Assessment for Internship - Solution

Rishi Ram S (rishi2016ram@gmail.com)(9842321577)

4.c) Solution for the System Design Question

Application Design

Give a detailed system design for one of the following
i. How to scale the application for 1M records.
ii. How to structure a database for lease data instead of using CSV

Architecture Overview

The system will follow a Modular Microservices Architecture with the following components:

- **API Gateway:** Handles requests and routes them to the appropriate microservices.
- **Data Parsing Service:** Extracts and validates data from CSV files.
- **Lease Management Service:** Core service for querying lease data and performing calculations.
- **Database Layer:** Stores and indexes lease data.
- **AI Query Service:** Integrates OpenAI's GPT API to handle natural language queries.
- **Report Generation Module:** Produces custom reports based on specific requirements.

System Design :

Scaling the application for 1M records

1. **Database Optimization:**

- Use partitioning based on StartDate or EndDate to distribute data across multiple tables or servers.
- Implement caching with tools like Redis or Memcached for frequently accessed queries (e.g., active leases in a period).
- Use read replicas for high read query volumes.

2. **Backend Design for Scalability**

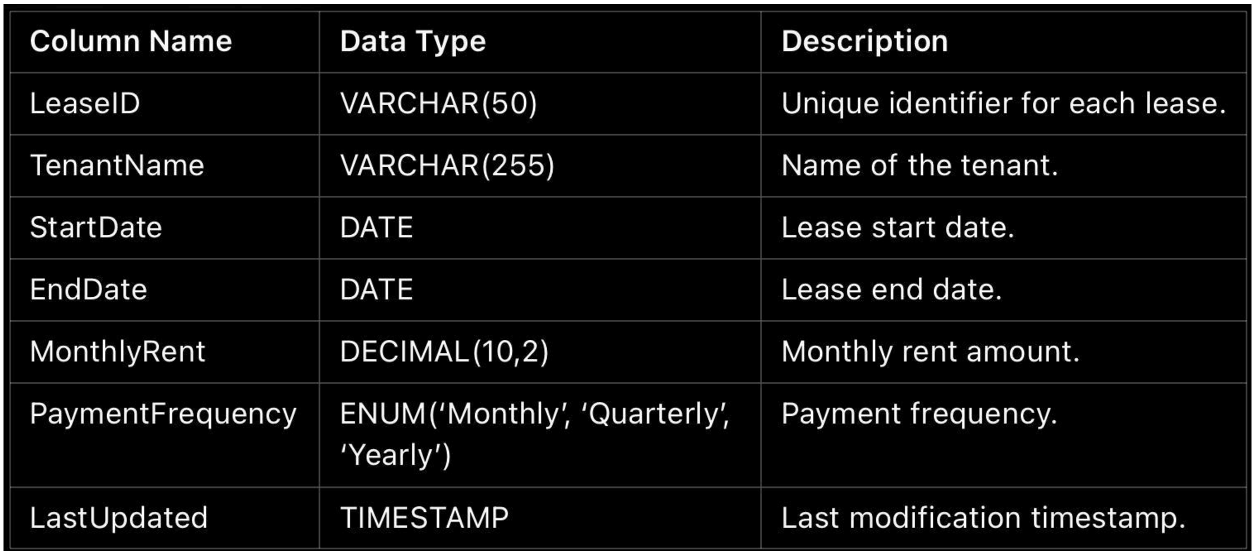
- Use an asynchronous processing model (e.g., RabbitMQ, Kafka) for handling heavy report-generation tasks.
- Employ horizontal scaling for microservices (e.g., deploy multiple instances using Kubernetes or AWS ECS)

Structuring a Database for the lease data instead of using CSV

Database Design

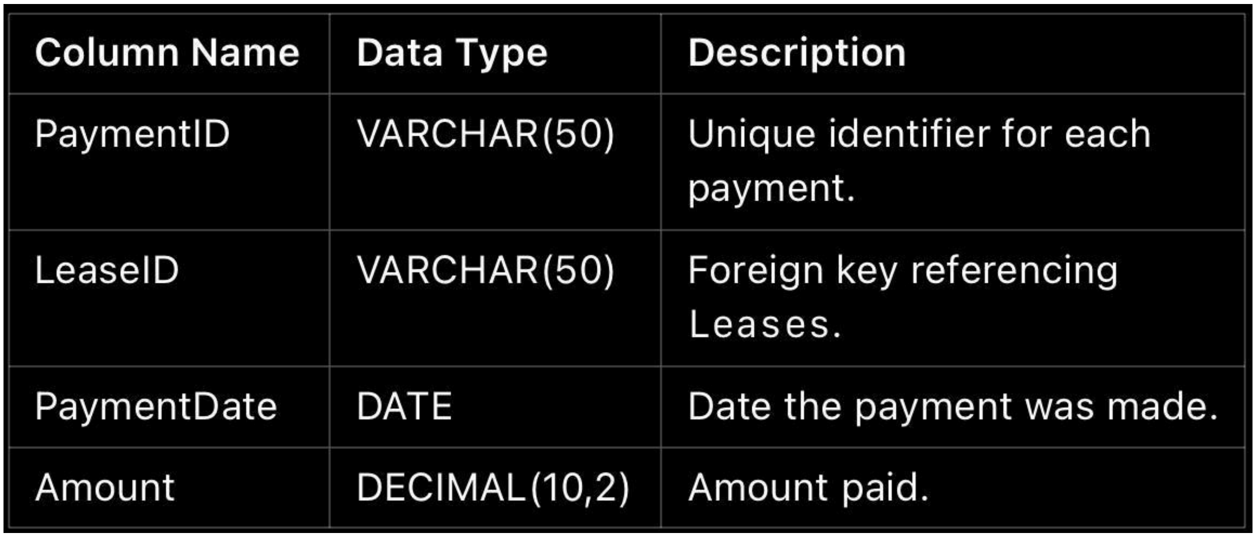
Switching from CSV to a relational database (RDBMS) is ideal for scaling and querying. The schema for lease data would include the following tables:

Table 1 : Leases



Column Name	Data Type	Description
LeaseID	VARCHAR(50)	Unique identifier for each lease.
TenantName	VARCHAR(255)	Name of the tenant.
StartDate	DATE	Lease start date.
EndDate	DATE	Lease end date.
MonthlyRent	DECIMAL(10,2)	Monthly rent amount.
PaymentFrequency	ENUM('Monthly', 'Quarterly', 'Yearly')	Payment frequency.
LastUpdated	TIMESTAMP	Last modification timestamp.

Table 2 : Payments



Column Name	Data Type	Description
PaymentID	VARCHAR(50)	Unique identifier for each payment.
LeaseID	VARCHAR(50)	Foreign key referencing Leases.
PaymentDate	DATE	Date the payment was made.
Amount	DECIMAL(10,2)	Amount paid.

Indexes

- Index on StartDate and EndDate for fast querying of active leases.
- Index on LeaseID for joining tables and efficient lookups.