

ESTRATEGIA

Trabajo Práctico - Gestión Bazaar



Gestión de Datos

2C-2022

Fecha de Entrega: 02/11/2022

N° de Versión de la Entrega: 1

Grupo N°33: NOT_FOUND

Alumno	Legajo	Email	Curso
Agustín Evans	174.426-4	aevans@frba.utn.edu.ar	K3572
Santiago D'Andrea	175.863-9	sadandrea@frba.utn.edu.ar	K3521
Tomás D'Andrea	177.762-2	tdandrea@frba.utn.edu.ar	K3521

Representante: Agustín Evans

ÍNDICE

1. Creación del Modelo de Datos	3
1.1. Creación de Estructuras	3
1.1.1. Productos y Variantes	3
1.1.2. Ventas	3
1.1.3. Compras	4
1.2. Representación (DER).....	4
2. Migración De Datos	5
2.1. Creación de las Tablas	5
2.2 Creación de los Stored Procedure	5
2.3. Decisiones y suposiciones tomadas	5

1. Creación del Modelo de Datos

Se desarrollará a continuación cuáles fueron las estrategias y decisiones que se tomaron para la construcción del modelo.

1.1. Creación de Estructuras

El modelo de datos diseñado consta de 24 entidades las cuales fueron creadas para organizar y normalizar los datos de una única tabla provista por la cátedra (tabla maestra), con el fin de poder cumplir con los nuevos requerimientos del sistema. A continuación se explicará el motivo y/o fin del creado de las distintas entidades y relaciones a partir de los distintos componentes y/o datos provistos.

1.1.1 Productos y Variantes

Para modelar los productos/variantes decidimos contar con la entidad **PRODUCTO** para poder representar los datos específicos de cada producto (nombre, descripción) la cual es parametrizada por las entidades **MARCA**, **CATEGORÍA** y **MATERIAL** con el fin de no tener descripciones repetitivas en cada uno de los productos.

Siguiendo con la misma lógica, para el modelado de las variantes contamos con la entidad **VARIANTE** con su correspondiente descripción y la parametrizamos con **TIPO_VARIANTE** para no tener descripciones de los tipos de cada variante de manera repetitiva.

Como el enunciado lo indica, un producto puede tener muchas variantes y cada variante puede estar en muchos productos, por lo que inevitablemente nos llevó a crear la entidad **PRODUCTO_VARIANTE** que nos permite representar aquellos datos que dependen de cada combinación de producto y variante en particular, como por ejemplo el precio de venta y compra del mismo.

Para esta última entidad decidimos que tenga su propio código (se asemeja al código del producto específico pero es diferente al código de PRODUCTO) que sea único para cada combinación de producto y variante, permitiéndonos, además, poder modelar productos únicos, es decir, que no tengan variantes (ya que la FK que referencia a la entidad variantes la creamos de tipo null).

1.1.2. Ventas

Para el modelado de las ventas creamos la entidad **VENTA** con los campos correspondientes a cada venta particular (fecha, costos e importes vinculados, importe total), la entidad **CLIENTE** para contar con los datos de los clientes que pertenecen a cada una de ellas, **LOCALIDAD** para las localidades de los mismos, con su respectivo código postal y provincia, la cual la normalizamos en la entidad **PROVINCIA**. Así mismo normalizamos su canal, medio de envío y medio de pago, creando la entidad **CANAL**, para modelar los distintos canales con sus respectivos costos de transacción, **MEDIO_PAGO** para los medios de pago y sus costos y porcentaje de descuento y la entidad **MEDIO_ENVIO** para los distintos tipos de envío. Como el precio de los mismos depende de las localidades de los clientes, nos llevó a realizar la entidad **ENVIO_LOCALIDAD** para poder representar los distintos precios según el tipo de envío y localidad.

1.1.3 Compras

De igual manera, para las transacciones asociadas a las compras creamos las entidades **COMPRA_PRODUCTO** **DESCUENTO_COMPRA** con una lógica similar que para las ventas, pero sin incluir el requerimiento de los tipos de descuentos.

[illegible]

2. Migración de Datos

Para poder realizar la migración de datos primero se debieron crear todos los componentes de la base de datos a partir del modelo de datos diseñado.

2.1 Creación de las tablas

Para crear las tablas fuimos organizando el orden de sus creaciones, realizando primero las tablas maestras paramétricas (Marca,categoría,etc), luego la tabla maestra a la que parametriza (en este caso Producto) hasta llegar a las tablas transaccionales (venta_producto, etc). Para crear las PK de las tablas que no contaban con su ID en la tabla Maestra, decidimos crearlas de forma autoincremental (int IDENTITY).

2.2 Creación de los Stored Procedure

Luego de crear las tablas, siguiendo un orden, creamos los stored procedure para migrar los datos de la tabla maestra a cada una de ellas según el caso. Para la migración de datos a cualquiera de las tablas paramétricas(categoría,marca,material,tipo_variante,canal,etc) nos bastó con hacer un insert de un select de la tabla maestra + definir el código(PK) autoincremental en el momento de crear la tabla.

Para las tablas con una o más FK la lógica fue similar, pero al tener que insertar las PK con las que se relaciona, debimos hacer un JOIN con esa misma tabla y matchear con los campos correspondientes.

2.3. Decisiones y suposiciones tomadas

A continuación se detallarán las suposiciones y decisiones que tomamos para la migración para resolver distintas cuestiones que nos encontramos al realizar diversas consultas en la tabla maestra.

- Para migrar las venta_producto pudimos notar que en la tabla maestra había filas de ventas iguales con el mismo producto_variante pero con cantidades distintas e igual precio. Para solucionar esto a la hora de migrar los datos, consideramos tomar un registro solo con la suma de las cantidades. Para las compra_producto consideramos lo mismo.
- Para migrar las venta_cupon pudimos notar que en la tabla maestra habían cupones iguales para una misma venta, por lo que decidimos considerar un solo registro de las mismas, permitiendo así que un cupón no se pueda utilizar más de una vez en la misma venta.
- Para migrar los tipos de descuento para cada descuento de cada venta supusimos que el valor 'Otros' que muestra la columna VENTA_DESCUENTO_CONCEPTO hace referencia a los descuentos especiales. Llegamos a dicha conclusión por descarte, ya que:

1. No podía ser un descuento por envío gratis debido a que en ninguna venta con este concepto el VENTA_DESCUENTO_IMPORTE tenía el mismo valor que el VENTA_ENVIO_PRECIO.
2. Tampoco podía ser un descuento por medio de pago ya que habían ventas que tenían, además de este concepto, otro VENTA_DESCUENTO_CONCEPTO (o por Transferencia o por Efectivo) y como una venta está asociada a un solo medio de pago, entonces no pueden haber dos VENTA_DESCUENTO_CONCEPTO por medio de pago para una venta.
3. Tampoco podía ser un descuento por cupón ya que los cupones se tratan por separado en la tabla.

Consecuentemente consideramos que los VENTA_DESCUENTO_CONCEPTO que dicen 'Transferencia' o 'Efectivo' representan a los descuentos por medio de pago. Los descuentos de envío gratis refieren a aquellos en los que el valor del importe es igual al precio del envío de la venta (en la tabla no hay ninguno) .

- Para migrar los producto_variante consideramos 2 precios para cada producto con su variante: uno de venta y otro de compra. Ambos los calculamos a partir de la última fecha de la venta/compra que se realizó de ese producto/variante.
- Para poder representar los descuentos por cupón decidimos que en la migración de la tabla VENTA_DESCUENTO se inserten también los registros de cada venta con el importe de descuento calculado por la sumatoria de todos los importes de los cupones utilizados en las mismas.