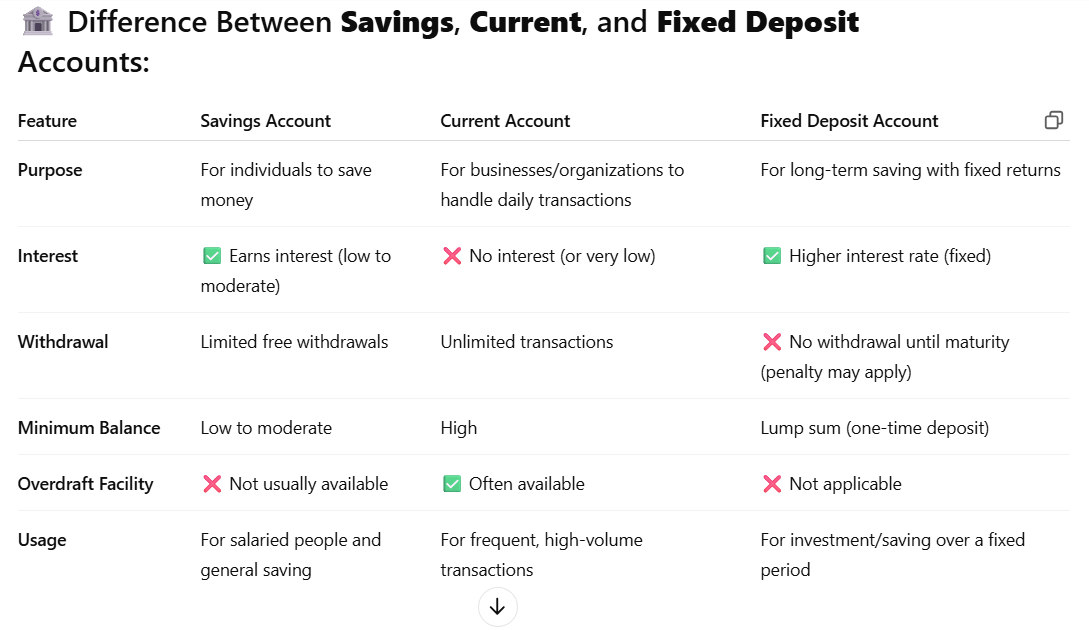
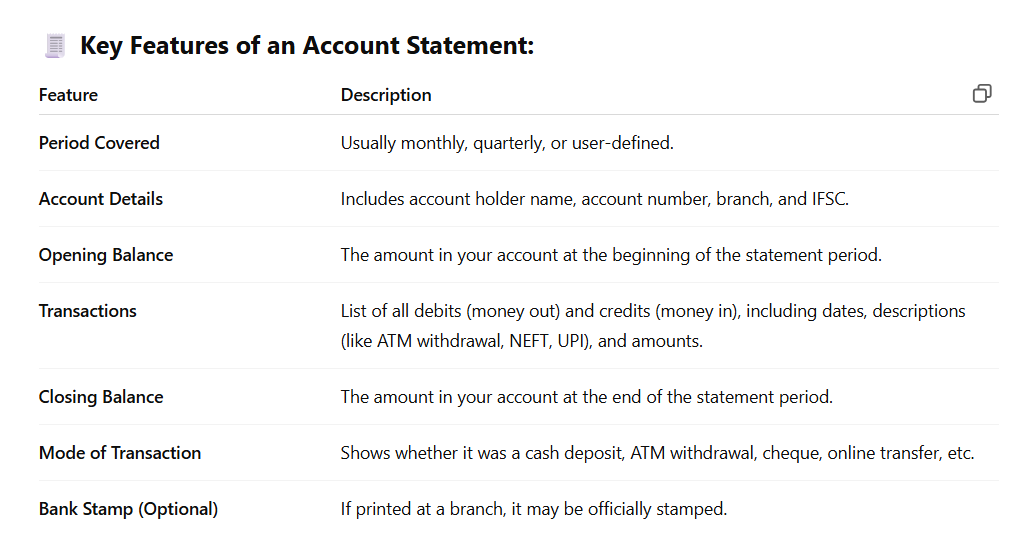
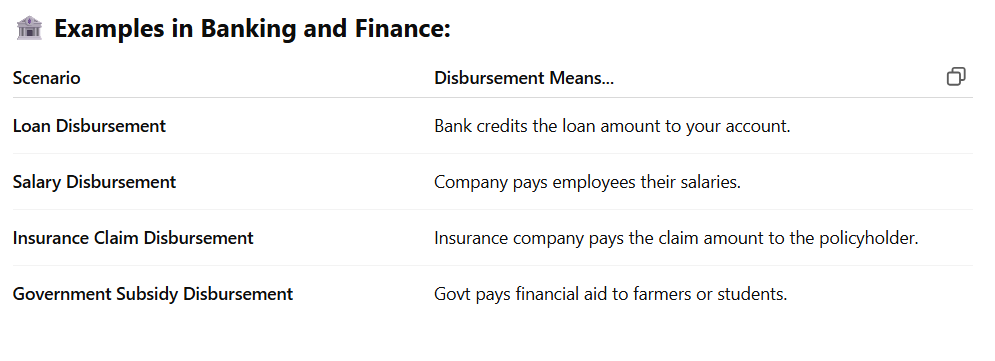
