Intro to databases with MS Access

There is a sense in which almost any non-trivial computer program is a database program – the sense that a computer program operates on some base of data. However, what we typically call a database program is one that operates on a base of highly structured data records, performing such operations as

- Sorting data
- Finding data – e.g., one might use a library catalogue, which should be regarded as a database, to determine if the library has a desired book in its collection.
- Filtering data – find a set of records that satisfy a given property. This is a generalization of the Find operation described above. E.g., one might ask what books by a given author are in the library’s holdings, and the library catalog database would then perform a filter operation to compile a list of all its holdings by that author.
- Creating customized reports

Databases you have probably used:

- A library’s catalog
- When you shop online, you interact with a vendor’s database of available products and services

You can start Access with a new database as follows: click Start, All Programs, Microsoft Office 2013, Access 2013. You might choose to start a Blank Desktop Database. Note you must save the database file when it is first created. Give the database a filename and choose
where to save it (disk and folder); click Create. This creates and saves the database. The screen you see appears as shown below.

The vertical panel at the left is the “navigation center” or “command center” or “control center” of the database. Eventually, this becomes a menu of the sub-documents (the “Access Objects”) of the database. By clicking on one of the sub-documents, you can open it for viewing and editing. Notice that this menu is initially empty, corresponding to the fact that we have just created an empty database.

If you go to the Create tab, you can choose to create, e.g., a “table” for your database. A table can be viewed in either the Datasheet view (which will appear if you click the Table button) or the Design view (which will appear if you click the Design button). The datasheet view is used to view and edit the data; the design view, for designing the fields
of the table. Notice the View button of the Home tab, which lets you toggle between these views of the table.

Let’s talk about the structure of a database and of a table. A database typically has at its heart at least one table. A table is a set of records; each record is made up of units of information variously called fields/components/members/properties/attributes. Every field has a data type corresponding to the nature of the data to be stored in the field. When you design a field, you can choose its data type from a listbox menu in the Data Type column of the design view (each row of the grid represents a field), as shown below.

Among the important data types:

- Short Text (called “Text” in older versions of Access) – typically used for non-numeric data, such as names, titles, etc. Notice the Field Size property. The value here (maximum permitted is 255) is
the maximum number of characters that can be stored in the field. E.g., if we used a field size of 10, we could only store the first 10 characters of a name in this field. Why not use the maximum field size of 255 in all cases, then? The larger the field size, the more memory your database uses.

- Long Text (called “Memo” in older versions of Access) is similar to Short Text, but the limit on the number of characters is not imposed by a field size.

- Suppose you want to present the data-entry user with a menu of selections for a field. E.g., you might have a Title field, for which a short list of titles such as Mr., Ms., Miss, Mrs., Dr., etc., are the appropriate selections. Offering a menu lets the user have the following advantages:
  - Selecting from a menu via a click of the mouse is often faster than typing.
  - Selecting from a menu is less error-prone than typing.
  - Offering a menu lets us standardize the representation of a value – e.g., in a Gender field, representing female by “F” only, rather than allowing all of “F”, “f”, “Fem”, “fem”, “Female”, or “female”. This, in turn, makes it much easier to design queries.

When you design the field, choose Lookup Wizard from the Data Type menu. This sets a “wizard” in motion, to guide you through the process of designing the menu. The dialogbox shown below appears.
You might choose the 2\textsuperscript{nd} option, to type in the menu values you want.

- Number – used for numeric data – data that we might wish to apply numeric operations to. Notice that the Field Size property for this data type is quite different than the Field Size property for Short Text. For Number, the Field Size requires the choice of a “sub-type”. The sub-types include
  - Byte – used for non-negative integer values between 0 and 255.
  - Integer – used for integer values between -32,767 and 32,767 – uses more memory than Byte
  - Long Integer – used for integer values between (about) -2 billion to 2 billion – uses more memory than Integer
- Single – short for “Single Precision Real Number” – used for real numbers (with the possibility of non-zero fractional parts, like 6.73). This sub-type offers 6 or 7 digits of accuracy in the sense of scientific notation, including the value 0 and a range of non-zero values with absolute values from about $10^{-37}$ to about $10^{37}$.

