**Banking App Requirements**

**Functions**

|  |  |  |
| --- | --- | --- |
| **Account (Debit Card)** | **Fixed Deposit** | **Credit Card** |
| Open account | Open account | Open account |
| Close account | Close account | Close account |
| Enable deposit |  | Enable deposit |
| Enable Withdrawal | Enable Withdrawal (Closure of account) | Enable Withdrawal |
|  | Policy Start Date and Maturity Date | Credit Card Start Date and End Date |
| Check Balance | Check Balance | Check Balance |
| Per Transaction Limit |  | Credit Limit |
| Monthly statements | Monthly statement | Monthly statements |
| Account Transaction History | Account Transaction History | Account Transaction History |
| User Detail Update Log | User Detail Update Log | User Detail Update Log |
| UpdateName | UpdateName | UpdateName |
| UpdateAddr | UpdateAddr | UpdateAddr |
| UpdateEmailAddr | UpdateEmailAddr | UpdateEmailAddr |
| UpdateContactNo | UpdateContactNo | UpdateContactNo |
| Messaging user when per transaction amount exceeds a prescribed level | Messaging user when policy expiry is due | Messaging user when monthly statement is available |
| User Details/Updating of user details | User Details/Updating of user details | User Details/Updating of user details |
| Request transaction limit change |  | Request credit limit change |
| Update Transaction limit |  | Update Credit Limit |
| Calculation of Interest | Calculation of Interest | Calculation of Interest |
|  | Renewal Option -  Auto renewal at expiry  On client request  Auto closure at expiry |  |
|  |  | Payment due date |
|  | Link to Debit card account |  |

**Details Requirements**

Primary key (PK), non duplicate values, Foreign key (FK), Not Null(NN)

**Data**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Java/ My SQL Datatype** | **Properties** | **Customer**  **Details** |
| **Customer** | String  varchar(10) | PK, NN | CustomerUIN |
|  | String  varchar | NN | CustomerName |
|  | String  varchar | NN | CustomerAddr |
|  | int  INT | NN | CustomerContactNo |
|  | String  varchar |  | CustomerEmailAddr |
|  | String  varchar | NN | CustomerResidentStatus   1. Citizen 2. PR 3. Foreigner |
|  | Byte  TINYINT | NN | CustomerStatus   1. Active 2. Dormant |
|  | java.sql.Date  DATETIME | NN | LastUpdateDateTime |
|  | java.sql.Date  DATETIME | NN | CreationDateTime |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Java/ My SQL Datatype** | **Properties** | **Customer**  **Details** |
| **Customer Account** | String  varchar | PK, NN | AccType   1. Debit Card 2. Fixed Deposit 3. Credit Card |
|  | int  INT | PK, NN, Auto-increment | AccNo |
|  | String  varchar(10) | FK, NN | AccHolderUIN |
|  | Byte  TINYINT | NN | CVV |
|  | Byte  TINYINT | NN | DebitCardPin |
|  | java.sql.Date  DATETIME | NN | LastUpdateDateTime |
|  | java.sql.Date  DATETIME | NN | CreationDateTime |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Java/ My SQL Datatype** | **Properties** | **AccType: 1**  **Debit Card** | **AccType: 2**  **Fixed Deposit** | **AccType: 3**  **Credit Card** |
| **Account Details** | String  varchar | PK, NN | AccType   1. DebitCard | AccType   1. Fixed Deposit | AccType   1. CreditCard |
|  | int  INT | PK, NN | AccNo | AccNo | AccNo |
|  | double DOUBLE |  | AccBal | AccBal | CreditLimitBal |
|  | double DOUBLE | NN, >0 | MinBal |  | SpendLimit |
|  | Byte  TINYINT | NN | AccStatus   1. Active 2. Dormant 3. Freeze | AccStatus   1. Active 2. Dormant 3. Freeze | AccStatus   1. Active   4. Suspended  5. Cancelled |
|  | double  DOUBLE | =0 No notification.  >0 Notification | TransNotifLimit |  | TransNotifLimit |
|  | String  varchar | NN |  | RenewalPrefer   1. Principal + Interest = new Principal 2. Principal = new Principal Interest transfer to Debit Acc 3. Wait for user instruction upon expiry 4. Close FD. Transfer Bal to Debit Acc |  |
|  | java.sql.Date  DATE | NN |  | StartDate | StartDate |
|  | java.sql.Date  DATE | NN |  | EndDate | EndDate |
|  | Int  INT |  |  | DebitAccNo | DebitAccNo |
|  | Byte  TINYINT | >0 |  |  | PaymentTerms |
|  | java.sql.Date  DATETIME | NN | LastUpdateDateTime | LastUpdateDateTime | LastUpdateDateTime |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Java/ My SQL Datatype** | **Properties** | **AccType: 1**  **Debit Card** | **AccType: 2**  **Fixed Deposit** | **AccType: 3**  **Credit Card** |
| **Account Transactions** | String  varchar | PK, NN | AccType  1.DebitCard | AccType  2.Fixed Deposit | AccType  3.CreditCard |
|  | int  INT | PK, NN | AccNo | AccNo | AccNo |
|  | double  DOUBLE |  | TransRec (+/-) | TransRec (+/-) | TransRec (+/-) |
|  | String  varchar | NN | TransType   1. Deposit 2. Withdrawal 3. Interest 4. Transfer 5. Debit Card Trans | TransType   1. Deposit 2. Withdrawal 3. Interest 4. Transfer | TransType   1. Payment 2. Withdrawal 3. Interest 4. Transfer 5. Credit Card Trans 6. Annual Fee 7. Late Payment Fee 8. FX Charges 9. Auto Payment |
|  | String  varchar | NN | TransDesc | TransDesc | TransDesc |
|  | java.sql.Date  DATETIME | NN | CreationDateTime | CreationDateTime | CreationDateTime |

