

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Database

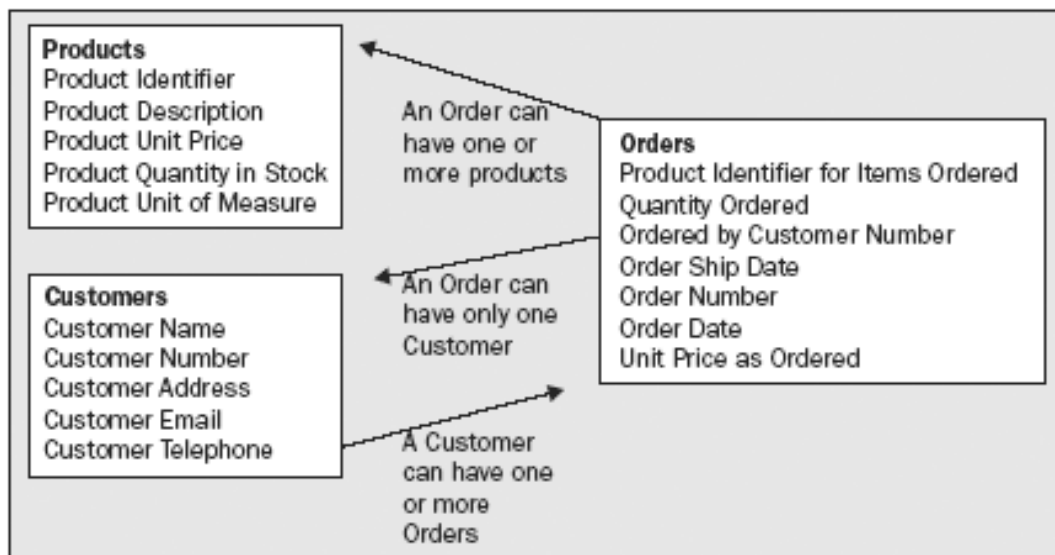
منبع پایگاه همکاران سیستم

پایگاه داده (Database) چیست ؟

پایگاه داده یک وسیله الکترونیکی جهت ذخیره داده (Data) بصورت سازمان یافته است. پس از اینکه داده ها در پایگاه داده ذخیره شد این داده ها قابل بازیابی ، پردازش می باشند و امکان تولید اطلاعات (Information) از داده ها وجود دارد.

فایل های تخت (Flat File) در مقابل پایگاه های داده رابطه دار (Relational Database)

ابتدایی ترین سطح پایگاه داده ها فایل ها تخت هستند که داده ها تنها در یک فایل ذخیره می شوند. در نظر بگیرد که یک صفحه اکسل را باز کرده اید و اطلاعات زیر را در آن وارد ساخته اید. با این کار شما یک دیتا بیس تخت ایجاد کرده اید. (شکل ۱)

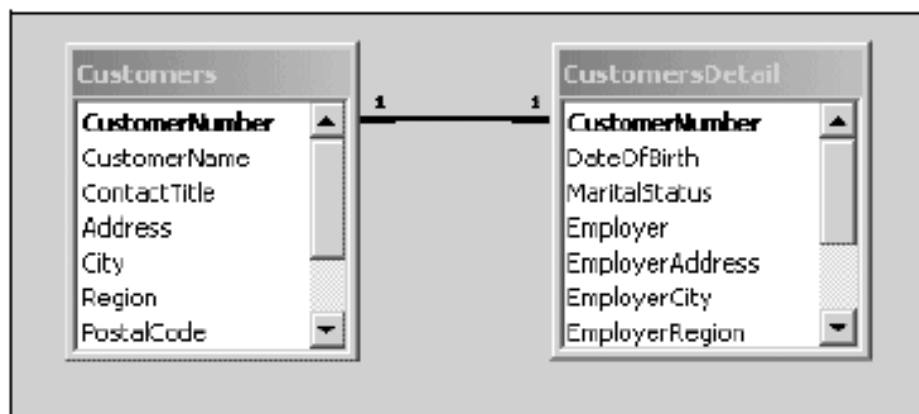


به قسمت های خاکستری رنگ توجه کنید داده های تکراری در فایل به چشم می خورند. این فایل های تخت دو مشکل اساسی دارند :

