Database Concepts 3.5

David M. Kroenke and David J. Auer

THE ACCESS WORKBENCH

Section 5

Relationships in Microsoft Access
At this point, we have created and populated the CONTACT, CUSTOMER, and SALESPE RSON tables in the Wallingford Motors CRM database. We have also learned how to create forms, reports, and queries in the preceding sections, and how to create and use view equivalent queries in Appendix C.

However, all the tables we’ve used so far have had 1:N relationships, which raised that question: How are 1:1 and N:M relationships managed in Access? In this section, we will cover the following objectives:

- Understand 1:1 relationships in Access.
- Understand N:M relationships in Access.

**N:M Relationships in Access**

We’ll start by discussing N:M relationships. This is actually a nonissue, because pure N:M relationships occur in data modeling. Remember that when a data model is transformed into a database design, an N:M relationship is broken down into two 1:N relationships. Each 1:N relationship is between a table resulting from one of the original entities in the N:M relationship and a new intersection table. If this doesn’t make sense to you, then review the chapter section “Representing N:M Strong Entity Relationships” and see Figures 5-13 and 5-15 for an illustration of how N:M relationships are converted to two 1:N relationships.
Since databases are built in DBMSs only as a final database design, DBMSs only need to work with the resulting 1:N relationships. Thus, as far as Access is concerned, there are no N:M relationships—only 1:N relationships!

**1:1 Relationships in Access**

1:1 relationships, however, are another matter—they definitely exist in Access. Since our current WMCRM database doesn’t contain a 1:1, let’s add one to it. We’ll let each SALESPERSON use one and only one car from the Wallingford Motors inventory as a demo vehicle. Our database design with this addition is shown in Figure AW-5-1.

![Figure AW-5-1 — The WMCRM Database with VEHICLE](image)

Note that both SALESPERSON and VEHICLE are optional in this relationship. First, a VEHICLE does not have to be assigned to a SALESPERSON, which makes sense as there will be a lot of cars in inventory and only a few SALESPERSONs. Second, a SALESPERSON does not have to take a demo car and may choose not to (yeah, right!). Also note that we have chosen to put the foreign key in SALESPERSON. This is a case where there is an advantage to putting the foreign key in one table or the other because if we’d put it in VEHICLE, the foreign key column (which would have been NickName) would have been NULL for every car except the few used as demo
vehicles. Finally, this is a table we’re using just to illustrate a 1:1 relationship—a functional VEHICLE table would have a lot more columns.

The column characteristics for the VEHICLE table are shown in Figure AW-5-2, and the data for the table is shown in Figure AW-5-3.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Key</th>
<th>Required</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory ID</td>
<td>AutoNumber</td>
<td>Primary Key</td>
<td>Yes</td>
<td>Surrogate Key</td>
</tr>
<tr>
<td>Model</td>
<td>Text(25)</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>VIN</td>
<td>Text(35)</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Figure AW-5-2 — Database Column Characteristics for the VEHICLE Table

<table>
<thead>
<tr>
<th>Inventory ID</th>
<th>Model</th>
<th>VIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AutoNumber]</td>
<td>HiStandard</td>
<td>G06HS123400001</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>HiStandard</td>
<td>G06HS123400002</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>HiStandard</td>
<td>G06HS123400003</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>HiLuxury</td>
<td>G06HL234500001</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>HiLuxury</td>
<td>G06HL234500002</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>HiLuxury</td>
<td>G06HL234500003</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>SUHi</td>
<td>G06HU345600001</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>SUHi</td>
<td>G06HU345600002</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>SUHi</td>
<td>G06HU345600003</td>
</tr>
<tr>
<td>[AutoNumber]</td>
<td>HiElectra</td>
<td>G06HE456700001</td>
</tr>
</tbody>
</table>

Figure AW-5-3 — Wallingford Motors VEHICLE Data

Let’s open the WMCRM.accdb database.

Opening the WMCRM.accdb Database

1. Start Microsoft Access.
2. Click the Microsoft Office Button, and then click the Open button. The Open dialog box is displayed. Browse to the WMCRM.accdb file, click the file name to highlight it, and then click the Open button.
3. The Security Warning bar appears with the database. Click the Security Warning Options... button to display Microsoft Office Security Options dialog box.
4. Click the “Enable this content” radio button to select this option, and then click the OK button.
We already know how to create a table and populate it with data, so we’ll go ahead and add the VEHICLE table and its data to the WMCRM.accdb database. Next, we need to modify SALESPERSON by adding the InventoryID column and populating it with data. The column characteristics for the new InventoryID column in the SALESPERSON table are shown in Figure AW-5-4, and the data for the column is shown in Figure AW-5-5 (Tina and Big Bill are driving the HiLuxury model, while Billy opted for a SUHi).

