

TP GESTION DE DATOS 1C – 2020 (UTN FRBA)

-Curso: K3014,

-Número de grupo: 13,

-Integrantes:

- 1) Laurenzi Mauro, legajo: 1637472,**
- 2) Piwko Ariel, legajo: 1593341,**
- 3) Decibe Kevin, legajo: 1473487,**
- 4) Anchipi Gabriela, legajo: 1072237.**

Índice:

- CONSIDERACIONES RESPECTO AL MODELO DE DATOS... 3

**- CONSIDERACIONES GENERALES RESPECTO A
CREACION DE TABLAS... 3**

**- CONSIDERACIONES GENERALES RESPECTO A
CREACION DE VISTAS... 6**

**- CONSIDERACIONES GENERALES RESPECTO A
CREACION DE TRIGGERS... 6**

**- CONSIDERACIONES GENERALES RESPECTO A
CREACION DE STORED PROCEDURES... 7**

- CONSIDERACIONES RESPECTO AL MODELO DE INTELIGENCIA DE NEGOCIOS... 9

Consideraciones respecto al Modelo de Datos.

Consideraciones generales respecto a Creación de Tablas:

-A la Tabla Cliente se le cargan las columnas referidas a “Cliente” en la tabla maestra y se le suma la columna “ID_CLIENTE” para ser usada como PK y poder identificar a cada cliente unívocamente. Se decide que cumpla con la propiedad IDENTITY para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

No se utiliza la columna “CLIENTE_DNI” como PK ya que, según los datos otorgados para la migración, no cumple con la condición de unicidad.

-A la Tabla Factura se le cargan todas las columnas referidas a “FACTURA” de la tabla maestra, así como el “ID_CLIENTE” y “ID_SUCURSAL”, y se decide agregar el campo “MONTO_TOTAL” para conocer el valor total que se le factura a un cliente, ya sea por adquisición de estadía(s), pasaje(s) o ambos. Este campo será, entonces, la suma de los precios totales de cada pasaje y/o estadía que se relaciona con la factura en las tablas Pasaje_Facturado y Estadía_Facturada.

-A la Tabla Sucursal se le cargan todas las columnas referidas a “SUCURSAL” de la tabla maestra, y se le suma la columna “ID_SUCURSAL” para ser usada como PK y poder identificar a cada sucursal unívocamente. Se decide que cumpla con la propiedad IDENTITY para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

Se decide no utilizar como PK los campos referentes a la dirección, mail o teléfono que provee la tabla maestra ya que el valor de estos puede cambiar con el tiempo, no siendo entonces aptos para ser usados como clave primaria.

-A la Tabla Estadía_Facturada se le carga la columna FACTURA_NRO para conocer a la factura de la cual la estadía es ítem, así como ESTADIA_CODIGO para conocer a la estadía (que previamente fue comprada por la agencia y por lo tanto debe existir en la tabla “Estadía”). Se decide cargar también los campos “FECHA_CHECK_IN” y “FECHA_CHECK_OUT”, los cuales se calculan como: fecha de inicio de la estadía (valor almacenado en FECHA_CHECK_IN) y fecha de inicio de la estadía + cantidad de noches reservadas (valor almacenado en FECHA_CHECK_OUT), entendiendo de esta forma que una Estadía puede venderse únicamente de manera completa. Es decir, por la misma cantidad de noches que fue reservada originalmente por la agencia de viajes al hotel. Se le agrega también el campo “ESTADIA_CANTIDAD_NOCHES” para verificar que las fechas de check in y check out sean correctas y se suma también el campo “PRECIO_ESTADIA”, cuyo valor será la suma de todas las habitaciones de la estadía que se vendan a un Cliente.

Se interpreta que un Cliente puede realizar la reserva de varias estadías distintas y los datos de estas se cargaran dentro de una misma factura.

