Database System Final Project

Table of Contents

Pg 1 - ***System Requirements***

Pg 3 - ***Contextual Data Flow Diagram***

Pg 4 - ***Entity Relationship Diagram***

Pg 5 - ***Normalized Database Model***

Pg 6 - ***Rationale of Database System***

Pg 7 - ***Implementation-Ready Database Model***

Pg 8 - ***Data Dictionary***

Pg 11 - ***SQL Query Ideas***

Pg 15 - ***Time Logs***

Pg 19 - ***Appendix***

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
    - 1.1. Each BANK needs to have a unique routing number, company name, CEO, and address.
        - 1.1.1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
    - 1.2. 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.
        - 1.2.1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
    - 1.3. 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.
        - 1.3.1. There are three types of loans: House, Car, and Business Loans.
    - 1.4. 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.
        - 1.4.1. 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.
    - 1.5. 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.
        - 1.5.1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.
    - 1.6. 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.

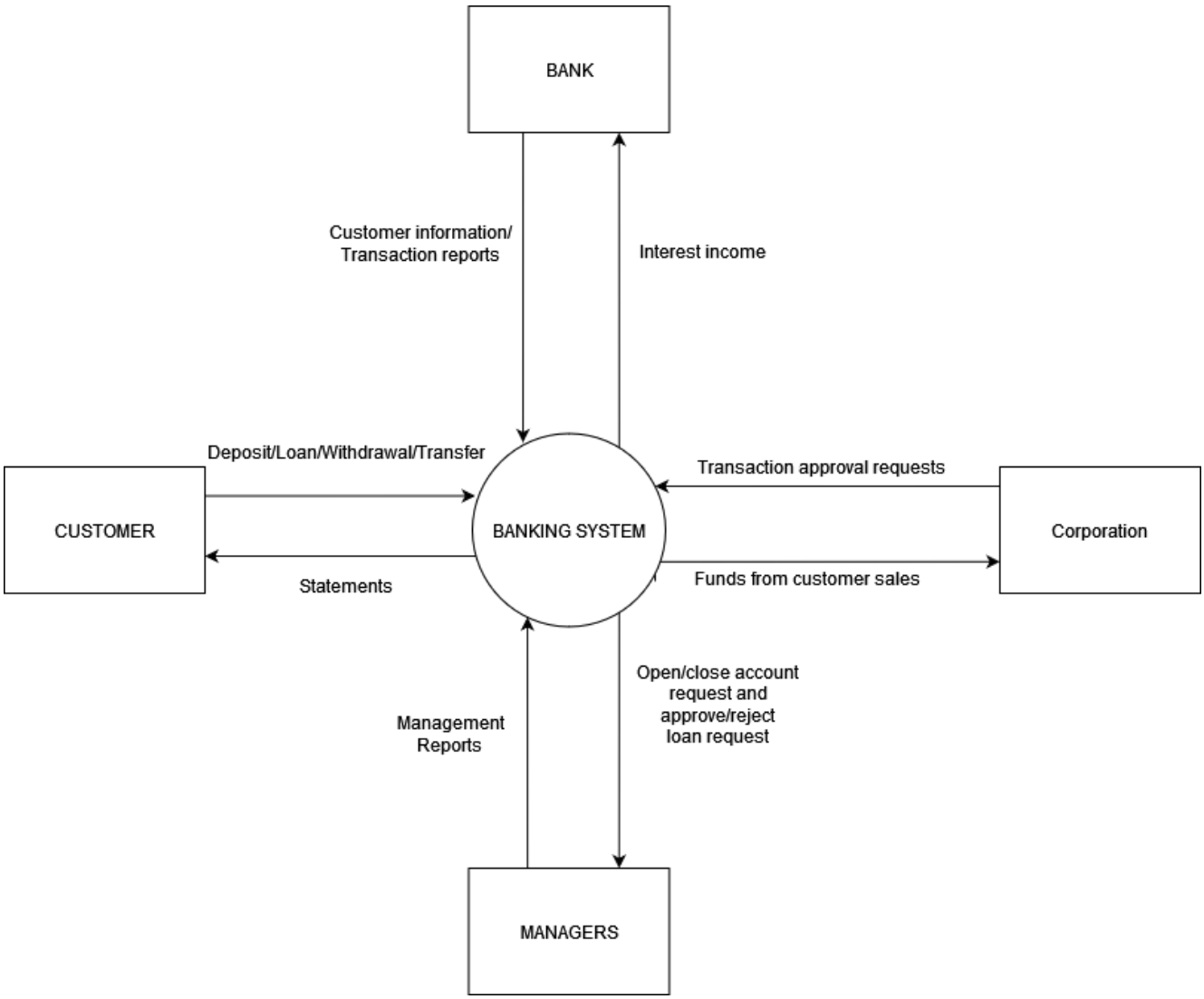
1.6.1. Each ADDRESS value should be separated into STREET ADDRESS, CITY, STATE, and ZIP CODE attributes.

