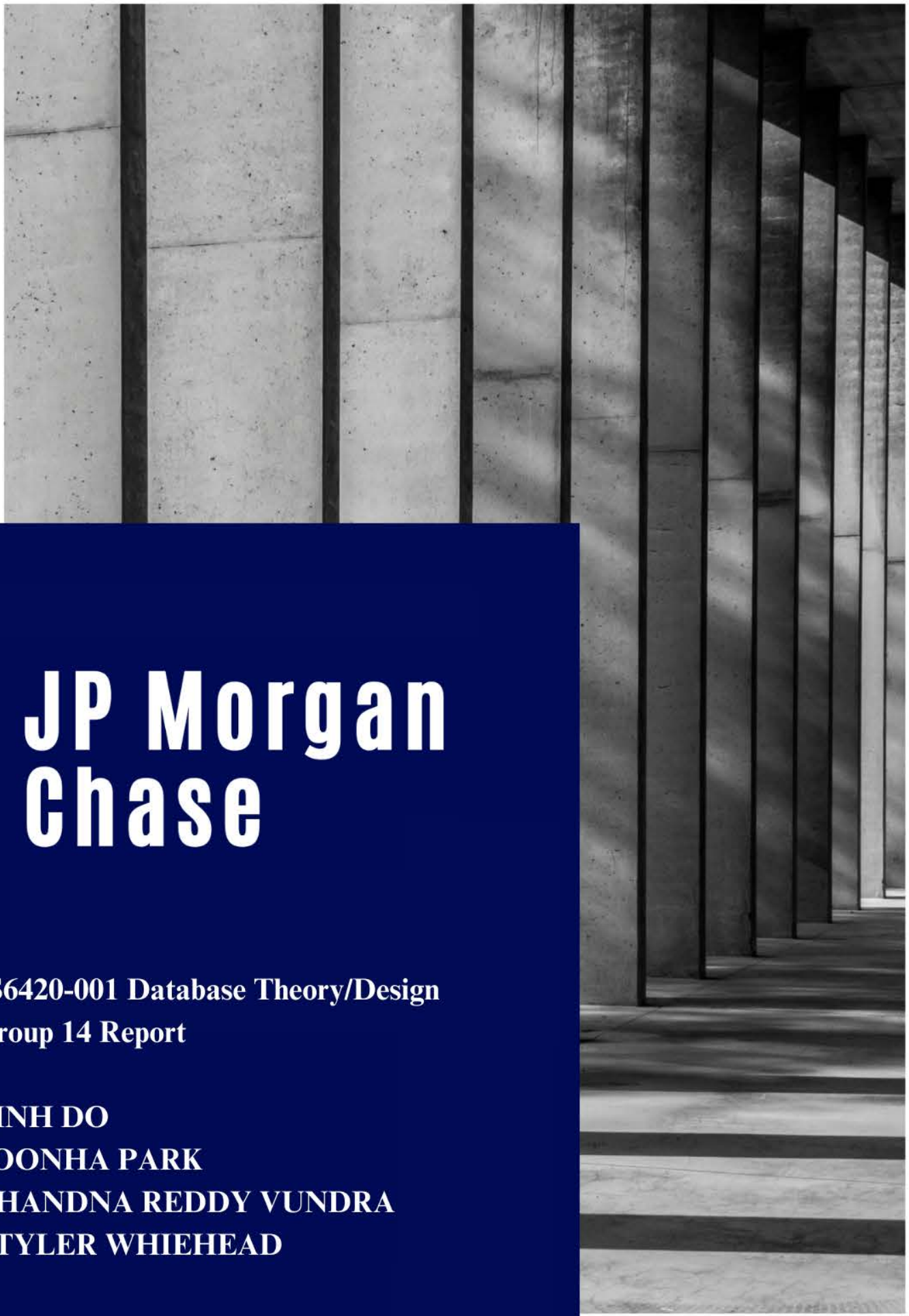


JP Morgan Chase

IS6420-001 Database Theory/Design
Group 14 Report

LINH DO
JOONHA PARK
CHANDNA REDDY VUNDRA
TTYLER WHIEHEAD

TABLE OF CONTENTS

EXECUTIVE SUMMARY

JP Morgan Chase, one of the largest financial institutions in the world, offers a comprehensive range of banking services to millions of customers globally. The organization is known for its innovative approach to financial services, including consumer and commercial banking, wealth management, and investment services. Chase's mobile and web applications provide customers with seamless access to their accounts, financial products, and transaction history, supporting their day-to-day banking needs. With a commitment to security, user convenience, and cutting-edge technology, Chase continuously enhances its digital platforms to deliver personalized and efficient services.

For this project, we focused on reverse-engineering the Chase banking application, which relies on a transactional database to support its operations. Our goal was to design a new database system that optimizes how the bank manages customer data, transactions, loans, and financial products. The new database includes key entities such as customer profiles, accounts, financial products, and transactions, ensuring real-time data access, security, and operational efficiency. This foundation supports core banking functions while enabling future scalability.

In addition to the system's current capabilities, we proposed an AI-powered Personal Finance Assistant (PFA) as a new business venture. This feature uses real-time data to provide customers with personalized financial advice and predictive insights, improving the customer experience and expanding Chase's service offerings. By integrating the PFA, the system enhances customer engagement, offering tailored recommendations based on individual financial behaviors.

The database design process focused on both technical robustness and ease of use, ensuring that the system can handle complex financial operations while delivering a user-friendly experience for employees and customers alike. With its emphasis on security, scalability, and customer personalization, this project aligns with JP Morgan Chase's mission to innovate and deliver exceptional service in the competitive banking landscape.

GENERAL DESCRIPTION

History

J.P. Morgan & Co. started out as the Bank of the Manhattan Company in New York City in 1799. Along with being a bank, it was established to provide clean water to the city. The Name J.P. Morgan & Co. started in 1871 and this was the beginning of the bank becoming a major player in commercial investment and private banking. The bank began merging with other banks. A few notable mergers were, merging with Chase National Bank in 1955. Chase Manhattan Bank then merged with JP Morgan &Co. Thus this is when they became known as JPMorgan Chase & Co(Chase Bank for short). Today Chase Bank serves millions of clients through commercial and consumer banking. They continue to be a major player in the financial industry that focuses on innovation, customer service, and Community Support.

Company Vision

Being one of the leading Financial service firms in the world, JP Morgan has a mission to serve corporations and individuals by delivering exceptional client service and supporting growth with their employees. Their purpose, “Make Dreams Possible for everyone, everywhere, every day.” best shows its support for the people around them. Three ways Chase plans to reach their goals are the following:

Racial Equity Commitment- Chase committed $30 billion to close the racial wealth gap. This also helped economic inclusion among underserved communities in the United States.

Community Investment- Chase focuses on creating a more inclusive and stronger economy through business community investments and policy advocacy. This includes supporting affordable housing, promoting financial health, being environmentally sustainable, and investing in local institutions and small businesses.

Innovative Solutions- driving sustainable, and inclusive growth for communities and the economy through leveraging business capabilities and philanthropic efforts.

Products/Services

Some of the products and Services offered by Chase Bank include the following:

Consumer & Community Banking- providing services to consumers and businesses, those services include deposit and investment products, cash management, payment solutions, mortgage origination servicing. THis also includes credit card issuance and auto loans.

Corporate & Investment Banking- Capital raising, Mergers and Acquisitions, and advisory Services. This also includes trading and market-making services.

Commercial Banking- Supporting businesses by lending, treasury services, and asset-based financing.

Asset & Wealth Management- retirement plans, wealth advisory to families, individuals and Institutions.

Transactional Database

Chase's Transactional database makes it so transactions with other companies are possible. Their database would include customer account details, deposits, withdrawals, payments, and other transactional data. Having this information is crucial when handling consumer checking and saving account data. Having this data encrypted and full of security restrictions allows for this information to be safely stored, and easily accessible for banking services and customer support.

BUSINESS REQUIREMENTS OF JP MORGAN CHASE

The project aims to modernize and streamline financial services for JP Morgan Chase customers by improving how they manage their personal finances, track transactions, and receive tailored product recommendations. One of the key innovations in this system is an AI-powered chatbot that offers personalized financial advice based on customer spending habits.

This system integrates core banking functions and AI-driven insights, focusing on several critical areas:

1. Customer Information Management

The system provides a centralized database for storing and managing customer information, ensuring data accuracy, privacy, and secure access. Customers are uniquely identified by a Customer ID, which allows efficient tracking of their account details, transactions, loans, and personal information.

2. Financial Product Management

The system supports comprehensive management of financial products, including credit cards, loans, savings accounts, and debit cards. Each product is linked to the customer and tracked for balances, interest rates, limits, and repayment statuses, ensuring accurate and up-to-date product information.

3. Transaction Tracking

Transaction management is a core function, enabling the accurate recording and categorization of customer transactions. Each transaction is stored with details such as amount, date, description, and category, facilitating seamless financial tracking and reporting.

4. Loan Management

The loan module allows customers to apply for, track, and manage their loans. It includes data such as loan amounts, interest rates, repayment schedules, and loan statuses. This ensures customers have clear insights into their borrowing activity, with the system providing reminders and status updates.

5. AI-Driven Financial Recommendations

A chatbot integrated into the system provides personalized financial product recommendations. Using AI algorithms, the chatbot analyzes customers' spending patterns, transaction history, and account status to suggest relevant products such as credit cards with better interest rates or personalized loan offers.

6. Chatbot and Customer Support

The system includes a chatbot that offers real-time support for customer queries, ranging from account status inquiries to transaction explanations. The chatbot uses historical transaction data to provide financial advice and support decisions, making customer interactions more efficient.

