

**Nombre:** Diego Pacheco Valdez

**No. Cuenta:** 195141

**Instrucciones Generales:** Le correctamente todos los apartados y las preguntas y contesta de acuerdo lo que se te pide, al terminar ponle tu nombre al documento y súbelo en formato de PDF. Ejemplo: Examen\_1\_Fulanito\_Rodriguez.pdf

### **Parte Práctica**

**Instrucciones:** leer y contestar las instrucciones de cada inciso. Recordar que al momento de subir el examen hay que subir el archivo de Python así recordar que debe estar bien comentado su o sus códigos

#### **Ejercicio 2: Fundamentos: Conexión a bases de datos y manipulación de datos en Python (20 puntos)**

**Conecte a la base de datos almacen. (2 puntos)**

```
import psycopg2

# Conectarse a la base de datos
conn = psycopg2.connect(
    dbname="almacen",
    user="postgres",
    password="1234",
    host="localhost"
)
```

Inserte un nuevo producto en la tabla productos. (3 puntos)

```
print("\n----- Inserción de un Nuevo Producto
-----")
pNombre = input("\nIntroduzca el Nombre del Producto: ")
pDesc = input("\nIntroduzca la Descripción del Producto: ")
pPrecio = float(input("\nIntroduzca el Precio del Producto: "))

cur.execute("""
    INSERT INTO productos (nombre, descripcion, precio)
    VALUES (%s, %s, %s)
    RETURNING id
""", (pNombre, pDesc, pPrecio))

# Acceder al id del Producto
idProducto = cur.fetchone()[0]

print(f"Producto con ID {idProducto} Creado Correctamente")

# Guardar el nuevo Producto
conn.commit()
```

```
----- Inserción de un Nuevo Producto -----
Introduzca el Nombre del Producto: Botellas
Introduzca la Descripción del Producto: Botellas de Vidrio para Rellenar
Introduzca el Precio del Producto: 145.00
Producto con ID 1 Creado Correctamente

----- Inserción de un Nuevo Cliente -----
Introduzca el Nombre del Cliente:
```

**Inserte un nuevo cliente en la tabla clientes. (3 puntos)**

```
print("\n----- Inserción de un Nuevo Cliente
-----")
cNombre = input("\nIntroduzca el Nombre del Cliente: ")
cEmail = input("\nIntroduzca el Correo del Cliente: ")

cur.execute("""
    INSERT INTO clientes (nombre, email)
    VALUES (%s, %s)
    RETURNING id
""", (cNombre, cEmail))

# Acceder al id del Cliente
idCliente = cur.fetchone()[0]

print(f"Cliente con ID {idCliente} Agregado Correctamente")

# Guardar el nuevo Cliente
conn.commit()
```

```
----- Inserción de un Nuevo Cliente -----
Introduzca el Nombre del Cliente:      Diego
Introduzca el Correo del Cliente:      diego@email.com
Cliente con ID 1 Agregado Correctamente
```

**Inserte una nueva venta relacionada con el cliente y el producto recién agregados en la tabla ventas. (5 puntos) \***

- Verifica que la cantidad en stock del producto es suficiente antes de realizar la venta. Si no es suficiente, lanza una excepción. (2 puntos)
- Actualiza la cantidad en stock del producto después de realizar la venta. (3 puntos)

**Recupere y muestre todos los registros de las tablas productos, clientes y ventas, incluyendo un join para mostrar los nombres de los clientes y productos en las ventas. (7 puntos)**

- Recuperar y mostrar registros de la tabla productos. (2.33 puntos)

```
print("\n----- Recuperar y Mostrar Registros de la
Tabla Productos -----")
# Se selecciona todo lo que hay en productos
cur.execute("SELECT * FROM productos")

# Almacenamos lo seleccionados en productos
productos = cur.fetchall()

# Se imprime producto por producto
for producto in productos:
    print(producto)

----- Recuperar y Mostrar Registros de la Tabla Productos -----
(1, 'Botellas', 'Botellas de Vidrio para Rellenar', Decimal('145.00'), datetime.
date(2024, 6, 21))
```