- Double – short for “Double Precision Real Number” – used for real numbers (with the possibility of non-zero fractional parts, like 6.73). This sub-type offers 13 to 15 digits of accuracy in the sense of scientific notation, including the value 0 and a range of non-zero values with absolute values from about $10^{-110}$ to about $10^{110}$. Double uses more memory than Single.

- Yes/No – a data type that (usually) presents the data-entry user with a checkbox in a table, and in some other types of documents of an Access database. However, this data type has multiple representations for its values. There are really just 2 values for this data type:
  - One of the values is represented by a checked checkbox, or in text by “Yes” or by “True” (without quotation marks), or numerically by any non-zero value.
  - The other value is represented by an unchecked checkbox, or in text by “No” or by “False” (without quotation marks), or numerically by 0.

- Currency – somewhat like Double – not restricted to integer values – but its values are displayed with a currency symbol – typically, the dollar sign - and, if useful, commas.
An important question: Consider a complex data value, say, a street address like

25 South Mullens Street apt. 25B

Should this be stored as the value of one field, called, say StreetAddress, with value “25 South Mullens Street apt. 25B”? Or, should it be stored in multiple fields, say, StreetNum with value 25

Street with value “South Mullens Street”

Apartment with value “apt. 25B”

? From the design standpoint, one field is simpler than three. However, the latter approach has important advantages that we will discuss later in the semester.

Notice the View button on the Home tab. This lets you toggle among various views of your database document. For a table, it toggles between the datasheet and design views.

A field might have a value that we anticipate will be used in a high percentage of records. In this case, it’s desirable for this value to be filled in automatically in new records, so that the data-entry user need only edit this field in records in which this default value is incorrect. E.g., most students have 0 unpaid parking tickets, so the corresponding field should be filled in automatically with the value 0. At NU, with about 60% female students, we might let “Ms.” be the default Title
value. Notice that each field has a Field Property called Default Value that can be filled in with a desired default value.

It’s desirable to build error-checking into the data-entry process. For example, if you intend to enter 10000 as a Tuition value but mistakenly enter 1000000 (for $1,000,000.00), Access will accept the entry if no such error-checking is in place. However, you can use the Validation Rule among the Field Properties to state the set of values considered appropriate for the field. E.g., we might want to limit the Tuition field to a range of $0 to $75,000 (note although $75,000 is much lower than this year’s full tuition, you might want your range to be valid through a number of future years’ inflation).

Validation rules are typically based on simple comparisons in which the value of the current field is assumed, without being written, to appear on one side of the comparison operator. For example,

<table>
<thead>
<tr>
<th>To state that</th>
<th>Validation rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>the field’s value is at most $75,000</td>
<td>&lt;=75000</td>
</tr>
<tr>
<td>The field’s value is at least 0</td>
<td>&gt;=0</td>
</tr>
<tr>
<td>0&lt;=The field’s value&lt;=$75,000</td>
<td>&gt;=0 and &lt;=75000</td>
</tr>
</tbody>
</table>

You can also use “or” in a validation rule, parallel to the way we used “and” in the last example.

Note if you want the error message that appears when a validation rule is violated to say something different from the default error message, use the Validation Text to specify the desired error message.
Suppose, e.g., we have listed the records of hundreds of students. Now, one of the students makes a payment, say, for tuition, and we want to update that student’s Tuition field accordingly. We have the problem that the name and tuition fields are too far apart to appear on screen simultaneously in our table. We saw in Excel that the analogous problem could be solve by using the Freeze Panes tool to split the screen. Access offers a very different solution to this problem – namely, the use of a “form.”

An Access form lets you focus your attention on any single record, showing the data for only that record. By using such a form, you can see all the fields of the one record only, and thereby eliminate the danger of modifying the wrong record.

On the Create tab, there are several buttons that can be used to create a form. For many purposes, the Form button is easiest to use. This button can be used to create a form that shows, in a default style, all fields of one record at a time, such as the following.
Notice the “navigation buttons” toward the button of the window. These include:

- Right arrowhead: go to next record
- Left arrowhead: go to previous record
- Last record (shows right arrowhead with vertical line segment) takes you to the last record
- First record (shows left arrowhead with vertical line segment) takes you to the first record
- New (blank) record (shows a right arrowhead with a burst of light) takes you to a blank record that can be used to enter the data of a new record, using the Form view of the form.