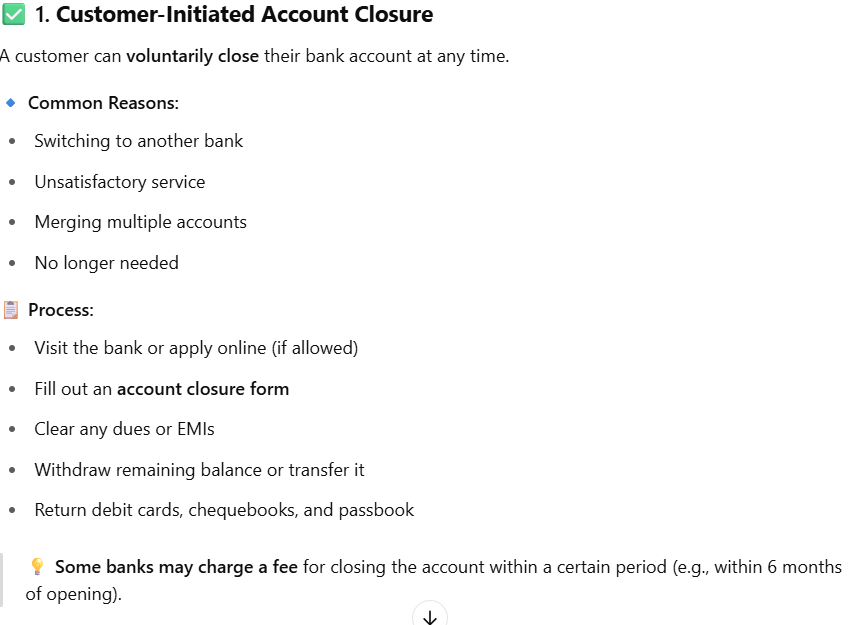
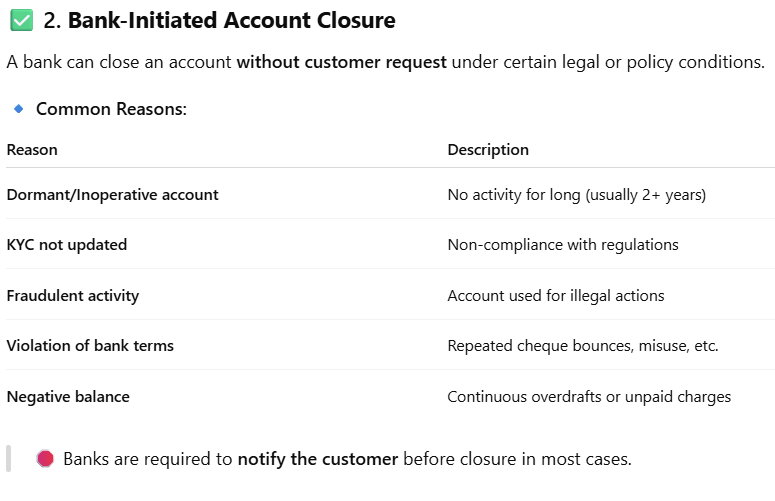
# Day 1 (14 - 05 - 2025)