1.7. 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.

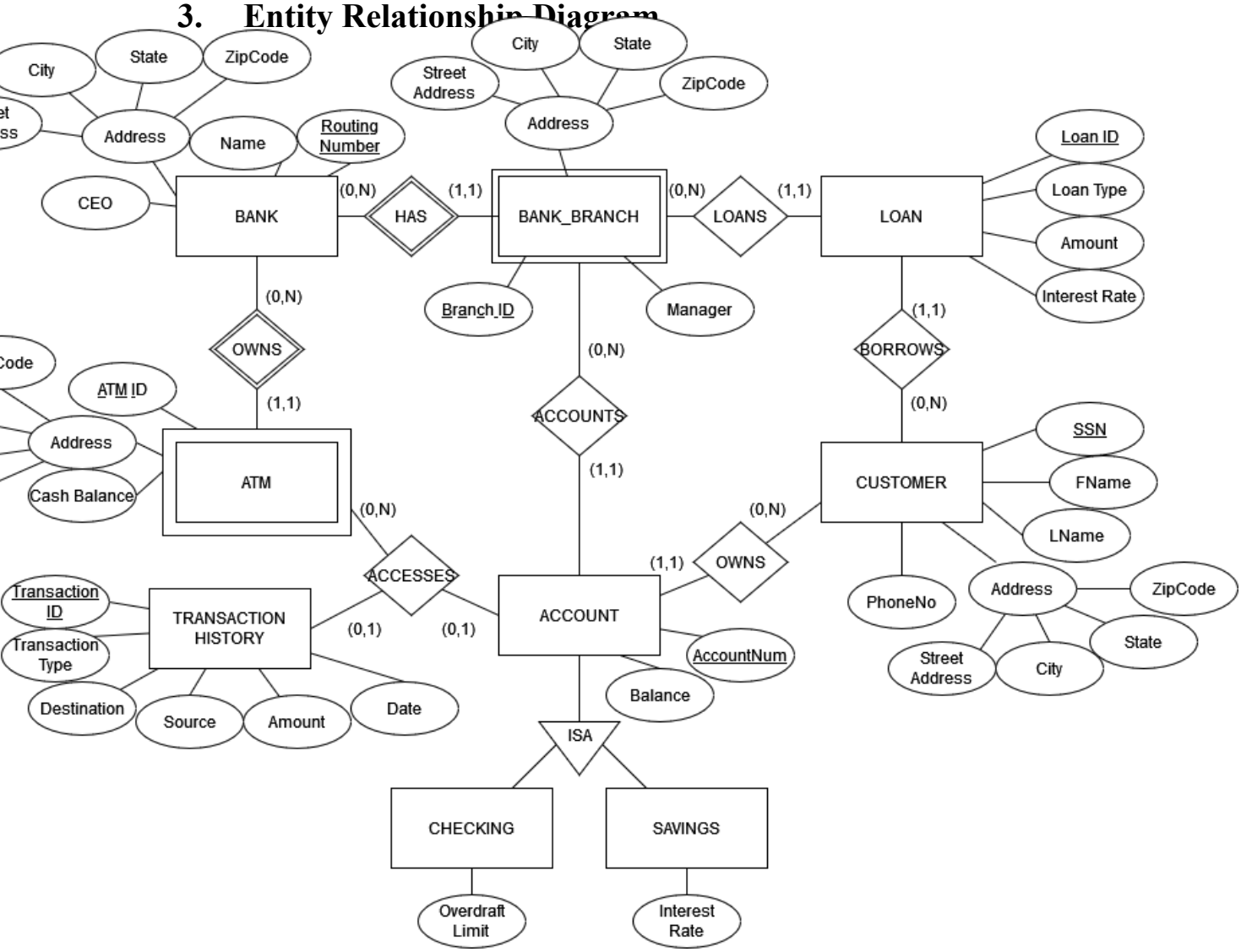
1.7.1. There are three transaction types: Deposit, Withdrawals, and Transfers

