ATM Management System Database

PREPARED BY

Robin Fernando

Github Username: robinfern94

Table of Contents

Project Summary

The goal of this project is to build an ATM Management System Database similar to the likes of Wells Fargo, Chase, Bank of America, etc. An ATM's basic features include depositing/withdrawing money, transferring funds, checking account balances, and even making payments to an optional credit card. A common issue that competing banks face is limited features on an ATM. More specifically, some customers have too complex of a transaction that an ATM would not be able to carry out. The customer would then be forced to head inside, get in a long line, and meet with the bank teller to fulfill their request. Not only is this inconvenient for the customer, but it also takes a toll on the bank employees since they would be unprepared for the overflow of impatient customers.

This project will help reduce the need to speak to a bank teller by opening up more denomination options, such as the availability of coins and uncommon bills. Doing this would accommodate the customers looking for specific withdrawal amounts. In addition, the project would also look into the available options when cashing in a check, such as depositing a portion and cashing the rest. This project also factors in the idea of security and would implement a hand recognition system to verify the customer. An extra layer of security would ask for the opposite hand in the case to bypass certain withdrawal limits that ATMs are programmed to strictly follow.

The features that this project brings would help satisfy more customer needs and reduce the workload of the bank staff. This ATM Management System Database will increase the overall usability of the ATM.

Database Requirements

1. Registered User:
    a. A registered user shall have one unique email.
    b. A registered user shall have at least one checking account.
    c. A registered user shall have zero or one savings account.
    d. A registered user shall open only one session at a time.
    e. A registered user shall be able to request specific bills and coins when withdrawing.
    f. A registered user shall be able to determine how to cash the check.
2. Account:
    a. An account shall belong to one or many registered users.
    b. An account shall have one unique account ID.
    c. An account shall have one user-encrypted password.
    d. An account shall have one creation date.
    e. Accounts shall be locked by a bank admin.
    f.
3. Savings Account:
    a. A savings account shall have a unique savings account ID.
    b. A savings account shall hold zero or more money.
4. Checking Account:
    a. A checking account shall have a unique checking account ID.
    b. A checking account shall hold zero or more money.
5. ATM:
    a. An ATM shall ask for the user's login credentials.
    b. An ATM shall update the account's balance.
    c. An ATM shall offer different user transaction options.
    d. An ATM shall accept zero or more checks.
    e. An ATM shall accept zero or more money.

f. An ATM shall disburse zero or more money.
g. An ATM shall send a warning when the internal money reaches a certain threshold.
h. An ATM shall monitor how many different types of bills it contains.
i. An ATM shall turn a check into a cash value.
j. An ATM shall warn the user when they have exceeded the withdrawal limit.
k. An ATM shall print out a receipt for the user.

6. Account Transactions:
    a. Each transaction shall have its own unique ID.
    b. A transaction shall have one or many types.
    c. A transaction type shall be to withdraw, deposit, transfer, or inquire balance.
7. Bank:
    a. A bank must have at least one employee.
    b. A bank must have at least one ATM.
    c. A bank shall receive checks and money from the ATM technician.
    d. A bank shall monitor an ATM's cash flow.
8. Employee
    a. Employees can supervise other employees.
    b. Employees shall work for one bank.
9. Admin:
    a. An admin shall be notified when an ATM is running low on money.
    b. An admin shall notify the technician to refill the ATM.
    c. An admin shall lock one or many accounts.
    d. An admin shall notify the registered user of the locked account via email.
    e. An admin shall have one email.
    f. An admin shall be able to close an account
10. ATM Technician:
    a. A technician shall be contacted by the admin.
    b. A technician shall refill the ATM with money.

c. A technician shall remove the checks from the ATM.

d. A technician shall return the checks to the bank.

11. Money:

    a. Money shall be loaded into an ATM.

    b. Money shall be withdrawn from an ATM.

    c. Money shall be deposited into an ATM.

    d. Money shall be zero or many bills.

    e. Money shall be zero or many coins.

    f. Money shall be split into specific amounts.

12. Check:

    a. Checks shall be accepted by an ATM.

    b. Checks shall be able to split their value.

    c. Checks shall be deposited into at least one account.

    d. Checks shall be turned into cash.

    e. Checks shall be picked up from ATMs by ATM Technicians.

