

```

-- 1. vista my_data
-- Crear el paquete foundicu
CREATE OR REPLACE PACKAGE foundicu AS
    current_user CHAR(10);
    PROCEDURE SetCurrentUser(p_user CHAR);
    FUNCTION GetCurrentUser RETURN CHAR;
END foundicu;
/
-- Crear el cuerpo del paquete foundicu
CREATE OR REPLACE PACKAGE BODY foundicu AS
    PROCEDURE SetCurrentUser(p_user CHAR) IS
    BEGIN
        current_user := p_user; -- Asignar el usuario actual
    END SetCurrentUser;

    FUNCTION GetCurrentUser RETURN CHAR IS
    BEGIN
        RETURN current_user; -- Devolver el usuario actual
    END GetCurrentUser;
END foundicu;
/
-- Crear la vista my_data (solo lectura)
CREATE OR REPLACE VIEW my_data AS
SELECT user_id,
       name || ' ' || surname1 || ' ' || NVL(surname2, '') AS full_name,
       address,
       email,
       phone
FROM users
WHERE user_id = foundicu.GetCurrentUser()
WITH READ ONLY;

--Pruebas
BEGIN
    foundicu.SetCurrentUser('0010126950'); -- Aquí pones el USER_ID del
usuario que desees probar
END;
/
SELECT * FROM my_data;

-- 2. Vista my_loans (operable para actualizar post)
-- Crear la vista my_loans
CREATE OR REPLACE VIEW my_loans AS
SELECT
    l.signature,
    l.user_id,
    l.stopdate,
    l.return,
    p.text AS post,
    p.post_date,
    p.likes,
    p.dislikes
FROM loans l
LEFT JOIN posts p ON l.signature = p.signature
WHERE l.user_id = foundicu.GetCurrentUser()
WITH CHECK OPTION;
-- Crear el trigger trg_update_my_loans
CREATE OR REPLACE TRIGGER trg_update_my_loans

```

```

INSTEAD OF UPDATE ON my_loans
FOR EACH ROW
BEGIN
    -- Si el nuevo post no es nulo y el antiguo es nulo
    IF :NEW.post IS NOT NULL AND :OLD.post IS NULL THEN
        -- Insertar el nuevo post con los valores predeterminados
        INSERT INTO posts (signature, user_id, stopdate, post_date, text,
likes, dislikes)
            VALUES (:OLD.signature, foundicu.GetCurrentUser(), :OLD.stopdate,
SYSDATE, :NEW.post, 0, 0);

        -- Si el nuevo post no es nulo y ya existía un post
    ELSIF :NEW.post IS NOT NULL THEN
        -- Actualizar el texto del post y la fecha
        UPDATE posts
        SET text = :NEW.post, post_date = SYSDATE
        WHERE signature = :OLD.signature;

        -- Si el nuevo post es nulo, no se actualiza el post
    ELSE
        RAISE_APPLICATION_ERROR(-20001, 'Solo se permite actualizar el
campo post.');
```

END IF;

```

        -- No se permite la actualización de ninguna otra columna, como
        'signature', 'user_id', 'stopdate', etc.
        IF :NEW.signature != :OLD.signature OR :NEW.user_id != :OLD.user_id
OR :NEW.stopdate != :OLD.stopdate THEN
            RAISE_APPLICATION_ERROR(-20002, 'No se puede actualizar este
campo.');
```

END IF;

```

END;
/

--Pruebas
BEGIN
    -- Asignar el usuario actual
    foundicu.SetCurrentUser('0010126950');
```

END;

```

/

-- Realizar un UPDATE en la vista para agregar un post
UPDATE my_loans
SET post = 'Este es un comentario sobre el préstamo.'
WHERE signature = 'KJ006' AND user_id = foundicu.GetCurrentUser();

select * from my_loans;
```