Data Management Deliverables for Personal Expense Tracker Project

1. Cleaned and Validated Datasets

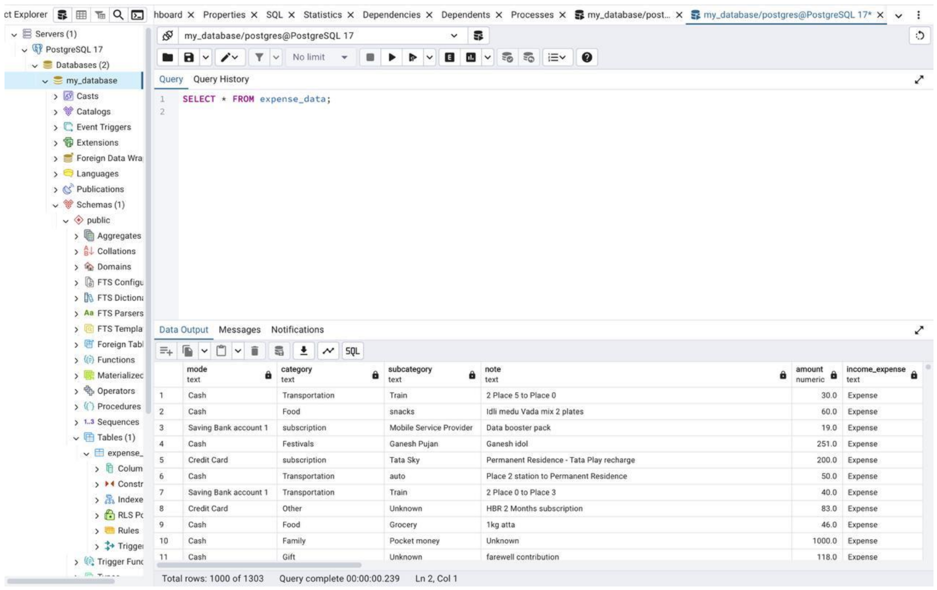
- **Initial Data**: The original dataset, `Daily Household Transactions.csv`, contained raw transaction records. This file included various transaction types (e.g., expenses, income) and metadata fields such as transaction category, subcategory, notes, amount, and mode.
- **Cleaning Process**: During data cleaning, issues such as missing values, incorrect data types, duplicates, and inconsistent categorization were identified and addressed.
    - **Duplicates**: Duplicate transactions were removed.
    - **Missing Values**: Missing values in essential columns (e.g., `amount`, `category`) were either filled or the rows were removed if essential information was missing.
    - **Data Type Conversion**: The `amount` field was converted to numeric format, and date fields were standardized to a consistent format.
    - **Categorization**: Standardized expense categories and subcategories for consistency.
- **Cleaned Dataset**: The cleaned data, `updated_data.csv`, is stored in PostgreSQL under the table `expense_data`, ready for analysis.

2. Database Management Reports

- **Database Structure**: The cleaned dataset was imported into PostgreSQL for efficient data management and querying. The database, `my_database`, contains the `expense_data` table, structured with the following fields:
    - `mode`: Transaction method (e.g., Cash, Credit Card).
    - `category`: General category of the transaction (e.g., Food, Transportation).
    - `subcategory`: Specifics within the main category (e.g., Snacks under Food).
    - `note`: Additional description or notes about the transaction.
    - `amount`: Numerical value of the transaction.
    - `income_expense`: Indicator of whether the transaction is an income or expense.
    - `currency`: The currency of the transaction amount (e.g., INR).

Queries Performed:

- **Full Table Query**: `SELECT * FROM expense_data;` - Displays all records in the `expense_data` table for review.

