Microsoft®

Access 2003

Quick Reference Sheets: Data Analysis Focus

Massachusetts Technology Corporation
Customized Training & Consulting Solutions

121 Franklin Street, Suite #100
Allston, MA 02134

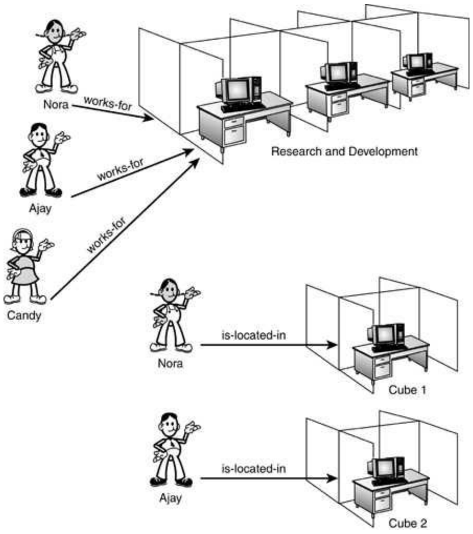
www.masstechnology.com

Database Terms and Concepts

<table>
<tr><th>Term / Concept</th><th>Description</th></tr>
<tr><td><b>Entities and Relationships</b></td><td>The very basics of what we are trying to model are <b>entities</b> and <b>relationships</b>. Entities are the things in the real world that we will store information about in the database. For example, we might choose to store information about employees and the departments they work for. In this case, an employee would be one entity and a department would be another. Relationships are the links between these entities. For example, an employee works for a department. Works-for is the relationship between the employee and department entities.

Relationships come in different degrees. They can be one-to-one, one-to-many (or many-to-one depending on the direction you are looking at it from), or many-to-many. A one-to-one relationship connects exactly two entities. If employees in this organization had a cubicle each, this would be a one-to-one relationship. The works-for relationship is usually a many-to-one relationship in this example. That is, many employees work for a single department, but each employee works for only one department.

<i>The is-located-in relationship is one-to-one. The works-for relationship is many-to-one.</i>

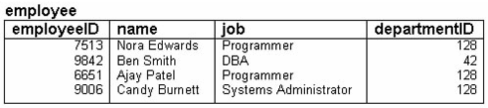


Nora
works-for
Ajay
works-for
Candy
works-for
Research and Development
Nora
is-located-in
Cube 1
Ajay
is-located-in
Cube 2

Note that the entities, the relationships, and the degree of the relationships depend on your environment and the business rules you are trying to model. For example, in some companies, employees may work for more than one department. In that case, the works-for relationship would be many-to-many. If anybody shares a cubicle or anybody has an office instead, the is-located-in relationship is not one-to-one.

When you are coming up with a database design, you must take these rules into account for the system you are modeling. No two systems will be exactly the same.</td></tr>
<tr><td><b>Relations or Tables</b></td><td>Microsoft Access is a relational database management system (RDBMS)—that is, it supports databases that consist of a set of <b>relations</b>. A relation in this sense is not your auntie, but a table of data. Note that the terms table and relation mean the same thing. In this guide, we will use the more common term table.

If you have ever used a spreadsheet, each sheet is typically a table of data. For example, an employee table may store employee IDs, names, jobs, and the department each employee works for.</td></tr>
<tr><td><b>Columns, Fields or Attributes</b></td><td>In database tables, each column, <b>field</b> or attribute describes some piece of data that each record in the table has. The terms column, field and attribute are used fairly interchangeably, but a column or field is really part of a table, whereas an attribute relates to the real-world entity that the table is modeling.</td></tr>
<tr><td><b>Rows, Records, Tuples</b></td><td>In the example of the employee table, each row represents a single employee record. You may hear these called rows, <b>records</b>, or tuples. Each row in the table consists of a value for each column/field in the table.

employee<table><tr><th>employeeID</th><th>name</th><th>job</th><th>departmentID</th></tr><tr><td>7513</td><td>Nora Edwards</td><td>Programmer</td><td>128</td></tr><tr><td>9842</td><td>Ben Smith</td><td>DBA</td><td>42</td></tr><tr><td>6651</td><td>Ajay Patel</td><td>Programmer</td><td>128</td></tr><tr><td>9006</td><td>Candy Burnett</td><td>Systems Administrator</td><td>128</td></tr></table></td></tr>
<tr><td><b>Keys</b></td><td>A <b>superkey</b> is a column (or set of columns) that can be used to identify a row in a table. A key is a minimal superkey. For example, in the employee table we could use the employeeID and the name together to identify any row in the table. We could also use the set of all the columns (employeeID, name, job, departmentID). These are both superkeys.

