

## Normalización y denormalización

Las relaciones que resultan de la observación del mundo real o de la transformación del esquema E-R elaborado en la etapa del modelo conceptual, pueden presentar algunos problemas, derivados de fallas en la percepción del sistema, en el diseño del esquema E-R o en el paso al modelo relacional; entre estos problemas cabe destacar los siguientes:

- Incapacidad para almacenar ciertos hechos.
- Redundancia, y por lo tanto, posibilidad de incoherencias.
- Ambigüedades.
- Pérdida de información (aparición de registros huérfanos).
- Pérdida de dependencias funcionales, es decir, de ciertas restricciones de integridad que dan lugar a interdependencias entre los datos.
- Aparición, en la base de datos, de estados que no son válidos en el mundo real, es decir, anomalías de inserción, modificación y eliminación.

### La necesidad de un método formal de diseño relacional

Veamos un ejemplo de un modelo de datos inadecuado y sus implicancias en el procesamiento de datos.

**FACTURA**

No. Factura	Nombre_Cliente	Fecha	Dirección	Distrito	Monto
001-01937	Comercial 2001 SAC	12/05/89	Av. La Paz 462	Miraflores	1519.00
001-01938	Andrés Villa	26/11/89	Calle 7 897	San Borja	2350.00
001-01939	Javier Arévalo	16/03/90	Jr. Cerro 123	Pueblo Libre	576.00
001-01940	Comercial 2001 SAC	26/11/90	Av. La Paz 462	Miraflores	1256.00
001-01941	Juan Acevedo Luque	13/12/90	Av. Arica 1301	Breña	1387.00
001-01942	Manuel Miranda SAC	17/12/90	Mz.C Lt31	Breña	2657.00
001-01943	Comercial 2001 SAC	21/12/90	Av. La Paz 462	Miraflores	5643.00

#### Redundancia

En el modelo observamos gran cantidad de **redundancia** ya que la Dirección del Cliente se repite con cada ocurrencia de facturas giradas a nombre de dicho cliente; algo similar ocurre con el Distrito.

Si el cliente *Comercial 2001 SAC* tiene registradas 500 facturas, su dirección y el distrito también estarán registradas 500 veces.

#### Anomalía de inserción

Si tenemos que registrar una nueva factura para un cliente ya registrado, debemos volver a ingresar su dirección y distrito, con el riesgo de digitar incorrectamente dichos datos. En este caso, el mismo cliente aparecería con direcciones distintas.

#### Anomalía de modificación

Si un cliente cambia de dirección, el cambio debe hacerse en cada una de las ocurrencias de dicho cliente. De manera inadvertida podemos cambiar un dato del cliente sin hacerlo en el resto de las ocurrencias del mismo cliente, lo que da lugar a incoherencias.

### Anomalía de eliminación

Si eliminamos una factura, y ésta es la única factura registrada para un cliente, se perderán todos los datos del cliente.

### Dependencia funcional

La teoría de la normalización se basa en el concepto de dependencias, hasta el punto que actualmente se la conoce también como **Teoría de las dependencias**.

La existencia de una dependencia no se puede demostrar, pero si afirmar por observación del mundo real que se trata de modelar.

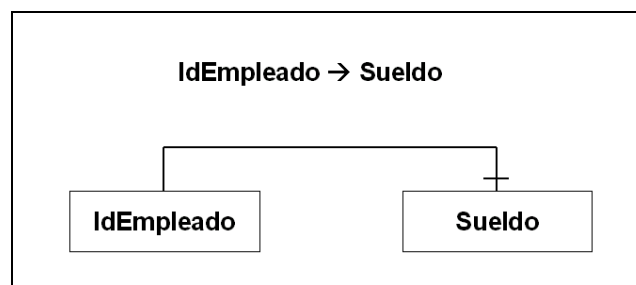
Las dependencias nos muestran algunas interrelaciones importantes existentes entre los atributos del mundo real, cuya semántica (significado) tratamos de incorporar a nuestra base de datos; son, por tanto, invariantes en el tiempo, siempre que no cambie el mundo real del cual proceden.

En el proceso de normalización es fundamental empezar identificando todas la dependencias funcionales del universo del caso cuyo diseño estamos realizando, y preocuparnos de conservar las dependencias funcionales a lo largo del proceso, de modo que el esquema relacional resultante de la normalización tenga las mismas dependencias funcionales que el esquema de partida.

