

TRABAJO PRÁCTICO DE GESTIÓN DE DATOS

GRUPO 32: NAME IS NULL

1C - 2020

Página 3	Explicación de cambios y alternativas descartadas de migración
Página 5	DER Migración
Página 6	Explicación de cambios y alternativas descartadas de modelo de BI
Página 8	DER Tabla de Hechos de Pasajes
Página 9	DER Tabla de Hechos de Estadías

Explicación de cambios y alternativas descartadas

Migración de tabla maestra:

Para realizar la migración correctamente tuvimos que modificar el DER realizado en la entrega anterior, ya que notamos algunas inconsistencias en nuestro modelo.

Para empezar, notamos que el DNI de los clientes no podía ser PRIMARY KEY ya que no cumplía con unicidad, es decir, dos clientes podían tener el mismo DNI. De haber existido una columna "tipo DNI", hubiéramos hecho que la PRIMARY KEY sea (DNI, tipo DNI), pero al no existir tal campo decidimos crear un IDENTITY que identifique puntualmente a cada cliente.

Decidimos agregar con el mismo razonamiento un IDENTITY en la tabla RUTA_AEREA, ya que nos dimos cuenta que mediante la misma ruta se podía llegar al mismo destino. Con un IDENTITY podemos diferenciar dentro de la ruta, cual es el origen y el destino del avión.

Luego de evaluar las diferentes alternativas, preferimos que el cliente sea guardado en la factura y no en la estadía como habíamos planteado anteriormente, ya que nos pareció más conveniente.

En la tabla sucursal nos pareció adecuado eliminar el campo "sucursal código" ya que podíamos utilizar como PRIMARY KEY al mail de cada sucursal. Dado que cada sucursal en su mail contenía el ID asociado pensamos que podría ser bueno utilizar ese número como PK en vez del mail entero, pero descartamos esta alternativa porque no lo vimos necesario.

Notamos que la forma correcta de identificar una habitación es referenciándola mediante su número y el hotel al que corresponde. Por lo tanto, modificamos nuestro modelo anterior teniendo en cuenta que la PRIMARY KEY debería ser (habitación número, hotel código). Para esto utilizamos un IDENTITY en la tabla hotel que asigne a cada uno una clave primaria válida.

Agregamos una tabla ítems ya que cada compra puede contar con muchos ítems dentro, esta tabla se asocia a la tabla compra a través de compra_numero.

Modificamos las relaciones que mantenía la tabla pasaje, de forma que ahora el pasaje esté incluido en la compra (si la empresa lo adquiere) o en la factura (si la empresa lo está vendiendo).