You can therefore use a form to enter the data of a new record. Also, you can edit the data of an existing record using a form. Will such changes show up in the table that the form is based upon? Perhaps not
immediately; if, however, you close the form and the table, and re-open the table, it will show changes of data you edited using the form.

You can use a form’s Design View to change its appearance in many ways. A form’s design view might look like the following:

Notice that many fields (as above, all fields) represented by 2 graphical controls:

- Usually, as shown above, on the left, a label that is frequently (as above) used to display the name of the field.
- Usually, on the right, a textbox used to show the value of the field.

We have seen that many of the controls are, by default, grouped so that an operation such as shrinking from the right edge is applied to all members of the group even if only one of them has the mouse’ focus.
If (as will often be the case) you prefer to operate on one control at a time, do the following to break up such a default group:

- Create a (non-default) group of those controls you wish to be able to operate upon one at a time.
- Click the Remove Layout button of the Arrange tab. (The default layout has these controls in a default group; you wish to remove such a layout.)

We will see that creating a non-default group is often very useful, so that you can apply the same operation to each of the members of the group simultaneously. You can hold down the Ctrl key as you click on controls you want in a group, in order to create the group. To break up an existing group, click on a blank spot in the form.

A textbox, as we have seen, is often used to display the value of a field. However, a textbox can also be used as a “calculated control” – i.e., it can be used to display the value of a calculation, by defining its display value using a formula. For example, we might display, for each student, the total owed in tuition, room and board, and fees.

In the Design view, using the Design tab, notice the “Controls” section of the tab. You can drag-and-drop any of these controls onto the form. Notice that when you drag a textbox onto the form, it is automatically accompanied by a label, to its left. If you don’t want this label to be present, click on it and strike Del. If you keep the label, you can click in it and edit its text (indeed, you can do so with any label).

A textbox, placed on the form as described above, is initially “unbound” to any definition of a display value. As long as it’s unbound, in the Form
view it will be blank. If you want the newly introduced textbox to display the value of a field, type the name of the field into the textbox.

A formula, as in Excel, starts with an equal sign. To use the value of a field (of the same record) in a formula, enclose the name of the field in brackets. Thus, a formula for the total of tuition, room and board, and fees:

\[=\text{Tuition}+\text{RoomAndBoard}+\text{Fees}\]

You can arrange for a group of controls to align vertically at either their left or right edges, or to align horizontally at either their top or bottom edges.

- Create the group you wish to align.
- On the Arrange tab, click the Align button and choose the edge at which you want the controls aligned.

Other tools on the Arrange tab allow you to size members of a group alike (e.g., To Widest – gives all members of the group the width of the widest member).

The Format tab can be used to apply many familiar (from Word and Excel) tools to data within a control such as a textbox or label. E.g., we selected a textbox and clicked the Align Left button so that the data of the textbox is aligned left within the textbox. Notice also on this tab tools for font properties (as in Word) and for number style properties (as in Excel).

You might choose, from the Controls of the Design tab, such graphic controls as a rectangle and a line, in order to create effects like the following.
On the Design tab, the Property Sheet button gives a task pane with a large number of properties that can be edited for the currently selected control or group of controls. For example, for our rectangle, we might choose the Border Style property to change the line style of the rectangle from Solid to Dashes, and the Border Width property to change the thickness of the lines from Hairline to 2 pt, yielding the following appearance.
Notice that the Design View of a form divides the form into

- **Form Header** – analogous to a page header; typically used for data and graphics you want displayed at the top of the form, for all records.
- **Detail section** – the heart of the form, typically used to specify the display of the data of any single record.
- **Form Footer** – analogous to a page footer; typically used for data and graphics you want displayed at the bottom of the form, for all records.

