

1D.A.W

# TABLA YOUTUBE

Iñaki Borrego Bau

---

Contenido

1. Completa la tabla comentarios ..... 2

2. Parte 1 ..... 3

3. Realiza las siguientes consultas. .... 7

# 1. Completa la tabla comentarios

- Inserta 5 filas nuevas en la tabla videos y comentarios:

INSERT INTO Videos (titulo, descripcion, url, user\_id)

VALUES ('Cocina mediterránea - Receta de paella', 'En este video te enseñamos a preparar una deliciosa paella mediterránea.', 'https://www.youtube.com/watch?v=abc123', 3),

('Tutorial de maquillaje para principiantes', 'Aprende cómo realizar un maquillaje natural perfecto paso a paso.', 'https://www.youtube.com/watch?v=def456', 4),

('Entrevista a Elon Musk sobre el futuro de SpaceX', 'El CEO de SpaceX comparte sus visiones y planes futuros para la exploración espacial.', 'https://www.youtube.com/watch?v=ghi789', 5),

('Concierto en vivo de Coldplay en París', 'Revive el emocionante concierto de Coldplay en la Ciudad de la Luz.', 'https://www.youtube.com/watch?v=jkl012', 2),

('Tutorial de yoga para principiantes', 'Sigue esta guía paso a paso para iniciarte en la práctica del yoga.', 'https://www.youtube.com/watch?v=mno345', 1);

INSERT INTO Comentarios (comentario, video\_id, user\_id)

VALUES ('¡Qué delicia! Me encanta la paella.', 6, 2),

('Muy útil, gracias por compartirlo.', 7, 1),

('Excelente entrevista, me inspira mucho.', 8, 3),

('Coldplay siempre entrega grandes performances en vivo.', 9, 4),

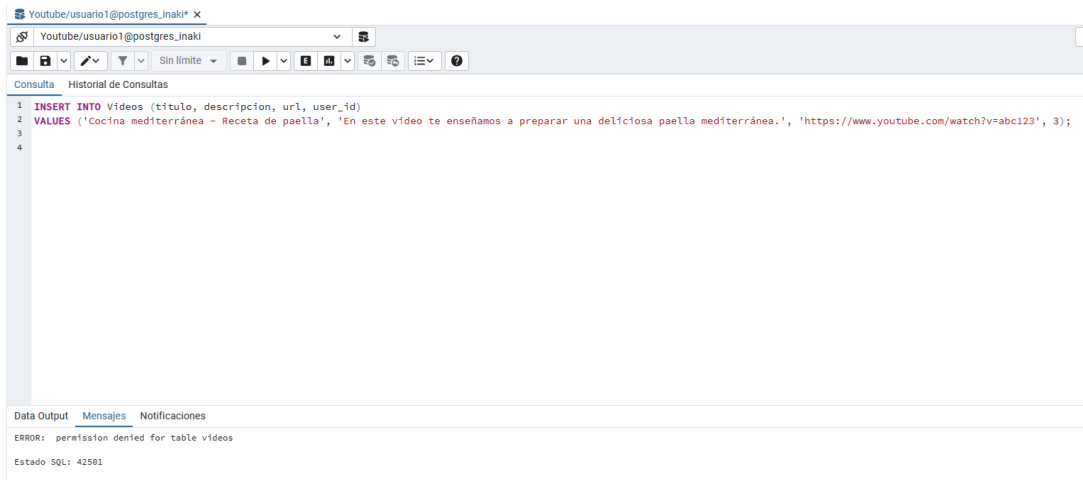
('¡Necesitaba esto! Gracias por el tutorial.', 10, 5);

## 2. Parte 1

- Crea un nuevo usuario con permisos de lectura para la tabla "videos". Luego, intenta realizar una inserción en la tabla "videos" y verifica si el usuario tiene permisos para realizar esta acción.

