



**UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO**

---



**FACULTAD DE INGENIERÍA**

**ALUMNO**  
**SILVERIO MARTÍNEZ ANDRÉS**

**MATERIA**  
**BASES DE DATOS**

**PROFESOR**  
**FERNANDO ARREOLA FRANCO**

**GRUPO**  
**1**

**EJERCICIO 3\_7**

**SEMESTRE 2024 - 2**

**UNAM**

## Ejercicio 3\_7

1) Separar el nombre del alumno y nombre del asesor (si aplica)

```
gpolt_2024_2_319254725=> SELECT alumno.nompila, alumno.appat, asesor.nompila, asesor.appat FROM alumno LEFT JOIN asesor ON alumno.ida  
sesor = asesor.idasesor;  
nompila | appat | nompila | appat  
-----  
Isaac | Lemus | Jorge | Campos  
David | Rivera | Jorge | Campos  
Mauricio | Barrientos | Jorge | Campos  
Mario | Tabura | Laura | Sandoval  
Luz | Rueda | Jorge | Campos  
Jorge | Santillan | |  
Pablo | Gonzales | Laura | Sandoval  
Gabriela | Suarez | |  
Gabriela | Gaytan | |  
Dayana | Plata | Fernando | Arreola  
(10 rows)
```

2) Alumno U Asesor (sólo en el atributo nombre)

```
gpolt_2024_2_319254725=> SELECT nompila FROM alumno UNION SELECT nompila FROM asesor;  
nompila  
-----  
Jorge  
Gabriela  
Mario  
Adolfo  
Laura  
Fernando  
Luz  
Mauricio  
(8 rows)
```

Donde, gracias a la unión, todos los nombres, tanto de asesores como alumnos se mostrarán en una misma tabla.

3) Alumno INTERSECT Asesor (sólo en el atributo nombre)

```
gpolt_2024_2_319254725=> SELECT nompila FROM alumno INTERSECT SELECT nompila FROM asesor;  
nompila  
-----  
Jorge  
(1 row)
```

Donde, de acuerdo al “intersect”, solo se mostrarán los valores que coincidan en ambas tablas, que, en este caso, hay dos personas diferentes llamadas Jorge en la tabla alumno y asesor.

#### 4) Alumno - Asesor y Asesor – Alumno

```
gpolt_2024_2_319254725=> SELECT nompila FROM alumno EXCEPT SELECT nompila FROM asesor;
nompila
-----
Luz
Gabriela
Mauricio
Mario
(4 rows)
```

```
postgres=# SELECT nompila FROM asesor EXCEPT SELECT nompila FROM alumno;
nompila
-----
Laura
Fernando
(2 rows)
```

Donde, para cada uno de ambos casos, se mostrarán los valores de la primera tabla introducida que no se encuentran en la segunda tabla, es por eso que Jorge no aparece, ya que se encuentra en ambas tablas.

#### 5) Alumno natural join Asesor y Alumno cross join asesor

```
gpolt_2024_2_319254725=> SELECT * FROM alumno NATURAL JOIN asesor;
nompila | appat | idasesor | idalumno
-----+-----+-----+-----
(0 rows)
```

gpolt\_2024\_2\_319254725=> SELECT \* FROM alumno CROSS JOIN asesor;

idalumno	nompila	appat	idasesor	idasesor	nompila	appat
1	Mauricio	Barrientos	as-1	as-1	Jorge	Campos
2	Mario	Tabura	as-2	as-1	Jorge	Campos
3	Luz	Rueda	as-1	as-1	Jorge	Campos
4	Jorge	Santillan		as-1	Jorge	Campos
5	Gabriela	Gaytan	as-3	as-1	Jorge	Campos
1	Mauricio	Barrientos	as-1	as-2	Laura	Sandoval
2	Mario	Tabura	as-2	as-2	Laura	Sandoval
3	Luz	Rueda	as-1	as-2	Laura	Sandoval
4	Jorge	Santillan		as-2	Laura	Sandoval
5	Gabriela	Gaytan	as-3	as-2	Laura	Sandoval
1	Mauricio	Barrientos	as-1	as-3	Adolfo	Millan
2	Mario	Tabura	as-2	as-3	Adolfo	Millan
3	Luz	Rueda	as-1	as-3	Adolfo	Millan
4	Jorge	Santillan		as-3	Adolfo	Millan
5	Gabriela	Gaytan	as-3	as-3	Adolfo	Millan
1	Mauricio	Barrientos	as-1	as-4	Fernando	Arreola
2	Mario	Tabura	as-2	as-4	Fernando	Arreola
3	Luz	Rueda	as-1	as-4	Fernando	Arreola
4	Jorge	Santillan		as-4	Fernando	Arreola
5	Gabriela	Gaytan	as-3	as-4	Fernando	Arreola

