

Part 1: Vocabulary

Microsoft Access: It is a *Database Management System (DBMS)* - a system to manage your *data* or *information*. You can think of a database as an electronic filing system. It organizes data in one place, but you can retrieve it anywhere you need it. By storing the data in one place, you are only required to update it in one place and the data is automatically updated everywhere. An Access Database is a collection of *Objects*. In this course, we will deal with four important *objects* (Table, Query, Form, & Report) explained in details below. (Open [Personnel.accdb](#) in Access Folder under My Document folder. Click on *Options* button on the *Security Warning* bar and select the radio button to *Enable this Content* and say *OK*.) Let's take a look at each object. Double-click on an object to open in its own tab.

1. **Table:** It consists of *columns (fields)* and *rows (records)* to organize your data. It looks a lot like an Excel spreadsheet, but it is not the same. All databases contain one or more tables. Tables are the main elements of a database. A database should have a separate table for every major subject.
 - **Field:** A field is a single piece of information. Think of it as a category of information as a column in a table. Each *column* is a field. For example: ID, company name, phone numbers.

Zip	Phone	HireDate
07087	201-863-4283	11/13/1989
12390	212-790-1253	1/3/1990
07032	201-439-6620	10/4/1985

The column containing phone numbers is a **field**

- **Record:** A collection of fields. For example, an employee's record might contain their ID, name, and an address. Think of it as a *row* in a table.

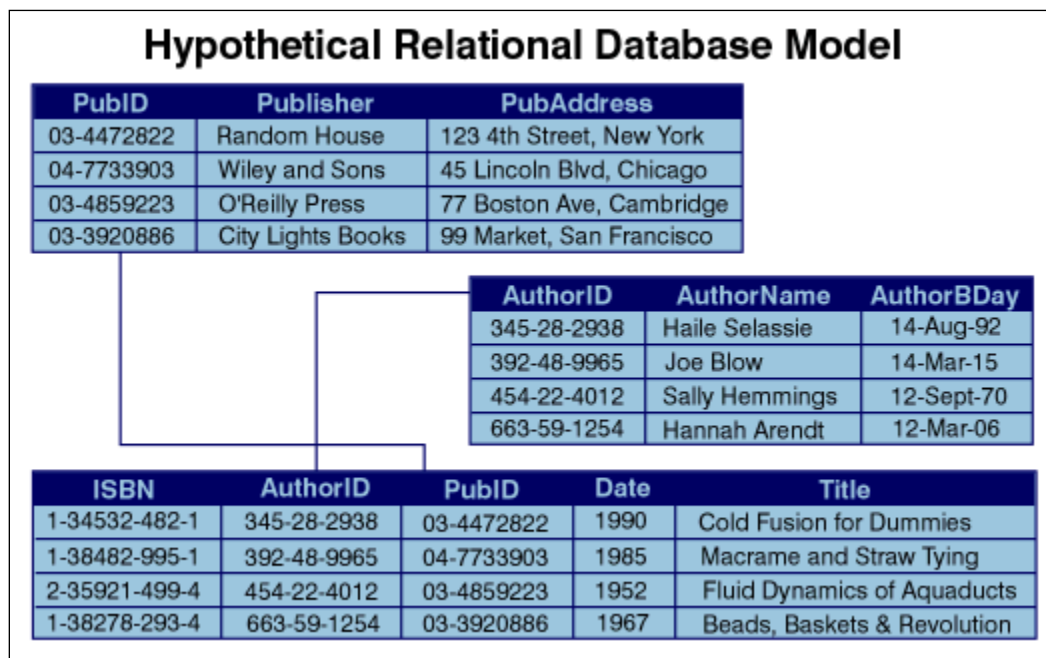
EmpID	FirstName	LastName	Address	City
EN1-10	Carol	Schaaf	2306 Palisade Ave.	Union City
EN1-12	Gayle	Murray	1855 Broadway	New York

The row containing information about Carol Schaaf is a **record**.

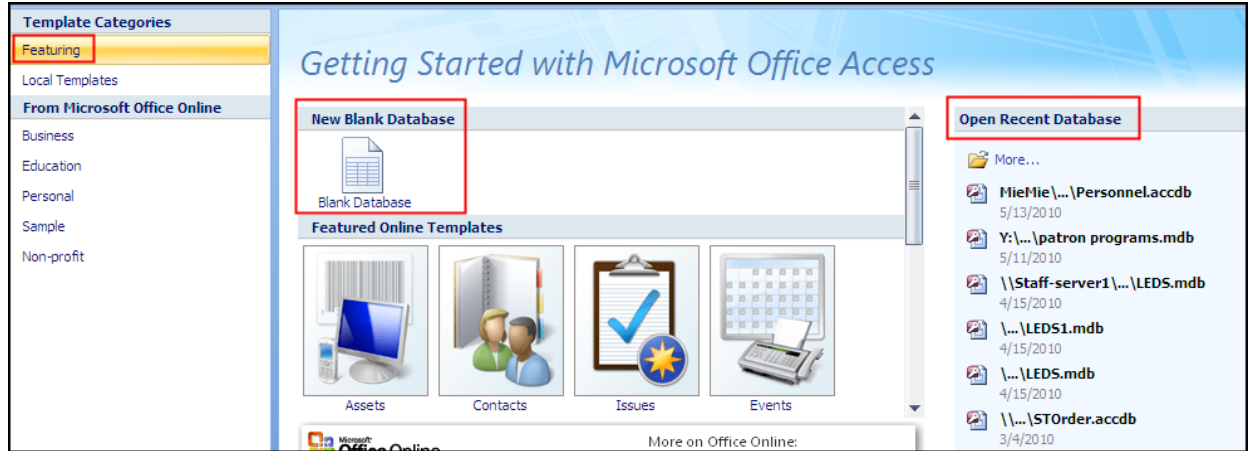
2. **Query:** It is a tool to ask a question of a database or to selectively extract information based on user-defined criteria. It can combine data from different tables, update data or perform calculations on your data. Examples: *How many students score between 70 and 80 on mid-term? How much money did we spend on blue copier paper during the months of Mar-May in 2008?* Extracted information may be used in *reports, forms*, or as the *basis* for other tables.
3. **Form:** It is primarily an input device for users to view, enter and edit data in tables, without having to understand table structures. Data is not stored in forms but in underlying tables.
4. **Report:** It is a tool to organize data in a specific layout for printing purpose. It may show only a subset of a data (*recordset*). They turn the data in your tables and queries into documents by calculating and displaying results. Example: *Calculate late fees for those customers who are late in payment for more than 90 days!*