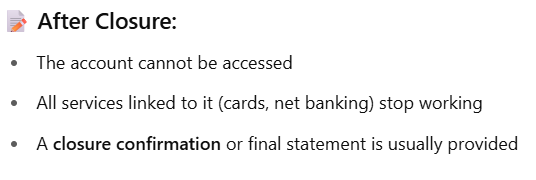


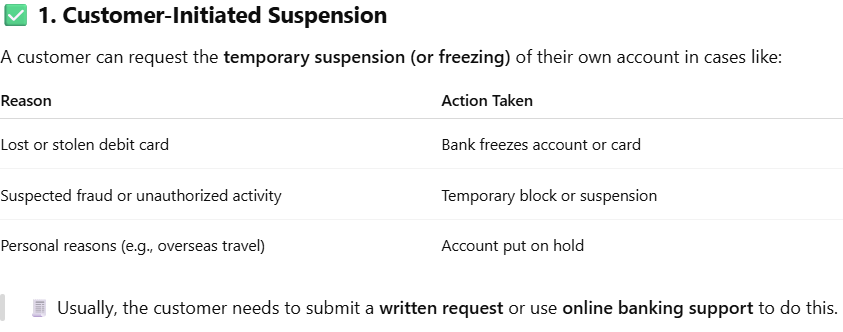


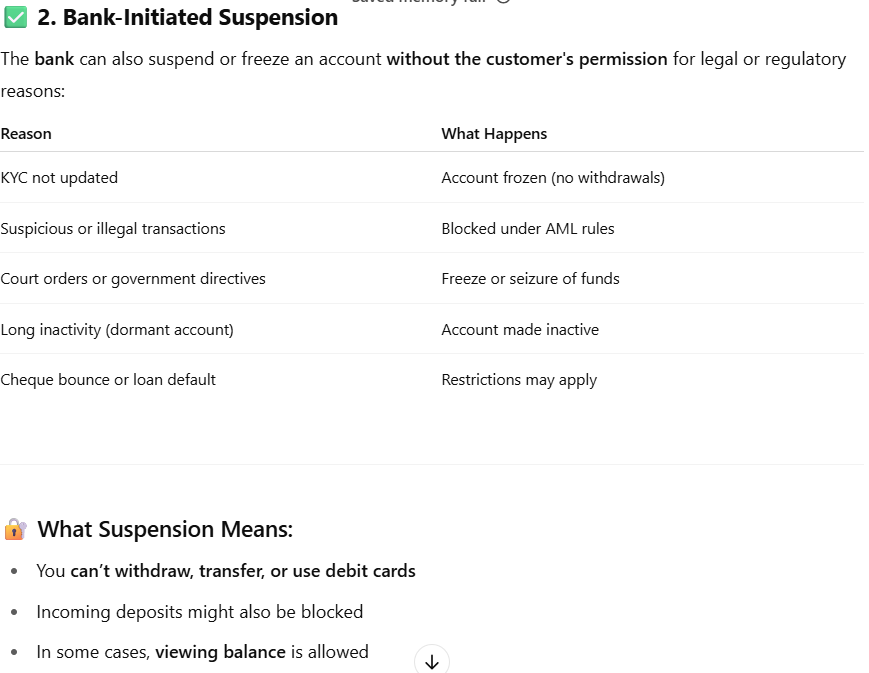


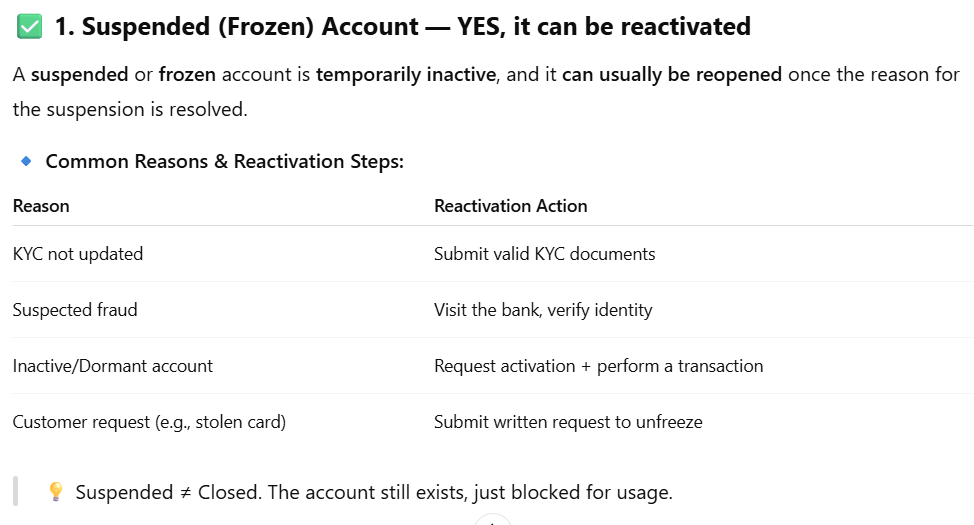


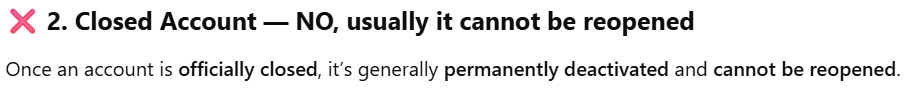




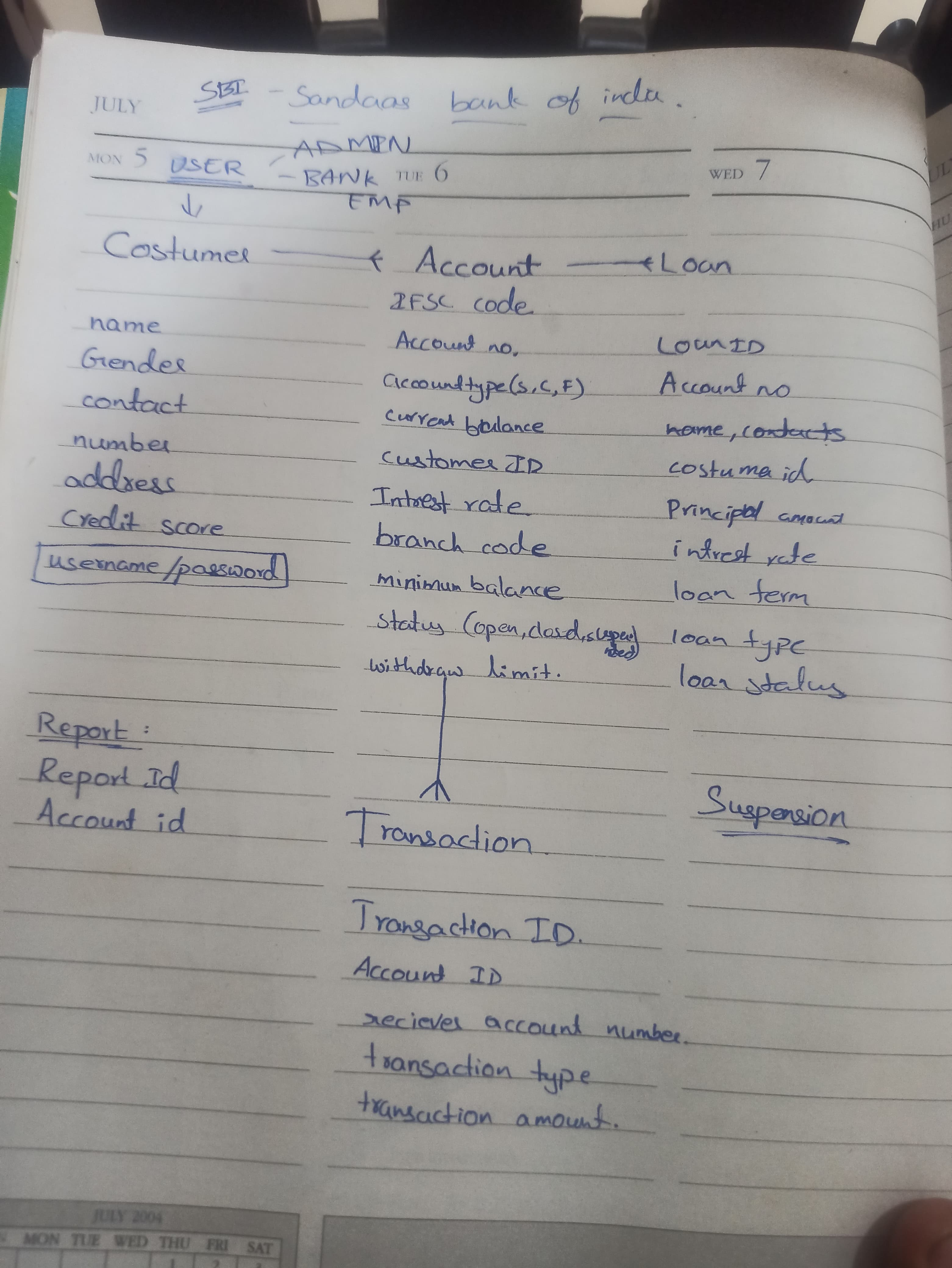


