**Film Rental Store Project**

**ABSTRACT**

The Film Rental Store project is a comprehensive exercise in building a RESTful API system for managing a film rental business. It uses the **Sakila database schema**, a widely recognized sample schema, to simulate real-world operations like customer management, film catalog updates, rentals, payments, and inventory tracking.  
This project emphasizes:

* **Enterprise-level design principles** (JSON-based APIs, HTTP status codes).
* **Security and testing best practices**.
* **Hands-on experience** with Spring Boot, JUnit, and Mockito.

**WHY AM I CREATING THIS PROJECT**

* **Real-world relevance:** Rental businesses are common, and understanding their workflows helps in designing scalable systems.
* **Skill development:** Learn REST API design, database integration, and testing.
* **Industry standards:** Practice global exception handling, proper HTTP status codes, and secure endpoints.
* **Portfolio building:** A complete project that demonstrates technical depth for interviews and assessments.

**WHAT AM I CREATING**

* A **Film Rental Store Management System** with:
  + **Customer operations:** Add, update, search by name, email, phone, country, city.
  + **Film operations:** Update title, rating, category, language, rental rate, and duration.
  + **Rental and payment workflows:** Rent films, update return dates, calculate revenue.
  + **Inventory management:** Track film copies across stores.
* **RESTful APIs** for all operations, following JSON standards.
* Integration with **Sakila schema** (tables: film, customer, rental, payment, inventory, staff, store).

**HOW AM I CREATING**

**Technology Stack**

* **Backend:** Java, Spring Boot.
* **Database:** MySQL with Sakila schema.
* **Testing:** JUnit and Mockito.

**Design Principles**

* All requests and responses in **JSON**.
* Use appropriate **HTTP status codes**:
  + 200 OK – Successful request.
  + 201 Created – New resource added.
  + 400 Bad Request – Invalid input.
  + 404 Not Found – Resource missing.
  + 500 Internal Server Error – Unexpected failure.
* **Global exception handling** for consistent error responses.

**Security**

* Secure **GET and POST** requests for customer data.
* Remove sensitive columns (username/password) from staff table; use a separate security table.

