

Proyecto Final

Traducción del Modelo Entidad Relación a Relacional

Procedimiento

Para describir el procedimiento de traducción lo que haremos es enlistar las relaciones del modelo relacional y explicar de qué entidades o relaciones provienen, así como de dónde provienen sus atributos.

NOTA: Los atributos y nombres de relación no estarán tildados, la razón es porque de esta manera los nombramos en SQL Server.

- La relación **Sucursales** proviene de la Entidad *Sucursal*, y sus atributos son exactamente los que se enlistan en la entidad del modelo ER:
 - *numero_sucursal*, un identificador único asociado a cada sucursal, que será de tipo *int* y será la **llave primaria**.
 - *ciudad*, *estado*, *calle* y *numero*, que son los atributos que componen el atributo compuesto *direccion*, el cual no aparece en el modelo relacional por ser compuesto. Los tres primeros tipos deben ser de tipo *varchar*, mientras que *numero* es un entero.
- La relación **Clientes** viene de la entidad *Cliente*, y entre sus atributos están los que se enlistan en la entidad, así como el número de la sucursal en la que se registró, el cual viene de la relación *registrar* entre *Cliente* y *Sucursal*, pues es N:1.
 - *correo_electronico* será del tipo *varchar* y servirá como la **llave primaria** de la relación.
 - *nombre*, *apellido_materno*, *apellido_paterno*, los atributos que corresponden al nombre completo, los tres de tipo *varchar*
 - *puntos*, los puntos que ha acumulado el cliente registrado. Este atributo tendrá como dominio los números enteros.
 - *calle*, *numero*, *ciudad*, *estado* son los que corresponden a la dirección, y es el mismo caso que de la relación anterior.
 - *telefono*, que será de tipo *varchar* para guardar los espacios o guiones entre los números.
 - *numero_sucursal*, el número de la sucursal en la que se registró, un entero y que pasa a esta tabla pues *Clientes* tenían una relación muchos a uno con *Sucursal*. Este atributo funciona como una **llave foránea**.
- La tabla **Empleados**, que viene de la entidad *Empleado*, y sus atributos son:
 - Aquellos que vienen de la entidad Empleado:
 - *rfc*, que escogimos para funcionar como **llave primaria**. Debe ser de tipo *char(13)*, pues el rfc de personas físicas tiene una longitud fija de 13 caracteres.
 - *nombre*, *apellido_paterno*, *apellido_materno*, correspondientes al nombre completo y el mismo caso que para Clientes.
 - *curp*, que debe ser de tipo *char(18)*.

- *calle, numero, ciudad, estado*, los correspondientes a la dirección y análogos al caso de las entidades anteriores.
- *fecha_nacimiento*, de tipo *date*. El atributo *edad* no pasa a este modelo pues es derivado.
- *numeor_seguro, cuenta_bancaria, tipo_sangre*, del tipo *varchar*.
- *tipo_transporte, licencia*. Atributos que no serán NULL sólo cuando el empleado sea Repartidor.
- Los atributos de la relación *trabajar*, así como el número de sucursal, pues esta relación es N:1 entre *Empleado* y *Sucursal*:
 - *salario, bonos*, ambos de tipo *money*.
 - *fecha_contratacion*, de tipo *date*.
 - *tipo_empleado*, el tipo del empleado, de tipo *varchar*.
 - *numero_sucursal*, el número de la sucursal en la que trabaja. Este atributo es una **llave foránea** que apunta a la relación Sucursales.
- La relación **Vendedores** viene de la entidad *Vendedor* y sólo tiene los atributos asociados a esta relación:
 - *rfc*, que debe ser de tipo *char(12)* pues se tratará de personas morales, y fungirá como **llave primaria** de la relación.
 - *nombre, teléfono*, ambos de tipo *varchar*.
- La tabla **MateriaPrima** viene de la entidad del mismo nombre y posee los mismos atributos que ésta:
 - *id_articulo*, de tipo entero, que será la **llave primaria**.
 - *nombre, tipo*, ambos de tipo *varchar*, el segundo nos dirá si es un ingrediente o un mobiliario u objeto.
- La tabla **Inventario**, que también proviene de una entidad del mismo nombre, y posee los siguientes atributos:
 - De la entidad *Inventario*:
 - *precio_unitario*, de tipo *money*
 - *cantidad*, de tipo entero, que con lo anterior nos ayuda a calcular el atributo derivado *cantidad_compra*, el dinero total que se gastó en la compra.
 - *decha_compra, fecha_caducidad*, ambas de tipo *date*.
 - *Inventario* está en relaciones N:1 con *MateriaPrima, Sucursal* y *Vendedor*, por lo que las llaves primarias de esta relación pasan como **llaves foráneas**: *id_articulo, numero_sucursal* y *rfc_proveedor*; estas tres junto con la fecha de compra forman la **llave primaria**.
- La tabla **Tipo** proviene de la entidad del mismo nombre y tiene solamente sus atributos, un *id_tipo*, que elegimos como **llave primaria**, y el *nombre* del tipo de platillo, de tipo *varchar*.

