

# **Hands-on 4: Operaciones SQL sobre la base de datos MexFlix**

Materia: Seminario de bases de datos

Fecha: 18/05/25

Nombre: Diego Martín Domínguez Hernández

Profesor: José Antonio Aviña Méndez

# 1. Sentencias SQL

```
-- Inserciones --

-- Usuarios
INSERT INTO usuario (UsuarioID, Nombre, Apellido, RFC, Direccion, Email, FechaRegistro)
VALUES
(1, 'Ana', 'García', 'GARA750101XXX', 'Av. México 123', 'ana.garcia@email.com', '2024-05-01'),
(2, 'Luis', 'Pérez', 'PERL800202YYY', 'Calle Reforma 456', 'luis.perez@email.com', '2024-05-02');

-- Membresías
INSERT INTO membresia (MembresiaID, UsuarioID, Tipo, Costo, FechaInicio, FechaFin)
VALUES
(1, 1, 'Premium', 299.00, '2024-05-01', '2024-06-01'),
(2, 2, 'Standard', 159.00, '2024-05-02', '2024-06-02');

-- Tarjetas
INSERT INTO tarjeta (TarjetaID, UsuarioID, Numero, Tipo, FechaExpiracion)
VALUES
(1, 1, '4111111111111111', 'VISA', '2027-01-01'),
(2, 2, '5500000000000004', 'MASTERCARD', '2026-12-01');

-- Dispositivos
INSERT INTO dispositivo (DispositivoID, Tipo)
VALUES
(1, 'smartphone'), (2, 'laptop'), (3, 'TV');

-- Categorías
INSERT INTO categoria (CategoriaID, Nombre)
VALUES
(1, 'Acción'), (2, 'Suspense'), (3, 'Documental');

-- Películas
INSERT INTO pelicula (PeliculaID, CategoriaID, Titulo, Duracion, FechaEstreno, EsEstreno)
VALUES
(1, 1, 'Misión Imposible', 120, '2023-12-01', TRUE),
(2, 2, 'El Código Da Vinci', 150, '2022-09-15', FALSE);

-- Series
INSERT INTO serie (SerieID, CategoriaID, Titulo, FechaEstreno)
VALUES
(1, 2, 'Breaking Bad', '2008-01-20'),
(2, 3, 'Planeta Tierra', '2006-03-05');

-- Actores
INSERT INTO actor (ActorID, Nombre, Apellido)
VALUES
(1, 'Tom', 'Cruise'), (2, 'Bryan', 'Cranston');

-- Serie_Actor
INSERT INTO serie_actor (SerieID, ActorID)
VALUES
(1, 2);
```

```

-- Episodios
INSERT INTO episodio (EpisodioID, SerieID, Titulo, Duracion, NumeroTemporada,
    NumeroEpisodio)
VALUES
(1, 1, 'Piloto', 58, 1, 1),
(2, 2, 'La vida en el agua', 60, 1, 1);

-- Consumo
INSERT INTO consumo (ConsumoID, UsuarioID, PeliculaID, EpisodioID, TipoContenido,
    FechaHoraInicio, FechaHoraFin, DispositivoID, EsRenta)
VALUES
(1, 1, 1, NULL, 'Pelicula', '2024-05-10 20:00', '2024-05-10 22:00', 1, TRUE),
(2, 2, NULL, 1, 'Episodio', '2024-05-11 18:00', '2024-05-11 19:00', 2, FALSE);

-- Sesiones
INSERT INTO sesion (SesionID, UsuarioID, DispositivoID, FechaHoraInicio, FechaHoraFin)
VALUES
(1, 1, 1, '2024-05-10 19:50', '2024-05-10 22:05'),
(2, 2, 2, '2024-05-11 17:55', '2024-05-11 19:05');

-- Facturas
INSERT INTO factura (FacturaID, UsuarioID, Fecha, MontoCuotaMensual,
    MontoRentasAdicionales, MontoTotal, Estado)
VALUES
(1, 1, '2024-05-31', 299.00, 50.00, 349.00, 'Al corriente'),
(2, 2, '2024-05-31', 159.00, 0.00, 159.00, 'Deudor');

-- Consultas --

-- Usuarios por tipo de membresía
SELECT u.Nombre, u.Apellido, m.Tipo
FROM usuario u
JOIN membresia m ON u.UsuarioID = m.UsuarioID;

-- Películas de accion estrenadas en 2023
SELECT Titulo, FechaEstreno
FROM pelicula p
JOIN categoria c ON p.CategoriaID = c.CategoriaID
WHERE c.Nombre = 'Acción' AND EXTRACT(YEAR FROM p.FechaEstreno) = 2023;

-- Contenido consumido en mayo 2024
SELECT u.Nombre, c.TipoContenido, p.Titulo AS Pelicula, e.Titulo AS Episodio, c.
    FechaHoraInicio
FROM consumo c
LEFT JOIN usuario u ON c.UsuarioID = u.UsuarioID
LEFT JOIN pelicula p ON c.PeliculaID = p.PeliculaID
LEFT JOIN episodio e ON c.EpisodioID = e.EpisodioID
WHERE c.FechaHoraInicio BETWEEN '2024-05-01' AND '2024-05-31';

-- Dispositivos usados por usuarios
SELECT DISTINCT u.Nombre, d.Tipo AS Dispositivo
FROM usuario u
JOIN sesion s ON u.UsuarioID = s.UsuarioID
JOIN dispositivo d ON s.DispositivoID = d.DispositivoID;

-- Consumo total y saldo actual del usuario
SELECT u.Nombre, f.MontoTotal, f.Estado

