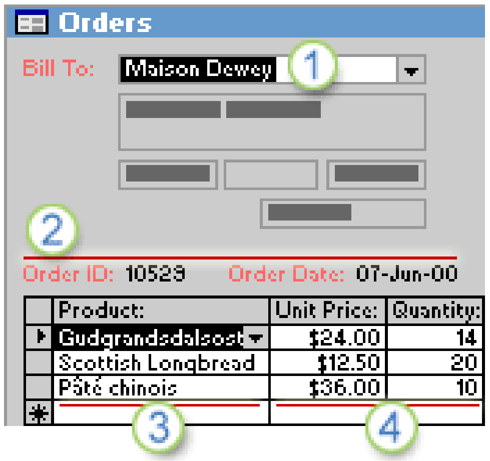
Guide to Table Relationships

Table of Contents

One of the goals of good database design is to remove data redundancy (duplicate data). To achieve that goal, you divide your data into many subject-based tables so that each fact is represented only once. You then provide Access with the means by which to bring the divided information back together — you do this by placing common fields in tables that are related. To do this step correctly, however, you must first understand the relationships between your tables, and then specify these relationships in your database.

Introduction

After you have created a table for each subject in your database, you must provide Access with the means by which to bring that information back together again when needed. You do this by placing common fields in tables that are related, and by defining relationships between your tables. You can then create queries, forms, and reports that display information from several tables at once. For example, the form shown on the next page includes information drawn from several tables:



Bill To: Maison Dewey

Order ID: 10529 Order Date: 07-Jun-00

Product:	Unit Price:	Quantity:
Gudgrandsdalsost	$24.00	14
Scottish Longbread	$12.50	20
Pâté chinois	$36.00	10

1 Information in this form comes from the Customers table...

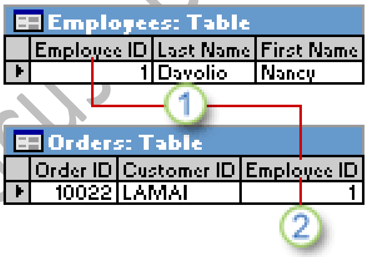
2 ...the Orders table...

3 ...the Products table...

4 ...and the Order Details table.

The customer name in the **Bill To** box is retrieved from the Customers table, the Order ID and the Order Date values come from the Orders table, the Product name comes from the Products table, and the Unit Price and Quantity values come from the Order Details table. These tables are linked to each other in a variety of ways to bring information from each into the form.

In the example, the fields in the tables must be coordinated so that they show information about the same order. This coordination is accomplished by using table relationships. A table relationship works by matching data in key fields — often a field with the same name in both tables. In most cases, these matching fields are the **primary key** from one table, which provides a unique identifier for each record, and a **foreign key** in the other table. For example, employees can be associated with orders for which they are responsible by creating a table relationship between the EmployeeID fields in the Employees and the Orders tables.



Employees: Table

Employee ID	Last Name	First Name
1	Davolio	Nancy

Orders: Table

Order ID	Customer ID	Employee ID
10022	LAMAI	1

1 EmployeeID appears in both tables – as a primary key ...

2 ... and as a foreign key.

Types of table relationships

There are three types of table relationships.

- A one-to-many relationship
    - Consider an order tracking database that includes a Customers table and an Orders table. A customer can place any number of orders. It follows that for any customer represented in the Customers table, there can be many orders represented in the Orders table. The relationship between the Customers table and the Orders table is, therefore, a one-to-many relationship.
    - To represent a one-to-many relationship in your database design, take the primary key on the "one" side of the relationship and add it as an additional field or fields to the table on the "many" side of the relationship. In this case, for example, you add a new field — the ID field from the Customers table — to the Orders table and name it Customer ID. Access can then use the Customer ID number in the Orders table to locate the correct customer for each order.
- A many-to-many relationship
    - Consider the relationship between a Products table and an Orders table. A single order can include more than one product. On the other hand, a single product can appear on many orders. Therefore, for each record in the Orders table, there can be many records in the Products table. In addition, for each record in the Products table, there can be many records in the Orders table. This type of relationship is called a many-to-many relationship because, for any product, there can be many orders and, for any order, there can be many products. Note that to detect existing many-to-many relationships between your tables, it is important that you consider both sides of the relationship.

