Royal University of Phnom Penh
Master of Data Science and Engineering Mr. CHHOY Sokunthaneth
Semester 3 - Data Analytics Course Mr. VEAN Viney

Peer-to-Peer Loan System

Final Report

1. Objective

To design a peer-to-loan system that allows people in need of cash to quickly and conveniently obtain loan, and people in excess of cash to optionally have another way to invest their surplus. By using peer to peer model, it is possible to reduce the risk level of investors when giving out loan to borrowers. Investors can diversify their loan portfolio and see borrower rating before deciding when to give loan. Borrowers can see many options of loan offering before committing to their preferred loan.

2. Conceptual Model

User Registration and Profile Creation:

- Users register on the platform and create profiles.
- User information such as name, email, and other relevant details are stored in the User table.

Loan Request Creation:

- Borrowers initiate loan requests by submitting details such as the amount requested and any required documents.
- Each loan request is assigned a unique RequestID and linked to the borrower's UserID in the LoanRequest table.

Investor Participation:

- Investors browse available loan requests and decide which ones to fund.
- Investors can contribute fully or partially to loan requests and specify the interest rate they are willing to offer.

- Each investor's participation in a loan request is recorded as a loan submission in the LoanSubmission table, linking the investor's UserID and the amount invested.

Loan Request Selection:

- Borrowers review the loan submissions received for their requests.
- Borrowers may choose to accept or reject loan submissions based on factors such as interest rate, total amount funded, and investor reputation.

Funding and Loan Approval:

- Once a borrower accepts a loan submission, the loan request is funded.
- The status of the loan request changes to "Funded" in the LoanRequest table.
- The loan submission status also changes to reflect its acceptance.

Payment Processing:

- Borrowers receive the funded amount and start repaying the loan according to the agreed-upon terms.
- Payments made by borrowers are recorded in the Payment table, linking to the corresponding loan submission.
- Investors receive payments as borrowers repay their loans, and their account balances are updated accordingly.

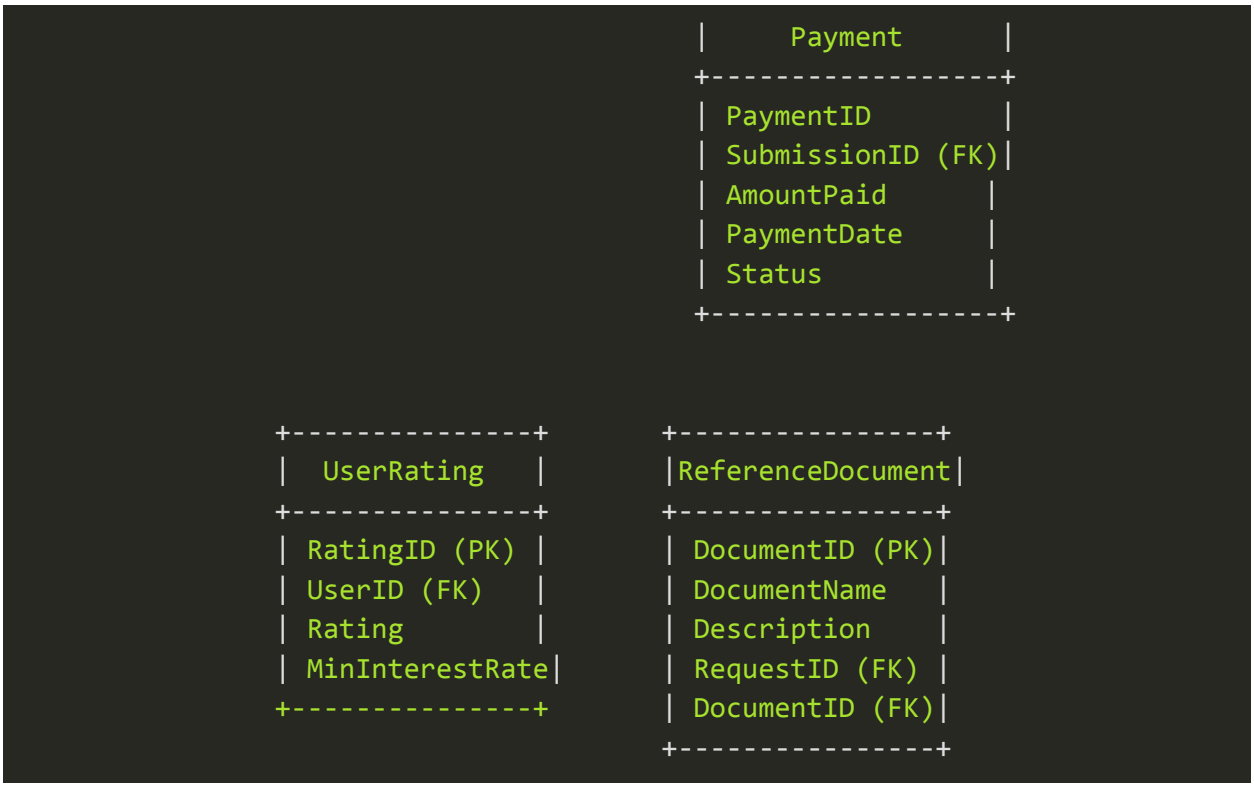
User Rating and Account Balance Management:

- User ratings are periodically updated based on factors such as repayment history, interaction with the platform, and other performance metrics.
- User ratings are stored in the UserRating table, linking to the respective UserID.
- Account balances of users are tracked in the AccountBalance table, ensuring accurate funds management for both investors and borrowers.

Document Submission and Verification:

- Borrowers may be required to submit reference documents along with their loan requests for verification purposes.
- The required documents are stored in the ReferenceDocument table, linked to the corresponding loan request.
- Platform administrators or automated systems may verify the submitted documents to ensure compliance with regulations and platform policies.

3. Relational Data Model



4. Data Collection

Since the peer-to-peer loan dataset is limited, we have decided to seed dummy data. The dummy data is constructed using Python, with an overall 175,000 records.

```python
import random
import psycopg2
from faker import Faker

# Create a Faker instance
fake = Faker()

# Establish a connection to the database
conn = psycopg2.connect(
    dbname='p2p-loan',
```

```
    user='admin',
    password='adminxx2',
    host='localhost',
    port='5434'
)

# Create a cursor object
cur = conn.cursor()

# generate target
N_USER = 1000
LOAN_REQUEST = 50000
LOAN_SUBMISSION = 100000
PAYMENT = 250000


# generate, insert into app_user
for _ in range(N_USER):
    fake_name = fake.name()
    fake_email = fake.email()
    fake_national_id = fake.ssn()
    fake_passport_id = fake.passport_number()
    fake_place_of_birth = fake.address()
    fake_current_address = fake.address()
    fake_phone_number = fake.phone_number()
    fake_date_of_birth = fake.date_of_birth()
    fake_occupation = fake.job()
    fake_organization = fake.company()
    fake_naitonality = fake.country()
    fake_created_at = fake.date_time_this_decade()

        cur.execute("INSERT INTO app_user (name, email, national_id,
passport_id,      place_of_birth,      current_address,      phone_number,
date_of_birth, occupation, organization, nationality, created_at) VALUES
(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)",
                                (fake_name, fake_email, fake_national_id,
fake_passport_id,       fake_place_of_birth,       fake_current_address,
```

```
fake_phone_number,          fake_date_of_birth,          fake_occupation,
fake_organization, fake_naitonality, fake_created_at))


# generate, insert into account_balance
fake_user_id = 0
for _ in range(N_USER):
    fake_user_id += 1
    fake_balance_amount = fake.random_int(min=20, max=50000)
    fake_currency = "USD"

     cur.execute("INSERT INTO account_balance (user_id, balance_amount,
currency) VALUES (%s, %s, %s)",
                (fake_user_id, fake_balance_amount, fake_currency))


# generate, insert into user_rating
fake_user_id = 0
for _ in range(N_USER):
    fake_user_id += 1
    fake_rating = fake.random_int(min=0, max=5)
    fake_min_interest_rate = round(random.uniform(2, 15), 2)
    fake_created_at = fake.date_time_this_decade()

           cur.execute("INSERT   INTO   user_rating   (user_id,   rating,
min_interest_rate, created_at) VALUES (%s, %s, %s, %s)",
                       (fake_user_id, fake_rating, fake_min_interest_rate,
fake_created_at))


# generate, insert into loan_request
for _ in range(LOAN_REQUEST):
    fake_user_id = fake.random_int(min=1, max=N_USER)
    fake_amount_requested = fake.random_int(min=50, max=20000)
    fake_ask_interest_rate = round(random.uniform(2, 12), 2)
