Gold Loan Management

Normalization:

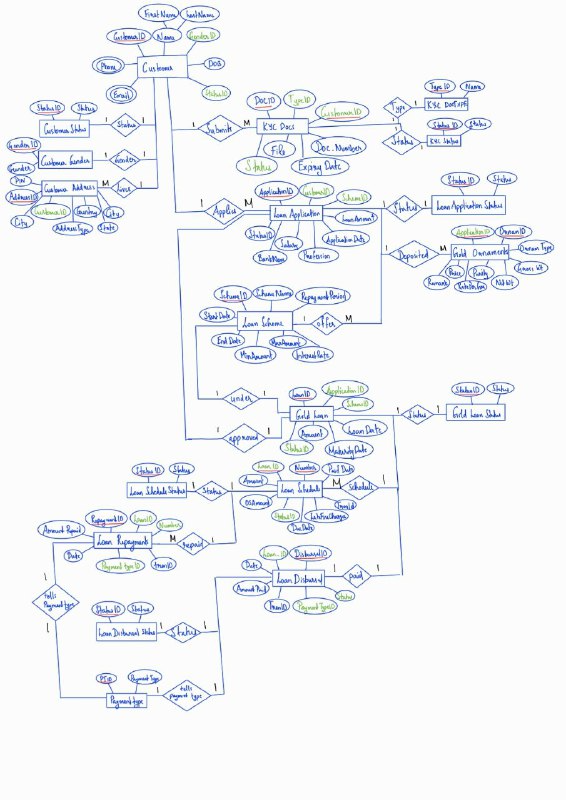
A large database defined as a single relation may result in data duplication. This repetition of data may result in:

* Making relations very large.
* It isn't easy to maintain and update data as it would involve searching many records in relation.
* Wastage and poor utilization of disk space and resources.
* The likelihood of errors and inconsistencies increases.

|  |  |
| --- | --- |
| **Normal Form** | **Description** |
| [1NF](https://www.javatpoint.com/dbms-first-normal-form) | A relation is in 1NF if it contains an atomic value. |
| [2NF](https://www.javatpoint.com/dbms-second-normal-form) | A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key. |
| [3NF](https://www.javatpoint.com/dbms-third-normal-form) | A relation will be in 3NF if it is in 2NF and no transition dependency exists. |
| BCNF | A stronger definition of 3NF is known as Boyce Codd's normal form. |

Normalization is a process of decomposing the relations into relations with fewer attributes.

ER-Diagram



1. **For Customer Entity:**

* Attributes:
  + CustomerID (INT, PK)
  + First\_Name (VARCHAR(50), NOT NULL)
  + Last\_Name (VARCHAR(50), NOT NULL)
  + GenderID (TINYINT, FK)
  + DOB (DATE)
  + StatusID (TINYINT, FK)
  + PhoneNumber
  + Email
* Functional Dependencies:
  + CustomerID -> First\_Name, Last\_Name, GenderID, DOB, StatusID.Phone Number, Email
  + **1NF(First Normal Form)**

The Customer table provided is not in the first normal form since there are multivalued attributes like email and phone number.

On Decomposition we get the following entities and attributes:

CustomerID -> First\_Name, Last\_Name, GenderID, DOB, StatusID

PhoneID -> Phone\_Number

EmailID -> Email

Here I have created two extra tables customer phone numbers and customer email where a primary key, a foreign key referencing to customer table and the respect phone\_number or email is present. Hence now the tables are in 1NF form

* + **2NF(Second Normal Form)**

Since all non key-attributes are fully functionally dependant on the primary key and the relation exists in 1NF, second normal form has been satisfied.

* + **3NF(Third Normal Form)**

Since there is no transition dependency and the relation exists in 2NF, third normal form has been satisfied

