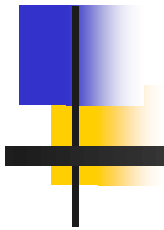
Introduction to Relationships in Databases

IOE 373 Lecture 02

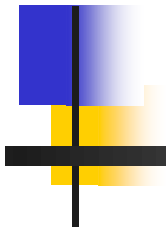


Two Most-Common Types

- One-to-Many
- Many-to-Many

One-to-Many

- A one-to-many relationship between two tables A and B means that
  - one record in table A can be related to many records in table B, but
  - Each record in table B can only be related to one record in table A.



Example 1

- Cars and Manufacturers
  - Every car model is made by one manufacturer:
    - Civic—Honda
    - Focus—Ford
    - Sonata—Hyundai
  - However, most manufacturers make more than one car model:
    - Honda: Civic, Accord, Fit, etc.
    - Ford: Focus, Fiesta, Mustang, Fusion, etc.

Partial Tables

Design Views

Models

Field Name	Data Type
ModelID	Number
ModelName	Text
Classification	Text

Manufacturers

Field Name	Data Type
ManufacturerID	Number
ManufacturerName	Text
ManufacturerCountry	Text

Datasheet Views

Models

ModelID	ModelName	Classification
1	Civic	Compact
2	Accord	Midsize
3	Fit	Subcompact
4	Focus	Compact
5	Fiesta	Subcompact
6	Mustang	Sport

Manufacturers

ManufacturerID	ManufacturerName	ManufacturerCountry
1	Honda	Japan
2	Ford	USA
3	BMW	Germany
4	Hyundai	South Korea

So how do we link these tables to indicate which Manufacturer makes each model?

Linking the tables; one-to-many

We add a ManufacturerID field to the Models table:

Models

Field Name	Data Type
ModelID	Number
ModelName	Text
Classification	Text
ManufacturerID	Number

We then populate this field with the appropriate ManufacturerID for each model:

Models

ModelID	ModelName	Classification	ManufacturerID
1	Civic	Compact	1
2	Accord	Midsize	1
3	Fit	Subcompact	1
4	Focus	Compact	2
5	Fiesta	Subcompact	2
6	Mustang	Sport	2

Manufacturers

ManufacturerID	ManufacturerName	ManufacturerCountry
1	Honda	Japan
2	Ford	USA
3	BMW	Germany
4	Hyundai	South Korea

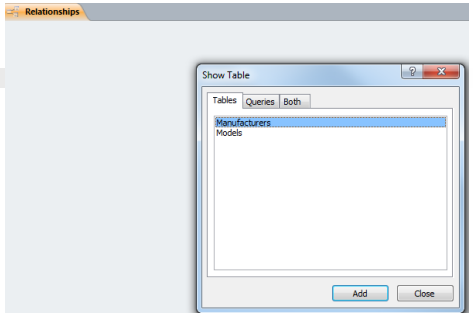
The ManufacturerID in the Models table is a *foreign key.*

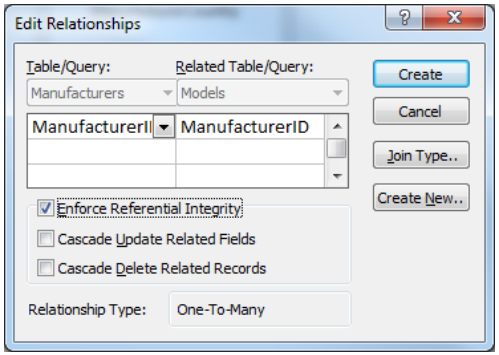
Definition

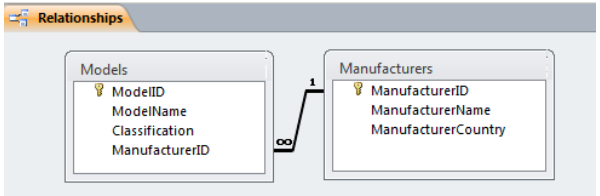
- Foreign Key: A field in one table which is not the primary key in that table but is the primary key of another table.
- Entering a value in the foreign key field indicates that the record in the first table is related to the record in the second table having that primary key value.

