Sri Lanka Institute of Information Technology

Assignment 1

MLB_01.01_04

ONLINE BANKING

Information system & Data Modeling – IT1090

B.Sc. (Hons) in Information Technology

Title: Online Banking System	
Batch Number: Y1.S2.IT.01.01	Group Number: MLB_01.01_04

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

Student ID	Name	Signature
IT236211238	Gunasekara D.T	
IT23547124	Wickramarathne W.A.P.G	Pavani gayara.
IT23553446	Perera W.A.D.M	Disan
IT23608818	De Silva W.M.I.M	
IT23550858	Jayawardhena R.M.M.K	

Submitted on: 06/10/2024

Table of Contents

1. Introduction

Welcome to Helix Banking! We are an online banking platform that is designed to cater a variety of banking facilities in one web application that is created to provide for the evolving needs of our customers. Prioritizing on convenience and accessibility, our online banking site allows customers to manage their finances effortlessly from any location, at any given time. Our customers have access to a wide range of services in our virtual platform, including account management, fund transfers, bill payments, and loan applications, all within a secure and user-friendly virtual space.

Our platform makes sure that even customers with minimal technical expertise can benefit from the platform effortlessly. At Helix banking, our website is intuitively designed, with clear menus and straightforward instructions, enabling our customers to complete transactions quickly and efficiently.

Be it checking account balances, opening Fixed Deposits, or even applying for a loan scheme, Helix Banking provides streamlined processes minimize the time and effort needed, making your banking needs hassle-free.

Moreover, at Helix Bank we offer personalized customer support to our valuable users which is available around the clock to assist customers with any queries or issues that they may face. Our web application prioritizes convenience, with features like responsive design that ensures a smooth experience of the platform across different devices, including smartphones, tablets, and desktops. Our customers can handle many of their banking needs whenever and wherever they want thanks to our user-friendly virtual environment, which frees them from having to travel to physical facilities or adhere to set banking hours.

2. Hypothetical Scenario

Helix Banking is an online banking platform created to serve customers with a myriad of financial services eliminating the burden of physical travel, allowing our users to conduct their banking needs from anywhere, at any time.

As a visitor to the website, you will be introduced to various promotions from Helix Banking, including checking and savings account offer, customer reviews, as well as Loan and fixed deposit (FD) proposals. By completing the sign up process, visiting users can become valued customers and gain access to the full suite of services Helix Bank has to offer.

Upon logging in as a registered customer, Helix banking provides access to a wide range of services to meet your banking needs. . You can transfer funds, pay bills, and even open loan or fixed deposit (FD) accounts. Customers can update their personal information at any time, and 24/7 customer support is available to assist with any issues to ensure a smooth banking experience for the customer. All customer data is securely stored in our databases, using top-tier security practices to ensure your information is always protected.

All banking activities at Helix Banking are closely monitored our staff members to ensure a smooth daily operations. Any customer support needs, Loan approvals, FD account openings, approving fund transactions as well as account openings are managed by the banking staff.
The system Administration team are responsible for maintaining and securing the online platform. They also oversee large fund transactions and handle the deletion of information from the system when necessary.

3. Requirements Analysis

3.1 Main Requirements

3.1.1 Functional Requirements

- The system must allow new customers to create new accounts by providing personal information.
- Customers must be able to login to the system securely using their login credentials.
- Customers should be able to view details of their bank accounts.
- The system must allow users to transfer money to and from their own personal accounts.
- The system must allow users to transfer money from their own account to other accounts in the same system.
- Customers must be able to conduct schedules transfers in the system.
- The system should allow customers to deposit funds to their accounts.
- The system should allow customers to withdraw funds from their accounts.
- Users must be able to view loan schemes using the system.
- Users must be able to apply for loan schemes using the loan scheme application form in the system.
- Users must be able to view Fixed Deposit (FD) schemes using the system.
- Users must be able to apply for FD's using the scheme application form in the system.
- The system must be able to send notifications for any completed banking activity.
- Customers must be able to contact customer support through the system for any customer needs.
- The system must allow users to securely log out from their accounts at any time.

3.1.2 Non-Functional Requirements

Performance:

- The system should be able to handle a growing user base without any hindrance in system performance.
- The system should have a high response time for key banking activities such as fund transfers and balance checking inquiries.

