CS5200 Database Management

2023 Fall

Group Project Report

Rui Zeng

Dominic Zhang

Chenqijia Zhang

Customer Investment Management System (CIMS)

# Background

The project is intended to design a database and its host program for brokers in financial investment industries. This system will allow users to manage the investment situation of their clients’ information, develop investment strategies based on visualized market data and perform CRUD operations on the database.

# README: Project Setup and Execution

**System Requirements:** Compatible with Windows, macOS

**Software Requirements:**

* Python (The program was based on Python 3.9)
* PyMySQL (Python MySQL client)
* Matplolib (Python plotting library)
* MySQL Server

PyMySQL and Matplotlib can be installed by pip install statements.

**Submission Specifications:**

* CIMS (SQL text file): Code for Database
* CIMS\_Host (py file): Python host program
* Sample data (SQL text file): self-generated sample data used for testing

**Technical Specifications:**

* Host Language and Libraries: Python (with PyMySQL for database interaction and Matplotlib for data visualization.)
* Database: MySQL, structured with tables for staff, customers, accounts, assets, transactions, etc.

**Database Setup:**

1. Run the CIMS (SQL text file) to build the database.
2. Run sample data file to import the sample data file.
3. Use CMD command to run the host file.

# User Flow

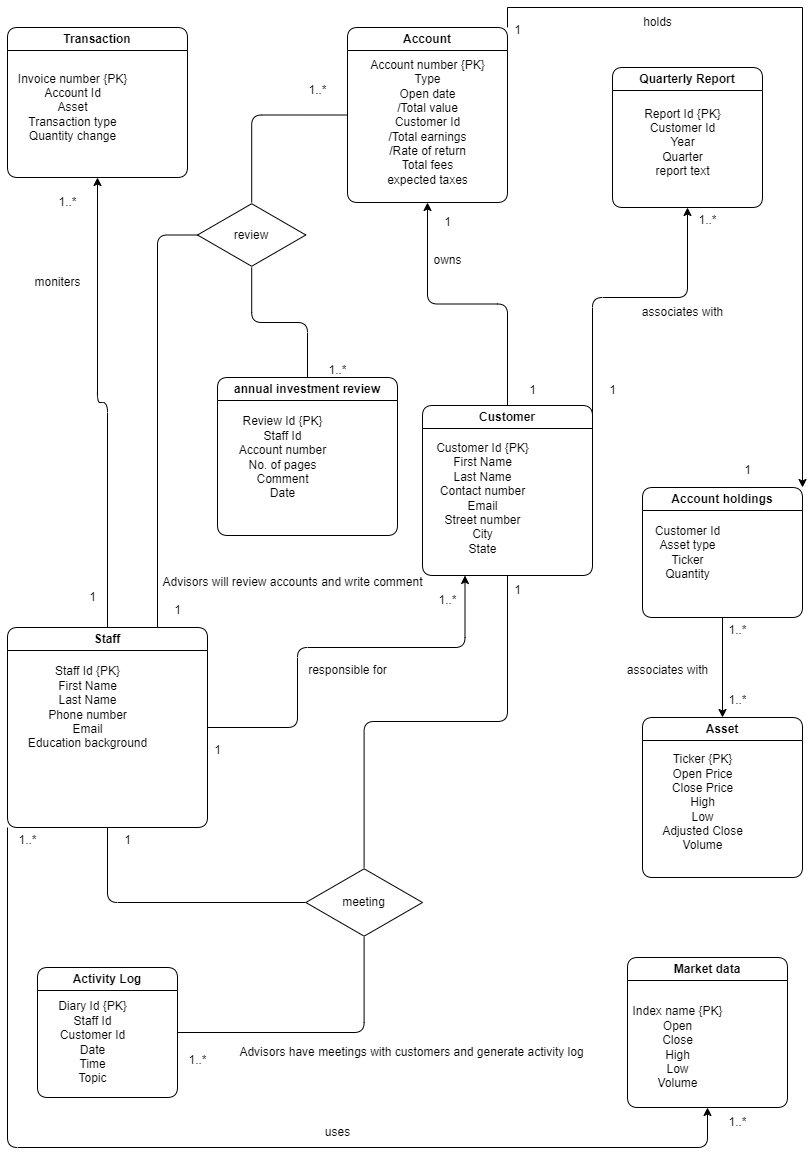
1. Staff Authentication: Enter staff ID and password.
2. Menu Navigation: Choose from options to view data, manage clients, execute transactions, etc.
3. Data Interaction, Update and Analysis: This includes:
   1. Viewing, visualizing, adding, updating, and deleting market data.
   2. Viewing, adding, updating, and deleting customer and accounts information.
   3. Record transactions.
   4. Documents management (Investment Reviews, Financial Reports).
   5. Customer service quality control (viewing staff-client interactions).