```

```

FROM usuario u
JOIN factura f ON u.UsuarioID = f.UsuarioID;

-- Actualizaciones --

-- Cambiar membresía a un usuario
UPDATE membresia
SET Tipo = 'Premium', Costo = 299.00, FechaInicio = '2024-05-15', FechaFin = '
    2024-06-15'
WHERE UsuarioID = 2;

-- Marcar a usuario "Al corriente" en factura
UPDATE factura
SET Estado = 'Al corriente'
WHERE UsuarioID = 2 AND Fecha = '2024-05-31';

-- Cambiar dispositivo de sesion a usuario
UPDATE sesion
SET DispositivoID = 3
WHERE SesionID = 1;

-- Marcar consumo 2 como renta
UPDATE consumo
SET EsRenta = TRUE
WHERE ConsumoID = 2;

-- Actualizar correo de usuario
UPDATE usuario
SET Email = 'ana.garcia2024@email.com'
WHERE UsuarioID = 1;

-- CREAR VISTAS --

-- Consumo mensual por usuario
SELECT u.Nombre, u.Apellido, c.TipoContenido, p.Titulo AS Pelicula, e.Titulo AS
    Episodio, c.FechaHoraInicio
FROM consumo c
LEFT JOIN usuario u ON c.UsuarioID = u.UsuarioID
LEFT JOIN pelicula p ON c.PeliculaID = p.PeliculaID
LEFT JOIN episodio e ON c.EpisodioID = e.EpisodioID
WHERE c.FechaHoraInicio BETWEEN '2024-05-01' AND '2024-05-31';

-- Usuarios y sus dispositivos
CREATE VIEW vista_usuarios_dispositivos AS
SELECT u.Nombre, d.Tipo AS Dispositivo
FROM usuario u
JOIN sesion s ON u.UsuarioID = s.UsuarioID
JOIN dispositivo d ON s.DispositivoID = d.DispositivoID;

-- Disparadores --

-- Crear tabla de auditoría pra registrar cambios en facturas
CREATE TABLE auditoria_factura (
    AuditoriaID SERIAL PRIMARY KEY,
    FacturaID INT,

```

```

    UsuarioID INT,
    Fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Operacion VARCHAR(20)
);

-- Función para el trigger
CREATE OR REPLACE FUNCTION registrar_auditoria_factura()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO auditoria_factura (FacturaID, UsuarioID, Operacion)
    VALUES (NEW.FacturaID, NEW.UsuarioID, TG_OP);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Cada que se actualice una factura, se registra en auditoría
CREATE TRIGGER trg_auditoria_factura
AFTER UPDATE ON factura
FOR EACH ROW
EXECUTE FUNCTION registrar_auditoria_factura();

```