If you insert an Image control (found on the Controls menu), you can then retrieve a saved image for the control to display as part of the process of placing this control on your form.
One of the most fundamental operations in the use of a database is the act of finding data, typically performed by a Find operation much that of Word, but with options that are peculiar to Access. A typical Find operation brings a dialogbox like the following:

It’s important to understand the listboxes:

- **Look in** – if you select Current Field, then only the current field (the one with the cursor) is searched. If you select Current Document, then all fields can be searched. Thus, if the cursor is in the LastName field when we want to find Georgetown, it can’t be found with the records shown above, but if we change the selection to Current Document, then Georgetown can be found in the Street field.

- **Match** – If we use Whole Field, then, in order to make a match, the Find What entry must match, completely, the value of the
field being searched. Other selections are “Any Part of Field” (allows “town” to match “Georgetown Avenue”) and At Start of Field (does not allow “town” to match “Georgetown Avenue”, but does allow “George” to match “Georgetown Avenue”)

- **Search** – if you select the menu entry “Up”, then only records above the current record are searched; if you select the menu entry “Down”, then only records below the current record are searched; if you select the menu entry “All”, then all records are searched.

Queries are often used to sort data, to filter data, and to enable you to view only a proper subset of the fields of a table. We can create a query by using, on the Create tab, either the Query Wizard or the Query Design button (I recommend the latter). If you click the Query Design button, the Show Table dialogbox, as shown below, appears.
The purpose of this dialogbox is to have you choose the datasheet(s) (table or query) you wish to base the new query upon. Choose the source datasheet(s), and for each you choose, click the Add button. After choosing the source datasheet(s) you wish to use, click the Close or the exit button for this dialogbox.

The grid at the bottom half of the window is where we design our query.

- Use the Field row to choose the fields you want in the query. As you select a field, the Table entry for that field’s column fills in automation with the name of the datasheet that the field is selected from.

- The Sort row of the design lets you choose how the data is to be sorted – by which field(s) and in which order (ascending or descending). If you wish to sort with respect to multiple fields – e.g., if you want to alphabetize the records, using LastName as the primary field of the sort and FirstName as the secondary (tiebreaking field) – the greater the priority of the sorting field, the further to the left it should be with respect to other sorting fields.

- The Show row has checkboxes. Those fields with a checked checkbox are shown in the datasheet view; those with unchecked checkboxes are not shown in the datasheet view.

- The rows labeled Criteria and Or, and the unlabeled rows below the Or row, are used to specify filtering conditions. These are written in the same style as Validation Rules. E.g., to select the records of those who owe at least $6,000 in Tuition, we can use
the filter, under Tuition, $\geq 6000$ as shown below.

- Suppose we want to select the records of students who have vehicles registered. In a query design, use either “Yes” or “True” (without quotation marks) in the VehicleRegistered field as a filter expression. Notice that this provides an example where one might reasonably choose not to show a field, since the filter guarantees that all records in this datasheet have the same value for the VehicleRegistered field.

Note that we can join simple filtering conditions by the OR operator by using different lines of the query design for the simple conditions. E.g., to select the records of students who have a vehicle registered or have unpaid parking tickets, the design shown below suffices.
This produces the following listing of records (note this shows you can copy data from a datasheet into a Word document).

<table>
<thead>
<tr>
<th>LastName</th>
<th>FirstName</th>
<th>UnpaidParkingTix</th>
<th>VehicleRegistered</th>
</tr>
</thead>
<tbody>
<tr>
<td>McKinney</td>
<td>Carol</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Flaherty</td>
<td>Martin</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>Pinchuk</td>
<td>Calvin</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Flaherty</td>
<td>Alice</td>
<td>2</td>
<td>Yes</td>
</tr>
</tbody>
</table>

By contrast, placing simple filtering conditions on the same line of the query design joins these conditions by the AND operator. E.g., the design shown below selects students who owe at least $6,000 in tuition and at least $5,000 in room and board.
Some tools for text filtering

- Text has order (a generalization of alphabetical order). E.g., to filter for students with last initials A-M, we can use the filter, under LastName, of <"N" (note we have to use “N”, not “M”, as the boundary, as, e.g., “McKinney” >”M” in alphabetical order, so “McKinney” would be excluded by using <”M” as the upper bound for the filter). Similarly, to filter for students with the last initials D-M, we can use the filter >=”D” and <”N”
This illustrates that you can use “and” or “or” to join simple filter conditions in the same field.