Al intentar dibujar las relaciones entre datos, se debe conocer qué atributos son dependientes de otros atributos.

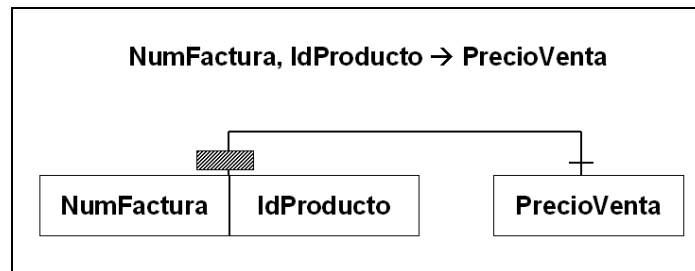
La frase funcionalmente dependiente se debe interpretar de la siguiente manera:

Si Sueldo es funcionalmente dependiente de IdEmpleado, entonces IdEmpleado identifica a Sueldo. En otras palabras, si conocemos el valor del código del empleado IdEmpleado, podemos identificar el valor del Sueldo asociado con éste.



## Dependencia funcional completa

Se dice que un atributo B tiene **dependencia funcional completa** o plena de A, si depende funcionalmente de A, pero no depende de ningún subconjunto de A.



El precio al que se factura un producto (PrecioVenta) depende de la combinación Factura (NumFactura) y Código del Producto (IdProducto). PrecioVenta no depende de IdProducto porque es un atributo distinto al Precio de Lista, y tampoco depende solo del atributo NumFactura. Por lo tanto, PrecioVenta tiene dependencia funcional completa de la combinación de atributos NumFactura e IdProducto.

En el caso que el atributo B depende de un subconjunto de A, y no de A, se dice que B tiene **dependencia funcional parcial** respecto a A.

## Descriptores equivalentes

Se dice que dos descriptores X e Y son equivalentes, si se cumple que:

$$X \rightarrow Y$$

$$Y \rightarrow X$$

Lo que puede representarse como:

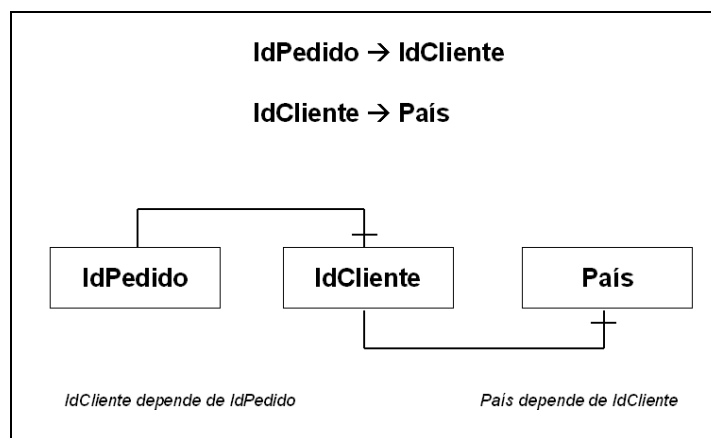
$$X \leftrightarrow Y$$

Por ejemplo:

$$\text{IdProducto} \leftrightarrow \text{NombreProducto}$$

## Dependencia funcional transitiva

Se dice que el atributo C tiene dependencia funcional transitiva respecto del atributo A, a través del atributo B si: B depende de A, y a su vez, C depende de B.



## Normalización

La **normalización** es un proceso que permite reemplazar las relaciones complejas entre los datos por relaciones más sencillas de tal manera que se facilita la comprensión y el desarrollo del procesamiento de datos.

La idea básica de la normalización es reducir una estructura de datos compleja (multidimensional) a un grupo de entidades y relaciones más simples.

Los objetivos principales de la normalización son facilitar la actualización de la base de datos, eliminar las redundancias y las anomalías de actualización, de manera tal que se maneje adecuadamente la integridad y consistencia de los datos. El modelo normalizado finalmente debe corresponder y reflejar la realidad a partir de la cual se diseñó.