Security:

- All customer data must be encrypted using standard encryption methods.
- Authentication and Authorization protocol should be used during any login processes.
- The system should ensure data privacy and prevent unauthorized personnel from attempting to access the system.

Reliability:

- The system must have reliable backup features and disaster recovery plans.
- The system should be designed to function even in the event of certain system errors.

Usability:

- The platform must be easy to navigate for all types of users.
- The system must be fully responsive when accessed on any device or on any user environment.

Efficiency:

- The website should be able to manage sever and database resources aiming to minimize maintenance costs.

Maintainability:

- The system should be able to uptake any system updates, changes, or additions without affecting the whole system.

Data Integrity:

- All transactions must ensure 100% data integrity during the transaction process.
- The system must be able to prevent any duplication of transactions to sustain the systems data integrity.

3.2 Data Requirements

Customer:

- Customer_ID
- First_name
- Last_name
- House_no
- Street
- City
- Code
- Phone
- Email

Account:

- Accout_no
- Balance
- Type
- Open_date
- Status

Transaction:

- Transaction_ID
- Description
- Account
- Amount
- Date

Bill Payment:

- Payment_ID
- Bill_type
- Amount
- Payment_date
- Status

Loan Application:

- Application_ID
- Type
- Status
- Request_amount
- Start_date
- Approve_date
- Installment_rates

FD Application:

- FD_ID
- Status
- Maturity_date
- Start_date
- Interest_rates

Customer Support:

- Request_ID
- Phone_number
- Emp_ID
- Service_type

Employee:

- Emp_ID
- First_name
- Last_name
- House_no
- Street
- City
- Code
- DOB
- Phone
- Email

Report:

- Report_ID
- Type
- Start_date
- Publish_date

4. Special Performance Requirements

- Fast customer support response availability:
  - When user seeks customer service, the system should enable quick responses without affecting the performance of the system.

- The system should send real time notifications without delays regardless of user location.

- The system should use predictive analytics to monitor performance in real time and notify adjustments to be made automatically.

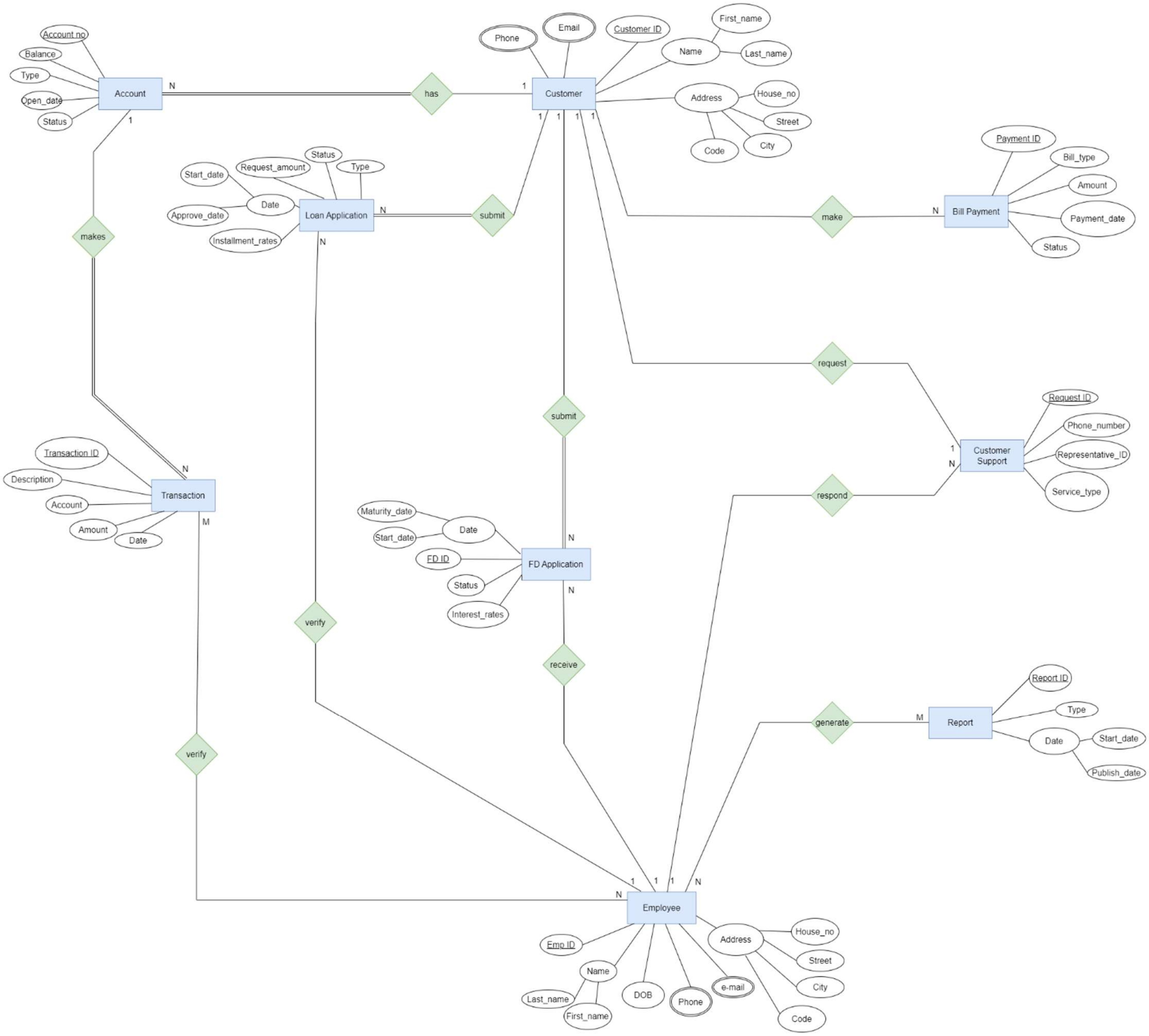
- The system must manage and support a high number of concurrent users performing different operations.