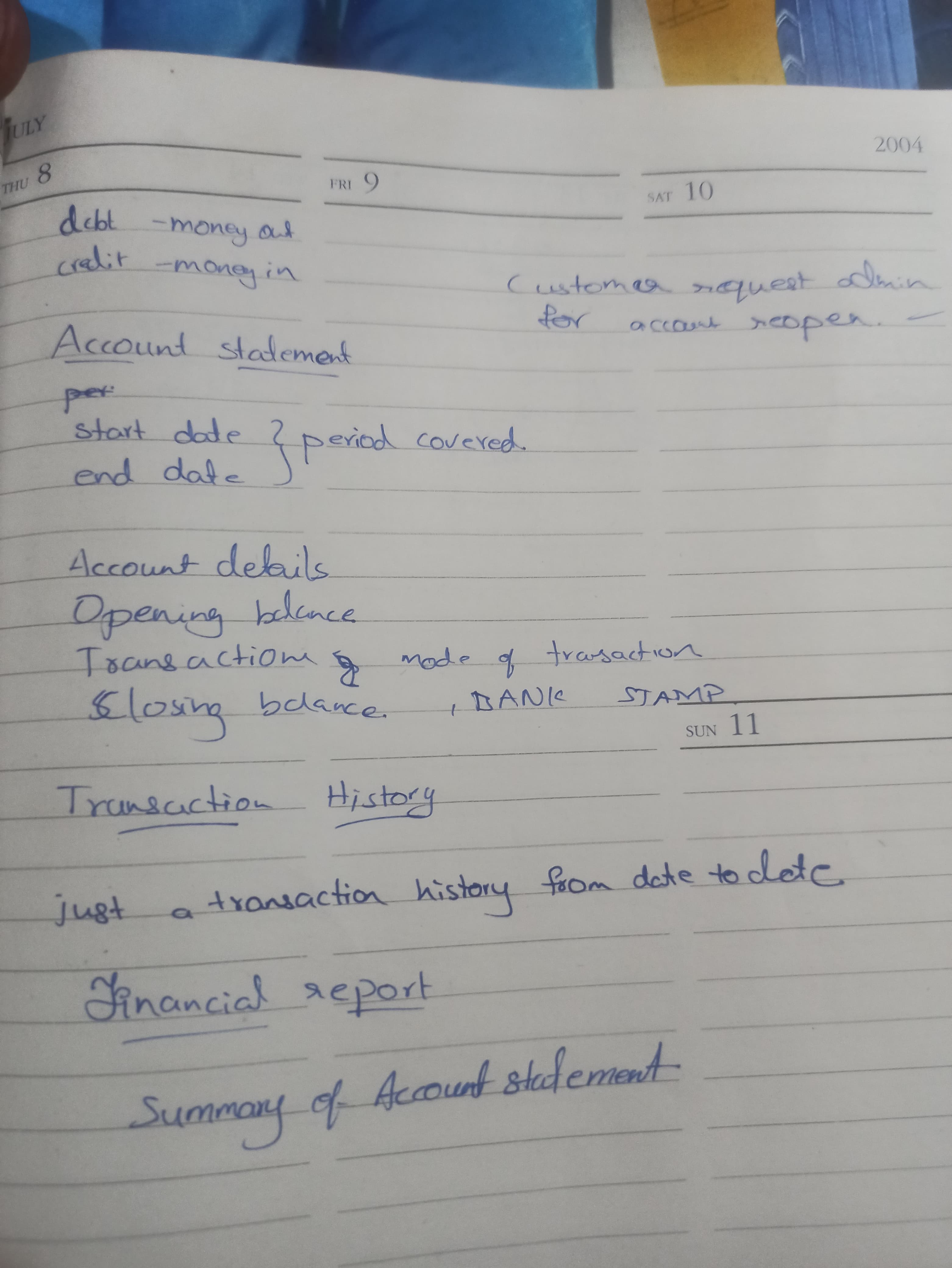




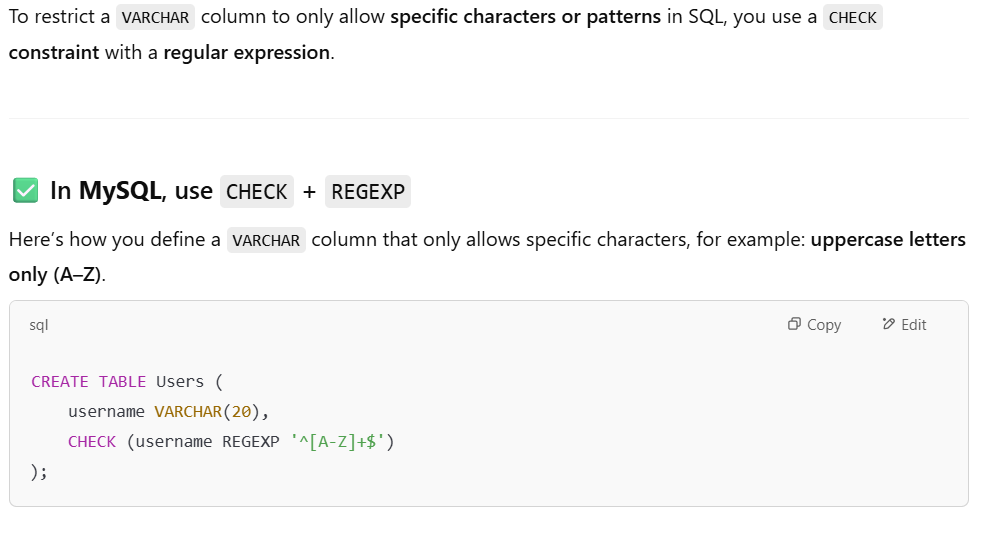


# Day 2 (15 - 05 - 2025)





# Day 3 (16 - 05 - 2025)

**REGULAR EXPRESSION CHECK IN SQL**





MUMBAI\_CITY - MVRK0000001

PUNE - MVRK0000002

NAGPUR - MVRK0000003

THANE - MVRK0000004

