Proyecto Final
Proceso de Normalización

Procedimiento

Vamos a normalizar todas las relaciones de nuestro esquema de base de datos de acuerdo a la **Forma Normal de Boyce Codd**, aunque si vemos que se pierden muchas dependencias funcionales, lo que haremos será pasarlo mejor 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 →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 Forma Normal de Boyce Codd.

- Clientes(correo, nombre, apellido_paterno, apellido_materno, puntos, calle, numero, estado, ciudad, telefono, numero_sucursal)
  - correo→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 Forma Normal de Boyce Codd.

- 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 →curp
  - curp→rfc
  - rfc→numero_seguro
  - curp→numero_seguro
  - numero_seguro→curp
  - numero_seguro→rfc
  - rfc→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→nombre, apellido_paterno, apellido_materno, tipo_empleado, tipo_sangre, fecha_nacimiento, calle, numero, estado, ciudad, cuenta_bancaria, tipo_transporte, licencia, numero_sucursal, salario, bonos
  - numero_seguro→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, tienen una super llave del lado izquierdo,y por lo tanto, no se presentan violaciones a la BCNF.

- Vendedores(rfc, nombre, telefono)
  - rfc→nombre,telefono

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

- MateriaPrima(id_articulo, nombre, tipo)
  - id_articulo→nombre
  - nombre→id_articulo
  - id_articulo→tipo
  - nombre→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 BCNF.

- Inventarios(id_articulo, numero_sucursal, fecha_compra, precio_unitario, cantidad, fecha_caducidad, rfc_provedor)
  - id_articulo, numero_sucursal, fecha_compra→ precio_unitario, cantidad, fecha_caducidad, rfc_provedor

  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 BCNF

- Tipo(id_tipo, nombre)
  - id_tipo→nombre
  - nombre→id_tipo

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

- Platillos(id_platillo, id_tipo, nombre)

- id_platillo→id_tipo, nombre

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

- IngredientesPlatillo(id_platillo, id_articulo, cantidad)
  - id_platillo, id_articulo→cantidad

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

- Salsas(nombre_salsa, picor)
  - nombre_salsa→picor

  De nuevo, no hay violaciones a la BCNF

- IngredientesSalsa(nombre_salsa, id_articulo, cantidad)
  - nombre_salsa, id_articulo→cantidad

  No hay violaciones a la BCNF

- Precios(id_platillo, fecha, precio)
  - id_platillo, fecha→precio

  No hay violación a la BCNF.

- Recomendaciones(id_platillo, nombre_salsa)
  En este caso sólo tenemos la dependencia funcional trivial (id_platillo, nombre_salsa → id_platillo, nombre_salsa), por lo que no hay violaciones a la BCNF.

- PresentacionSalsas(nombre_salsa, tamanio)
  En esta relación se presenta el mismo caso que en la anterior (sólo tenemos nombre_salsa, tamanio → nombre_salsa, tamanio), por lo que no hay violaciones a la BCNF.

- PreciosSalsas(nombre_salsa, tamanio, fecha, precio)
  - nombre_salsa, tamanio, fecha→precio

  No hay violaciones a la BCNF.

- Promociones(id_promocion, tipo_descuento, dia, tipo_producto)
  - id_promocion, → tipo_descuento, dia, tipo_producto

  No hay violaciones a la BCNF

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

No hay violaciones a la BCNF pues numero_ticket es una llave.

- ComponentesPedido(numero_ticket, id_platillo, id_promocion, nombre_salsa, tamanio, cantidad_platillo, cantidad_salsa)
  - numero_ticket $\twoheadrightarrow$ id_platillo
  - numero_ticket $\twoheadrightarrow$ nombre_salsa, tamanio
  - numero_ticket $\rightarrow$ id_promocion
  - numero_ticket, id_platillo $\rightarrow$ cantidad_platillo
  - numero_ticket, nombre_salsa, tamanio $\rightarrow$ cantidad_salsa

