



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**



FACULTAD DE INGENIERÍA

ALUMNO

SILVERIO MARTÍNEZ ANDRÉS

MATERIA

BASES DE DATOS

PROFESOR

FERNANDO ARREOLA FRANCO

GRUPO

1

EJERCICIO 2_7

SEMESTRE 2024 - 2

UNAM

Ejercicio 2_7

De las siguientes tablas:

id_Alumno	nombre_Alumno	id_Asesor
1	Mauricio Barrientos	as-1
2	Mario Tabura	as-2
3	Luz Rueda	as-1
4	Jorge Santillan	
5	Gabriela Gaytan	as-3

id_Asesor	nombre_Asesor
as-1	Jorge Campos
as-2	Laura Sandoval
as-3	Adolfo Millan
as-4	Fernando Arreola

Se desea saber:

- Se desea conocer aquellos alumnos que ya cuentan con asesor. Interesa el nombre del asesor y del alumno.
- Se desea conocer aquellos alumnos que ya cuentan con asesor y aquellos que no. Interesa el nombre del asesor y del alumno.
- Se desea conocer aquellos asesores que ya cuentan con tesista y aquellos que no. Interesa el nombre del asesor y del alumno.
- Se desea conocer aquellos alumnos que ya cuentan con asesor, así como los alumnos sin asesor y los asesores sin alumnos. Interesa el nombre del asesor y del alumno.
- Se desea conocer aquellos alumnos sin asesor y los asesores sin alumnos. Interesa el nombre del asesor y del alumno.
- Se desea conocer aquellos alumnos sin asesor y los asesores sin alumnos, Interesa el nombre del asesor y del alumno. Emplear joins.
- Se desea conocer aquellos asesores sin alumnos. Interesa el nombre del asesor. Emplear joins.

Para ello se implementó las tablas “alumno” y “asesor” en postgres para poder resolver este ejercicio, obteniendo los siguientes resultados:

- Para conocer a los alumnos que ya cuentan con asesor:

```
gpo1t_2024_2_319254725=> SELECT alumno.nompila, alumno.appat, asesor.nompila AS nomAsesor, asesor.appat AS apasesor FROM alumno INNER JOIN asesor ON alumno.idasesor = asesor.idasesor;
```

nompila	appat	nomasesor	apasesor
Mauricio	Barrientos	Jorge	Campos
Mario	Tabura	Laura	Sandoval
Luz	Rueda	Jorge	Campos
Gabriela	Gaytan	Adolfo	Millan

(4 rows)

Para ello, se utilizó un INNER JOIN, con el propósito de unir el “idasesor” que es coincidente en ambas tablas, es decir, que el alumno si tenga un asesor asignado

- Para conocer a los alumnos que cuentan y no con asesor

```
gpo1t_2024_2_319254725=> SELECT alumno.nompila AS alumno, asesor.nompila AS asesor FROM alumno LEFT JOIN asesor ON alumno.idasesor = asesor.idasesor;
```

alumno	asesor
Mauricio	Jorge
Mario	Laura
Luz	Jorge
Jorge	
Gabriela	Adolfo

(5 rows)

Para conocer a los alumnos que cuentan con asesor y los que no, se utilizó un LEFT JOIN, donde, de la tabla del lado izquierdo, que en este caso es “alumno”, se devolverán todas las filas coincidentes del lado izquierdo, respecto al derecho, mostrando así los alumnos con y sin asesor.

- Para conocer aquellos asesores que cuentan y no con un tesista

```
gpo1t_2024_2_319254725=> SELECT alumno.nompila AS alumno, asesor.nompila AS asesor FROM alumno RIGHT JOIN asesor ON alumno.idasesor = asesor.idasesor;
```

alumno	asesor
Mauricio	Jorge
Mario	Laura
Luz	Jorge
Gabriela	Adolfo
	Fernando

(5 rows)

Caso contrario al inciso anterior, se utilizó un RIGHT JOIN, donde, de la tabla del lado derecho, que en este caso es “asesor”, se devolverán todas las filas coincidentes del lado derecho, respecto al izquierdo, mostrando así los asesores que cuentan con y sin tesista.

- Para conocer los alumnos que cuentan y no cuentan con asesor, así como los asesores sin alumnos

```
gpolt_2024_2_319254725=> SELECT alumno.nompila AS alumno, asesor.nompila AS asesor FROM alumno FULL OUTER JOIN asesor ON alumno.idasesor = asesor.idasesor;
```

alumno	asesor
Mauricio	Jorge
Mario	Laura
Luz	Jorge
Jorge	
Gabriela	Adolfo
	Fernando

(6 rows)

Para este caso, se utilizó el FULL JOIN, ya que buscábamos que todos los datos de ambas tablas se vieran presentes, mostrando así los alumnos que cuentan con asesor, así como los asesores sin alumnos.

- Para conocer a los alumnos sin asesor y los asesores sin alumnos

```
gpolt_2024_2_319254725=> SELECT alumno.nompila AS alumno, asesor.nompila AS asesor FROM alumno FULL OUTER JOIN asesor ON alumno.idasesor = asesor.idasesor WHERE alumno.idasesor IS NULL OR asesor.idasesor IS NULL;
```

alumno	asesor
Jorge	
	Fernando

(2 rows)

Igual que en el inciso anterior, se utilizó un FULL JOIN para concatenar todos los datos de ambas tablas, pero al ponerle un WHERE, donde el id del asesor sea nulo en ambas tablas, es cuando se nos muestra el resultado, donde Jorge no tiene un asesor, y Fernando no asesora a nadie

- Para conocer a los alumnos sin asesor

```
gpo1t_2024_2_319254725=> SELECT alumno.nompila AS alumno, asesor.nompila AS asesor FROM alumno FULL OUTER JOIN asesor ON alumno.idase  
sor = asesor.idasesor WHERE asesor.idasesor IS NULL;  
alumno | asesor  
-----+-----  
Jorge  |  
(1 row)
```

- Para conocer a los asesores sin alumnos

```
gpo1t_2024_2_319254725=> SELECT alumno.nompila AS alumno, asesor.nompila AS asesor FROM alumno FULL OUTER JOIN asesor ON alumno.idase  
sor = asesor.idasesor WHERE alumno.idalumno IS NULL;  
alumno | asesor  
-----+-----  
| Fernando  
(1 row)
```

Aquí, algo similar al inciso anterior, lo que se hizo fue poner un WHERE, donde el id del alumno fuera nulo, con el propósito de conocer el asesor que no asesora a ningún alumno, en este caso, Fernando.