



## Práctica 08: Normalización de una B. D.

### Fundamentos de Bases de Datos

#### Integrantes:

- Aceves Higareda Mayra Guadalupe  
ahigareda.mayra@ciencias.unam.mx  
310646246
- Badillo Lora Carlos  
carlosb@ciencias.unam.mx  
415083504
- Moreno Ruíz Jesús Fernando  
chuchini@ciencias.unam.mx  
414001967
- Servín Mote Edson  
edson@ciencias.unam.mx  
308292503

**Profesor:** Gerardo Avilés Rosas

**Laboratorio:** Erick Orlando Matla Cruz

---

---

## Forma Normal de las Tablas del caso de uso “El pastor enchilado”

A continuación se presentan las tablas correspondientes al diagrama del caso de uso “El pastor enchilado”, con su respectivo análisis sobre la forma normal en la que se encuentran:

### 1. Horario

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldHorario*, pero no con el de la segunda forma normal. Los atributos sobre el cierre y apertura de la sucursal dependen del día, y el día depende del id del horario, por lo que no todos los atributos dependen de la llave.

*Dependencias funcionales:*

- *nldHorario* → *sNombre*
- *nldHorario* → *nldSucursal*
- *nldHorario* → *sDia* → *tHoraCierre*, *tHoraApertura*

### 2. Direccion

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldDireccion*, pero no con el de la segunda forma normal. El atributo sobre el código postal de la dirección depende la colonia, y la colonia depende del id de la dirección, por lo que no todos los atributos dependen de la llave.

*Dependencias funcionales:*

- *nldDireccion* → *sCalle*, *sColonia*
- *nldDireccion* → *sObservaciones*
- *nldDireccion* → *nldMunicipio*
- *sColonia* → *sCp*

### 3. CMunicipio

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldMunicipio*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldMunicipio* determina al resto.

*Dependencias funcionales:*

- *nldMunicipio* → *sMunicipio*, *nldEstado*

### 4. CEstado

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldEstado*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad

---

---

entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldEstado* determina al resto.

*Dependencias funcionales:*

■ *nldEstado* → *sEstado*

## 5. Proveedor\_Telefono

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave compuesta está conformada por *nldProveedor* y por *nldTelefono*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldProveedor* y *nldTelefono* determinan al resto.

*Dependencias funcionales:*

■ *nldProveedor, nldTelefono* → *nldTipoTelefono*

## 6. Direccion\_Persona

Se cumplen las formas normales por vacuidad.

## 7. Persona

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldPersona*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldPersona* determina al resto.

*Dependencias funcionales:*

■ *nldPersona* → *sApp, aApm, sNombre, dFechaNacimiento, sCorreo*

## 8. Persona\_Telefono

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave compuesta está conformada por *nldPersona* y por *nldTelefono*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldPersona* y *nldTelefono* determinan al resto.

*Dependencias funcionales:*

■ *nldPersona, nldTelefono* → *nldTipoTelefono*

## 9. Telefono

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldTelefono*. Cumple con la segunda forma normal, donde los atributos no llave

---

---

dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldTelefono* determina al resto.

*Dependencias funcionales:*

■ *nldTelefono* → *sTelefono*

## 10. CTipoTelefono

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldTipoTelefono*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldTipoTelefono* determina al resto.

*Dependencias funcionales:*

■ *nldTipoTelefono* → *sTipoTelefono*

## 11. Empleado

Todos sus atributos son atómicos por lo que cumple con la primera forma normal. Los atributos no llave dependen únicamente de la llave. No hay transitividad entre los atributos y, por último, cumple con la forma normal de Boyce-Codd ya que el determinante es llave.

*Dependencias funcionales:*

■ *nldEmpleado* → *dFechaContratacion*, *sRFC*, *nldPersona*, *nldSucursal*, *sNombre*

## 12. Cliente

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldCliente*, pero no con el de la segunda forma normal. El atributo sobre el saldo depende del monedero, por lo que no todos los atributos dependen de la llave.

*Dependencias funcionales:*

■ *nldCliente* → *sMonedero*, *nldPersona*

■ *sMonedero* → *sSaldo*

## 13. DetalleOrden

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldDetalleOrden*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldDetalleOrden* determina al resto.

