PERSONAL EXPENSE TRACKER

Ankita Taneja
GUVI – HCL
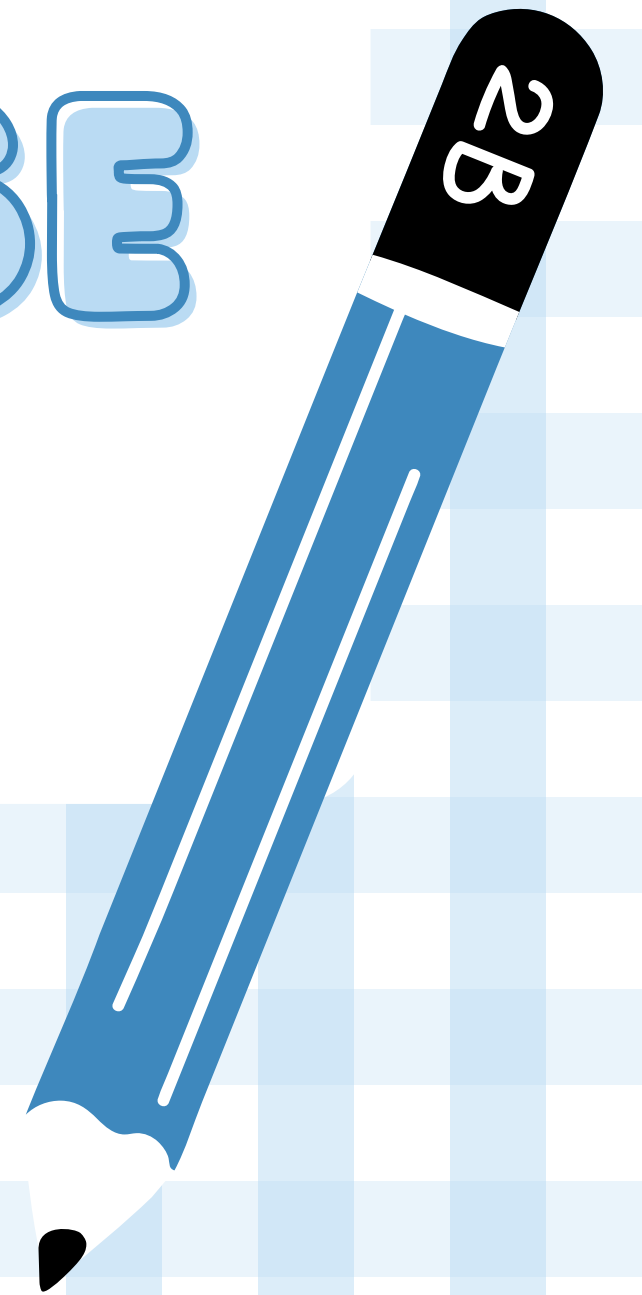
TABLE OF CONTENTS

TECHNOLOGY USED

1 Python – Anaconda Navigator Jupyter

2 SQL – Intergrated Commands

3 Streamlit

4 Command Prompt

5 Canva – Presentation

GENERATE DATA - FAKER LIBRARY

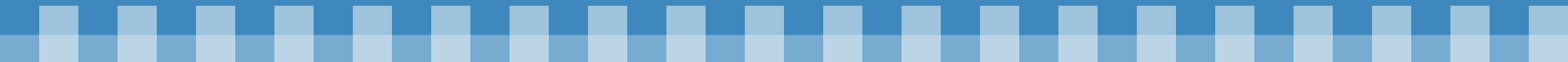**Total Balance of Savings:** $1,11,001

- **Data Simulation**: Use the Faker library to generate a realistic dataset that depicts a person's expense throughout the month. Create 12 different tables for each month.
- *Faker* is a Python package that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you.
- To **install**, Open CMD Prompt:
  Install with pip: pip install Faker
- **Import** faker library:
  from faker import Faker
- **Call**:
  fake.name()
  fake.text()
  fake.address()