Se le suma la columna “ID_FACT_ESTADIA” para ser usada como PK y poder identificar a cada estadía facturada unívocamente. Se decide que cumpla con la propiedad IDENTITY para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

-Se crea la Tabla “Habitación_Vendida”, que actúa como tabla puente entre Estadía_Facturada y Habitación. En esta se almacenan los datos de la Estadía vendida o facturada (ID_FACT_ESTADIA) y de la habitación del hotel que se corresponde con la estadía (ID_HABITACION), se le agrega además el precio cobrado de la misma. De esta manera, se entiende que un Cliente puede reservar varias habitaciones del hotel del cual la agencia le vendió una estadía, y el precio de todas estas habitaciones que el cliente reserve se sumara al MONTO_TOTAL de una única factura.

-A la tabla “Pasaje_Facturado” se le cargan los campos referentes al registro de Factura del que se desprende (FACTURA_NRO) y el pasaje previamente cargado en la tabla Pasaje (PASAJE_CODIGO). Se le agrega además el precio de venta de este pasaje. Se le suma la columna “ID_FACT_PASAJE” para ser usada como PK y poder identificar a cada pasaje facturado o vendido unívocamente. Se decide que cumpla con la propiedad IDENTITY para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

Se interpreta que la agencia le puede vender mas de un pasaje a un cliente dentro de una misma venta, quedando todos estos ligados a una única factura.

-A la Tabla Compra se le cargan los campos “COMPRA_NUMERO”, el cual es utilizado como PK, “COMPRA_FECHA” y se agrega el campo “COSTO_TOTAL”, el cual tendrá el valor del costo de la estadía y/o el/los pasaje/s que hayan sido comprados por la agencia. A medida que se vayan cargando nuevos ítems (estadía o pasajes) se ira sumando el costo de los mismos al COSTO_TOTAL del registro de Compra que estos tienen como FK.

-A la Tabla Pasaje se le carga el campo “PASAJE_CODIGO”, el cual es utilizado como PK, la compra de la cual es ítem (COMPRA_NRO), el costo que significo el pasaje para la agencia, y el precio de venta del mismo, así como el vuelo y la butaca especifica. Se entiende que, dentro de una misma compra, la agencia puede adquirir varios pasajes distintos.

-A la Tabla Vuelo se le cargan: fecha de salida y llegada, ID_RUTA para conocer el origen y destino del mismo, el avión que se usara como transporte y el nombre de la aerolínea. Se le suma la columna “ID_VUELO” para ser usada como PK y poder identificar a cada vuelo unívocamente. Se decide que cumpla con la propiedad IDENTITY para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

-A la Tabla “Ruta Aerea” se le cargan los datos referentes a origen y destino del vuelo, el código de ruta aérea y se le suma la columna “RUTA_ID” para ser usada como PK y poder identificar cada ruta aérea unívocamente ya que el campo “RUTA_AEREA_CODIGO” no cumple con el principio de unicidad y, por ende, se descarta su uso como PK. Se decide que cumpla con la propiedad **IDENTITY** para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

-A la Tabla Avion se le cargan los dos únicos campos referentes a cada avión: su identificador, el cual se utiliza como PK, y su modelo.

-A la Tabla Butaca se le cargan los campos referentes al tipo y numero, así como el avión al cual pertenece. Se agrega también el campo “BUTACA_ID” para ser usado como PK y se decide que cumpla con la propiedad **IDENTITY** para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

-A la Tabla Estadía almacena los datos de las estadías compradas por la agencia y se le cargan los campos: **ESTADIA_CODIGO**, utilizado como PK, **COMPRA_NRO** para referenciar a la instancia de Compra de la cual la estadía en particular es ítem, fecha de inicio, cantidad de noches, y el **HOTEL_ID** para conocer el hotel en el cual se reservó la estadía. Se entiende que, dentro de una misma compra, la agencia solo puede reservar una única estadía.