* + **BCNF(Boyce Codd’s Normal Form)**

In the customer table all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Customer Status Table**:
   * Attributes:
     + StatusID (TINYINT, PK)
     + STATUS (VARCHAR(50), NOT NULL)
   * Functional Dependencies:
     + StatusID -> STATUS
   * NF:
     + The Customer Status table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
   * 2NF:
     + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
   * 3NF:
     + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
   * BCNF:
     + In the Customer Status table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.
2. **Customer Gender Table**:
   * Attributes:
     + GenderID (TINYINT, PK)
     + Gender (VARCHAR(50), NOT NULL)
   * Functional Dependencies:
     + GenderID -> Gender
   * NF:
     + The Customer Gender table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
   * 2NF:
     + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
   * 3NF:
     + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
   * BCNF:
     + In the Customer Gender table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.
3. **Customer Address Table**:
   * Attributes:
     + CustomerID (INT, FK)
     + AddressID (INT, PK)
     + Address (VARCHAR(255), NOT NULL)
     + AddressType (VARCHAR(50))
     + Country (VARCHAR(50))
     + State (VARCHAR(50))
     + City (VARCHAR(50))
     + Pincode (VARCHAR(6))
   * Functional Dependencies:
     + AddressID -> CustomerID,Address, AddressType, Country, State, City, Pincode
   * NF:
     + The Customer Address table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
   * 2NF:
     + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
   * 3NF:
     + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
   * BCNF:
     + In the Customer Address table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.
4. **KYC docs Table**:
   * Attributes:
     + CustomerID (INT, FK, PK)
     + DOC ID (INT, PK)
     + DOC\_NUMBER (INT, NOT NULL)
     + ExpiryDate (DATE)
     + TypeID (TINYINT, FK)
     + Status (TINYINT, FK)
   * Functional Dependencies:
     + DOC ID -> DOC\_NUMBER, CustomerID, ExpiryDate, TypeID, Status
   * 1NF(First Normal Form):
     + The KYC docs table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
   * 2NF(Second Normal Form):
     + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
   * 3NF(Third Normal Form):
     + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
   * BCNF(Boyce Codd’s Normal Form):
     + In the KYC docs table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.
5. **KYC Status Table**:
   * Attributes:
     + StatusID (TINYINT, PK)
     + STATUS (VARCHAR(50), NOT NULL)
   * Functional Dependencies:
     + StatusID -> STATUS
   * 1NF:
     + The KYC Status table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
   * 2NF:
     + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
   * 3NF:
     + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
   * BCNF:
     + In the KYC Status table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.
6. **KYC DOCType Table**:
   * Attributes:
     + TypeID (TINYINT, PK)
     + Name (VARCHAR(50), NOT NULL)
   * Functional Dependencies:
     + TypeID -> Name
   * 1NF:
     + The KYC DOCType table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
   * 2NF:
     + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
   * 3NF:
     + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
   * BCNF:
     + In the KYC DOCType table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.
7. **Loan Application Table**:

* Attributes:
  + ApplicationID (INT, PK)
  + CustomerID (INT, FK)
  + SchemeID (INT, FK)
  + ApplicationDate (DATE)
  + LoanAmount (DECIMAL(10,2))
  + Profession (VARCHAR(100))
  + Salary (DECIMAL(10,2))
  + Bank\_card\_number (VARCHAR(16))
  + Status (TINY INT, FK)
* Functional Dependencies:
  + ApplicationID -> CustomerID, SchemeID, ApplicationDate, LoanAmount, Profession, Salary, Bank\_card\_number, Status
* 1NF:
  + The Loan Application table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Loan Application table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **LoanApplication Status Table**:

* Attributes:
  + StatusID (TINYINT, PK)
  + STATUS (VARCHAR(50), NOT NULL)
* Functional Dependencies:
  + StatusID -> STATUS
* 1NF:
  + The Loan Application Status table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Loan Application Status table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Gold Ornament Table**:

* Attributes:
  + ApplicationID (INT, FK, PK)
  + OrnamID (INT, PK)
  + Ornam\_Type (VARCHAR(50))
  + Gross\_Weight (DECIMAL(10,2))
  + Net\_Weight (DECIMAL(10,2))
  + Purity (DECIMAL(5,2))
  + RatePerGram (DECIMAL(10,2))
  + Price (DECIMAL(10,2))
  + Remark (VARCHAR(255))
* Functional Dependencies:
  + ApplicationID, OrnamID -> Ornam\_Type, Gross\_Weight, Net\_Weight, Purity, RatePerGram, Price, Remark
* 1NF:
  + The Gold Ornament table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Gold Ornament table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Loan Scheme Table**:

* Attributes:
  + Scheme\_ID (INT, PK)
  + Scheme\_name (VARCHAR(100))
  + StartDate (DATE)
  + EndDate (DATE)
  + MaxAmount (DECIMAL(10,2))
  + MinAmount (DECIMAL(10,2))
  + InterestRate (DECIMAL(5,2))
  + RepaymentPeriod (INT)
* Functional Dependencies:
  + Scheme\_ID -> Scheme\_name, StartDate, EndDate, MaxAmount, MinAmount, InterestRate, RepaymentPeriod
* 1NF:
  + The Loan Scheme table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Loan Scheme table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Gold Loan Table**:

* Attributes:
  + Loan\_ID (INT, PK)
  + Application\_ID (INT, FK)
  + LoanAmount (DECIMAL(10,2))
  + LoanDate (DATE)
  + MaturityDate (DATE)
  + Scheme\_ID (INT, FK)
  + Status (TINYINT, FK)
* Functional Dependencies:
  + Loan\_ID -> Application\_ID, LoanAmount, LoanDate, MaturityDate, Scheme\_ID, Status
* 1NF:
  + The Gold Loan table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Gold Loan table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **GoldLoan Status Table**:

* Attributes:
  + StatusID (TINYINT, PK)
  + STATUS (VARCHAR(50), NOT NULL)
* Functional Dependencies:
  + StatusID -> STATUS
* 1NF:
  + The GoldLoan Status table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the GoldLoan Status table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Loan Schedule Table**:

* Attributes:
  + Loan\_ID (INT, FK, PK)
  + Numb (INT, PK)
  + Amount (DECIMAL(10,2))
  + OSAmount (DECIMAL(10,2))
  + PaidDate (DATE)
  + DueDate (DATE)
  + LateFineCharges (DECIMAL(10,2))
  + Status (TINYINT, FK)
* Functional Dependencies:
  + Loan\_ID, Numb -> Amount, OSAmount, PaidDate, DueDate, LateFineCharges, Status
* 1NF:
  + The Loan Schedule table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Loan Schedule table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **LoanSchedule Status Table**:

* Attributes:
  + StatusID (TINYINT, PK)
  + STATUS (VARCHAR(50), NOT NULL)
* Functional Dependencies:
  + StatusID -> STATUS
* 1NF:
  + The LoanSchedule Status table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the LoanSchedule Status table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Loan Disbursal Table**:

* Attributes:
  + Disbursal\_ID (INT, PK)
  + LoanID (INT, FK)
  + AmountPaid (DECIMAL(10,2))
  + PaymentDate (DATE)
  + PaymentTypeID (TINYINT, FK)
  + Transaction\_ID (INT)
  + Status (TINYINT, FK)
* Functional Dependencies:
  + Disbursal\_ID -> LoanID, AmountPaid, PaymentDate, PaymentTypeID, Transaction\_ID, Status
* 1NF:
  + The Loan Disbursal table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Loan Disbursal table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **LoanDisbursal Status Table**:

* Attributes:
  + StatusID (TINYINT, PK)
  + STATUS (VARCHAR(50), NOT NULL)
* Functional Dependencies:
  + StatusID -> STATUS
* 1NF:
  + The LoanDisbursal Status table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the LoanDisbursal Status table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Payment Type Table**:

* Attributes:
  + PTID (TINYINT, PK)
  + PaymentType (VARCHAR(50), NOT NULL)
* Functional Dependencies:
  + PTID -> PaymentType
* 1NF:
  + The Payment Type table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Payment Type table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

1. **Loan Repayment Table**:

* Attributes:
  + RepaymentID (INT, PK)
  + LoanID (INT, FK)
  + Number (INT, FK)
  + AmountRepaid (DECIMAL(10,2))
  + PaymentDate (DATE)
  + PaymentTypeID (TINYINT, FK)
  + Transaction\_ID (INT)
* Functional Dependencies:
  + RepaymentID -> LoanID, Number, AmountRepaid, PaymentDate, PaymentTypeID, Transaction\_ID
* 1NF:
  + The Loan Repayment table provided is in the first normal form since there are no multivalued attributes or each attribute contains atomic values without any repeating groups.
* 2NF:
  + Since all non-key attributes are fully functionally dependent on the primary key and the relation exists in 1NF, second normal form has been satisfied.
* 3NF:
  + Since there is no transitive dependency and the relation exists in 2NF, third normal form has been satisfied.
* BCNF:
  + In the Loan Repayment table, all the right-hand side of all the relations is always being determined by a super key or a candidate key. Hence we can conclude that this relation is in the BCNF form. Every determinant is a candidate key.

