

Project Report : Banking System Database

1. PROJECT TITLE

Banking System Database

2. OBJECTIVE

The objective of this project is to design and implement a database system that efficiently manages banking operations such as customer details, account management, transactions, employee records, credit cards, loans, and branch details.

The system aims to provide structured storage and quick retrieval of data, ensuring data integrity and supporting analytical insights for better decision-making.

3. PROBLEM STATEMENT

Banks deal with a massive volume of customer data and financial transactions daily. Manual handling of these records leads to redundancy, inconsistency, and inefficiency. Therefore, a structured database system is required to:

- Maintain accurate customer and account records
 - Track transactions securely
 - Manage loans and credit cards
 - Monitor branch and employee performance
-

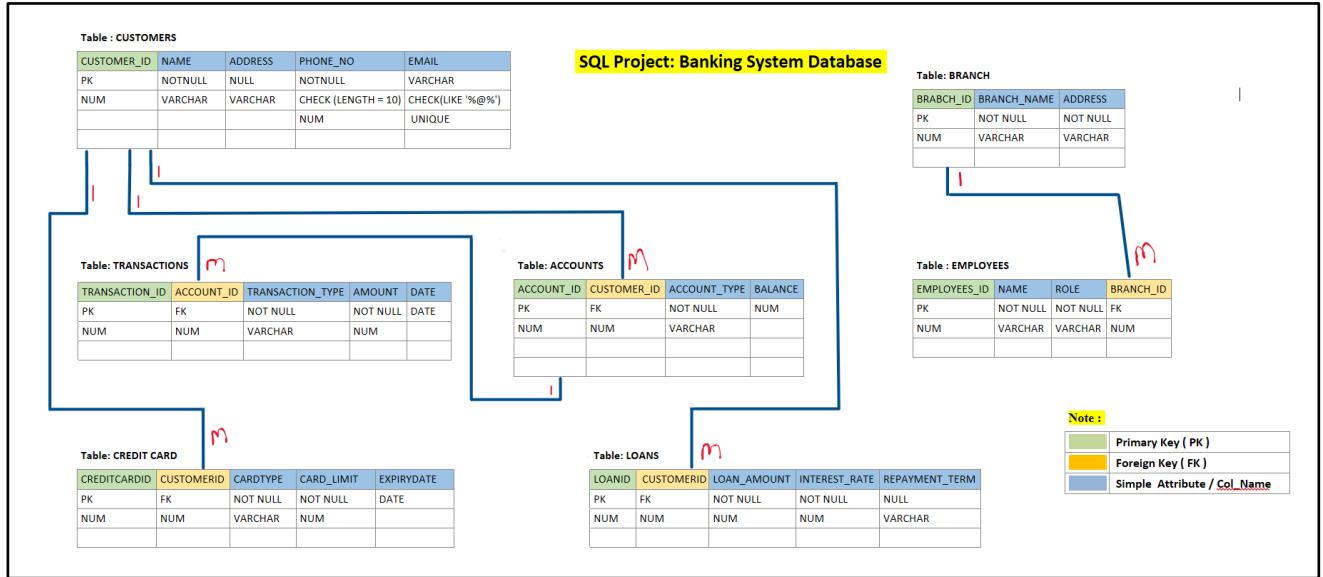
4. SYSTEM OVERVIEW

The Banking System Database consists of the following main entities:

- Customers
- Account
- Transaction
- Branch
- Employee
- CreditCard
- Loan

These entities are interconnected to form a complete relational database structure that supports real-time banking operations.

5. ER DIAGRAM



Relationship Explanation

- One CUSTOMERS can have many ACCOUNTS (1 to M)
- One ACCOUNT can have many TRANSACTIONS (1 to M)
- One CUSTOMERS can have many CREDITCARDS (1 to M)
- One CUSTOMERS can have many LOANS (1 to M)
- One BRANCH can have many EMPLOYEES (1 to M)

Table Details

1. CUSTOMERS Table

Purpose: Stores customer personal and contact information

Columns:

- **CUSTOMERID** – Primary Key, Number
 - **NAME** – Varchar (30), NOT NULL
 - **AGE** – Number, NULL
 - **PHONENO** – Number (10), CHECK(LENGTH(PHONENO)=10)
 - **EMAIL** – Varchar (100), UNIQUE, CHECK (EMAIL LIKE '%@%')
 - **ADDRESS** – Varchar (50), NULL
-