However, we don't need all those columns to identify a row. We need only (for example) the employeeID. This is a <b>minimal superkey</b>—that is, a minimized set of columns that can be used to identify a single row. So, employeeID is a key.

In the employee table, we could identify an employee by name or by employeeID. These are both keys. We call these candidate keys because they are candidates from which we will choose the primary key. <b>The primary key</b> is the column or set of columns that we will use to identify a single row from within a table. In this case we will make employeeID the primary key. This will make a better key than name because it is common to have two people with the same name.

<b>Foreign keys</b> represent the links between tables. For example, if you look back at the employee table above, you can see that the departmentID column holds a department number. This is a foreign key: The full set of information about each department will be held in a separate table, with the departmentID as the primary key in that table.</td></tr>
<tr><td><b>Functional Dependencies</b></td><td>The term <b>functional dependency</b> comes up less often than the ones previously mentioned, but we will need to understand it to understand the normalization process that’s important to designing effective and efficient databases.

If there is a functional dependency between column A and column B in a given table, which may be written A —> B, then the value of column A determines the value of column B. For example, in the employee table, the employeeID functionally determines the name (and all the other attributes in this particular example).</td></tr>
<tr><td><b>Schemas</b></td><td>The term <b>schema</b> or database schema simply means the structure or design of the database—that is, the form of the database without any data in it. If you like, the schema is a blueprint for the data in the database.

We can describe the schema for a single table in the following way: <code>employee(employeeID, name, job, departmentID)</code>

A traditional convention is to use a solid underline for the attributes that represent the primary key and a broken underline for any attributes that represent foreign keys. Primary keys that are also foreign keys have both a solid and a broken underline.</td></tr>
</table>

What is Microsoft Access?

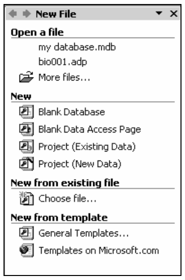
A database is a collection of information organized as to make it easy to view it, search it, retrieve the right detail, and collect the necessary facts in the easiest, most timely, and most effortless manner possible

Organization means method, assumes discipline, and also anticipates efficient manner in using that information. Unless you are creating small applications for personal use, you will usually need to share your data with other people (users, database developers, etc) or other machines. To make your job easier, Microsoft Access provides database information and the tools you need to use your database in one package. To be organized, you will divide your database into different related parts. The method of management you'll use makes Microsoft Access a Database Management System (DBMS). When you create a database in Microsoft Access, you create a file that includes different parts of your database. These are referred to as tables, queries, forms, reports, etc.

Since you define these parts, you also organize them in a manner that helps some parts of your database to supply specific information to others. In one part, you would cover one category of data, such as people's personal information (name, date of birth, salary, hobbies, etc). In another you would cover what they buy in a store or other related transactions. Microsoft Access is a relational database used on desktop computers to manage information on different levels for different purposes. It can be used for personal information management, in a small business to organize and manage all data, or in an enterprise to communicate with servers.