- Recuperar y mostrar registros de la tabla clientes. (2.33 puntos)

```
print("\n----- Recuperar y Mostrar Registros de la
Tabla Clientes -----")
# Se selecciona todo lo que hay en clientes
cur.execute("SELECT * FROM clientes")

# Almacenamos lo seleccionados en clientes
clientes = cur.fetchall()

# Se imprime cliente por cliente
for cliente in clientes:
    print(cliente)

----- Recuperar y Mostrar Registros de la Tabla Clientes -----
(1, 'Diego', 'diego@email.com', datetime.date(2024, 6, 21))
```

- Recuperar y mostrar registros de la tabla ventas con join para incluir nombres de clientes y productos. (2.34 puntos)

```
# Recuperar y mostrar registros de la tabla ventas con join
para incluir nombres de clientes y productos
print("\n----- Recuperar y mostrar registros de la
tabla ventas -----")
print("\n----- con join para incluir nombres
-----")
print("\n----- de clientes y productos
-----")

# En este execute se consiguen los datos a mostrar
cur.execute("""
    SELECT ventas.id, clientes.nombre, productos.nombre,
ventas.fecha_ventas
    FROM ventas
    JOIN clientes ON ventas.id_cliente = clientes.id
    JOIN productos ON ventas.id_producto = productos.id
""")

# Ahora que ya tenemos el join, lo capturamos con el
cur.fetchall() y lo almacenamos en ventas

ventas = cur.fetchall()
for venta in ventas:
    print(venta)
```

```
----- Recuperar y mostrar registros de la tabla ventas -----
----- con join para incluir nombres -----
----- de clientes y productos -----
(1, 'clien1', 'producto1', datetime.date(2024, 6, 21))
(2, 'clien2', 'producto2', datetime.date(2024, 6, 21))
(3, 'Rodrigo', 'producto1', datetime.date(2024, 6, 21))
(4, 'Rodrigo', 'Cartas de Magic', datetime.date(2024, 6, 21))
```

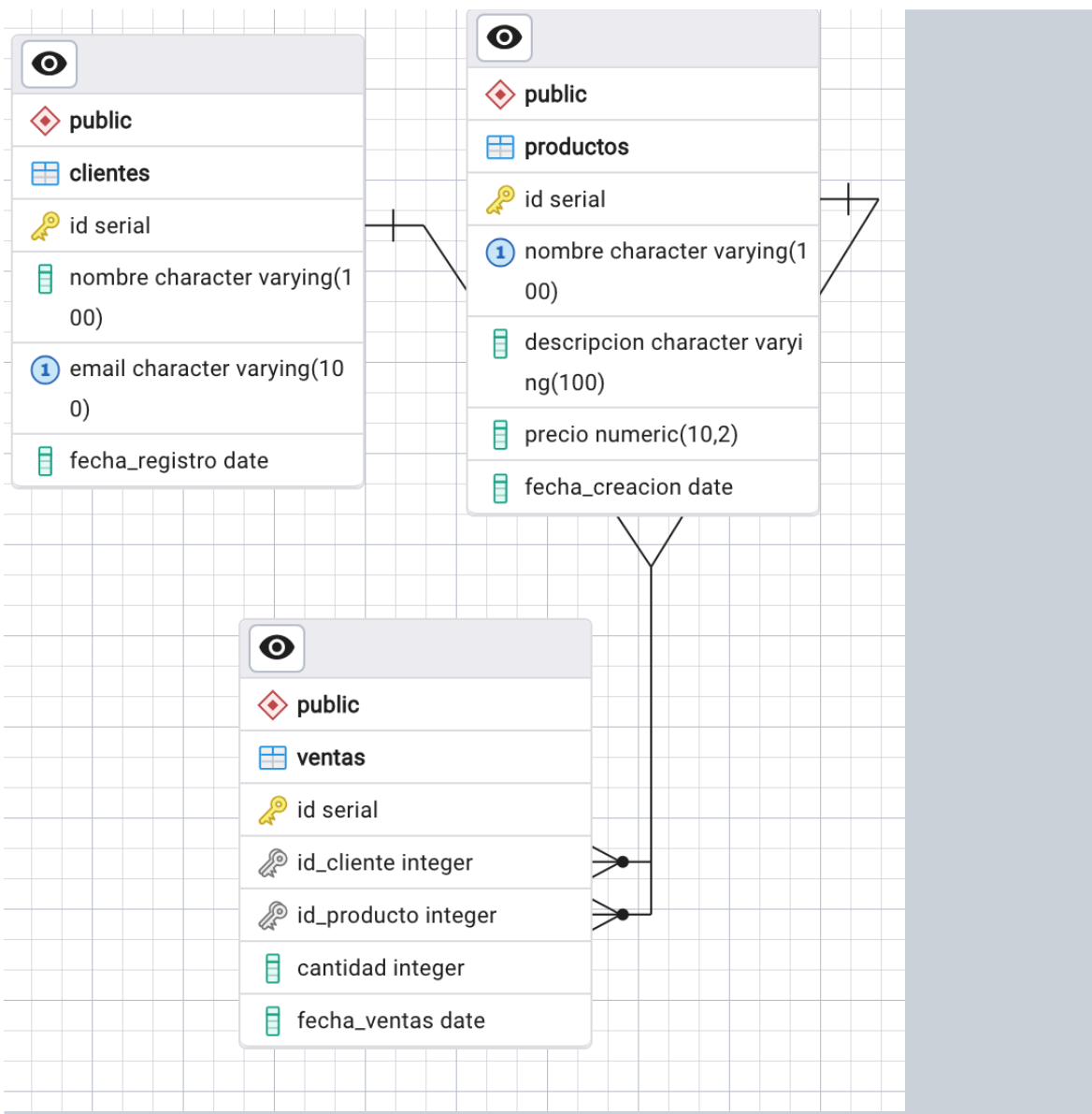
## RELOADED Ejercicio 1: Fundamentos: Creación de tablas y manejo de transacciones en SQL (20 puntos)

