

# Clase 14 practica SQL

Esta práctica consiste en reforzar los conceptos ya vistos, ya que serán de gran utilidad para la creación de la base de datos de películas de nuestro proyecto dividido en 3 entregables.

Para iniciar ya el proceso de desarrollo, en donde ustedes serán los encargados de bases de datos y hacer que el backend funcione, realizaremos lo siguiente:

1. Si estamos en los computadores de la universidad, únicamente instalaremos pgadmin, <https://www.pgadmin.org/download/> (<https://www.pgadmin.org/download/>), no será necesario instalar postgres.
2. En los computadores personales la instalación de estas herramientas las hicimos previamente.
3. Instalar visual estudio code <https://code.visualstudio.com/download> (<https://code.visualstudio.com/download>)
4. Instalar node JS <https://nodejs.org/en/> (<https://nodejs.org/en/>)
5. Para proba los servicios <https://insomnia.rest/> (<https://insomnia.rest/>) o <https://www.postman.com/downloads/> (<https://www.postman.com/downloads/>)

**Recuerde tener instalado git y clonar previamente el repositorio de trabajo**

## Conexión a base de datos:

Para conectarse a la base de datos puede utilizar la de la red local (la cual solo funcionará en clase) o la remota la cual está siendo ejecutada sobre el servicio RDS de AWS. La ip se otorgará en el salón de clase, por favor no comparta las credenciales con personas ajenas al curso por seguridad.

Para iniciar la conexión abre pgadmin y sobre la opción **Servers** del menú de la izquierda, da clic derecho y elige **create server**

**Create - Server**

General Connection SSL SSH Tunnel Advanced

Name: Nombre de la conexión

Server group: Servers

Background: ☐

Foreground: ☐

Connect now?: ☒

Comments:

⚠ Either Host name, Address or Service must be specified.

Close Reset Save

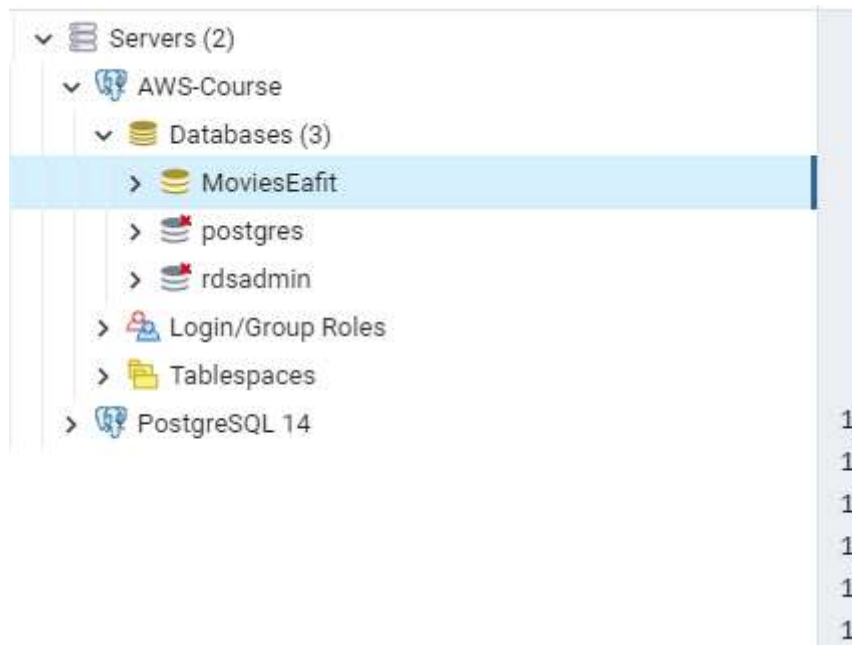
Proporcione un nombre a la conexión y luego de clic en el tab connection:

The screenshot shows a 'Create - Server' dialog box with the 'Connection' tab selected. The 'Host name/address' field is empty and has a red warning icon next to it. The 'Port' is set to 5432. The 'Maintenance database' is set to 'MoviesEafit'. The 'Username' is 'su\_usuario'. 'Kerberos authentication?' is disabled. The 'Password' field is masked with dots. 'Save password?' is disabled. The 'Role' and 'Service' fields are empty. A red error message at the bottom states: 'Either Host name, Address or Service must be specified.' The 'Save' button is highlighted in blue.