13. Supply System:

    a. The supply system shall keep track of the checks and cash inside the ATM.

14. Debit Card:

    a. A debit card shall belong to one and only one user.

    b. A debit card shall have only one pin code.

    c. A debit card shall be inserted into an ATM.

15. Region

    a. A region shall have zero or many ATMs.

16. Receipt:

    a. A receipt shall be printed for the current session of transactions.

Detailed List of Main Entities, Attributes, and Keys

1. Registered User (Strong)
    a. user_id: key, numeric
    b. full_name: alphanumeric, composite
        i. name
        ii. last
    c. email: alphanumeric, unique
    d. dob: date, composite
        i. year
        ii. month
        iii. day
2. Account (Weak)
    a. account_id: key, numeric
    b. user: key, numeric
    c. language: alphanumeric
    d. created: composite, date
    e. password: alphanumeric
    f. is_locked: numeric
3. Checking (Weak)
    a. checking_acc_id: key, numeric
    b. acc_id: key, numeric
    c. balance: numeric
4. Savings (Weak)
    a. savings_acc_id: key, numeric
    b. acc_id: key, numeric
    c. balance: numeric
5. ATM (Strong)

    a. atm_id: key, numeric
    b. bank: key, numeric
    c. status: alphanumeric
    d. police_station: key, alphanumeric
    e. network: key, alphanumeric
    f. region: key, alphanumeric
    g. keypad: key, alphanumeric
    h. handprint_scanner: key, alphanumeric
6. Bank (Strong)
    a. bank_id: key, numeric
    b. bank_name: alphanumeric
    c. address: alphanumeric, multi-value, composite
        i. street
        ii. zipcode
        iii. state
        iv. country
7. Employee (Weak)
    a. employee_id: key, numeric
    b. bank_id: key, numeric
    c. is_supervisor: numeric
    d. employee_email: alphanumeric, unique
8. Admin (Weak)
    a. employee_id: key, numeric
    b. admin_id: key, numeric
9. ATM Technician (Weak)
    a. employee_id: key, numeric
    b. tech_id: key, numeric
10. Teller (Weak)
    a. employee_id: key, numeric

