COMMON TABLE EXPRESSION (CTE) & STORE PROCEDURE

QUESTION 1

Using W3 database from the "products" table, calculate the discount based on the following criteria, and at the end show the total (sum) of given discounts on all products.

- Price less than 25$: No discount
- Price more than 25$ and less than 50$: 15% discount
- Price more than 50$: 20% discount

Expected final view:

```
******************************
Total Discounted Price        | Total Discount |
******************************
12345.00$                     | 1234.00$       |
```

QUESTION 2

Write a Stored Procedure to filter products in W3 database given the category name, and a price range (minimum price and maximum price) as input arguments. Please note that the input arguments of this stored procedure are optional, and it should work with any number of input arguments.

Write the command to run that Stored Procedure.

QUESTION 3

✓ Initial Task:

- Write a stored procedure sp_products that accepts the following parameters:
- @query (optional): A keyword to search for in the product name. If no keyword is provided, return all products.
- @minPrice (optional): The minimum price of the products to be included in the results. The default value is 0.

- @maxPrice (optional): The maximum price of the products. If not provided, it should default to the highest price in the Products table.

✓ Updated Task:

- You need to modify the sp_products procedure to also filter products by their category. The new version of the procedure should include:
- @categoryID (optional): The category ID to filter by. If this value is NULL, the procedure should use 2 as the default category.

QUESTION 4

You are tasked with analyzing customer orders for a company. Your goal is to create a stored procedure named sp_CustomersOrders that generates a report summarizing customer purchases based on the following conditions:

Input Parameters:

@CustomerName (optional): A partial or full customer name for filtering. If not provided, it should return all customers.

@MinAmount (optional): The minimum total payment made by the customer. The default value is 0.

@MaxAmount (optional): The maximum total payment made by the customer. If not provided, it should default to the highest total payment among all customers.

@city (optional): The city where the customer is located. If not provided, the procedure should return customers from all cities.

@country (optional): The country where the customer is located. If not provided, the procedure should return customers from all countries.

Requirements:

The report should include customer name, contact name, city, country, total amount paid, total number of orders, and total quantity of products ordered.

The report should be grouped by customer name, contact name, city, and country.

Customers should only be displayed if they match the search criteria, and the total amount paid should be between the specified @MinAmount and @MaxAmount.

The results should be ordered by the total amount paid in descending order.

QUESTION 5

Create two stored procedures in SQL Server to generate a pattern of asterisks (*) and hashes (#). The first procedure, create_pattern, uses variables to build the pattern and then prints it. The second procedure, create_pattern2, directly prints the pattern without using intermediate variables.

- **create_pattern(@n INT = 2)**: This procedure generates a pattern where the number of asterisks decreases and the number of hashes increases with each iteration. It uses variables to store the parts of the pattern before printing.
- **create_pattern2(@n INT = 2)**: This procedure achieves the same pattern but prints it directly without using intermediate variables.

QUESTION 6

Create a stored procedure sp_search_customers to find customers based on Country or City.