ASSIGNMENT – 3

NAME : HEMANTH KUMAR.B

REG NO : 192324238

COURSE : DATABASE MANAGEMENT SYSTEM

COURSE CODE : CSA0593

**SCENERIO:**

Create a database for a Financial Management System.

-Design tables for accounts, transactions, loans, and investments.

-Implement a backup strategy to ensure data consistency and minimize downtime in case of system failure.

-Write stored procedures for:Automating backups at specific intervals,Restoring a backup in the event of data loss.

-Simulate crash recovery scenarios using database transaction logs.

**Database Design for Financial Management System**

**1. Accounts Table**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| account\_id | INT | Primary Key, unique identifier for each account |
| account\_type | VARCHAR(50) | Type of account (e.g., savings, checking, loan) |
| account\_balance | DECIMAL(10,2) | Current balance of the account |
| account\_holder\_name | VARCHAR(100) | Name of the account holder |
| account\_creation\_date | DATE | Date the account was created |
|  |  |  |
|  |  |  |

**2. Transactions Table**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| transaction\_id | INT | Primary Key, unique identifier for each transaction |
| account\_id | INT | Foreign key referencing the account ID |
| transaction\_type | VARCHAR(50) | Type of transaction (e.g., deposit, withdrawal, transfer) |
| transaction\_amount | DECIMAL(10,2) | Amount of the transaction |
| transaction\_date | DATE | Date of the transaction |

**3. Loans Table**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| loan\_id | INT | Primary Key, unique identifier for each loan |
| account\_id | INT | Foreign key referencing the account ID |
| loan\_amount | DECIMAL(10,2) | Amount of the loan |
| interest\_rate | DECIMAL(5,2) | Interest rate on the loan |
| start\_date | DATE | Start date of the loan |
| end\_date | DATE | End date of the loan |

**4. Investments Table**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
|  |  |  |
| investment\_id | INT | Primary Key, unique identifier for each investment |
| account\_id | INT | Foreign key referencing the account ID |
| investment\_type | VARCHAR(50) | Type of investment (e.g., stocks, bonds, mutual funds) |
| investment\_amount | DECIMAL(10,2) | Amount invested |
| purchase\_date | DATE | Date of purchase |

**Backup Strategy**

1. **Full Backups:**
   * Create full backups of the entire database at regular intervals (e.g., weekly).
2. **Differential Backups:**
   * Create differential backups that only store the data that has changed since the last full backup.
3. **Incremental Backups:**
   * Create incremental backups that store only the data that has changed since the last full or differential backup.

**Stored Procedures**

**1. Automate Backups:**

SQL

CREATE PROCEDURE AutomateBackup

AS

BEGIN

-- Implement backup logic using appropriate backup tools

-- (e.g., SQL Server Agent, Oracle RMAN)

-- Specify backup type (full, differential, or incremental)

-- and backup destination

END

**2. Restore Backup:**

SQL

CREATE PROCEDURE RestoreBackup

@backup\_file\_path VARCHAR(255)

AS

BEGIN

-- Implement restore logic using appropriate backup tools

-- Specify the backup file to restore from

END

**Crash Recovery**

* **Transaction Logs:**
  + Record all database transactions, including changes made to data.
  + Use transaction logs to roll back incomplete transactions or recover lost data.
* **Checkpoint Mechanism:**
  + Periodically write the database's current state to disk, ensuring that all committed transactions are durable.
* **Recovery Process:**
  + Read the transaction log to identify the last checkpoint.
  + Roll back any uncommitted transactions after the checkpoint.
  + Replay committed transactions to restore the database to a consistent state.

By implementing these strategies, you can ensure the integrity and availability of your financial management system's data.