ORGANIZATIONAL DATA STORAGE MANAGEMENT SYSTEM FOR BANKS.

Author: KATHIRAVAN A

Version: 0.1

Author ID: 201021048

# Abstract:

The Data Storage Management System for banks is a software tool created to help banks manage their employees and customers efficiently. It's divided into sections for different types of bank staff, such as managers and clerks, and various customer categories, including those with loans or savings accounts. This system ensures data is organized and accessible securely. It operates using a structured method of storing information and functions online. The primary aim is to securely store essential data and retrieve it promptly when required. By implementing this system, banks can streamline their operations and provide better service to their customers.

# 1 Introduction:

In today's fast-paced world, banks play a crucial role in our daily lives, managing our money and facilitating financial transactions. Behind the scenes, there's a lot of information that needs to be organized and managed efficiently to keep everything running smoothly. This is where the Bank Database Management System (DBMS) comes into play. The Bank DBMS is like a digital filing cabinet for a bank, storing and organizing all the important information about employees and customers. Picture it as a big, virtual bookshelf with different sections for different types of people and accounts. There are different roles within a bank: there are managers who oversee operations, clerks who handle transactions, and other staff members who keep things running behind the scenes. Each of these roles has specific tasks and responsibilities, and the DBMS helps keep track of who's doing what. On the customer side, banks serve a diverse range of people with various financial needs. Some have savings accounts where they keep their money safe, while others have loans they're paying off. The DBMS keeps track of all these accounts and helps ensure that customers' needs are met promptly and accurately.

In the digital age, security is a top priority for banks. They need to make sure that sensitive information, like account numbers and personal details, is kept safe from hackers and unauthorized access. The Bank DBMS uses advanced technology and encryption methods to protect this information, giving customers peace of mind when they trust their money to the bank. Overall, the Bank DBMS is an essential tool for modern banks, helping them manage their operations efficiently, serve their customers effectively, and maintain the highest standards of security and reliability. Without it, the complex world of banking would be much more challenging to navigate.

## 1.1 Purpose of project:

The main reason to create this project is to help banks do their job better. Banks have a lot of information to manage, like details about their employees and customers. Our project, the Bank Database Management System (DBMS), will make it easier for banks to keep track of all this important information. With the Bank DBMS, banks will be able to organize their data neatly and access it quickly whenever they need to. This will help them serve their customers faster and more efficiently. Plus, it'll make sure that sensitive information, like account numbers and personal details, stays safe and secure. Overall, our goal is to make banking easier and safer for everyone involved – from the bank employees who use the system every day to the customers who trust their money with the bank.

## 1.2 Scope of the project:

**Employee Management Module:**

Creation and maintenance of employee records, including personal details, roles, and privileges. Functionality for categorizing employees into distinct roles such as Manager, Clerk, Normal Employee, and Non-technical Worker. Implementation of authentication mechanisms to control access to employee data based on their roles and privileges.

**Customer Management Module:**

Storage and management of customer information, including account details, transaction history, and contact information. Handling various types of customers, such as Account Holders, Loan Purchased Customers, and Saving Account Customers. Support for account creation, updates, and deletion, along with functionalities for managing customer accounts and transactions.

**Authentication Module:**

Implementation of secure login and logout functionalities for admin users. Verification of admin credentials to ensure authorized access to the DBMS. Integration of security measures such as password encryption and session management to protect against unauthorized access.

## 1.3 Objectives:

**Efficient Data Management:** The primary objective of the project is to develop a Bank Database Management System (DBMS) that facilitates the efficient storage and management of employee and customer data within banking institutions.

**Streamlined Operations:** The project aims to streamline banking operations by providing a structured framework for organizing and accessing critical information, thus improving overall efficiency and productivity.

**User-Friendly Interface:** Providing a user-friendly interface for bank employees to interact with the DBMS is another objective. Intuitive design and navigation will contribute to increased user adoption and productivity.

**Enhanced Security:** Implement robust security measures to protect sensitive data from unauthorized access and breaches, safeguarding the integrity and confidentiality of customer and employee information.