- The system must comply with relevant regulations and standards to ensure legal and operational integrity of the system.

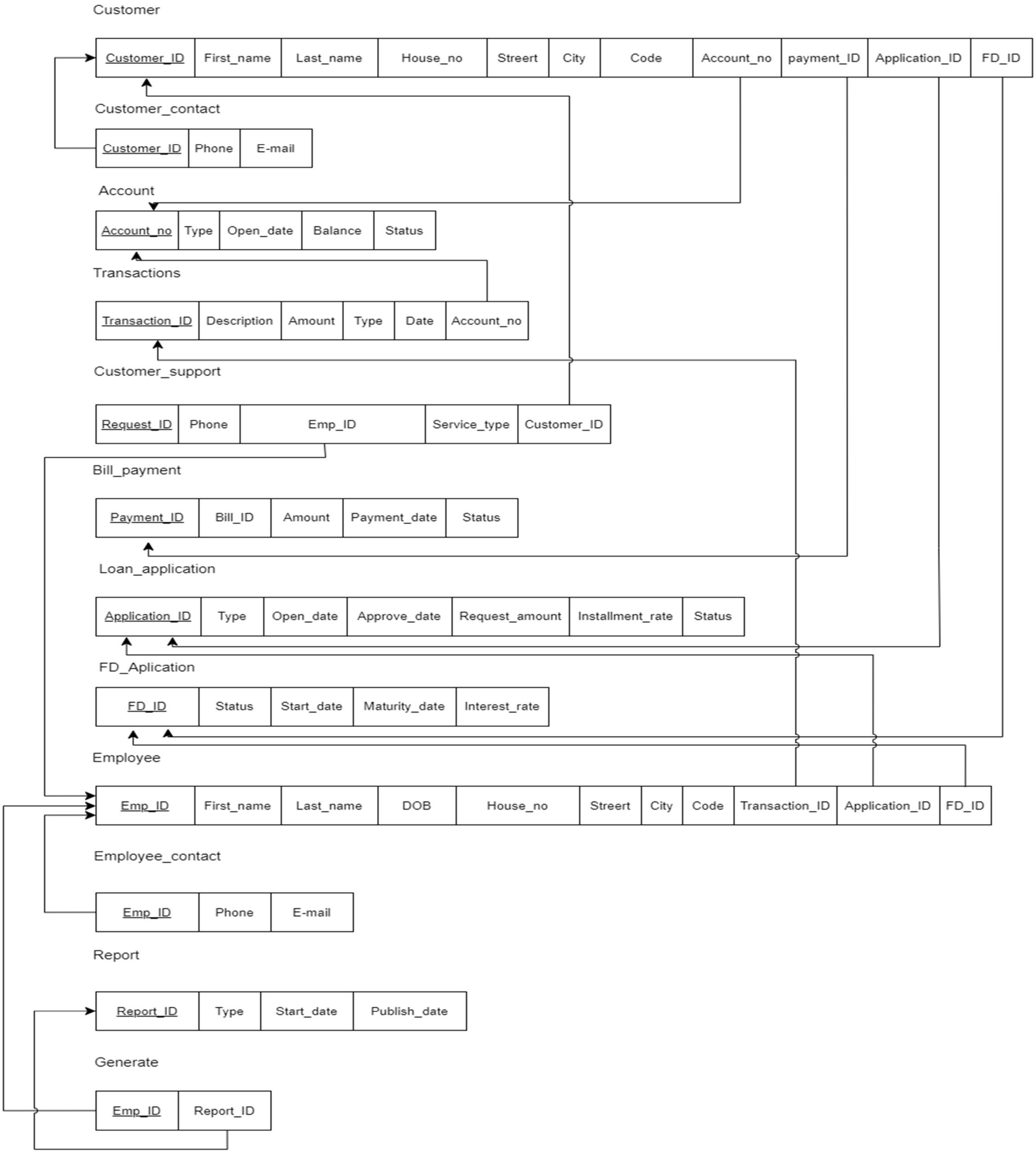
5. Security Requirements

- The system should implement automatic session timeouts for sensitive banking operations.
- The system should allow certain transactions to be processed if the user is within a certain geographical area.
- There should only be one user account per e-mail address in the system.
- The system should detect high operations if any are being performed and notify relevant personnel to ensure the operation is conducted securely.
- The user account password must be a strong password which consists of uppercase letters, lowercase letters, digits and special characters.
- All of the data in the system should be backed up in the database to ensure customer information protection and data integrity.

6. Entity Relationship (ER) Model



7. Relational Schema



Customer

Customer_ID	First_name	Last_name	House_no	Streert	City	Code	Account_no	payment_ID	Application_ID	FD_ID

Customer_contact

Customer_ID	Phone	E-mail

Account

Account_no	Type	Open_date	Balance	Status

Transactions

Transaction_ID	Description	Amount	Type	Date	Account_no

Customer_support

Request_ID	Phone	Emp_ID	Service_type	Customer_ID

Bill_payment

Payment_ID	Bill_ID	Amount	Payment_date	Status

Loan_application

Application_ID	Type	Open_date	Approve_date	Request_amount	Installment_rate	Status

FD_Aplication

FD_ID	Status	Start_date	Maturity_date	Interest_rate

Employee

Emp_ID	First_name	Last_name	DOB	House_no	Streert	City	Code	Transaction_ID	Application_ID	FD_ID