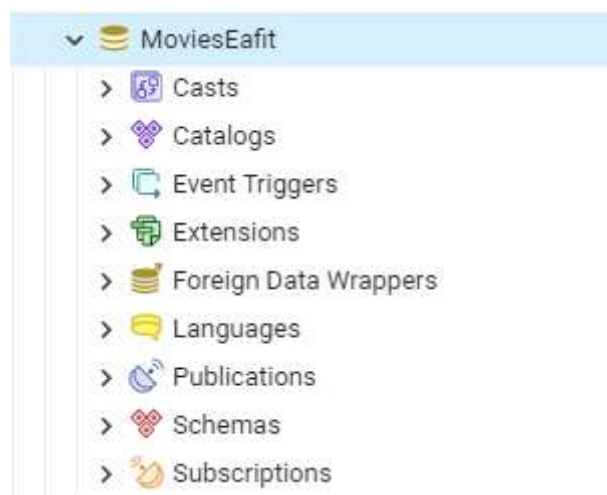
Luego de clic en save y ya tendrá acceso a la base de datos.

## Visualización de base de datos

Cuando realice la conexión despliegue la opción con el nombre asignado previamente y elija la base de datos con la que trabajaremos de aquí en adelante "Movies Eafit".



Las bases de datos, tiene la siguiente estructura:



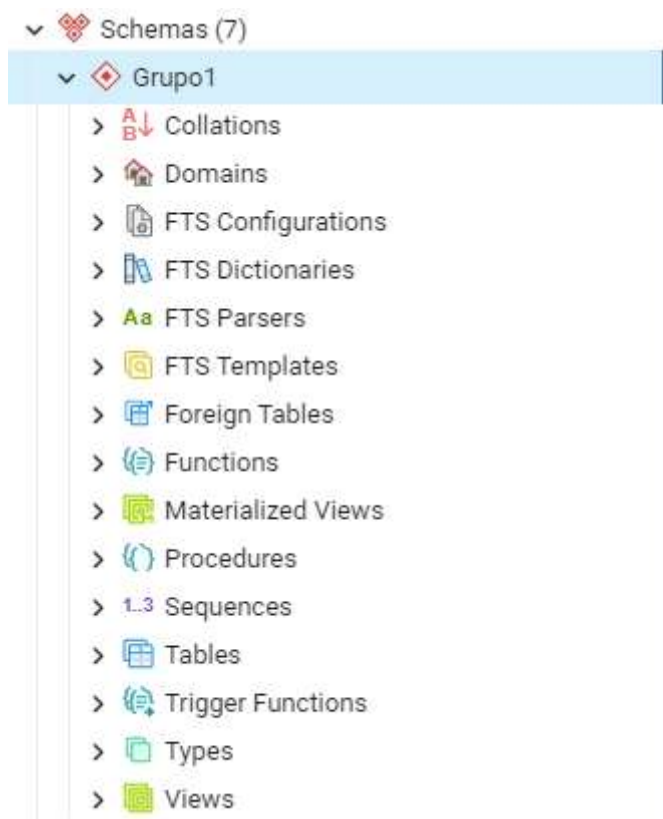
Para ver los esquemas disponibles donde empezará a trabajar cada grupo, desplegar Schemas para ver más.

Cada grupo tiene la siguiente asignación de esquemas:

- sa\_ma\_hu (Grupo 1)
- cvj (Grupo 2)
- miguapa (Grupo 3)
- sapos\_g (Grupo 4)
- dinamita (Grupo 5)
- avcoro (Grupo 6)

