**7PAM2005-0901-2023 - Data Mining and Discovery**

**Name:** Sanshiya Rameshkumar

**Student ID :** 21066861

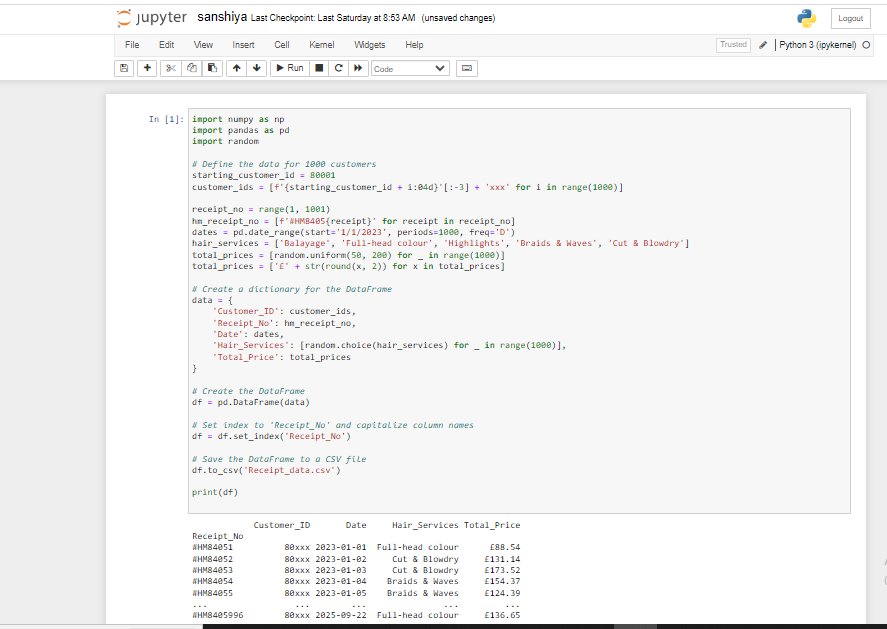
**Title:** Enhancing Headmasters Hair Salon through Comprehensive Sales Analysis

**GitHub Link:** <https://github.com/Sanshiya-23/Enhancing-Headmasters-Hair-Salon-through-Comprehensive-Sales-Analysis.git>

**Abstract:**

This report delves into a meticulous analysis of Headmasters Hair Salon's operations, focusing on three key aspects: receipt data, customer data, and ratings data. The receipt data table unravels the intricacies of how the salon communicates service information and total costs to its customers. Concurrently, the customer data table offers insights into customer profiles, including essential details for targeted marketing strategies aimed at customer retention and loyalty. Lastly, the ratings data sheds light on the performance of the salon's hair stylists, identifying their strengths and areas for improvement in various hair services. Leveraging Python code, a dataset comprising 1000 rows for each table was generated and imported into SQL for comprehensive analysis. The findings from this study aim to provide actionable recommendations to enhance the overall performance and customer satisfaction at Headmasters Hair Salon.

**Report Justification**

**Receipt Data:  
  
**

This code generates and manipulates data to create a DataFrame representing receipt information for a fictional hair salon.

**Customer Information Setup**: Customer IDs are generated for 1000 customers starting from 80001, with partial anonymization. Receipt numbers are created with a specific format (#HM8405x) for 1000 receipts. Dates are generated for a span of 1000 days starting from January 1, 2023.

**Hair Service and Total Price Generation:** A list of hair services is defined, including Balayage, Full-head colour, Highlights, Braids & Waves, and Cut & Blowdry. Total prices are randomly generated for each receipt, ranging from £50 to £200, and formatted accordingly.

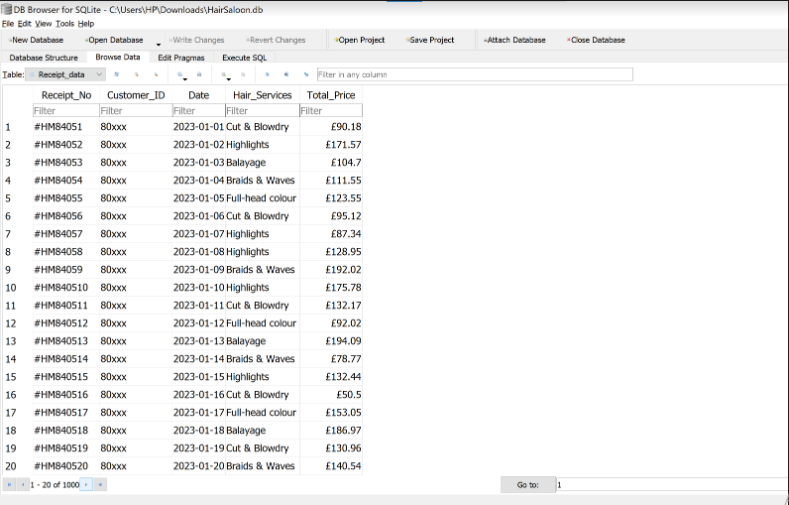
**DataFrame Creation:** A dictionary named 'data' is created, containing keys such as 'Customer\_ID', 'Receipt\_No', 'Date', 'Hair\_Services', and 'Total\_Price'. The dictionary is used to create a Pandas DataFrame named 'df'.

