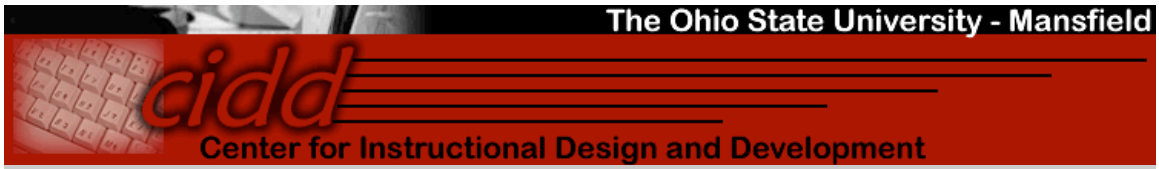


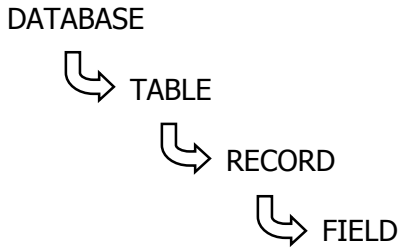
Microsoft Access 2000

Written by Mike Collura
The Ohio State University - Mansfield

The concept of a database can be thought of as follows: a database is a set of information related to a specific application, such as all of the information related to the students at a university. In the context of Microsoft Access, a database can be viewed as a large repository (like a file cabinet) in which tables, reports, queries, and other objects are stored.

Database Structure

The structure of a Microsoft Access is very hierarchical.



The primary storage unit within a database is a **table**. A database can contain multiple tables. A table organizes (and to a certain degree formats) the data to make it more useful. In our student information database, for example, we may have a table of student demographic information. A table contains a set of records.

A **record** contains data about a single unit within the table, such as a person or a tuition transaction. Using the student demographics example, a record is all the information about a single student in the table. A record is made up of a set of fields.

A **field** is a single entity within each record. It contains a single piece of information about that record. If we had Joe Student's record, his last name might be a field while another field might contain his address and still another his state and so on.

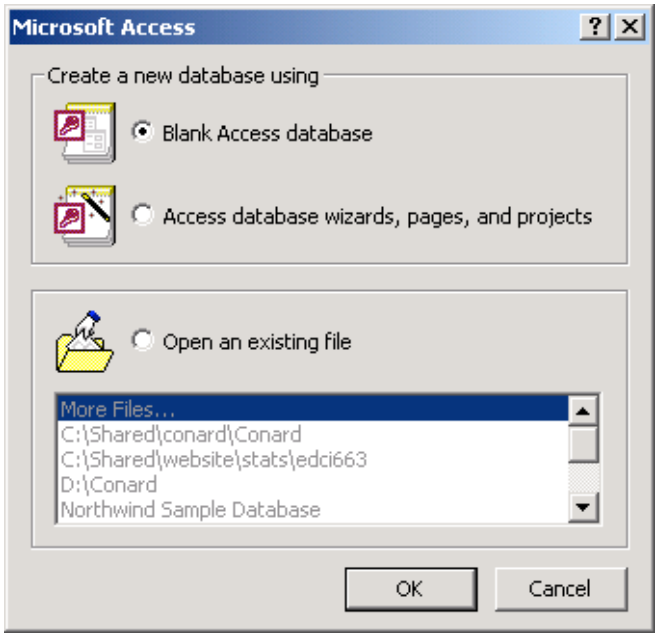
Microsoft Access is categorized as a **relational database**. This allows you to link records from two or more tables based on the contents of a field. For example, Joe Student pays tuition and fees each quarter. In addition to the student demographic information, a database might contain a table with every tuition payment made by all students. Creating relationship between the two tables will allow the user to search for tuition payments that were made only by Joe Student.

Database Objects

In addition to tables, a Microsoft Access database can also contain queries, reports, and forms. A **query** is simply a question you ask of the database in order to extract specific information. A **report** provides the ability to arrange, summarize, perform calculations on, and group data from one or more tables for printable output. **Forms** provide an alternate way of entering and accessing data in a table.

Launching Access

When you launch Access you are presented with dialog box seen below. There are 3 options to choose from at this point, 2 of which refer to creating a new database.



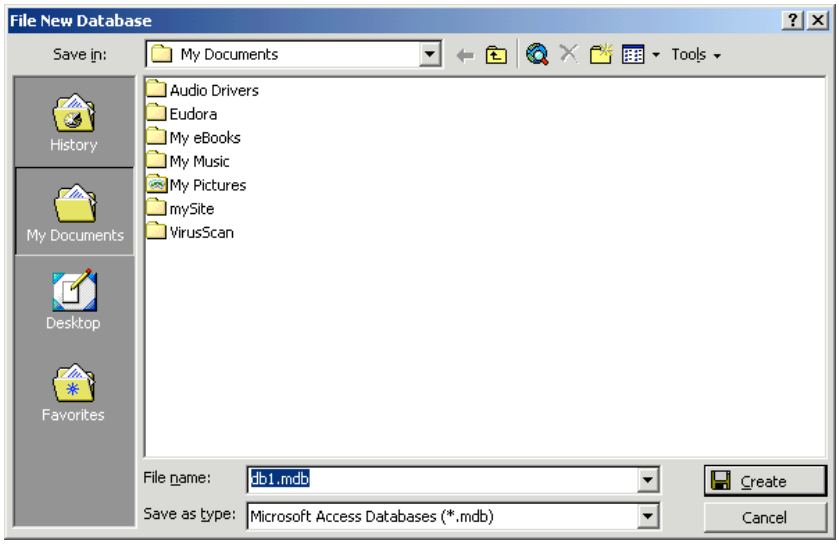
Blank Access database: This option creates a database with no objects (i.e. no tables, queries, forms or reports).