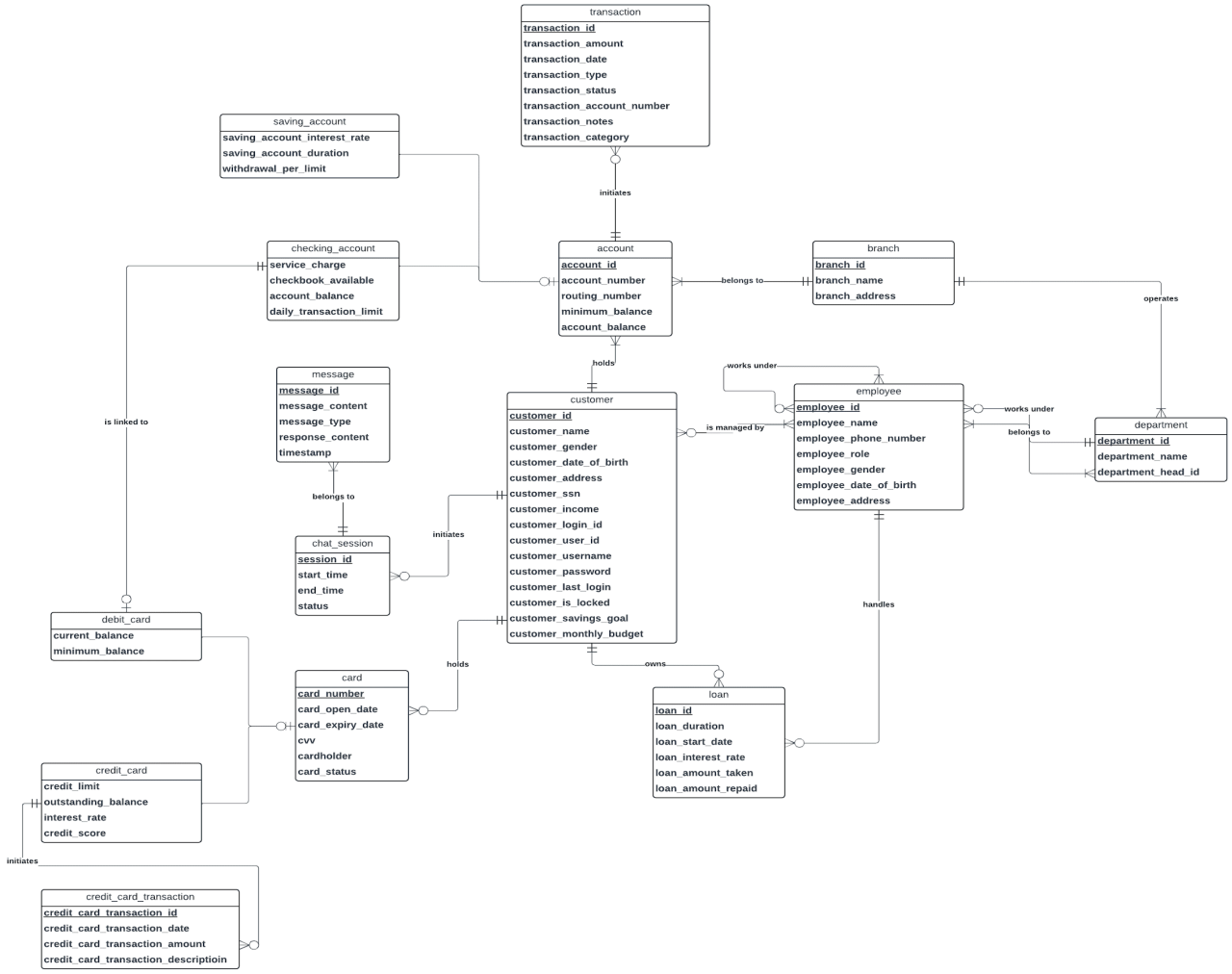
7. Branch and Employee Management

The system tracks branch locations and employee information, ensuring the proper management of resources across all branches. Employees are managed via a secure system that allows only authorized access to sensitive customer data, thus maintaining confidentiality and security.

8. Financial Security and User Access Control

To safeguard customer data, a robust role-based access control system is in place, limiting access to sensitive information based on user roles. This ensures that employees and customers can only view or modify data that is relevant to their roles, maintaining both privacy and integrity of the information.

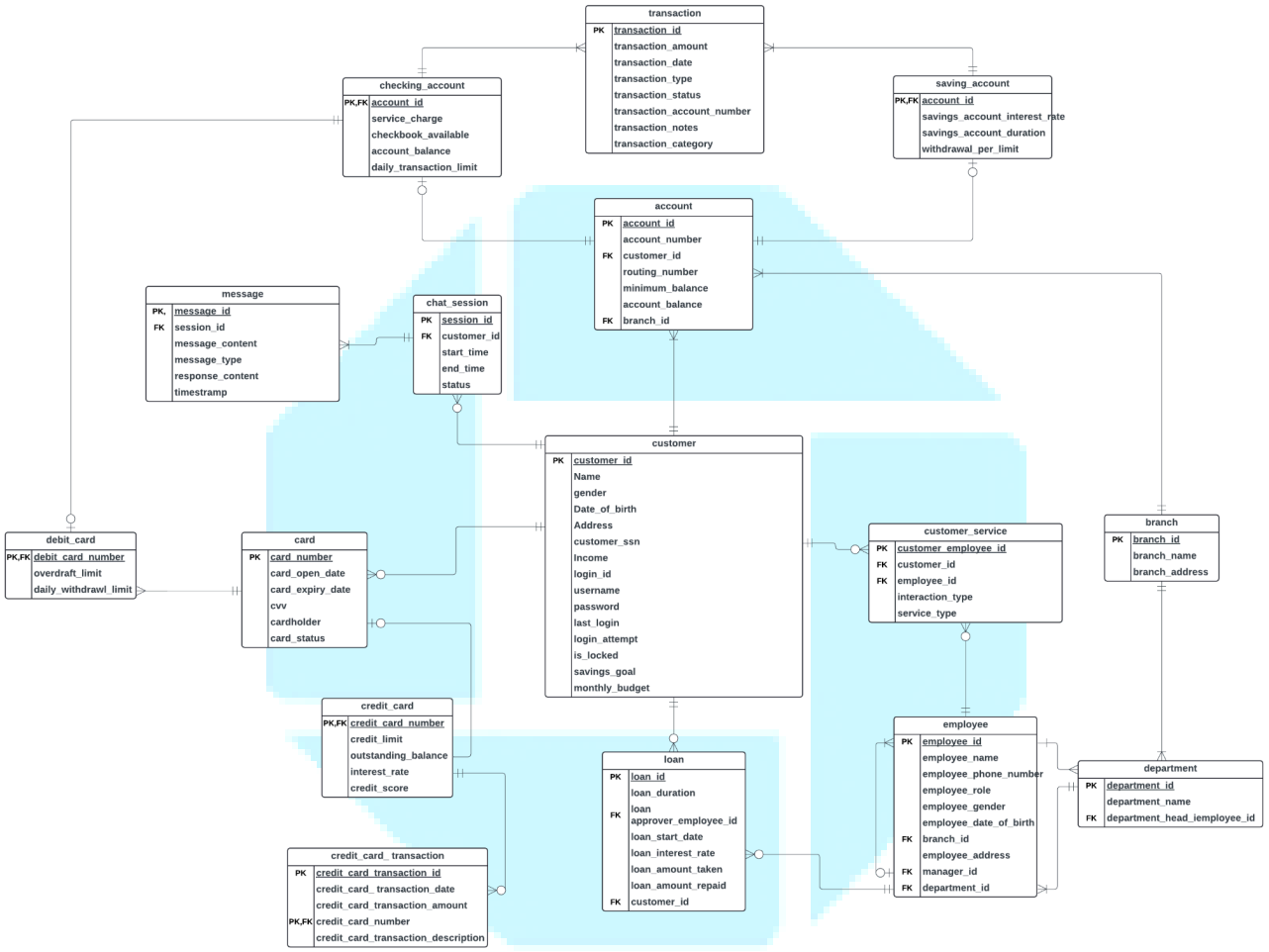
CONCEPTUAL MODEL



The JP Morgan Chase conceptual model centers around the customer entity, which connects to various types of accounts like checking, savings, credit cards, debit cards, and loans. These accounts are linked to specific transactions, such as deposits, withdrawals, and payments, recorded in the transaction entity. Customers also engage with the bank through branches, interacting with employees, while chat sessions and messages track customer service interactions.

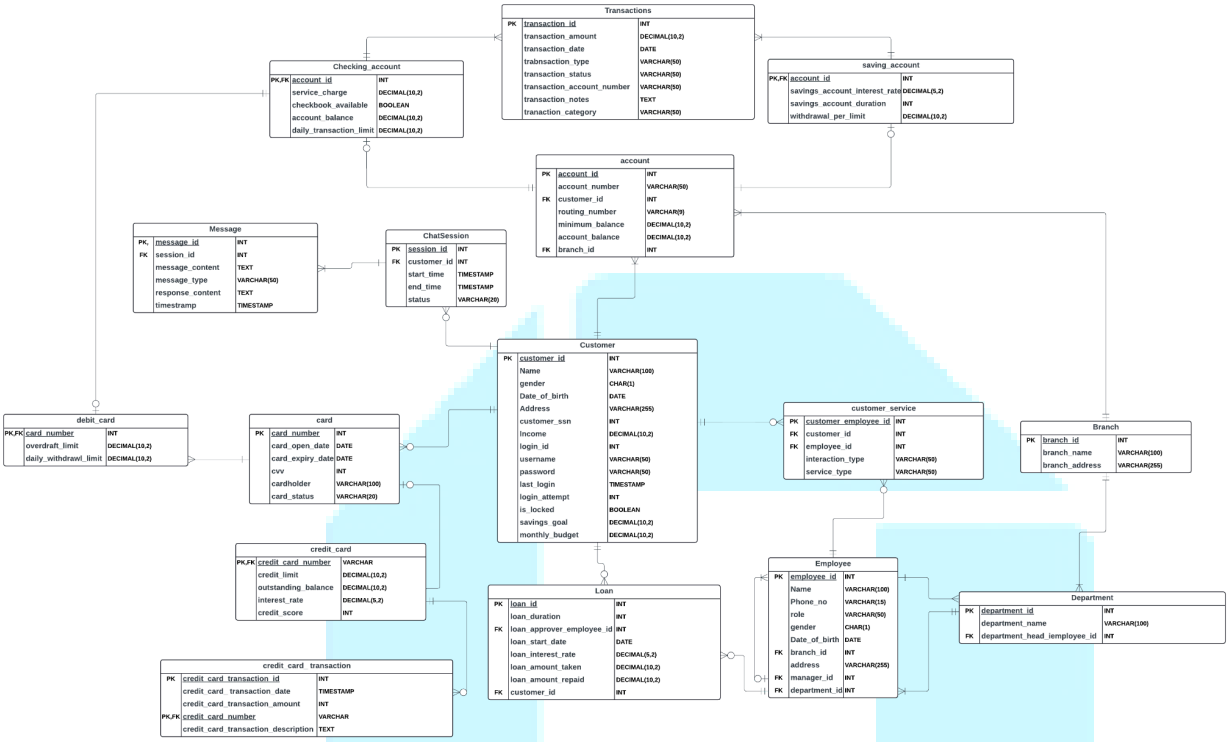
The account entity plays a key role in monitoring customers' financial status, with employees managing customer needs and handling loans. The model captures the hierarchical structure of departments and employees, emphasizing the customer’s relationship with banking products and the overall operations of the bank.