NASHIK - MVRK0000005

HYDERABAD - MVRK0000006

RANGA\_REDDY - MVRK0000007

MEDCHAL\_MALKAJGIRI - MVRK0000008

WARANGAL - MVRK0000009

NIZAMABAD - MVRK0000010

KHAMMAM - MVRK0000011

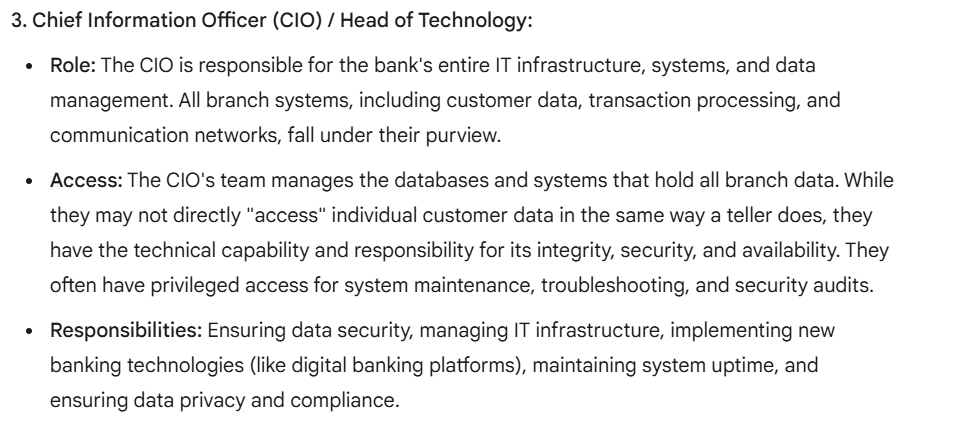
CHENNAI - MVRK0000012

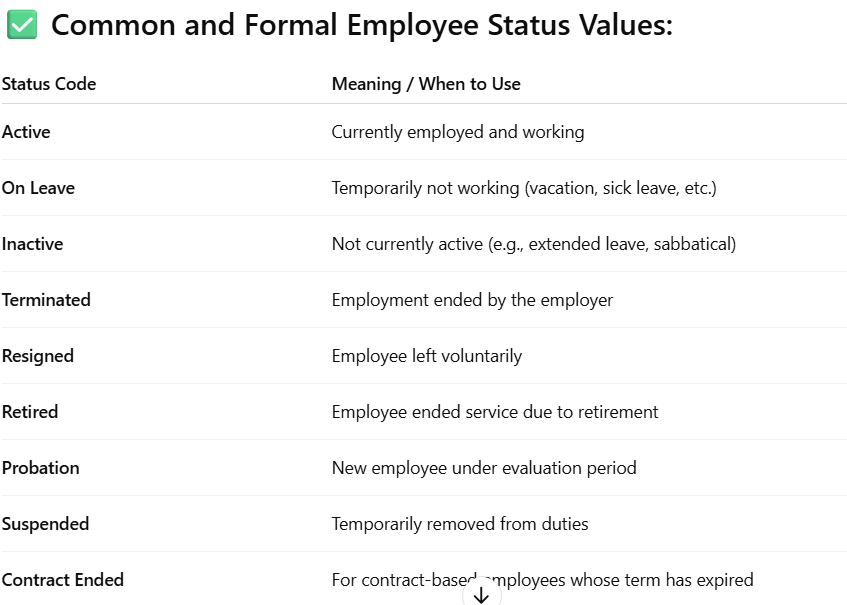
