Date: February 24th, 2018
Team Members: Gregory Matthews / Georgi Simeonov

Project Description:

Our project will replicate the services that a typical brokerage firm would provide to investors. The database will allow brokers in the firm to access their client's personal information and account information, access investment trading history of a particular broker, as well as query a client's history of past statements, and similarly a broker's commissions. The database also provides the transaction data of every client account under the firm, such as their withdrawal and deposit history. All of the particular trades that are executed by a broker are also related to the type of investment products the brokerage firm has open to their clients. All of the entities provided in our database are focused for the primary use of a broker, as well as very simple client usage for transaction information. The database is designed for the most optimal broker usability.

Entity sets, Relationship sets, Business Rules:

Brokers:

- Brokers have a key: bid
- Attributes consist of {bid, first name, last name, DOB, type}
- Each Broker makes at least one Trade

Clients:

- Clients have a key: SSN
- Attributes consist of {first name, last name, DOB, SSN, address, phone number}
- Each Client has at least one Account
- Each Client has at least one Transaction

Account:

- Accounts have a key: Account ID (aid)
- Attributes consist of {aid, type, IRS_status, date_created}
- Each Account has exactly one Client
- Each Account has exactly one Statement (per year)

Trades:

- Trades have a key: Trade ID (trid)
- Attributes consist of {trid, type, action, shares, price}
- Each Trade is made by exactly one Broker
- Each Trade is exactly one type of Investment Product

Commissions:

- Commissions have a key: Commission ID (cid)
- Attributes consist of {cid, amount, type, commission_date}
- Each Commission is received by exactly one Broker

Statements:

- Statements have a key: Statement ID (sid)
- Attributes consist of {sid, type, statement_date}
- Each statement is received by exactly one Account

Transactions:

- Transactions have a key: Transaction ID (tid)
- Attributes consist of {tid, action, amount, transaction_date}
- Each transaction is made by exactly one Account

Investment Products:

- Investment products have a key: Investment ID (iid)
- Attributes consist of {iid, type}
- Each Investment Product is exactly one type of Trade

Translation to the ER Diagram:

We translated the ER diagram based on the relational schema based on the techniques learnt in class. Most entity sets were translated in to tables. InvestmentProducts and Trades (as well as type_of) were combined into one table because they have one to one relationship. None of the relationship sets got their own tables. They were added to the entity that has the most restrictive relation (bold arrow - “exactly one”). For example Make was added to Transactions just as Receive was added to Commissions. Also Has(with Since) was added to Accounts. If any connection could be interpreted as exactly one, its table got a foreign key the table that is related to - Accounts has a foreign key to Clients, Statements has a foreign key to Accounts, Transactions has a foreign key to Accounts and etc. There were some exceptions - the relationship between Accounts and Brokers. It has no restrictions from our ER diagram. Although having an account without a broker did not make sense in our scenario so we added a foreign key to Accounts referencing Brokers. Later after receiving feedback that foreign key was not applicable to the actual ER diagram so it was removed. The relationship is many-to -many (as stated above - it has not restrictions). So Associated_with became a table called Broker_Account_Bridge (instead of bridge the term junction can be used).

The tables we created in the database(schema.sql) are listed below:

- Commissions
- Trades
- Transactions
- Statements
- Accounts

- Brokers
- Broker Account Bridge (added after the feedback)
- Clients

Data Acquisition:

The data was manually generated as insert statements which can be found in the schema.sql file. It can be loaded multiple times from the psql interface (\i schema.sql). All tables have been populated in sufficient number of rows in order to show the relational integrity between tables.

User Interfaces:

Our team has implemented a user interface using python notebooks via Jupyter. The psycopg2 library allowed us to convert SQL queries into corresponding python interpretations, while the pandas library was used to print graphical table representations of the query. Some of the following user level functionalities that our UI supports are as follows:

- Show client IRS status information corresponding to each Broker account, given IRS status code.
- List account transaction amounts given a particular date range and amount threshold.
- Output the total sum of shares, and total price of trades handled by each broker, given the type of trade (sell, buy, limit)
- List the top “k” earning brokers based on commission amounts, where k is the number of brokers to display.
- Show the total transaction amounts for each broker based on whether the type is deposit or withdrawal.