# TABLE OF CONTENTS

- **Introduction** ............................................................................................................................ 1
- **Relationships** .......................................................................................................................... 1
  - Using Related Tables .................................................................................................................. 1
  - Creating a Relationship ............................................................................................................. 2
- **Using Operators in Queries** ....................................................................................................... 6
  - Using Comparison Operators .................................................................................................... 6
  - Using an AND Condition ............................................................................................................ 7
  - Using an OR Condition ............................................................................................................... 9
  - Using the BETWEEN AND Operator ....................................................................................... 11
  - Using the Expression Builder .................................................................................................. 12
  - Using the Wildcard Character ................................................................................................. 13
- **Designing Advanced Queries** .................................................................................................. 14
  - Creating a Calculated Field ...................................................................................................... 14
  - Creating a Function Query ....................................................................................................... 15
  - Creating a Parameter Query ................................................................................................... 16
  - Creating a Concatenation in a Query ...................................................................................... 17
  - Filtering a Query ....................................................................................................................... 18
- **Class Exercises** ...................................................................................................................... 20
- **How To Learn More** ................................................................................................................ 21
  - For Intermediate Access Users .............................................................................................. 21
  - For Access Users New to Office 2007 .................................................................................... 21
  - Printed Material ....................................................................................................................... 22
- **Training and Support** ............................................................................................................. 22
  - IT Training ................................................................................................................................. 22
  - Troubleshooting and Support ................................................................................................. 22
INTRODUCTION

This training guide will acquaint you with the intermediate functions of Microsoft Access 2007. This guide provides an intermediate understanding of database functions and capabilities, and the tools necessary to begin your own exploration with Access 2007.

RELATIONSHIPS

Using Related Tables

Tables can be joined, or related to one another. This means that information from fields within different tables can be used. Joining tables is a useful way to:

- Avoid duplicate information in various tables.
- Create reports, forms, and queries from the related data tables and save them in the database file.
- Create smaller, more efficient tables that can be related when you need access to the data.

When you relate tables, the table from which you select the first join field is the **Primary** table, and the table to which you drag the join field to, is the **Related** table. The tables must have some **common fields** that contain the same type of data. One of the fields in the primary table must be the **Primary Key**. The primary key is used to prevent duplicate entries. The common fields in both tables must have the same or equivalent data types and; if they are **Number** fields, they must have the same field size.

For example, you can create a table consisting of customer names and addresses. You can also include a unique identification number for each customer, which would be the **primary key** in the table (see Figure 1). You can create this number or allow Access to create it for you.

Figure 1 - Table of Names and Addresses

<table>
<thead>
<tr>
<th>ID #</th>
<th>Names</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jones</td>
<td>5352 Oak Tree Drive</td>
<td>Edmonds</td>
<td>WA</td>
<td>98026</td>
</tr>
<tr>
<td>2</td>
<td>White</td>
<td>563 S. Monroe Blvd</td>
<td>Fort Wayne</td>
<td>IN</td>
<td>46825</td>
</tr>
<tr>
<td>3</td>
<td>Smith</td>
<td>21458 Clark Road</td>
<td>Valencia</td>
<td>CA</td>
<td>91355</td>
</tr>
<tr>
<td>4</td>
<td>Lee</td>
<td>1622 16th Street West</td>
<td>Washington</td>
<td>DC</td>
<td>20520</td>
</tr>
<tr>
<td>5</td>
<td>Fulbright</td>
<td>975 Walker Court</td>
<td>Golden Valley</td>
<td>MN</td>
<td>55422</td>
</tr>
</tbody>
</table>

*Primary key*
You could then create a separate table consisting only of orders placed by customers. This table would also contain the field for the unique customer identification number, but not the customers’ names and addresses (see Figure 2).

