|  |
| --- |
| SENIOR\_DEVELOPERS |
| FRBA - BUS 2013 |
| -ESTRATEGIA -DER |
|  |
| Índice.  Script  Explicación del modelo de datos creado  DER  Vistas  Triggers |

|  |
| --- |
|  |

**Script**

Durante el script inicial se procede en primer lugar al borrado de esquema y entidades, en el caso de que existan. Esto es para facilitar el trabajo si quisiera ejecutar de nuevo el script.

Luego se realiza la creación de entidades con sus respectivos campos y claves (ver explicación del modelo de datos).

Se incluyen también las vistas utilizadas como así también los procedimientos y funciones escogidas para cumplir con las nuevas restricciones del sistema y poder adecuarse a la situación actual.

Finalmente se procede con la migración de datos, la cual consiste en cargar todas las tablas creadas utilizando la totalidad de los datos pertenecientes a la tabla maestra, estableciendo un diseño que pueda solucionar el problema de normalización y desorganización presente en dicha tabla.

**Explicación del modelo de datos creado**

A continuación se detallan las tablas creadas, el porqué de sus relaciones, sus claves primarias, sus claves foráneas y el agregado de campos extras a los obligatorios.

Usuario:

Su primary key es el campo username, ya que creemos que es un campo que diferencia y hace únicos a los usuarios, en este caso.

Se relaciona con la tabla:

-Rol (a través de la foreign key rol\_ID). Un usuario puede tener un rol.

Debido a que un usuario de perfil “Administrador” que intenta acceder a todas las funcionalidades del sistema puede realizar el login de forma incorrecta, se debe agregar el campo “intentosFallidos” en el cual se irán sumando los mismos. Esto ayudará además a la hora de inhabilitar a un usuario luego de haberse registrado 3 intentos fallidos. Por ello, también debemos introducir el campo “habilitado”.

Rol:

Su primary key es el campo “ID”, ya que creemos que es un campo que diferencia y hace único a cada rol. Observamos que un rol (tanto administrativo como cliente) posee diferentes funcionalidades, mientras que cada una de dichas funcionalidades pueden ser asignadas a muchos roles. Por esto, se decidió crear la tabla Rol\_Funcionalidad (explicada a continuación) que refleja esta situación.

Además, para cuando se requiera realizar una baja lógica de algún rol consideramos necesario agregar el campo “habilitado”.

Rol\_Funcionalidad:

Su primary key está formada por los campos “Rol\_ID” y “Funcionalidad\_ID”, los cuales serán también foreign key. Consideramos necesaria esta tabla con el objetivo de tener disponible un rol con cada una de sus funcionalidades a la hora de quitar alguna de ellas (se quitan de a una por vez) o cuando se desea agregar nuevas funcionalidades para dicho rol.

Estará formada a partir de las tablas:

-Rol

-Funcionalidad

Funcionalidad: Su primary key es el campo ID, ya que creemos que es un campo que diferencia y hace única a cada funcionalidad.

Se relaciona con la tabla:

-Rol\_Funcionalidad (una funcionalidad puede pertenecer a distintos roles).

Como el sistema realizará altas, bajas y modificaciones de ciudades, recorridos y micros necesitaremos crear esas 3 tablas.

Ciudad:

Contiene posibles orígenes o destinos que un cliente puede escoger cuando decide realizar un viaje. Hace única a una ciudad la primary key “ID”.

Recorrido:

Cada recorrido se diferenciará por su primary key “ID” y a los campos obligatorios se le agregará el campo “habilitado”, ya que en caso de querer dar de baja un recorrido lo haremos de forma lógica. Cada recorrido puede tener dos ciudades por lo que estará relacionado con la tabla Ciudad a través de las foreign key “ciudadOrigen\_ID” y “ciudadDestino\_ID”.

También debemos realizar altas, bajas y modificaciones de los micros necesarios para realizar un viaje o el transporte de una encomienda por lo que confeccionamos la tabla Micro.

Micro:

Estará diferenciado con la primary key patente que lo hará único. Hemos tomado la decisión de no agregar campos extras en la tabla, pero decidimos analizar cada butaca del micro por separado ya que posee atributos propios de una butaca advirtiendo además que un micro puede tener muchas de ellas y cada una de ellas puede ser distintas entre sí (pasillo o ventanilla, por ejemplo). Lo mismo sucede con el servicio brindado por dicho micro. Poseerá

Butaca:

Se diferenciarán con un NRO pero como puede repetirse en otro micro, la primary key debe contener también el micro al que pertenece la butaca, por medio del campo “micro\_patente”. Este campo será además foreign key para poder relacionar la butaca con el micro correspondiente. Tendrá que contener campos que indiquen si pertenece al primer o segundo piso, y si está del lado del pasillo o de la ventanilla.

Servicio:

Su primary key será directamente su nombre y agregamos el campo extra “porcentajeAdicional” ya que cada recorrido tiene un precio base entre la ciudad de origen y destino, y en función de dicho precio base, se le aplica un porcentaje adicional en función del tipo de servicio que se esté brindando.

RegistroMicroFueraServicio:

Esta tabla es creada ya que cuando un ómnibus está fuera de servicio se debe indicar la fecha en la que el mismo volverá a estar disponible. Este atributo no es propio de la situación de un micro fuera de servicio. Su primary key está formada por el micro (“micro\_patente”) y por la fecha de fuera de servicio “fecha”. La patente nos permite relacionar este registro con el micro en particular, por lo que será FK además de PK.

