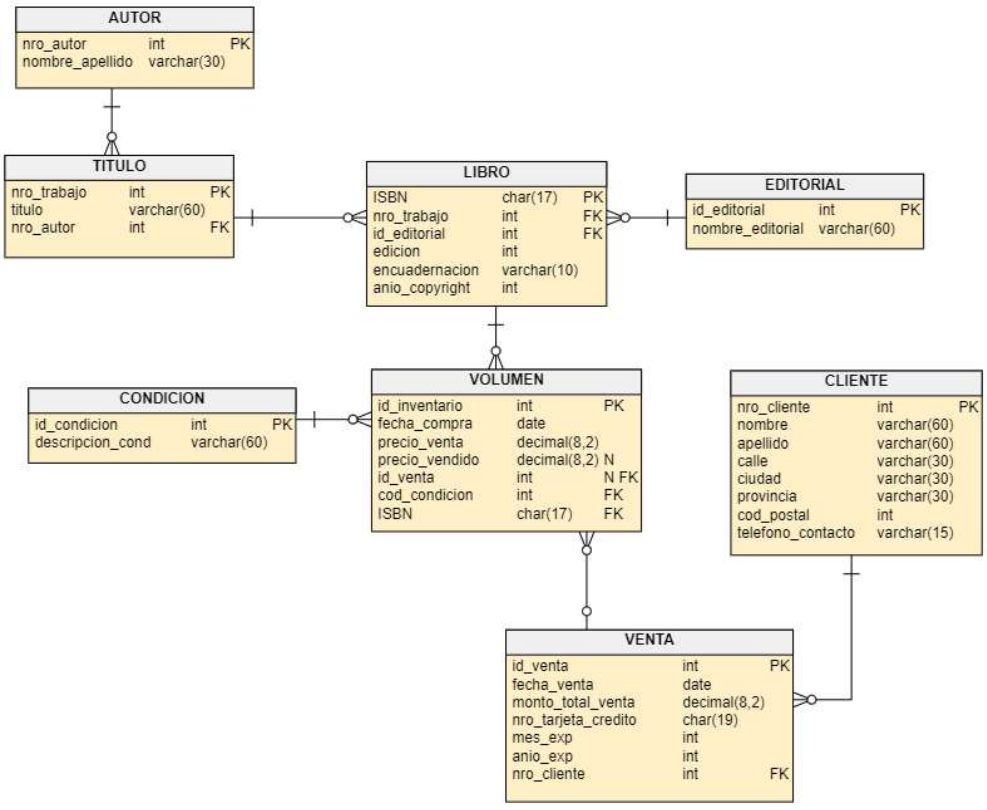
Comenzado el	jueves, 5 de diciembre de 2024, 09:30
Estado	Finalizado
Finalizado en	jueves, 5 de diciembre de 2024, 11:06
Tiempo empleado	1 hora 35 minutos

Información

El siguiente esquema de base de datos corresponde a un Sistema de librería en el que se guarda la información de libros, autores, editoriales, libros en stock (tabla volumen), clientes y ventas



El script para su creación se encuentra en el siguiente link

El script para la carga de los datos se encuentra en el siguiente link

Pregunta 1

Finalizado

Se puntúa como 0 sobre 1,00

Utilizando el esquema de las librerías dar la consulta SQL para obtener los clientes que han realizado compras de libros de al menos tres autores distintos. La consulta debe devolver el número de cliente, el nombre y apellido del cliente, y la cantidad de autores diferentes cuyos libros ha comprado. El resultado debe estar ordenado por número de cliente

SELECT c.nro_cliente, c.nombre,c.apellido, COUNT(DISTINCT t.nro_autor) AS cantidad_autores_distintos

