed then all the records related to the student must be deleted from fees table, marks table and so on.

What is the effect of enabling 'Cascade Update Related Fields' when defining relationship?

If cascade update related fields option is enabled, Access automatically updates the key field value of related records in the child tables if the primary key field value from master table is updated. For example, if a student roll number is changed then all the roll numbers of that student must be updated in fees table, marks table and so on.

What are controls? How to use them? Give examples for control.

Controls are objects that can be placed in a form. The different controls are available in the Tool Box. After selecting the control in the toolbox it can be placed on the forms. After placing the control, we can set its properties to serve our request. Examples for controls are, textbox, label, List Box, Check Box, Option Buttons etc.

What is the advantage of Report Wizard over an Auto Report?

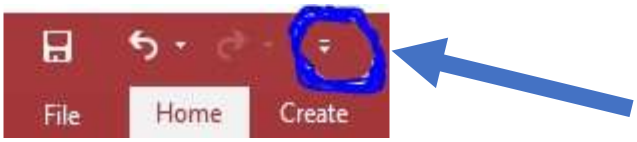
It takes a little more work to create a report with the report wizard than with the Auto Report but you have a lot more control over what the final result looks like. This is an advantage of Report Wizard over an Auto Report.

What is a macro?

A macro is a set of instructions, which can be executed repeatedly. it is useful for automating certain routine tasks like printing reports etc. The macro once created can be executed from buttons or any other controls.

Encrypting the database with a password

You can add more buttons to the Quick Access toolbar like the *switchboard manager* and the *Set Database password* commands



Click the quick access toolbar

Then click *More Commands* > quick Access > Commands Not on the Ribbon

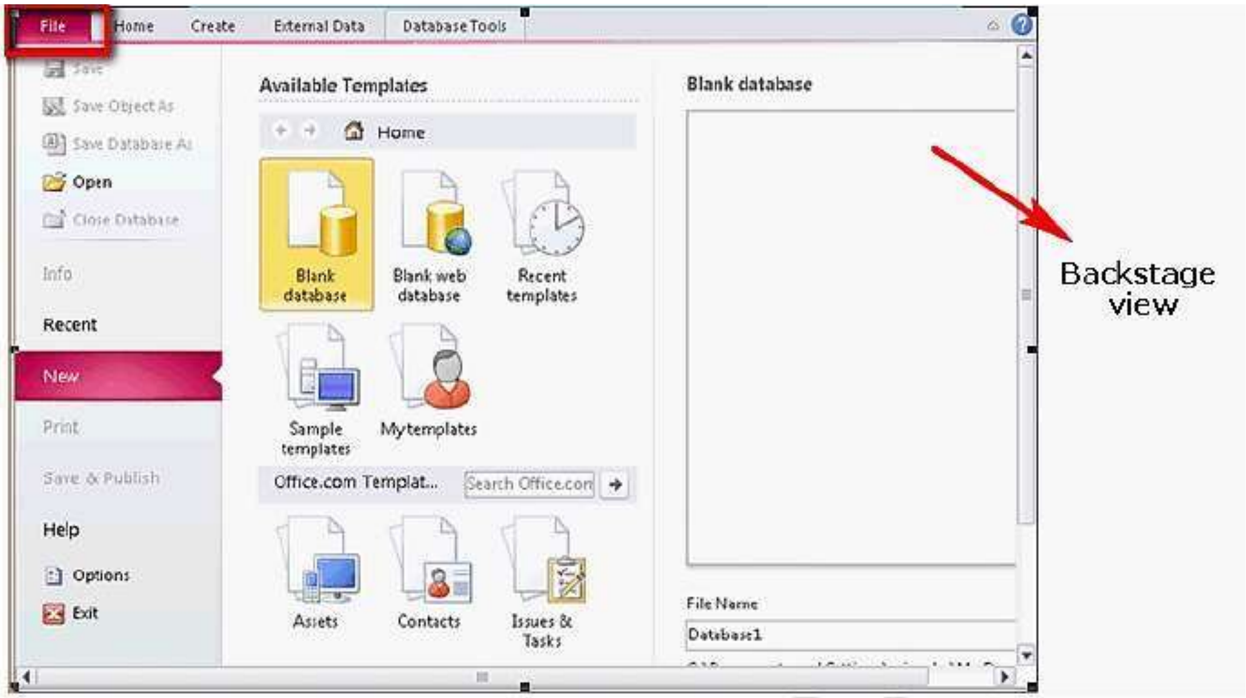
What is VBA?

Visual Basic is a stand-alone language that allows you to build complete applications in a windows environment. Visual Basic for Applications (VBA) is a hosted language, which means it runs within another application. VBA is most commonly used with the common office applications (Word, Excel, Powerpoint, Access), but it is also licensed to third party applications (WordPerfect, Corel Draw).

VBA is a subset of Visual Basic (VB). It is very similar to VB, but it does not contain all the functionalities of VB since it is hosted within another environment. The hosting environment (such as MS Word or Access) builds the foundation for using VBA

What is the backstage view in Access 2010?

The Backstage view is a full-screen user interface model in which commands related to managing MS Access and Access database are arranged in the form of buttons and pages. The
File tab in the upper-left corner of the program window displays the Backstage view. The Backstage view allows a user to create, save, and check hidden metadata or personal information. It also helps in discovering and using the features that are not related to the authoring features on the Ribbon.



- **You want to hide the Navigation Pane using a macro while working on Microsoft Office Access 2010. Show the steps**

  Click on the *Create tab*.

  Click the Macro button in the *Macros & Code group*.

  Click the Show All Actions button in the Show/Hide group on the *Design tab.*

  Select the *RunMenuCommand* option from the drop-down list.

  Select the *WindowHide* option from the Command box.

- **James works as a Database Designer for Tech Perfect Inc. He wants to prevent the Navigation Pane from appearing by default in Microsoft Office Access 2010**

  File > Options > Current database > Navigation

  N.B: You can also hide or show the navigation pane by pressing **F11**

  Object dependency in the *DATABASE TOOLS* tab can be used to view tables/quarries or other objects related to the Form

- **Factors to Consider when designing a new database.**

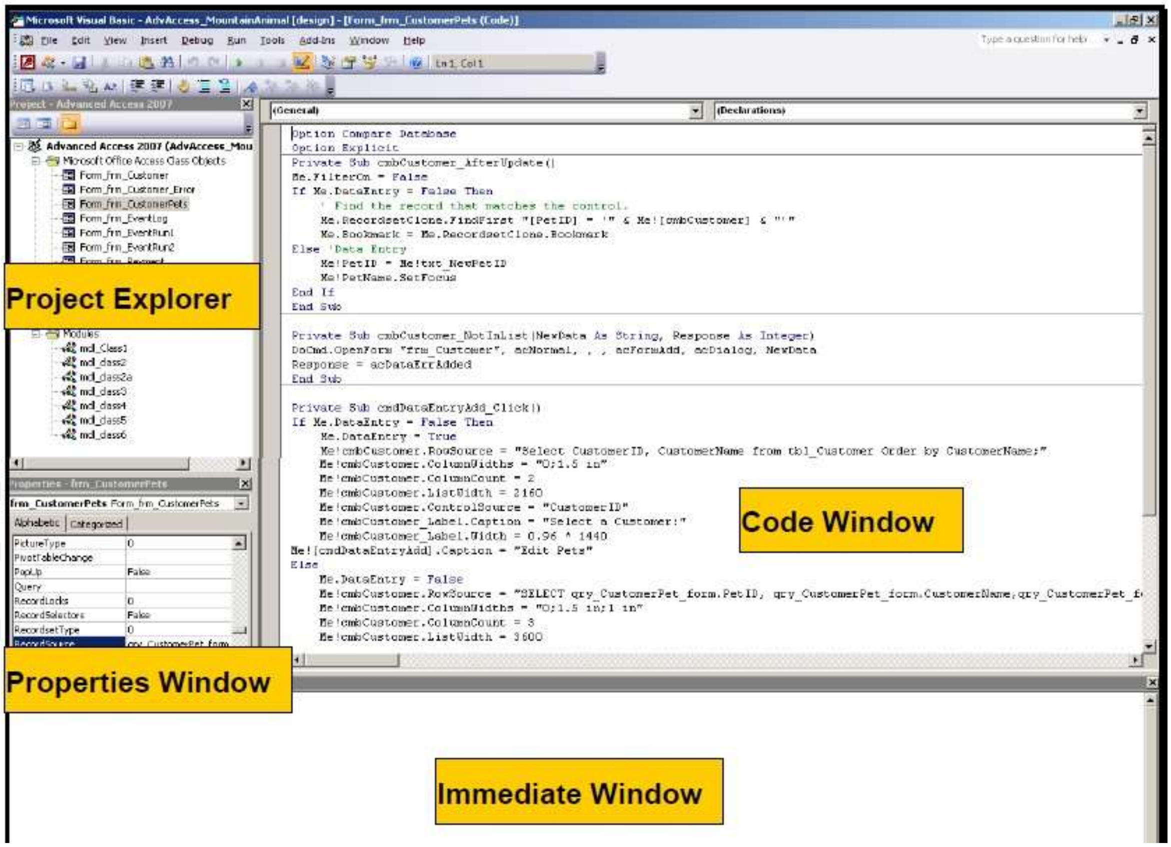
Designing Databases

The most basic object in a database is a table in which you store your data. The following guidelines will help you create efficient tables.

- **One table per subject:** The first rule in creating database tables is to ensure that every table stores data about one subject.
- **Give every table a primary key:** Another important consideration in creating a table is to assign a primary key. This field contains a value that sets a record apart from all other records in the table.
- **Include foreign keys:** A final thing you can do to make your tables more efficient is to include foreign keys. When a primary key from one table is stored in another table, it is called a foreign key. You can use foreign keys to create relationships between tables.

- **Nitty gritty details about databases.**
  - Opening existing databases.
  - How to open database objects.
  - How to use and switch between Design view and datasheet view
  - Closing database objects
  - Renaming database objects
  - Deleting database objects
  - Deleting Records

V.B.A Code Editor Window.



PictureType	0
PivotTableChange	
PopUp	False
Query	
RecordLocks	0
RecordSelectors	False
RecordsetType	0
RecordSource	qry_CustomerPet_form

```
Option Compare Database
Option Explicit
Private Sub cmbCustomer_AfterUpdate()
Me.FilterOn = False
If Me.DataEntry = False Then
    ' Find the record that matches the control.
    Me.RecordsetClone.FindFirst "[PetID] = '" & Me![cmbCustomer] & "'"
    Me.Bookmark = Me.RecordsetClone.Bookmark
Else 'Data Entry
    Me!PetID = Me!txt_NewPetID
    Me!PetName.SetFocus
End If
End Sub

Private Sub cmbCustomer_NotInList(NewData As String, Response As Integer)
DoCmd.OpenForm "frm_Customer", acNormal, , , acFormAdd, acDialog, NewData
Response = acDataErrAdded
End Sub

Private Sub cmdDataEntryAdd_Click()
If Me.DataEntry = False Then
    Me.DataEntry = True
    Me!cmbCustomer.RowSource = "Select CustomerID, CustomerName from tbl_Customer Order by CustomerName;"
    Me!cmbCustomer.ColumnWidths = "0;1.5 in"
    Me!cmbCustomer.ColumnCount = 2
    Me!cmbCustomer.ListWidth = 2160
    Me!cmbCustomer.ControlSource = "CustomerID"
    Me!cmbCustomer_Label.Caption = "Select a Customer:"
    Me!cmbCustomer_Label.Width = 0.96 * 1440
Me![cmdDataEntryAdd].Caption = "Edit Pets"
Else
    Me.DataEntry = False
    Me!cmbCustomer.RowSource = "SELECT qry_CustomerPet_form.PetID, qry_CustomerPet_form.CustomerName,qry_CustomerPet_fo
    Me!cmbCustomer.ColumnWidths = "0;1.5 in;1 in"
    Me!cmbCustomer.ColumnCount = 3
    Me!cmbCustomer.ListWidth = 3600
```

The *immediate* window is used to execute and view the output of VBA codes using the command *Debug.Print* when we press enter to execute. It is used to text codes that aren't associated with any database objects

WHAT IS A *SWITCHBOARD*?