-A la Tabla Hotel se le cargan todos los campos **HOTEL_NOMBRE**, **HOTEL_CALLE**, **HOTEL_NRO_CALLE**, **HOTEL_CANT_ESTRELLAS** y se le agrega la PK “**HOTEL_ID**” para poder identificar a cada hotel reservado por la agencia unívocamente. Se decide que cumpla con la propiedad **IDENTITY** para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

-A la Tabla Habitación se le carga los campos **HOTEL_ID**, **HABITACION_NUMERO**, **HABITACION_PISO**, **HABITACION_FRENTE**, **TIPO_HABITACION_CODIGO**, el costo de la misma y el precio de venta. Se le suma además la PK **HABITACION_ID** para poder identificar a cada habitación comprada por la agencia unívocamente. Se decide que cumpla con la propiedad **IDENTITY** para que la misma pueda insertarse de manera automática ante una nueva instanciación de la tabla a la hora de realizar la migración de datos.

-Se agrega la Tabla Habitación_X_Estadía para que funcione como tabla puente entre la Estadía comprada por la agencia y las habitaciones del hotel que se han reservado, en esta tabla se indican no solo los identificadores de la estadía y la habitación en específico, sino también el costo de la habitación para que este sea sumado al costo total de la compra. Se entiende que dentro de una misma estadía comprada se pueden reservar varias habitaciones y que esas habitaciones podrán volver a ser reservadas en próximas estadías compradas.

-A la tabla Tipo_Habitacion se le cargan únicamente el código del tipo de habitación, que es utilizado como PK, y la descripción de cada tipo de habitación existente.

Consideraciones generales respecto a la creación de vistas:

- Se crea la vista “ItemsMixtos_Facturas” para poder ver la(s) Estadía(s) y Pasaje(s) que pudieran llegar a ser ítem de una misma factura, así como el precio de cada uno, y el monto total facturado al cliente al que se le vendió estos ítems.
- Se crea la vista “Estadías_Factura” para ver específicamente la o las estadías que fueron vendidas a un cliente, el precio total de la misma y el monto total facturado.
- Se crea la vista “Pasajes_Factura” para ver específicamente el o los pasajes que fueron vendidos a un cliente, el precio total de cada uno y el monto total facturado.
- Se crea la vista “ItemsMixtos_Compras” para poder ver la Estadía y el o los Pasajes que pudieran llegar a ser ítem de una misma compra, y el costo total abonado por la agencia.
- Se crea la vista “Compra_Estadía” para ver específicamente cada estadía individual que fue comprada por la agencia, junto con las fechas, el costo total, la fecha de compra, y el hotel referente.
- Se crea la vista “Pasaje_Estadía” para ver específicamente cada pasaje comprado por la agencia, el código de vuelo, y el costo total de compra.
- Se crea la vista “Vuelos_Programados” para ver la lista de vuelos comprados por la agencia, junto con el origen y destino, fechas de salida y llegada, y la aerolínea.

Consideraciones generales respecto a la creación de triggers:

- Se crea el trigger “sumar_pasaje_a_costo_total” para que, por cada nueva instanciación de la tabla “Pasaje”, el costo del mismo se sume al costo total de la compra asociada a la adquisición del pasaje.
- Se crea el trigger “sumar_hab_a_costo_total” para que, análogamente al trigger anterior, se sume el costo de cada habitación reservada al costo total de la compra asociada a la reserva.
- Se crea el trigger “crear_compra_tras_instancia_de_pasaje” para que, una vez que se instancia la tabla “Pasaje”, se instancie la tabla “Compra” con el código de compra insertado en la instancia de “Pasaje”, en el caso de que el registro dentro de la tabla “Compra” aun no exista. Este trigger no resulta necesario para la migración de datos de la tabla maestra ya que cada pasaje tiene ya asociado una compra.
- Se crea el trigger “crear_compra_tras_instancia_de_estadia” para que, de forma análoga al trigger anterior, se instancie la tabla “Compra” con el código de compra insertado en la instancia de “Estadía” en el caso de que la compra aun no exista. Tampoco resulta necesario para la migración de datos.