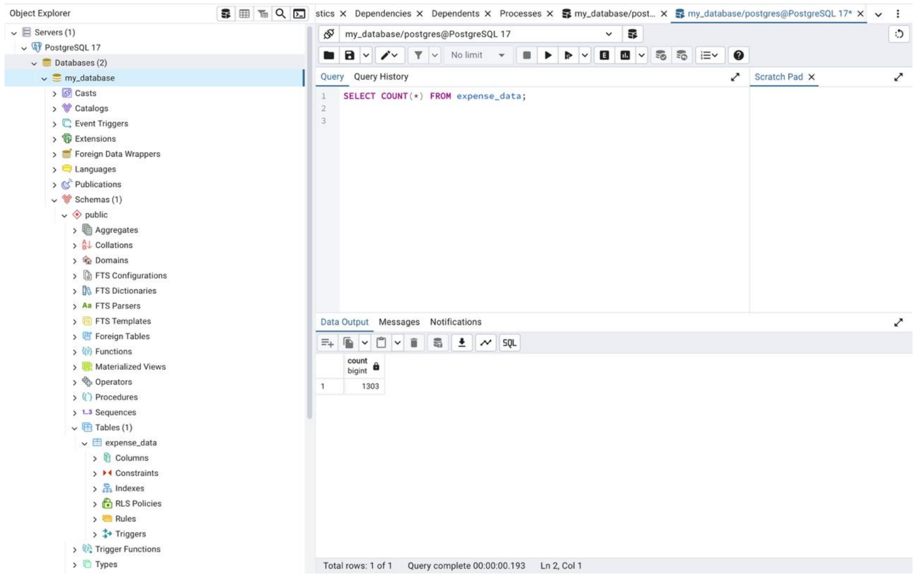


```sql
SELECT * FROM expense_data;
```

	mode text	category text	subcategory text	note text	amount numeric	income_expense text
1	Cash	Transportation	Train	2 Place 5 to Place 0	30.0	Expense
2	Cash	Food	snacks	Idli medu Vada mix 2 plates	60.0	Expense
3	Saving Bank account 1	subscription	Mobile Service Provider	Data booster pack	19.0	Expense
4	Cash	Festivals	Ganesh Pujan	Ganesh idol	251.0	Expense
5	Credit Card	subscription	Tata Sky	Permanent Residence - Tata Play recharge	200.0	Expense
6	Cash	Transportation	auto	Place 2 station to Permanent Residence	50.0	Expense
7	Saving Bank account 1	Transportation	Train	2 Place 0 to Place 3	40.0	Expense
8	Credit Card	Other	Unknown	HBR 2 Months subscription	83.0	Expense
9	Cash	Food	Grocery	1kg atta	46.0	Expense
10	Cash	Family	Pocket money	Unknown	1000.0	Expense
11	Cash	Gift	Unknown	farewell contribution	118.0	Expense

Total rows: 1000 of 1303   Query complete 00:00:00.239   Ln 2, Col 1

- **Record Count**: `SELECT COUNT(*) FROM expense_data;` - Confirms the total number of transactions in the dataset.

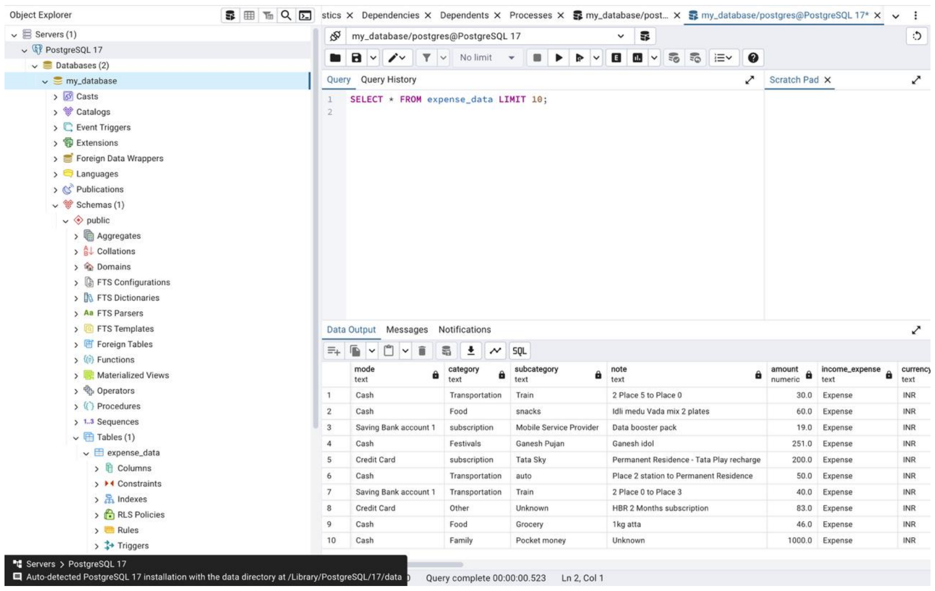


```sql
SELECT COUNT(*) FROM expense_data;
```

	count bigint
1	1303

Total rows: 1 of 1   Query complete 00:00:00.193   Ln 2, Col 1

- **Sample Query**: `SELECT * FROM expense_data LIMIT 10;` - Fetches a sample of 10 records for quick inspection