۱- تکرر داده ها

۲- مشکلات نگهداری پایگاه داده

برای رفع این مشکلات می توانید از پایگاه داده رابطه دار بهره بگیرید. این نوع پایگاه داده ها اطلاعات را در جداول ۱ مجزا نگهداری می کنند. برای مثال می توان اطلاعات فوق را در جداول مشتری ، سفارش و اقلام سفارش نگاه داشت و فیلدهای هر جدول به صورت زیر است : (شکل ۲)



بنابراین اطلاعاتی نظیر کد مشتری ، نام مشتری ، نام خانوادگی مشتری ، آدرس مشتری ، شهر مشتری ، ایالت مشتری و ... در جدول مشتری قرار می گیرد.
شماره سفارش ، کد مشتری ، تاریخ سفارش در جدول سفارشات قرار می گیرد.
شماره قلم سفارش ، شماره سفارش ، توضیح سفارش ، مقدار سفارش ، قیمت قلم و واحد آن در جدول اقلام سفارش قرار می گیرد.
برای استفاده از این جداول باید بین آنها ارتباط برقرار شود که این موضوع در ادامه مورد بررسی قرار خواهیم داد.

تعیین نیازمندیهای یک پایگاه داده

قبل از آنکه پایگاه داده طراحی شود باید نیازمندیهای آن را معین کنیم. برای تشخیص نیازهای پایگاه داده یک سازمان رهنمودهای زیر را در نظر داشته باشید :

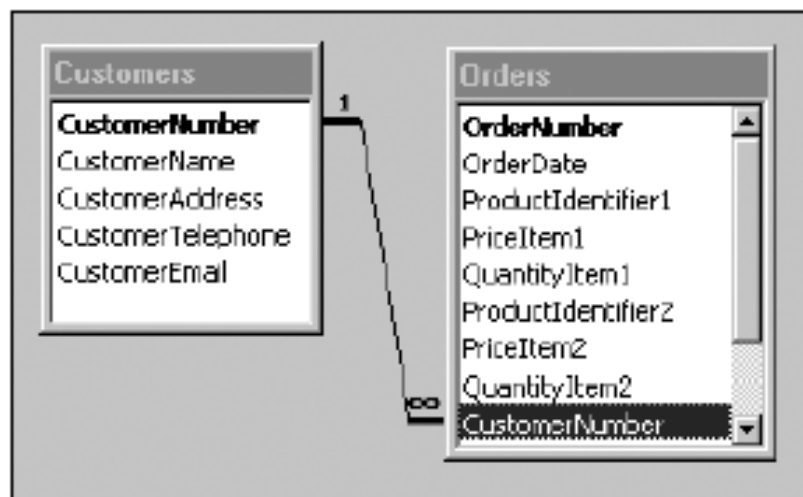
رهنمود ۱ : پایگاه داده های موجود را به دقت بررسی کنید.

رهنمود ۲ : با تک تک کاربران مصاحبه کنید

رهنمود ۳: از فرمهای شرکت کپی تهیه کنید
رهنمود ۴: به دقت گزارش های موجود را بررسی کرده اند نیازهای گزارشی را نیز مد نظر قرار دهید.