LOGICAL MODEL



The relational model refines the conceptual design by establishing primary keys (PKs), foreign keys (FKs), and clear relationships to maintain data integrity. It resolves many-to-many relationships with bridge tables and manages multi-valued attributes through specialized tables. The supertype-subtype relationships are handled by splitting them into separate entities like account, checking_account,savings_account, card,debit_card, and credit_card, capturing subtype-specific attributes. This structured approach ensures data consistency, efficient querying, and smooth database implementation and maintenance.

Physical Model



The Physical Model defines the data types and parameters in which the data is put into the database. This builds off of the Logical Model and adds a bit more information on how the layout. For attributes used a range of data types which included, INT, DECIMAL, DATE, and TEXT. These data types specify the kind of data that attribute holds and makes it easier to organize and manage it.

NEW VENTURE IDEAS

AI Personalized Financial Chatbot

We propose an innovative expansion of Chase Bank's personal AI assistant services, AI Personalized Financial Chatbot. This new venture aims to enhance Chase's AI Chatbot service by offering real-time, personalized financial insights, adaptive budgeting, and predictive analysis through AI technology. This strategy addresses critical limitations in Chase's existing services, such as static budgeting tools, lack of predictive analytics, and the absence of real-time financial recommendations.

The new chatbot is designed to give Chase customers a more personalized and proactive approach to managing their finances. The chatbot will deliver tailored budgeting, saving, investing, and loan management recommendations by analyzing a customer's spending habits, financial goals, and income patterns. The AI-driven insights will allow customers to make more informed financial decisions, optimize their spending, and achieve their financial objectives more effectively.

New Core Entities in DataBase

These entities are fundamental to managing customer interactions and the chatbot, ensuring all communications are appropriately tracked and structured for seamless user experiences.

1. **Chat Session Entity:** The **ChatSession** entity is designed to manage each interaction session between a user and the AI assistant.

- **session_id** serves as the unique identifier for each chat session, ensuring that multiple user interactions are grouped under a single session.
- **user_id** is a foreign key linking the chat session to a specific user, which allows the AI to provide personalized insights based on the user's financial data and history.
- The **start_time** and **end_time** fields log the exact duration of the chat session, which can help track the time spent on customer inquiries and provide insights into user engagement.
- **Status** tracks the session's current state (e.g., active, completed, escalated to human support).

2. **Message Entity:** The **Message** entity captures the individual messages exchanged during each chat session.

- **message_id** serves as the primary key for each message, ensuring that each piece of communication is unique and can be tracked.
- **session_id** links each message to the corresponding chat session, allowing for all messages to be grouped and ordered within the session.
- **message_content** holds the text or data of the message, whether it's a question from the user or a response from the AI assistant.

message_type distinguishes between messages the user sends (e.g., queries) and those the system generates (e.g., responses or recommendations).

- **response_content** stores the AI's actual recommendation or response based on the user's query.
- **Timestamp** tracks when each message was sent, providing a detailed communication log.

REQUIREMENTS REVIEW

1. Customer Information Management

- **Status**: Complete
- **Summary**: The system now efficiently handles the storage and management of customer data, assigning a unique ID to each customer. This ensures secure and organized access to personal details, account information, transactions, and loan history, maintaining a high standard of accuracy and privacy.

2. Financial Product Management

- **Status**: Complete
- **Summary**: A robust module is in place to manage a range of financial products, such as credit cards, loans, and savings accounts. All products are linked to customer profiles, allowing for real-time updates on balances, rates, limits, and repayments, thus ensuring the information is both current and actionable

3. Transaction Tracking

- **Status**: Complete
- **Summary**: Transaction processing and categorization are fully functional. Each transaction records essential details like the amount, date, and description. This provides a clear and organized view of customer spending, making it easier to track and manage their financial activities.

4. Loan Management

- **Status**: Complete
- **Summary**: The loan management system allows for the tracking and management of all aspects of customer loans, including amounts, interest rates, and repayment schedules. Alerts and reminders help customers stay on top of their payments and understand their borrowing activity.

5. AI-Driven Financial Recommendations

- **Status**: In Development
- **Summary**: An AI-powered chatbot is being integrated to provide personalized product suggestions based on customer transaction history and financial patterns. The chatbot will recommend relevant services, such as credit card options or loan products, tailored to individual needs.

6. Chatbot and Customer Support

- **Status**: In Progress
- **Summary**: The chatbot system, aimed at improving customer interaction, is being developed. It will handle queries related to accounts and transactions while offering real-time advice. With the help of AI, the chatbot will use past transaction data to provide more precise financial guidance.

7. Branch and Employee Management

- **Status**: Complete
- **Summary**: The system now provides efficient management of branch locations and employee information. It also ensures secure, role-based access to customer data, ensuring that only authorized personnel can access sensitive information, helping to safeguard privacy and optimize resource use.

8. Security and Access Control

- **Status**: Complete
- **Summary**: A secure, role-based access system has been established, limiting access to sensitive data based on user roles. This ensures compliance with data security protocols while protecting customer information and maintaining data integrity.

ETHICAL CONSIDERATIONS

Data Privacy and Security Policy at JP Morgan Chase

JP Morgan Chase's concern for customer financial information and privacy is critical to all its activities, especially in creating AI-based solutions such as the Personal Finance Assistant (PFA). JP Morgan Chase complies with industry standards for data protection and customer privacy, including the Gramm-Leach-Bliley Act (GLBA) and California Consumer Privacy Act (CCPA), to comply with the highest levels of data protection and customer confidentiality. These laws protect financial information and customers' rights using Chase's service.Important Compliance Information JP Morgan Chase considers these major compliance issues surrounding data privacy and security.

1. **CFI Customer Financial Information (CFI):** Typically, financial companies share proprietary information like account balances, transactions, savings objectives, etc. Special efforts are made to ensure this information's privacy, integrity, and availability according to data minimization and use requirements.

2. **Access Management**: JP Morgan Chase uses role-based access control (RBAC) to restrict system users to specific personnel. This means customer financial information can be viewed only by those with a business needing access to or control of it.

3. **Encryption:** Data encryption is used on all customer data at rest and in motion to prevent unauthorized access or capture. Security encryption is the best in the industry, and your information stays private during all transactions and storage.

4. **Audit Trails:** JP Morgan Chase tracks user actions in its systems in an audit trail. These records are accountable and traceable; if anyone gets access or you suspect a data breach, the bank can take action immediately. Logbooks are vital to investigations in the event of an incident.

5. **Third-Party Sharing of Data:** The organization engages with third parties (such as cloud hosts and software providers) in strict data-sharing arrangements. These agreements ensure that third parties will be bound by JP Morgan Chase's data privacy and financial rules, such as the GLBA and CCPA.

6. **Training & Awareness of employees:** JP Morgan Chase offers data privacy, security, and compliance training. This includes providing explicit procedures for dealing with, storing, and sharing customer money and educating all employees about current security measures.

7. **Customer Consent and Sharing of Data:** Customers trust that JP Morgan Chase will know how their data is collected, used, and shared. Law and regulations explicitly require exigent approval for communication with stakeholders or third-party providers.

8. **Monitoring of Continuous Compliance:** There is always a compliance check for rules, such as GLBA or CCPA. JP Morgan Chase actively keeps abreast of security risks and checks for vulnerabilities and prepares for them. Our systems are audited so that all systems and processes remain compliant with evolving requirements for financial data protection.

JP Morgan Chase's strict compliance with industry best practices and data integrity will help to protect customers' financial information. JP Morgan Chase protects confidential data while maintaining customer confidence and compliance via access control, encryption, audit trails, and ongoing compliance monitoring.

CONCLUSION & NEXT STEPS

In conclusion, the JP Morgan Chase database redesign has successfully modernized key components of the bank's digital infrastructure, including customer information management, financial product tracking, and transaction monitoring. These enhancements have improved the system’s efficiency, security, and ease of use, ensuring better customer experiences and streamlined operations. With these robust foundations in place, the system is ready to support future growth and handle more complex banking functions.

One of the most exciting future directions is the development of the AI-Powered Personal Finance Assistant (PFA). Once fully integrated, this tool will provide customers with personalized financial insights and real-time advice, helping them make more informed decisions about saving, budgeting, and investing. Future iterations could incorporate advanced predictive analytics and financial forecasting, allowing the PFA to offer even deeper insights tailored to each customer’s financial behavior.

Security and privacy will remain critical as the system expands. Enhancing encryption protocols, improving role-based access controls, and implementing real-time fraud detection will be essential to maintaining the integrity of customer data. As transaction volumes increase, ongoing efforts to optimize database performance and scalability will ensure that the system can handle growing demands while maintaining high efficiency and reliability.

Looking ahead, further opportunities lie in improving the user experience with more intuitive interfaces and dashboards, enabling customers to have a more seamless and personalized interaction with the system. Integrating with third-party services, such as fintech applications and investment tools, could also expand the platform’s ecosystem, providing a more comprehensive financial management experience. These future features will enable JP Morgan Chase to continue leading in innovative, secure, and customer-focused banking services.

References

Code of Conduct and Code of Ethics for Finance Professionals.JPMorganChase.

https://www.jpmorganchase.com/about/governance/code-of-conduct

Privacy and security.JPMorganChase.https://www.jpmorganchase.com/legal/privacy.htm

APPENDIX

Group Contribution Table

TEAM MEMBER	HOURS SPENT	DESCRIPTION OF WORK	ADDITIONAL COMMENTS
LINH DO	10	Conceptual Model, Business Requirements, Requirements Review, Presentation, Powerpoint Slides	
JOONHA PARK	10	New Venture Idea Summary, Ethical Consideration, Report, SQL StatementPresentation	
CHANDNA REDDY VUNDRA	10	Logical Model, Executive Summary, Conclusion & Next steps, Presentation, Powerpoint Slides	
Tyler Whitehead	10	Physical Model, General Description, Presentation, Model slides	

Detailed Business Requirements

Requirement	Entity	Status
Customer Information Management	Customer	Completed
Financial Product Management	Card Debit Card Credit Card Account Checking Account Saving Account	Completed

Transaction Tracking	Transaction Credit Card Transaction	Completed
Loan Management	Loan	Completed
AI-Driven Financial Recommendations	Message Chat Session Transaction Customer	Completed
Chatbot and Customer Support	Message Chat Session Transaction Customer	Completed
Branch and Employee Management	Branch Employee Department	Completed
Financial Security and User Access Control	Customer Transaction	Completed