# 2.Functional Requirements

# 2.1. Modules:

## 2.1.1 Employee Management Module:

The Employee Management Module is a crucial component of the Bank Database Management System (DBMS), responsible for handling the information related to bank employees. This module encompasses the creation, organization, and maintenance of employee records, ensuring that pertinent details such as personal information, roles, and privileges are accurately stored and managed. Within the Employee Management Module, different types of employees are categorized based on their roles and responsibilities within the bank. These categories typically include:

### Higher Employees:

Higher employees are responsible for overseeing the day-to-day operations of the bank branch or department. Their roles may include setting goals, managing staff, and ensuring compliance with organizational policies and procedures. Managers typically have elevated privileges within the DBMS, allowing them access to sensitive information and administrative functionalities.

### Normal Employee:

Normal employees encompass various staff members who perform general duties within the bank, such as administrative tasks, customer service, or support roles. While their responsibilities may vary, normal employees are essential for the overall functioning of the bank branch or department. Their access to the DBMS may be restricted compared to managers and clerks, depending on their specific roles and privileges.

### Non-technical Workers:

Non-technical workers include staff members who perform manual or non-technical tasks within the bank, such as janitorial staff, security personnel, or maintenance workers. While their roles may not directly involve customer interactions or financial transactions, they still play a crucial role in maintaining the physical infrastructure and security of the bank premises. Non-technical workers may have limited access to the DBMS, primarily for updating their personal information or accessing relevant operational documents.

## 2.1.2 Customer Management Module:

The Customer Management Module of the Bank Database Management System (DBMS) is a pivotal component dedicated to managing the diverse array of customers served by the bank. This module encompasses various functionalities tailored to different types of customers and their respective banking needs.

### Account Holders:

Account holders are individuals or entities who have opened accounts with the bank, such as savings accounts, checking accounts, or investment accounts. The DBMS stores and manages account holder information, including personal details, account types, account balances, transaction history, and contact information. Functionalities for account creation, updates, and deletion are provided, along with features for managing account preferences and settings.

### Loan Purchased Customers:

Loan purchased customers are individuals or businesses that have acquired loans or credit facilities from the bank. The DBMS tracks loan details, including loan types, loan amounts, interest rates, repayment schedules, and outstanding balances. It facilitates the management of loan accounts, including loan origination, disbursement, repayment, and monitoring of payment schedules.

### Savings Account Customers:

Savings account customers are individuals who hold savings accounts with the bank to deposit and save money. The DBMS manages savings account information, including account balances, interest rates, deposit transactions, and withdrawals. It provides functionalities for savings account management, such as depositing funds, withdrawing funds, transferring money between accounts, and setting up automatic transfers or recurring deposits.

## 2.1.3 User Authentication module:

The Authentication Module for Admin within the Bank Database Management System (DBMS) is critical for ensuring secure access to the system's administrative functions. This module provides robust authentication mechanisms to verify the identity of admin users and control their access privileges within the DBMS.

### Secure Login Mechanism:

The module implements a secure login mechanism for admin users, requiring them to enter valid credentials such as username and password. Passwords are encrypted and stored securely within the system to prevent unauthorized access or data breaches.

## Access Control:

Admin users are granted specific access privileges based on their roles and responsibilities within the organization. Access control mechanisms ensure that admin users can only access functionalities and data relevant to their assigned roles, preventing unauthorized access to sensitive information.

### Logout Mechanism:

Admin users can securely logout from the system to terminate their active sessions and prevent unauthorized access to their accounts. The logout mechanism ensures that session data is cleared, and users are redirected to the login page to authenticate again for subsequent access.

### Logging and Auditing:

The module logs authentication-related activities, including login attempts, successful logins, failed login attempts, and logout events. Audit logs provide visibility into user authentication activities, helping administrators monitor system access and identify potential security incidents or anomalies.

## 2.1.4 Bank data management:

The Bank Data Management module is pivotal within the Bank Database Management System (DBMS), comprising three integral components: Bank Data, Bank History, and Bank Details. Each component serves a distinct purpose in managing essential information related to the bank's operations and historical records.

