



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Decisiones de diseño

PRESENTA

Aguirre Muñoz Leonardo 318017686
Becerril Lara Francisco Javier 317114490
Isunza Alvarez Marcos Guillermo 419002921
Sosa Hedding Etzael Iván 316259305
Zenteno Pompa Juan Carlos 316251608

PROFESOR

Gerardo Avilés Rosas

AYUDANTES

Gerardo Uriel Soto Miranda
Valeria Fernanda Manjarrez
Ricardo Badillo Macías
Carlos Augusto Escalona Navarro

ASIGNATURA

Fundamentos de Bases de Datos

1. Consideraciones de Diseño

Entidad: Persona

Atributos compuestos

- Direccion
 - 1. Calle
 - 2. C.P. (Código postal)
 - 3. Colonia
 - 4. Alcadia
 - 5. Exterior
- NombreC. (Nombre Completo)
 - 1. ApellidoP. (Apellido Paterno)
 - 2. ApellidoM. (Apellido Materno)
 - 3. Nombre

Atributos simples

- Nacimiento
- PK
- IdPersona

Atributos calculados

- Edad

Atributos multivaluados

- Correo
- Telefono

Consideramos Persona como Superentidad para poder desglosar dos subentidades.

- Cliente
- Empleado

Ambas subentidades conservan los mismos atributos, en cambio para Empleado tiene dos atributos definidos.

- CURP
- Salario

Empleado tiene otras dos subentidades:

- Cajero
- Encargado
- Gerente

Entidad Producto Atributos Simples

- IDProducto

-
- Precio

- Marca

Entidad Departamento Atributos Simples

- IDepartamento

- Nombre

Subentidades Departamento

Abarrotes

- Atributos Simples

1. Empaque
2. FechaPreparacion

Subentidades Departamento

1. Perecedero
2. Imperecedero

Atributos Simples

- Refrigeracion
- FechaCaducidad

Electronicos

- Atributos Simples

1. Watts
2. Descripcion

- Atributos Compuestos

1. Categoria
 - Telecomunicacion
 - Informatica
 - Alumbrado
 - Vigilancia

Entidad Venta

- Atributos Simples

1. IDVenta
2. CantidadProducto

- Atributo Calculado

1. Total

2. Relaciones

- Departamento **Conformar** Sucursal
- Venta **Realizar** Sucursal
- Empleado **Trabajar** Sucursal
- Cajero **Efectuar** Venta
- Venta **Adquirir** Cliente
- Producto **Contener** Departamento

3. Justificación Entidad-Relación

A continuación, enumeramos las entidades y relaciones que participan en el Diagrama Entidad Relación que modela nuestra Base de Datos. Adicionalmente, proporcionamos una ventana al proceso de raciocinio colectivo que llevo acabo nuestro equipo a la hora de concebir formalmente, en el sentido Platónico de las ideas, nuestra Base de Datos y su estructura dentro del marco de trabajo de la Entidad Relación.

3.1. Persona

La entidad Persona la utilizaremos para almacenar datos sobre los Clientes y los Empleados (mediante herencia) ya que comparten atributos tales como nombre, dirección, fecha de nacimiento, correo(s) electrónico(s), teléfono(s). Cómo sólo nos importa almacenar estos dos tipos de personas, la especialización será total y por temas de simplicidad no dejaremos que un Empleado sea un Cliente. Nota: Podría haberse elegido (o puede elegirse) usar el RFC como identificador de persona, pues con los datos que tenemos es suficiente para obtenerlo.

3.2. Empleado

La subentidad Empleado se especializa en Gerente, Encargado y Cajero.

3.3. Departamento

La entidad Departamento tiene un identificador y un nombre. Se especializa en Abarrotes y Electrónicos. A su vez, Abarrotes se especializa en Perecederos e Imperecederos, que cuentan con fecha de preparación y tipo de empaque (lata, bolsa, etc.). Los Perecederos tienen una fecha de caducidad y un atributo para decir si necesitan refrigeración. Por su parte, los Electrónicos tienen una descripción, la cantidad de Watts que necesitan para funcionar y la categoría a la que pertenecen como indica la especificación de la práctica.

3.4. Producto

La entidad Producto sólo tiene un identificador, la marca y el precio (original) del producto.

3.5. Sucursal

La entidad Sucursal cuenta con un identificador y algunos atributos necesarios como el nombre, la dirección, el horario de la tienda y otros datos solicitados.

3.6. Venta

La entidad Venta se utiliza como la generadora del ticket de compra, por eso su identificador.

3.7. ListaCompra

Finalmente, aunque no se menciona explícitamente, diseñamos una entidad ListaCompra para llevar el rastreo de los productos que compra un Cliente. Esta cuenta con un identificador y un atributo calculado que es Total, el cual hará la suma del costo de todos los Productos adquiridos.

- Para la parte de las relaciones, algunas son muy intuitivas como que los Empleados trabajan en una Sucursal y tienen un horario con hora de entrada y salida. Al sólo poder trabajar en una y que una Sucursal puede tener varios empleados es uno a muchos. Como sí puede haber Sucursales sin Empleados, es parcial de un lado y total del otro. También la relación Conformar entre Departamento y Sucursal es intuitiva (aunque quizá no necesaria). Contener es la relación entre Departamento y Producto para relacionar lo que contiene cada Departamento de la tienda. Un Producto puede estar en un Departamento pero un Departamento puede tener varios Productos. Esta tiene un atributo para llevar el registro del Stock que hay de cierto Producto en el Departamento.