Figure 2 - Table of ID Numbers and Products

<table>
<thead>
<tr>
<th>ID #</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PC Label Printer</td>
</tr>
<tr>
<td>2</td>
<td>12- Sheet Crosscut Shredder</td>
</tr>
<tr>
<td>3</td>
<td>Vertical Wood File Cabinet</td>
</tr>
<tr>
<td>4</td>
<td>Black Leather Executive Chair</td>
</tr>
<tr>
<td>5</td>
<td>Fire-Safe Security File Cabinet</td>
</tr>
</tbody>
</table>

By relating the two tables through the common customer identification number field, the customers’ names, addresses, and telephone numbers do not have to be entered for every order.

Access includes **two basic types** of relationships:

- **One-to-many**: one record from the primary table matches many records from the related table; for example, one customer record matches many order records.

- **One-to-one**: one record from the primary table matches one record from the related table. Access determines the relationship type automatically when you create a relationship.

**Creating a Relationship**

You create relationships between tables or queries in the **Relationships** window. The **Relationships** window displays a graphic representation of the relationships in the database.

1. Open database *1Relationship.accdb*
2. Select the **DatabaseTools** tab.
3. Click **Relationships** in the **Show/Hide** group (see Figure 3).

Figure 3 - Relationships Command
4. On the **Design** tab, in the **Relationships** group, click the **Show Table** button (see Figure 4).

![Figure 4 - Show Table Button](image)

5. From the **Tables** tab, select the tables you’d like to relate from the **Show Table** dialog box, **tblBookOrderDetails** (see Figure 5).

![Figure 5 - Show Table Dialog Box](image)

6. Click the **Add** button.
7. Click **Close**. Results are shown in Figure 5a.

![Figure 5a - Show Tables](image)
8. Drag the fields to be included in the relationship: **OrderID** field from **tblBookOrders5**.
9. Drop the field on **OrderID** in the **tblBookOrderDetails** table.
10. The **Edit Relationships** window appears (see Figure 6).

Figure 6 - Edit Relationships Window

![Edit Relationships Window](image)

When working in the **Relationships** window, you can reposition the field lists to view the relationships more easily. In addition, the field name representing the primary key appears in **bold in the tables**. All tables must be closed before you can create relationships.

**Note:** You can select multiple tables in the **Show Table** dialog box by holding the **Ctrl** key and clicking on each table. Clicking the **Add** button adds all the selected tables to the **Relationships** window.

**Setting Referential Integrity**

When you create a relationship between two tables, you can set **referential integrity**. Referential integrity is a built-in set of rules Access uses to make sure that the relationship is valid. Referential integrity can also prevent accidental deletion or editing of data.

In order to use referential integrity, the following conditions must be true:

- The related field in the primary table is the primary key.
- The related fields in both tables have the same data type.
- Both tables belong to the same database.

When you set referential integrity, you must observe the following three rules:

- You cannot enter data in the join field in the related table that does not have a match in the join field in the primary table.
- You cannot delete records from the primary table if there are matching records in the related table.
• You cannot edit primary key values in the primary table if related records exist.

Figure 7 - Enforce Referential Integrity Selected

However, if you want to perform the changes listed above and still maintain referential integrity, you can select the **Cascade Update Related Fields** and **Cascade Delete Related Records** options in the **Edit Relationships** dialog box (refer to Figure 7). When either or both of these options are selected, Access makes the necessary changes to the related tables automatically to maintain referential integrity.

**Note:** It is recommended that these two options be used only after careful consideration since the changes cannot be undone.

11. Make your selection.
12. Click the **Create** button to close the dialog box.

When the referential integrity option is activated, Access displays symbols above the join line in the **Relationships** window to indicate the type of relationship: one-to-one or one-to-many. The number 1 above the join line next to a field list indicates "one", while the mathematical symbol for **infinity** (which resembles a horizontal 8) indicates "many" (see Figure 8).

Figure 8 - Relationship Between Two Tables
Notes: Double-clicking the middle segment of any join line opens the **Edit Relationships** window with the selected join displayed. Double-clicking the beginning or ending segment of any join line opens the **Edit Relationships** window, but with no join selected. You can then use the Table or Query list to select the desired join.