SQL STATEMENT

```
-- Drop tables if they exist
DROP TABLE IF EXISTS Customer cascade;
DROP TABLE IF EXISTS Branch cascade;
DROP TABLE IF EXISTS Employee cascade;
DROP TABLE IF EXISTS Department cascade;
DROP TABLE IF EXISTS Credit_Card cascade;
DROP TABLE IF EXISTS Credit_Card_Transaction cascade;
DROP TABLE IF EXISTS Checking_Account cascade;
DROP TABLE IF EXISTS Card cascade;
DROP TABLE IF EXISTS Transactions cascade;
DROP TABLE IF EXISTS Debit_Card cascade;
DROP TABLE IF EXISTS Account cascade;
DROP TABLE IF EXISTS Saving_Account cascade;
DROP TABLE IF EXISTS Loan cascade;
DROP TABLE IF EXISTS ChatSession cascade;
DROP TABLE IF EXISTS Message cascade;
DROP TABLE IF EXISTS Customer_Service cascade;
-- Step 1: Create tables without foreign keys
-- Customer Table
CREATE TABLE Customer (
    customer_id INT PRIMARY KEY,
    name VARCHAR(100),
    gender CHAR(1),
    date_of_birth DATE,
    address VARCHAR(255),
    customer_ssn INT,
    income DECIMAL(10, 2),
    login_id INT,
    username VARCHAR(50),
    password VARCHAR(50),
    last_login TIMESTAMP,
    login_attempt INT,
    is_locked BOOLEAN,
    savings_goal DECIMAL(10, 2),
    monthly_budget DECIMAL(10, 2)
);
-- Branch Table
CREATE TABLE Branch (
    branch_id INT PRIMARY KEY,
    branch_name VARCHAR(100),
    branch_address VARCHAR(255)
);
-- Department Table
```

```sql
CREATE TABLE Department (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(100),
    department_head_employee_id INT
);
-- Employee Table
CREATE TABLE Employee (
    employee_id INT PRIMARY KEY,
    name VARCHAR(100),
    phone_no VARCHAR(15),
    role VARCHAR(50),
    gender CHAR(1),
    date_of_birth DATE,
    branch_id INT,
    address VARCHAR(255),
    manager_id INT,
    department_id INT
);
-- Credit Card Table
CREATE TABLE Credit_Card (
    credit_card_number VARCHAR PRIMARY KEY,
    credit_limit DECIMAL(10, 2),
    outstanding_balance DECIMAL(10, 2),
    interest_rate DECIMAL(5, 2),
    credit_score INT
);
-- Credit Card Transaction Table
CREATE TABLE Credit_Card_Transaction (
    credit_card_transaction_id INT PRIMARY KEY,
    credit_card_transaction_date TIMESTAMP,
    credit_card_transaction_amount INT,
    credit_card_number VARCHAR,
    credit_card_transaction_description TEXT
);
-- Checking Account Table
CREATE TABLE Checking_Account (
    account_id INT PRIMARY KEY,
    service_charge DECIMAL(10, 2),
    checkbook_available BOOLEAN,
    account_balance DECIMAL(10, 2),
    daily_transaction_limit DECIMAL(10, 2)
);
-- Card Table
CREATE TABLE Card (
    card_number INT PRIMARY KEY,
    card_open_date DATE,
    card_expiry_date DATE,
    cvv INT,
    cardholder VARCHAR(100),
    card_status VARCHAR(20)
);
```

```sql
-- Transactions Table
CREATE TABLE Transactions (
    transaction_id INT PRIMARY KEY,
    transaction_amount DECIMAL(10, 2),
    transaction_date DATE,
    transaction_type VARCHAR(50),
    transaction_status VARCHAR(50),
    transaction_account_number VARCHAR(50),
    transaction_notes TEXT,
    transaction_category VARCHAR(50)
);
-- Debit Card Table
CREATE TABLE Debit_Card (
    debit_card_number INT PRIMARY KEY,
    overdraft_limit DECIMAL(10, 2),
    daily_withdrawal_limit DECIMAL(10, 2)
);
-- Account Table
CREATE TABLE Account (
    account_id INT PRIMARY KEY,
    account_number VARCHAR(50),
    customer_id INT,
    routing_number VARCHAR(9),
    minimum_balance DECIMAL(10, 2),
    account_balance DECIMAL(10, 2),
    branch_id INT
);
-- Saving Account Table
CREATE TABLE Saving_Account (
    account_id INT PRIMARY KEY,
    savings_account_interest_rate DECIMAL(5, 2),
    savings_account_duration INT,
    withdrawal_per_limit DECIMAL(10, 2)
);
-- Loan Table
CREATE TABLE Loan (
    loan_id INT PRIMARY KEY
    loan_duration INT,
    loan_approver_employee_id INT
    loan_start_date DATE,
    loan_interest_rate DECIMAL(5, 2),
    loan_amount_taken DECIMAL(10, 2),
    loan_amount_repaid DECIMAL(10, 2),
    customer_id INT
);
-- Chat Session Table
CREATE TABLE ChatSession (
    session_id INT PRIMARY KEY,
    customer_id INT,
    start_time TIMESTAMP,
    end_time TIMESTAMP,
```

```
    status VARCHAR(20)
);
-- Message Table
CREATE TABLE Message (
    message_id INT PRIMARY KEY,
    session_id INT,
    message_content TEXT,
    message_type VARCHAR(50),
    response_content TEXT,
    timestamp TIMESTAMP
);
-- Customer Service Table
CREATE TABLE Customer_Service (
    customer_employee_id INT PRIMARY KEY,
    customer_id INT,
    employee_id INT,
    interaction_type VARCHAR(50),
    service_type VARCHAR(50)
);
-- Step 2: Add foreign keys after all tables are created
ALTER TABLE Employee
ADD FOREIGN KEY (branch_id) REFERENCES Branch(branch_id),
ADD FOREIGN KEY (manager_id) REFERENCES Employee(employee_id),
ADD FOREIGN KEY (department_id) REFERENCES Department(department_id);
ALTER TABLE Department
ADD FOREIGN KEY (department_head_employee_id) REFERENCES
Employee(employee_id);
ALTER TABLE Credit_Card_Transaction
ADD FOREIGN KEY (credit_card_number) REFERENCES
Credit_Card(credit_card_number);
ALTER TABLE Checking_Account
ADD FOREIGN KEY (account_id) REFERENCES Account(account_id);
ALTER TABLE Debit_Card
ADD FOREIGN KEY (debit_card_number) REFERENCES Card(card_number);
ALTER TABLE Account
ADD FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
ADD FOREIGN KEY (branch_id) REFERENCES Branch(branch_id);
ALTER TABLE Saving_Account
ADD FOREIGN KEY (account_id) REFERENCES Account(account_id);
ALTER TABLE Loan
add foreign key (loan_approver_employee_id) REFERENCES
Employee(employee_id),
ADD FOREIGN KEY (customer_id) REFERENCES Customer(customer_id);
ALTER TABLE ChatSession
ADD FOREIGN KEY (customer_id) REFERENCES Customer(customer_id);
ALTER TABLE Message
ADD FOREIGN KEY (session_id) REFERENCES ChatSession(session_id);
ALTER TABLE Customer_Service
ADD FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
ADD FOREIGN KEY (employee_id) REFERENCES Employee(employee_id);
-- Customer Table Insert
```

```
INSERT INTO Customer (customer_id, name, gender, date_of_birth, address,
customer_ssn, income, login_id, username, password, last_login,
login_attempt, is_locked, savings_goal, monthly_budget)
VALUES
(1, 'John Doe', 'M', '1985-01-01', '123 Elm Street', 123456789, 55000.00,
1, 'johndoe', 'password123', '2024-10-01 10:00:00', 1, false, 5000.00,
2000.00),
(2, 'Jane Smith', 'F', '1990-05-05', '456 Oak Street', 987654321,
62000.00, 2, 'janesmith', 'password456', '2024-10-01 11:00:00', 0, false,
6000.00, 2500.00),
(3, 'Alice Johnson', 'F', '1988-08-08', '789 Maple Street', 112233445,
48000.00, 3, 'alicej', 'password789', '2024-10-02 09:30:00', 2, true,
4000.00, 1800.00),
(4, 'Bob Brown', 'M', '1975-02-20', '234 Birch Street', 998877665,
75000.00, 4, 'bobbrown', 'password111', '2024-10-02 12:45:00', 0, false,
10000.00, 3000.00),
(5, 'Charlie Davis', 'M', '1995-03-15', '567 Cedar Street', 223344556,
82000.00, 5, 'charlied', 'password222', '2024-10-03 08:00:00', 1, false,
12000.00, 3500.00),
(6, 'Daisy Evans', 'F', '1992-07-25', '890 Pine Street', 554433221,
68000.00, 6, 'daisye', 'password333', '2024-10-04 10:30:00', 1, false,
7000.00, 2700.00),
(7, 'Eve Foster', 'F', '1981-11-11', '123 Spruce Street', 665544332,
55000.00, 7, 'evef', 'password444', '2024-10-04 11:45:00', 3, true,
4500.00, 2000.00),
(8, 'Frank Green', 'M', '1993-04-05', '456 Willow Street', 887766554,
50000.00, 8, 'frankg', 'password555', '2024-10-05 09:15:00', 0, false,
5500.00, 2200.00),
(9, 'Grace Hall', 'F', '1994-06-12', '789 Redwood Street', 998877776,
60000.00, 9, 'graceh', 'password666', '2024-10-05 14:00:00', 1, false,
6500.00, 2400.00),
(10, 'Henry Ivy', 'M', '1986-09-09', '234 Chestnut Street', 332211554,
70000.00, 10, 'henryi', 'password777', '2024-10-06 16:30:00', 2, true,
8000.00, 2800.00),
(11, 'Isabella Jones', 'F', '1989-10-10', '567 Walnut Street', 776655443,
72000.00, 11, 'isabellaj', 'password888', '2024-10-07 17:15:00', 1, false,
8500.00, 3000.00),
(12, 'Jack King', 'M', '1991-12-15', '890 Poplar Street', 554433776,
58000.00, 12, 'jackk', 'password999', '2024-10-08 08:45:00', 0, false,
6000.00, 2300.00),
(13, 'Karen Lee', 'F', '1996-03-18', '123 Dogwood Street', 221133445,
63000.00, 13, 'karenl', 'password101', '2024-10-09 10:30:00', 1, false,
7000.00, 2500.00),
(14, 'Leo Martin', 'M', '1982-07-20', '456 Fir Street', 998844556,
78000.00, 14, 'leom', 'password202', '2024-10-09 11:00:00', 2, true,
9000.00, 3200.00),
(15, 'Mia Nelson', 'F', '1987-08-30', '789 Alder Street', 665533221,
54000.00, 15, 'mian', 'password303', '2024-10-09 13:00:00', 3, false,
4500.00, 2200.00);
-- Branch Table Insert
INSERT INTO Branch (branch_id, branch_name, branch_address)
```

```
VALUES
(1, 'Downtown', '123 Main St'),
(2, 'Uptown', '456 Broadway Ave'),
(3, 'East Side', '789 East Rd'),
(4, 'West Side', '234 West Blvd'),
(5, 'North Side', '567 North St'),
(6, 'South Side', '890 South Rd'),
(7, 'Central', '123 Center St'),
(8, 'Suburban', '456 Suburb Dr'),
(9, 'Riverside', '789 River Ln'),
(10, 'Hilltop', '234 Hilltop Ave'),
(11, 'Valley', '567 Valley Rd'),
(12, 'Lakeside', '890 Lake Rd'),
(13, 'Woodland', '123 Woodland Dr'),
(14, 'Mountain', '456 Mountain Rd'),
(15, 'Seaside', '789 Seaside Dr');
-- Employee Table Insert
INSERT INTO Employee (employee_id, name, phone_no, role, gender,
date_of_birth, branch_id, address, manager_id, department_id)
VALUES
(1, 'Anna Smith', '123-456-7890', 'Manager', 'F', '1980-03-15', 1, '123
Main St', NULL, 1),
(2, 'Ben Williams', '321-654-0987', 'Teller', 'M', '1985-05-25', 1, '456
Oak St', 1, 1),
(3, 'Cathy Johnson', '555-666-7777', 'Loan Officer', 'F', '1990-08-11', 2,
'789 Pine St', 1, 2),
(4, 'David Lee', '444-333-2222', 'Clerk', 'M', '1987-12-22', 3, '234 Cedar
St', 1, 2),
(5, 'Emily Brown', '999-888-7777', 'Manager', 'F', '1981-07-01', 4, '567
Spruce St', NULL, 1),
(6, 'Frank Green', '111-222-3333', 'Security', 'M', '1975-11-15', 5, '890
Birch St', 5, 1),
(7, 'Grace Hall', '444-555-6666', 'HR', 'F', '1992-10-08', 6, '234 Willow
St', 5, 1),
(8, 'Henry Ivy', '777-888-9999', 'Technician', 'M', '1983-09-02', 7, '567
Redwood St', 7, 2),
(9, 'Ivy Nelson', '222-333-4444', 'Cashier', 'F', '1994-04-06', 8, '890
Walnut St', 8, 3),
(10, 'Jack King', '123-321-4567', 'Supervisor', 'M', '1986-02-11', 9, '234
Poplar St', 9, 3),
(11, 'Karen Davis', '654-987-3210', 'Analyst', 'F', '1991-03-30', 10, '567
Dogwood St', 10, 4),
(12, 'Leo Martin', '555-666-1111', 'Loan Officer', 'M', '1989-05-21', 11,
'890 Fir St', 11, 5),
(13, 'Mia Nelson', '222-111-4444', 'Accountant', 'F', '1992-12-17', 12,
'123 Alder St', 12, 6),
(14, 'Noah Parker', '333-555-7777', 'Marketing', 'M', '1987-09-27', 13,
'456 Seaside Dr', 13, 7),
(15, 'Olivia Scott', '999-000-1234', 'Manager', 'F', '1982-11-07', 14,
'789 Hilltop Rd', NULL, 8);
-- Department Table Insert
```

```
INSERT INTO Department (department_id, department_name, department_head_employee_id)
VALUES
(1, 'HR', NULL),
(2, 'Loan Services', NULL),
(3, 'Customer Service', NULL),
(4, 'Security', NULL),
(5, 'Technical', NULL),
(6, 'Marketing', NULL),
(7, 'Accounting', NULL),
(8, 'Management', NULL);
-- Credit Card Table Insert
INSERT INTO Credit_Card (credit_card_number, credit_limit, outstanding_balance, interest_rate, credit_score)
VALUES
('1234567812345678', 10000.00, 2000.00, 3.5, 700),
('2234567812345678', 15000.00, 5000.00, 4.0, 750),
('3234567812345680', 12000.00, 3000.00, 4.5, 680),
('4234567812345681', 18000.00, 6000.00, 3.8, 720),
('5234567812345682', 20000.00, 7000.00, 5.0, 690),
('6234567812345683', 25000.00, 8000.00, 2.9, 710),
('7234567812345684', 9000.00, 1000.00, 3.6, 760),
('8234567812345685', 8000.00, 2000.00, 3.7, 740),
('9234567812345686', 14000.00, 3000.00, 3.4, 710),
('1034567812345687', 16000.00, 4000.00, 2.8, 750),
('1134567812345688', 11000.00, 1200.00, 3.0, 780),