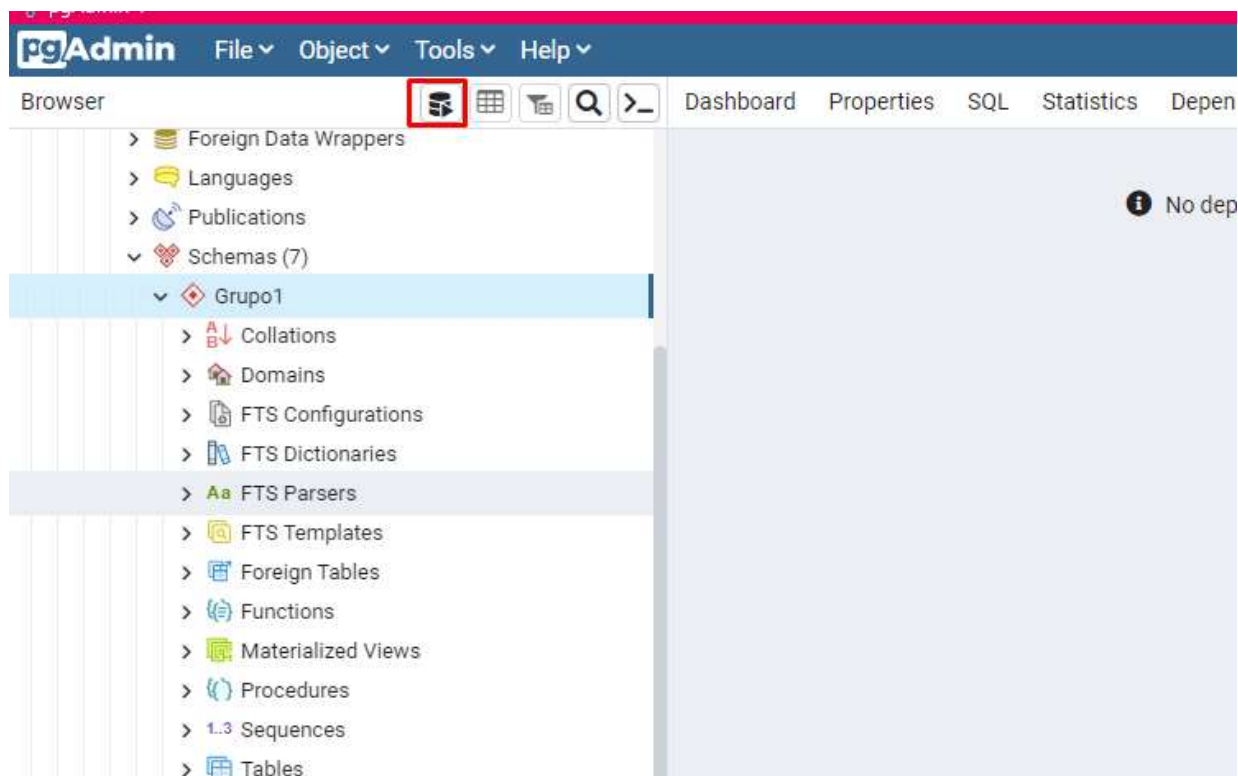
(Si alguno falta en la lista por favor informar)

Para ver el contenido del esquema, desplegar alguno de los disponibles. Aquí se podrán ver las tablas, relaciones entre tablas, procedimientos, funciones, tipos de datos, triggers, vistas.

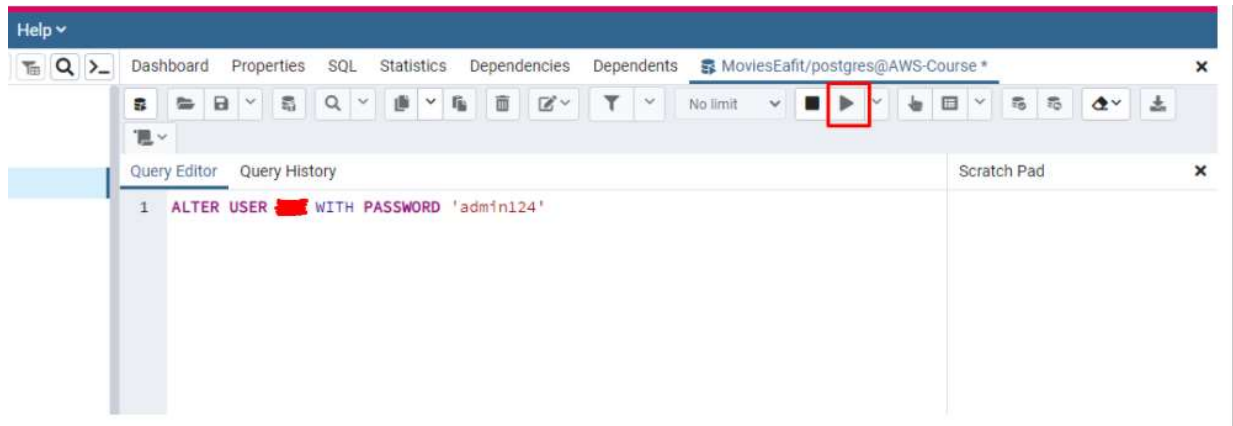


Considerando el trabajo 1 (aún falta el proceso de revisión y la normalización del resultado inicial), vamos a contruir la estructura de la base de datos, para ellos vamos a tomar el modelo de cada uno entregado en el trabajo (recuerden que luego debemos hacer los ajustes de normalización, por ahora solo tomaremos el modelo tal cual lo tenemos). Vamos a crear las tablas USANDO SENTENCIAS SQL, NO LA INTERFAZ GRÁFICA EN ESTE CASO.