Access database wizards, pages, and projects: This option allows you to create a database from one of a number of available predefined database templates. You are taken through a wizard that asks you a series of questions and asks you to choose among various options. At the conclusion of the wizard the required objects are created for you.

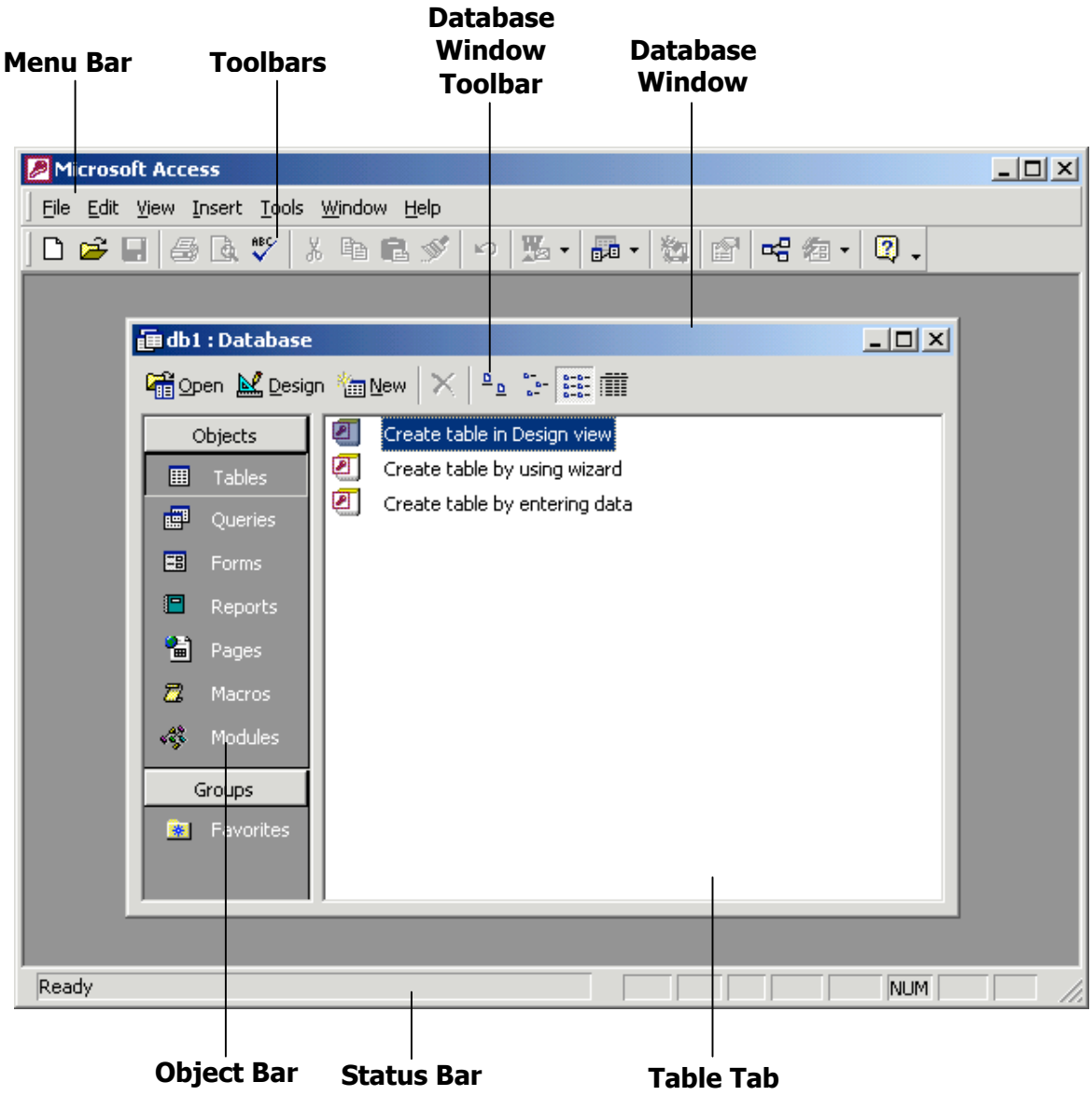
Open an existing file:
This option allows you to open a database that you have already created and saved to a disk. Listed below the option is a list of the recent files that you have worked on. You can choose a file from the list or choose More Files… to browse your available disk drives.

