

Consideraciones de Diseño: Reglas de Negocio vs Nuestra Base

1. La carta de sus locales incluye tacos, burritos, quesadillas, gringas, tortas, platillos mexicanos, salsas y bebidas (se adjuntan algunos ejemplos de menús de taquerías, para que se den alguna idea), los precios se suelen colocar con etiquetas fluorescentes y en caso de que deban ser actualizados, se pega una nueva etiqueta encima. A don Pepe le interesa poder llevar un registro histórico de los precios de todos sus productos y de los insumos de los mismos, ya que le interesa saber en qué momento puede sacar alguna promoción.

Los alimentos y bebidas están almacenados en la tabla **comida**. El registro histórico para alimentos, salsas e insumos se guardan su respectiva tabla.

2. Se desea poder otorgar tickets por cada consumo (puede ser por comensal o mesa), el cual debe tener la siguiente información: fecha, sucursal, mesero que atendió, detalle de los productos solicitados con el subtotal en cada caso y el total del consumo.

La fecha, datos del cliente y mesero que atendió se guardan en una tabla **ticket**, el cálculo del subtotal se recupera mediante una función SQL, y el detalle de los productos se puede recuperar de las tablas **registrar_comida** y **registrar_salsas**. Desplegar toda la información junta se deja a la aplicación (que no se implementa).

3. Se necesita identificar la cantidad aproximada de porciones de ingredientes necesaria para cada una de las preparaciones que se realizan en la taquería, esto con el fin de minimizar los desperdicios generados (los datos no tienen que ser reales, ten en cuenta que es solo una simulación, solo trata que tengan algún sentido).

Los ingredientes para comidas y salsas se almacenan en las tablas **ingredientes_comida** e **ingredientes_salsa**. Sin embargo no existe restricción si alguien quiere almacenar ingredientes de bebidas envasadas.

4. Se desea crear un programa llamado Taquero de corazón, el cual otorgará el 10% del consumo en puntos, los cuales podrán ser intercambiados por productos dentro de la taquería, cada punto equivale a un peso.

Se agrega un procedimiento almacenado que incrementa los puntos de un cliente. Si los puntos son NULL es que no está en el programa.

5. De igual forma, suele manejar promociones como: “Lunes de 2x1 en tacos al pastor”, miércoles de “Taco Chinito” en el que todos los tacos cuestan \$10.00 o “Viernes de tortuga” (2x1 en tortas).

Las promociones se manejan dentro de la función **genera_monto**. Solo se manejan por ahora 10% de descuento en Pozoles y 25% de descuento en Aguas.

6. La taquería ofrece servicio a domicilio por lo que se necesitarán los datos de los clientes (nombre completo, CURP, dirección, teléfono, correo electrónico); con este servicio, los clientes podrán pagar solo con efectivo o tarjeta de crédito/débito.

Los datos de los clientes se almacenan en la tabla **cliente**. Las restricciones de pago están implementadas en un disparador que se activa al insertar un ticket.

7. Se desea tener un control de pago a empleados a través del pago vía nómina y otorgar seguridad social; para lograr esto, se requiere la información personal de los empleados (nombre completo, dirección, RFC, edad, número de seguridad social, antigüedad, dirección, teléfono y correo electrónico). Adicionalmente se busca crear un programa de bonos para empleados, con el que cada empleado que

cumpla 2 años laborando en la taquería se le agregará a su pago de nómina \$1,500.00. Los clientes también pueden ser alguno de los empleados, en cuyo caso, el servicio será gratuito.

Los datos de los empleados se guardan en la tabla **empleado**. El pago de nómina junto con el bono después de cumplir 2 años se genera mediante la función **genera_nomina**. El descuento del 100% en consumo se aplica en la función **genera_monto**.

8. Se debe considerar una solución que pueda ser escalable para poder eventualmente controlar sucursales, ya que don Pepe tiene pensando abrir al menos una en los próximos meses y posteriormente, abrir una en cada alcaldía de la CDMX y más adelante, una en cada uno de los estados de la República Mexicana.

De entrada la base está poblada con 16 sucursales(una por alcaldía), sin embargo se pueden agregar más sin restricción alguna ya que cualquier relación es independiente de la sucursal.

9. Derivado del punto anterior, los empleados sólo podrán estar ligados a una sucursal; en el caso de los clientes, no se debe duplicar su información en la base de datos.

Los datos de todos los empleados están unificados en una sola tabla(**empleado**), lo mismo los clientes(**cliente**), por lo que no se puede encontrar información duplicada.

