Laboratorio 4 Bases de datos
Roberto Rios, 20979
Mario De Leon, 19019

Ejercicio 1

1.1

lab04/postgres@PostgreSQL 14

Query Editor | Query History

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

Data Output | Explain | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 95 msec.
```

```
1  ALTER TABLE curso ADD PRIMARY KEY (codigo);
```

Data Output | Explain | Messages | Notifications

```
ALTER TABLE

Query returned successfully in 45 msec.
```

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  INSERT INTO estudiante VALUES ('3112121434543', '02-02-2022', 'Roberto Francisco', 'Rios Morales'),
2      ('2637584921234', '02-02-2022', 'Mario David', 'De Leon Muralles');
```

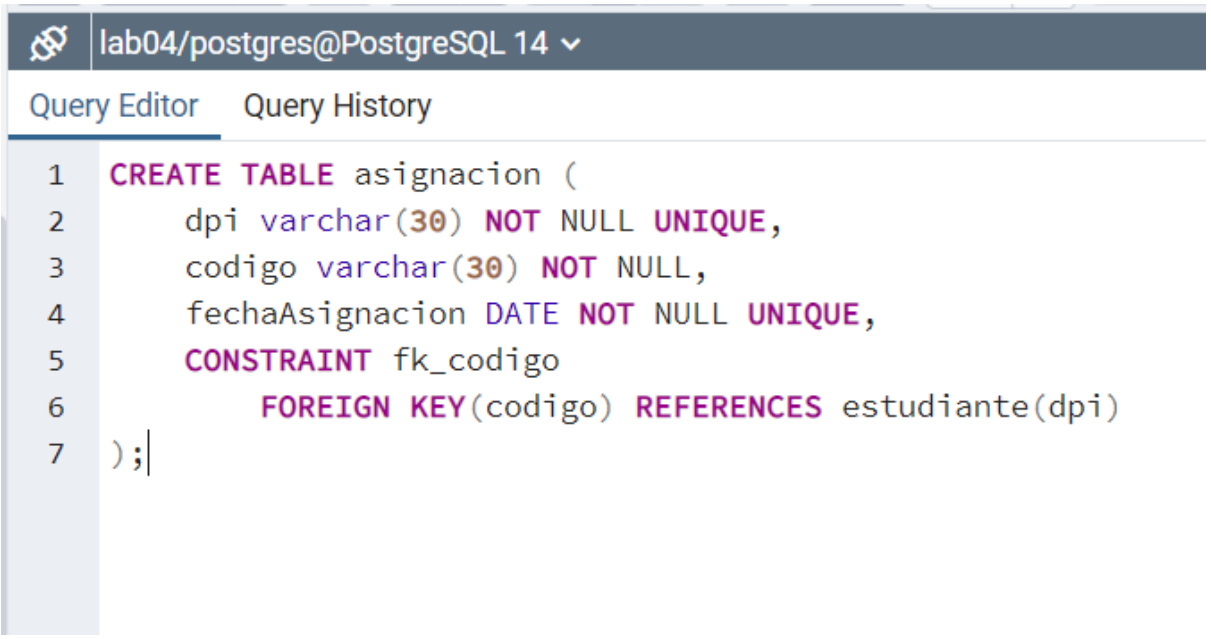
Data Output | Explain | Messages | Notifications

```
INSERT 0 2

Query returned successfully in 55 msec.
```


lab04/postgres@PostgreSQL 14
Query Editor
Query History
1
INSERT INTO curso VALUES ('CC3057', 'Bases de datos 1');
Data Output
Explain
Messages
Notifications
INSERT 0 1
Query returned successfully in 39 msec.

1.2


lab04/postgres@PostgreSQL 14
Query Editor
Query History
1
CREATE TABLE asignacion (
2
dpi varchar(30) NOT NULL UNIQUE,
3
codigo varchar(30) NOT NULL,
4
fechaAsignacion DATE NOT NULL UNIQUE,
5
CONSTRAINT fk_codigo
6
FOREIGN KEY(codigo) REFERENCES estudiante(dpi)
7
);

1.3

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```sql
CREATE TABLE asignacion (
    dpi varchar(30) NOT NULL,
    codigo varchar(30) NOT NULL,
    fechaAsignacion DATE NOT NULL,
    CONSTRAINT fk_dpi
        FOREIGN KEY(dpi) REFERENCES estudiante(dpi),
    CONSTRAINT fk_codigo
        FOREIGN KEY(codigo) REFERENCES curso(codigo),
    UNIQUE (dpi, codigo, fechaAsignacion)
);
```

Data Output | Explain | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 48 msec.
```

1.3.1

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```sql
INSERT INTO asignacion VALUES ('2637584921234', 'CC3057', '2022-02-02'),
    ('3112121434543', 'CC3057', '2022-02-02');
```

1.3.2.

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```sql
INSERT INTO asignacion VALUES ('2413949960108', 'CC3057', '2022-02-02');
```

Data Output | Explain | Messages | Notifications

```
ERROR:  inserción o actualización en la tabla «asignacion» viola la llave foránea «fk_dpi»
DETAIL:  La llave (dpi)=(2413949960108) no está presente en la tabla «estudiante».
SQL state: 23503
```

1.3.3.

