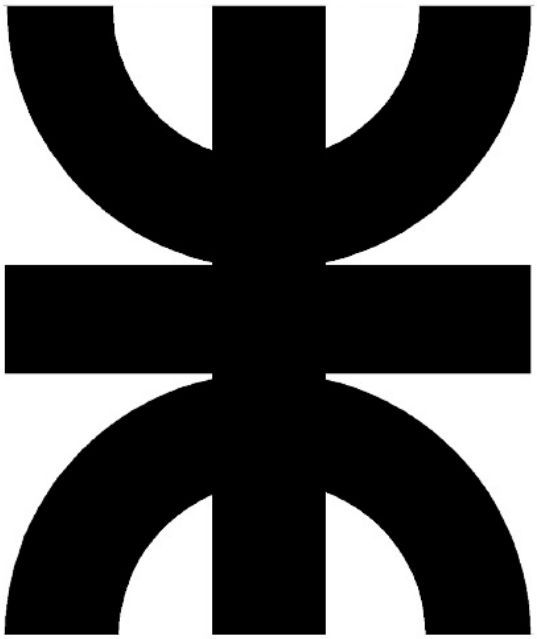
UTN
FACULTAD REGIONAL BUENOS AIRES
Ingeniería en Sistemas de Información



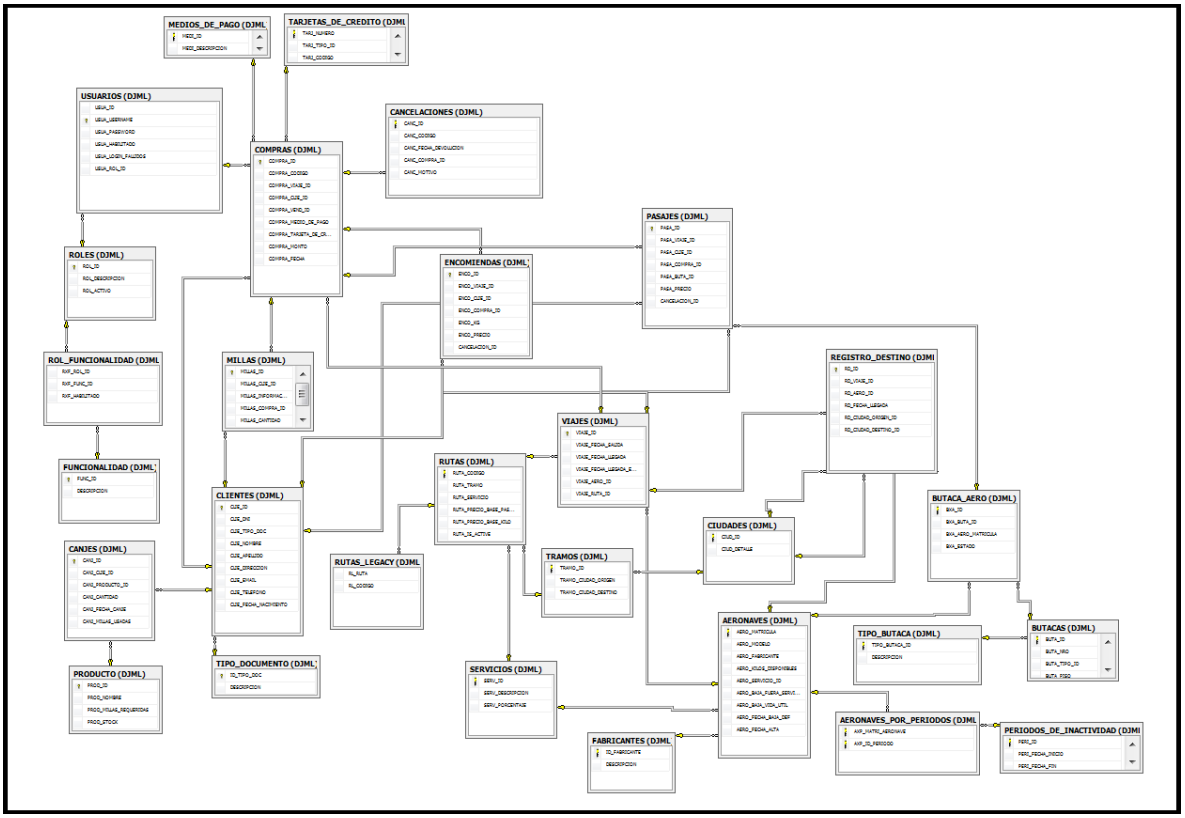
Gestión de Datos: FRBA AEROLINEAS 2C – 2015