Creating the Relationship

- In Access, open the Relationship designer (Database Tools/Relationships)
- Select the tables that are involved in the relationship:
- Drag the foreign key field from one table to the other.
- Check the Referential Integrity box and click “Create”.







One-to-Many

- The relationship is now formalized in Access.
- Access will enforce the relationship—you will not be allowed to enter a ManufacturerID in the Models table which doesn't have a matching entry in the Manufacturers table.
- Therefore, you will probably want to populate the Manufacturers table before the Models table.

Representing One-to-Many Relationships - Recap

- A one-to-many relationship is expressed in Access and other DBMS's by using a *foreign key*.
- A foreign key is a field in a table which is not the primary key of that table, but is the primary key of another table.

Consider…

In the table below, what seems to be too simplistic?

Manufacturers

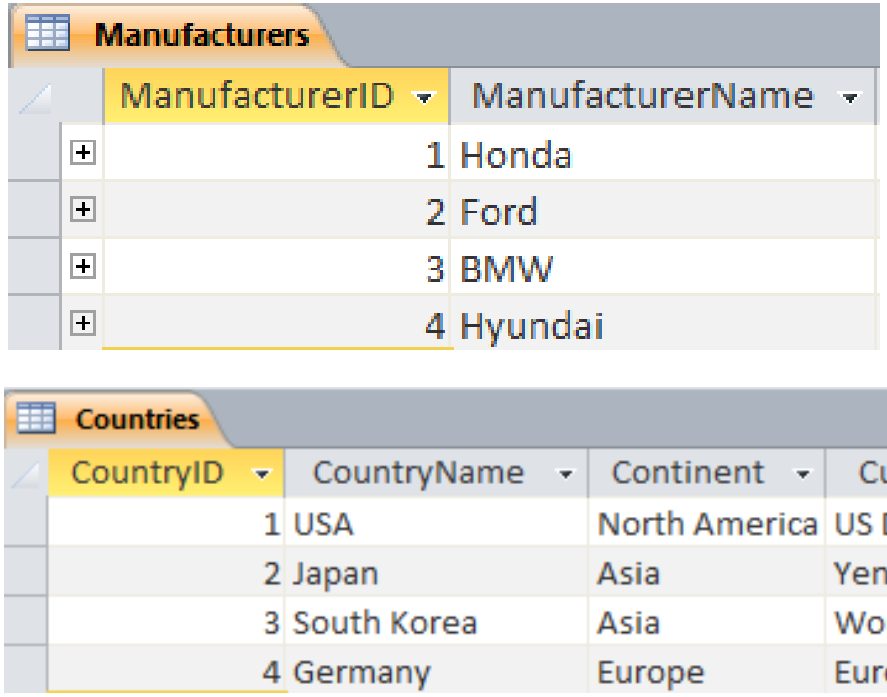
ManufacturerID	ManufacturerName	ManufacturerCountry
1	Honda	Japan
2	Ford	USA
3	BMW	Germany
4	Hyundai	South Korea

- Ignore that there aren't enough manufacturers listed.
- Ignore that there are a lot of other fields that we could add to this table.
- What is wrong with the ManufacturerCountry field?

There isn't just one country per company

- Most large car companies manufacture vehicles in multiple countries.
- In general, it isn't accurate to say that "Hondas are made in Japan" or "Fords are made in the US."
- How will we represent that many companies build cars in multiple countries, and that many countries host factories from many companies? (The US, for example, has factories for all four of the manufacturers listed.)

How are we going to link manufacturers to countries, then???



Manufacturers

ManufacturerID	ManufacturerName
1	Honda
2	Ford
3	BMW
4	Hyundai

Countries

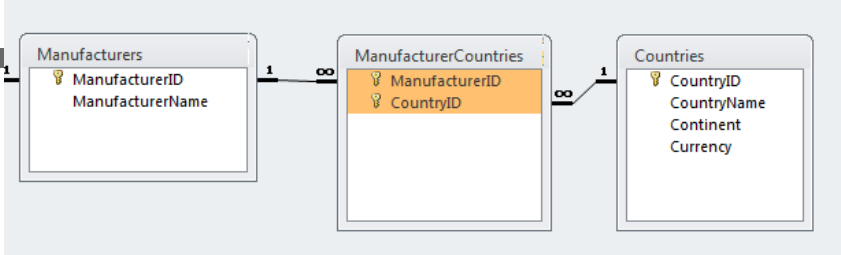
CountryID	CountryName	Continent	Currency
1	USA	North America	US Dollar
2	Japan	Asia	Yen
3	South Korea	Asia	Won
4	Germany	Europe	Euro

(The first step is hinted at here: Create a Countries table!)