1.7.2. The destination and source of each transaction will be the account numbers of their respective accounts.

2. Contextual Data Flow Diagram



3. Entity Relationship Diagram



4. Normalized Database Model

Bank

Routing Number (PK)	Name	CEO	Street Address	City	State	Zipcode
000000001	Suntrust	William Rogers	303 Peachtree St NE	Atlanta	GA	30408
000000002	Bank Of America	Brian Moynihan	100 North Tyron Street,	Charlotte	NC	28255
000000003	Chase	Jamie Dimon	270 Park Avenue	New York	NY	10017

ATM

ATM ID (PK)	Bank ID (PK)	Cash Balance	Street Address	City	State	Zipcode
0001	000000001	$10,000.01	840 Oak Rd SW	Lawrenceville	GA	30044
0002	000000002	$25,000.50	4155 Lawrenceville Hwy #5	Lilburn	GA	30047
0003	000000003	$30,000.20	4004 Lawrenceville Hwy,	Lilburn	GA	30047

Transaction History

Transaction ID (PK)	ATM ID (FK)	Bank ID (FK)	Source (FK)	Destination (FK)	Transaction Type	Amount	Date
000000000000001	0001	000000001	000000000000	000000000001	Deposit	$240	12/5/21
000000000000002	0002	000000002	000000000000	000000000002	Deposit	$5000	5/5/21
000000000000003	0003	000000003	000000000003	000000000000	withrawal	$5	9/15/21

Account

Account Number (PK)	Branch ID (FK)	Bank ID (FK)	Owner ID (FK)	Balance
000000000001	01	000000001	132-453-3543	$542.23
000000000002	02	000000002	523-345-4352	$124.2
000000000003	03	000000003	923-328-8237	$12431.54

Checking

Checking Number (FK)	Overdraft Limit
000000001	$45.12
000000002	$50.50
000000003	$35.87

Bank Branch

Branch ID (PK)	Bank ID (FK)	Address	City	State	Manager
01	000000001	221 Hunter Rd	Louisville	IL	Wilbert Smith
02	000000002	124 Five Point	Glendale	NY	John Key
03	000000003	5312 Sugarloaf Pkwy	Duluth	GA	Ali Naqvi

Saving Account

Savings Num (FK)	Interest Rate
Row 1	22.24%
Row 2	16.00%
Row 3	19.312%

Loans

Loan ID (PK)	Branch ID (FK)	Bank ID (FK)	Customer ID (FK)	Loan Type (FK)	Amount	Interest Rate
000000000000001	01	000000001	000000001	House Loan	$245,000.53	2.32%
000000000000002	02	000000002	000000002	Car Loan	$29,000	4.00%
000000000000003	03	000000003	000000003	Business Loan	$2000	12.0%

