INTRODUCTION TO COMPUTER APPLICATIONS

IN

DATABASE SYSTEMS

(MICROSOFT ACCESS 2007)

By

Ndukwe I.G

Cs 102

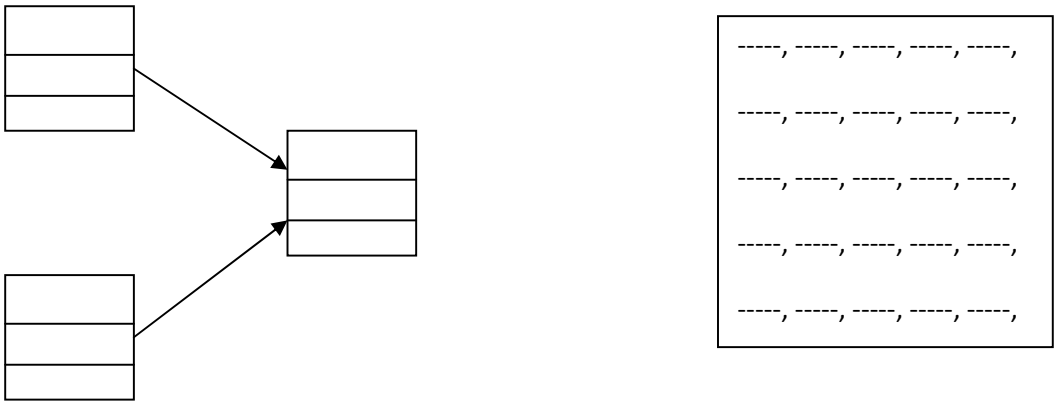
CHAPTER ONE

Introduction to Database Concepts and Terminology

Microsoft Access (MS Access) is a Relational Database Management System *(RDBMS) which* enables you to store, organize, and manipulate collections of information in an electronic format.

Access 2007 manages data electronically, it enables you to manipulate and extract the data with ***ease*** and ***flexibility***.

Relational Database Vs Flat File Database



What is a Flat File?

A flat file database is where you have a simple text file and all the information is separated with a comma delimiter (or some kind of delimiter). Each of the records is a kind of delimited by a new line character.

The problem with a flat file is that when you get into very large quantities of data, it becomes very cumbersome and complex to maintain the relationship between records.

If you have ever handled a flat file database, you might have used a programming language like java to deal with the data to do whatever sorting you need to and it can be quite confusing and painful. One common example of a flat file is Microsoft Excel.

What is a Relational Database?

A Relational Database is a collection of information that is organized in form of tables such that some of the tables linked to each other.

You might have a table that represents Students and another table Departments. Each table have column representing their different data types.

Example

Student table could have columns like Id, First Name, Last Name, Sex etc. and the Department table could have columns like Dept Id, Department Name etc.

You maintain the relationships between tables with constraints like primary key and foreign key.

The way data is stored in a Relational Database is much different. If we were to open up the actual data file in a relational database we will see that data is usually stored in a binary format unlike flat files which are stored in text formats. So you can use Micro Soft Access to access these binary files and display their content in text formats that is readable.

> ***NOTE***
>
> *For example, suppose you are working with a personal database and you need to pull the names and extensions of all students who belong to a particular department.*
>
> *If your paper files are organized alphabetically by last name, you may need to check every file to determine the department in which each student belongs. This could take hours when using a flat file.*
>
> *With Access, this could take just minutes to do.*

RDBMS Terminologies

Before you plan and design a database, it helps to be familiar with the terminology that is common to a relational database management system.

What is a Database

A ***database*** is a collection of related information stored in tables.

In a relational database, two or more tables can be linked to appear as one table.

Also repetition of data items is eliminated, except for those items the tables are linked by.

Common examples of databases include: *phone books, mailing lists, student records, and billing records.*

What is a Table

A ***table*** is a collection of related data stored in **rows** and **columns**.

Since Access 2000 is a relational database management system, you can store data in separate tables, and extract or display related data together in one table.

> ***NOTE***
>
> *Since Access 2007 is a relational database management system, you can store data in separate tables, and extract or display related data together in one table.*

Tables are composed of ***Records***, ***Fields*** and ***Data*** **Values**.