تعیین اطلاعاتی که باید ردیابی شود

پس از مصاحبه با افراد سازمان شما باید اطلاعاتی که باید ردیابی شود را مشخص کنید. فرض کنید در یک سازمان به یک لیستی مانند لیست زیر دست پیدا کردید (شکل ۳)

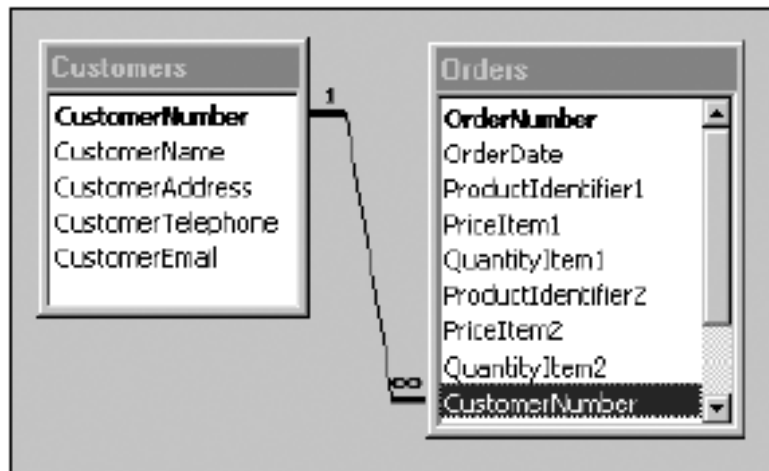


تعیین طراحی منطقی پایگاه داده

پس از شناسایی نیازها شما باید طرح اولیه و پیش نویس پایگاه داده ها را تهیه کنید. در واقع قبل از اینکه آنرا بصورت الکترونیکی ایجاد کنید روی یک کاغذ نقشه اولیه آنرا می کشید به این کار اصطلاحاً طراحی منطقی پایگاه داده (Logical Database Design) می گویند.

تعریف جدول ها (موجودیت ها) و فیلدها (ویژگیها)

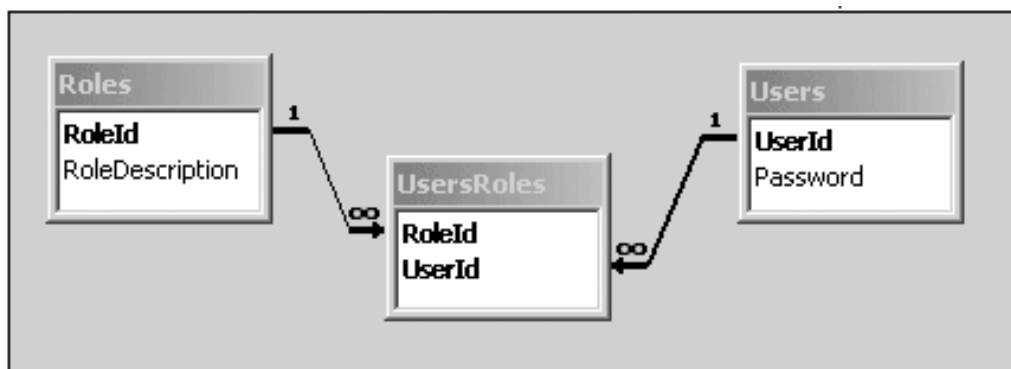
اولین قدم در طراحی منطقی پایگاه داده تعریف جداول است که موجودیت نیز نامیده می شود. جدولها گروه های مرتبطی از اطلاعات می باشند. به مثال قبل در مورد سفارش و مشتری دقت نمایید (شکل ۴)



فیلدها که ویژگی نیز نامیده می شوند در واقع عنصر اساسی جدولها هستند. مانند فیلد نام مشتری یا آدرس مشتری در جدول مشتری .

هویت دادن به جدول ها و فیلدها

پس از تعیین جداول و فیلدها باید به آنها هویت بخشید . نمونه ای از آنها را مشخص نمود و نوع هر فیلد را تعیین کرد. برای این کار می توانید از لیستی که قبلا تهیه کردید (شکل ۳) کمک بگیرید. برای مثال در شکل ۵ نمونه ای از این جدول آمده است:



برای مثال در جدول کالا شرح کالای ۱۲۳۴۵ ، توبا (Tofa) می باشد. این فیلد از نوع داده متنی می باشد و تخمین ما این است که این شرح حداکثر ۲۵ کاراکتر خواهد بود.