Many-to-Many Relationship

- So: A company can build cars in many countries, and a country can have factories from many companies.
- This is what is called a *many-to-many relationship*.
- You can't express a many-to-many relationship simply by drawing a line between two tables.
- Many-to-many relationships require a third table, called an *intersection table*.

The solution



Manufacturers

ManufacturerID	ManufacturerName
1	Honda
2	Ford
3	BMW
4	Hyundai

ManufacturerCountries

ManufacturerID	CountryID
1	1
1	2
2	1
2	2
3	1
3	4
4	1
4	3

Countries

CountryID	CountryName	Continent	Currency
1	USA	North America	US Dollar
2	Japan	Asia	Yen
3	South Korea	Asia	Won
4	Germany	Europe	Euro

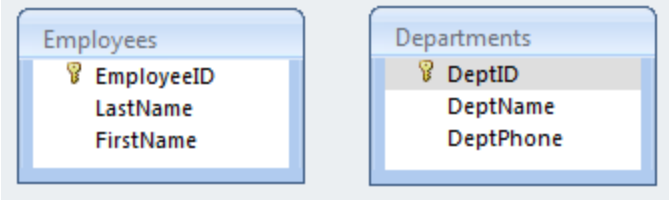
- Each row in the ManufacturerCountries table indicates that THAT manufacturer makes cars in THAT country.
- For example: The first row says that Honda makes cars in the USA. The next row says that Honda makes cars in Japan.
- Note that this intersection table uses BOTH fields as its primary key. In each column, the values are not unique. However, the combination of the two values is unique—we don't enter that Honda makes cars in the USA *twice*.

Creating the Many-to-Many Relationship

1. Remove the ManufacturerCountry field from the Manufacturers table.
2. Create a table for countries.
3. Create a new table which matches ManufacturerID's with CountryID's.
4. This table has a primary key which combines the primary keys of the other two tables.
5. Create the relationships—two one-to-many relationships.

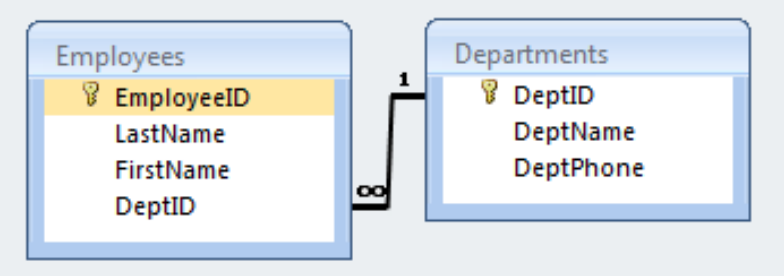
Relationship Type as Business Decision

- A business uses a database to track employees and departments.

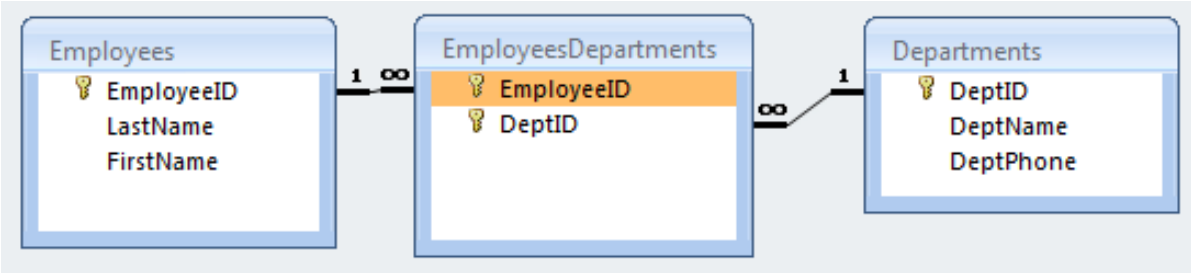


- If the business determines that an employee can only work for one department at a time, the relationship is one-to-many;
- If an employee can work for more than one department, the relationship is many-to-many.