Tables:

1. Customer Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| CustomerID | INT | PK | Unique identifier for the customer |
| First\_Name | VARCHAR(50) | NOT NULL | First name of the customer |
| Last\_Name | VARCHAR(50) | NOT NULL | Last name of the customer |
| GenderID | TINYINT | FK | Gender of customer |
| DOB | DATE |  | Date of birth of the customer |
| StatusID | TINYINT | FK | Status of the customer |

1. Customer Status Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| StatusID | TINYINT | PK | Unique identifier for the status |
| STATUS | VARCHAR(50) | NOT NULL | Status of customer. Eg: Active,Inactive |

1. Customer Gender Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| GenderID | TINYINT | PK | Unique identifier for the gender |
| Gender | VARCHAR(50) | NOT NULL | Gender of customer. Eg: Male, Female, Inactive |

1. Customer Address Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| CustomerID | INT | FK | Reference to the customer |
| AddressID | iNT | PK | Unique identifier for address |
| Address | VARCHAR(255) | NOT NULL | Address of the customer |
| AddressType | VARCHAR(50) |  | Address type. Eg: Permanent,Temporary |
| Country | VARCHAR(50) |  | Country of the customer's address |
| State | VARCHAR(50) |  | State of the customer's address |
| City | VARCHAR(50) |  | City of the customer's address |
| Pincode | VARCHAR(6) |  | Pincode of the customer's address |

1. Phone Number Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| CustomerID | INT | FOREIGN KEY | Reference to the customer |
| PhoneID | INT | PK | Unique identifier for phone number |
| Phone\_Number | VARCHAR(10) | NOT NULL | Phone number of the customer |

1. Email Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| CustomerID | INT | FOREIGN KEY | Reference to the customer |
| EmailID | INT | PK | Unique identifier for email. |
| Email | VARCHAR(255) | NOT NULL | Email address of the customer |

1. KYC docs Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| CustomerID | INT | FK,PK | ID of the customer referenced in the customer table. |
| DOC ID | INT | PK | Unique identifier for the customer |
| DOC\_NUMBER | INT | NOT NULL | Document Number |
| ExpiryDate | DATE |  | Expiry Date of the document |
| TypeID | TINYINT | FK | Type of document. Eg: 1 - Aadhar,2 - PanCard |
| Status | TINYINT | FK | Status of the KYC verification.Eg: Verified, Not Verified. |
|  |  |  |  |

1. KYC Status Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| StatusID | TINYINT | PK | Unique identifier for the status |
| STATUS | VARCHAR(50) | NOT NULL | Status of KYC. Eg: Verfied,Not Verified |

1. KYC DOCType Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| TypeID | TINYINT | PK | Unique identifier for the status |
| Name | VARCHAR(50) | NOT NULL | Type of kyc docuemnt. Eg: Aadhar,pancard,etc |

1. Loan Application Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| ApplicationID | INT | PK | Unique identifier for the loan application |
| CustomerID | INT | FK | Reference to the customer |
| SchemeID | INT | FK | Reference to gold scheme table |
| ApplicationDate | DATE |  | Date of loan application |
| LoanAmount | DECIMAL(10,2) |  | Amount applied for loan |
| Profession | VARCHAR(100) |  | Profession of the applicant |
| Salary | DECIMAL(10,2) |  | Salary of the applicant |
| Bank\_card\_number | VARCHAR(16) |  | Bank card number of the applicant |
| Status | TINY INT | FK | Status of the loan application. Eg: Accpeted, Rejected, Pending. |
|  |  |  |  |
|  |  |  |  |

1. LoanApplication Status Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| StatusID | TINYINT | PK | Unique identifier for the status |
| STATUS | VARCHAR(50) | NOT NULL | Status of LoanApplication. Eg: Pending,Rejected,Accepted |

1. Gold OrnamentTable:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| ApplicationID | INT | FK,PK | Reference to loan application table |
| OrnamID | INT | PK | Unique identifier for gold ornament table |
| Ornam\_Type | VARCHAR(50) |  | Type of the gold ornament |
| Gross\_Weight | DECIMAL(10,2) |  | Gross weight of the gold ornament |
| Net\_Weight | DECIMAL(10,2) |  | Net weight of the gold ornament |
| Purity | DECIMAL(5,2) |  | Purity of the gold ornament |
| RatePerGram | DECIMAL(10,2) |  | Rate of per gram of gold in ornament |
| Price | DECIMAL(10,2) |  | Price of the gold ornament |
| Remark | VARCHAR(255) |  | Remark on the ornament. Eg: Ring is damaged. |