Creating a Database

1. To launch Microsoft Access, click Start > Programs > Microsoft Access.
2. From the Task Pane, click Blank Database.

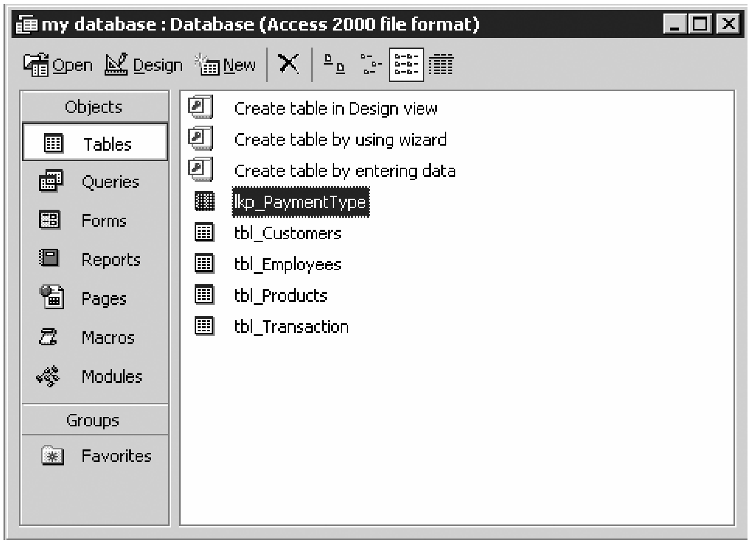


3. Navigate to the location where you'd like to save your database, and enter a database name.

   *Unlike most desktop applications, you create the database file **before** you begin adding content.*

4. Click OK.

The Database Window



Tables are the core of the database. Data is stored in rows (records) and columns (fields).

Queries allow users to view, calculate, filter, change, sort and examine the data stored in tables.

Forms are custom screens that provide an intuitive way to enter and update data in a table or query

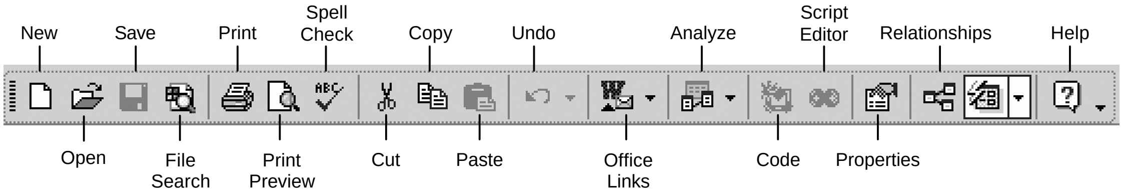
Reports present data from a table or query in a printed format. They can be customized and saved for later use.

Pages are forms saved as web pages so that the database can be accessed via the Internet

Macros automate common tasks and can be invoked by clicking a button, pressing a shortcut key or through a Visual Basic event.

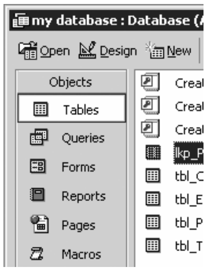
Modules are groups of procedures that are written in Visual Basic and used to automate tasks and create applications through Access.

The Standard Toolbar



Creating a Table

1. To create a new table, ensure the *Tables* item is selected in the Database window.

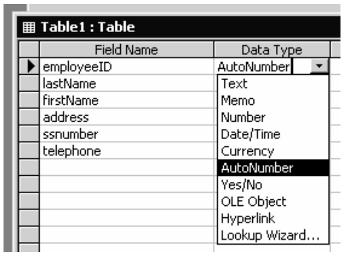


2. Click the New button in the Database window.

3. Choose Design View.

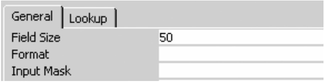
4. Populate the table with your field names, with the following recommendations for ease of use and portability:

   - Create one field that always has a unique value, and give it a relevant name. Examples include *ID, ProductID, InvoiceID, EmployeeID.* Make this field a **primary key** by clicking on the 🔑 button on the toolbar. This will ensure at least one field contains unique data for each record.
   - For portability to other database programs and external applications, avoid spaces and special characters (e.g., parenthesis, dashes, asterisks, etc.) in field names. Names such as *LastName, FirstName, SSNumber* are ideal.
   - Choose data types and lengths appropriate to the field content.



Field Name	Data Type
employeeID	AutoNumber
lastName	
firstName	
address	
ssnumber	
telephone	

Figure: Data Types



Field Size 50
Format
Input Mask

Figure: Text Field Size

The most common field types are as follows:

Data Type	Description
Autonumber	Each time a new record is created, a sequential **number** is automatically populated into the table. This is an ideal field for primary key fields such as invoice number, product IDs, etc.
Text (Default)	A field for small amounts of text, such as names, addresses, zip codes, telephone numbers. The maximum available size for a text field is 255 characters. If you will never use this field as a source for mathematical calculations, then it's a good choice.
Memo	Stores long text entries – up to 64,000 characters long.
Number	Stores numbers that can be used in calculations.
Currency	Similar to the Number data type. Stores numbers and symbols that represent money.
Date/Time	Stores dates, times or both.

