



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Trabajo Practico 1C2023 - Entrega

Modelo de datos y migración

FRBA - Delivery

Gestión de datos

Docentes:

- Enrique Reinoso
- Juan Zaffaroni
- Max Ariosti

GRUPO: LOCKE N° 56

Alumno:

- Juan Pablo Bruno

ÍNDICE

ÍNDICE

Corrección

USUARIO y REPARTIDOR

PAQUETE y TIPO PAQUETE

TIPO MOVILIDAD

LOCALIDAD

LOCALIDAD_REPARTIDOR

ESTADO_ENVIO_MENSAJERIA

MEDIO DE PAGO

TIPO_MEDIO_PAGO

ENVIO_MENSAJERIA

LOCAL, TIPO_LOCAL y CATEGORIA_LOCAL

HORARIO_LOCAL

PRODUCTO_LOCAL

PEDIDO

ITEM_PEDIDO

ESTADO_PEDIDO

CUPON y CUPON_RECLAMO

CUPON_PEDIDO y CUPON_RECLAMO_PEDIDO

Estrategia modelo BI

Tablas de Hechos

Dimension_CategoriaLocal

Dimensión Tiempo

Dimensiones: TipoLocal, Local, CategoriaLocal, Tipo Movilidad,

EstadoEnvioMensajeria, TipoPaquete, TipoReclamo, EstadoReclamo

Corrección

Cambios realizados al modelo

Luego de la corrección decidimos guardar únicamente el estado(activo) de las entidades y no la traza de ellos.

Es por eso que realizamos cambios en las tablas:

- **#Estado_Pedido:** La convertimos en una tabla de tipo
- **#Estado_Envio_Mensajeria:** La convertimos en una tabla de tipo
- **#Estado_Reclamo:** La convertimos en una tabla de tipo

- **#Envio_Mensajeria:** Agregamos la columna ENVI_ESTADO donde se guarda el estado activo del envio mensajeria. Anteriormente este se guardaba en la tabla #estado_envio_mensajeria. Un envio_mensajeria puede pasar por muchos estados Ej realizado, confirmado, en camino, etc pero siempre va a tener un único estado activo y va a ser en el cual se encuentre el envío en el momento que se consulta.

- **#Pedido:** Agregamos la columna PEDI_ESTADO donde se guarda el estado activo del pedido. Anteriormente este se guardaba en la tabla #estado_pedido. Un pedido puede pasar por muchos estados Ej realizado, confirmado, en camino, etc pero siempre va a tener un único estado activo y va a ser en el cual se encuentre el pedido en el momento que se consulta.

- **#Reclamo:** Agregamos la columna RECL_ESTADO donde se guarda el estado activo del reclamo. Anteriormente este se guardaba en la tabla #estado_reclamo. Un reclamo puede pasar por muchos estados Ej iniciado, en revisión, solucionado, etc pero siempre va a tener un único estado activo y va a ser en el cual se encuentre el reclamo en el momento que se consulta.

USUARIO y REPARTIDOR

En estas entidades decidimos utilizar una PK autoincremental debido a que priorizamos la simplicidad del sistema y consistencia.

Según los datos de la tabla maestra anterior podríamos haber utilizado el DNI como PK ya que actuaba naturalmente como una, debido a que no había usuarios con DNI

REPETIDOS. Pero esto no me asegura que en el futuro va a seguir siendo así ya que al usar el DNI como PK puedo tener problemas con la sensibilidad de los datos, cambios en el DNI, escalabilidad y portabilidad y reutilización del DNI

Un campo int autoincremental permite una mayor escalabilidad y flexibilidad en comparación con el DNI. Puedes generar nuevos registros de usuarios sin depender de la disponibilidad de un DNI único.

PAQUETE y TIPO PAQUETE

La entidad paquete representa el envío específico de la mensajería. Su PK es el número de envío de la mensajería. Todos los detalles del paquete están asociados al Tipo de paquete, es por eso que esta entidad es la que guarda todos los detalles sobre el paquete. Como por ejemplo el PAQU_TIPO_PRECIO, PAQU_ALTO_MAX, etc.

La entidad Paquete únicamente tiene los campos PAQU_ENVIO(PK) y PAQU_TIPO que es FK a la tabla TIPO_PAQUETE.