    fake_status = fake.random_element(
        elements=('submitted', 'started', 'finished', 'rejected'))
```

```
    fake_created_at = fake.date_time_this_decade()

      cur.execute("INSERT INTO loan_request (user_id, amount_requested,
ask_interest_rate, status, created_at) VALUES (%s, %s, %s, %s, %s)",
                                    (fake_user_id,  fake_amount_requested,
fake_ask_interest_rate, fake_status, fake_created_at))


# generate, insert into loan_submission
for _ in range(LOAN_SUBMISSION):
    fake_request_id = fake.random_int(min=1, max=LOAN_REQUEST)
    fake_investor_id = fake.random_int(min=1, max=N_USER)
    fake_amount_invested = fake.random_int(min=50, max=20000)
    fake_bid_interest_rate = round(random.uniform(7, 20), 2)
    fake_status = fake.random_element(
        elements=('submitted', 'accepted', 'rejected'))
    fake_created_at = fake.date_time_this_decade()

     cur.execute("INSERT INTO loan_submission (request_id, investor_id,
amount_invested, bid_interest_rate, status, created_at) VALUES (%s, %s,
%s, %s, %s, %s)",
                                      (fake_request_id,  fake_investor_id,
fake_amount_invested,        fake_bid_interest_rate,        fake_status,
fake_created_at))


# generate, insert into payment
for _ in range(PAYMENT):
    fake_submission_id = fake.random_int(min=1, max=LOAN_SUBMISSION)
    fake_amount_paid = fake.random_int(min=20, max=10000)
    fake_payment_date = fake.date_time_this_decade()
    fake_status = fake.random_element(
        elements=('submitted', 'accepted', 'rejected'))
    fake_created_at = fake.date_time_this_decade()

       cur.execute("INSERT  INTO  payment  (submission_id,  amount_paid,
payment_date, status, created_at) VALUES (%s, %s, %s, %s, %s)",
```

```
                                (fake_submission_id,  fake_amount_paid,
fake_payment_date, fake_status, fake_created_at))


# generate, insert into reference_document
for _ in range(LOAN_REQUEST*2):
    fake_document_name = fake.random_element(
            elements=('property-title', 'vehicle-title', 'national_id',
'driver_license'))
    fake_description = fake.sentence()
    fake_request_id = fake.random_int(min=1, max=LOAN_REQUEST)
    fake_created_at = fake.date_time_this_decade()

          cur.execute("INSERT  INTO  reference_document  (document_name,
description, request_id, created_at) VALUES (%s, %s, %s, %s)",
                  (fake_document_name, fake_description, fake_request_id,
fake_created_at))


# Commit the changes and close the connection
conn.commit()
cur.close()
conn.close()
```

5. Sensitive Data and Data Quality Issues

- PII (Personally Identifiable Information):
  - Name
  - Email
  - NationID
- CFI (Consumer Financial Information):
  - BalanceID
  - BalanceAmount
  - AmountRequested

    - AmountInvested
    - InterestRate
    - AmountPaid
    - MinInterestRate
    - PaymentDate
- CPNI (Customer Proprietary Network Information):
    - N/A
- PHI (Protected Health Information):
    - N/A

Data quality challenges:

- LoanRequest: the user might input too low or too high the amount requested. Thus it should range input should be introduced.
- LoanSubmission: the user might input too low or too high-interest rates. Thus it should range input should be introduced.
- ReferenceDocument: the user might input random reference documents if not verified. Thus staff dedicated to reviewing reference documents should be introduced.
- Payment: the user might scan payment QR directly to borrowers/investors. Thus deep payment should be introduced.