UsersRoles : Table		
	UserId	RoleId
▶	jdoe	2
	jdoe	5
	jsmith	1
	jsmith	2
	jsmith	3
	jsmith	4
	jsmith	5
*		
Record: 1 of		

Users : Table		
	UserId	Password
▶	jdoe	*****
	jsmith	*****
*		
Record: 1 of		

Roles : Table		
	RoleId	RoleDescription
▶	1	Add
	2	Edit
	3	Update
	4	Delete
	5	View
*		
Record: 1 of		

یا در جدول مشتری نام مشتری ۱۲۳۴۵۶ ، جان ای.دویی می باشد. این فیلد از نوع داده متنی می باشد و تخمین ما این است که این شرح حداکثر ۴۵ کاراکتر خواهد بود.

CUSTOMERS TABLE			
Field	Example	Type of Data	Estimated Size of Data
CustomerNumber (PK)	123456	Numeric	Positive number with no decimals
CustomerName	Jane A. Doe	Text	45 characters
CustomerAddress	123 Somewhere St., Anytown, IN 46060 USA	Text	65 characters
CustomerTelephone	317-111-2222	Text	12 characters
CustomerEmail	jdoe@yahoo.com	Text	50 characters

و سرانجام فیلدهای جدول سفارش مانند شکل فوق خواهد بود.

تعیین کلید اصلی (Primary Key)

کلید اصلی یک فیلد یا مجموعه ای از فیلدها است که یک رکورد (Record) در جدول را یکتا (Unique) می کنند. برای مثال در جدول مشتری شماره مشتری یک کلید اصلی می باشد. هر مشتری یک شماره دارد و هر مشتری نشان دهنده یک هویت است
یک در جدول سفارش ، شماره سفارش کلید اصلی می باشد. هیچ دو سفارشی یک شماره مشابه نخواهند داشت.

تعیین کلید خارجی یا کلید مرتبط کننده (Foreign Key)

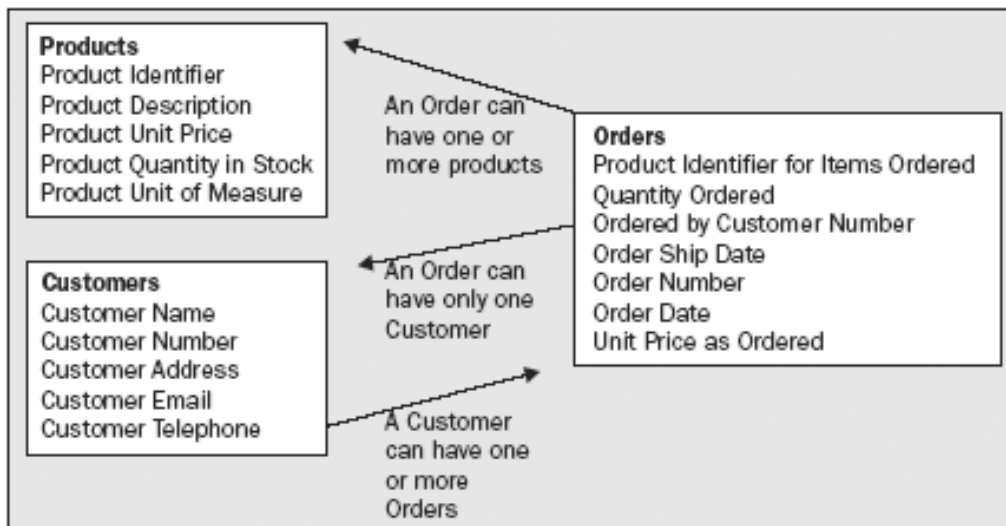
کلید خارجی یک فیلد یا مجموعه ای از فیلدها است که یک جدول را به کلید اصلی جدول دیگر مرتبط می سازد بنابراین این دو جدول مرتبط می شوند. برای مثال شماره مشتری و شماره کالا کلیدهای اصلی جداول مشتری و کالا می باشند که به عنوان کلید خارجی در جدول سفارش عمل می کنند. به شکل ۶ دقت نمایید.