*Dependencias funcionales:*

■ *nldDetalleOrden* → *nCantidad*, *bDomicilio*, *nldSucursalProducto*, *nldOrden*, *nldCliente*

---

## 14. Orden

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldOrden*, pero no con el de la segunda forma normal. El atributo sobre el día que se realiza la orden depende del atributo fecha, además de que genera redundancia, por lo que no todos los atributos dependen de la llave.

*Dependencias funcionales:*

- *nldOrden* → *dTotal*, *dFecha*, *nldEmpleado*
- *dFecha* → *sDia*

## 15. Promocion

Cumple con el requisito de la primera forma normal (atributos atómicos) donde la llave primaria es *nldPromocion*. Cumple con la segunda forma normal, donde los atributos no llave dependen de la llave. Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos. Cumple con la forma normal de Boyce-Codd, donde *nldPromocion* determina al resto.

*Dependencias funcionales:*

- *nldPromocion* → *sPromocion*, *sDia*, *nldSucursalProducto*

## 16. Sucursal

*Dependencias funcionales:*

- *nldSucursal*, *sNombre* → *nldDireccion*, *nldTelefono*

Esta tabla no cumple con el requisito de la primera forma normal (sólo tener atributos atómicos), ya que tiene una llave compuesta por *nldSucursal* y por *sNombre*, donde podemos tener redundancia al crear la llave. Por ejemplo, si la tabla estuviera poblada con los siguientes valores:

<u>nldSucursal</u>	<u>sNombre</u>	nldDireccion	nldTelefono
1	"0Universidad"	345	98
10	"Universidad"	456	309

Podemos ver que si creamos la llave compuesta, tenemos valores duplicados con el identificador "10Universidad".

## 17. Sucursal\_Producto

Cada uno de los atributos son atómicos por lo que está en primera forma normal. Los atributos no llave dependen de la llave (**nldSucursalProducto**). No existe transitividad entre los atributos y de igual forma el determinante es llave por lo que la tabla está en FNBC.

*Dependencias funcionales:*

- *nldSucursalProducto* → *dPrecio*, *dFechaActualización*, *bActivo*, *nldSucursal*, *nldProducto*

---

---

## 18.Producto

La tabla cumple con la primera forma normal al tener atributos atómicos. Así mismo también cumple la segunda forma normal ya que su atributo no llave (sProducto) depende del atributo llave. Tampoco hay dependencias transitivas por lo que cumple el requisito para la tercera forma normal. También cumple con la forma normal Boyce-Codd puesto que el determinante es llave.

*Dependencias funcionales:*

■ nldProducto → sProducto

## 19.Proveedor

La tabla cumple con la primera forma normal ya que todos sus atributos son atómicos. De igual forma los atributos no llave son dependientes de la llave. No existe transitividad entre los atributos y el determinante es llave. Por lo ante está en la FNBC.

*Dependencias funcionales:*

■ nldProveedor → sProveedor, sRFC, nldDireccion

## 20.Proveedor\_Ingrediente

La tabla cumple con el requisito de sólo poseer atributos atómicos para estar en primera forma normal. Sin embargo, no cumple los requisitos para estar en segunda forma normal (todo atributo no llave depende del atributo llave).

\* **Nota:** Se considera a la tabla **Sucursal** sin normalizar, por lo tanto aún mantenemos como llave foránea a **nldNombre** y **nldSucursal**.

*Dependencias funcionales:*

■ nldProveedor, nldIngrediente → dFecha, dCosto, nCantidad, nldSucursal, nldNombre

■ nCantidad, nldIngrediente → dCosto (La cantidad de un ingrediente en particular determinará el costo de éste).

## 21.Ingrediente\_Producto

La tabla cumple con la primera forma normal al ser todos sus atributos atómicos. De igual forma el atributo no llave **sPorción** es dependiente de la llave primaria compuesta por **nldProducto** y **nldIngrediente**. Tampoco existen dependencias transitivas entre atributos no llave. Por último, se cumple que todo determinante es llave, en este caso la llave primaria compuesta que ya mencionamos antes.

*Dependencias funcionales:*

■ nldProducto, nldIngrediente → sPorcion

---

---

## 22. Ingrediente

La tabla se encuentra en la forma normal de Boyce-Codd puesto que sus atributos son atómicos, el único atributo no llave depende de la llave, no hay dependencias transitivas y el determinante es llave.

*Dependencias funcionales:*

■ nIdIngrediente → sIngrediente

## 23. Salsa\_Producto

Se cumplen las formas normales por vacuidad.

