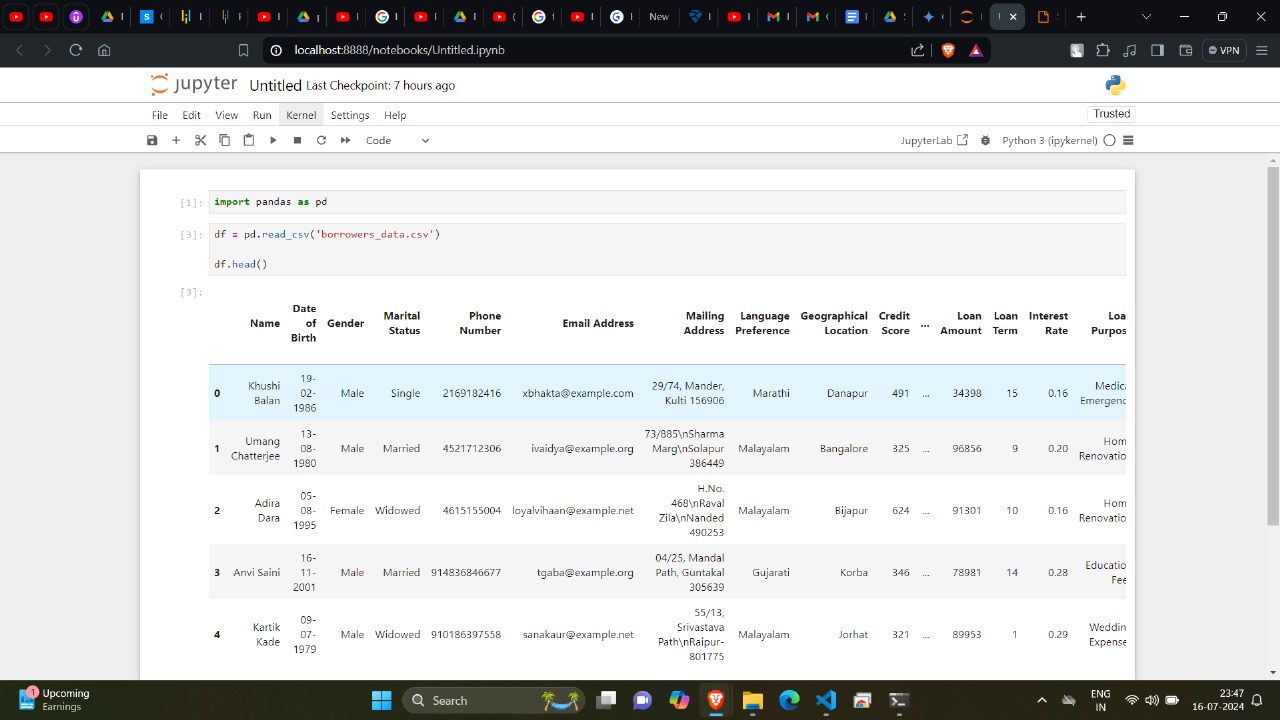
This code performs data preparation and export steps to get your data ready for import into an SQL database. The script utilizes the Pandas library in Python, Jupyter.

**1. Library Import:**

The code begins by importing the Pandas library, a powerful tool for data manipulation and analysis in Python.