Creating a New Database

One feature that is very different in Access compared to other Windows programs is that you must save the database file before you can do anything to or with it. If you choose to create a new database in the above dialog box you will then be prompted with the dialog box to the right. This simply requires you to name and save the new database.



The Access Workspace



Creating Tables

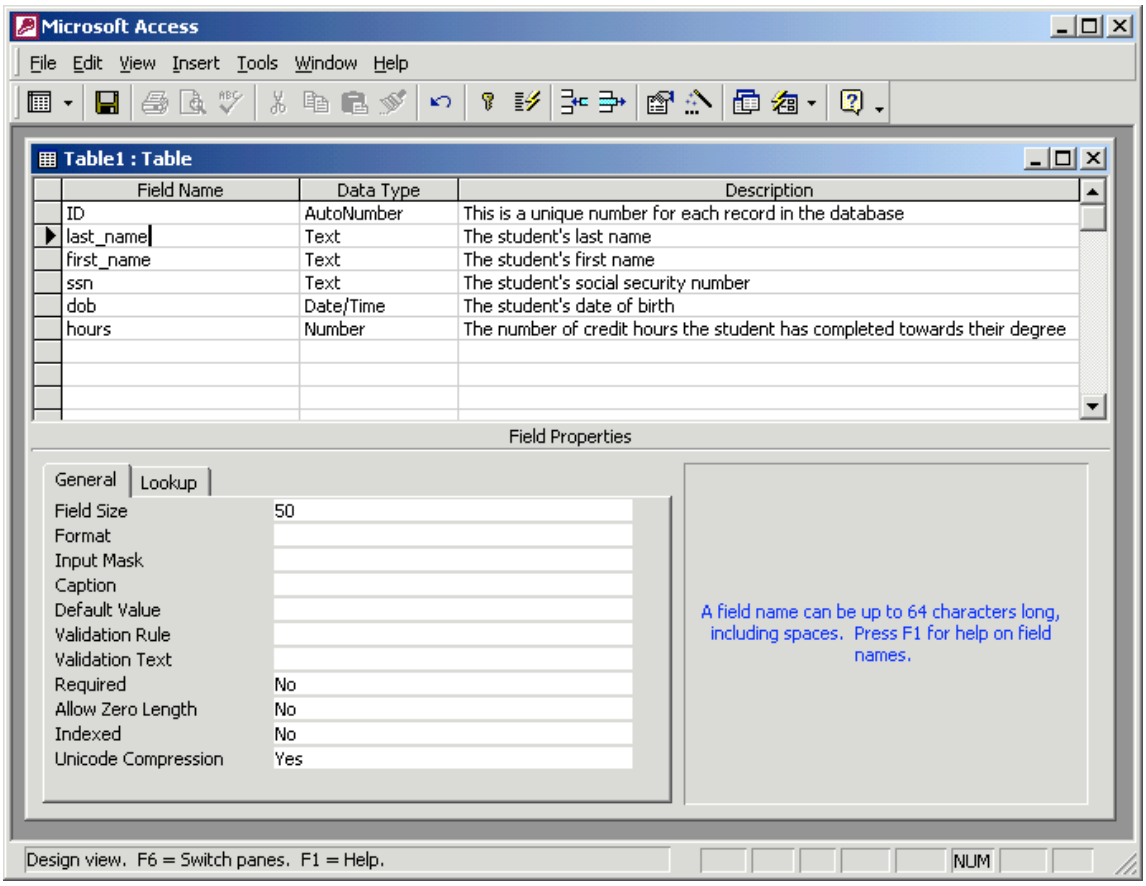
As can be seen in the table tab of the window on the previous page, there are 3 methods to creating a new table.

Create table in Design view: This method requires that you have some idea of the characteristics of the data that you want to apply to your database. You define the name, type of data, any unique characteristics that the data might have, and the field length for each field of the database. This is the most powerful of the three methods.

Create table by using wizard: This method allows you to select from a number of predefined tables, and their existing fields, based on the input you give when presented with a series of questions and options.

Create table by entering data: This method displays a blank datasheet on which you just start entering the data. Field names are simply numbered and the data type is determined for you based on the data you input.

In the picture below you will see the Design view of a table. When designing a table you must give each field a name and define it's data type. Specifying a description and the field properties are optional but encouraged.



Field Data Types

- **Text:** holds any alphanumeric character (letter, number, or special character). Is limited to 256 characters.

- **Number:** is restricted to the plus or minus sign (+ or -), numerals, and the decimal point; the decimal point must be counted as part of the field length. Use this data type anytime you want to perform calculations that involve the contents of the field.

- **Yes/No:** contains Y (yes) or N (no) and is thus always one position in length. A yes is stored as a –1, whereas a no is stored as a 0.

- **Memo:** holds large documents (64,000 characters). Use memo fields when you want to store narrative or descriptive information.

- **Date/Time:** contains eight positions and automatically has the slashes (/) in the correct locations. An empty Date field appears as / / .

- **Currency:** holds money-related data to be used in calculations. Is does a better job of rounding dollars and cents than a number field. Once you have specified the Currency data type, you usually have to define how the data is to be displayed and stored.