### Bank History:

The Bank History module keeps track of important events and milestones in the bank's journey, offering a detailed record of its growth and development. It includes details like the date the bank was founded, any mergers or partnerships it entered into, its expansion efforts including new branches, changes in regulations affecting the bank, and notable achievements or awards received. Authorized users can view these historical records and make updates to ensure accuracy when necessary. This module serves as a valuable resource for understanding the bank's evolution and legacy.

### Bank Details:

The Bank Details module contains specific information and attributes concerning the bank's operations, facilities, and services, offering a comprehensive overview of its infrastructure and offerings. This includes details about bank branches such as location, contact information, and operating hours, as well as a list of products and services provided by the bank such as accounts, loans, and investment options. Additionally, the module describes the bank's physical infrastructure, encompassing office spaces, ATMs, and technological resources. Authorized users have the capability to access and review this detailed information about the bank's operations and facilities, and they can also update and modify branch information, service offerings, and infrastructure details as needed. This module serves as a crucial resource for understanding and managing the bank's operational framework and service portfolio.

# 3 Non-Functional Requirements:

**Security:** Implement robust authentication mechanisms to ensure only authorized users access the system. Data encryption and secure transmission protocols

**Usability:** The user interface should be intuitive and user-friendly, requiring minimal training for users to navigate and perform tasks.provide error messages to assist users in using the system effectively.

**Documentation:** Comprehensive documentation should be provided, including user manuals, system architecture diagrams, and technical specifications, to facilitate system maintenance and support.

# 3 High Level Design:

The Bank Database Management System (DBMS) will be built using modern web technologies, including React.js for the frontend, Node.js for the backend, and a relational database management system (RDBMS) such as MySQL or PostgreSQL for data storage. The high-level design outlines the architectural components and interactions of the system.

## 3.1 Client-Side (Frontend):

**React.js Framework:** The frontend of the Bank DBMS will be developed using React.js, a popular JavaScript library for building user interfaces. React.js enables the creation of interactive and responsive UI components, providing a seamless user experience.

**User Interface (UI):** The UI will consist of intuitive and user-friendly interfaces for accessing different functionalities of the DBMS, including employee and customer management, authentication, and data visualization.

**Component-Based Architecture:** React.js follows a component-based architecture, allowing for modular development and reusability of UI components across the application.

## 3.2 Server-Side (Backend):

**Node.js Runtime Environment:** The backend of the Bank DBMS will be implemented using Node.js, a runtime environment for executing JavaScript code on the server side. Node.js provides a non-blocking, event-driven architecture, making it well-suited for building scalable and high-performance applications.

**RESTful API:** Node.js will be used to develop RESTful APIs to handle client requests and interact with the database. These APIs will facilitate communication between the frontend and backend components of the DBMS.

## 3.3 Data storage and management:

The Bank Database Management System (DBMS) utilizes MySQL as the primary database management system for storing and managing data effectively. MySQL is a robust relational database management system known for its scalability, reliability, and performance, making it an ideal choice for handling the bank's vast data requirements.

# 4.Low Level Design:

# 5 Diagrams:

## 5.1 Use case diagram:

A use case diagram illustrates the interactions between users (actors) and the system, showcasing the various functionalities and features of the Bank Database Management System (DBMS) from the perspective of different users or stakeholders.