    b. teller_id: key, numeric

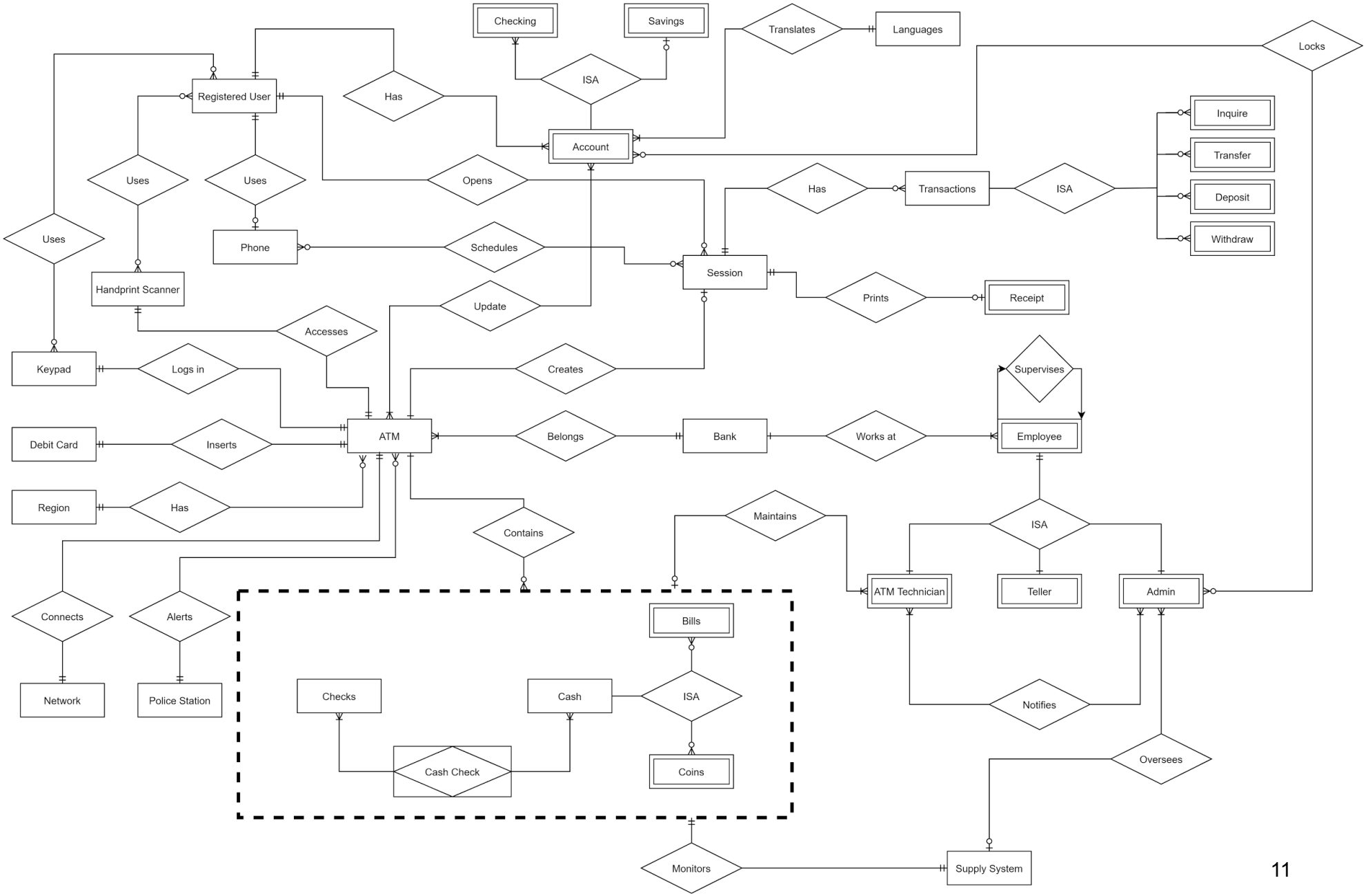
11. Cash (Strong)
    a. cash_id: key, numerical
    b. amount: numerical, derived
12. Bills (Weak)
    a. bill_type: key, numerical
    b. bill_amount: numerical
13. Coins (Weak)
    a. coin_type: key, numerical
    b. coin_amount: numerical
14. Checks (Strong)
    a. check_id: key, numerical
    b. check_value: numerical
15. Debit Card (Strong)
    a. debit_card_id: key, numerical
    b. card_no: numerical
    c. name: alphanumerical
    d. pin: numerical
    e. exp_date: alphanumerical
    f. CVV: numerical
16. Handprint Scanner (Strong)
    a. scanner_id: key, numerical
    b. left_handprint: image
    c. right_handprint: image
17. Supply System (Strong)
    a. supply_id: key, alphanumerical
    b. total_value: numerical, derived
    c. check_co: numerical, derived
18. Region (Strong)

    a. region_id: key, alphanumerical
    b. description: alphanumerical

19. Session (Strong)
    a. session_id: key, numerical
    b. session_create: numerical
    c. user: key, numerical
20. Transaction (Strong)
    a. transaction_id: key, numerical
    b. session_id: key, numerical
21. Inquire (Weak)
    a. inquire_id: key, numerical
    b. account: numerical
22. Withdraw (Weak)
    a. withdraw_id: key, numerical
    b. withdraw_amount: numerical
    c. account: numerical
23. Deposit (Weak)
    a. deposit_id: key, numerical
    b. deposit_amount: numerical
    c. account: numerical
24. Transfer (Weak)
    a. transfer_id: key, numerical
    b. transfer_amount: numerical
    c. account: numerical
25. Receipt (Weak)
    a. receipt_id: key, numerical
    b. session_id: key, numerical
26. Phone (Strong)
    a. phone_id: key, numerical

b. user_phone: key, alphanumeric

Entity Relationship Diagram (ERD)