La integridad de datos, se puede asociar en general a restricciones que permitan asegurar que los cambios o actualizaciones que efectúan los usuarios en la base de datos, no resulten en una pérdida de consistencia de los datos, es decir se debe contar con mecanismos que permitan proteger la base de datos de modificaciones indeseadas.

El concepto de normalización de los datos tuvo su origen en la teoría relacional numérica y fue planteado formalmente por Edgar F. Codd.

Codd encontró en esta teoría una forma elegante de presentar un método para el trabajo de modelar los datos; ésta es la razón por la cual se debe seguir estos procedimientos, llamados formas normales, de manera rígida y en forma ordenada. Esto quiere decir que primero se debe pasar a la Primera Forma normal, luego a la Segunda Forma Normal luego a la Tercera y así sucesivamente. No se debe alterar esta secuencia.

Codd planteó inicialmente la Primera Forma Normal (1FN), la Segunda Forma Normal (2FN) y la Tercera Forma Normal (3FN); posteriormente en 1971 Heath reformuló la 3FN y en 1977 Fagin definió la Cuarta Forma Normal (4FN).

La presentación formal de la teoría de la normalización exige un bagaje matemático, con sus correspondientes algoritmos, y es solo indispensable en una formación académica universitaria. No se pretende desarrollar esta formalización, si no más bien presentar estas ideas de una manera intuitiva para que los usuarios puedan comprender esta teoría de manera muy simple y práctica.

## Primera Forma Normal (1FN)

Una entidad se encuentra en la primera forma normal (1FN) si:

- Todos sus atributos son simples (no descomponibles), y
- No tiene grupos repetitivos.

Un atributo que forma parte de un grupo repetitivo es aquel que no permite presentar a la entidad en forma plana. Esto quiere decir, que no se puede dibujar a la entidad en un plano debido a que puede admitir varios valores en una misma instancia de la entidad; esto es en una misma celda. Por ejemplo, la entidad FACTURA.

<b>Factura</b>			<b>No. Factura</b> 001-08937		
			<b>No. Guía</b> 001-09956		
			<b>Fecha</b> 28 diciembre 2002		
<b>Cliente</b> Comercial Arenales S.A.C		<b>Código</b> CAR0547		<b>RUC</b> 10201092626	
<b>Código</b>	<b>Descripción</b>	<b>Unidad</b>	<b>Cantidad</b>	<b>Precio Unitario</b>	<b>Subtotal</b>
P00003	CARAMELOS FRUTAS ARCOR	Pqte. 520 gr	300	2.20	660.00
P00007	CHOCOLATE DOÑA PEPA FIELD	Pqte. X 6 Unidades	1000	3.30	3300.00
P00010	WAFER CHOCOLATE FIELD	Pqte. X 9 Unidades	500	1.00	500.00
P00012	CHOCOLATE MOSTRO FIELD	Pqte. X 6 Unidades	500	2.20	1100.00
<b>Total Venta</b>					5560.00
<b>IGV</b>					1000.80
<b>Monto Factura</b>					6560.80

FACTURA tiene los siguientes atributos:

- Número de la Factura (NumFac)
- Número de la Guía de Remisión (NumGuia)
- Fecha de Emisión (Fecha)
- Nombre del Cliente (NomCli)
- Código del Cliente (IdCliente)
- RUC del Cliente (RUC)
- Código del Producto (IdProducto)
- Descripción del Producto (Descripcion)
- Unidad de Medida (Unidad)
- Cantidad Despachada (Cantidad)
- Precio Unitario (Precio)
- Subtotal del Producto (Subtotal)
- Total de la Venta (TotalVenta)
- Monto del IGV (IGV)
- Monto de la Factura (Monto)

El siguiente dibujo muestra que la FACTURA no se puede representar en un plano, salvo que presentemos los datos redundantes.

NumFac	...	IdProducto	Descripcion	Unidad	Cantidad	Precio	Subtotal	...	Monto
001-08937		P00003	Caramelo frutas	Pqte. 520 gr	300	2.20	660.00		6560.80
		P00007	Chocolate Doña	Pqte. X 6	1000	3.30	3300.00		
		P00010	Wafer Chocolate	Pqte. X 9	500	1.00	500.00		
		P00012	Chocolate Mostro Field	Pqte. X 6 Unidades	500	2.20	1100.00		

Como se observa en el dibujo, la entidad FACTURA no se encuentra en 1FN ya que tiene un grupo repetitivo formado por los atributos de los productos consignados en la FACTURA. Por lo tanto, la entidad FACTURA se encuentra en una **forma no normalizada** (FNN).