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  INSERT INTO asignacion VALUES ('3112121434543', 'FF2017-435', '2022-02-02');
```

Data Output | Explain | Messages | Notifications

```
ERROR:  inserción o actualización en la tabla «asignacion» viola la llave foránea «fk_codigo»
DETAIL:  La llave (codigo)=(FF2017-435) no está presente en la tabla «curso».
SQL state: 23503
```

1.4

"CASCADE" borra los registros de la tabla independiente cuando se borra el registro de la tabla principal (usando DELETE); o CASCADE también puede actualizar el valor de una clave secundaria cuando se actualiza el valor de la clave referenciada (usando UPDATE).

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  ALTER TABLE asignacion
2  ADD CONSTRAINT fk_dpi
3      FOREIGN KEY(dpi) REFERENCES estudiante(dpi) ON DELETE CASCADE
```

Primero mostramos a los estudiantes en 'estudiante':

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  SELECT * FROM public.estudiante
2  ORDER BY dpi ASC
```

Data Output | Explain | Messages | Notifications

	dpi [PK] character varying (30)	fechanacimiento date	nombres character varying (30)	apellidos character varying (50)
1	2637584921234	2022-02-02	Mario David	De Leon Muralles
2	3112121434543	2022-02-02	Roberto Francisco	Rios Morales
3	4837483738383	2021-12-22	Ye	East
4	4837483738388	2021-12-22	Yeeeeee	West

Y luego mostramos a los dpi's de los estudiantes en 'asignacion':

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  SELECT * FROM public.asignacion
2  LIMIT 100
3
```

Data Output | Explain | Messages | Notifications

	dpi character varying (30)	codigo character varying (30)	fechaasignacion date
1	2637584921234	CC3057	2022-02-02
2	3112121434543	CC3057	2022-02-02
3	4837483738388	CC3000	2022-02-02
4	4837483738388	CC3057	2022-02-02

Ahora borramos al estudiante 'Roberto Rios':

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  DELETE FROM estudiante WHERE dpi='3112121434543'
2
```

Data Output | Explain | Messages | Notifications

```
DELETE 1

Query returned successfully in 54 msec.
```

Y podemos ver que desapareció de la tabla ‘estudiante’ y de ‘asignacion’:

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  SELECT * FROM public.estudiante
2  ORDER BY dpi ASC
```

Data Output | Explain | Messages | Notifications

	dpi [PK] character varying (30)	fechanacimiento date	nombres character varying (30)	apellidos character varying (50)
1	2637584921234	2022-02-02	Mario David	De Leon Muralles
2	4837483738383	2021-12-22	Ye	East
3	4837483738388	2021-12-22	Yeeeeee	West

lab04/postgres@PostgreSQL 14

Query Editor | Query History

```
1  SELECT * FROM public.asignacion
2
```

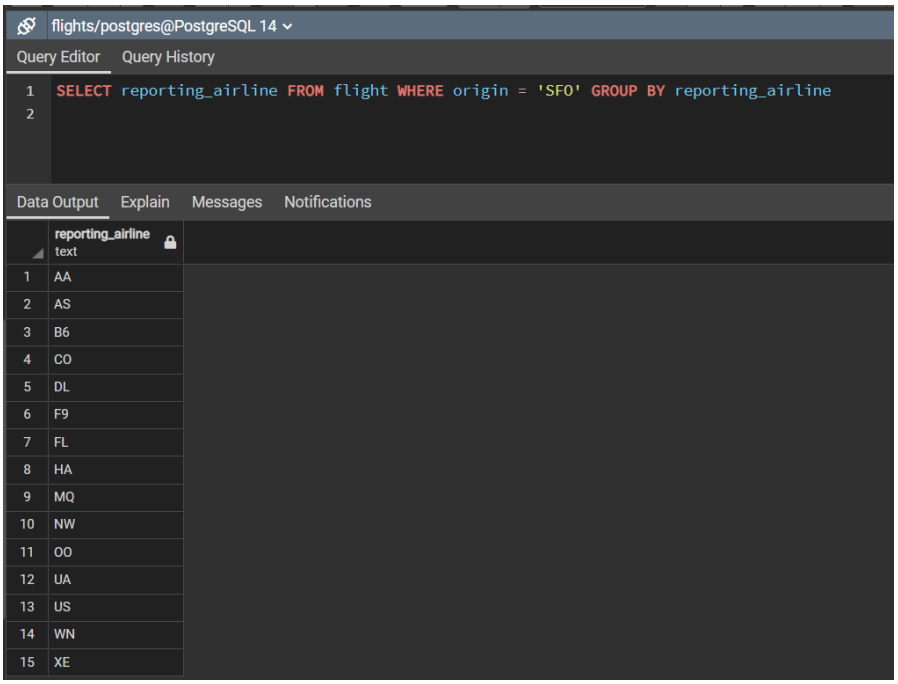
Data Output | Explain | Messages | Notifications