A switchboard is simply a form with command button that open other switchboard or perform tasks such as opening forms and printing reports. Switchboard is a tool in Access to create a database application. You can use Switchboard Manager to create switchboards.

Switchboard is used for easier navigation between forms and reports within a database

Procedure for Adding a Switchboard.

Select the **File** ribbon in the upper left-hand corner of Access and click the **Options** button.

- Select **Customize Ribbon** in the list of categories that appears on the left-hand side of the Access Options window.
- Select **Database Tools** in the list of tabs on the right-hand pane, and click the **New Group** button at the bottom of the pane.
- The **New Group (Custom)** entry should now be created and selected automatically. Click the **Rename** button and rename it from **New Group** to **Switchboard**. Click **OK.**
- Above the left pane, select **Commands Not in the Ribbon** from the **Choose commands from:** drop-down list.
- In the left pane, scroll down the list and find **Switchboard Manager**. Click the **Add** button to add it to the newly created **Switchboard (Custom)** group. Click **OK**.

N.B:

Alternative Menu Systems

Instead of building a menu system using the Switchboard Manager wizard, you also have the option of creating a **custom menu system** *using a blank form* and adding to that form command buttons that open the database forms and reports. The advantages to this approach are that you have much greater design flexibility and you can have an unlimited number of items on the menu, compared with the maximum of eight items on a Switchboard menu. The downside to this approach is that it is not as easy as building a menu system with the Switchboard Manager wizard.

Another new way to create a menu system in Access is with a **Navigation Form.** These special forms have prebuilt templates with horizontal and vertical navigation buttons that are used to quickly access Forms and Reports. Navigation Forms will be briefly introduced in this chapter as an alternative to a Switchboard menu system.

Procedure for Adding Items to the Switchboard

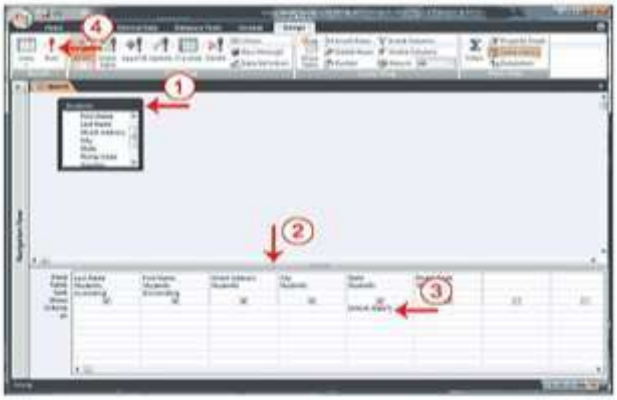
- ✓ Click the switchboard manager button (from Quick Access toolbar or the Database Tools Tab)
- ✓ Access will inform you that no switchboard was found, do you want to create a new one? Click *YES*
- ✓ The manager will open with a new default switchboard highlighted, click *EDIT* button
- ✓ On the edit switchboard page, click *New*
- ✓ In the edit switchboard item page, give your switchboard button a label
- ✓ Back at the edit switchboard page, click new and repeat the process to add other forms and reports etc.

QUERY PARAMETERS.

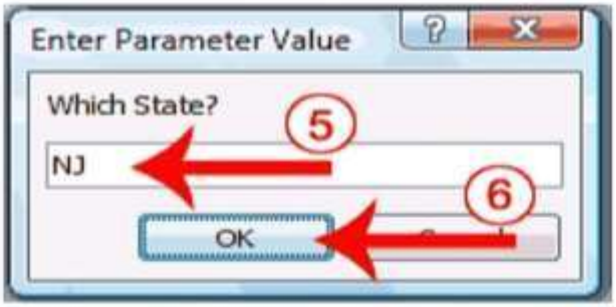
If the conditions of the query should be entered at the time of execution of query, we specify the query parameter instead of criteria value or constant. Query parameters are specified inside large braces **[ ]** Thus, when the query is run, the values of parameters are asked and accordingly the records will be presented.

To set query parameter, type the message inside large brackets in the criteria box of required field. To specify data type of parameter you can select *Parameter option from Query menu*. In the parameter column, type the message entered in Criteria box then, selects the type of the value that the user has to enter when query is executed in the Data Type column from the drop-down list.

To create a parameter query:



1. Open a table or query in Query Design view.
2. Create your query.
3. On the Criteria line, type the prompt within square brackets.
4. Click the Run button. Access prompts you.


Enter Parameter Value

Which State?

NJ

OK

Note: If you want to make your user prompt more flexible, use one of the following formats.

Like "*" & [*Prompt*] & "*"	Returns all records that contain the value you enter. Example: If you enter **ad,** Access returns all records that include the sequence ***ad*** anywhere in the field.
Like "*" & [*Prompt*]	Returns all records that end with the value you enter. Example: If you enter **S,** Access returns all records that end with *S*.
Like [*Prompt*] & "*"	Returns all records that begin with the value you enter. Example: If you enter **S,** Access returns all records that begin with *S*.
> [Prompt]	Find all records with a value greater than the value you enter. Example: If you enter

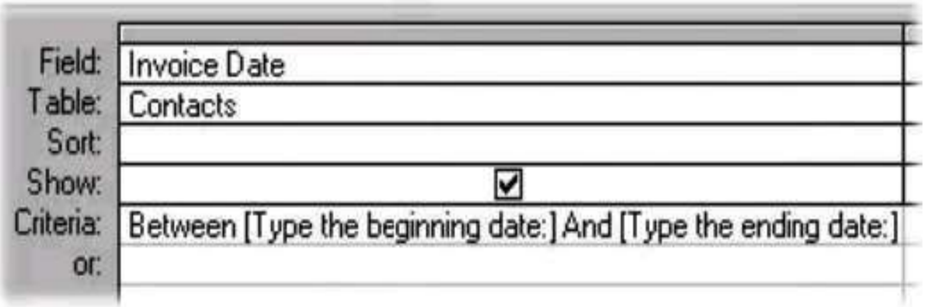
✓ Don'ts in Query Parameters.

When you are designing a query parameter, make sure the [prompts] is not exactly the same as one of your field names. E.g. if you have a field called Last Name then such a [Last Name] wouldn't work. Solution is to turn it into a question by adding a question mark [Last Name?].

Don't use illegal characters inside your parameter prompts like **! . & [ ]**

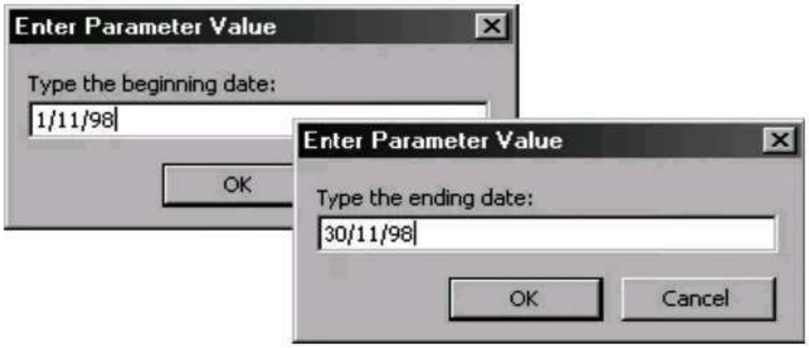
Don't write too much, you are allowed only 40 to 50 characters……anything else just cuts off.

✓ Using Multiple Parameter query.



Field: Invoice Date
Table: Contacts
Sort:
Show: ☑
Criteria: Between [Type the beginning date:] And [Type the ending date:]
or:

In this example the query would display all the record which contained dates in the range *1 November 1998 - 30 November 1998* in the *Invoice Date* field...



Enter Parameter Value
Type the beginning date:
1/11/98
OK

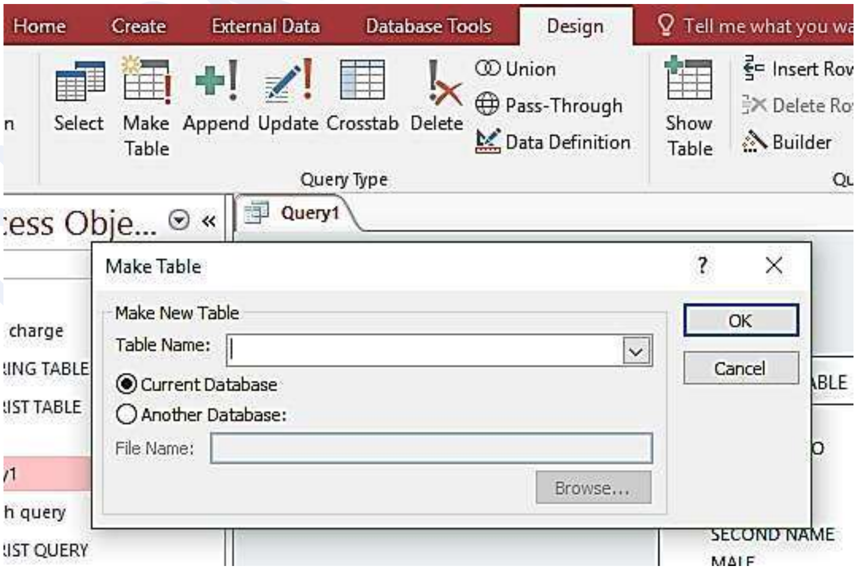
Enter Parameter Value
Type the ending date:
30/11/98
OK
Cancel

✓ Using a Query to Make a Table.

You can use a query to create a table, this is useful when you want to create a new table that includes the fields and data from an existing table

- Open the query on which you want to base your new table on in design view
- Enter all the criteria you desire then finally click *Make table* command, give the new table a name in the current database

N.B: Always run the query to update the table in case you have changes



Make Table
Make New Table
Table Name:
● Current Database
○ Another Database:
File Name:
Browse...
OK
Cancel

Domain (Functions)

The domain functions are like the grouping functions you use in a totals query. These functions take a range of records, and then perform a calculation or lookup to arrive at a single value Access includes eight domain functions:

- **DSum** calculates the sum of multiple values. You can use it to calculate an order's total price.
- **DAvg** calculates the average of multiple values. You can use it to calculate a product's average price.
- **DCount** counts the number of matching records. You can use it to count the number of items in an order, or the number of orders a customer made.
- **DMin** and **DMax** find the smallest or largest value in a series. You can use these functions to find bargainbasement or top-priced products.
- **DFirst** and **DLast** extract the first or last value in a series. If you sort a list of orders by date, you can get the oldest or most recent order.
- **DLookup** finds a value that meets specific criteria. You can use it to hunt through a table and find the product name for a given product ID.

All the domain functions take the same three parameters. The first parameter is the field (or calculated expression) you want to retrieve or use in your calculation. The second parameter is the table or query you're using. The third parameter contains any filter conditions you want to use to narrow down the number of rows. If you're trying to find the average price of all the beverages sold by Boutique Fudge, you use the Price field (as the first parameter), the Products table (the second parameter), and use the Beverages category (the third parameter) as a filter to include just the products you want.

SQL and how it works

SQL means *Structured Query Language*. Behind a query is always an SQL-statement that specifies what to compute. Behind the scenes, every query is actually a text command written in a specialized language called SQL which is a staple of the database world, and it's supported in all major database products, albeit with minor variations. *SELECT, FROM, WHERE*

SELECT <the fields to show in the result> FROM <one or more tables> e.g.

SELECT [tblSecurityLevel].[SecurityID], [tblSecurityLevel].[SecurityLevel] FROM tblSecurityLevel;

Data Dictionary.

All definitions of elements in the system are described in detail in a Data dictionary.

The elements of the system that are defined are: Dataflow, Processes, and Data stores.

If a database administrator wants to know the definition of a data item name or the content of a particular dataflow, the information should be available in the dictionary.

Data Bank.

A **Data Bank** can be defined as a collection of data, usually for several users, and available to several organizations. A Data Bank is therefore, a collection of databases.

Notes.