-Se crea el trigger “cargar_costo_hab_x_estadia” para que se cargue de forma automática el costo correcto de cada habitación reservada, y así evitar errores que pueden surgir con la carga manual. Esto no es necesario para la migración de datos.

-Se crea el trigger “cargar_precio_hab_en_factura” para que se cargue el precio de venta de cada habitación vendida en la estadía facturada y la factura correspondiente (se suma al monto total), de forma automática.

-Se crea el trigger “cargar_precio_pas_en_factura” para que se cargue el precio de venta de cada pasaje vendido en la factura correspondiente, sumándose al monto total.

-Se crea el trigger “cargar_precio_hab_vendida” para asegurarnos que el precio de cada habitación en la tabla Habitación_Vendida sea el correcto, previniendo errores de carga manual. No resulta necesario en migración de datos.

-Se crea el trigger “cargar_precio_pas_vendido” para asegurarnos que el precio de cada pasaje en la tabla Pasaje_Facturado sea el correcto, previniendo errores de carga manual. No resulta necesario en migración de datos.

-Se crea el trigger “cargar_fechas_check_in_out” para que se carguen automáticamente los datos de fecha de check in y check out de cada estadía vendida, en base a la fecha de inicio de la estadía que previamente fue comprada por la agencia (fecha check in) y la cantidad de noches reservadas, que se le suman a la fecha de inicio (check out).

Consideraciones generales respecto a la creación de stored procedures:

-Se crea la procedure “cargarDatosEnCliente” para la migración de datos de la Tabla Maestra a la Tabla “Cliente”. Al no encontrar un campo que permita identificar de forma unívoca a cada cliente ingresado en la Tabla Maestra, se optó por seleccionar la combinación CLIENTE_NOMBRE, CLIENTE_APELLIDO y CLIENTE_DNI para obtener cada cliente distinto y cargarlo en la Tabla “Cliente”.

-Se crea la procedure “cargarDatosEnSucursal” para la migración de datos de la Tabla Maestra a la Tabla “Sucursal”. Al no encontrar un campo que permita identificar de forma unívoca a cada sucursal ingresada en la Tabla Maestra, se optó por seleccionar la combinación SUCURSAL_DIR, SUCURSAL_MAIL y SUCURSAL_TEL para obtener cada sucursal distinta y cargarla en la Tabla “Sucursal”, a pesar de que estos son campos cuyos valores pueden ir variando con el tiempo.

-Se crea la procedure “cargarDatosEnFactura” para la migración de datos en la tabla “Factura”, utilizando como PK a FACTURA_NRO para identificar cada factura distinta que se presenta en los datos de la Tabla Maestra.

-Se crea la procedure “cargarDatosEnCompra” para la migración de datos en la tabla “Compra”, utilizando como PK a COMPRA_NUMERO para identificar cada compra distinta que se presenta en los datos de la Tabla Maestra.

-Se crea la procedure “cargarDatosEnHotel” para la migración de datos en la tabla “Hotel”. Al no encontrar un campo que permita identificar de forma unívoca a cada hotel ingresado en la Tabla Maestra, se optó por seleccionar la combinación EMPRESA_RAZON_SOCIAL, HOTEL_CALLE

y HOTEL_NRO_CALLE para obtener cada hotel distinto y cargarlo en la Tabla “Hotel”, a pesar de que estos son campos cuyos valores pueden ir variando con el tiempo.

-Se crea la procedure “cargarDatosEnEstadia” para la migracion de datos en la tabla “Estadía”. Utilizando como PK para identificar a cada estadía distinta al campo ESTADIA_CODIGO. Se verifico además que el campo FACTURA_NRO sea nulo para que solo se tomen las estadías compradas por la agencia, que luego pueden ser vendidas a clientes.

