

My Favorite Company: A Customer Product Ordering Database Powered by Microsoft Access 2016

Matthew Fisher

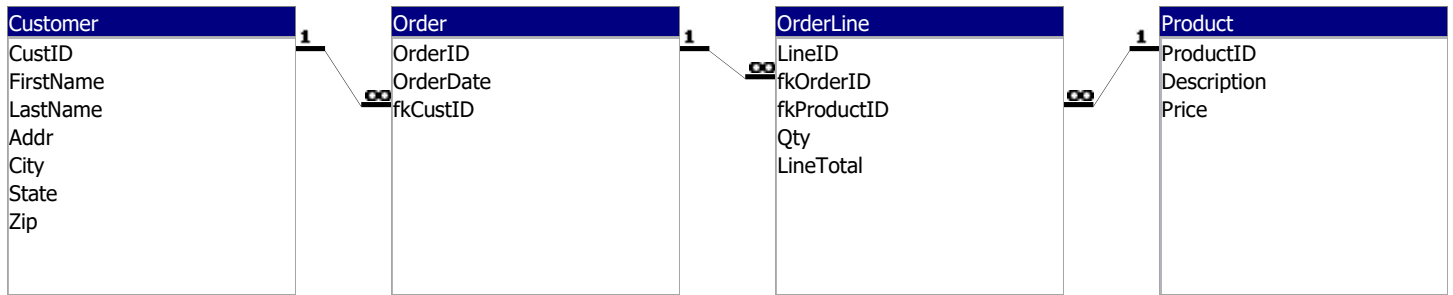
BA325-OL1: Competing w/ Information Technology

Portland State University: Professor Gerbing

July 15, 2017

ID	FirstName	LastName	Addr	City	State	Zip
1	Sam	Smith	44 NE 44th St	Portland	OR	97201
2	Sally	Surich	55 SW 55th St	Portland	OR	97231
3	Matt	Fisher	1234 SW Mai	Portland	OR	97211
4	Lisa	Mitchell	1234 SE Cou	Portland	OR	97201

ProductID	Description	Price
1	Jacket Liner	\$199.00
2	Pant Liner	\$199.00
3	Gloves	\$139.00



My Favorite Company: Database Structure
(Tables, Fields, & Relationships)

As professor Gerbing of Portland State University states, “a crucial element of database structure is to relate the tables with the appropriate one-to-many relationships.” Each table and the fields they manage can be “tied together” to create what is called a “one-to-many” relationship. A one-to-many relationship is a lot like it sounds, it links one table to potentially many other completely separate tables. All tables still need to exist with the same relational database, and there needs to be at least one “primary key” linked to at least one “foreign key” for it to qualify as a one-to-many relationship.

In the case of this database, we have a minimum four tables in the My Favorite Company or ‘MyFirst’ database that each have a primary key and foreign key assigned to specific fields within each table. The first table is titled ‘Customer’ and holds the customer information with a primary key assigned as ‘CustID’. This primary key is linked to the foreign key of the second database titled ‘Order’ which contains part of our order information. The foreign key field is titled ‘fkCustID’ in the Order table. The primary key ‘OrderID’ in the Order table is then linked to the foreign key of the third table titled ‘OrderLine’ which handles the calculation portion of the database. The OrderLine table is unique in that it has two foreign keys. The primary key OrderID is linked to the ‘fkOrderID’ foreign key in the Orderline table and the second foreign key ‘kfProductID’ is linked to the primary key titled ‘ProductID’ in the fourth and final table in the database, ‘Product’. The OrderLine table is the only table with an unlinked primary key though it still has one defined. The final table in the database, Product, handles the description and pricing of the available pricing in the database.

These one-to-many relationships all come together to form a functioning relational database structure. This structure is the minimum required to produce a working invoicing database. The database can be called on to produce desired information to appear on a form that is pre-designed. This form is then potentially utilized by both the customer and the business to complete orders. The relationships within the database can be edited, though the above described structure works best for the type of information being stored on the scale that My Favorite Company requires.

With these relationships in place the database and the information contained in each table can be used to create reports, run queries, or generate invoices. My Favorite Company utilizes this database for each of these functionalities.

