**Database System Final Project**

**Table of Contents**

**Pg 1 -** [***System Requirements***](#sdp2d84ykqem)

**Pg 3 -** [***Contextual Data Flow Diagram***](#es46ms4n3boa)

**Pg 4 -** [***Entity Relationship Diagram***](#13lq46qs1vtz)

**Pg 5 -** [***Normalized Database Model***](#jthmcw9r6f6u)

**Pg 6 -** [***Rationale of Database System***](#jthmcw9r6f6u)

**Pg 7 -** [***Implementation-Ready Database Model***](#7cp89phagnk8)

**Pg 8 -** [***Data Dictionary***](#le8mp1jlic3r)

**Pg 11 -** [***SQL Query Ideas***](#nn4eyzb1aoyo)

**Pg 16 -** [***Appendix***](#nb99it4in9gn)

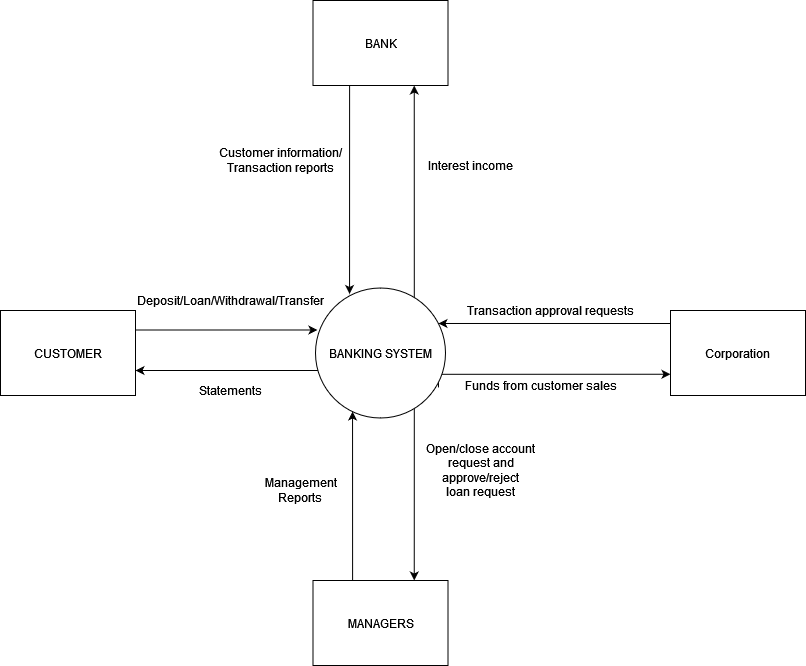
**Member Names (Database Maniac):**

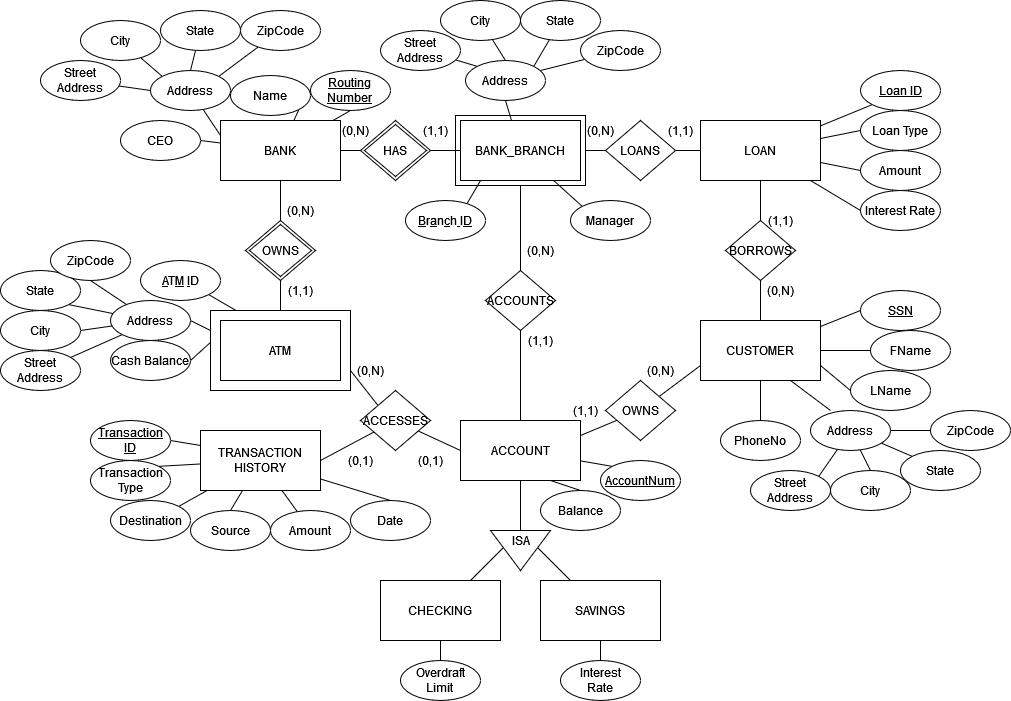
* JT Whetstone
* Xiangjian (Jay) Wu
* Wilbert Liu
* Abazar Naqvi

