Primer Parcial

10 de septiembre de 2020

Indicaciones generales

- Este es un examen **individual** con una duración de **120 minutos: de 13:00 a 15:00**.
- En `e-aulas` puede acceder a las diapositivas, los enunciados de los talleres y a la sección correspondiente a este parcial.
- Celulares y otros dispositivos electrónicos deben estar apagados y ser guardados dentro de las maletas.
- El estudiante solo podrá disponer de hojas en blanco como borrador de apuntes (opcional).
- El estudiante puede tener hojas manuscritas de resumen (opcional). Estas hojas deben estar marcadas con nombre completo.
- La actividad en `e-aulas` se cerrará a la hora en punto acordada. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de `e-aulas` será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.
  Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.
- **Cualquier incumplimiento de lo anterior conlleva la anulación del examen.**
- Las respuestas deben estar totalmente justificadas.
- **Entrega**: Un solo archivo de texto con las respuestas a cada ejercicio, el cual debe enviar a `e-aulas`.
  Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.

Considere el modelo relacional de la Figura 1.

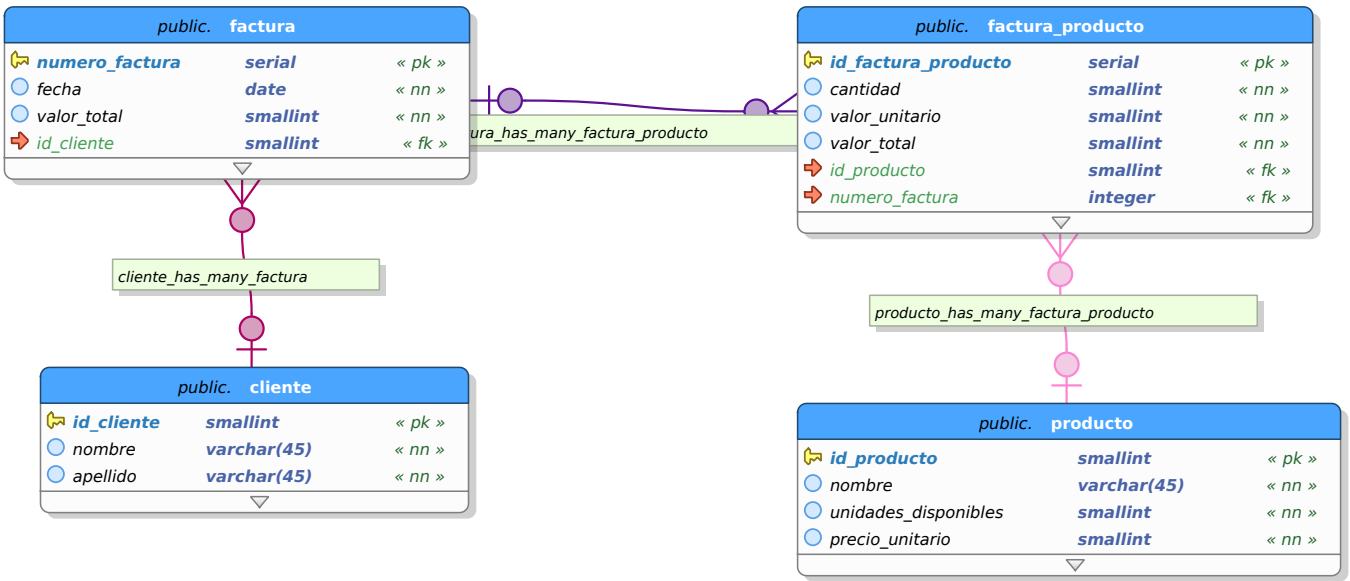

Figura 1: Modelo relacional

Este modelo tiene las siguientes sentencias DDL:

```sql
CREATE TABLE cliente(
    id_cliente smallint NOT NULL,
    nombre varchar(45) NOT NULL,
    apellido varchar(45) NOT NULL,
    PRIMARY KEY (id_cliente)
);
CREATE TABLE producto(
    id_producto smallint NOT NULL,
    nombre varchar(45) NOT NULL,
    unidades_disponibles smallint NOT NULL,
    precio_unitario smallint NOT NULL,
    PRIMARY KEY (id_producto)
);
CREATE TABLE factura(
    numero_factura serial NOT NULL,
    fecha date NOT NULL,
    valor_total smallint NOT NULL,
    id_cliente smallint,
    PRIMARY KEY (numero_factura),
    FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente)
);
CREATE TABLE factura_producto(
    id_factura_producto serial NOT NULL,
    cantidad smallint NOT NULL,
    valor_unitario smallint NOT NULL,
    valor_total smallint NOT NULL,
    id_producto smallint,
    numero_factura integer,
    PRIMARY KEY (id_factura_producto),
    FOREIGN KEY (id_producto) REFERENCES producto (id_producto),
    FOREIGN KEY (numero_factura) REFERENCES factura (numero_factura)
);
```

Además, tiene registros dados por las siguientes sentencias DML.

```sql
insert into cliente (id_cliente, nombre, apellido)
values (10, 'Clark', 'Kent');
insert into cliente (id_cliente, nombre, apellido)