CRUD = Operations

new/update/delete/find

Types

-attributes/data

Data

Java datatypes/ SQL datatypes

Biz Operations

Tables Schema

Object Hierarchy(interface?)

Test Scenarios (validation of data)

Tx vs Op Data

Frequently used Static

Consistent Secure

https://dev.mysql.com/doc/ndbapi/en/mccj-using-clusterj-mappings.html

|  |  |
| --- | --- |
| **Java Data Type** | **MySQL Column Type** |
| boolean, [Boolean](http://java.sun.com/javase/6/docs/api/java/lang/Boolean.html) | [BIT(1)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html) |
| byte, [Byte](http://java.sun.com/javase/6/docs/api/java/lang/Byte.html) | [BIT(1)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html) to [BIT(8)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html), [TINYINT](https://dev.mysql.com/doc/refman/8.0/en/integer-types.html) |
| short, [Short](http://java.sun.com/javase/6/docs/api/java/lang/Short.html) | [BIT(1)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html) to [BIT(16)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html), [SMALLINT](https://dev.mysql.com/doc/refman/8.0/en/integer-types.html), [YEAR](https://dev.mysql.com/doc/refman/8.0/en/year.html) |
| int, [Integer](http://java.sun.com/javase/6/docs/api/java/lang/Integer.html) | [BIT(1)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html) to [BIT(32)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html), [INT](https://dev.mysql.com/doc/refman/8.0/en/integer-types.html) |
| long, [Long](http://java.sun.com/javase/6/docs/api/java/lang/Long.html) | [BIT(1)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html) to [BIT(64)](https://dev.mysql.com/doc/refman/8.0/en/bit-type.html), [BIGINT](https://dev.mysql.com/doc/refman/8.0/en/integer-types.html), [BIGINT UNSIGNED](https://dev.mysql.com/doc/refman/8.0/en/integer-types.html) |
| float, [Float](http://java.sun.com/javase/6/docs/api/java/lang/Float.html) | [FLOAT](https://dev.mysql.com/doc/refman/8.0/en/floating-point-types.html) |
| double, [Double](http://java.sun.com/javase/6/docs/api/java/lang/Double.html) | [DOUBLE](https://dev.mysql.com/doc/refman/8.0/en/floating-point-types.html) |
| [java.math.BigDecimal](http://java.sun.com/javase/6/docs/api/java/math/BigDecimal.html) | [NUMERIC](https://dev.mysql.com/doc/refman/8.0/en/fixed-point-types.html), [DECIMAL](https://dev.mysql.com/doc/refman/8.0/en/fixed-point-types.html) |
| [java.math.BigInteger](http://java.sun.com/javase/6/docs/api/java/math/BigInteger.html) | [NUMERIC](https://dev.mysql.com/doc/refman/8.0/en/fixed-point-types.html) (precision = 0), [DECIMAL](https://dev.mysql.com/doc/refman/8.0/en/fixed-point-types.html) (precision = 0) |