So by having a database, you can store and retrieve data, sort it, analyze it, summarize it, and report results in moments. It can combine data from various files, avoiding duplication of data. By not duplicating, you are reducing file sizes. Not only that, you can keep the data consistent by entering only once. Overall, using Access can even make data entry more efficient and accurate. A key benefit of Access is avoiding the complications of multiple lists.

Relational Database: Access creates a *Relational Database* which means that data is stored in various separate *tables* by *subject* or *task*, but the data is related and can be brought together by creating “joins” in ways that you specify. Relationships among tables link data from individual tables to increase its usefulness.




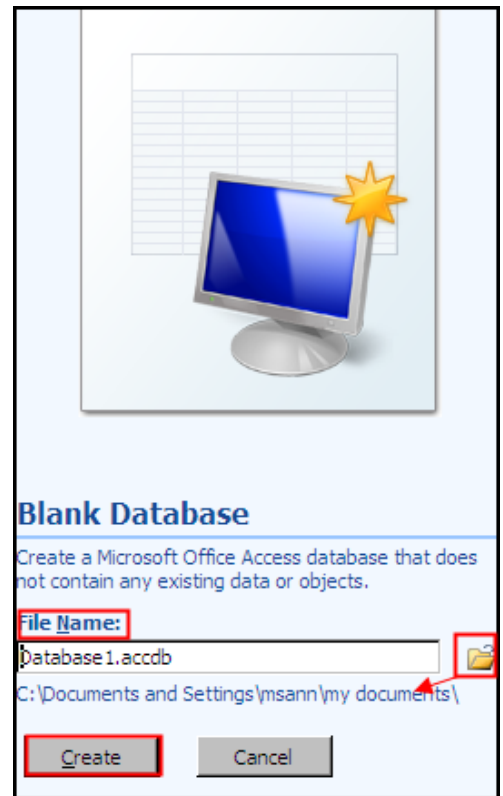
Getting Started: Screen Elements



1. **Featuring:** May create a database from an already pre-built template.
2. **New Blank Database:** Click to create blank database from scratch.
3. **Open Recent Database:** Appears on the right task pane. It shows recently used databases. May click to open an existing database.

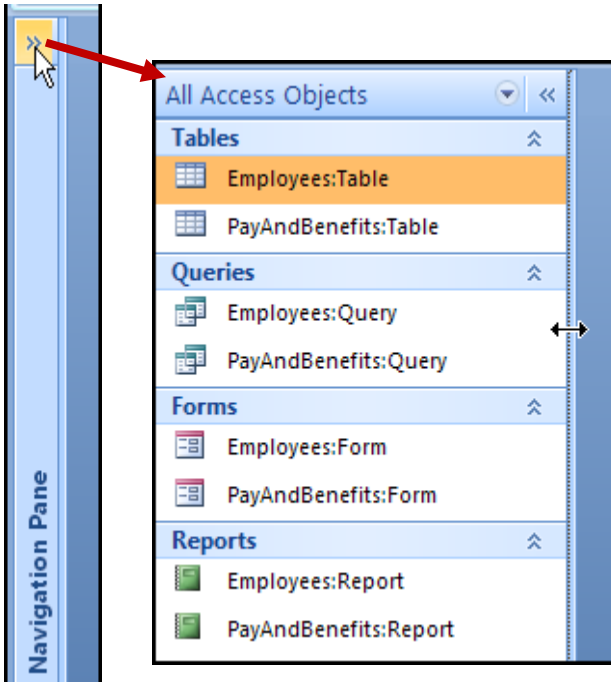
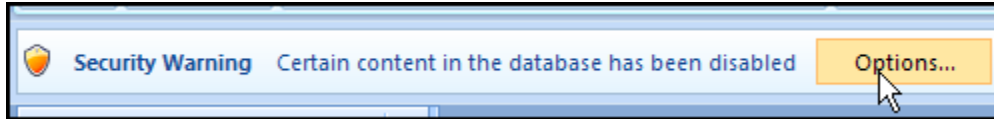
By clicking on **New Blank Database** button, the right task pane changes as shown on the right:

- Enter a database name in **File Name** box.
- Click on  to select where to save database. The file path below should indicate the location where database will be saved.
- Click on **Create** button.

