GESTIÓN DE DATOS

GRUPO N° 38	CURSO: K3114

PROFESOR: Marcelo mozcuzza

NOMBRE GRUPO: PEDIDOS_APP

ENTREGA N°: 1

TÍTULO: Entrega de modelo de datos y migración

INTEGRANTES PRESENTES EL DÍA QUE SE REALIZÓ	
Martín Spagnol	176546
Luciano Rodriguez	1771930
Gabriel Soares de Oliveira	2027770
Mateo Michini	1758160

RESPONSABLE: Martín Spagnol

Diagrama Entidad Relación

Usuario

Cada usuario es único, es por eso que lleva como atributo el DNI como PK (primary key) para poder ser identificado.

Usuario_Dirección

Cada puede tener registrado en el sistema varias direcciones que serán de utilidad para referenciarlas en el pedido. La misma cuenta con un código de ID propio como atributo PK para identificar unívocamente a las direcciones.

Medio de pago

Los usuarios registrados en el sistema deben contar con un medio de pago, estos deben realizarlo a través de un tipo de tarjeta donde el número de la misma será utilizado como PK en un atributo.

Cupones y sus tipos

Se modelaron dos ramas de cupones: cupón y cupón reclamo. Ambas entidades cuentan con un ID único q los identifica y encuentran relacionadas con USUARIO_X_CUPON que hace de intermediario entre CUPÓN y USUARIO, la misma cuenta con una PK con un id especifico.

En el caso de cupón, que no es cupón reclamo, lo encontramos relacionado al pedido mediante PEDIDO_X_CUPON la cual contiene a cupón y pedido como FK.

Luego para cupón reclamo contiene los datos propios del cupón y como FK al reclamo que se aplicó este cupón.

Reclamo y operador

Para identificar a los operadores utilizamos como PK OPERADOR_DNI. Esta entidad de operador la asociamos a la de Reclamo a través de otra entidad RECLAMO_X_OPERADOR

que tiene como FK al reclamo asociado con el operador, y esta entidad cuenta con su propio identificador RECLAMO_x_USUARIO_NRO, de esta forma desnormalizamos el reclamo con el operador. Por otro lado, el reclamo tiene asociado (según el enunciado) el pedido al cual se le realizó, por este motivo también se agregó la clave del pedido y usuario a sus atributos.

Pedidos

Cada Pedido tiene los atributos que se pidieron en el TP, este cuenta con una PK llamada PEDIDO_NRO y con medio de pago, local y usuario como FK.

Los pedidos pueden tener uno o más productos en sí mismos y es por eso que utilizamos la tabla PRODUCTO_X_PEDIDO para referenciarlos con las FK de producto_local, pedido y local de manera única con una PK denominada PRODUCTO_X_PEDIDO_NRO.

Producto

Los productos se ven reflejados en la entidad PRODUCTO_LOCAL, el cual tiene como atributos un nombre, precio y descripción, así como una FK al Local al que dispone este producto. Los mismos cuentan con una PK denominada PRODUCTO_LOCAL_CODIGO

Local y Horario

Un local tiene su id que denominamos LOCAL_CODIGO y los atributos mencionados en el enunciado. Desnormalizamos el horario y creamos otra entidad con su propio PK y como atributo FK tiene al local asociado, junto con atributos como el día, apertura del local ese día y la hora de cierre del mismo.

Envío

Un pedido está relacionado a un envió donde cuenta con la FK de pedido y del repartidor asociado a este. Este cuenta con una PK para identificar el envío denominado ENVIO_NRO

Repartidor y tipo movilidad

Un repartido cuenta con su DNI como PK. Se normalizo la movilidad respecto del repartido creando una entidad TIPO_MOVILIDAD con su propio PK denominado TIPO_MOVILIDAD_COD y su movilidad como atributos.

Envío mensajería y paquete

Envío mensajería cuenta con todos los atributos mencionados en el enunciado, utilizando como FK a usuario, repartidor, medio de pago y paquete.

Paquete cuenta con su PK denominada PAQUETE_NRO y una FK para envío mensajería

Modelo BI

Dimensiones Obligatorias

Tiempo

Utilizamos como PK una id asociada para cada combinacion de año y mes que tenemos el sistema

Dia

Como varios dias tienen la misma inicial asociada, decidimos que su PK sea el mismo nombre del dia, en ves de asociarle una id, ya que esto seria mas complicado a la hora de migrar y comparar con las fk, debido a que deberiamos hacer una funcion que depende del dia de la semana nos devuelva el id correcto

Rango Horario

Decidimos que utilizariamos id como PK ya que resultaria en un manejo mas facil, y generamos un atributo llamado rango horario descripcion que indica cual es el rango horario, para asignar las id hicimos una funcion, que verifica una fecha y retorna el id asociado al rango horario de esa fecha

Provincia

Utilizamos como PK un ID asociado a cada provincia, y un atributo que contenga el nombre de la provincia

Localidad

Utilizamos como PK un ID asociado a cada localidad, y un atributo que contenga el nombre de la localidad, pero como puede darse el caso, que exista una misma localidad pero distinta provincia, hicimos una fk a provincia, lo que hace que si existen localidades con el mismo nombre pero distinta provincia, tendran un id distinto