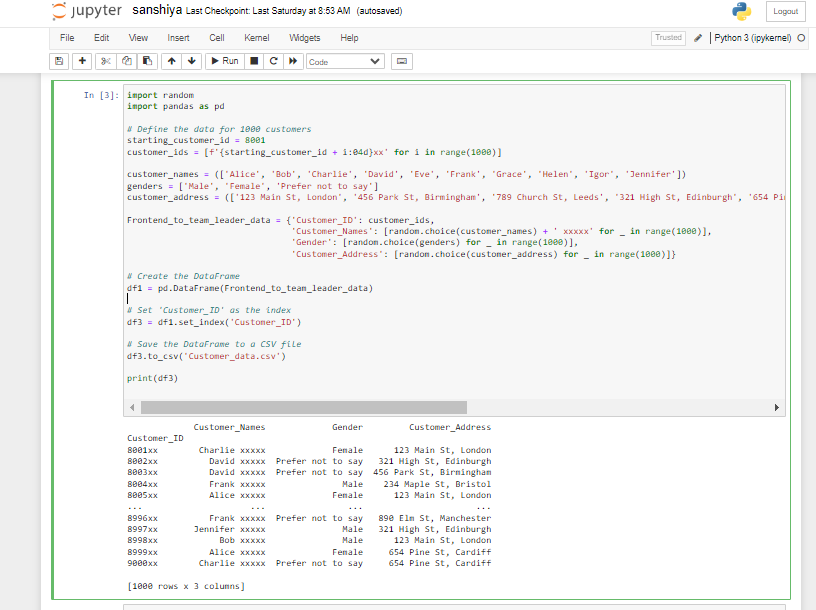
**DataFrame Manipulation:** The 'Receipt\_No' column is set as the index for the DataFrame. Column names are capitalized for consistency.

**Save to CSV:** The DataFrame is saved to a CSV file named 'Receipt\_data.csv'.

**Print the DataFrame:** The DataFrame is printed to the console.

In essence, the code simulates a scenario where a hair salon maintains records of customer transactions, including anonymized customer IDs, receipt numbers, dates of service, types of hair services availed, and total prices. The data is then organized into a Pandas DataFrame and used random library to create data instead of faker seed library to make the report quite realistic and saved to a CSV file for further analysis or record-keeping.



**Customer Data:** 

This code generates a fictional dataset representing customer information for a company.

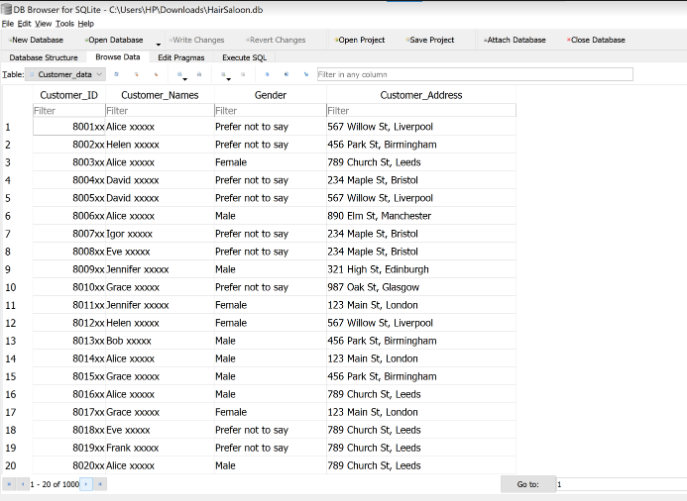
**Customer Information Setup:** Customer IDs are generated for 1000 customers starting from 8001, with partial anonymization by adding 'xx' at the end. Customer names, genders, and addresses are defined. Customer names have 'xxxxx' added at the end for additional privacy.