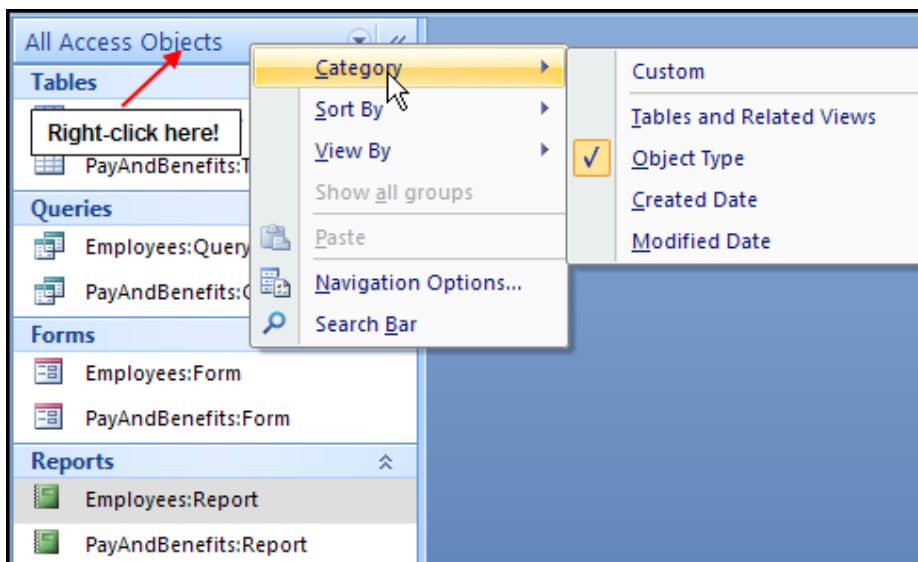


Part 2: Navigating in Access

- To open an existing database, click on the any database appears on the right task pane. You will see the security alert warning right below the ribbon. Click on the **Options** and **Enable the Security Content** before you work with the database.

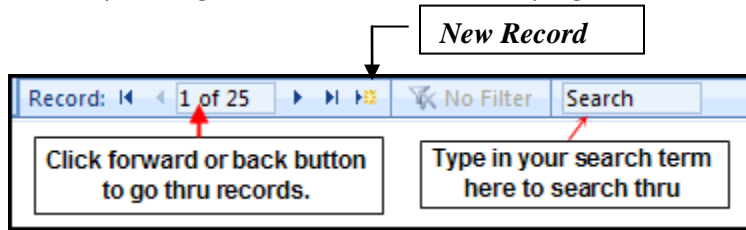


You will see the Navigation Pane on the left side of the window. By clicking on the double arrow or by using Toggle Key F11, you can open or close the Navigation Pane. In the Navigation Pane, all the objects that are defined in the database are displayed in categories. In the figure on the left, **All Access Objects** is the category, and **Tables**, **Queries**, **Forms**, and **Reports** are the groups. Each category is organized into groups that are displayed as bars. You can also change the width of the Navigation Pane by placing the mouse on the right edge; when it changes to a double-sided arrow, drag the mouse to left or right to decrease or increase.



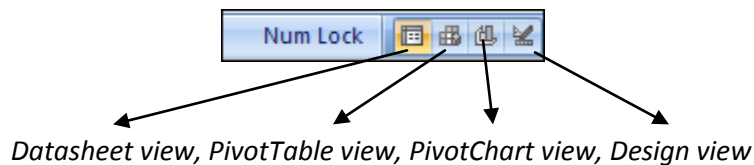
You may also right-click on the top of the Navigation Pane to display the short-cut menu to specify how objects are displayed and sorted.

Double-clicking on each object in the Navigation pane will open the object in the *Viewing Window* on the right in the *Tabbed Document Window Viewing* feature. All database objects: tables, queries, forms and reports will be opened as tabs in the same window. Each tab can be closed by clicking on the 'x' button on the top right of the viewing window.



Each tabbed document window will have its own *Record Navigation Bar* at the bottom with *Search* box when relevant.

The *Microsoft Office Status Bar* at the right bottom corner will let user switch between different views in which the database objects can be viewed. See below.

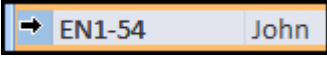




Open [EnterData.accdb](#) database. Enable the Security Options as usual. Scroll through it to see the data that the table contains. Use these following keyboard functions to navigate *within* records.

- F2:** Toggles between 'edit' and 'selection' modes.
- Home:** moves the cursor to the beginning of the current record.
- End:** Moves the cursor to the end of the current record.
- Ctrl-Home:** Moves the cursor to the first field of the first record.
- Ctrl-End:** Moves the cursor to the last field of the last record.
- Tab:** Moves one field to the right.
- Shift-Tab:** Moves one field to the left.

Just like in Excel, point to the left of the first column in the margin area will turn the mouse into a dark arrow. Clicking on the dark arrow, you will select the entire record in that row. Likewise, selecting on any header field and clicking the down arrow will select the entire field in that column. Let's add a new employee in this table by using the following data. Click on to add a new record. Use *Tab* key to move to the next field on the right.


EmpID: EN1-54
 FirstName: John
 LastName: Wu
 Address: 445 Honeymoon Lane
 City: New York
 State: NY
 Zip: 13398
 Phone: (212)745-1244
 HireDate: 10/21/1995

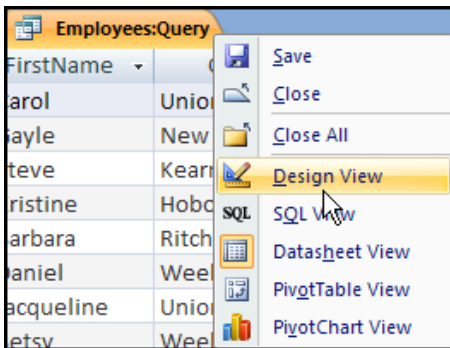
Deleting Records: Click on the *Record Selector*  for the record you wish to delete and hit the 'delete' key  on Home tab. Accept the resulting prompt. **Records are irreversibly deleted. Always backup your current data before deleting.** (Instructions: Go to Office Button>Manage>Back Up Database to backup your data before making any irreversible deletes. A current date will be added to your database name and will be saved as a separate copy.)

Sorting Records: Click anywhere within the column you wish to sort. Click on either "Sort Ascending" or "Sort Descending" button . When a field column is sorted, the data contained in the other fields will realign as well, so as to keep records in tact.