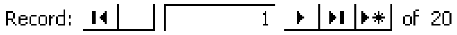
5. Click the 💾 (Save) button. We recommend you name the table with a prefix of *tbl_* and remove spaces or special characters such as parenthesis, periods, or asterisks. Examples include *tbl_Employee, tbl_Product, tbl_Customer.*

6. Click the ▭ Datasheet View button to switch from the design view to the Datasheet view.

Navigating a Datasheet (Tables and Queries)

The Navigation Bar

The navigation bar is found near the bottom of most table and query windows. Database information can be directly added and edited from tables and certain queries and forms.



Function	Action
Add a New Record	Click the **New Record** button on the navigation bar.
Select a Record	Click the **Record selector** to the left of the record.
Save a Record	Click on a record other than the current one. The record will be automatically saved.
Delete a Record	Select the record and click the **Delete Record** button on the toolbar. *IMPORTANT: There is only one undo level in Microsoft Access, so be careful when deleting records as you may not be able to undo this action.*
Spell Check	Click the **Spelling button** on the toolbar.
Print Preview:	Click the **Print Preview** button on the toolbar.
Find Information in a Table	Place the cursor *in the field* that contains the value you want to search for, then click the **Find button** on the toolbar or press **<Ctrl>** + **<F>**. Type the value to search for in the **Find What box** and click **Find Next**. Click **Find Next** until you've found what you're looking for.
Replace Information in a Table	Place the cursor *in the field* that contains the value you want to replace, then select **Edit** ✍ **Replace** from the menu. Type the value you want to search for in the **Find What box** and the new value in the **Replace With** box. Click **Find Next** until you've found what you're looking for, and then click **Replace** or **Replace All** to replace every instance of the value. *IMPORTANT: There is only one undo level in Microsoft Access, so be careful when updating records as you may not be able to undo this action.*
Sort Information	Place the cursor in the field to sort by and click either the **Sort Ascending** or the **Sort Descending** buttons on the toolbar.
Filter By Selection	Place the cursor in the record and field that matches the selection criterion and click the **Filter By Selection** button on the toolbar.
Filter By Form	Click the **Filter By Form** button on the toolbar. Click the **down arrow** next to the field that contains the selection criterion and select the value you want the filtered records to match. You can use more than one field to filter data.
Remove a Filter	Click the **Remove Filter** button on the toolbar.

Datasheet Shortcuts

Function	Shortcut
Print Current View	<Ctrl> + <P>
Undo	<Ctrl> + <Z>
New Record	<Ctrl> + < + >
Delete Record	<Ctrl> + < - >
Save Record	<Shift> + <Enter>
Cancel Changes	<Esc>
Insert Date	<Ctrl> + < ; > (semi-colon)
Insert Time	<Shift> + <Ctrl> + < : >
Insert Value from Same Field in Previous Record	<Ctrl> + < ‘ > (apostrophe)
Next Field	<Tab>
Previous Field	<Shift> + <Tab>
First Record	<Ctrl> + < arrow up >
Last Record	<Ctrl> + < arrow up >

Establishing Table Relationships

When you store data, you have to create a table for each subject or type of information you are tracking. To bring the data from multiple tables together in a query, form, or report, you must define relationships between the tables. Relationships match data in key fields—usually a field with the same name in both tables.

In most cases, you will define a relationship between the primary key (which provides a unique identifier for each record from one table) and a foreign key in the other table. In the figure below a vendor (the primary key) is associated with the products that it is connected to (the foreign key) through the relationship created between the Suppliers table and the Products table using the SupplierID fields in both tables.

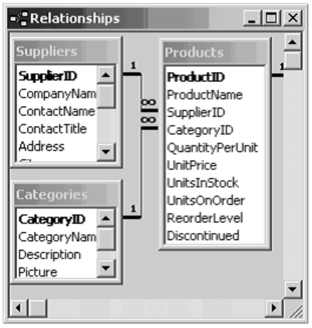
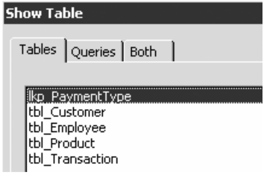


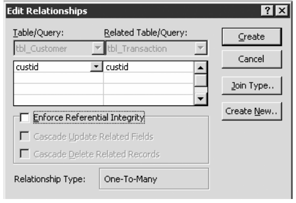
Figure: The Relationships window

To begin setting up relationships, first close any tables you have open. You can't create or modify relationships between open tables. Then follow these steps:

1. Press <F11> to switch to the Database window.
2. Click **Relationships** on the toolbar.
3. If you haven't defined any relationships in your database thus far, the **Show Table** dialog box is automatically displayed.