Además, descartamos la idea de utilizar cursores como en el siguiente ejemplo ya que ralentizaba el sistema.

```
/*CREATE PROCEDURE migracionAviones
AS
BEGIN
```

```

        DECLARE avionesCursor

        CURSOR    FOR    SELECT    AVION_IDENTIFICADOR,    AVION_MODELO    from
gd_esquema.Maestra

        DECLARE @avion_identificador NVARCHAR(50) , @avion_modelo NVARCHAR(50)
        OPEN avionesCursor
        FETCH NEXT FROM avionesCursor INTO @avion_identificador, @avion_modelo


        WHILE @@FETCH_STATUS = 0
            BEGIN
                IF (NOT EXISTS(SELECT 1 FROM AVION WHERE AVION_IDENTIFICADOR
= @avion_identificador))
                    BEGIN
                        INSERT INTO AVION
                        VALUES (@avion_identificador, @avion_modelo)

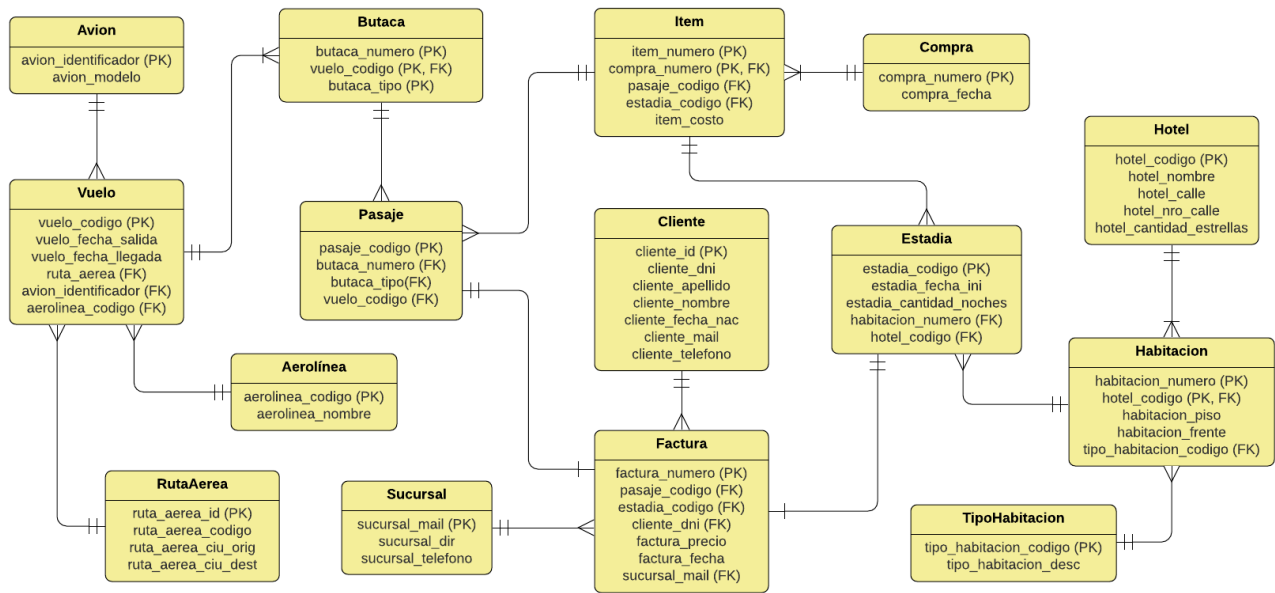
                        FETCH    NEXT    FROM    avionesCursor    INTO
@avion_identificador, @avion_modelo
                    END;
                FETCH NEXT FROM avionesCursor INTO @avion_identificador,
@avion_modelo
            END;

        CLOSE avionesCursor;
        DEALLOCATE avionesCursor;
END;
GO*/

```

Por último, decidimos realizar el script con un formato modular, es decir stored procedures separados que se ejecutan todos dentro del mismo procedure sin cursores, ya que probamos realizando un procedure con un solo cursor que abarque toda la migración, pero resultó ser más lento en comparación.

DER migración de datos:



Migración de tabla maestra:

Realizamos la tabla de hechos conforme a lo requerido por el modelo STAR, esta tabla se compone con las PRIMARY KEYS de cada una de las tablas de dimensiones creadas.

Las tablas de dimensión se nutren de los datos obtenidos de la migración realizada en base a la tabla maestra otorgada por la cátedra.

Utilizamos dos tablas de hechos, una para Estadías (F_Estadía) y otra para pasajes (F_Pasaje) para evitar tener nulos en las PRIMARY KEYS, esto mismo sucedía al querer insertar todo en una misma tabla de hechos. Consideramos la utilización de una sola tabla a pesar de esto (insertando en lugar de NULL un caracter que indique que el registro estaba vacío) pero entendimos que no debíamos cargar hechos inexistentes en una dimensión, por esta razón decidimos utilizar las dos tablas mencionadas anteriormente.

Tablas de hechos:

Contiene las PKs de las tablas dimensiones. Para poder ordenar la información de forma adecuada, las tablas de hechos contienen las columnas Cantidad_facturado, Cantidad_comprado, Monto_total_facturado, Monto_total_comprado, de esta forma podremos filtrar por los parámetros que elijamos para obtener información sobre cualquiera de estas columnas.

Se dividen en dos para evitar insertar nulos dentro de las PK. En el diagrama se encuentran graficadas repetidamente las dimensiones en cada una (como D_Clientes o D_Sucursales) sólo para un mayor entendimiento del diseño, es claro que ambas dimensiones junto con todas las que aparecen repetidas sólo existen una vez dentro del modelo ya que hacen referencia a la misma tabla.