FROM CLIENTE c

JOIN VENTA v ON c.nro_cliente = v.nro_cliente

JOIN VOLUMEN vol ON v.id_venta = vol.id_venta

JOIN LIBRO l ON vol.ISBN = l.ISBN

JOIN TITULO t ON l.nro_trabajo = t.nro_trabajo

GROUP BY c.nro_cliente, c.nombre, c.apellido

HAVING COUNT(DISTINCT t.nro_autor) >= 3

ORDER BY c.nro_cliente;

```
SELECT c.nro_cliente, c.nombre, c.apellido, COUNT(DISTINCT t.nro_autor) AS autores_diferentes
FROM   cliente c JOIN venta v ON c.nro_cliente = v.nro_cliente
JOIN volumen vol ON v.id_venta = vol.id_venta
JOIN libro l ON vol.ISBN = l.ISBN
JOIN titulo t ON l.nro_trabajo = t.nro_trabajo
GROUP BY c.nro_cliente, c.nombre, c.apellido
HAVING COUNT(DISTINCT t.nro_autor) > 2
ORDER BY 1;
```

Comentario:
Bien

Pregunta 2

Finalizado

Se puntúa como 0 sobre 1,00

Utilizando el esquema de las librerías:

a) Cree una vista denominada Vista_Libros_Cuero que contenga los datos solo de los libros encuadernados en cuero, incluya el ISBN, titulo, id_editorial, edicion, anio_copyright

b) ¿La vista creada anteriormente podría haber sido actualizable? Justifique

--A)

CREATE VIEW Vista_Libros_Cuero AS

SELECT l.ISBN, t.titulo,l.id_editorial, l.edicion, l.anio_copyright

FROM LIBRO l

JOIN TITULO t ON l.nro_trabajo = t.nro_trabajo

WHERE l.encuadernacion = 'cuero';

--B)

no es actualizable automáticamente debido al JOIN entre tablas. Pero podría hacerse actualizable implementando triggers específicos para manejar las modificaciones.

a)

```
CREATE VIEW Vista_Libros_Cuero AS
SELECT L.ISBN, T.titulo, L.id_editorial, L.edicion, L.anio_copyright
FROM LIBRO L
JOIN TITULO T ON(T.nro_trabajo = L.nro_trabajo)
WHERE UPPER(L.encuadernacion) like '%CUERO%';
```

b) La vista no puede ser actualizable porque son necesarias las 2 tablas en el FROM dado que se requiere listar el titulo de los libros que está en la tabla TITULO.

Comentario:
Bien

Pregunta 3

Finalizado

Se puntúa como 0 sobre 1,00

Incorpore los siguientes controles en SQL estándar mediante el recurso declarativo más restrictivo y utilizando sólo las tablas y atributos necesarios. Justifique el tipo de restricción definida en cada caso.

1. Para cada libro en stock (Tabla volumen) si el id de venta es distinto de nulo el precio vendido también debe ser distinto de nulo.
2. controlar que no haya más de 10 volúmenes nuevos (condición del libro igual a 1) para cada libro en stock (registros en la tabla VOLUMEN)

--1)

ALTER TABLE VOLUMEN

ADD CONSTRAINT chk_id_venta_precio_vendido

CHECK (

id_venta IS NULL OR precio_vendido IS NOT NULL

);

1-

```
ALTER TABLE VOLUMEN
ADD CONSTRAINT chk_venta_precio
CHECK((id_venta IS NULL AND precio_vendido IS NULL) OR
(id_venta IS NOT NULL AND  precio_vendido IS NOT NULL));
```

El tipo de restricción es de tupla

2)

```
ALTER TABLE VOLUMEN
ADD CONSTRAINT chk_max_volumen
CHECK(NOT EXISTS(
            SELECT 1
            FROM VOLUMEN
            WHERE cod_condicion = 1
            GROUP BY ISBN
            HAVING COUNT(*) > 10));
```

El tipo de restricción es de tabla

Comentario:

a) Mal

b) sin resolver

Pregunta 4

Finalizado

Se puntúa como 0 sobre 1,00

Escriba los triggers completos necesarios de aquellos chequeos que no pueden ser soportados por Postgresql de manera declarativa

--2)

CREATE OR REPLACE FUNCTION verificar_volumenes_nuevos()