Crea una base de datos llamada almacen. (1 punto)

```
psql (16.3)
Type "help" for help.

empresa=# CREATE DATABASE almacen;
CREATE DATABASE
empresa=#
```

**Dentro de esta base de datos, crea tres tablas: productos, clientes y ventas y mostrar el diagrama EDR. (3 puntos)**



- Crear tabla productos con los campos especificados. (1 punto)

```
almacen=# CREATE TABLE productos(
id SERIAL PRIMARY KEY,
nombre VARCHAR(100) UNIQUE NOT NULL,
descripcion VARCHAR(100),
precio NUMERIC(10, 2) NOT NULL,
fecha_creacion DATE DEFAULT CURRENT_DATE);
CREATE TABLE
almacen=#
```

- Crear tabla clientes con los campos especificados. (1 punto)

```
CREATE TABLE
almacen=# CREATE TABLE clientes(
almacen=# id SERIAL PRIMARY KEY,
almacen=# nombre VARCHAR(100) NOT NULL,
almacen=# email VARCHAR(100) UNIQUE NOT NULL,
almacen=# fecha_registro DATE DEFAULT CURRENT_DATE);
CREATE TABLE
almacen=#
```

- Crear tabla ventas con los campos especificados. (1 punto)

```
almacen=# CREATE TABLE ventas(
almacen=# id SERIAL PRIMARY KEY,
almacen=# id_cliente INT NOT NULL REFERENCES clientes(id),
almacen=# id_producto INT NOT NULL REFERENCES productos(id),
almacen=# cantidad INT NOT NULL,
almacen=# fecha_ventas DATE DEFAULT CURRENT_DATE);
CREATE TABLE
```

**La tabla productos debe tener los siguientes campos: (3 puntos)**

- id (clave primaria, autoincremental). (0.75 puntos)
- nombre (único). (0.75 puntos)
- descripcion. (0.5 puntos)
- precio. (0.5 puntos)
- fecha\_creacion (por defecto, la fecha actual). (0.5 puntos)

**La tabla clientes debe tener los siguientes campos: (3 puntos)**

- id (clave primaria, autoincremental). (0.75 puntos)

- nombre. (0.75 puntos)
- email (único). (0.75 puntos)
- fecha\_registro (por defecto, la fecha actual). (0.75 puntos)
- 

**La tabla ventas debe tener los siguientes campos: (3 puntos)**

- id (clave primaria, autoincremental). (0.75 puntos)
- id\_cliente (clave foránea que referencia a clientes(id)). (0.75 puntos)
- id\_producto (clave foránea que referencia a productos(id)). (0.75 puntos)
- cantidad. (0.5 puntos)
- fecha\_venta (por defecto, la fecha actual). (0.25 puntos)



