COMP1111

Week 4

Defining Table Relationships

- Today we will be covering table relationships, and what they are.
- There are 3 types of table relationships in Microsoft Access
    - One-to-many
    - Many-to-one
    - One-to-one

Defining Table Relationships

- In a relational database, relationships enable you to prevent redundant data.
- For example, if you are designing a database that will track information about books, you might have a table named "Titles" that stores information about each book, such as the book’s title, date of publication, and publisher.
- There is also information that you might want to store about the publisher, such as the publisher's telephone number, address, and ZIP Code/Postal Code.
- If you were to store all this information in the "Titles" table, the publisher’s telephone number would be duplicated for each title that the publisher prints.

Defining Table Relationships

- A better solution is to store the publisher's information only one time, in a separate table that we will call "Publishers."
- You would then put a pointer in the "Titles" table that references an entry in the "Publishers" table.
- To make sure that your data stays synchronized, you can enforce referential integrity between tables.
- Referential integrity relationships help make sure that information in one table matches information in another.
- For example, each title in the "Titles" table must be associated with a specific publisher in the "Publishers" table. A title cannot be added to the database for a publisher that does not exist in the database.
- Logical relationships in a database enable you to efficiently query data and create reports.

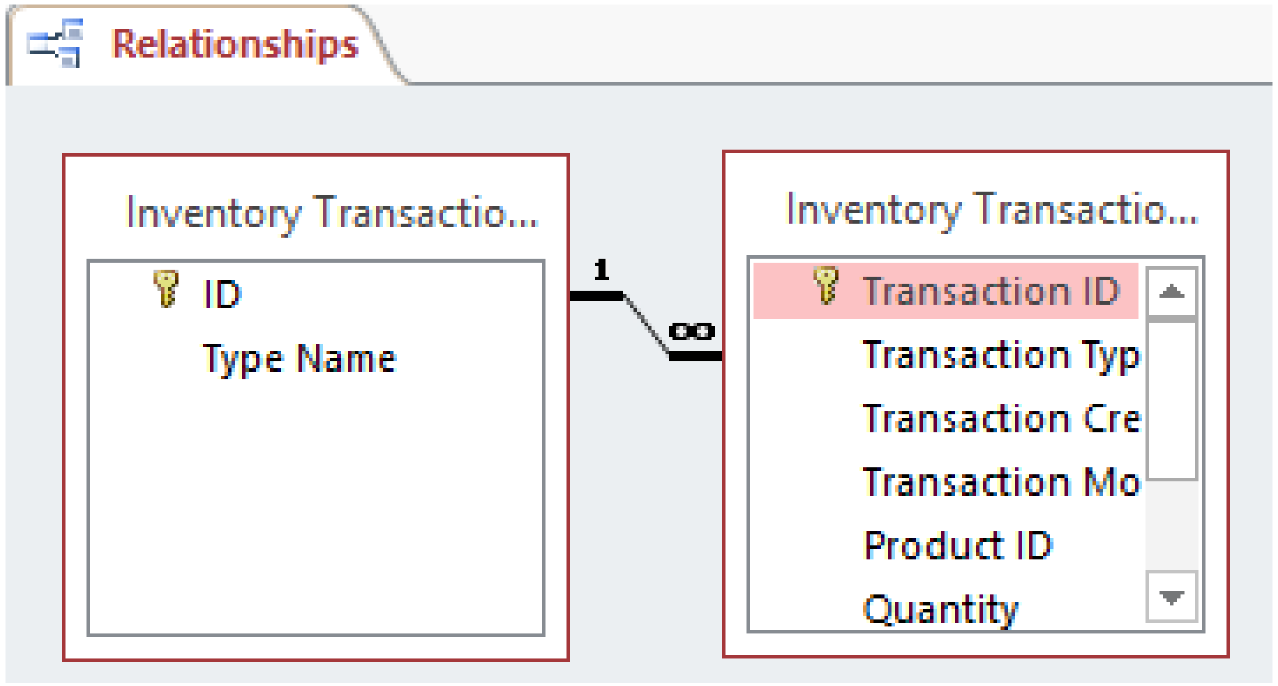
Table Relationships

- A relationship works by matching data in key columns, usually columns (or fields) that have the same name in both tables.
- In most cases, the relationship connects the primary key, or the unique identifier column for each row, from one table to a field in another table.
- The column in the other table is known as the "foreign key”.
- For example, if you want to track sales of each book title, you create a relationship between the primary key column (let's call it **title_ID**) in the "Titles" table and a column in the "Sales" table that is named **title_ID**.
- The **title_ID** column in the "Sales" table is the foreign key.
- There are three kinds of relationships between tables. The kind of relationship that is created depends on how the related columns are defined.

One-to-many Relationships

- A one-to-many relationship is the most common kind of relationship.
- In this kind of relationship, a row in table A can have many matching rows in table B.
- But a row in table B can have only one matching row in table A.
- For example, the "Publishers" and "Titles" tables have a one-to-many relationship. That is, each publisher produces many titles. But each title comes from only one publisher.
- A one-to-many relationship is created if only one of the related columns is a primary key or has a unique constraint.
- In the relationship window in Access, the primary key side of a one-to-many relationship is denoted by a number 1. The foreign key side of a relationship is denoted by an infinity symbol.