**Enhancements**

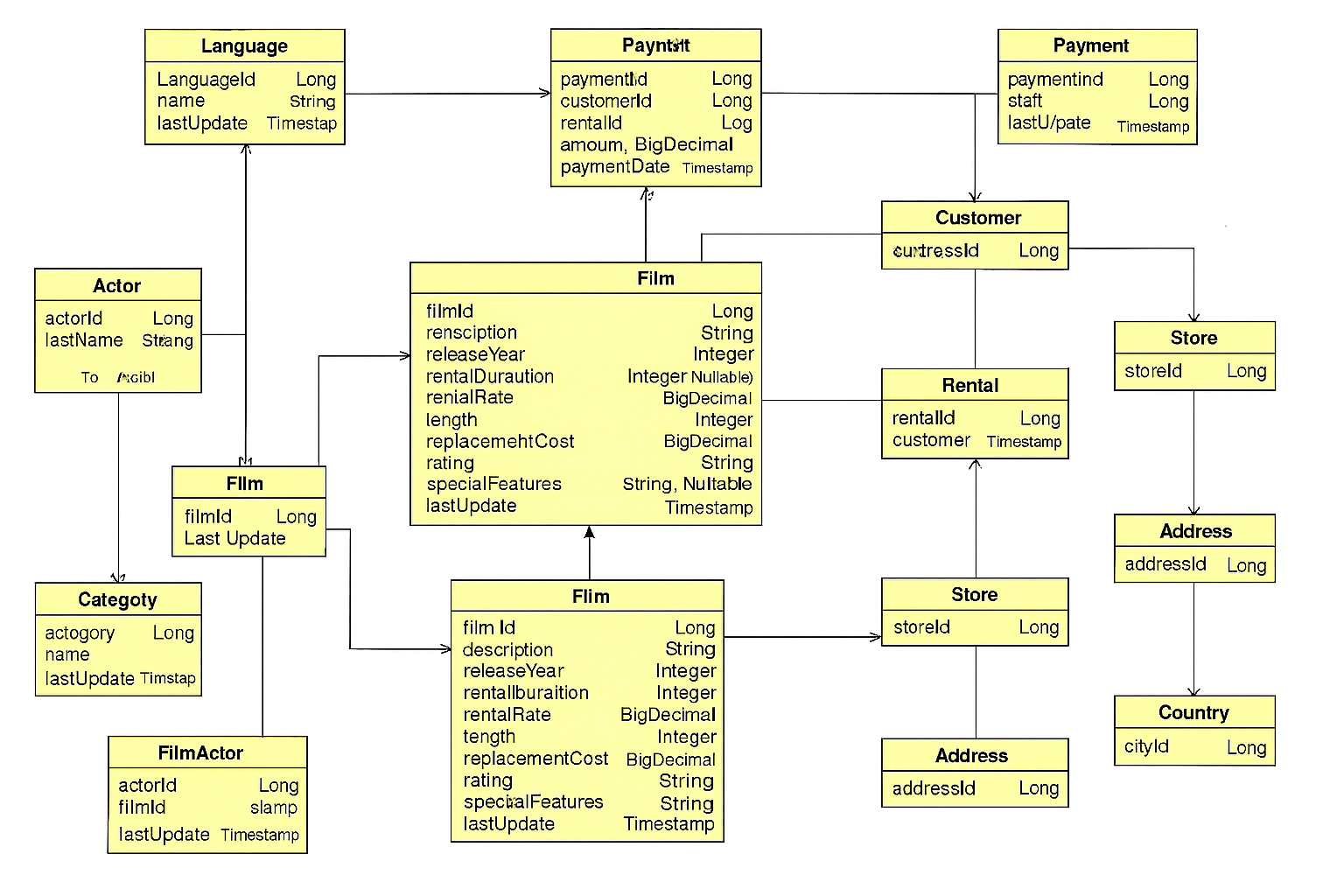
* Convert **Staff picture** column to URL.
* Implement **custom endpoints** like:
  + /api/films/topten – Top 10 most rented films.
  + /api/payment/revenue/datewise – Revenue reports.

**FINAL OUTCOME AS A LEARNER**

**By completing this project, I :**

* **Master End-to-End API Development  
  Understand how to design, implement, and deploy RESTful APIs that interact seamlessly with relational databases.**
* **Gain Expertise in Database Connectivity  
  Learn how to integrate Java applications with MySQL using ORM tools and handle complex queries efficiently.**
* **Apply Best Practices in RESTful Design  
  Implement proper HTTP methods, status codes, and JSON standards for clean and maintainable APIs.**
* **Develop Robust Error Handling Skills  
  Use global exception handling and custom error responses to ensure reliability and user-friendly feedback.**
* **Strengthen Security Knowledge  
  Learn how to secure endpoints, manage authentication, and protect sensitive data in enterprise applications.**
* **Enhance Testing Proficiency  
  Gain confidence in writing unit and integration tests using JUnit and Mockito, ensuring code quality and reliability.**
* **Improve Code Maintainability and Scalability  
  Understand how to structure projects for long-term growth, modularity, and easy enhancements.**
* **Build Reporting and Analytics Skills  
  Implement endpoints for revenue calculation, top rented films, and other business insights.**
* **Experience Real-World Business Logic  
  Work with workflows like rentals, payments, and inventory management that mimic actual business processes.**
* **Boost Problem-Solving and Debugging Skills  
  Learn to troubleshoot API issues, database constraints, and performance bottlenecks effectively.**
* **Create a Portfolio-Ready Project  
  Showcase a complete, production-like application that demonstrates technical depth and architectural thinking.**

**CLASS DIAGRAM**

****

**USECASE DIAGRAM:**

**A diagram of a diagram

AI-generated content may be incorrect.**