## Inserta al menos dos registros en cada tabla. (2 puntos)

- Insertar registros en productos. (0.67 puntos)

```
# Inserción de productos
print("\n----- Productos -----")
p1Name = input("\nIntroduzca el Nombre del Primer Producto: ")
p1Desc = input("\nIntroduzca la Descripción del Primer Producto: ")
p1Price = float(input("\nIntroduzca el Precio del Primer Producto: "))

cur.execute("""
    INSERT INTO productos (nombre, descripcion, precio)
    VALUES (%s, %s, %s)
    RETURNING id
""", (p1Name, p1Desc, p1Price))

# Capturamos el id del producto para mostrarlo y confirmar
que correctamente se haya guardado
producto1_id = cur.fetchone()[0]
print(f"Producto 1 insertado con ID: {producto1_id}")
conn.commit()

p2Name = input("\nIntroduzca el Nombre del Segundo Producto: ")
p2Desc = input("\nIntroduzca la Descripción del Segundo Producto: ")
p2Price = float(input("\nIntroduzca el Precio del Segundo Producto: "))

cur.execute("""
    INSERT INTO productos (nombre, descripcion, precio)
    VALUES (%s, %s, %s)
    RETURNING id
""", (p2Name, p2Desc, p2Price))

# Capturamos el id del producto para mostrarlo y confirmar
que correctamente se haya guardado
producto2_id = cur.fetchone()[0]
print(f"Producto 2 insertado con ID: {producto2_id}")
conn.commit()
```

```
----- Productos -----  
Introduzca el Nombre del Primer Producto: Botas  
Introduzca la Descripción del Primer Producto: Botas Estilo Dr Martens  
Introduzca el Precio del Primer Producto: 145.00  
Producto 1 insertado con ID: 4  
Introduzca el Nombre del Segundo Producto: Gel  
Introduzca la Descripción del Segundo Producto: Gel Para Peinar  
Introduzca el Precio del Segundo Producto: 89.99  
Producto 2 insertado con ID: 5
```

- Insertar registros en clientes. (0.67 puntos)

```
# Inserción de clientes
print("\n----- Clientes -----")
c1Name = input("\nIntroduzca el Nombre del Primer Cliente:
")
c1Email = input("\nIntroduzca el Correo del Primer Cliente:
")

cur.execute("""
    INSERT INTO clientes (nombre, email)
    VALUES (%s, %s)
    RETURNING id
""", (c1Name, c1Email))
# Capturamos el id del cliente para mostrarlo y confirmar
que correctamente se haya guardado
cliente1_id = cur.fetchone()[0]
print(f"Cliente 1 insertado con ID: {cliente1_id}")
conn.commit()

c2Name = input("\nIntroduzca el Nombre del Segundo Cliente:
")
c2Email = input("\nIntroduzca el Correo del Segundo Cliente:
")

cur.execute("""
    INSERT INTO clientes (nombre, email)
    VALUES (%s, %s)
    RETURNING id
""", (c2Name, c2Email))
# Capturamos el id del cliente para mostrarlo y confirmar
que correctamente se haya guardado
cliente2_id = cur.fetchone()[0]
print(f"Cliente 2 insertado con ID: {cliente2_id}")
conn.commit()
```

```
-----Inserción de registros-----  
".com"  
----- Clientes -----  
  
Introduzca el Nombre del Primer Cliente: Alfredo  
".com"  
Introduzca el Correo del Primer Cliente: alfredo@email.com  
Cliente 1 insertado con ID: 5  
  
Introduzca el Nombre del Segundo Cliente: Mario  
"%s", (%s, %s)  
Introduzca el Correo del Segundo Cliente: mario@email.com  
Cliente 2 insertado con ID: 6
```