Table 1.1

ID	Last Name	First Name	Level	Score	Grade
01	Eze	Obi	100L	60	B
02	John	Bot	200L	40	D
03	Mike	Juliet	100L	70	A

*This is an example of a student **table**.*

A ***Record*** is a row in a table.

Table 1.2

ID	Last Name	First Name	Level	Score	Grade
01	Eze	Obi	100L	60	B
02	John	Bot	200L	40	D
03	Mike	Juliet	100L	70	A

*The colored row of the students table shows an example of a **record**.*

Records can contain all the information for one **person**, **thing** or **event**.

> ***NOTE***
>
> *In the table shown here, each record contains all the information for one student.*

A ***field*** is a column in a table. **Fields** contains categories of information.

Table 1.3

ID	Last Name	First Name	Level	Score	Grade
01	Eze	Obi	100L	60	B
02	John	Bot	200L	40	D
03	Mike	Juliet	100L	70	A

*The colored column of this table shows an example of a **field**.*

For example, the table displayed here contains six fields: *ID, Last Name, First name, Level, Score, and Grade.*

Table 1.4

ID	Last Name	First Name	Level	Score	Grade
01	Eze	Obi	100L	60	B
02	John	Bot	200L	40	D
03	Mike	Juliet	100L	70	A

*The colored cell of this table shows an example of a **Data Value**.*

CHAPTER TWO

Introduction to Access 2007

In this chapter, we will explore the Access environment. Then we will discuss some of Access objects and how they interact in a database.

Access Objects

The components of a database are called **objects.** *The most basic object is a **Table**; Since a table is where all the data is stored.*

Other objects include: **Forms, Queries, Reports, Pages, Macrons, and Modules.**

HANDS ON

- *Open your Micro Soft Access 2007, a screen is displayed containing three dialog boxes; The **Left Dialog Box** is titled "Template Category", the **Middle Dialog Box** is titled "Getting Started with Microsoft Office Access", while the **Right Dialog Box** is titled "Open Recent Databases".*
- *The **Featuring Tab** in the **Left Dialog Box** is highlighted by default.*
- *The **Middle Dialog Box** gives you a choice of a choice of creating a new **Database** or creating a database based on an already existing **Template.***
- *Click on the **Blank Database** button. A text box labeled **File Name** with default value **Database1** is automatically displayed in the **Right Dialog Box**.*
- *Enter the database name in this text box. (Under this text box is the path where this database file will be housed. e.g. C:\Users\nduxy\Documents).*
- *Click on Create button*

Congratulations!! You have just created a Database file in MS Access.

Title Bar

Across the top of the application window is the **Title Bar**.

The **Title Bar** contains the ***Office Button, Database Name, Minimize Button, Maximize Button*** *and* ***Close Button.*** *However, the **Customize Quick Access Tool Bar** could be set to the **Title Bar** or below the ribbon**.***

The **Customize Quick Access Tool Bar** has the **Save, Undo and Redo** buttons by default. It can be customized by clicking on the drop down button which drops down a menu list of items that can be selected to display more quick access buttons.

Menu Bar

Bellow the Title Bar is the **Menu Bar.** Access 2007 menus are adaptive, which means that the available commands change automatically depending on the activity you are performing.

NOTE

*When you click on a menu item in the **Menu Bar,** it displays all the tools that can be used for that menu in the Tool Bar.*

HANDS ON

Click on each menu bare options and observe the contents.

Tool Bar

Below the Menu Bar is the **Tool Bar,** which is used to perform common commands quickly.

NOTE

*Like the Menu bar, the tools on the **Tool Bar** are also adaptive, and the available commands change automatically as you work.*

Tool Bar Buttons

The **Tool Bar Buttons** have icons to help you identify the commands they represent. However, if you don't recognize a button, you can place the mouse pointer directly over the button and pause to see a **Screen Tip**.

NOTE

***Screen Tips** describe or identify the button commands.*

Ribbon

The **Ribbon** holds both the Menu Bar and the Tool Bar.

Status Bar

Along the button of the screen is the **Status Bar.** The **Status Bar** displays messages that enable you to use the menus more efficiently.

Create Button in the Menu Bar

Observe the **Objects** displayed when you click on the **Create Button** in the menu bar.