  En este caso, para la Forma Normal de Boyce Codd, sólo tenemos que fijarnos en las dependencias que son funcionales. También hay que notar que es esta relación, toda llave candidata debe necesariamente contener a los atributos numero_ticket, id_platillo, nombre_salsa y tamanio, pues no están del lado derecho de ninguna dependencia funcional.
  Tomando esto en cuenta, hay que notar que numero_ticket$\rightarrow$id_promocion y las demás dependencias funcionales son violaciones a la BCNF, y por lo tanto, tenemos que proceder con el proceso de Normalización.
  Elegimos como dependencia inicial numero_ticket$\rightarrow$id_promocion. Separamos ComponentesPedido en dos relaciones, una con los atributos de esta dependencia funcional, y otra con los demás atributos y el lado izquierdo de esta DF:
  - PromocionesPedido(numero_ticket, id_promocion), con $numero\_ticket \rightarrow id\_promocion$
  - Y(numero_ticket, id_platillo, nombre_salsa, tamanio, cantidad_salsa), con: $numero\_ticket, id\_platillo \rightarrow cantidad\_platillo$
    $numero\_ticket, nombre\_salsa, tamanio \rightarrow cantidad\_salsa\}$

  La primer relación ya está en BCNF, sin embargo, X no, puesto que de nuevo, como numero_ticket, nombre_salsa, tamanio e is_platillo no están del lado derecho de ninguna DF, tienen que ser parte de cualquier súper llave, y como ninguna DF tiene del lado izquierdo una súper llave, todas son violaciones a la BCNF
  Para continuar el proceso de normalización, elegimos $numero\_ticket, id\_platillo \rightarrow cantidad\_platillo$
  Separamos X en dos relaciones, una con los atributos de la DF y otra que tenga el lado izquierdo y el resto:
  - PlatillosPedido(numero_ticket, id_platillo, cantidad_platillo), con $numero\_ticket, id\_platillo \rightarrow cantidad\_platillo$
  - X(numero_ticket, id_platillo, nombre_salsa, tamanio, cantidad_platillo, cantidad_salsa), con:
    $numero\_ticket, nombre\_salsa, tamanio \rightarrow cantidad\_salsa\}$

  De nuevo, la primer relación ya está en BCNF, sin embargo, Y no, puesto que de nuevo, como numero_ticket, nombre_salsa, tamanio e is_platillo no están del lado derecho de ninguna DF, tienen que ser parte de cualquier súper llave, y como ninguna DF tiene del lado izquierdo una súper llave, todas son violaciones a la BCNF
  Para continuar el proceso de normalización, elegimos $numero\_ticket, nombre\_salsa, tamanio \rightarrow$

cantidad_salsa
Separamos Y en dos relaciones, una con los atributos de la DF y otra que tenga el lado izquierdo y el resto:

- SalsasPedido(numero_ticket, nombre_salsa, tamanio, cantidad_salsa), con $\{numero\_ticket, nombre\_salsa,$ *cantidad_salsa*$\}$
- Z(numero_ticket, id_platillo, nombre_salsa, tamanio, cantidad_platillo), solo con DF triviales.

Note que no perdimos ninguna DF en el proceso de normalización.
Ahora, Z no tiene dependencias funcionales no triviales, pero aún sigue teniendo redundancia puesto que mantiene las DMV:

- numero_ticket $\twoheadrightarrow$ id_platillo
- numero_ticket $\twoheadrightarrow$ nombre_salsa, tamanio

ya 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. Es importante notar que decidimos esto pues no perdimos ninguna DF en la normalización y nos pareció que valía la pena pensar en la siguiente forma normal.
Iniciamos con las DMV: numero_ticket $\twoheadrightarrow$ id_platillo y numero_ticket $\twoheadrightarrow$ nombre_salsa, tamanio. 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 numero_ticket $\twoheadrightarrow$ nombre_salsa, tamanio. Separamos Z en dos relaciones, una que contiene a los atributos de esta DMV, y otra con los atributos restantes y los del lado izqiuerdo de la DMV, que en este caso sólo es numero_ticket:

- SalsasPedido1(numero_ticket, nombre_salsa, tamanio), con una única DMV: numero_ticket $\twoheadrightarrow$ nombre_salsa, tamanio.
- PlatillosPedido1(numero_ticket, id_platillo), con una única DMV: numero_ticket $\twoheadrightarrow$ 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 Z es la anterior.
Ahora note que estas relaciones son redundantes, pues dos que ya teníamos, SalsasPedido y PlatillosPedido, tienen exactamente esta información con una columna extra que está determinada de manera única por los atributos de estas dos relaciones.
Como no queremos relaciones redundantes, no tomamos estas dos relaciones resultantes en cuenta y nos quedamos sólo con las demás que obtuvimos en el proceso de normalización. Así, DetallesPedido queda separada en tres relaciones:

- SalsasPedido(numero_ticket, nombre_salsa, tamanio)

- PlatillosPedido(numero_ticket, id_platillo)
- PromocionesPedido(numero_ticket, id_promocion)