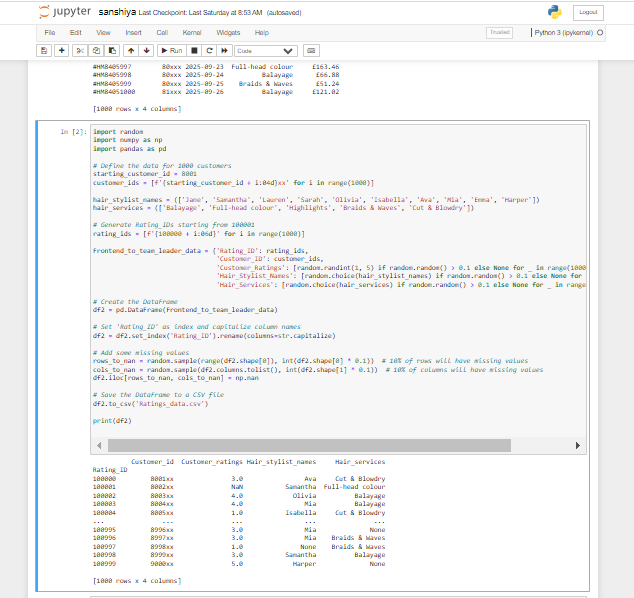
**DataFrame Creation:** A dictionary named 'Frontend\_to\_team\_leader\_data' is created, containing keys such as 'Customer\_ID', 'Customer\_Names', 'Gender', and 'Customer\_Address'. The dictionary is used to create a Pandas DataFrame named 'df1'.

**DataFrame Manipulation:** The 'Customer\_ID' column is set as the index for the DataFrame, resulting in a new DataFrame named 'df3'. This could be useful for quick lookup based on customer IDs.

**Save to CSV:** The DataFrame is saved to a CSV file named 'Customer\_data.csv'. Print the DataFrame: The DataFrame with anonymized customer information is printed to the console.

In essence, the code simulates a scenario where a company maintains records of customer details, including anonymized customer IDs, names, genders, and addresses. The data is organized into a Pandas DataFrame and saved to a CSV file for further analysis, record-keeping, or sharing within the team.



**Ratings Data:**

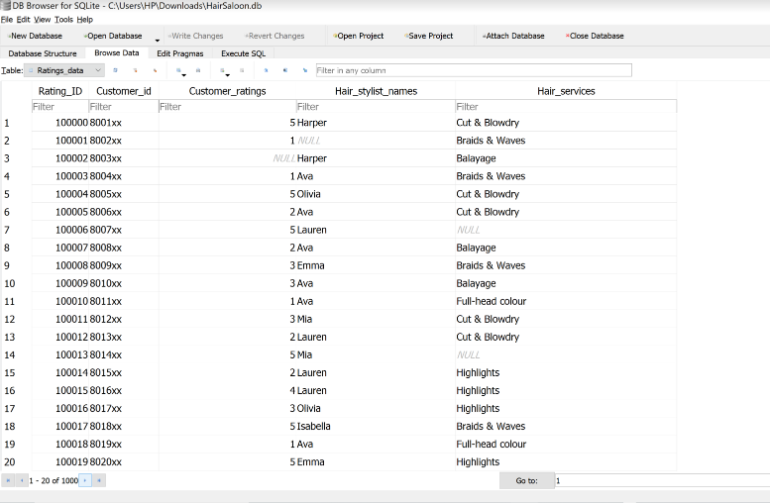
This code generates a simulated dataset representing customer ratings for a hair salon. **Customer Information Setup:** Customer IDs are generated for 1000 customers starting from 8001, with partial anonymization by adding 'xx' at the end. Hair stylist names and types of hair services are defined.

**Rating Information Generation:** Rating IDs are generated starting from 100001. A dictionary named 'Frontend\_to\_team\_leader\_data' is created, containing keys such as 'Rating\_ID', 'Customer\_ID', 'Customer\_Ratings', 'Hair\_Stylist\_Names', and 'Hair\_Services'. Customer ratings, hair stylist names, and types of hair services are randomly assigned. Approximately 10% of values are left as missing (None) to simulate incomplete data.