- La tabla **Platillos**, que viene de la entidad *Platillo*, y tiene los siguientes atributos:
 - *id_platillo*, un entero único que servirá como **llave primaria de la relación**.
 - *nombre*, de tipo *varchar*, el nombre del platillo.
 - El *id_tipo* del tipo de platillo que es, esta es una **llave foránea** que apunta a la tabla Tipo, y viene de que la relación entre Platillo y Tipo en E/R es N:1.
- Como la relación entre *Platillo* y *MateriaPrima* en el modelo ER es N:M, esta relación se tiene que traducir a una tabla, la cual llamamos **IngredientesPlatillo**. Sus atributos son:
 - *id_platillo*, que es una **llave foránea** del id del platillo al que pertenece el ingrediente.
 - *id_articulo*, también una **llave foránea**, que es el id del identiicador, y que forma la **llave primaria** junto a *id_platillo*.
 - *cantidad*, de tipo entero.
- La entidad débil *Precios* pasa como una relación, y además de los atributos de la entidad, que son *precio* (de tipo *money*) y *fecha* (de tipo *date*), se envía como *llave foránea* el identificador del platillo al que pertenece. La **llave primaria** consiste de este identificador y la fecha en la que se agregue el precio.
- La entidad *Salsa* pasa como una tabla **Salsas**, el cual tiene como atributos sólo los de dicha entidad, *nombre*, que será su **llave primaria** y *picor*, su nivel de picor, ambos de tipo *varchar*.
- La tabla **IngredientesSalsas** sale de la relación entre *Salsa* y *MateriaPrima* que es N:M. Es análoga a **IngredientesPlatillo**.
- La entidad Presentación es una entidad débil, que se traduce como a tabla **Presentaciones-Salsa**, y la cual tiene el atributo *tamano*, y el identificador de la salsa, *nombre_salsa*. Ambos atributos son la **llave** de esta relación.
- La entidad *PreciosSalsa* se traduce de manera análoga a *Precios*, por lo que tenemos una tabla **PreciosSalsas**, en donde la **llave primaria** es el identificador de la presentación (nombre y tamaño de la salsa, que pasan como **llaves foráneas**) y la fecha en que se puso el precio, y también tenemos un atributo que guarda el precio.
- Al ser de cardinalidad N:M, la relación entre Platillos y Salsas pasa como una tabla, **Recomendaciones**. Lo único que lleva esta tabla son dos **llaves foráneas**, el nombre de la salsa y el id del platillo. Ambas funcionan en conjunto como la **llave primaria**.
- La entidad *Promocion* se convierte en la tabla **Promociones**. Sus atributos son:
 - *id_promocion*, un identificador de tipo entero que servirá como **llave primaria**.
 - *dia*, de tipo entero, el día en que aplica la promoción.
 - *tipo_descuento*, de tipo *varchar*.
 - *id_tipo*, el identificador del tipo en el que aplica. Este atributo pasa a la tablal pues la relación entre Promociones y Tipos es N:1.

- La entidad *Pedido*, pasa como una tabla, **Pedidos**. Los atributos de esta tabla serán:
 - *numero_ticket*, un número único que servirá como **llave primaria**.
 - *no_mesa*, el número de mesa (si aplica) del pedido, de tipo entero
 - *metodo_pago* El método con el que se pagó el pedido.
 - *fecha* la fecha en la que se efectuó.
 - Todos los atributos anteriores son los correspondientes a la entidad pedido, excepto *total*, que no aparece por ser derivado. Adicionalmente agregamos dos atributos que son llaves foráneas provenientes de relaciones N:1 en el modelo ER:
 - *numero_sucursal*, que apunta a Sucursales.
 - *correo_cliente*, que apunta a Clientes.
- la última tabla en nuestro modelo relacional es **ComponentesPedido**, que es la resultante de la relación n-aria que hay entre pedidos, presentaciones de salsas, platillos y promociones en el modelo ER. Traduciendo tenemos que crear una relación donde estén las llaves primarias de todas estas relaciones, por lo que nos quedamos con los atributos:
 - *numero_ticket*
 - *id_platillo*
 - *nombre_salsa, tamano*
 - *id_promocion*

Todos ellos **llaves foráneas** y en su conjunto forman la **llave primaria**.