- The Database is organizational, while a Data Bank is multi-organizational in use.
- The Database & the Data Bank have similar construction and purpose. The only difference is that, the term **Data Bank** is used to describe a larger capacity base, whose contents are mostly of historical references (i.e., the Data Bank forms the basis for data or information that is usually generated periodically). On the other hand, the contents of the Database are used frequently to generate information that influences the decisions of the concerned organization.

Relationships Settings

Once the database primary keys are set properly and compatible data types have been assigned to the primary and foreign keys, you are ready to set the database relationships. There are three additional options you can enable when establishing the database relationships.

- **Referential Integrity:** It is generally a good idea to enforce referential integrity for the relationships in the database. This ensures that foreign key data can be added to a table only if a related primary key already exists. Enforcing this option also prevents "orphaned data" from appearing in your database. Orphaned data are records that exist in the database but are not related to any other data. Because of this, orphaned data are considered "bad" data and should be avoided.
- **Cascade Update:** If referential integrity is enforced, you have the option to also enable Cascade Update for the relationship. It is usually a good idea to enable Cascade Update. This option will automatically update any foreign key values when the primary key field is updated. In other words, it ensures that the data records remain linked together across multiple tables, even if you update the primary key value in the table.
- **Cascade Delete:** If referential integrity is enforced, you have the option to enable Cascade Delete for the relationship. It is not necessarily a good or bad idea to use this option. Instead, you have to decide whether it is something you want for your database system. When enabled, Cascade Delete will automatically delete related foreign key records from the database when you delete a primary key record. If you have multiple tables and relationships with Cascade Delete enabled, it becomes very easy to quickly delete numerous related records from the database. Again, this powerful option should be considered carefully when deciding whether or not to enable it.

Types of Quaries on the Wizard.

- Simple Query Wizard gets you started with an ordinary query, which displays a subset of data from a table. This query is the kind you created in the previous section.
- Crosstab Query Wizard generates a crosstab query, which lets you summarize large amounts of data using different calculations.
- Find Duplicates Query Wizard is similar to the Simple Query Wizard, except it adds a filter expression that shows only records that share duplicated values. If you forgot to set a primary key or to create a unique index for your table, this option can help you clean up the mess.
- Find Unmatched Query Wizard is similar to the Simple Query Wizard, except it adds a filter expression that finds unlinked records in related tables. You could use this to find an order that isn't associated with any particular customer.

Relationships vs. Joins

It's important to understand the differences between a relationship and a query join:

Relationship. A permanent link between two tables, which is stored in your database. When creating a relationship in the database, you have the option of switching on referential integrity, a set of rules that prevents inconsistent data in related tables.

Join. A query feature that lets you combine related data from two tables into one set of results. The join doesn't affect how you enter or edit that information in the underlying tables. If you have a relationship in place, Access assumes you want to use a join to link those tables together in a query, which only makes sense.

If you add two unrelated tables, Access tries to help you out by guessing a relationship. If it spots a field with the same data type and the same name in both tables, it adds a join on this field. This action often isn't what you want

Tips

- ✓ You can add extra tables to a query without using the Show Table window. All you need to do is drag the table from the navigation pane and drop it on the query design surface.
- ✓ If you use this calculation with the date January 10, 2013, the new date becomes January 24, 2013.

  ExtendedDeadline: [DueDate] + 14

- ✓ Using subtraction, you can find the number of days between any two dates. Here's how you calculate how long it was between the time an order was placed and when it was shipped:

  ShippingLag: [ShipDate] - [OrderDate]

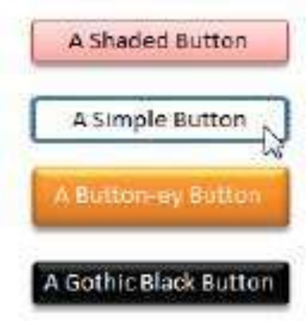
  If the ship date occurred 12 days after the order date, you'd see a value of 12.

- ✓ You can even use the ampersand to tack text alongside numeric values. If you want the slightly useless text "The price is" to appear before each price value, use this calculated field:

  Price: "The price is: " & [Price]

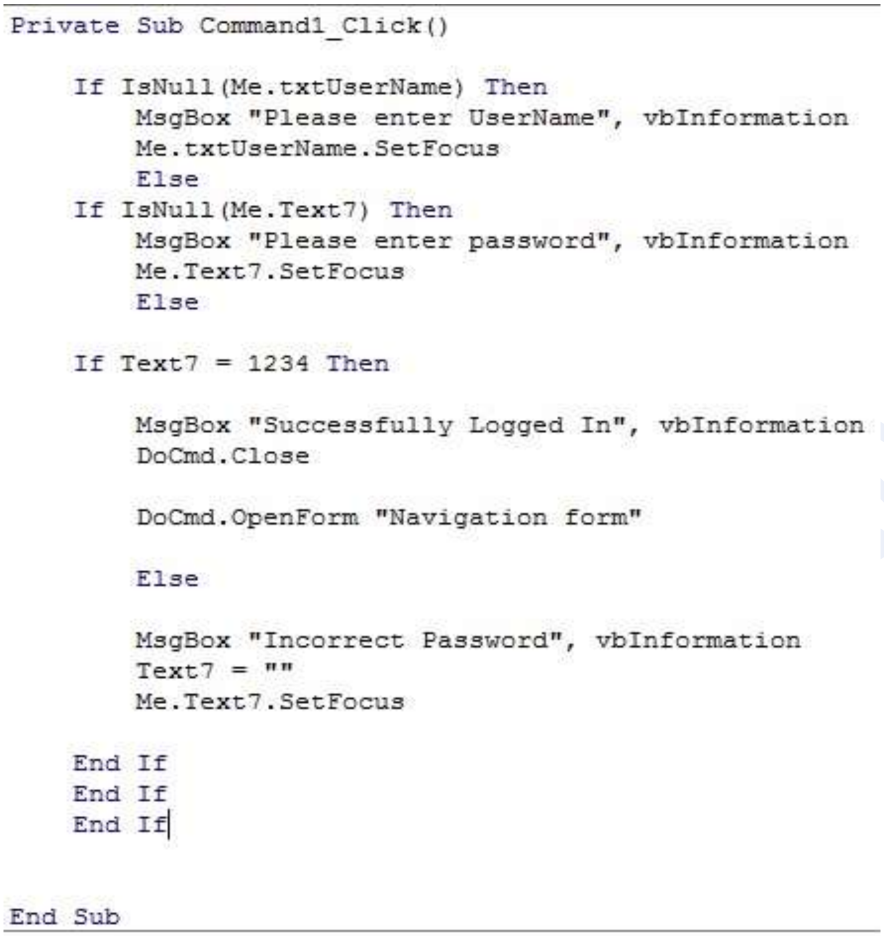
- ✓ **Showing Web Pages with the Web Browser**

  Access includes a Web Browser control that lets you embed a web browser window in a form. This way, you can display web pages that are in some way related to your record data.
- ✓ **Command** buttons can be formatted select it and pick a fancy style from the Form Design Tools | Format→Control Formatting→Quick Styles menu.

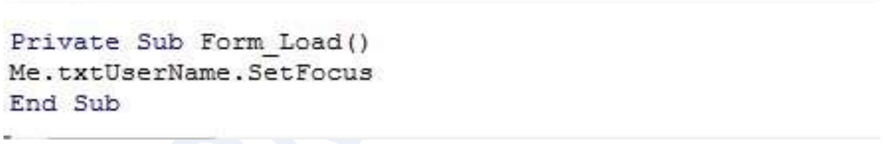


How To Create A Login Dialog Box For Single-User Databases.

Click *Create Tab*>More Forms>*Modal Dialog*
Attach the following code to OK button on the form, ensure the form is loaded on *startup*



```
Private Sub Command1_Click()

    If IsNull(Me.txtUserName) Then
        MsgBox "Please enter UserName", vbInformation
        Me.txtUserName.SetFocus
        Else
    If IsNull(Me.Text7) Then
        MsgBox "Please enter password", vbInformation
        Me.Text7.SetFocus
        Else

    If Text7 = 1234 Then

        MsgBox "Successfully Logged In", vbInformation
        DoCmd.Close

        DoCmd.OpenForm "Navigation form"

        Else

        MsgBox "Incorrect Password", vbInformation
        Text7 = ""
        Me.Text7.SetFocus

    End If
    End If
    End If


End Sub
```



```
Private Sub Form_Load()
Me.txtUserName.SetFocus
End Sub
```

```
Private Sub Form_Close()

    If Text7 = 1234 Then
            DoCmd.OpenForm "Navigation form"
    Else
            DoCmd.CloseDatabase
    End If
End Sub
```

Text 7	*Properties*	Under	Format	******
			Input mask	password
Exit	*Macro*		Choose	*Quit Access*

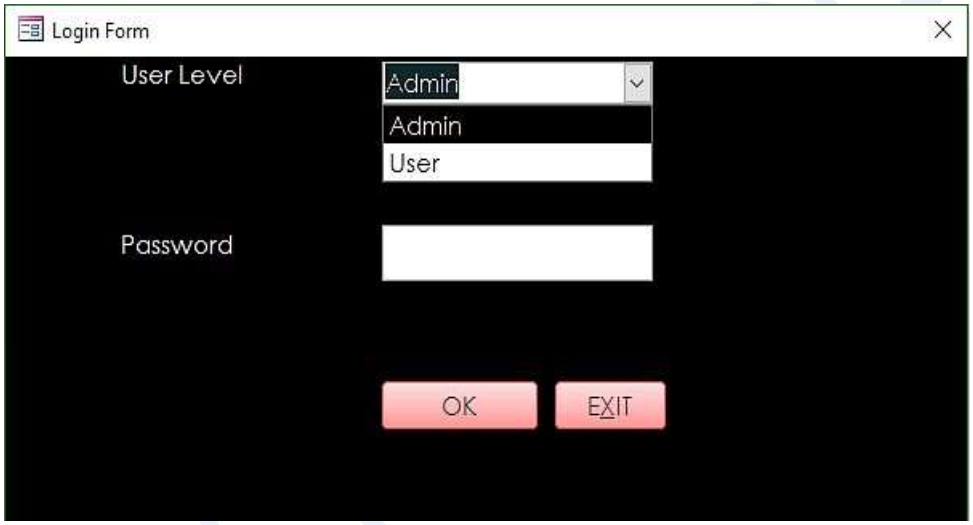
N.B: Login forms based on look up fields from tables are much more secured and can be used for multi-user databases, otherwise Encrypting the whole database with a password is the most recommended security option over the single user login form.

CREATING LOOKUP FIELDS

Creating lookup fields can help improve the efficiency of data entry process in a database, a lookup field can display a user-friendly value that is bound to another value in the source data table or value list.

a) **Using a Lookup list:** the field gets its data from an existing table/query *(bound)*

- Open the table in datasheet view and click the column header
- Click the *Lookup & Relationship* datatype
- In the L&R select *I want the lookup field to get value from a table/query*
- Select the field you require, leave the *Hide key column* check box selected
- Name the field. *Allow multiple values* must be selected to enable storing multiple values



b) **Using a lookup value list:** the field gets its data from a list of values that you type in when you create the field. *(ideal for constant data)*

- Open the table in design view
- Click the cell in the data type that corresponds to the column that you want to define as a lookup field, click down arrow and select a data type and then click *Lookup wizard*
- In the lookup wizard, select *I will type in the values I want,* next select no. of columns you want, type the values you want displayed in the field and click next
- If you selected more fields then you have to choose which one becomes the unique identifier
- Next type a label name for the field
- If you are creating a multi-value lookup, then select *Allow Multiple Values* - Click finish and click Yes to change the changes.

N.B: You can change the filed properties of the lookup field in design view using the lookup table

Using the DLookUp Function.

It is used to get values of a particular field from a specified set of records. This function can be used in a *VBA* module, a *query expression, a macro,* or even a calculated *control* on a form or report.

Syntax.