You can also open the **Edit Relationships** dialog box by right-clicking the middle segment of the join line and selecting the **Edit Relationship** command.

**Deleting a Join Line**

Deleting a join line removes the relationship between two tables. If you no longer need to relate the tables or you want to create a different relationship, you must delete the join line.

You must select a join line before you can delete it:

1. Right-click on the middle segment of a join line to select it.
2. Click **Delete**.
3. Click **Yes** to permanently remove the selected relationship from the database.

Note: You can also delete a join line by clicking its middle segment and pressing the **Delete** key on your keyboard.

**USING OPERATORS IN QUERIES**

**Using Comparison Operators**

To select specific records, you can enter criteria in the **Criteria** row of the query design grid. The simplest criterion requires that records match a single value to be included in the **RecordSet**.

![Figure 1 - Comparison Operators](image)

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td>less than</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>less than or equal to</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>greater than</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>greater than or equal to</td>
</tr>
<tr>
<td><code>=</code></td>
<td>equal to</td>
</tr>
<tr>
<td><code>&lt;&gt;</code></td>
<td>not equal to</td>
</tr>
<tr>
<td><code>Not</code></td>
<td>reverse logic</td>
</tr>
</tbody>
</table>

Use a comparison operator to compare a specified value with all the values in a field. For example, you may want to find all customers who bought anything but a car. You can use a combination of comparison operators and field values to write an expression defining the desired criteria.
When you run the query, only the records with values meeting the criteria you specify appear in the RecordSet.

**Note**: Access automatically inserts number symbols (#) around date values and quotation marks (" ") around alphanumeric values. Access does not insert any symbols or characters around numeric values.

Figure 2 - Creating a Query with Tables and Criteria

**Using an AND Condition**

Many times, a query requires more than one condition to obtain the desired result. For example, if you want to find all customers in Connecticut or Massachusetts who bought anything other than a car (see Figure 3), you would need two conditions:

**State=CT and Category <> cars**

Figure 3 - Combined AND Criteria

**Note**: To combine two criteria in this way, you use the **And** logical operator.
You can use the **And** operator in a single field or in different fields. In a single field, you can use the **And** operator to find records that fall into a range. For example, to find customers whose sales dates fall between 2/25/08 and 4/30/08, you type both criteria on a single line in the **Criteria** row under the appropriate field.

>=2/25/08 and <=4/30/08 in the **SaleDate** field
The **And** operator also allows you to impose conditions in two or more different fields. For example, to find customers in CT with the last name Bond, you type each criterion on a single line in the **Criteria** row under the appropriate fields.

Enter **CT** in the **State** field and **Bond** in the **LastName** field in the “Criteria” row (see Figure 7).

**Figure 7 - Criteria for a Query**

![Figure 7 - Criteria for a Query](image1)

**Notes**: Criteria entered in the same **Criteria** row of the design grid create an **And** condition. Criteria entered in different **Criteria** rows create an **Or** condition.

**Figures 8 - Results of CT and Bond Query**

![Figure 8 - Results of CT and Bond Query](image2)

**Using an OR Condition**

Many times, a query requires more than one condition to obtain the desired result. For example, if you want to find customers in CT or with the last name of Bond, you would need two conditions:

**State=CT as well as LastName = Bond**

The records only need to meet one of the conditions in order to be included in the RecordSet. To combine two criteria in this way, you use the **Or** logical operator.

You can use the **Or** operator in a single field or in different fields. In a single field, you type the criteria on two separate lines under the same field. In different fields, you type the criteria on two separate lines under the appropriate fields. For example, to find all customers with sales dates on or before August 24, 2007 or credit limits above $80,000 type

**<=8/24/07** in the **Criteria** row under the **SaleDate** field and **>80000** in the **or** row under the **CreditLimit** field (see Figure 9).
You can create additional **Or** and **And** operators by typing criteria into the **Criteria** row, the **Or** row, or any row below the **Or** row.