('1234567812345689', 22000.00, 7000.00, 4.6, 690),
('1334567812345690', 10000.00, 2000.00, 3.3, 720),
('1434567812345691', 15000.00, 5000.00, 3.5, 730),
('1534567812345692', 13000.00, 500.00, 3.0, 740);


-- Credit Card Transaction Table Insert
INSERT INTO Credit_Card_Transaction (credit_card_transaction_id, credit_card_transaction_date, credit_card_transaction_amount, credit_card_number, credit_card_transaction_description)
VALUES
(1, '2024-10-01 10:00:00', 100.00, '1234567812345678', 'Groceries'),
(2, '2024-10-01 11:30:00', 50.00, '2234567812345678', 'Gas'),
(3, '2024-10-02 14:00:00', 200.00, '3234567812345680', 'Electronics'),
(4, '2024-10-02 17:15:00', 150.00, '4234567812345681', 'Clothing'),
(5, '2024-10-03 09:00:00', 300.00, '5234567812345682', 'Furniture'),
(6, '2024-10-03 13:45:00', 75.00, '6234567812345683', 'Restaurant'),
(7, '2024-10-04 12:30:00', 125.00, '7234567812345684', 'Movies'),
(8, '2024-10-04 15:00:00', 250.00, '8234567812345685', 'Books'),
(9, '2024-10-05 10:15:00', 80.00, '9234567812345686', 'Taxi'),
(10, '2024-10-05 14:00:00', 60.00, '1034567812345687', 'Gift'),
(11, '2024-10-06 09:45:00', 220.00, '1134567812345688', 'Travel'),
(12, '2024-10-06 12:30:00', 40.00, '1234567812345689', 'Coffee'),
(13, '2024-10-07 08:00:00', 350.00, '1334567812345690', 'Gym Equipment'),
(14, '2024-10-07 10:15:00', 90.00, '1434567812345691', 'Shoes'),
```

```
(15, '2024-10-08 09:30:00', 110.00, '1534567812345692', 'Hotel');


-- Checking Account Table Insert
INSERT INTO Checking_Account (account_id, service_charge,
checkbook_available, account_balance, daily_transaction_limit)
VALUES
(1, 15.00, true, 1000.00, 5000.00),
(2, 10.00, false, 2000.00, 6000.00),
(3, 12.50, true, 3000.00, 7000.00),
(4, 18.00, true, 5000.00, 8000.00),
(5, 20.00, false, 7000.00, 9000.00),
(6, 15.50, true, 8000.00, 10000.00),
(7, 25.00, true, 6000.00, 7500.00),
(8, 10.00, false, 9000.00, 6500.00),
(9, 5.00, true, 12000.00, 8500.00),
(10, 15.00, true, 15000.00, 10000.00),
(11, 12.00, true, 5000.00, 12000.00),
(12, 17.50, false, 8000.00, 11000.00),
(13, 20.00, true, 14000.00, 9500.00),
(14, 25.00, false, 20000.00, 15000.00),
(15, 22.00, true, 25000.00, 18000.00);
-- Card Table Insert
INSERT INTO Card (card_number, card_open_date, card_expiry_date, cvv,
cardholder, card_status)
VALUES
(101, '2021-01-01', '2024-12-31', 123, 'John Doe', 'active'),
(102, '2021-05-01', '2024-12-31', 456, 'Jane Smith', 'active'),
(103, '2022-01-01', '2025-12-31', 789, 'Alice Johnson', 'active'),
(104, '2022-06-01', '2025-12-31', 101, 'Bob Brown', 'blocked'),
(105, '2023-01-01', '2026-12-31', 112, 'Charlie Davis', 'active'),
(106, '2023-03-01', '2026-12-31', 131, 'Daisy Evans', 'blocked'),
(107, '2021-11-01', '2024-12-31', 145, 'Eve Foster', 'active'),
(108, '2023-07-01', '2026-12-31', 151, 'Frank Green', 'active'),
(109, '2021-09-01', '2024-12-31', 166, 'Grace Hall', 'active'),
(110, '2021-10-01', '2024-12-31', 177, 'Henry Ivy', 'blocked'),
(111, '2022-04-01', '2025-12-31', 188, 'Isabella Jones', 'active'),
(112, '2023-02-01', '2026-12-31', 199, 'Jack King', 'active'),
(113, '2023-06-01', '2026-12-31', 211, 'Karen Lee', 'blocked'),
(114, '2021-08-01', '2024-12-31', 234, 'Leo Martin', 'active'),
(115, '2022-12-01', '2025-12-31', 345, 'Mia Nelson', 'blocked');
-- Transactions Table Insert
INSERT INTO Transactions (transaction_id, transaction_amount,
transaction_date, transaction_type, transaction_status,
transaction_account_number, transaction_notes, transaction_category)
VALUES
(1, 100.00, '2024-10-01', 'debit', 'completed', '123456789', 'Groceries
purchase', 'Food'),
(2, 50.00, '2024-10-02', 'credit', 'completed', '987654321', 'Salary
deposit', 'Income'),
```

```
(3, 75.00, '2024-10-03', 'debit', 'completed', '112233445', 'Restaurant bill', 'Food'),
(4, 200.00, '2024-10-04', 'debit', 'completed', '998877665', 'Electronics purchase', 'Shopping'),
(5, 150.00, '2024-10-05', 'debit', 'completed', '223344556', 'Clothing purchase', 'Shopping'),
(6, 500.00, '2024-10-06', 'credit', 'completed', '554433221', 'Freelance payment', 'Income'),
(7, 300.00, '2024-10-07', 'debit', 'completed', '665544332', 'Furniture purchase', 'Home'),
(8, 20.00, '2024-10-08', 'debit', 'completed', '887766554', 'Coffee shop', 'Food'),
(9, 400.00, '2024-10-09', 'debit', 'completed', '998877776', 'Gym equipment', 'Fitness'),
(10, 50.00, '2024-10-10', 'debit', 'completed', '332211554', 'Taxi fare', 'Transport'),
(11, 100.00, '2024-10-11', 'credit', 'completed', '776655443', 'Refund from store', 'Refund'),
(12, 150.00, '2024-10-12', 'debit', 'completed', '554433776', 'Gas station', 'Transport'),
(13, 600.00, '2024-10-13', 'credit', 'completed', '221133445', 'Bonus payment', 'Income'),
(14, 75.00, '2024-10-14', 'debit', 'completed', '998844556', 'Movie tickets', 'Entertainment'),
(15, 120.00, '2024-10-15', 'debit', 'completed', '665533221', 'Grocery shopping', 'Food');
-- Debit Card Table Insert
INSERT INTO Debit_Card (debit_card_number, overdraft_limit, daily_withdrawal_limit)
VALUES
(1, 500.00, 1000.00),
(2, 1000.00, 2000.00),
(3, 1500.00, 3000.00),
(4, 2000.00, 4000.00),
(5, 2500.00, 5000.00),
(6, 3000.00, 6000.00),
(7, 3500.00, 7000.00),
(8, 4000.00, 8000.00),
(9, 4500.00, 9000.00),
(10, 5000.00, 10000.00),
(11, 5500.00, 11000.00),
(12, 6000.00, 12000.00),
(13, 6500.00, 13000.00),
(14, 7000.00, 14000.00),
(15, 7500.00, 15000.00);
-- Account Table Insert
INSERT INTO Account (account_id, account_number, customer_id, routing_number, minimum_balance, account_balance, branch_id)
VALUES
(1, 'ACC001', 1, '111000111', 100.00, 5000.00, 1),
(2, 'ACC002', 2, '222000222', 200.00, 6000.00, 2),
```

```
(3, 'ACC003', 3, '333000333', 300.00, 7000.00, 3),
(4, 'ACC004', 4, '444000444', 400.00, 8000.00, 4),
(5, 'ACC005', 5, '555000555', 500.00, 9000.00, 5),
(6, 'ACC006', 6, '666000666', 600.00, 10000.00, 6),
(7, 'ACC007', 7, '777000777', 700.00, 11000.00, 7),
(8, 'ACC008', 8, '888000888', 800.00, 12000.00, 8),
(9, 'ACC009', 9, '999000999', 900.00, 13000.00, 9),
(10, 'ACC010', 10, '101010101', 1000.00, 14000.00, 10),
(11, 'ACC011', 11, '111111111', 1100.00, 15000.00, 11),
(12, 'ACC012', 12, '121212121', 1200.00, 16000.00, 12),
(13, 'ACC013', 13, '131313131', 1300.00, 17000.00, 13),
(14, 'ACC014', 14, '141414141', 1400.00, 18000.00, 14),
(15, 'ACC015', 15, '151515151', 1500.00, 19000.00, 15);
-- Saving Account Table Insert
INSERT INTO Saving_Account (account_id, savings_account_interest_rate,
savings_account_duration, withdrawal_per_limit)
VALUES
(1, 1.50, 12, 500.00),
(2, 1.75, 24, 600.00),
(3, 2.00, 36, 700.00),
(4, 2.25, 48, 800.00),
(5, 2.50, 60, 900.00),
(6, 1.80, 12, 1000.00),
(7, 1.60, 24, 1100.00),
(8, 2.10, 36, 1200.00),
(9, 2.30, 48, 1300.00),
(10, 1.90, 60, 1400.00),
(11, 2.05, 12, 1500.00),
(12, 2.35, 24, 1600.00),
(13, 2.45, 36, 1700.00),
(14, 2.55, 48, 1800.00),
(15, 2.65, 60, 1900.00);
-- Loan Table Insert
INSERT INTO Loan (loan_id, loan_duration, loan_approver_employee_id,
loan_start_date, loan_interest_rate, loan_amount_taken,
loan_amount_repaid, customer_id)
VALUES
(1, 12, 1, '2024-01-01', 3.50, 10000.00, 2000.00, 1),
(2, 24, 2, '2024-02-01', 4.00, 15000.00, 5000.00, 2),
(3, 36, 3, '2024-03-01', 4.50, 20000.00, 7000.00, 3),
(4, 48, 4, '2024-04-01', 5.00, 25000.00, 10000.00, 4),
(5, 60, 5, '2024-05-01', 3.75, 30000.00, 12000.00, 5),
(6, 12, 6, '2024-06-01', 4.25, 12000.00, 2000.00, 6),
(7, 24, 7, '2024-07-01', 3.90, 18000.00, 5000.00, 7),
(8, 36, 8, '2024-08-01', 4.10, 22000.00, 7000.00, 8),
(9, 48, 9, '2024-09-01', 4.30, 25000.00, 10000.00, 9),
(10, 60, 10, '2024-10-01', 4.60, 30000.00, 15000.00, 10),
(11, 12, 11, '2024-11-01', 3.80, 10000.00, 5000.00, 11),
(12, 24, 12, '2024-12-01', 4.20, 15000.00, 6000.00, 12),
(13, 36, 13, '2025-01-01', 4.40, 20000.00, 8000.00, 13),
(14, 48, 14, '2025-02-01', 4.50, 25000.00, 10000.00, 14),
```

```
(15, 60, 15, '2025-03-01', 4.75, 30000.00, 15000.00, 15);
-- Chat Session Table Insert
INSERT INTO ChatSession (session_id, customer_id, start_time, end_time,
status)
VALUES
(1, 1, '2024-10-01 09:00:00', '2024-10-01 09:15:00', 'completed'),
(2, 2, '2024-10-01 10:00:00', '2024-10-01 10:10:00', 'completed'),
(3, 3, '2024-10-02 11:00:00', '2024-10-02 11:20:00', 'completed'),
(4, 4, '2024-10-02 12:00:00', '2024-10-02 12:25:00', 'completed'),
(5, 5, '2024-10-03 13:00:00', '2024-10-03 13:30:00', 'completed'),
(6, 6, '2024-10-03 14:00:00', '2024-10-03 14:10:00', 'completed'),
(7, 7, '2024-10-04 15:00:00', '2024-10-04 15:15:00', 'completed'),
(8, 8, '2024-10-04 16:00:00', '2024-10-04 16:20:00', 'completed'),
(9, 9, '2024-10-05 17:00:00', '2024-10-05 17:25:00', 'completed'),
(10, 10, '2024-10-05 18:00:00', '2024-10-05 18:15:00', 'completed'),
(11, 11, '2024-10-06 09:00:00', '2024-10-06 09:30:00', 'completed'),
(12, 12, '2024-10-06 10:00:00', '2024-10-06 10:15:00', 'completed'),
(13, 13, '2024-10-07 11:00:00', '2024-10-07 11:20:00', 'completed'),
(14, 14, '2024-10-07 12:00:00', '2024-10-07 12:15:00', 'completed'),
(15, 15, '2024-10-08 13:00:00', '2024-10-08 13:25:00', 'completed');
-- Message Table Insert
INSERT INTO Message (message_id, session_id, message_content,
message_type, response_content, timestamp)
VALUES
(1, 1, 'Hello, I need help with my account', 'user_message', 'Sure, how
can I assist you?', '2024-10-01 09:00:00'),
(2, 1, 'I forgot my password', 'user_message', 'Let me reset that for
you', '2024-10-01 09:05:00'),
(3, 2, 'What are your opening hours?', 'user_message', 'We are open from 9
AM to 5 PM', '2024-10-01 10:05:00'),
(4, 2, 'Thank you!', 'user_message', 'You''re welcome!', '2024-10-01
10:07:00'),
(5, 3, 'Can I change my address online?', 'user_message', 'Yes, you can
update it in your profile', '2024-10-02 11:05:00'),
(6, 4, 'I need help with a loan application', 'user_message', 'I will
transfer you to the loan department', '2024-10-02 12:10:00'),
(7, 5, 'How do I get a new debit card?', 'user_message', 'You can request
it through our app', '2024-10-03 13:10:00'),
(8, 6, 'Can I cancel a transaction?', 'user_message', 'Please provide the
transaction details', '2024-10-03 14:05:00'),
(9, 7, 'I am having trouble logging in', 'user_message', 'Let me assist
you with that', '2024-10-04 15:05:00'),
(10, 8, 'Is there a fee for transferring money?', 'user_message', 'Yes, a
small fee applies', '2024-10-04 16:10:00'),
(11, 9, 'How do I set up direct deposit?', 'user_message', 'You can do
that through your employer', '2024-10-05 17:10:00'),
(12, 10, 'What is the interest rate on savings accounts?', 'user_message',
'Our current rate is 2.5%', '2024-10-05 18:05:00'),
(13, 11, 'Can I open a new checking account?', 'user_message', 'Yes, I can
help with that', '2024-10-06 09:10:00'),
```

```
(14, 12, 'How do I get a copy of my statement?', 'user_message', 'You can download it from your account', '2024-10-06 10:10:00'),
(15, 13, 'I need to update my contact information', 'user_message', 'You can do that online', '2024-10-07 11:10:00');

-- Customer Service Table Insert
INSERT INTO Customer_Service (customer_employee_id, customer_id, employee_id, interaction_type, service_type)
VALUES
(1, 1, 1, 'phone', 'account assistance'),
(2, 2, 2, 'in-person', 'loan inquiry'),
(3, 3, 3, 'email', 'address update'),
(4, 4, 4, 'chat', 'password reset'),
(5, 5, 5, 'in-person', 'debit card issue'),
(6, 6, 6, 'phone', 'transaction inquiry'),
(7, 7, 7, 'chat', 'login assistance'),
(8, 8, 8, 'email', 'money transfer fee'),
(9, 9, 9, 'phone', 'direct deposit setup'),
(10, 10, 10, 'in-person', 'interest rate inquiry'),
(11, 11, 11, 'email', 'checking account opening'),
(12, 12, 12, 'phone', 'statement request'),
(13, 13, 13, 'chat', 'contact information update'),
(14, 14, 14, 'in-person', 'loan application'),
(15, 15, 15, 'phone', 'debit card activation');


SELECT * FROM Customer;
```