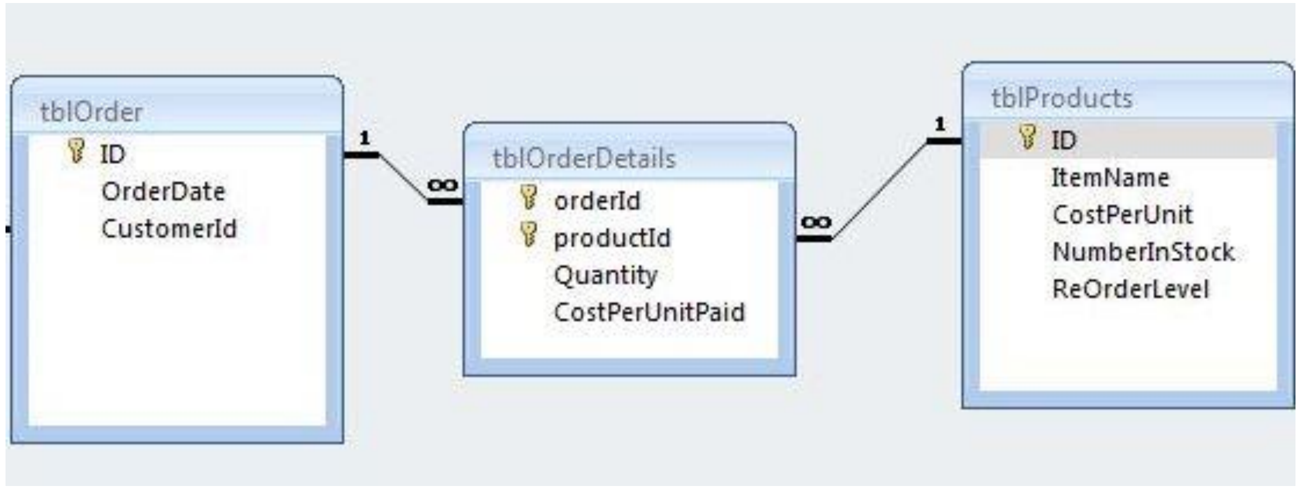
One-to-many Relationships



Many-to-many Relationships

- In a many-to-many relationship, a row in table A can have many matching rows in table B, and vice versa.
- You create such a relationship by defining a third table that is called a junction table.
- The primary key of the junction table consists of the foreign keys from both table A and table B.
- For example, the “Order" table and the “Products" table have a many-to-many relationship that is defined by a one-to-many relationship from each of these tables to the “OrderDetails" table.
- The primary key of the “orderDetails" table is the combination of the **ord_ID** column (the “Order" table’s primary key) and the **prod_ID** column (the “Products" table’s primary key).

Many-to-many Relationships



Referential Integrity

- Referential integrity means that the foreign key in any referencing table must always refer to a valid row in the referenced table.
- Referential integrity ensures that the relationship between two tables remains synchronized
- While creating Relationships, you could also enable Cascade Update, and Cascade Delete.

What Are Queries

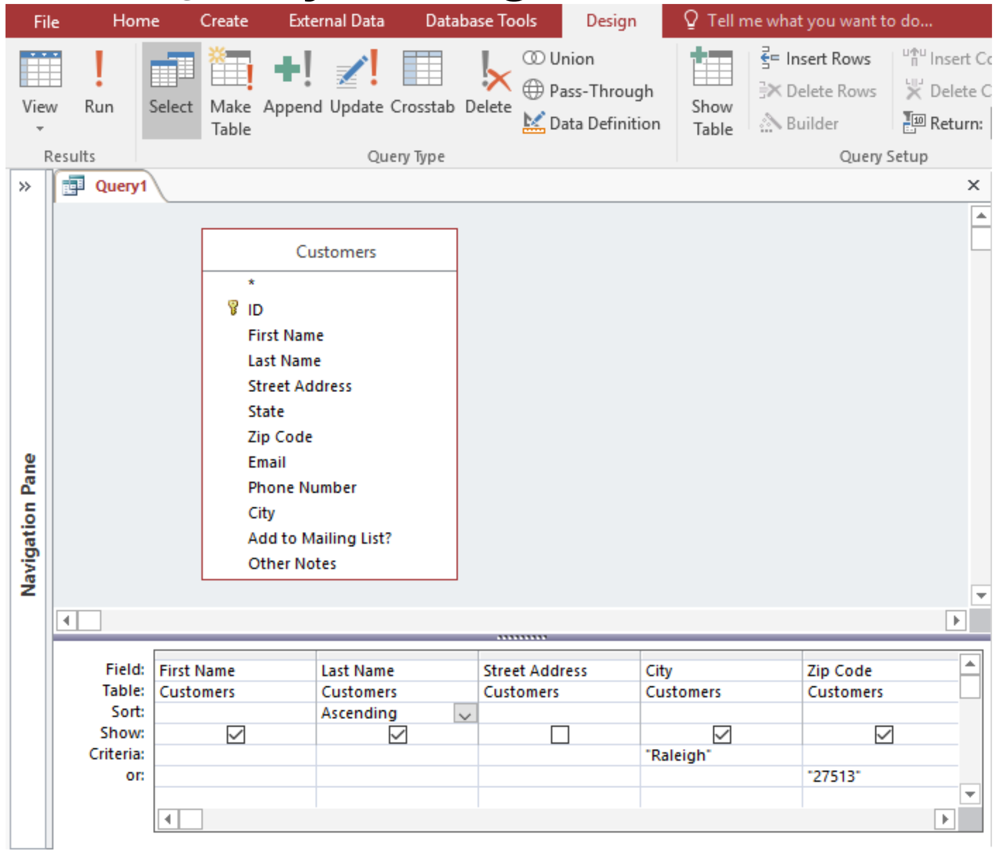
- Queries are a way of **searching** for and **compiling** data from one or more tables.
- Running a query is like asking a **detailed question** of your database.
- When you build a query in Access, you are **defining specific search conditions** to find exactly the data you want.