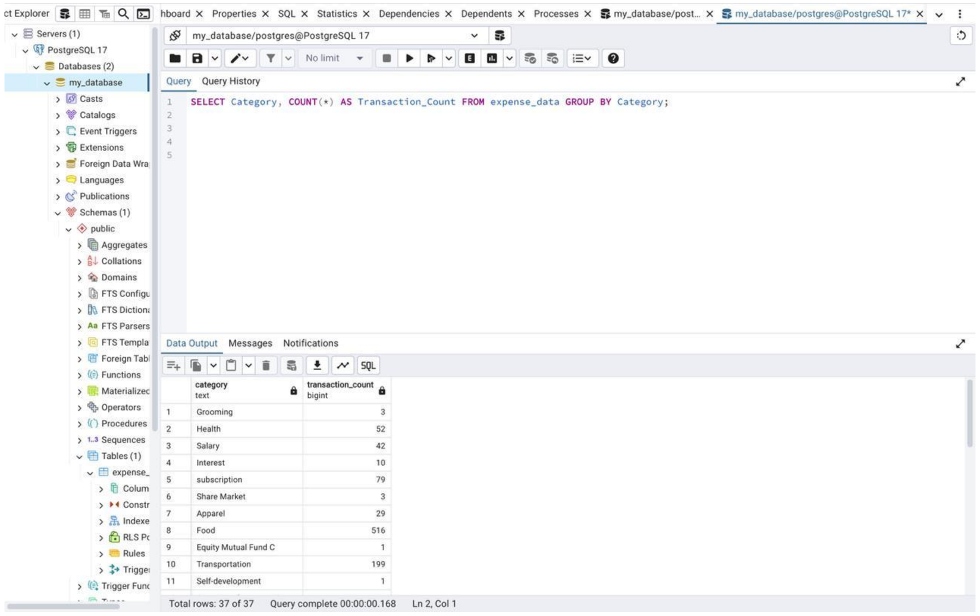
```sql
SELECT * FROM expense_data LIMIT 10;
```

	mode text	category text	subcategory text	note text	amount numeric	income_expense text	currency text
1	Cash	Transportation	Train	2 Place 5 to Place 0	30.0	Expense	INR
2	Cash	Food	snacks	Idli medu Vada mix 2 plates	60.0	Expense	INR
3	Saving Bank account 1	subscription	Mobile Service Provider	Data booster pack	19.0	Expense	INR
4	Cash	Festivals	Ganesh Pujan	Ganesh idol	251.0	Expense	INR
5	Credit Card	subscription	Tata Sky	Permanent Residence - Tata Play recharge	200.0	Expense	INR
6	Cash	Transportation	auto	Place 2 station to Permanent Residence	50.0	Expense	INR
7	Saving Bank account 1	Transportation	Train	2 Place 0 to Place 3	40.0	Expense	INR
8	Credit Card	Other	Unknown	HBR 2 Months subscription	83.0	Expense	INR
9	Cash	Food	Grocery	1kg atta	46.0	Expense	INR
10	Cash	Family	Pocket money	Unknown	1000.0	Expense	INR

Auto-detected PostgreSQL 17 installation with the data directory at /Library/PostgreSQL/17/data

Query complete 00:00:00.523 Ln 2, Col 1

- **Category-Based Transaction Count**: `SELECT category, COUNT(*) AS transaction_count FROM expense_data GROUP BY category;` - Provides a count of transactions per category.