تعریف روابط در پایگاه داده

به مثال مشتری ، کالا و سفارش کالا برمیگردیم .میتوان گزاره های زیر را بیان کرد :

یک سفارش می تواند شامل چند کالا باشد
یک سفارش تنها به یک مشتری مربوط است
هر مشتری می تواند چندین سفارش داشته باشد

به زبان پایگاه داده ، این گزاره ها روابط میان جدولها را بیان می کنند. این روابط را می توان در نمودار موجودیت ها (ERD) نمایش داد (شکل ۷)



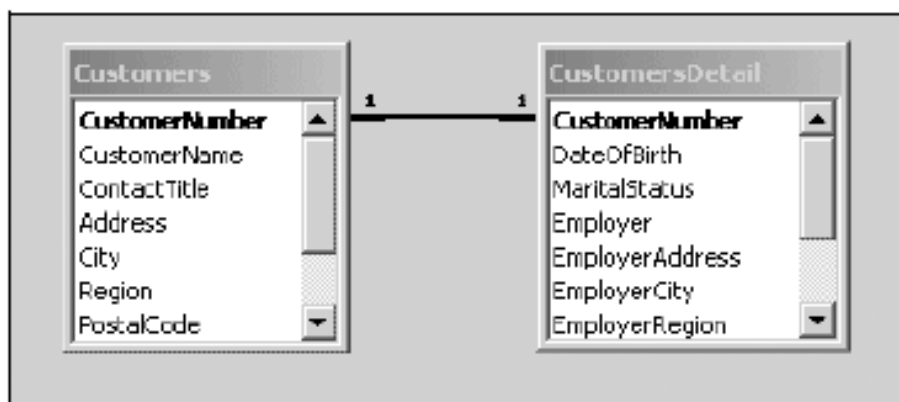
روابط میان موجودیت ها می توان به ۳ صورت باشد :

- ۱- یک به یک (One to One)
- ۲- یک به چند (One to Many)
- ۳- چند به چند (Many to Many)

در ادامه به تشریح هریک از این روابط می پردازیم

رابطه یک به یک

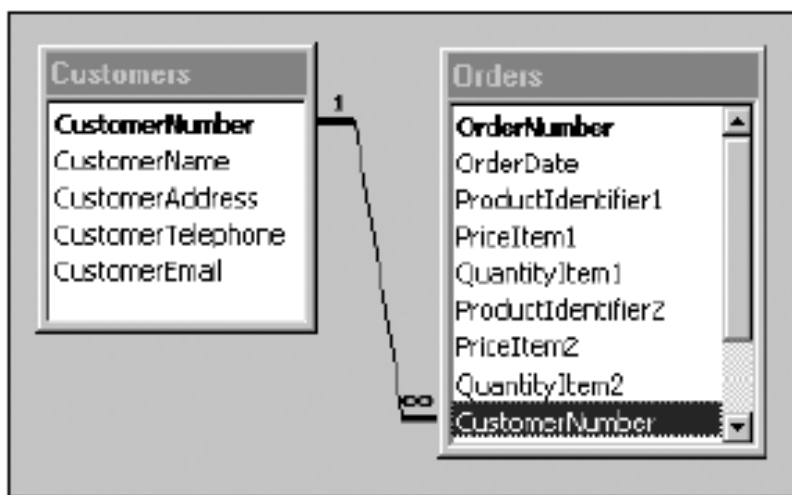
رابطه یک به یک مشابه تناظر یک به یک در ریاضیات است . در این نوع از رابطه، هر رکورد از جدول اول تنها و تنها با یک رکورد در جدول دوم رابطه دارد. برای مثال فرض کنید اطلاعات هر مشتری ۱۰۰ فیلد دارد و پایگاه داده ما این تعداد فیلد را پشتیبانی نمی کند بنابراین بخشی از اطلاعات مشتری را در جدولی بنام CustomerDetial قرار می دهیم. این جدول با جدول Customer رابطه یک به یک دارد. فیلدی که این دو جدول را بهم مرتبط می سازد فیلد Customernumber می باشد. (شکل ۸)



به اعداد ۱ و ۱ روی خط مرتبط کننده دقت کنید. این نشان دهنده رابطه یک به یک است.