How Are Queries Used

- Queries are far more powerful than the simple searches or filters you might use to find data within a table.
- This is because queries can draw their information from **multiple** tables.
- For example, while you could use a **search** in the customers table to find the name of one customer at your business or a **filter** on the orders table to view only orders placed within the past week, neither would let you view both customers and orders at once.
- However, you could easily run a **query** to find the name and phone number of every customer who's made a purchase within the past week.
- A well-designed query can give information you might not be able to find out just by examining the data in your tables.

Query Design View

- We will be using Query Design View to create our queries.
- Click on Create -> Query Design



Field:	First Name	Last Name	Street Address	City	Zip Code
Table:	Customers	Customers	Customers	Customers	Customers
Sort:		Ascending			
Show:	☑	☑	☐	☑	☑
Criteria:				"Raleigh"	
or:					"27513"

Wildcards

- Wildcards can be used when you run select, update, and delete queries.
  - Asterisk *
    - Matches any number of characters, and can be used anywhere in a character string
      - Wh* finds what, white, and why, but not a awhile or watch
  - Question Mark ?
    - Matches any single alphanumeric character
      - B?ll finds ball, bell, and bill
  - Hashtag #
    - Matches any single numeric character
      - 1#3 finds 103, 113, and 123

Greater Than / Less Than

- Can be used in select, update, and delete queries
- Used to specify values that are greater/less than a specific value
  - Greater than >
    - Shows values that are greater than value on the right side
      - > 100 will return values that are greater than 100 (101, 110)
      - >= 100 will return values that are greater than or equal to 100 (100, 101, 110)
  - Less than <
    - Shows values that are less than value on the right side
      - < 100 will return values that are less than 100 (99, 90, 82)
      - <= 100 will return values that are less than or equal to 100 (100, 99, 90)