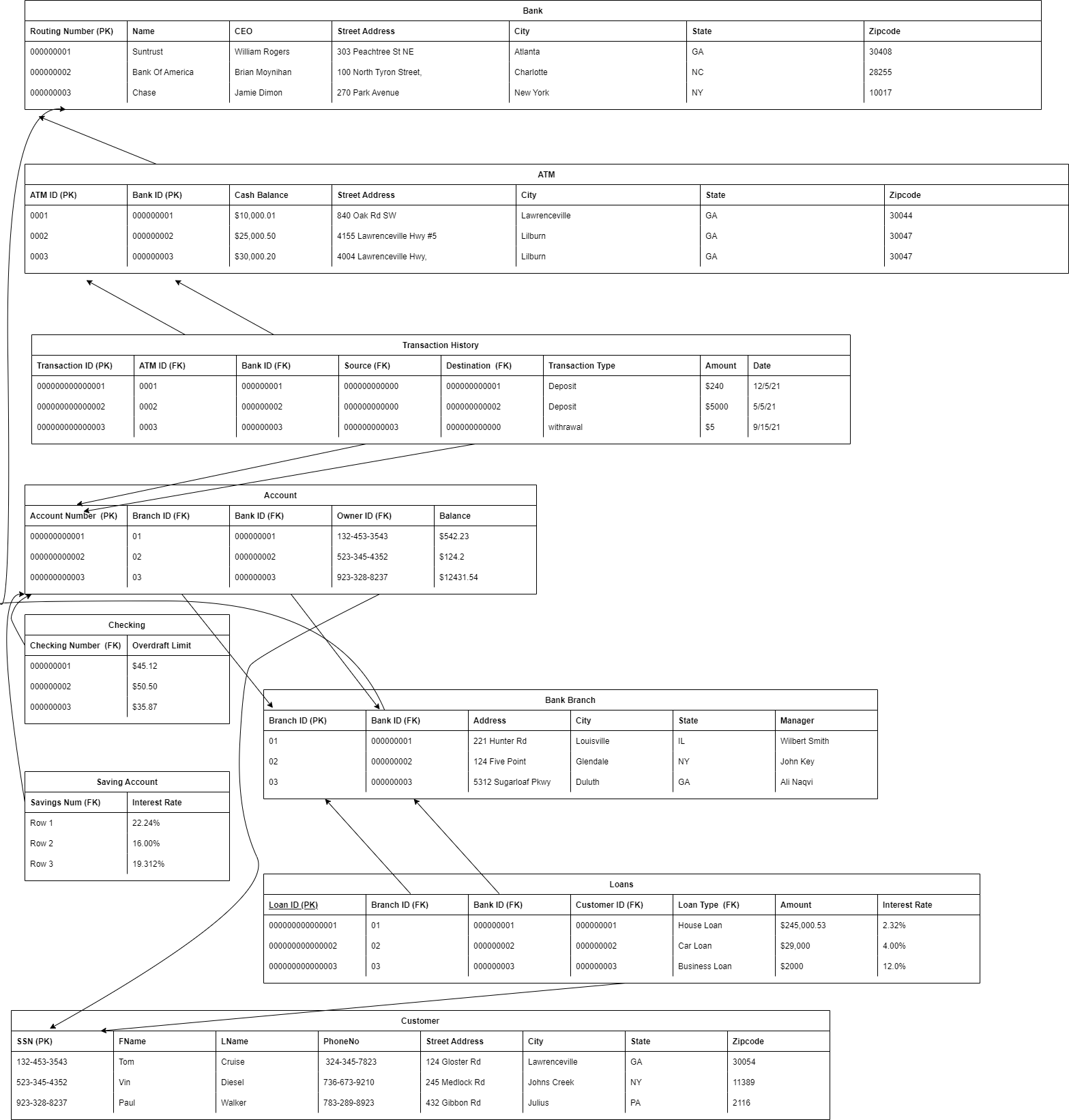
1. **System Requirements**

The purpose of this system is to track information about various banks and their branches, as well as the customers they serve and the accounts those customers create within the bank. Banks should be able to input customer information and transaction reports, then receive information about accrued income based off of interest rates. Managers should be able to input management reports of accounts and loans then receive account and loan requests. Corporations should use this database to input transaction information and receive the necessary funds from the appropriate customer accounts. Finally, customers should be able to use this database to request transactions and view the statements as a result of these transactions. This system is currently designed for a set of local bank systems, and does not support more than one routing number per bank entity.

1. Client Requirements
   1. Each BANK needs to have a unique routing number, company name, CEO, and address.
      1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
   2. BANK needs to be able to have many BANK BRANCHES that are identified through their parent BANK and their own Branch ID number. They should have their own office addresses, managers, and their own branch names that are typically modeled after the city they reside in. A BANK does not have to have a BANK BRANCH, but a BANK BRANCH must have a parent BANK. A BANK’s headquarters will be referenced as Branch No 1 if any account was made at the headquarters.
      1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
   3. A BANK BRANCH should be able to give out multiple LOANs that have their own unique identification number, a loan type associated with it, amount, and interest rate. A LOAN must have a BANK BRANCH but a branch does not have to have any loans out.
      1. There are three types of loans: House, Car, and Business Loans.