CREATE USER usuario1 WITH PASSWORD 'usuario1';

GRANT SELECT ON videos TO usuario1;



The screenshot shows a PostgreSQL client window titled 'Youtube/usuario1@postgres\_inaki'. The SQL editor contains the following code:

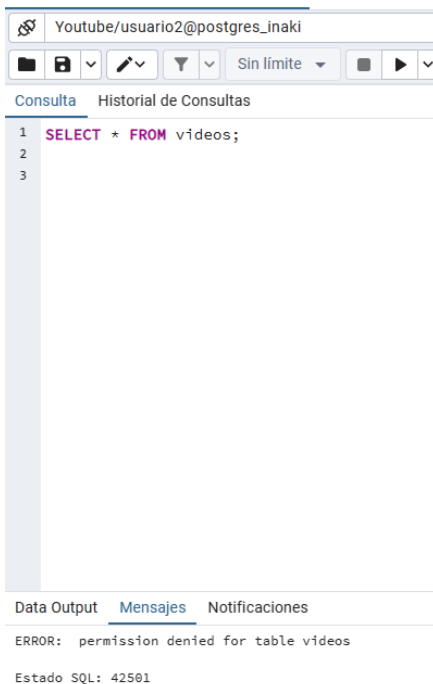
```
1 INSERT INTO Videos (titulo, descripción, url, user_id)
2 VALUES ('Cocina mediterránea - Receta de paella', 'En este video te enseñamos a preparar una deliciosa paella mediterránea.', 'https://www.youtube.com/watch?v=abc123', 3);
3
4
```

Below the editor, the 'Data Output' tab is selected, displaying the error message: 'ERROR: permission denied for table videos'. The 'Estado SQL' is 42501.

- Crea un nuevo usuario con permisos de escritura para la tabla "videos". Luego, intenta realizar una consulta SELECT en la tabla "videos" y verifica si el usuario tiene permisos para realizar esta acción.

CREATE USER usuario2 WITH PASSWORD 'usuario2';

GRANT INSERT, UPDATE, DELETE ON videos TO usuario2;



The screenshot shows a PostgreSQL client window titled 'Youtube/usuario2@postgres\_inaki'. The SQL editor contains the following code:

```
1 SELECT * FROM videos;
2
3
```

Below the editor, the 'Data Output' tab is selected, displaying the error message: 'ERROR: permission denied for table videos'. The 'Estado SQL' is 42501.

- Concede permiso de lectura y escritura en la tabla videos de tu usuario creado anteriormente. Verifica si el usuario tiene permisos para realizar esta acción

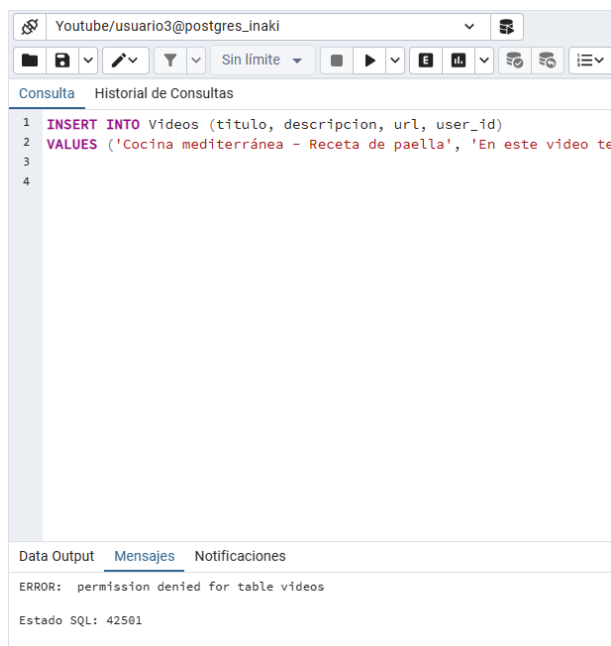
```
GRANT SELECT, INSERT, UPDATE, DELETE ON videos TO usuario3;
```

\*Ya tenia echa la de abajo y recado las cosas.

- Revoca los permisos de escritura del usuario que has creado en el ejercicio anterior y otórgale permisos de lectura en su lugar. Intenta realizar una inserción en la tabla "videos" y verifica si el usuario tiene permisos para realizar esta acción.

```
REVOKE SELECT, INSERT, UPDATE, DELETE ON videos FROM usuario3;
```

```
GRANT SELECT ON videos TO usuario3;
```



- Crea un nuevo usuario llamado "practicass" con la contraseña "becario" y otorgarle permisos de lectura en las tablas videos y comentarios. Verifica si el usuario tiene permisos para realizar esta acción.

```
CREATE USER practicas WITH PASSWORD 'becario';
```

```
GRANT SELECT ON videos, comentarios TO practicas;
```