Employee_contact

Emp_ID	Phone	E-mail

Report

Report_ID	Type	Start_date	Publish_date

Generate

Emp_ID	Report_ID

8. SQL Queries

8.1 Creating Database Tables

```sql
create database HelixBankingDB

use HelixBankingDB

select * from Transactions

/*Table creation*/

--Customer Table
CREATE TABLE Customer (
    Customer_ID VARCHAR(20) NOT NULL,
    First_name VARCHAR(50) NOT NULL,
    Last_name VARCHAR(50) NOT NULL,
    House_no VARCHAR(20) NOT NULL,
    Street VARCHAR(100) NOT NULL,
    City VARCHAR(50) NOT NULL,
    Code VARCHAR(10) NOT NULL,
    Account_no INT NOT NULL,
    Payment_ID VARCHAR(20) NOT NULL,
    Application_ID VARCHAR(20) NOT NULL,
    FD_ID VARCHAR(20) NOT NULL,

       CONSTRAINT Customer_PK PRIMARY KEY(Customer_ID),

       CONSTRAINT AccountNo_FK FOREIGN KEY (Account_no) REFERENCES Account(Account_no),
       CONSTRAINT BillPayment_FK FOREIGN KEY (Payment_ID) REFERENCES 
Bill_payment(Payment_ID),
       CONSTRAINT LoanApplication_FK FOREIGN KEY (Application_ID) REFERENCES 
Loan_application(Application_ID),
       CONSTRAINT FDApplication_FK FOREIGN KEY (FD_ID) REFERENCES FD_Application(FD_ID)
);


--Customer contact Table
CREATE TABLE Customer_contact (---
    Customer_ID VARCHAR(20) NOT NULL,
    Phone VARCHAR(15) NOT NULL,
    Email VARCHAR(100) CHECK (Email LIKE '%_@%_.__%') NOT NULL,

        CONSTRAINT Customer_Contact_PK PRIMARY KEY(Customer_ID),

        CONSTRAINT Customer_Contact_FK FOREIGN KEY (Customer_ID) References Customer
(Customer_ID)
);
```

```
--Account Table
CREATE TABLE Account (---
    Account_no INT  NOT NULL,
    Type VARCHAR(50) NOT NULL,
    Open_date DATE NOT NULL,
    Balance DECIMAL(10, 2) NOT NULL,
    Status VARCHAR(20) NOT NULL,

       CONSTRAINT Account_PK PRIMARY KEY(Account_no),
);


--Transactions Table
CREATE TABLE Transactions (-----
    Transaction_ID VARCHAR(20) NOT NULL,
    Description VARCHAR(255) NOT NULL,
    Amount DECIMAL(10, 2) NOT NULL,
    Type VARCHAR(50) NOT NULL,
    Date DATE NOT NULL,
       Account_no INT  NOT NULL,

       CONSTRAINT Transactions_PK PRIMARY KEY(Transaction_ID),

       CONSTRAINT Account_FK FOREIGN KEY (Account_no) REFERENCES Account(Account_no)
);


--Employee table
CREATE TABLE Employee (
    Emp_ID VARCHAR(20) NOT NULL,
    First_name VARCHAR(50) NOT NULL,
    Last_name VARCHAR(50) NOT NULL,
    DOB DATE NOT NULL,
    House_no VARCHAR(20) NOT NULL,
    Street VARCHAR(100) NOT NULL,
    City VARCHAR(50) NOT NULL,
    Code VARCHAR(10) NOT NULL,
    Transaction_ID VARCHAR(20) NOT NULL,