   4. A BANK BRANCH should also be able to account for multiple ACCOUNTs. Each of these accounts should have a distinct account number as their identification. They should also have a financial balance.However, a BANK BRANCH does not have to have an ACCOUNT and an ACCOUNT must have a BANK BRANCH.
      1. An ACCOUNT will either be a CHECKING or SAVINGS account. The difference between the two should be that a CHECKING account will have an overdraft limit and a SAVINGS account will have an accompanying interest rate.
   5. A BANK must also be able to have many ATMs to reach out to customers. These ATMs must be associated with one BANK only and are identified by their parent BANK ID and their own specific ATM ID number. They must also be able to store a cash balance within the machine and have their ADDRESS be available to the database system. A BANK does not have to have an ATM.
      1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
   6. A CUSTOMER should be identified by their Social Security Number (denoted as SSN). The customer’s first name, last name, phone number, and address should also be stored. A CUSTOMER should be able to borrow a loan. They can have zero to many loans, but a LOAN must only belong to one CUSTOMER. A CUSTOMER should be able to own many ACCOUNTS, but an individual does not have to have one to take out a LOAN or do business with the BANK. An ACCOUNT must belong to at least one and only one CUSTOMER.
      1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
   7. There should also be a table of TRANSACTION HISTORY that should be accessible to accounts and ATMs. Each transaction will have a specific transaction identification number. It must also store the date, amount, destination, source, and transaction type of every transaction.
      1. There are three transaction types: Deposit, Withdrawals, and Transfers
      2. The destination and source of each transaction will be the account numbers of their respective accounts.
2. **Contextual Data Flow Diagram**