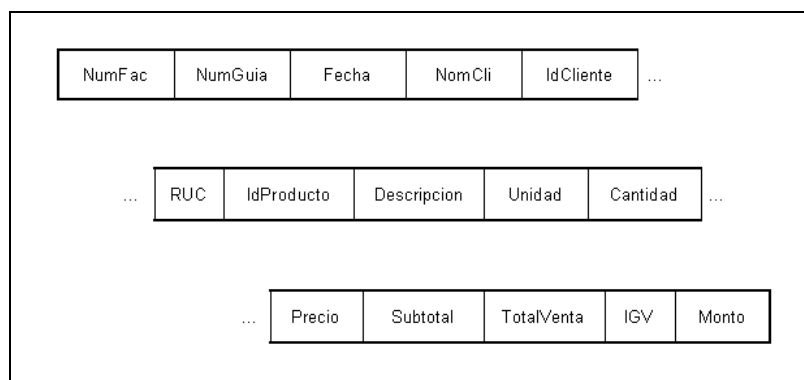
## Cómo obtener la Primera Forma Normal

Para obtener la 1FN procederemos de la siguiente manera:

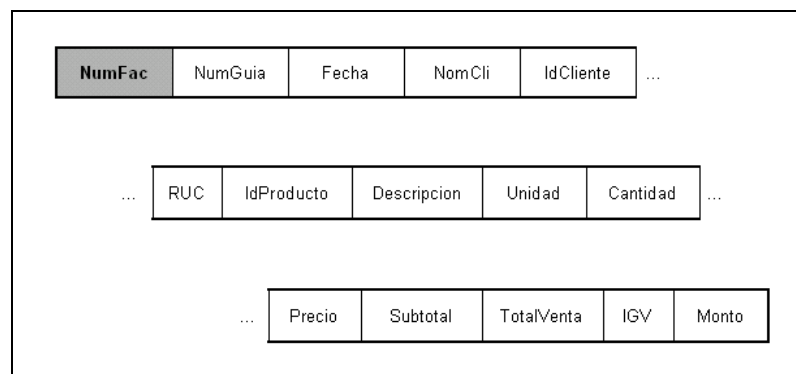
1. Identificar la clave primaria en la entidad compleja
2. Identificar los grupos repetitivos en la entidad compleja.
3. Reescribir la entidad compleja original quitándole los grupos repetitivos.
4. Crear una nueva entidad con cada uno de los grupos repetitivos
5. Identificar la clave primaria en las nuevas entidades.
6. Añadir la clave primaria de la entidad original a las nuevas entidades.

## Paso 1: Identificar la clave primaria en la entidad compleja

- Representar la entidad compleja con todos sus atributos.



- Utilizando los criterios de Unicidad y Minimidad, identificar la clave primaria. Identificamos al atributo NumFac como la PK de la entidad compleja.



## Paso 2: Identificar los grupos repetitivos en la entidad compleja

- Para identificar los grupos repetitivos tratar de representar los atributos de la entidad en una tabla de filas y columnas donde cada fila es una instancia de la entidad, y cada columna representa un atributo. Si para una instancia de la entidad se tienen múltiples valores para una celda, entonces el atributo en dicha celda es parte de un grupo repetitivo.

NumFac	...	IdProducto	Descripcion	Unidad	Cantidad	Precio	Subtotal	...	Monto
001-08937		P00003	Caramelo frutas	Pqte. 520 gr	300	2.20	660.00		6660.80
		P00007	Chocolate Doña	Pqte. X 6	1000	3.30	3300.00		
		P00010	Wafer Chocolate	Pqte. X 9	500	1.00	500.00		
		P00012	Chocolate Mostro Field	Pqte. X 6 Unidades	500	2.20	1100.00		

- En nuestro caso, el grupo repetitivo está formado por los atributos: IdProducto, Descripcion, Unidad, Cantidad, Precio y Subtotal.