- **Faker → Generate Data → DB Creation → Insert into SQL Database → Run SQL Queries → Analyze in Python → Visualize in Streamlit → Final Dashboard**

SQL QUERIES

- 1. What is the total amount spent in each category?
- 2. What is the total amount spent using each payment mode?
- 3. What is the total cashback received across all transactions?
- 4. Which are the top 5 most expensive categories in terms of spending?
- 5. How much was spent on transportation using different payment modes?
- 6. Which transactions resulted in cashback?
- 7. What is the total spending in each month of the year?
- 8. Which months have the highest spending in categories like "Travel," "Entertainment," or "Gifts"?
- 9. Are there any recurring expenses that occur during specific months of the year (e.g., insurance premiums, property taxes)?
- 10. How much cashback or rewards were earned in each month?
- 11. How has your overall spending changed over time (e.g., increasing, decreasing, remaining stable)?
- 12. What are the typical costs associated with different types of travel (e.g., flights, accommodation, transportation)?
- 13. Are there any patterns in grocery spending (e.g., higher spending on weekends, increased spending during specific seasons)?
- 14. Define High and Low Priority Categories
- 15. Which category contributes the highest percentage of the total spending?

```sql
-- 1. Total amount spent in each category
SELECT category, SUM(amount_paid) AS total_spent
FROM expenses
GROUP BY category
ORDER BY total_spent DESC;


-- 2. Total amount spent using each payment mode
SELECT payment_mode, SUM(amount_paid) AS total_spent
FROM expenses
GROUP BY payment_mode;


-- 3. Total cashback received across all transactions
SELECT SUM(cashback) AS total_cashback
FROM expenses;


-- 4. Top 5 most expensive categories in terms of spending
SELECT category, SUM(amount_paid) AS total_spent
FROM expenses
GROUP BY category
ORDER BY total_spent DESC
LIMIT 5;


-- 5. Amount spent on transportation using different payment modes
SELECT payment_mode, SUM(amount_paid) AS total_spent
FROM expenses
WHERE category = 'Transportation'
GROUP BY payment_mode;
```

```sql
-- 6. Transactions that resulted in cashback
