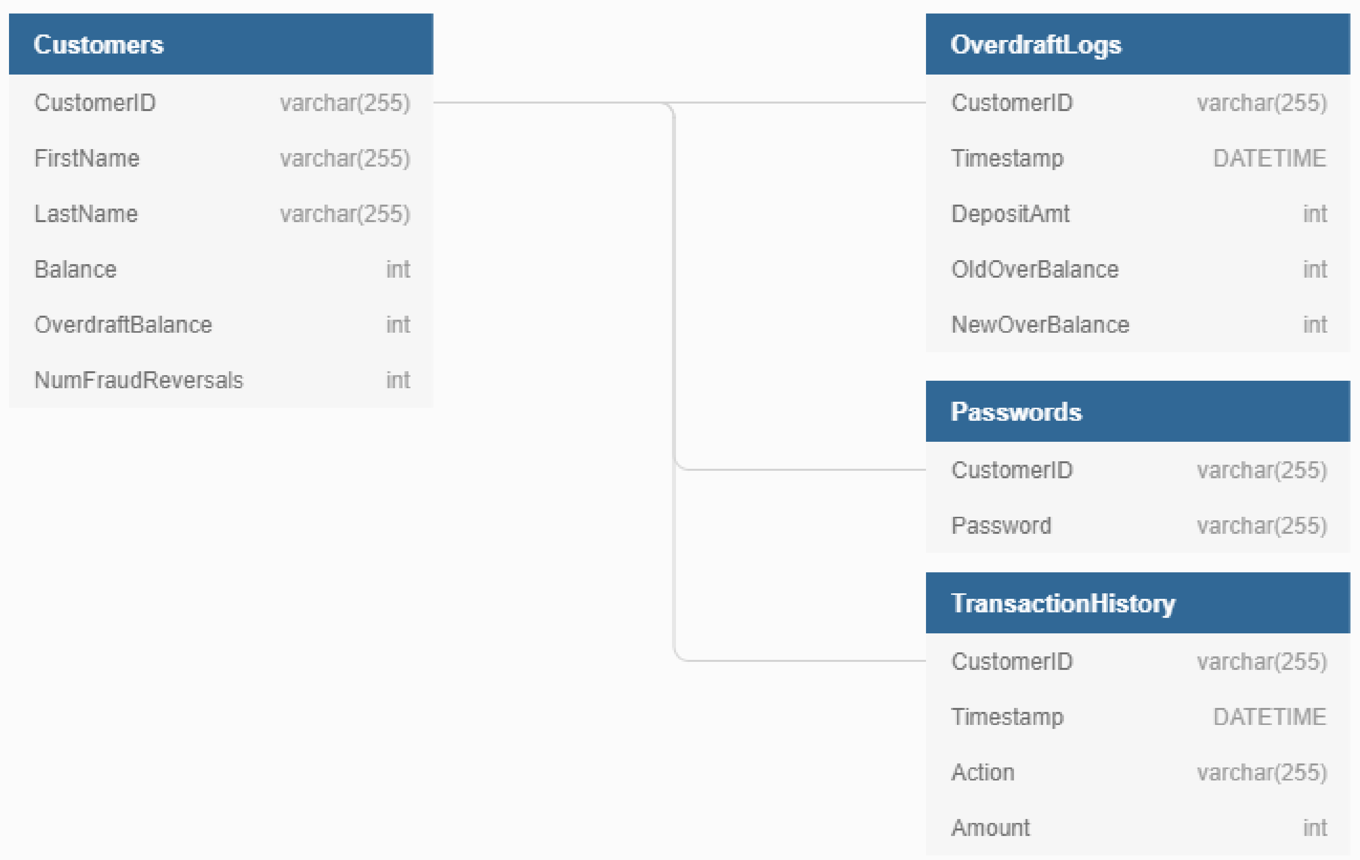
Testudo Bank Database Schema Documentation

Owner: Adithya Solai (adithyasolai7@gmail.com)

Schema Diagram

The Testudo Bank application uses a **MySQL** database with the following schema:



Customers	
CustomerID	varchar(255)
FirstName	varchar(255)
LastName	varchar(255)
Balance	int
OverdraftBalance	int
NumFraudReversals	int

OverdraftLogs	
CustomerID	varchar(255)
Timestamp	DATETIME
DepositAmt	int
OldOverBalance	int
NewOverBalance	int

Passwords	
CustomerID	varchar(255)
Password	varchar(255)

TransactionHistory	
CustomerID	varchar(255)
Timestamp	DATETIME
Action	varchar(255)
Amount	int

Schema Definition

Customers Table

- `CustomerID` is a **primary key** for the `Customers` table because it uniquely identifies each row in the table since each Customer is only given one row entry in this table.
- `CustomerID` can be used as a foreign key in the `Passwords`, `TransactionHistory` and `OverdraftLogs` tables.
  - A **foreign key** means that `CustomerID` values from the `Customers` table will correspond to `CustomerID` values in these other tables. Therefore, you can fetch all passwords, transactions, and overdraft repayments belonging to a customer using their `CustomerID`.
- `Balance` and `OverdraftBalance` are `int` datatypes. Intuitively, this shouldn't support storage of a dollar amount value due to decimal values needed to represent cents. However, **the workaround is to treat this column as the # of pennies, not the # of dollars.** Real commercial banks implement balance in their DBs like this to prevent issues with floating-point roundoffs.
- `OverdraftBalance` starts at 0 for all customers, and is updated once a customer withdraws beyond their balance.
- `OverdraftBalance` should only ever store values >= 0 and <= 1000.
- `NumFraudReversals` is the # of Fraud Disputes that the customer has done. The customer's account is frozen after they have done 2 Fraud Disputes.

Passwords Table

- `CustomerID` is a primary key in the `Passwords` table since each customer can only have 1 password at a time.

TransactionHistory Table

- `CustomerID` is not a primary key in the `TransactionHistory` table because a single customer can do many transactions, which will all get logged in this table.
- `Action` column is used to specify whether the transaction was a `"Deposit"` or `"Withdraw"`.
- `Amount` stored as an `int` to follow the pennies paradigm used in the `Customers` table for columns like `Balance` and `OverdraftBalance`.

OverdraftLogs Table

- A table that logs every time a customer pays off some (or all) of their overdraft balance.
- `CustomerID` is not a primary key in the `OverdraftLogs` table because a single customer can do many repayments of their overdraft balance, which will each get logged in this time.

- `DepositAmt`, `OldOverBalance`, and `NewOverBalance` are all stored as an `int` to follow the pennies paradigm used in the `Customers` table for columns like `Balance` and `OverdraftBalance`.
- The **candidate key** of this table is `CustomerID, Timestamp, NewOverBalance`.
    - A **candidate key** is the minimal group of columns needed to uniquely identify every row in a table.
    - Using just `CustomerID` and `Timestamp` columns is not enough to be a **candidate key** because the `DATETIME` MySQL data type used for `Timestamp` is in the form `YYYY-MM-DD hh:mm:ss`, meaning this table is granular only up to seconds. This means there can be two repayments from the same customer that are logged with the same timestamp.
    - The `DepositAmt` of two repayments occurring within the same second could be the same, so we don't gain anything from including this column in our candidate key.
    - The `NewOverBalance` must be different after each time a customer makes a re-payment, so this can be used to distinguish payments made within the same second.
    - **This forces us to enforce that any deposits made must be > 0 to avoid the same `NewOverBalance` in two re-payments made within the same second.**
    - There is an edge case where a customer can make a repayment, withdraw more on credit back to the previous `OverdraftBalance` amount, and then make another repayment for the same amount all within one second. We will assume this is not possible.