### Paso 3: Reescribir la entidad compleja original quitándole los grupos repetitivos

- La entidad compleja FACTURA sin el grupo repetitivo (que momentáneamente denominaremos entidad A) queda como sigue:

Entidad A								
NumFac	NumGuia	Fecha	NomCli	IdCliente	RUC	TotalVenta	IGV	Monto

### Paso 4: Crear una nueva entidad con cada uno de los grupos repetitivos

- Con el grupo repetitivo identificado en la entidad compleja FACTURA formamos la siguiente entidad, que denominaremos entidad B.

Entidad B						
IdProducto	Descripcion	Unidad	Cantidad	Precio	Subtotal	



**Paso 5: Identificar la clave primaria en las nuevas entidades.**

- En la nueva entidad B, identificamos al atributo IdProducto como la PK de la nueva entidad.

Entidad B						
<b>IdProducto</b>	Descripcion	Unidad	Cantidad	Precio	Subtotal	

**Paso 6: Añadir la clave primaria de la entidad original a las nuevas entidades**

- Añadimos a la nueva entidad B la clave primaria de la entidad original FACTURA, de modo tal que no se pierda la relación entre los atributos de la entidad original y de la nueva entidad. En la nueva entidad se obtiene una clave primaria compuesta.

Entidad B						
<b>NumFac</b>	<b>IdProducto</b>	Descripcion	Unidad	Cantidad	Precio	Subtotal

Luego de aplicar estos pasos a la entidad compleja FACTURA obtenemos dos entidades: A y B. Ambas se encuentran en la primera forma normal (1FN); es decir, pueden representarse en forma planar.

Entidad A								
<b>NumFac</b>	NumGuia	Fecha	NomCli	IdCliente	RUC	TotalVenta	IGV	Monto

Entidad B						
<b>NumFac</b>	<b>IdProducto</b>	Descripcion	Unidad	Cantidad	Precio	Subtotal

## Segunda Forma Normal (2FN)

Una entidad se encuentra en la segunda forma normal (2FN) si:

- Ya se encuentra en la primera forma normal, y
- Todos los atributos no claves tienen dependencia funcional completa de la clave primaria.

### Cómo obtener la Segunda Forma Normal

Para obtener la segunda forma normal procedemos de la siguiente manera:

1. Analizar la dependencia funcional de los atributos no claves identificando los que tienen dependencia parcial.
2. Separar los atributos con dependencia parcial de la clave y formar con ellos una nueva entidad.
3. Añadir el atributo clave a la nueva entidad.

#### Paso 1: Analizar la dependencia funcional de los atributos no claves

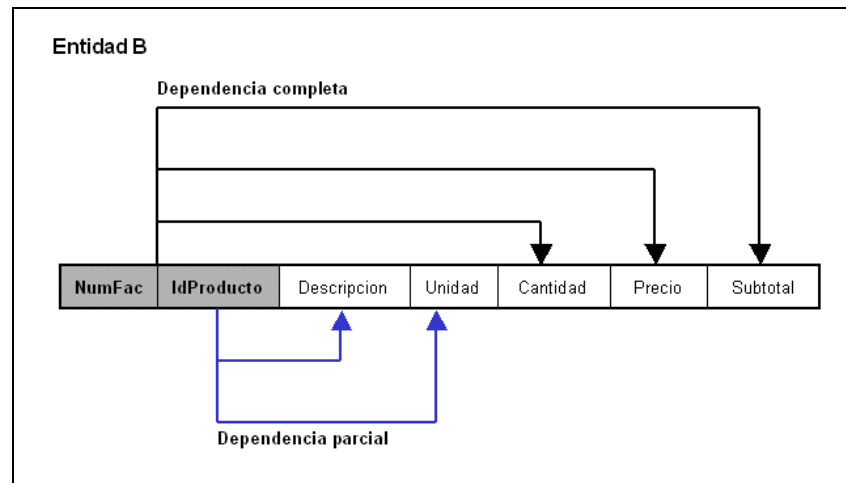
Para la entidad A:

Entidad A								
NumFac	NumGuia	Fecha	NomCli	IdCliente	RUC	TotalVenta	IGV	Monto

- En el caso de la entidad A podemos observar que la PK está formada por un solo atributo, luego es imposible encontrar alguna dependencia parcial en cualquier atributo que se identifique con la PK, ya que al depender funcionalmente de la PK dependerá de toda, y no de una parte de ella.
- Podemos concluir entonces que, si una entidad ya está en 1FN, y su PK es simple (formada por un solo atributo), no tendrá dependencias parciales, entonces ya se encuentra en 2FN. O como corolario, que solo existirá posibilidad de encontrar dependencias parciales en una entidad si su PK es compuesta (formada por más de un atributo). Por lo tanto, la entidad A ya se encuentra en 2FN.

