

**Enunciado:**

1. Considere el modelo relacional de la Figura 1.

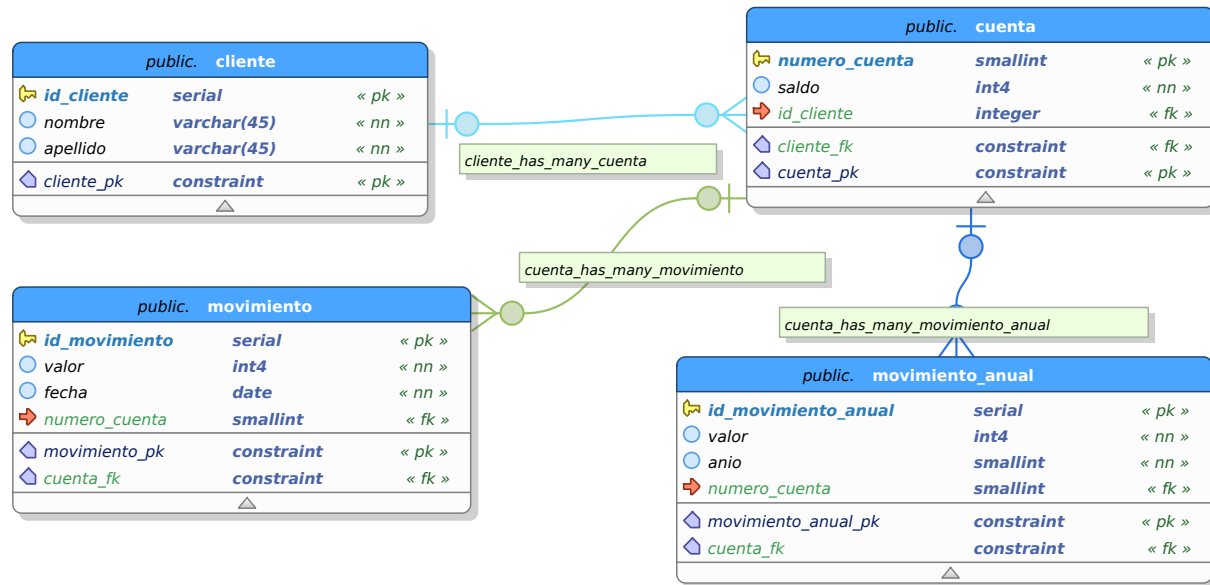


Figura 1: Modelo relacional.

Este modelo tiene las siguientes sentencias DDL:

```

1 CREATE TABLE cliente(
2     id_cliente SERIAL NOT NULL,
3     nombre VARCHAR(45) NOT NULL,
4     apellido VARCHAR(45) NOT NULL,
5     PRIMARY KEY (id_cliente)
6 );
7
8 CREATE TABLE cuenta(
9     numero_cuenta SMALLINT NOT NULL,
10    saldo INT4 NOT NULL,
11    id_cliente INTEGER,
12    PRIMARY KEY (numero_cuenta),
13    FOREIGN KEY (id_cliente) REFERENCES cliente (
14        id_cliente)
15 );
16
17 CREATE TABLE movimiento(
18     id_movimiento SERIAL NOT NULL,
19     valor INT4 NOT NULL,
20     fecha DATE NOT NULL,
21     numero_cuenta SMALLINT,
22     PRIMARY KEY (id_movimiento),
23     FOREIGN KEY (numero_cuenta) REFERENCES cuenta (
24         numero_cuenta)
25 );

```

```
23 );  
24  
25 CREATE TABLE movimiento_anual(  
26     id_movimiento_anual SERIAL NOT NULL,  
27     valor INT4 NOT NULL,  
28     anio SMALLINT NOT NULL,  
29     numero_cuenta SMALLINT,  
30     PRIMARY KEY (id_movimiento_anual),  
31     FOREIGN KEY (numero_cuenta) REFERENCES cuenta (  
        numero_cuenta)  
32 );
```

La base de datos cuenta con los datos de las siguientes sentencias:

```
1 INSERT INTO cliente (nombre, apellido)  
2     VALUES ('Pedro', 'Picapiedra');  
3 INSERT INTO cliente (nombre, apellido)  
4     VALUES ('Pablo', 'Marmol');  
5 INSERT INTO cliente (nombre, apellido)  
6     VALUES ('Vilma', 'Picapiedra');  
7 INSERT INTO cuenta (numero_cuenta, saldo, id_cliente)  
8     VALUES (1000, 1100000, 1);  
9 INSERT INTO cuenta (numero_cuenta, saldo, id_cliente)  
10    VALUES (2000, 4000000, 2);  
11 INSERT INTO cuenta (numero_cuenta, saldo, id_cliente)  
12    VALUES (3000, 2500000, 3);
```

Note que en las 2 primeras sentencias, debido a que `id_cliente` es `serial`, este atributo se asigna automáticamente.

