

Proyecto Final

Proceso de Normalización

Procedimiento

Vamos a normalizar todas las relaciones de nuestro esquema de base de datos de acuerdo a la **Tercera Forma Normal**. Lo primero que haremos es indicar cada relación en nuestra base de datos con su respectivo conjunto de dependencias funcionales. El proceso con el que se obtuvieron dichas dependencias funcionales se encuentra detallado en el documento de Traducción.

NOTA: En el documento los nombres de relación y atributos se escribirán sin tildes, de manera que concuerden con los scripts de SQL.

- Sucursales(numero_sucursal, calle, numero, ciudad, estado)
 - numero_sucursal \rightarrow calle, numero, ciudad, estado

En esta relación las únicas dependencias funcionales no triviales tienen del lado izquierdo a la llave de la relación (numero_sucursal), por lo tanto no hay violaciones a la Tercera Forma Normal.

- Clientes(correo, nombre, apellido_paterno, apellido_materno, puntos, calle, numero, estado, ciudad, telefono, numero_sucursal)
 - correo \rightarrow nombre, apellido_paterno, apellido_materno, puntos, calle, numero, estado, ciudad, telefono, numero_sucursal

De nuevo se presenta el caso de que todas las dependencias funcionales tienen del lado izquierdo una llave y no hay violación a la Tercera Forma Normal.

- Empleados(rfc, nombre, apellido_paterno, apellido_materno, curp, tipo_empleado, tipo_sangre, fecha_nacimiento, calle, numero, estado, ciudad, cuenta_bancaria, numero_seguro, tipo_transporte, licencia, numero_sucursal, salario, bonos, fecha_contratacion)
 - rfc \rightarrow curp
 - curp \rightarrow rfc
 - rfc \rightarrow numero_seguro
 - curp \rightarrow numero_seguro
 - numero_seguro \rightarrow curp
 - numero_seguro \rightarrow rfc
 - rfc \rightarrow nombre, apellido_paterno, apellido_materno, tipo_empleado, tipo_sangre, fecha_nacimiento, calle, numero, estado, ciudad, cuenta_bancaria, tipo_transporte, licencia, numero_sucursal, salario, bonos
 - curp \rightarrow nombre, apellido_paterno, apellido_materno, tipo_empleado, tipo_sangre, fecha_nacimiento, calle, numero, estado, ciudad, cuenta_bancaria, tipo_transporte, licencia, numero_sucursal, salario, bonos

- $\text{numero_seguro} \rightarrow \text{nombre, apellido_paterno, apellido_materno, tipo_empleado, tipo_sangre, fecha_nacimiento, calle, numero, estado, ciudad, cuenta_bancaria, tipo_transporte, licencia, numero_sucursal, salario, bonos}$

En este caso hay que hacer notar que tanto curp como numero_seguro y como rfc son llaves candidatas de esta relación, por lo que de nuevo, al todas las dependencias funcionales tener una de ellas del lado izquierdo, no se presentan violaciones a la 3NF.

■ $\text{Vendedores}(\text{rfc, nombre, telefono})$

- $\text{rfc} \rightarrow \text{nombre, telefono}$

El rfc es una llave, por lo que no hay violaciones a la 3NF en esta relación.

■ $\text{MateriaPrima}(\text{id_articulo, nombre, tipo})$

- $\text{id_articulo} \rightarrow \text{nombre}$
- $\text{nombre} \rightarrow \text{id_articulo}$
- $\text{id_articulo} \rightarrow \text{tipo}$
- $\text{nombre} \rightarrow \text{tipo}$

Como tanto id_articulo como nombre forman una llave para esta relación, todas las dependencias funcionales tienen del lado izquierdo una súper llave, por lo que no hay violaciones a la 3NF.

■ $\text{Inventarios}(\text{id_articulo, numero_sucursal, fecha_compra, precio_unitario, cantidad, fecha_caducidad, rfc_proveedor})$

- $\text{id_articulo, numero_sucursal, fecha_compra} \rightarrow \text{precio_unitario, cantidad, fecha_caducidad, rfc_proveedor}$

Note que las únicas dependencias funcionales tienen a id_articulo , numero_sucursal y fecha_compra del lado izquierdo, y estos atributos forman una llave, por lo que no hay violaciones a la 3NF

■ $\text{Tipo}(\text{id_tipo, nombre})$

- $\text{id_tipo} \rightarrow \text{nombre}$
- $\text{nombre} \rightarrow \text{id_tipo}$

Note que en esta relación tanto nombre como id_tipo son llaves candidatas y por lo tanto no hay violación a la 3NF.

■ $\text{Platillos}(\text{id_platillo, id_tipo, nombre})$

- $\text{id_platillo} \rightarrow \text{id_tipo, nombre}$

Note en esta relación todas las dependencias funcionales tienen una llave del lado izquierdo, por lo que no hay violación a la 3NF.

- IngredientesPlatillo(id_platillo, id_articulo, cantidad)

- id_platillo, id_articulo \rightarrow cantidad

Note en esta relación todas las dependencias funcionales tienen una llave del lado izquierdo, por lo que no hay violación a la 3NF.