- **AutoNumber:** allows you to number the records as they appear in a table, query, and so forth. This type of data cannot be updated.

- **OLE object:** stores objects from other Windows applications that support object linking and embedding (OLE). When you display a record that contains an OLE field, you can view the OLE object (graphic image, graph, worksheet, and so forth) by double-clicking the field. Windows then launches the parent application for the OLE object.

- **Hyperlink:** stores hyperlinks that can be used to jump to a web site or open an Office document.

- **Lookup:** is created by the Lookup Wizard. It allows you to choose a value from another table or from a list of values using a combo box.

Special note on fields: it is not a good practice to enter field names that have spaces in them. If you must use a two word field name, separate the words with the underscore (_). For example, first name should be first_name.

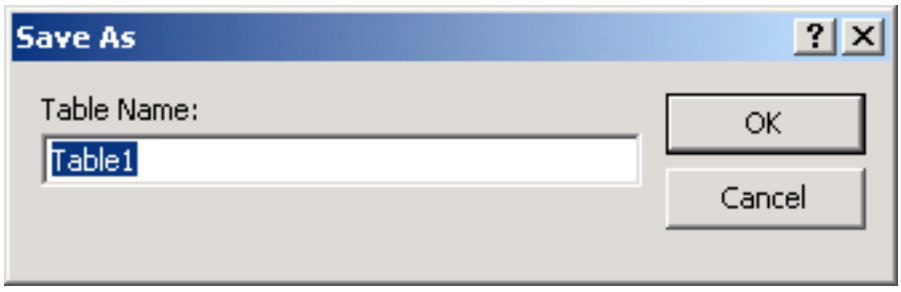
Field Properties

As mentioned, it is always a good idea to specify field properties such as the optimal field length and perhaps how the data is formatted. Field length, alignment, fill characters, color, and other features are controlled for each field via the Field Properties box. Examples of formatting and fill characters are listed below.

- **Length:** specifies the maximum number of characters the field can contain.
- **Format:** options are different depending on data type. For example, enter the greater than sign (>) to convert the text in the field to all caps or select one of the various data formats using the drop down list if the data type is Date/Time.
- **Input Mask:** formats the data entered based on predefined characteristics. For example if you use the telephone number input mask, you enter the number as 1234567890 and it displays at (123) 456-7890. This saves time for the person doing data entry.
- **Default Value:** specifies the initial value of the data for each record.
- **Validation Rule:** specifies requirements for the data being entered into the field. For example, you can specify that the data must be either autumn, winter, spring or summer. Attempting to input any other data would result in an error.
- **Required:** Ensures that data must be entered into that field for each record. For example, requiring a social security number or student ID ensures that you have something unique about that individual in the database.

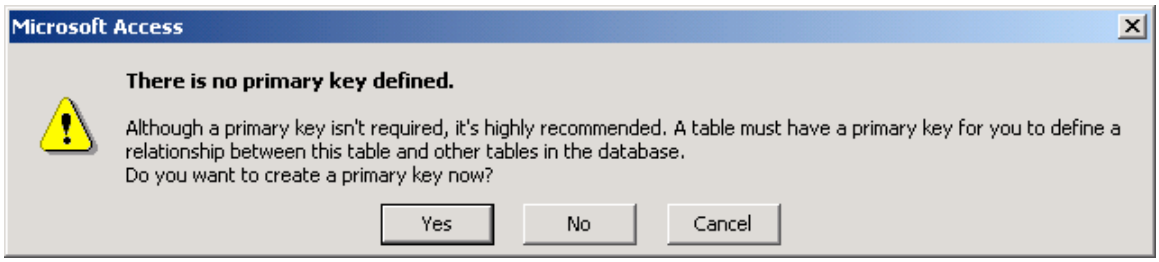
Saving your Table

Once you have completed your table design you must save the table. When you click the save button on the toolbar you will be prompted to name the table.



It's a good practice to give the table a meaningful name so it identifies its function within the database.

When save your table for the first time you will be prompted with the following dialog box warning that there is no primary key defined.



A **primary key** is a unique identifier for each record. This means that there is a field in the table that has a unique value for each record. Common primary keys are social security numbers, student IDs (like the first half of your OSU email – doe.1) or the numbers created by using the AutoNumber data type. If you are creating a database table that will eventually relate to another table, that table must have a primary key.