Para ello damos clic en el siguiente icono de la barra superior.

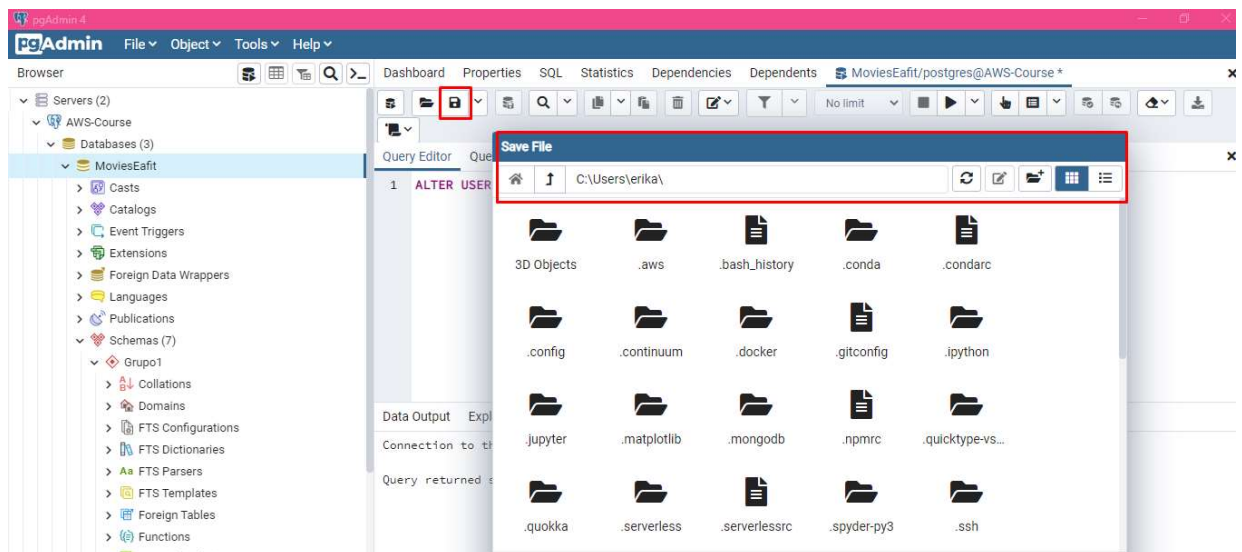


Cambiaremos la clave por una que solo conozca el equipo. Ejecute el comando tal cual se muestra en la imagen. Si el cambio fue exitoso usted perderá conexión con la base de datos, por lo que deberá conectarse de nuevo.



Y ahora sí, sobre el mismo archivo cree las sentencias de creación de las tablas definidas por usted en un inicio, revise las clases anteriores para lograr esto. E inserte tres registros por cada tabla. Deben contener los constraints.

Al finalizar presione el icono de guardar para almacenar el archivo. Este script deberá subirlo al git utilizando los comandos vistos previamente en clase.



## Parte 2:

- Vamos a practicar las sentencias update y delete. Crea una sentencia que actualice la contraseña de un usuario siguiendo

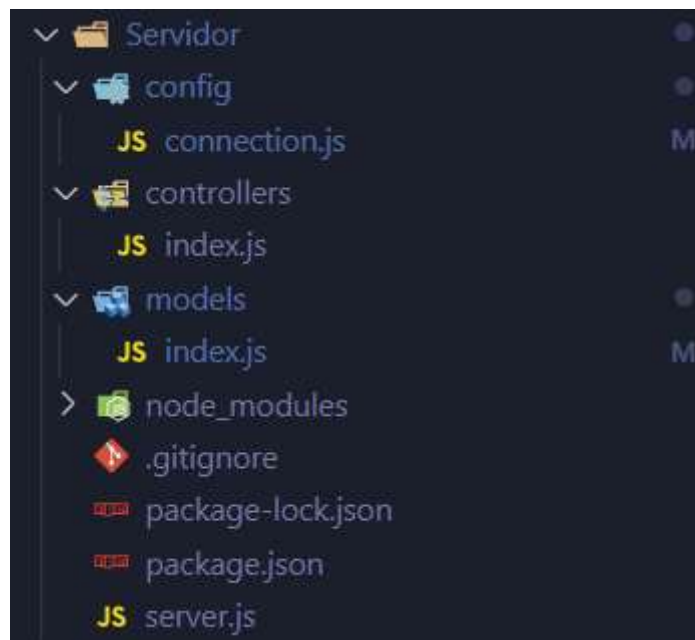
```
UPDATE table_name
SET column = value
Where column = condition
```