RETURNS TRIGGER AS $$

BEGIN

IF (SELECT COUNT(*)

FROM VOLUMEN

WHERE ISBN = NEW.ISBN AND cod_condicion = 1) >= 10 THEN

RAISE EXCEPTION 'No se pueden agregar más de 10 volúmenes nuevos para el libro con ISBN %', NEW.ISBN;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_verificar_volumenes_nuevos

BEFORE INSERT OR UPDATE ON VOLUMEN

FOR EACH ROW

EXECUTE FUNCTION verificar_volumenes_nuevos();

```
create trigger tr_stock_libros_nuevos
before insert or update of cod_condicion, ISBN
on volumen
for each row
when (NEW.cod_condicion = 1 )
execute procedure fn_stock_libros_nuevos();


create or replace fn_stock_libros_nuevos()
returns Trigger as $$
declare
  cantidad integer;
begin
  select count(*) into cantidad
  from volumen
  where ISBN = NEW.ISBN
  and cod_condicion = 1;
  if (cantidad > 9) then
    raise exception 'Ya existen 10 ejemplares nuevos de este libro.';
  end if;
  return NEW;
end $$
language 'plpgsql';
```

Comentario:
Falta especificar

```
bupdate of cod_condicion, ISBN


y
```

```
when (NEW.cod_condicion = 1 )
```

Pregunta 5

Finalizado

Se puntúa como 0 sobre 1,00

Se desea crear una tabla denominada VENTAS_DIARIAS con el monto total vendido y la cantidad de libros vendidos diariamente.

```
CREATE TABLE VENTAS_DIARIAS (
   fecha date not null,
   monto_total_vendido numeric(20,2) not null,
   cantidad_libros int,
   CONSTRAINT PK_VENTAS_DIARIAS PRIMARY KEY (fecha));
```

1) De la sentencia de insert para llenar dicha tabla.

2) Implemente el/los triggres necesarios para mantener actualizada dicha tabla

--1)

INSERT INTO VENTAS_DIARIAS (fecha, monto_total_vendido, cantidad_libros)

SELECT v.fecha_venta AS fecha, SUM(vol.precio_vendido) AS monto_total_vendido,COUNT(vol.ISBN) AS cantidad_libros

FROM VENTA v

JOIN VOLUMEN vol ON v.id_venta = vol.id_venta

WHERE vol.precio_vendido IS NOT NULL

GROUP BY v.fecha_venta

ORDER BY v.fecha_venta;

--2)

```
CREATE OR REPLACE FUNCTION actualizar_ventas_diarias()
RETURNS TRIGGER AS $$
BEGIN
  IF EXISTS (SELECT 1 FROM VENTAS_DIARIAS WHERE fecha = NEW.fecha_venta) THEN
    UPDATE VENTAS_DIARIAS
    SET
      monto_total_vendido = (
        SELECT SUM(vol.precio_vendido)
        FROM VENTA v
        JOIN VOLUMEN vol ON v.id_venta = vol.id_venta
        WHERE v.fecha_venta = NEW.fecha_venta AND vol.precio_vendido IS NOT NULL
      ),
      cantidad_libros = (
        SELECT COUNT(vol.ISBN)
        FROM VENTA v
        JOIN VOLUMEN vol ON v.id_venta = vol.id_venta
        WHERE v.fecha_venta = NEW.fecha_venta
      )
    WHERE fecha = NEW.fecha_venta;
  ELSE
    INSERT INTO VENTAS_DIARIAS (fecha, monto_total_vendido, cantidad_libros)
    SELECT
      NEW.fecha_venta,
```

```
        SUM(vol.precio_vendido),
        COUNT(vol.ISBN)
    FROM VENTA v
    JOIN VOLUMEN vol ON v.id_venta = vol.id_venta
    WHERE v.fecha_venta = NEW.fecha_venta;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_actualizar_ventas_diarias_venta
AFTER INSERT OR UPDATE OR DELETE ON VENTA
FOR EACH ROW
EXECUTE FUNCTION actualizar_ventas_diarias();

CREATE TRIGGER trigger_actualizar_ventas_diarias_volumen
AFTER INSERT OR UPDATE OR DELETE ON VOLUMEN
FOR EACH ROW
EXECUTE FUNCTION actualizar_ventas_diarias();
```

```
INSERT INTO VENTAS_DIARIAS(fecha, monto_total_vendido, cantidad_libros)
SELECT V.fecha_venta, SUM(monto_total_venta), COUNT(*)
FROM VENTA V
JOIN VOLUMEN VO ON (VO.id_venta = V.id_venta)
GROUP BY V.fecha_venta;
```

```
CREATE TRIGGER tr_actualizar_ventas_diarias
BEFORE INSERT OR UPDATE OF fecha_venta, monto_total_venta OR DELETE
ON VENTA
FOR EACH ROW
EXECUTE FUNCTION fn_actualizar_ventas_diarias();
```

```
```