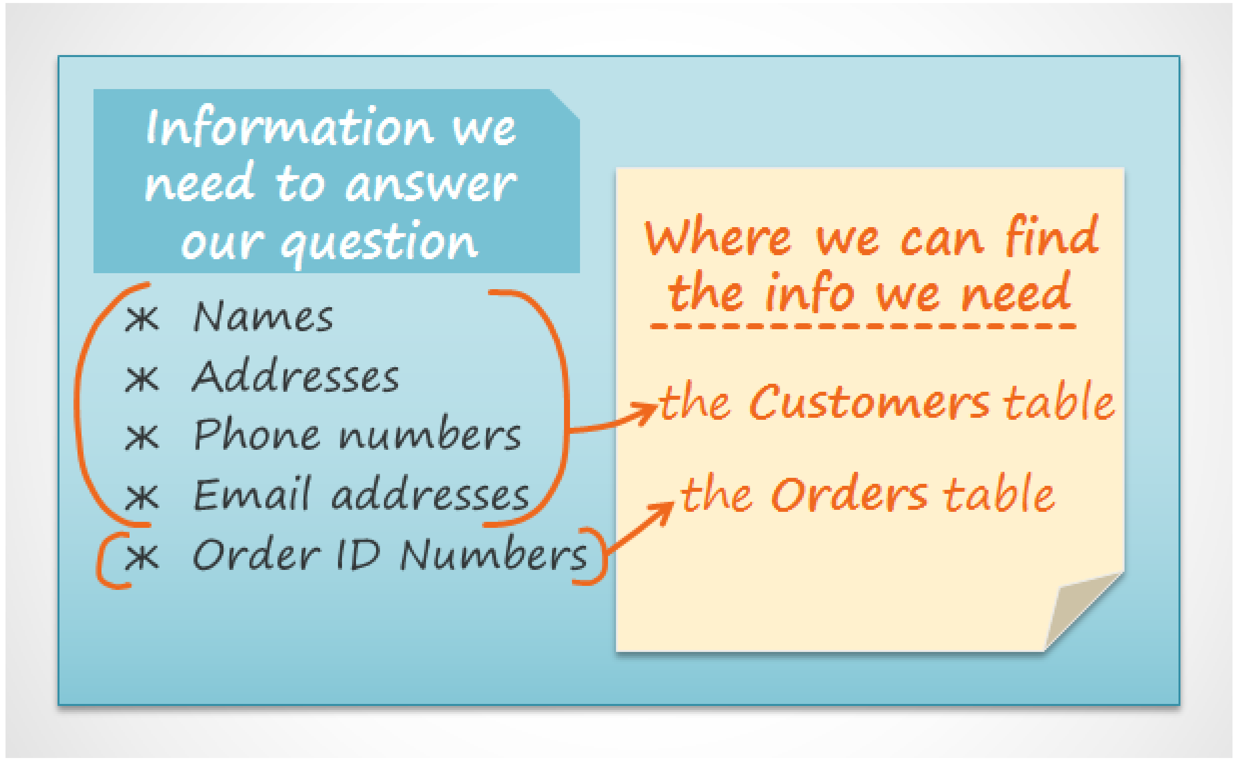
Multi-Table Queries

- Most queries you design in Access will likely use **multiple tables**, allowing you to answer more complex questions.
- Queries can be difficult to understand and build if you don't have a good idea of what you're trying to find and how to find it.
- A one-table query can be simple enough to make up as you go along, but to build anything more powerful you'll need to plan the query in advance.

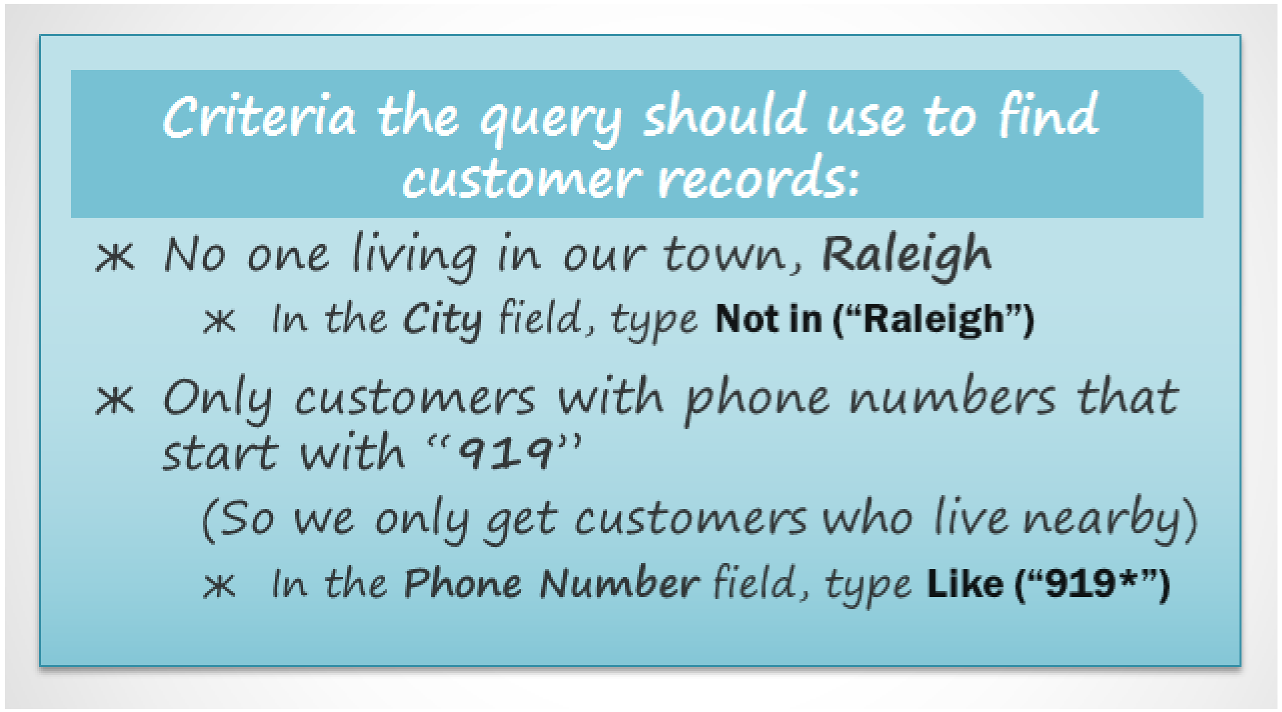
Planning A Query

- **Pinpoint** exactly what you want to know. If you could ask your database any question, what would it be? Building a query is more complicated than just asking a question, but knowing precisely what question you want to answer is essential to building a useful query.
- **Identify** every type of information you want included in your query results. Which fields contain this information?
- **Locate** the fields you want to include in your query. Which tables are they contained in?
- **Determine** the criteria the information in each field needs to meet. Think about the question you asked in the first step. Which fields do you need to search for specific information? What information are you looking for? How will you search for it?

Planning A Query



Planning A Query



Criteria the query should use to find customer records:

- No one living in our town, **Raleigh**
  - In the **City** field, type **Not in ("Raleigh")**
- Only customers with phone numbers that start with "919"

  (So we only get customers who live nearby)
  - In the **Phone Number** field, type **Like ("919*")**