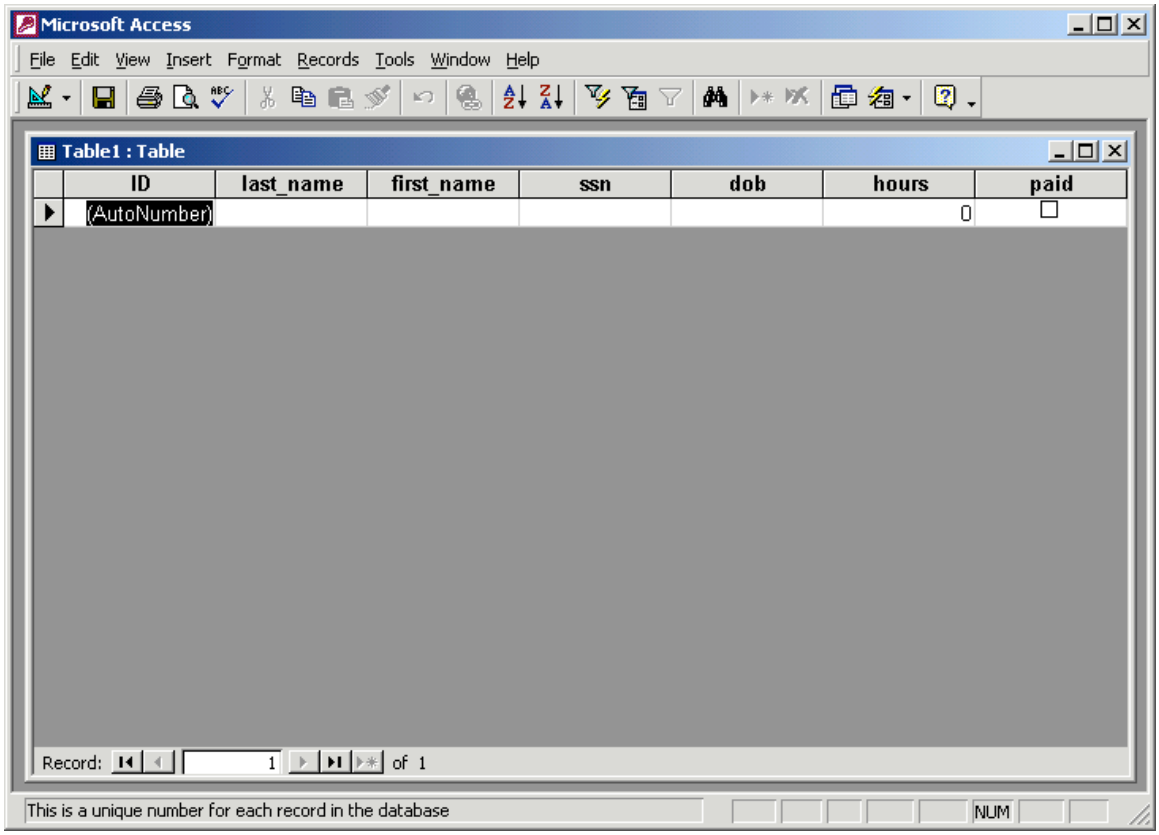
To define a primary key from the Design view right-click on the field that you would like to define as the primary key and select Primary Key from the context menu that pops up.

Entering Data

To enter data you must switch from the table Design view to the Datasheet view. On the left most edge of your toolbar you will find the view button.

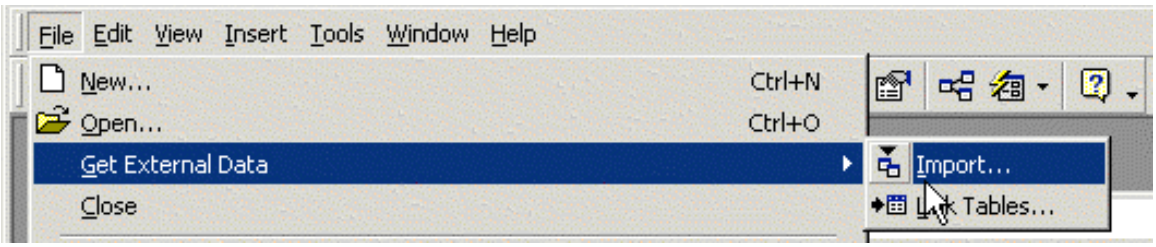
You are now ready to explore the wonderful world of data entry in Access. On the following page you will see an example of a blank table in the datasheet view. To enter data in the table you simply begin typing in the datasheet. Once you leave the field you've just entered data in, Access saves the table automatically. So, every entry is saved for you! If you have required fields you must enter data into them. If you don't, access will prompt you with an error message.

The Datasheet view of a table looks very much like an Excel spreadsheet with its rows and columns. The field names are in the gray area at the top of each column as seen below.



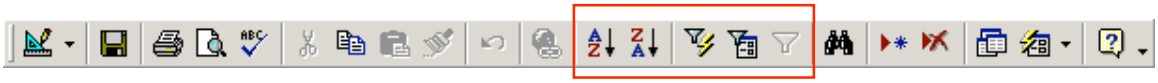
Once you've entered data you can edit it at any time by simply clicking in the desired field and changing the entry. If, however, you used an AutoNumber field, you cannot change that entry.

One feature worth noting is that you can import data from a spreadsheet, text file, or variety of other sources using the menu pictured below. I will caution you that this isn't always easy because your file must have the same number of columns, fields, etc. as the table you are trying to import it into. You can, however, have the data imported into a new table that you can format after the fact.



Sorting and Filtering your Table

Once you have tables of data the need is going to arise to sort that data and do simple searches for common elements. The toolbar below highlights the sort and filter buttons.



