

## Estrategia del Trabajo Práctico “ZAFFA\_FLIGHTS”.

Nombre y Apellido	Legajo	Contacto
Federico Bancalari	1637794	fedebanca97@gmail.com
Facundo Oviedo	1678577	oviedoe@yahoo.com
Pablo Matías Pardo	1538720	pardmatias@est.frba.utn.edu.ar

## Contenido

Modelo de datos .....	3
Introducción: .....	3
Consideraciones: .....	3
Funciones: .....	3
Tablas temporales: .....	3
Stored Procedures: .....	4
Índices: .....	4
DER: .....	4
Modelo de Inteligencia de Negocios (BI) .....	5
Introducción: .....	5
Consideraciones: .....	5
Funciones: .....	6
Vistas: .....	6
Stored Procedures: .....	6
Observaciones: .....	7
DER Modelo BI .....	7

## Modelo de datos

### **Introducción:**

El siguiente documento especifica las decisiones tomadas en el grupo, relacionado al procedimiento de la migración.

### **Consideraciones:**

Tabla Ruta: Consideramos que es una buena decisión crear un nuevo campo id que sea la pk de nuestra tabla, para identificar los tramos de las rutas que tienen combinación.

Tabla Hotel: En la tabla maestra encontramos que el hotel tiene información sobre su ubicación, pero no detalla la ciudad a la que pertenece, en cambio en la ruta nos encontramos con una ciudad, la cual nos pareció importante normalizar, permitiendo a futuro lograr el vínculo de la ciudad con el hotel. Existe la consulta pertinente a esta consideración hacia el staff de gestión de datos.

Tabla pasajes: En algunos casos la tupla de vuelo y butaca se repiten para distinto pasaje, esto no puede pasar porque la misma butaca en el mismo vuelo no debería ser vendida 2 veces. Por lo tanto, estos casos los consideramos anómalos, decidimos insertarlos en otra tabla para luego, si corresponde, poder tratarlos.

Creímos conveniente realizar la migración en un esquema diferente al de maestra para separar la lógica de los datos.

### **Funciones:**

Estas funciones se crean fuera del store procedure y tienen en común que su responsabilidad es la de normalizar datos.

1. Fx\_codigoEmpresa Descripción: Normalizamos la razón social de la empresa.
2. Fx\_codigoButaca Descripción: Normalizamos la descripción del tipo de butaca.
3. Fx\_codigoCiudad Descripción: Normalizamos la descripción de la ciudad.

### **Tablas temporales:**

Creemos conveniente crear tablas temporales para mejorar la velocidad en acceso a los datos dentro de ellas, relacionando los datos que estas tienen en común.

Realizando varios select encontramos que la información se agrupa en 3 tablas generales: estadías, vuelos y clientes. Esto logra recorrer todos los campos de la tabla maestra menos cantidad de veces, logrando mejorar el uso de recursos. Estos a su vez van a ser llamado dentro de un store procedure para que estén contenidos dentro de una transacción

```
#ESTADIAS_EN_GENERAL  
#VUELOS_EN_GENERAL  
#SOBRE_CLIENTES
```

### Stored Procedures:

Hicimos un procedimiento “ejecutar\_migracion”, que contiene las inserciones a todas las tablas de nuestro modelo, respetando el orden de las FK y utilizando las funciones ya creadas.

Primero creamos las tablas temporales, nos pareció muy importante incluir dentro de este procedure el único índice que necesitamos, finalizando con la ejecución de todos los insert que requieren lo anteriormente mencionado.

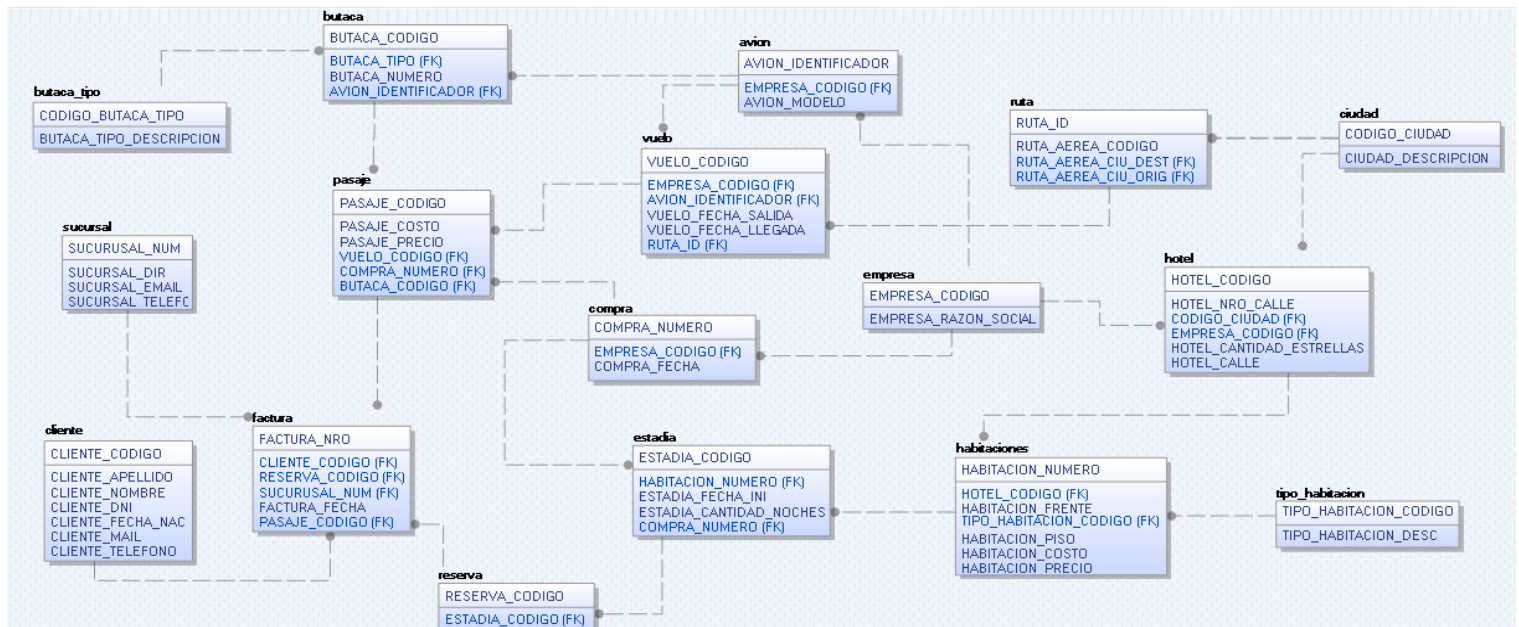
Por último, finalizada la migración, realizamos un último procedimiento para insertar los datos anómalos en la tabla “pasajes\_anomalos”, como lo mencionamos anteriormente

