Tutorial: Las 3 Formas Normales

Por Fred Coulson

Las preguntas pueden ser dirigidas a http://phlonx.com/contact

Bajado de http://phlonx.com/resources/nf3/

Tabla de Contenido

Introducción

Este trabajo está concebido para ser un seminario muy breve dirigido a los principiantes que quieren conseguir un dominio conceptual del proceso de normalización de bases de datos. Yo encuentro muy difícil visualizar estos conceptos usando solamente las palabras, por lo que me auxiliaré, tanto como sea posible, de imágenes y gráficos.

Para demostrar los importantes principios tratados, tomaremos el clásico ejemplo de una **Factura** y la llevaremos hasta la Tercera Forma Normal. También construiremos, por el camino, un **Modelo Entidad - Relación** de la Base de Datos (BD).

Nota Importante: Esto no es una descripción de cómo diseñar e implementar una BD. Las muestras de BD en forma de imágenes de pantalla no fueron pensadas para ser tomadas literalmente, sino simplemente como ayuda visual para mostrar como los datos son redistribuidos a medida que la estructura de la tabla se va transformando, cada vez más, en normalizada.

Los puristas y los académicos pueden no estar interesados en este tipo de acercamiento a este tema. Yo no trataré asuntos tales como las ventajas y desventajas de la normalización. Para aquellos que quieren profundizar en esta materia, al final se da una lista de referencias.

Para la mayoría, las primeras tres formas normales son las mas comúnmente aceptadas. Cuando la gente se sienta a diseñar una BD, ya ellos, con frecuencia, tienen en mente una estructura parcialmente normalizada ya que la normalización es una vía natural de percibir las relaciones entre los datos y para ello no se requieren habilidades matemáticas o teóricas.

En realidad, usualmente toma cierto trabajo denormalizar una BD (es decir, quitar las relaciones naturalmente eficientes que genera la estructura normalizada de los datos). La denormalización es una tarea bastante común pero no será tratada en esta presentación.

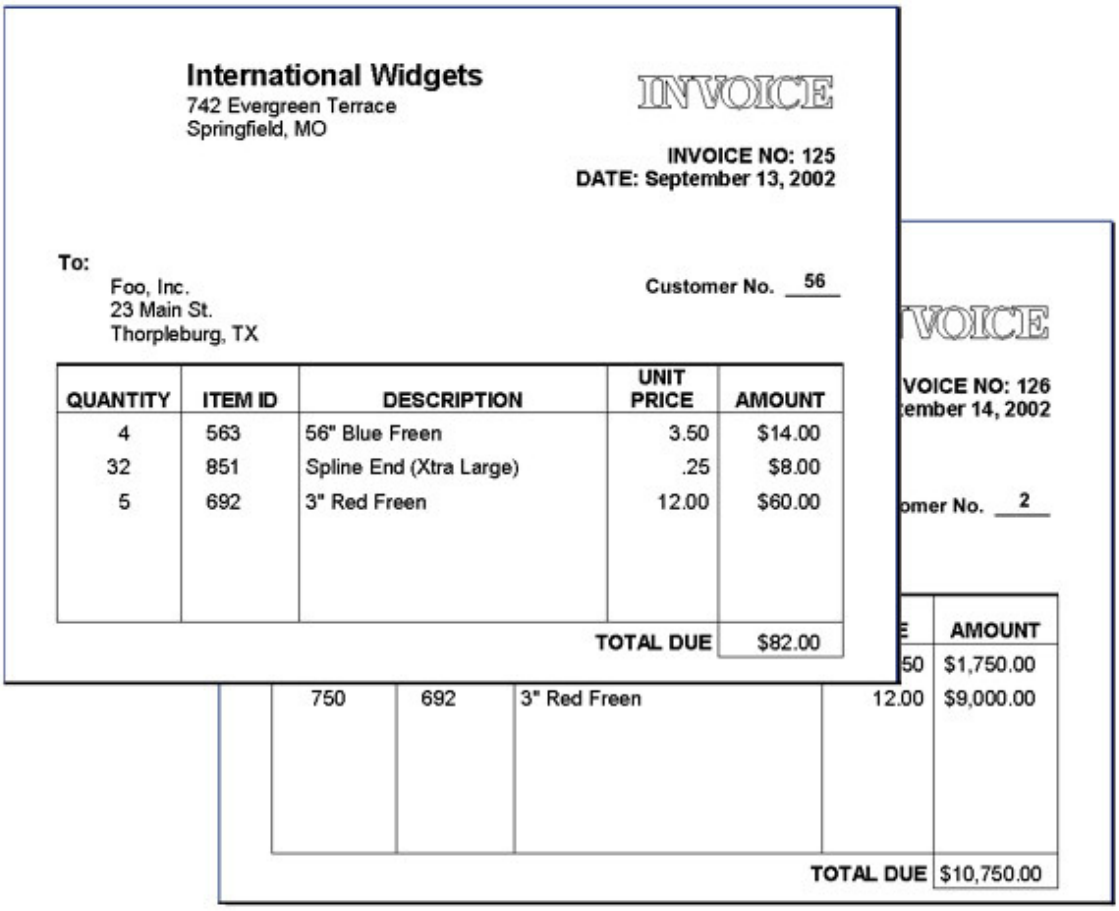
Para empezar: Primero, memorice las 3 formas normales de tal forma que pueda recitarlas cuando duerma. El significado se irá aclarando por el camino. Solo memoricemos por ahora:

1. No elementos repetidos o grupos de elementos
2. Sin dependencias parciales de llaves concatenadas
3. Sin dependencias de atributos que no son llaves

El Problema:
Manteniendo la Secuencia de las Facturas

Considere una factura típica (Figura A).