	customer_id	name	gender	date_of_birth	address	customer_ssn	income	login_id	username	password	last_login	login_attempt	is_locked	savings_goal
1	1	John Doe	M	1985-01-01	123 Elm Street	123,456,789	55,000	1	johndoe	password123	2024-10-01 10:00:00.000	1	[ ]	5,000
2	2	Jane Smith	F	1990-05-05	456 Oak Street	987,654,321	62,000	2	janesmith	password456	2024-10-01 11:00:00.000	0	[ ]	6,000
3	3	Alice Johnson	F	1988-08-08	789 Maple Street	112,233,445	48,000	3	alicej	password789	2024-10-02 09:30:00.000	2	[v]	4,000
4	4	Bob Brown	M	1975-02-20	234 Birch Street	998,877,665	75,000	4	bobbrown	password111	2024-10-02 12:45:00.000	0	[ ]	10,000
5	5	Charlie Davis	M	1995-03-15	567 Cedar Street	223,344,556	82,000	5	charlied	password222	2024-10-03 08:00:00.000	1	[ ]	12,000
6	6	Daisy Evans	F	1992-07-25	890 Pine Street	554,433,221	68,000	6	daisye	password333	2024-10-04 10:30:00.000	1	[ ]	7,000
7	7	Eve Foster	F	1981-11-11	123 Spruce Street	665,544,332	55,000	7	evef	password444	2024-10-04 11:45:00.000	3	[v]	4,500
8	8	Frank Green	M	1993-04-05	456 Willow Street	887,766,554	50,000	8	frankg	password555	2024-10-05 09:15:00.000	0	[ ]	5,500
9	9	Grace Hall	F	1994-06-12	789 Redwood Street	998,877,776	60,000	9	graceh	password666	2024-10-05 14:00:00.000	1	[ ]	6,500
10	10	Henry Ivy	M	1986-09-09	234 Chestnut Street	332,211,554	70,000	10	henryi	password777	2024-10-06 16:30:00.000	2	[v]	8,000
11	11	Isabella Jones	F	1989-10-10	567 Walnut Street	776,655,443	72,000	11	isabellaj	password888	2024-10-07 17:15:00.000	1	[ ]	8,500
12	12	Jack King	M	1991-12-15	890 Poplar Street	554,433,776	58,000	12	jackk	password999	2024-10-08 08:45:00.000	0	[ ]	6,000
13	13	Karen Lee	F	1996-03-18	123 Dogwood Street	221,133,445	63,000	13	karenl	password101	2024-10-09 10:30:00.000	1	[ ]	7,000
14	14	Leo Martin	M	1982-07-20	456 Fir Street	998,844,556	78,000	14	leom	password202	2024-10-09 11:00:00.000	2	[v]	9,000
15	15	Mia Nelson	F	1987-08-30	789 Alder Street	665,533,221	54,000	15	mian	password303	2024-10-09 13:00:00.000	3	[ ]	4,500

```
SELECT * FROM Branch;
```