Grupo: DJML

Integrantes	Legajos
Diego Alfonsin	135.151-5
Marcos Mezzabotta	146.884-4
Lucas Quintana	147.033-4
Juan Urbano	138.538-0

Índice

DER:



Para mayor legibilidad nuestro DER tambien se encuentra en un archivo por separado llamado DER.PNG

Lógica Principal:

Para la resolución del Trabajo Práctico se trató de utilizar la mayor cantidad de conceptos vistos en clase en relación a SQL y T SQL, a su vez añade en una parte dominante código ADO.NET para la ejecución de consultas desde la aplicación (conexión con la base de datos, modificación, actualización, etc.)

Consideraciones:

✓ Todos los campos creados a partir de la normalización respetan los mismos tipos de datos que la tabla MAESTRA. Para ello se crearon los respectivos Constrains para las tablas.

✓ El DER fue diseñado con la herramienta provista por Microsoft SQL Server Management Studio, de manera que refleja transparente y confiablemente el modelo de datos real.

✓ Para la carga inicial se crearon diversos procedimientos que agrupan la lógica de carga de cada tabla. La ejecución de estos procedimientos se realiza en un orden establecido para respetar la integridad referencial de las diversas tablas.

✓ El sistema comienza con el form de bienvenidos y el usuario elije que opción

seleccionar. Si selecciona Kiosco se le muestran las funcionalidades correspondientes a ese tipo de usuario. Si selecciona Administrador se le muestran las funcionalidades correspondientes para ese tipo de usuario previo a esto debiendo pasar por un login.

✓ Al haber DNI repetidos en la tabla maestra para salvar ciertos casos, se creó la PRIMARY KEY clie_id como un IDENTITY y además se creó la tabla Tipo_Documento para poder modelar distintos Tipos de Documentos con sus respectivos números. Para salvar los casos de los datos repetidos se tomo la decisión de migrar a todos los Clientes con distintos Tipos de Documento como lo permite el nuevo modelado de datos.

✓ La encriptación de las contraseñas no está realizada en la migración de datos sino una vez que realiza el primer logeo al sistema.

✓ Como no hay un AMB de formas de pago, se da por entendido que solo existen las formas de pago Efectivo y con Tarjeta aunque la solución propuesta permita agregar otros tipos de pago.

✓ En cuanto a la configuración de la hora, utilizamos el formato de 24 hs.

✓ Para el código identificatorio de las rutas también se generó un ruta_codigo PRIMARY KEY de tipo IDENTITY. Para modelar esta parte se crearon las tablas RUTAS, TRAMOS Y CIUDADES. La tabla RUTAS tiene asociada a ella un TRAMO. Un TRAMO tiene asociado una CIUDAD_ORIGEN y una CIUDAD_DESTINO. Y la tabla CIUDADES tiene todas las ciudades disponibles en el sistema hasta el momento. Se recuerda que no habia que desarrollar el AMB de ciudades pero el sistema por su modelo permite el alta de nuevas ciudades.

✓ Con estos cambios en las rutas no nos resultaba necesario el campo ‘Ruta_Codigo’ de la tabla maestra. Nuestra solución para esto fue crear una tabla que llamamos RUTA_LEGACY en donde registramos el código de ruta de la maestra y la PK de la tabla RUTAS.