SELECT *
FROM expenses
WHERE cashback > 0;


-- 7. Total spending in each month of the year
SELECT MONTH(date) AS month, SUM(amount_paid) AS total_spent
FROM expenses
GROUP BY MONTH(date)
ORDER BY month;


-- 8. Highest spending months for Travel, Entertainment, or Gifts
SELECT MONTH(date) AS month, category, SUM(amount_paid) AS total_spent
FROM expenses
WHERE category IN ('Travel', 'Entertainment', 'Gifts')
GROUP BY MONTH(date), category
ORDER BY total_spent DESC;


-- 9. Recurring expenses in specific months (e.g., insurance, property taxes)
SELECT MONTH(date) AS month, category, COUNT(*) AS frequency
FROM expenses
WHERE category IN ('Insurance', 'Bills')
GROUP BY MONTH(date), category
ORDER BY frequency DESC;


-- 10. Cashback or rewards earned in each month
SELECT MONTH(date) AS month, SUM(cashback) AS total_cashback
FROM expenses
GROUP BY MONTH(date)
ORDER BY month;
```

```sql
-- 11. Overall spending trend (monthly)
SELECT MONTH(date) AS month, SUM(amount_paid) AS total_spent
FROM expenses
GROUP BY MONTH(date)
ORDER BY month;


-- 12. Costs associated with different travel types (assuming description field)
SELECT description, AVG(amount_paid) AS avg_cost
FROM expenses
WHERE category = 'Travel'
GROUP BY description
ORDER BY avg_cost DESC;


-- 13. Grocery spending patterns (weekends vs weekdays)
SELECT
    CASE
        WHEN DAYOFWEEK(date) IN (1, 7) THEN 'Weekend'
        ELSE 'Weekday'
    END AS day_type,
    AVG(amount_paid) AS avg_spent
FROM expenses
WHERE category = 'Groceries'
GROUP BY day_type;


-- 14. High vs Low Priority Categories
SELECT category, SUM(amount_paid) AS total_spent,
    CASE
        WHEN category IN ('Bills', 'Insurance', 'Health') THEN 'High Priority'
        ELSE 'Low Priority'
    END AS priority
```

```sql
    END AS priority
FROM expenses
GROUP BY category, priority
ORDER BY total_spent DESC;


-- 15. Category contributing highest percentage of total spending
SELECT category,
        SUM(amount_paid) AS category_spent,
        (SUM(amount_paid) / (SELECT SUM(amount_paid) FROM expenses) * 100) AS percent_total
FROM expenses
GROUP BY category
ORDER BY percent_total DESC
LIMIT 1;
```
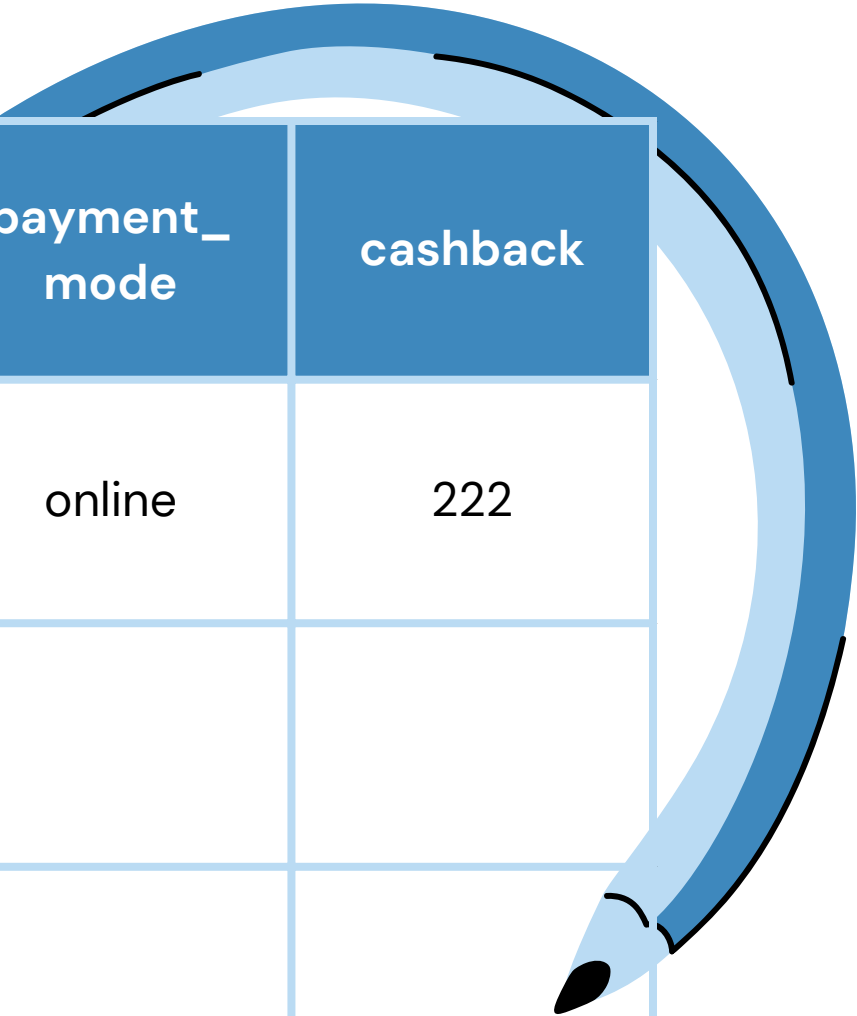
EXPENSES TRACKER