Viaje:

Un usuario administrativo deberá poder armar una ruta de viaje para luego ser vendida. Creamos la tabla Viaje, cada uno de los cuales tendrá un único ID (PK). Cada viaje deberá contener un recorrido y el micro que realizará el viaje. Así tendrá dos claves foráneas: ”recorrido\_ID” y “micro\_patente”. Además de las fechas correspondientes se decide agregar el atributo “estado” el cual indicará si está realizándose el viaje (está vigente) o no. Esto facilitará las cosas cuando se desee asignar algún micro a un nuevo viaje. Puede ocurrir que el micro aún no haya completado su recorrido por lo cual no estará habilitado. Cuando el micro llegue a destino se cambiará dicho estado de vigencia.

Pasaje y Encomienda:

Cuando un cliente se acerca para realizar la compra de un pasaje/encomienda observamos que se deberá vincular la compra con ese cliente, asignar un viaje, un número de compra y un precio. Pero hay un campo que depende del tipo de compra, en el caso de un pasaje se asigna un número de butaca y en caso de encomienda se asigna un pesaje en KG. Es por esta razón que se crean dos tablas: Encomienda y Pasaje. Ambas estarán identificadas por un “ID” y tendrán como foreign key a “compra\_NRO”.

Un cliente podrá realizar varias compras a la vez por lo cual una compra puede tener asignados muchos pasajes. Para esto, se crea la tabla RegistroCompra vinculada con Pasaje, Encomienda y con el cliente en cuestión (FK: “cliente\_DNI”).

Cliente:

Su primary key es el campo DNI, ya que creemos que es un campo que diferencia y hace únicas a las personas (clientes en este caso).

Se relaciona con las tablas TarjetaCredito (un cliente puede tener muchas tarjetas de crédito), Encomienda, Pasaje y RegistroCompra.

A los campos referidos al cliente observados en la tabla maestra hemos decidido agregar 3 nuevos.

-genero: Se necesita tener el género de un cliente ya que dependiendo del mismo y de su edad es posible realizar descuentos al valor del pasaje.

-esDiscapacitado: se agrega ya que si un cliente es discapacitado tendrá el beneficio de que no serán cobrados más de 2 pasajes por compra.

-puntos: este atributo incluirá los puntos acumulados de “pasajero frecuente” que un cliente obtiene cuando finaliza un viaje o el transporte de una o más encomiendas.

TarjetaCredito:

Creamos esta tabla porque si la compra se está realizando desde una terminal kiosco, el único medio de pago será por medio de tarjeta de crédito. Su primary key es el campo “NRO” y posee una foreign key formada por el campo “cliente\_DNI” la cual permite a una tarjeta de crédito relacionarse con su respectivo cliente.

Se relaciona con la tabla Cliente (una tarjeta de crédito pertenece a un único cliente). Será necesario que contenga el número de la tarjeta de crédito, su código de seguridad, su fecha de vencimiento y el tipo de tarjeta de crédito ya que dependiendo del tipo de tarjeta se podrá realizar la compra en cuotas.

Cancelación/Devolución.

Cuando un cliente decide cancelar un pasaje y/o encomienda deberá registrarse dicha situación. Se crea la tabla Cancelacion identificada con un “ID” , una clave foránea “compra\_NRO” la cual relacionará a la compra que fue cancelada., el motivo y la fecha de cancelación.

Además, la cancelación puede ser de muchos pasajes o muchas encomiendas que pertenecían al mismo cliente y como tienen campos diferentes es necesario diferencias las cancelaciones por lo cual se crean las tablas CancelacionEncomienda y CancelacionPasaje.

Canje de puntos.

Se crean las tablas Canje que contendrá el DNI del cliente (“cliente\_DNI”) que tendrá cierta cantidad de puntos y atributos propios del canje en cuestión. El canje es identificado con un “ID” (PK) y se relaciona con el producto a canjear a través de la FK “producto\_ID”.

También se crea la tabla Producto que lo identifica con un “ID” (PK), los puntos necesarios para poder obtenerlo, el nombre y el stock.



**Vistas**

En el script se realiza la creación de vistas para obtener los clientes con sus respectivos pasajes/encomiendas, otra que mostrará los puntos de un cliente a la fecha y otra en la que se observan las compras que se realizaron en efectivo.

Además se incluye la creación de vistas especiales necesarias para la confección de los listados estadísticos.

Vistas creadas:

V\_DestinosPasajes: necesaria para obtener los destinos para los cuales se compró pasaje.

V\_MicrosPorcentajeLleno: necesaria para obtener los micros y su porcentaje de butacas llenas.

V\_ClientePuntos: mostrará los puntos acumulados a la fecha.

V\_DestinoPasajeCancelado: permite obtener los destinos con pasajes cancelados.

V\_MicrosDiasFueraDeServicio: necesaria para obtener los dias que un micro estuvo fuera de servicio.

La creación de estas vistas nos permitirá obentener más fácilmente los listados estadísticos que necesitamos y el top 5 de cualquiera de las condiciones planteadas anteriormente.

**Triggers**

Hemos decidido utilizar triggers durante situaciones en la cual se produce un evento el cual genera que se deban realizar ciertas operaciones a partir de él.

Triggers utilizados:

tr\_bajaRecorrido: Cuando se desea dar de baja un determinado recorrido se cancelan todos los viajes que estaban asignados a dicho recorrido.

tr\_cancelacionViaje: Cuando se cancela un viaje automáticamente se cancelan todos los pasajes y encomiendas correspondientes a ese viaje.

tr\_deshabilitarRol: Cuando se deshabilita un rol es necesario deshabilitar también a todos los usuarios que posean dicho rol.

tr\_canjeProducto: Cada vez que un cliente realiza el canje de un producto se debe reducir automáticamente el stock del mismo.