	branch_id	branch_name	branch_address
1	1	Downtown	123 Main St
2	2	Uptown	456 Broadway Ave
3	3	East Side	789 East Rd
4	4	West Side	234 West Blvd
5	5	North Side	567 North St
6	6	South Side	890 South Rd
7	7	Central	123 Center St
8	8	Suburban	456 Suburb Dr
9	9	Riverside	789 River Ln
10	10	Hilltop	234 Hilltop Ave
11	11	Valley	567 Valley Rd
12	12	Lakeside	890 Lake Rd
13	13	Woodland	123 Woodland Dr
14	14	Mountain	456 Mountain Rd
15	15	Seaside	789 Seaside Dr

```
SELECT * FROM Employee;
```

	employee_id	name	phone_no	role	gender	date_of_birth	branch_id	address	manager_id	department_id
1	1	Anna Smith	123-456-7890	Manager	F	1980-03-15	1	123 Main St	[NULL]	1
2	2	Ben Williams	321-654-0987	Teller	M	1985-05-25	1	456 Oak St	1	1
3	3	Cathy Johnson	555-666-7777	Loan Officer	F	1990-08-11	2	789 Pine St	1	2
4	4	David Lee	444-333-2222	Clerk	M	1987-12-22	3	234 Cedar St	1	2
5	5	Emily Brown	999-888-7777	Manager	F	1981-07-01	4	567 Spruce St	[NULL]	1
6	6	Frank Green	111-222-3333	Security	M	1975-11-15	5	890 Birch St	5	1
7	7	Grace Hall	444-555-6666	HR	F	1992-10-08	6	234 Willow St	5	1
8	8	Henry Ivy	777-888-9999	Technician	M	1983-09-02	7	567 Redwood St	7	2
9	9	Ivy Nelson	222-333-4444	Cashier	F	1994-04-06	8	890 Walnut St	8	3
10	10	Jack King	123-321-4567	Supervisor	M	1986-02-11	9	234 Poplar St	9	3
11	11	Karen Davis	654-987-3210	Analyst	F	1991-03-30	10	567 Dogwood St	10	4
12	12	Leo Martin	555-666-1111	Loan Officer	M	1989-05-21	11	890 Fir St	11	5
13	13	Mia Nelson	222-111-4444	Accountant	F	1992-12-17	12	123 Alder St	12	6
14	14	Noah Parker	333-555-7777	Marketing	M	1987-09-27	13	456 Seaside Dr	13	7
15	15	Olivia Scott	999-000-1234	Manager	F	1982-11-07	14	789 Hilltop Rd	[NULL]	8

```
SELECT * FROM Department;
```

	department_id	department_name	department_head_employee_id
1	1	HR	[NULL]
2	2	Loan Services	[NULL]
3	3	Customer Service	[NULL]
4	4	Security	[NULL]
5	5	Technical	[NULL]
6	6	Marketing	[NULL]
7	7	Accounting	[NULL]
8	8	Management	[NULL]

```
SELECT * FROM Credit_Card;
```

	credit_card_number	credit_limit	outstanding_balance	interest_rate	credit_score
1	1234567812345678	10,000	2,000	3.5	700
2	2234567812345678	15,000	5,000	4	750
3	3234567812345680	12,000	3,000	4.5	680
4	4234567812345681	18,000	6,000	3.8	720
5	5234567812345682	20,000	7,000	5	690
6	6234567812345683	25,000	8,000	2.9	710
7	7234567812345684	9,000	1,000	3.6	760
8	8234567812345685	8,000	2,000	3.7	740
9	9234567812345686	14,000	3,000	3.4	710
10	1034567812345687	16,000	4,000	2.8	750
11	1134567812345688	11,000	1,200	3	780
12	1234567812345689	22,000	7,000	4.6	690
13	1334567812345690	10,000	2,000	3.3	720
14	1434567812345691	15,000	5,000	3.5	730
15	1534567812345692	13,000	500	3	740

```
SELECT * FROM Credit_Card_Transaction;
```

SELECT * FROM Credit_Card_Transaction | Enter a SQL expression to filter results (use Ctrl+Space)

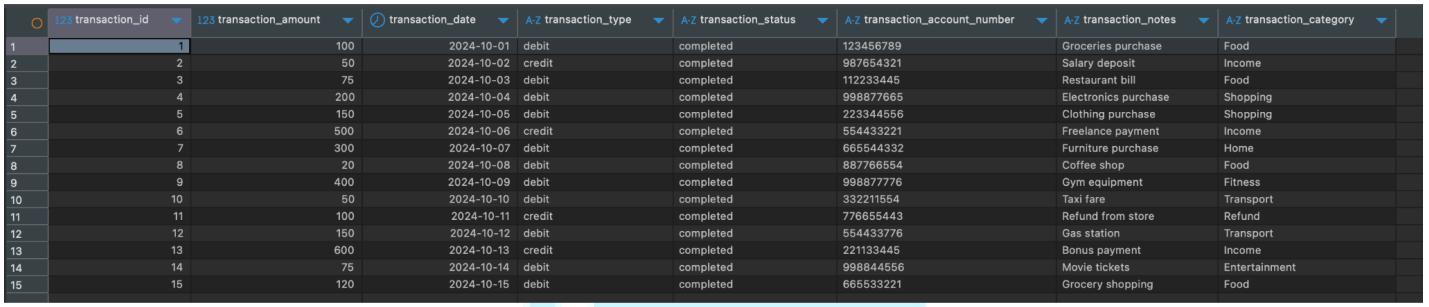
Grid | Text

	credit_card_transaction_id	credit_card_transaction_date	credit_card_transaction_amount	credit_card_number	credit_card_transaction_description
1	1	2024-10-01 10:00:00.000	100	1234567812345678	Groceries
2	2	2024-10-01 11:30:00.000	50	2234567812345678	Gas
3	3	2024-10-02 14:00:00.000	200	3234567812345680	Electronics
4	4	2024-10-02 17:15:00.000	150	4234567812345681	Clothing
5	5	2024-10-03 09:00:00.000	300	5234567812345682	Furniture
6	6	2024-10-03 13:45:00.000	75	6234567812345683	Restaurant
7	7	2024-10-04 12:30:00.000	125	7234567812345684	Movies
8	8	2024-10-04 15:00:00.000	250	8234567812345685	Books
9	9	2024-10-05 10:15:00.000	80	9234567812345686	Taxi
10	10	2024-10-05 14:00:00.000	60	1034567812345687	Gift
11	11	2024-10-06 09:45:00.000	220	1134567812345688	Travel
12	12	2024-10-06 12:30:00.000	40	1234567812345689	Coffee
13	13	2024-10-07 08:00:00.000	350	1334567812345690	Gym Equipment
14	14	2024-10-07 10:15:00.000	90	1434567812345691	Shoes
15	15	2024-10-08 09:30:00.000	110	1534567812345692	Hotel

```
SELECT * FROM Checking_Account;
```

	account_id	service_charge	checkbook_available	account_balance	daily_transaction_limit
1	1	15	[v]	1,000	5,000
2	2	10	[ ]	2,000	6,000
3	3	12.5	[v]	3,000	7,000
4	4	18	[v]	5,000	8,000
5	5	20	[ ]	7,000	9,000
6	6	15.5	[v]	8,000	10,000
7	7	25	[v]	6,000	7,500
8	8	10	[ ]	9,000	6,500
9	9	5	[v]	12,000	8,500
10	10	15	[v]	15,000	10,000
11	11	12	[v]	5,000	12,000
12	12	17.5	[ ]	8,000	11,000
13	13	20	[v]	14,000	9,500
14	14	25	[ ]	20,000	15,000
15	15	22	[v]	25,000	18,000

```
SELECT * FROM Transactions;
```



	transaction_id	transaction_amount	transaction_date	transaction_type	transaction_status	transaction_account_number	transaction_notes	transaction_category
1	1	100	2024-10-01	debit	completed	123456789	Groceries purchase	Food
2	2	50	2024-10-02	credit	completed	987654321	Salary deposit	Income
3	3	75	2024-10-03	debit	completed	112233445	Restaurant bill	Food
4	4	200	2024-10-04	debit	completed	998877665	Electronics purchase	Shopping
5	5	150	2024-10-05	debit	completed	223344556	Clothing purchase	Shopping
6	6	500	2024-10-06	credit	completed	554433221	Freelance payment	Income
7	7	300	2024-10-07	debit	completed	665544332	Furniture purchase	Home
8	8	20	2024-10-08	debit	completed	887766554	Coffee shop	Food
9	9	400	2024-10-09	debit	completed	998877776	Gym equipment	Fitness
10	10	50	2024-10-10	debit	completed	332211554	Taxi fare	Transport
11	11	100	2024-10-11	credit	completed	776655443	Refund from store	Refund
12	12	150	2024-10-12	debit	completed	554433776	Gas station	Transport
13	13	600	2024-10-13	credit	completed	221133445	Bonus payment	Income
14	14	75	2024-10-14	debit	completed	998844556	Movie tickets	Entertainment
15	15	120	2024-10-15	debit	completed	665533221	Grocery shopping	Food

```
SELECT * FROM Card;
```

