**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Steps:**

1. **Understand Recursive Algorithms:**
   * Explain the concept of recursion and how it can simplify certain problems.
2. **Setup:**
   * Create a method to calculate the future value using a recursive approach.
3. **Implementation:**
   * Implement a recursive algorithm to predict future values based on past growth rates.
4. **Analysis:**
   * Discuss the time complexity of your recursive algorithm.
   * Explain how to optimize the recursive solution to avoid excessive computation.

IF OBJECT\_ID('OrderDetails', 'U') IS NOT NULL DROP TABLE OrderDetails;

IF OBJECT\_ID('Orders', 'U') IS NOT NULL DROP TABLE Orders;

IF OBJECT\_ID('Products', 'U') IS NOT NULL DROP TABLE Products;

IF OBJECT\_ID('Customers', 'U') IS NOT NULL DROP TABLE Customers;

-- Database Schema

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name VARCHAR(100),

Region VARCHAR(50)

);

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2)

);

CREATE TABLE Orders (

OrderID INT,

CustomerID INT,

OrderDate DATE,

CONSTRAINT PK\_Orders PRIMARY KEY NONCLUSTERED (OrderID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Sample Data

INSERT INTO Customers (CustomerID, Name, Region) VALUES

(1, 'Alice', 'North'),

(2, 'Bob', 'South'),

(3, 'Charlie', 'East'),

(4, 'David', 'West');

INSERT INTO Products (ProductID, ProductName, Category, Price) VALUES

(1, 'Laptop', 'Electronics', 1200.00),

(2, 'Smartphone', 'Electronics', 800.00),

(3, 'Tablet', 'Electronics', 600.00),

(4, 'Headphones', 'Accessories', 150.00);

INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES

(1, 1, '2023-01-15'),

(2, 2, '2023-02-20'),

(3, 3, '2023-03-25'),

(4, 4, '2023-04-30');

INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES

(1, 1, 1, 1),

(2, 2, 2, 2),

(3, 3, 3, 1),

(4, 4, 4, 3);

-- Exercise 1: Non-Clustered Index

SELECT \* FROM Products WHERE ProductName = 'Laptop';

CREATE NONCLUSTERED INDEX IX\_Products\_ProductName ON Products(ProductName);

SELECT \* FROM Products WHERE ProductName = 'Laptop';

-- Exercise 2: Clustered Index on OrderDate

SELECT \* FROM Orders WHERE OrderDate = '2023-01-15';

CREATE CLUSTERED INDEX IX\_Orders\_OrderDate ON Orders(OrderDate);

SELECT \* FROM Orders WHERE OrderDate = '2023-01-15';

-- Exercise 3: Composite Index on CustomerID + OrderDate

SELECT \* FROM Orders WHERE CustomerID = 1 AND OrderDate = '2023-01-15';

CREATE NONCLUSTERED INDEX IX\_Orders\_CustomerID\_OrderDate ON Orders(CustomerID, OrderDate);

SELECT \* FROM Orders WHERE CustomerID = 1 AND OrderDate = '2023-01-15';