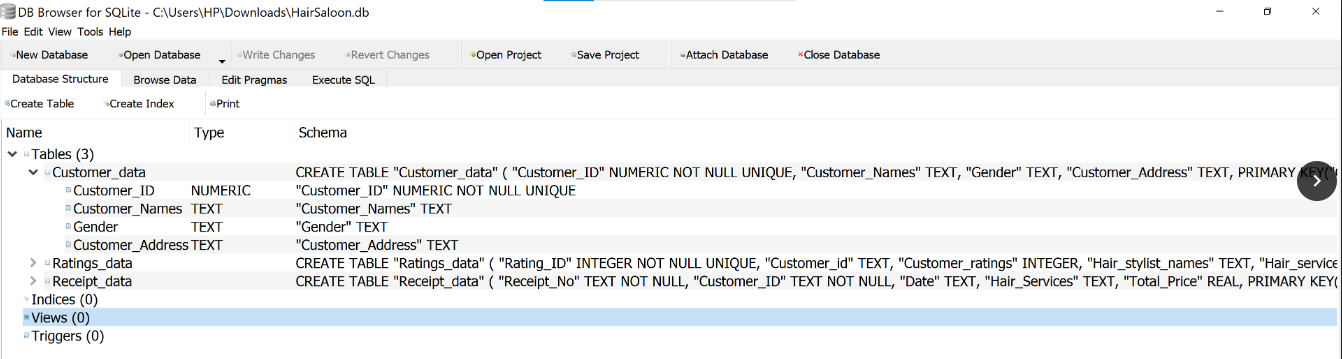
**DataFrame Creation:** The dictionary is used to create a Pandas DataFrame named 'df2'. **DataFrame Manipulation:** The 'Rating\_ID' column is set as the index for the DataFrame. Column names are capitalized for consistency.

**Adding Missing Values:** Approximately 10% of the rows and 10% of the columns in the DataFrame are selected, and their values are replaced with NaN (missing) to simulate incomplete data.

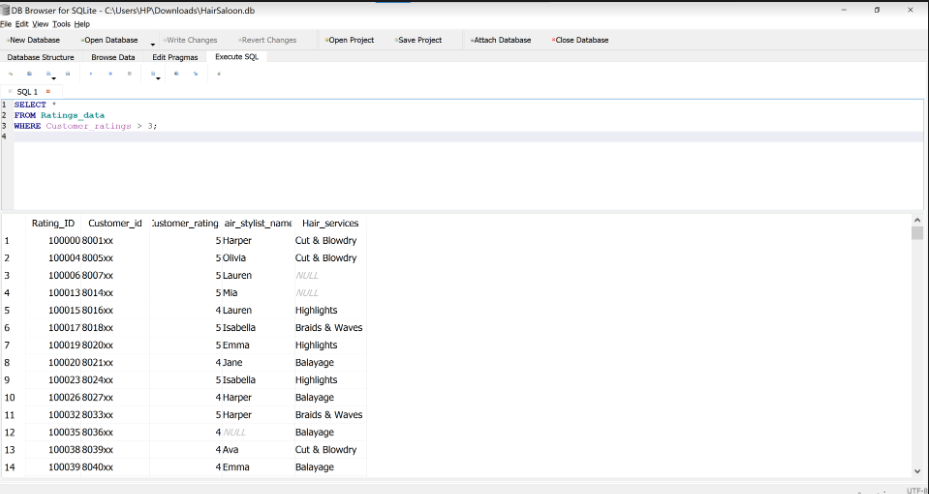
**Save to CSV:** The DataFrame is saved to a CSV file named 'Ratings\_data.csv'. Print the DataFrame: The DataFrame with customer ratings and associated information, including some missing values, is printed to the console. In essence, the code simulates a scenario where a hair salon collects customer ratings for various hair services provided by different stylists. The data is organized into a Pandas DataFrame, and missing values are introduced to mimic the real-world scenario of incomplete feedback or data entry errors. The resulting dataset is saved to a CSV file for further analysis or record-keeping.