    - To represent a many-to-many relationship, you must create a third table, often called a junction table, that breaks down the many-to-many relationship into two one-to-many relationships. You insert the primary key from each of the two tables into the third table. As a result, the third table records each occurrence, or instance, of the relationship. For example, the Orders table and the Products table have a many-to-many relationship that is defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.
- A one-to-one relationship
    - In a one-to-one relationship, each record in the first table can have only one matching record in the second table, and each record in the second table can have only one matching record in the first table. This type of relationship is not common because, most often, the information related in this way is stored in the same table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. When you do identify such a relationship, both tables must share a common field.

Why create table relationships?

You can create table relationships explicitly by using the Relationships window, or by dragging a field from the **Field List** pane. Access uses table relationships to know how to join tables when you need to use them in a database object. There are several reasons why you should create table relationships before you create other database objects, such as forms, queries and reports.

- Table relationships inform your query designs
    - To work with records from more than one table, you often must create a query that joins the tables. The query works by matching the values in the primary key field of the first table with a foreign key field in the second table. For example, to return rows that list all of the orders for each customer, you construct a query that joins the Customers table with the Orders table based on the Customer ID field. In the Relationships window, you can manually specify the fields to join. However, if you already have a relationship defined between the tables, Access supplies the default join, based on existing the table relationship. In addition, if you use one of the query wizards, Access uses the information it gathers from the table relationships you have already defined to present you with informed choices and to prepopulate property settings with appropriate default values.
- Table relationships inform your form and report designs
    - When you design a form or report, Access uses the information it gathers from the table relationships you have already defined to present you with informed choices and to prepopulate property settings with appropriate default values.
- Table relationships are the foundation upon which you can enforce referential integrity to help prevent orphan records in your database. An orphan record is a record with a reference to another record that does not exist — for example, an order record that references a customer record that does not exist.
    - When you design a database, you divide your information into tables, each of which has a primary key. You then add foreign keys to related tables that reference those primary keys. These foreign key-primary key pairings form the basis for table relationships and multi-table queries. It is important, therefore, that these foreign key-primary key references stay synchronized. Referential integrity helps ensure that references stay synchronized and is dependent upon table relationships.

Understanding referential integrity

When you design a database, you divide your information into many subject-based tables to minimize data redundancy. You then provide Access with the means by which to bring the data back together by placing common fields into related tables. For example, to represent a one-to-many relationship you take the primary key from the "one" table and add it as an additional field to the "many" table. To bring the data back together, Access takes the value in the "many" table and looks up the corresponding value in the "one" table. In this way the values in the "many" table reference the corresponding values in the "one" table.

Suppose you have a one-to-many relationship between Shippers and Orders and you want to delete a Shipper. If the shipper you want to delete has orders in the Orders table, those orders will become "orphans" when you delete the Shipper record. The orders will still contain a shipper ID, but the ID will no longer be valid, because the record that it references no longer exists.

The purpose of referential integrity is to prevent orphans and keep references in sync so that this hypothetical situation never occurs.

You enforce referential integrity by enabling it for a table relationship. Once enforced, Access rejects any operation that would violate referential integrity for that table relationship. This means Access will reject both updates that change the target of a reference, and deletions that remove the target of a reference. However, it is possible you might have a perfectly valid need to change the primary key for a shipper that has orders in the Orders table. For such cases, what you really need is for Access to automatically update all the effected rows as part of a single operation. That way, Access ensures that the update is completed in full so that your database is not left in an inconsistent state, with some rows updated and some not. For this reason Access supports the Cascade Update Related Fields option. When you enforce referential integrity and choose the Cascade Update Related Fields option, and you then update a primary key, Access automatically updates all fields that reference the primary key.