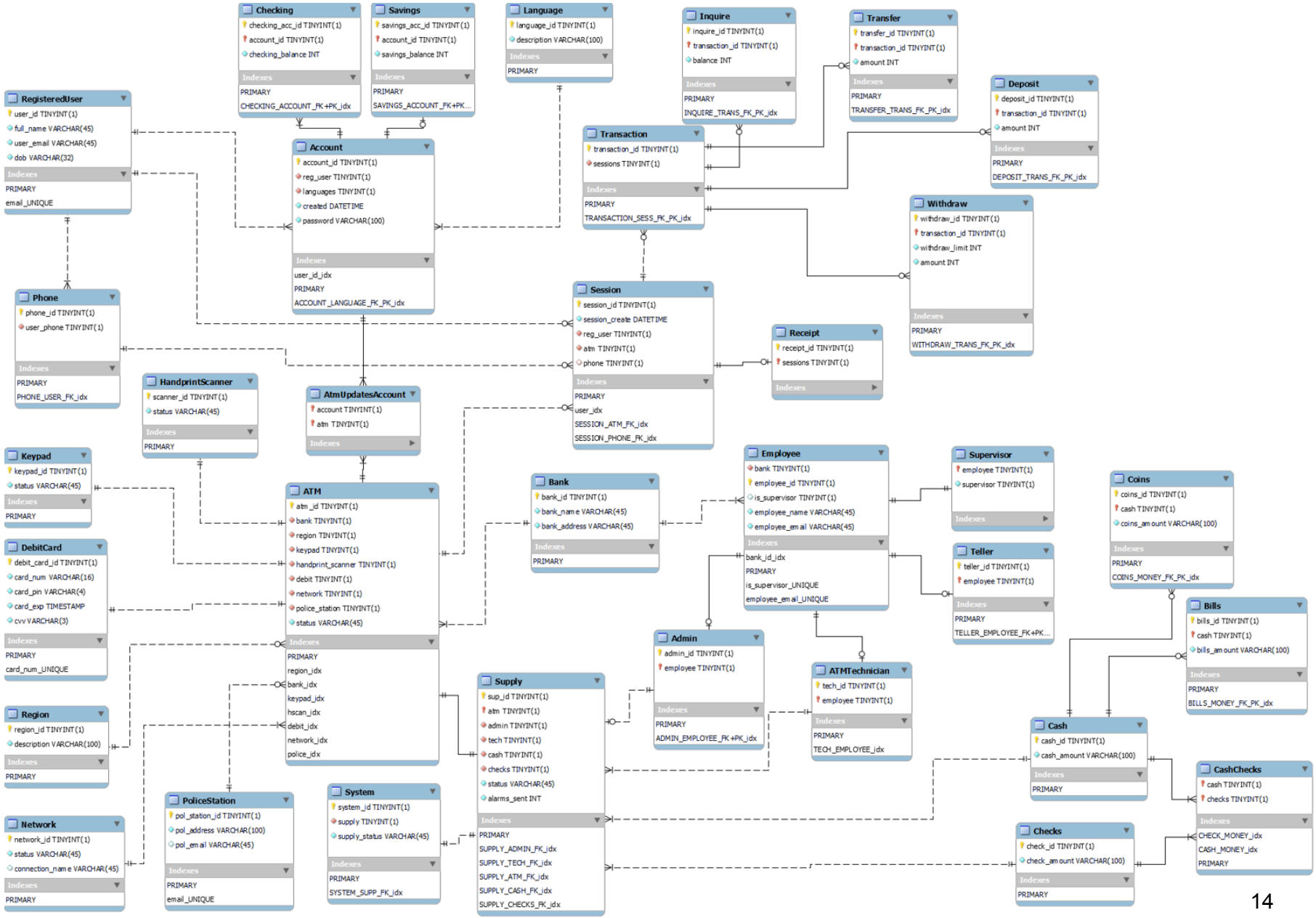


Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/ Fail	Error Description
1	Registered User	Has	Account	M to N	Fail	Registered Users can have multiple types of accounts.
2	Registered User	Uses	Debit Card	1 to 1	Pass	None
3	Registered User	Uses	Handprint Scanner	1 to Many	Pass	None
4	Account	ISA	Savings	1 to 1	Fail	Not required to have a savings account
5	Account	ISA	Checking	1 to M	Fail	Multiple checking accounts are possible.
6	Debit Card	Login	ATM	1 to 1	Pass	None
7	Handprint Scanner	Login	ATM	M to 1	Fail	Both hands are used.
8	ATM	Has	Region	1 to 1	Pass	None
9	ATM	Update	Account	M to N	Pass	None
10	ATM	Creates	Session	1 to 1	Pass	None
11	ATM	Belongs	Bank	M to 1	Fail	Banks own multiple ATMs.
12	ATM	Contains	Checks	1 to M	Pass	None
13	ATM	Contains	Money	1 to M	Pass	None
14	Session	Has	Inquire	1 to M	Pass	None