2. BRANCH Table

Purpose: Stores bank branch information

Columns:

- **BRANCHID** – Primary Key, Number
 - **BNAME** – Varchar (35), NOT NULL
 - **ADDRESS** – Varchar (50), NOT NULL
-

3. ACCOUNTS Table

Purpose: Stores customer account details

Columns:

- **ACCOUNT_ID** – Primary Key, Number
 - **CUSTOMERID** – Foreign Key → CUSTOMER(CUSTOMERID), Number
 - **ACCOUNT_TYPE** – Varchar (15), NOT NULL
 - **BALANCE** – Number, NOT NULL
-

4. TRANSACTIONS Table

Purpose: Records all banking transactions

Columns:

- **TRANSACTION_ID** – Primary Key, Number
- **ACCOUNT_ID** – Foreign Key → ACCOUNT(ACCOUNT_ID), Number
- **TRANSACTION_TYPE** – Varchar (20) (Debit / Credit)
- **AMOUNT** – Number, NOT NULL
- **TRANSACTION_DATE** - DATE

5. EMPLOYEES Table

Purpose: Stores bank employee information

Columns:

- **EMPID –** Primary Key, Number
 - **ENAME –** Varchar (30), NOT NULL
 - **ROLE –** Varchar (20), NULL
 - **BRANCHID –** Foreign Key → BRANCH(BRANCHID), Number
-

6. CREDITCARD Table

Purpose: Stores customer credit card details

Columns:

- **CREDITCARDID –** Primary Key, Number
 - **CUSTOMERID –** Foreign Key → CUSTOMER(CUSTOMERID), Number
 - **CARDTYPE –** Varchar (30), NOT NULL
 - **CARD_LIMIT –** Number, NOT NULL
 - **EXPIRYDATE –** Date
-

7. LOAN Table

Purpose: Stores customer loan information

Columns:

- **LOANID –** Primary Key, Number
- **CUSTOMERID –** Foreign Key → CUSTOMER(CID), Number
- **LOANAMOUNT –** Number, NOT NULL
- **INTERESTRATE –** Number, NOT NULL
- **REPAYMENTTERM –** Varchar (50), NULL

Step-by-step process with all modifications, in correct Oracle SQL, ready to run

1. CUSTOMERS Table

```
SQL>
SQL> CREATE TABLE CUSTOMERS
  2  ( CUSTOMERID NUMBER,
  3    NAME VARCHAR(30) NOT NULL,
  4    ADDRESS VARCHAR(50) NULL,
  5    PHONENO NUMBER(10) CHECK(LENGTH(PHONENO)=10),
  6    EMAIL VARCHAR(25) NULL,
  7    CONSTRAINTS CUSTOMERID_PK_CUST_PROJECT PRIMARY KEY(CUSTOMERID)
  8  );
Table created.

SQL> alter table customers
  2  drop column email;
Table altered.

SQL>

SQL> ALTER TABLE CUSTOMERS
  2  ADD EMAIL VARCHAR2(100) UNIQUE
  3  CHECK (EMAIL LIKE '%@%');
Table altered.
```

2. BRANCH Table

```
SQL>
SQL> CREATE TABLE BRANCH
  2  (
  3    BRANCHID NUMBER,
  4    BNAME VARCHAR(35) NOT NULL,
  5    ADDRESS VARCHAR(20) NOT NULL,
  6    CONSTRAINTS BRANCHID_PK_BRAN_PROJECT PRIMARY KEY(BRANCHID)
  7  );
Table created.
```

3. ACCOUNTS Table

```
SQL>
SQL> CREATE TABLE ACCOUNTS
  2  (
  3    ACCOUNT_ID NUMBER,
  4    CUSTOMERID NUMBER,
  5    ACCOUNT_TYPE VARCHAR(15) NOT NULL,
  6    BALANACE NUMBER,
  7    CONSTRAINTS ACCOUNT_ID_PK_ACCT_PROJECT PRIMARY KEY(ACCOUNT_ID),
  8    CONSTRAINTS CUSTOMERID_FK_ACCT_PROJECT FOREIGN KEY(CUSTOMERID) REFERENCES CUSTOMERS(CUSTOMERI
D)
  9  );
Table created.

SQL> alter table ACCOUNTS
  2  rename column BALANACE to BALANCE;
Table altered.

SQL>
```