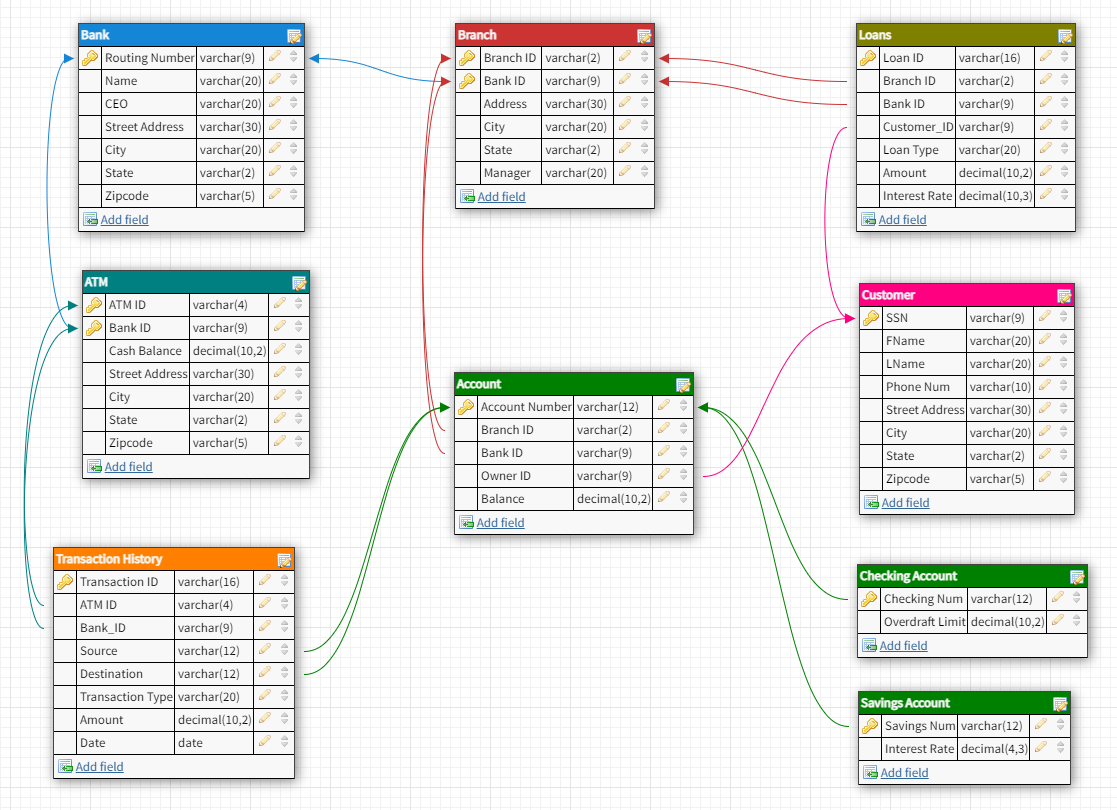


1. **Entity Relationship Diagram**
2. **Normalized Database Model**



1. **Rationale of Database System**

* The scalability of this database system is rather flexible. The system allows for multiple banks and multiple branches, and can also be scaled down to just one bank with multiple routing numbers. However, the system would need to be modified once the requirement of multiple routing numbers for each bank arises.
* Our DBMS of choice is MySQL. It is relatively cheap and we have the most experience with it. It also supports the horizontal scaling of our project well, making it a good choice considering the current limitations of our schema. Additionally, its architecture allows for low query times and vertical scaling, allowing us to store a large number of entries and query them efficiently.

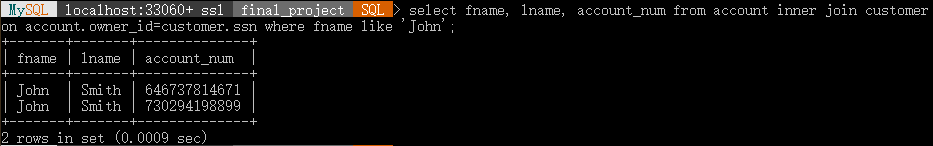
1. **Implementation-Ready Database Model**