TIPO MOVILIDAD

Decidimos crear una entidad TIPO_MOVILIDAD como una tabla de enumeración de los posibles valores de un REPA_TIPO_MOVILIDAD.

El enunciado nos dice que:

“El repartidor está dado de alta con un tipo de Movilidad, esto corresponde al vehículo que usa para hacer los repartos (moto, bicicleta, etc). Puede tener únicamente uno asociado.”

Esto nos llevó a analizar la estrategia de utilizar constraints para restringir los valores de la columna REPA_TIPO_MOVILIDAD. Debido que los valores que posibles eran únicamente ['Auto', 'Bicicleta', 'Motocicleta']

pero optamos por la estrategia de la tabla debido que nos otorga:

-Mayor mantenimiento y flexibilidad: Al utilizar tablas de valores o enumeraciones, puedes almacenar de manera centralizada y estructurada los valores permitidos para una determinada columna. Esto facilita el mantenimiento y la actualización de esos valores en un solo lugar. Si necesitas modificar o agregar nuevos valores permitidos, solo debes

realizar cambios en la tabla de valores, sin tener que modificar las constraints en todas las tablas relacionadas.

-Mayor Legibilidad y documentación: Al utilizar tablas de valores, los valores permitidos para una columna se vuelven más legibles y comprensibles para los desarrolladores y los usuarios. Al ver el contenido de la tabla de valores, es más fácil entender qué opciones están disponibles para esa columna. Además, estas tablas pueden servir como documentación clara de los valores permitidos en la base de datos.

LOCALIDAD

La tabla LOCALIDAD al igual que TIPO_MOVILIDAD únicamente tiene una columna y es una tabla de valores o enumeración. Su PK es el nombre de la LOCALIDAD y en ella se encuentran todas las localidades de los usuarios, locales y repartidores.

LOCALIDAD_REPARTIDOR

Es una tabla intermedia para romper una relación muchos a muchos entre Repartidor y Localidad.

“Los repartidores están dados de alta por localidades, en función de las cuales se le asignan los envíos. Un repartidor puede estar activo en una sola localidad.”

Al leer esto “*Un repartidor puede estar activo en una sola localidad*” podríamos elegir la estrategia de agregar una columna Localidad en la Tabla Repartidor, pero esto sería un error porque vemos en la tabla maestra que un repartidor puede tener más de una localidad. Tenemos que poner especial énfasis en la palabra **ACTIVO**.

Conclusión, un repartidor puede tener muchas LOCALIDADES pero estar ACTIVO en una SOLA y esta tabla nos permite relacionar a UN REPARTIDOR con TODAS SUS LOCALIDADES. Sin importar si ese repartidor se encuentra activo en esa localidad o no.

ESTADO_ENVIO_MENSAJERIA

Esta tabla relaciona a un Envío con su Estado.

En el modelo anterior no se registraban los distintos estados del envio_mensajería únicamente se registraba “Estado Mensajería Entregado”

“El sistema registra el estado en que se encuentra el pedido, como ser: realizado, confirmado, en camino origen, en camino destino, entregado, cancelado, etc.”

Esta tabla nos va a permitir identificar única entre cada combinación de nro de envío y estado de envío. Esto asegura que no haya duplicados y que cada registro en la tabla sea único y distintivo.

Además esto nos permite agregar nuevos registros en la tabla

"ESTADO_ENVIO_MENSAJERIA" cada vez que el estado de un envío cambie, y así mantener un registro completo y ordenado de todos los estados anteriores del envío.

MEDIO_DE_PAGO

“El usuario ingresa a través del sistema el medio de pago con el cual quiere pagar. Este puede ser a través de tarjeta de crédito o débito de distintos emisores. La tarjeta que se utilice debe estar previamente registrada y asociada al usuario para que este pueda seleccionarla al realizar el pago, sino debe darla de alta al momento de realizar el pedido.”

Como el medio de pago está relacionado a un usuario tiene una FK a la tabla USUARIO.