Month: February

Budget: $15,000

6 Data Set Columns

- id
- date
- category
- amount
- payment_mode
- cashback

id	date	category	amount	payment_mode	cashback
1	xxxx-xx-xx	Shopping	854	online	222

STREAMLIT

- Streamlit is an open-source Python library that makes it easy to create and share custom web apps for machine learning and data science.
- By using Streamlit you can quickly build and deploy powerful data applications.
- For more information about the open-source library, see the Streamlit Library documentation.
- Streamlit lets you build dashboards, generate reports, or create chat apps.

STREAMLIT APP FEATURES

1. FILTER THE DATA BY MONTH, PAYMENT MODE AND CATEGORY
2. INTERACTIVE CHARTS
3. DASHBOARD KPI'S
4. SQL DRIVEN INSIGHTS

🔍 **Filters**

Filter by Category:

All ⌄

Filter by Payment Mode:

All ⌄

Filter by Month:

All ⌄

COMMAND PROMPT


```
Command Prompt - streamlit                    +    ⌄

C:\Users\LENOVO\GUVI\Project 1 Personal Expenses>init_db.py

C:\Users\LENOVO\GUVI\Project 1 Personal Expenses>
[main 2025-12-03T13:27:35.804Z] update#setState idle
[main 2025-12-03T13:28:05.815Z] update#setState checking for updates
[main 2025-12-03T13:28:05.947Z] update#setState idle

C:\Users\LENOVO\GUVI\Project 1 Personal Expenses>data_simulation.py

C:\Users\LENOVO\GUVI\Project 1 Personal Expenses>

C:\Users\LENOVO\GUVI\Project 1 Personal Expenses>streamlit run atreamlit_app.py
Usage: streamlit run [OPTIONS] [TARGET] [ARGS]...
Try 'streamlit run --help' for help.

Error: Invalid value: File does not exist: atreamlit_app.py

C:\Users\LENOVO\GUVI\Project 1 Personal Expenses>streamlit run streamlit_app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://10.23.134.187:8501

2025-12-03 19:01:52.028 Please replace `use_container_width` with `width`.

`use_container_width` will be removed after 2025-12-31.

For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
```

EXPENSE TRACKER

Total Balance of Savings: $39,102

5 TABS

- DASHBOARD
- SQL INSIGHTS
- ADD EXPENSE
- RAW DATA
- DOWNLOAD

💰 **ADVANCED PERSONAL EXPENSE DASHBOARD**

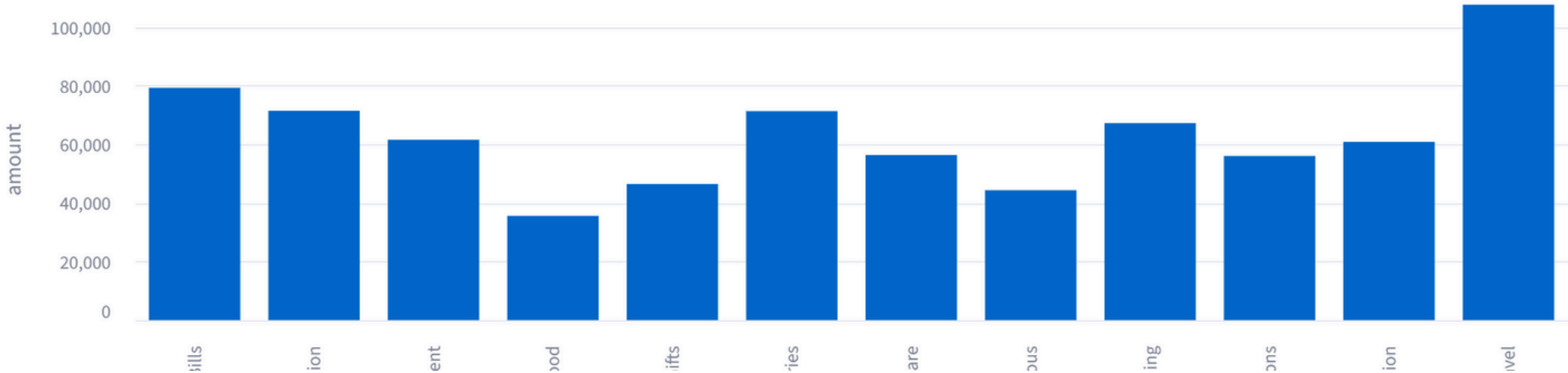Analyze your expenses with SQL-powered insights, filters, charts & KPIs.