4. If you need to add the tables you want to relate and the **Show Table** dialog box isn't displayed, click **Show Table** on the toolbar.
5. Double-click the names of the tables you want to relate, and then close the **Show Table** dialog box by clicking the Close button.
6. Drag the field that you want to relate from one table to the related field in the other table.
7. To drag multiple fields, press the CTRL key, click each field, and then drag them.
8. In most cases, you drag the primary key field (which is displayed in bold text) from one table to a similar field (often with the same name) called the foreign key in the other table.
9. The **Edit Relationships** dialog box is displayed. Check the field names displayed in the two columns to ensure they are correct. You can change them if necessary.



Table/Query: tbl_Customer
Related Table/Query: tbl_Transaction
custid custid
□ Enforce Referential Integrity
□ Cascade Update Related Fields
□ Cascade Delete Related Records
Relationship Type: One-To-Many

10. **(Optional)** Check the Enforce Referential Integrity box to avoid creating “orphan” data in foreign tables. If you want to change the “join type” between the tables, click the **Join Type button**, select the type of join and click **OK**.
11. Click the **Create** button to create the relationship.
12. Repeat steps 4 through 7 for each pair of tables you want to relate, then close the Relationships window and save your changes.

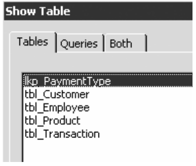
Tip: To start seeing how relationships are set up, look at the Northwind.mdb sample database that is installed with Microsoft Office. On the **Help** menu, point to **Sample Databases**, and then click **Northwind Sample Database**. Once the Relationships window is open you can print the relationships. Study them to determine why fields were connected to one another.

Creating a Query

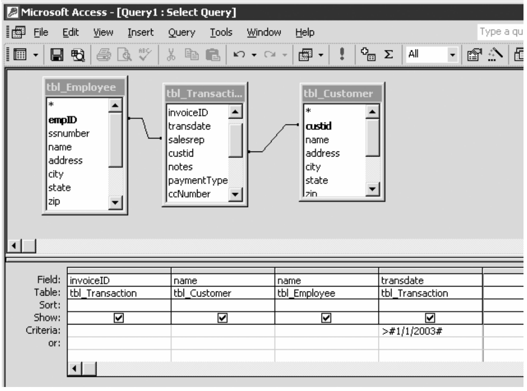
A query is a request you present to the database, and the database displays its response to you or performs an action. A SELECT query contains logic in the form *"show me these items from these sources that meet these criteria*." You can create a query either through the query wizard (simplest), or the Query Design View (most accurate).

To create a Select Query

1. From the Database Window, click Queries and double-click **Create query in Design view**.
2. Select the table you wish to query and click **Add**.



3. Repeat for all the tables you want to add to the query.
4. Click **Close** when you're finished.
5. Find the field you want to query (use the scroll bars) and drag the field down to the QBE grid.



Field:	invoiceID	name	name	transdate
Table:	tbl_Transaction	tbl_Customer	tbl_Employee	tbl_Transaction
Sort:				
Show:	☑	☑	☑	☑
Criteria:				>#1/1/2003#
or:				

6. Repeat until you've added all the fields you want to include in the query.
7. If you want to filter records, enter the criteria in the Criteria row using the guidelines in this table.

Criteria Example	Type	Result / Description
"Connecticut"	Text	Dislays records where the field equals **Connecticut**.
Between #1/1/00# and #12/31/00# *…or…* >= #1/1/00# and #12/31/00#	Numeric, Text	Displays records where the date is between 1/1/00 and 12/31/00.
Not "Boston" and <> ""	Text	Displays records where the field is not blank, and does not equal "Boston"
Like "be*"	Text	Displays records where the field starts with **be**, such as **be**aches, **be**fore, **be**ar.
Between A* and C*	Text	Displays records where the field starts with **A B** or **C.**
> 500	Numeric	Displays records whose field value is greater than 500
Between 500 and 600 *…or…* >= 500 and <=600	Numeric	Displays records where the field value is between 500 and 600
IS NULL	Numeric, Text	Displays records where the field is blank.
IS NOT NULL	Numeric, Text	Displays records where the field is not blank.