|  |  |
| --- | --- |
| **Java Data Type** | **MySQL Column Type** |
| [Java.util.Date](http://java.sun.com/javase/6/docs/api/java/util/Date.html) | [DATETIME](https://dev.mysql.com/doc/refman/8.0/en/datetime.html), [TIMESTAMP](https://dev.mysql.com/doc/refman/8.0/en/datetime.html), [TIME](https://dev.mysql.com/doc/refman/8.0/en/time.html), [DATE](https://dev.mysql.com/doc/refman/8.0/en/datetime.html) |
| [Java.sql.Date](http://java.sun.com/javase/6/docs/api/java/sql/Date.html) | [DATE](https://dev.mysql.com/doc/refman/8.0/en/datetime.html) |
| [Java.sql.Time](http://java.sun.com/javase/6/docs/api/java/sql/Time.html) | [TIME](https://dev.mysql.com/doc/refman/8.0/en/time.html) |
| [Java.sql.Timestamp](http://java.sun.com/javase/6/docs/api/java/sql/Timestamp.html) | [DATETIME](https://dev.mysql.com/doc/refman/8.0/en/datetime.html), [TIMESTAMP](https://dev.mysql.com/doc/refman/8.0/en/datetime.html) |

|  |  |
| --- | --- |
| **Java Data Type** | **MySQL Column Type** |
| [String](http://java.sun.com/javase/6/docs/api/java/lang/String.html) | [CHAR](https://dev.mysql.com/doc/refman/8.0/en/char.html), [VARCHAR](https://dev.mysql.com/doc/refman/8.0/en/char.html), [TEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html) |
| byte[] | [BINARY](https://dev.mysql.com/doc/refman/8.0/en/binary-varbinary.html), [VARBINARY](https://dev.mysql.com/doc/refman/8.0/en/binary-varbinary.html), [BLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html) |