-
- La ListaCompra se relaciona con Producto mediante Formar para poder tener almacenada una lista (ticket) de cada producto que se vende. Cuenta con dos atributos, uno para considerar descuentos posibles (Promoción) y otro para registrar el costo final del Producto (Descuento) ya tomando en cuenta posibles ofertas. Es muchos a muchos porque una ListaCompra puede tener varios Productos y un producto puede estar en varias ListaCompra.
 - Un Cliente compra una o varias ListaCompras y como es Cliente, debe hacerlo pero una ListaCompra es única por Cliente. Una ListaCompras genera una Venta (un ticket) y sólo una. La relación tiene un atributo para contar el número de Productos que tiene la ListaCompra.
 - Una Sucursal realiza una Venta (no necesariamente vende por lo que es parcial) pero una Venta se hace en una única sucursal, un Cajero cobra la Venta pero no necesariamente debe cobrar, puede no ocurrir y un Cliente efectua una Venta y una Venta siempre viene de un único cliente. Esta última relación tiene un atributo para saber qué método de pago utilizó (efectivo, tarjeta o mixto, es decir, una combinación de ambos).

4. Justificación

A continuación, detallamos más a fondo algunas decisiones de diseño que llevamos durante el periodo de gestación conceptual de nuestra Base de Datos, junto con algunas decisiones difíciles que cometimos en el camino.

1. Primero trabajamos con las entidades fuertes: Producto, ListaCompra, Venta, Sucursal. En cuanto a Departamento y Persona, al ser entidades que manejan herencia se siguió la idea vista en clase para los casos de uso Cine y Metro CDMX donde la entidad genera una tabla y se le asigna un atributo extra para diferenciar las subentidades (por ejemplo, en Cine la entidad Persona genera una tabla y se crean 3 atributos booleanos para diferenciar si la persona es Actor, Director o Productor). A la hora de la práctica, empleamos ésta técnica con la entidad Departamento para los casos de Electronicos, Perecederos e Imperecederos. Por otro lado, con Persona tenemos un caso particular puesto que Cliente se relaciona con Venta, al igual que Cajero. Para solucionar esto, sí dividimos la entidad Persona en Empleado y Cliente. Del lado de Empleado utilizamos el mismo recurso que con el Cine (Personas) o Departamento, para diferenciar a un Cajero, Gerente o Encargado. Así, Empleado se puede relacionar con Venta (la relación Cobrar) y Cliente se relaciona con Venta (Efectuar) y son distinguibles. Aquí las únicas dudas son si esto está bien resuelto debido a que podría dividirse cada una de las subentidades, generando 4 tablas prácticamente idénticas con 2 tablas cada una para el teléfono y correo y cada una relacionada con Sucursal para denotar la relación Trabajo.
2. Después seguimos con las relaciones que generan tabla que son Formar (muchos a muchos) y Generar (uno a uno).
3. Finalmente conectamos todo conforme las relaciones lo indicaban. La mayoría se trataban de relaciones 1:N que están bien descritas en las notas de clase (La clase con participación total y cardinalidad muchos absorbe el identificador de la entidad con cardinalidad uno y participación parcial). En el caso de Conformar y Comprar la única diferencia es la participación. En donde las relaciones generan tabla, es decir, Formar y Generar se siguió lo visto en clase. Formar es análogo al ejemplo del cine. La relación absorbe la cardinalidad de las entidades y es participación obligatoria para asegurar que lo que se encuentre ahí proviene de las entidades que une. En Generar sólo modificamos la cardinalidad.
4. Como última nota, al separar Cliente y Empleado, cada uno genera una tabla para su Correo y Teléfono.

5. Consideraciones extra

En la población de nuestra Base de Datos elegimos añadir algunas sucursales adicionales que no son referenciadas por ningún departamento, empleado de cualquier tipo o venta. Ésto con el propósito de poder mostrar algunas operaciones que serían posibles en la Base de Datos a la hora de manejar información de múltiples sucursales, pero sin el *bloat* y la expansión exponencial del tamaño de todas las consultas que habría conllevado el poblar con empleados, ventas, etc. referenciantes a cada una de éstas sucursales en igual medida a las primeras 15 sucursales en la base.

6. Instrucciones para la probar la ejecución de los procedimientos almacenados y los triggers

Para probar la ejecución de los procedimientos almacenados y los disparadores podemos realizar lo siguiente:

1. Ingresar a pgAdmin y cargar los datos
2. Hacer click derecho en el servidor donde queremos cargar los datos y elegir la opción Query Tool, cargar las instrucciones del DDL y ejecutarlos.
3. Nuevamente desplegar Query tool para copiar y pegar las instrucciones del DML y ejecutarlo
4. Nuevamente desplegar Query tool para copiar y pegar las instrucciones de los procedimientos y los triggers que se encuentran en el script programacion.sql
5. Para ejecutar los procedimientos se debe de realizar en orden. Primero se selecciona las instrucciones del la creación del procedimiento, las cuales empiezan con las palabras CREATE PROCEDURE y se ejecutan, después se selecciona la instrucción que empieza por call() y se ejecuta para probarlo.
6. Para ejecutar los disparadores se realiza de manera análoga, primero se selecciona y se ejecutan las instrucciones para la creación del trigger y después se selecciona las instrucciones para habilitarlo las cuales empiezan por CREATE TRIGGER
7. Para probar su funcionamiento seleccionar las instrucciones para probar su funcionamiento, estas instrucciones empiezan con la palabra **update**.