DLookUp(*exp, domain, criteria*)

e.g. =DLookUp("[Password]","[tblSecurityLevel]","[SecurityID]=1")

Where *[Password]* is the field whose value we are looking

[tblSecurityLevel] is the name of my table/domain

[SecurityID]=1 is my criteria, I want the password for user ID 1

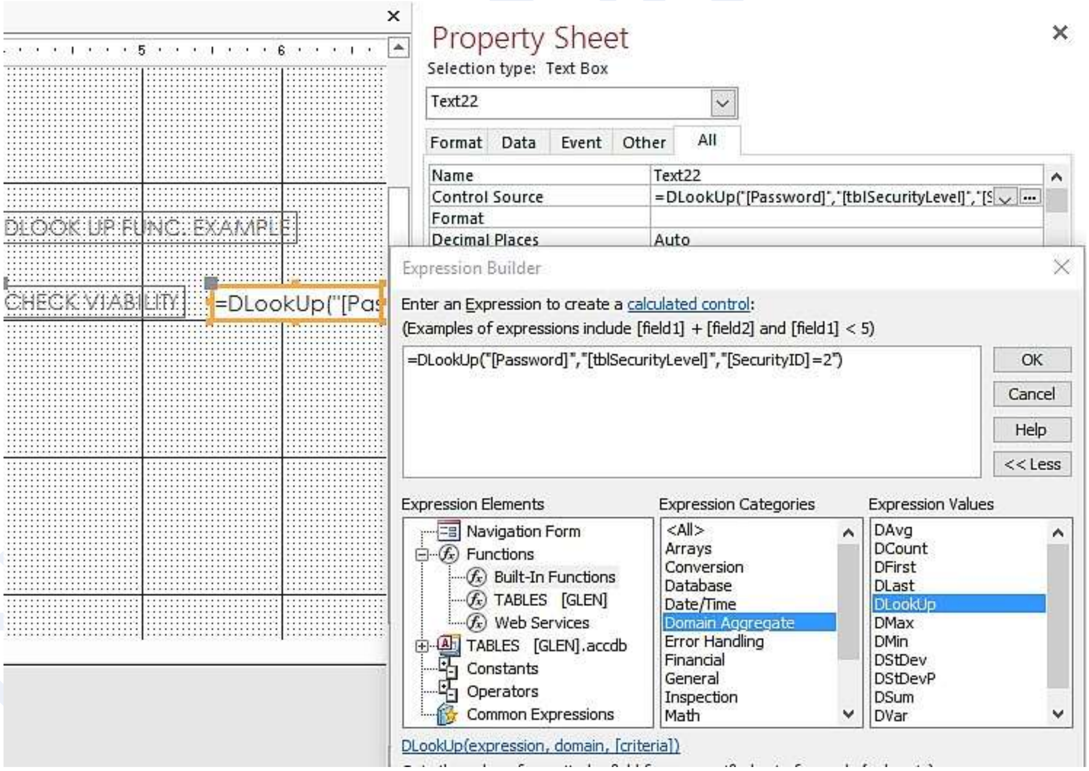
Expression: identifies the field whose value you want to return, you can include name of a field in a table, form control, a constant or even another inbuilt function.

Domain: a string expression identifying the object

Criteria: used to restrict the range of data

The DlookUp function returns a single field value based on the criteria, if not supplied or met, the function returns a NULL.

It can be used in a calculated field in a Query (*under criteria row),* Form controls (*type it under control source).*



Property Sheet

Selection type: Text Box

Text22

Name	Text22
Control Source	=DLookUp("[Password]","[tblSecurityLevel]","[S
Format	
Decimal Places	Auto

Expression Builder

Enter an Expression to create a calculated control:

(Examples of expressions include [field1] + [field2] and [field1] < 5)

```
=DLookUp("[Password]","[tblSecurityLevel]","[SecurityID]=2")
```

DLookUp(expression, domain, [criteria])

Developer Secret ☺

When you Hold down the **SHIFT KEY** while opening your secured database, you can **BYPASS** all the security code set during login.

Example Code if used in V.B.A

```
Private Sub Command42_Click()
Dim x As Integer x = DLookup("[Password]",
    "tblSecurityLevel")

              If x = Text40 Then

                        MsgBox "Correct Password"

              Else

                        MsgBox "Invalid Password"

              End If

End Sub
```

OR

Me.Text40.Value=Dlookup("password", "tblSecurityLevel", "SecurityID=1")

N.B: ***Me.Text40.Value=DCount("phoneNumber", "tblContacts", "FirstName= 'ben'")***

THE IIf FUNCTION.

This function returns one of two parts of an expression depending on the evaluation. Can be used with any builder, *Expression builder*, *Code*, and also in a *Query Syntax:*

IIf(expr, truepart, falsepart)

Example:

Suppose you have a *Customer table* that contains a field named *CountryRegion* having details of countries where customers come from.
In a Form you want to display whether Luo is the first language of the Customer if he comes from Kenya, you can add a control and use IIf in its Control Source property

=IIf([CountryRegion]="Kenya", "Luo", "Non-Luo")

Nested IIf Expressions.

=IIf([CountryRegion]="Kenya", "Luo", IIf([CountryRegion]="France", "French", "American"))

=IIf([Due Date]<Date(), "OVERDUE", IIf([Due Date]=Date(), "Due Today","Not Yet Due"))

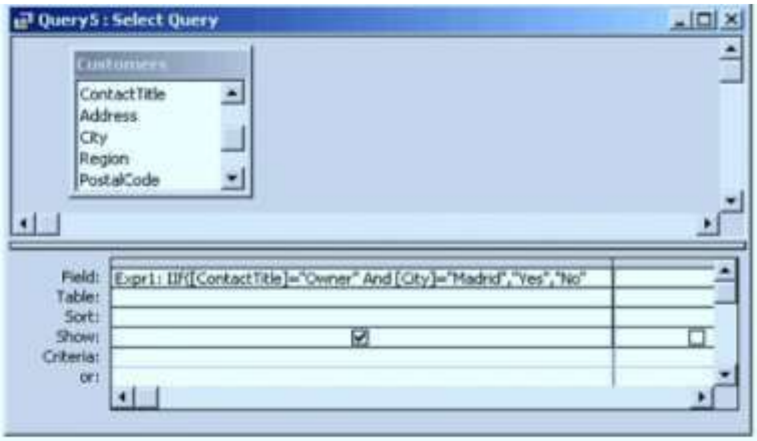
=IIf([PurchaseDate]<#1/1/2016#, "Old", "New")

=IIf([Average]>90,"A",IIf([Average]>=80,"B","C"))

Use of IIf function in a Query.

Frequently used to create calculated fields

Language:IIf(CountryRegion]=”Kenya”, “Luo”, “Non-Luo”)

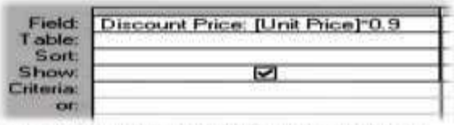


Field: Expr1: IIf([ContactTitle]="Owner" And [City]="Madrid","Yes","No")

In the example above, the iif function will return "Yes" if both the ContactTitle = "Owner" and City = "Madrid". If one or both of these conditions is not met, it will return "No".

HOW TO FORMAT A DYNASET.

Calculating with Fields and Constants



Field: Discount Price: [Unit Price]*0.9

Discount Price: [Unit Price]*0.9

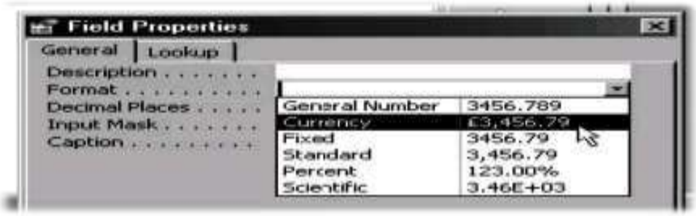
In this example the query creates a new field called ***Discount Price*** and displays in it the value in the ***Unit Price*** field multiplied by ***0.9***.

Note: This is the same as multiplying by 90% or subtracting 10% but the query does not recognise the % symbol and we have to devise an alternative expression.

Formatting the New Field

When you are performing calculations, you may wish to see your results formatted in a particular way - currency for example. Normally field formatting is specified in a the design of a table, but as the field is being created by the query there is another way...

In the query's design view, right-click anywhere in the new field's column and choose **Properties...** from the shortcut menu (or click somewhere in the new field and choose **View > Properties** from the menu). This opens the **Field Properties** dialog box...

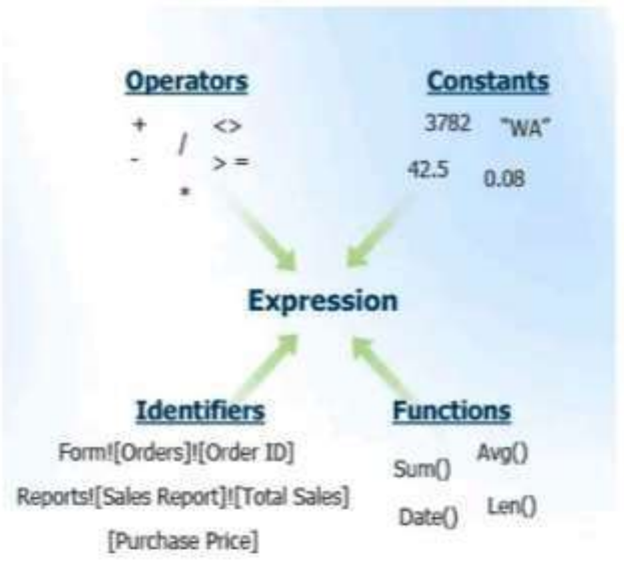


General Number	3456.789
Currency	£3,456.79
Fixed	3456.79
Standard	3,456.79
Percent	123.00%
Scientific	3.46E+03

What does the term *EXPRESSION* mean in Ms. Access?

Expressions consist of *Operators, Functions, Constants,* and *Identifiers (*for example, the names of fields, tables, forms etc.).
You can create expressions using the ***Expression Builder*** or you can select an expression from the many ***inbuilt*** expressions in Access. Keyboard shortcut for launching the expression builder after selecting a *Control* is (CTRL + F2) in Access_2016.

Mostly expressions are used in *control source* and *default value* properties for a control, you can use expressions to calculate values, validate data, and even set default values for fields or controls.



For example, the following expression contains all four of these elements:

=Sum([Purchase Price])*0.08

=DCount("[Retired]","[Assets]","[Retired]=Yes")	Uses the **DCount** function to return the number of Yes values in the Retired field (a Yes/No field) in the Assets table.

Example.

Expressions that refer to *values* or *controls* in other Access objects can be written like this

=Forms! [Orders] ! [OrderNo]

It shows the value of the *OrderNo* control on the form called *Orders*

=Forms! [Customers] ! [Commission] * 1.06

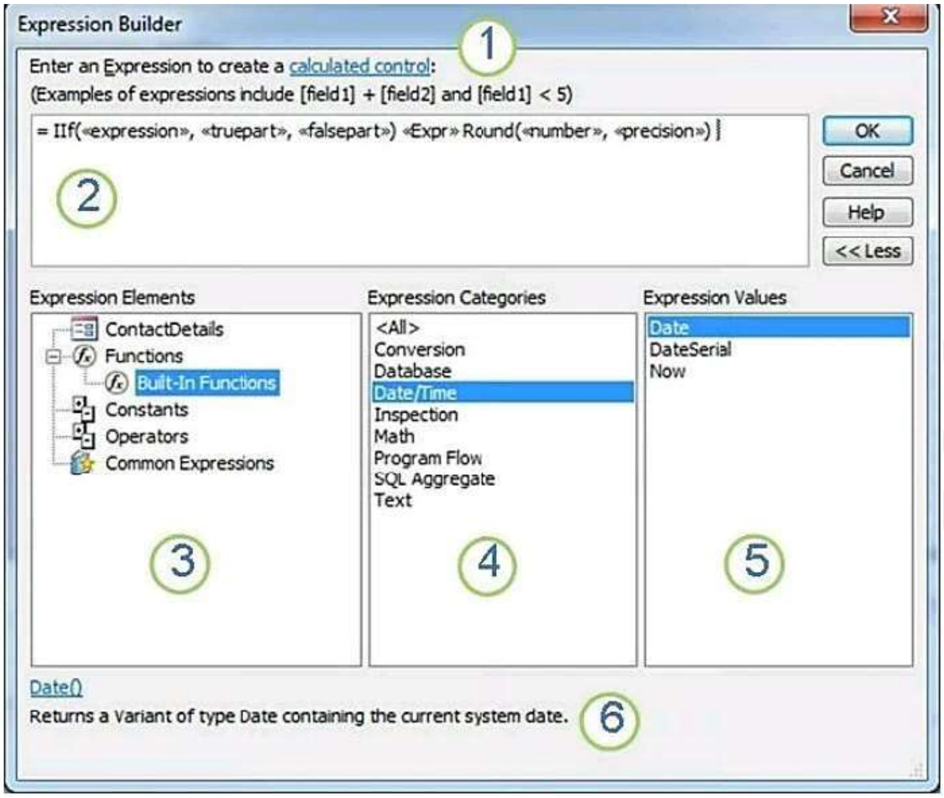
It increases the value of commission in the Customers form by *6%*

Expressions can aslo be formed in the footer area of a Report

e.g. **=”Page” &[Page]** or =**[Page]**

Check effect under print preview.