Resuelva lo siguiente:

- a) Desarrolle una aplicación en Python que permita hacer las siguientes transacciones:
  - Primera transacción
    - Retirar de la cuenta *1000* el valor *200000*
    - Registrar en la tabla `movimiento` el valor *200000*, fecha *2019-10-10*, cuenta *1000*
    - Depositar en la cuenta *2000* el valor *200000*
    - Registrar en la tabla `movimiento` el valor *200000*, fecha *2019-10-10*, cuenta *2000*
  - Segunda transacción
    - Retirar de la cuenta *1000* el valor *100000*
    - Registrar en la tabla `movimiento` el valor *100000*, fecha *2020-01-10*, cuenta *1000*
    - Depositar en la cuenta *3000* el valor *100000*

- Registrar en la tabla `movimiento` el valor *100000*, fecha *2020-01-10*, cuenta *3000*
  - Tercera transacción
    - Retirar de la cuenta *2000* el valor *500000*
    - Registrar en la tabla `movimiento` el valor *500000*, fecha *2020-03-10*, cuenta *2000*
    - Depositar en la cuenta *3000* el valor *500000*
    - Registrar en la tabla `movimiento` el valor *500000*, fecha *2020-03-10*, cuenta *3000*
  - Cuarta transacción
    - Retirar de la cuenta *3000* el valor *400000*
    - Registrar en la tabla `movimiento` el valor *400000*, fecha *2020-03-10*, cuenta *3000*
    - Depositar en la cuenta *4000* el valor *400000*
    - Registrar en la tabla `movimiento` el valor *400000*, fecha *2020-03-10*, cuenta *4000*
- b) Ofrezca una conclusión de cada transacción ejecutada
2. Con base en la base de datos del ejercicio 1, desarrolle una función en `plpgsql` que permite calcular por cada cuenta y por cada año, el valor (en dinero) de los movimientos realizados.
  3. Con base en la base de datos del ejercicio 1, desarrolle un disparador que genere los mismos resultados del ejercicio 2.
  4. Considere el modelo relacional de la Figura 4.
    - a) Escriba el código DDL que genera las tablas, relaciones y restricciones de integridad que se observan en la Figura 4. Para modelar las restricciones de integridad necesarias tenga en cuenta la siguiente descripción del modelo:
      - El modelo relacional permite a una firma inmobiliaria almacenar la información relacionada con las propiedades que se están alquilando; y los habitantes, que pueden ser quienes solicitan el alquiler (hosts) o sus invitados, ya sean familiares o pareja (guests).
      - La información requerida para registrar un habitante en el sistema es su nombre completo; su tipo de documento - que no debe ser diferente a CC (Cédula de ciudadanía), CE (Cédula de extranjería), PA (Pasaporte), RC (Registro civil), o TI (Tarjeta de identidad); el número de documento; y un correo electrónico (único por habitante). La información mencionada anteriormente es obligatoria.
      - Adicionalmente, por cada habitante se debe llevar el registro del host con el que vive. Esta última restricción es opcional para los habitantes que son hosts, es decir, quienes figuran en el alquiler de cada propiedad.

