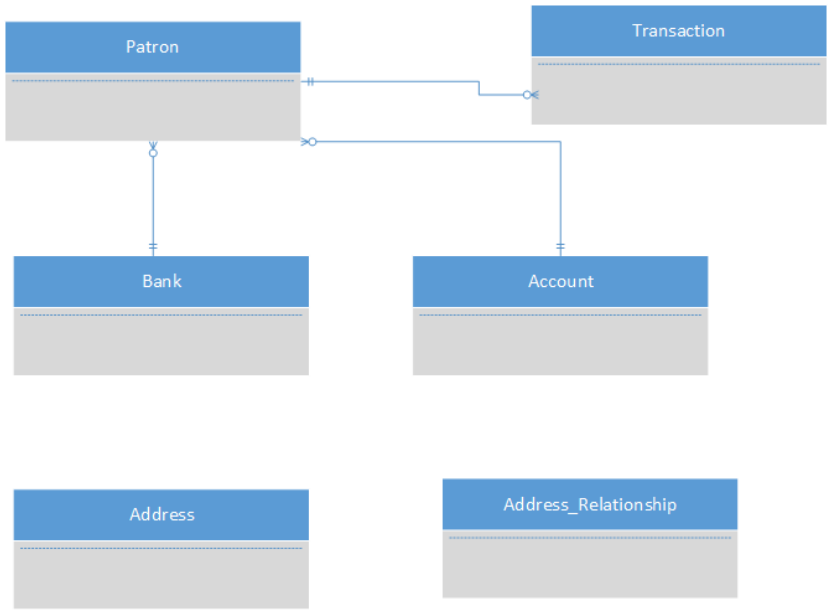
JDBC

Create a Bank - Patron CRUD application based on the following information: -

1) Build a database
    a. With the tables shown in the data.xlsx spreadsheet.
    b. Column level constraints are also shown in this spreadsheet.
    c. The tables have the following relationship.



2) Create the following classes and methods: -

Class: BankRepository		
S.No.	**Method**	**Description**
1	Result add(Patron)	Adds a new record in the Patron table.
2	Result remove(Patron)	Deletes a record from the Patron table.
3	Result update(Patron)	Updates a record in the Patron table.
4	Patron findPatron(id)	Finds a patron by id
5	List<Patron> findPatron(name)	Finds a patron by name
6	Result transact(Transaction)	Adds a record to the transaction table.
7	Transaction findTransaction(id)	Finds a transaction record by id
8	Result add(Bank bank)	Adds a record to the bank table
9	Result remove(Bank bank)	Deletes a record from the bank table
10	Result update(Bank)	Updates a record in the Bank table.
11	Bank findBank(Bank)	Finds a bank by id
12	List<Bank> findBank(name)	Returns a list of bank by name
13	Result add(Account)	Adds a record in the Account table
14	Result update(Account)	Updates a record in the Account table

15	Result delete(Account)	Deletes a record in the Account table
16	Acount findAccount(id)	Finds an account by id.
Enum: Result (SUCCESS, FAILURE)		
Class: Bank(id int, String name)		
Class: Patron(id int, name String, image byte[])		
Class: Transaction(id int, Account account, amount double, AccountType accountType)		
Enum: AccountType(CREDIT, DEBIT)		
Class: Account(int id, Bank bank, Patron patron)		
Address(int id, String street1, String street2, String city, String zip, String country, String zip)		

3) Create JUnit tests for only the following: -
    a. Result transact(Transaction) method in the BankRepository class.
    b. Transaction findTransaction(id) method in the BankRepository class.