How to Use an Existing Access Database

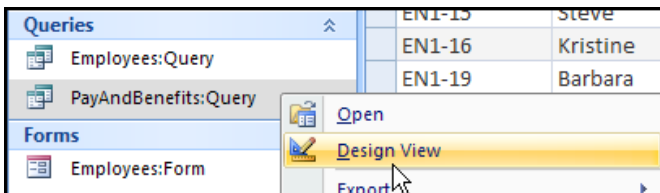
Open **Personnel.accdb** database. To view data in a table, or to run a query, or to view data using a form, or to view the data in a report, simply double-click the desired item on left navigation pane. Or right-click the desired item and choose Open. Notice each item is opened in its own tab view and can be closed individually. Also notice that you can view items in different

views by simply clicking on the view buttons on the Status Bar  or by using any of the methods below.



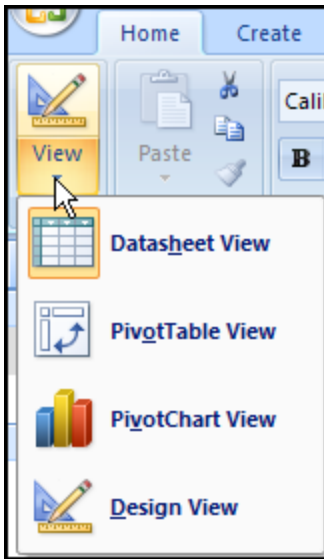
Right-click on the tab and select any view

OR



Right-click on an object in the navigation pane and select a different view

OR



Simply click on the drop-down arrow on **View** under **Home** tab and select a different view

Part 3: Database Creation

Designing database files need to be carefully thought out. The following steps are clearly stated on Microsoft Access Online Help page.

1. *Determine the purpose of your database (who, what, when, why, how)*
This helps prepare you for the remaining steps. For example:
 - The database will contain information on employees and benefit plan enrollment.
 - It will be used by HR staff only.
2. *Find and organize the information required*
Gather all of the types of information you might want to record in the database, such as employee records and their benefit plans.
3. *Turn information items into columns (Fields)*
Decide what information you want to store in each table. Each item becomes a field, and is displayed as a column in the table. Ask other people who will use the database for ideas as well. For example, an Employees table might include fields such as Last Name and Hire Date. Breakdown information into smallest units such as - Employee Names into Last and First Name fields.
4. *Divide the information into tables*
Divide your information items into major entities or subjects, such as Employees or Benefit Plans. Each subject then becomes a table. A table can have multiple fields.
5. *Apply the normalization rules*
A fundamental principle of data normalization is the same data should not be stored in multiple places. Technically, the process of breaking your data down into related tables is called normalization. Apply the data normalization rules to see if your tables are structured correctly. Make adjustments to the tables, as needed.
6. *Specify primary keys*
Choose each table's primary key. The primary key is a column that is used to uniquely identify each row. An example might be Employee ID. **The primary key is values that should never be blank, that rarely change. And you DO NOT allow duplicate values.**

7. *Set up the table relationships*

Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.

8. *Refine your design*

Analyze your design for errors. Create the tables and add a few records of sample data. See if you can get the results you want from your tables. Make adjustments to the design, as needed.

Your database will contain tables, forms, reports and queries to support those forms and reports. The most complex part of creating database is designing the tables. Your plan should include how many database files to have, how many tables to include within a file, what fields to create, etc. Some ideas to bear in mind when designing tables:

- Who will need to use data from that table and/or what type of questions will be asked of the table?
- Are the fields broken down to their least common denominator? Example: Are the names broken down into Last Name and First Name Fields?
- Avoid data duplication in *multiple* tables at all cost.

Example of Bad Data:

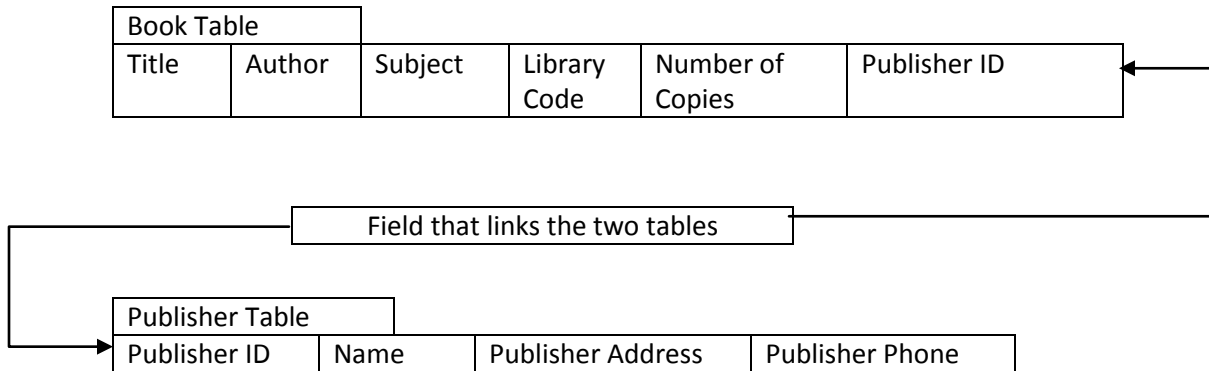
Book Table							
Title	Author	Subject	Library Code	Number of Copies	Publisher	Publisher address	Publisher phone

In above example, placing the Publisher information in the Book Table can create following problems:

1. Problem maintaining your data: In this table, the publisher information fields are in many book records (i.e. repeated information). That means, you will have to track down all those individual book record when/if publisher information needs to be updated. It's cumbersome.
2. Accuracy of your data: Having to key in the same detail in several records can lead to errors.
3. Size of your database: The details on publisher information are repeated many times resulting in unnecessary large database.

Solution to this Problem:

1. Create two tables: a table for the Book and a new table for the Publisher.
2. Create a new field, Publisher ID, in each table and assign that field as a primary key (unique field) to identify each publisher uniquely. This field will then be used to link the tables when and as required.