			Balance			
15	Session	Has	Withdraw	1 to M	Pass	None
16	Session	Has	Deposit	1 to M	Pass	None
17	Session	Has	Transfer	1 to M	Pass	None
18	Session	Prints	Receipt	1 to 1	Pass	None
19	Withdraw	Has	Limit	M to 1	Pass	None
20	Money	ISA	Bills	1 to M	Pass	None
21	Money	ISA	Coins	1 to M	Pass	None
22	Checks	Cash Check	Money	M to N	Pass	None
23	Bank	Monitors	Cash Flow	Aggregation	Pass	None
24	Employee	Works at	Bank	M to 1	Pass	None
25	Employee	Supervise	Employee	Recursive	Pass	None
26	Employee	ISA	ATM Tech	1 to 1	Pass	None
27	Employee	ISA	Teller	1 to 1	Pass	None
28	Employee	ISA	Admin	1 to 1	Pass	None
29	ATM Tech	Maintains	Supply System	M to 1	Pass	None
30	Admin	Oversees	Supply System	M to 1	Fail	Not just one admin per system.
31	Admin	Notifies	ATM Tech	M to N	Pass	None
32	Supply System	Monitors	Cash Flow	Aggregation	Pass	None

Database Model / EER



EER Table

Table	FK	ON DELETE	ON UPDATE	Comment
Account	user	ON CASCADE	ON CASCADE	If a user is deleted, the account from the user should be deleted as well so others can use the system.
Account	language	ON CASCADE	ON CASCADE	The language preference is needed to determine how to display the info on the ATM screen.
Checking	account	ON CASCADE	ON CASCADE	If the account is deleted, the checking should also be deleted with it.
Savings	account	ON CASCADE	ON CASCADE	If the account is deleted, the savings should also be deleted with it.
AtmUpdates Account	account	ON CASCADE	ON CASCADE	If the account is deleted, there is nothing for the ATM to update.
AtmUpdates Account	atm	ON CASCADE	ON CASCADE	If the ATM is deleted, no actions can be done on the account.
ATM	bank	ON CASCADE	ON CASCADE	The ATM must belong to a bank.
ATM	region	ON CASCADE	ON CASCADE	ATMs must be located in a certain region.
ATM	keypad	ON CASCADE	ON CASCADE	Required for login.
ATM	handprint scanner	ON CASCADE	ON CASCADE	Required for login.
ATM	debit card	ON CASCADE	ON CASCADE	Required for login.
ATM	network	SET NULL	ON CASCADE	If the network is deleted from the ATM, wait until a new network connection is made.
ATM	police station	ON CASCADE	ON CASCADE	A main priority of our system would be safety; without being able to contact the police, the ATM should not be able to function.
Supply	atm	ON CASCADE	ON CASCADE	No ATM means no supply.
Supply	admin	ON CASCADE	ON CASCADE	An admin is always needed to monitor.

Supply	atm tech	SET NULL	ON CASCADE	If the tech is deleted, set null until a new role is found to fill.
Supply	cash	ON CASCADE	ON CASCADE	If the cash is deleted, then the supply should be deleted as well.
Supply	checks	ON CASCADE	ON CASCADE	If the check is deleted, then the supply should be deleted too.
System	supply	ON CASCADE	ON CASCADE	If there is no supply to monitor, then there is no use for a system.
Employee	bank	ON CASCADE	ON CASCADE	If there is no longer a place to work, there are no employees.
Admin	employee	ON CASCADE	ON CASCADE	If the employee is deleted, the admin should also be deleted because they must be an employee first.
ATM Tech	employee	ON CASCADE	ON CASCADE	If the employee is deleted, the technician should also be deleted because they must be an employee first.
Teller	employee	ON CASCADE	ON CASCADE	If the employee is deleted, the teller should also be deleted because they must be an employee first.
Supervisor	employee	ON CASCADE	ON CASCADE	If the employee is deleted, the supervisor should also be deleted because they must be an employee first.
Cash Checks	cash	ON CASCADE	ON CASCADE	Cash is needed to turn checks into cash.
Cash Checks	check	NO ACTION	ON CASCADE	No checks means there's nothing to convert.
Bills	cash	ON CASCADE	ON CASCADE	Bills are a defining table for Cash.
Coins	cash	ON CASCADE	ON CASCADE	Coins is a defining table for Cash
Session	user	ON CASCADE	ON CASCADE	If a user is deleted, then their sessions should be deleted as well.
Session	phone	SET NULL	ON CASCADE	The phone is not required to open a session
Receipt	session	ON CASCADE	ON CASCADE	If the session is deleted, then the receipt should be deleted as well.
Transaction	session	ON CASCADE	ON CASCADE	If the session is deleted, then there are no transactions.
Inquire	transaction	ON CASCADE	ON CASCADE	If the transaction is deleted, there is no inquiry.