## 24. Salsa\_Ingrediente

La tabla se encuentra en la forma normal de Boyce-Codd puesto que todos sus atributos son atómicos, el único atributo no llave depende de la llave, no hay dependencias transitivas y el determinante es una llave primaria la cual es compuesta.

*Dependencias funcionales:*

■ nIdSalsa, nIdIngrediente → sPorcion

## 25. Salsa

La tabla cumple con los requisitos de todas las formas normales hasta la de Boyce-Codd.

*Dependencias funcionales:*

■ nIdSalsa → sSalsa, nIdPicor

## 26. CPicor

La tabla se encuentra en la forma normal de Boyce-Codd por lo que también cumple con las primeras tres formas normales.

*Dependencias funcionales:*

■ nIdPicor → sPicor

---

---

## Normalización de las Tablas

### 1. Horario

**\*Nota:** consideramos que ya se realizó la normalización de la tabla **Sucursal** que quita el atributo **sNombre**

- **2FN** - Nos conviene que el atributo *sDia* forme parte de la llave, haciéndola compuesta junto con *nldHorario*. Por lo que las dependencias funcionales quedarían de la siguiente manera:

*Dependencias funcionales:*

- *nldHorario, sDia* → *tHoraCierre, tHoraApertura, nldSucursal*

Esto con el propósito de no crear más dependencias y hacer mínimo el conjunto, teniendo en cuenta que no se pierde información.

- **3FN** - No existe transitividad entre atributos.
- **FNBC** - *nldHorario* ya determina al resto de los atributos de la tabla *Horario*, así como *sDia* determina al resto de los atributos de la tabla *Dia*.

### 2. Direccion

- **2FN** - Creamos dos nuevas tablas: *CColonia* y *CodigoPostal*. *CColonia* cuenta con los atributos *nldColonia* (llave primaria de la tabla), *sColonia*, la llave foránea de la tabla *Municipio* (ya que queremos mantener la referencia del municipio al cual pertenece el código postal) y *nldCp* (la llave foránea del código postal), quitando a éste último de la tabla *Direccion*.

*CodigoPostal* tiene los atributos *nldCp* (llave primaria) y *sCp*.

Separamos la tabla colonia de código postal, ya que esto facilita las consultas en la base de datos. Así, las tablas quedarían de la siguiente manera:

Direccion				
nldDireccion	sCalle	sObservaciones	nldColonia	nldMunicipio

CColonia			
nldColonia	sColonia	nldMunicipio	nldCp

CodigoPostal	
nldCp	sCp



- 
- 
- **3FN** - No existe transitividad entre atributos.
  - **FNBC** - *nldDireccion* ya determina al resto de los atributos de la tabla *Direccion*, así como *nldColonia* determina al resto de los atributos de la tabla *CodigoPostal*.

### 3. CMunicipio

- **FNBC** - La tabla ya está normalizada.

### 4. CEstado

- **FNBC** - La tabla ya está normalizada.

### 5. Proveedor\_Telefono

- **FNBC** - La tabla ya está normalizada.

### 6. Direccion\_Persona

- **FNBC** - La tabla ya está normalizada por vacuidad.

### 7. Persona

- **FNBC** - La tabla ya está normalizada.

### 8. Persona\_Telefono

- **FNBC** - La tabla ya está normalizada.

### 9. Telefono

- **FNBC** - La tabla ya está normalizada.

### 10. CTipoTelefono

- **FNBC** - La tabla ya está normalizada.

### 11. Empleado

- **FNBC** - La tabla ya está normalizada.

---

## 12. Cliente

- **2FN** - Creamos una nueva tabla Monedero, con los atributos *nldMonedero*, *sMonedero* y *sSaldo*, quitando a éstos 2 últimos de la tabla Cliente.

Cliente		
<b><i>nldCliente</i></b>	<i>nldPersona</i>	<i>nldMonedero</i>

Monedero		
<b><i>nldMonedero</i></b>	<i>sMonedero</i>	<i>sSaldo</i>

- **3FN** - No existe transitividad entre atributos.
- **FNBC** - *nldCliente* ya determina al resto de los atributos de la tabla *Cliente*, así como *nldMonedero* determina al resto de los atributos de la tabla *Monedero*.

## 13. DetalleOrden

- **FNBC** - La tabla ya está normalizada.