The Expression Builder.



Enter an Expression to create a calculated control:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

```
= IIf(«expression», «truepart», «falsepart») «Expr» Round(«number», «precision»)
```

Date()
Returns a Variant of type Date containing the current system date.

1. **Instructions and Help link** Get information about the context in which you are entering the expression.

2. **Expression box** Type your expression here, or add expression elements by double-clicking items in the element lists below.

> NOTE: If you do not see the element lists (items 3, 4, and 5), click **More >>** to the right of the Expression box.

3. **Expression Elements list** Click an element type to view its categories in the Expression Categories list.

4. **Expression Categories list** Click a category to view its values in the Expression Values list. If there are no values in the Expression Values list, double-click the category item to add it to the Expression box.

5. **Expression Values list** Double-click a value to add it to the Expression box.

6. **Help and information about the selected expression value** If available, click the link to see a Help article about the selected expression value.

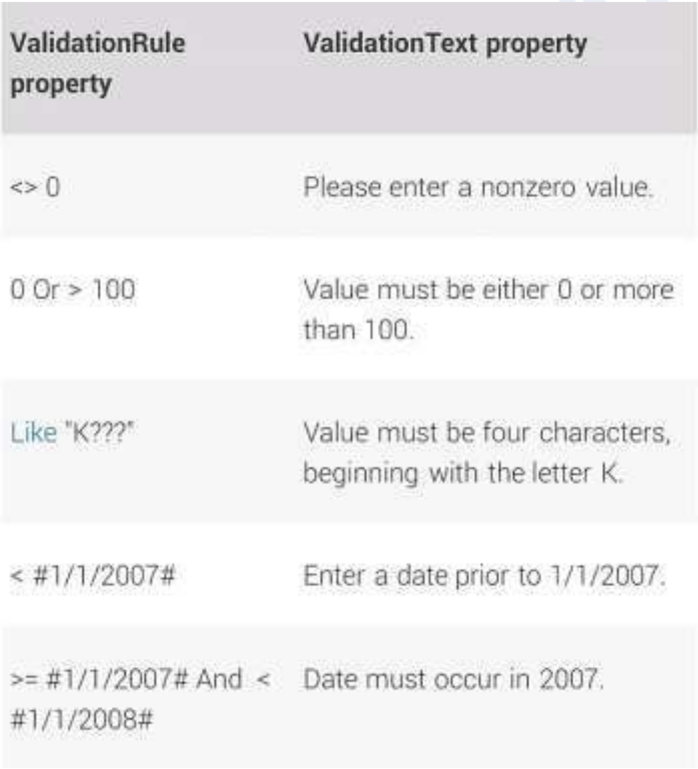
Expressions in Queries.

Expression builder can also be used in Queries under *Criteria* in Access 2016.

Field	**LoanDate**	*Other Examples*
Criteria	<Date( ) – 30	NOT Like "C*"
	This displays loans more than 30 days old.	NOT "U.S.A
		In ("Kenya","Uganda")
		"Kenya" OR "Uganda" same

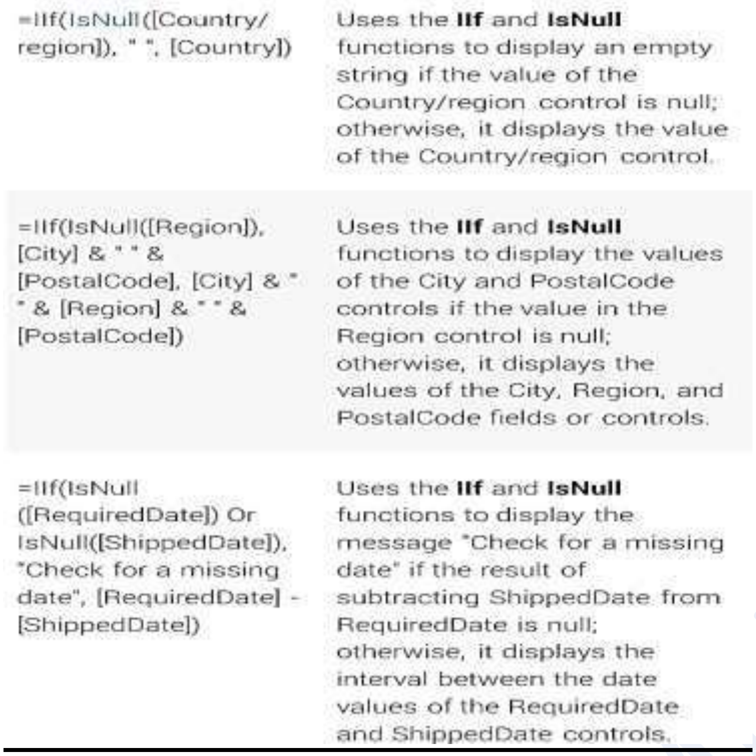
Validation Rules Using Expressions.

Suppose you want to automatically insert the date and time into a field whenever someone adds a new record you might use **=NOW( )**
Validation Text and *Validation Rule* can be added to Not Only tables but even *Controls* as well e.g. textbox under properties, you can have **>#1/1/2010#**



ValidationRule property	ValidationText property
<> 0	Please enter a nonzero value.
0 Or > 100	Value must be either 0 or more than 100.
Like "K???"	Value must be four characters, beginning with the letter K.
< #1/1/2007#	Enter a date prior to 1/1/2007.
>= #1/1/2007# And < #1/1/2008#	Date must occur in 2007.

Inbuilt Functions Expressions.



=IIf(IsNull([Country/region]), " ", [Country])	Uses the **IIf** and **IsNull** functions to display an empty string if the value of the Country/region control is null; otherwise, it displays the value of the Country/region control.
=IIf(IsNull([Region]), [City] & " " & [PostalCode], [City] & " " & [Region] & " " & [PostalCode])	Uses the **IIf** and **IsNull** functions to display the values of the City and PostalCode controls if the value in the Region control is null; otherwise, it displays the values of the City, Region, and PostalCode fields or controls.
=IIf(IsNull ([RequiredDate]) Or IsNull([ShippedDate]), "Check for a missing date", [RequiredDate] - [ShippedDate])	Uses the **IIf** and **IsNull** functions to display the message "Check for a missing date" if the result of subtracting ShippedDate from RequiredDate is null; otherwise, it displays the interval between the date values of the RequiredDate and ShippedDate controls.

The Subform Control

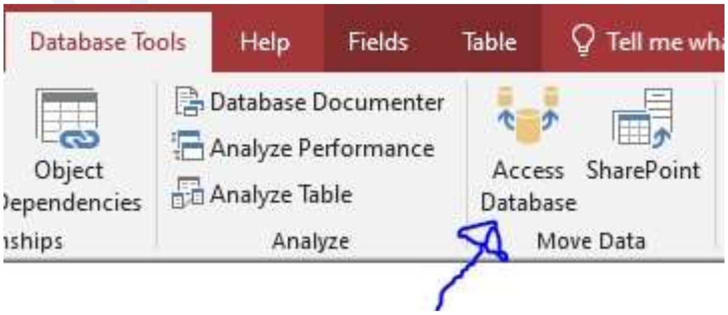
Access shows linked records using the sub form control. You can add the sub form control to any form to show linked records. It's available in the ribbon's Form Design Tools | Design→Controls section with all the other controls. If you add it by hand, Access prompts you to pick the table you want to show.

Three properties determine what the sub form control shows. First, the Source Object property identifies the object in the database that has the related records. You can choose an existing table, query, or form.

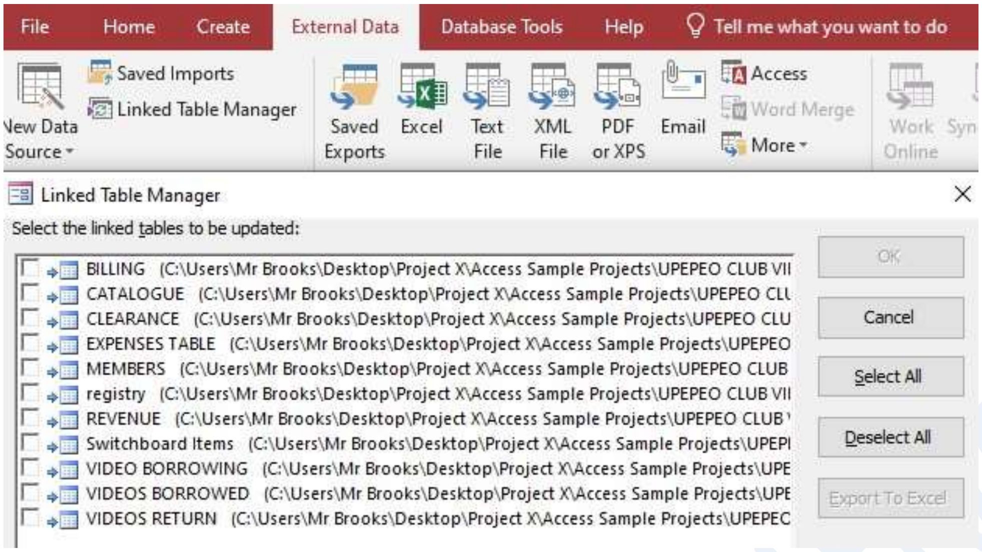
CREATION: After creating relationships for tables, when you select fields from those different tables during Form creation by the wizard. A sub form is automatically created.

How to SPLIT an Access Database.

You can split your normal database into two files, *Front end* (quarries, forms, reports, modules) with tables pointing to the *backend* (tables) where the Real tables reside.



If a location of the frontend changes, there is no effect. Still data will be updated on the backend But if the location of the backend is changed. You can update the new links using the **Linked table manager.**

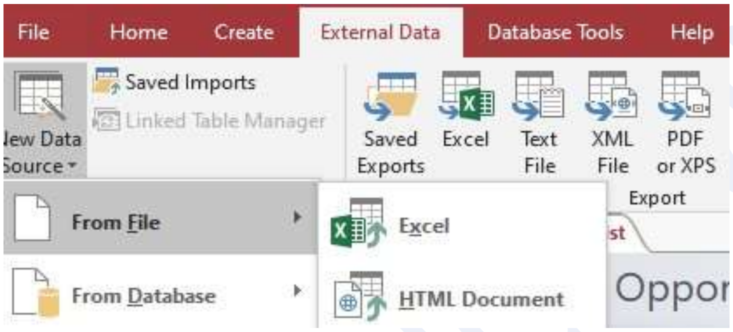


Linked Table Manager

Select the linked tables to be updated:

□ BILLING (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO CLUB VII
□ CATALOGUE (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO CLU
□ CLEARANCE (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO CLU
□ EXPENSES TABLE (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO
□ MEMBERS (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO CLUB
□ registry (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO CLUB VII
□ REVENUE (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEO CLUB
□ Switchboard Items (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPI
□ VIDEO BORROWING (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPE
□ VIDEOS BORROWED (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPE
□ VIDEOS RETURN (C:\Users\Mr Brooks\Desktop\Project X\Access Sample Projects\UPEPEC

How to Import & Export Data.