رابطه یک به چند

در این نوع از رابطه هر رکورد از جدول اول می تواند با چند رکورد در جدول دوم رابطه داشته باشد. مثال مشتری و سفارش نمونه ای مناسب برای این رابطه است. هر مشتری می تواند چندین سفارش داشته باشد یعنی هر رکورد (مشتری) از جدول Customer می تواند با چند رکورد (سفارش) از جدول Order مرتبط باشد (شکل ۹)



به عدد ۱ و علامت بی نهایت روی خط مرتبط کننده دقت کنید این نشان دهنده رابطه یک به چند است و اینچنین خوانده می شود : هر رکورد از جدول مشتری می تواند با چند رکورد از جدول سفارش مرتبط باشد.

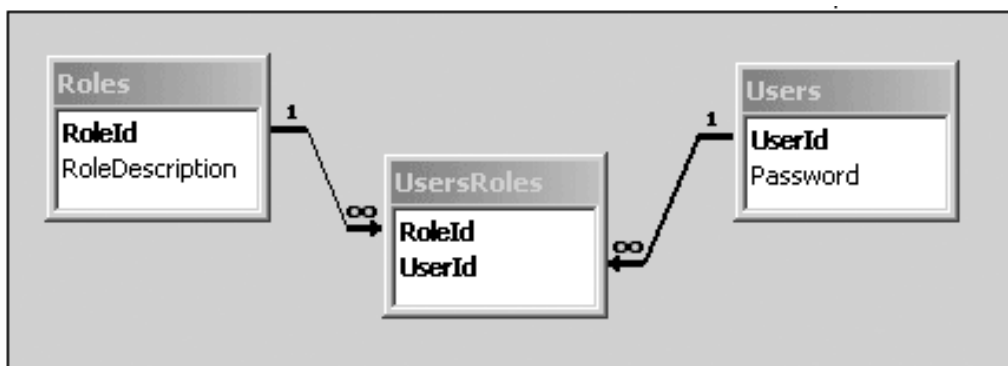
رابطه چند به چند

اینجاست که دنیا آشفته و پیچیده می شود. در این نوع از رابطه هر رکورد از جدول اول می تواند با چند رکورد در جدول دوم رابطه داشته باشد و بالعکس. اما چگونه می توان این پیچیدگی و آشفتگی را حل کرد؟ با استفاده از یک جدول سوم یا جدول واسط می توان این مشکل را برطرف نمود. فرض کنید که یک سیستم کاربران (User) زیادی دارد و هر کاربر می تواند نقش های (Role) متفاوتی داشته باشد به عبارت دیگر :

هر کاربر می تواند چند نقش داشته باشد

هر نقش می تواند به چند کاربر مربوط باشد

برای ارتباط برقرار کردن میان دو جدول کاربران (Users) و نقش ها (Roles) جدول واسطی میان این دو جدول بنام نقش کاربران (Userroles) ایجاد می کنیم. اکنون هر دو جدول کاربران و نقش ها با این جدول رابطه یک به چند دارند. بنابراین یک رابطه چند به چند را به دو رابطه یک به چند تبدیل نمودیم : (شکل ۱۰)



به زبان دیگر :

One to Many + Many to Many = One to Many

به اطلاعات این جداول دقت نمایید :

The screenshot displays three database tables in a management tool:

- UsersRoles : Table**

UserId	RoleId
jdoe	2
jdoe	5
jsmith	1
jsmith	2
jsmith	3
jsmith	4
jsmith	5
- Users : Table**