Figura A: Factura



International Widgets
742 Evergreen Terrace
Springfield, MO

INVOICE

INVOICE NO: 125
DATE: September 13, 2002

To:
Foo, Inc.
23 Main St.
Thorpleburg, TX

Customer No. 56

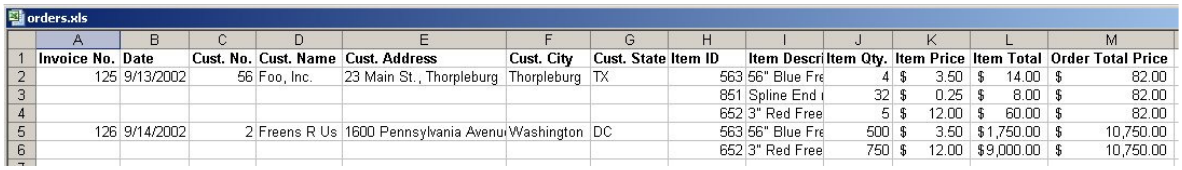
QUANTITY	ITEM ID	DESCRIPTION	UNIT PRICE	AMOUNT
4	563	56" Blue Freen	3.50	$14.00
32	851	Spline End (Xtra Large)	.25	$8.00
5	692	3" Red Freen	12.00	$60.00
			TOTAL DUE	$82.00

VOICE
VOICE NO: 126
ember 14, 2002
omer No. 2

				AMOUNT
			50	$1,750.00
750	692	3" Red Freen	12.00	$9,000.00
			TOTAL DUE	$10,750.00

Aquellos que tienen una mente ordenada pero no son muy conocedores de las BD relacionales podrían intentar capturar los datos de una Factura en una hoja de cálculo como Microsoft Excel.

Figura A-1: cuadricula de las órdenes



orders.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Invoice No.**	**Date**	**Cust. No.**	**Cust. Name**	**Cust. Address**	**Cust. City**	**Cust. State**	**Item ID**	**Item Descri**	**Item Qty.**	**Item Price**	**Item Total**	**Order Total Price**
2	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	563	56" Blue Fre	4	$ 3.50	$ 14.00	$ 82.00
3								851	Spline End (	32	$ 0.25	$ 8.00	$ 82.00
4								652	3" Red Free	5	$ 12.00	$ 60.00	$ 82.00
5	126	9/14/2002	2	Freens R Us	1600 Pennsylvania Avenu	Washington	DC	563	56" Blue Fre	500	$ 3.50	$1,750.00	$ 10,750.00
6								652	3" Red Free	750	$ 12.00	$9,000.00	$ 10,750.00

Esto no es un mal intento ya que en la hoja de cálculo se va almacenando cada compra que hace cada consumidor. Pero que pasaría si comenzamos a hacernos preguntas más complejas como por ejemplo:

- ¿Cuantos “3" Red Freens“ ordenó Freens R Us en el 2002?
- ¿Cuales han sido las ventas totales de 56” Blue Freens en el estado de Texas?
- ¿Que ítems fueron vendidos en Julio del 2003?

En la medida que la cuadricula crece, se va complicando el hecho de encontrar estas respuestas. Al tratar de organizar los datos de forma que podamos, razonablemente, encontrar las respuestas a estas preguntas, estamos comenzando a realizar el proceso de normalización.

Primera Forma Normal:
No elementos repetidos o Grupos de Elementos

Mire las filas 2, 3 y 4 de la cuadricula de la Figura A-1, ellas representan toda la información que tenemos para una simple factura (Factura # 125).

En la jerga de las bases de datos, este grupo de filas se refiere a una simple fila de una base de datos. Nunca piense que una fila de una base de datos está formada, como aquí, por tres filas de una cuadrícula: esto es una desafortunada ambigüedad del lenguaje. Los teóricos de las bases de datos tienen una palabra especial que ayuda un poco con esta ambigüedad: ellos llaman a la "cosa" contenida en las filas 2, 3 y 4, **tupla.** Nosotros no vamos a usar esa palabra en este trabajo (y si usted es afortunado, nunca la oirá mencionar nunca más en su carrera con las bases de datos). Aquí, nos referiremos a esta cosa como **fila**.

De esta forma, la Primera Forma Normal (PFN) nos indica que debemos deshacernos de los elementos repetidos. ¿Cuales son?

Volvemos a centrar nuestra atención en la factura #125 en la Figura A-1. Las casillas H2, H3 y H4 contienen una lista de ítems de los ID Numbers (números de identificación). Esto es una *columna* dentro de la primera fila de nuestra base de datos (BD). De la misma forma, I2–I4 constituyen una simple columna; lo mismo con J2–J4, K2–K4, L2-L4 y M2-M4. Las columnas de la BD son nombradas algunas veces como **atributos** (las filas y las columnas son tuplas y atributos).

Usted podrá notar que cada una de estas columnas contienen una lista de valores. Son, precisamente, estas listas las cuestionadas por NF1: NF1 aborrece las listas o arreglos con valores repetidos dentro de una simple columna de una BD. NF1 anhela la **atomicidad:** la indivisibilidad de un atributo dentro de partes similares.

Por lo tanto, queda claro que tenemos que hacer algo con los datos repetidos de ítems de información dentro de la fila para la factura #125. En la Figura A-1 son las celdas:

- H2 hasta M2
- H3 hasta M3
- H4 hasta M4

Datos similares se repiten dentro de la fila de la factura #125. Podemos satisfacer la necesidad de la atomicidad de la PFN simplemente: separando cada item de estas listas en su propia fila.