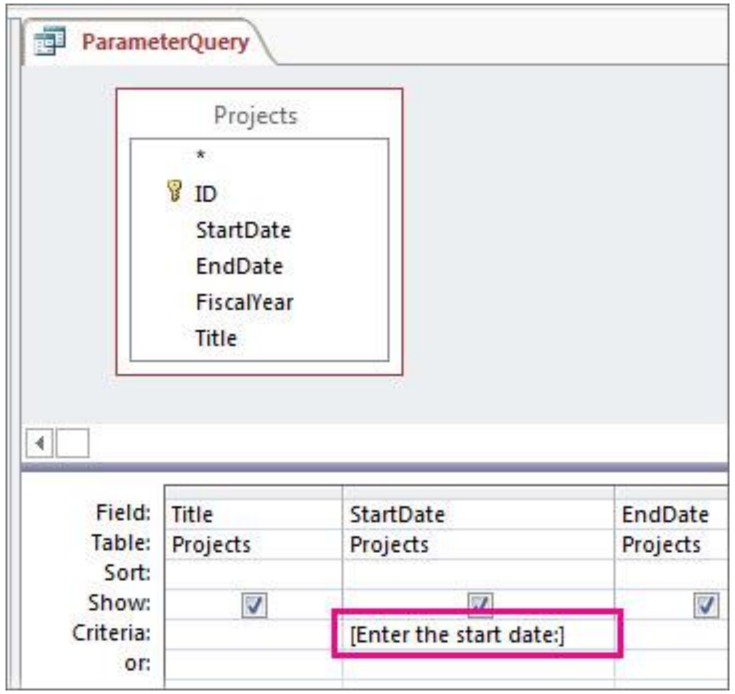
Create a Multi-Table Query

- Select the **Query Design** command from the **Create** tab on the Ribbon.
- In the dialog box that appears, select each table you want to include in your query and click **Add**.
- You can press and hold the **Ctrl** key on your keyboard to select more than one table. When we planned our query, we decided we needed information from the **Customers** and **Orders** tables, so we'll add these.
- Set field **criteria** by entering the desired criteria in the criteria row of each field. We want to set two criteria: **Not in ("Raleigh")** in the **City** field, and **Like ("919*")** in the **Phone Number** field. This will find customers who do not live in Raleigh but who do live in the 919 area code.
- After you have set your criteria, **run** the query by clicking the **Run** command on the **Design** tab.
- If you want, **save** your query by clicking the **Save** command in the Quick Access Toolbar. When prompted to name it, type the desired name, then click **OK**.

Create a Multi-Table Query

- Creating a parameter is similar to adding a normal criterion to a query.
- Create a select query, and then open the query in Design view.
- In the **Criteria** row of the field you want to apply a parameter to, enter the text that you want to display in the parameter box, enclosed in square brackets. For example, **[Enter the start date:]**

Table Query Parameter


ParameterQuery

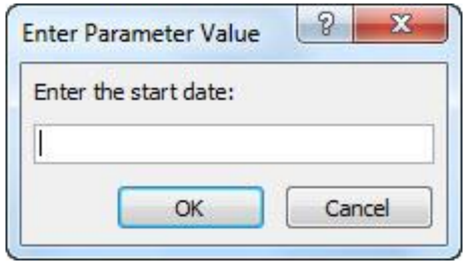
Projects

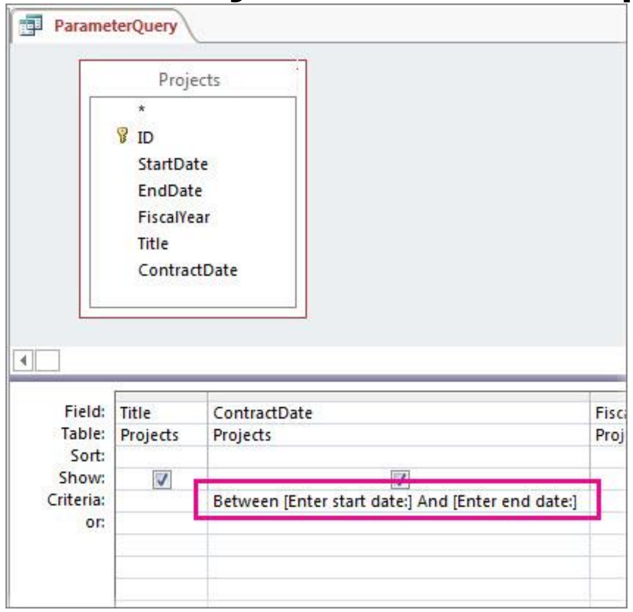
- *
- ID
- StartDate
- EndDate
- FiscalYear
- Title

Field:	Title	StartDate	EndDate
Table:	Projects	Projects	Projects
Sort:			
Show:	✓	✓	✓
Criteria:		[Enter the start date:]	
or:			

Table Query Parameter

- When you run the query, the prompt appears without the square brackets.
- You can use multiple parameters in a criterion. For example, **Between [Enter the start date:] And [Enter the end date:]** will generate two prompts when you run the query.