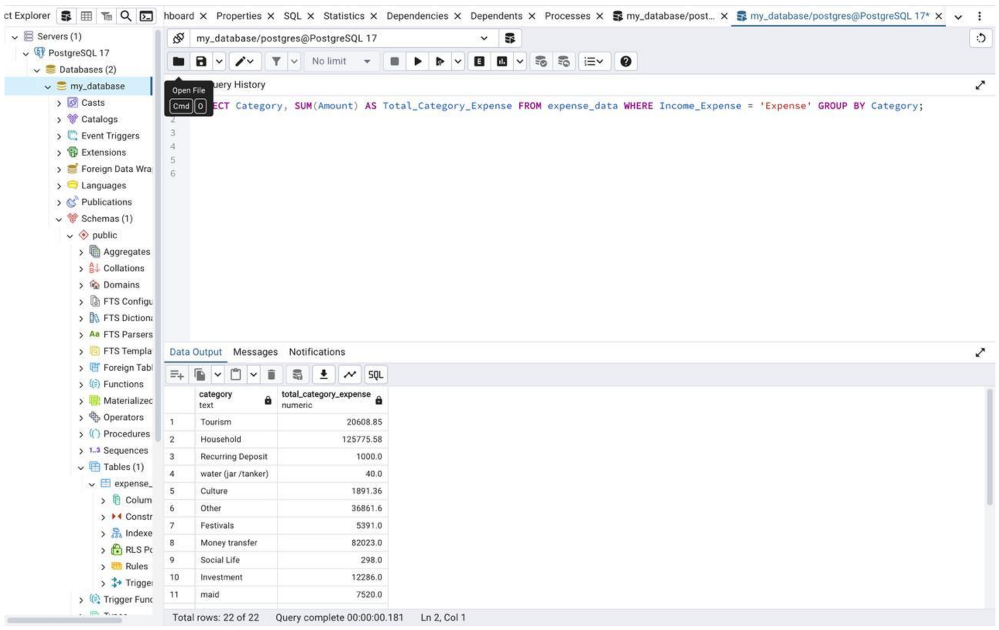


```sql
SELECT Category, COUNT(*) AS Transaction_Count FROM expense_data GROUP BY Category;
```

	category text	transaction_count bigint
1	Grooming	3
2	Health	52
3	Salary	42
4	Interest	10
5	subscription	79
6	Share Market	3
7	Apparel	29
8	Food	516
9	Equity Mutual Fund C	1
10	Transportation	199
11	Self-development	1

Total rows: 37 of 37 Query complete 00:00:00.168 Ln 2, Col 1

- **Category-Based Total Expense**: `SELECT category, SUM(amount) AS total_category_expense FROM expense_data WHERE income_expense = 'Expense' GROUP BY category;` - Sums up expenses per category



```
ECT Category, SUM(Amount) AS Total_Category_Expense FROM expense_data WHERE Income_Expense = 'Expense' GROUP BY Category;
```

	category text	total_category_expense numeric
1	Tourism	20608.85
2	Household	125775.58
3	Recurring Deposit	1000.0
4	water (jar /tanker)	40.0
5	Culture	1891.36
6	Other	36861.6
7	Festivals	5391.0
8	Money transfer	82023.0
9	Social Life	298.0
10	Investment	12286.0
11	maid	7520.0

Total rows: 22 of 22   Query complete 00:00:00.181   Ln 2, Col 1

3. Data Quality Assessments

- **Data Completeness**: Verified that all essential columns (`amount`, `category`, `income_expense`) contain data with minimal missing values after cleaning.
- **Data Consistency**: Ensured that categories and subcategories are standardized, avoiding inconsistencies in spelling or naming conventions.
- **Accuracy Checks**: Cross-validated transaction amounts and categorization based on descriptions in the `note` field.
- **Integrity in PostgreSQL**: Ensured that the data uploaded to PostgreSQL maintained integrity by using constraints (e.g., numeric validation on `amount`) and setting up data types accordingly.

4. Data Collection Methodologies

- **Data Sources**: Transaction data was collected from user inputs and uploads, encompassing various daily transactions.
- **Data Entry Standards**: Established guidelines for entering data to ensure consistency. Categories were predefined to minimize manual entry errors, and descriptive notes were encouraged for clarity.
- **Automation Potential**: Data was loaded into PostgreSQL, allowing for automated data updates and additional processing if needed for future transactions.