- Salsas(nombre_salsa, picor)

- nombre_salsa \rightarrow picor

De nuevo, no hay violaciones a la 3NF

- IngredientesSalsa(nombre_salsa, id_articulo, cantidad)

- nombre_salsa, id_articulo \rightarrow cantidad

No hay violaciones a la 3NF

- Precios(id_platillo, fecha, precio)

- id_platillo, fecha \rightarrow precio

No hay violación a la 3NF.

- Recomendaciones(id_platillo, nombre_salsa)

En este caso sólo tenemos la dependencia funcional trivial (id_platillo, nombre_salsa \rightarrow id_platillo, nombre_salsa), por lo que no hay violaciones a la 3NF.

- PresentacionSalsas(nombre_salsa, tamano)

En esta relación se presenta el mismo caso que en la anterior (sólo tenemos nombre_salsa, tamano \rightarrow nombre_salsa, tamano)

- PreciosSalsas(nombre_salsa, tamano, fecha, precio)

- nombre_salsa, tamano, fecha \rightarrow precio

No hay violaciones a la 3NF.

- Promociones(id_promocion, tipo_descuento, dia, tipo_producto)

- id_promocion, \rightarrow tipo_descuento, dia, tipo_producto

No hay violaciones a la 3NF

- Pedidos(numero_ticket, numero_sucursal, metodo_pago, no_mesa, fecha, correo_cliente)

- numero_ticket \rightarrow numero_sucursal, metodo_pago, no_mesa, fecha, correo_cliente

No hay violaciones a la 3NF

- ComponentesPedido(numero_ticket, id_platillo, id_promocion, nombre_salsa, tamano)

- $\text{numero_ticket} \rightarrow \text{id_platillo}$
- $\text{numero_ticket} \rightarrow \text{nombre_salsa}, \text{tamano}$
- $\text{numero_ticket} \rightarrow \text{id_promocion}$

En este caso, para la tercer forma normal, sólo tenemos que fijarnos en la última de las dependencias multivaluadas, pues es la única que es funcional. También hay que notar que es esta relación, toda llave candidata debe necesariamente contener a los atributos `numero_ticket`, `id_platillo`, `nombre_salsa` y `tamano`, pues no están del lado derecho de ninguna dependencia funcional.

Tomando esto en cuenta, hay que notar que $\text{numero_ticket} \rightarrow \text{id_promocion}$ es una violación a la 3NF, y por lo tanto, tenemos que proceder con el proceso de Normalización.

En este caso, F consta de una sola DF, $F = \{\text{numero_ticket} \rightarrow \text{id_promocion}\}$. Este conjunto ya es mínimo, pues es la única dependencia que podemos usar para el cálculo de cerraduras. Siguiendo el procedimiento de Normalización, para cada DF en F se crea una tabla cuyos atributos sean únicamente los de F :

- `PromocionesPedido(numero_ticket, id_promocion)`, $F = \{\text{numero_ticket}, \text{id_promocion}\}$

Y como no hay una relación con los atributos de una llave candidata, y como

$$\begin{aligned} &\{\text{numero_ticket}, \text{id_platillo}, \text{nombre_salsa}, \text{tamano}\}^+ = \\ &\{\text{numero_ticket}, \text{id_platillo}, \text{nombre_salsa}, \text{tamano}, \text{id_promocion}\} \end{aligned}$$

Creamos otra relación con los atributos `numero_ticket`, `id_platillo`, `nombre_salsa` y `tamano`:

- `X(numero_ticket, id_platillo, nombre_salsa, tamano)`, con sólo dependencias funcionales triviales

Ahora, `X` no tiene dependencias funcionales no triviales, pero aún sigue teniendo redundancia puesto que estamos repitiendo la información de cada platillo de un pedido un número de veces igual al número de salsas que hubo en el pedido y viceversa. Esto no nos parece ideal, por lo que decidimos, únicamente para esta relación, puesto que no es de interés para la base de datos guardar esta información, Hacer uso de la cuarta forma normal.

Iniciamos con las DMV: $\text{numero_ticket} \twoheadrightarrow \text{id_platillo}$ y $\text{numero_ticket} \twoheadrightarrow \text{nombre_salsa}, \text{tamano}$. Ambas son violaciones a la 4NF pues son no triviales y del lado izquierdo no tienen una superllave.

Siguiendo el algoritmo de Normalización, elegimos una de ellas. Tomamos $\text{numero_ticket} \twoheadrightarrow \text{nombre_salsa}, \text{tamano}$. Separamos `X` en dos relaciones, una que contiene a los atributos de esta DMV, y otra con los atributos restantes y los del lado izquierdo de la DMV, que en este caso sólo es `numero_ticket`:

- `SalsasPedido(numero_ticket, nombre_salsa, tamano)`, con una única DMV: $\text{numero_ticket} \twoheadrightarrow \text{nombre_salsa}, \text{tamano}$.
- `PlatillosPedido(numero_ticket, id_platillo)`, con una única DMV: $\text{numero_ticket} \twoheadrightarrow \text{id_platillo}$.

En este caso ya no quedan ni dependencias funcionales ni DMV no triviales (pues las dos que teníamos ahora tienen a todos los atributos de sus respectivas relaciones), y por lo tanto, la cuarta forma normal de `X` es la anterior. Así, `DetallesPedido` queda separada en tres relaciones:

- SalsasPedido(numero_ticket, nombre_salsa, tamaño)
- PlatosPedido(numero_ticket, id_plato)
- PromocionesPedido(numero_ticket, id_promocion)