

# Entrega 2 - Torres

## 1. Introducción

Este documento corresponde a la segunda entrega del proyecto final de base de datos. Se amplía el contenido de la entrega 1 incluyendo el desarrollo de vistas, funciones personalizadas, procedimientos almacenados (Stored Procedures), triggers, e inserciones de datos que permiten ejecutar y probar correctamente la estructura de la base de datos.

El sistema continúa enfocado en la gestión del área de Recursos Humanos para una empresa argentina ficticia.

---

## 2. Vistas

A continuación se detallan las vistas implementadas, su objetivo y las tablas que intervienen:

### - vista\_empleados\_por\_departamento

- **Objetivo:** Listar todos los empleados junto a su departamento y puesto.
- **Tablas involucradas:** Empleado, Departamento, Puesto.

sql

CopiarEditar

```
CREATE VIEW vista_empleados_por_departamento AS
SELECT e.legajo, e.nombre, e.apellido, d.nombre AS departamento,
p.nombre AS puesto
FROM Empleado e
JOIN Departamento d ON e.id_dpto = d.id_dpto
JOIN Puesto p ON e.id_puesto = p.id_puesto;
```

---

### - vista\_asistencias\_totales

- **Objetivo:** Mostrar la cantidad total de asistencias registradas por empleado.
- **Tablas involucradas:** Empleado, Asistencia.

sql

CopiarEditar

```
CREATE VIEW vista_asistencias_totales AS
SELECT e.legajo, e.nombre, e.apellido, COUNT(a.id_asistencia) AS
total_asistencias
FROM Empleado e
LEFT JOIN Asistencia a ON e.legajo = a.legajo
GROUP BY e.legajo, e.nombre, e.apellido;
```

---

### - vista\_empleados\_con\_obra\_social

- **Objetivo:** Mostrar los empleados y su respectiva obra social.
- **Tablas involucradas:** Empleado, ObraSocial.

sql

CopiarEditar

```
CREATE VIEW vista_empleados_con_obra_social AS
SELECT e.legajo, e.nombre, e.apellido, o.nombre AS obra_social
FROM Empleado e
JOIN ObraSocial o ON e.id_obra = o.id_obra;
```

---

## 3. Funciones Personalizadas

### - fn\_calcular\_edad

- **Objetivo:** Calcular la edad actual de una persona dada su fecha de nacimiento.
- **Tabla utilizada:** Empleado (aplicable a cualquier fecha en realidad).

sql

CopiarEditar

```
CREATE FUNCTION fn_calcular_edad(fecha_nac DATE)
RETURNS INT
DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, fecha_nac, CURDATE());
END;
```

---

### - fn\_total\_asistencias

- **Objetivo:** Devolver la cantidad de asistencias registradas para un empleado.
- **Tabla utilizada:** Asistencia.

sql

CopiarEditar

```
CREATE FUNCTION fn_total_asistencias(empleado_legajo INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total
    FROM Asistencia
    WHERE legajo = empleado_legajo;
    RETURN total;
END;
```

---

## 4. Stored Procedures

### - sp\_registrar\_asistencia

- **Objetivo:** Registrar una asistencia para un empleado con su fecha y estado.
- **Tablas involucradas:** Asistencia.

sql

CopiarEditar

```
CREATE PROCEDURE sp_registrar_asistencia(
    IN p_legajo INT,
    IN p_fecha DATE,
    IN p_estado ENUM('Presente', 'Ausente', 'Licencia', 'Tardanza')
)
BEGIN
    INSERT INTO Asistencia (legajo, fecha, estado)
    VALUES (p_legajo, p_fecha, p_estado);
END;
```

---

### - **sp\_actualizar\_obra\_social**

- **Objetivo:** Actualizar la obra social de un empleado existente.
- **Tabla involucrada:** Empleado.

sql

CopiarEditar

```
CREATE PROCEDURE sp_actualizar_obra_social(
    IN p_legajo INT,
    IN p_nueva_obra INT
)
BEGIN
    UPDATE Empleado
    SET id_obra = p_nueva_obra
    WHERE legajo = p_legajo;
END;
```

---

## 5. Trigger

### - **trg\_log\_asistencia\_insert**

- **Objetivo:** Registrar automáticamente en una tabla auxiliar (LogAsistencia) cada vez que se agrega una nueva asistencia.
- **Tablas involucradas:** Asistencia, LogAsistencia (creada específicamente).

sql

CopiarEditar

```
CREATE TABLE IF NOT EXISTS LogAsistencia (
    id_log INT PRIMARY KEY AUTO_INCREMENT,
    legajo INT,
    fecha DATE,
    estado VARCHAR(20),
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TRIGGER trg_log_asistencia_insert
AFTER INSERT ON Asistencia
FOR EACH ROW
BEGIN
    INSERT INTO LogAsistencia (legajo, fecha, estado)
    VALUES (NEW.legajo, NEW.fecha, NEW.estado);
END;
```

---

## 6. Inserción de Datos

Para probar todas las funciones, vistas y procedimientos creados, se cargaron datos ficticios de ejemplo:

```
sql
CopiarEditar
-- Departamentos
INSERT INTO Departamento (nombre, descripcion) VALUES
('Recursos Humanos', 'Área encargada de la gestión del personal'),
('Sistemas', 'Área de soporte y desarrollo tecnológico'),
('Administración', 'Área de finanzas y contabilidad');

-- Obras Sociales
INSERT INTO ObraSocial (nombre, telefono, direccion) VALUES
('OSDE', '011-4321-1000', 'Av. Corrientes 1234, CABA'),
('Swiss Medical', '011-4321-2000', 'Av. Santa Fe 4567, CABA');

-- Puestos
INSERT INTO Puesto (nombre, descripcion, id_dpto) VALUES
('Analista de RRHH', 'Gestiona procesos de RRHH', 1),
('Programador', 'Desarrolla software', 2),
('Contador', 'Maneja la contabilidad', 3);

-- Empleados
INSERT INTO Empleado (nombre, apellido, dni, fecha_nac, edad,
lugar_nacimiento, id_puesto, id_dpto, id_obra, telefono, email)
VALUES
('Juan', 'Pérez', '30123456', '1988-05-10', 36, 'Buenos Aires', 1,
1, 1, '11-4567-8901', 'juan.perez@email.com'),
('María', 'González', '29876543', '1990-08-22', 33, 'Rosario', 2, 2,
2, '341-555-1234', 'maria.gonzalez@email.com');
```

```
('Carlos', 'Fernández', '31234567', '1985-03-15', 39, 'Córdoba', 3, 3, 1, '351-333-9876', 'carlos.fernandez@email.com');
```

```
-- Asistencias
```

```
INSERT INTO Asistencia (legajo, fecha, estado) VALUES
```

```
(1, '2025-07-14', 'Presente'),
```

```
(2, '2025-07-14', 'Ausente'),
```

```
(3, '2025-07-14', 'Presente');
```

```
-- Contactos de emergencia
```

```
INSERT INTO ContactoEmergencia (legajo, nombre, telefono, parentesco) VALUES
```

```
(1, 'Ana Pérez', '11-4567-0001', 'Hermana'),
```

```
(2, 'Luis González', '341-555-0002', 'Padre'),
```

```
(3, 'Sofía Fernández', '351-333-0003', 'Esposa');
```

---

## 7. Archivos adjuntos

Se entregan dos archivos `.sql`:

- `entrega2_objetos.sql`: contiene las vistas, funciones, stored procedures y triggers.
- `entrega2_datos.sql`: contiene inserciones para poblar la base de datos con registros de prueba.