<https://dbdesigner.page.link/ehkJVKnxzHHAWmJc7>

1. **Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | Field Name | Data Type and length | Constraint | Description |
| Bank | Routing\_number | char(9) | Primary key | Uniquely identify bank |
| Bank | Name | varchar(20) | Not null | Name of the bank |
| Bank | CEO | varchar(20) | Not null | Name of the CEO |
| Bank | Address | varchar(30) | Not null | Address of the bank |
| Bank | City | varchar(20) | Not null | The city of the address |
| Bank | State | char(2) | Not null | The state of the address |
| Bank | Zip\_code | char(5) | Not null | The zip code of the address |
| ATM | ATM\_id | char(4) | Partial Key | Combined with Bank\_id uniquely identify ATM |
| ATM | Bank\_id | char(9) | Partial Key | Refer back to table Bank attribute Routing\_number |
| ATM | Cash\_balance | decimal(10,2) | Not null | Cash balance |
| ATM | Address | varchar(30) | Not null | Address of the ATM |
| ATM | City | varchar(20) | Not null | The city of the address |
| ATM | State | char(2) | Not null | The state of the address |
| ATM | Zip\_code | char(5) | Not null | The zip code of the address |
| Transaction History | Transaction\_id | char(16) | Primary key | Uniquely identify each transaction |
| Transaction History | ATM\_id | char(4) | Foreign key | Refer back to table ATM attribute ATM\_id |
| Transaction History | Bank\_id | char(9) | Foreign key | Refer back to table ATM attribute Bank\_id |
| Transaction History | Source | char(12) | Foreign key | Refer back to table Account attribute Account\_number |
| Transaction History | Destination | char(12) | Foreign key | Refer back to table Account attribute Account\_number |
| Transaction History | Transaction\_type | varchar(20) | Not null | The transaction type of each history |
| Transaction History | Amount | decimal(10,2) | Not null | Amount of each transaction |
| Transaction History | Date | date | Not null | Date of transaction in format of yyyy-mm-dd |
| Bank Branch | Branch\_id | char(2) | Partial keys | Combined with Bank\_id uniquely identify bank branch |
| Bank Branch | Bank\_id | char(9) | Partial keys | Refer back to table Bank attribute Routing\_number |
| Bank Branch | Address | varchar(30) | Not null | Address of the bank branch |
| Bank Branch | City | varchar(20) | Not null | The city of the address |
| Bank Branch | State | char(2) | Not null | The state of the address |
| Bank Branch | Zip\_code | char(5) | Not null | The zip code of the address |
| Bank Branch | Manager\_id | varchar(20) | Not null | Id of the manager |
| Account | Account\_number | char(12) | Primary key | Uniquely identify each account |
| Account | Branch\_id | char(2) | Foreign key | Refer back to table Bank branch attribute Branch\_id |
| Account | Bank\_id | char(9) | Foreign key | Refer back to table Bank branch attribute Bank\_id |
| Account | Owner\_id | char(9) | Foreign key | Refer back to table Customer attribute SSN |
| Account | Balance | decimal(10,2) | Not null | Amount of balance on the account in format xx.xx |
| Loans | Loan\_id | char(16) | Primary key | Uniquely identify each loan |
| Loans | Branch\_id | char(2) | Foreign key | Refer back to table Bank branch attribute Branch\_id |
| Loans | Bank\_id | char(9) | Foreign key | Refer back to table Bank attribute Routing\_number |
| Loans | Customer\_id | char(9) | Foreign key | Refer back to table Customer attribute SSN |
| Loans | Loan\_type | varchar(20) | Not null | Type of loan |
| Loans | Amount | decimal(10,3) | Not null | Amount of loan |
| Loans | Interest\_rate | decimal(10,3) | Not null | Amount of interest rate |
| Customer | SSN | char(9) | Primary key | Uniquely identify a customer |
| Customer | FName | varchar(20) | Not null | Customer first name |
| Customer | LName | varchar(20) | Not null | Customer last name |
| Customer | PhoneNo | char(10) | Not null | Phone number of the customer |
| Customer | Address | varchar(30) | Not null | Mailing address of the customer |
| Customer | City | varchar(20) | Not null | The city of the address |
| Customer | State | char(2) | Not null | The state of the address |
| Customer | Zip\_code | char(5) | Not null | The zip code of the address |
| Checking Account | Account\_number | char(12) | Foreign key | Refer back to table Account attribute Account\_number |
| Checking Account | Overdraft\_limit | decimal(10,2) | Not null | Overdraft limit for each account |
| Savings Account | Account\_number | char(12) | Foreign key | Refer back to table Account attribute Account\_number |
| Savings Account | Interest\_rate | decimal(4,3) | Not null | Saving account interest rate |