Figure A-2: cuadrícula allanada de las ordenes

orders.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Invoice No.	Date	Cust. No.	Cust. Name	Cust. Address	Cust. City	Cust. State	Item ID	Item Descri	Item Qty.	Item Price	Item Total	Order Total Price
2	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	563	56" Blue Fre	4	$ 3.50	$ 14.00	$ 82.00
3	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	851	Spline End (	32	$ 0.25	$ 8.00	$ 82.00
4	125	9/13/2002	56	Foo, Inc.	23 Main St., Thorpleburg	Thorpleburg	TX	652	3" Red Free	5	$ 12.00	$ 60.00	$ 82.00
5	126	9/14/2002	2	Freens R Us	1600 Pennsylvania Avenu	Washington	DC	563	56" Blue Fre	500	$ 3.50	$ 1,750.00	$ 10,750.00
6	126	9/14/2002	2	Freens R Us	1600 Pennsylvania Avenu	Washington	DC	652	3" Red Free	750	$ 12.00	$ 9,000.00	$ 10,750.00

Me parece estar oyendo a todos diciendo: estamos tratando de reducir las duplicaciones y aquí estamos introduciendo mas! Vean cuanta duplicidad en los datos de los clientes!

No se preocupen. El tipo de duplicidad que hemos introducido hasta este punto será abordada cuando lleguemos a la **Tercera Forma Normal (TFN).** Por favor, sea paciente, esto es un paso necesario dentro del proceso.

Hasta ahora hemos hablado solamente de la mitad de la historia de la PFN. Concretamente hablando, la PFN aborda dos cuestiones:

1.- Una fila de dato no puede contener grupos repetidos de datos similares (atomicidad)
2.- Cada fila tiene que tener un único identificador (o **llave primaria**)

Hemos estado tratando el tema de la atomicidad pero, para centrar nuestra atención en las llaves primarias, daremos un adios a las cuadrículas y moveremos nuestros datos hacia los sistemas de administración de BD relacionales (RDBMS en inglés). Usaremos Microsoft Access para crear la tabla de órdenes, como en la Figura B:

Figura B: tabla de ordenes

orders : Table

order_id	order_date	customer_id	customer_name	customer_addre	customer_city	customer	item_id	item_description	item_qty	item_price	item_total_price	order_total_price
125	9/13/2002	56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX	563	56" Blue Freen	4	$3.50	$14.00	$82.00
125	9/13/2002	56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX	851	Spline End (Xtra	32	$0.25	$8.00	$82.00
125	9/13/2002	56	Foo, Inc.	23 Main St., Th	Thorpleburg	TX	652	3" Red Freen	5	$12.00	$60.00	$82.00
126	9/14/2002	2	Freens R Us	1600 Pennsylva	Washington	DC	563	56" Blue Freen	500	$3.50	$1,750.00	$10,750.00
126	9/14/2002	2	Freens R Us	1600 Pennsylva	Washington	DC	652	3" Red Freen	750	$12.00	$9,000.00	$10,750.00

Record: 6 of 6

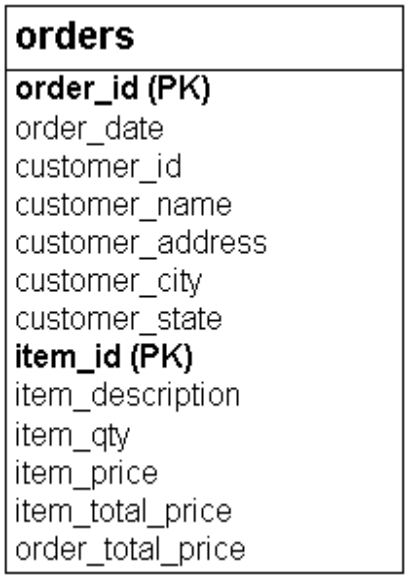
Esto se ve tan bonito como la cuadrícula perola diferencia es que dentro de un RDBMS podemos identificar la **llave primaria**. Una llave primaria es una columna (o grupo de columnas) que de forma única identifica a *cada fila.*

> *Un valor que de forma única identifique una fila se llama **llave primaria**.*
>
> *Cuando este valor está formado por dos o mas columnas, lo llamamos **llave primaria concatenada**.*

Como usted podrá ver en la Figura B, no hay una columna que de forma única identifique cada fila. Sin embargo, si ponemos un número de columnas juntas, podemos satisfacer esta demanda.
Las dos columnas juntas que de forma única identifican a cada fila son **order_id** y **item_id**: no hay otras dos columnas que tengan la misma combinación que **order_id** y **item_id**. Por lo tanto, pueden ser usados como llave primaria de la tabla. Aunque están en dos columnas diferentes de la tabla, son tratados como una sola entidad. La llamaremos **llave primaria concatenada**.

La estructura fundamental de la tabla de órdenes puede ser representada como se muestra en la **Figura C**:

Figura C: estructura de la tabla orders



Identificamos las columnas que forman la llave primaria con la notación **PK**. La Figura C es el comienzo de nuestro Diagrama Entidad Relación (DER).

Nuestro esquema de BD ya satisface los dos requerimientos de la PFN: atomicidad y unicidad. De esta forma, nuestra tabla cumple con los criterios básicos de una BD relacional.

¿Que viene ahora?

Segunda Forma Normal:
Sin Dependencias Parciales en Llaves Concatenadas.

Seguidamente probamos cada tabla si tiene **dependencias parciales sobre la llave concatenada.** Esto significa que para una tabla que tiene una llave primaria concatenada, cada columna de la tabla, que no forma parte de la llave primaria, tiene que depender de la llave concatenada completamente. Si hay alguna columna que solamente dependa de una parte de la llave concatenada, entonces decimos que la tabla completa no cumple la Segunda Forma Normal (SFN) y tenemos que crear otra tabla para rectificar este fallo.

¿Aún no está claro? Para entender esto, tomemos la tabla de órdenes columna a columna y para cada una hagámonos la pregunta:

> ¿Puede esta columna existir sin una de las partes de la llave primaria concatenada?