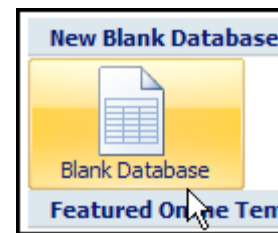


By doing so will have the following benefits:

- Each set of publisher details is stored (and therefore keyed in) only once eliminating possible human error. Should any of the publisher details change, only one record needs to be updated in the Publisher Table.
- The Book Table will be considerably smaller in size without needing to have repeated information on publisher details.
- In your Book Table, if a publisher is wrongly identified in a book record, you have only one field to correct, rather than all of the publisher's fields.

Creating a Blank Database

- Open Microsoft Access Program on your desktop.
- Click on Blank Database button on the top left. The right pane will open.
- Select the location to save your database and give it a name, Click "Create". We will actually create one on page 12.



Scenario:

We want to create a simple library database. We will go through Database Creation steps laid out in Step 3 above.

1. The purpose of the library database is to:
 - a. Record, update and edit details of all the books held in the library – such as information on title, author, publisher, subjects, etc.
 - b. Give the physical location of the book in the library, such as row, shelf numbers.

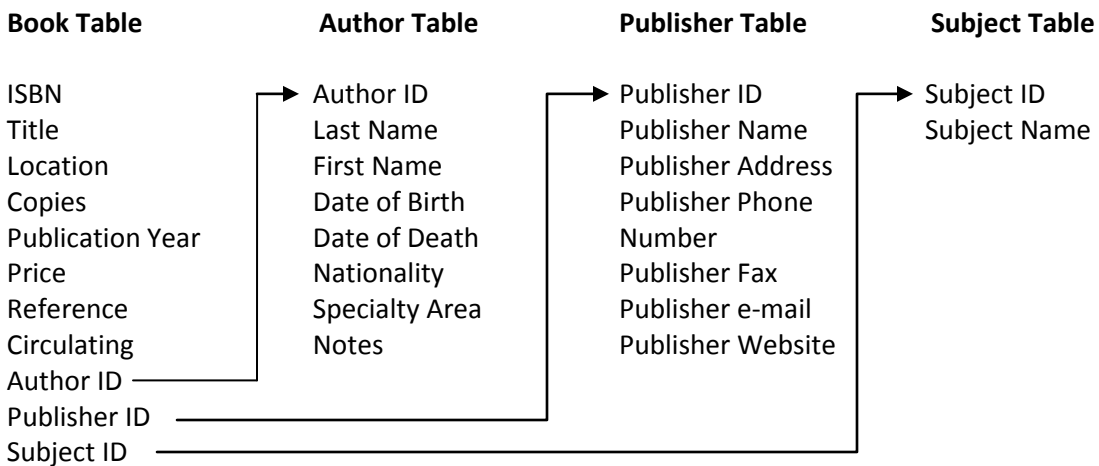
By having the above information, we will be able to provide our patrons with reports:

- a. A list of all the books by certain author or by a certain publisher
- b. A list of all the books in certain subject area: such as all cooking books
- c. Physical location(s) of a certain book or list of books

2. Following information is collected to create this database:

ISBN
 Book Title
 Author (Name, date of birth, date of death, nationality, specialist area, and any other pertinent information)
 Publisher (Name, address, phone, fax, email, website)
 Subjects
 Location (row and shelf numbers)
 Number of copies
 Publication Year
 Price
 Reference or Circulating

3. We will turn that above information into *Fields*. You want to make sure that information is broken down into smallest units – such as Author’s name into Last Name and First Name.
4. We will put the *fields* into logical groups or categories and create necessary *tables*. A table can have multiple *fields*. So we may divide the *fields* into possible *tables* as below.

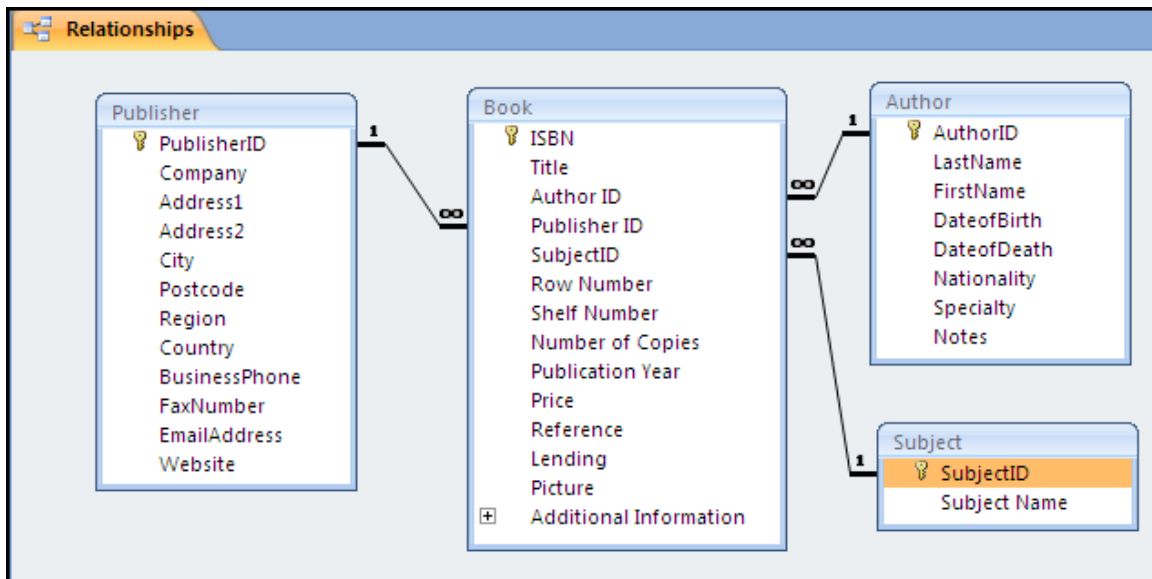


While dividing the *fields* into *Tables* above, consider following facts:

- You do not want to duplicate information – for example, you do not want to repeat Author, Publisher, or Subject information for every book you record. So you will divide that detailed information into separate tables. We will then create relationship between these tables and pull the needed information into the Book Table. By doing so,

you are using a *field* (Author ID, Publisher ID, or Subject ID) in more than one table and it is preferable to do so rather than redundant information in the *Tables*.