4. TRANSACTIONS Table

```
SQL> CREATE TABLE TRANSACTIONS
  2  (
  3    TRANSACTION_ID NUMBER,
  4    ACCOUNT_ID NUMBER,
  5    TRANSACTION_TYPE VARCHAR2(20) NOT NULL,
  6    AMOUNT NUMBER NOT NULL,
  7    TRANSACTION_DATE DATE,
  8    CONSTRAINT TRANSACTION_ID_PK_TRAN_PROJECT PRIMARY KEY (TRANSACTION_ID),
  9    CONSTRAINT ACCOUNT_ID_FK_TRAN_PROJECT FOREIGN KEY (ACCOUNT_ID)
 10      REFERENCES ACCOUNTS(ACCOUNT_ID)
 11  );
Table created.

SQL> SELECT * FROM USER_CONSTRAINTS
  2  WHERE TABLE_NAME = 'TRANSACTIONS';

OWNER          CONSTRAINT_NAME          C TABLE_NAME          SEARCH_CONDITION
SCOTT          SYS_C005676              C TRANSACTIONS        "TRANSACTION_TYPE" IS NOT NULL
SCOTT          SYS_C005677              C TRANSACTIONS        "AMOUNT" IS NOT NULL
SCOTT          TRANSACTION_ID_PK_TRAN_PROJECT P TRANSACTIONS
SCOTT          ACCOUNT_ID_FK_TRAN_PROJECT   R TRANSACTIONS
```

5. EMPLOYEES Table

```
SQL> CREATE TABLE EMPLOYEES
  2  (
  3    EMPID NUMBER,
  4    ENAME VARCHAR(15) NOT NULL,
  5    ROLE VARCHAR(15) NULL,
  6    BRANCHID NUMBER,
  7    CONSTRAINTS EMPID_PK_EMPL_PROJECT PRIMARY KEY(EMPID),
  8    CONSTRAINTS BRANCHID_FK_EMPL_PROJECT FOREIGN KEY(BRANCHID)
  9      REFERENCES BRANCH(BRANCHID)
 10  );
Table created.

SQL> SELECT * FROM USER_CONSTRAINTS
  2 WHERE TABLE_NAME = 'EMPLOYEES';

OWNER          CONSTRAINT_NAME          C_TABLE_NAME          SEARCH_CONDITION
SCOTT          SYS_C005691              C_EMPLOYEES          "ENAME" IS NOT NULL
SCOTT          EMPID_PK_EMPL_PROJECT    P_EMPLOYEES          NULL
SCOTT          BRANCHID_FK_EMPL_PROJECT R_EMPLOYEES          NULL

SQL>
```

6. CREDITCARD Table

```
SQL>
SQL> CREATE TABLE CREDITCARD
  2  (
  3    CREDITCARDID NUMBER,
  4    CUSTOMERID NUMBER,
  5    CARDTYPE VARCHAR(30) NOT NULL,
  6    LIMIT NUMBER NOT NULL,
  7    EXPIRYDATE DATE,
  8    CONSTRAINTS CREDITCARDID_PK_CRED_PROJECT PRIMARY KEY(CREDITCARDID),
  9    CONSTRAINTS CUSTOMERID_FK_CRED_PROJECT FOREIGN KEY(CUSTOMERID)
 10      REFERENCES CUSTOMERS(CUSTOMERID)
 11  );
Table created.

SQL>
SQL> ALTER TABLE CREDITCARD
  2  RENAME COLUMN LIMIT TO CARD_LIMIT;

Table altered.
```