My Favorite Company

800.866.6666 <http://www.myfavoritecompany.com>

Order ID	<input type="text" value="2"/>
Order Date	<input type="text" value="7/12/2017"/>
Customer ID	<input type="text" value="3"/>
First Name	<input type="text" value="Matt"/>
Last Name	<input type="text" value="Fisher"/>
Address	<input type="text" value="1234 SW Main St."/>
City	<input type="text" value="Portland"/>
State	<input type="text" value="OR"/>
Zip	<input type="text" value="97211"/>

Order

PrdID	Description	Price	Qty	LineTotal
<input type="text" value="1"/>	Jacket Liner	<input type="text" value="\$199.00"/>	<input type="text" value="2"/>	<input type="text" value="\$398.00"/>
<input type="text" value="2"/>	Pant Liner	<input type="text" value="\$199.00"/>	<input type="text" value="15"/>	<input type="text" value="\$2,985.00"/>
			Subtotal	<input type="text" value="\$3,383.00"/>

My Favorite Company

800.866.6666 <http://www.myfavoritecompany.com>

Order ID	<input type="text" value="3"/>
Order Date	<input type="text" value="7/13/2017"/>
Customer ID	<input type="text" value="1"/>
First Name	<input type="text" value="Sam"/>
Last Name	<input type="text" value="Smith"/>
Address	<input type="text" value="44 NE 44th St"/>
City	<input type="text" value="Portland"/>
State	<input type="text" value="OR"/>
Zip	<input type="text" value="97201"/>

Order

PrdID	Description	Price	Qty	LineTotal
<input type="text" value="1"/>	Jacket Liner	\$199.00	<input type="text" value="1"/>	\$199.00
<input type="text" value="2"/>	Pant Liner	\$199.00	<input type="text" value="3"/>	\$597.00
<input type="text" value="3"/>	Gloves	\$139.00	<input type="text" value="2"/>	\$278.00
			Subtotal	<input type="text" value="\$1,074.00"/>

Zip	FirstName	LastName
97201	Sam	Smith
97201	Lisa	Mitchell

Customer Report

Zip	Count	Customer ID	LastName	FirstName
97201		4	Mitchell	Lisa
		1	Smith	Sam
	2			
97211		3	Fisher	Matt
	1			
97231		2	Surich	Sally
	1			
Total Count:	4			

Reflection: My Favorite Company Database

This database illustrates the minimum requirements to produce a working invoicing database and introduces the four primary functions any database performs. These are the manipulation and organization of data, and the retrieval and summarization of that data. The same structure utilized in this database can be utilized to create simple invoices for almost any company/firm. Additional tables could be added to include other types of information, and fields could be altered to contain anything from photos to functioning buttons for options such as printing generated invoices.

Beyond invoices, it could be used as a product catalog meant to manage the descriptions of inventory a company holds. It could also be easily utilized for general inventory, general transaction records, surveys, sales lead records, supply chain records, human resources department records, etc. A creative example of how this database could be altered for a different use is that it could be used to generate tickets and manage ticketing for a concert venue. Concertgoers or staff could enter customer information and/or payment information into a similarly designed relational database that could then be used to generate tickets and track the records of attendees.

Nearly all information that can be recorded digitally could be managed within a relational database and as well, queried and/or utilized for generating reports. In practice, the more simplified the information is that is contained in the database and the less of it there is means the database is more like to run smoothly, efficiently, and free of errors. This is important to keep in mind when considering the structure and amount of information used by the intended database. The purpose for the creation of this specific database, invoicing, is one of the most common uses of a relational database and therefore the perfect starting example for creating a sample database for use by a small business. Potential uses are only limited by digital storage ability, capacity, and perhaps the range of ability of the software used to design the database. Microsoft Access can be used pretty intuitively by those who are familiar with Microsoft Office related products, and Access allowed this database to be built without having to overcome any significant learning curves. Microsoft Access is widely available and relatively easy to use compared to some databases built on legacy software platforms.

Lastly, the concept of one-to-many relationships is important to understand within the database environment and outside of it. Computer programming and logic structure utilizes similar concepts in the way their operations are designed. With the information technology industry continue to grow in size and value, understanding fundamental database operations can bring added value to the workplace. Implementing databases to manage any range of tasks could bring added efficiency and increased value to a company. The flexibility and functionality offered by modern enterprise database software is extremely robust and continues to create revenues in the trillions of dollars for firms around the globe. Relational databases are common and their trend of use will continue well into the future.