UserId	Password
jdoe	*****
jsmith	*****
- Roles : Table**

RoleId	RoleDescription
1	Add
2	Edit
3	Update
4	Delete
5	View

یکپارچگی ارجاع (Integrity Referential)

با ایجاد رابطه میان جدول از مزیتی بنام یکپارچگی ارجاعی استفاده می کنیم . مزیت یکپارچگی بطور خلاصه به موارد زیر اشاره دارد :

۱- به روز رسانی آشناری : یعنی هر گاه فیلد کلیدی در یگ جدول تغییر کرد این تغییر باید در جدول دیگر که از طریق این فیلد با جدول مربوطه در ارتباطند نیز اعمال گردد.

۲- حذف آشناری : یعنی هر گاه قرار است فیلد کلیدی از یگ جدول حذف شود باید اطمینان حاصل شود که در تمام جدول مرتبط حذف صورت پذیرفته است.

توجه داشته باشید که یکپارچگی ارجاعی اگرچه معمولاً یک مزیت بشمار می رود ولی گاهی اوقات مساله ساز نیز می شود.

نرمال سازی (Normalization)

ساختن یک پایگاه داده مانند ساختن یک خانه است. آیا برای ساختن یک خانه دانستن تعداد اتاق های مورد نیاز کافی است و یا اینکه باید به پارامترهای دیگر نیز توجه داشت تا بتوان نقشه ای بهینه برای خانه تهیه کرد. نرمال سازی در واقع فرایند تهیه طرح بهینه برای یک پایگاه داده است. نرمال سازی مراحل دارد و باید به یک قانون توجه داشت :

هرچه در مرحله بالاتری از نرمال سازی باشید پایگاه داده شما طرح و ساختاری کاراتر خواهد داشت.

نرمال سازی سطح اول (First normal form)

در نرمال سازی سطح اول گروه های تکراری را حذف نمایید.

این گروه های تکراری میتوانند دو حالت را در بر می گیرند :

۱- در یک فیلد داده های زیاد و طولانی قرار می گیرد.

۲- دو فیلد مشابه وجود دارد.

برای مثال جدول مشتری را در نظر بگیرید (شکل ۱۱)

CUSTOMERS TABLE			
Field	Example	Type of Data	Estimated Size of Data
CustomerNumber (PK)	123456	Numeric	Positive number with no decimals
CustomerName	Jane A. Doe	Text	45 characters
CustomerAddress	123 Somewhere St., Anytown, IN 46060 USA	Text	65 characters
CustomerTelephone	317-111-2222	Text	12 characters
CustomerEmail	jdoe@yahoo.com	Text	50 characters

در این جدول به فیلد آدرس مشتری Customer Address توجه کنید. اطلاعات زیادی در این فیلد قرار می گیرند و عملاً قابلیت بازیابی اطلاعات کاهش می یابد. این فیلد را می توان به پنج فیلد آدرس مشتری، شهر مشتری، استان مشتری، کد پستی و کشور مشتری تقسیم نمود. (شکل ۱۲)

CUSTOMERS TABLE			
Field	Example	Type of Data	Estimated Size of Data
CustomerNumber (PK)	123456	Numeric	Positive number with no decimals
CustomerFirstName	Jane	Text	15 characters
CustomerMiddleName	A.	Text	15 characters
CustomerLastName	Doe	Text	25 characters
CustomerAddress	123 Somewhere St.	Text	30 characters
CustomerCity	Anytown	Text	20 characters
CustomerState	IN	Text	2 characters
CustomerZip	46060	Text	9 characters
CustomerCountry	USA	Text	20 characters
CustomerTelephone	317-111-2222	Text	12 characters
CustomerEmail	jdoe@yahoo.com	Text	50 characters

اکنون به جدول سفارشات توجه نمایید (شکل ۱۳). در این جدول فیلدهای تکراری مانند شماره ، قیمت و مقدار کالا وجود دارد.