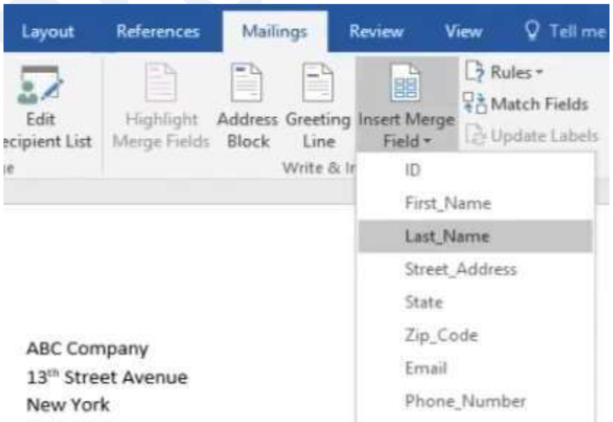
How to Mail Merge in Access

It’s a feature that is used to send one letter to several addresses. From Access once you have a table of recipients, on the ***External Data Tab*** click Word Merge.

On the Merge wizard choose *Create new Document* and link to data and click Okay.

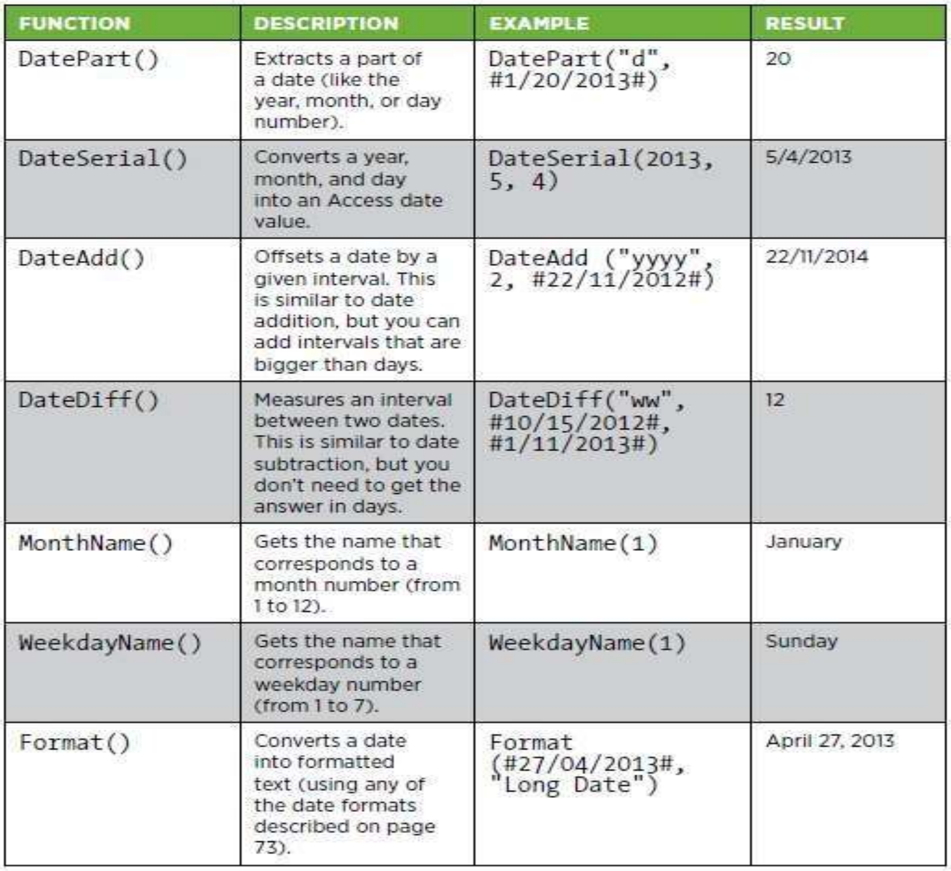
Immediately a new Ms. Word window will open. Follow the steps on the side pane. In step 3 remember to use an *existing list.*

In step 4 write your letter and from the Mailings tab, click Insert merge fields. Complete the procedure in step 6

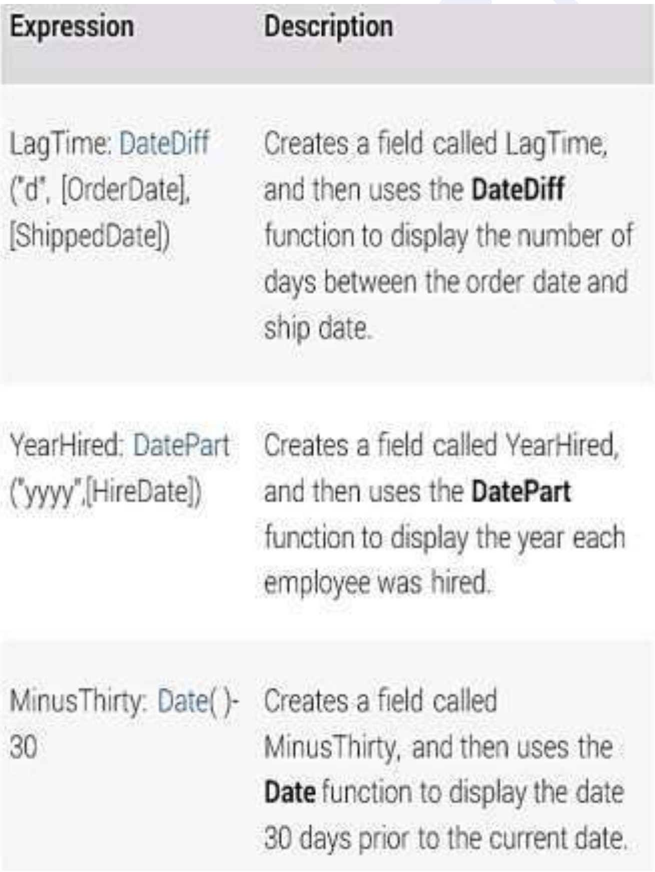


ABC Company
13th Street Avenue
New York

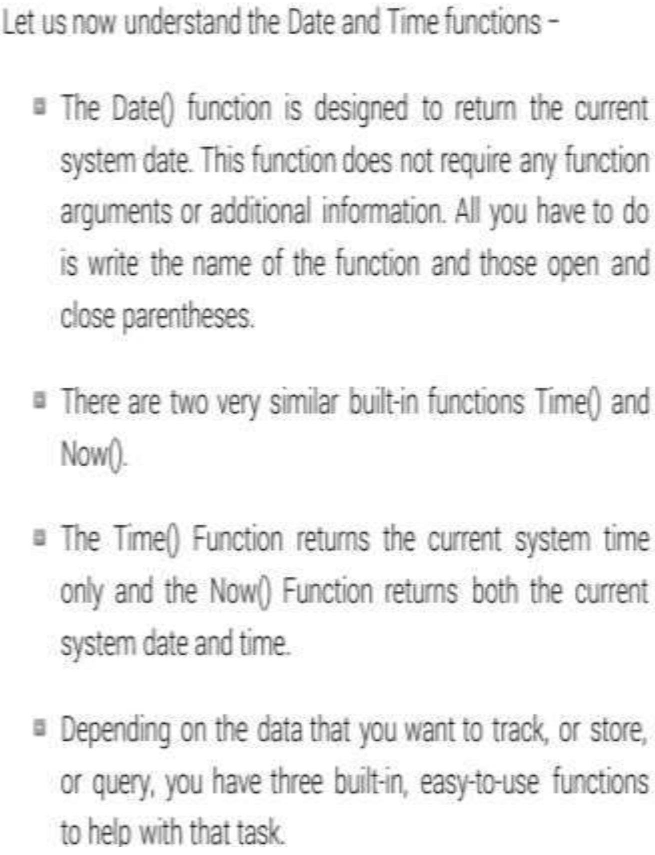
DATE FUNCTIONS.



FUNCTION	DESCRIPTION	EXAMPLE	RESULT
DatePart()	Extracts a part of a date (like the year, month, or day number).	DatePart("d", #1/20/2013#)	20
DateSerial()	Converts a year, month, and day into an Access date value.	DateSerial(2013, 5, 4)	5/4/2013
DateAdd()	Offsets a date by a given interval. This is similar to date addition, but you can add intervals that are bigger than days.	DateAdd ("yyyy", 2, #22/11/2012#)	22/11/2014
DateDiff()	Measures an interval between two dates. This is similar to date subtraction, but you don't need to get the answer in days.	DateDiff("ww", #10/15/2012#, #1/11/2013#)	12
MonthName()	Gets the name that corresponds to a month number (from 1 to 12).	MonthName(1)	January
WeekdayName()	Gets the name that corresponds to a weekday number (from 1 to 7).	WeekdayName(1)	Sunday
Format()	Converts a date into formatted text (using any of the date formats described on page 73).	Format (#27/04/2013#, "Long Date")	April 27, 2013



Expression	Description
LagTime: DateDiff ("d", [OrderDate], [ShippedDate])	Creates a field called LagTime, and then uses the **DateDiff** function to display the number of days between the order date and ship date.
YearHired: DatePart ("yyyy",[HireDate])	Creates a field called YearHired, and then uses the **DatePart** function to display the year each employee was hired.
MinusThirty: Date( )- 30	Creates a field called MinusThirty, and then uses the **Date** function to display the date 30 days prior to the current date.



Let us now understand the Date and Time functions −

- The Date() function is designed to return the current system date. This function does not require any function arguments or additional information. All you have to do is write the name of the function and those open and close parentheses.
- There are two very similar built-in functions Time() and Now().
- The Time() Function returns the current system time only and the Now() Function returns both the current system date and time.
- Depending on the data that you want to track, or store, or query, you have three built-in, easy-to-use functions to help with that task.

GROUPS OF CODES IN V.B.A

Modules – A container of code to help organize it which may be assigned to form or report

Functions – code which is grouped together and can return a value. Reusable code which can be called repeatedly. We group bits of codes together so that it's easier to manage.

Sub routines – code which is grouped together and does not return a value. Reusable code which can be called repeatedly. We group bits of codes together so that it's easier to manage.
Sub routines codes performed when a form or a report triggers an *Event*.

Classes – A module of code which outlines the behavior and properties of an *object.*

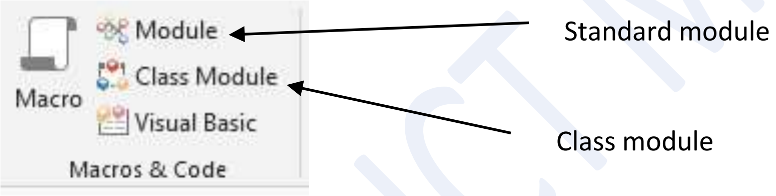
Properties – parts of an object that holds values for that object

e.g. Color=red, Automatic transmission=false, speed=60

Methods – behavior that the object will perform

e.g. TurnLeft(), ApplyBreak()

MODULES.



if you place your code on the *class module*, it will only be accessible/embedded to that object (Form/report). if you place your code on the *standard module*, it will be accessible to multiple objects within the database.

A module contains a *standalone code* routine that runs only when you tell it to. Initially, a brand-new module has just one line of code, which looks like this:

Option Compare Database

This line is an instruction that tells Visual Basic how to handle operations that compare pieces of text. Ordinarily, Visual Basic has its own rules about how to deal with text, but this statement tells it to use the Access settings instead.
Before you write code that actually does anything, you should add one more instruction to the top of your code file. Just before (or after) the Option Compare Database instruction, add this:

Option Explicit

This instruction tells Visual Basic to use stricter error checking, which catches common typos when using variables. You can tell Visual Basic to add the Option Explicit line automatically to all new code files. To do so, select Tools→Options, check the Require Variable Declaration option, and then click OK. Access experts always use this setting.
As with other Access database objects, when you close the Visual Basic editor, Access prompts you to save any newly created modules. If you don't want to wait that long, choose File→Save [DatabaseName] where DatabaseName is the name of your database file.