Tablas de dimensiones:

Creamos dimensiones para todas aquellas tablas del modelo OLTP que contenían información que consideramos relevante para el modelo OLAP. En algunos casos, la información existente nos permitió inferir otros datos que fueron calculados e insertados en cada tabla de dimensión, como la edad del cliente o la cantidad de vuelos que salen o entran de una ciudad determinada.

Las dimensiones creadas son:

1. D_Fecha_Facturación: Almacena la fecha de cada venta realizada.
2. D_Fecha_Compra: Almacena la fecha de cada compra realizada.
3. D_Ruta_Aerea
4. D_Tipo_de_Butaca: Con su correspondiente vuelo.
5. D_Cliente
6. D_Tipo_de_Habitación: Calculamos la cantidad de camas según el código de habitación, asumiendo que existe una relación cantidad/código.
7. D_Ciudad_Entrante y D_Ciudad_Saliente: Dimensiones creadas para conocer, por ejemplo, cuántos vuelos entran y salen de una determinada ciudad.
8. D_Sucursal
9. D_Aviones
10. D_Proveedores: Muestra el nombre del proveedor, ya sea una aerolínea o una cadena de hoteles, y el tipo de servicio que brinda el mismo.

Finalmente, creamos dos VIEWS para constatar que los datos hayan sido ingresados correctamente según los requerimientos dictados por la cátedra.

Diagrama de la tabla de hechos de Pasajes:

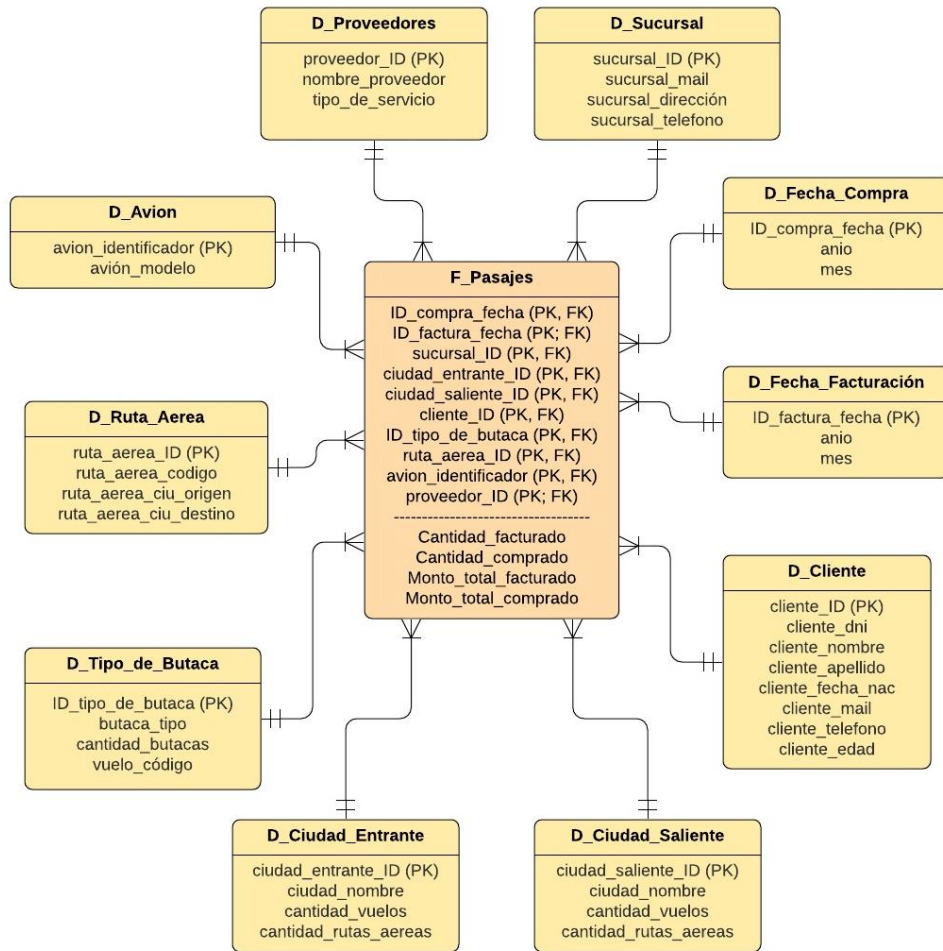


Diagrama de la tabla de hechos de Estadías:

