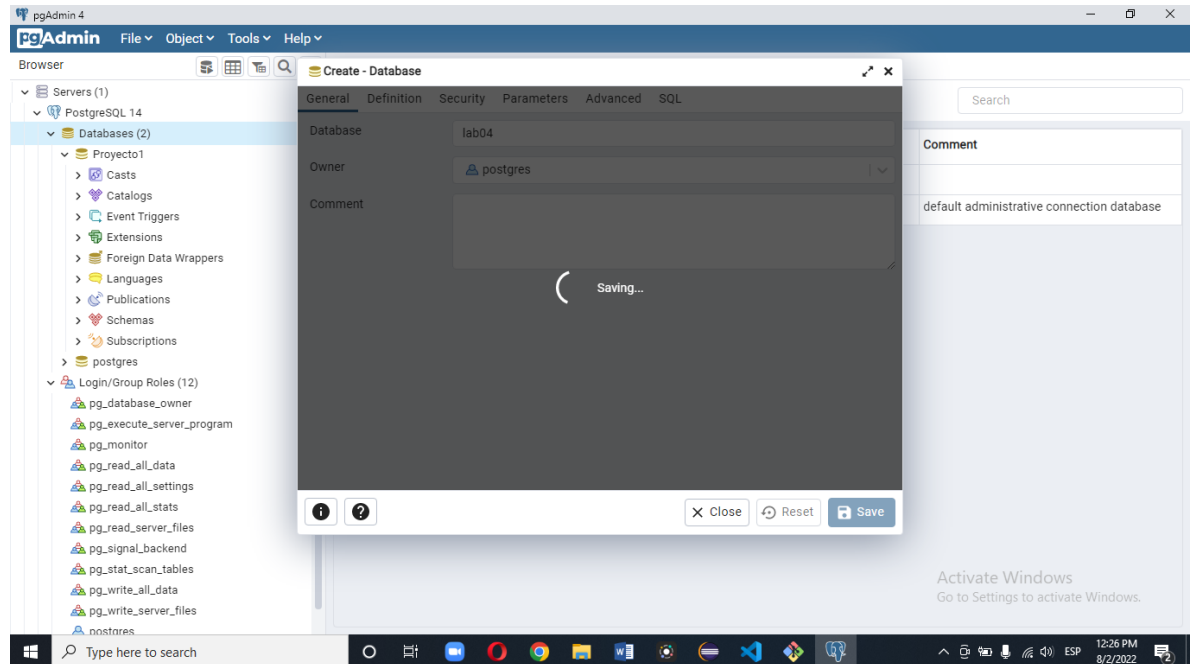


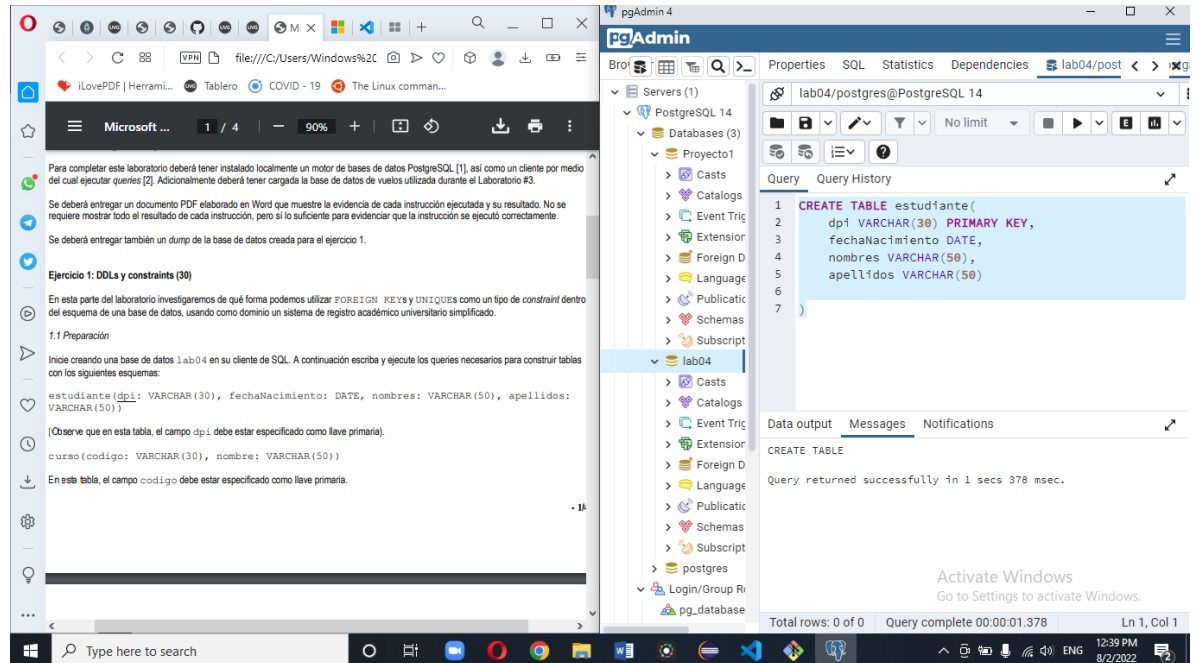
Laboratorio 04

DDL's y Constraints

1. Creación de la base de datos del lab.



Creación de la tabla estudiante.



Creación de la tabla curso.

The screenshot shows a Windows desktop with a PDF document open in Microsoft Word and pgAdmin 4. The PDF document contains the following text:

Adicionalmente se busca que el estudiante profundice y practique sobre los conceptos de queries utilizando el conjunto completo de características de consultas SQL, incluyendo JOINS, subqueries y agregaciones, para responder a consultas avanzadas sobre bases relacionales en PostgreSQL.

Introducciones generales y observaciones

Para completar este laboratorio deberá tener instalado localmente un motor de bases de datos PostgreSQL [1], así como un cliente del cual ejecutar queries [2]. Adicionalmente deberá tener cargada la base de datos de vuelos utilizada durante el Laboratorio #3.

Se deberá entregar un documento PDF elaborado en Word que muestre la evidencia de cada instrucción ejecutada y su resultado, requiere mostrar todo el resultado de cada instrucción, pero si lo suficiente para evidenciar que la instrucción se ejecutó correctamente.

Se deberá entregar también un dump de la base de datos creada para el ejercicio 1.

Ejercicio 1: DDLs y constraints (30)

En esta parte del laboratorio investigaremos de qué forma podemos utilizar FOREIGN KEYS y UNIQUES como un tipo de const del esquema de una base de datos, usando como dominio un sistema de registro académico universitario simplificado.

1.1 Preparación

Inicie creando una base de datos lab04 en su cliente de SQL. A continuación escriba y ejecute los queries necesarios para const con los siguientes esquemas:

```
estudiante(dpi VARCHAR(30), fechaNacimiento DATE, nombres VARCHAR(50), apellidos VARCHAR(50))
```

(Observe que en esta tabla, el campo dpi debe estar especificado como llave primaria).

```
curso(codigo VARCHAR(30), nombre VARCHAR(50))
```

En esta tabla, el campo codigo debe estar especificado como llave primaria.

pgAdmin 4 shows the following SQL queries:

```
CREATE TABLE estudiante(  
1  dpi VARCHAR(30) PRIMARY KEY,  
2  fechaNacimiento DATE,  
3  nombres VARCHAR(50),  
4  apellidos VARCHAR(50)  
5  )  
6  
7  
8  
9  CREATE TABLE curso(  
10  codigo VARCHAR(30) PRIMARY KEY,  
11  nombre VARCHAR(50)  
12  )
```

Query returned successfully in 229 msec.

Inserts de 5 estudiantes

The screenshot shows a Windows desktop with a PDF document open in Microsoft Word and pgAdmin 4. The PDF document contains the following text:

Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de los datos
Semestre
CC3088 Base de datos 1

Seguidamente escriba los INSERTS necesarios para registrarse a usted y a otros estudiantes y a este curso dentro de las tablas.

Recuerde que el código del presente curso es CC3088.

Note que, dado que el campo estudiante.dpi es llave, la base de datos le impide insertarse a usted dos veces.

1.2 Creación de constraints e integridad referencial (10 puntos)

A continuación, utilizando FOREIGN KEYS y UNIQUES defina el esquema de una tabla asignacion que contenga el dpi de estudiante, el código de un curso y la fecha de la asignación, y que no permita ingresar tuplas con un estudiante o curso inexistentes.

Adicionalmente no se debe permitir tener dos veces al mismo estudiante en el mismo curso con la misma fecha de asignación. Esto lograrse con PRIMARY KEYS, pero en este ejercicio debe lograrlo por medio de constraints UNIQUE.

1.3 Verificación de constraints (5 puntos)

A fin de dejar evidencia que las políticas implementadas se cumplen, construya y ejecute queries para:

1.3.1. Su asignación y la de otro estudiante a este curso el día de hoy;

1.3.2. El ingreso de una asignación con código de estudiante estudiante DPI 2413 94996 0108 a este curso (este estudiante la tabla de estudiantes, por lo que la inserción debe resultar en un error)

1.3.2. Su asignación a un curso con código FP2017-435 (este curso no existe en la tabla de cursos, por lo que la inserción debe resultar en un error)

pgAdmin 4 shows the following SQL query:

```
INSERT INTO estudiante(dpi, fechaNacimiento, nombres,  
VALUES(  
12  '1234567890',  
13  '2002/01/20',  
14  'Diego Andres',  
15  'Alonzo Medinilla'  
16  ),  
17  
18  
19  
20  
21  
22  
23
```

Query returned successfully in 266 msec.

Inserts del curso

The screenshot shows a web browser window on the left displaying a document from the Universidad del Valle de Guatemala. The document contains instructions for a database exercise, including writing INSERT statements and defining constraints. On the right, the pgAdmin 4 interface is open, showing the 'lab04/postgres@PostgreSQL 14' database. The 'Query' tab is active, displaying an SQL INSERT statement:

```
INSERT INTO curso(codigo, nombre)
VALUES(
  'CC3088',
  'Base de datos 1'
)
```

 The 'Messages' tab shows an error:

```
ERROR: duplicate key value violates unique constraint
"curso_pkey"
DETAIL: Key (codigo)=(CC3088) already exists.
SQL state: 23505
```

2. Creación de constraints para integridad referencial.

The screenshot shows the pgAdmin 4 interface with the 'Query' tab active. The SQL query displayed is:

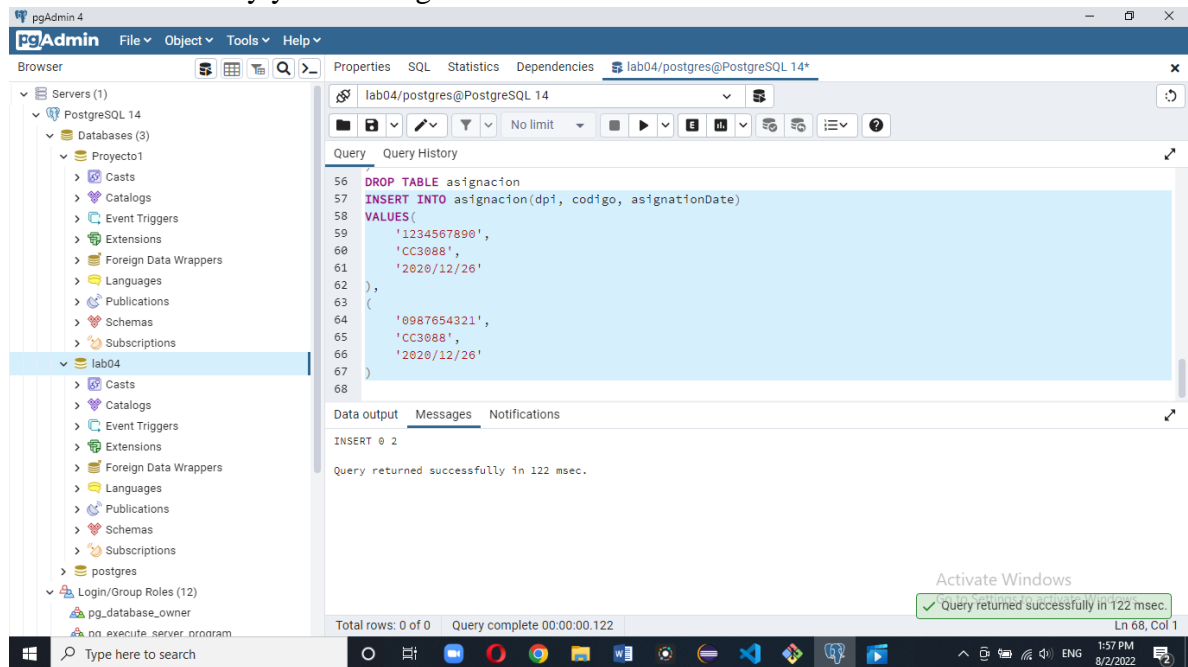
```
CREATE TABLE asignacion(
  dpi VARCHAR(30) NOT NULL,
  codigo VARCHAR(30) NOT NULL,
  asignacionDate DATE,
  FOREIGN KEY (dpi) REFERENCES estudiante,
  FOREIGN KEY (codigo) REFERENCES curso,
  CONSTRAINT dca UNIQUE (dpi, codigo, asignacionDate)
)
```

 The 'Messages' tab shows the successful execution of the query:

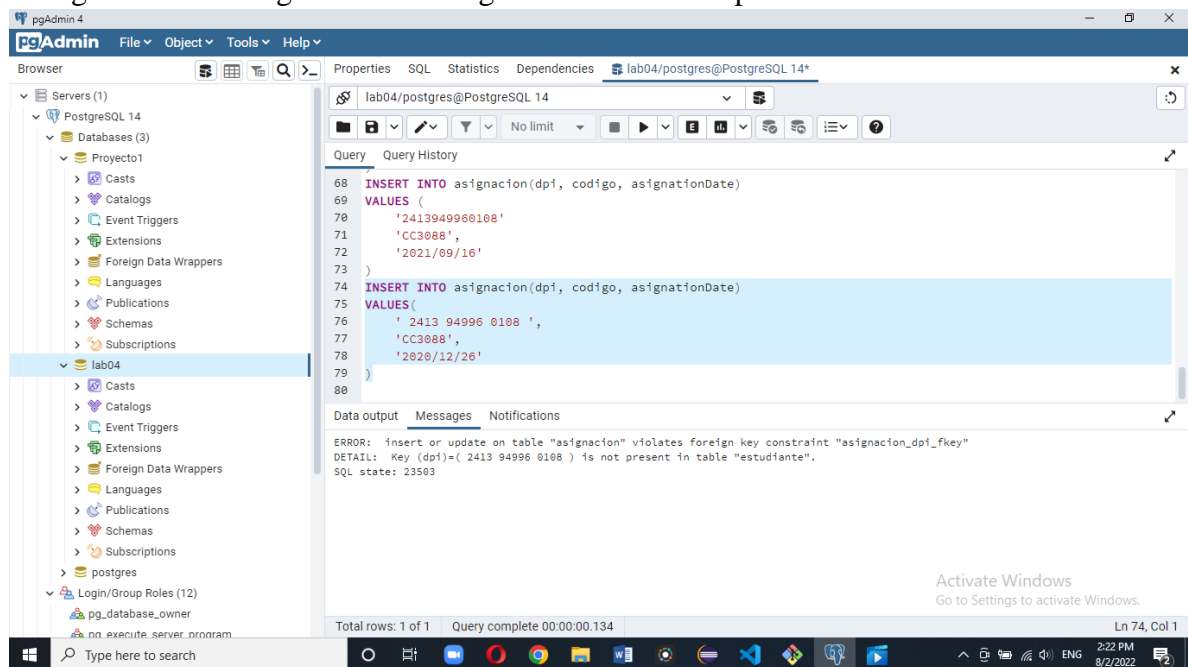
```
Query returned successfully in 150 msec.
```

3. Verificación de Constraints

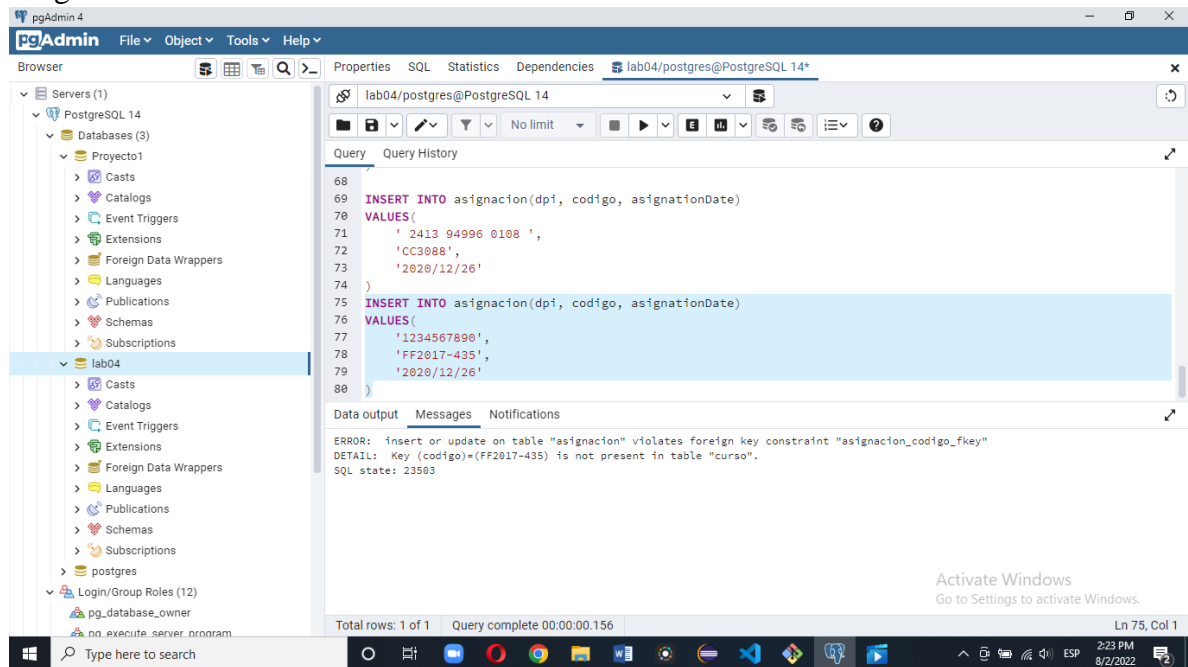
a. Insert de Gabriela y yo en la asignación



b. El ingreso de una asignación de código de estudiante no presente.

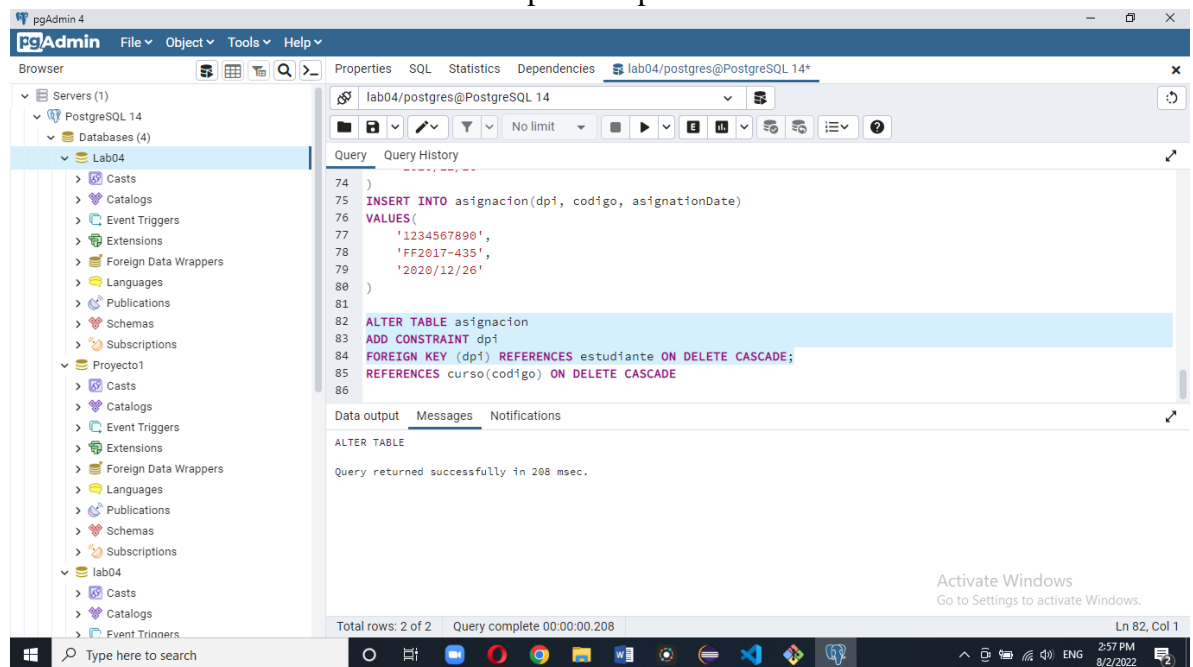


c. Asignación mía a un curso inexistente.

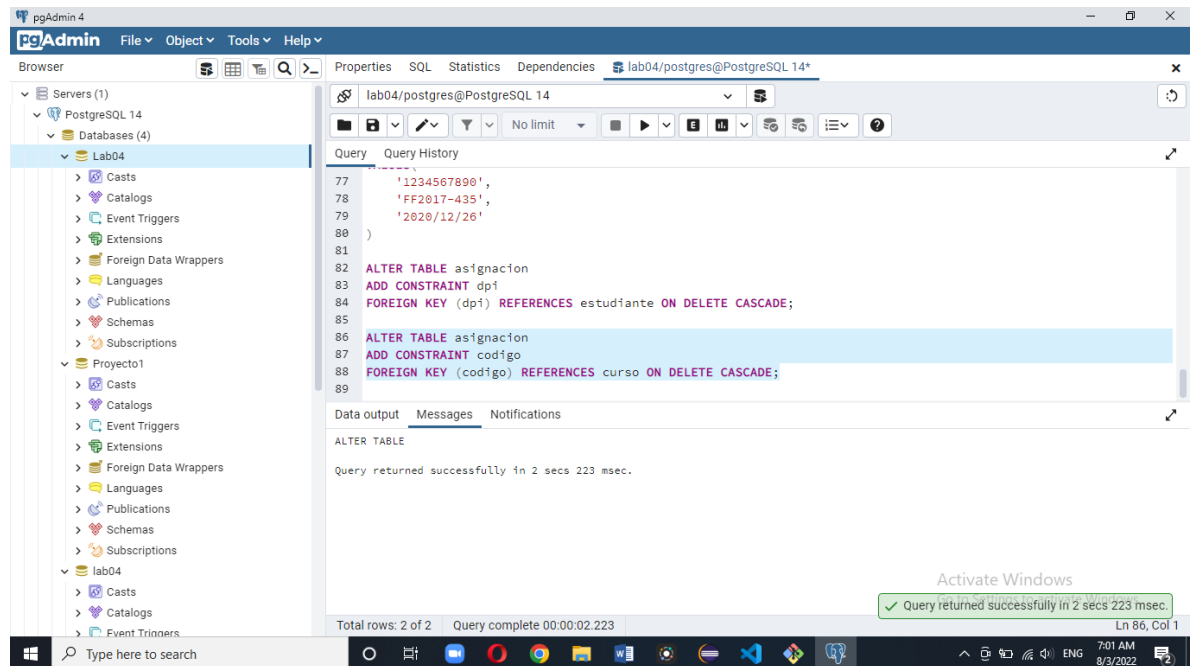


4. Comportamientos derivados

Alteración de la eliminación en cascada para el dpi.

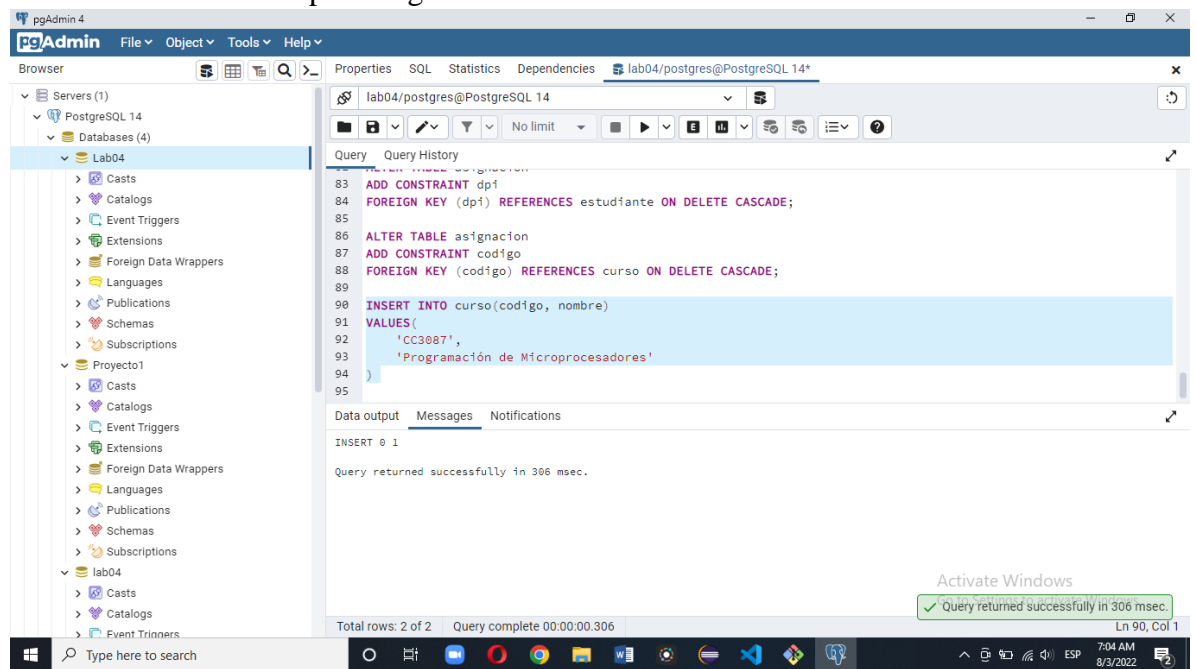


Alteración de la tabla asignación para la eliminación en cascada cuando se elimine el código del curso.

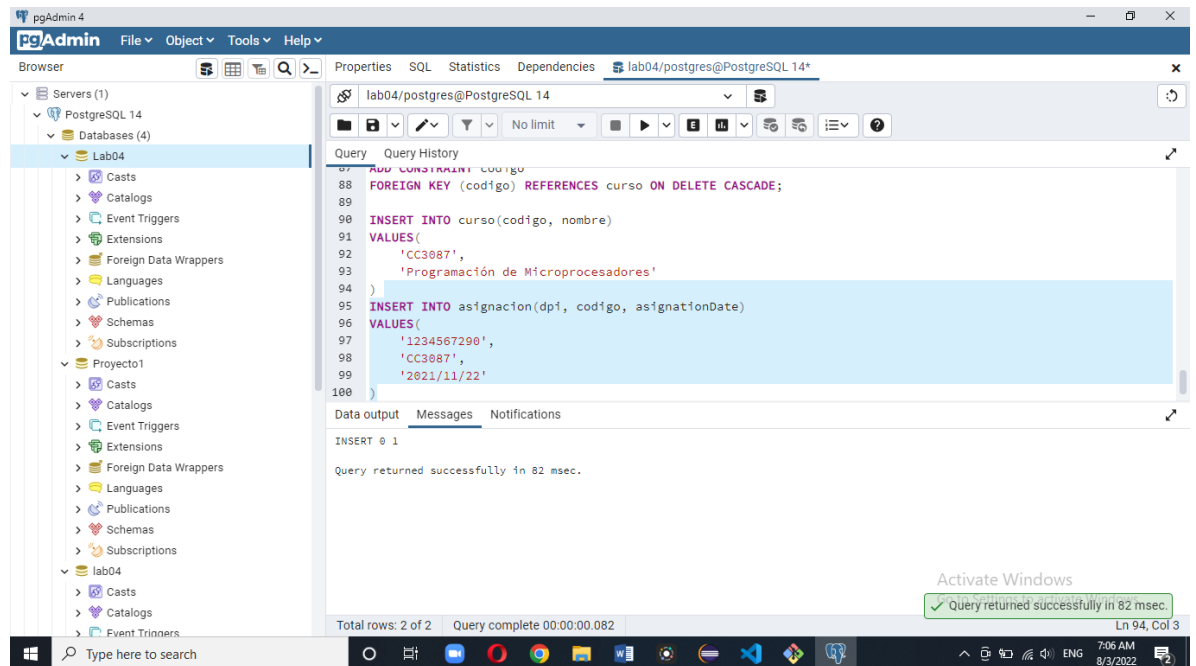


Prueba:

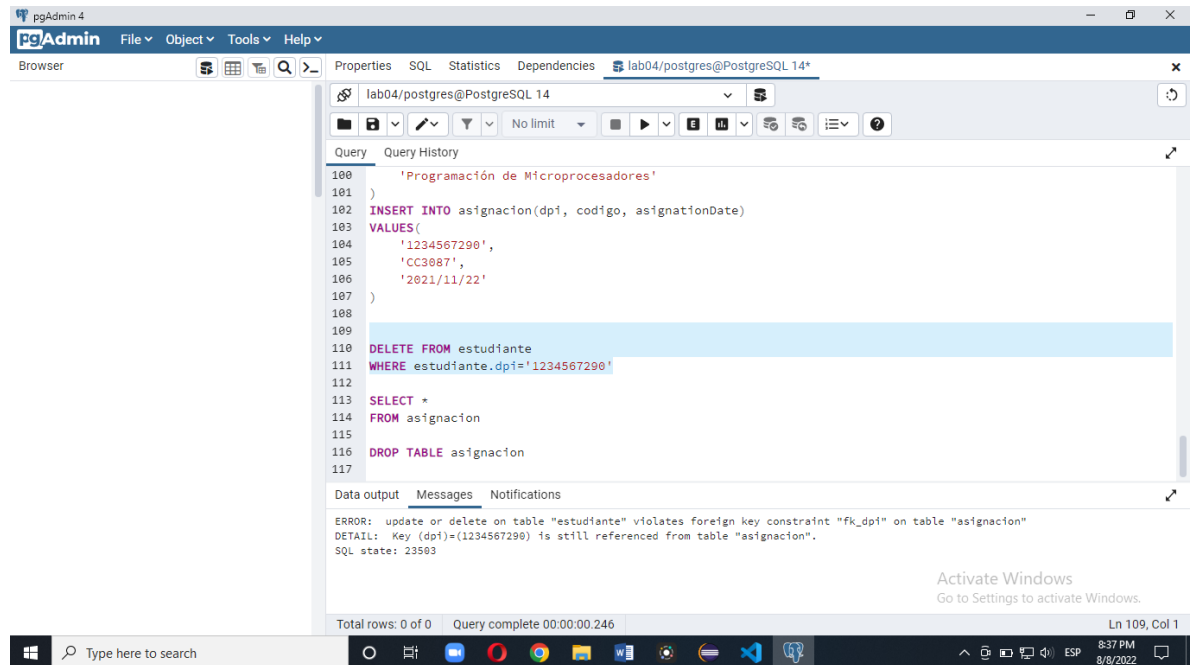
Creación de otro curso para asignarse.



Asignación al curso de Microprocesadores por parte del estudiante Sebastian Najera.



Eliminación en cascada



5. Código de la primera parte

```
CREATE TABLE estudiante(  
  dpi VARCHAR(30) PRIMARY KEY,  
  fechaNacimiento DATE,  
  nombres VARCHAR(50),
```

apellidos VARCHAR(50)

)

CREATE TABLE curso(

codigo VARCHAR(30) PRIMARY KEY,

nombre VARCHAR(50)

)

INSERT INTO estudiante(dpi, fechaNacimiento, nombres, apellidos)

VALUES(

'1234567890',

'2002/01/20',

'Diego Andres',

'Alonzo Medinilla'

),

(

'0987654321',

'2003/04/21',

'Gabriela Elizabeth',

'Marroquin Sigui'

),

(

'1234567290',

'2001/03/22',

'Sebastian Andres',

'Lopez Najera'

),

(

'1232167290',


```
'1999/03/22',
'Raun Argueta',
'Lopez Aristondo'
),
(
'123567290',
'2005/03/22',
'Rodrigo Andres',
'Kefalas Olmeca'
)
INSERT INTO curso(codigo, nombre)
VALUES(
'CC3088',
'Base de datos 1'
)
CREATE TABLE asignacion(
dpi VARCHAR(30) NOT NULL,
codigo VARCHAR(30) NOT NULL,
asignationDate DATE,
CONSTRAINT fk_dpi
FOREIGN KEY (dpi) REFERENCES estudiante,
CONSTRAINT fk_codigo
FOREIGN KEY (codigo) REFERENCES curso,
UNIQUE (dpi, codigo, asignationDate)
)
ALTER TABLE asignacion
DROP CONSTRAINT dca
DROP TABLE asignacion
```

Diego Andrés Alonzo Medinilla 20172

```
INSERT INTO asignacion(dpi, codigo, asignationDate)
```

```
VALUES(
```

```
'1234567890',
```

```
'CC3088',
```

```
'2020/12/26'
```

```
),
```

```
(
```

```
'0987654321',
```

```
'CC3088',
```

```
'2020/12/26'
```

```
)
```

```
INSERT INTO asignacion(dpi, codigo, asignationDate)
```

```
VALUES(
```

```
' 2413 94996 0108 ',
```

```
'CC3088',
```

```
'2020/12/26'
```

```
)
```

```
INSERT INTO asignacion(dpi, codigo, asignationDate)
```

```
VALUES(
```

```
'1234567890',
```

```
'FF2017-435',
```

```
'2020/12/26'
```

```
)
```

```
ALTER TABLE asignacion
```

```
DROP CONSTRAINT dpi
```

Diego Andrés Alonzo Medinilla 20172

ALTER TABLE asignacion

ADD CONSTRAINT dpi

FOREIGN KEY (dpi) REFERENCES estudiante(dpi) ON DELETE CASCADE

ALTER TABLE asignacion

DROP CONSTRAINT codigo

ALTER TABLE asignacion

ADD CONSTRAINT codigo FOREIGN KEY (codigo) REFERENCES curso(codigo) ON DELETE CASCADE;

INSERT INTO curso(codigo, nombre)

VALUES(

'CC3087',

'Programación de Microprocesadores'

)

INSERT INTO asignacion(dpi, codigo, asignationDate)

VALUES(

'1234567290',

'CC3087',

'2021/11/22'

)

DELETE FROM estudiante

WHERE estudiante.dpi='1234567290'

SELECT *

FROM asignacion

DROP TABLE asignación

Consultas avanzada en SQL

1. ¿Qué aerolíneas de las que vuelan desde SFO se retrasan más en llegar?
 - a. Las aerolíneas que realizaron un vuelo al menos desde SFO son: F9, OO, US, AS, CO, UA, B6, MQ, HA, AA, FL, XE, DL, WN y NW. (Esto lo copié directamente del archivo de instrucciones porque para que iba a copiar 1 a 1=.

The screenshot shows the pgAdmin 4 interface with the following components:

- Browser:** A tree view on the left showing database objects like `arrtime`, `createtime`, `reporting_airline`, `flightnum`, `tailnum`, `actualelapsedtime`, `crselapsedtime`, `airtime`, `arrdelay`, `depdelay`, `origin`, `dest`, `distance`, `taxiin`, `taxiout`, `cancelled`, `cancellationcode`, `diverted`, `carrierdelay`, `weatherdelay`, `nasdelay`, `securitydelay`, `lateaircraftdelay`, `Constraints`, `Indexes`, `RLS Policies`, and `Rules`.
- Query Editor:** The main window showing SQL queries. The first query is:

```
1 SELECT arrtime
2 FROM flight
3 WHERE flight.arrtime IS NOT null
4 ORDER BY flight.arrtime DESC
5 LIMIT 15
```

The second query is:

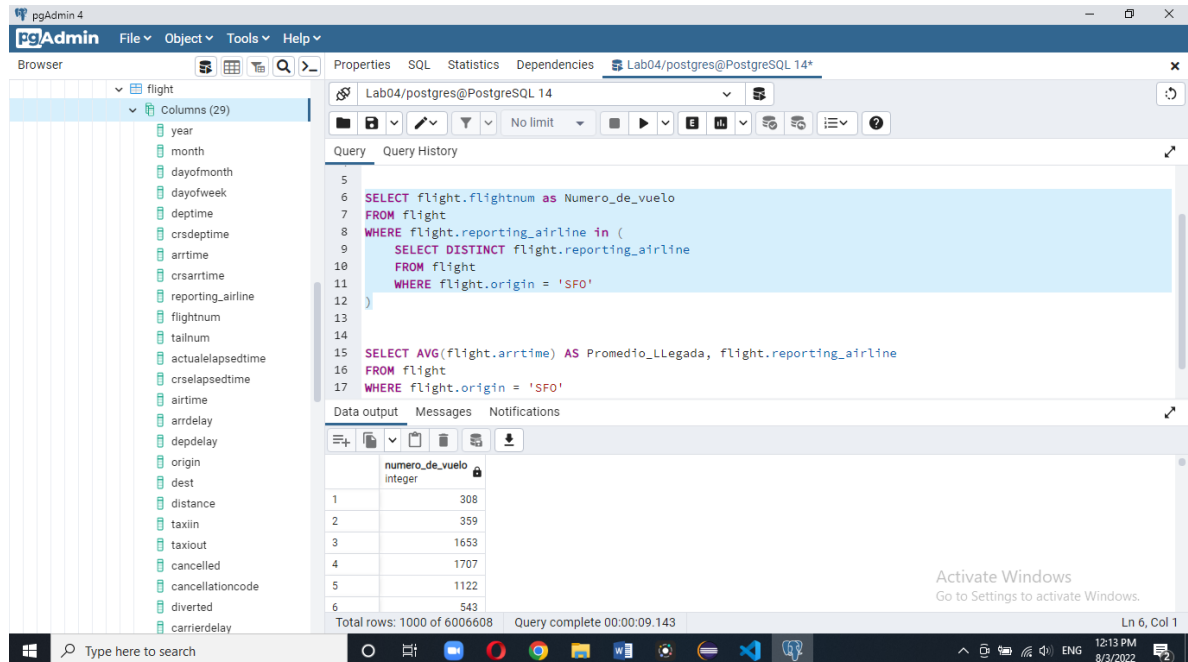
```
7 SELECT DISTINCT flight.reporting_airline
8 FROM flight
9 WHERE flight.origin = 'SFO'
```

The third query is:

```
11 SELECT flight.distance
12 FROM flight
13 ORDER BY flight.distance DESC
```
- Data Output:** The bottom panel showing the results of the second query. It displays a table with the following data:

reporting_airline
NW
CO
FL
OO
AS
XE
- Status Bar:** At the bottom, it shows "Total rows: 15 of 15" and "Query complete 00:00:07.792".

- b. Identificación de aquellos vuelos que pertenecen dentro de las aerolíneas que han volado desde SFO.



The screenshot shows the pgAdmin 4 interface. The 'Query' tab is active, displaying the following SQL code:

```

5
6 SELECT flight.flightnum AS Numero_de_vuelo
7 FROM flight
8 WHERE flight.reporting_airline IN (
9     SELECT DISTINCT flight.reporting_airline
10    FROM flight
11    WHERE flight.origin = 'SFO'
12 )
13
14
15 SELECT AVG(flight.arrtime) AS Promedio_Llegada, flight.reporting_airline
16 FROM flight
17 WHERE flight.origin = 'SFO'

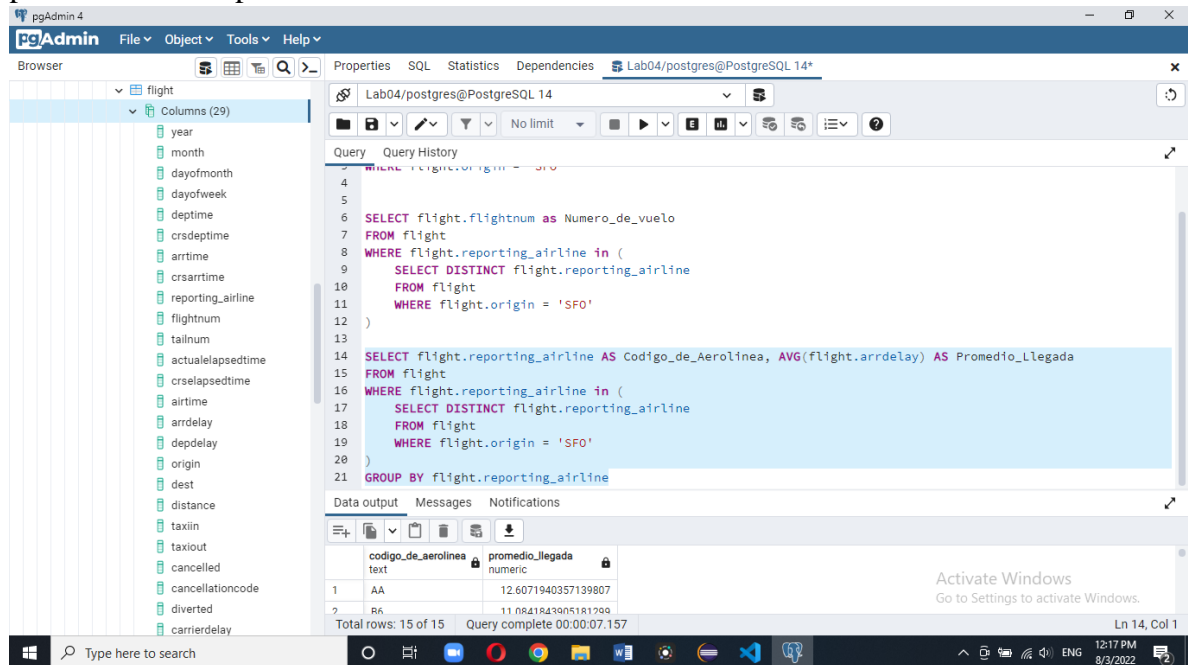
```

The 'Data output' tab shows the results of the query:

numero_de_vuelo	Promedio_Llegada
308	
359	
1653	
1707	
1122	
543	

Total rows: 1000 of 6006608. Query complete 00:00:09.143. Ln 6, Col 1.

- c. Computación del promedio de retraso de todas las aerolíneas. Código y promedio de tiempo.



The screenshot shows the pgAdmin 4 interface. The 'Query' tab is active, displaying the following SQL code:

```

4
5
6 SELECT flight.flightnum AS Numero_de_vuelo
7 FROM flight
8 WHERE flight.reporting_airline IN (
9     SELECT DISTINCT flight.reporting_airline
10    FROM flight
11    WHERE flight.origin = 'SFO'
12 )
13
14 SELECT flight.reporting_airline AS Código_de_Aerolinea, AVG(flight.arrdelay) AS Promedio_Llegada
15 FROM flight
16 WHERE flight.reporting_airline IN (
17     SELECT DISTINCT flight.reporting_airline
18    FROM flight
19    WHERE flight.origin = 'SFO'
20 )
21 GROUP BY flight.reporting_airline

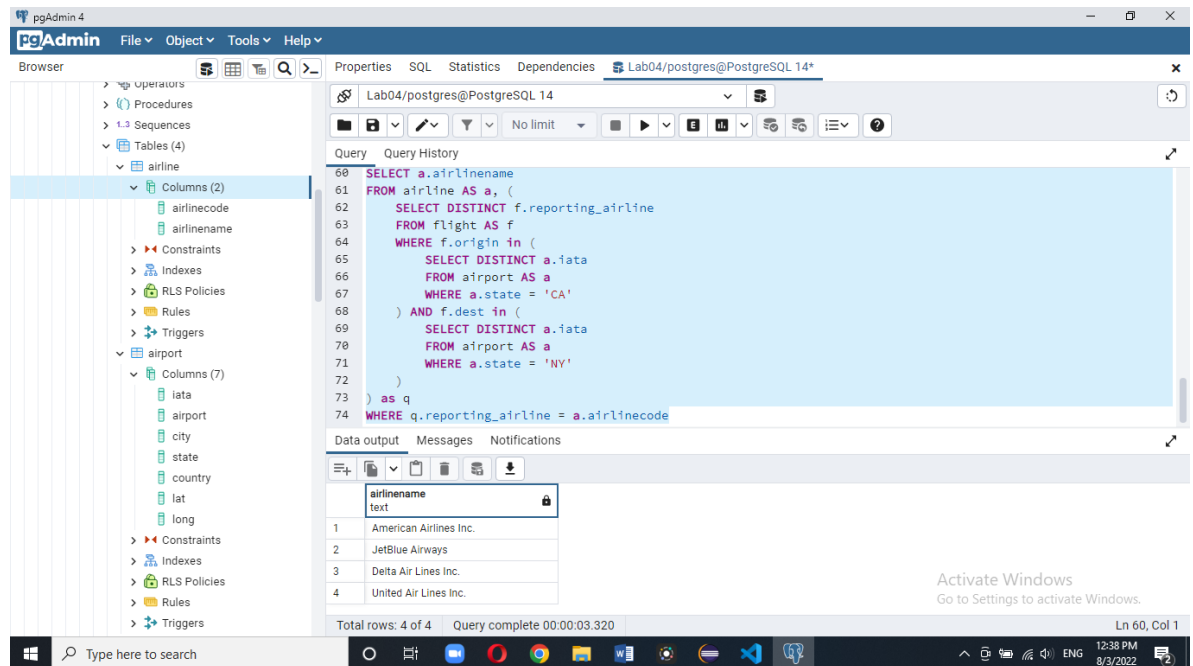
```

The 'Data output' tab shows the results of the query:

código_de_aerolinea	Promedio_Llegada
AA	12.6071940357139807
AA	11.684184905181199

Total rows: 15 of 15. Query complete 00:00:07.157. Ln 14, Col 1.

2. ¿Qué aerolíneas vuelan de California a Nueva York?



Las aerolíneas que vuelan de California a Nueva York son:

- American Airlines Inc.
 - JetBlue Airways
 - Delta Air Lines Inc.
 - United Air Lines Inc.
3. ¿Qué aerolíneas ha realizado vuelos del estado de California al de Nueva York, y también de California a Nuevo México?

Las aerolíneas que van de California a Nuevo México son:

- "SkyWest Airlines Inc."
- "United Air Lines Inc."
- "Southwest Airlines Co."

d. "ExpressJet Airlines Inc. (1)"

The screenshot shows the pgAdmin 4 interface. On the left, the 'airline' table is selected under 'Tables (4)'. The 'state' column is highlighted. The main pane displays a SQL query (lines 52-66) that selects airline names from the 'airline' table, filtered by flights originating from California ('CA') and destined for New Mexico ('NM'). The results pane shows four rows of data:

airlinecode	airlinename
1	SkyWest Airlines Inc.
2	United Air Lines Inc.
3	Southwest Airlines Co.
4	ExpressJet Airlines Inc. (...)

The status bar indicates 'Total rows: 4 of 4' and 'Query complete 00:00:04.233'. A green message box at the bottom right states: 'Successfully run. Total query runtime: 4 secs 233 msec. 4 rows affected.'

Por ende las aerolíneas que van tanto de California a Nueva York como de California a Nuevo México son:

The screenshot shows the pgAdmin 4 interface. On the left, the 'airline' table is selected under 'Tables (4)'. The 'state' column is highlighted. The main pane displays a SQL query (lines 68-82) that selects airline names from the 'airline' table, filtered by flights originating from California ('CA') and destined for either New York ('NY') or New Mexico ('NM'). The results pane shows five rows of data:

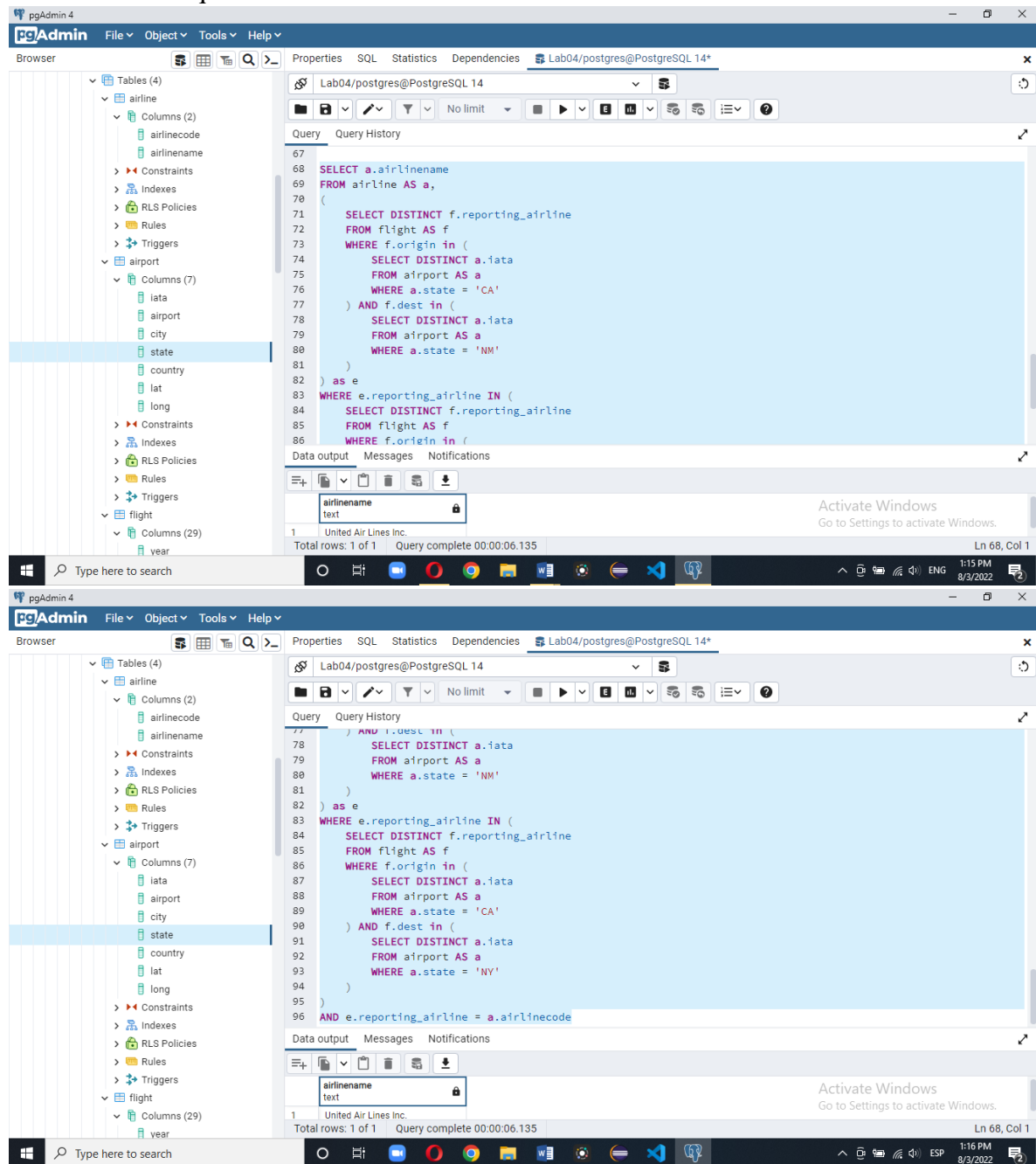
airlinecode	airlinename
1	American Airlines Inc.
2	JetBlue Airways
3	Delta Air Lines Inc.
4	SkyWest Airlines Inc.
5	United Air Lines Inc.

The status bar indicates 'Total rows: 7 of 7' and 'Query complete 00:00:04.028'. A green message box at the bottom right states: 'Successfully run. Total query runtime: 4 secs 28 msec. 7 rows affected.'

- e. "American Airlines Inc."
- f. "JetBlue Airways"
- g. "Delta Air Lines Inc."
- h. "SkyWest Airlines Inc."
- i. "United Air Lines Inc."

- j. "Southwest Airlines Co."
- k. "ExpressJet Airlines Inc. (1)"

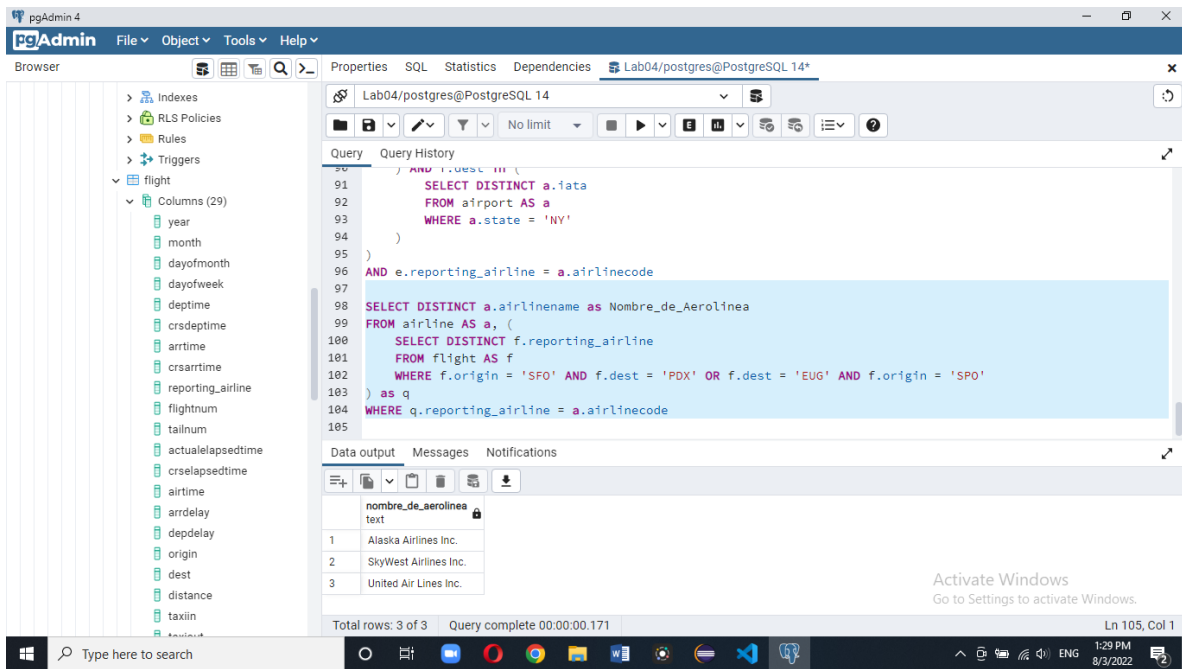
Única aerolínea que realiza ambos vuelos:



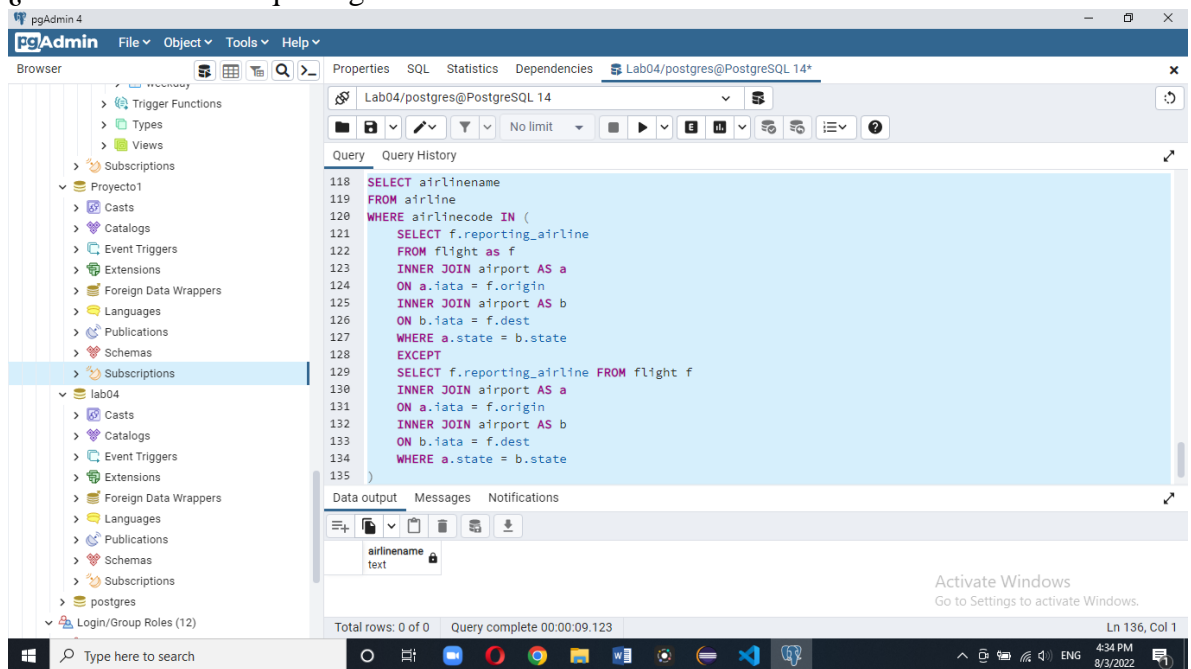
Teniendo en común United Air lines Inc. Como la única aerolínea que realiza ambos vuelos.

- 4. ¡Quiero maximizar mis millas!
- Las aerolíneas que sí realizan vuelos de San Francisco (SFO) a Portland (PDX) y Eugene (EUG) son:
 - a. "Alaska Airlines Inc."

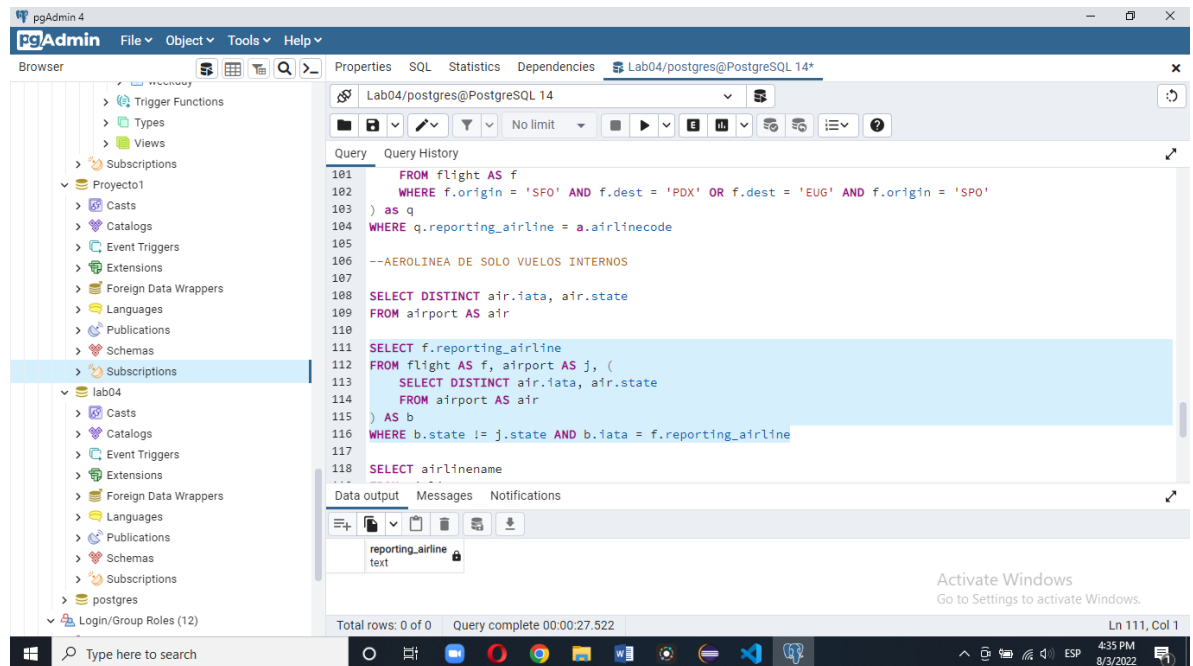
- b. "SkyWest Airlines Inc."
- c. "United Air Lines Inc."



