



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería



ALUMNO:

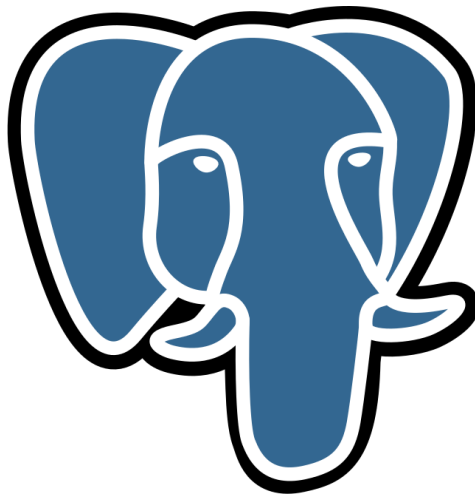
- López Mendoza Ricardo

MATERIA:

Bases de Datos

Tarea :

EJERCICIOS TEMA 7





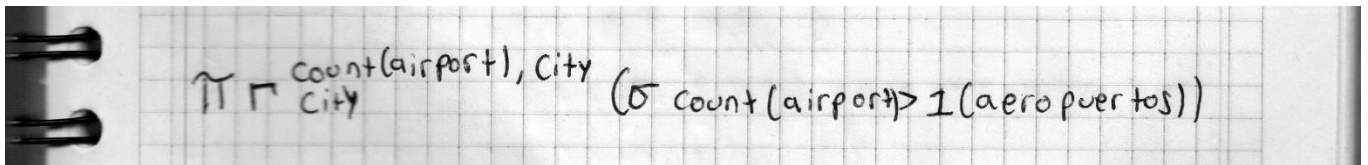
Bases de datos

Ejercicios tema VII Ing.
Fernando A.

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)
- 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)

1. Indicar las ciudades que tienen más de un aeropuerto. Agregar su notación correspondiente en álgebra relacional.



Resultado de la consulta

```
registro_vuelos=# Select COUNT(airport) NUM_AEROPUERTOS,city,(select now()) FECHA,(select current_user) usuario from
registro_vuelos-# aeropuertos group by city HAVING COUNT(airport)>1;
registro_vuelos
 num_aeropuertos | city | fecha | usuario
-----+-----+-----+-----
2 | Jackson | 2022-12-11 22:13:39.988328-06 | postgres
2 | Springfield | 2022-12-11 22:13:39.988328-06 | postgres
2 | Albany | 2022-12-11 22:13:39.988328-06 | postgres
2 | Columbia | 2022-12-11 22:13:39.988328-06 | postgres
2 | New York | 2022-12-11 22:13:39.988328-06 | postgres
2 | Jacksonville | 2022-12-11 22:13:39.988328-06 | postgres
2 | Wilmington | 2022-12-11 22:13:39.988328-06 | postgres
2 | San Diego | 2022-12-11 22:13:39.988328-06 | postgres
2 | Chicago | 2022-12-11 22:13:39.988328-06 | postgres
2 | Houston | 2022-12-11 22:13:39.988328-06 | postgres
2 | Charleston | 2022-12-11 22:13:39.988328-06 | postgres
2 | Portland | 2022-12-11 22:13:39.988328-06 | postgres
2 | Columbus | 2022-12-11 22:13:39.988328-06 | postgres
2 | Rochester | 2022-12-11 22:13:39.988328-06 | postgres
(14 filas)

Duración: 1.865 ms
```

Select COUNT(airport) NUM_AEROPUERTOS,city,(select now()) FECHA,(select current_user) usuario from
aeropuertos group by city HAVING COUNT(airport)>1;

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

π airline (
 σ airline not like '%Inc.' and
 σ airline not like '%Co.' (aerolineas))

```
registro_vuelos=# Select airline,(select now()) FECHA,(select current_user) usuario from
registro_vuelos=# aerolineas where airline not LIKE '%Inc.' and airline not like '%Co.';
registro_vuelos

```

airline	fecha	usuario
JetBlue Airways	2022-12-11 22:12:58.150932-06	postgres
Spirit Air Lines	2022-12-11 22:12:58.150932-06	postgres
Atlantic Southeast Airlines	2022-12-11 22:12:58.150932-06	postgres
Virgin America	2022-12-11 22:12:58.150932-06	postgres

(4 filas)

Duración: 0.580 ms

```
registro_vuelos=#
```

Codigo sql

Select airline,(select now()) FECHA,(select current_user) usuario from
 aerolineas where airline not LIKE '%Inc.' and airline not like '%Co.';

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

```
registro_vuelos=# Select airport aeropuertos,(select now()) FECHA,(select current_user) usuario from
registro_vuelos=# vuelos v join aeropuertos A
registro_vuelos=# On V.origin_airport= A.iata_code or V.destination_airport= A.iata_code
registro_vuelos=# Where v.late_aircraft_delay = (select max(late_aircraft_delay) from vuelos);
registro_vuelos

```

aeropuertos	fecha	usuario
Dallas/Fort Worth International Airport	2022-12-11 22:12:12.904044-06	postgres
Honolulu International Airport	2022-12-11 22:12:12.904044-06	postgres