	dpi character varying (30)	codigo character varying (30)	fechaasignacion date
1	2637584921234	CC3057	2022-02-02
2	4837483738388	CC3000	2022-02-02
3	4837483738388	CC3057	2022-02-02

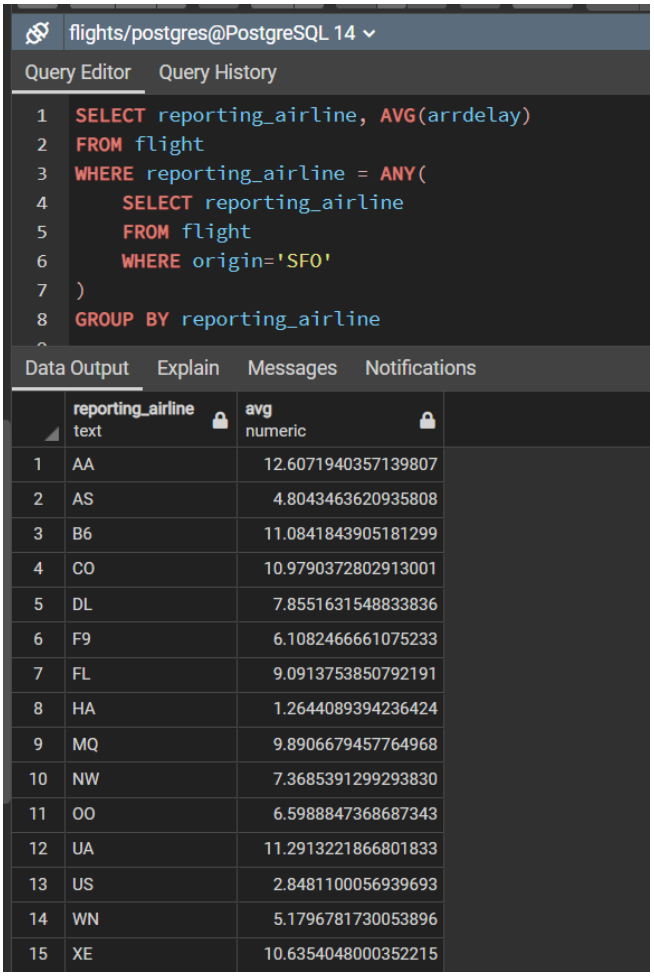
2

2.1

2.1.1


flights/postgres@PostgreSQL 14
Query Editor
Query History
1 SELECT reporting_airline FROM flight WHERE origin = 'SFO' GROUP BY reporting_airline
2
Data Output
Explain
Messages
Notifications
reporting_airline
text
1 AA
2 AS
3 B6
4 CO
5 DL
6 F9
7 FL
8 HA
9 MQ
10 NW
11 OO
12 UA
13 US
14 WN
15 XE

2.1.2


flights/postgres@PostgreSQL 14
Query Editor
Query History
1 SELECT reporting_airline, AVG(arrdelay)
2 FROM flight
3 WHERE reporting_airline = ANY(
4 SELECT reporting_airline
5 FROM flight
6 WHERE origin='SFO'
7)
8 GROUP BY reporting_airline
Data Output
Explain
Messages
Notifications
reporting_airline
text
avg
numeric
1 AA 12.6071940357139807
2 AS 4.8043463620935808
3 B6 11.0841843905181299
4 CO 10.9790372802913001
5 DL 7.8551631548833836
6 F9 6.1082466661075233
7 FL 9.0913753850792191
8 HA 1.2644089394236424
9 MQ 9.8906679457764968
10 NW 7.3685391299293830
11 OO 6.5988847368687343
12 UA 11.2913221866801833
13 US 2.8481100056939693
14 WN 5.1796781730053896
15 XE 10.6354048000352215

2.2

Query Editor | Query History

```sql
SELECT * FROM airline
--se van a unir tablas empezando por el codigo--
    WHERE airlinecode IN(
        SELECT fl.reporting_airline FROM flight fl
        --columna iata de tabla airport--
            INNER JOIN airport AS a1
                ON a1.iata = fl.origin
            INNER JOIN airport AS a2
                ON a2.iata = fl.dest
            WHERE a1.state = 'CA' AND a2.state = 'NY'
    )

```

Data Output | Explain | Messages | Notifications

	airlinecode text	airlinename text
1	AA	American Airlines Inc.
2	B6	JetBlue Airways
3	DL	Delta Air Lines Inc.
4	UA	United Air Lines Inc.