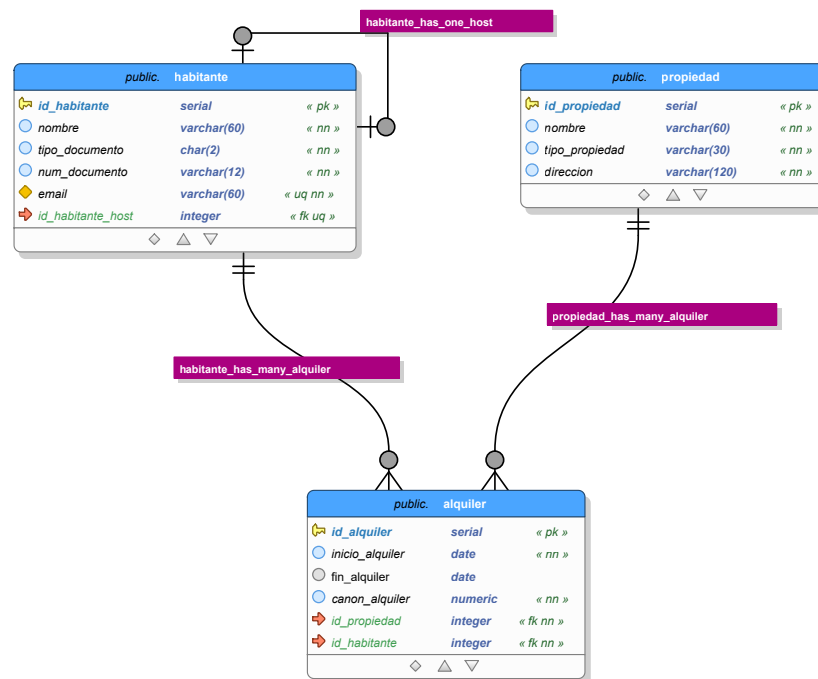


Figura 4: Modelo relacional de alquiler de vivienda.

- Para registrar una propiedad, es obligatorio tener un nombre para registrarla; el tipo de propiedad, que debe ser **Casa** o **Apartamento**; y la dirección.
  - Para registrar un alquiler es necesario contar con la fecha de inicio del alquiler, su valor por defecto debe ser la fecha actual, la cual puede ser asignada con la función `now() : DATE`. La fecha de finalización es opcional. El canon de alquiler es un valor que no puede ser negativo. Cada registro de alquiler debe tener una propiedad y un habitante registrados previamente en el sistema.
- b) Inserte una nueva tupla en la tabla `propiedad` usando como `id_propiedad` el valor 54
  - c) Inserte nuevas tuplas en la tabla `propiedad` usando sentencias `INSERT` en las que se haga uso del código auto-incremental del `id_propiedad`
  - d) Inserte un habitante que tenga valor `NULL` en `id_habitante_host`, luego inserte uno donde su `id_habitante_host` sea el `id_habitante` del primero, y finalmente inserte uno donde su `id_habitante_host` sea un valor cualquiera que no exista como ID en la tabla `habitante`. Concluya
  - e) Inserte un registro de alquiler que relacione a un habitante con una propiedad, haga uso del auto-incremental de `id_alquiler`, y de la fecha `DEFAULT` en el atributo `fin_alquiler`
  - f) Inserte un registro de alquiler que relacione a un habitante con una propiedad, estableciendo un `canon_alquiler` de -500000. Concluya

- g) El modelo relacional está incompleto. Para completarlo se requiere que exista un atributo nuevo en la tabla **propiedad** llamado foto, que se usará para guardar una foto de cada propiedad. Modifique el DDL de la tabla propiedad para incluir la foto de cada propiedad.
5. Se planea contratar un auditor de alquileres de los inmuebles. Para esto se requiere crear un perfil especial con las siguientes características:
- El perfil tendrá únicamente permisos de lectura sobre la tabla **habitante**.
  - El perfil podrá acceder a toda la información de alquileres, excepto por el atributo **canon\_alquiler** (es necesario generar una vista). Adicionalmente tendrá permisos de modificar información sobre la vista, pero no podrá insertar ni borrar información.
  - El perfil tendrá todos los permisos sobre la tabla **propiedad**

Escriba las sentencias DCL necesarias para crear el rol con las correspondientes autorizaciones.

6. Observe la relación **habitante** presentada en la Figura 4. En esta tabla se observa que cada habitante puede haber sido invitado a vivir en alguna propiedad por otro habitante. Si ese es el caso, en la tupla del habitante invitado debería aparecer el **id\_habitante** de la persona que lo invitó (específicamente en el campo **id\_habitante\_host**). Escriba una consulta SQL que permita obtener el id y el nombre de todos los habitantes invitados de forma directa o indirecta dado un **id\_habitante\_host** particular. Se adjuntan ejemplos de datos de la tabla **habitante** (Tabla 1) y de la salida esperada (Tabla 2).

habitante		
id_habitante	nombre	id_habitante_host
1	Miguel	NULL
2	Carlos	1
3	Jose	1
4	Juan	2
5	Pedro	2
6	Camila	3
7	Diana	3
8	Adriana	7
9	Dora	7
10	Laura	7

Tabla 1: Ejemplo de datos en tabla habitante.

habitante	
id_habitante	nombre
6	Camila
7	Diana
8	Adriana
9	Dora
10	Laura

Tabla 2: Ejemplo de salida esperada para `id_habitante_host = 3`.