(2 filas)

Duración: 2046.020 ms (00:02.046)

Codigo sql

Select airport aeropuertos,(select now()) FECHA,(select current_user) usuario from
 vuelos v join aeropuertos A
 On V.origin_airport= A.iata_code or V.destination_airport= A.iata_code
 Where v.late_aircraft_delay = (select max(late_aircraft_delay) from vuelos);

- 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,min(precio) precio_minimo ,(select now()) FECHA,(select current_user) usuario from
datos_clase=# articulo where precio=(select min(precio) from articulo)group by categoria;
          registro_vuelos
categoria | precio_minimo | fecha                | usuario
-----+-----+-----+-----
accesorios |          120 | 2022-12-11 22:30:19.38413-06 | postgres
(1 fila)

Duración: 0.972 ms
datos_clase=#
```

CODIGO SQL

```
Select categoria,min(precio) precio_minimo ,(select now()) FECHA,(select current_user) usuario from
articulo where precio=(select min(precio) from articulo)group by categoria;
```

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

```
          registro_vuelos
nombre      | terminacion | fecha                | usuario
-----+-----+-----+-----
Frontier Airlines Inc. | 9           | 2022-12-11 22:46:56.603078-06 | postgres
Virgin America         | X           | 2022-12-11 22:46:56.603078-06 | postgres
(2 filas)

Duración: 3.558 ms
registro_vuelos=#
```

CODIGO SQL

```
select airline nombre,substr(iata_code,2) terminacion,(select now()) FECHA,(select current_user) usuario
from
aerolineas where iata_code LIKE '_X' OR iata_code LIKE '_9';
```

- 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 aeropuertos,(select now()) FECHA,(select current_user) from
registro_vuelos=# Aeropuertos
registro_vuelos=# Where latitude between 40 and 41 and longitud<(select avg(longitude) from aeropuertos where latitude
between 40 and 41);
          registro_vuelos
aeropuertos | fecha                | current_user
-----+-----+-----
Arcata Airport | 2022-12-11 23:00:58.772194-06 | postgres
Elko Regional Airport | 2022-12-11 23:00:58.772194-06 | postgres
Central Nebraska Regional Airport | 2022-12-11 23:00:58.772194-06 | postgres
Yampa Valley Airport(Yampa Valley Regional) | 2022-12-11 23:00:58.772194-06 | postgres
Lincoln Airport(Lincoln Municipal) | 2022-12-11 23:00:58.772194-06 | postgres
Redding Municipal Airport | 2022-12-11 23:00:58.772194-06 | postgres
Salt Lake City International Airport | 2022-12-11 23:00:58.772194-06 | postgres
Valdez Airport | 2022-12-11 23:00:58.772194-06 | postgres
(8 filas)

Duración: 2.116 ms
registro_vuelos=#
```

CODIGO SQL

Select airport aeropuertos,(select now()) FECHA,(select current_user) from

Aeropuertos

Where latitude between 40 and 41 and longitude<(select avg(longitude) from aeropuertos where latitude between 40 and 41);

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

```
registro_vuelos=# Select count(flight_number),airline,day,(select now()) FECHA,(select current_user)
registro_vuelos-# From aeropuertos A join vuelos v
registro_vuelos-# On v.origin_airport=A.iata_code
registro_vuelos-# Where v.cancelled is not null and city='Honolulu'
registro_vuelos-# group by airline,day;
count | airline | day | fecha | current_user
-----+-----+-----+-----+-----
  80 | AA | 1 | 2022-12-11 23:39:38.284905-06 | postgres
  78 | AA | 2 | 2022-12-11 23:39:38.284905-06 | postgres
  78 | AA | 3 | 2022-12-11 23:39:38.284905-06 | postgres
  81 | AA | 4 | 2022-12-11 23:39:38.284905-06 | postgres
  79 | AA | 5 | 2022-12-11 23:39:38.284905-06 | postgres
  77 | AA | 6 | 2022-12-11 23:39:38.284905-06 | postgres
  78 | AA | 7 | 2022-12-11 23:39:38.284905-06 | postgres
  78 | AA | 8 | 2022-12-11 23:39:38.284905-06 | postgres
  75 | AA | 9 | 2022-12-11 23:39:38.284905-06 | postgres
  76 | AA | 10 | 2022-12-11 23:39:38.284905-06 | postgres
```

```
  2 | VX | 21 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 22 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 23 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 24 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 25 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 26 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 27 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 28 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 29 | 2022-12-11 23:39:38.284905-06 | postgres
  2 | VX | 30 | 2022-12-11 23:39:38.284905-06 | postgres
  1 | VX | 31 | 2022-12-11 23:39:38.284905-06 | postgres
(217 filas)

Duración: 1694.670 ms (00:01.695)
```

CODIGO SQL

Select count(flight_number),airline,day,(select now()) FECHA,(select current_user) USUARIO

From aeropuertos A join vuelos v

On v.origin_airport=A.iata_code

Where v.cancelled is not null and city='Honolulu'

group by airline,day;