2.3

Query Editor | Query History

```
1  SELECT * FROM airline
2  --se van a unir tablas empezando por el codigo--
3      WHERE airlinecode IN(
4          SELECT fl.reporting_airline FROM flight fl
5          --columna iata de tabla airport--
6              INNER JOIN airport AS a1
7                  ON a1.iata = fl.origin
8              INNER JOIN airport AS a2
9                  ON a2.iata = fl.dest
10             WHERE a1.state = 'CA' AND a2.state = 'NY'
11     )AND airlinecode IN(
12         SELECT fl.reporting_airline FROM flight fl
13         --columna iata de tabla airport--
14             INNER JOIN airport AS a1
15                 ON a1.iata = fl.origin
16             INNER JOIN airport AS a2
17                 ON a2.iata = fl.dest
18             WHERE a1.state = 'CA' AND a2.state = 'NM'
19     )
20
```

Data Output | Explain | Messages | Notifications

	airlinecode text	airlinename text
1	UA	United Air Lines Inc.

2.4 Esta vez ya no son estados, sino ciudades

Query Editor | Query History

```sql
SELECT * FROM airline
--se van a unir tablas empezando por el codigo--
    WHERE airlinecode IN(
        SELECT fl.reporting_airline FROM flight fl
        --columna iata de tabla airport--
            INNER JOIN airport AS a1
                ON a1.iata = fl.origin
            INNER JOIN airport AS a2
                ON a2.iata = fl.dest
            WHERE a1.iata = 'SFO' AND a2.iata = 'PDX'
    )AND airlinecode IN(
        SELECT fl.reporting_airline FROM flight fl
        --columna iata de tabla airport--
            INNER JOIN airport AS a1
                ON a1.iata = fl.origin
            INNER JOIN airport AS a2
                ON a2.iata = fl.dest
            WHERE a1.iata = 'SFO' AND a2.iata = 'EUG'
    )
```

Data Output | Explain | Messages | Notifications

	airlinecode text	airlinename text
1	OO	SkyWest Airlines Inc.
2	UA	United Air Lines Inc.

2.5

Query Editor | Query History

```sql
SELECT * FROM airline
--se van a unir tablas empezando por el codigo--
    WHERE airlinecode IN(
        SELECT fl.reporting_airline FROM flight fl
        --columna iata de tabla airport--
            INNER JOIN airport AS a1
                ON a1.iata = fl.origin
            INNER JOIN airport AS a2
                ON a2.iata = fl.dest
        --el estado va a ser el mismo--
            WHERE a1.state = a2.state

        EXCEPT

        SELECT fl.reporting_airline FROM flight fl
        --columna iata de tabla airport--
            INNER JOIN airport AS a1
                ON a1.iata = fl.origin
            INNER JOIN airport AS a2
                ON a2.iata = fl.dest
        --excluir aquellos donde el estado no es el mismo--
            WHERE a1.state != a2.state
    )

```

Data Output | Explain | Messages | Notifications

airlinecode text	airlinename text

2.6

QUERY del LAB 3:

Query Editor | Query History

```
1  SELECT al.airlinename, AVG(fl.arrdelay) FROM flight fl
2   JOIN airline al
3      ON al.airlinecode = fl.reporting_airline
4      GROUP BY al.airlinename
5          HAVING AVG(fl.arrdelay)>10
6          ORDER BY AVG(fl.arrdelay)
7          DESC;
```

Data Output | Explain | Messages | Notifications

	airlinename text	avg numeric
1	American Airlines Inc.	12.6071940357139807
2	PSA Airlines Inc.	11.8174676839980073
3	Mesa Airlines Inc.	11.7751814336008094
4	United Air Lines Inc.	11.2913221866801833
5	JetBlue Airways	11.0841843905181299
6	Continental Air Lines Inc.	10.9790372802913001
7	ExpressJet Airlines Inc. (1)	10.6354048000352215
8	ExpressJet Airlines Inc.	10.2080024157137816

QUERY del LAB 4:

Query Editor | Query History

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

Data Output | Explain | Messages | Notifications

	porcentaje double precision
1	0.4