|  |
| --- |
| Luis Fernández Fernández y Nicolás Ruiz Martínez  10-25-2023 |

|  |
| --- |
| UAM EPS |
| Memoria Practica1 |
| Bases de Datos |

## Claves primarias y extranjeras de cada tabla:

aircrafts\_data (**aircraft\_code**, model)

airports\_data (**airport\_code**, airport\_name, city, coordinates, timezone)

boarding\_passes (**ticket\_no**  ticket\_flights. ticket\_no, **flight\_id** ticket\_flights.flight\_id, boarding\_no, seat\_no)

bookings (**book\_ref**, book\_date, total\_amount)

flights (**flight\_id**, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport  airports\_data.airport\_code, arrival\_airport  airports\_data.airport\_code, aircraft\_code  aircrafts\_data.aircraft\_code, actual\_departure, actual\_arrival)

seats (**seat\_no**, **aircraft\_code** aircrafts\_data.aircraft\_code, fare\_conditions)

ticket\_flights (**ticket\_no** tickets.ticket\_no, **flight\_id** flights.flight\_id, fare\_conditions, amount)

tickets (**ticket\_no**, book\_ref bookings.book\_ref, passenger\_id, passenger\_name, contact\_data)

## Diagrama del modelo relacional

Todas las querys se han obtenido en formato html para una mejor visualización de la información usando el comando \H en psql para cambiar la salida a dicho formato.

## Query 1:

Lista el número de reservas (bookings) que contengan un billete de ida y vuelta agrupadas por aeropuerto. Un billete es de ida y vuelta si el aeropuerto de origen del primer vuelo es idéntico al aeropuerto de destino del último vuelo. El resultado mostrará dos columnas con el código del aeropuerto de salida, así como el número de reservas que contienen billetes de ida y vuelta para ese aeropuerto. Igualmente, el resultado se ordenará de forma ascendente usando el atributo (departure airport).

No se ha podido hacer

## Query 2:

El precio de una reserva (booking.total amount) puede ser calculado a partir del precio de cada vuelo (ticket flights.amount). Crea una consulta que para cada reserva muestre el precio de la misma calculado a partir del valor almacenado en ticket flights.amount. La salida debe mostrar tres columnas conteniendo los atributos booking.book ref, booking.total amount y el valor calculado por vuestra consulta. Ordena la consulta de forma ascendiente usando el atributo bookigs.book ref.

Se ha hecho un select y otro y un join y blablablá

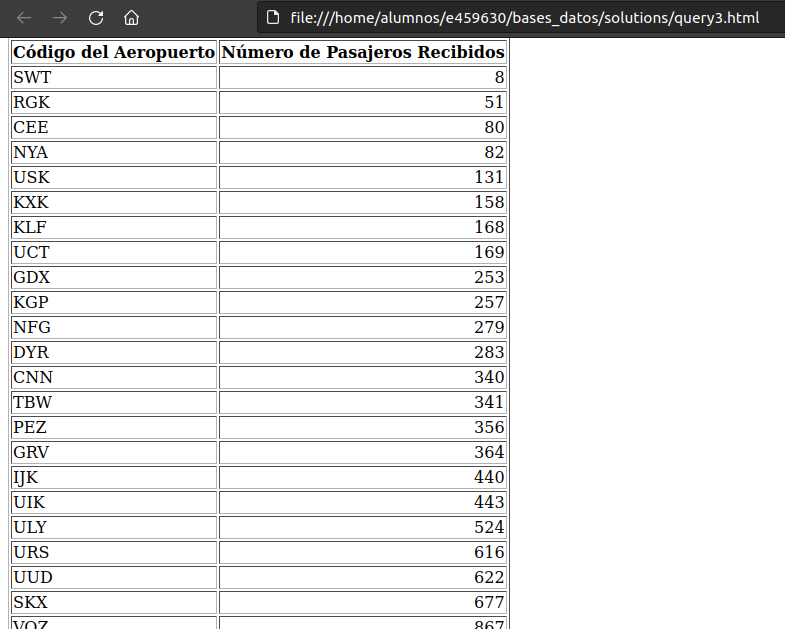
A screenshot of a computer

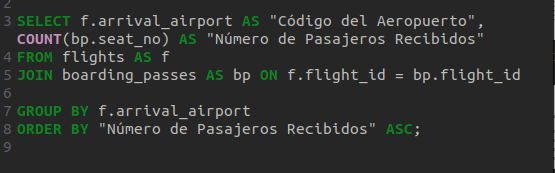
Description automatically generatedA screenshot of a computer

Description automatically generated

## Query 3:

Crea una consulta que muestre por pantalla el código de aeropuerto y el número de pasajeros recibidos, y ordena la misma de forma ascendiente por el número de pasajeros recibidos. Se considera que un pasajero ha volado al aeropuerto si se ha emitido una tarjeta de embarque. Una tarjeta de embarque para un vuelo del aeropuerto X al Y incrementa en 1 el número de pasajeros recibidos en el aeropuerto Y .





## Query 4:

Vuelo con más asientos vacíos. Muestra en la salida el atributo flight id y el número de asientos vacíos. En caso de empate muestra todos los vuelos que hayan empatado.

## Query 5:

Muestra reservas para los cuales no se haya emitido alguna de las tarjetas de embarque. La salida debe mostrar los atributos book ref y flight id. Ordenar el resultado de forma ascendente por la pareja (book ref, flight id).

## Query 6:

Por definición dos vuelos (flights) realizan el mismo trayecto si su número de vuelo (flight no) es el mismo. Se solicita el trayecto donde la media de los retrasos acumulados sea mayor. Se define retraso como la diferencia entre los valores actual arrival - scheduled arrival. Mostrar en la línea de salida flight no y el retraso medio. En caso de empate deben aparecer todos los trayectos con mayor retraso.