Decidimos utilizar como PK el MEDI_NRO_TARJETA ya que es un campo que nunca va a tener repetidos y identifica unívocamente al medio de pago

Según el enunciado los únicos tipos de medio de pago son Crédito o Débito pero en la tabla Maestra vemos que también se encuentra el Efectivo. En el caso de que el TIPO_MEDIO_PAGO sea “Efectivo” vemos en la tabla maestra que ingresa un valor en la columna MEDI_NRO_TARJETA que cumple con Unicidad.

TIPO_MEDIO_PAGO

La tabla TIPO_MEDIO_PAGO al igual que TIPO_MOVILIDAD únicamente tiene una columna y es una tabla de valores o enumeración. Su PK es el TIPO_MEDI_PAGO y en ella se encuentran todos los posibles tipos de medios de pago.

ENVIO_MENSAJERIA

Es la entidad principal del caso de uso Registro del servicio de mensajería.

Esta entidad está asociada a Usuario, Repartidor y Paquete.

Debido a que tanto la ENVI_DIRE_ORIG como ENVI_DIRE_DESTINO

“No es necesario que esta dirección esté previamente cargada o registrada como el caso del pedido.”

No se tiene que relacionar con la tabla DIRECCION_USUARIO

Tuvimos que eliminar un CONSTRAINT CHECK para la columna ENVI_KM > 0 ya que todas las filas de la tabla maestra esta columna se encuentra en 0.

Tuvimos que eliminar un CONSTRAINT CHECK para que la FECHA_ENTREGA > FECHA debido a que varias filas en la tabla maestra tienen cargada la información incorrectamente.

Decidimos poner algunos CONSTRAINT CHECKS para todas las columnas que sean sobre precio o valor sea >= 0.

LOCAL, TIPO_LOCAL y CATEGORIA_LOCAL

La entidad TIPO_LOCAL únicamente tiene una columna y es una tabla de valores o enumeración. En ella se encuentran todos los posibles tipos de local.

La entidad CATEGORIA_LOCAL únicamente tiene una columna y es una tabla de valores o enumeración. En ella se encuentran todas las posibles categorías de local.

Actualmente la tabla se encuentra vacía ya que anteriormente no se registraban las categorías del local, pero el nuevo modelo ya permite incorporarlas.

HORARIO_LOCAL

Debido a que los horarios dependen del Local, del día y observando los datos de la tabla maestra vemos que un Local y un Día pueden aparecer más de una vez dependiendo de si el Local cierra y vuelve a abrir más tarde en el mismo día. Para resolver esta situación y obtener una identificación unívoca de cada fila optamos por utilizar como PK LOCAL, DIA Y HORA_APERTURA.

Además decidimos crear un índice compuesto (HORA_LOCAL, HORA_DIA) ya que creemos que es muy probable realizar búsquedas basadas en la combinación de esas columnas y el índice compuesto en las mismas puede mejorar significativamente el rendimiento de las consultas.

PRODUCTO_LOCAL

Esta entidad tiene como PK (codigo_producto, local_nombre). Representa el catálogo de productos de cada local.

PEDIDO

Es la entidad principal del caso de uso Registro de pedidos.

Esta entidad está asociada a Medio de pago, Usuario, Repartidor y Local.

“Los usuarios pueden tener varias direcciones previamente cargadas asociadas a su cuenta. Si el pedido es a una dirección nueva, el usuario debe darla de alta para poder seleccionarla.”

Debido a esto sabemos que la entidad PEDIDO tiene que relacionarse con DIRECCION_USUARIO. Ya que el pedido tiene que referenciar la dirección del usuario ya guardada en el sistema.

ITEM_PEDIDO

Esta tabla relaciona un producto de un local con un pedido

Tuvimos que realizar una modificación al momento de ingresar los datos ya que en la tabla maestra la combinación PEDIDO_NRO, PRODUCTO_CODIGO Y LOCAL no representaba unívocamente a una fila (entonces tampoco era una combinación válida para PK) porque aparecía había filas que aparecían varias veces esas combinaciones solamente con diferente ITEM_CANTIDAD.

La decisión que tomamos para solucionar este problema fue sumar las columnas item_cantidad. Y así insertar una única fila a la tabla ITEM_PEDIDO según la combinación (PEDIDO_NRO, PRODUCTO_CODIGO, LOCAL) y permitiéndome usar estas 3 columnas como una PK compuesta.