NOTE

*The **Create Button** displays all the **database objects** which are **Tables, Forms, Reports, Queries** and **Macros.***

***What is a* Database?**

A **database** is a collection of information organized to provide efficient retrieval.

Table Object

A **Table** a database object that holds information about like items and organizes them in form of **rows** and **columns**.

Query Object

A **Query** is a way to analyze, select, and ask questions about data that are stored in tables.

NOTE

You can use a query to locate records in one or more tables, perform calculations and totals, or select certain kind of data.

Queries look much like tables. However, queries are much more selective.

For example, we can run a query that displays just the students Name and their ID.

Form Object

A **Form** is an object used for entering and viewing information stored in a table or query.

HANDS ON

- *Open your Micro Soft Access 2007*
- *The **Featuring Tab** in the **Left Dialog Box** is highlighted by default.*
- *The **Middle Dialog Box** gives you a choices creating a database based on an already existing **Template.***
- *Click on the **Student** button. A text box labeled **File Name** with default value **Student** is automatically displayed in the **Right Dialog Box**.*
- *Click on Download button*

Congratulations!! You have just created a Database file in MS Access.

HANDS ON

- *Click and Open the **Student Detail Form.***
- *Observe that this **form** contains the same fields with that of the **Students Table**.*

A **form** retrieves its data from the table on which it is based. **Forms** also display only one record at a time, and are often used for data-entry purpose.

Report

A **Report** enables you to analyze and print data in a specific format. Although you can also print tables and queries, reports give you greater control over the printout format.

HANDS ON

Click on Students Report

Reports can include data contained in one or more tables, or they can include the results of a query.

Page

A **Page** is an object used to build Access forms and reports that run on the web.

CHAPTER THREE

Database Planning and Design

The most important step in creating your database is planning.

Planning is important because what you include in the database, and how you store the data, will affect what you can do with it.

field is the most basic piece of information in a database file. When you design fields for your database, each field should contain the smallest meaningful value possible.

For example, in a database that includes student's name, you would separate first and last names into two fields as shown in tables 2.1 and 2.2.

Table 2.1

Id	Name	Level	Score	Grade
01	Eze, Obi	100L	60	B
02	John, Bot	200L	40	D
03	Mike, Juliet	100L	70	A

This table holds both the Surname and First Name of a Student in one table.

Table 2.2

Id	Last Name	First Name	Level	Score	Grade
01	Eze	Obi	100L	60	B
02	John	Bot	200L	40	D
03	Mike	Juliet	100L	70	

This table holds the Student Surname in one field and then the Student First Name in another.

Likewise, you would break addresses into street address, city, state, and zip code fields.

NOTE

When you are designing your database, you are tempted to create one large table that includes all of the necessary fields. However, in most cases, this is not the most effective design.

Each table should focus on one topic and should store only the data related to that topic.

*This indicates that your table is in the **1st Normal Form**.*

When you are designing your database, it may be tempting to create one large table that includes all of the necessary fields. However, in most cases, this is not the most efficient design.

Each table should focus on one topic and should store only the data related to that topic.

Storing data in separate tables provide the flexibility to extract and summarize data. It also makes managing data easier, and eliminates duplicate data.

Table 2.3

ID	Last Name	First Name	Sex	Dept ID	Department
01	Eze	Obi	M	001	Mathematics
02	John	Bot	M	002	Physics
03	Mike	Juliet	F	001	Mathematics

This table combines both Student and Department information.

Table 2.3

ID	Last Name	First Name	Sex	Dept ID
01	Eze	Obi	M	001
02	John	Bot	M	002
03	Mike	Juliet	F	001

This table contains only student information

Table 2.4

Dept ID	Department
001	Mathematics
002	Physics

This table contains Department information

NOTE

Suppose you want to change the department for a set of students.

If your personal database includes separate tables such as Student and Department tables in table 2.3 and 2.4, you will need to change the department in one place, rather than in the records for every student affected by this change.

Normalization

Normalizing your tables is also a very important planning step in database design. The database community set few rules or guidelines for database normalization. These rules are called **normal forms** and starts from the first normal form till the fifth normal form.

In this chapter we will only discuss two important normal forms which are **1NF** and **2NF**.