- The asterisk can be used as a wildcard (matching any text value). E.g., to find the students whose last initials are “D” or “F”, you can use the filter

  \[ \text{D* or F*} \]

Note it is possible that a query will produce 0 records. E.g., a filter might not be satisfied by any of the data source’s records, so that the filter produces an empty set of records.

- To select a text value of 0 characters (such a value is a “null string”), use the filter

  \[ \text{Is Null} \]
E.g., the design shown below can be used to select students with home addresses in the US, since we left the Country field null for such students.

Similarly, you can use the filter Is Not Null to find records with a non-null value in the field filtered for. E.g., this expression, used as a filter in the Country field, might be used to select the records of all “international” students.

Reports allow you to combine features of datasheets and forms. You can create a report on the Create tab, using any of several buttons in the Reports section of the tab (I recommend the Report Wizard). The Report Wizard starts the following process:

1. Select the datasheet(s) to use as sources of records for the report you are editing, and the fields you will use.
2. If your report is based on related datasheets, then you’re asked “How do you want to view your data?” This requires you to choose one of the datasheets used as a data source. Depending on your choice, you may end up implicitly grouping (see below) by one of the fields.

3. Choose whether to add grouping levels, and, if so, which field(s) to group by. Grouping is a special form of sorting the records. Each distinct value of the field(s) chosen for grouping determines a group of records – e.g., if we group by StateOrProvince, we have a group of NY records, a group of ONT records, etc. Records of a group are listed consecutively, perhaps with a group header and/or a group footer. The groups are ordered – e.g., NY group of records before ONT group of records.

4. You can choose to sort by up to 4 fields (in addition to any fields you chose to group by). Field 1 is the primary field of the sort; field 2 is secondary; etc. In a grouped report, the Summary Options button provides the easiest way to place summary statistics (for records of each group – not summaries for the entire report) in the group footer.


6. Choose a title for the report.

If, from the last step, we chose to preview the report (it can be printed from the Preview), we can exit the preview by clicking the Exit Preview
button. This takes us to the report’s Design View, which has much in common with a form’s design view.

For example, the Detail section is similar, typically used to show the value of the fields for individual records. Note each field appears once in the Detail section, but, in the Report view, all records use the field. Thus, all records appear in the Report view, one after another, in the format designed for one record in the Detail section.

Notice also the headers and footers.

- Report header – used for data to appear at the beginning of the report
- Page header – used for data to appear at the beginning of each page the report
- Group header (named for the group – e.g., “StateOrProvince header” rather than “Group header”) – used for data to appear at the beginning of each group of records. Typical uses: identify the group; column headers (here instead of in page header)
- Page footer – used for data to appear at the bottom of each page the report
- Report footer – used for data to appear at the end of the report
- Group footer (named for the group – e.g., “StateOrProvince footer” rather than “Group footer”) – used for data to appear at the end of each group of records. Typical use: statistics that summarize the records of the group.

You can use a calculated control, as in a form. Other tools familiar from Form Design include the Format tab, the Arrange tab, and graphical controls and Property Sheet of the Design tab.
In addition, you can use calculated controls for summary statistics, using the SUM, AVG, MIN, and MAX functions. To compute such a summary statistic over all records of the report, place the calculated control in the report footer; to compute such a summary statistic over all records of a group of records, place the calculated control in the group footer. E.g., to compute the total tuition owed by all students, use the formula (in a textbox in the Report Footer)

```
=SUM([Tuition])
```

You may have noticed that the Report Wizard never asked about filtering the records used. You can create a report based on a filtered set of records by basing the report on a filtered datasheet, e.g., a query with the desired filtering.

Note the placement of a calculated control has a powerful effect on what it calculates.

- If placed in the Detail section, the control’s formula should calculate using the data of the current record, for each record.
- If placed in a group footer with a formula that uses one of the summary functions, it will compute the summary for all records of the current group, for each group.
- If placed in a report footer with a formula that uses one of the summary functions, it will compute the summary for all records of the report.
- If placed in a page footer, it will usually yield an error message.