✓ Para poder mantener los números de pasaje y encomienda, creamos una tabla llamada CLAVES donde consultamos y actualizamos a medida que estos se venden.

✓ La tabla COMPRAS tiene una FK a la tabla USUARIOS para representar al empleado que vendió el pasaje o encomienda. En caso de haberse tramitado por una terminal kiosko este campo será NULL y nos servirá para identificar este caso.

✓ Para realizar las cancelaciones de los pasajes y encomiendas solo se tienen en cuenta aquellos que fueron comprados a partir del nuevo sistema propuesto.

✓ Las millas obtenidas por los clientes que viajaron en la empresa son acreditadas luego que se registra la llegada a destino de la aeronave en cuestión. Cabe aclarar que solo computan millas aquellos clientes que viajaron luego del nuevo sistema ya que antes la empresa no ofrecía ese servicio.

✓ En la funcionalidad de compras se le da la posibilidad al cliente de que pueda cargar N números de pasajes n X números de encomiendas para mejorar la solución propuesta.

✓ Para la opción "remplazar por aeronave similar" en baja de aeronaves se creó un Procedimiento con cursores que busca aeronaves de la misma flota y valida que la aeronave por la que se va a remplazar este desocupada en las fechas de la aeronave a dar de baja. En caso de no haber ninguna aeronave disponible o en su defecto ninguna aeronave que cumpla sus mismas condiciones se crea la nueva aeronave de matricula "AUX-NNN" que va a ser la que suplante a la aeronave dada de baja.

Estructuras:

A continuación se realiza una descripción de las diferentes estructuras empleadas.

Stored Procedures

✓ Procedimientos para la carga inicial:

Estructura: La estructura es la misma para todo el script inicial, se mostrará un ejemplo de StoredProcedure para mayor interpretación.

Se describirá el ejemplo de la tabla Roles.

1. Se crea el procedimiento:
a. CREATE PROCEDURE DJML.CREAR_ROLES
2. Se crea la tabla donde se alojaran los datos, con sus respectivos Constraint los cuales servirán para restringir y limitar el tipo de dato que se puede insertar dentro de la tabla:
a. CREATE TABLE DJML.ROLES(
ROL_ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
ROL_DESCRIPCION VARCHAR(40) NOT NULL,
ROL_ACTIVO BIT NOT NULL)
3. Sí se pudo crear exitosamente muestra una leyenda " Se creó la tabla correctamente"
4. Por último se procede a migrar insertando (si corresponde) en la tabla creada, los datos correspondientes extraídos de la única Tabla Maestra provista por la cátedra. Con la lógica acorde para extraer de manera correcta todas las tablas con dicha información, sin que se pierda ningún campo de estas.
5. Finalmente se ejecutan todos los procedimientos creados anteriormente:
a. exec DJML.CREAR_ROLES
La lista de los procedimientos que tienen la funcionalidad descrita anteriormente es:

EXEC DJML.CREAR_ROLES

EXEC DJML.CREAR_FUNCIONALIDADES

EXEC DJML.CREAR_USUARIOS

EXEC DJML.CREAR_SERVICIOS

EXEC DJML.CREAR_CIUDADES

EXEC DJML.CREAR_RUTAS

EXEC DJML.CREAR_AERONAVES

EXEC DJML.CREAR_VIAJES

EXEC DJML.CREAR_PASAJEENCOMIENDA

EXEC DJML.CREAR_COMPRAS

EXEC DJML.CREAR_CLIENTES

EXEC DJML.CREAR_CANJES

Además de los objetos de base de datos ya mencionados contamos con:

- Una vista (v_rutas) que usamos para mostrar las rutas en su respectivo ABM.
- El procedimiento Proc_Aeronaves, anteriormente nombrado, para el reemplazo de una aeronave.
- Un trigger (Insertar_Millas) para asignarle las millas al cliente.