Si la respuesta es "si" – aunque sea una vez – entonces la tabla falló a la SFN.

Veamos la Figura C otra vez para recordar la estructura de la tabla **orders**.

Figura C: estructura de la tabla orders

orders
order_id (PK)
order_date
customer_id
customer_name
customer_address
customer_city
customer_state
item_id (PK)
item_description
item_qty
item_price
item_total_price
order_total_price

Primero, recordemos el significado de las dos columnas de la llave primaria:

- **order_id** identifica la factura de donde proviene este item
- **item_id** es el identificador único del renglón del inventario. Usted puede pensar que este número es un número de parte, un número de control de inventario, etc.

No analizamos estas columnas (ya que ellas son parte de la llave primaria). Ahora consideraremos las restantes columnas…
order_date es la fecha en que fue hecha la orden. Es obvio que depende de **order_id;** la fecha de la orden tiene que tener una orden si no sería solo una fecha. ¿Pero puede una fecha de orden existir sin un **item_id**?
La respuesta inmediata es si: **order_date** depende de **order_id** pero no de **item_id**. Algunos de ustedes pueden objetar esto partiendo de que usted podría tener una orden fechada sin ítems (una factura vacia). Pero eso no es lo que estamos diciendo: Todo lo que nosotros estamos tratando de establecer aquí es si una orden en particular en una fecha en particular depende de un item en particular. Claro que no. El problema de cómo prevenir que ordenes vacías caigan en la discusión de las "reglas de negocios" y pudieran ser resueltas aplicando

lógica, no es una cuestión a resolver por la Normalización.

Por lo tanto: **order_date** no aprobó la SFN. 

De esta forma, nuestra tabla no ha aprobado la SFN. Continuemos probando las otras columnas. Debemos encontrar todas las columnas que no aprueben la prueba para, entonces, hacer algo especial con ellas.

customer_id es el número de identificación del consumidor que puso la orden. ¿Depende el de **order_id**? No: un consumidor puede existir sin solicitar ninguna orden. Depende el de **item_id**? No: por la misma razón. Esto es interesante: **customer_id** (junto con las otras columnas customer_*) no depende de ninguno de los miembros de la llave primaria. ¿Que hacemos con estas columnas?

No debemos preocuparnos por ellas hasta que lleguemos a la TFN. Por ahora, las marcamos como desconocidas. 

Item_description es la próxima columna que no es por si misma parte de la llave primaria. Esta es la descripción del item que está en el inventario. Es obvio que esta columna depende de **item_id**. ¿Pero puede existir sin una **order_id**?

Si! Un item del inventario (junto con su descripción) puede estar en el almacén por un largo tiempo y nunca ser vendido… Puede existir independientemente de la orden. **Item_description** falló la prueba. 

Item_qty se refiere al número de ítems comprados en una factura en particular. ¿Puede la cantidad existir sin el **item_id**? Imposible: no podemos hablar de la "cantidad de nada" (por lo menos en el diseño de BD). ¿Puede una cantidad existir sin una orden? No: una cantidad que es comprada por medio de una factura no tiene sentido sin la factura. De esta forma, esta columna no viola la SFN: **item_qty** depende de las dos partes de la la llave primaria concatenada.

Item_price es similar a **item_description**. Depende de **item_id** pero no de **order_id**, de aqui que viola la SFN. 

Item_total_price es un caso difícil. De un lado, parece que depende de ambas **order_id** y **item_id**, en cuyo caso pasa la SFN. Por otro lado, es un valor derivado de otros: es el producto de **item_qty** y **item_price**. ¿Que hacer con este campo?

De hecho, este campo no pertenece a nuestra BD. El puede ser fácilmente calculado e incluirlo en la BD sería redundante (y fácilmente podría introducir problemas). Por tanto, lo descartaremos y no hablaremos mas de el.

El valor **order_total_price**, que es la suma de todos los campos **item_total_price** para una orden en particular, es otro valor derivado de otros valores por tanto, lo descartamos.

He aquí el resultado del análisis de la tabla **orders** para la SFN:

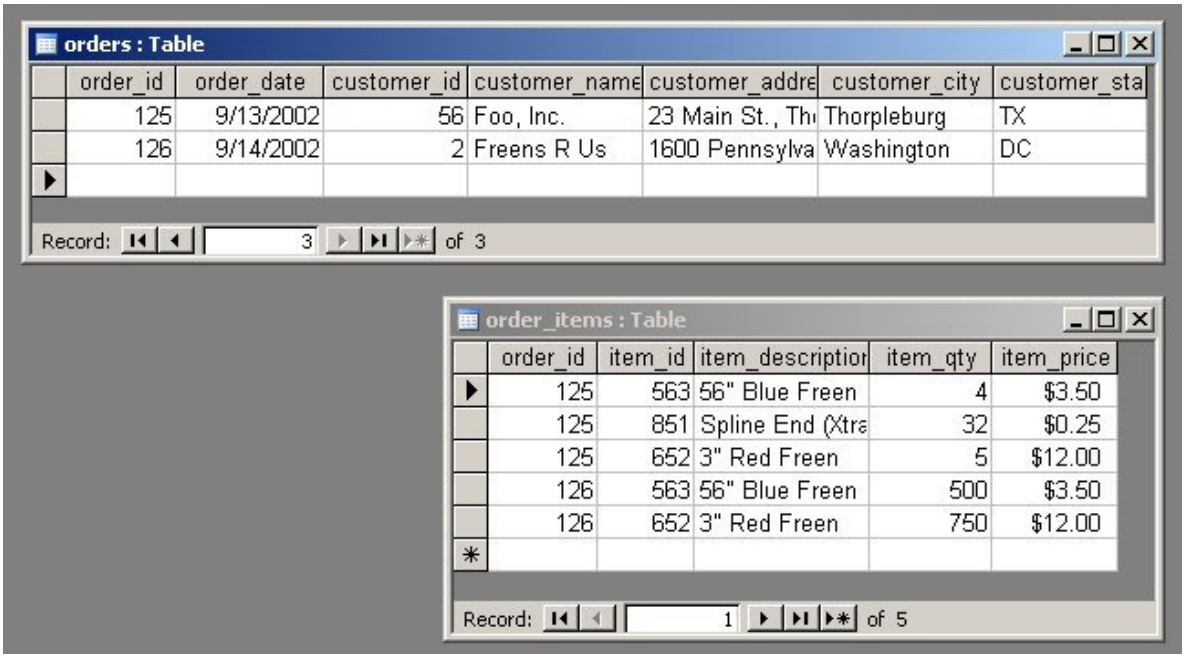
Figura C (revisada)