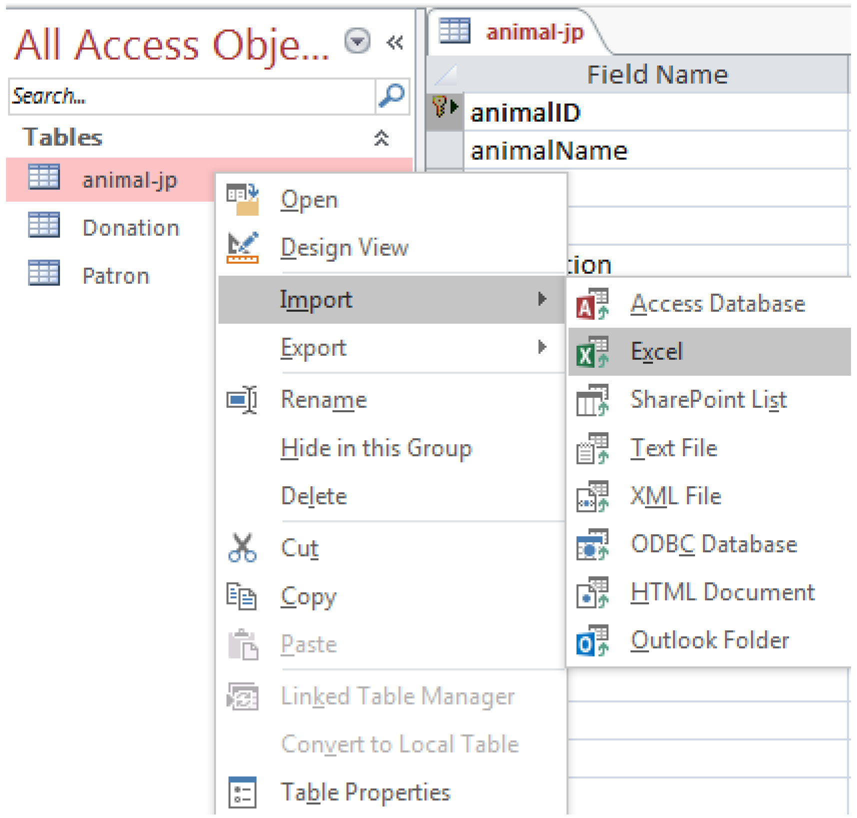


Between [Enter start date:] And [Enter end date:]

Importing Table Data

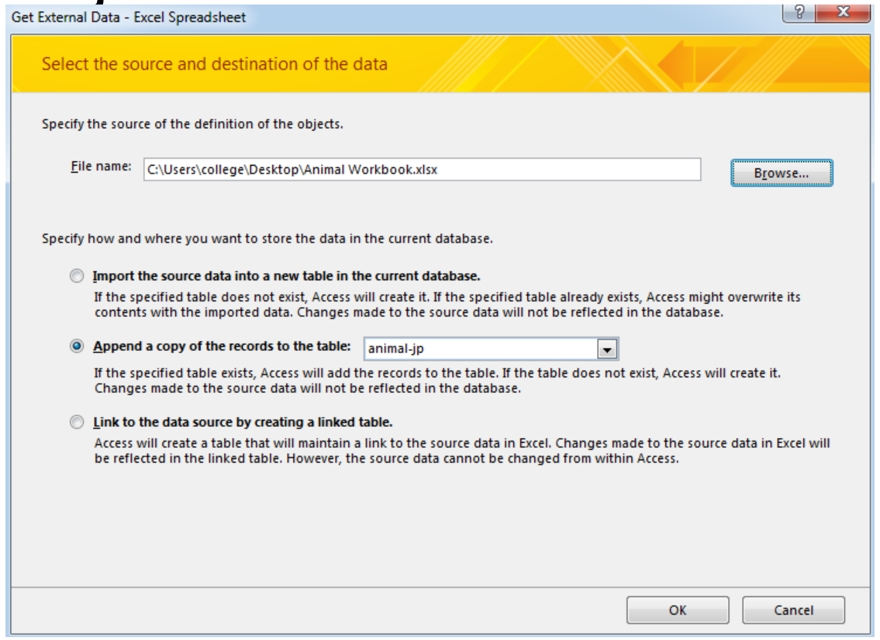
- Create a new table, using Table Design view.
- The fields in your table should match the columns from your Excel file, which you will be using to import.
- Be sure to set a primary key, in our case it will be animalID
- Once the table has been saved, you can then import the data by right clicking on the table, and selecting import and then excel.

Importing Table Data



Importing Table Data

- Browse for your Excel file
- Select append a copy to the table
- Select the table we just created


Get External Data - Excel Spreadsheet

Select the source and destination of the data

Specify the source of the definition of the objects.

File name: C:\Users\college\Desktop\Animal Workbook.xlsx Browse...

Specify how and where you want to store the data in the current database.

○ **Import the source data into a new table in the current database.**
If the specified table does not exist, Access will create it. If the specified table already exists, Access might overwrite its contents with the imported data. Changes made to the source data will not be reflected in the database.