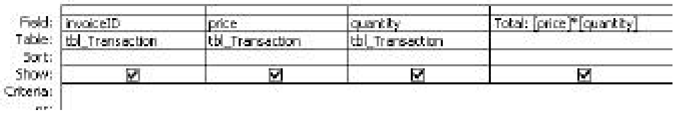
8. To run the query, click the ! **Run** button.
9. Once you've finished, close the query window, click **Yes** to save the query, enter a query name, and click **OK**.

Creating a Calculated Field in a Query

You can use field values in a table to create a new column with the calculated result. For example, to calculate the total quantity multiplied by the product price, into a column named *Total*, you would enter in the design grid:

```
Total:[price]*[quantity]
```

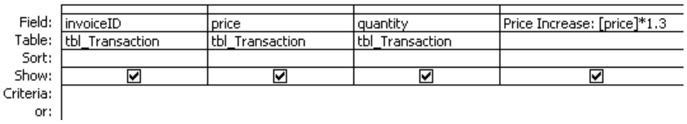
Note that the colon is used to separate the column name from the expression and field names are enclosed in square brackets ([ ]); numbers are not.



Field:	invoiceID	price	quantity	Total: [price]*[quantity]
Table:	tbl_Transaction	tbl_Transaction	tbl_Transaction	
Sort:				
Show:	☑	☑	☑	☑
Criteria:				

Expressions can also include a single field that has a value to be adjusted, such as calculating values for a price increase. To calculate a 30 percent increase in the amount of a product price and display the results in a column you want to name *Price Increase*, you would enter in the design grid:

```
Price Increase:[price]*1.3
```



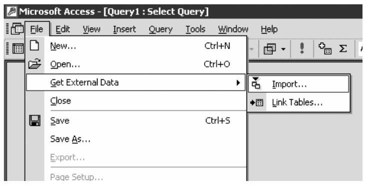
Field:	invoiceID	price	quantity	Price Increase: [price]*1.3
Table:	tbl_Transaction	tbl_Transaction	tbl_Transaction	
Sort:				
Show:	☑	☑	☑	☑
Criteria:				
or:				

Importing a Spreadsheet

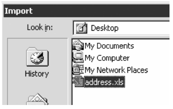
You can import multiple file types into an Access database, and create a new table out of spreadsheets, delimited text files, other database files and more.

Create a New Table from a Spreadsheet:

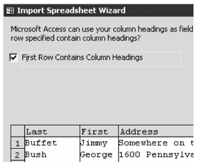
1. **Begin the import process.** From your File menu, choose **Get External Data**. Then choose **Import**.



2. **Select the file to import.** Select the Microsoft Excel spreadsheet that will serve as the source of data for your new database. Use the dialog box presented to navigate through your file system to the correct spreadsheet and file type.



3. **Select the appropriate worksheet.** Select the Worksheet or Named Range that you wish to import. Most likely, you'll want to accept the default option of Sheet1, unless you have a more complex Excel file. Be aware that the first row of your data may contain field headings. If so, be sure to check the *First Row Contains Column Headings* box.



Import Spreadsheet Wizard

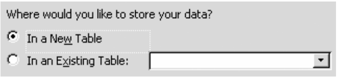
Microsoft Access can use your column headings as field
row specified contain column headings?

☑ First Row Contains Column Headings

	Last	First	Address
1	Buffet	Jimmy	Somewhere on t
2	Bush	George	1600 Pennsylva

Notice that a preview of the data is provided in the bottom portion of the window. Take this opportunity to give the data a glance to ensure it's the correct data source. Click **Next** to continue.

4. **Choose a destination type.** If you're trying to create a new table, you'll want to check the *In a New Table* radio box and click **Next** to continue. If you were importing data into an existing table, you could specify the destination table here. However, column headings in the Excel file must match with field names in the existing Access table.



Where would you like to store your data?

● In a New Table

○ In an Existing Table:

5. Click **Next** to continue and then click **Finish**.