| 📊 Dashboard | 📇 SQL Insights | ➕ Add Expense | 📄 Raw Data | ⬇ Download |

📄 **Raw Expense Data**

	id	date	category	amount	payment_mode	cashback
0	1	2024-01-26	Shopping	446.36	Online	0
1	2	2024-01-20	Gifts	111.56	Cash	0
2	3	2024-01-05	Food	247.36	Cash	0
3	4	2024-01-06	Transportation	410.46	Online	0
4	5	2024-01-01	Food	124.04	Online	5.99
5	6	2024-01-03	Gifts	327.84	Cash	0
6	7	2024-01-22	Miscellaneous	565.46	Cash	0
7	8	2024-01-09	Groceries	756.33	Cash	0
8	9	2024-01-12	Shopping	739.48	Cash	0
9	10	2024-01-08	Miscellaneous	1000.98	Online	0

SQL INSIGHTS

💰 ADVANCED PERSONAL EXPENSE DASHBOARD

Analyze your expenses with SQL-powered insights, filters, charts & KPIs.

| 📊 Dashboard | 🔢 SQL Insights | ➕ Add Expense | 📄 Raw Data | ⬇️ Download |

🧮 SQL Analytics

Select a query to run:

Spending by payment mode ⌄

Run Query

	payment_mode	total
0	Cash	402261.74
1	Online	355888.5

💰 **ADVANCED PERSONAL EXPENSE DASHBOARD**

Analyze your expenses with SQL-powered insights, filters, charts & KPIs.