It's also possible you might have a valid need to delete a row and all related records — for example, a Shipper record and all related orders for that shipper. For this reason, Access supports the Cascade Delete Related Records option. When you enforce referential integrity and choose the Cascade Delete Related Records option, and you then delete a record on the primary key side of the relationship, Access automatically deletes all records that reference the primary key.

View table relationships

To view your table relationships, click **Relationships** on the **Database Tools** tab. The Relationships window opens and displays any existing relationships. If no table relationships have yet been defined and you are opening the Relationships window for the first time, Access prompts you to add a table or query to the window.

Open the Relationships window

1) On the **File** tab, click **Open**.
2) In the **Open** dialog box, select and open the database.
3) On the **Database Tools** tab, in the **Relationships** group, click **Relationships**.
4) If the database contains relationships, the Relationships window appears. If the database does not contain any relationships and you are opening the Relationships window for the first time, the **Show Table** dialog box appears.
    a) Click **Close** to close the dialog box.

b) On the **Design** tab, in the **Relationships** group, click **All Relationships**.

- This displays all of the defined relationships in your database.

  Note that hidden tables (tables for which the **Hidden** check box in the table's **Properties** dialog box is selected) and their relationships will not be shown unless the **Show Hidden Objects** check box is selected in the **Navigation Options** dialog box.

A table relationship is represented by a relationship line drawn between tables in the Relationships window. A relationship that does not enforce referential integrity appears as a thin line between the common fields supporting the relationship. When you select the relationship by clicking its line, the line thickens to indicate it is selected. If you enforce referential integrity for this relationship, the line appears thicker at each end. In addition, the number **1** appears over the thick portion of the line on one side of the relationship, and the infinity symbol (∞) appears over the thick portion of the line on the other side.

When the Relationships window is active, you can select from the following commands on the ribbon:

On the **Design** tab, in the **Tools** group:

- **Edit Relationships**: Opens the **Edit Relationships** dialog box. When you select a relationship line, you can click **Edit Relationships** to change the table relationship. You can also double-click the relationship line.
- **Clear Layout**: Removes all tables and relationships from display in the Relationships window. Note that this command only hides the tables and relationships — it does not delete them.
- **Relationships Report**: Creates a report that displays the tables and relationships in your database. The report shows only the tables and relationships that are not hidden in the Relationships window.

On the **Design** tab, in the **Relationships** group:

- **Show Table**: Opens the **Show Table** dialog box so that you can select tables and queries for viewing in the Relationships window.
- **Hide Table**: Hides the selected table in the Relationships window.
- **Direct Relationships**: Displays all relationships and related tables for the selected table in the Relationships window, if they are not already displayed.
- **All Relationships**: Displays all of the relationships and related tables in your database in the Relationships window.
  - Note that hidden tables (tables for which the **Hidden** check box in the table's **Properties** dialog box is selected) and their relationships will not be shown unless Show Hidden Objects is selected in the Navigation Options dialog box.
- **Close**: Closes the Relationships window. If you made any changes to the layout of the Relationships window, you are asked whether to save those changes.

Create a table relationship

You can create a table relationship by using the Relationships window, or by dragging a field onto a datasheet from the **Field List** pane. When you create a relationship between tables, the common fields are not required to have the same names, although it is often the case that they do. Rather, those fields must have the same data type. If the primary key field is an AutoNumber field, however, the foreign key field can be a Number field if the **FieldSize** property of both fields is the same. For example, you can match an AutoNumber field and a Number field if the **FieldSize** property of both fields is Long Integer. When both common fields are Number fields, they must have the same **FieldSize** property setting.