- Make sure your *table* structure includes the *fields* required for the reports you intend to produce from this database.
 - Do not worry about calculated *fields* in your *tables*. You can perform calculations when you need them in your *queries, forms* and *reports*.
5. As we created necessary fields above for our tables, we have applied the normalization rules by eliminating duplication of data.
 6. We will assign *Primary Keys* where necessary. *Primary Key* is a *field* (or combination of *fields*) that would uniquely identify each record held within each *table*. In our example above, in *Book Table*, ISBN is the most unique field since there is no two ISBN alike. In *Author, Publisher, & Subject Tables*, there is no unique information already in original fields. Even though Publisher phone numbers seem unique, it can change often. So we would assign unique ID in each table by using auto number data type thus born *Author ID, Publisher ID, and Subject ID fields*.
 7. Based on what we have so far, we can visualize how these tables will relate to each other as follows. The *Book Table* is related to each of the other tables within the database as indicated below. The lines between the tables indicate which fields are related to each other and they are called *join lines* in Access. The type of relationship in each case is called one-to-many (the most common type of relationship between tables). Between the *Publisher Table* and the *Book Table*, one record in each *Publisher Table* is related to many records in the *Book Table*. Same rules apply to other relationships between the *Book Table and Author & Subject* tables. We will simply add locations of items as Row Number and Shelf Number in *Book Table* since there is no benefit of separating these fields out into a separate table.

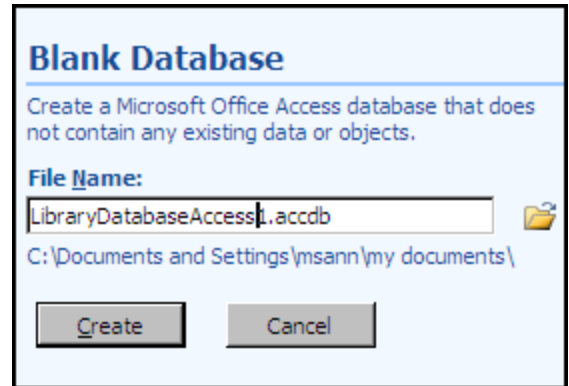


8. We will take a look at refining the design, if necessary, after we enter a few data.

Creating an Access Database

You have designed the database on paper. Now, it's the time to create one in Access. Follow these steps below.

1. Open Access. Click on Blank Database. Click on Open File icon  on right and select "Desktop" in Save in location and name the database "LibraryDatabaseAccess1" and save as Microsoft Office 2007 Access Database (.accdb) and click on "Create" button to save it.

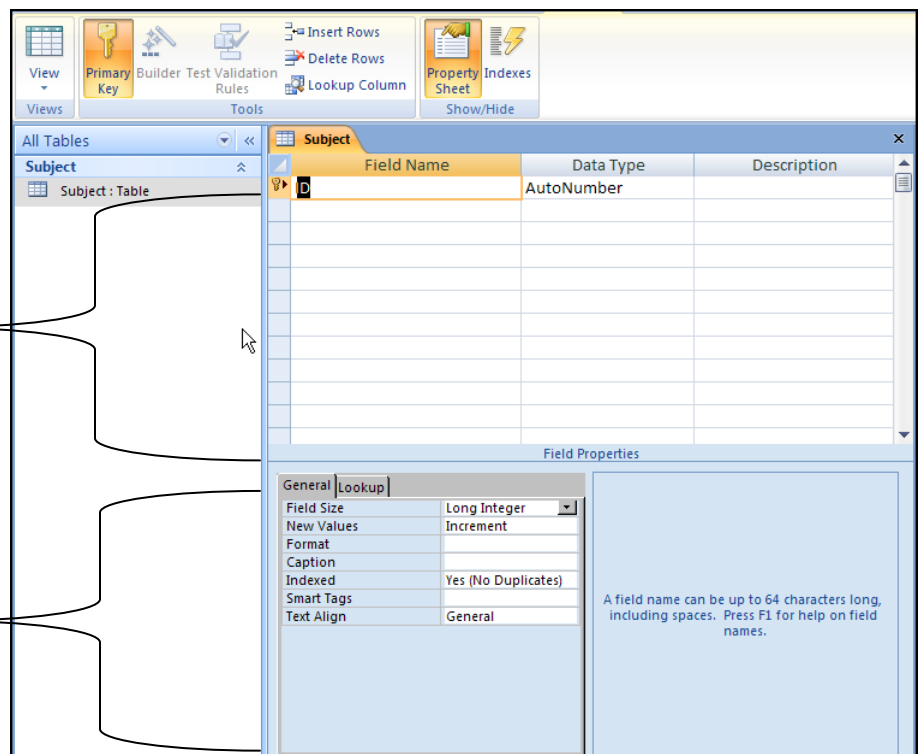





2. Create Tables. After step 1 above, a new table is produced automatically, and it is displayed in Datasheet view – which is used for data entry and editing purposes. We will create our table in Design view to be able to set up the structure of each table. Only in Design view, can we add and delete fields, specify data types and assign properties to the fields. Click on the "Design View" button on the top left. You will be required to give a name and save the table. Name the table "Subject".
- Design View has two sections: the top half to create "Fields" by using "Field Names" and define their data type, the bottom half to assign "Field Properties". *Note:* "Fields names" can be as long as 64 characters but long field names increase the odds of user error since they are referred to by name with every expression, function, or control that accesses them.



Create "fields" and define their "data type" in this pane.

Input "field properties" in this pane that would determine the nature of the data accepted into a field.



Design View vs. Table View: On the bottom right screen, you can switch views by clicking on different button  "Design View"  and "Datasheet View" .