Customer

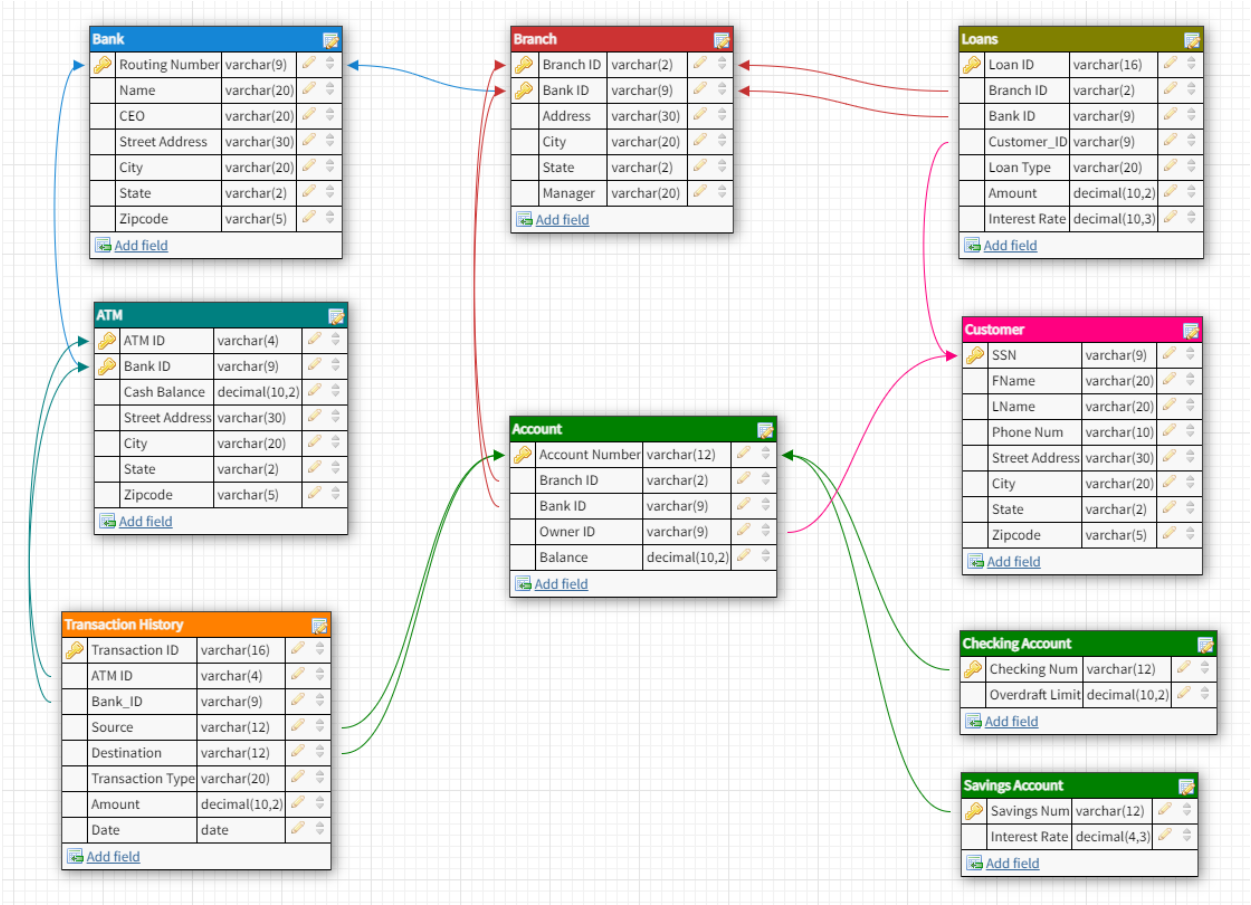
SSN (PK)	FName	LName	PhoneNo	Street Address	City	State	Zipcode
132-453-3543	Tom	Cruise	324-345-7823	124 Gloster Rd	Lawrenceville	GA	30054
523-345-4352	Vin	Diesel	736-673-9210	245 Medlock Rd	Johns Creek	NY	11389
923-328-8237	Paul	Walker	783-289-8923	432 Gibbon Rd	Julius	PA	2116

5. 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.

