

1. Inserción de datos:

```
insert into Departamento (nombre, presupuesto) values
('Tecnologia', 100000000),
('Contratacion', 50000000),
('Tecnologia', 150000000),
('Administrativo', 105000000),
('Salud', 500000000);
```

```
insert into Empleado (nombre, cargo, salario, id_departamento, fecha_ingreso) values
('Juan','Gerente',12250000 , 1 , '2024-10-30'),
('Carlos','Empleado',600000 , 2 , '2023-10-30'),
('Felipe','Gerente',12250000 , 3 , '2023-10-30'),
('Jaime','Subgerente',100000 , 4 , '2025-09-25'),
('Tomas','CEO',31000000 , 5 , '2020-10-30');
```

```
insert into Proyecto (nombre, fecha_inicio, presupuesto, id_departamento) values
('InnovaTech','2025-09-30', 25000000 , 1),
('FastContract','2024-09-30',25000000, 2),
('ManageProcess','2025-05-30',25000000, 3),
('Logistica PCT','2025-10-20',25000000, 4),
('FarmaGo','2025-08-30',25000000, 5);
```

```
insert into Asignacion (id_empleado, id_proyecto, horas_trabajadas) values
(1, 1, 30),
(2, 2, 60),
(3, 3, 75),
(4, 4, 10),
(5, 5, 95);
```

2. Desarrollo de Reto 6: Impedir que se inserte o actualice un empleado con salario inferior a 1.200.000

2.1. Trigger después de insertar un empleado con salario (menor a) < 1.200.000.

```
DELIMITER $$
CREATE TRIGGER empleado_salario
BEFORE INSERT ON Empleado
FOR EACH ROW
BEGIN
    IF NEW.salario < 1200000 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El salario no puede ser inferior a 1.200.000';
    END IF;
END$$
DELIMITER ;
```

- Empleado con salario de 600.000, resultado esperado: “El salario no puede ser inferior a ...”

```
insert into Empleado (nombre, cargo, salario, id_departamento, fecha_ingreso) values
('Cesar','Pasante',600000 , 5 , '2020-10-30');
```

Resultado:

✖ 127 08:33:02 insert into Empleado (nombre, cargo, salario, id_departamento, fecha_ingreso) values ('Cesar','Pasante',600000 , 5 , '2...

Error Code: 1644. El salario no puede ser inferior a 1.200.000

2.2. Después de insertar un empleado con salario (mayor a) > 1.200.000.

- Empleado con salario de 6.000.000, resultado esperado: “Correcto”

```
insert into Empleado (nombre, cargo, salario, id_departamento, fecha_ingreso) values
('Cesar','Pasante',6000000 , 5 , '2020-10-30');
```

Resultado:

✔ 128 08:37:47 insert into Empleado (nombre, cargo, salario, id_departamento, fecha_ingreso) values ('Cesar','Pasante',6000000 , 5 , '... 1 row(s) affected

2.3. Trigger antes de actualizar empleado con salario (menor a) < 1.200.000

```
-- Trigger: BEFORE UPDATE
DELIMITER $$
CREATE TRIGGER empleado_salario_update_check
BEFORE UPDATE ON Empleado
FOR EACH ROW
BEGIN
    IF NEW.salario < 1200000 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El salario no puede ser inferior a 1.200.000';
    END IF;
END$$
DELIMITER ;
```

2.4. Actualizar empleado con salario de 900.000 donde tengan cargo “Gerente”

```
update Empleado set salario = 900000 where cargo in('Gerente');
```

Resultado:

❌ 130 08:44:49 update Empleado set salario = 900000 where cargo in('Gerente')

Error Code: 1644. El salario no puede ser inferior a 1.200.000

3. Procedimiento: ActualizarSalariosMinimos()

```
DELIMITER $$
CREATE PROCEDURE ActualizarSalariosMinimos()
BEGIN
    UPDATE Empleado
    SET salario = 1200000
    WHERE salario < 1200000;
END$$
DELIMITER ;
```

Prueba: `call ActualizarSalariosMinimos();`

Resultado:

08:49:48 call ActualizarSalariosMinimos() Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column To disable safe mode, toggle the option in Preferences -> SQL Editor and reconnect. 0.000 sec

Observación: Debido a los permisos de administrador fue imposible probar este proceso.

4. Función: EmpleadosPorDebajoSalarioMin()

```
DELIMITER $$
CREATE FUNCTION EmpleadosPorDebajoSalarioMin(salario_min DECIMAL(10,2))
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE cantidad INT;

    SELECT COUNT(*) INTO cantidad
    FROM Empleado
    WHERE salario < salario_min
    ORDER BY nombre;

    RETURN cantidad;
END$$
DELIMITER ;
```

Prueba: `SELECT EmpleadosPorDebajoSalarioMin(1200000);`

Resultado esperado: ('Carlos','Empleado',600000 , 2 , '2023-10-30') y ('Jaime','Subgerente',100000 , 4 , '2025-09-25') o sea dos (2) empleados por debajo.

Resultado:

	EmpleadosPorDebajoSalarioMin(1200000)
▶	2

5. Transacción: Revertir inserciones inválidas.

```
DELIMITER $$
CREATE PROCEDURE RevertirInsercionesInvalidas()
BEGIN
    START TRANSACTION;
    INSERT INTO Asignacion (id_empleado, id_proyecto, horas_trabajadas) VALUES
    (1, 1, 30),
    (2, 2, 60),
    (3, 3, 75),
    (4, 4, 10),
    (5, 5, -5); -- horas_trabajadas < 0 no son aceptadas

    IF EXISTS (
        SELECT 1
        FROM Asignacion
        WHERE horas_trabajadas < 0) THEN
        ROLLBACK;
        SELECT 'Transacción invalida: se encontraron inserciones erroneas.' AS Resultado;
    ELSE
        COMMIT;
        SELECT 'Transacción completada exitosamente.' AS Resultado;
    END IF;
END$$
DELIMITER ;
```

Resultado: Arroja un error puesto que existe una restricción CHECK que impide que se inicie la transacción.

08:59:45 CALL RevertirInsercionesInvalidas() Error Code: 4025. CONSTRAINT
`asignacion.horas_trabajadas` failed for `parcial`.`asignacion` 0.016 sec

Solución propuesta: Hacer un alter table y quitar esa restricción para así correr la transacción y que arroje el resultado esperado a la inserción (5, 5, -5) = invalido.

```
155 • describe Asignacion;
156 • alter table Asignacion
157     modify horas_trabajadas INT;
158 • CALL RevertirInsercionesInvalidas();
159
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

Resultado
Transacción invalida: se encontraron inserciones erroneas.