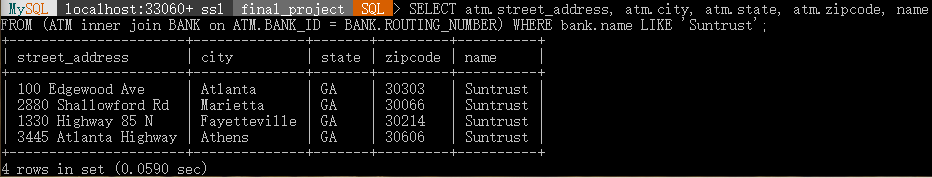
1. **SQL Query Ideas**
   1. List all the accounts owned by customer “John Smith”.



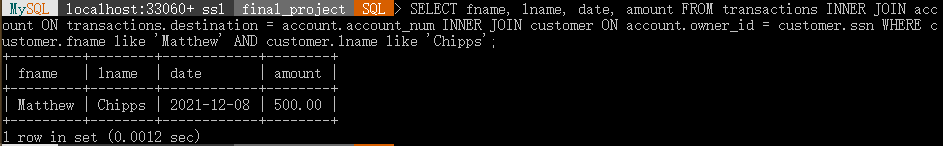
* 1. List the street addresses of which ATMs need to be refilled? (cash balance below $500)



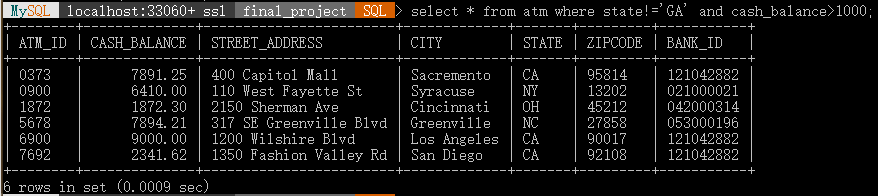
* 1. Show location of all ATMs owned by Suntrust.



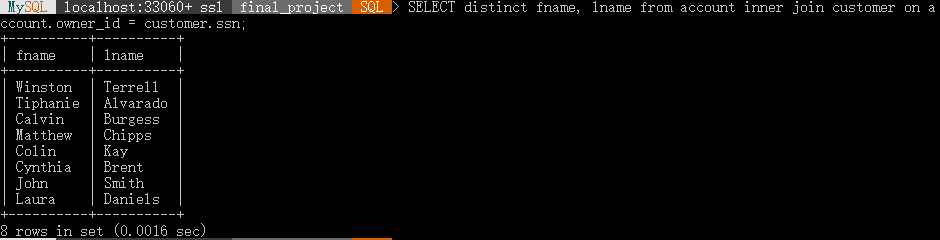
* 1. Find first name, last name, date, and the total dollar amount of transactions done by “Matthew Chipps”



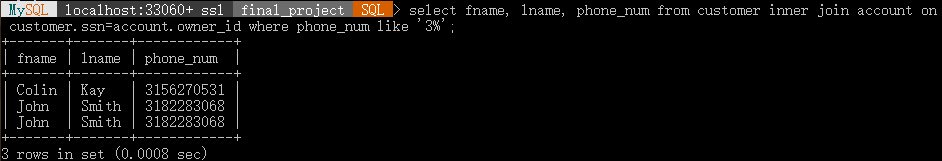
* 1. Show the information of all ATMs outside of Georgia with a cash balance greater than $1000.



