COP 5725: Database Management Systems

Project Deliverable 1

Group 25

Shangde Gao - gao.shangde@ufl.edu
Srija Gurijala - srijagurijala@ufl.edu
Dimitrios Melissourgos - dmelissourgos@ufl.edu
Andrei Sura - asura@ufl.edu
Mukul Yadav - mchand.yadav@ufl.edu

Contents

Czech Bank Financial Data Analysis and Demographics

Motivation

Banks relentlessly collect data from their customers, in order to both better serve them and protect themselves from risky loans. They are in need of good software tools in order to analyze the data and adjust their strategies.

We are developing a web application with the purpose of helping the bank personnel to make decisions by analyzing the historical transaction data and by showing trends in the bank's customers history. Another use case for the historical data trends is to help the marketing departments target their efforts based on the demographics and geographical locations of the best performing and worst performing categories.

Our team was able to find a data set which focuses on the economic transactions of a bank and its customers from the Czech Republic from 1995 until 1999. Among other information it includes the characteristics of the bank account and the applicant, their previous loans and the status (if any), the locality of the customers and various other such details as mentioned in the 'Data Description' section. The source of our data is the website https://data.world. It is further cited in detail under section 'Data Source'.

Main Functions

Banks offer their services to several different customers, with different characteristics and different history. Additionally, customers often require loans and credit from the banks, which imposes a risk on them. In order for the bank to decide if they should provide the loan to a customer or not, they need to utilize the data they have collected on that customer, their demographics, etc. Our application's main function will be to use the acquired data that is stored in a database, in order to perform queries on them and show trends on the bank's customers and the market of Czech Republic in general. The graphs produced by these queries could benefit the bank staff make decisions and accept or deny some of its services to people.

Every bank officer will need to register their information into the system and create an account. Then, they will be able to login and have access on the database and the queries available in our software.

They will be able to see statistics, general trends and the information of individual customers. An example of a general statistic that can benefit the bank is to check the location of most of its existing customers in order to decide where their new physical store should be built. Finally, the software will present graphs showing trends of customers over the years, for example, the increase/decrease in the number of young vs older customers applying for loans. It will show the geographical locations of the customers, the percentage of the types of customers the bank currently has and so on.

Relational Database Advantages

The utilization of a relational database is useful to our project, since the amount of data is significant (i.e. more than one million records). The row/column count for each file type is listed below:

File	**Rows**	**Columns**
account	4501	4
card	893	4
client	5370	3
disp	5370	4
district	78	16
loan	683	7
order	6,472	6
trans	1,056,321	10
TOTAL	**1,079,688**	

The summarization and grouping queries will benefit from the database indexes which will connect the tables based on the **account_id** column present in six out of eight tables, among other common columns. Given that the district file contains unique names we will be able to add another source of information (containing district area and geographical coordinates) without modifying the existing table - this is the logical independence factor in action helping the development of the software solution.

Data Description

The dataset we have chosen for our project is a financial dataset of a bank in Czech Republic. The data has been collected from 1995 until 1999. There are 8 tables in our data set: Account, Card, Client, Disposition, District, Loan, Order and Transaction. The explanation of our tables is given below.

Account

The Account table holds basic information on the bank account of the client. It has the following columns:

account_id: This is the identification of the account and the primary key of this table.

district_id: The location of the branch that issued the account.

frequency: The frequency of the statement issuance by the bank. It can be monthly, weekly, or after each transaction.

date: The date the account was created.

Card

The Card table holds information about the credit cards that the bank has issued. It has the following columns:

card_id: This is the identification of the credit card and the primary key of this table.

disp_id: The disposition of the card to a bank account.

type: There are three different types of credit cards: junior, classic and gold.

issued: The date that the card was issued.

Client

The Client table holds minimal information (due to de-identification) about the bank's client. It has the following columns:

client_id: This is the client identifier and the primary key of this table.

birth_number: This field holds information about both the birth date and sex of the client. The birth date is stored in the format YYMMDD for men and YYMM+50DD for women.

district_id: The address of the client.

Disposition

The Disposition table holds information about the relationship of the client and the account. It has the following columns:

disp_id: This is the record identifier and the primary key of this table.

client_id: The identification of a client. It can be linked to the Client table with this field.

account_id: The identification of an account. It can be linked to the Account table.

type: The type of disposition, which is either owner or user.