¿Que hacemos con una tabla que falla la SFN, como esta que analizamos? Primero tomamos la segunda mitad de la llave primaria concatenada (**ítem_id**) y la ponemos en su propia tabla.

Todas las columnas que dependen de **ítem_id,** ya sea completamente o parcialmente, irán con ella a una nueva tabla. Llamaremos a esta nueva tabla **order_items** (ver **Figura D**).

Los otros campos que dependen de la primera parte de la llave primaria (**order_id**) y los que no estamos seguros, se quedan donde están.

Figura D: tabla orders y tabla order_items

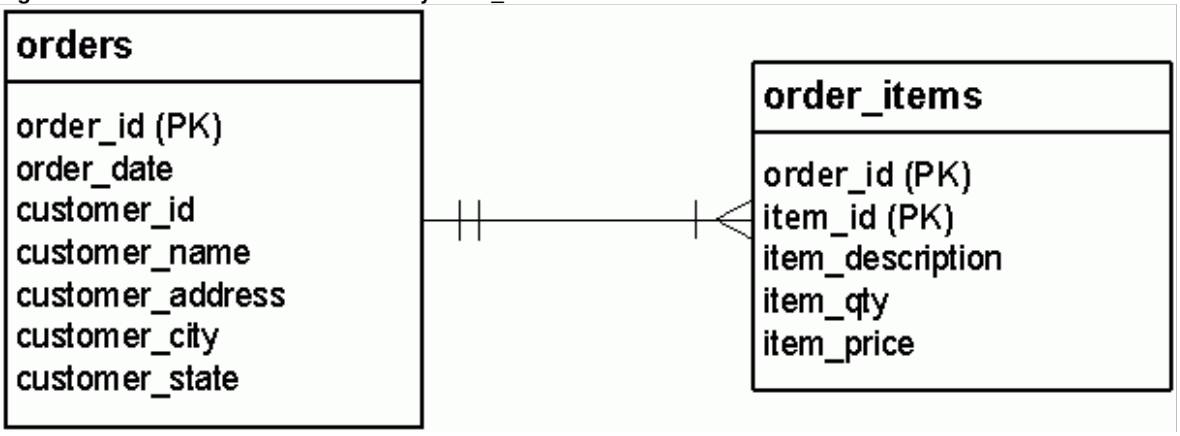


Hay varias cosas que destacar:

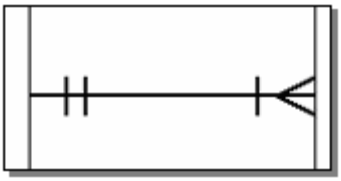
1. Hemos traído una copia de la columna **order_id** de la tabla **order_items**. Esto permite a cada **order_item** "recordar" a cual orden pertenece.
2. La tabla **orders** tiene menos columnas que antes.
3. La tabla **orders** ya no tiene una llave primaria concatenada. La llave primaria consiste ahora en una sola columna, **order_id**.
4. La tabla **order_items** si tiene una llave primaria concatenada.

Esta es la estructura de la tabla (**Figura E**):

Figura E: estructura de las tablas orders y order_items



Si usted es nuevo en esto de los Diagramas Entidad Relación, preste especial atención a la línea que conecta estas dos tablas. Esta línea significa,



Cada orden puede ser asociada con cualquier número de **order_items,** y **por lo menos** uno;

Cada **order_item** está asociado con una orden y **solo** una;

Hay otras formas de representar estas relaciones tabla a tabla; en este caso yo he usado una de las muchas formas estándar.

Segunda Forma Normal: Fase II

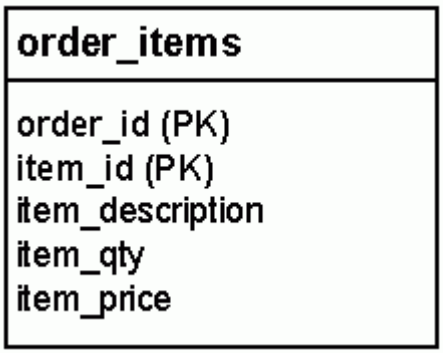
Pero, un momento, hay mas!

Recuerde que la SFN solo se aplica a tablas con llave primaria concatenada. Ahora que **orders** tiene una simple columna como llave primaria, pasa la SFN. Felicidades!
order_items, sin embargo, aún tiene una llave primaria concatenada. Tenemos que pasarla una vez más por el análisis de la SFN y ver si es capaz de pasarlo. Hacemos la misma pregunta que hicimos antes:

> ¿*Puede esta columna existir **sin** una o la otra parte de la llave primaria concatenada*?

Primero, nos remitimos a la **Figura F**, para que nos recuerde la estructura de la tabla **order_items**.