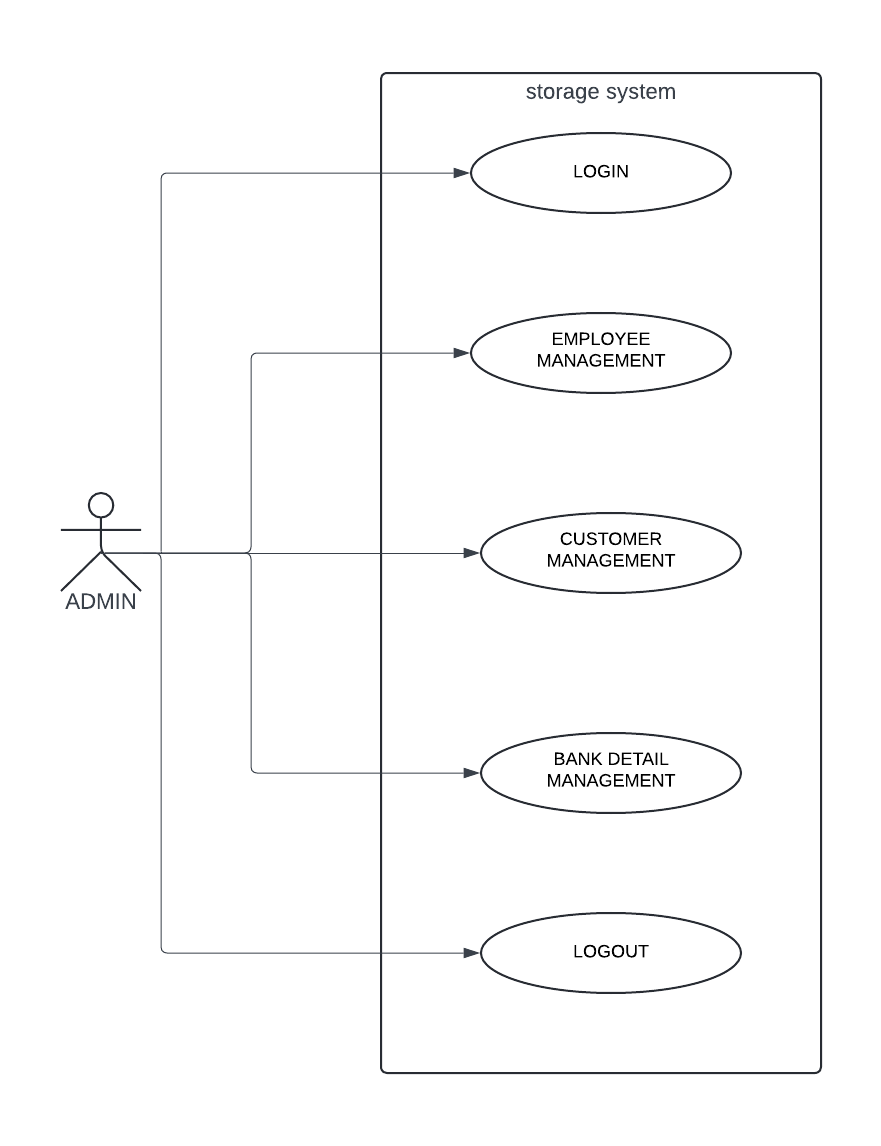


Fig 1: Use case diagram

## 5.2 Class diagram:

A class diagram provides a visual representation of the structure and relationships of the classes within the Bank Database Management System (DBMS). It outlines the entities (classes) in the system, their attributes, methods, and associations with other classes.