- Criteria entered into the **same** **Criteria** row across the design grid create an **And** condition.
- Criteria entered into **different** **Criteria** rows create an **Or** condition.
- You can also create a combination of **And** and **Or** conditions.

**Note**: You can create an **Or** condition for a single field in the **Criteria** row using the **Or** operator. Typing **IL or GA or AZ** in the **Criteria** row of the **State** field finds all records in any of the three regions.
Using the BETWEEN AND Operator

You can use the **Between And** operator in a query to find data that is between two values. You can use this operator with a text, numeric, or date field. For example, to find all records of customers with credit limits between 8000 and 20,000, you would enter **Between 8000 And 20000** in the Criteria row under the CreditLimit field (see Figure 11).

**Figure 11 - BETWEEN AND Query**

The **Between And** operator is **inclusive**. All records with values that include or fall between the stated criteria are included in the RecordSet. Space must be included between the criteria and the words **Between** and **And**. If you do not include the proper spacing, the **Data type mismatch in criteria expression** error message will appear (see Figure 12).

**Figure 12 - Data Mismatch Error**

**Figure 13 - Between And Query Results for CreditLimit Between 8000 And 20000**
Using the Expression Builder

When you enter criteria, you are actually creating an expression. Expressions are calculations and can include database objects, operators, and values. Objects range from table fields to controls in a form. Operators are standard mathematical operators used in calculations, such as +, -, *, /, ( ), < >. Values can be numbers, dates, text, and built-in functions, as well as fields, control, and property identifiers.

You can create an expression by typing the expression elements, or you can use the Expression Builder. The Expression Builder is a tool that provides all the elements needed to build the expression.

The Expression Builder displays the Expression box in its top pane, a row of operator buttons below the Expression box, and three lower panes that display categories, subcategories, and values, respectively.

New expressions appear in the Expression box. You can use a combination of methods to build a new expression. You can type some elements of the expression and select others, such as operators, functions, and values from the element panes.

If you make a mistake, the Undo button allows you to undo previous actions, one at a time. Additionally, you can select and delete any elements you want to remove from the Expression box.

Open up the Expression Builder by right-clicking the Criteria row and selecting the Build... option. The Expression Builder window, as seen in Figure 14 below, will be displayed.

Figure 14 - Expression Builder Window

Notes: The Expression Builder is available in Design view for any database object for which you need to create an expression. For example, you can use the Expression Builder to create a validation rule for a table field.
You can insert an element from an element pane into the Expression box by double-clicking the element or by selecting it and clicking the Paste button in the Expression Builder.

**Using the Wildcard Character**

Wildcard characters are used in a query to find records when the criterion contains a pattern (such as all last names beginning with M) or is only partly known (such as the proper spelling - Kline or Klein). Wildcards take the place of one or several letters in a Text field or numbers in a Date/Time field.

The two most common wildcards are listed in the following table (see Figure 15).

Figure 15 - Chart of Wildcard Characters

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Representing</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Any single letter or number</td>
<td>Sm?th finds Smith and Smyth, whereas ?andy finds Sandy, Randy, etc.</td>
</tr>
</tbody>
</table>
| *        | One or more letters or numbers| M* finds all records that start with M. 8/*80 finds all dates in August, 2000.  
*ball* finds all records that have the word ball anywhere in the field. |

**Note**: Wildcards are not case sensitive. For example, *ill finds Bill and bill.

When you use wildcard characters (? and *) Access automatically inserts the word Like before the criteria, and quotation marks (” ”) around text.

Figure 16 - Query Using Wildcard
DESIGNING ADVANCED QUERIES

Creating a Calculated Field

Access allows you to create expressions that calculate new field values. For example, you can create an expression that multiplies the value in the Quantity field by the value in the Price field to calculate total sales.

In expressions, field names are enclosed in square brackets ([ ]); numbers are not. For example, to calculate 8% of ordered total amount and display the results in a column named Tax, enter Tax:[price]*.08 in the design grid. The colon separates the column from the expression (see Figure 1, below).