- **Data Type:** Selecting an accurate data type for each field is vital to your data. Data type will not only determine what sort of data will be accepted into a field but also how the data can be manipulated. See data type descriptions in the table below.

Data Type (Source: Microsoft.com)

Data type	Description	Example
Text	A Text field can store any character or number (that does not require calculation) and is limited to a length of 255 characters.	A product name, phone number, part number, postal codes
Memo	Use for lengthy text and numbers. Stores up to 63,999 characters.	Notes or descriptions
Number	A Number field can store only numbers, and these numbers can be used for calculations except calculations involving money (use Currency type).	A unit count, such as 200
Date/Time	A Date/Time field stores a date and a time.	An order date, such as 10/10/2008 5:21 P.M.
Currency	A Currency field stores currency values, numbers, and decimal-place formatting, which can be used in calculations.	An item price, such as \$41.99
AutoNumber	Use for unique sequential (incrementing by 1) or random numbers that are automatically inserted when a record is added.	A customer identification number, such as 123243
Yes/No	Boolean Data	Yes and No values or fields that contain 1 of 2 choices such as On/Off, True/False
OLE Object	An OLE object field stores objects created by programs other than Access and linked to or embedded in an Access table.	An attachment, such as an Excel worksheet, a Word document, a graphic, or a sound
Hyperlink	Use for hyperlinks. It can be a UNC path (e.g. C:\USER\DOCS\LETTER.TXT) or a URL.	A website address or a file location on your computer
Attachment	Any supported type of file	Images, spreadsheets, documents, charts, etc.
Lookup Wizard	Use to create a field to select from another list of values or from a separate table.	List of names or Products to select rather than user to type in

- "Description" option box is used to provide the user with a detailed description of the type of data contained in the field. The content of the Description field in the Properties

dialog box will be displayed under Description in the Database window on the Status Bar (bottom left). Forms created based on those tables with descriptions and they will carry those descriptions on them. It is a simple and effective way to provide the user with interactive help as they work with tables and forms.

Field Property	Description	Notes
Field Size	The Field Size property sets the maximum number of characters in a field for text and number.	Text from 1 to 255 characters. Though smaller limits save computer memory, you want to make sure your largest data item will fit.
	Number field sizes: <ul style="list-style-type: none"> • Byte (single byte) • Integer (2-byte) • Decimal • Long Integer (4-byte) 	Values: <ul style="list-style-type: none"> • -0 to 255 • Positive or negative numbers (no decimals) • Up to 28 digits after the decimal point • Twice storage as Integer
Format	The Format property specifies how the field should appear on the screen.	An order date will be displayed as "June 14, 2008".
Decimal Places	For Number and Currency fields	Specify the number of decimal places to display.
Input Mask	The Input Mask property specifies a pattern for data entry into a field. Can use on Text, Number, Date/Time, and Currency data types.	An order date must be entered in the form "06/14/2008".
Caption	The Caption property provides the label for a field used on a <i>form</i> or <i>report</i> .	The field's label is "Employee's hire date".
Default Value	The Default Value property provides the default information for all new records.	The order date is always today's date unless edited. OR to ensure that Access always displays the current system date in the Date field, use the formula: =Date(). Note: when users need to enter a different value in the cell, they can type over the default one.
Validation Rule	The Validation Rule property validates data entered by users before saving data.	The quantity ordered must be greater than 1.
Validation Text	The Validation Text property displays a message when data is rejected as invalid.	"Please enter an order quantity greater than 1."
Required*	The Required property defines a field as requiring data for the record to be complete.	e.g. A purchase order must have a ship-to address. It simply determines if a Null (blank) entry will be allowed or not. Use this to represent a <i>missing or unknown</i> value.

Field Property	Description	Notes
Allow Zero*	The Allow Zero property allows a record to be complete with a field that contains no data. Only applies to Text, Memo, & Hyperlink table fields.	Not all customers have an e-mail address, and some records will not contain any data in that field. Use a zero-length string (a single space) to represent the <i>absence</i> of a value.
Indexed	The Indexed property speeds up searching and can ensure that data is unique.	Frequently used data items, such as part number and city codes, will be easier to locate and manage.
Unicode Compression	Text, memo and hyperlink fields	Indicate whether you want Access to save space if only plain text is entered. Leave at default – yes
SmartTags		Allows SmartTag to be applied to field.

*If you want Microsoft Access to store a zero-length string instead of a Null value when you leave a field blank, set both the AllowZeroLength and Required properties to Yes.

More on Input Mask:

An Input Mask allows you to control what values a database user can enter into a field. Example, an Input Mask for a phone number field might appear as: (____)____ - _____. Access provides many predefined input masks for most situations but you can create your own customized masks. The Input Mask Wizard is only available for text and date fields. For numeric fields, you must enter the formatting symbols yourself.

Input Mask Symbols	
Symbol	Description
0	Digital 0 to 9 (required)
9	Digital 0 to 9 (optional)
A	Letter or digit (required)
a	Letter or digit (optional)
#	Digit or space
&	Any character or space (required)
C	Any character or space (optional)
L	Letter A-Z (required)
?	Letter A-Z (optional)
>	Make following characters uppercase
<	Make following characters lowercase
!	Causes the field to be filled from right to left, rather than left to right. The exclamation point can be inserted either at the beginning or end of the mask.
\	Causes the following character to be displayed as a literal character. Example, \L is displayed as L. Entry is required.
""	Used to display any text that you want users to see
Password	Creates a password entry field in Design mode. When users type in this field, Access stores the characters but displays asterisks "*" .