Create a table relationship by using the Relationships window

1) On the **File** tab, click **Open**.
2) In the **Open** dialog box, select and open the database.
3) On the **Database Tools** tab, in the **Relationships** group, click **Relationships**.
4) If you have not yet defined any relationships, the **Show Table** dialog box automatically appears. If it does not appear, on the **Design** tab, in the **Relationships** group, click **Show Table**.
   a) The **Show Table** dialog box displays all of the tables and queries in the database. To see only tables, click **Tables**. To see only queries, click **Queries**. To see both tables and queries, click **Both**.
5) Select one or more tables or queries and then click **Add**. When you have finished adding tables and queries to the Relationships window, click **Close**.
6) Drag a field (typically the primary key) from one table to the common field (the foreign key) in the other table. To drag multiple fields, press the CTRL key, click each field, and then drag them.
   a) The **Edit Relationships** dialog box appears.
   b) Verify that the field names shown are the common fields for the relationship. If a field name is incorrect, click the field name and select a new field from the list.
   c) To enforce referential integrity for this relationship, select the **Enforce Referential Integrity** check box.
   d) Click **Create**.
7) The relationship line is drawn between the two tables. If you selected the **Enforce Referential Integrity** check box, the line appears thicker at each end.
   a) In addition, again only if you selected the **Enforce Referential Integrity** check box, the number **1** appears over the thick portion of the line on one side of the relationship, and the infinity symbol (∞) appears over the thick portion of the line on the other side.

Notes:

- **To create a one-to-one relationship**: Both of the common fields (usually the primary key and foreign key fields) must have a unique index. This means the **Indexed** property for these fields should be set to **Yes (No Duplicates)**. If both fields have a unique index, Access creates a one-to-one relationship.
- **To create a one-to-many relationship**: The field on the "one" side (typically the primary key) of the relationship must have a unique index. This means the **Indexed** property for this field should be set to **Yes (No Duplicates)**. The field on the "many" side should *not* have a unique index. It can have an index, but it must allow duplicates. This means the **Indexed** property for this field should be set to either **No**, or **Yes (Duplicates OK)**. When one field has a unique index and the other does not, Access creates a one-to-many relationship.

Create a table relationship by using the Field List pane

You can add a field to an existing table that is open in Datasheet view by dragging it from the **Field List** pane. The **Field List** pane shows fields available in related tables and also fields available in other tables. When you drag a field from an "other" (unrelated) table and then complete the Lookup Wizard, a new one-to-many relationship is automatically created between the table in the **Field List** pane and the table to which you dragged the field. This relationship, created by Access, does not enforce referential integrity by default. To enforce referential integrity, you must edit the relationship.

Open a table in Datasheet view

1) On the **File** tab, click **Open**.
2) In the **Open** dialog box, select and open the database.
3) In the Navigation Pane, right-click the table to which you want to add the field and create the relationship, and then click **Open**.

Open the Field List pane

1) Press ALT+F8.
   a) The **Field List** pane appears.

The **Field List** pane shows all of the other tables in your database, grouped into categories. When you work with a table in Datasheet view, Access displays fields in either of two categories in the **Field List** pane: **Fields available in related tables** and **Fields available in other tables**. The first category lists all of the tables that have a relationship with the table you are currently working with. The second category lists all of the tables with which your table does not have a relationship.

In the **Field List** pane, when you click the plus sign (**+**) next to a table name, you see a list of all the fields available in that table. To add a field to your table, drag the field that you want from the **Field List** pane to the table in Datasheet view.

Add a field and create a relationship from the Field List pane

1) In the **Field List** pane, under **Fields available in other tables**, click the plus sign (**+**) next to a table name to display the list of fields in that table.
2) Drag the field that you want from the **Field List** pane to the table that is open in Datasheet view.
3) When the insertion line appears, drop the field in position.
   a) The **Lookup Wizard** starts.
4) Follow the instructions to complete the **Lookup Wizard**.
   a) The field appears in the table in Datasheet view.

When you drag a field from an "other" (unrelated) table and then complete the Lookup Wizard, a new one-to-many relationship is automatically created between the table in the **Field List** and the table to which you dragged the field. This relationship, created by Access, does not enforce referential integrity by default. To enforce referential integrity, you must edit the relationship.

Delete a table relationship

To remove a table relationship, you must delete the relationship line in the Relationships window. Carefully position the cursor so that it points at the relationship line, and then click the line. The relationship line appears thicker when it is selected. With the relationship line selected, press DELETE. Note that when you remove a relationship, you also remove referential integrity support for that relationship, if it is enabled. As a result, Access will no longer automatically prevent the creation of orphan records on the "many" side of a relationship.

