Microsoft Access
Understanding Relationships

Academic Health Center Training
training@health.ufl.edu
(352) 273-5051

What is a **Relational Database**?

• A relational database is a collection of tables from which data can be accessed in many different ways without having to reorganize the database tables.
  - That is, the tables can “talk” to each other. We can link (relate) our tables to find:
    - Which doctors are seeing a patient
    - Which students are in which class
    - Which item is selling the most on Friday’s

What is a **Relational Database**?

• A relational database allows data structures, storage and retrieval operations, and integrity constraints.
  - Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency
Review of the Basic Design Rules

- **Organizing Data**
  - Data is split between tables to prevent data duplication and entry repetition

- **No Derived Fields**
  - Only the linking fields, the primary keys, should be in more than one table. If any other field can be found in another table, it should not be repeated.

- **Data is broken down into Smallest Logical Parts**
  - Smallest “Sortable” parts. Remember it’s much easier to pull fields together than it is to pull a field apart.

---

Review of the Basic Design Rules

- **Descriptive Field Names**
  - Name your fields (columns) as clear as possible in each table. Be Clear, Be Concise and Be Consistent.

- **Unique Field Names**
  - Keep fields unique across tables, and keep them as clear as possible in each table.

- **Unique Records**
  - Each of your tables should have unique records. We ensure this by setting one field to be a Primary Key.

---

Table of Yards

<table>
<thead>
<tr>
<th>Yards</th>
<th>Yard Number</th>
<th>Owner</th>
<th>Address</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table of Birdfeeders

<table>
<thead>
<tr>
<th>Bird Feeders</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird Feeder Number</td>
<td>Material (wood, plastic...)</td>
</tr>
<tr>
<td></td>
<td>Location in Yard</td>
</tr>
</tbody>
</table>

Relating Yards and Birdfeeders

<table>
<thead>
<tr>
<th>Bird Feeders</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird Feeder Number</td>
<td>Material</td>
</tr>
<tr>
<td></td>
<td>Location in Yard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Yards</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Yard Number</td>
<td>Owner</td>
</tr>
<tr>
<td></td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td>Phone Number</td>
</tr>
</tbody>
</table>

There must be one field in both tables that is the same, so that the database knows how the tables connect. It’s best to use the Primary key as the link.

If we put the Birdfeeder in the Yard table, we will have to count each one. BF1, BF2...

Whenever you find yourself numbering the fields in this way, it’s a sign you’re on the wrong track.
Relating Yards and Birdfeeders

But if we reverse the direction, the link (relationship) makes more sense.

Each Birdfeeder can only be in one Yard, but each Yard can have many Birdfeeders. This is called a one to many (1 - ∞) Relationship.

One to Many Relationships

• One to Many relationships are the most common relationships.
  • One Birdfeeder is visited by Many Birds
  • One Yard contains Many Birdfeeders
  • One Patient has Many Prescriptions
  • One Insurance has Many Patients
  • One Student attends Many Classes

• One to Many includes One to None.

• A record MUST be in the One (primary) table in order to appear in the Many table.

Primary Key linked to Non Primary Key
One to One Relationships

- One to One relationships can often combine the data into one table.
  - One Birdfeeder is located in One place in the Yard
  - One Yard has One Address
  - One Patient has One Home Phone Number
  - One Insurance has One Contact Person
  - One Student has One Gatorlink ID

- Access determines the “primary” table based on the direction you create your relationship.

---

One to One Relationships

Primary Key linked to Primary Key

<table>
<thead>
<tr>
<th>Patients</th>
<th>Patient Contact Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pt Medical Record #</td>
<td>Pt Medical Record #</td>
</tr>
<tr>
<td>Pt Last Name</td>
<td>Pt Email Address</td>
</tr>
<tr>
<td>Pt First Name</td>
<td></td>
</tr>
<tr>
<td>Pt Birth Date</td>
<td></td>
</tr>
</tbody>
</table>

---

One to One Relationships

Reasons you may use a One to One…

- You have more than 255 fields
  - the maximum number of columns (fields)
- You have a large set of related data that doesn’t need to be accessed every time you look up that item
  - medical history, map of the location, transcript from previous school
Many to Many Relationships

- Many to Many relationships are very common.
  - Many Students are taught by Many Teachers
  - Many Patients see Many Doctors
  - Many Medications are taken by Many Patients
  - Many Customers buy Many Products

- You cannot create a “true” relationship between these tables because there can be no uniqueness in either side of the relationship.