Figura F



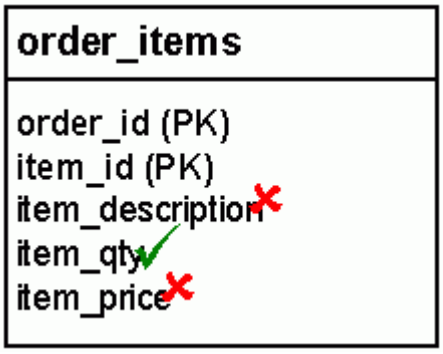
Ahora considere las columnas que no son parte de la llave primaria...
item_description depende de **item_id**, pero no de **order_id**. De manera que (sorpresa), esta columna, una vez mas falla la SFN.

item_qty depende de ambos miembros de la llave primaria, por lo que no viola la SFN.

item_price depende de **item_id** pero no de **order_id**, de manera que si viola la SFN.

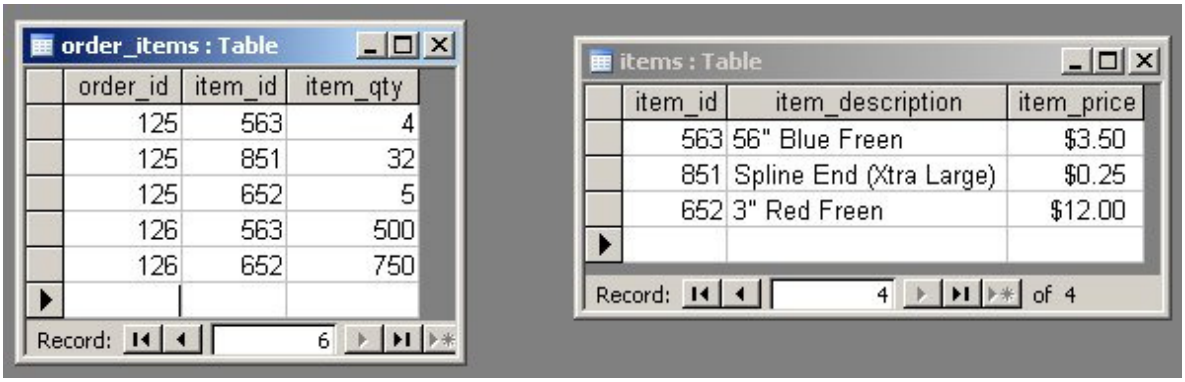
Debemos sentirnos bien ahora. Aquí está el diagrama de la tabla mencionada:

Figura F (revisada):



De esta forma, con los campos que fallaron la SFN, creamos una nueva tabla que llamaremos **items**:

Figura G: tablas order_items y items



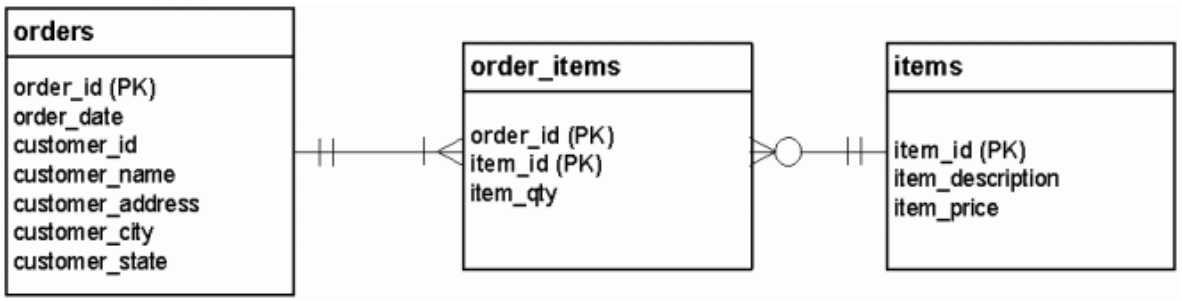
Pero, hay algo mal. Cuando hicimos nuestro primer pase por la prueba de la SFN, sacamos todos los campos que dependían de **ítem_id** y los pusimos en una tabla nueva. Esta vez, estamos seleccionando solamente los campos que fallaron la prueba, o sea, **item_qty** se queda donde está. ¿Por qué? ¿Cuál es la diferencia ahora?

La diferencia consiste es que en el primer pase, quitamos la llave **ítem_id** de la tabla **orders** conjuntamente, debido a la relación uno-a-muchos entre las tablas **orders** y **order_items.** Por tanto, el campo **ítem_qty** tiene que seguir a **ítem_id** hacia la nueva tabla.

En el segundo pase, **item_id** no fue quitado de la tabla **order_items** debido a la relación *muchos-a-uno* entre **order_items** y **ítems.** Por lo tanto, ya que **ítem_qty** no viola la SFN en esta ocasión, le es permitido quedarse en la tabla con las dos partes de la llave primaria de las que depende.

Esto queda claro con el Nuevo DER. Aquí se ve como la tabla **ítems** encaja dentro de todo el esquema de la BD:

Figura H:



La linea que conecta las tablas **items** y **order_items** significa lo siguiente:

> *Cada item puede ser asociado con cualquier número de líneas de cualquier número de factura, incluso cero;*
> *cada **order-item** está asociado con un item, y **solo** uno.*

Estas dos líneas son ejemplos de relaciones de uno a muchos. Esta estructura de tres tablas, considerada como una unidad, es la forma en que expresamos la relación muchos-a-muchos:

> *Cada orden puede tener muchos items; cada item puede pertenecer a muchas órdenes.*

Note que esta vez, no traeremos una copia de la columna **order_id** a la tabla nueva. Esto es asi porque los ítems individuales no necesitan tener reconocimiento de las órdenes a las que ellos pertenecen. La tabla **order_items** tiene el cuidado de recordar esta relación entre las columnas **order_id** y **ítem_id**. Tomadas juntas estas dos columnas conforman la llave primaria de **order_items** pero tomadas separadamente son las llaves externas o los punteros a otras filas en otras tablas. Hablaremos mas de la llaves externas cuando lleguemos a la TFN.