Dependencias Funcionales

- **Sucursales(numero_sucursal, calle, numero, ciudad, estado)**
 En esta relación, las únicas dependencias funcionales son: $\text{numero_sucursal} \rightarrow \text{calle}$, $\text{numero_sucursal} \rightarrow \text{numero}$, $\text{numero_sucursal} \rightarrow \text{ciudad}$ y $\text{numero_sucursal} \rightarrow \text{estado}$, pues el número de sucursal determina de manera única la dirección de la sucursal (cada sucursal tiene una única dirección, que no comparte con ninguna otra sucursal). Los atributos de la dirección no tienen dependencias funcionales entre sí, ni siquiera ciudad y estado, pues puede haber ciudades del mismo nombre en distintos estados de la república.
 Aplicando la regla de la unión nos queda la dependencia funcional: $\text{numero_sucursal} \rightarrow \text{estado, ciudad, calle, numero}$.
- **Clientes(correo_electronico, nombre, apellido_paterno, apellido_materno, puntos, calle, numero, ciudad, estado, telefono, numero_sucursal)**
 Sabemos que el correo electrónico le corresponde a una única persona, y de esa manera determina al nombre y la dirección. Además al estar asociado a una cuenta, determina a los puntos que tiene y a la sucursal en la que se registró.
 Esto nos da la dependencia: $\text{correo} \rightarrow \text{nombre, apellido_paterno, apellido_materno, puntos, calle, numero, estado, ciudad, telefono, numero_sucursal}$.
 Es la única DF, pues el nombre puede coincidir con dos o más clientes, y los demás no se determinan entre sí.

- **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)**

En este caso tenemos que tanto el curp como el rfc determinan a una única persona, y por lo tanto se determinan entre sí. Es decir, tenemos $\text{curp} \rightarrow \text{rfc}$ y $\text{rfc} \rightarrow \text{curp}$. De igual manera, el número de seguro también es personal y por lo tanto también tenemos $\text{curp} \rightarrow \text{numero_seguro}$, $\text{rfc} \rightarrow \text{numero_seguro}$, $\text{numero_seguro} \rightarrow \text{curp}$, $\text{numero_seguro} \rightarrow \text{rfc}$. Estos tres atributos no pueden aparecer del lado derecho de ninguna otra dependencia funcional pues ni el nombre completo, aún con la fecha de nacimiento, ni la dirección ni información laboral corresponden únicamente a una persona. (Ni siquiera la cuenta bancaria, pues puede haber cotitulares). Y entre los demás atributos no hay dependencias funcionales pues el nombre (ni por partes ni completo) determina la dirección o la información laboral, pues el nombre puede repetirse, ni la información laboral depende de la dirección ni viceversa.

Sin embargo, los tres atributos que pueden determinar de manera única a la persona sí tienen unos únicos valores de los demás atributos asociados, por lo que también tenemos:

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

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

En esta relación, un único rfc determina un nombre de la compañía y un teléfono, y son las únicas dependencias funcionales pues los nombres de las empresas se pueden repetir.

Así, la única dependencia funcional que tenemos es $\text{rfc} \rightarrow \text{nombre, telefono}$

- **MateriaPrima(id_articulo, nombre, tipo)**

En este caso, el identificador del artículo determina al nombre y al tipo de manera única, por lo que tenemos: $\text{id_articulo} \rightarrow \text{nombre}$ y $\text{id_articulo} \rightarrow \text{tipo}$

Adicionalmente, los nombres de artículos no se repiten, por lo que también determinan de manera única al identificador y al tipo, por lo que también tenemos: $\text{nombre} \rightarrow \text{tipo}$ y $\text{nombre} \rightarrow \text{id_articulo}$

- **Inventario(id_articulo, numero_sucursal, fecha_compra, precio_unitario, cantidad, fecha_caducidad, rfc_proveedor)**

En el inventario se guardan compras, donde una compra está determinada por el artículo comprado, la sucursal que lo compra y la fecha en que se compra. Esto determina de manera única a los demás atributos pues el precio unitario en que se dio la venta, la cantidad que se compró, la fecha de caducidad y el vendedor son únicos por compra. Esto nos da la DF:

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

Y esta es la única DF pues los demás atributos no determinan un valor único de los demás, ni tampoco lo hacen con los de la izquierda-

- **Tipo(id_tipo, nombre)**

Aquí tenemos que un id corresponde a un único nombre de tipo, pero también lo inverso. Entonces tenemos: $\text{id_tipo} \rightarrow \text{nombre}$ y $\text{nombre} \rightarrow \text{id_tipo}$.