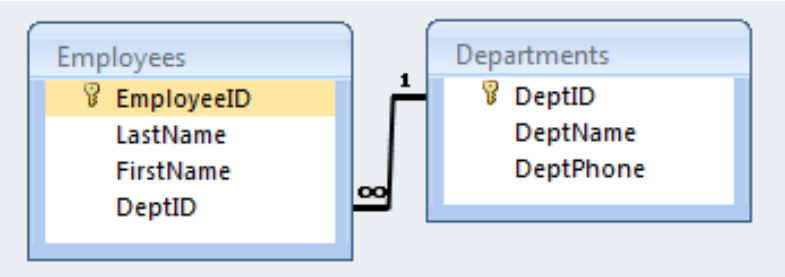
- To create the one-to-many relationship, we add a foreign key (DeptID) to the Employees table, and link it to the DeptID in the Departments table:



- To create the many-to-many relationship, we create an intersection table.
- The intersection table's primary key is compound, and includes the primary key of the two related tables:

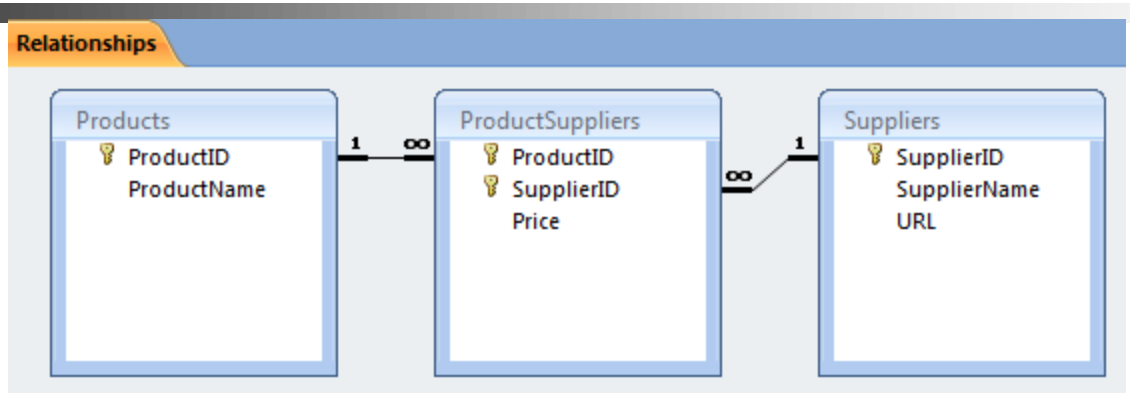


Summary: One-to-Many Relationships



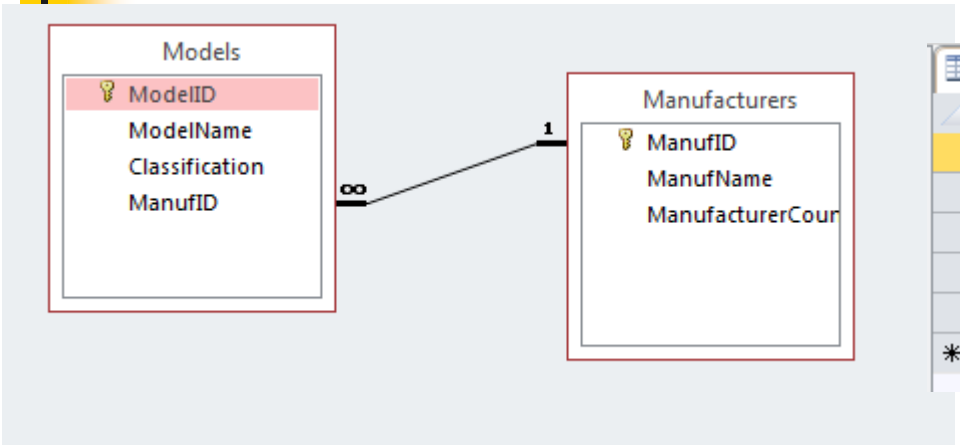
- Things to know:
  - Occurs when you can make a statement like this: "Each Department can have more than one employee, but each employee can be in at most one department."
  - Requires a foreign key field to be added to the table on the "many" side.
  - The foreign key field links to the primary key of the table on the "one" side.