📊 Dashboard | 📋 SQL Insights | ➕ Add Expense | 📄 Raw Data | ⬇️ Download

➕ **Add New Expense** 🔗

Date

2025/12/03

Category

Payment Mode

Cash ⌄

Description

Amount

0.00 − +

💰 **ADVANCED PERSONAL EXPENSE DASHBOARD**

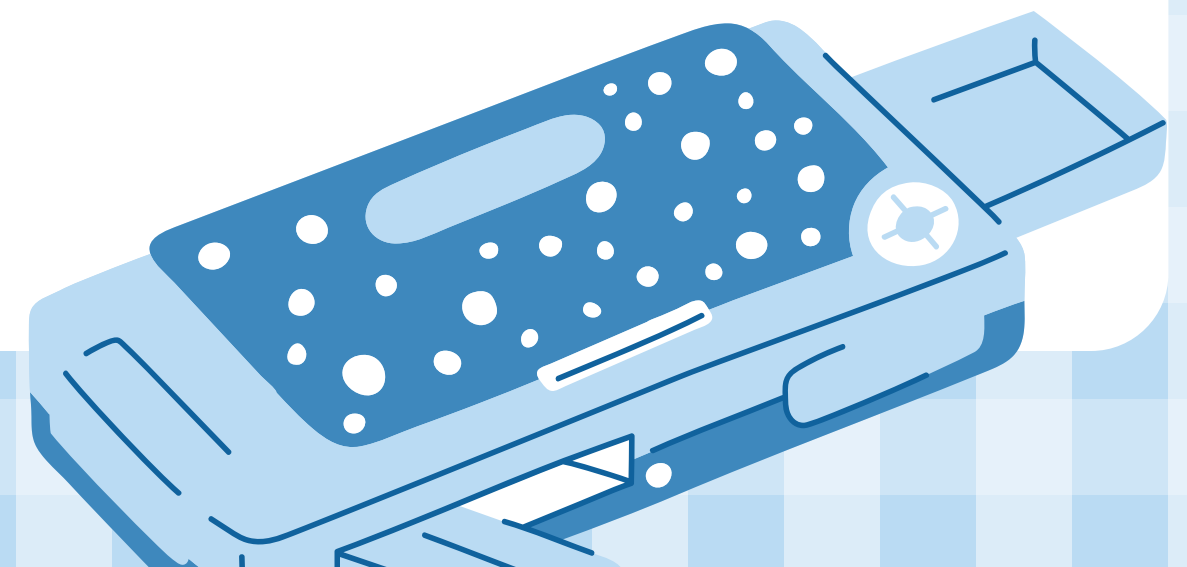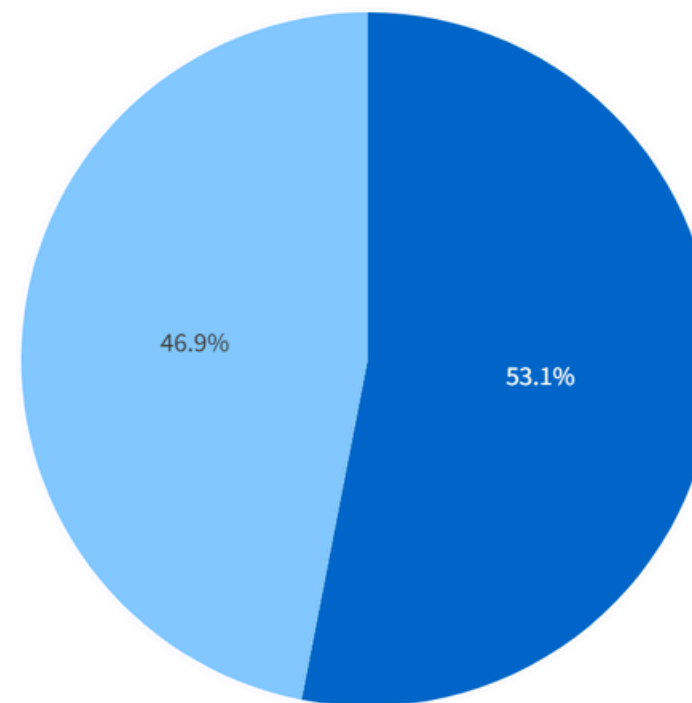Analyze your expenses with SQL-powered insights, filters, charts & KPIs.

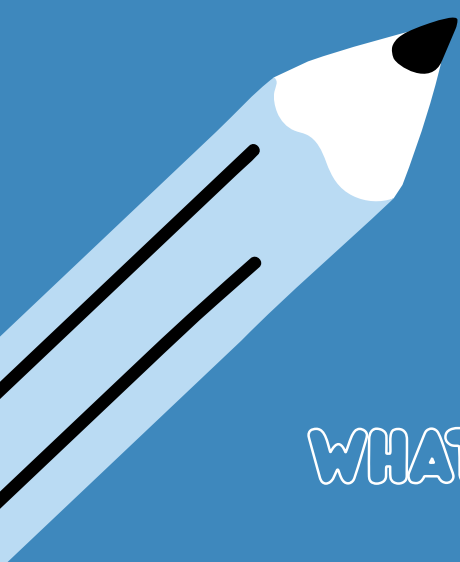📊 Dashboard | 📋 SQL Insights | ➕ Add Expense | 📄 Raw Data | ⬇️ Download

⬇️ **Download Your Data**

Download CSV

CONCLUSION

WHAT THE PROJECT SOLVES :

PROVIDES A COMPLETE SIMULATION OF PERSONAL EXPENSE TRACKING USING SYNTHETIC (FAKER-GENERATED) DATA.

HELPS USERS UNDERSTAND THEIR SPENDING HABITS, IDENTIFY TRENDS, AND MONITOR FINANCIAL DISCIPLINE.

OFFERS A STRUCTURED SYSTEM TO ANALYZE EXPENSES USING SQL AND VISUALIZE INSIGHTS THROUGH AN INTERACTIVE STREAMLIT DASHBOARD.