Hay que destacar que nuestra tabla nueva no tiene llave primaria concatenada, de esta forma, pasa la SFN. Hasta aquí, hemos logrado alcanzar la SFN!

Tercera Forma Normal:
Sin Dependencia de Atributos que nos son llaves

Al fin, retornamos al problema de la repetición de la información de los Consumidores. Como está nuestra BD ahora, si un consumidor hace mas de una orden, tenemos que introducir todas esas informaciones de contactos de los consumidores una vez mas. Eso sucede porque hay columnas en la tabla **orders** que dependen de "atributos que no son llaves".

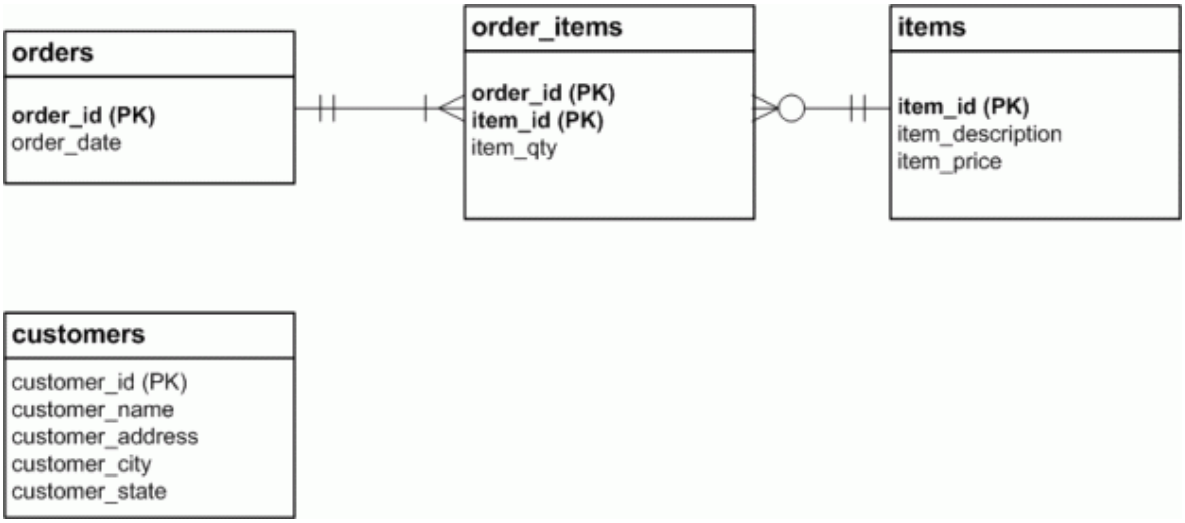
Para entender mejor este concepto, considere la columna **order_date.** ¿Puede ella existir independientemente de la columna **order_id**? *No*: una fecha de orden no tiene sentido sin una orden. **order_date** digamos que depende de un atributo llave (**order_id** es el "atributo llave" porque es la llave primaria de la tabla).

¿Qué hay con **customer_name** – puede existir por si solo, fuera de la tabla **orders**?

Si: Es completamente razonable hablar de un consumidor sin necesidad de hablar de ordenes o facturas. Lo mismo pasa con **customer_address**, **customer_city** y **customer_state.** Estas cuatro columnas actualmente dependen de **customer_id,** que no es una llave en esta tabla (es un atributo que no es llave).

Estos campos pertenecen a su propia tabla, con **customer_id** como llave primaria (ver Figura I).

Figura I:

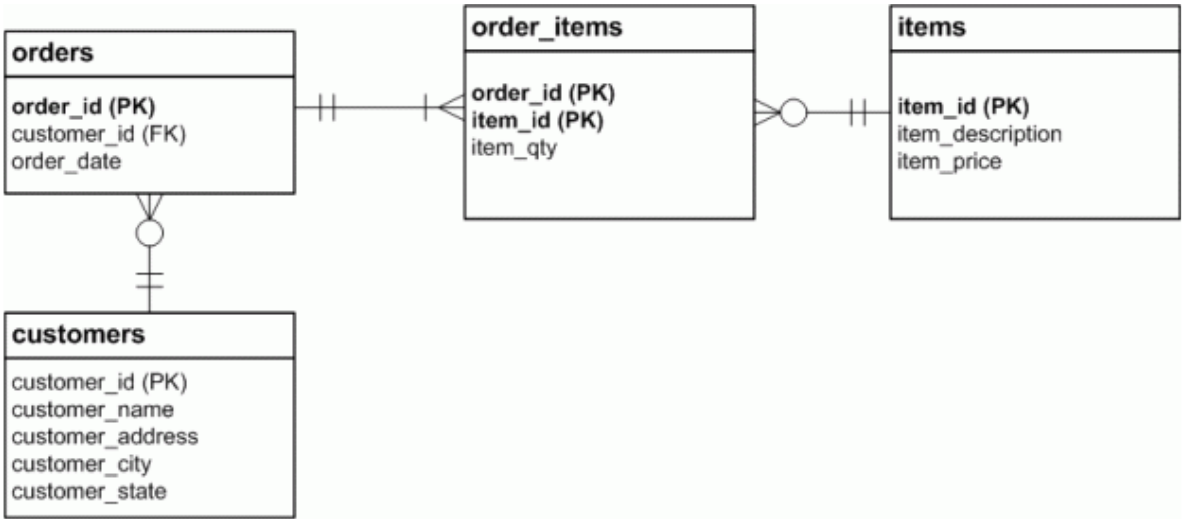


Sin embargo, usted notará en la Figura I que hemos cortado la relación entre la tabla **orders** y los datos del Consumidor que antes estaban en ella.

Pero esto no se quedará asi.

