

# **Universidad Nacional Autónoma de México**

## **Facultad de Ingeniería**

### **Bases de Datos Grupo 1**

#### **Ejercicios Tema VII**

**De los Cobos García Carlos Alberto**

**12 de diciembre de 2022**

1. Considerar lo siguiente:

- Validar los scripts
  - Ejecutar los scripts sql y posteriormente, cargar los archivos csv
  - Agregar como resultado, la consulta en formato texto, y una captura de pantalla con el resultado obtenido.
  - En cada consulta, deberá mostrarse el tiempo que tomó en ejecutarse la consulta. (Investigar)
    - Para ello, basta con colocar \timing on.
  - Dentro del resultado de su consulta, deberá agregar una columna extra que incluya la fecha y hora del sistema y otra para el usuario que ejecutó la consulta. (Investigar)
    - Para agregar las columnas se coloca CURRENT\_TIMESTAMP y USER.
1. Indicar las ciudades que tienen más de un aeropuerto. Agregar su notación correspondiente en álgebra relacional.

```
registro_vuelos=# select city, current_timestamp, user from aeropuertos group by 1 having count(airport)>1;
```

| city         | current_timestamp             | user     |
|--------------|-------------------------------|----------|
| Jackson      | 2022-12-12 18:07:29.610758-06 | postgres |
| Springfield  | 2022-12-12 18:07:29.610758-06 | postgres |
| Albany       | 2022-12-12 18:07:29.610758-06 | postgres |
| Columbia     | 2022-12-12 18:07:29.610758-06 | postgres |
| New York     | 2022-12-12 18:07:29.610758-06 | postgres |
| Jacksonville | 2022-12-12 18:07:29.610758-06 | postgres |
| Wilmington   | 2022-12-12 18:07:29.610758-06 | postgres |
| San Diego    | 2022-12-12 18:07:29.610758-06 | postgres |
| Chicago      | 2022-12-12 18:07:29.610758-06 | postgres |
| Houston      | 2022-12-12 18:07:29.610758-06 | postgres |
| Charleston   | 2022-12-12 18:07:29.610758-06 | postgres |
| Portland     | 2022-12-12 18:07:29.610758-06 | postgres |
| Columbus     | 2022-12-12 18:07:29.610758-06 | postgres |
| Rochester    | 2022-12-12 18:07:29.610758-06 | postgres |

(14 filas)

Duración: 2.571 ms

## ALGEBRA RELACIONAL

- Nombre de las aerolíneas que no terminan en Inc. ni en Co. Agregar su notación correspondiente en álgebra relacional.

```
registro_vuelos=# select airline,current_timestamp,user
registro_vuelos-# from aerolineas
registro_vuelos-# where airline not like '%Inc.' and airline not like '%Co.';
```

| airline                     | current_timestamp             | user     |
|-----------------------------|-------------------------------|----------|
| JetBlue Airways             | 2022-12-12 18:15:40.409192-06 | postgres |
| Spirit Air Lines            | 2022-12-12 18:15:40.409192-06 | postgres |
| Atlantic Southeast Airlines | 2022-12-12 18:15:40.409192-06 | postgres |
| Virgin America              | 2022-12-12 18:15:40.409192-06 | postgres |

(4 filas)

Duración: 1.974 ms

## ALGEBRA RELACIONAL

- Indicar los nombres de los aeropuertos que estuvieron implicados en el vuelo que presentó el mayor retraso de llegada.

```
registro_vuelos=# select origin_airport,destination_airport,arrivale_delay,current_timestamp,user
registro_vuelos-# from vuelos
registro_vuelos-# where arrivale_delay=(select max(arrivale_delay) from vuelos);
```

```

origin_airport | destination_airport | arrivale_delay | current_timestamp | user
-----+-----+-----+-----+-----
BHM            | DFW                  | 1971           | 2022-12-12 19:15:10.383191-06 | postgres
(1 fila)

Duración: 10148.810 ms (00:10.149)

```

4. Mostrar aquella categoría (tabla artículo) que tiene el precio mínimo. La información debe estar agrupada (Implica que la consulta no sale con sólo selects y wheres).

```

datos_clase=# select categoria,current_timestamp,user
datos_clase=# from articulo
datos_clase=# where precio=(select min(precio) from articulo);

categoria | current_timestamp | user
-----+-----+-----
accesorios | 2022-12-12 19:25:13.230604-06 | postgres
(1 fila)

Duración: 74.537 ms

```

5. Se desea conocer el nombre de aquellas aerolíneas cuyo segundo carácter del iata code termina en X ó 9. Debe incluirse una columna que muestre dicha terminación.

```

registro_vuelos=# select airline,iata_code,current_timestamp,user
registro_vuelos=# from aerolineas
registro_vuelos=# where iata_code like '%X' or iata_code like '%9';

airline | iata_code | current_timestamp | user
-----+-----+-----+-----
Frontier Airlines Inc. | F9 | 2022-12-12 19:39:28.146613-06 | postgres
Virgin America | VX | 2022-12-12 19:39:28.146613-06 | postgres
(2 filas)

Duración: 84.134 ms