7. LOAN Table

```
SQL> CREATE TABLE LOAN
  2  (
  3    LOANID NUMBER,
  4    CUSTOMERID NUMBER,
  5    LOANAMOUNT NUMBER NOT NULL,
  6    INTERESTRATE NUMBER NOT NULL,
  7    REPAYMENTTERM VARCHAR(50) NULL,    CONSTRAINT LOANID_PK_LOAN_PROJECT PRIMARY KEY (LOANID),
  8    CONSTRAINT CUSTOMERID_FK_LOAN_PROJECT FOREIGN KEY (CUSTOMERID)
  9      REFERENCES CUSTOMERS(CUSTOMERID)
 10  );
Table created.

SQL> SELECT * FROM USER_CONSTRAINTS
  2 WHERE TABLE_NAME = 'LOAN';

OWNER          CONSTRAINT_NAME          C_TABLE_NAME          SEARCH_CONDITION
SCOTT          SYS_C005684              C_LOAN               "LOANAMOUNT" IS NOT NULL
SCOTT          SYS_C005685              C_LOAN               "INTERESTRATE" IS NOT NULL
SCOTT          LOANID_PK_LOAN_PROJECT    P_LOAN
SCOTT          CUSTOMERID_FK_LOAN_PROJECT R_LOAN

SQL> |
```

SAMPLE INSERT DATA

```
SQL> select * from CUSTOMERS;
+-----+-----+-----+-----+
| CUSTOMERID | NAME | ADDRESS | PHONENO | EMAIL |
+-----+-----+-----+-----+
| 101 | Kartik Shinde | Yavatmal | 9876543210 | kartik@gmail.com |
| 102 | Amit Patil | Nagpur | 9123456789 | amit@outlook.com |
| 103 | Neha Deshmukh | Pune | 9988776655 | neha@yahoo.com |
+-----+-----+-----+-----+
SQL> select * from ACCOUNTS;
+-----+-----+-----+
| ACCOUNT_ID | CUSTOMERID | ACCOUNT_TYPE | BALANCE |
+-----+-----+-----+
| 1001 | 101 | Savings | 50000 |
| 1002 | 102 | Current | 80000 |
| 1003 | 103 | Savings | 30000 |
+-----+-----+-----+
SQL> select * from TRANSACTIONS;
+-----+-----+-----+-----+
| TRANSACTION_ID | ACCOUNT_ID | TRANSACTION_TYPE | AMOUNT | TRANSACTI |
+-----+-----+-----+-----+
| 1 | 1001 | Debit | 2000 | 23-NOV-25 |
| 2 | 1001 | Credit | 5000 | 23-NOV-25 |
| 3 | 1002 | Debit | 10000 | 23-NOV-25 |
+-----+-----+-----+-----+
SQL> select * from CREDITCARD;
+-----+-----+-----+-----+
| CREDITCARDID | CUSTOMERID | CARDTYPE | CARD_LIMIT | EXPIRYDAT |
+-----+-----+-----+-----+
| 301 | 101 | VISA | 100000 | 31-DEC-27 |
| 302 | 102 | MasterCard | 150000 | 30-JUN-26 |
+-----+-----+-----+-----+
SQL> select * from LOAN;
+-----+-----+-----+-----+
| LOANID | CUSTOMERID | LOANAMOUNT | INTERESTRATE | REPAYMENTTERM |
+-----+-----+-----+-----+
| 401 | 102 | 300000 | 8.5 | 5 Years |
| 402 | 103 | 200000 | 7.8 | 3 Years |
+-----+-----+-----+-----+
SQL> select * from BRANCH;
+-----+-----+-----+
| BRANCHID | BNAME | ADDRESS |
+-----+-----+-----+
| 1 | SBI Yavatmal | Main Road |
| 2 | SBI Nagpur | Civil Line |
+-----+-----+-----+
SQL> select * from EMPLOYEES;
+-----+-----+-----+-----+
| EMPID | ENAME | ROLE | BRANCHID |
+-----+-----+-----+-----+
| 201 | Rahul | Manager | 1 |
| 202 | Sneha | Cashier | 2 |
+-----+-----+-----+-----+
SQL> commit ;
Commit complete.
SQL>
```

8. ADVANTAGES OF THE SYSTEM

- Reduces data redundancy
 - Ensures data accuracy
 - Improves transaction tracking
 - Enhances report generation
 - Better customer management
-

9. SAMPLE INSIGHTS SUPPORTED

- Most active customers
 - Highest balance accounts
 - Daily transaction volume
 - Branch-wise employee performance
 - Loan distribution trends
-

10. CONCLUSION

The Banking System Database provides a structured, efficient, and scalable solution for managing banking operations. By using normalization and relational constraints, the system ensures data integrity, easy maintenance, and powerful analytical capabilities. This database can be further integrated with application layers for real-time banking systems.