OPERACIONES CREATE (INSERCIÓN)

1. Inserción en Oficinas:

INSERT INTO Oficinas (Nombre, Direccion, Telefono)

VALUES ('Oficina Centro', 'Calle Mayor 15', '964123456')

RETURNING ID;

En esta consulta:

- No especificamos el ID porque es SERIAL (autoincremental)

- RETURNING ID nos devuelve el ID generado automáticamente

- Esta inserción debe realizarse primero porque los inmuebles dependen de las oficinas

2. Inserción en Inmuebles:

INSERT INTO Inmuebles (

Tipo, Superficie, Direccion, Zona,

PrecioVenta, PrecioAlquiler, Propietario,

TelefonoPropietario, OficinaID, TieneLlaves

) VALUES (

'Piso', 90, 'C/Linda 24, zona Capuchinos', 'Capuchinos',

14000000, NULL, 'Luis Herranz',

'964223344', 1, true

) RETURNING Ref;

Aspectos importantes:

- NULL en PrecioAlquiler indica que no está disponible para alquiler

- OficinaID debe corresponder a una oficina existente (integridad referencial)

- RETURNING Ref nos da la referencia generada que necesitaremos para las características

- El orden de los campos debe coincidir con el orden de los valores

3. Inserción en CaracteristicasVivienda:

INSERT INTO CaracteristicasVivienda (

Ref, NumHabitaciones, NumBanos, NumAseos,

NumCocinas, Altura, TieneGasCiudad,

TienePuertaBlindada, TieneParquet, TieneCalefaccion

) VALUES (

1234, 3, 1, 0, 1, 2, true, true, true, false

);

Detalles importantes:

- Ref debe ser una referencia válida de un inmueble existente

- Los campos booleanos (tiene\_\*) aceptan true/false

- Esta tabla solo se usa para pisos, casas y villas

OPERACIONES READ (CONSULTA)

1. Consulta básica con JOIN:

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID;

Explicación:

- LEFT JOIN mantiene todos los inmuebles aunque no tengan oficina asignada

- El alias 'i' y 'o' simplifican la escritura

- i.\* selecciona todos los campos de inmuebles

- as NombreOficina renombra la columna para claridad

2. Consulta condicional:

SELECT \* FROM Inmuebles

WHERE LOWER(Zona) LIKE LOWER('%capuchinos%');

Desglose:

- LOWER() convierte a minúsculas para búsqueda insensible a mayúsculas

- LIKE con % permite búsqueda parcial (ejemplo: "zona capuchinos norte")

- El % antes y después busca la palabra en cualquier posición

3. Consulta compleja con CASE:

SELECT i.\*,

CASE WHEN i.Tipo != 'Local' THEN

(SELECT json\_build\_object(

'habitaciones', cv.NumHabitaciones,

'baños', cv.NumBanos,

'aseos', cv.NumAseos,

'gas\_ciudad', cv.TieneGasCiudad

) FROM CaracteristicasVivienda cv

WHERE cv.Ref = i.Ref)

ELSE

(SELECT json\_build\_object(

'diafano', cl.Diafano,

'puertas', cl.NumPuertas,

'aire', cl.TieneAireAcondicionado

) FROM CaracteristicasLocal cl

WHERE cl.Ref = i.Ref)

END as caracteristicas

Análisis detallado:

- CASE WHEN determina el tipo de inmueble

- json\_build\_object crea un objeto JSON con las características

- Las subconsultas obtienen las características específicas

- as caracteristicas nombra la columna resultante

4. Consulta con agregación:

SELECT Zona,

COUNT(\*) as total\_inmuebles,

AVG(PrecioVenta) as precio\_medio\_venta,

AVG(PrecioAlquiler) as precio\_medio\_alquiler

FROM Inmuebles

GROUP BY Zona;

Funciones utilizadas:

- COUNT(\*) cuenta el número de registros

- AVG() calcula la media

- GROUP BY agrupa los resultados por zona

OPERACIONES UPDATE (ACTUALIZACIÓN)

1. Actualización simple:

UPDATE Inmuebles

SET PrecioVenta = 150000, PrecioAlquiler = 800

WHERE Ref = 1234;

Componentes:

- SET especifica los campos a modificar

- WHERE limita qué registros se actualizan

- Sin WHERE se actualizarían todos los registros

2. Actualización con condiciones:

UPDATE CaracteristicasVivienda

SET TieneGasCiudad = true, TieneCalefaccion = true

WHERE Ref = 1234;

Consideraciones:

- Solo afecta a un registro específico

- Múltiples campos pueden actualizarse a la vez

- Los valores booleanos se expresan como true/false

OPERACIONES DELETE (ELIMINACIÓN)

1. Eliminación simple:

DELETE FROM Inmuebles WHERE Ref = 1234;

Efectos:

- ON DELETE CASCADE elimina automáticamente registros relacionados

- La eliminación es permanente

- WHERE es crucial para evitar eliminaciones no deseadas

AGREGAR INMUEBLE (equivalente a agregarInmueble())

Para agregar un inmueble necesitamos hacer las inserciones en orden debido a las relaciones entre tablas:

-- 1. Primero insertamos el inmueble básico

INSERT INTO Inmuebles (

Tipo, Superficie, Direccion, Zona,

PrecioVenta, PrecioAlquiler, Propietario,

TelefonoPropietario, OficinaID, TieneLlaves

) VALUES (

'Piso', -- Tipo de inmueble

90, -- Superficie en m2

'C/Linda 24', -- Dirección

'Capuchinos', -- Zona

140000, -- Precio venta

NULL, -- Sin precio alquiler

'Luis Herranz', -- Propietario

'964223344', -- Teléfono

1, -- ID de la oficina

true -- Tiene llaves

) RETURNING Ref; -- Obtenemos la referencia generada

-- 2. Luego agregamos sus características específicas

-- Si es una vivienda (Piso, Casa o Villa):

INSERT INTO CaracteristicasVivienda (

Ref, NumHabitaciones, NumBanos, NumAseos,

NumCocinas, TieneGasCiudad, TienePuertaBlindada,

TieneParquet, TieneCalefaccion

) VALUES (

1234, -- Usar la referencia obtenida antes

3, -- Habitaciones

1, -- Baños

1, -- Aseos

1, -- Cocinas

true, -- Gas ciudad

true, -- Puerta blindada

false, -- Sin parquet

true -- Con calefacción

);

-- Si es un local:

INSERT INTO CaracteristicasLocal (

Ref, Diafano, NumPuertas,

TieneAireAcondicionado, SuperficieAltillo

) VALUES (

5678, -- Referencia del local

true, -- Es diáfano

2, -- Número de puertas

true, -- Tiene aire acondicionado

10.5 -- Metros cuadrados de altillo

);

MOSTRAR INMUEBLES (equivalente a mostrarInmuebles())

-- Consulta base para mostrar todos los inmuebles con sus datos

SELECT

i.Ref,

i.Tipo,

i.Superficie,

i.Direccion,

i.Zona,

i.PrecioVenta,

i.PrecioAlquiler,

i.Propietario,

i.TelefonoPropietario,

i.TieneLlaves,

o.Nombre as NombreOficina,

-- Características según el tipo

CASE WHEN i.Tipo != 'Local' THEN

(SELECT row\_to\_json(cv) FROM CaracteristicasVivienda cv WHERE cv.Ref = i.Ref)

ELSE

(SELECT row\_to\_json(cl) FROM CaracteristicasLocal cl WHERE cl.Ref = i.Ref)

END as Caracteristicas

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

ORDER BY i.Ref;

BUSCAR INMUEBLES (equivalente a buscarInmuebles())

-- 1. Búsqueda por zona

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

WHERE LOWER(i.Zona) LIKE LOWER('%capuchinos%');

-- 2. Búsqueda por tipo

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

WHERE i.Tipo = 'Piso';

-- 3. Búsqueda por rango de precio de venta

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

WHERE i.PrecioVenta BETWEEN 100000 AND 200000;

-- 4. Búsqueda por oficina

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

WHERE i.OficinaID = 1;

-- 5. Búsqueda por características de vivienda

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

LEFT JOIN CaracteristicasVivienda cv ON cv.Ref = i.Ref

WHERE cv.NumHabitaciones >= 3;

-- 6. Búsqueda por características de local

SELECT i.\*, o.Nombre as NombreOficina

FROM Inmuebles i

LEFT JOIN Oficinas o ON i.OficinaID = o.ID

LEFT JOIN CaracteristicasLocal cl ON cl.Ref = i.Ref

WHERE cl.Diafano = true;

ELIMINAR INMUEBLE (equivalente a eliminarInmueble())

-- Gracias al ON DELETE CASCADE, solo necesitamos eliminar el inmueble

-- y las características y visitas se eliminarán automáticamente

DELETE FROM Inmuebles WHERE Ref = 1234;

CONSULTAS ADICIONALES ÚTILES (basadas en el enunciado original):

-- Ver inmuebles disponibles tanto para venta como alquiler

SELECT \*

FROM Inmuebles

WHERE PrecioVenta IS NOT NULL

AND PrecioAlquiler IS NOT NULL;

-- Listar inmuebles por zona con sus visitas

SELECT i.Ref, i.Tipo, i.Direccion, i.Zona,

COUNT(v.ID) as NumeroVisitas

FROM Inmuebles i

LEFT JOIN Visitas v ON v.RefInmueble = i.Ref

GROUP BY i.Ref, i.Tipo, i.Direccion, i.Zona

ORDER BY i.Zona, NumeroVisitas DESC;

-- Buscar inmuebles similares en la misma zona

SELECT i2.\*

FROM Inmuebles i1, Inmuebles i2

WHERE i1.Ref = 1234 -- Referencia del inmueble base

AND i1.Zona = i2.Zona -- Misma zona

AND i1.Tipo = i2.Tipo -- Mismo tipo

AND i2.Ref != i1.Ref -- Excluir el mismo inmueble

AND ABS(i2.Superficie - i1.Superficie) < 20; -- Superficie similar

-- Ver historial de visitas de un inmueble

SELECT v.FechaHora, v.Cliente, v.Comentario

FROM Visitas v

WHERE v.RefInmueble = 1234

ORDER BY v.FechaHora DESC;

-- Resumen de inmuebles por oficina

SELECT o.Nombre as Oficina,

COUNT(\*) as TotalInmuebles,

SUM(CASE WHEN i.PrecioVenta IS NOT NULL THEN 1 ELSE 0 END) as EnVenta,

SUM(CASE WHEN i.PrecioAlquiler IS NOT NULL THEN 1 ELSE 0 END) as EnAlquiler

FROM Oficinas o

LEFT JOIN Inmuebles i ON i.OficinaID = o.ID

GROUP BY o.ID, o.Nombre;