Amrita Vishwa Vidyapeetham

Department of Artificial Intelligence

21AIE304 Big Data and DataBase Management Systems

Practice Problems -3

Date: 10-10-2023

Instructions

a. Please compile the practice problems into a Word document for PP1 and save it in your OneDrive folder.
b. Create a section titled 'Practice Problem 3.' For each question, provide the associated code in text format immediately below the question.
c. Kindly refrain from including any screenshots.

1. Suppose you have a DataFrame containing information about user activities on a website, and you want to perform the following tasks:

   Extract the domain from the page_url.
   Count the number of visits for each user on each domain.
   Find the top domain for each user based on the number of visits.

   Assume the data looks like

```
val data = Seq( (1, "https://example.com/spark/page1"),
(2, "https://example.com/spark/page2"),
(1, https://example.com/spark/page3),
(3, "https://example.com/hadoop/page1"),
(2, https://example.com/spark/page4),
(3, "https://example.com/spark/page5"),
(1, https://anotherdomain.com/page6),
(2, "https://anotherdomain.com/page7") )
```

Write an SQL query to generate a report that includes the book's title, original price, and the discounted price based on the criteria above.

2. Suppose you have a CSV file named **`shopping_data.csv`** with the following attributes: **`customer_id,product_id,quantity,price.`** Compute the below

   Total Spending per Customer: Calculate the total spending for each customer and display the result.

Most Purchased Product: Identify the product that has been purchased the most and display its details.
Average Price per Product: Calculate the average price for each product and display the result.

3. Suppose you have a DataFrame containing information about employees, and you want to add a new column called "performance_category" based on the "performance_score" column. The categorization should be as follows:

   If the performance score is greater than or equal to 90, the category is "Excellent."
   If the performance score is between 80 and 89 (inclusive), the category is "Good."
   If the performance score is between 70 and 79 (inclusive), the category is "Average."
   If the performance score is below 70, the category is "Poor."

A sample dataframe looks like below

val data = Seq((1, "John", 95), (2, "Alice", 85), (3, "Bob", 78), (4, "Eva", 60), (5, "Mike", 88))

4. You have three datasets: employees, departments, and projects. The employees dataset contains information about employees, the departments dataset contains information about departments, and the projects dataset contains information about projects assigned to employees. You need to perform the following tasks:

   Task 1: Inner Join - Employee and Department
   Join the employees and departments datasets using an inner join based on the department_id column.

   Task 2: Left Join - Employee and Projects
   Join the employees and projects datasets using a left join based on the employee_id column.

   Task 3: Right Join - Projects and Employees
   Join the projects and employees datasets using a right join based on the employee_id column.

   Task 4: Full Outer Join - Employee, Department, and Projects
   Join the employees, departments, and projects datasets using a full outer join based on common columns.

5. Suppose you have a dataset of marketing campaign results with columns like "campaign_id," "conversion_rate," and "cost_per_conversion." The goal is to analyze the effectiveness of each campaign and calculate the overall marketing ROI.

The sample data is

val data = Seq(("campaign_1", 0.1, 50.0), ("campaign_2", 0.15, 60.0), ("campaign_3", 0.12, 55.0), ("campaign_4", 0.2, 70.0), ("campaign_5", 0.18, 65.0))

6. Suppose you have a dataset of travel bookings with columns like "booking_id," "destination," and "travel_date." The goal is to identify popular travel destinations and analyze booking trends.

The sample data is

val data = Seq(("booking_1", "City A", "2023-01-01"), ("booking_2", "City B", "2023-01-02"), ("booking_3", "City A", "2023-01-03"), ("booking_4", "City C", "2023-01-04"), ("booking_5", "City B", "2023-01-05"))