---

Many to Many Relationships

Non Primary Key linked to Non Primary Key

<table>
<thead>
<tr>
<th>Appointments</th>
<th>Prescriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>App ID Number</td>
<td>Prescription Number</td>
</tr>
<tr>
<td>Pt Medical Record #</td>
<td>Pt Medical Record #</td>
</tr>
</tbody>
</table>

Access sees this as an Indeterminate relationship

You cannot Enforce Referential Integrity

---

Many to Many Relationships (indeterminate)

- Indeterminate relationships are often found when we are linking tables, because Linked Tables cannot have a primary key.

- If you have imported or created a table, it’s very rare to have a need for an indeterminate Relationship.

- These relationships show Access that the data saved in the field from the first table is the same kind of data saved in the second table, but there can be no data integrity rules applied on indeterminate relationships.
Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

Jack bought:
2 Hats
4 Pairs of Socks
3 Pairs of Shoes

Shoes bought by:
Jack Johnson
Jill Jones
Jerry Jacks

Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

Many to Many Relationship

Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

Customer1
Customer2
Customer3
......

One Product can be purchased by an unlimited number of Customers.
Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip
Product1
Product2
Product3
......

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

One Customer can purchase an unlimited number of Products.

Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

One Product can be sold many times

Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

SALES
Sales ID
Customer ID
Product ID
Date
Quantity

One Customer can have many sales

Many to Many Relationships Example

CUSTOMERS
Customer ID
First
Last
Address
City
State
Zip

PRODUCTS
Product ID
Product
Supplier
Description
Units
Cost
Price

SALES
Sales ID
Customer ID
Product ID
Date
Quantity

One Product can be sold many times
Many to Many Relationships Example

Products by Customer

First ____________ Last __________________
Address ______________________________________
City_____________ State ____ Zip Code ________

<table>
<thead>
<tr>
<th>Product</th>
<th>Date</th>
<th>Qty</th>
</tr>
</thead>
</table>

Many to Many Relationships Example

Customers by Products

Product __________________________________
Supplier__________________________________
Description _______________________________
Units_________ Cost______ Price________

<table>
<thead>
<tr>
<th>Customer</th>
<th>Date</th>
<th>Qty</th>
</tr>
</thead>
</table>

Relationship?

<table>
<thead>
<tr>
<th>Patients</th>
<th>Medications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient ID</td>
<td>Med ID</td>
</tr>
<tr>
<td>First</td>
<td>Medication</td>
</tr>
<tr>
<td>Last</td>
<td>Description</td>
</tr>
<tr>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td></td>
</tr>
<tr>
<td>Zip</td>
<td></td>
</tr>
</tbody>
</table>
Patients

Patient ID
First
Last
Address
City
State
Zip

Medications

Med ID
Medication Description

Patient Meds

PM ID
Patient ID
Med ID
Dosage
Directions

One Patient can take many Medications.

One Kind of Medication can be taken by Many Patients.

Relationships – Many to Many

Primary Docs

Doc ID
Name
Phone
Pager
email

One Patient will have only ONE primary Doctor.
One Primary Doctor can have MANY patients.

Relationship – One to Many

Relationship?

Relationship?
Relationship?

Patients
- Patient ID
- First
- Last
- Address
- City
- State
- Zip

Med History
- Patient ID
- Health Q1
- Health Q2
- Health Q3
- Health Q4

Relationship – One to One

Patients
- Patient ID
- First
- Last
- Address
- City
- State
- Zip

Med History
- Patient ID
- Health Q1
- Health Q2
- Health Q3
- Health Q4

One Patient will have only ONE Medical History. Each Medical History will belong to only ONE patient.

Let’s Practice...