ProductIdentifier1 (FK)	12345	Numeric	Positive number with no decimals
PriceItem1	\$19.00	Currency	\$00.00 to \$10,000.00
QuantityItem1	2	Numeric	0 to 9,999
ProductIdentifier2 (FK)	2345	Numeric	Positive number with no decimals
PriceItem2	\$8.50	Currency	\$00.00 to \$10,000.00
QuantityItem2	3	Numeric	0 to 9,999
ProductIdentifier3 (FK)	3456	Numeric	Positive number with no decimals
PriceItem3	\$13.00	Currency	\$00.00 to \$10,000.00
QuantityItem3	4	Numeric	0 to 9,999
ProductIdentifier4 (FK)	4567	Numeric	Positive number with no decimals
PriceItem4	\$15.00	Currency	\$00.00 to \$10,000.00
QuantityItem4	5	Numeric	0 to 9,999

برای نرمال کردن این جدول فیلدی تحت عنوان شماره سفارش به جدول اضافه می کنیم و فیلدهای اضافه را حذف می کنیم :

ORDERS TABLE			
Field	Example	Type of Data	Estimated Size of Data
OrderNumber (PK)	1000	Numeric	Positive number with no decimals
ProductIdentifier (PK)	12345	Numeric	Positive number with no decimals
OrderDate	Aug. 1, 2001	Date	Valid date
CustomerNumber (FK)	123456	Numeric	Positive number with no decimals
Price	\$ 15.00	Currency	\$00.00 to \$10,000.00
Quantity	5	Numeric	0 to 9,999
OrderShipDate	Aug. 3, 2001	Date	Valid date

این اطلاعات در جدول بصورت زیر خواهند بود .

Orders : Table							
	OrderNumber	ProductIdentifier	OrderDate	CustomerNumber	Price	Quantity	OrderShipDate
▶	1000	14	5/27/2001	12345	\$ 15.00	5	5/28/2001
	1000	23	5/27/2001	12345	\$ 13.25	1	5/28/2001
	1000	25	5/27/2001	12345	\$ 9.99	2	5/28/2001
	1001	7	5/27/2001	23456	\$ 17.32	1	5/28/2001
	1001	18	5/27/2001	23456	\$ 25.00	2	5/28/2001

CUSTOMERS TABLE			
Field	Example	Type of Data	Estimated Size of Data
CustomerNumber (PK)	123456	Numeric	Positive number with no decimals
CustomerFirstName	Jane	Text	15 characters
CustomerMiddleName	A.	Text	15 characters
CustomerLastName	Doe	Text	25 characters
CustomerAddress	123 Somewhere St.	Text	30 characters
CustomerCity	Anytown	Text	20 characters
CustomerState	IN	Text	2 characters
CustomerZip	46060	Text	9 characters
CustomerCountry	USA	Text	20 characters
CustomerTelephone	317-111-2222	Text	12 characters
CustomerEmail	jdoe@yahoo.com	Text	50 characters

ProductIdentifier1 (FK)	12345	Numeric	Positive number with no decimals
PriceItem1	\$19.00	Currency	\$00.00 to \$10,000.00
QuantityItem1	2	Numeric	0 to 9,999
ProductIdentifier2 (FK)	2345	Numeric	Positive number with no decimals
PriceItem2	\$8.50	Currency	\$00.00 to \$10,000.00
QuantityItem2	3	Numeric	0 to 9,999
ProductIdentifier3 (FK)	3456	Numeric	Positive number with no decimals
PriceItem3	\$13.00	Currency	\$00.00 to \$10,000.00
QuantityItem3	4	Numeric	0 to 9,999
ProductIdentifier4 (FK)	4567	Numeric	Positive number with no decimals
PriceItem4	\$15.00	Currency	\$00.00 to \$10,000.00
QuantityItem4	5	Numeric	0 to 9,999