-Se crea la procedure “cargarDatosEnTipoHabitacion” para la migracion de datos en la tabla “Tipo_Habitacion”, utilizando como PK para identificar a cada tipo de habitación al campo “TIPO_HABITACION_CODIGO”.

-Se crea la procedure “cargarDatosEnHabitacion” para la migracion de datos en la tabla “Habitación”. Utilizando como campos para identificar cada habitación distinta los mismos que se utilizan en la procedure “cargarDatosEnHotel”.

-Se crea la procedure “cargarDatosEnHabitación_X_Estadia” para la migracion de datos en la tabla “Habitación_X_Estadia”. Utilizando una combinación de datos pertenecientes a cada hotel y habitación para identificar univocamente las habitaciones reservadas.

-Se crea la procedure “cargarDatosEnEstadiaFacturada” para la migracion de datos en la tabla “Estadía_Facturada”. Utilizando, como en otras tablas anteriormente nombradas, una combinacion de campos para identificar cada registro univoco de la tabla.

-Se crea la procedure “cargarDatosEnHabitacionVendida” para la migracion de datos en la tabla “Habitación_Vendida”. Utilizando, como en otras tablas anteriormente nombradas, una combinacion de campos para identificar cada registro univoco de la tabla.

-Se crea la procedure “cargarDatosEnAvion” para la migracion de datos en la tabla “Avion”. Utilizando al campo AVION_IDENTIFICADOR como PK.

-Se crea la procedure “cargarDatosEnButaca” para la migración de datos en la tabla “Butaca”. Utilizando, como en otras tablas anteriormente nombradas, una combinacion de campos para identificar cada registro univoco de la tabla.

-Se crea la procedure “cargarDatosEnRutaAerea” para la migracion de datos en la tabla “Ruta_Aerea”. Utilizando, como en otras tablas anteriormente nombradas, una combinacion de campos para identificar cada registro univoco de la tabla.

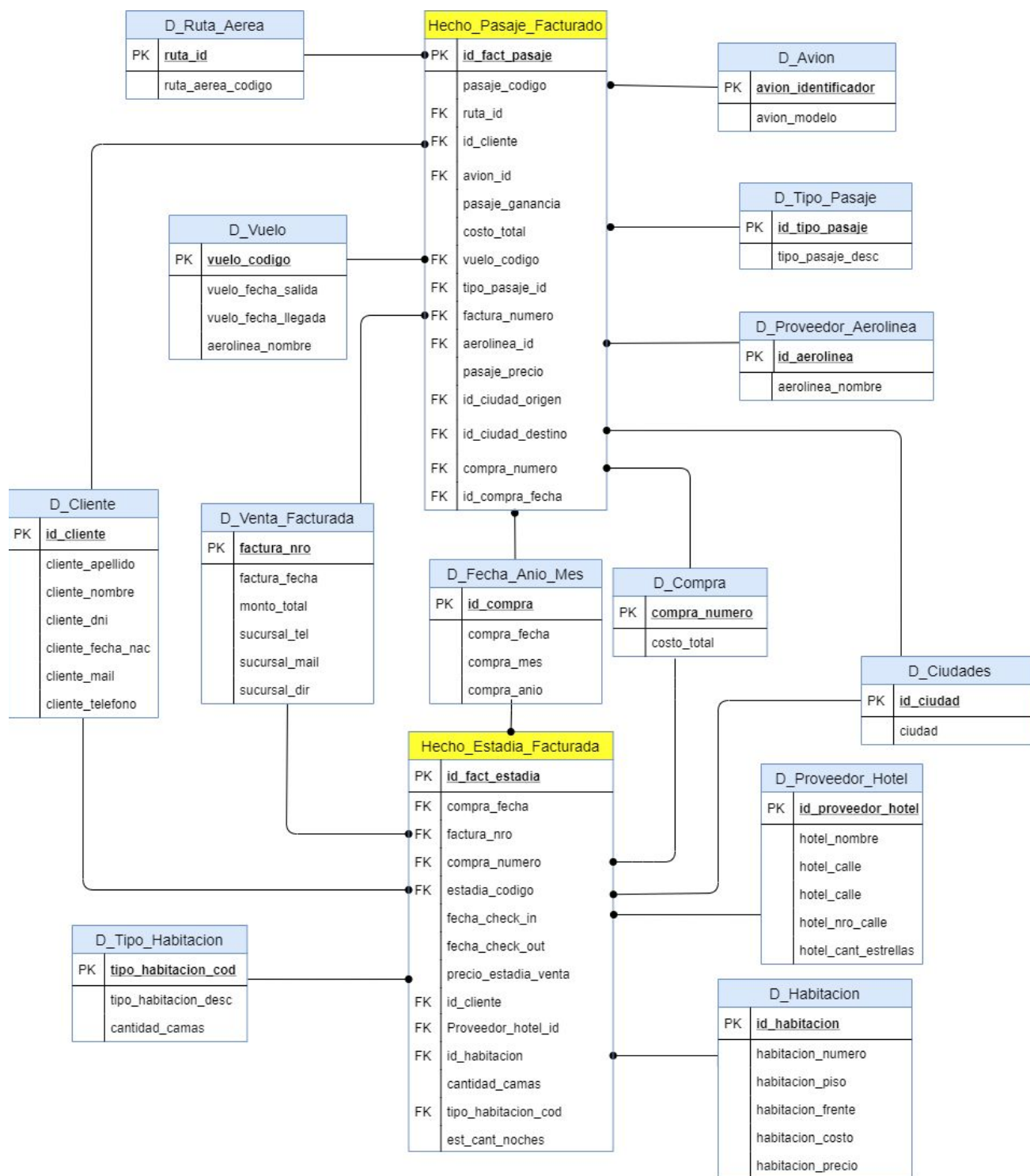
-Se crea la procedure “cargarDatosEnVuelo” para la migracion de datos en la tabla “Vuelo”. Utilizando, como en otras tablas anteriormente nombradas, una combinacion de campos para identificar cada registro univoco de la tabla.