First Normal Form (1NF)

The **1NF** sets very crucial rules to creating database tables:

1. There are no repeating or duplicate fields.
2. Each cell contains a single value
3. Each record is uniquely identified by a primary key

Table 2.4

Course Code	**Course Title**	**Name1**	**Name2**	**Name3**
MTH 101	Calculus	Eze, Obi	John, Bot	Mike, Juliet
CS 102	Computer Applications	Olu, Ayo	Peter, Oke	Jim, Paul
STA 111	Probability	Bala, Ali	Oluwa, Seun	Eze, Obi

RULE 1: A table with atomic data cannot have multiple columns with the same type of data. We have to group Name1, Name2 and Name3 fields into one Name field as shown in table 2.5.

Table 2.5

Course Code	**Course Title**	**Name**
MTH 101	Calculus	Eze, Obi
MTH 101	Calculus	John, Bot
MTH 101	Calculus	Mike, Juliet
CS 102	Computer Applications	Olu, Ayo
CS 102	Computer Applications	Peter, Oke
CS 102	Computer Applications	Jim, Paul
STA 111	Probability	Bala, Ali
STA 111	Probability	Oluwa, Seun
STA 111	Probability	Eze, Obi

RULE 2: A column with atomic data cannot have several values of the same type of data in name column. We have to spilt the Name field into Last Name and First Name fields in table 2.6.

Table 2.6

Course Code	Course Title	First Name	Last Name
MTH 101	Calculus	Eze	Obi
MTH 101	Calculus	John	Bot
MTH 101	Calculus	Mike	Juliet
CS 102	Computer Applications	Olu	Ayo
CS 102	Computer Applications	Peter	Oke
CS 102	Computer Applications	Jim	Paul
STA 111	Probability	Bala	Ali
STA 111	Probability	Oluwa	Seun
STA 111	Probability	Eze	Obi

RULE 3: Each record is this table is not uniquely identified by a **primary key**.

We have to uniquely identify each table in as shown in Table 2.7.

Table 2.7

ID	Course Code	Course Title	First Name	Last Name
01	MTH 101	Calculus	Eze	Obi
02	MTH 101	Calculus	John	Bot
03	MTH 101	Calculus	Mike	Juliet
04	CS 102	Computer Applications	Olu	Ayo
05	CS 102	Computer Applications	Peter	Oke
06	CS 102	Computer Applications	Jim	Paul
07	STA 111	Probability	Bala	Ali
08	STA 111	Probability	Oluwa	Seun
09	STA 111	Probability	Eze	Obi

This table is now in its **first normal form** (**1NF**).

NOTE

To be 1NF, a table must follow these two rules:

- *Each row of data must contain atomic values. (Rule 1 & 2)*
- *Each row must have a unique identifier known as Primary key. (Rule 3)*

PRIMARY KEY

A **Primary Key** is a column in your table that makes each record unique.

> ***NOTE***
>
> *A **Primary Key** is used to uniquely identify each record. This means that the data in the primary key column cannot be repeated.*

Primary Key Rules

1. **A Primary Key cannot be NULL**
   *If primary keys can be **NULL**, it then means that it cannot be unique because other records can also be **NULL**.*
2. **A Primary Key must be given a value when the record is inserted**
   *When you insert a record without primary key, you run the risk of ending up with **NULL** primary key and duplicate rows in your table, which violates the **1NF***
3. **A Primary Key must be compact**
   A primary key should contain only the information it needs to be unique and nothing extra.
4. **A Primary key value cannot be changed**
   If you could change the value of your key, you would risk accidentally setting it to a value you already used. Remember, it has to remain unique.

Second Normal Form (2NF)

The concept of removing the delicacy of data comes in the **Second Normal Form (2NF).**

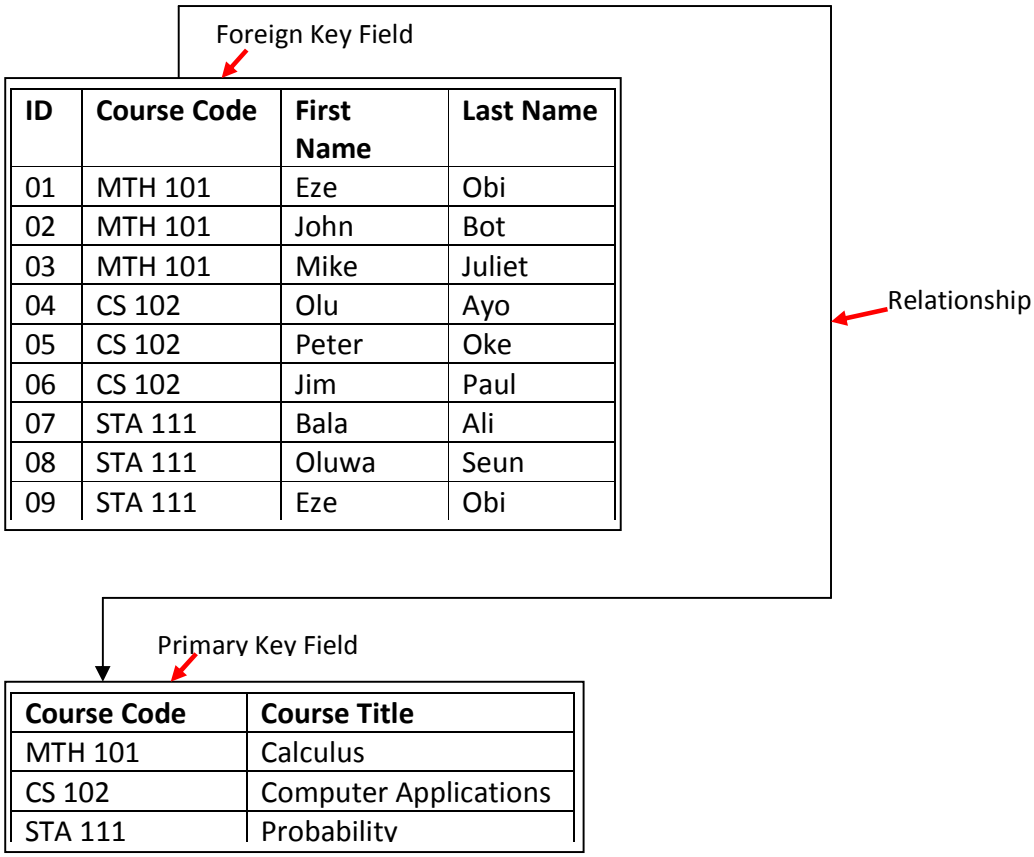
1. It should meet all the requirements of the **1NF**
2. It should remove subsets of data that apply to multiple rows of a table and place them in separate tables
3. Create relationships between the new table and their predecessors through the use of a foreign key.

The **1NF** deals with atomicity and non redundancy of data whereas **2NF** deals with the relationship between the composite key column and the non-key column. To achieve the next progressive level your table should satisfy the requirement of **1NF** then move towards the **2NF**.

Table 2.8

ID	Course Code	Course Title	First Name	Last Name
01	MTH 101	Calculus	Eze	Obi
02	MTH 101	Calculus	John	Bot
03	MTH 101	Calculus	Mike	Juliet
04	CS 102	Computer Applications	Olu	Ayo
05	CS 102	Computer Applications	Peter	Oke
06	CS 102	Computer Applications	Jim	Paul
07	STA 111	Probability	Bala	Ali
08	STA 111	Probability	Oluwa	Seun
09	STA 111	Probability	Eze	Obi

This Table is in the **1NF** but not in the **2NF**



ID	Course Code	First Name	Last Name
01	MTH 101	Eze	Obi
02	MTH 101	John	Bot
03	MTH 101	Mike	Juliet
04	CS 102	Olu	Ayo
05	CS 102	Peter	Oke
06	CS 102	Jim	Paul
07	STA 111	Bala	Ali
08	STA 111	Oluwa	Seun
09	STA 111	Eze	Obi

Course Code	Course Title
MTH 101	Calculus
CS 102	Computer Applications
STA 111	Probability

Figure 2.1

In Figure 2.1, both of the tables are now in the **2NF**

FOREIGN KEY

A **Foreign Key** is a column in a table that references the **PRIMARY KEY** of another table.

Relational Database Management System (RDBMS)

As you may know, Access is a relational database management system.

Relationships give you the flexibility to combine information from many tables.

To access information from one table, you must be able to join the tables in queries, or link them in a form of report. Two tables can be joined, or linked, only if they have a common field or fields.

Relationships

There are three types of relationships that can exist between two tables:

- one-to-one (1:1)
- one-to-many (1:M)
- many-to-many (M:M)

Only tables with 1:1 or 1:M relationships should be joined.