5. ¿Existen aerolíneas que hagan únicamente vuelos internos?

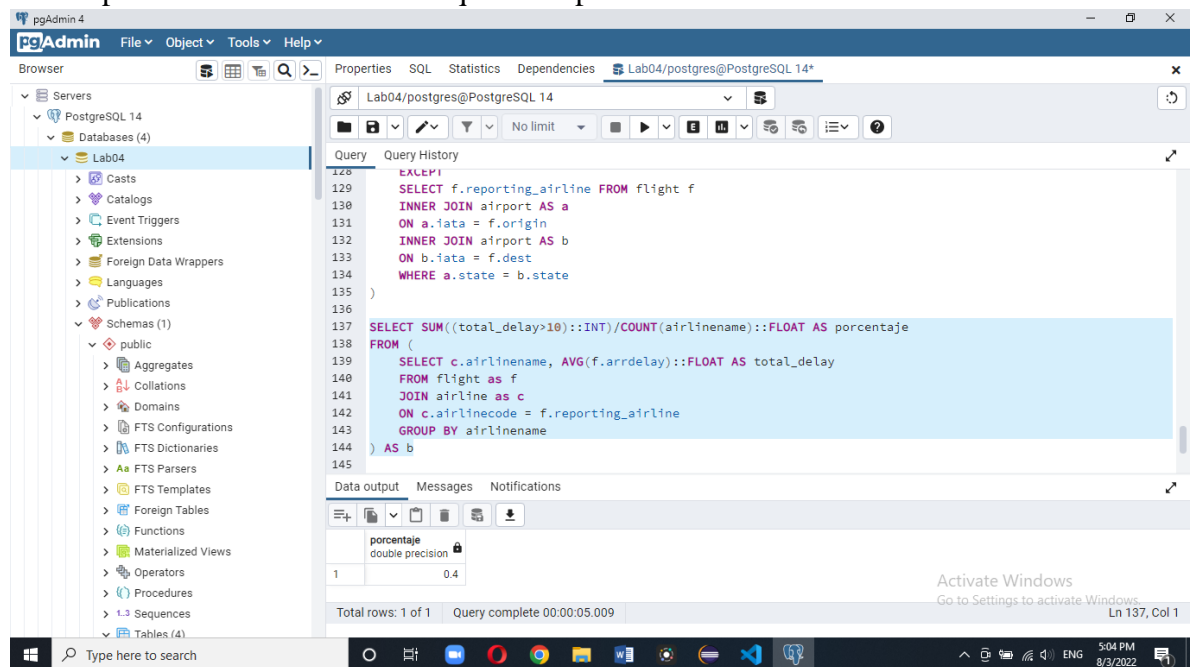


No hay aerolíneas con vuelos internos.



No hay aerolíneas con vuelos internos.

6. ¿Qué proporción de aerolíneas se retrasan en llegar un promedio 10 minutos o más con respecto al total de aerolíneas que han operado vuelos?



7. Código de la segunda parte
- ```

SELECT DISTINCT flight.reporting_airline
FROM flight
WHERE flight.origin = 'SFO'

```

```
SELECT flight.flightnum as Numero_de_vuelo
FROM flight
WHERE flight.reporting_airline in (
 SELECT DISTINCT flight.reporting_airline
 FROM flight
 WHERE flight.origin = 'SFO'
)
```

--Estas 2 queries no son lo mismo aunque lo parezcan

```
SELECT flight.reporting_airline AS Codigo_de_Aerolinea, AVG(flight.arrdelay)
AS Promedio_Llegada
FROM flight
WHERE flight.reporting_airline in (
 SELECT DISTINCT flight.reporting_airline
 FROM flight
 WHERE flight.origin = 'SFO'
)
GROUP BY flight.reporting_airline
```

```
SELECT DISTINCT flight.reporting_airline AS Codigo_de_Aerolinea,
AVG(flight.arrdelay) AS Promedio_Llegada
FROM flight
WHERE flight.origin = 'SFO'
GROUP BY flight.reporting_airline
--Esas dos de arriba
```

```
SELECT DISTINCT a.iata
FROM airport AS a
WHERE a.state = 'CA'
```

```
SELECT DISTINCT a.iata
FROM airport AS a
WHERE a.state = 'NY'
```

```
SELECT DISTINCT f.reporting_airline
FROM flight AS f
WHERE f.origin in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'CA'
) AND f.dest in (
```

```
SELECT DISTINCT a.iata
FROM airport AS a
WHERE a.state = 'NY'
)

SELECT a.airlinename
FROM airline AS a, (
 SELECT DISTINCT f.reporting_airline
 FROM flight AS f
 WHERE f.origin in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'CA'
) AND f.dest in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'NM'
)
) as q
WHERE q.reporting_airline = a.airlinecode

SELECT a.airlinename
FROM airline AS a,
(
 SELECT DISTINCT f.reporting_airline
 FROM flight AS f
 WHERE f.origin in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'CA'
) AND f.dest in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'NM'
)
) as e
WHERE e.reporting_airline IN (
 SELECT DISTINCT f.reporting_airline
 FROM flight AS f
 WHERE f.origin in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'CA'
```

```
) AND f.dest in (
 SELECT DISTINCT a.iata
 FROM airport AS a
 WHERE a.state = 'NY'
)
)
AND e.reporting_airline = a.airlinecode
```

```
SELECT DISTINCT a.airlinename as Nombre_de_Aerolinea
FROM airline AS a, (
 SELECT DISTINCT f.reporting_airline
 FROM flight AS f
 WHERE f.origin = 'SFO' AND f.dest = 'PDX' OR f.dest = 'EUG' AND f.origin =
'SPO'
) as q
WHERE q.reporting_airline = a.airlinecode
```

--AEROLINEA DE SOLO VUELOS INTERNOS

```
SELECT DISTINCT air.iata, air.state
FROM airport AS air
```

```
SELECT f.reporting_airline
FROM flight AS f, airport AS j, (
 SELECT DISTINCT air.iata, air.state
 FROM airport AS air
) AS b
WHERE b.state != j.state AND b.iata = f.reporting_airline
```

```
SELECT airlinename
FROM airline
WHERE airlinecode IN (
 SELECT f.reporting_airline
 FROM flight as f
 INNER JOIN airport AS a
 ON a.iata = f.origin
 INNER JOIN airport AS b
 ON b.iata = f.dest
 WHERE a.state = b.state
 EXCEPT
 SELECT f.reporting_airline FROM flight f
 INNER JOIN airport AS a
 ON a.iata = f.origin
```

```
 INNER JOIN airport AS b
 ON b.iata = f.dest
 WHERE a.state = b.state
)
```

```
SELECT SUM((total_delay>10)::INT)/COUNT(airlinename)::FLOAT AS
porcentaje
FROM (
 SELECT c.airlinename, AVG(f.arrdelay)::FLOAT AS total_delay
 FROM flight as f
 JOIN airline as c
 ON c.airlinecode = f.reporting_airline
 GROUP BY airlinename
) AS b
```

```
SELECT SUM((total_delay>10)::INT)/COUNT(airlinename)::FLOAT AS
porcentaje
FROM(
 SELECT al.airlinename, AVG(fl.arrdelay)::FLOAT AS total_delay
 FROM flight as fl
 JOIN airline as al
 ON al.airlinecode = fl.reporting_airline
 GROUP BY airlinename
) AS alias_1
```