COIMBATORE - MVRK0000013

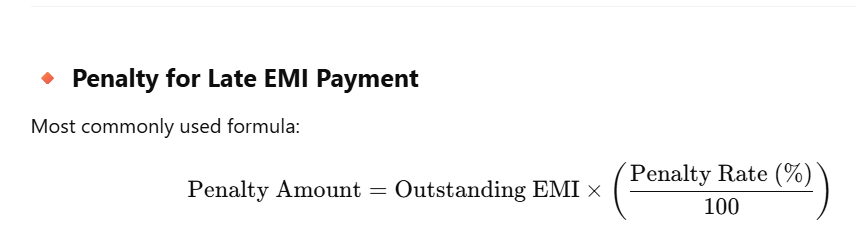
MADURAI - MVRK0000014

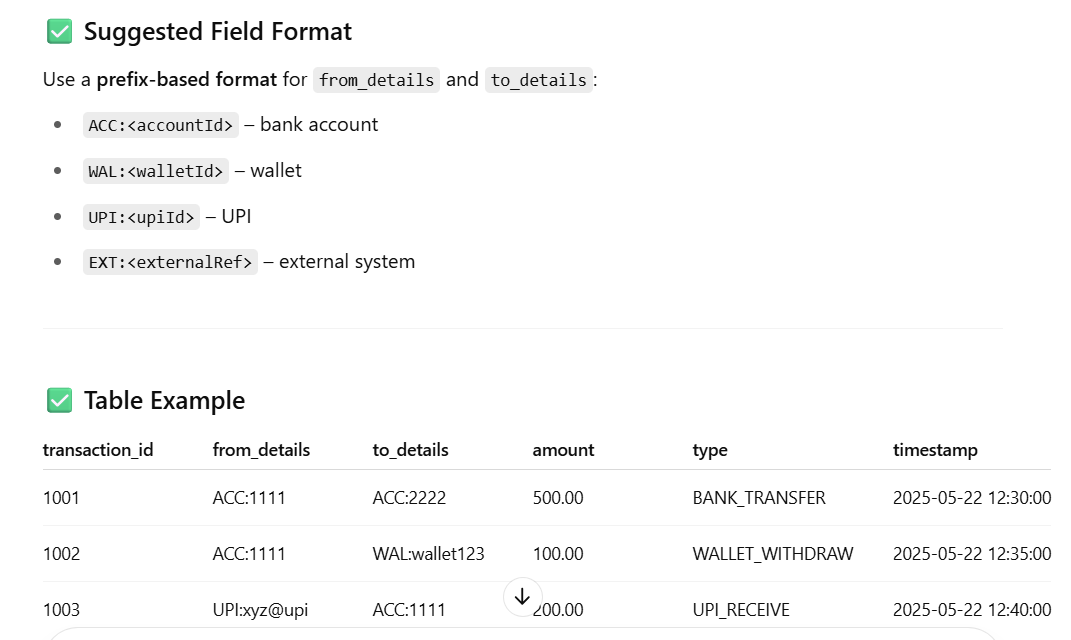
TIRUCHIRAPPALLI - MVRK0000015

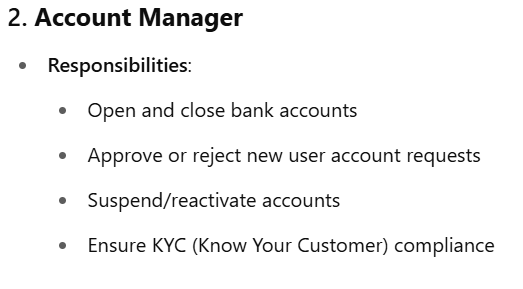
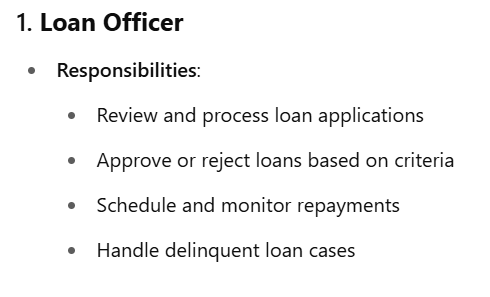
SALEM - MVRK0000016