Youtube/practicass@postgres\_inaki

Consulta Historial de Consultas

1 `Select *FROM comentarios;`

Data Output Mensajes Notificaciones

	id_comentario [PK] integer	comentario character varying (255)	video_id integer	user_id integer
1	1	precioso	5	
2	2	bello	2	
3	3	elegante	4	
4	4	encantador	3	
5	5	gracioso	3	
6	6	precioso	1	

- Otorga permisos de lectura en la tabla "videos" a un usuario y luego revoca los permisos de lectura de un campo específico, por ejemplo, "descripción". Intenta realizar una consulta SELECT en la tabla "videos" y verifica si el usuario tiene permisos para ver el campo "descripción".

```
CREATE USER usuario4 WITH PASSWORD 'usuario4';
```

```
GRANT SELECT ON TABLE Videos TO nombre_usuario;
```

```
REVOKE SELECT (descripcion) ON TABLE Videos FROM usuario4;
```

NO ME FUNCIONA

El chat por resolverlo me dice esto

```
CREATE OR REPLACE VIEW Videos_vista AS
```

```
SELECT id_video, titulo, url, user_id
```

```
FROM Videos;
```

```
GRANT SELECT ON Videos_vista TO nombre_usuario;
```

```
SELECT * FROM Videos_vista;
```

- Permisos de lectura solo para el campo titulo de la tabla videos al usuario "practicass y verifica si el usuario tiene permisos para realizar esta acción. Realiza las siguientes consultas.

GRANT SELECT (titulo) ON TABLE Videos TO practicas;

The screenshot shows the pgAdmin 4 web interface. At the top, the browser address bar displays 'youtube/practicas@PostgreSQL 16'. Below the address bar is a toolbar with icons for file operations (new, open, save, copy, paste, delete), a search icon, and a 'No limit' dropdown. The main content area is divided into two panes. The left pane, titled 'Query', shows a SQL query: `SELECT titulo FROM Videos;`. The right pane, titled 'Query History', is empty. Below the query panes is a 'Data Output' section. It has tabs for 'Messages' and 'Notifications'. The 'Data Output' tab is active, showing a table with 5 rows and 1 column. The table has a header row with the column name 'titulo' and a data type 'character varying (255)'. The rows contain the following titles: 'Salir Corriendo', 'LOS SECRETOS - Pero A Tu Lado (acustico) video', 'Mix: Double Dare - We Belong', 'CANTADITAS DE SIEMPRE by DJ Kachu', and 'JERUSALEMA CHALLENGE IES ALMENARA'. A lock icon is visible in the top right corner of the table.

	titulo character varying (255)
1	"Salir Corriendo
2	LOS SECRETOS - Pero A Tu Lado (acustico) video
3	Mix: Double Dare - We Belong
4	CANTADITAS DE SIEMPRE by DJ Kachu
5	JERUSALEMA CHALLENGE IES ALMENARA

### 3. Realiza las siguientes consultas.

- Consulta el número de videos subidos por cada usuario.

```
SELECT u.username, COUNT(v.id_video) AS num_videos_subidos
FROM Usuarios u
LEFT JOIN Videos v ON u.Id_Usuario = v.user_id
GROUP BY u.username;
```

	username character varying (255) 🔒	num_videos_subidos bigint 🔒
1	jordi	0
2	carles	1
3	gerard	1
4	juan	2
5	andrei	1

- Consulta la cantidad de comentarios realizados por cada usuario en la tabla Comentarios.





```
SELECT u.username, COUNT(c.id_comentario) AS num_comentarios_realizados
FROM Usuarios u
LEFT JOIN Comentarios c ON u.Id_Usuario = c.user_id
GROUP BY u.username;
```

	username character varying (255) 🔒	num_comentarios_realizados bigint 🔒
1	jordi	3
2	carles	4
3	gerard	1
4	juan	5
5	andrei	5

- Consulta los comentarios realizados por el usuario juan que contengan la palabra "encantador".

```
SELECT *
FROM Comentarios
WHERE user_id = (SELECT Id_Usuario FROM Usuarios WHERE username = 'juan')
AND comentario LIKE '%encantador%';
```



<b>id_comentario</b> [PK] integer 	<b>comentario</b> character varying (255) 	<b>video_id</b> integer 	<b>user_id</b> integer 
--	--	---	---