Biz Operations

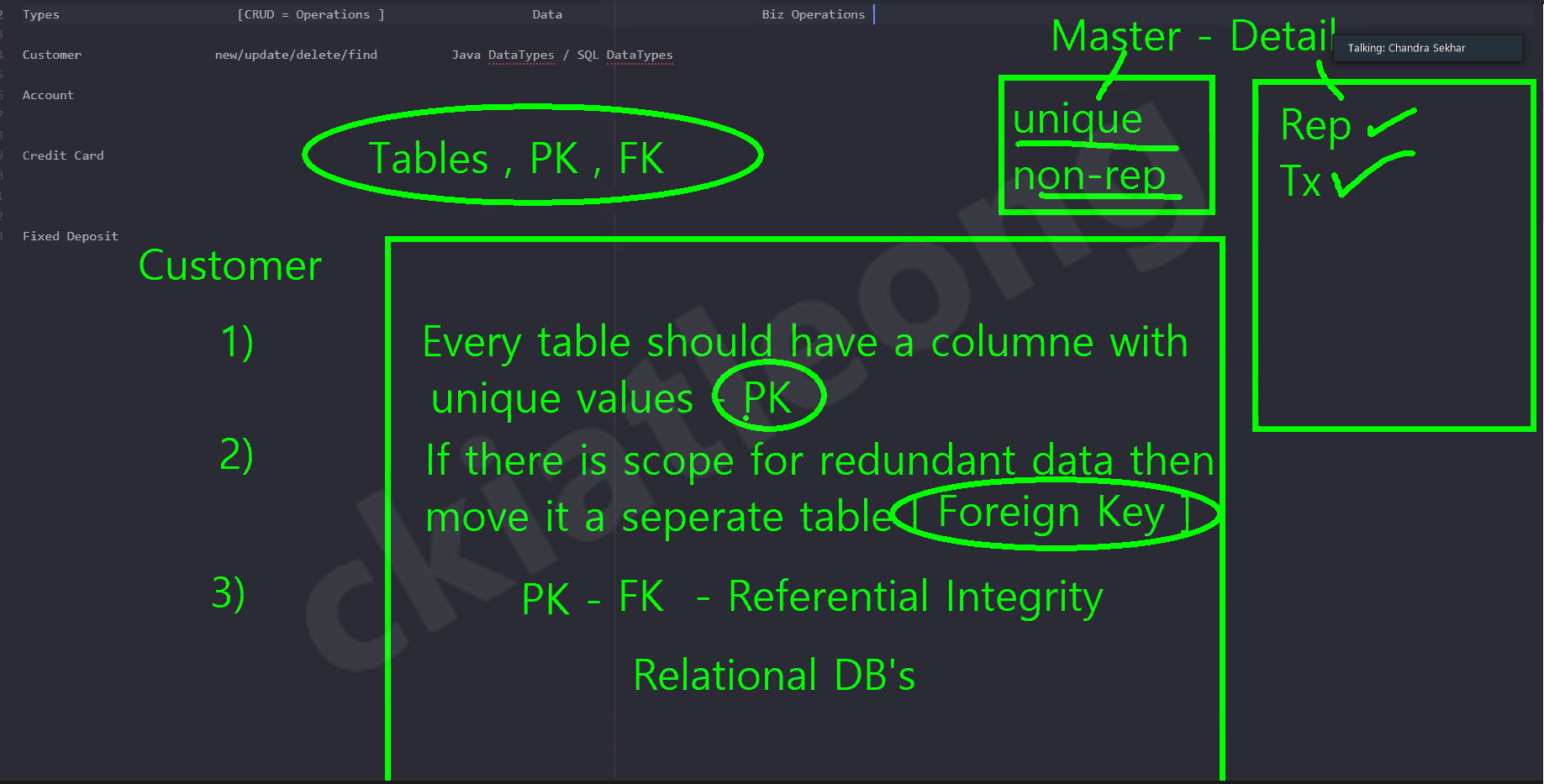


Table schema

Object hierarchy -- interface, ...

Test scenarios and data validation

**Deposit**

Start

1. Customer informs BankExec at the counter he/she wants to deposit some money and handed over a deposit slip.
2. BankExec verify Customer’s personal details.
3. BankExec asks for deposit amount and the deposit account number.
4. BankExec counted the cash handed over by Customer.
5. BankExec verified and obtained verbal confirmation from the Customer on the cash amount received.
6. BankExec executed the Deposit function on the system.
7. BankExec presented a printed copy of the deposit slip to Customer.
8. BankExec asks Customer if there are other services required. Perform other service(s) if positive response.

End

**Withdrawal**

Start

1. Customer informs BankExec at the counter he/she wants to withdraw some money and handed over a withdrawal slip.
2. BankExec verify Customer’s personal details.
3. BankExec asks for withdrawal amount and the withdrawal account number.
4. BankExec executed the Withdrawal function on the system.
5. BankExec printed copy of the withdrawal slip.
6. BankExec counted the cash to be handed over to Customer.
7. BankExec handed over the cash and withdrawal slip to Customer.
8. Customer verified and verbally confirmed the cash amount received.
9. BankExec asks Customer if there are other services required. Perform other service(s) if positive response.

End