Tenemos que restaurar la relación creando una entidad llamada **llave externa** (indicada en nuestro diagrama como **(FK)**) en la tabla **orders**. Una llave externa es, en esencia, una columna que apunta a la llave primaria en otra tabla. La **Figura J** muestra esta relación, y muestra completamente nuestro DER:

Figura J: DER Final



La relación que ha sido establecida entre las tablas **orders** y **customers** puede ser expresada de la siguiente manera:

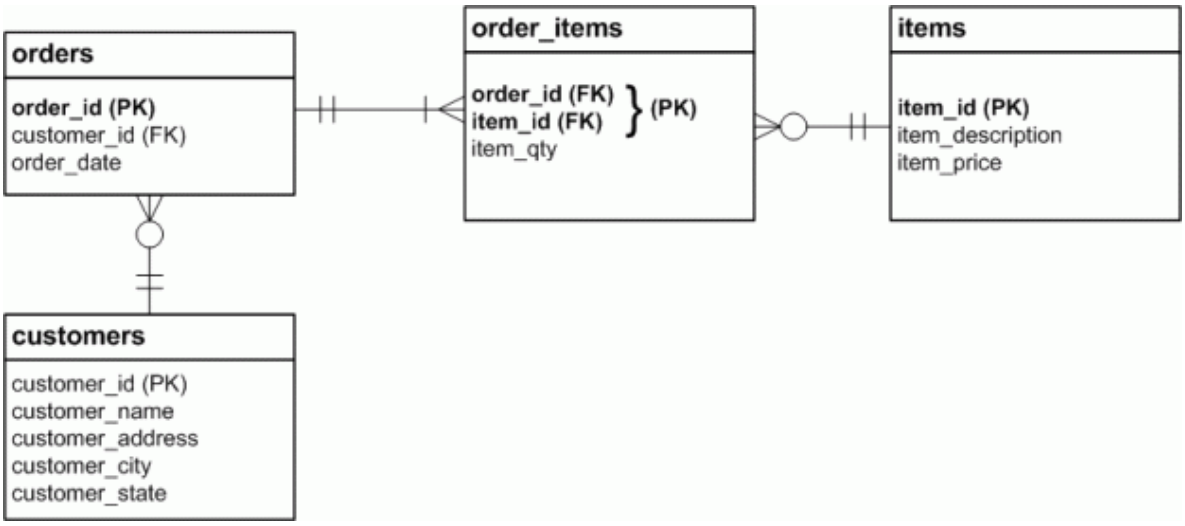
> *cada orden es hecha por un, y **solo** un consumidor;*
> *cada consumidor puede hacer cualquier número de órdenes, incluso cero.*

Una puntualización final...

Notará que las columnas **order_id** y **item_id** en **order_items** cumplen una doble función: no solamente como llave primaria (concatenada)**,** sino que también**,** individualmente, sirven como llaves externas de las tablas **orders** y **ítems** respectivamente.

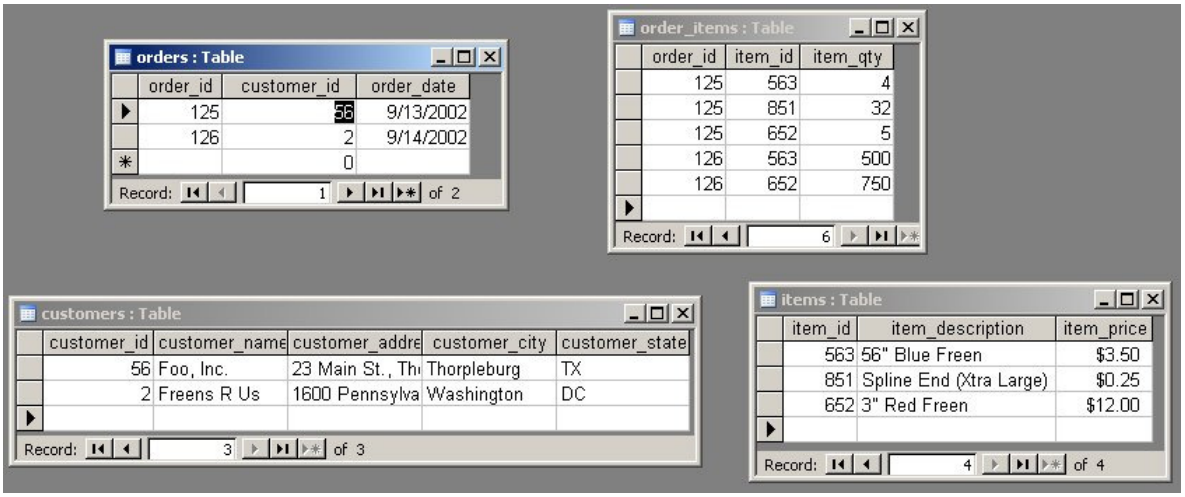
La **Figura J.1** muestra este hecho, y nuestro DER completo:

Figura J1: DER Final



Y finalmente, vemos como quedan los datos en cada una de las cuatro tablas. Note que la TFN remueve mejor las columnas de la tabla, que las **filas**.

Figura K:



Referencias para Futuras Lecturas

Huelga decir que, hay mucho mas de que hablar en este tema. Si usted quiere leer mas sobre la teoría y la práctica de las Tres Formas Normales, aquí tiene algunas sugerencias:

The Art of Analysis, por Dr. Art Langer, dedica un espacio considerable a la normalización. Springer-Verlag Telos (Enero 15, 1997) ISBN: 0387949720

Seminario del Dr. Codd 1969 artículo sobre normalización de BD.: www.acm.org/classics/nov95

El artículo sobre normalización de Wikipedia "Wikipedia article on normalization" discute las cinco formas normales: en.wikipedia.org/wiki/Database_normalization