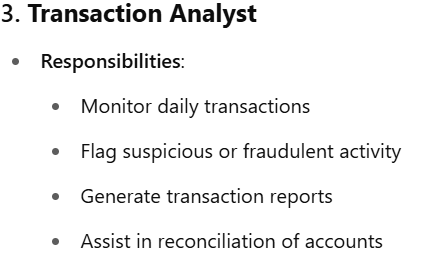


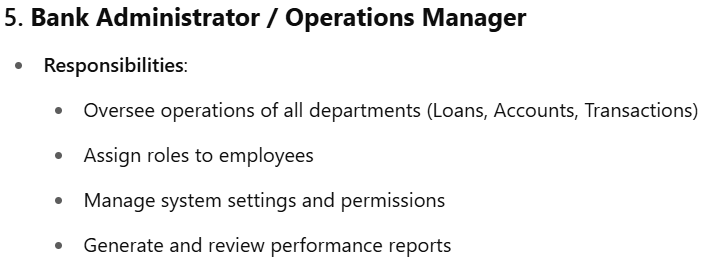












INSERT INTO account\_type (id,account\_type,interest\_rate,minimum\_balance,transaction\_limit,transaction\_amount\_limit,withdraw\_limit) VALUES (1,'SAVINGS', 3.00, 1000.00, 20, 50000.00, 10),(2,'CURRENT', 2.00, 2000.00, 50, 100000.00, 25),(3,'BUSINESS', 3.50, 10000.00, 0, 0.00, 0),(4,'FIXED', 5.50, 5000.00, 5, 20000.00, 1),(5,'JOINT\_SAVINGS', 3.00, 1500.00, 15, 40000.00, 8),(6,'JOINT\_CURRENT', 2.50, 2500.00, 40, 90000.00, 20),(7,'JOINT\_BUSINESS', 3.25, 15000.00, 0, 0.00, 0),(8,'JOINT\_FIXED', 5.75, 6000.00, 4, 25000.00, 1);

**BRANCHES**

INSERT INTO branch (id, ifsc, branch\_name, address, status, contact\_number, email) VALUES

(1, 'MVRK0000001', 'MUMBAI\_CITY', '123 Marine Drive, Mumbai, Maharashtra', 'ACTIVE', '9123456780', 'mumbai\_city.mvrkBank@gmail.com'),

(2, 'MVRK0000002', 'PUNE', '456 JM Road, Pune, Maharashtra', 'ACTIVE', '9123456781', 'pune.mvrkBank@gmail.com'),

(3, 'MVRK0000003', 'NAGPUR', '789 Sitabuldi, Nagpur, Maharashtra', 'ACTIVE', '9123456782', 'nagpur.mvrkBank@gmail.com'),

(4, 'MVRK0000004', 'THANE', '101 Ghodbunder Road, Thane, Maharashtra', 'ACTIVE', '9123456783', 'thane.mvrkBank@gmail.com'),

(5, 'MVRK0000005', 'NASHIK', '202 College Road, Nashik, Maharashtra', 'ACTIVE', '9123456784', 'nashik.mvrkBank@gmail.com'),

(6, 'MVRK0000006', 'HYDERABAD', '303 Banjara Hills, Hyderabad, Telangana', 'ACTIVE', '9123456785', 'hyderabad.mvrkBank@gmail.com'),

(7, 'MVRK0000007', 'RANGA\_REDDY', '404 LB Nagar, Ranga Reddy, Telangana', 'ACTIVE', '9123456786', 'ranga\_reddy.mvrkBank@gmail.com'),

(8, 'MVRK0000008', 'MEDCHAL\_MALKAJGIRI', '505 Kompally, Medchal-Malkajgiri, Telangana', 'ACTIVE', '9123456787', 'medchal\_malkajgiri.mvrkBank@gmail.com'),

(9, 'MVRK0000009', 'WARANGAL', '606 Hanamkonda, Warangal, Telangana', 'ACTIVE', '9123456788', 'warangal.mvrkBank@gmail.com'),

(10, 'MVRK0000010', 'NIZAMABAD', '707 Bodhan Road, Nizamabad, Telangana', 'ACTIVE', '9123456789', 'nizamabad.mvrkBank@gmail.com'),

(11, 'MVRK0000011', 'KHAMMAM', '808 Wyra Road, Khammam, Telangana', 'ACTIVE', '9123456790', 'khammam.mvrkBank@gmail.com'),