values (20, 'Bruce', 'Wayne');
insert into cliente (id_cliente, nombre, apellido)
values (30, 'Petter', 'Parker');

insert into producto (id_producto, nombre, unidades_disponibles,
   precio_unitario)
values (100, 'Cuaderno', 250, 3500);
insert into producto (id_producto, nombre, unidades_disponibles,
   precio_unitario)
values (200, 'Marcador', 120, 2800);
insert into producto (id_producto, nombre, unidades_disponibles,
   precio_unitario)
```

```
values (300, 'CD', 85, 750);
insert into producto (id_producto, nombre, unidades_disponibles,
   precio_unitario)
values (400, 'Lapiz', 800, 500);
```

1. [0.5 ptos.] Escriba una operación de álgebra relacional que permita mostrar el nombre de los productos y las respectivas cantidades vendidas en cada factura.

Escriba las sentencias SQL para obtener los siguientes resultados:

2. [0.5 ptos.] Creación de una factura a Petter Parker que contiene 2 cuadernos y 2 lápices.
3. [0.5 ptos.] Consulta de nombre de producto y unidades vendidas, ordenados del producto más vendido al menos vendido.
4. [1.0 ptos.] Creación de una vista que despliega todos las facturas con sus respectivos clientes.
5. [1.0 ptos.] Creación de un Trigger que actualiza el numero de unidades disponibles de un producto cuando se crea una factura que contiene dicho producto. Por ejemplo, considere que una factura incluye 2 cuadernos, entonces las unidades disponibles el registro *Cuaderno* en la tabla `Producto` deben decrementarse en 2.
6. [1.0 ptos.] Consulta de nombre y apellido de clientes con la cantidad de facturas y el promedio del valor total facturado por cada cliente, para los clientes que tienen mas de una factura.
7. [0.5 ptos.] Con base en el siguiente código fuente en Python (no se requiere ejecutar), indique que cambios se generan en la base de datos.

```
import psycopg2

con = psycopg2.connect(user = "postgres",
                       password = "postgres",
                       host = "localhost",
                       port = "5432",
                       database = "postgres")

def una_transaccion():
    cur = con.cursor()
    cur.execute("INSERT INTO factura_producto (cantidad,
       valor_unitario, valor_total) VALUES (12, 3500, 42000, 100,
        5)")
    con.commit()
    cur.execute("INSERT INTO factura (numero_factura, fecha,
       valor_total, id_cliente) VALUES (5, today(), 54500, 20)")
    con.commit()
    cur.execute("INSERT INTO factura_producto (cantidad,
       valor_unitario, valor_total) VALUES (25, 500, 12500, 400,
       5)")
    con.commit()
```