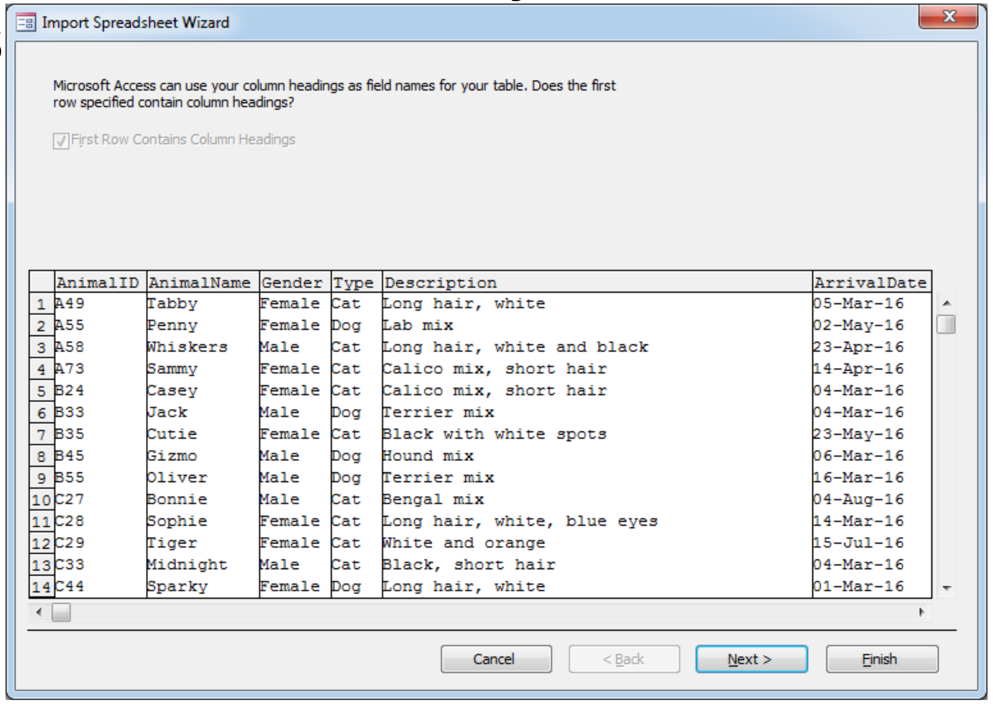
● **Append a copy of the records to the table:** animal-jp
If the specified table exists, Access will add the records to the table. If the table does not exist, Access will create it. Changes made to the source data will not be reflected in the database.

○ **Link to the data source by creating a linked table.**
Access will create a table that will maintain a link to the source data in Excel. Changes made to the source data in Excel will be reflected in the linked table. However, the source data cannot be changed from within Access.

OK Cancel

Importing Table Data

- On the next page, the import spreadsheet wizard will appear
- This screen will show you some of the data, and you can confirm everything is in the correct fields



Microsoft Access can use your column headings as field names for your table. Does the first row specified contain column headings?

☑ First Row Contains Column Headings

	AnimalID	AnimalName	Gender	Type	Description	ArrivalDate
1	A49	Tabby	Female	Cat	Long hair, white	05-Mar-16
2	A55	Penny	Female	Dog	Lab mix	02-May-16
3	A58	Whiskers	Male	Cat	Long hair, white and black	23-Apr-16
4	A73	Sammy	Female	Cat	Calico mix, short hair	14-Apr-16
5	B24	Casey	Female	Cat	Calico mix, short hair	04-Mar-16
6	B33	Jack	Male	Dog	Terrier mix	04-Mar-16
7	B35	Cutie	Female	Cat	Black with white spots	23-May-16
8	B45	Gizmo	Male	Dog	Hound mix	06-Mar-16
9	B55	Oliver	Male	Dog	Terrier mix	16-Mar-16
10	C27	Bonnie	Male	Cat	Bengal mix	04-Aug-16
11	C28	Sophie	Female	Cat	Long hair, white, blue eyes	14-Mar-16
12	C29	Tiger	Female	Cat	White and orange	15-Jul-16
13	C33	Midnight	Male	Cat	Black, short hair	04-Mar-16
14	C44	Sparky	Female	Dog	Long hair, white	01-Mar-16

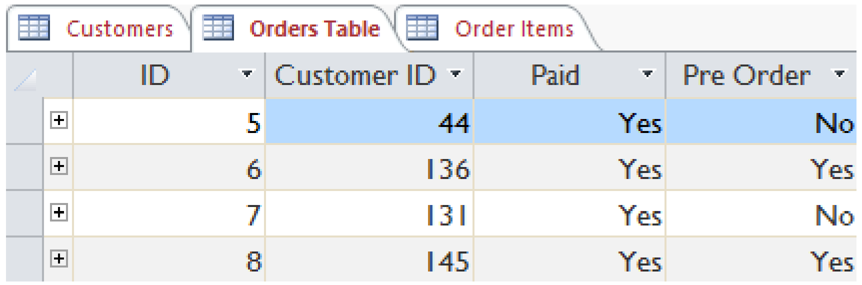
Working With Forms

- While you can always enter data directly into database tables, you might find it easier to use **forms**.
- Forms ensure you're entering the right data in the right location and format. This can help keep your database accurate and consistent.

Working With Forms

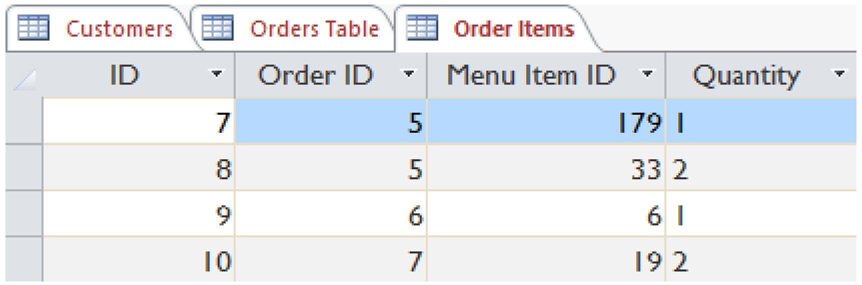
- The use of forms ensures that all the data entered goes exactly where it’s supposed to go: into one or more related tables.
- While entering data into simple tables is fairly straightforward, data entry becomes more complicated as you start populating tables with records from elsewhere in the database.
- For instance, the **orders table** in a bakery's database might link to information on customers, products, and prices drawn from related tables.
- For example, in the Orders Table below the Customer ID field is linked to the Customers table.