Figure 1 - Calculated Field

Calculated fields are created in queries. You can also use criteria to remove nonessential records, thereby allowing the query to run faster. The results of your query can then be used to generate a report.

Figure 2 - Tax Field Results

Notes: The field names used in an expression must be the same as the field names that appear in the table.

You can also use the Expression Builder to create a calculated field by selecting any blank field row and clicking the Build... button on the Query Setup group in the Design tab.

Another way of using the Expression Builder is by right-clicking any blank field row and selecting the Build option from the drop-down menu.
Formatting a Calculated Field

Once you have created a calculated field, you can change its properties as you would any other field on the design grid. Right-click on the Tax field and select Properties from the drop down menu (see Figure 3, below). The Format property determines how data appear in Datasheet view. For example, you can change the properties of a calculated field so that the field values display as currency. Click on the Format field, click the down arrow and select Currency from the menu (see Figure 4).

You can also change the format of a calculated field by right-clicking anywhere in it and selecting the Properties command.

Creating a Function Query

Access allows you to create a query that groups records by a selected field and applies a function that calculates values on other fields in the query according to your needs. For example, you can group records in a table by state and then select the Count function to find out how many customers (records) are in each state (field). You can also group by customer name (field) and calculate the Sum of each customer's orders (record values.)
There are several types of functions from which you can choose. The most commonly used functions are listed in the following table:

![Chart of Functions](image)

**Note:** You can perform more than one calculation on a specific field. In such a case, you must add the field to the design grid a second time and create the desired expression.

You can add a **Total** row by right-clicking anywhere in the design grid and selecting the **Totals** option.

![Selecting the Totals Option](image)

### Creating a Parameter Query

If you want to run a query with different criteria each time, you can create a **Parameter** query. A parameter query is a query that prompts the user for information when the query is run. Access then uses the information as the criteria and runs the query. The resulting RecordSet only includes those records that meet the criteria. This option allows you to avoid displaying the query in **Design** view each time you want to change the criteria.

You enter the text that will display in the prompt in the **Criteria** row under the appropriate field in the design grid and enclosed in square brackets ([ ]). You can set up a parameter query to prompt the user for more than one piece of information as well.
1. Enter the message you want to display in the **Criteria** row prompt.

![Figure 10 - Parameter Query](image1)

2. Enter the category you want to query. For example: cars, SUVs, trucks or minivans and then click the **OK** button.

![Figure 11 - Category Prompt](image2)

3. The query displays the desired information (see Figure 12).

![Figure 12 - Query Results for Cars](image3)

**Note:** You can add multiple parameters to a query. When you run the query, a prompt will appear for each parameter in it.

### Creating a Concatenation in a Query

Access allows you to combine two or more fields into one field. This process, known as **concatenation**, enables you to add field name text strings to one another. The text strings can follow each other with or without spaces. You can add other characters between the text strings if needed. For example, you can combine the individual **City**, **State**, and **Postal Code** fields into one field called **Address**. You can have the comma and space characters appear between the field text strings. This concatenation process can be performed by creating a query to combine two or more fields.
When typing expressions for concatenation, the first part of the expression defines the name of the new field. The second part of the expression defines the fields which are to be concatenated. These field names must be surrounded by brackets. The ampersand (&) appears between the field name brackets. Any additional characters that are to appear between the fields are surrounded by double quotes.

For example, the expression `FullName: [Last Name]&" , &"[First Name]` placed in Field: concatenates the Last Name and First Name fields and inserts the concatenated text string into a field called FullName. The new field displays the last name, a comma, a space, and the first name for each record in the table.

Figure 13 - Concatenation Example for FullName.

Notes: You may use concatenation to create text strings in forms and reports. Expression Builder can be used to concatenate text. Select any blank Field row and click the Build button on the Query Design toolbar.

Filtering a Query

You can apply filters to queries in the same way you apply filters to a table or form. Since the data you want to filter sometimes appears in two or more tables, you might need to create a multiple table query. Once the query is created, you can apply a filter to temporarily isolate the records that you want to view.