# Conceptual Design

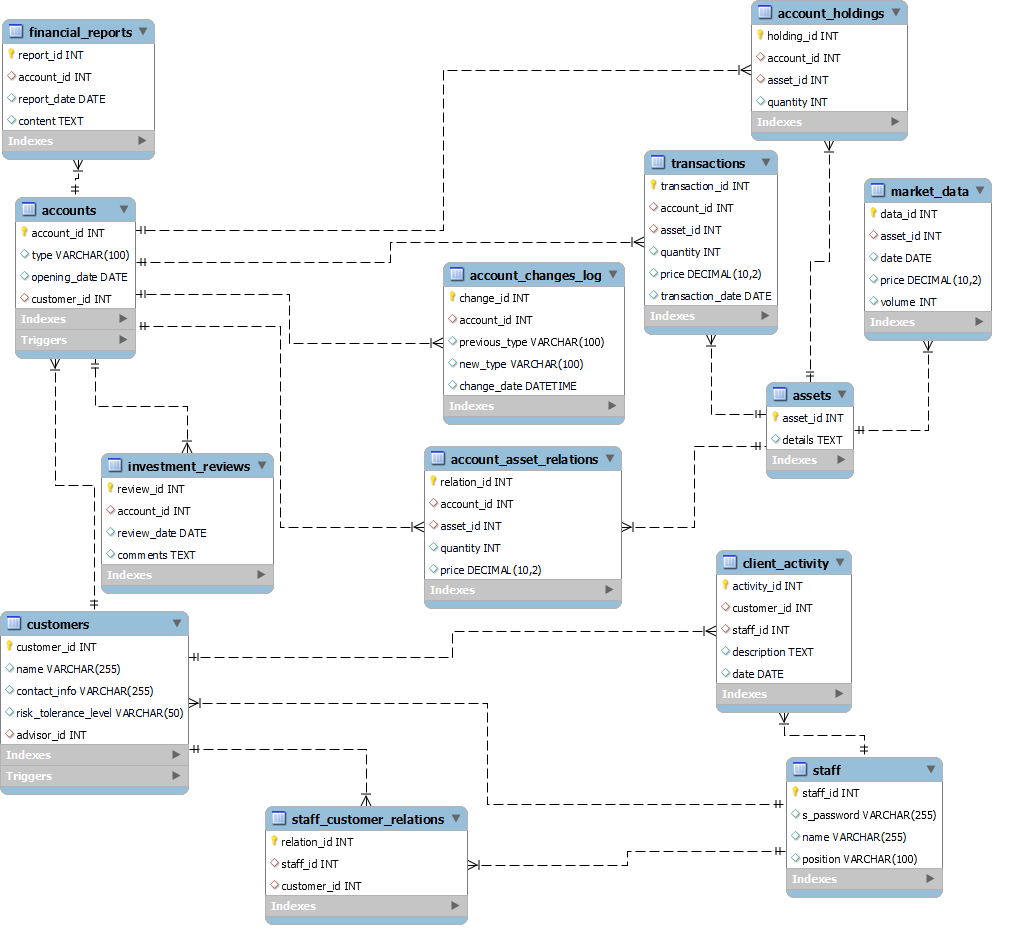
The database contains multiple entities including staffs, customers, accounts, etc. The entities and relationship among them are as follows:

* **Staffs:** A staff member can be an advisor to multiple customers.
* **Customers:** A customer can have only one account and is linked to a staff advisor.
* **Accounts:** An account is associated with a customer and records the investment-related statistics, such as total earnings and rate of return.
* **Assets:** Assets entity is a summary of all the assets our clients hold or have held before. It can be involved in transactions across various accounts.
* **Investment Reviews:** Manager-generated reports linked to an account, providing feedback on investment performance. This is used internally to monitor and evaluate the investment performance, and investment recommendations.
* **Financial reports:** Manager-generated reports providing financial performance of accounts to both clients and staffs.
* **Client activities:** Record of interactions between clients and staffs. This table is used to improve client services and coordinate internal management.
* **Market Data:** Record of market data that help to develop investment strategies.

# Entity-relationship diagram



# Logical Design



# Lessons Learned

**Python Proficiency:** Developed a deeper understanding of Python, especially in database interactions using PyMySQL and data visualization with Matplotlib. Mastered the use of Python for building complex command-line applications, handling exceptions, and managing external libraries.

**MySQL Expertise:** Gained substantial experience in designing and managing MySQL databases. This includes writing complex SQL queries, creating stored procedures, setting up relational tables with appropriate constraints, and implementing triggers for maintaining data consistency.

# Insights

**Problem-Solving Strategies:** Enhanced abilities in debugging and optimizing Python code. Learned to effectively troubleshoot SQL query issues, leading to more efficient and reliable database interactions.

**Data Handling:** Improved skills in data manipulation and analysis, particularly in converting raw financial data into meaningful insights through Python scripts and SQL queries.

**User Interface Design:** Gained insights into designing user-friendly command-line interfaces, understanding the importance of clear prompts, easy navigation, and presenting data in a readable format.

# Future Work

**Database Expansion:** Integration with more complex financial data and analytics tools.

**Functionality Enhancements:** Advanced user management, automated data processing.

**Justification:** Future enhancements will be driven by user needs and technological advancements.

**User Experience Improvement:** Develop better GUI in the future, improve operation logics to ensure better user experience.