(20 rows)

En el caso del natural join, no se mostrará ningún resultado, ya que, como ninguno de los valores que toma el manejador tiene alguna relación, no se mostrará nada (a diferencia que si se hace un join tradicional y se especifica que columnas estamos uniendo)

Y para el caso del cross join, combinará cada fila de la primera tabla con cada fila de la segunda tabla, haciendo que no sea necesario el tener algún valor en común entre ambas tablas.

- 6) Modificar la tabla Alumno, agregue los siguientes atributos: carrera varchar(40) y edad smallint

```

gpoit_2024_2_319254725=> ALTER TABLE alumno ADD carrera varchar(40), ADD edad smallint;
ALTER TABLE
gpoit_2024_2_319254725=> \d alumno;
Table "public.alumno"
  Column |          Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
idalumno | integer                |           | not null |
nompila  | character varying(50)  |           |         |
appat    | character varying(60)  |           |         |
idasesor | character varying(6)   |           |         |
carrera  | character varying(40)  |           |         |
edad     | smallint               |           |         |
Indexes:
  "idalumno_pk" PRIMARY KEY, btree (idalumno)
Foreign-key constraints:
  "idasesor_fk" FOREIGN KEY (idasesor) REFERENCES asesor(idasesor) ON UPDATE CASCADE ON DELETE SET NULL

```

7) Insertar los siguientes 5 registros:

id\_Alumno,nombre,ap\_paterno, carrera,edad,id\_asesor

6,Isaac,Lemus,Petrolera,30,as-1

7,Gabriela,Suarezs,Industrial,24,as-3

8,Pablo,Gonzalez,Computacion,23,as-2

9,David,Rivera,Industrial,25,as-1

10,Dayana,Plata,Computacion,24,as-4

```

gpoit_2024_2_319254725=> INSERT INTO alumno (idAlumno, nompila, appat, carrera, edad, idAsesor)
gpoit_2024_2_319254725-> VALUES
gpoit_2024_2_319254725-> ('6', 'Isaac', 'Lemus', 'Petrolera', '30', 'as-1'),
gpoit_2024_2_319254725-> ('7', 'Gabriela', 'Suarez', 'Industrial', '24', 'as-3'),
gpoit_2024_2_319254725-> ('8', 'Pablo', 'Gonzales', 'Computación', '23', 'as-2'),
gpoit_2024_2_319254725-> ('9', 'David', 'Rivera', 'Industrial', '25', 'as-1'),
gpoit_2024_2_319254725-> ('10', 'Dayana', 'Plata', 'Computacion', '24', 'as-4');
INSERT 0 5
gpoit_2024_2_319254725=> SELECT * FROM alumno;
 idalumno | nompila | appat | idasesor |  carrera  | edad
-----|-----|-----|-----|-----|-----
        1 | Mauricio | Barrientos | as-1 |          |
        2 | Mario   | Tabura    | as-2 |          |
        3 | Luz     | Rueda     | as-1 |          |
        4 | Jorge   | Santillan |      |          |
        5 | Gabriela | Gaytan    | as-3 |          |
        6 | Isaac   | Lemus     | as-1 | Petrolera |    30
        7 | Gabriela | Suarez    | as-3 | Industrial |    24
        8 | Pablo   | Gonzales  | as-2 | Computación |    23
        9 | David   | Rivera    | as-1 | Industrial |    25
       10 | Dayana  | Plata     | as-4 | Computacion |    24
(10 rows)

```

8) Actualizar los 5 registros iniciales para asignar valores en los atributos agregados