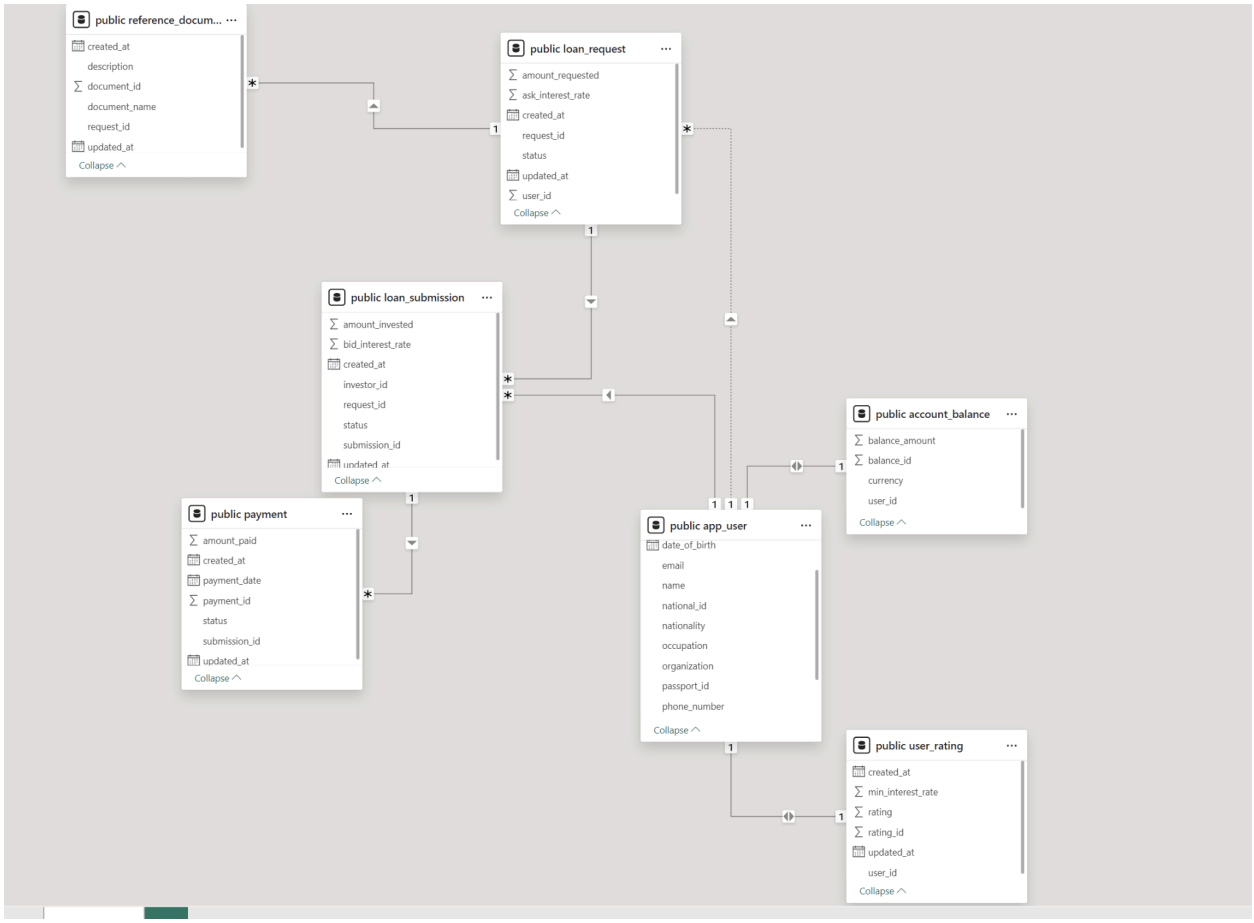
6. Analytics Questions

From the peer-to-peer loan system, we would like to find insights for the following

