```sql
-- 1. Evitar los "posts" de usuarios institucionales
CREATE OR REPLACE TRIGGER prevent_institutional_posts
BEFORE INSERT ON posts
FOR EACH ROW
DECLARE
    v_user_type CHAR(1);
BEGIN
    SELECT type INTO v_user_type FROM users WHERE user_id = :NEW.user_id;
    IF v_user_type = 'L' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Usuarios institucionales no
pueden hacer posts.');
    END IF;
END;
/


-- Pruebas
--Inserta una biblioteca, debe descartarlo
INSERT INTO municipalities (TOWN, PROVINCE, POPULATION)
VALUES ('Barcelona', 'Barcelona', 1000);

INSERT INTO routes (ROUTE_ID)
VALUES ('R001');

INSERT INTO drivers (PASSPORT, EMAIL, FULLNAME, BIRTHDATE, PHONE,
ADDRESS, CONT_START)
VALUES ('PASSPORT001', 'driver@example.com', 'Juan Pérez',
TO_DATE('1980-06-15', 'YYYY-MM-DD'), 123456789, 'Calle Falsa 456',
TO_DATE('2025-01-01', 'YYYY-MM-DD'));

INSERT INTO assign_drv (PASSPORT, TASKDATE, ROUTE_ID)
VALUES ('PASSPORT001', TO_DATE('2024-11-23', 'YYYY-MM-DD'), 'R001');

INSERT INTO stops (TOWN, PROVINCE, ADDRESS, ROUTE_ID, STOPTIME)
VALUES ('Barcelona', 'Barcelona', 'Calle Falsa 123', 'R001', 30);

INSERT INTO bibuses (PLATE, LAST_ITV, NEXT_ITV)
VALUES ('B001', TO_DATE('2024-01-01', 'YYYY-MM-DD'),
TO_DATE('2025-01-01', 'YYYY-MM-DD'));

INSERT INTO assign_bus (PLATE, TASKDATE, ROUTE_ID)
VALUES ('B001', TO_DATE('2024-11-23', 'YYYY-MM-DD'), 'R001');

INSERT INTO services (TOWN, PROVINCE, BUS, TASKDATE, PASSPORT)
VALUES ('Barcelona', 'Barcelona', 'B001', TO_DATE('2024-11-23',
'YYYY-MM-DD'), 'PASSPORT001');

INSERT INTO books (TITLE, AUTHOR, COUNTRY, LANGUAGE, PUB_DATE, ALT_TITLE,
TOPIC, CONTENT, AWARDS)
VALUES ('Libro de prueba', 'Autor de prueba', 'España', 'Español', 2024,
NULL, 'Ficción', 'Contenido de prueba', NULL);

INSERT INTO editions (ISBN, TITLE, AUTHOR, LANGUAGE, NATIONAL_LIB_ID)
VALUES ('ISBN123456789', 'Libro de prueba', 'Autor de prueba', 'Español',
'LIB001');

INSERT INTO copies (SIGNATURE, ISBN, CONDITION, COMMENTS, DEREGISTERED)
VALUES ('P002', 'ISBN123456789', 'N', 'Nuevo estado', NULL)

INSERT INTO users (USER_ID, ID_CARD, NAME, SURNAME1, SURNAME2, BIRTHDATE,
TOWN, PROVINCE, ADDRESS, EMAIL, PHONE, TYPE)
```

```sql
VALUES ('BIB001', 'BIB123456', 'Biblioteca', 'Central', NULL,
TO_DATE('2000-01-01', 'YYYY-MM-DD'), 'Barcelona', 'Barcelona', 'Avenida
de la Biblioteca 1', 'contacto@biblioteca.es', 934567890, 'L');

INSERT INTO loans (SIGNATURE, USER_ID, STOPDATE, TOWN, PROVINCE, TYPE,
TIME, RETURN)
VALUES ('P002', 'BIB001', TO_DATE('2024-11-23', 'YYYY-MM-DD'),
'Barcelona', 'Barcelona', 'L', 0, NULL);

INSERT INTO posts (SIGNATURE, USER_ID, STOPDATE, POST_DATE, TEXT, LIKES,
DISLIKES)
VALUES ('P002', 'BIB001', TO_DATE('2025-04-02', 'YYYY-MM-DD'),
TO_DATE('2025-04-03', 'YYYY-MM-DD'), 'Texto de prueba de usuario
institucional', 0, 0);




-- 2. Actualizar "fecha de baja" cuando una copia está deteriorada
CREATE OR REPLACE TRIGGER update_deregistration_date
BEFORE UPDATE ON copies
FOR EACH ROW
BEGIN
    IF :NEW.condition = 'D' AND :OLD.condition <> 'D' THEN
        SELECT SYSDATE INTO :NEW.deregistered FROM dual;
    END IF;
END;
/


INSERT INTO copies (SIGNATURE, ISBN, CONDITION, COMMENTS, DEREGISTERED)
VALUES ('C1234', '978-84-339-6114-3', 'G', 'Buen estado', NULL);


-- Verificar que la fecha de baja es NULL
SELECT * FROM copies WHERE SIGNATURE = 'C1234';


-- Actualizar la condición a 'D' (Deteriorado)
UPDATE copies SET CONDITION = 'D' WHERE SIGNATURE = 'C1234';


-- Verificar que la fecha de baja se ha actualizado
SELECT * FROM copies WHERE SIGNATURE = 'C1234';




-- 3. Crear tablas históricas para usuarios y préstamos