    Application_ID VARCHAR(20) NOT NULL,
    FD_ID VARCHAR(20) NOT NULL,

       CONSTRAINT EmpID_PK PRIMARY KEY(Emp_ID),

       CONSTRAINT Transaction_FK FOREIGN KEY (Transaction_ID) REFERENCES
Transactions(Transaction_ID),
       CONSTRAINT EmpLoanApplication_FK FOREIGN KEY (Application_ID) REFERENCES
Loan_application(Application_ID),
       CONSTRAINT EmpFDApplication_FK FOREIGN KEY (FD_ID) REFERENCES
FD_Application(FD_ID)
);
```

```
--Employee contact Table
CREATE TABLE Employee_contact (
    Emp_ID VARCHAR(20) NOT NULL,
    Phone VARCHAR(15) NOT NULL,
    Email VARCHAR(100) CHECK (Email LIKE '%_@%_.__%') NOT NULL,

       CONSTRAINT EmployeeContact_PK PRIMARY KEY(Emp_ID),

    CONSTRAINT EmployeeContact_FK FOREIGN KEY (Emp_ID) REFERENCES Employee(Emp_ID)
);

--Customer Support Table
CREATE TABLE Customer_support (
    Request_ID VARCHAR(20)  NOT NULL,
    Phone VARCHAR(15) NOT NULL,
    Emp_ID VARCHAR(20) NOT NULL,
    Service_type VARCHAR(50) NOT NULL,
    Customer_ID VARCHAR(20) NOT NULL,

       CONSTRAINT CustomerSupport_PK PRIMARY KEY(Request_ID),

    CONSTRAINT CustomerSupport_FK1 FOREIGN KEY (Customer_ID) REFERENCES
Customer(Customer_ID),
    CONSTRAINT CustomerSupport_FK2 FOREIGN KEY (Emp_ID) REFERENCES Employee(Emp_ID)
);

--Bill payment Table
CREATE TABLE Bill_payment (----
    Payment_ID VARCHAR(20) NOT NULL,
    Bill_ID VARCHAR(20) NOT NULL,
    Amount DECIMAL(10, 2) NOT NULL,
    Payment_date DATE  NOT NULL,
    Status VARCHAR(20) NOT NULL,

       CONSTRAINT BillPayment_PK PRIMARY KEY(Payment_ID)
);

--loan Application Table
CREATE TABLE Loan_application (---
    Application_ID VARCHAR(20) NOT NULL,
    Type VARCHAR(50) NOT NULL,
    Open_date DATE NOT NULL,
    Approve_date DATE NOT NULL,
    Request_amount DECIMAL(10, 2) NOT NULL,
    Installment_rate DECIMAL(5, 2) NOT NULL,
    Status VARCHAR(20) NOT NULL

       CONSTRAINT LoanApplication_PK PRIMARY KEY(Application_ID)
);
```

```
--Report Table
CREATE TABLE Report (-----
    Report_ID VARCHAR(20) NOT NULL,
    Type VARCHAR(50) NOT NULL,
    Start_date DATE NOT NULL,
    Publish_date DATE  NOT NULL,

       CONSTRAINT Report_PK PRIMARY KEY(Report_ID),
);


--Generate Table
CREATE TABLE Generate (
    Emp_ID VARCHAR(20) NOT NULL,
    Report_ID VARCHAR(20)  NOT NULL,

    CONSTRAINT Generate_PK PRIMARY KEY (Emp_ID, Report_ID),

    CONSTRAINT Generate_FK1 FOREIGN KEY (Emp_ID) REFERENCES Employee(Emp_ID),
    CONSTRAINT Generate_FK2 FOREIGN KEY (Report_ID) REFERENCES Report(Report_ID)
);
```