1) On the **Database Tools** tab, in the **Relationships** group, click **Relationships**.
   a) The Relationships window appears. If you have not yet defined any relationships and this is the first time you are opening the Relationships window, the **Show Table** dialog box appears. If the dialog box appears, click **Close**.
   b) All tables that have relationships are displayed, showing relationship lines.

2) Click the relationship line for the relationship that you want to delete. The relationship line appears thicker when it is selected.
3) Press the DELETE key.
   a) Or, right-click and then click **Delete**.
4) Access might display the message: “Are you sure you want to permanently delete the selected relationship from your database?” If this confirmation message appears, click Yes.

Note: If either of the tables employed in the table relationship are in use, perhaps by another person or process, or in an open database object (such as a form), you will not be able to delete the relationship. You must first close any open objects that use these tables before you can remove the relationship.

Change a table relationship

You change a table relationship by selecting it in the Relationships window and then editing it. Carefully position the cursor so that it points at the relationship line, and then click the line to select it. The relationship line appears thicker when it is selected. With the relationship line selected, double-click it or click **Edit Relationships** in the **Tools** group on the **Design** tab. The **Edit Relationships** dialog box appears.

Make your changes in the Edit Relationships dialog box

1) On the **Database Tools** tab, in the **Relationships** group, click **Relationships**.
   a) The Relationships window appears. If you have not yet defined any relationships and this is the first time you are opening the Relationships window, the **Show Table** dialog box appears. If the dialog box appears, click **Close**.
2) All tables that have relationships are displayed, showing relationship lines.
3) Click the relationship line for the relationship that you want to change. The relationship line appears thicker when it is selected.
4) Double-click the relationship line.
5) Make your changes, and then click **OK**.

The **Edit Relationships** dialog box allows you to change a table relationship. Specifically, you can change the tables or queries on either side of the relationship, or the fields on either side. You can also set the join type, or enforce referential integrity and choose a cascade option.

Join types

When you define a table relationship, the facts about the relationship inform your query designs. For example, if you define a relationship between two tables, and you then create a query that employs those tables, Access automatically selects the default matching fields based upon the fields specified in the relationship. You can override these initial default values in your query, but the values supplied by the relationship will often prove to be the correct ones. Because matching and bringing together data from more than one table is something you will do frequently in all but the most simple databases, setting defaults by creating relationships can be time saving and beneficial.

A multiple table query combines information from more than one table by matching the values in common fields. The operation that does the matching and combining is called a join. For instance, suppose you want to display customer orders. You create a query that joins the Customers table and the Orders table on the Customer ID field. The query result contains customer information and order information for only those rows where a corresponding match was found.

One of the values you can specify for each relationship is the join type. The join type tells Access which records to include in a query result. For example, consider again a query that joins the Customers table and the Orders table on the common fields that represent the Customer ID. Using the default join type (called an inner join), the query returns only the Customer rows and the Order rows where the common fields (also called the joined fields) are equal.

However, suppose you want to include all Customers — even those who have not yet placed any orders. To accomplish this, you must change the join type from an inner join to what is called a left outer join. A left outer join returns all of the rows from the table on the left side of the relationship and only those that match from the table on the right. A right outer join returns all of the rows on the right and only those that match on the left.

Note: In this case, "left" and "right" refer to the position of the tables in the **Edit Relationships** dialog box, not the Relationships window.

You should think about the result you will most often want from a query that joins the tables in this relationship, and then set the join type accordingly.

Set the join type

1) In the **Edit Relationships** dialog box, click **Join Type**.
    a) The Join Properties dialog box appears.
2) Click your choice, and then click **OK**.

The following table (using the Customers and Orders tables) shows the three choices that are displayed in the **Join Properties dialog** box, the type of join they use, and whether all rows or matching rows are included for each table.

