

PRIMER PARCIAL

9 de septiembre de 2020

Indicaciones generales

- Este es un examen **individual** con una duración de **120 minutos: de 16:00 a 18:00**.
- 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.

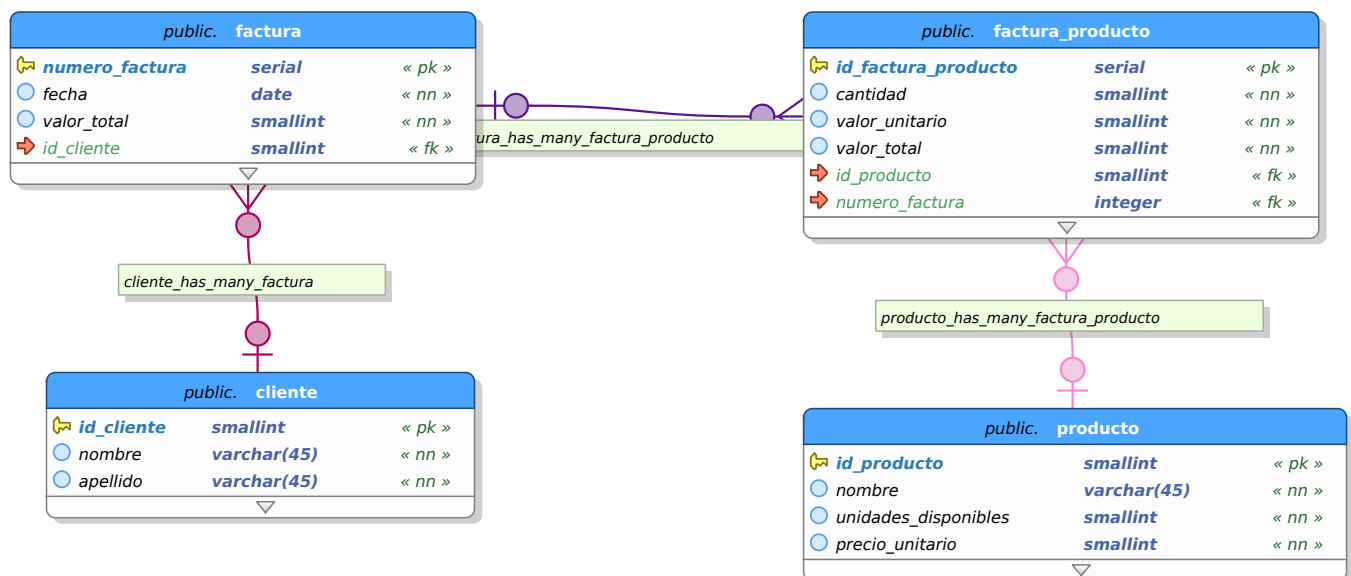


Figura 1: Modelo relacional

Para construir un esquema de prueba de este modelo relacional, se recomienda crear una nueva base de datos con nombre **facturacion** y ejecutar las sentencias SQL adjuntas en la plataforma en el archivo **facturacion.sql**, las cuales se presentan a continuación:

■ Sentencias DDL:

```
1      CREATE TABLE cliente(  
2          id_cliente SMALLINT NOT NULL,  
3          nombre VARCHAR(45) NOT NULL,  
4          apellido VARCHAR(45) NOT NULL,  
5          PRIMARY KEY (id_cliente)  
6      );  
7      CREATE TABLE producto(  
8          id_producto SMALLINT NOT NULL,  
9          nombre VARCHAR(45) NOT NULL,  
10         unidades_disponibles SMALLINT NOT NULL,  
11         precio_unitario SMALLINT NOT NULL,  
12         PRIMARY KEY (id_producto)  
13     );  
14     CREATE TABLE factura(  
15         numero_factura serial NOT NULL,  
16         fecha DATE NOT NULL,  
17         valor_total SMALLINT NOT NULL,  
18         id_cliente SMALLINT,  
19         PRIMARY KEY (numero_factura),  
20         FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente)  
21     );  
22     CREATE TABLE factura_producto(  
23         id_factura_producto serial NOT NULL,  
24         cantidad SMALLINT NOT NULL,  
25         valor_unitario SMALLINT NOT NULL,  
26         valor_total SMALLINT NOT NULL,  
27         id_producto SMALLINT,  
28         numero_factura integer,  
29         PRIMARY KEY (id_factura_producto),  
30         FOREIGN KEY (id_producto) REFERENCES producto (  
31             id_producto),  
32         FOREIGN KEY (numero_factura) REFERENCES factura (  
33             numero_factura)  
34     );
```

■ Sentencias DML:

```
1      INSERT INTO cliente (id_cliente, nombre, apellido)  
2      VALUES (10, 'Clark', 'Kent');  
3      INSERT INTO cliente (id_cliente, nombre, apellido)  
4      VALUES (20, 'Bruce', 'Wayne');  
5      INSERT INTO cliente (id_cliente, nombre, apellido)  
6      VALUES (30, 'Petter', 'Parker');  
7
```



```
8      INSERT INTO producto (id_producto, nombre,  
      unidades_disponibles, precio_unitario)  
9      VALUES (100, 'Cuaderno', 250, 3500);  
10     INSERT INTO producto (id_producto, nombre,  
      unidades_disponibles, precio_unitario)  
11     VALUES (200, 'Marcador', 120, 2800);  
12     INSERT INTO producto (id_producto, nombre,  
      unidades_disponibles, precio_unitario)  
13     VALUES (300, 'CD', 85, 750);  
14     INSERT INTO producto (id_producto, nombre,  
      unidades_disponibles, precio_unitario)  
15     VALUES (400, 'Lapiz', 800, 500);
```

1. [0.5 ptos.] Escriba una operación de álgebra relacional que permita mostrar el nombre y apellido de los clientes, junto a la fecha de factura y valor total de factura de cada una de las facturas.

Escriba las sentencias SQL para obtener los siguientes resultados:

2. [0.5 ptos.] Creación de una factura a Clark Kent que contiene 3 cuadernos.
3. [1.0 ptos.] Creación de una vista que despliega todas las facturas con sus respectivos clientes.
4. [1.0 ptos.] Para este punto es posible ejecutar las sentencias DML que se encuentran en el punto 4 de la plantilla de respuestas `facturacion.sql`. Consulta de nombre y apellido de clientes con la cantidad de facturas de cada cliente y el valor total facturado por cada cliente, para aquellos clientes con más de una factura.
5. [0.5 ptos.] Consulta de nombre de producto y unidades vendidas, ordenados del producto más vendido al menos vendido.
6. [1.0 ptos.] Cree una función de plpgsql que reciba como parámetro el id de un cliente y retorne una tabla con el listado de los productos comprados, sin repeticiones y ordenados alfabéticamente.
7. [0.5 ptos.] Con base en el siguiente código fuente en Python (sin ejecutarlo), indique que cambios se generan en la base de datos.

```
1  import psycopg2  
2  
3  con = psycopg2.connect(user = "postgres",  
4                          password = "postgres",  
5                          host = "localhost",  
6                          port = "5432",  
7                          database = "facturacion")  
8  
9  def una_transaccion():  
10     cur = con.cursor()  
11     cur.execute("INSERT INTO factura (fecha, valor_total,  
                id_cliente) VALUES (now()::DATE, 100000,10);")
```



```
12      con.commit()
13      cur.execute("DELETE FROM cliente WHERE id_cliente = 10;")
14      con.commit()
15      cur.execute("INSERT INTO cliente (id_cliente,nombre,apellido
16                  ) VALUES (10, 'Super','Man');")
17      con.commit()
```