	card_number	card_open_date	card_expiry_date	cvv	cardholder	card_status
1	1	2021-01-01	2024-12-31	123	John Doe	active
2	2	2021-05-01	2024-12-31	456	Jane Smith	active
3	3	2022-01-01	2025-12-31	789	Alice Johnson	active
4	4	2022-06-01	2025-12-31	101	Bob Brown	blocked
5	5	2023-01-01	2026-12-31	112	Charlie Davis	active
6	6	2023-03-01	2026-12-31	131	Daisy Evans	blocked
7	7	2021-11-01	2024-12-31	145	Eve Foster	active
8	8	2023-07-01	2026-12-31	151	Frank Green	active
9	9	2021-09-01	2024-12-31	166	Grace Hall	active
10	10	2021-10-01	2024-12-31	177	Henry Ivy	blocked
11	11	2022-04-01	2025-12-31	188	Isabella Jones	active
12	12	2023-02-01	2026-12-31	199	Jack King	active
13	13	2023-06-01	2026-12-31	111	Karen Lee	blocked
14	14	2021-08-01	2024-12-31	212	Leo Martin	active
15	15	2022-12-01	2025-12-31	234	Mia Nelson	blocked
16	101	2021-01-01	2024-12-31	123	John Doe	active
17	102	2021-05-01	2024-12-31	456	Jane Smith	active
18	103	2022-01-01	2025-12-31	789	Alice Johnson	active
19	104	2022-06-01	2025-12-31	101	Bob Brown	blocked
20	105	2023-01-01	2026-12-31	112	Charlie Davis	active
21	106	2023-03-01	2026-12-31	131	Daisy Evans	blocked
22	107	2021-11-01	2024-12-31	145	Eve Foster	active
23	108	2023-07-01	2026-12-31	151	Frank Green	active
24	109	2021-09-01	2024-12-31	166	Grace Hall	active
25	110	2021-10-01	2024-12-31	177	Henry Ivy	blocked
26	111	2022-04-01	2025-12-31	188	Isabella Jones	active
27	112	2023-02-01	2026-12-31	199	Jack King	active
28	113	2023-06-01	2026-12-31	211	Karen Lee	blocked
29	114	2021-08-01	2024-12-31	234	Leo Martin	active
30	115	2022-12-01	2025-12-31	345	Mia Nelson	blocked

```
SELECT * FROM Debit_Card;
```

	123 debit_card_number	123 overdraft_limit	123 daily_withdrawal_limit
1	1	500	1,000
2	2	1,000	2,000
3	3	1,500	3,000
4	4	2,000	4,000
5	5	2,500	5,000
6	6	3,000	6,000
7	7	3,500	7,000
8	8	4,000	8,000
9	9	4,500	9,000
10	10	5,000	10,000
11	11	5,500	11,000
12	12	6,000	12,000
13	13	6,500	13,000
14	14	7,000	14,000
15	15	7,500	15,000

```
SELECT * FROM Account;
```

	123 account_id	A-Z account_number	123 customer_id	A-Z routing_number	123 minimum_balance	123 account_balance	123 branch_id
1	1	ACC001	1	111000111	100	5,000	1
2	2	ACC002	2	222000222	200	6,000	2
3	3	ACC003	3	333000333	300	7,000	3
4	4	ACC004	4	444000444	400	8,000	4
5	5	ACC005	5	555000555	500	9,000	5
6	6	ACC006	6	666000666	600	10,000	6
7	7	ACC007	7	777000777	700	11,000	7
8	8	ACC008	8	888000888	800	12,000	8
9	9	ACC009	9	999000999	900	13,000	9
10	10	ACC010	10	101010101	1,000	14,000	10
11	11	ACC011	11	111111111	1,100	15,000	11
12	12	ACC012	12	121212121	1,200	16,000	12
13	13	ACC013	13	131313131	1,300	17,000	13
14	14	ACC014	14	141414141	1,400	18,000	14
15	15	ACC015	15	151515151	1,500	19,000	15

```
SELECT * FROM Saving_Account;
```

	123 account_id	123 savings_account_interest_rate	123 savings_account_duration	123 withdrawal_per_limit
1	1	1.5	12	500
2	2	1.75	24	600
3	3	2	36	700
4	4	2.25	48	800
5	5	2.5	60	900
6	6	1.8	12	1,000
7	7	1.6	24	1,100
8	8	2.1	36	1,200
9	9	2.3	48	1,300
10	10	1.9	60	1,400
11	11	2.05	12	1,500
12	12	2.35	24	1,600
13	13	2.45	36	1,700
14	14	2.55	48	1,800
15	15	2.65	60	1,900

```
SELECT * FROM Loan;
```

	loan_id	loan_duration	loan_approver_employee_id	loan_start_date	loan_interest_rate	loan_amount_taken	loan_amount_repaid	customer_id
1	1	12	1	2024-01-01	3.5	10,000	2,000	1
2	2	24	2	2024-02-01	4	15,000	5,000	2
3	3	36	3	2024-03-01	4.5	20,000	7,000	3
4	4	48	4	2024-04-01	5	25,000	10,000	4
5	5	60	5	2024-05-01	3.75	30,000	12,000	5
6	6	12	6	2024-06-01	4.25	12,000	2,000	6
7	7	24	7	2024-07-01	3.9	18,000	5,000	7
8	8	36	8	2024-08-01	4.1	22,000	7,000	8
9	9	48	9	2024-09-01	4.3	25,000	10,000	9
10	10	60	10	2024-10-01	4.6	30,000	15,000	10
11	11	12	11	2024-11-01	3.8	10,000	5,000	11
12	12	24	12	2024-12-01	4.2	15,000	6,000	12
13	13	36	13	2025-01-01	4.4	20,000	8,000	13
14	14	48	14	2025-02-01	4.5	25,000	10,000	14
15	15	60	15	2025-03-01	4.75	30,000	15,000	15

```
SELECT * FROM ChatSession;
```

Grid Text

	session_id	customer_id	start_time	end_time	status
1	1	1	2024-10-01 09:00:00.000	2024-10-01 09:15:00.000	completed
2	2	2	2024-10-01 10:00:00.000	2024-10-01 10:10:00.000	completed
3	3	3	2024-10-02 11:00:00.000	2024-10-02 11:20:00.000	completed
4	4	4	2024-10-02 12:00:00.000	2024-10-02 12:25:00.000	completed
5	5	5	2024-10-03 13:00:00.000	2024-10-03 13:30:00.000	completed
6	6	6	2024-10-03 14:00:00.000	2024-10-03 14:10:00.000	completed
7	7	7	2024-10-04 15:00:00.000	2024-10-04 15:15:00.000	completed
8	8	8	2024-10-04 16:00:00.000	2024-10-04 16:20:00.000	completed
9	9	9	2024-10-05 17:00:00.000	2024-10-05 17:25:00.000	completed
10	10	10	2024-10-05 18:00:00.000	2024-10-05 18:15:00.000	completed
11	11	11	2024-10-06 09:00:00.000	2024-10-06 09:30:00.000	completed
12	12	12	2024-10-06 10:00:00.000	2024-10-06 10:15:00.000	completed
13	13	13	2024-10-07 11:00:00.000	2024-10-07 11:20:00.000	completed
14	14	14	2024-10-07 12:00:00.000	2024-10-07 12:15:00.000	completed
15	15	15	2024-10-08 13:00:00.000	2024-10-08 13:25:00.000	completed

```
SELECT * FROM Message;
```

	message_id	session_id	message_content	message_type	response_content	timestamp
1	1	1	Hello, I need help with my account	user_message	Sure, how can I assist you?	2024-10-01 09:00:00.000
2	2	1	I forgot my password	user_message	Let me reset that for you	2024-10-01 09:05:00.000
3	3	2	What are your opening hours?	user_message	We are open from 9 AM to 5 PM	2024-10-01 10:05:00.000
4	4	2	Thank you!	user_message	You're welcome!	2024-10-01 10:07:00.000
5	5	3	Can I change my address online?	user_message	Yes, you can update it in your profile	2024-10-02 11:05:00.000
6	6	4	I need help with a loan application	user_message	I will transfer you to the loan department	2024-10-02 12:10:00.000
7	7	5	How do I get a new debit card?	user_message	You can request it through our app	2024-10-03 13:10:00.000
8	8	6	Can I cancel a transaction?	user_message	Please provide the transaction details	2024-10-03 14:05:00.000
9	9	7	I am having trouble logging in	user_message	Let me assist you with that	2024-10-04 15:05:00.000
10	10	8	Is there a fee for transferring money?	user_message	Yes, a small fee applies	2024-10-04 16:10:00.000
11	11	9	How do I set up direct deposit?	user_message	You can do that through your employer	2024-10-05 17:10:00.000
12	12	10	What is the interest rate on savings accounts?	user_message	Our current rate is 2.5%	2024-10-05 18:05:00.000
13	13	11	Can I open a new checking account?	user_message	Yes, I can help with that	2024-10-06 09:10:00.000
14	14	12	How do I get a copy of my statement?	user_message	You can download it from your account	2024-10-06 10:10:00.000
15	15	13	I need to update my contact information	user_message	You can do that online	2024-10-07 11:10:00.000

```
SELECT * FROM Customer_Service;
```

	123 customer_employee_id	123 customer_id	123 employee_id	A-Z interaction_type	A-Z service_type
1	1	1	1	phone	account assistance
2	2	2	2	in-person	loan inquiry
3	3	3	3	email	address update
4	4	4	4	chat	password reset
5	5	5	5	in-person	debit card issue
6	6	6	6	phone	transaction inquiry
7	7	7	7	chat	login assistance
8	8	8	8	email	money transfer fee
9	9	9	9	phone	direct deposit setup
10	10	10	10	in-person	interest rate inquiry
11	11	11	11	email	checking account opening
12	12	12	12	phone	statement request
13	13	13	13	chat	contact information update
14	14	14	14	in-person	loan application
15	15	15	15	phone	debit card activation