Choice	Relational join	Left table	Right table
1) Only include rows where the joined fields from both tables are equal.	Inner join	Matching rows	Matching rows
2) Include ALL records from 'Customers' and only those records from 'Orders' where the joined fields are equal.	Left outer join	All rows	Matching rows
3) Include ALL records from 'Orders' and only those records from 'Customers' where the joined fields are equal.	Right outer join	Matching rows	All rows

When you choose option 2 or option 3, an arrow is shown on the relationship line. This arrow points to the side of the relationship that shows only matching rows.

Make changes in the Join Properties dialog box

1) On the **Database Tools** tab, in the **Relationships** group, click **Relationships**.
2) Click the relationship line for the relationship that you want to change.
   a) The relationship line appears thicker when it is selected.
3) Double-click the relationship line.
   a) The **Edit Relationships** dialog box appears.
4) Click **Join Type**
5) In the **Join Properties** dialog box, click an option, and then click **OK**.
6) Make any additional changes to the relationship, and then click **OK**.

Enforce referential integrity

The purpose of using referential integrity is to prevent orphan records and to keep references synchronized so that you don't have any records that reference other records that no longer exist. You enforce referential integrity by enabling it for a table relationship. Once enforced, Access rejects any operation that would violate referential integrity for that table relationship. Access rejects updates that change the target of a reference, and also deletions that remove the target of a reference.

Turn referential integrity on or off

1) In the Relationships window, click the relationship line for the relationship that you want to change.
   a) The relationship line appears thicker when it is selected.
2) Double-click the relationship line.
3) Select the **Enforce Referential Integrity** check box.
4) Make any additional changes to the relationship, and then click **OK**.

After you have enforced referential integrity, the following rules apply:

- You cannot enter a value in the foreign key field of a related table if that value doesn't exist in the primary key field of the primary table — doing so creates orphan records.
- You cannot delete a record from a primary table if matching records exist in a related table. For example, you cannot delete an employee record from the Employees table if there are orders assigned to that employee in the Orders table. You can, however, choose to delete a primary record *and* all related records in one operation by selecting the **Cascade Delete Related Records** check box.
- You cannot change a primary key value in the primary table if doing so would create orphan records. For example, you cannot change an order number in the Orders table if there are line items assigned to that Order in the Order Details table. You can, however, choose to update a primary record *and* all related records in one operation by selecting the **Cascade Update Related Fields** check box.

Notes:

- If you have difficulty enabling referential integrity, note that the following conditions are required in order to enforce referential integrity:
    - The common field from the primary table must be a primary key or have a unique index.
    - The common fields must have the same data type. The one exception is that an AutoNumber field can be related to a Number field that has a **FieldSize** property setting of **Long Integer**.
    - Both tables must exist in the same Access database. Referential integrity cannot be enforced on linked tables. However, if the source tables are in Access format, you can open the database in which they are stored and enable referential integrity in that database.

Set the cascade options

You might encounter a situation in which you have a valid need to change the value on the "one" side of a relationship. In such a case, you need Access to automatically update all of the effected rows as part of a single operation. That way, the update is completed in full so that your database is not left in an inconsistent state — with some rows updated and some not. Access helps you avoid this problem by supporting the Cascade Update Related Fields option. When you enforce referential integrity and choose the Cascade Update Related Fields option, and you then update a primary key, Access automatically updates all fields that reference the primary key.

You might also need to delete a row and all related records — for instance, a shipper record and all related orders for that shipper. For this reason, Access supports the Cascade Delete Related Records option. When you enforce referential integrity and choose the Cascade Delete Related Records option, Access automatically deletes all records that reference the primary key when you delete the record that contains the primary key.

Turn cascade update and/or cascade delete on or off

1) In the Relationships window, click the relationship line for the relationship that you want to change. The relationship line appears thicker when it is selected.
2) Double-click the relationship line.
3) Select the **Enforce Referential Integrity** check box.
4) Select either the **Cascade Update Related Fields** or **Cascade Delete Related Records** check box, or select both.
5) Make any additional changes to the relationship, and then click **OK**.

Note: If the primary key is an AutoNumber field, selecting the **Cascade Update Related Fields** check box will have no effect, because you cannot change the value in an AutoNumber field.