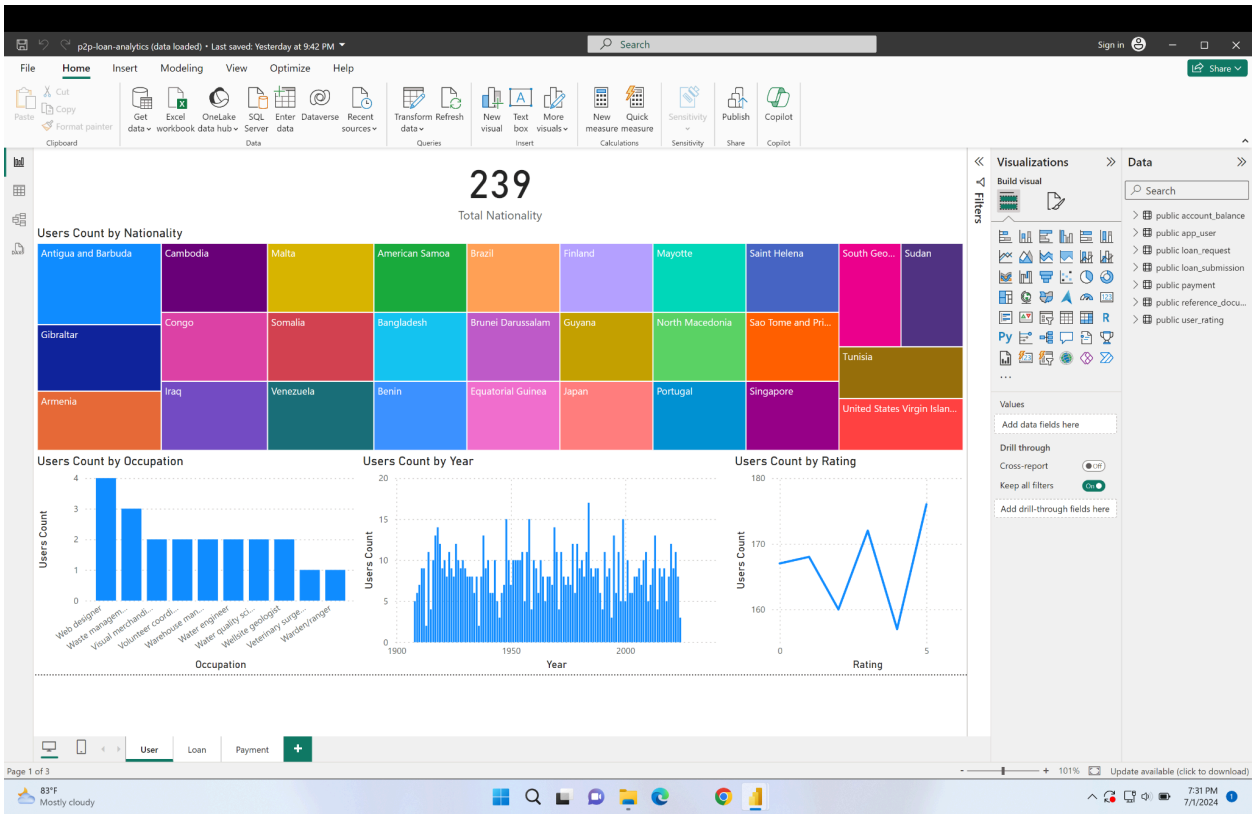
- What is the demographic of users?
- What is the most popular cause of borrowing?
- What is the average user rating that receive loan?
- What is the distribution of borrower / investors age?
- What is profession of the majority of borrowers / investors?
- What is the yearly borrowing / investing amount?
- What is the average user rating?
- What is the average asking interest rate / bidding interest rate?
- What is the average payment status?7. PowerBI Implementation