Note also that some familiar functions from Excel are available, including ROUND and IF.
It’s often desirable to have multiple tables in a database, and to establish “relations” between/among these multiple tables. E.g., in our current database, we might construct a separate table with data about the colleges of the university.

In this second table, we use the College field as a “primary key.” This is a field for which two records are not permitted to have the same value.

We added a College field in the Student Data table, and chose its data type via Lookup Wizard, and the radio button for its menu entries taken from another table – the College Data table and its College field. This is one way of creating a relationship between tables, as illustrated by the Database Tools tab’s Relationships button:
Thus, this use of the Lookup Wizard “related” the College field of College Data and the College field of Student Data.

What is the use of creating a relationship?

1. Relationships allow Access to mimic human reasoning in reaching deductive conclusions. E.g., a record of the Student Data table might have Peter LaPierre in the Hospitality College, and a record of our College Data table might show that the Hospitality College has its office in St. Vincent’s. Human intelligence deduces that Peter must go to St. Vincent’s to do something in his college’s office. Because of the relationship we have created, Access can
reach the same conclusion, e.g., if we build a query based on both tables.

2. From the above, we might react: “So what? Access could reach the same conclusions if all the data were in a single table.” But by using multiple tables, we can save a lot of work in data-entry, and a lot of memory in storing the database. E.g., suppose we had 3,000 student records of the same 16 fields (including the College field) currently in the Student Data table, plus the additional 3 more fields of the College Data table, all the in Student Data table. This would give a total 3000 * 19 = 57,000 cells for the table. If, instead, we use the current structure, expanded to 3,000 student records, we would have a total of 3000*16=48,000 cells in the Student Data table, plus 4 records * 4 fields per record = 16 cells of the College Data table, plus one relationship: a total of 48,016 cells and one relationship.

You can delete a relationship by using the Relationships display. Click on the polygonal line that illustrates the relationship to be deleted, and strike the Del key.

A stronger relationship than that created by the Lookup Wizard can be created via the Relationships display, as follows.

- Open the Relationships display (Database Tools, Relationships).
- Make sure the datasheets you wish to relate are closed.
• Make sure that on the Relationships display, the datasheets you wish to relate are in view. If necessary, click the Show Table button and choose the datasheet(s) to be shown.

• A relationship is between fields of distinct datasheets. Drag and drop one of these fields to the other. The Edit Relationships dialog box appears.

• Check the checkbox labeled Enforce Referential Integrity. This sets up an error-check: related fields, as a result of this checked checkbox, must be of the same data type.

• Click the Create button.

As a result, a relationship is created. If one of the related fields is a primary key, the relationship is “one to many,” symbolized by the 1 and ∞ symbols on the polygonal line in the Relationships display:
The 1 symbol is near the primary key; the $\infty$ symbol is adjacent to the other related field. Since records of College Data represent colleges, and records of Student Data represent students, this is interpreted as the database recognizing that there is 1 college per student, but possibly many students per college.

Note when have a report designed as shown below,
and attempt to view the report (using the Report View), we get a screen like the following:
If Access asks you for a Parameter Value, it typically means there is an error in a formula so that Access doesn’t understand the reference named (here, Total Owed). Typically, the reference is used in the notation for a field reference, but no field by that name exists. If you enter a value, Access will use the value entered, but it’s likely to produce incorrect results.

We have used the following forms of “data integration” (the use of data produced in one software package, in a document of a different software package), between Access and Word:

- Copy-and-paste of cells of a datasheet into a Word document.
• Copy-and-paste of a screen image of an Access object into a Word document.

Other forms of Access-Word data integration are discussed below.

Suppose we want to use an Access report in a Word document, with the capability of editing the Access data in the Word document, so that the version in the Word document looks like an Access report, not just like a Word table. A simple copy-and-paste of the Print Preview doesn’t work, because copy-and-paste starts with blocking data, but an attempt to block data in the Print Preview of a report fails – instead, it causes the magnification to change. What you can do is the following:

1. “Export” the report. This is a special form of what otherwise might be called a “Save As” operation. An exported document is saved in a file format that is typically not the “usual” format. In the current case, we want to export the report in Rich Text Format (RTF).
2. “Import” the desired report from its saved RTF file into your Word document.