The Filter By Selection feature allows you to quickly and easily filter a query to display only those records in which the selected value appears. Conversely, the

Filter Excluding Selection feature filters out the selected value, leaving only those records that do not contain the selected value.
To filter a query in **Datasheet** view:

1. Click in the field containing the text you would like to filter.
2. Click the **Filter** button on the **Sort & Filter** group of the **Home** tab (see Figure 15).

![Figure 15 - Filter Command](image)

3. The **Filter** window opens (see Figure 16).

![Figure 16 - Filtering a Query](image)

4. Make your selections.
5. Click **OK**.
6. To remove the filter, click the **Remove Filter** button on the **Sort & Filter** group of the **Home** tab (see Figure 17).

![Figure 17 - Remove Filter Command](image)
**Note:** To filter a query, you can also right-click the field you want to filter, and either **Filter By Selection** (select) or **Filter Excluding Selection** (deselect) options from the dropdown list.

### CLASS EXERCISES

For the following exercises, open the **1Relationships** database.

1. Create a query using the **Orders** table that displays **CustomerID** and **OrderDate** for orders made between January 1, 2008 (enter as 1/1/2008) and January 30, 2008 (enter as 1/30/2008). Use the **AND** operator. Name the query **qryOrdersJan08**. (Hint: Create tab > Query Design)

2. Create a query using the **Customer** table that displays **ContactName** and **ContactTitle** for all Sales Managers or Marketing Managers. Use the **OR** operator. Name the query **qrySalesMarketingManagers**.

3. Create a query using the **Orders** table that displays **CustomerID**, **OrderDate** and **RequiredDate** for required orders between February 1, 2008 and February 28, 2008. Use the **BETWEEN AND** operator. Name the query **qryRequiredOrdersFeb08**.

4. Create a query using the **Customers** table that displays **CustomerID**, **ContactTitle** and **ContactName** for names that start with J. Use **wildcard characters**. Name the query **qryJContacts**

5. Create a query using the **Orders** table that displays **CustomerID** and **Freight**. Create a calculated field called **DiscountFreight** that subtracts $2 from the freight price. Name the query **qryDiscountFreight**. Format the field as **Currency**.

6. Create a query using the **Customers** table that displays **CustomerID**, **ContactTitle** and **ContactName**. Create a parameter query that queries the **ContactTitle** field. Name the query **qryContactTitle**

7. Create a query using the **Customers** table that displays a concatenation of **ContactName**, **CompanyName**. Name the query **qryContactCompany**.
HOW TO LEARN MORE

For Intermediate Access Users

The Microsoft.com website offers many different online tutorials. They are simple, convenient, and can be accessed via the internet. Below are some topics and links for these online tutorials.

Online Tutorials, Quick Reference Cards and How To’s

Guide to the Access 2007 user interface


Quick Reference Card for Up to speed with Access 2007


Import Data from Excel to a new table in Access


For Access Users New to Office 2007

On-line Training Tutorial, Quick Reference Cards & How To’s

What’s New in Microsoft Access 2007?


Get Up to Speed with Access 2007


Guide to Access 2007 Templates


Which file format should I use in Access 2007?


Converting a Database to the Access 2007 Format

**Printed Material**

There are many books available to help you learn to use Microsoft Office Access 2007. Here is one:

**Microsoft Access 2007 Bible** (includes a comprehensive reference CD) by Grohn, Stockman, Powell, Prague, Irwin, and Reardon.

---

**TRAINING AND SUPPORT**

**IT Training**

IT Training & Development offers training in many different applications at various skill levels. See what is coming up over the next few months by checking our website at:

[www.csun.edu/it/training](http://www.csun.edu/it/training)

**Contact Us:**

IT Training & Development  
Phone: (818) 677-1700 or x1700 (on campus)  
Email: training@csun.edu

**Troubleshooting and Support**

If you experience problems getting started with Office 2007, contact the Help Center at x1400 or helpcenter@csun.edu.