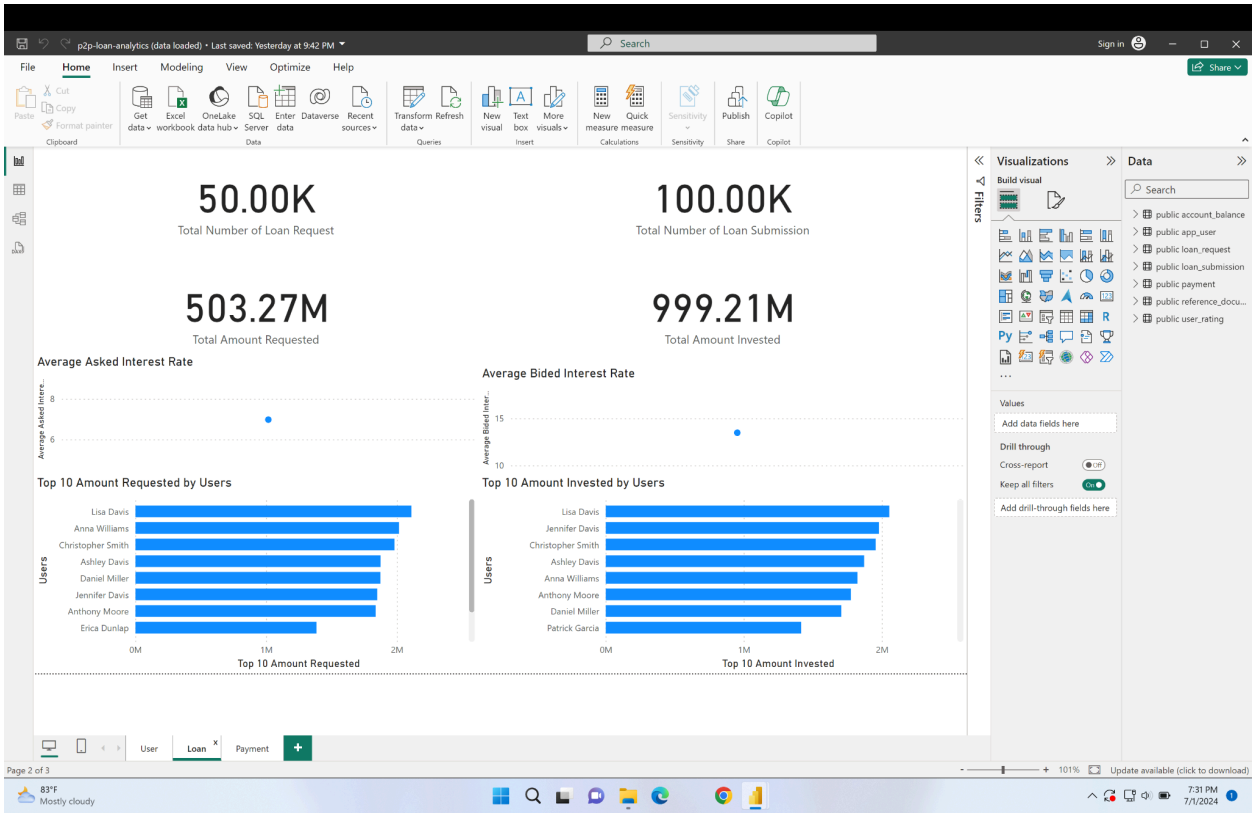
7. Server Implementation

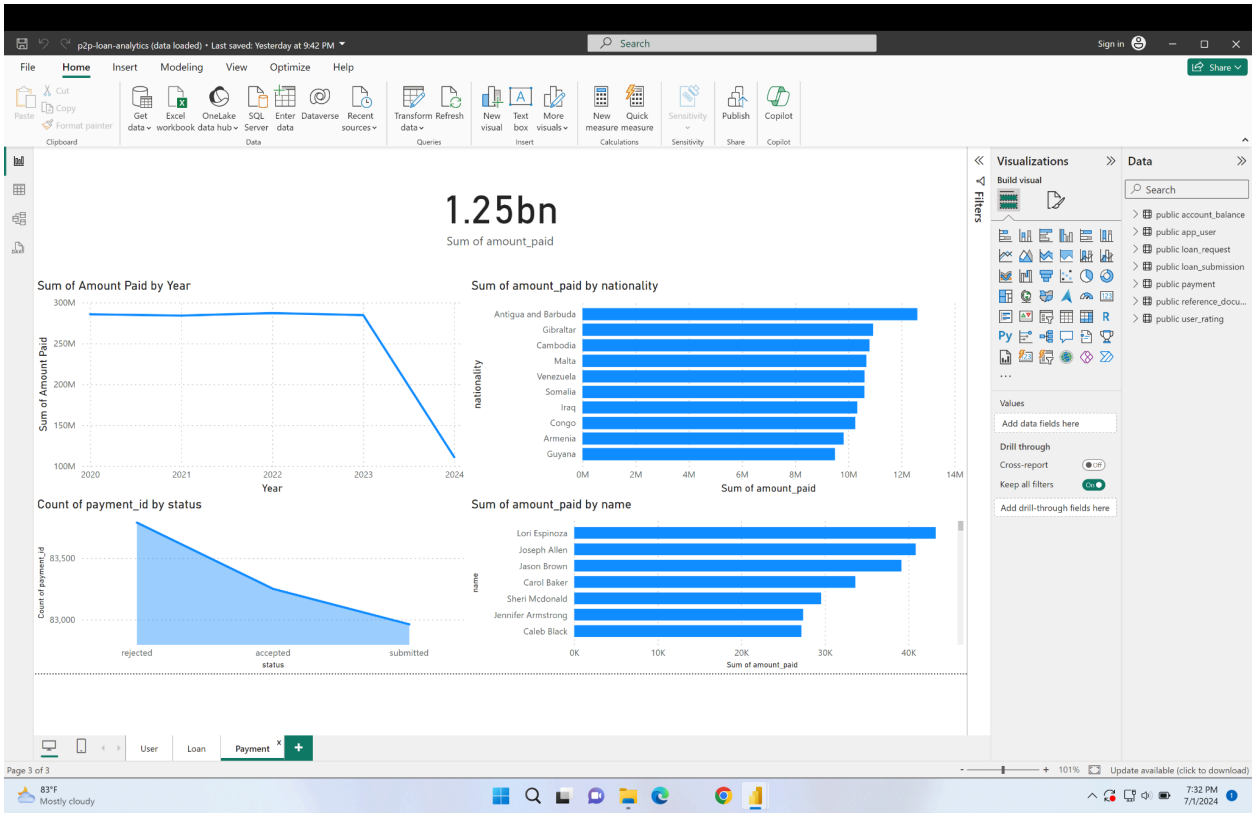
The dataset is stored in PostgreSQL then loaded to PowerBI for analysis.



8. PowerBI Implementation




50.00K
Total Number of Loan Request
100.00K
Total Number of Loan Submission
503.27M
Total Amount Requested
999.21M
Total Amount Invested
Average Asked Interest Rate
Average Bided Interest Rate
Top 10 Amount Requested by Users
Lisa Davis
Anna Williams
Christopher Smith
Ashley Davis
Daniel Miller
Jennifer Davis
Anthony Moore
Erica Dunlap
Users
0M
1M
2M
Top 10 Amount Requested
Top 10 Amount Invested by Users
Lisa Davis
Jennifer Davis
Christopher Smith
Ashley Davis
Anna Williams
Anthony Moore
Daniel Miller
Patrick Garcia
Users
0M
1M
2M
Top 10 Amount Invested
Visualizations
Data
Filters
User
Loan
Payment



9. Conclusion

Peer-to-peer loan system offer a promising way for borrower to find cash quickly, and investor a convenient way to invest their saving, given the risk is properly managed for investors. The analytics allow us to visualize how this system might work in production.