To sort the data, for example, alphabetically by last name simply click in one of the last name entries and click the Sort Ascending (A to Z) button. When using these buttons on numeric fields, Sort Ascending starts at the lowest number and orders the records to the highest. Conversely, Sort Descending starts at the highest number and orders the records to the lowest number.

Applying a filter allows you to perform very simple queries on your data right from the datasheet. There are two types of filters, By Selection and By Form.

- **Filter By Selection:** This type lets you filter your data based on the field you have clicked in (selected). For example, lets say you want to show only those students that are from Ohio.
    - Click in one of the fields that has OH as the entry for state
    - Click the Filter By Selection button
    - What do you see?
    - To remove the filter click the Remove Filter button

- **Filter By Form:** This type lets you choose from one of the existing field entries as the criteria for filtering. The advantage of this type of filter is that you can specify more than on search criterion. For example, what if you wanted the names of the seniors from Indiana?
    - Click the Filter By Form button
    - Notice now that when you click inside one of the fields a selection arrow appears.
    - When you click on the arrow you see a list of all the unique entries within the table for that field.
    - Click in State and select IN
    - Click in rank and select Seniors
    - Click the Apply Filter button
    - What do see now?
    - To remove the filter click the Remove Filter button

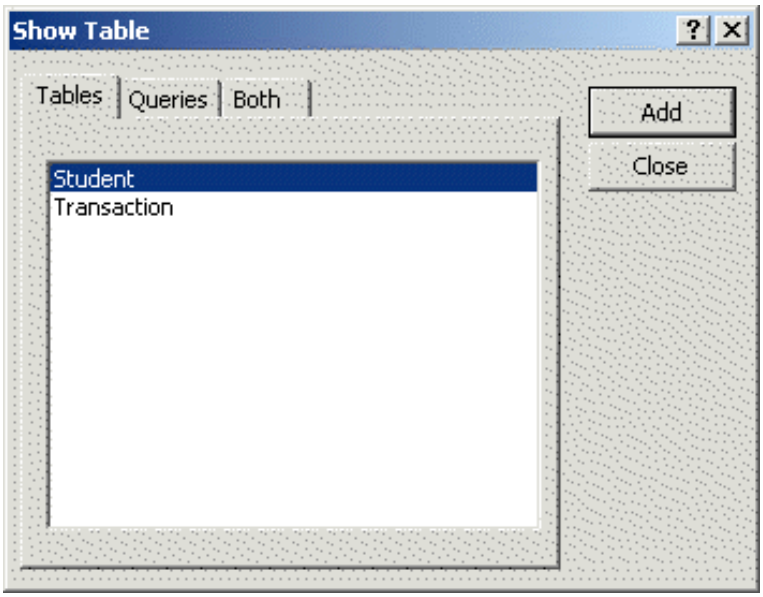
These methods are great ways to search your data but they have a few obvious limitations. First, filters cannot be saved so if you want to perform the same searches over and over you have to reapply the filter every time. Second, you can't generate reports on filtered data. Finally, they have limited search power.

Enter Queries!

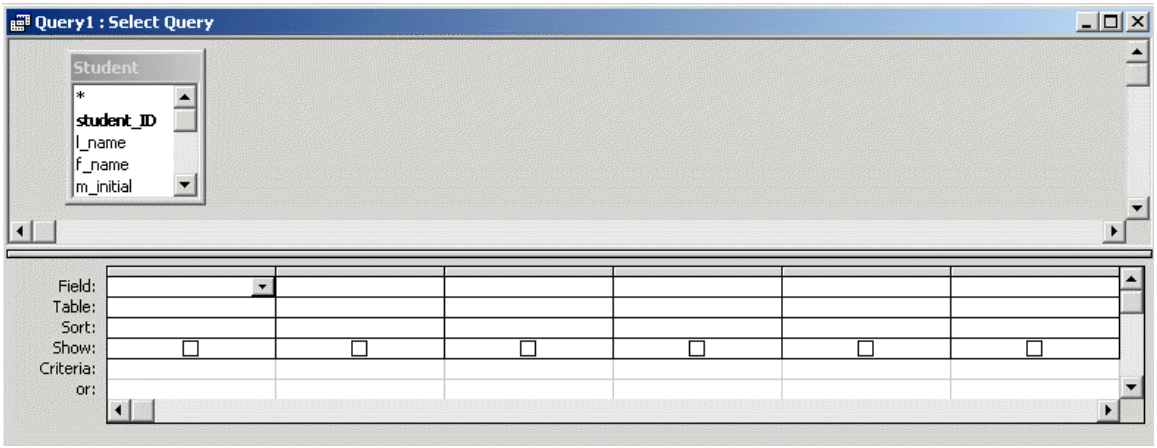
Single-Table Queries

Queries are separate database objects that can be saved and reused over and over. To create a query, from the database window click on the Query button in the object bar. There are two options for designing queries, in Design View or by using a wizard. By now these options should be clear to you. We will be using the design view.

When you double-click Create query by Design view you are first presented with the dialog box pictured below.



It is from here that you specify which table you would like to perform a query on. You can choose more than one. At this point we are only going to perform single-table queries. We will revisit multi-table queries later. So, select Student and click Add. This will take you to the query design window seen below.