### Índices:

El único índice que tenemos en nuestro script se realizó para la tabla reservas sobre el campo estadia\_codigo logrando optimizar el tiempo del join en la migración de la factura.

El proceso de insert en la factura sin este índice tardaba aproximadamente 5 min, luego de implementarlo paso a tardar 6 segundos.

### DER:



## Modelo de Inteligencia de Negocios (BI)

### **Introducción:**

La siguiente sección, detalla las decisiones tomadas en lo que respecta al modelo de BI de nuestro TP

### **Consideraciones:**

Creamos dos tablas de hechos, una para pasajes y otra para estadías, no podíamos hacer una única pues las PK son mutuamente excluyentes. En ambas hay diferentes dimensiones que tomamos en cuenta para cada una (vistas en el DER del modelo estrella), de primeras están la mayoría de las que se piden en el enunciado, tomando en cuenta los siguientes detalles

- Año y mes no eran una tabla originalmente y no vimos necesidad de que la tenga pues la llenamos con la información de año y mes que sacamos tanto de la tabla de facturas como la de compra

- La dimensión cliente solicitada, tenía como PK CLIENTE\_NUM, esta era una columna que no servía para hacer ninguna intersección, pues siempre ocurre que un cliente compra, un pasaje o una estadía y nada más, por lo que decidimos, en su lugar usar como dimensión la edad del cliente (calculada al momento de hacer la compra, no la edad presente, porque creemos que era más relevante para tomar decisiones), esto nos impidió graficar en el DER al cliente como dimensión pues en ese caso es necesario que la PK de cliente vaya a la tabla de hechos, pero los datos lo sacamos de cliente

- Añadimos como dimensión a las tablas estadía y pasaje, de modo que la PK compuesta de las tablas hechos\_estadia y hechos\_pasaje sea UNIQUE

En la tabla de hechos de pasaje pusimos una única vez el campo ciudad y no ciudad\_origen y ciudad\_destino por separado, para que sea más cómodo de ver desde el select de la view

Usamos esquemas diferentes pues en la vida real el modelo OLAP y el OLTP suelen estar separados como bases de datos diferentes, pero a fines prácticos

tampoco vimos necesario crear una nueva base de datos independiente de la ya creada

Al momento de migrar las PK de las dimensiones a la tabla de hechos, hicimos un INNER JOIN con factura y no un OUTER JOIN, por lo que los pasajes y estadias que fueron comprados pero no vendidos, no los podemos ver al hacer un select \*, esta decisión está tomada en base a que consideramos que a nuestra empresa le son más relevantes los datos de los pasajes vendidos antes que los comprados, por lo tanto, la mayoría de la información que mostramos es sobre las ventas, y la que mostramos sobre las compras, no usa en ningún momento datos que no mostramos, si no los que si (por ejemplo año\_compra y mes\_compra o empresa\_codigo)

Un detalle extra a mencionar es que todas las estadias compradas fueron vendidas, así que sus promedios y cantidades dan igual

### **Funciones:**

fx\_cantidad\_camras: Esta función es necesaria al momento de calcular las intersecciones que toman en cuenta la cantidad de camas, donde uno ingresa el TIPO\_HABITACION\_CODIGO y la función devuelve si tiene una, dos, tres o cuatro según el tipo

### **Vistas:**

Generamos vistas para que pueda hacerse select \* de ellas según las distintos cortes entre dimensiones, simplificando incluso más las queries para comodidad del usuario

### **Stored Procedures:**

MigracionAModeloBI: Es el procedure que hace los cálculos e inserts necesarios para que nuestro modelo OLAP funcione como tiene que funcionar

### Observaciones:

Cuando hagan select sobre la tabla de hechos, van a notar que la edad de los clientes a veces son números muy altos, consideramos que es un error de enunciado y lo dejamos así.

### DER Modelo BI