Summary: Many-to-Many Relationships



- Things to know:
  - Occurs when you can make a statement like this: "Each Product may be available from more than one supplier, and a supplier may provide more than one product."
  - Requires intersection table (like ProductSuppliers above)
  - Primary key of intersection table is always compound and always includes the full primary keys of the two entity tables (Products, Suppliers)
  - The intersection table may also contain data that applies specifically to the intersection, such as Price above (different suppliers may have different prices for products).

Appendix

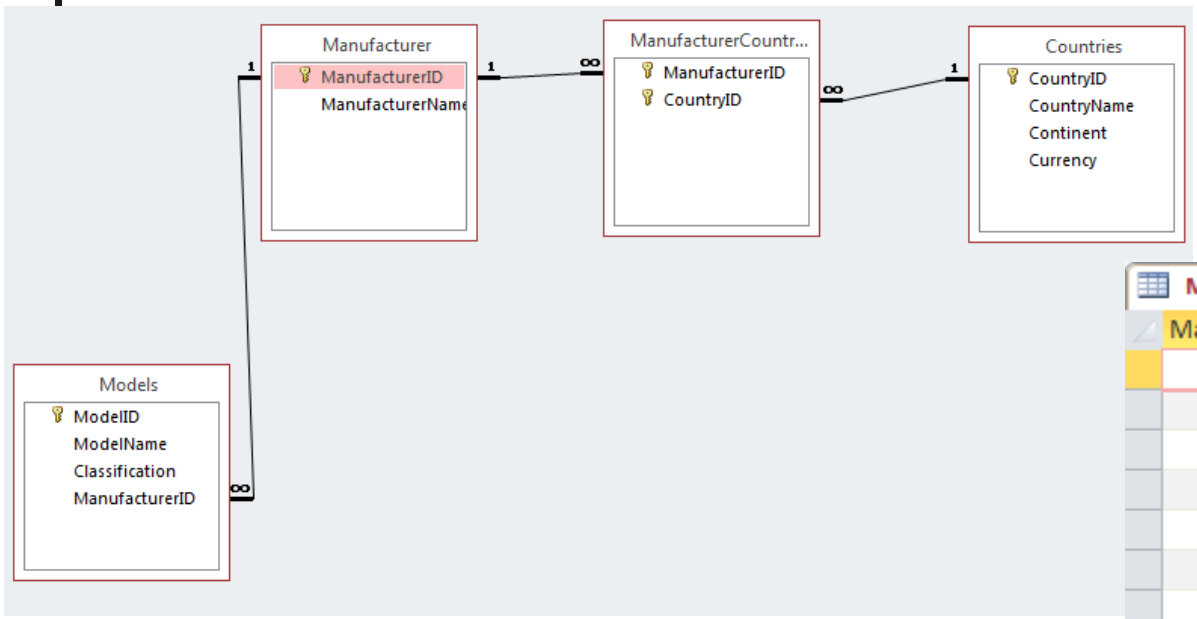


Manufacturers

ManufID	ManufName	Manufacture	Click to Add
1	Honda	Japan	
2	Ford	USA	
3	BMW	Germany	
4	Hyundai	South Korea	
5	GM	USA	
0			

Models

ModelID	ModelName	Classificatio	ManufID	Click to Add
1	Civic	Compact	1	
2	Accord	Midsize	1	
3	Fit	Subcompact	1	
4	Focus	Compact	2	
5	Fiesta	Subcompact	2	
6	Mustang	Sport	2	
7	Cruze	Compact	5	
0			0	



ManufacturerCountries

Manufacture	CountryID	Click to Add
1	1	
1	2	
2	1	
2	2	
3	1	
3	4	
4	1	
4	3	
*		

Relationships | Countries

CountryID	CountryNam	Continent	Currency	Click to Add
1	Usa	NorthAmerica	US Dollar	
2	Japan	Asia	Yen	
3	South Korea	Asia	Won	
4	Germany	Europe	Euro	
0				