The top half of this window shows the table(s) you are querying and a list of those tables' fields. The bottom is where you specify search criteria.

Pictured below is the Query Toolbar. The highlighted buttons are the 3 most important on the toolbar. The button with the exclamation point is the Run button. As the name implies, this button executes the query.

Let's begin by performing the following queries:

1. Which students are from Ohio?
   - Fields: l_name, f_name, state
   - Criteria: OH (notice what Access does to your search criteria)

2. List the current alumni and their graduating class (the year they graduated).
   - Fields: l_name, f_name, rank, grad_date
   - Criteria: Alumni

3. Which students are Freshman from Ohio?
   - Fields: l_name, f_name, rank, state
   - Criteria: freshman **AND** OH

4. Which students are from Ohio, Indiana, or Pennsylvania?
   - Fields: l_name, f_name, state
   - Criteria: OH **OR** IN **OR** PA (caution with IN)

5. Which students are Freshman from Ohio, Indiana, or Pennsylvania?
   - Fields: l_name, f_name, rank, state
   - Criteria: freshman **AND** OH **OR** IN **OR** PA

6. List the students born since 1980.
   - Fields: l_name, f_name, dob
   - Criteria: **>**12/31/79

7. List the students born in the 1970s.
   - Fields: l_name, f_name, dob
   - Criteria: **BETWEEN** 12/31/69 **AND** 1/1/80

The words AND, OR, NOT, NOR, IN, and BETWEEN are know as Boolean operators and are special words used by Access to search data.

Performing Summary Calculations in Queries

Access gives us the capabilities to do common summary calculations in a query such as summation and average. To perform such calculations we must first activate the Totals row in our query design window. We do this by clicking the Totals button on the toolbar pictured on the previous page. This is the button with the Σ (sigma) on it.

Now lets perform the following queries:

Using only the Students table:

1. How many students are from Ohio?
   - Fields: state
   - Total: **Count**
   - Criteria: **=**"OH"

Using only the Transaction table: (You can delete the table being used by clicking on it in the upper half of the query design window and hitting the delete key. To add another table use the Show Table button – the one with the yellow plus sign.)

1. What is the average amount that students have paid?
   - Fields: paid
   - Total: **Avg**

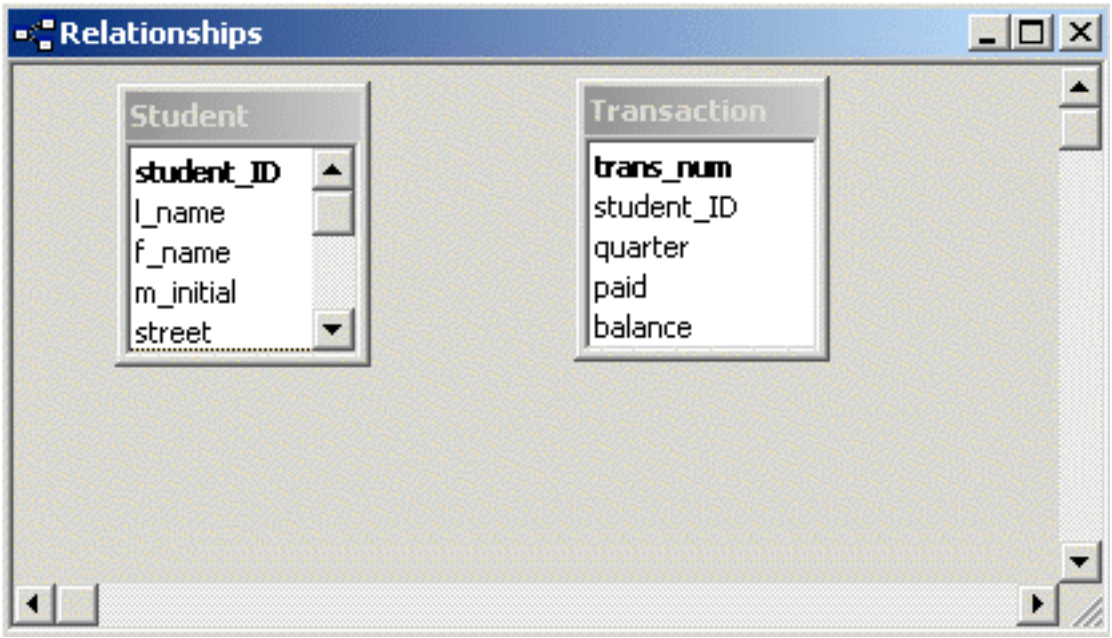
2. What is the total balance due for all the students combined?
   - Fields: balance
   - Total: **Sum**

3. What is the highest, lowest, and average balance due?
   - Fields: balance 3 times
   - Total: **Max, Min, Avg**

Relationships

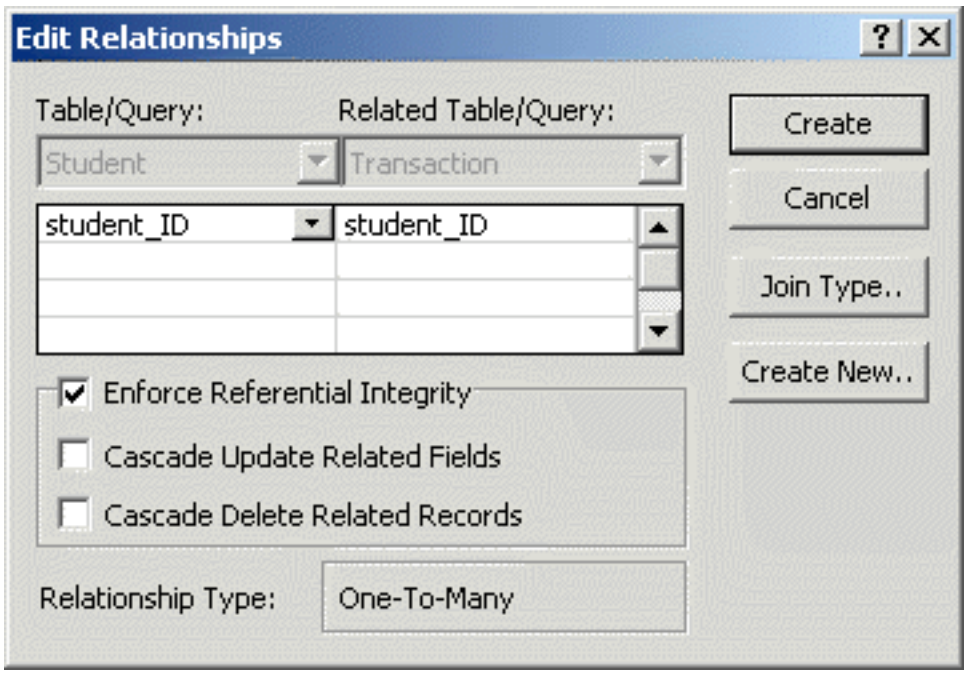
The next important concept goes all the way back to page 2 of this document. Remember we said that Access is a relational database. To build relationships between tables we have to link them in the relationships window. You can launch the relationship window using the button circled on the toolbar pictured below. You cannot have any database object open to perform this operation.

When you launch this tool you will see the same Show Table dialog box that you saw when you began a query. This time add both tables and then click close. The result is pictured in the window below.



It is from here that you link the two tables to create a relationship. Notice that the primary key in each table is in bold type. There is another type of key that needs mentioning at this point. A **foreign key** is the field in one table that is the primary key in another. In our case, the student_ID is the foreign key of the Transaction table because it is the primary key of the Student table. The Student table does not have a foreign key. It is the existence of this foreign key in the Transaction table that allows us to create the relationship. Create the relationship as follows:

- Click and hold on the student_ID field in the Student table
- Drag the pointer from the Student table into the Transaction table
- Put the pointer directly over the student_ID field (in the Transaction table)
- Let go of the mouse button
- This action will launch the dialog box pictured on the following page



Take a good look at what you see here. It shows us the field in one table that we are linking to the field in the other. In addition, it shows us the type of relationship we are creating, in our case One-to-Many. There are 3 types of relationships

- One-to-One – one record in the first table (the one on the left in the box above) is related to one (and only one) record in the other table
- One-to-Many – one record in the first table is (or can be) related to one or more records in the other table
- Many-to-Many – one or more records in the the first table is (or can be) related to one or more records in the other table

Finally this box also gives us the opportunity to **Enforce Referential Integrity**. This is a concept that creates a security net of sorts. In essence, it prevents us from deleting a record in the first table if it's related to one or more records in the other.

Think of it like this, if a student has transaction records (in the Transaction table) and we delete the student (from the Student table) we will no longer know who the transactions (the records in the Transaction table) belong to. This feature prevents us from deleting the student if their student_ID is referenced in the Transaction table.

When you click create you will notice a line linking the two tables. When you close this window you will be prompted to save and must do so to keep the relationship intact.

Creating Multi-table Queries

Now we are ready to create queries that search data from both tables simultaneously.

Lets create the following queries. When you are prompted to add the tables choose them both. Notice that the link between the tables is displayed in the query window.

1. Does Morgan Oliver owe us any money?
   - Fields: Student.l_name, Transaction.balance
   - Criteria: Oliver, >0

2. If so, how much?
   - Fields: Student.l_name, Transaction.balance
   - Total: Sum
   - Criteria: Oliver, >0

3. Who is NOT from Ohio and owes us money?
   - Fields: Student.l_name, Transaction.balance, Student.state
   - Total: Sum
   - Criteria: >0, **NOT** OH

4. Which quarter(s) has Tim Duncan already paid towards?
   - Fields: Student.l_name, Transaction.paid, Transaction.quarter
   - Total: Sum
   - Criteria: Duncan, >0

5. How many transactions do we have on file for William Anderson?
   - Fields: Student.l_name, Transaction.number
   - Total: Count
   - Criteria: Anderson

Other Database Objects

If you are interested in creating forms or reports I recommend just using wizards. The design views of these objects are very complicated. You can typically get the results you are looking for from the wizards.