1. Loan Scheme Tabke

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| Scheme\_ID | INT | PK | Unique identifier for the gold scheme |
| Scheme\_name | VARCHAR(100) |  | Name of the gold loan scheme |
| StartDate | DATE |  | Start date of the scheme |
| EndDate | DATE |  | End date of the scheme |
| MaxAmount | DECIMAL(10,2) |  | Max amount that can be borrowed in the scheme |
| MinAmount | DECIMAL(10,2) |  | Min amount that can be borrowed in the scheme |
| InterestRate | DECIMAL(5,2) |  | Interest rate of the gold loan scheme |
| RepaymentPeriod | INT |  | Repayment period of the gold loan scheme(in months). Eg: 36 (3 years) |
|  |  |  |  |

1. Gold Loan Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| Loan\_ID | INT | PK | Unique identifier for the gold loan |
| Application\_ID | INT | FK | Reference to the Loan Application table |
| LoanAmount | DECIMAL(10,2) |  | Amount of the gold loan |
| LoanDate | DATE |  | Date on which the loan was granted |
| MaturityDate | DATE |  | MaturityDate of the gold loan |
| Scheme\_ID | INT | FK | Reference to the gold loan scheme |
| Status | TINYINT | FK | Status of the gold loan |
|  |  |  |  |

1. GoldLoan Status Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| StatusID | TINYINT | PK | Unique identifier for the status |
| STATUS | VARCHAR(50) | NOT NULL | Status of Gold Loan. Eg: Ongoing,Closed |

1. Loan Schedule Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| Loan\_ID | INT | FK,PK | Refence to gold loan table. |
| Numb | INT | PK | Unique Identifier for Loan Schedule Table |
| Amount | DECIMAL(10,2) |  | Amount to be paid for that particular month or to clear the entire loan. |
| OSAmount | DECIMAL(10,2) |  | Outstanding Amount to be paid |
| PaidDate | DATE |  | Date when payments was done. |
| DueDate | DATE |  | Date when payment is due. |
| LateFineCharges | DECIMAL(10,2) |  | Fine for when customer misses the due date of an emi payment. |
| Status | TINYINT | FK | Status of the Loan Schedule.Eg: 1-Unpaid,2-Partially Paid,3.Fully paid |
|  |  |  |  |

1. LoanSchedule Status Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| StatusID | TINYINT | PK | Unique identifier for the status |
| STATUS | VARCHAR(50) | NOT NULL | Status of Loan Schedule Eg: Unpaid,Partially Paid, Fully Paid |

1. Loan Disbursal table

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| Disbursal\_ID | INT | PK | Unique identifier for the loan disbursal |
| LoanID | INT | FK | References to the gold loan table. |
| AmountPaid | DECIMAL(10,2) |  | Amount paid to the customer as loan amount. |
| PaymentDate | DATE |  | Date of payment to customer. |
| PaymentTypeID | TINYINT | FK | Type of payment for the loan. Eg: Cheque, BankTransfer, Cash. |
| Transaction\_ID | INT |  | Transaction ID related to the disbursal |
| Status | TINY INT | FK | Status of the loan disbursal. Eg: 0-Unpaid, 1-Paid |

1. LoanDisbursal Status Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| StatusID | TINYINT | PK | Unique identifier for the status |
| STATUS | VARCHAR(50) | NOT NULL | Status of Loan Disbursal Eg: Unpaid,Paid |

1. Payment Type Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| PTID | TINYINT | PK | Unique identifier for the Payment Type |
| PaymentType | VARCHAR(50) | NOT NULL | Types of payment menthods. Eg: Cehque, Bank Transfer , Cash etc. |

1. Loan Repayment Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data Type | Constraint | Description |
| RepaymentID | INT | PK | Unique identifier for the Loan Repayment |
| LoanID | INT | FK | Reference to the Loan Schedule |
| Number | INT | FK | Refernce to the Loan Schedule Table |
| AmountRepaid | DECIMAL(10,2) |  | Amount Repaid to the gold loan |
| PaymentDate | DATE |  | Date of the payment |
| PaymentTypeID | TINYINT | FK | Type of payment for the lump sum payment. Eg: Cheque, BankTransfer, Cash. |
| Transaction\_ID | INT |  | Transaction ID related to the payment |