Para la entidad B:

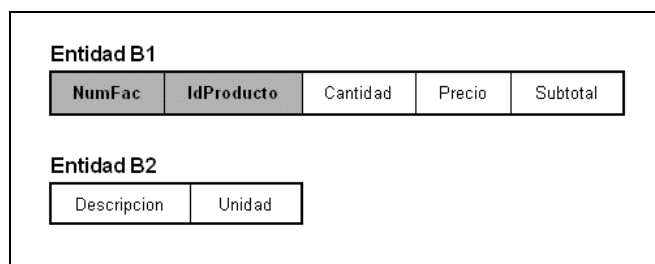
En el gráfico mostrado en la página siguiente podemos observar que los atributos Descripción (nombre y descripción del producto) y Unidad (unidad de medida en la que se comercializa el producto) se pueden identificar solo con conocer el IdProducto (el código del producto debe ser único para cada producto). Luego, no necesitan a toda la PK de la entidad para ser identificados, por lo tanto tienen dependencia funcional parcial de la PK.



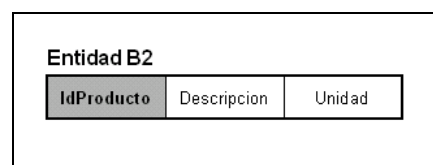
## Paso 2: Separar los atributos con dependencia parcial

Para la entidad B:

- Separamos los atributos Descripcion y Unidad que son los que tienen dependencia parcial de la clave, y formamos una nueva entidad. A las entidades obtenidas de este modo las denominaremos B1 y B2.

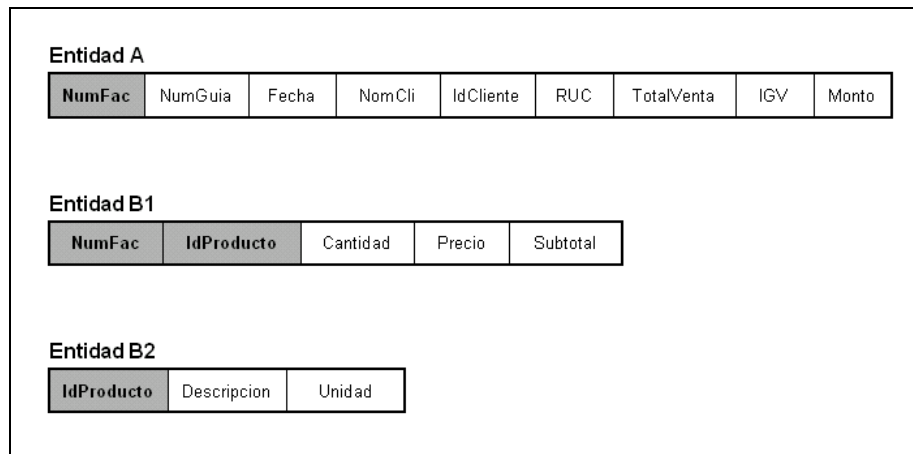


## Paso 3: Añadir el atributo clave a la nueva entidad



- Para que no se pierda la relación entre la entidad B1, y la nueva entidad B2 generada a partir de ella, le añadimos a la nueva entidad el atributo del que dependen parcialmente los atributos en la entidad B1. Luego, le añadimos el atributo IdProducto.

Hasta este momento, a partir de la entidad compleja original FACTURA, se han generado las siguientes entidades:



### Tercera Forma Normal (3FN)

Una entidad se encuentra en la tercera forma normal (3FN) si:

- Ya está en la segunda forma normal, y
- No tiene atributos con dependencia funcional transitiva; es decir, atributos que se pueden obtener a partir de su relación con otros atributos.