1,Petrolera,27

2, Telecomunicaciones, 24

3, Computacion, 27

4, Industrial, 25

5, Computacion, 19

```
gpo1t_2024_2_319254725=> UPDATE alumno SET carrera = 'Petrolera', edad = '27' WHERE idalumno = 1;
UPDATE 1
gpo1t_2024_2_319254725=> UPDATE alumno SET carrera = 'Telecomunicaciones', edad = '24' WHERE idalumno = 2;
UPDATE 1
gpo1t_2024_2_319254725=> UPDATE alumno SET carrera = 'Computacion', edad = '27' WHERE idalumno = 3;
UPDATE 1
gpo1t_2024_2_319254725=> UPDATE alumno SET carrera = 'Industrial', edad = '25' WHERE idalumno = 4;
UPDATE 1
gpo1t_2024_2_319254725=> UPDATE alumno SET carrera = 'Computacion', edad = '19' WHERE idalumno = 5;
UPDATE 1
gpo1t_2024_2_319254725=> SELECT * FROM alumno;
idalumno | nompila | appat | idasesor | carrera | edad
-----+-----+-----+-----+-----+-----
6 | Isaac | Lemus | as-1 | Petrolera | 30
7 | Gabriela | Suarez | as-3 | Industrial | 24
8 | Pablo | Gonzales | as-2 | Computación | 23
9 | David | Rivera | as-1 | Industrial | 25
10 | Dayana | Plata | as-4 | Computacion | 24
1 | Mauricio | Barrientos | as-1 | Petrolera | 27
2 | Mario | Tabura | as-2 | Telecomunicaciones | 24
3 | Luz | Rueda | as-1 | Computacion | 27
4 | Jorge | Santillan | | Industrial | 25
5 | Gabriela | Gaytan | as-3 | Computacion | 19
(10 rows)
```

9) Validar que los datos coincidan con los esperados

Ya se está verificando al mostrar los datos de salida de cada una de las actualizaciones solicitadas

10) Nombre completo del alumno de mayor edad

```
gpo1t_2024_2_319254725=> SELECT nompila, appat, edad FROM alumno WHERE edad = (SELECT MAX(edad) FROM alumno);
nompila | appat | edad
-----+-----+-----
Isaac | Lemus | 30
(1 row)
```

Para ello, se realizó una subconsulta en donde regresa la edad máxima de entre todos los alumnos, siendo Isaac el más grande de todos con 30 años

11) Nombre completo del alumno de menor edad

```
gpo1t_2024_2_319254725=> SELECT nompila, appat, edad FROM alumno WHERE edad = (SELECT MIN(edad) FROM alumno);
nompila | appat | edad
-----+-----+-----
Gabriela | Gaytan | 19
(1 row)
```

Similar al punto anterior, se realiza la misma subconsulta, pero por medio de una funcion de agregación “MIN”, se obtiene el alumno que es mas joven de entre todos, siendo en este caso Gabriela

12)cantidad de alumnos por carrera

```
gpolt_2024_2_319254725=> SELECT COUNT(carrera), carrera FROM alumno GROUP BY carrera;
```

count	carrera
3	Industrial
1	Telecomunicaciones
4	Computacion
2	Petrolera

(4 rows)

Para ello, mediante un “COUNT” que nos ayudará a contar las veces que se repite la carrera y un “GROUP BY”, para agrupar el nombre de las carreras que se repiten.

13)Nombre de la carrera que tiene a la persona más joven. Agrupar los datos.

```
gpolt_2024_2_319254725=> SELECT carrera, MIN(edad) AS maxEdad FROM alumno GROUP BY carrera ORDER BY maxEdad ASC LIMIT 1;
```

carrera	maxedad
Computacion	19

(1 row)

Para este inciso, se hizo el uso de tres cláusulas importantes como “GROUP BY” para agrupar las carreras repetidas y solo se muestre una vez , “ORDER BY”, para mostrar hasta arriba al alumno de menor edad y “LIMIT 1” para solo mostrar el dato que nos interesa, que es la carrera con la persona más joven

14)Nombre de la carrera que tiene a la persona más grande. Agrupar los datos.

```
gpolt_2024_2_319254725=> SELECT carrera, MAX(edad) AS maxEdad FROM alumno GROUP BY carrera ORDER BY maxEdad DESC LIMIT 1;
```

carrera	maxedad
Petrolera	30

(1 row)

Para este inciso, se hizo el uso de tres cláusulas importantes como “GROUP BY” para agrupar las carreras repetidas y solo se muestre una vez (aunque también si no lo pones, el manejador no te permite mostrar los resultados), “ORDER BY”, para mostrar hasta arriba al alumno de mayor edad y “LIMIT 1” para solo mostrar el dato que nos interesa, que es la carrera con la persona más vieja

15)Nombre de la carrera que tiene a la persona más joven. Usar subconsultas.

```
gpolt_2024_2_319254725=> SELECT carrera, edad FROM alumno WHERE edad = (SELECT MIN(edad) FROM alumno);
```

carrera	edad
Computacion	19

(1 row)

Ahora, mediante el uso de una subconsulta se obtuvo el alumno con la edad más pequeña, no hay nada más que agregar

