

Memoria Practica 2

Search

En este apartado se realizaron dos consultas a la base de datos, una compleja y la otra bastante sencilla.

Search Querys

La primera se encarga de encontrar todos los vuelos de determinado aeropuerto a otro en una fecha específica, incluyendo escalas de máximo un transbordo. Lo cual le añadía un poco de complejidad, al igual que había que enseñar la cantidad de asientos libres de los vuelos. En la primera práctica se implementó una vista que guardaba todos los asientos libres de cada vuelo, por lo que se usa esa vista en la query (cada vez que se corre el makefile, se crea la base de datos de nuevo y se corre exclusivamente esa query que es necesaria para el buen funcionamiento de esta práctica). La consulta es la siguiente:

```
"WITH connected_flights AS ("
"SELECT "
"f1.scheduled_departure AS departure_time_1, "
"f1.departure_airport AS departure_airport_1, "
"f2.scheduled_arrival AS arrival_time_2, "
"f2.arrival_airport AS arrival_airport_2, "
"f1.flight_id AS flight_id_1, "
"f2.flight_id AS flight_id_2, "
"1 AS num_flights_connection "
"FROM flights f1 "
"JOIN flights f2 ON f1.arrival_airport = f2.departure_airport "
"WHERE f1.departure_airport = ? "
"      AND f2.arrival_airport = ? "
"      AND DATE(f1.scheduled_departure) = ? "
"      AND f2.scheduled_arrival < f1.scheduled_departure + INTERVAL '24 hours' "
"      AND f2.scheduled_arrival > f1.scheduled_departure "
"      AND f2.scheduled_departure > f1.scheduled_arrival), "
"one_way_flights AS ( "
"SELECT "
"      scheduled_departure AS departure_time_1, "
"      departure_airport AS departure_airport_1, "
"      scheduled_arrival AS arrival_time_2, "
"      arrival_airport AS arrival_airport_2, "
"      flight_id AS flight_id_1, "
"      0 AS flight_id_2, "
"      0 AS num_flights_connection "
"FROM flights "
"WHERE departure_airport = ? AND arrival_airport = ? "
"      AND DATE(scheduled_departure) = ? ), "
"union_table_flights AS ( "
"SELECT * FROM connected_flights "
"UNION "
"SELECT * FROM one_way_flights "
") "
"SELECT "
"      flight_id_1, "
"      flight_id_2, "
"      departure_time_1, "
"      departure_airport_1, "
"      arrival_time_2, "
"      arrival_airport_2, "
"      LEAST(frs1.free_seats, frs2.free_seats) AS free_seats, "
"      num_flights_connection, "
"      (arrival_time_2 - departure_time_1) as duration "
"FROM union_table_flights t1 "
"JOIN TABLE FREE SEATS frs1 ON t1.flight_id_1 = frs1.flight_id "
"LEFT JOIN TABLE FREE SEATS frs2 ON t1.flight_id_2 = frs2.flight_id "
"WHERE frs1.free_seats > 0 AND (frs2.free_seats IS NULL OR frs2.free_seats > 0) "
"ORDER BY (arrival_time_2 - departure_time_1) ASC"
```

Al final del día la query consiste en primero sacar todos los vuelos con escalas de máximo un transbordo. En esta agregamos los 3 valores (*from*, *to*, *date*) que nos proporciona el usuario, para poder encontrar lo que nos está solicitando. Además la cantidad de tiempo no puede durar más de 24 horas, por lo que agregamos esa condición también.

Lo segundo es buscar los vuelos directos en la fecha proporcionada. Después hacemos una unión de ambas tablas, ya que, las dos tienen la misma información.

Por último seleccionamos los datos de interés de los vuelos que imprimimos en la parte derecha del menú y algo muy importante es que en esta parte de la consulta hacemos el *join* con la vista *table_free_seats*. Así obtenemos cuantos asientos libres hay en los vuelos que se encontraron y si alguno está lleno, no lo mostramos en la respuesta.

Para la vista en detalle de los itinerarios usamos una query bastante sencilla, que es la siguiente:

```
"SELECT "  
"    f.flight_id, "  
"    f.aircraft_code, "  
"    f.scheduled_departure, "  
"    f.scheduled_arrival "  
"    FROM flights f "  
"    WHERE f.flight_id = ? OR f.flight_id = ?"
```

En esta solo accedemos directamente a *flights* y devolvemos los atributos que se nos piden (*flight_id*, *aircraft_code*, *scheduled_departure*, *scheduled_arrival*) de la tabla *flights*. Si el vuelo es directo él *OR* no hace mucha diferencia ya que busca en un índice que no existe.

Implementación en C

Para poder enseñar los datos en el menú tuvimos que hacer varios cambios en los archivos presentados por el profesor. El primero fue añadir un campo para agregar la fecha en el menú. Después de implementar esto, teníamos que usar *odbc* para hacer una conexión a la base de datos y así poder manipularlos con un lenguaje de programación (c). Definimos la query que íbamos a correr, además de juntarla con los parámetros dados por el usuario. Al combinarlos nos regresaba cada fila asociada con la consulta hecha por el usuario. Está la guardamos en la variable *choices* y después la proyectamos en la parte derecha del menú. Cada vez que guardamos una fila en *choices*, también guardamos el *flight_id_1*, el *flight_id_2* y los *num_connections*. Los cuales luego serán utilizados para hacer una visualización a detalles del itinerario.

De igual forma se implementó una función para observar el detalle del itinerario. Cuando se presiona el tabulador y presionamos *enter* en uno de los itinerarios de los vuelos, somos capaces de ver información a detalle del o de los vuelos que forman parte de ese itinerario.