8.2 Inserting Data

```
/*Adding table details*/

/*Add Account table details*/
 INSERT INTO Account(Account_no,Type,Open_date,Balance,Status)
 VALUES
 ('801234567','Savings Account','2001-1-5','50742.62','Active'),
 ('801654321','Current Account','1998-7-31','5690.00','Active'),
 ('801897680','Savings Account','2011-8-9','235000.30','Active'),
 ('801617592','Savings Account','2003-12-15','600000.50','Active'),
 ('801529600','Joint Account','2019-11-5','45000.00','Active');

 /*Add Bill Payment table details*/
 INSERT INTO Bill_payment(Payment_ID,Bill_ID,Amount,Payment_date,Status)
 VALUES
 ('P001','BID001','10000.00','2024-7-12','Active'),
 ('P002','BID002','22000.00','2024-8-24','Active'),
 ('P003','BID003','290000.00','2023-12-9','Active'),
 ('P004','BID004','15000.00','2022-7-31','Active'),
 ('P005','BID005','1000000.00','2024-4-2','Active');

/*Add Loan Application table details*/
INSERT INTO
Loan_application(Application_ID,Type,Open_date,Approve_date,Request_amount,Installment_rate,Status)
VALUES
('A0001','Personal','2024-1-5','2024-4-5','500000.00','6.50','Active'),
('A0002','Mortgage','2024-8-19','2024-2-12','2000000.00','7.42','Pending'),
('A0003','Personal','2023-11-4','2024-1-5','100000.00','6.50','Active'),
('A0004','Student','2023-10-28','2023-12-15','290000.00','4.52','Active'),
('A0005','Small Business','2024-6-5','2024-8-25','75000.00','7.50','Active');




/*Add FD Application table details*/
INSERT INTO FD_Application(FD_ID,Status,Start_date,Maturity_date,Interest_rate)
VALUES
('F001','Active','2024-3-12','2024-6-12','7.20'),
('F002','Active','2022-3-4','2027-8-1','9.00'),
('F003','Active','2024-8-31','2024-11-30','12.12'),
('F004','Active','2023-5-19','2024-5-19','8.18'),
('F005','Active','2024-7-29','2024-10-29','7.20');

/*Add Report table details*/
INSERT INTO Report(Report_ID,Type,Start_date,Publish_date)
VALUES
('Rep001','Admin','2023-5-6','2024-7-4'),
('Rep002','Balance','2024-5-5','2024-7-4'),
('Rep003','Profit and loss','2022-5-6','2023-7-4'),
('Rep004','Risk','2019-9-3','2019-12-4'),
('Rep005','Admin','2020-2-28','2023-4-4');
```

```sql
/*Add Transaction table details*/
INSERT INTO Transactions(Transaction_ID,Description,Amount,Type,Date,Account_no)
VALUES
('T001','Money is added to an account via ATM','5000.00','Deposit','2024-4-
17','801234567'),
('T002','Money is taken out of an account via electronic
transfer','20000.00','Withdrawal','2024-8-25','801654321'),
('T003','A return of funds to an account from a service','650000.00','Refund','2023-1-
27','801897680'),
('T004','Money is added to an account','90000.00','Deposit','2024-11-8','801617592'),
('T005','Money is taken out of an account via electronic