Note Once you've saved your module, you can see it in the Access window's navigation pane. If you're using the Tables and Related Views mode, your module appears in the Unrelated Objects category. If you're using the Object Type mode, it appears in a separate Module category. If you double-click a module in the navigation pane, Access opens it in the Visual Basic editor.

Inside every module (except the empty ones) are one or more Visual Basic subroutines. A subroutine is a named unit of code that performs a distinct task. In the VB language, subroutines start with the word Sub followed by the name of the subroutine. Subroutines end with the statement End Sub. Here's an example subroutine

```
Sub MyCodeRoutine()
' The following statement shows a message box.
MsgBox "Witness the power of my code." End
Sub
```

Sometimes Access coders use the word Me in front of field names and control names. So instead of write this:

Textbox1 = "" They write this: ***Me.*** *Textbox1 = ""*

Technically, Me is a programming object that represents the current form. You might wonder why anyone would type more to achieve the same thing. The reason is because of Visual Basic's handy IntelliSense. When you type Me followed by a period, the Visual Basic editor pops open a long list of every ingredient on the current form, the suggestions you can then accept by pressing the space bar. So, while using Me makes no difference to the way code works, it does help some programmers type their code more quickly.

In Short:

Modules are code sheets in VBA Project used to create general procedures (sub-programs) that are not associated with any event on objects like Forms or Reports. A VBA project normally uses at least one *module* to *store* the necessary *functions* and *subroutines* known as *Procedures.*

A *subroutine* is a piece of code that performs a set of actions/calculations while

A *function* is like a subroutine except that it returns a single value.

VBA allows the developer to define functions or subroutines as either *Public* or *Private* using those keywords. Public functions are accessible throw-out the *Modules* within the database application while Private Procedure or Function is only accessible within a specific module.

Access VBA comes with library of built in functions used to perform various task such as mathematical calculations and error handling, Examples: date (), Exp( ), Sqr ()

e.g. Myroot = Sqr (84)

Msgbox=Myroot

You can interact with objects in three ways:

- ***Properties***. Properties are pieces of information about an object. You change properties to modify the object or how it behaves. A text box object has a FontSize property that controls its text size.

  Example: *TextBox1.Backcolor = vbYellow*

  Properties describes characteristics of the object (adjective) e.g. color, shape, price

- ***Methods***. Methods are actions you can perform with an object. For instance, every form has a Requery method that lets you rerun the query that gets its data. Methods describes actions of the object (verb) e.g. run, walk, jog

- ***Events***. Events are notifications that an object sends out, which you can respond to with your code. You can react to button clicks by using the button control's On Click event.

METHODS

Methods let you perform actions with an object. In many cases, calling a method does more than set a property. In fact, a single method may launch a complex operation that affects many properties. The Requery method tells your form to get the latest data from the database, and then to refresh all its controls.

Examples: *TextBox1.Setfocus*

Class – a template for creating objects

The DoCmd Object

Microsoft Access provides an object that allows you to open one object from another. This object is called DoCmd. The DoCmd object can be used to open/Close a form or report from another form. It is equipped with methods (also called functions; we will soon learn what a function is) that you can call to perform the desired operation.

For example, to open a form, call the OpenForm using the following syntax:

DoCmd.OpenForm *FormToOpen*

OR DoCmd.Close OR DoCmd.CloseDatabase

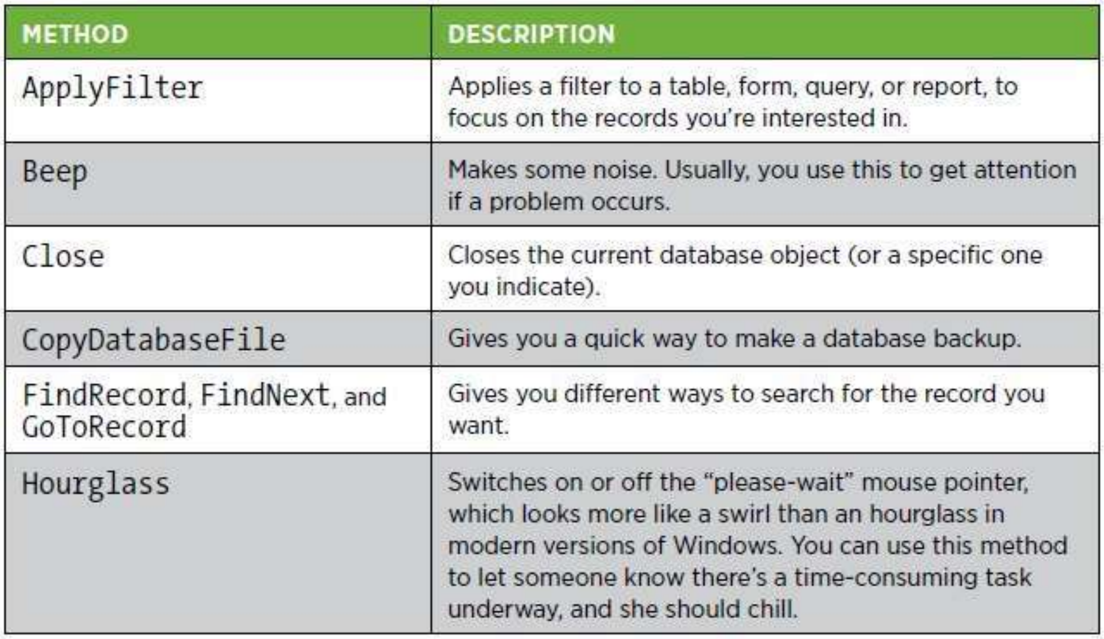
The DoCmd object doesn't have any properties.

Instead, it's made up of methods that perform different actions.

Like most of the DoCmd methods, OpenForm can use several optional parameters.

Visual Basic prompts you by showing the list of possible parameters as you type.

Useful Methods of the DoCmd Object



METHOD	DESCRIPTION
ApplyFilter	Applies a filter to a table, form, query, or report, to focus on the records you're interested in.
Beep	Makes some noise. Usually, you use this to get attention if a problem occurs.
Close	Closes the current database object (or a specific one you indicate).
CopyDatabaseFile	Gives you a quick way to make a database backup.
FindRecord, FindNext, and GoToRecord	Gives you different ways to search for the record you want.
Hourglass	Switches on or off the "please-wait" mouse pointer, which looks more like a swirl than an hourglass in modern versions of Windows. You can use this method to let someone know there's a time-consuming task underway, and she should chill.

```
Private Sub Ship_Click()
        ' Close this form.
        DoCmd.Close
        ' Switch back to the ShipOrders form.
        DoCmd.OpenForm "ShipOrders" '
        Update the order.
        ' The StatusID for Shipped is 4.
        Forms("ShipOrders").StatusID = 4
        DoCmd.RunCommand acCmdSaveRecord

End Sub
```

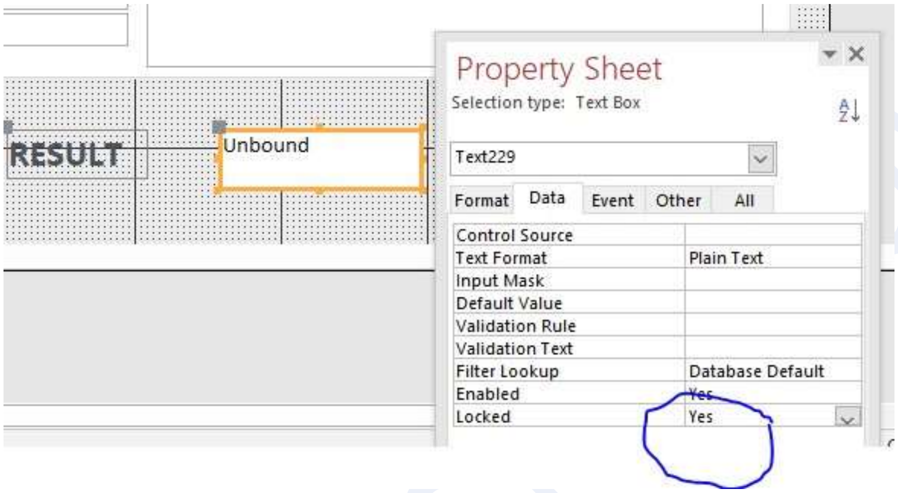
VARIABLES.

Variables hold a place in memory to store data. Variable data type determines how much memory to allocate.

To declare variables, we use the keyword **DIM,** e.g. *Dim age As Integer*
Assigning a value e.g. *age=25* or stname= "bob"

Examples of V.B.A data types.
Boolean, byte, integer, long integer, single, Double, currency, date, string, variant etc.

TEXT BOX LOCKING



Property Sheet

Selection type: Text Box

Text229

Format | Data | Event | Other | All

Control Source	
Text Format	Plain Text
Input Mask	
Default Value	
Validation Rule	
Validation Text	
Filter Lookup	Database Default
Enabled	Yes
Locked	Yes

If you don't want the users to enter anything e.g. a value on a text box. You can lock it under properties pane. **Additional Tips:**

- **When a picture or other graphic image is placed in the report header section it will appear____** Once in the beginning of the report
- In one-to-many relationship the table in 'one' side is called **Parent** and on 'many' side is called **Child**
- **A small button with three dots usually displayed at the right of field properties box** Build button
- **A database language concerned with the definition of the whole database structure and schema is** DDL
- **The database language that allows you to access or maintain data in a database** DML
- Open Exclusive locks whole database so that no one can access it whereas Open locks only the record that is being accessed
- **A composite key is** Made up of two or more fields to uniquely identify records
- **In databases, Locking level is also called as** Granularity
- According to the *Leszynski Naming Conventions*, which of the following is the prefix for a text box control? **txt**

SAMPLE CODES.

If Then

```
Private Sub Selection( )
        Dim score As Integer
        Score=80
                If score >=80 Then
                        Msgbox “Excellent pass with” & score
                End if
End Sub
```

If Then Else

```
Private Sub Selection( )
        Dim score As Integer
        Score=80
                If score >=80 Then
                        Msgbox “Excellent pass with” & score
                Else
                        Msgbox “More room for Improvement” & score
                End if
End Sub
```

Nested If

```
Private Sub Nestedif( )
        Dim score As Integer
        InputBox “Enter marks between 0 and 100”
                If score >=80 Then
        Msgbox “Your grade is A”
        ElseIf score >=50
        Msgbox “Your grade is B”
        Else    If score >=30
                        Msgbox “Your grade is C”
                Else
                        Msgbox “Your Grade is E”
                End if
                End if
                End if
End Sub
```

SUB PROCEDURE CODE:

```
Sub ChangeColor()
txtCountry.BackColor = 16763293
End Sub
```

Select CASE

```
    Private Sub Mycase()
        Dim choice As Integer
        Choice=InputBox("Enter 1 to 3")
            Select Case choice
            Case 1
                MsgBox ("Orange")
            Case 2
                MsgBox ("banana")
            Case 3
                MsgBox ("apple")
        Case Else
                MsgBox ("Invalid Choice")
            End Select
    End Sub
```

WHILE…WEND

Write a program that displays ten numbers from 0 to 10

```
    Private Sub Whileloop()
Dim x As integer
        x=0
            While x < 10
                MsgBox x
                X=x+1
            Wend
    End Sub
```

FOR Loop

Write a program to display numbers between 0 to 10 /   plus even numbers

```
Private Sub ForLoop()
    Dim n As Integer
    For n=0 To 10
    MsgBox n
    Next n
End sub
```

```
Private Sub ForLoop()
    Dim n As Integer
     For n=0 To 10 Step 2
    MsgBox n
    Next n
End sub
```

Do…Until

```
Private Sub ForLoop()
Dim n As Integer
        n=1
        Do Until n=10
        MsgBox n
        n=n+1
        Loop
End sub
```

```
Private Sub Form_Timer()
        'blink label is the name of textbox control
                With blinklabel
                .ForeColor = IIf(.ForeColor = 1, 255, 1)
        End With
        End Sub
```