One-To-One Relationship

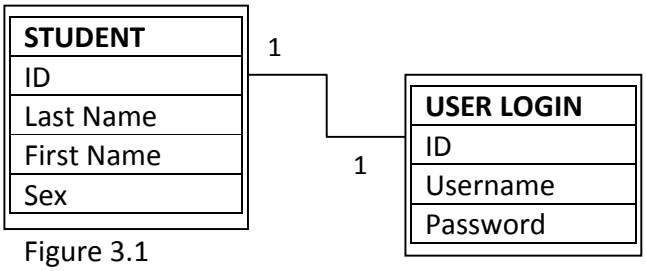


Figure 3.1

These two tables in Figure 3.1 have a one-to-one relationship. A one-to-one relationship exists when one of each data values in the common field of the first table matches one of these fields in the second table.

In this case, the ID field appears in both the STUDENT and USER LOGIN tables, so it can be used to join the two tables. However, One Student owns one User Login and one User Login is owned by one Student.

One-To-Many Relationship

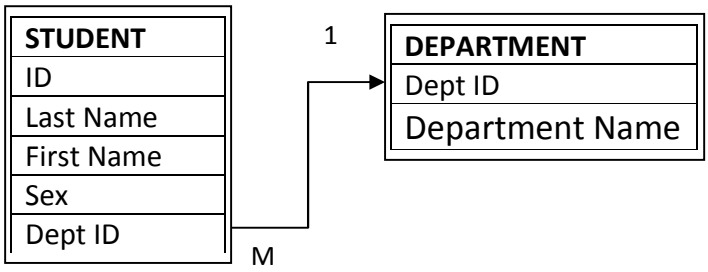


Figure 3.2

A one-to-many relationship exists when one of each data value in the common field of the first table matches more than one of the values in the second table.

For example, in Figure 3.2, Many Students can belong to one Department and only one Department can have many students.

Many-to-Many Relationship

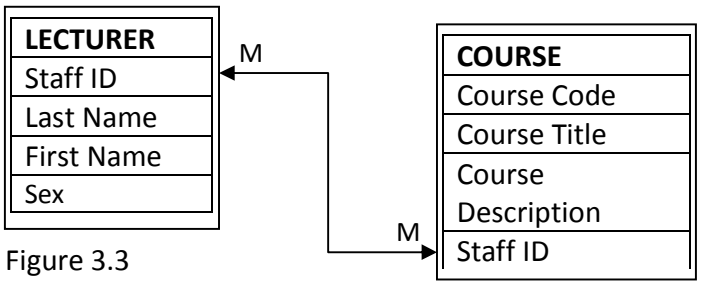


Figure 3.3

Many Lecturers can take many courses and a course can be taken by many lecturers.

Tables with a many-to-many relationship in Figure 3.3should not be joined. Avoid many-to-many relationship by redesigning tables, or by linking each table to a third, intermediate table as shown in Figure 3.4.

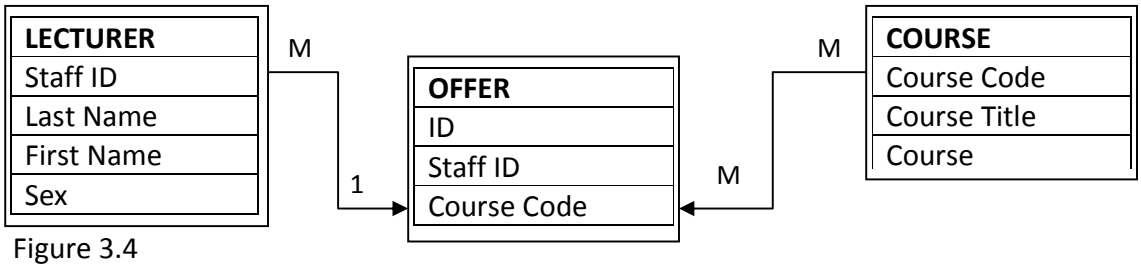


Figure 3.4

CHAPTER FOUR

Examining a Table

Access table contains all of your data and are the foundation of your databases.

In this chapter, we will learn how to navigate in a table. Then we will examine the table in both Datasheet view and Design view to compare the table structure to its underlying design.