16)Nombre de la carrera que tiene a la persona más grande. Usar subconsultas

```
gpolt_2024_2_319254725=> SELECT carrera, edad FROM alumno WHERE edad = (SELECT MAX(edad) FROM alumno);
```

carrera	edad
Petrolera	30

(1 row)

Igual al punto anterior, se uso una subconsulta para obtener el alumno, esta vez, con la edad más alta

17)Promedio de edad por carrera

```
gpolt_2024_2_319254725=> SELECT carrera, AVG(edad) AS promEdad FROM alumno GROUP BY carrera;
```

carrera	promedad
Industrial	24.666666666666667
Telecomunicaciones	24.000000000000000
Computacion	23.250000000000000
Petrolera	28.500000000000000

(4 rows)



Para este inciso se utilizó la función de agregación “AVG”, que sirve para calcular promedios entre los valores que se le lleguen a pasar a esta función. Además se agrupó la carrera.

18) borrar al asesor Adolfo Millan

```
gpo1t_2024_2_319254725=> DELETE FROM asesor WHERE idasesor = 'as-3';  
DELETE 1  
gpo1t_2024_2_319254725=> SELECT * FROM asesor;  
 idasesor | nompila | appat  
-----+-----+-----  
 as-1     | Jorge   | Campos  
 as-2     | Laura   | Sandoval  
 as-4     | Fernando | Arreola  
(3 rows)
```

19) Actualizar el id del asesor Fernando Arreola, asignar "as-5"

```
gpo1t_2024_2_319254725=> UPDATE asesor SET idasesor = 'as-5' WHERE idasesor = 'as-4';  
UPDATE 1  
gpo1t_2024_2_319254725=> SELECT * FROM asesor;  
 idasesor | nompila | appat  
-----+-----+-----  
 as-1     | Jorge   | Campos  
 as-2     | Laura   | Sandoval  
 as-5     | Fernando | Arreola  
(3 rows)
```

20) inicie sesión en dos terminales , conectándose en ambas a la base de datos donde almaceno los datos del presente ejercicio  
- en ambas sesiones, inicie una transacción

```
postgres=# BEGIN; -- T1 postgres=# BEGIN; -- T2  
BEGIN  
postgres=# █ BEGIN  
postgres=# █
```

- en la ventana A, actualice el apellido del alumno con id = 7 por "Suarez"

```
postgres=# UPDATE alumno SET appat = 'Mariya' WHERE idalumno = 7;  
UPDATE 1
```

Aquí, en vez de cambiar el apellido por 'Suarez', lo cambié por 'Mariya', ya que el registro con id 7, ya contaba con el apellido 'Suarez', y si se cambiaba o no, no se mostrarían lo que el profesor quiere que veamos que pasa aquí.

- en la ventana B, seleccione toda la información del alumno con id = 7. ¿Qué observa?

```
postgres=# SELECT * FROM alumno;
```

idalumno	nompila	appat	idasesor	carrera	edad
6	Isaac	Lemus	as-1	Petrolera	30
9	David	Rivera	as-1	Industrial	25
1	Mauricio	Barrientos	as-1	Petrolera	27
2	Mario	Tabura	as-2	Telecomunicaciones	24
3	Luz	Rueda	as-1	Computacion	27
4	Jorge	Santillan		Industrial	25
8	Pablo	Gonzales	as-2	Computacion	23
7	Gabriela	Suarez		Industrial	24
5	Gabriela	Gaytan		Computacion	19
10	Dayana	Plata	as-5	Computacion	24

(10 rows)

Claramente se observa que no se hizo el cambio, ya que, como yo lo veo, al iniciar una transacción, esta se "aislará" en el modo de que, el cambio que hagamos ahí afecte o no (dependiendo si se pone un COMMIT o un ROLLBACK) lo que ya llevamos de nuestra base de datos. En esta ocasión, al no haber terminado la transacción con, ya sea COMMIT o ROLLBACK, no se notará cambio alguno en la otra terminal.

- en la ventaba A, ingrese commit

```
postgres=# COMMIT;  
COMMIT
```

- en la ventana B, seleccione toda la información del alumno con id = 7, posteriormente rollback. ¿Qué observa?

```
postgres=# SELECT * FROM alumno WHERE idalumno = 7;
 idalumno | nompila | appat | idasesor | carrera | edad
-----+-----+-----+-----+-----+-----
         7 | Gabriela | Mariya |          | Industrial | 24
(1 row)
```

Lo que mencionaba, al momento de poner el COMMIT en la ventana A, los cambios se han realizado, es por eso que ya en la ventana B, se ven reflejados esos cambios.