**2. Reading CSV Data:**

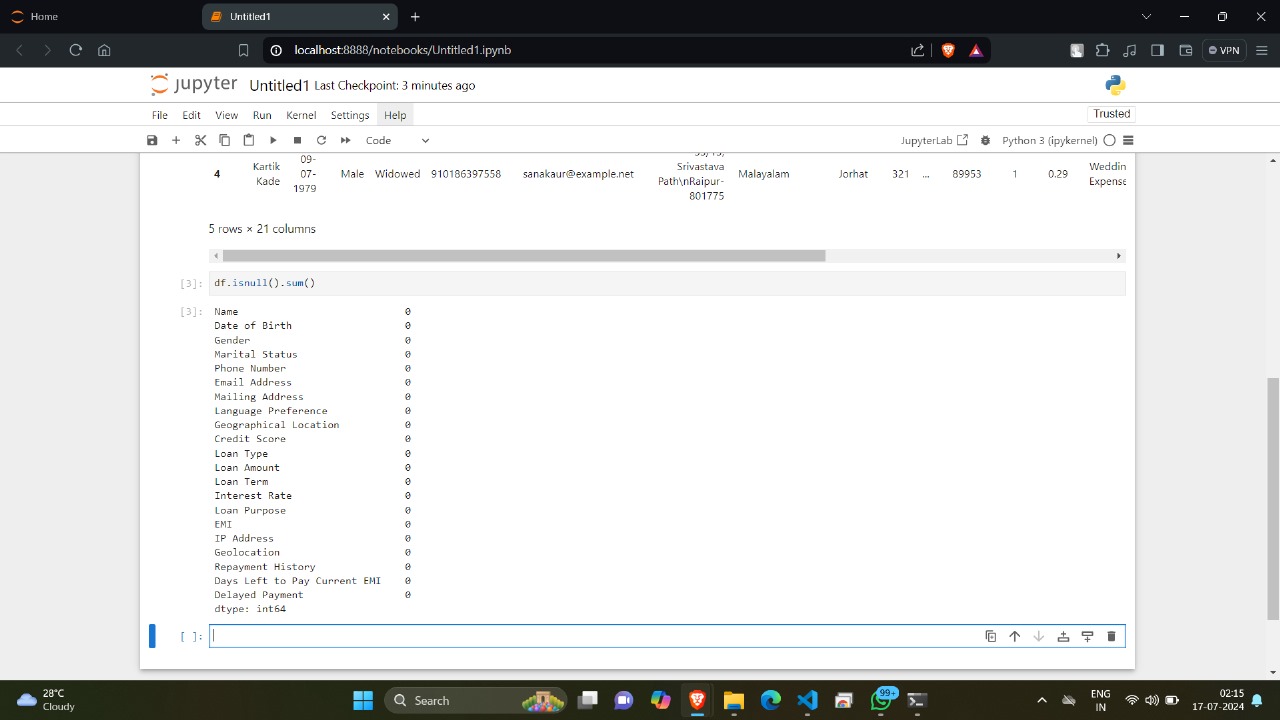
The script then reads your data from a CSV (Comma-Separated Values) file. This file should be located in the same directory as your Python script.

**3. Data Cleaning:**

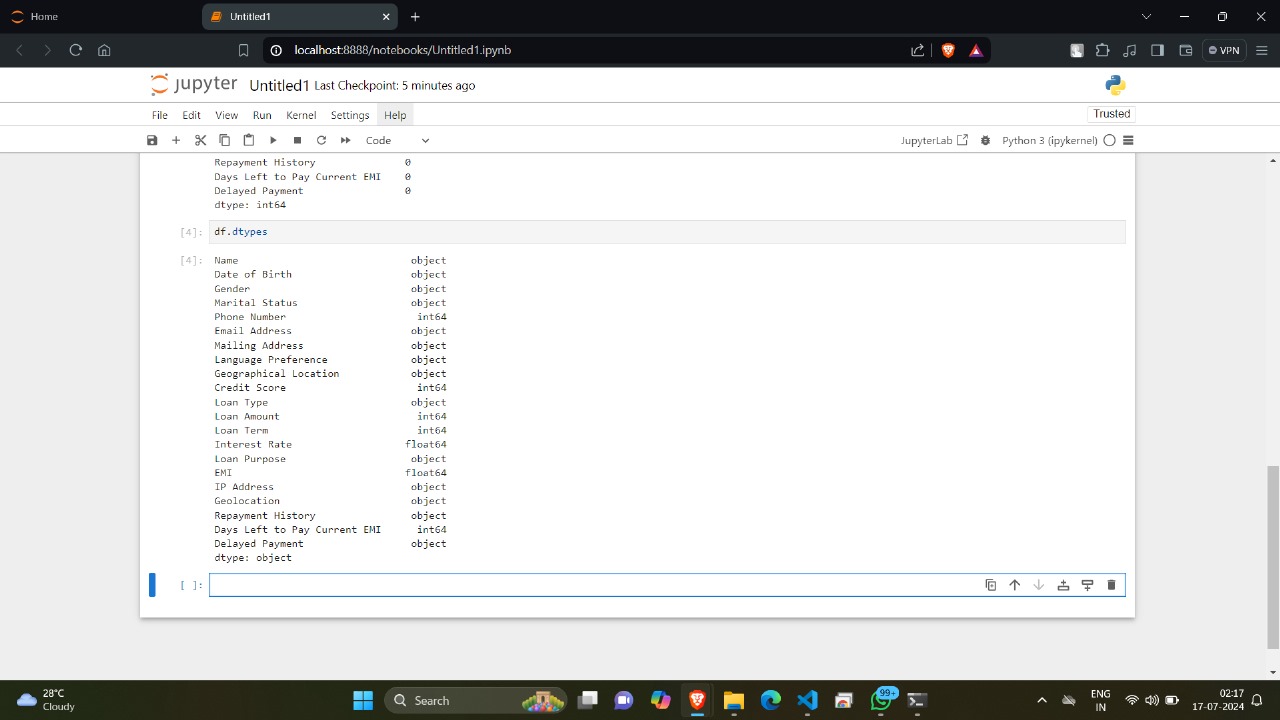
This crucial step ensures your data is clean and consistent before import into SQL. Here's what the code does:

Drop Unnecessary Columns: You can identify and remove any columns that are irrelevant for your SQL analysis. This helps streamline your data and improve efficiency.

Missing Value Check: The code checks for missing values (like empty cells) in your data. You can choose to remove rows with missing values, impute them with estimated values, or handle them based on your specific needs.



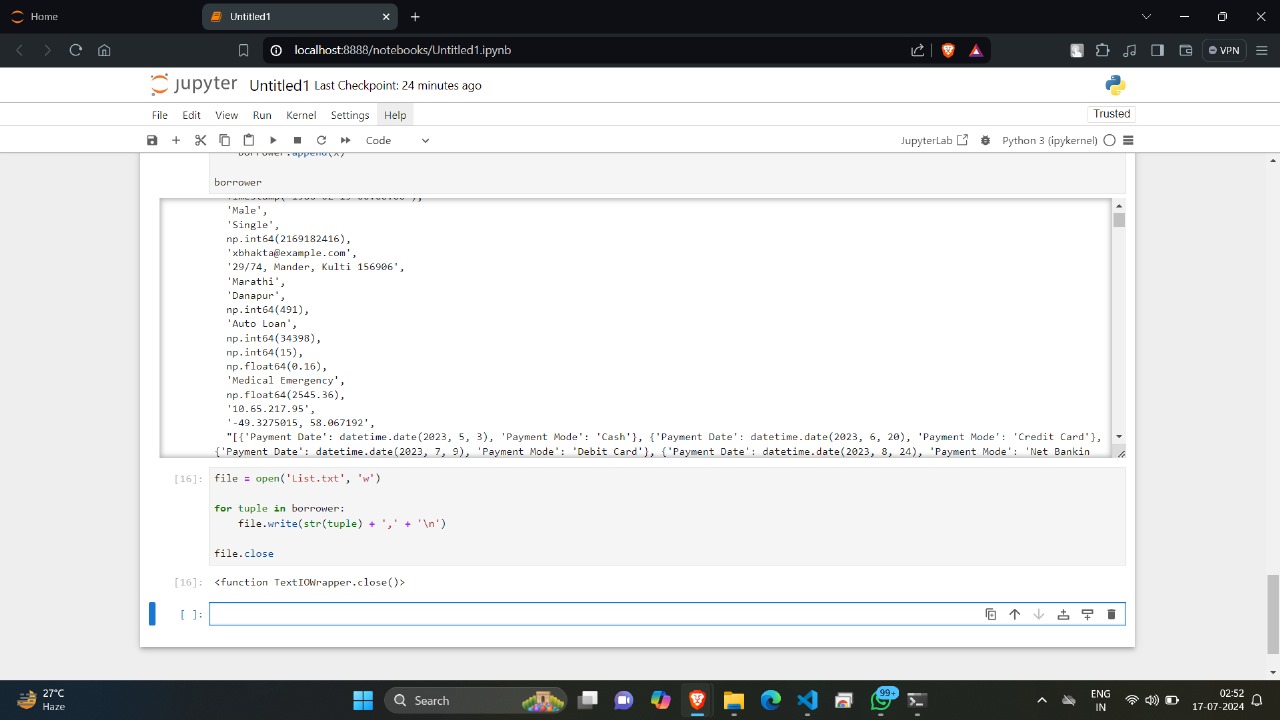
Data Type Handling: Pandas allows you to ensure your data adheres to the correct data types in your SQL database. This might involve converting columns from strings to numbers, dates, or other appropriate data types.