#### Cómo obtener la tercera forma normal

1. Analizar la dependencia funcional de los atributos no claves identificando los que tienen dependencia funcional transitiva.
2. Separar los atributos con dependencia transitiva y formar con ellos una nueva entidad.
3. Establecer la clave primaria de la nueva entidad.

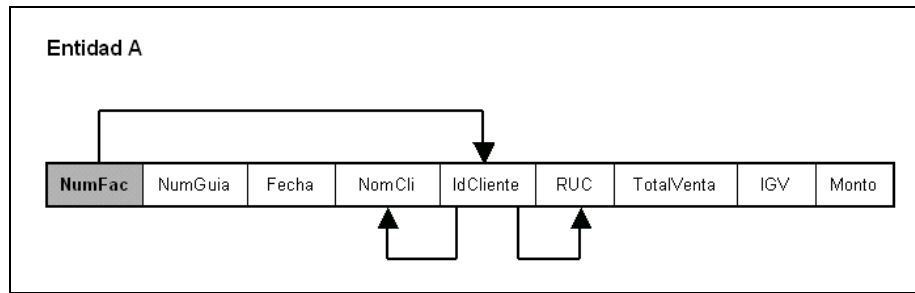
#### Paso 1: Analizar la dependencia funcional transitiva

Para las entidades B1 y B2:

- En nuestro ejemplo observamos que los atributos no claves de las entidades B1 y B2 tienen dependencias funcionales directas, luego no existen dependencias transitivas, por lo tanto, estas entidades ya se encuentran en la tercera forma normal.

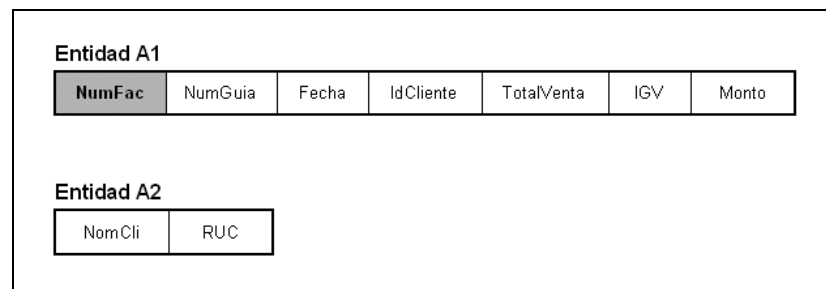
Para la entidad A:

- El nombre del cliente (NomCli) se puede identificar con el código del cliente (IdCliente), a su vez, el IdCliente se puede identificar con el número de factura (NumFac) que es la PK de la entidad; por lo tanto, NomCli tiene dependencia transitiva de NumFac a través de IdCliente.



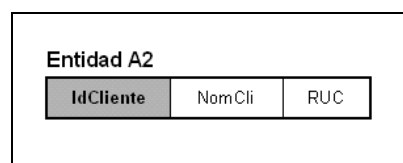
## Paso 2: Separar los atributos con dependencia transitiva

- En la entidad A separamos los atributos NomCli y RUC que tienen dependencia transitiva de la clave NumFac a través de IdCliente. Las nuevas entidades (A1 y A2) quedan como se muestra:



## Paso 3: Establecer la clave primaria de la nueva entidad

- Para la nueva entidad A2, la clave primaria es IdCliente.



Finalmente, a partir de la entidad compleja original FACTURA hemos obtenido las siguientes entidades:

**Entidad A1: FACTURA**

NumFac	NumGuia	Fecha	IdCliente	TotalVenta	IGV	Monto
--------	---------	-------	-----------	------------	-----	-------

**Entidad A2: CLIENTE**

IdCliente	NomCli	RUC
-----------	--------	-----

**Entidad B1: DETALLE\_FACTURA**

NumFac	IdProducto	Cantidad	Precio	Subtotal
--------	------------	----------	--------	----------

**Entidad B2: PRODUCTO**

IdProducto	Descripcion	Unidad
------------	-------------	--------

## Denormalización

Es el proceso de introducir redundancia intencional en las entidades normalizadas para mejorar el rendimiento de las consultas a la base de datos.

En la aplicación de base de datos se debe identificar los procesos críticos para tomar la decisión de denormalizar algunas entidades, en base a criterios como alta frecuencia de consultas, gran volumen de datos, data sumamente volátil, ó data de prioridad explícita.