District

This table holds information about the demographic of the country. It doesn't have a single primary key. It has the following columns:

a1: This is the district code. The Czech Republic is divided into districts. An integer has been assigned to each of these districts.

a2: This is the district name. We can associate the district number with the district code and the region.

a3: The region in which clients are located.

a4: The number of inhabitants.

a5: Number of municipalities with less than 499 inhabitants.

a6: Number of municipalities with number of inhabitants 500 - 1999.

a7: Number of municipalities with number of inhabitants 2000 - 9999.

a8: Number of municipalities with more than 10000 inhabitants.

a9: Number of cities.

a10: Ratio of urban inhabitants.

a11: Average salary.

a12: Unemployment rate of 1995.

a13: Unemployment rate of 1996.

a14: Number of entrepreneurs per 1000 inhabitants.

a15: Number of crimes committed in 1995.

a16: Number of crimes committed in 1996.

Loan

This table holds information about the loans the bank has given. It has the following columns:

loan_id: This is the identifier of the loan and the primary key of this table.

account_id: The identification of the account. It can be linked with the Account table.

date: The date when the loan was granted.

amount: The amount of money the bank gave as a loan.

duration: The duration of the loan.

payments: The amount of the monthly payments.

status: There are 4 different status for the loan denoted by 4 letters: 'A' stands for contract finished, 'B' stands for contract finished and loan not paid, 'C' stands for running contract and 'D' stands for running contract and client in debt.

Order

This table holds information about the permanent orders from debit (payments). It has the following columns:

order_id: The order identifier and the primary key for this table.

account_id: The account the order was issued for. It can be linked with the Account table.

bank_to: The bank of the recipient of the payment. Each bank has a two-letter code.

account_to: The account of the recipient.

amount: The amount of money that is being paid/transferred.

k_symbol: This is the characterization of the payment in 4 different categories: Insurance payment, household payment, leasing payment and loan payment.

Transaction

This table holds information about the transactions of the bank's clients. It has the following columns:

trans_id: The transaction identifier and the primary key of this table.

account_id: The account associated with this transaction. It can be linked with the Account table.

date: The date of the transaction.

type: The type of the transaction as in deposit or withdrawal.

operation: There are 5 different operations: credit card withdrawal, credit in cash, collection from another bank, withdrawal in cash and remittance to another bank.

amount: The amount of money of this transaction.

balance: The remaining amount after the transaction.

k_symbol: The transaction characterization in 7 different categories: Insurance payment, statement payment, interest credited, sanction interest if negative balance, household payment, pension and loan payment.

bank: The bank of the partner which the payment is issued for. Each bank has a two-letter code.

account: The account of the partner which the payment is issued for.

Note: We might be able to create an extra table that associates the district data to coordinates, in order to create an accurate heatmap.

Queries

1. Plot the remaining account balance of the clients in time. We will project 2 lines on this graph color-coded for males and females. We will use filters to look for specific ages in this line graph.

This can provide useful information to the user, since they can observe the trend of the clients' remaining amounts, in case they ask the bank for a loan.

2. Plot the transaction amount in time. We will project 2 lines on this graph color-coded for credit transactions and withdrawals. It is important for a banker to know if their client is spending more and more money over time, or if they are trying to cut back. We can also filter the information in this line graph.

3. Plot the number of each type of card issued per year over all the available years in the data set. In this bar chart we can place filters on the years we are interested in. The importance of this graph lies within the fact that some cards are premium, and the bank might be lending more money to some clients by giving them a higher credit limit.

4. Plot the number of people of each loan status over the available years in the data set. In this bar chart we can place filters on the amount of the loan that was taken and the time period the user wants to look at. This query is useful since the user can see how many loans are active/inactive and which ones have been paid/not paid.

5. Heatmap of the inhabitants on the various districts. This could be beneficial to the bank in order to decide where its next physical store should be.

6. Plot the creation of entrepreneur accounts over time using time filters. This is a line graph that can be used by the bank to identify if in a specific amount of time it has attracted more entrepreneurs.