Importante agregar siempre el where a la sentencia update, ya que si se hace con dicha clausula, pueden afectar todos los registros.

- Utiliza la sentencia delete para eliminar un registro de la base de datos `DELETE FROM table_name WHERE column = condition;`

Teniendo en cuenta lo anterior, vamos a crear un CRUD básico de prueba que nos ayudará a entender como funciona el backend de una aplicación web, además de como se integran las bases de datos en la vida real. En este caso nos será de ayuda para entender mejor el desarrollo de la aplicación moviesEafit

Para hacer la práctica, descargar los cambios de la rama develop `git pull origin develop`, con esto ya tendremos la nueva carpeta Servidor, la cual nos permitira hacer un CRUD por medio de servicios rest, exponiendo endpoints que luego serán utilizados por un front o sitio web para consultar e interactuar con la base de datos.



Esta estructura está basada en MVC donde tenemos un modelo y un controlador. Para hacer las modificaciones, hacer los siguientes pasos:

1. Ir a la carpeta config y en el script connection añadir las credenciales de conexión, son las mismas utilizada en pgadmin4.

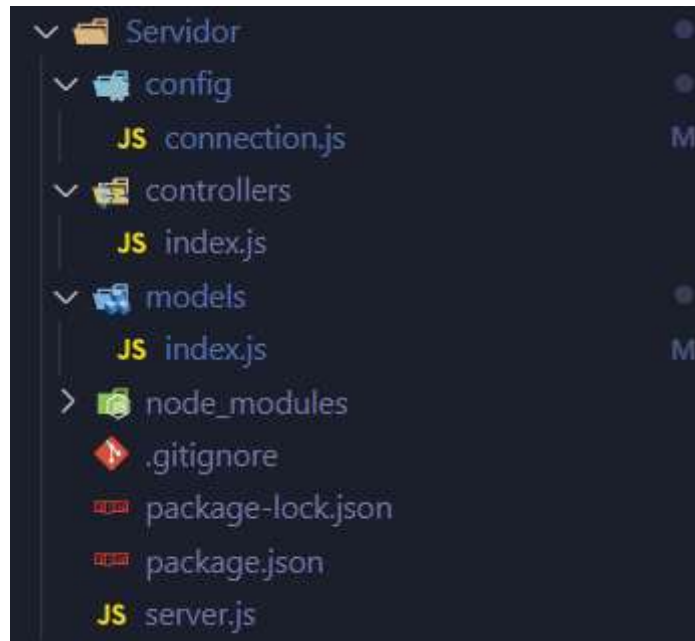
```
const { Pool } = require("pg")

const pool = new Pool({
  user: "",
  host: "",
  database: "",
  password: "",
  port: 5432,
});
```

```
pool.on('connect', () => {
  console.log('MoviesEafit DB up!');
});

module.exports = pool
```

2. En la carpeta models, se encuentra el script que permite agregar las setencias SQL para interactuar con la base de datos y crear, actualizar, eliminar e insertar datos. Cree las sentencias necesarias para obtener un usuario, insertar usuario, actualizar usuario, eliminar usuario. Haga esto mismo para la tabla de peliculas.



3. En la carpeta controller se encuentra el script que permite agregar los puntos de acceso para enviar las peticiones desde un cliente o interfaz y permitir la interacción con la base de datos.

Los puntos de acceso esperados para ejecutar con éxito esta práctica son:

/\*\* Rutas esperadas para este ejercicio practico:

- /api/login
- /api/profile or /api/user
- /api/profile/create or /api/user/create
- /api/profile/update or /api/user/update
- /api/profile/crete or /api/user/delete
- /api/movies
- /api/movies/:id
- /api/movies/create
- /api/movies/update
- /api/movies/delete
- /

**Ahora, intenta identificar, que otro servicios necesarias para interactuar con la base de datos y las demás tablas**