ESTADO_PEDIDO

Esta tabla relaciona a un Pedido con su Estado.

En el modelo anterior no se registraban los distintos estados del pedido únicamente se registraba “Estado Mensajería Entregado” y “Estado Mensajería Cancelado”

“El sistema registra el estado en que se encuentra el pedido, como ser: realizado, confirmado, en camino origen, en camino destino, entregado, cancelado, etc.”

Esta tabla nos va a permitir identificar única entre cada combinación de nro de pedido y estado de pedido. Esto asegura que no haya duplicados y que cada registro en la tabla sea único y distintivo.

Además esto nos permite agregar nuevos registros en la tabla "ESTADO_PEDIDO" cada vez que el estado de un pedido cambie, y así mantener un registro completo y ordenado de todos los estados anteriores del pedido.

CUPON y CUPON_RECLAMO

Primero habíamos seleccionado la estrategia de que el CUPÓN y CUPON_RECLAMO estén relacionados. La tabla CUPONES contenía a TODOS LOS CUPONES y si era un CUPON_RECLAMO se encontraba en la tabla intermedia y se le agregaba la FK del reclamo.

Pero este modelo nos presentó problemas debido a que ocurría que los CUPONES y los CUPONES_RECLAMO la forma que se generaba el nro_cupon para insertar en la base de datos era la misma y ya el CUPON_NRO dejaba de identificar unívocamente a un usuario.

Por eso decidimos modelar a los CUPONES Y CUPON_RECLAMO como tablas diferentes y así puedo tratar tanto a los CUPONES como a los CUPONES_RECLAMO con una única pk que es cupon_nro.

Además la forma en que está modelado los cupones que no son originados por un reclamo no pueden referenciar a un reclamo.

CUPON_PEDIDO y CUPON_RECLAMO_PEDIDO

“Al momento de realizar el pedido, el usuario puede seleccionar si quiere utilizar cupones de descuento de los que tenga disponibles. De ser así los descuentos realizados se muestran en pantalla y se descuentan del total del pedido.”

Todos los descuentos aplicados a un pedido deben ser guardados en la BD del sistema, asociados a dicho pedido.”

Son las tablas que guardan los cupones utilizados en un pedido. Como esta modelado permite tener n cupones x 1 pedido.

Estrategia modelo BI

Tablas de Hechos

Decidimos realizar tres tablas de hechos según los tres casos de uso distintos que teníamos para mantener la integridad, la claridad en la estructura del modelo de datos y facilitar el análisis de datos en contextos específicos.

La razón detrás de tener tablas de hechos separadas es que las dimensiones compartidas representan contextos comunes en los que se analizan las métricas, mientras que las dimensiones diferentes implican un contexto diferente.

Definimos un nivel de granularidad por rango horario

Dimension_CategoriaLocal

Debido a que nuestra tabla *Categoria_Local* del modelo Relacional no tenía valores, al realizar el INSERT dentro de la tabla *Dimension_CategoriaLocal* está nos quedaría también vacía, que es lo que nos sucedió.

Esto nos generaba una gran problemática, debido a que la Tabla *Hechos_Pedido* tiene una PK/FK llamada *IdCategoriaLocal* que hace referencia a esta tabla. Y ya que un campo que es PK no puede contener valores nulos, tuvimos que insertar al menos un valor.

Dimensión_Tiempo

Inicialmente creamos esta dimensión con tres columnas. Un Id Autoincrement, Mes y año. Luego decidimos dividir esta dimensión en dos dimensiones. *Dimension_Anio* y *Dimension_Mes*.

Este cambio fue fomentado por una mayor claridad de datos y facilidad para interpretarlos. Además al separar el año del mes nos permitió evitar que se dupliquen los registros de cada mes por cada año.

Dimensiones: TipoLocal, Local, CategoriaLocal, Tipo Movilidad, EstadoEnvioMensajeria, TipoPaquete, TipoReclamo, EstadoReclamo

En todas estas dimensiones decidimos utilizar un INT AUTOINCREMENTAL debido a que priorizamos el rendimiento y las claves primarias enteras autoincrementales son más eficientes en términos de rendimiento al realizar operaciones de búsqueda, indexación y uniones en comparación con claves primarias compuestas o con claves naturales más largas.