transfer','150000.00','Withdrawal','2024-10-30','801529600');

/*Add Customer table details*/
INSERT INTO
Customer(Customer_ID,First_name,Last_name,House_no,Street,City,Code,Account_no,Payment_ID
,Application_ID,FD_ID)
VALUES
('C001','Chamidu','Jayasingha','86','Methsiri
road','Malabe','1120','801234567','P001','A0001','F001'),
('C002','Madusanka','Fernando','34','Isurupura
road','Malabe','3456','801654321','P002','A0002','F002'),
('C003','Ahzan','Mohomad','567','Negombo
road','Divulapitiya','8800','801897680','P003','A0003','F003'),
('C004','Virajini','Weerasingha','92','Gamunu road','Colombo
5','2213','801617592','P004','A0004','F004'),
('C005','Piumi','Jayawardhena','06','Kirindivita
road','Aththanagalla','9090','801529600','P005','A0005','F005');

/*Add Customer contact table details*/
INSERT INTO Customer_contact(Customer_ID,Phone,Email)
VALUES
('C001','0779578432','chamidu34@.gmail.com'),
('C002','0719076440','fernando660@.gmail.com'),
('C003','0779995543','ahMo@.gmail.com'),
('C004','0768332111','virajiniWeeransin7@.gmail.com'),
('C005','0716861828','jayawardhenapiumi@.gmail.com');

/*Add Employee table details*/
INSERT INTO
Employee(Emp_ID,First_name,Last_name,DOB,House_no,Street,City,Code,Transaction_ID,Applica
tion_ID,FD_ID)
VALUES
('E001','Nissanka','Dharmasiri','1995-12-23','22','Methsiri
road','Kaduwela','1120','T001','A0001','F001'),
('E002','Menaka','Fernando','1990-1-15','456','Vijaya
road','Malabe','3456','T002','A0002','F002'),
('E003','Vithushan','Rajakaruna','1999-7-31','4','Kurunagala
road','Meerigama','8800','T003','A0003','F003'),
('E004','Dammika','Disanayake','1988-6-4','120','Wilson road','Colombo
10','2213','T004','A0004','F004'),
('E005','Nethmi','Sewmini','2001-9-20','98','Vijithapura
road','Panadura','9090','T005','A0005','F005');
```

```
/*Add Employee contact table details*/
INSERT INTO Employee_contact(Emp_ID,Phone,Email)
VALUES
('E001','0777555790','niss@.gmail.com'),
('E002','0719076441','menaka10@.gmail.com'),
('E003','0779999543','rajakaruna22@.gmail.com'),
('E004','0768322111','Disanayakedammika08@.gmail.com'),
('E005','0716801827','sewmi99@.gmail.com');

/*Add Customer support table details*/
INSERT INTO Customer_support(Request_ID,Phone,Emp_ID,Service_type,Customer_ID)
VALUES
('Req001','0779578432','E001','Transaction Support','C001'),
('Req002','0779578432','E002', 'Account Error','C002'),
('Req003','0779578432','E003', 'Account Cancellation','C003'),
('Req004','0779578432','E004', 'Account Error','C004'),
('Req005','0779578432','E005','Bill Payment Support','C005');

/*Add Generate table details*/
INSERT INTO Generate(Emp_ID,Report_ID)
VALUES
('E001','Rep001'),
('E002','Rep002'),
('E003','Rep003'),
('E004','Rep004'),
('E005','Rep005');
```