(12, 'MVRK0000012', 'CHENNAI', '909 Anna Salai, Chennai, Tamil Nadu', 'ACTIVE', '9123456791', 'chennai.mvrkBank@gmail.com'),

(13, 'MVRK0000013', 'COIMBATORE', '111 Avinashi Road, Coimbatore, Tamil Nadu', 'ACTIVE', '9123456792', 'coimbatore.mvrkBank@gmail.com'),

(14, 'MVRK0000014', 'MADURAI', '222 Alagar Kovil Road, Madurai, Tamil Nadu', 'ACTIVE', '9123456793', 'madurai.mvrkBank@gmail.com'),

(15, 'MVRK0000015', 'TIRUCHIRAPPALLI', '333 Rockfort, Tiruchirappalli, Tamil Nadu', 'ACTIVE', '9123456794', 'tiruchirappalli.mvrkBank@gmail.com'),

(16, 'MVRK0000016', 'SALEM', '444 Fairlands, Salem, Tamil Nadu', 'ACTIVE', '9123456795', 'salem.mvrkBank@gmail.com');

//Validate Account opening application dto object

AccountOpeningApplicationDtoValidation.*validateAccountOpeningApplicationDto*(applicationDto);

//Check whether given fields are null

if(branchName==null) {

throw new InvalidInputException("Invalid Branch Name provided..!!!");

}

if(bankAccountType==null) {

throw new InvalidInputException("Invalid Bank Account Type provided..!!!");

}

//Get branch by branch name

Branch branch=branchRepository.getByName(branchName);

//If no branch available with given name throw exception

if(branch==null) {

throw new ResourceNotFoundException("Invalid Branch Name...!!!");

}

//Extract Account opening application

AccountOpeningApplication application=applicationDto.getApplication();

//Set branch for the application

application.setBranch(branch);

//Get account type by its name

AccountType accountType= accountTypeRepository.getAccountTypeByName(bankAccountType);

//Check whether account type exists in database

if(accountType==null) {

throw new ResourceNotFoundException("Invalid Bank Account type provided Name...!!!");

}

//Set account type to the Account opening application

application.setAccountType(accountType);

application.setStatus(*ApplicationStatus*.***PENDING***);

/\*

\* Now here check if account is joint type,

\* if yes account holders should be greater than or equal to 2

\* then fetch the customer details of all the account holders using their mail and

\* store in list of customers

\* if the current Actor(customer) is not in the list of customers then account opening application cannot be created due to identity invalid

\* if all the conditions are met then save account holders in account holder and get the ids

\* save updated account holders in the list

\* add application to database

\* now update CustomerAccountHolderApplication with account holder id, customer id, application id

\* \*/

AccountOpeningApplicationDtoValidation.*validateListSize*(accountType.getAccountType(), applicationDto.getAccountHolderList());

List<Customer> customerList=new ArrayList<>();

for(AccountHolder a : applicationDto.getAccountHolderList()) {

Customer customer=customerRepository.getByContactNumber(a.getContactNumber());

if(customer==null) {

throw new ResourceNotFoundException("Cannot submit application,No customer with given phone number"+a.getContactNumber());

}

customerList.add(customer);

}

Customer presentCustomer=customerRepository.getByUsername(username);

Boolean isPresent=false;

for(Customer c: customerList) {

if(c.equals(presentCustomer)) {

isPresent=true;

}

}

if(isPresent==false) {

throw new IdentityNotMatchException("Customer's identity is absent in the given account holder list...!!! ");

}

application.setApplicationDateTime(LocalDateTime.*now*());

//Add a condition to check weather application is already available or not

//Boolean isAlreadyAvailable=true;

//SAVING step1: add account holders

applicationDto.setAccountHolderList(accountHolderRepository.saveAll(applicationDto.getAccountHolderList()));

//SAVING step2: add application

application=accountOpeningApplicationRepository.save(application);

applicationDto.setApplication(application);

//SAVING step3: Create list of CustomerAccountOpeningApplication and save

List<CustomerAccountOpeningApplication> customerAccountOpeningApplication=new ArrayList<>();

//Create CustomerAccountOpeningApplication object and add it to list

for(int i=0;i<customerList.size();i++) {

AccountHolder accountHolder=applicationDto.getAccountHolderList().get(i);

CustomerAccountOpeningApplication customerAccountOpeningApplicationTemp=new CustomerAccountOpeningApplication();

customerAccountOpeningApplicationTemp.setAccountHolder(accountHolder);

customerAccountOpeningApplicationTemp.setCustomer(customerList.get(i));

customerAccountOpeningApplicationTemp.setAccountOpeningApplication(application);

if(customerList.get(i).equals(presentCustomer)) {

customerAccountOpeningApplicationTemp.setCustomerApproval(*ApplicationStatus*.***ACCEPTED***);

}

else {

customerAccountOpeningApplicationTemp.setCustomerApproval(*ApplicationStatus*.***PENDING***);

}

customerAccountOpeningApplication.add(customerAccountOpeningApplicationTemp);

}//For loop

customerAccountOpeningApplicationRepository.saveAll(customerAccountOpeningApplication);