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Key</th>
<th>Required</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>InventoryID</td>
<td>Int</td>
<td>Foreign</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**Figure AW-5-4 — Database Column Characteristics for the InventoryID Column**

<table>
<thead>
<tr>
<th>NickName</th>
<th>LastName</th>
<th>FirstName</th>
<th>...</th>
<th>InventoryID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tina</td>
<td>Smith</td>
<td>Tina</td>
<td>...</td>
<td>4</td>
</tr>
<tr>
<td>Big Bill</td>
<td>Jones</td>
<td>William</td>
<td>...</td>
<td>5</td>
</tr>
<tr>
<td>Billy</td>
<td>Jones</td>
<td>Bill</td>
<td>...</td>
<td>7</td>
</tr>
</tbody>
</table>

**Figure AW-5-5 — SALESPERSON InventoryID Data**

There is nothing here that we don’t know how to do—we altered the CUSTOMER table in a similar way in Section 3 of "The Access Workbench"—so we will go ahead and add the InventoryID column and its data to the SALESPERSON table. This is an easier table alteration to make than the one we made to CUSTOMER because the InventoryID column in SALESPERSON is NOT NULL so we do not have to set it to NULL after entering the data.

Now, we’re ready to establish the relationship between the two tables.

**Creating the Relationship Between SALESPERSON and VEHICLE**

1. Click the **Database Tools** command tab to display the **Database Tools** ribbon.
2. Click the **Shutter Bar Open/Close Button** to compress the Navigation Pane.
3. Click the **Relationships** button in the **Show/Hide** group. The Relationships tabbed document window appears.
**NOTE: WARNING!** We are about to encounter another peculiarity of Access. The next steps lead to that situation, not the final outcome that we want. Remember that we want a 1:1 relationship and see if you can figure out what’s happening as we go along.

4. Click the **Show Table** button in the **Relationships** group of the **Design** ribbon.

5. In the Show Table dialog box, click **VEHICLE** to select it, and then click the **Add** button to add VEHICLE to the Relationships window.

6. In the Show Table dialog box, click the **Close** button to close the dialog box.

7. You can rearrange and resize the table objects in the Relationships window using standard Windows drag-and-drop techniques. Rearrange the **SALESPERSON**, **CUSTOMER**, **CONTACT**, and **VEHICLE** table objects until they appear as shown in Figure AW-5-6.

![The new VEHICLE table](image)

Figure AW-5-6 — The Relationships Window with the Current Relationship Diagram

**NOTE:** Remember that we create a relationship between two tables in the Relationships window by dragging a primary key column and dropping it on top of the corresponding foreign key column.

8. Click and hold the **column name InventoryID** in the **VEHICLE** table object, and then drag it over the **column name InventoryID** in the **SALESCUSTOMER** table. The Edit Relationships dialog box appears.

9. Click the **Enforce Referential Integrity** check box.

10. Click the **Create** button to create the relationship between VEHICLE and SALESPERSON.

11. Right-click the **relationship line** between SALESPERSON and CUSTOMER, and then click **Edit Relationship** in the shortcut menu that appears. The **Edit Relationships dialog box** appears.

12. The relationship between the tables now appears in the Relationships window as shown in Figure AW-5-7 on the next page.
But now we have a serious problem: the relationship that was created is a 1:N relationship, not the 1:1 relationship that we wanted. Let’s fix it. There must be a way to fix the relationship somewhere on the Edit Relationship dialog box! Unfortunately, there isn’t. Go ahead and try everything you can think of, it won’t work. This is the peculiarity of Access that we mentioned.

So just what is the trick to creating a 1:1 relationship in Access? The trick is setting the indexed property of the foreign key column (InventoryID in SALESPERSON in this case) as Indexed (no duplicates) as shown in Figure AW-5-8 on the next page. As long as the same value can occur more than once in the foreign key column, Access will create a 1:N relationship instead of the desired 1:1 relationship.
To create the 1:1 relationship, we will need to delete the existing relationship, modify the InventoryID property in SALESPERSON, and create a new relationship between the tables.

**Deleting the Incorrect Relationship Between SALESPERSON and VEHICLE**

1. Click the OK button on the Edit Relationships dialog box.
2. Right-click on the **relationship line** between VEHICLE and SALESPERSON to display the shortcut menu, then click **Delete**.
3. A dialog box will be displayed with the message “Are you sure you want to permanently delete the selected relationship from your database?” Click the **Yes** button.
5. A dialog box will be displayed with the message “Do you want to save the changes to the layout of ‘Relationships’?” Click the **Yes** button.
6. Click the **Shutter Bar Open/Close Button** to expand the Navigation Pane.