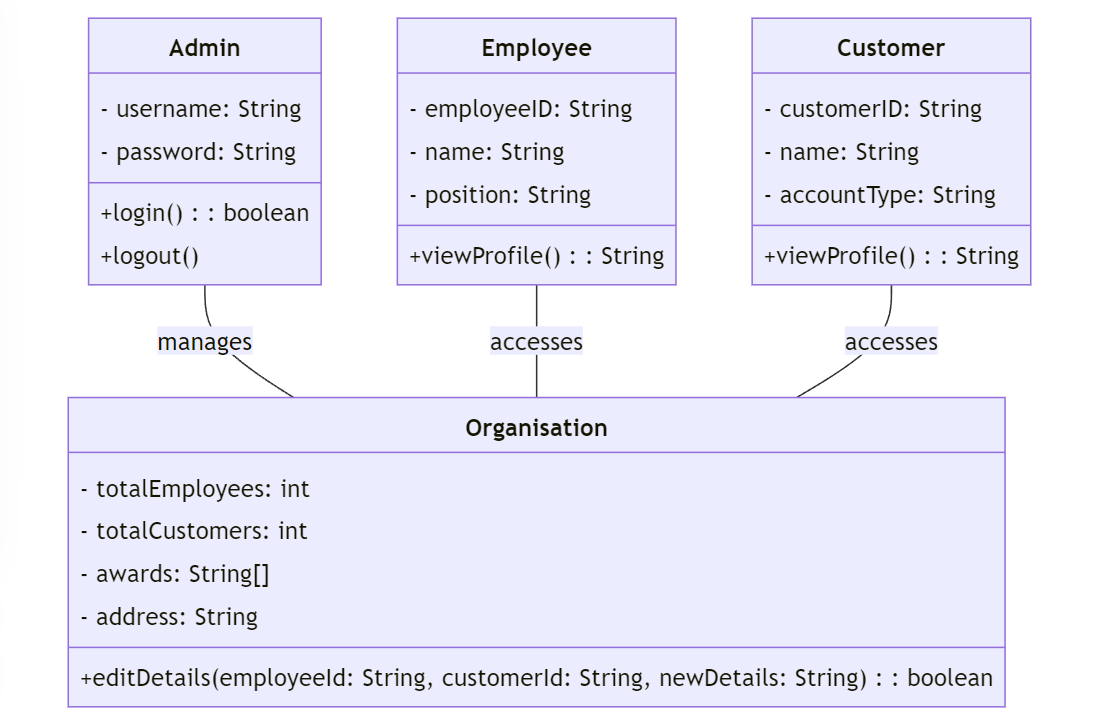


Fig 2: Class diagram

## 5.3 Sequence diagram:

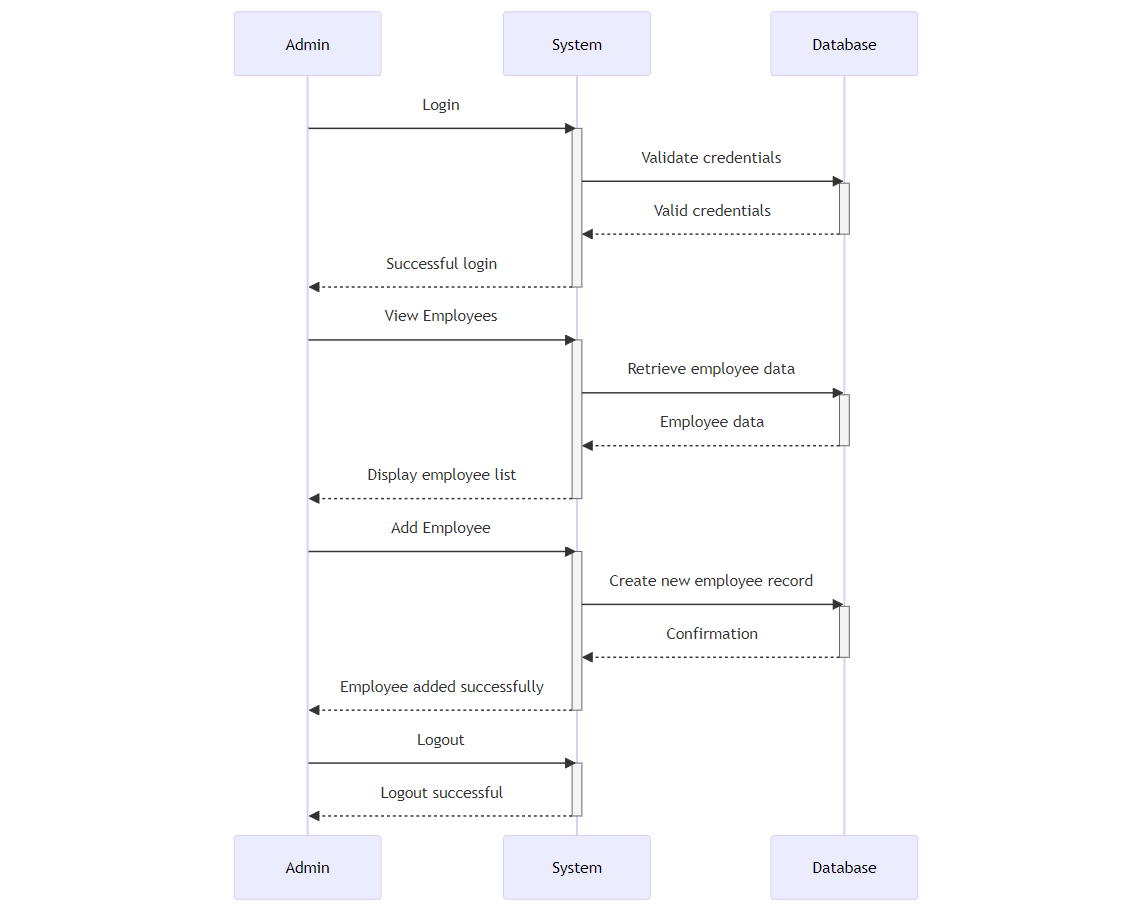
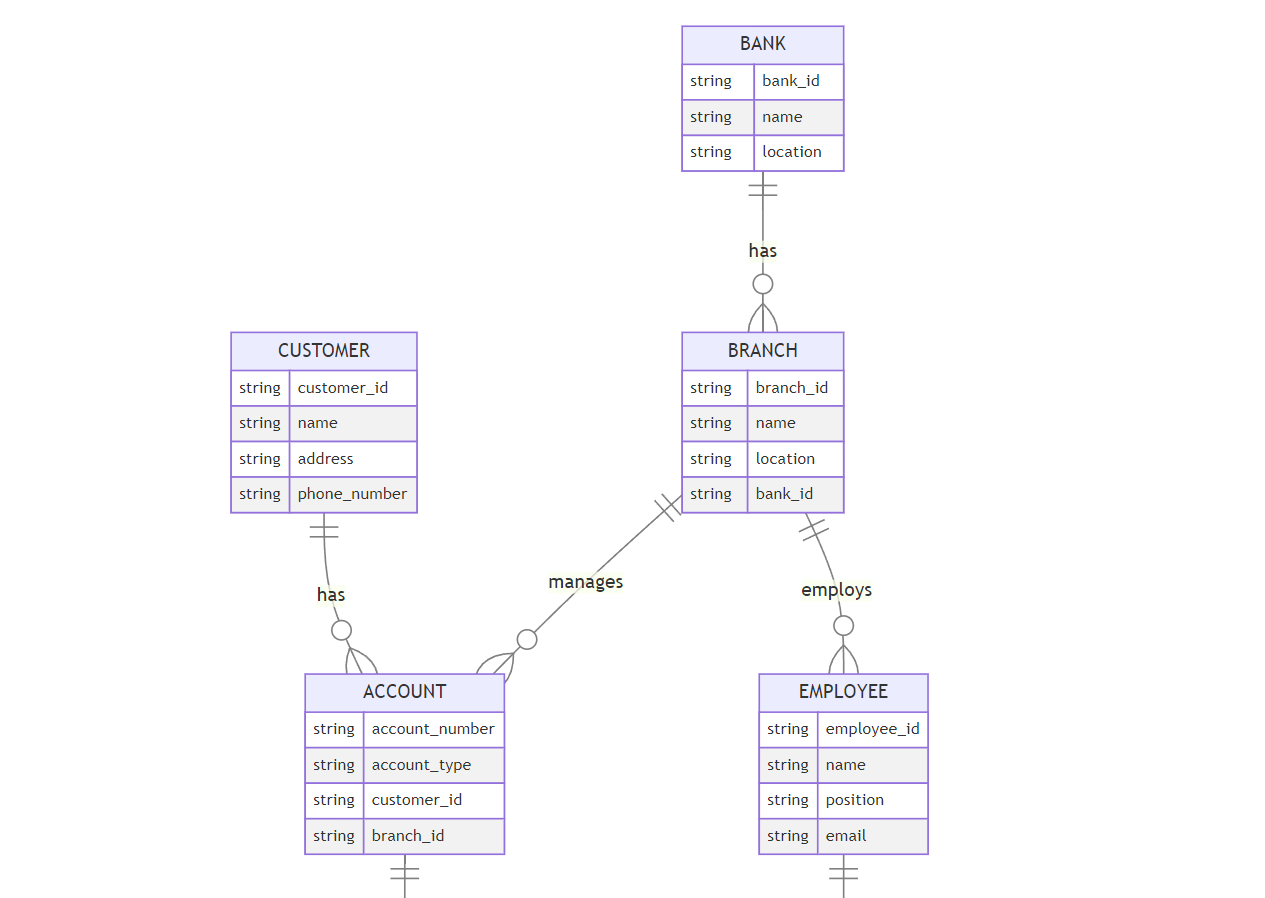