When you open a table in Access, the table is displayed in **Datasheet view**.

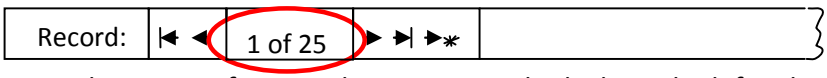
Datasheet view provides a tabular view of your data, where each column is a field and each row is a record.

HANDS ON

- *Table automatically Appears in a Datasheet View with Table1*
- *Rename Table1 to Student Table*
- *Click on Add New Field to Add fields*

Table Navigation Methods

As you can see in the status bar, the table is displayed in Datasheet view. Observe the record indicator in the status area. The active record is 1 of 25.



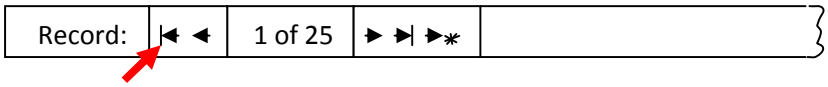
Another way to determine If a record is active is to look along the left side of the table. An orange colored square in the selection area, called the record indicator, points to the active record.

When a table is larger than your screen, you can use the vertical and Horizontal Scroll bars to view fields or records that appear beyond the current viewing area. However, when you use the scroll bar, the active record does not change.

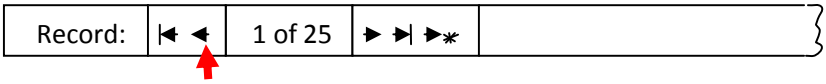
One way to move to another record is to use the Up and Down arrow keys on the keyboard.

You can also use the **Page Up** and **Page Down** Keys to move an entire screen at a time.

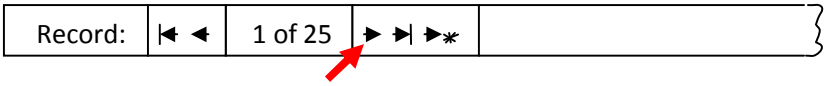
The Button on the far left of the status area is called the **First Record** button and its used to move to the first record in the table.



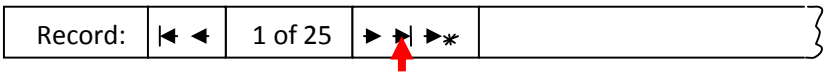
The **Previous Record** button is next to the **First Record** button. You can use this button to move the record indicator to the previous record in the table. The **Previous Record** button dims when there is no record before it.



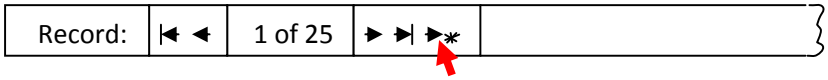
The next button in the status area is the **Next Record** button. As you probably guessed, the **Next Record** button is used to move to the next record in the table.



The next button is called the **Last Record** button is used to jump to the last record in the table.



The **New Record** button is the last button on the status area. You can use it to add a new record to the end of a table.



Suppose you wanted to move to a particular record in the table. You could do this by typing the record number in the specific Record Number field in the status area and pressing the enter key.

Datasheet View

The **Datasheet view** displays tables in form of rows and columns like Excel sheet.

Design View

The **Design View** is used to define and modify tables and other design elements.

In the **Design View**, the Field List pane in the upper portion of this window contains the name of every field in the table. It also contains the field data type and an optional field description.

In the **Design View** the key symbol next to any field name indicates the primary key.

> ***NOTE***
>
> *To toggle to **Design View** from the **Datasheet View,** you click on the **View Button.***
>
> *Also notice that the appearance of both the **View Button** in the **Tool Bar** and the **Menu Bar** changes when you are in the view. This is a prove that both the **Tool Bar** and **Menu Bar** are adaptive.*

> ***TIPS***
>
> *When you design a table, you decide what fields to include, the order in which the fields will appear in the table, the type of data each field will contain, and the size and format of the fields.*

Primary Key

A **Primary Key** is a field (or combination of fields) that uniquely identifies each record in a table. Every table includes a primary key.

Data Type

The **Data Type** determines the kind of data that can be entered in a field and the kind of operations Access can perform on the data.

Field Property Pane

This pane is used to define the characteristics of each field in a table.