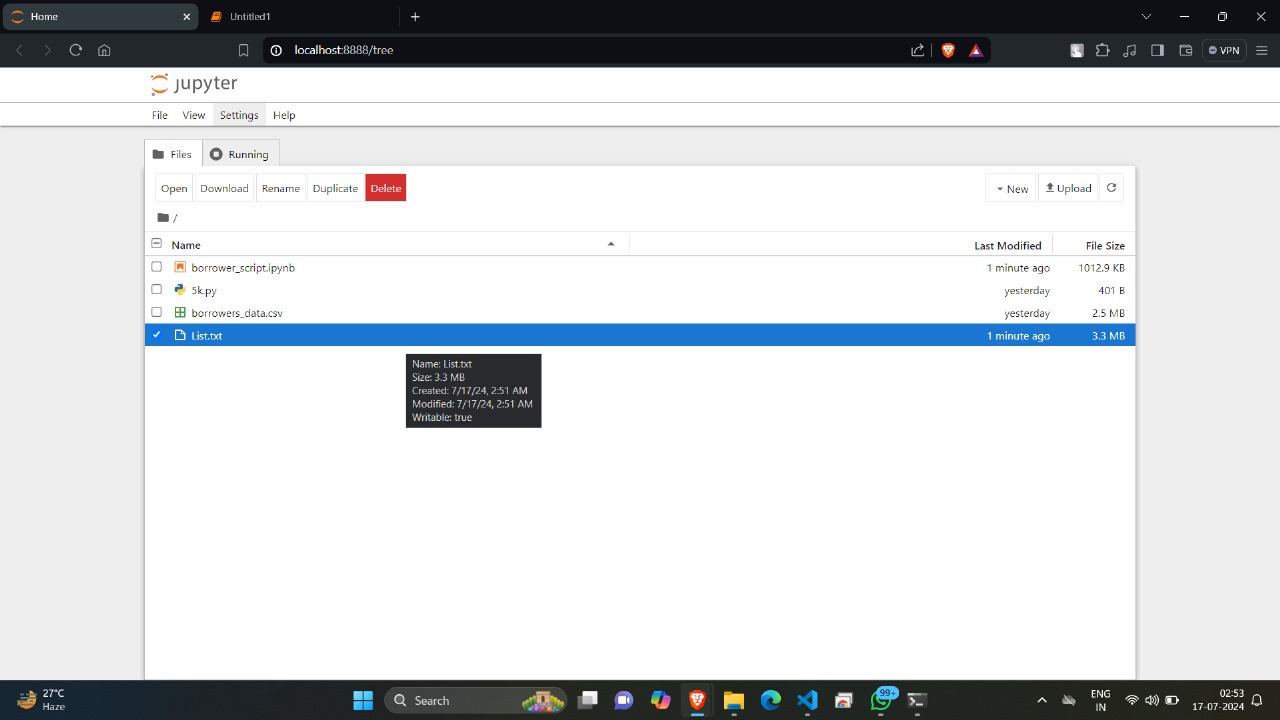
**4. Dataframe Export:**

Once your data is clean, the code prepares it for export:

List of Tuples: The script converts the Pandas DataFrame (a structured data table) into a list of tuples. This format is compatible with how data is typically stored in relational databases like SQL.



Save as TXT File: The list of tuples is then saved as a text file (e.g., "prepared\_data.txt"). This file can be easily imported into your SQL database using tools or commands specific to your SQL environment.



**5. Data Import into SQL (Separate Process):**

This ReadMe focuses on data preparation and export.