The Export step: Using a view with the External Data tab available (note if you use an older version of Access, you may need to be in a view other than the Print Preview), on the External Data tab, click More, Word. The resulting dialogbox:
Choose disk, folder, and file name in familiar fashion. You may find the Browse button useful in specifying disk and folder. Note the default file extension, .rtf.

The “import” step: open the saved RTF file. It will open as a Word document. Copy the data you want from the RTF file into the Word document where you want it.

Another form of integration of Access data into Word documents: “mail merge” allows you to create a form or “template” document with “blanks to be filled in,” and then “fill in the blanks.” In particular, we typically use mail merge to create multiple documents that have the same general form, but are customized so that they’re all different. The customization is done by “merging” the data of a record of a
database into the template document. The merge process will produce one customized version of the template for each record in the data source.

Creating a template is a Word process, using the Mailings tab.

1. Click Start Mail Merge, Letters
2. This step seems to require that the database to be used be closed. Click Select Recipients, Use Existing List. Retrieve the database file that has the records of recipients (typically, including their addresses and other useful data), and connect the template document to the datasheet that has the desired data.
3. Edit the template document. When it is necessary to specify customized data (a “blank to be filled in”), click the Insert Merge Field button and choose the field whose value should be merged into the template.
4. The Preview Results button is a toggle. One setting allows you to view the results of merging the data of any individual record into the template. The other setting takes you back to the view of the template in which you can edit the template.
5. To obtain hardcopies of all merged documents, click Finish & Merge, Print Documents.

You can filter a table from the Home tab, in more than one way.
The Selection button offers a menu of filters that can be applied to the column that has the cursor. These selections may be partially determined by the cell that has the cursor, e.g.,

the above illustrates how the menu is affected with the cursor in a cell with the value “MD”.
Notice that either the Toggle Filter button above the datasheet or the Filter button below the datasheet can used to remove the filter currently applied; or, if a filter has been removed without being “cleared,” clicking either of these buttons reapplies the filter.

To clear a filter, click Advanced, Clear All Filters. If you do so when a filter is in use, the filter is removed.

More complex filters can be built using the Advanced menu. If, from this menu, you select Filter By Form, you reach a screen like the following, which is a design view for your filter.
In any of the fields, you can enter a filter expression (as before, in the style of a Validation Rule). E.g., to show the records of students who owe over $5,000 in tuition, under Tuition we can enter the filter \( >5000 \) and click Toggle Filter.

Notice if you do not clear a filter, it may be present the next time you design a Filter By Form.

Notice the second tab to appear for the filter design is labeled Or. This indicates that if you want to filter, using multiple conditions joined by the OR operator, then the individual conditions are specified on different tabs of the filter design. E.g, to find the set of students who owe over $5,000 in tuition or who have unpaid parking tickets, we can use
In parallel with filtering in the design of a query, if we use the *same* tab to specify multiple filtering conditions, then the conditions are joined by the AND operator. E.g., to find the set of students who owe over $5,000 in tuition and who have unpaid parking tickets, we can use

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field</th>
<th>Filter condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look For</td>
<td>Tuition</td>
<td>&gt;5000</td>
</tr>
<tr>
<td>Or</td>
<td>ParkingTix</td>
<td>&gt;0</td>
</tr>
</tbody>
</table>

When you filter a table with respect to a Yes/No field, you should regard the filter specification as having 3 possible “values” – yes/true/checked, no/false/unchecked, and unspecified. A blue fill in the checkbox of the filter design means “unspecified” or “I don’t care”. To specify that you want an unchecked checkbox, click in the checkbox twice, as a result of which, the checkbox will be unchecked and will not have a blue fill in the design, so that when the filter is applied, only records in which the checkbox is unchecked will be visible.

If you wish to join simple conditions of the same field by the AND or the OR operator, you can use “and” or “or” in the same
cell of the filter design. E.g., to find students with last initials G or L, we can use, under LastName, the filter

\[ G^* \text{ or } L^* \]