2- Ejercicio 1: (video relacionado: 2.4: Sistemas OLTP vs OLAP. Contenido relacionado: OLTP y OLAP performance)

El Cloud Arquitect de tu empresa ha notado algunos problemas en la base de datos de la empresa por eso te pide a ti como Data engineer que lo ayudes a analizar el performance de las siguientes secuencias de querys. La base de datos actualmente se encuentra en PostgreSQL.

```
-- 1. Crear tablas:
CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
  Email VARCHAR(100),
  Phone VARCHAR(20)
);
CREATE TABLE Orders (
  OrderID INT PRIMARY KEY,
  CustomerID INT,
  OrderDate DATE,
  TotalAmount DECIMAL(10,2),
  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
CREATE TABLE OrderItems (
  OrderItemID INT PRIMARY KEY,
  OrderID INT,
  ProductID INT,
  Quantity INT,8
```

```
Price DECIMAL(10,2),
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);
CREATE TABLE Products (
 ProductID INT PRIMARY KEY,
 ProductName VARCHAR(100),
 Category VARCHAR(50),
 Price DECIMAL(10,2)
);
```

-- 2. Insertar Data:

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email,
Phone)
VALUES (1, 'John', 'Doe', 'john.doe@example.com', '1234567890');

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES (1, 1, '2022-01-01', 100.00);

INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity,
Price)
VALUES (1, 1, 1, 2, 50.00);

INSERT INTO Products (ProductID, ProductName, Category, Price)
VALUES (1, 'Product A', 'Category A', 50.00);
```

--3. Hacer algunas modificaciones:

```
UPDATE Customers
```

```
SET Phone = '9876543210'

WHERE CustomerID = 1;

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES (2, 1, '2022-02-01', 200.00);

INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity,
Price)
VALUES (2, 2, 1, 3, 50.00);
```

Preguntas Cloud Arquitect:

1. ¿Cuándo ejecutas los INSERT/UPDATE de la sección 2 y 3, notas algo raro en la ejecución cuando lo haces en PostgreSQL o todo fluye rápidamente?
   R// Si ejecutarás lo anterior en Redshift (un servicio de clustering) las inserciones se pueden tomar de 6-10 seg, algo bastante elevado teniendo en cuenta que solo estamos insertando 4 valores. Te puedes imaginar en un caso donde tengamos millones de registros por minuto colapsaría el sistema. Esto se debe en gran medida porque Redshift es un sistema para trabajar análogo a sistemas OLAP donde no se hacen tantas operaciones como Insert, Update y Delete sino más bien sentencias como GROUPBY y JOINS. Al hacer lo mismo en PostgreSQL seguro notarás que fluye todo mucho más rápido
2. ¿Este código corresponde a acciones realizadas por un sistema OLAP o OLTP? ¿Por qué?
   R// Si respondiste OLTP, estás en lo correcto. Esto es porque estamos realizando operaciones como Insert y Update que son típicas de estos sistemas.
3. ¿Cuál sería el orden correcto de borrado de las tablas creadas previamente?
   R// OrderItems > Orders > Customers > Products. Esto es a que existen dependencias transitorias entre las tablas que no te permitirán que borres primero por ejemplo la tabla Customers

2- Ejercicio 2 (Video relacionado: 2.4: Sistemas OLTP vs OLAP. Contenido relacionado: Consultas en sistemas OLTP y OLAP)

Recientemente el Data manager de la empresa te contacta para que lo ayudes a obtener métricas del nuevo negocio de la empresa, los dueños necesitan saber si vale la pena continuar vendiendo el producto estrella, llamado 'Product A'

```
--1. Crear Tablas
CREATE TABLE Sales (
  SalesID INT PRIMARY KEY,
  ProductID INT,
  DateID INT,
  CustomerID INT,
  Quantity INT,
  Amount DECIMAL(10,2)
);

CREATE TABLE Products (
  ProductID INT PRIMARY KEY,
  ProductName VARCHAR(100),
  Category VARCHAR(50)
);

CREATE TABLE Dates (
  DateID INT PRIMARY KEY,
  Date DATE,
```

```
DayOfWeek VARCHAR(20),
  Month VARCHAR(20),
  Year INT
);

CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
  Email VARCHAR(100),
  Phone VARCHAR(20)
);
```

--2. Insertar la data

```
INSERT INTO Sales (SalesID, ProductID, DateID, CustomerID, Quantity,
Amount)
VALUES (1, 1, 1, 1, 2, 100.00);

INSERT INTO Products (ProductID, ProductName, Category)
VALUES (1, 'Product A', 'Category A');

INSERT INTO Dates (DateID, Date, DayOfWeek, Month, Year)
VALUES (1, '2022-01-01', 'Saturday', 'January', 2022);

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone)
VALUES (1, 'John', 'Doe', 'john.doe@example.com', '1234567890');
```

Para esto te pide que le proporciones lo siguiente:

1. Obtener el monto total de ventas por categoría de producto (piensa que tienes millones de datos, por ende, debes hacerlo escalable y eficiente)
   R// Aquí hay un ejemplo de una posible solución

   ```
   SELECT p.Category, SUM(s.Amount) AS TotalSalesAmount
   FROM Sales s
   JOIN Products p ON s.ProductID = p.ProductID
   GROUP BY p.Category;
   ```

2. Obtener las ventas por mes para el producto estrella A
   R// Aquí una posible solución:

   ```
   SELECT d.Month, SUM(s.Amount) AS TotalSalesAmount
   FROM Sales s
   JOIN Products p ON s.ProductID = p.ProductID
   JOIN Dates d ON s.DateID = d.DateID
   WHERE p.ProductName = 'Product A'
   GROUP BY d.Month;
   ```