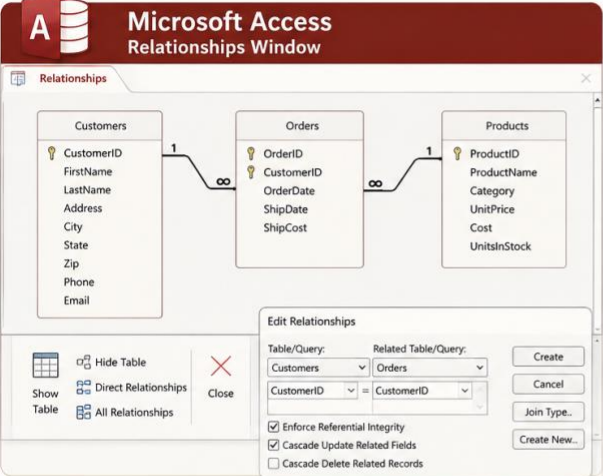


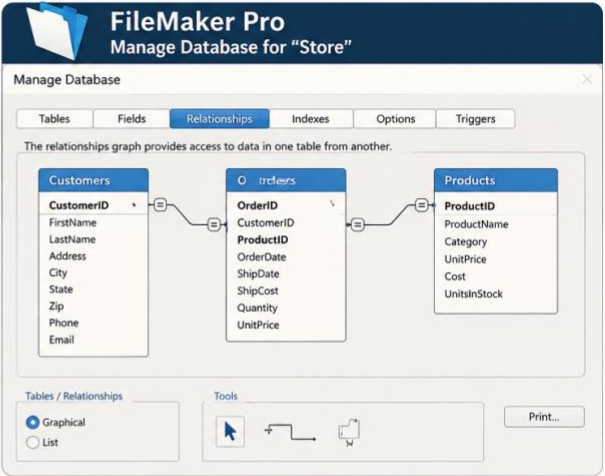
Relational Database Joins: Access vs. FileMaker

Linking Tables Through Common Fields



Microsoft Access Relationships Window

The Relationships window shows three tables: Customers, Orders, and Products. Customers is linked to Orders (1 to many), and Orders is linked to Products (many to many). The 'Edit Relationships' pane at the bottom shows the relationship between Customers and Orders, with options to enforce referential integrity and cascade updates/deletes.



FileMaker Pro Manage Database

The Manage Database window shows the same three tables. The Relationships tab displays a graphical relationship graph where Customers is linked to Orders, and Orders is linked to Products. The 'Tools' section at the bottom includes a 'Graphical' view option and a 'Print...' button.

ACCESS JOIN EXAMPLE

- Drag matching fields between tables.
- "1" means one record on the "one" side.
- "∞" means many records on the "many" side.
- Enforce Referential Integrity keeps your data accurate.
- Cascading options update or delete related records automatically.

Key symbols:  = Primary Key,  = Relationship

FILEMAKER JOIN EXAMPLE

- Add your tables, then drag a field from one table to the related field in another.
- Matching field types connect the tables.
- The relationship line shows how data is linked.
- FileMaker handles the join automatically when you show related data.

Key symbols:  = Relationship,  = Primary Key



WHAT A JOIN DOES

A join combines information from related tables using a common field (Primary Key ↔ Foreign Key). This eliminates duplicated data and keeps your database accurate, efficient, and easy to maintain.

A relational database stores information in separate, related tables that share common fields. This storage system reduces duplication and makes data easier to organize, update, and retrieve.

Why Relational Databases Beat Spreadsheets

The Secret Is That They Remember Things Only Once

One of the first questions people ask when learning databases is, "Why can't I just use Excel?" For many small jobs, you certainly can. If you're keeping track of a Christmas card list or a collection of baseball cards, a spreadsheet may be all you'll ever need.

But as your information grows, spreadsheets begin asking the same questions over and over again. Databases don't. That's the real advantage of a **relational database**.

Imagine Running a Small Business

Suppose you own a sporting goods store. You need to keep track of your customers, the products you sell, and every purchase they make.

An Excel spreadsheet might look something like this:

Customer	City	Product	Price
----------	------	---------	-------

John Smith	Denver	Tennis Racquet	\$150
John Smith	Denver	Tennis Balls	\$8
John Smith	Denver	Running Shoes	\$95

Notice what's happening. John Smith's address is repeated over and over again. If he moves, you may have to change his address dozens—or even hundreds—of times. That's inefficient, and it's also an invitation to mistakes.

A Better Way

A relational database stores information only once. Instead of one giant table, it creates several smaller ones. One table stores customers, another stores products, and a third stores orders. Each table becomes responsible for only one kind of information.

Instead of repeating John Smith's address every time he buys something, the Orders table simply stores his **Customer ID**. The database already knows where John Smith lives. It doesn't have to ask again.

The Magic Number

Every customer receives a unique identification number. Every product receives one too. Those identification numbers become the glue that holds the database together.

Think of them as license plates. Two people may both be named John Smith, but no two customers have the same Customer ID. That's how the database always knows exactly who you're talking about.

Where Customers Meet Products

Now comes the clever part. Instead of storing customer information inside the Products table—or product information inside the Customers table—we create a third table.

Database designers call it a **junction table**. I prefer to think of it as the **meeting place**, because that's exactly what it is. This table simply records the Customer ID, Product ID, purchase date, quantity, and price paid. Everything else is already stored somewhere else.

Why This Matters

Suppose John Smith moves across town. In a spreadsheet, you might have to update hundreds of records. In a relational database, you change one record in the Customers table, and every invoice, order, report, and purchase history automatically reflects the new address.

The same principle applies when a product description changes, a supplier is replaced, or inventory levels are updated. One correction updates the entire system while preserving the integrity of your historical records.

Behind the Curtain

When database designers connect those tables, they create what is known as a **relationship**. Programs such as Microsoft Access and FileMaker let you create those relationships visually by dragging matching fields together.

Once the relationships exist, the software quietly performs all the difficult work behind the scenes. The user simply sees information that appears to flow effortlessly from one screen to another.

Why Businesses Love Relational Databases

One well-designed relational database can answer questions like:

- Which customers bought this product?
- Which products has this customer purchased?
- Which items are almost out of stock?
- Who hasn't ordered in six months?
- What were our sales last quarter?

Those answers often appear in seconds. Trying to answer the same questions in a giant spreadsheet can become an exercise in frustration as the amount of information continues to grow.

The Bottom Line

Relational databases require a little more planning than spreadsheets, but they reward that effort with better organization, fewer typing mistakes, faster reporting, and information that stays consistent even as a business grows.

That's why nearly every major organization—from hospitals and universities to banks and airlines—depends on relational databases every day. Once you understand the simple idea of **storing information only once and connecting it through relationships**, you've taken the biggest step toward thinking like a database designer.