Fig 3: State diagram

## 5.4 Entity-Relationship Diagram:



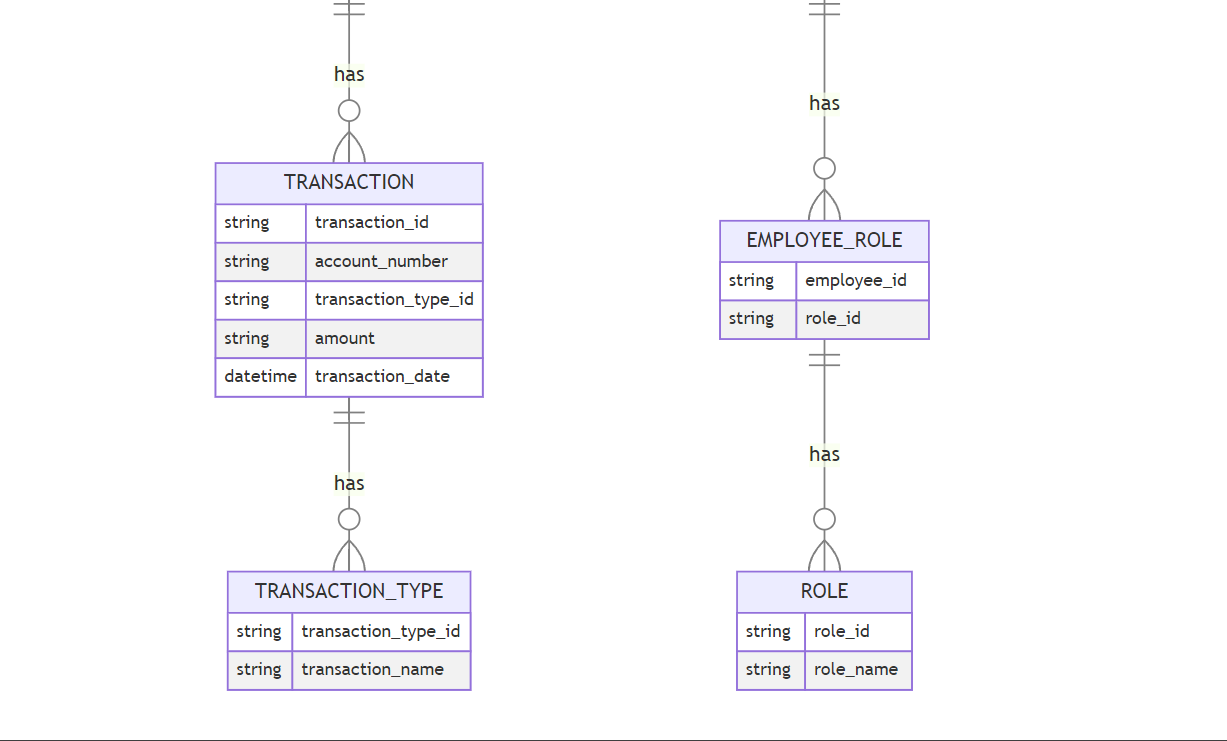


Fig 5: Entity Relationship diagram

## 5.5 Flow Chart

