Access: Relationships

Table Relationships
In a relational database, information about a particular subject is stored in its own table. The purpose of this is so that you do not need to store redundant data. For example, if you have a database with information about students and classes you would want to store the information about classes separately so that you do not have to enter all the details about a particular class such as time, place, instructor, credits, etc. over again for every student enrolled in that class.

A relationship works by matching data in key columns, usually columns with the same name in both tables. In most cases, the relationship matches the primary key from one table, which provides a unique identifier for each row, with the same type of field in the other table. The inclusion of this field in the second table is called a “foreign key”. A foreign key is one or more table fields (columns) that refer to the primary key field or fields in another table.

After the tables are created, relationships are defined by linking the related data together through the field(s) they have in common. In the example below, ID# in the Faculty table is the primary key for that table. In the Contracts Table, PI is essentially the same field as ID#, containing the same type of information. PI is a foreign key in the Contracts Table.

Note: Linking fields do not have to have the same name. They must be of the same data type and length for a relationship to be established, however.

<table>
<thead>
<tr>
<th>FACULTY</th>
<th>Name</th>
<th>Lname</th>
<th>Title</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>511-11-1234</td>
<td>Fred</td>
<td>Parson</td>
<td>Assoc. Prof</td>
<td>481-1111</td>
</tr>
<tr>
<td>522-22-2211</td>
<td>Julie</td>
<td>Brewer</td>
<td>Professor</td>
<td>491-2222</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONTRACTS</th>
<th>Sponsored by</th>
<th>Beg Date</th>
<th>End Date</th>
<th>Budget</th>
<th>Account</th>
<th>PI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSF</td>
<td>1/1/00</td>
<td>12/31/02</td>
<td>500000</td>
<td>00-9998</td>
<td>522-22-2211</td>
<td></td>
</tr>
<tr>
<td>USDA</td>
<td>3/1/01</td>
<td>3/1/02</td>
<td>100000</td>
<td>00-7823</td>
<td>522-22-2211</td>
<td></td>
</tr>
</tbody>
</table>

In order to match that information with related data stored in another table, you must define a relationship between the two tables. The definition of a relationship is: an association established between common fields (columns) in two tables. In order to eventually create a relationship, you must make sure that when you design your tables that you include a common field in both.

There are three types of relationships between tables. The type of relationship that is created depends on how the related columns are defined. A relationship can be one-to-one, one-to-many, or many-to-many.
Primary Key

Each table should have a field that constitutes a unique identifier. This field will be the table’s primary key. Advantages of assigning a primary key:

-- an index is automatically created for the primary key. An index speeds up queries and other operations;
-- when you view records in a form or datasheet, Access displays them in primary key order by default;
-- when you add data to your table, Access doesn't allow records with the same primary key value as another existing record;
-- you must specify a primary key if you want to create a default relationship between tables or if you want to update two or more related tables at the same time.

Types of Table Relationships

1) One-to-Many:

The most common type of relationship is the **one-to-many**. In a one-to-many relationship a record in Table A can have more than one matching record in Table B, but a record in table B has at most one matching record in Table A. An example of this would be Orders and Employees. A single employee may have sold more than one order, but a particular customer’s order was sold by only one employee. This type of relationship is created by including the primary key field from the table on the one side as an additional field (foreign key) in the table on the many side.

2) One-to-One:

In a **one-to-one** relationship, a row in Table A can have no more than one matching row in Table B, and vice versa. A one-to-one relationship is created if both of the related columns are primary keys. This type of relationship is not common because information related in this way would usually be included in a single table.

Examples of the use of a **one-to-one** relationship would be to divide information for security reasons, to divide a table with many columns, store data that is short-lived or data that applies only to a subset of the primary table. For example you might have one table containing general, non-confidential information and another containing the employee’s personal information. The general table could be accessible to all employees and the other could be secured to only those in Human Resources. Usually tables in a one-to-one relationship have the same field as a primary key in both.
3) **Many-to-Many:**

In a **many-to-many** relationship, a row in table A may have many matching rows in table B, and vice versa. An example of a many to many situation would be students and classes. One class has many students in it and likewise one student can have many classes.

A **many-to-many** relationship between tables is only possible by creating a third table (called a junction table) whose primary key consists of two fields – the keys from both Tables A and B. To accommodate this situation, create a junction table containing the primary key from Table A and the primary key from Table B. Other pertinent fields may also be included in this table. After you have created the junction table you will then set the relationships by creating a one-to-many relationship between Table A and the junction table and between Table B and the junction table.

**Defining Relationships Between Tables**

When creating relationships between tables, the related fields do not have to have the same names. However, related fields must have the same data type. An exception is relating an AutoNumber field from a table with another field. An AutoNumber field may be matched with a Number field if the **Field Size** property of both fields is the same. For example, you can match an AutoNumber field and a Number field if the **Field Size** property of both fields is Long Integer. Even when both matching fields are Number fields, they must have the same **Field Size** property setting.

**Creating Relationships**

To create a relationship between two tables:

1. Click the Relationships button on the Database Tools tab.
2. Drag the field that you want to relate from one table to the related field in the other table. 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.
3. The Edit Relationships dialog box is displayed. Check the field names displayed in the two columns to ensure they are correct. If they are not correct, you can change them here.
4. Set the relationship options if desired. For information about a specific item in the Edit Relationships dialog box, click the question mark button, and then click the item.
5. Click the Create button to create the relationship. When you close the Relationships window, Microsoft Access asks if you want to save the layout. Whether you save the layout or not, the relationships you create are saved in the database.

Referential Integrity
Referential Integrity refers to rules that you can enforce to prevent the accidental changing or deletion of data in related tables. This helps ensure that your data is not out of sync and that you do not end up with “orphan” records, i.e., a record in one table that has lost its related information from another table. Referential integrity also helps ensure that information in one table matches information in another. For example, each order in the Orders table must be associated with a specific employee in the Employee table.

Cascade Update and Cascade Delete
For relationships in which referential integrity is enforced, you can specify whether you want Microsoft Access to automatically cascade update or cascade delete related records.

If you click to select the Cascade Update Related Fields check box when you define a relationship, any time that you change the primary key of a record in the primary table, Microsoft Access automatically updates the primary key to the new value in all related records. For example, if you change a customer's ID in the Customers table, the CustomerID field in the Orders table is automatically updated for every one of that customer's orders so that the relationship is not broken. Microsoft Access cascades updates without displaying any message.

NOTE: If the primary key in the primary table 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.

If you select the Cascade Delete Related Records check box when you define a relationship, any time that you delete records in the primary table, Microsoft Access automatically deletes related records in the related table. For example, if you delete a customer record from the Customers table, all the customer's orders are automatically deleted from the Orders table (this includes records in the Order Details table related to the Orders records). When you delete records from a form or datasheet with the Cascade Delete Related Records check box selected, Microsoft Access warns you that related records may also be deleted. However, when you delete records using a delete query, Microsoft Access automatically deletes the records in related tables without displaying a warning.

Editing a Relationship
You can edit a relationship at any time and change the referential integrity rules by opening the relationships grid and double-clicking on the relationship line between two tables. Any changes you make to the relationship will be automatically saved as soon as you complete the change.

Deleting a Relationship
You may wish to delete a relationship. Reasons for this could be because you need to make changes to a related table and Access is preventing you from doing so because of the relationship between that table and another. To delete a relationship, open the Relationships window, click once on the relationships line and press the Delete key on the keyboard.