Transfer	transaction	ON CASCADE	ON CASCADE	If the transaction is deleted, there is no transfer.
Deposit	transaction	ON CASCADE	ON CASCADE	If the transaction is deleted, there is no deposit.
Withdraw	transaction	ON CASCADE	ON CASCADE	If the transaction is deleted, there is no withdrawal.

Testing Table

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
Registered User	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Registered User	Update	Fail	Duplicate entry '4' for key	Update to a new key that is not being used already.
Language	Delete	Pass	None	None
Language	Update	Pass	None	None
Account	Delete	Pass	None	None
Account	Update	Pass	None	None
Checking	Delete	Pass	None	None
Checking	Update	Pass	None	None
Savings	Delete	Pass	None	None
Savings	Update	Pass	None	None
Handprint Scanner	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Handprint Scanner	Update	Pass	None	None
Keypad	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Keypad	Update	Pass	None	None
Debit Card	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an

				object not in use.
Debit Card	Update	Pass	None	None
Region	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Region	Update	Pass	None	None
Network	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Network	Update	Pass	None	None
Police Station	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Police Station	Update	Pass	None	None
Bank	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Bank	Update	Pass	None	None
ATM	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	This is the DELETE call that is messing up all of the above tests. Reorder or delete an object that is not being tested already.
ATM	Update	Pass	None	None
ATM updates Account	Delete	Pass	None	None
ATM updates	Update	Pass	None	None

Account				
Checks	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Checks	Update	Pass	None	None
Cash	Delete	Pass	None	None
Cash	Update	Fail	Duplicate entry '50' for key	50 already exists, try another key.
Cash Checks	Delete	Pass	None	None
Cash Checks	Update	Pass	None	None
Bills	Delete	Pass	None	None
Bills	Update	Pass	None	None
Coins	Delete	Pass	None	None
Coins	Update	Pass	None	None
Employee	Delete	Pass	None	None
Employee	Update	Pass	None	None
ATM Technician	Delete	Pass	None	None
ATM Technician	Update	Pass	None	None
Teller	Delete	Pass	None	None
Teller	Update	Pass	None	None
Admin	Delete	Pass	None	None
Admin	Update	Pass	None	None
Supervisor	Delete	Pass	None	None

Supervisor	Update	Pass	None	None
Supply	Delete	Pass	None	None
Supply	Update	Pass	None	None
System	Delete	Pass	None	None
System	Update	Pass	None	None
Phone	Delete	Pass	None	None
Phone	Update	Pass	None	None
Session	Delete	Fail	Cannot delete or update a parent row: a foreign key constraint fails	There's a DELETE call later in the test that is trying to delete the same thing by reference. Maybe reorder the DELETE's or try to delete an object not in use.
Session	Update	Pass	None	None
Transaction	Delete	Pass	None	None
Transaction	Update	Pass	None	None
Inquire	Delete	Pass	None	None
Inquire	Update	Pass	None	None
Transfer	Delete	Pass	None	None
Transfer	Update	Pass	None	None
Deposit	Delete	Pass	None	None
Deposit	Update	Pass	None	None
Withdraw	Delete	Pass	None	None
Withdraw	Update	Pass	None	None
Receipt	Delete	Pass	None	None
Receipt	Update	Pass	None	None