## 14. Orden

- **2FN** - Quitamos el atributo *sDia* de la tabla porque genera redundancia en la información. Podemos determinar el día a partir del atributo *dFecha*. Así, todos los atributos no llave dependen de la llave primaria *nldOrden*.
- **3FN** - Cumple con la tercera forma normal, ya que no existe transitividad entre los atributos.
- Cumple con la forma normal de Boyce-Codd, donde *nldOrden* determina al resto.

## 15. Promocion

- **FNBC** - La tabla ya está normalizada.

## 16. Sucursal

- **1FN** - Quitamos el atributo *sNombre* de la tabla, quedando únicamente como llave primaria *nldSucursal*. Esto implica eliminar las llaves foráneas *sNombre* de la tabla Horario, Empleado, Proveedor\_Ingrediente y Sucursal\_Producto.
- **2FN** - Todos los atributos no clave, ya dependen de una llave primaria.
- **3FN** - No existe transitividad en los atributos.
- **FNBC** - La llave *nldSucursal* determina al resto de los atributos.

## 17. Sucursal\_Producto

- **FNBC** - La tabla ya está normalizada.

---

---

## 18.Producto

- **FNBC** - La tabla ya está normalizada.

## 19.Proveedor

- **FNBC** - La tabla ya está normalizada.

## 20.Proveedor\_Ingrediente

- **1FN** - Al haber actualizado la tabla Sucursal, **sNombre** ya no forma parte de la tabla y sólo queda como llave foránea **nIdSucursal**. Con ésto se cumple que los atributos son atómicos.
- **2FN** - Creamos una nueva tabla en donde se tenga la información del costo de un ingrediente dependiendo de su cantidad. (Suponemos que los proveedores ya tienen **especificadas** las cantidades por las cuales venden sus productos.

Proveedor_Ingrediente_Costo			
<b>nIdProveedorIngredienteCosto</b>	nCantidad	nIdIngrediente	dCosto

- **3FN** - No es necesario hacer nada puesto que no existe transitividad entre los atributos.
- **FNBC** - Se cumple puesto que el determinante es llave primaria.

## 21.Ingrediente\_Producto

- **FNBC** - La tabla ya está normalizada.

## 22.Ingrediente

- **FNBC** - La tabla ya está normalizada.

## 23.Salsa\_Producto

- **FNBC** - La tabla ya está normalizada por vacuidad.

## 24.Salsa\_Ingrediente

- **FNBC** - La tabla ya está normalizada.

---

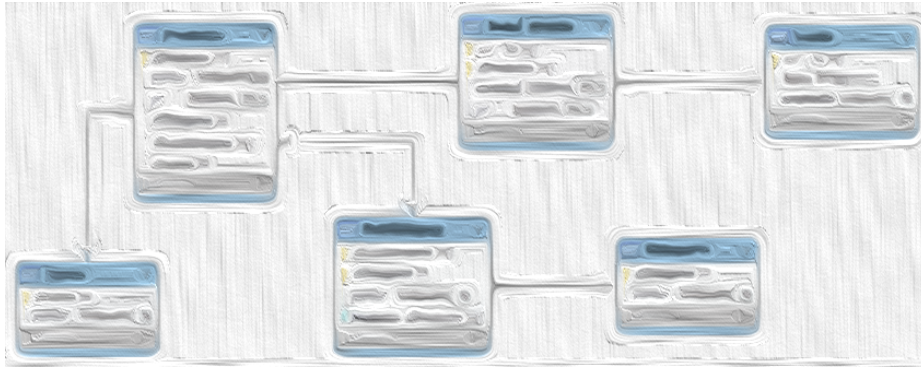
## 25. Salsa

- **FNBC** - La tabla ya está normalizada.

## 26. CPicor

- **FNBC** - La tabla ya está normalizada.

### *Justificación del Diagrama Propuesto*



El diagrama que proponemos es una modificación al diagrama del caso de uso “El pastor enchilado” con el cual hemos estado trabajando, ya que prácticamente consideraba todos los datos que se quieren almacenar en la base de datos, solamente había que normalizar algunas tablas.

Las modificaciones más importantes que realizamos fue separar la colonia y el código postal de la dirección, el monedero y su saldo del cliente, así como determinar el costo total de un ingrediente adquirido con un proveedor a partir de la cantidad solicitada. Todo esto, manteniendo las mismas relaciones que se tenían, adaptadas a las nuevas entidades creadas.

Es una buena solución, puesto que al dividir las entidades (tablas) respecto a las dependencias funcionales, facilitamos las consultas que se realizarán en el futuro en la base de datos.