- **Platillos(id_platillo, nombre, id_tipo)**

En este caso, las únicas dependencias funcionales son: $\text{id_platillo} \rightarrow \text{nombre}$ y $\text{id_platillo} \rightarrow \text{id_tipo}$, pues a un platillo le corresponde un único tipo, y un único nombre. Y no tenemos la dependencia $\text{nombre} \rightarrow \text{id_tipo}$ puesto que podemos tener dos especiales con el mismo nombre, por ejemplo “El demoledor” puede ser tanto una Torta como un taco, y no tenemos restricción de que dos platillos no pueden llamarse de la misma manera.

Aplicando la regla de la unión, tenemos $\text{id_platillo} \rightarrow \text{nombre, id_tipo}$.

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

Note que aquí sólo tenemos que el platillo y el ingrediente determinan una única cantidad, pues el platillo puede tener muchos ingredientes y viceversa. Así, la única dependencia funcional en este caso es: $\text{id_platillo, id_articulo} \rightarrow \text{cantidad}$.

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

En este caso, para un platillo en específico y una fecha, hay un único precio asociado. De esta forma, tenemos:

$\text{id_platillo, fecha} \rightarrow \text{precio}$.

- **Salsas(nombre_salsa, picor)**

En esta relación, sólo tenemos que una salsa tiene un único nombre y un único nivel de picor, por lo que sólo tenemos: $\text{nombre_salsa} \rightarrow \text{picor}$.

- **IngredientesSalsas(nombre_salsa, id_articulo, cantidad)**

Este caso es similar al de los ingredientes de los platillos, el nombre de la salsa y el id del artículo determinan únicamente a la cantidad y es la única dependencia funcional. Por lo tanto sólo tenemos $\text{nombre_salsa, id_articulo} \rightarrow \text{cantidad}$.

- **PresentacionesSalsa(nombre_salsa, tamaño)**

En este caso sólo tenemos las dependencias funcionales triviales, pues una misma salsa puede tener varios tamaños y viceversa.

- **PreciosSalsas(nombre_salsa, tamaño, fecha, precio)**

Este es un caso análogo al de Precios, pero en esta relación, tenemos que el nombre de la salsa, el tamaño, y la fecha, determina el precio.

Así, tenemos: $\text{nombre_salsa, tamaño, fecha} \rightarrow \text{precio}$.

- **Recomendaciones(id_platillo, nombre_salsa)**

En este caso sólo tenemos las triviales, pues para un mismo platillo puede tener recomendadas varias salsas y viceversa.

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

En esta relación sólo tenemos id_promocion, \rightarrow tipo_descuento, dia, tipo_producto, ya que tanto los días, los tipos de descuentos y los tipos de producto se pueden repetir en distintas promociones.

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

Cada numero_ticket tiene asociado un único Pedido, y cada pedido sólo se paga con un método, en una única sucursal, si es el caso en una única mesa; se hizo en una única fecha, y la hizo un único cliente.

Así, tenemos numero_ticket \rightarrow numero_sucursal, metodo_pago, no_mesa, fecha, correo_cliente. Los demás atributos se pueden repetir en múltiples pedidos, y ninguno puede determinar a otro.

- **ComponentesPedido(numero_ticket, id_platillo, id_promocion, nombre_salsa, tamaño)** En esta relación hay una única dependencia funcional: numero_ticket \rightarrow id_promocion, pues hemos decidido que las promociones no sean acumulables, por lo que a cada pedido le corresponde una única promoción.

Sin embargo, en un mismo pedido puede haber muchos platillos y muchos tipos de salsa. Pero más aún, note que por cada tipo de salsa en un pedido, repetimos esta información por cada platillo pedido; y por cada platillo pedido en un pedido, repetimos la información por cada salsa pedida. Esto nos da un par de dependencias multivaluadas que pueden resultar útiles más adelante:

numero_ticket \twoheadrightarrow id_platillo y *numero_ticket \twoheadrightarrow nombre_salsa, tamaño*