10. Se busca tener un control de inventarios especificando el día de compra, fecha de caducidad, cantidad, marca del producto adquirido, precio de compra (ingredientes, mesas, sillas, bancos, platos, servilleteros, etcétera).

Se crea la tabla **productos_inventario** donde se almacena el inventario. Este inventario es único por sucursal. Sin embargo este inventario no tiene correlación con los productos que se registran en los tickets. Se delega esta responsabilidad a la aplicación(No implementada).

11. También interesa almacenar información de los proveedores, ya que en un futuro interesará hacer alianzas comerciales, a fin de obtener precios más competitivos.

Los proveedores se almacenan en la relación **proveedor**. Se almacenan su rfc y un teléfono.

12. Un rubro importante es lo relacionado a las salsas, ya que las de este lugar son muy famosas y se desea presentarlas en una sección detallando: ingredientes, nivel de picor (dulce, bajo, medio, alto, extremo) y recomendación de platillo a acompañar. De igual forma, derivado del éxito de sus salsas, los clientes pueden obtenerlas para sus eventos, las presentaciones que se tienen son: frasco de 250 mg., de medio kilo o bien por litro.

Se almacenan los datos de las salsas en la tabla **salsas**.

13. Es posible que un cliente no desee dar sus datos (por cuestiones del aviso de privacidad), en este caso, no se le niega el servicio y se debe mantener una imagen que llamaremos “cliente default” al cual se ligará el servicio. Este cliente existe por cada sucursal. Los datos en este caso son: nombre completo (sucursal), CURP (dejaremos el RFC de la taquería), dirección, teléfono, correo electrónico (estos son los que corresponden a la taquería). Este tipo de cliente no puede hacer pedido a domicilio.

Esto se maneja mediante el procedimiento almacenado **agregar_cliente_default**. Se verifica en la tabla **empleado** mediante un disparador que un cliente default no pueda ser empleado. Así mismo existe un disparador para verificar que un pedido a domicilio no sea registrado a nombre de un cliente default.

14. Existen 5 tipos de empleados: parrilleros, taqueros, meseros, cajeros, tortilleros, repartidores.

Esta información se encuentra en la tabla **empleado**. Un empleado solo puede y debe tener una especialidad.

15. En el caso de los repartidores, además de la información definida para los empleados, se requiere del número de licencia (si aplica) y si cuenta o no con transporte (motocicleta o bicicleta). En el caso de contar con transporte, se requieren los siguientes datos: marca, modelo, tipo (motocicleta/bicicleta).

Los transportes se almacenan en la relación **transporte**. Se verifica que solo los repartidores puedan almacenar vehículos.

16. Por cuestiones de mercadotecnia, se necesita saber las distintas formas en que los clientes pagan los servicios: efectivo, tarjeta de crédito, o puntos etc.

Esta información se almacena en la tabla **ticket**.

Modelo Entidad Relación

1. Entidades:

- cliente
Se crea esta entidad para almacenar los datos de los clientes(nombre, dirección, CURP, puntos del sistema taquero de corazón).
PK: CURP.
- empleado
Se crea la entidad empleado como especialización de cliente, esto porque los empleados también son clientes. A su vez empleado se especializa en cajero, repartidor, mesero, etc.
PK: CURP. Heredada de cliente.
- transporte(Entidad débil)
Dado que se deben guardar los datos de los vehículos de los repartidores se crea esta relación.
PK: id_transporte(discriminante) + CURP(Pk de empleado).
- proveedor
Está indicado en las reglas de negocio que se desea almacenar datos de los proveedores. Se almacena solamente rfc, nombre y los teléfonos.
PK: rfc
- sucursal
Se crea la entidad sucursal y se almacenan nombre, direccion, email, telefono.
PK: id_sucursal. Decidimos crear este identificador para evitar conflictos con el nombre cuando se creen más sucursales
- productos_inventario(Entidad débil)
Se guarda el stock(por sucursal), nombre, marca, descripción, caducidad y un histórico del día y el precio de compra.
PK: id_producto(discriminante) + id_sucursal
- ticket
Se genera como llave un número de ticket, se guardan el tipo de servicio, fecha, método de pago, subtotal y total(calculados). Se pide además se guarde el nombre del mesero y la sucursal(se toma de los datos del mesero).
PK: num_ticket
- productos_venta
Se crea esta entidad madre que guarda un historico(precio, fecha), una lista de ingredientes(ingrediente, cantidad) y la clave de venta. Se especializa en comida y salsas.
PK: clave_venta
- comida
Hereda los datos de productos_venta y le añade la descripción.
PK: clave_venta. Heredada de productos_venta.
- salsas
Hereda los datos de productos_venta y le añade recomendacion(platillo), nivel de picor y cantidad de la presentación.
PK: clave_venta. Heredada de productos_venta.