A few examples of Input Mask^{Microsoft.com}:

THIS INPUT MASK	PROVIDES THIS TYPE OF VALUE	NOTES
(000) 000-0000	(206) 555-0199	In this case, you must enter an area code because that section of the mask (000, enclosed in parentheses) uses the 0 placeholder.
(999) 000-0000!	(206) 555-0199 () 555-0199	In this case, the area code section uses the 9 placeholder, so area codes are optional. Also, the exclamation point (!) causes the mask to fill in from left to right.
(000) AAA-AAAA	(206) 555-TELE	Allows you to substitute the last four digits of a U.S. style phone number with letters. Note the use of the 0 placeholder in the area code section, which makes the area code mandatory.
"SSN"999\-99\-9999	SSN 231-54-6226	Quoted SSN is a literal entry followed by the number placeholder 9 with literal dashes in between.

More on Validation Rule:

There are three types of validation rules.

Field Level: This rule is applied when you enter data into the field. You cannot tab to the next field until you enter something that satisfies the rule, or undo your entry.

Table Level: The rule is applied after all fields have been entered, just before the record is saved. Use this rule to compare values across different fields, or to delay validation until the last moment before the record is saved.

Form Level: Validation rule may be applied to object controls on a form. The easiest and fastest way to apply a validation rule to a form is to first add the rule to the underlying table field, and then use the automated form-creation tools that Access provides to create a form.

Validation rules can be Relational or Boolean operators. A few examples below:

OPERATOR	FUNCTION	EXAMPLE
=	Equal to	= "ASCPL" Only allow the value "ASCPL"
>=	Greater than or equal to.	[EndDate]>=[StartDate] Enter an ending date on or after the start date.
<=	<=50	Allows values 50 or lower
>	Greater than	>1 value must be greater than 1
<	Less than	<Date() i.e. <current date The date cannot be in future, such as Birth date
<>	Not equal to	<>0 Enter a nonzero value
AND	Specifies that all the data that you enter must be true or false within limits that you specify.	>=#1/1/2009# AND <#12/31/2009# Date must occur in 2009 Note You can also use AND to combine validation rules. For example: NOT "UK" AND LIKE "U*" .

OPERATOR	FUNCTION	EXAMPLE
OR	Specifies that one or more pieces of data can be true.	"M" OR "F" Enter M for Male or F for female
NOT	Tests for converse values. Use before any comparison operator except IS NOT NULL.	NOT >10 The same as value must be <=10
IN	Tests for values equal to existing members in a list. Comparison value must be a comma-separated list enclosed in parentheses.	IN ("Tokyo","Paris","Moscow")
BETWEEN	Tests for a range of values. You must use two comparison values — low and high — and you must separate those values with the AND separator.	BETWEEN 100 AND 1000 (the same as: >=100 AND <=1000)
IS NOT NULL	Forces users to enter values in the field. This is the same as setting the Required field property to Yes . However, when you enable the Required property and a user fails to enter a value, Access displays a somewhat unfriendly error message. Typically, your database is easier to use if you use IS NOT NULL and enter a friendly message in the Validation Text property.	IS NOT NULL
LIKE	Matches pattern strings in Text and Memo fields.	LIKE "Geo*"

Keep these rules in mind as you create expressions:

- Surround the names of table fields with square brackets, like so: **[RequiredDate]<=[OrderDate]+30**.
- Surround dates with pound signs (#), like so: **<#01/01/2007#** (In 2007, # signs will be placed automatically.)
- Surround text values with double quotation marks, like so: **IN ("Tokyo","Paris","Moscow")**. Also, note that you separate items with commas, and you place lists inside parentheses.

Author Table:

Create Eight Fields:

Author ID: AutoNumber; set this key to be the primary key

Last Name: Text; limit to 20 characters; Indexed = Yes; Duplicates OK

First Name: Text; limit to 20 characters; Indexed = Yes; Duplicates OK

Date of Birth: Date/Time – pick a format from the drop down list; Type in Input Mask code 00/00/0000;;# or click the Build button beside the Input Mask field and work through the Wizard (zero digits make the user required to fill in all digits and # sign indicates the spacing required for all those digits)

Date of Death: Applies same rule as *Date of Birth* above; Also place the validation rule to make [Date of Death]>[Date of Birth] at the Table Properties Level by turning on the Property Sheet from the Design Tab in the Ribbon

Nationality: Text; limit to 30 characters; Indexed = Yes; Duplicates OK

Specialty: Text; limit to 20 characters; Indexed = Yes; Duplicates OK; Enter Description: "Main area of author's work"

Notes: Memo for notable notes or long descriptions

Save and name the table as *Author* and close the Design view and enter the following data:

Author ID	LastName	FirstName	Date of Birth	Date of Death	Nationality	Specialty	Notes
1	Peterson	Brian			English	Romantic Fiction	Best seller "Hollywood Days"
2	McDonald	Alastair	1/12/1936		Scottish	Poetry	
3	Jackson	Marion	6/24/1955		English	Travel	
4	Adamson	Pauline			Australian	Children's Fiction	
5	Duncan	Wilma	7/4/1938		American	Travel	
6	Ferguson	John	4/3/1903	10/5/1988	Irish	Gardening	
7	Jackson	Allan	12/10/1940	10/2/1993	Scottish	Travel	
8	Smith	Dick			Canadian	Children's Fiction	Best seller "The Mountie"
9	Schmit	Hans	12/12/1952		German	Computing	PC applications
10	Camembert	Marion			French	Gardening	Best seller "Flowering Shrubs"
11	Meunier	Luc			French	Cooking	
12	Allan	Isabelle	5/10/1965		Scottish	Travel	
13	Stephen	Moira			Scottish	Computing	PC application
14	MacDonald	Donald	12/12/1920	3/1/1978	Scottish	Travel	
15	Williams	Peter	10/10/1930		Irish	Science	School textbooks - mainly physics
16	Borthwick	Anne			Welsh	Astronomy	TV personality
17	Ferguson	Alan	12/4/1945		English	Computing	Mainly Computer Science
18	Wilson	Peter	1/3/1930		Canadian	Children's Fiction	Famous for "Worst Wizard" books
19	Smith	Ann			American	Computing	PC applications
20	Watson	George	10/22/1945		Austrian	Travel	Writer of TV Books