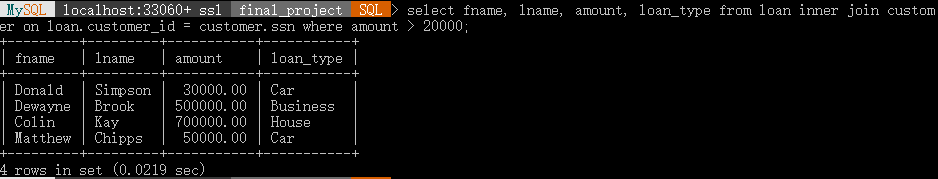
* 1. Show names for all customers who have accounts with any bank in the system.



* 1. Show all names of customers who own an account and whose phone number starts with 3.



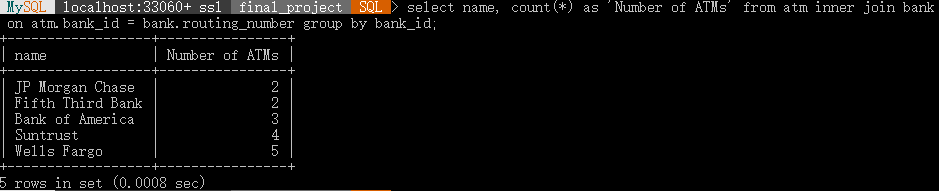
* 1. Show names, loan amounts, and loan types for all customers with loans > 20000



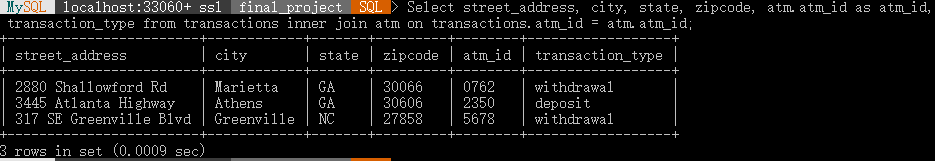
* 1. Show names of customers and the manager name of the bank branch they have an account in.



* 1. Show the number of ATMs each bank owns



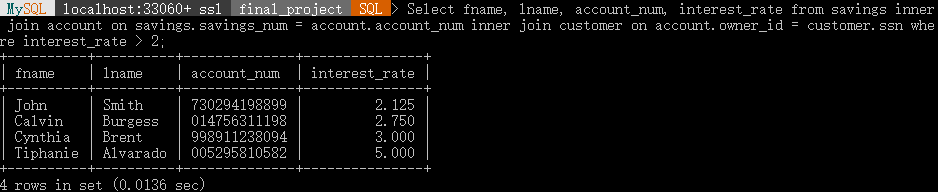
* 1. Show the address of every atm used in a transaction and the transaction type.



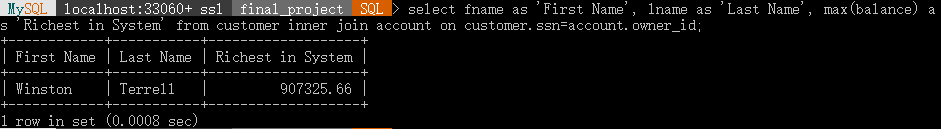
* 1. Show all customer who last name start with A or B or C and order it alphabetically



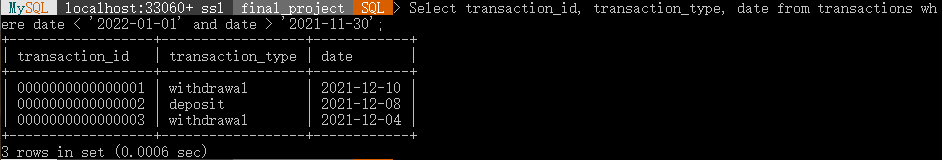
* 1. Show names and account number for all customers who have a savings account with an interest rate greater than 2%



* 1. Find the richest person in this database system



* 1. Find all transactions done in december of 2021



1. **Appendix**

Phase 0: “Hi, please proceed with the Banking System Project. Choose a Team Name, add it to the document and re-upload . Try to bring in additional complexity by adding more features. Please follow guidelines given by the professor for Phase-1.”

Phase 1 (pgs 1-3): No feedback from TA.