Comentario:
Regular la función si es delete no existe NEW

Pregunta 6

Finalizado

Se puntúa como 0 sobre 1,00

Para las siguientes preguntas seleccione la opción correcta y justifique en cada caso.

¿Qué sucede si una vista con WITH CHECK OPTION incluye otra vista con su propia cláusula WITH CHECK OPTION?

1. No se puede crear este tipo de vistas anidadas.
2. Solo se aplica la restricción de la primera vista.
3. Se aplican las restricciones de ambas vistas, garantizando que las operaciones cumplan con todos los criterios.
4. La cláusula WITH CHECK OPTION se ignora en vistas anidadas.

¿Cuándo es útil utilizar la acción ON DELETE SET DEFAULT en una FOREIGN KEY?

1. Cuando se quiere evitar eliminar registros en la tabla referenciante.
2. Cuando se desea que el valor de la clave foránea en la tabla referenciante cambie a un valor predeterminado al eliminarse el registro en la tabla referenciada.
3. Cuando se desea actualizar automáticamente los registros secundarios al cambiar los primarios.
4. Cuando se necesita restringir los cambios en la clave foránea.

¿Qué diferencia tiene un control declarativo sobre un trigger?

1. Permite mayor personalización de las condiciones de verificación.
2. El control declarativo verifica la carga previa de la base de datos en cambio el trigger no.
3. Permite ejecutar acciones complejas en otras tablas cuando se cumple una condición.
4. Permite manipular múltiples registros de forma dinámica al cumplir una condición.

¿Qué sucede si una vista con WITH CHECK OPTION incluye otra vista con su propia cláusula WITH CHECK OPTION?

La respuesta correcta es la 3 ya que Cada WITH CHECK OPTION asegura que las operaciones cumplan con las condiciones de todas las vistas involucradas, manteniendo la consistencia en la jerarquía.

¿Cuándo es útil utilizar la acción ON DELETE SET DEFAULT en una FOREIGN KEY?

La respuesta correcta es la 2 ya que Permite asignar un valor por defecto a la clave foránea en la tabla referenciante, evitando relaciones rotas tras la eliminación en la tabla referenciada.

¿Qué diferencia tiene un control declarativo sobre un trigger?

La respuesta correcta es la 1 ya que Los triggers permiten implementar lógica más compleja y personalizada, mientras que los controles declarativos son más rígidos pero más eficientes.

Comentario:
Mal la rta.

¿Qué diferencia tiene un control declarativo sobre un trigger?

La respuesta correcta es la 1 ya que Los triggers permiten implementar lógica más compleja y personalizada, mientras que los controles declarativos son más rígidos pero más eficientes.

Bien el resto

Actividad previa

◄ Avisos

Ir a...

Siguiente actividad

Parcial de Final Libre ►

Mantente en contacto

Facultad, Pabellón Central Paraje Arroyo Seco. Campus Universitario. (B7001BBO) Tandil. Buenos Aires, Argentina

https://exa.unicen.edu.ar/

(+54) (0249) 438-5650 Conmutador: int. 2000

moodle@exa.unicen.edu.ar

Descargar la app para dispositivos móviles