-Se crea la procedure “cargarDatosEnPasaje” para la migracion de datos en la tabla “Pasaje”. Utilizando el campo “PASAJE_CODIGO” como PK.

-Se crea la procedure “cargarDatosEnPasajeFacturado” para la migracion de datos en la tabla “Pasaje_Facturado”. Utilizando, como en otras tablas anteriormente nombradas, una combinacion de campos para identificar cada registro univoco de la tabla.

Consideraciones respecto al Modelo de Inteligencia de Negocios.

Para lograr un seguimiento más sencillo de nuestro Modelo BI y complementar las decisiones tomadas, decidimos adjuntar nuestro modelo esquemático de las Dimensiones y las Tablas de Hechos a continuación.



- Primeramente, siguiendo las reglas del enunciado, las Dimensiones a crear y las Tablas de Hechos estarán en el mismo esquema en el cual se encuentran las Tablas Relacionales.
- Se parte del concepto del modelo estrella, utilizando esta técnica para hacer corresponder un modelo multidimensional sobre una base de datos de tipo relacional.
- Las tablas de dimensiones y hechos se encuentran relacionadas entre sí por medio de FKs.
- Se crearon las dimensiones requeridas por enunciado y algunas que se consideraron que pueden llegar a ser de ayuda a la hora de realizar consultas.
- Se decidió que las dimensiones D_Cliente, D_Venta_Facturada, D_Fecha_Año_Mes, D_Compra y D_Ciudades sean compartidas por ambas tablas de hechos por tener atributos en común.
- Teniendo en cuenta nuestro Modelo Relacional, con proveedores de hotelería separados de proveedores de aerolíneas, se decidió crear dos dimensiones de proveedores, una para pasaje (D_Proveedor_Aerolinea) y otra para estadía (D_Proveedor_Hotel), para que no queden campos en null. Aprovechando esto, se agregaron más atributos en caso de querer hacer consultas específicas para cada proveedor. Se creó un ID para D_Proveedor_Aerolinea que no existía en el Modelo Relacional (sólamente existía el nombre), para poder fácilmente identificarlos unívocamente.
- D_Proveedor_Hotel se cargará con los datos de la tabla “Hotel”
- Para modelar la dimensión año y mes requerida por enunciado se crea la dimensión D_Fecha_Año_Mes la cual su PK es id_fecha, que se refiere al número de compra, para cumplir unicidad. Los atributos compra_año y compra_mes son de tipo smallint y tinyint respectivamente para ocupar menos espacio en memoria. Se carga con datos de la tabla “Compra”.
- Se crea la dimensión “D_Cliente” y se crea un Stored Procedure para cargar con información perteneciente a la tabla “Cliente” del modelo relacional.
- Se crea la dimensión “D_Venta_Facturada” y se crea un Stored Procedure para cargar con información que proviene de las tablas “Factura” y “Sucursal” del modelo relacional.
- Se crea la dimensión “D_Compra” y se crea un Stored Procedure para cargar con información que proviene de la tabla “Compra” del modelo relacional.
- Se crea la dimensión “D_Ruta_Aerea” y se crea un Stored Procedure para cargar con los datos de la tabla “Ruta_Aerea”.
- Se crea la dimensión “D_Vuelo” y se crea un Stored Procedure para cargar con los datos de la tabla “Vuelo”.
- Se crea la dimensión “D_Avion” y se crea un Stored Procedure para cargar con los datos de la tabla “Avion”.
- Se crea la dimensión “D_Tipo_Pasaje”, tomando datos de la tabla Butaca, tales como el ID de la butaca como el ID del tipo de pasaje, y el tipo de butaca, como la descripción del tipo de pasaje.
- Se crea la dimensión “D_Ciudades”, y se crea un Stored Procedure para cargar datos de las ciudades origen y destino de la tabla “Ruta_Aerea” indistintamente, con un UNION sobre esa misma tabla.

- Se crea la tabla de hechos “Hecho_Pasaje_Facturado” y se crea un Stored Procedure para cargar con datos de múltiples tablas del modelo relacional y algunas dimensiones. Utilizamos un order by para agilizar la ejecución.

- Se crea la dimensión “D_Hotel” y se crea un Stored Procedure para cargarla con los datos de la tabla relacional “Hotel”.

- Se crea la dimensión “D_Tipo_Habitacion” y se crea un Stored Procedure para cargar con datos de la tabla “Tipo_Habitacion”. Calculamos cantidad_cameras a partir del dato “tipo_habitacion_desc” tomando las siguientes consideraciones:

- Si la habitación es simple, tiene 1 cama.
- Si la habitación es doble, tiene 2 camas.
- Si la habitación es triple, tiene 3 camas.
- Si la habitación es cuádruple, tiene 4 camas.
- Si la habitación es king, tiene 1 cama.

- Se crea la dimensión “D_Habitacion” y se crea un Stored Procedure para cargarla con datos de la tabla relacional “Habitacion”

- Se crea la tabla de hechos “Hecho_Estadia_Facturada” y se crea un Stored Procedure para cargarla con datos de múltiples tablas del modelo relacional. También ordenamos el select utilizando un order by para agilizar la ejecución.

- Calculamos $\text{pasaje_ganancia} = \text{pasaje_precio} - \text{costo_total}$, que se traduce como el precio de venta del pasaje menos el precio de compra original. Consideramos que es sumamente importante visualizar el resultado de esta cuenta, ya sea positivo (hubo ganancia) o negativo (hubo pérdida), de cara a las decisiones que se podrían tomar a futuro.

- Agregamos Foreign Keys a ambas tablas de hechos para relacionarlas con sus respectivas dimensiones.

- Finalmente se ejecutan todos los stored procedures para cargar todas las tablas con los datos previamente mencionados.