6. Implementation-Ready Database Model



https://dbdesigner.page.link/ehkJVKnxzHHAWmJc7

7. 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

Data Dictionary

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

Data Dictionary

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

8. SQL Query Ideas

8.1. List all the accounts owned by customer “John Smith”.

```
MySQL localhost:33060+ ssl final_project SQL > select fname, lname, account_num from account inner join customer
on account.owner_id=customer.ssn where fname like 'John';
+-------+-------+--------------+
| fname | lname | account_num  |
+-------+-------+--------------+
| John  | Smith | 646737814671 |
| John  | Smith | 730294198899 |
+-------+-------+--------------+
2 rows in set (0.0009 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > select atm_id, street_address from atm where cash_balance<500;
+--------+----------------------------+
| atm_id | street_address             |
+--------+----------------------------+
| 0023   | 21 East State St           |
| 0101   | 5070 Southport Supply Rd SE |
| 1109   | 350 Main St                |
| 2350   | 3445 Atlanta Highway       |
+--------+----------------------------+
4 rows in set (0.0016 sec)
```

8.3. Show location of all ATMs owned by Suntrust.

```
MySQL localhost:33060+ ssl final_project SQL > SELECT atm.street_address, atm.city, atm.state, atm.zipcode, name
FROM (ATM inner join BANK on ATM.BANK_ID = BANK.ROUTING_NUMBER) WHERE bank.name LIKE 'Suntrust';
+----------------------+--------------+-------+---------+----------+
| street_address       | city         | state | zipcode | name     |
+----------------------+--------------+-------+---------+----------+
| 100 Edgewood Ave     | Atlanta      | GA    | 30303   | Suntrust |
| 2880 Shallowford Rd  | Marietta     | GA    | 30066   | Suntrust |
| 1330 Highway 85 N    | Fayetteville | GA    | 30214   | Suntrust |
| 3445 Atlanta Highway | Athens       | GA    | 30606   | Suntrust |
+----------------------+--------------+-------+---------+----------+
4 rows in set (0.0590 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > SELECT fname, lname, date, amount FROM transactions INNER JOIN acc
ount ON transactions.destination = account.account_num INNER JOIN customer ON account.owner_id = customer.ssn WHERE c
ustomer.fname like 'Matthew' AND customer.lname like 'Chipps';
+---------+--------+------------+--------+
| fname   | lname  | date       | amount |
+---------+--------+------------+--------+
| Matthew | Chipps | 2021-12-08 | 500.00 |
+---------+--------+------------+--------+
1 row in set (0.0012 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > select * from atm where state!='GA' and cash_balance>1000;
+--------+--------------+-----------------------+-------------+-------+---------+-----------+
| ATM_ID | CASH_BALANCE | STREET_ADDRESS        | CITY        | STATE | ZIPCODE | BANK_ID   |
+--------+--------------+-----------------------+-------------+-------+---------+-----------+
| 0373   |      7891.25 | 400 Capitol Mall      | Sacremento  | CA    | 95814   | 121042882 |
| 0900   |      6410.00 | 110 West Fayette St   | Syracuse    | NY    | 13202   | 021000021 |
| 1872   |      1872.30 | 2150 Sherman Ave      | Cincinnati  | OH    | 45212   | 042000314 |
| 5678   |      7894.21 | 317 SE Greenville Blvd | Greenville  | NC    | 27858   | 053000196 |
| 6900   |      9000.00 | 1200 Wilshire Blvd    | Los Angeles | CA    | 90017   | 121042882 |
| 7692   |      2341.62 | 1350 Fashion Valley Rd | San Diego   | CA    | 92108   | 121042882 |
+--------+--------------+-----------------------+-------------+-------+---------+-----------+
6 rows in set (0.0009 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > SELECT distinct fname, lname from account inner join customer on a
ccount.owner_id = customer.ssn;
+----------+----------+
| fname    | lname    |
+----------+----------+
| Winston  | Terrell  |
| Tiphanie | Alvarado |
| Calvin   | Burgess  |
| Matthew  | Chipps   |
| Colin    | Kay      |
| Cynthia  | Brent    |
| John     | Smith    |
| Laura    | Daniels  |
+----------+----------+
8 rows in set (0.0016 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > select fname, lname, phone_num from customer inner join account on
 customer.ssn=account.owner_id where phone_num like '3%';
+-------+-------+------------+
| fname | lname | phone_num  |
+-------+-------+------------+
| Colin | Kay   | 3156270531 |
| John  | Smith | 3182283068 |
| John  | Smith | 3182283068 |
+-------+-------+------------+
3 rows in set (0.0008 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > select fname, lname, amount, loan_type from loan inner join custom
er on loan.customer_id = customer.ssn where amount > 20000;
+---------+---------+-----------+-----------+
| fname   | lname   | amount    | loan_type |
+---------+---------+-----------+-----------+
| Donald  | Simpson |  30000.00 | Car       |
| Dewayne | Brook   | 500000.00 | Business  |
| Colin   | Kay     | 700000.00 | House     |
| Matthew | Chipps  |  50000.00 | Car       |
+---------+---------+-----------+-----------+
4 rows in set (0.0219 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > select fname, lname, manager as 'Manager of Bank Branch' from bran
ch inner join account on branch.bank_id=account.bank_id inner join customer on account.owner_id=customer.ssn group by
 fname;
+----------+----------+------------------------+
| fname    | lname    | Manager of Bank Branch |
+----------+----------+------------------------+
| Winston  | Terrell  | Stacey Stratford       |
| Tiphanie | Alvarado | Patrick Hart           |
| Calvin   | Burgess  | Ian Carmichael         |
| Matthew  | Chipps   | Bryant Smith           |
| Colin    | Kay      | Brennan Leicester      |
| Cynthia  | Brent    | Ian Carmichael         |
| John     | Smith    | Bryant Smith           |
| Laura    | Daniels  | Bryant Smith           |
+----------+----------+------------------------+
8 rows in set (0.0014 sec)
```

