



UNIVERSIDAD NACIONAL AGRARIA LA MOLINA

Facultad de Ciencias
Departamento Académico de Estadística e Informática

Sistemas de Gestión de Base de Datos II

Sesión Práctica: Consola MySQL con XAMPP

Práctica Dirigida – SGBD II (UNALM)

Trabajo en consola con MySQL (XAMPP) - Duración: 120 minutos

```
-- ■ PRÁCTICA DIRIGIDA - SGBD II (UNALM)
-- Trabajo en consola con MySQL (XAMPP)
-- Duración: 120 minutos

-- ■■ 1. CONECTARSE A MYSQL COMO ROOT
-- cd C:\xampp\mysql\bin
-- mysql -u root -p

-- ■■ 2. CREAR UNA NUEVA BASE DE DATOS DE PRÁCTICA
CREATE DATABASE IF NOT EXISTS bd_gestion;
USE bd_gestion;

-- ■■ 3. CREAR TABLAS DE UNA EMPRESA DE TRANSPORTE
CREATE TABLE buses (
    id_bus INT AUTO_INCREMENT PRIMARY KEY,
    placa VARCHAR(10) NOT NULL,
    modelo VARCHAR(50),
    capacidad INT,
    en_servicio BOOLEAN DEFAULT TRUE
);

CREATE TABLE conductores (
    id_conductor INT AUTO_INCREMENT PRIMARY KEY,
    nombres VARCHAR(100) NOT NULL,
    licencia VARCHAR(15) UNIQUE,
    fecha_ingreso DATE
);

CREATE TABLE rutas (
    id_ruta INT AUTO_INCREMENT PRIMARY KEY,
    origen VARCHAR(100),
    destino VARCHAR(100),
    distancia_km DECIMAL(6,2)
);

-- ■■ 4. INSERTAR DATOS DE EJEMPLO
INSERT INTO buses (placa, modelo, capacidad) VALUES
('B1234', 'Mercedes Sprinter', 15),
('C5678', 'Hyundai County', 20);

INSERT INTO conductores (nombres, licencia, fecha_ingreso) VALUES
('Luis Ramos', 'A1234567', '2020-02-01'),
('Carmen Vidal', 'B7654321', '2021-08-15');

INSERT INTO rutas (origen, destino, distancia_km) VALUES
('Lima', 'Cañete', 135.5),
('Lima', 'Chosica', 30.2);

-- ■■ 5. GESTIÓN DE USUARIOS Y PRIVILEGIOS
CREATE USER 'control_transporte'@'localhost' IDENTIFIED BY 'Control2025!';
GRANT SELECT, INSERT, UPDATE ON bd_gestion.* TO 'control_transporte'@'localhost';
FLUSH PRIVILEGES;

-- ■■ 6. MODIFICACIONES AVANZADAS
ALTER USER 'control_transporte'@'localhost' WITH MAX_USER_CONNECTIONS 1;
ALTER USER 'control_transporte'@'localhost' PASSWORD EXPIRE INTERVAL 30 DAY;

-- ■■ 7. CREAR PROCEDIMIENTOS NUEVOS (NO REPETIDOS)
DELIMITER //
CREATE PROCEDURE ListarConductores()
BEGIN
    SELECT * FROM conductores;
END //

CREATE PROCEDURE BusesEnRuta(IN servicio BOOLEAN)
BEGIN
    SELECT * FROM buses WHERE en_servicio = servicio;
```

```

END //

CREATE PROCEDURE RutaDetalle(IN id INT, OUT detalle VARCHAR(255))
BEGIN
    SELECT CONCAT(origen, ' a ', destino, ' - ', distancia_km, ' km') INTO detalle
    FROM rutas WHERE id_ruta = id;
END //
DELIMITER ;

-- ■■ 8. LLAMADAS Y VALIDACIONES
CALL ListarConductores();
CALL BusesEnRuta(TRUE);
CALL RutaDetalle(1, @desc);
SELECT @desc;

-- ■■ 9. REVOKE Y ELIMINACIÓN DE USUARIO
REVOKE ALL PRIVILEGES ON bd_gestion.* FROM 'control_transporte'@'localhost';
DROP USER 'control_transporte'@'localhost';

-- ■■ 10. ADICIONAL: TRIGGER DE AUDITORÍA DE SERVICIO
CREATE TABLE historial_servicio (
    id_historial INT AUTO_INCREMENT PRIMARY KEY,
    id_bus INT,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    estado_anterior BOOLEAN,
    estado_nuevo BOOLEAN
);

DELIMITER //
CREATE TRIGGER trg_actualizar_estado
BEFORE UPDATE ON buses
FOR EACH ROW
BEGIN
    IF OLD.en_servicio <> NEW.en_servicio THEN
        INSERT INTO historial_servicio (id_bus, estado_anterior, estado_nuevo)
        VALUES (OLD.id_bus, OLD.en_servicio, NEW.en_servicio);
    END IF;
END //
DELIMITER ;

-- Prueba del trigger:
UPDATE buses SET en_servicio = FALSE WHERE id_bus = 1;
SELECT * FROM historial_servicio;

```