Working With Forms



ID	Customer ID	Paid	Pre Order
5	44	Yes	No
6	136	Yes	Yes
7	131	Yes	No
8	145	Yes	Yes

- In fact, in order to see the entire order you would also have to look at the **Order Items table**, where the menu items that make up each order are recorded.

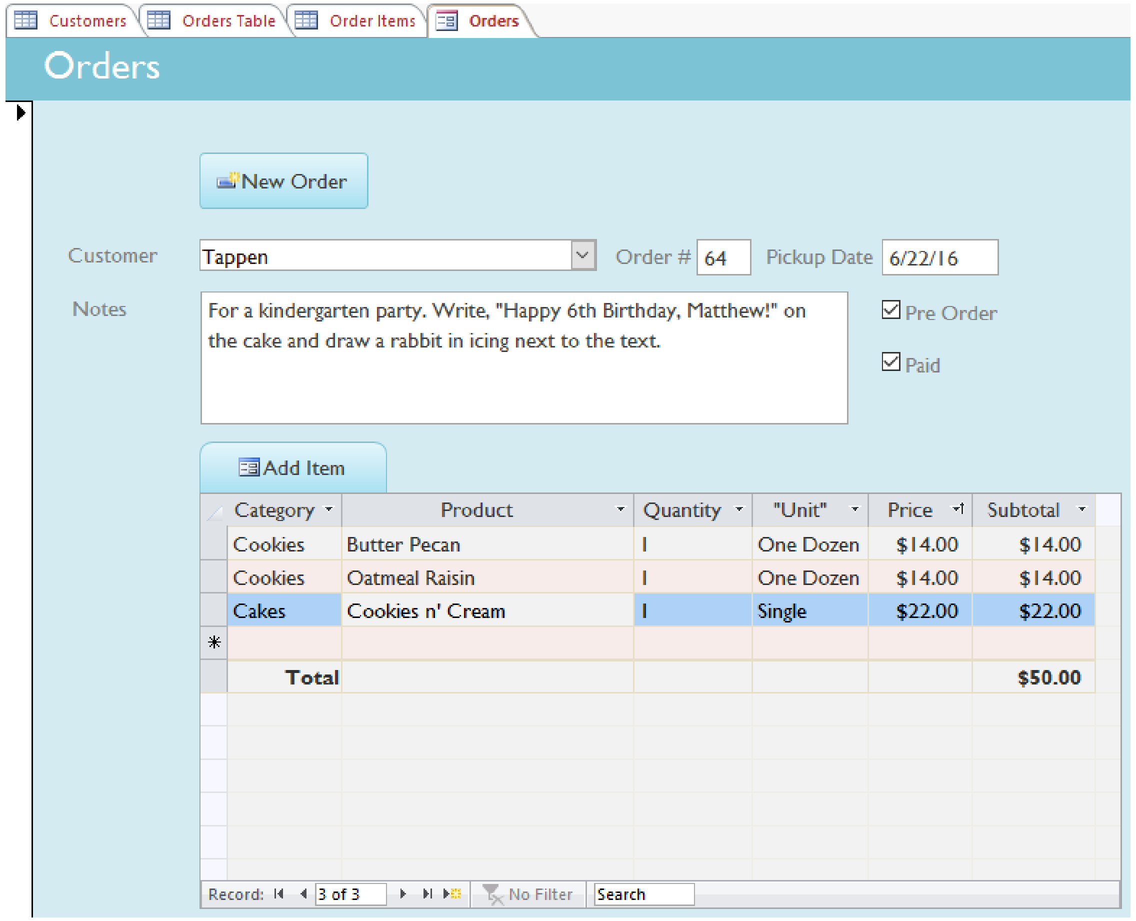


ID	Order ID	Menu Item ID	Quantity
7	5	179	1
8	5	33	2
9	6	6	1
10	7	19	2

Working With Forms

- The records in these tables include **ID numbers** of records from other tables.
- You can't learn much just by glancing at these records because the ID numbers don't tell you much about the data they relate to.
- Plus, because you have to look at two tables just to view one order, you might have a difficult time even finding the right data.
- It's easy to see how viewing or entering many records this way could become a difficult and tedious task.

Working With Forms



Orders

Customer: Tappen

Order #: 64

Pickup Date: 6/22/16

Notes: For a kindergarten party. Write, "Happy 6th Birthday, Matthew!" on the cake and draw a rabbit in icing next to the text.

☑ Pre Order

☑ Paid

Category	Product	Quantity	"Unit"	Price	Subtotal
Cookies	Butter Pecan	1	One Dozen	$14.00	$14.00
Cookies	Oatmeal Raisin	1	One Dozen	$14.00	$14.00
Cakes	Cookies n' Cream	1	Single	$22.00	$22.00
Total					**$50.00**

Record: 3 of 3 No Filter Search

GCFLearnFree

- Information from http://www.gcflearnfree.org/access2016 and other web resources.