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.

```
Select * from aerolineas cross join dblink('dbname=datos_clase','select * from cliente') as
cliente ( iata_code character varying, id_Cliente varchar(13) );
```

9. Cantidad de vuelos cancelados por día.

```
registro_vuelos=# Select count(flight_number) vuelos_cancelados,day DIA,(select now()) FECHA,(select current_user) USUARIO
registro_vuelos=# From vuelos
registro_vuelos=# Where cancelled is not null
registro_vuelos=# group by day;
registro_vuelos
vuelos_cancelados | dia | fecha | usuario
-----
189477 | 1 | 2022-12-11 23:48:28.985627-06 | postgres
195986 | 2 | 2022-12-11 23:48:28.985627-06 | postgres
190007 | 3 | 2022-12-11 23:48:28.985627-06 | postgres
190893 | 4 | 2022-12-11 23:48:28.985627-06 | postgres
189766 | 5 | 2022-12-11 23:48:28.985627-06 | postgres
191232 | 6 | 2022-12-11 23:48:28.985627-06 | postgres
187598 | 7 | 2022-12-11 23:48:28.985627-06 | postgres
193964 | 8 | 2022-12-11 23:48:28.985627-06 | postgres
194224 | 9 | 2022-12-11 23:48:28.985627-06 | postgres
189288 | 10 | 2022-12-11 23:48:28.985627-06 | postgres
190756 | 11 | 2022-12-11 23:48:28.985627-06 | postgres
190872 | 12 | 2022-12-11 23:48:28.985627-06 | postgres
195089 | 13 | 2022-12-11 23:48:28.985627-06 | postgres
188611 | 14 | 2022-12-11 23:48:28.985627-06 | postgres
192950 | 15 | 2022-12-11 23:48:28.985627-06 | postgres
195899 | 16 | 2022-12-11 23:48:28.985627-06 | postgres
191319 | 17 | 2022-12-11 23:48:28.985627-06 | postgres
191393 | 18 | 2022-12-11 23:48:28.985627-06 | postgres
193284 | 19 | 2022-12-11 23:48:28.985627-06 | postgres
195707 | 20 | 2022-12-11 23:48:28.985627-06 | postgres
189413 | 21 | 2022-12-11 23:48:28.985627-06 | postgres
192725 | 22 | 2022-12-11 23:48:28.985627-06 | postgres
193560 | 23 | 2022-12-11 23:48:28.985627-06 | postgres
185017 | 24 | 2022-12-11 23:48:28.985627-06 | postgres
187317 | 25 | 2022-12-11 23:48:28.985627-06 | postgres
187387 | 26 | 2022-12-11 23:48:28.985627-06 | postgres
191920 | 27 | 2022-12-11 23:48:28.985627-06 | postgres
191401 | 28 | 2022-12-11 23:48:28.985627-06 | postgres
179441 | 29 | 2022-12-11 23:48:28.985627-06 | postgres
178771 | 30 | 2022-12-11 23:48:28.985627-06 | postgres
103812 | 31 | 2022-12-11 23:48:28.985627-06 | postgres
(31 filas)
```

CODIGO SQL

```
Select count(flight_number) vuelos_cancelados,day DIA,(select now()) FECHA,(select current_user) USUARIO
From vuelos
Where cancelled is not null
group by day;
```

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

```
registro_vuelos=# SELECT DISTINCT( A.airline) AEROLINEA, (select now()) FECHA,(select current_user) USUARIO
registro_vuelos=# From aerolineas A join vuelos V
registro_vuelos=# On V.airline=A.iata_code
registro_vuelos=# Where distance=(select max(distance) from vuelos);
registro_vuelos
aerolinea | fecha | usuario
-----
Delta Air Lines Inc. | 2022-12-12 00:01:52.004169-06 | postgres
Hawaiian Airlines Inc. | 2022-12-12 00:01:52.004169-06 | postgres
(2 filas)

Duración: 2437.738 ms (00:02.438)
registro_vuelos=#
```

CODIGO SQL

```
SELECT DISTINCT( A.airline) AEROLINEA, (select now()) FECHA,(select current_user) USUARIO
```

```
From aerolineas A join vuelos V
```

```
On V.airline=A.iata_code
```

```
Where distance=(select max(distance) from vuelos);
```

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

```
registro_vuelos=# SELECT DISTINCT (airport) aeropuerto, (select now()) FECHA,(select current_user) USUARIO
registro_vuelos=# From vuelos V join aeropuertos A
registro_vuelos=# On V.origin_airport=A.iata_code
registro_vuelos=# Where air_time=(select max(air_time) from vuelos);
registro_vuelos
aeropuerto                                | fecha                                | usuario
-----+-----+-----
John F. Kennedy International Airportá(New York International Airport) | 2022-12-12 00:09:55.767113-06 | postgres
(1 fila)

Duración: 2504.482 ms (00:02.504)
registro_vuelos=#
```

CODIGO SQL

```
SELECT DISTINCT (airport) aeropuerto, (select now()) FECHA,(select current_user) USUARIO
```

```
From vuelos V join aeropuertos A
```

```
On V.origin_airport=A.iata_code
```

```
Where air_time=(select max(air_time) from vuelos);
```