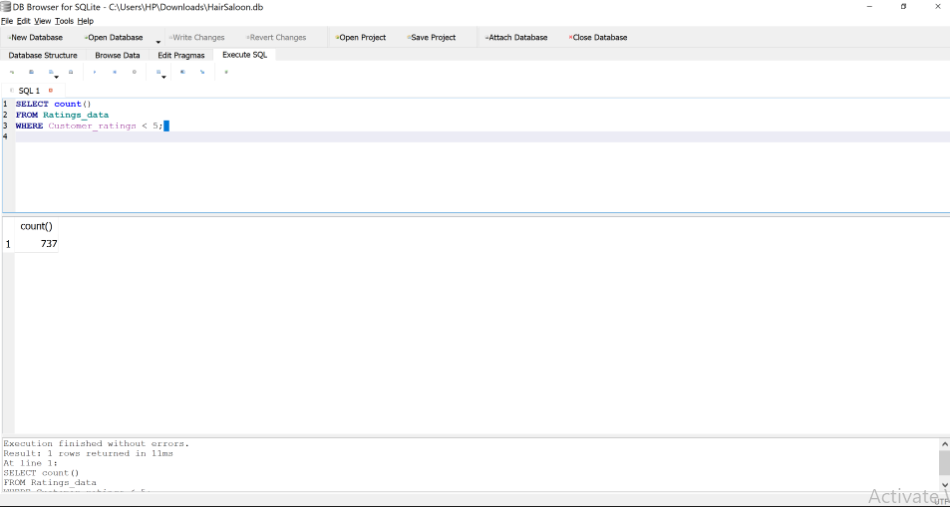
**Database Scheme:**

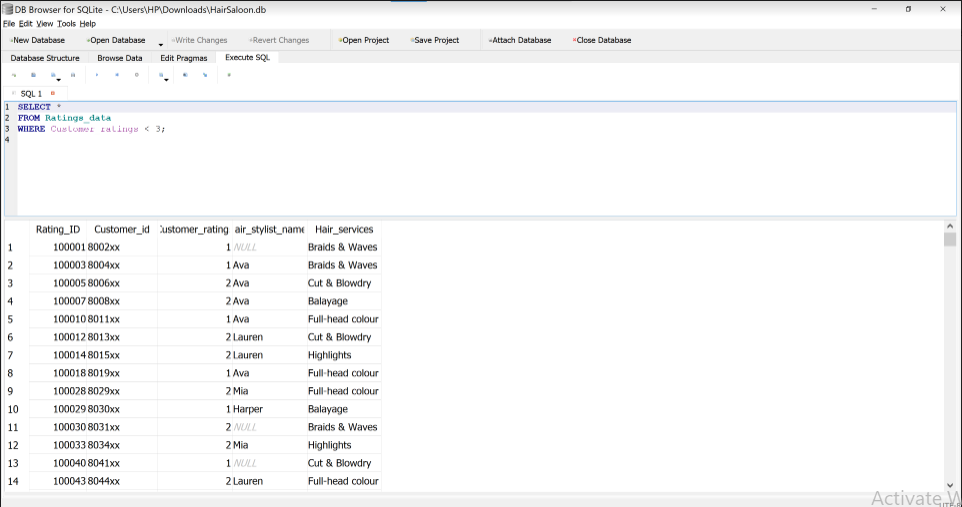
****

**SQL Query:**

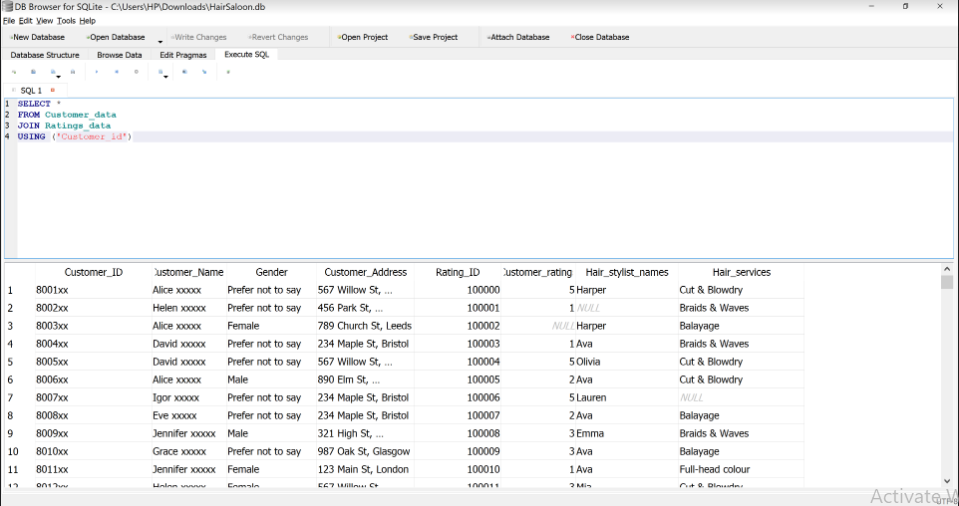
****

I conducted an analysis on Ratings\_data, isolating entries where customer ratings exceeded 3. This targeted approach aims to spotlight outstanding stylist performance, fostering a culture of excellence and improvement.



Leveraging the COUNT operator, I quantified our salon's positive ratings, providing a numerical measure of customer satisfaction. This insight is pivotal for recognizing strengths and steering improvements in service quality.  
  


By executing an SQL query for ratings below 3, I pinpointed areas in stylist performance that require improvement. This focused analysis guides targeted enhancements for an elevated customer experience.



I’ve added the JOIN operator in an SQL query to seamlessly merge customer and rating data. This integration enriches our understanding, fostering a holistic view for informed decisions and enhanced services.