8.10. Show the number of ATMs each bank owns

```
MySQL localhost:33060+ ssl final_project SQL > select name, count(*) as 'Number of ATMs' from atm inner join bank
 on atm.bank_id = bank.routing_number group by bank_id;
+------------------+----------------+
| name             | Number of ATMs |
+------------------+----------------+
| JP Morgan Chase  |              2 |
| Fifth Third Bank |              2 |
| Bank of America  |              3 |
| Suntrust         |              4 |
| Wells Fargo      |              5 |
+------------------+----------------+
5 rows in set (0.0008 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > Select street_address, city, state, zipcode, atm.atm_id as atm_id,
 transaction_type from transactions inner join atm on transactions.atm_id = atm.atm_id;
+------------------------+------------+-------+---------+--------+------------------+
| street_address         | city       | state | zipcode | atm_id | transaction_type |
+------------------------+------------+-------+---------+--------+------------------+
| 2880 Shallowford Rd    | Marietta   | GA    | 30066   | 0762   | withdrawal       |
| 3445 Atlanta Highway   | Athens     | GA    | 30606   | 2350   | deposit          |
| 317 SE Greenville Blvd | Greenville | NC    | 27858   | 5678   | withdrawal       |
+------------------------+------------+-------+---------+--------+------------------+
3 rows in set (0.0009 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > select fname, lname from customer where lname like 'A%' or lname l
ike 'B%' or lname like 'C%' order by lname;
+----------+----------+
| fname    | lname    |
+----------+----------+
| Roger    | Abrams   |
| Tiphanie | Alvarado |
| Barbara  | Anne     |
| Cynthia  | Brent    |
| Dewayne  | Brook    |
| Calvin   | Burgess  |
| Matthew  | Chipps   |
| Stephen  | Crocker  |
+----------+----------+
8 rows in set (0.0010 sec)
```