Rango etario

Decidimos que utilizariamos id como PK ya que resultaria en un manejo mas facil, y generamos un atributo llamado rango etario descripcion que indica cual es el rango etario, para asignar las id hicimos un procedure que genera todos los posibles rangos etarios dados, luego hicimos tres funciones para cada rango etario posible (operador/cliente/repartidor) donde cada funcion devuelve el id, correspondiente a el rango etario, de la persona, esto lo modelamos asi y no con varias tablas, ya que nos parecia rebundante y con funciones podiamos solucionarlo

Tipo de Medio de Pago

Utilizamos como PK un ID asociado a cada tipo de medio de pago, creamos un atributo llamado tipo de medio de pago, donde contiene la informacion del medio pago

Local tipo

Utilizamos como PK un ID asociado a cada tipo de local, creamos un atributo llamado tipo de tipo local descripcion, donde contiene la informacion de cual tipo de local es

Local

Utilizamos como PK el nombre del local, tambien una fk hacia tipo de local

Local Categoria

Utilizamos como PK un ID asociado a cada categoria de local, creamos un atributo llamado categoria local descripcion, donde contiene la informacion de la categoria de local que es

Tipo paquete

Utilizamos como PK un ID asociado a cada tipo de paquete, creamos un atributo llamado paquete tipo, donde contiene informacion de que tipo de paquete es

Estados pedido/reclamo/envios

Decidimos que para todas las tablas estados se utilize un ID como PK, y un atributo llamado estado, que indica el estado

Estados pedido/reclamo/envios

Decidimos que para todas las tablas estados se utilize un ID como PK, y un atributo llamado estado, que indica el estado

Dimensiones de funcionalidad

A la hora de pensar en los requisitos decidimos que las dimensiones obligatorias, mantengan la menor cantidad de FK entre si, asi luego creamos dimensiones especificas donde contengan todas las FK necesarias tanto como para los pedidos,estados,reclamos. Esto asi nos facilitara a la hora crear las vistas con lo requerimientos necesarios

Hecho pedido

Esta tabla tiene un ID como PK, a su vez tendra atributos sobre el total usado de cupones en el pedido, el total del servicio(que es la suma de los productos y la tarifa) el tiempo estimado de entrega,calificacion dada en el pedido, la fecha del pedido y la fecha de entrega del pedido, estos datos los sacamos de varias tablas, a su vez tendra varias FK hacia las dimensiones como a rango etario, que tendra dos ya que tendremos el rango etario repartidor y el rango etario usuario(decidimos hacerlo asi, ya que nos parecia irrelevante en

el modelo de negocio tener los datos del usuario y calcularrlo, por lo que a la hora de migrar directamente utilizamos la funcion que le asigna a que rango etario pertenece cada uno),el estado del pedido,tiempo, etc.

Hecho envio mensajeria

Esta tabla tiene un ID como PK, a su vez tendra atributos sobre el valor asegurado, el precio del seguro, el tiempo estimado de entrega, la fecha del envio y la fecha de entrega del envio, a su vez tendra varias FK hacia las dimensiones como a rango etario, que tendra dos ya que tendremos el rango etario repartidor y el rango etario usuario(decidimos hacerlo asi, ya que nos parecia irrelevante en el modelo de negocio tener los datos del usuario y calcularrlo, por lo que a la hora de migrar directamente utilizamos la funcion que le asigna a que rango etario pertenece cada uno),el estado del envio,tiempo, etc.

Hecho envio mensajeria

Esta tabla tiene un ID como PK, a su vez tendra atributos sobre el valor del cupon asociado al reclamo, la fecha del reclamo y la fecha solucion, a su vez tendra varias FK hacia las dimensiones como a rango etario, que tendra dos ya que tendremos el rango etario operador ,el estado del reclamo,tiempo, etc.

Migracion y modelo de datos

Creacion de tablas

A partir de las correcciones del DER y revisación de la tabla maestra, decidimos rediseñar nuestro DER para luego facilitar la migración. Hacer coincidir los tipos de datos y nombres similares dentro de nuestras tablas fue algo casi necesario y excluyente para que podamos obtener una buena migración. El tipo de datos, era meramente necesario ya que si no la migración sería imposible. Los nombres, procuramostenerlos lo más parecido posible para así no generar confusiones en las migraciones.

Vimos necesario que antes de crear cada tabla, constraint y procedures sea necesario eliminarlos o dropearlos de la base de datos para así poder probar varias veces y que se vuelvan a generar constantemente para no generar errores. Realizamos los drops en orden, ya que fue necesario primero eliminar los constraint que estaban relacionados con las tablas, luego las tablas creadas y por último los procedur

Constraints

Decidimos escribir las Constraints aparte de las tablas, realizando alter tables, donde definíamos las Foreign Keys de cada tabla. Esto nos ayudó para poder generar una relación pertinente entre las tablas donde nos asegurábamos que el lugar donde íbamos a buscar la clave foránea era el adecuado.

Stored Procedures

Al momento de migrar los datos, decidimos utilizar Stored Procedures llamados “MIGRAR_NOMBRE_DE_TABLA”. En ellos fuimos obteniendo los datos necesarios de gd_esquema.Maestra y pasarlos a la tabla indicada por el procedure en cuestión.

Fuimos limitando los posibles datos que podrían migrar en nuestras tablas realizando IS NOT NULL en los campos necesarios para no obtener campos nulos y así no generar inconsistencias en las tablas. A su vez, realizamos JOIN con otras tablas para poder obtener los datos adecuados.

Los JOIN nos obligaron a tener que realizar las ejecuciones de los procedures en orden, ya que para ciertas tablas, necesitábamos cargar otras tablas previamente.