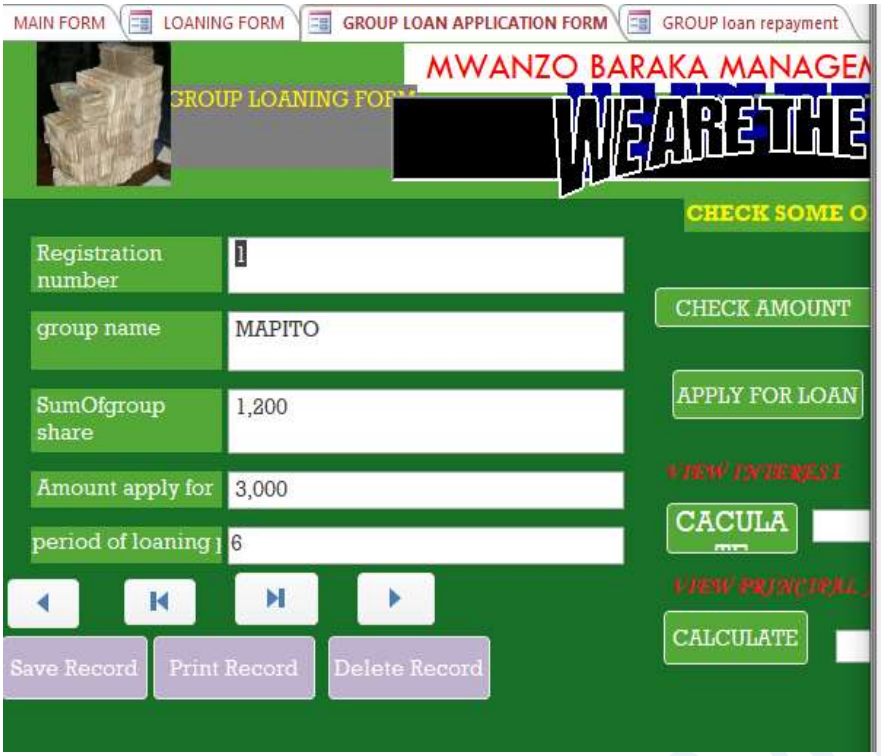
```
        Function GetFullName() As String
                Dim strFirstName, strLastName As String
                strFirstName = txtFirstName
                strLastName = txtLastName
                GetFullName = strFirstName & " " & strLastName
        End Function
```

```
        Private Sub cmdMessage4_Click()
                MsgBox "You are about to embark on a long journey." & _
                vbCrLf & "If your courage is still fresh, " & _ "now is the time to let us know!",
                _ vbOKCancel + vbQuestion, _ "Accept or Cancel the Mission"
        End Sub
```

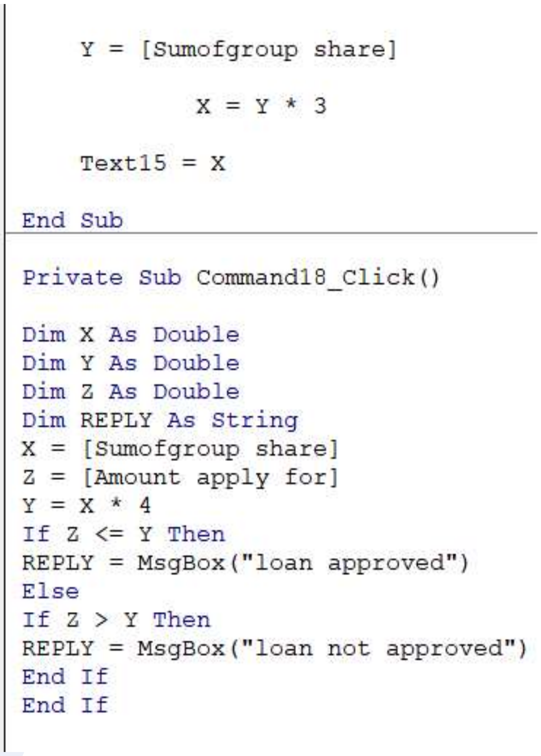
Hiding the Navigation Buttons

```
        Private Sub Form_Load() Me.NavigationButtons
                = False
        End Sub
```

```
        Private Sub Form_Current() If
                Gender = "Male" Then
                Detail.BackColor = 16772055
                FormHeader.BackColor = 16752478
                FormFooter.BackColor = 14511872
                End If
        End Sub
```

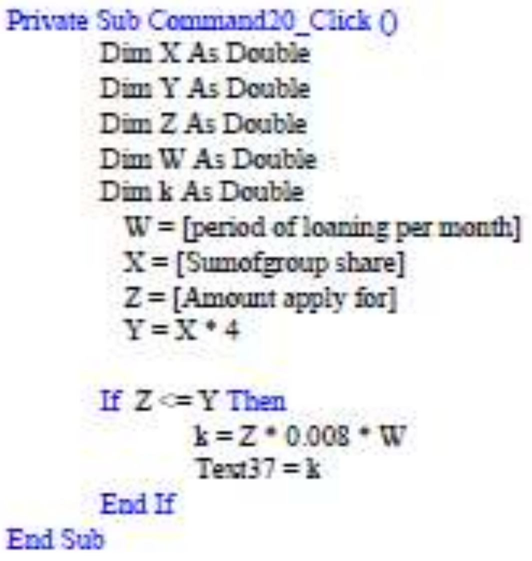


Registration number	1
group name	MAPITO
SumOfgroup share	1,200
Amount apply for	3,000
period of loaning	6

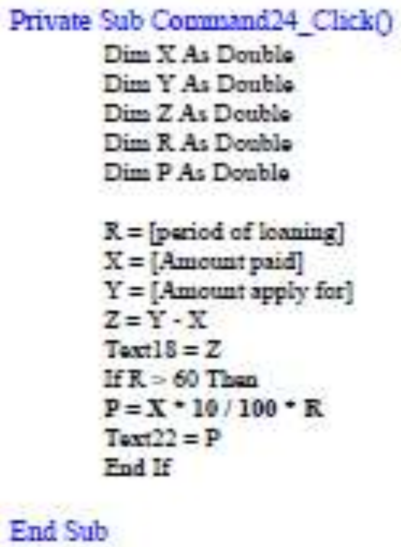


```
    Y = [Sumofgroup share]

          X = Y * 3

    Text15 = X

End Sub
```

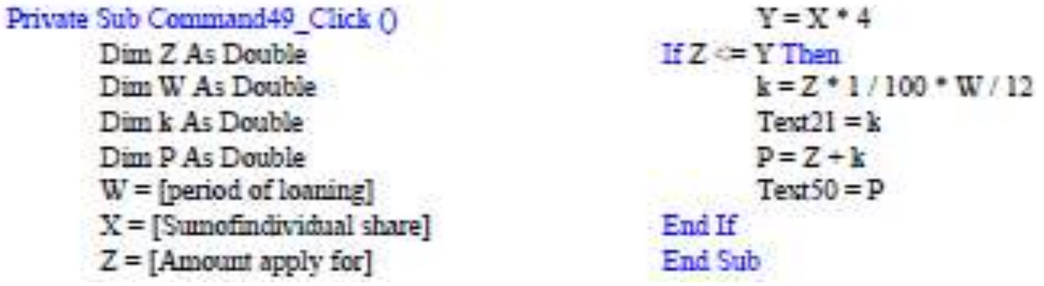
```
Private Sub Command18_Click()

Dim X As Double
Dim Y As Double
Dim Z As Double
Dim REPLY As String
X = [Sumofgroup share]
Z = [Amount apply for]
Y = X * 4
If Z <= Y Then
REPLY = MsgBox("loan approved")
Else
If Z > Y Then
REPLY = MsgBox("loan not approved")
End If
End If
```



```
Private Sub Command20_Click ()
        Dim X As Double
        Dim Y As Double
        Dim Z As Double
        Dim W As Double
        Dim k As Double
          W = [period of loaning per month]
          X = [Sumofgroup share]
          Z = [Amount apply for]
          Y = X * 4

        If  Z <= Y Then
                k = Z * 0.008 * W
                Text37 = k
        End If
End Sub
```



```
Private Sub Command24_Click()
        Dim X As Double
        Dim Y As Double
        Dim Z As Double
        Dim R As Double
        Dim P As Double

        R = [period of loaning]
        X = [Amount paid]
        Y = [Amount apply for]
        Z = Y - X
        Text18 = Z
        If R > 60 Then
        P = X * 10 / 100 * R
        Text22 = P
        End If

End Sub
```

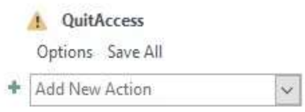


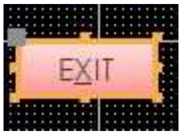
```
Private Sub Command49_Click ()
        Dim Z As Double
        Dim W As Double
        Dim k As Double
        Dim P As Double
        W = [period of loaning]
        X = [Sumofindividual share]
        Z = [Amount apply for]
        Y = X * 4
If Z <= Y Then
        k = Z * 1 / 100 * W / 12
        Text21 = k
        P = Z + k
        Text50 = P
End If
End Sub
```

SAMPLE CODE FOR MULTI-USER DATABASE

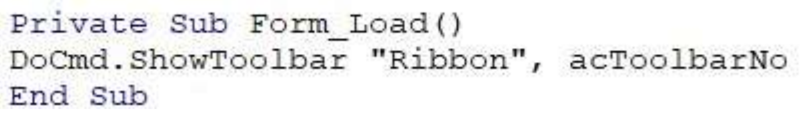
```
Option Compare Database

Private Sub Form_Load()
Me.txtUserName.SetFocus

End Sub

Private Sub Command1_Click()
Dim User As String
Dim UserLevel As Integer
Dim TempPass As String
Dim ID As Integer
Dim UserName As String
Dim TempID As String


   If IsNull(Me.txtUserName) Then
MsgBox "Please enter UserName", 
vbInformation
        Me.txtUserName.SetFocus
        Else
        If IsNull(Me.txtPassword) Then
MsgBox "Please enter password", 
vbInformation
Me.txtPassword.SetFocus
        Else
        TempID = Me.txtUserName.Value
        UserName = DLookup("[UserName]", 
"tblUser")
        UserLevel = DLookup("[UserType]", 
"tblUser")
        TempPass = DLookup("[UserPassword]", 
"tblUser")
        UserLogin = DLookup("[UserLogin]", 
"tblUser")
        DoCmd.Close
        If (TempPass = "password") Then
MsgBox "Please change Password", 
vbInformation
        DoCmd.OpenForm "Travel Form"
        Else
        If UserLevel = 1 Then
        DoCmd.OpenForm "navigation form"
        End If
        End If
        End If
        End If
End Sub
```

Attached Macro on Exit Button





To **HIDE** the ribbon from not showing…. attach on form load event

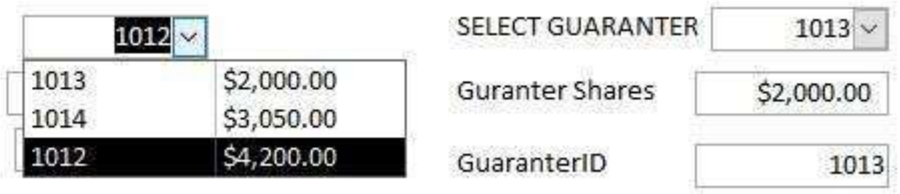


```
Private Sub Form_Load()
DoCmd.ShowToolbar "Ribbon", acToolbarNo
End Sub
```

Piece of code for writing on the status bar

```
SysCmd acSysCmdSetStatus, "im loading"
```

Piece of Code for storing Multiple Values from a COMBO BOX Selection on a Form



1012

1013	$2,000.00
1014	$3,050.00
1012	$4,200.00

SELECT GUARANTER	1013
Guranter Shares	$2,000.00
GuaranterID	1013

```
Private Sub Combo14_AfterUpdate()
Me.GuaranterID = Me.Combo14.Column(0)
End Sub
```

‘’means only pick item in index 0 which is the ID and store it in a field called GuaranterID