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

```
MySQL localhost:33060+ ssl final_project SQL > Select fname, lname, account_num, interest_rate from savings inner
 join account on savings.savings_num = account.account_num inner join customer on account.owner_id = customer.ssn whe
re interest_rate > 2;
+----------+----------+--------------+---------------+
| fname    | lname    | account_num  | interest_rate |
+----------+----------+--------------+---------------+
| John     | Smith    | 730294198899 |         2.125 |
| Calvin   | Burgess  | 014756311198 |         2.750 |
| Cynthia  | Brent    | 998911238094 |         3.000 |
| Tiphanie | Alvarado | 005295810582 |         5.000 |
+----------+----------+--------------+---------------+
4 rows in set (0.0136 sec)
```

8.14. Find the richest person in this database system

```
MySQL localhost:33060+ ssl final_project SQL > select fname as 'First Name', lname as 'Last Name', max(balance) a
s 'Richest in System' from customer inner join account on customer.ssn=account.owner_id;
+------------+-----------+-------------------+
| First Name | Last Name | Richest in System |
+------------+-----------+-------------------+
| Winston    | Terrell   |         907325.66 |
+------------+-----------+-------------------+
1 row in set (0.0008 sec)
```

8.15. Find all transactions done in december of 2021

SQL Query Ideas

```
MySQL localhost:33060+ ssl final_project SQL > Select transaction_id, transaction_type, date from transactions wh
ere date < '2022-01-01' and date > '2021-11-30';
+------------------+------------------+------------+
| transaction_id   | transaction_type | date       |
+------------------+------------------+------------+
| 0000000000000001 | withdrawal       | 2021-12-10 |
| 0000000000000002 | deposit          | 2021-12-08 |
| 0000000000000003 | withdrawal       | 2021-12-04 |
+------------------+------------------+------------+
3 rows in set (0.0006 sec)
```

9. Time Logs

Name	Task	Time spend (hour)
Jay	Data Dictionary	2.5
	Contextual Data Flow Diagram	1 (coop work with all)
	SQL query	2.5(coop work with Wilbert)
	System requirements	2 (coop work with all)
	EER Diagram	0.75 (coop work with all)
	System requirements (modifying with common error document)	1.5
	Rationale of Database Model	1 (coop work with JT and Wilbert)
	Formatting	0.5
Wilbert Liu	Signed and Verified by Wilbert Liu	
	Signed and Verified by Abazar Naqvi	
	Signed and Verified by JT Whetstone	

Time Logs

Name	Task	Time spend (hour)
Wilbert	System Requirements	2.25 (coop work with all)
	Contextual Data Flow Diagram	1 (coop work with all)
	EER Diagram	1.5
	EER Diagram	0.75 (coop work with all)
	Database Schema	2.5
	SQL Queries	2.5 (coop work with Jay)
	Database Rationale	1 (coop work with JT and Jay)
	Signed and Verified by XiangJian (Jay) Wu	
	Signed and Verified by Abazar Naqvi	
	Signed and Verified by JT Whetstone	

Time Logs

Name	Task	Time spend (hour)
Abazar	EER Diagram	0.75 (coop work with all)
	Database Schema	2.5 (coop work with all)
	Normalized Diagram 3nf	6
	Signed and Verified by XiangJian (Jay) Wu	
Wilbert Liu	Signed and Verified by Wilbert Liu	
	Signed and Verified by JT Whetstone	

Time Logs

Name	Task	Time spend (hour)
JT Whetstone	System Requirements	2.5
	Contextual Data Flow Diagram	0.5
	EER Diagram	5
	Normalized Database Model	0.25(coop work with all)
	Rationale of Database	1 (coop work with Jay and Wilbert)
	Implementation-Ready Model	0.25 (coop work with Wilbert)
	SQL Queries and Creation	3.5
	Signed and Verified by XiangJian (Jay) Wu	
Wilbert Liu	Signed and Verified by Wilbert Liu	
	Signed and Verified by Abazar Naqvi	

10. 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.