7. Create a bar chart of the various statement frequencies. This is useful in order for the bank to find out when its clients prefer to receive their statements.

8. Create a histogram of the amounts for each type of loan status and color-code the number of loans in each amount. We will create filters in order for the user to see only the amounts they are interested in.

9. Create a pie chart with the transaction characterization for the Transaction table, such as insurance payment, statement payment, interest credited, sanction interest if negative balance, household payment, pension and loan payment. With this graph the bank will know which payments are most common and which ones are more rare in order to adjust its fees accordingly.

10. Heatmap of the transactions. This can be very useful if the bank is considering targeting certain parts of the country with mail offers.

Some of the queries generated in this software are dependent or have relationships with the other queries. The analysis logic of this research is firstly to discuss the general trend or distribution of one type

of financial information and then discuss the trend or distribution of the detailed perspectives. The trend and distribution can be temporal and spatial, i.e. considering the changing patterns of different information along the time period or the condition in different geographic regions. The specific dependence relationships of some of the queries are as follows:

This project mainly discusses the financial issues from the perspective of balance, transaction and loan. From the perspective of transaction, we firstly plot the general trend of transaction amount over time (Query 2), and then consider the amount of transactions in different districts (Query 5), the transaction of entrepreneurs (Query 6), and the statement frequencies (Query 7). From the perspective of a loan, we firstly considered the general number of applicants of different types of loans over time (Query 4), and then specifically discuss the frequency distribution of each type of loan status and in each amount bucket (Query 9).

Additionally, some of the queries are generated based on the results of the previous queries. For example, the trend by number of opened entrepreneur accounts over time is generated by selecting the entrepreneur accounts from the general temporal pattern of transactions.

Software Requirements

The software stack will consist of a web-based UI that provides navigation controls to browse between the pages which present new knowledge we derive from our relational data model (Figure 1).

The team is planning to use the following technologies for implementing the project requirements:

- Drivers: provided by Oracle for accessing the proprietary database
- Python SqlAlchemy - Object Relational Mapper
  https://docs.sqlalchemy.org/en/13/dialects/oracle.html
- Javascript: for implementing the visualizations

The frontend will stream data from backend layer through REST APIs that translates UI query into a select used to fetch data from the Oracle DB instance.

Frontend: ReactJS framework with a visualization toolkit such as d3.js, chartjs, react-vis, or highcharts. The decision on which library to use will be made as soon as we start to experiment with the backend.
Backend: Flask/Spring/nNodeJS based RESTful APIs for returning data to the frontend.
DB: CISE hosted Oracle DB instance as persistence layer.

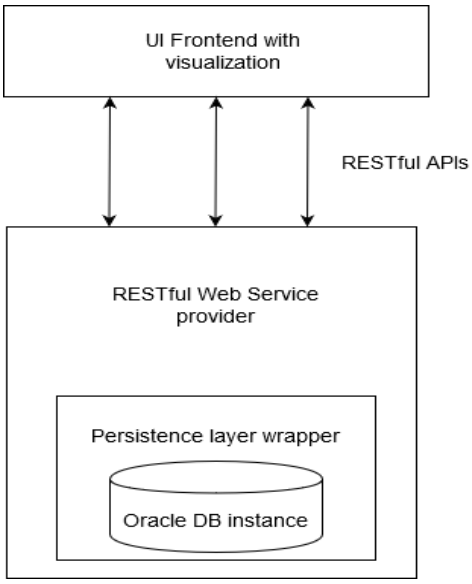

Figure 1: Full Stack

Data Source

The dataset to be used for drawing aforementioned inferences is hosted at data.world site published under Public domain license by @lpetrocelli. This database was prepared by Petr Berka and Marta Sochorova. It is one among few data repositories on bank finance data available for open access. The dataset captures real anonymized Czech bank transactions, account info, and loan records released for PKDD'99 Discovery Challenge.

It captures relations between clients account, disposition, orders, transactions, loan, credit card details as well as demographic details in form of district of originating and destination bank account branch location.

References:

- http://lisp.vse.cz/pkdd99/berka.htm
- https://www.researchgate.net/post/Is_there_any_public_database_for_financial_transactions_or_at_least_a_synthetic_generated_data_set