```

6. Proporcionar el nombre de los aeropuertos cuya latitud se encuentre entre 40 y 41, y su longitud sea menor que el promedio de la longitud. Nota: el promedio se toma de aquellas observaciones cuya latitud se encuentre entre 40 y 41.

```

registro_vuelos=# select airport, current_timestamp, user
registro_vuelos=# from aeropuertos
registro_vuelos=# where latitude between 40 and 41 and longitude=(select avg(latitude) from aeropuertos where latitude between 40 and 41);

```

```

airport | current_timestamp | user
-----+-----+-----
(0 filas)

Duración: 87.406 ms

```

7. ¿Cuántos aviones por aerolínea y día, fueron cancelados saliendo del aeropuerto de Honolulu?

```

registro_vuelos=# select count(flight_number),airline,day_of_week,current_timestamp,user
registro_vuelos=# from vuelos
registro_vuelos=# where cancelled = '1' and origin_airport in (select airport from aeropuertos where city='Honolulu')
registro_vuelos=# group by airline, day_of_week
registro_vuelos=# order by day_of_week;

```

```

count | airline | day_of_week | current_timestamp | user
-----+-----+-----+-----+-----
(0 filas)

```

8. Hacer un cross join entre la tabla cliente y la tabla aerolíneas. Obviamente ambas tablas forman parte de distintas BDs, debe encontrar la forma de hacerlo.

9. Cantidad de vuelos cancelados por día.

```

registro_vuelos=# select count(cancelled),day_of_week,current_timestamp,user
registro_vuelos=# from vuelos
registro_vuelos=# where cancelled = '1'
registro_vuelos=# group by day_of_week
registro_vuelos=# order by day_of_week;

```

```

count | day_of_week | current_timestamp | user
-----+-----+-----+-----+-----
21073 | 1 | 2022-12-12 21:13:35.845005-06 | postgres
15072 | 2 | 2022-12-12 21:13:35.845005-06 | postgres
10729 | 3 | 2022-12-12 21:13:35.845005-06 | postgres
12291 | 4 | 2022-12-12 21:13:35.845005-06 | postgres
8805 | 5 | 2022-12-12 21:13:35.845005-06 | postgres
8749 | 6 | 2022-12-12 21:13:35.845005-06 | postgres
13165 | 7 | 2022-12-12 21:13:35.845005-06 | postgres
(7 filas)

Duración: 2926.905 ms (00:02.927)

```

10. Seleccionar el nombre de los aeropuertos cuya segunda letra del iata code sea K ó X, sin usar operadores and, not u or. Puede usar alguna función propia de postgres.

```
registro_vuelos=# select airport,current_timestamp,user
registro_vuelos=# from aeropuertos
registro_vuelos=# where iata_code like any({'_K%','_X%'});
```

| airport                                 | current_timestamp             | user     |
|---|-------------------------------|----------|
| King Salmon Airport                     | 2022-12-12 21:21:36.994871-06 | postgres |
| Elko Regional Airport                   | 2022-12-12 21:21:36.994871-06 | postgres |
| General Mitchell International Airport  | 2022-12-12 21:21:36.994871-06 | postgres |
| Muskegon County Airport                 | 2022-12-12 21:21:36.994871-06 | postgres |
| Will Rogers World Airport               | 2022-12-12 21:21:36.994871-06 | postgres |
| Rock Springs-Sweetwater County Airport  | 2022-12-12 21:21:36.994871-06 | postgres |
| Texarkana Regional Airport (Webb Field) | 2022-12-12 21:21:36.994871-06 | postgres |

(7 filas)

Duración: 1.438 ms

11. Indicar el nombre(s) de la aerolínea cuya distancia de vuelo es la mayor.

```
registro_vuelos=# select distinct airline,current_timestamp,user
registro_vuelos=# from vuelos
registro_vuelos=# where distance=(select max(distance) from vuelos);
```

| airline | current_timestamp             | user     |
|---------|-------------------------------|----------|
| DL      | 2022-12-12 21:25:10.562767-06 | postgres |
| HA      | 2022-12-12 21:25:10.562767-06 | postgres |

(2 filas)

Duración: 4382.203 ms (00:04.382)

12. Indicar el nombre del aeropuerto de origen donde se presentó el mayor tiempo de vuelo.

```
registro_vuelos=# select distinct origin_airport,current_timestamp,user
registro_vuelos=# from vuelos
registro_vuelos=# where air_time=(select max(air_time) from vuelos);
```

| origin_airport | current_timestamp             | user     |
|----------------|-------------------------------|----------|
| JFK            | 2022-12-12 21:30:59.623735-06 | postgres |

(1 fila)

Duración: 4309.677 ms (00:04.310)