- Insertar registros en ventas. (0.66 puntos)

```
# Inserción de ventas
print("\n----- Venta 1 -----")
v1cl = int(input("\nIntroduzca el ID de un Cliente para
Venta 1: "))
v1p = int(input("\nIntroduzca el ID de un Producto para
Venta 1: "))
v1c = int(input("\nIntroduzca la Cantidad Vendida para Venta
1: "))

cur.execute("""
    INSERT INTO ventas (id_cliente, id_producto, cantidad)
    VALUES (%s, %s, %s)
""", (v1cl, v1p, v1c))

print("Venta 1 insertada correctamente.")

print("\n----- Venta 2 -----")
v2cl = int(input("\nIntroduzca el ID de un Cliente para
Venta 2: "))
v2p = int(input("\nIntroduzca el ID de un Producto para
Venta 2: "))
v2c = int(input("\nIntroduzca la Cantidad Vendida para Venta
2: "))

cur.execute("""
    INSERT INTO ventas (id_cliente, id_producto, cantidad)
    VALUES (%s, %s, %s)
""", (v2cl, v2p, v2c))

print("Venta 2 insertada correctamente.")

conn.commit()
```

```

VIC))
----- Venta 1 -----
insertada correctamente.")
Introduzca el ID de un Cliente para Venta 1: 2
Ventas 2 -----")
Introduzca el ID de un Producto para Venta 1: 2
("\nIntroduzca el ID de un Cliente para Venta 1: 34
Introduzca la Cantidad Vendida para Venta 1: 34
Venta 1 insertada correctamente.
("\nIntroduzca el ID de un Producto para Venta 2: 3
----- Venta 2 -----
"\nIntroduzca la Cantidad Vendida para Venta 2: 134
Introduzca el ID de un Cliente para Venta 2: 1
Introduzca el ID de un Producto para Venta 2: 3
Introduzca la Cantidad Vendida para Venta 2: 134
Venta 2 insertada correctamente.

```

**Realiza una transacción que haga lo siguiente: (5 puntos)**

- Inserte un nuevo cliente. (1 punto)
- Inserte una venta asociada al cliente y a uno de los productos. (2 puntos)
- Asegúrate de que la transacción se complete sólo si ambos insertos se realizan correctamente. (1 punto)
- Si alguno de los insertos falla, revierte toda la transacción. (1 punto)

```

print("\n----- La Transacción -----")

# Insertar un nuevo cliente y asociar una venta con un
producto
cName = input("\nIntroduzca el Nombre del Cliente para la
nueva venta: ")
cEmail = input("\nIntroduzca el Correo del Cliente para la
nueva venta: ")
pId = int(input("\nIntroduzca el ID del Producto para la
nueva venta: "))
vC = int(input("\nIntroduzca la Cantidad Vendida para la
nueva venta: "))

try:
    # Insertar un nuevo cliente
    cur.execute("""
        INSERT INTO clientes (nombre, email)
        VALUES (%s, %s)
        RETURNING id
    """, (cName, cEmail))

    cliente_id = cur.fetchone()[0]
    print(f"\nCliente insertado con ID: {cliente_id}")

    # Insertar una venta asociada al nuevo cliente y al
    producto especificado
    cur.execute("""
        INSERT INTO ventas (id_cliente, id_producto,
cantidad)
        VALUES (%s, %s, %s)
    """, (cliente_id, pId, vC))

    print("Nueva venta insertada correctamente.")

    conn.commit()
    print("Transacción completada exitosamente.")

except Exception as e:
    # Si ocurre un error, revertir la transacción
    conn.rollback()
    print(f"Error durante la transacción: {e}")
    print("Transacción revertida.")
finally:
    cur.close()

```

```

----- La Transacción -----

Introduzca el Nombre del Cliente para la nueva venta: Luigi
Introduzca el Correo del Cliente para la nueva venta: luigi@email.com
Introduzca el ID del Producto para la nueva venta: 2
Introduzca la Cantidad Vendida para la nueva venta: 45
Introduzca la Cantidad Vendida para la nueva venta: 45
que la transacción se complete sólo si ambos insertos se
Cliente insertado con ID: 7
Nueva venta insertada correctamente.
Transacción completada exitosamente.

```

```

----- La Transacción -----

Introduzca el Nombre del Cliente para la nueva venta: afdS
Introduzca el Correo del Cliente para la nueva venta: saf@email.com
Introduzca el ID del Producto para la nueva venta: 6
Introduzca la Cantidad Vendida para la nueva venta: 345
Cliente insertado con ID: 8
Error durante la transacción: insert or update on table "ventas" violates foreign
key constraint "ventas_id_producto_fkey"
DETALLE: Key (id_producto)=(6) is not present in table "productos".
Transacción revertida.

```