CREATE TABLE historic_users AS SELECT * FROM users WHERE 1=0;
CREATE TABLE historic_loans AS SELECT * FROM loans WHERE 1=0;


CREATE OR REPLACE TRIGGER archive_user_data
BEFORE DELETE ON users
FOR EACH ROW
BEGIN
    -- Insertar el usuario en la tabla de históricos
    INSERT INTO historic_users
```

```sql
    (USER_ID, ID_CARD, NAME, SURNAME1, SURNAME2, BIRTHDATE, TOWN,
PROVINCE, ADDRESS, EMAIL, PHONE, TYPE, BAN_UP2)
    VALUES
    (:OLD.USER_ID, :OLD.ID_CARD, :OLD.NAME, :OLD.SURNAME1, :OLD.SURNAME2,
:OLD.BIRTHDATE, :OLD.TOWN, :OLD.PROVINCE,
     :OLD.ADDRESS, :OLD.EMAIL, :OLD.PHONE, :OLD.TYPE, :OLD.BAN_UP2);


    -- Insertar los préstamos del usuario en la tabla de históricos
    INSERT INTO historic_loans
    SELECT * FROM loans WHERE user_id = :OLD.user_id;


    -- Eliminar los préstamos del usuario en la tabla original
    DELETE FROM loans WHERE user_id = :OLD.user_id;
END;
/



-- Insertar un municipio
INSERT INTO municipalities (TOWN, PROVINCE, POPULATION)
VALUES ('Madrid', 'Madrid', 1000);


-- Insertar una ruta
INSERT INTO routes (ROUTE_ID)
VALUES ('R001');


-- Insertar un conductor
INSERT INTO drivers (PASSPORT, EMAIL, FULLNAME, BIRTHDATE, PHONE,
ADDRESS, CONT_START)
VALUES ('PASSPORT001', 'driver@example.com', 'Juan Pérez',
TO_DATE('1980-06-15', 'YYYY-MM-DD'), '912345678', 'Calle Gran Vía 123',
TO_DATE('2025-01-01', 'YYYY-MM-DD'));


-- Asignar el conductor a la ruta
INSERT INTO assign_drv (PASSPORT, TASKDATE, ROUTE_ID)
VALUES ('PASSPORT001', TO_DATE('2024-11-23', 'YYYY-MM-DD'), 'R001');


-- Insertar una parada en la ruta
INSERT INTO stops (TOWN, PROVINCE, ADDRESS, ROUTE_ID, STOPTIME)
VALUES ('Madrid', 'Madrid', 'Calle Gran Vía 456', 'R001', 15);


-- Insertar un bibliobús
INSERT INTO bibuses (PLATE, LAST_ITV, NEXT_ITV)
VALUES ('B001', TO_DATE('2024-01-01', 'YYYY-MM-DD'),
TO_DATE('2025-01-01', 'YYYY-MM-DD'));


-- Asignar el bibliobús a la ruta
INSERT INTO assign_bus (PLATE, TASKDATE, ROUTE_ID)
VALUES ('B001', TO_DATE('2024-11-23', 'YYYY-MM-DD'), 'R001');


-- Insertar un servicio en la ruta
INSERT INTO services (TOWN, PROVINCE, BUS, TASKDATE, PASSPORT)
```

```sql
VALUES ('Madrid', 'Madrid', 'B001', TO_DATE('2024-11-23', 'YYYY-MM-DD'),
'PASSPORT001');


-- Insertar un libro
INSERT INTO books (TITLE, AUTHOR, COUNTRY, LANGUAGE, PUB_DATE, ALT_TITLE,
TOPIC, CONTENT, AWARDS)
VALUES ('Historia de Madrid', 'Autor Desconocido', 'España', 'Español',
2020, NULL, 'Historia', 'Contenido de prueba', NULL);


-- Insertar una edición del libro
INSERT INTO editions (ISBN, TITLE, AUTHOR, LANGUAGE, NATIONAL_LIB_ID)
VALUES ('ISBN987654321', 'Historia de Madrid', 'Autor Desconocido',
'Español', 'LIB001');


-- Insertar un ejemplar del libro
INSERT INTO copies (SIGNATURE, ISBN, CONDITION, COMMENTS, DEREGISTERED)
VALUES ('S001', 'ISBN987654321', 'N', 'En buen estado', NULL);


-- Insertar un usuario institucional (Biblioteca)
INSERT INTO users (USER_ID, ID_CARD, NAME, SURNAME1, BIRTHDATE, TOWN,
PROVINCE, ADDRESS, EMAIL, PHONE, TYPE)
VALUES ('USER001', '12345678X', 'Biblioteca', 'Central',
TO_DATE('1980-01-01', 'YYYY-MM-DD'), 'Madrid', 'Madrid', 'Calle Mayor 1',
'contacto@biblioteca.es', '912345678', 'L');


-- Insertar un préstamo para la biblioteca
INSERT INTO loans (SIGNATURE, USER_ID, STOPDATE, TOWN, PROVINCE, TYPE,
TIME, RETURN)
VALUES ('S001', 'USER001', TO_DATE('2024-11-23', 'YYYY-MM-DD'), 'Madrid',
'Madrid', 'L', 0, NULL);




DELETE FROM users WHERE USER_ID = 'USER001';
SELECT * FROM historic_users WHERE USER_ID = 'USER001';
SELECT * FROM historic_loans WHERE USER_ID = 'USER001';
```