2. Relaciones.

- poseer(Relación débil)
Entre transporte(total, M) y repartidor(parcial, 1).
Un repartidor puede almacenar varios transportes, pero un transporte solo puede pertenecer a un repartidor, en caso de que se borre al repartidor se borran los transportes.
- trabajar
Entre empleado(total, M) y sucursal(parcial, 1).
Un empleado debe trabajar en una sola sucursal. La sucursal puede tener cualquier número de empleados.
- surtir
Entre proveedor(parcial, M) y sucursal(parcial, N).
No hay restricciones sobre el trabajo entre proveedores y sucursales.
- almacenar(Relación débil)
Entre sucursal(parcial, 1) y productos_inventario(total, M).
El inventario está fuertemente ligado a cada sucursal, por lo tanto si se borra la sucursal, se borran los datos de los productos en inventario.
- registrar
Entre productos_venta(parcial, 1) y ticket(parcial, 1).
Contiene el atributo cantidad de donde posteriormente se calcularán el subtotal de un ticket. Fuera de eso no existen restricciones entre las entidades que participan en la relación.
- atiende
Entre ticket(total, M) y mesero(parcial, 1).
Dado que se debe guardar el nombre del mesero tenemos participación total y limitamos a un mesero responsable de un ticket.
- pagar
Entre ticket(total, M) y cliente(parcial, 1).
Dado que se deben guardar los datos del cliente tenemos participación total. Solo un cliente puede pagar un determinado ticket.

Modelo Relacional

- transporte
Generada a partir de la entidad débil transporte.
PK: curp(empleador) + id.transporte
- empleado
La especialización desde cliente genera esta relación, sin embargo la especialización en cajero, repartidor, etc. queda plasmada en atributos para cada uno, es decir, no crea una nueva relación.
PK: curp FK: id.sucursal(Añadida por la relación entre sucursal y empleado)
- cliente
Generada a partir de la entidad cliente
PK: curp
- ticket
La entidad ticket en el modelo E/R se transforma en una relación junto con sus atributos.
PK: num.ticket
FK: curp.empleado(Añadida por la relación entre ticket y empleado)
FK: curp.cliente(Añadida por la relación entre ticket y cliente)
- comida y salsas
La entidad productos_venta se descompone en dos tablas con los atributos de la entidad madre, más los atributos de cada entidad hija.
PK: clave_venta

- historico_comida e historico_salsas
Como teníamos estos atributos multivaluados en la entidad productos_venta se crea para cada entidad hija una tabla.
PK: clave_venta(ya sea comida o salsas)
PK: precio
PK: fecha
- proveedor
La entidad proveedor genera esta tabla.
PK: rfc
- telefono_proveedor
El atributo multivaluado telefono genera esta tabla.
PK: rfc
PK: telefono
- surtir
La relación surtir al no tener restricciones nos da esta tabla de referencia.
FK: rfc
FK: id_sucursal
- sucursal
Sucursal crea su propia tabla junto con sus atributos.
PK: id_sucursal
- productos_inventario
La entidad débil productos_inventario genera tabla y añade como llave primaria la llave de sucursal.
PK: id_producto
PK: id_sucursal
- historico_inventario
El atributo multivaluado historico_inventario genera tabla.
PK: id_producto
PK: id_sucursal
PK: dia_compra
PK: precio_compra
- ingredientes_comida e ingredientes_salsas
Al ser atributo multivaluado genera tabla, como la relación productos_venta tiene especialización sin traslape y total se generan dos tablas, una para comida y otra para salsas.
PK: clave_venta
PK: nombre
PK: cantidad
- registrar_comida y registrar_salsas
La relación entre productos_venta y ticket genera una tabla para comida y otra para salsas dada la especialización de productos_venta. PK: clave_venta
PK: num.ticket

Nombres

1. Sela Farrera Ortega - 311617670
2. Ernesto Cárdenas Torres - 306155406
3. Maria Dadmy Nolasco Botello - 309127802
4. Yamil Salazar Gonzalez - 306037445
5. Jose Angel Hernandez Morales - 315137903