---

**Figure AW-5-8 — Setting the Indexed Property Value in the SALESPERSON Table**

Select the **InventoryID** column

Set the Indexed property to **Yes** *(No Duplicates)*
Setting the Indexed Property of the InventoryID Column in SALESPERSON

1. Open the SALESPERSON table in Design View.
2. Select the InventoryID field. The InventoryID field properties are displayed in the General tab.
3. Click the Indexed text field. A drop-down list arrow appears on the right end of the text field. Click the drop-down list arrow to display the list, and select Yes (No Duplicates). The result appears as shown in Figure AW-5-8.
4. Click the Save button to save the completed changes to the SALESPERSON table.
5. Close the SALESPERSON table.

Creating the Correct 1:1 Relationship Between SALESPERSON and VEHICLE

1. Click the Database Tools command tab to display the Database Tools ribbon.
2. Click the Shutter Bar Open/Close Button to compress the Navigation Pane.
3. Click the Relationships button in the Show/Hide group. The Relationships tabbed document window appears.
4. Click and hold the column name InventoryID in the VEHICLE table object, and then drag it over the column name InventoryID in the SALECUSTOMER table. The Edit Relationships dialog box appears.
5. Click the Enforce Referential Integrity check box.
6. Click the Create button to create the relationship between VEHICLE and SALESPERSON.
7. Right-click the relationship line between SALESPERSON and CUSTOMER, and then click Edit Relationship in the shortcut menu that appears. The Edit Relationships dialog box appears.
8. Note that the correct one-to-one relationship between the tables now appears in the Relationships window as shown in Figure AW-5-9 on the next page.
9. Click the Cancel button on the Edit Relationships dialog box.
11. A dialog box will be displayed with the message “Do you want to save the changes to the layout of ‘Relationships’?” Click the Yes button.
12. Click the Shutter Bar Open/Close Button to expand the Navigation Pane.
We have successfully created the 1:1 relationship that we wanted—we just had to learn the Access way of doing it!

**Closing the Database and Exiting Access**

That completes the work we'll do in this section of “The Access Workbench.” As usual, we will finish by closing the database and Access.

**Closing the WMCRM Database and Exiting Access**

1. To close the WMCRM database and exit Access, click the Microsoft Access **Close** button in the upper right corner of the Microsoft Access window.
Access Workbench Exercises

AW.5.1 Using an IE Crow’s Foot E-R diagram, draw a database design for the Wedgewood Pacific Corporation (WPC) database as completed at the end of Section 3 of “The Access Workshop.”

AW.5.2 This chapter’s “The Access Workshop” described how to create 1:1 relationships in Access. In particular, we added the business rule that each salesperson at Wallingford Motors can have one and only one vehicle as a demo car. Suppose that the rule has been changed so that each salesperson can have one or more cars as demo vehicles.

A. Using an IE Crow’s Foot E-R diagram, redraw the database design in Figure AW-5-1 to show the new relationship between VEHICLE and SALESPERSON. Which table(s) is (are) the parent(s) in the relationship? Which table(s) is (are) the child(ren)? In which table(s) do you place a foreign key?

B. Start with the Wallingford Motors database that you’ve created so far (WMCRM.accdb) as it exists after working through all the steps in this section’s “The Access Workshop” (If you haven’t completed these actions, do so now). Make two copies of the WMCRM.accdb database (Note: Copying an Access database is discussed in Section 4 of “The Access Workbench”). Name the first copy as WMCRM-AW05-v01.accdb, and name the second copy as WMCRM-AW05-v02.accdb.

C. Modify the WMCRM-AW05-v02.accdb database to implement the new relationship between VEHICLE and SALESPERSON.

AW.5.3 This chapter’s “The Access Workshop” described how to create 1:1 relationships in Access. In particular, we added the business rule that each salesperson at
Wallingford Motors can have one and only one vehicle as a demo car. Suppose that the rule has been changed so that (1) each salesperson can have one or more cars as demo vehicles, and (2) each demo vehicle can be shared by two or more salespersons.

A. Using an IE Crow’s Foot E-R diagram, redraw the database design in Figure AW-5-1 to show the new relationship between VEHICLE and SALESPERSON. Which table(s) is (are) the parent(s) in the relationship? Which table(s) is (are) the child(ren)? In which table(s) do you place a foreign key?

B. Start with the Wallingford Motors database that you’ve created so far (WMCRM.accdb) as it exists after working through all the steps in this section’s “The Access Workshop” (If you haven’t completed these actions, do so now). Copy the WMCRM.accdb database and rename the copy as WMCRM-AW05-v03.accdb (NOTE: Copying an Access database is discussed in Section 4 of “The Access Workbench”).

C. Modify the WMCRM-AW05-v03.accdb to implement the new relationship between VEHICLE and SALESPERSON.