Loan Management System
Database Testing Summary

Summary of Database Testing Results

The database testing for the Loan Management System focused on validating key database operations, ensuring data integrity, and assessing performance. A total of **9 test cases** were executed, covering core functionalities such as Loan Management, Client Management, and Payments Management. The DB had a 56% pass rate.

During the testing process, several issues were identified:

1. **Critical Issues:** 1 (duplicate entries caused by a trigger).
2. **High Severity Issues:** 1 (missing index on a heavily queried table).
3. **Other Issues:** 2 low-severity bugs and redundant queries affecting data consistency.

Key observations highlighted good adherence to foreign key constraints and data validation rules.

1. Overview

1. **Project/Module Name:** Loan Management System
2. **Objective:** Validate database operations, integrity, constraints, and performance under various scenarios.

2. Testing Scope

1. **Tested Features**
    - CRUD operations, stored procedures, triggers, foreign key constraints, and performance optimization.
2. **Test Data**
    - The testing involved realistic sample datasets simulating scenarios, including loan records, client and payment records.

3. Test Results

A total of **9 test cases** were executed. The results are summarized below:

Test Case ID	Test Case Description	Expected Outcome	Actual Outcome	Status (PASS/FAI
TC_DB_001	Validate loan creation	Loan record created	Loan record created	PASS
TC_DB_002	Invalid loan creation	Loan record not created	Loan record created	FAIL
TC_DB_003	Valid payment creation	payment record created	payment record created	PASS
TC_DB_004	Invalid payment creation	payment record bot created	payment record created	FAIL
TC_DB_005	Valid client creation	Client record created	Client record created	PASS
TC_DB_006	Invalid client creation	Client record not created	Client record not created	PASS
TC_DB_007	Check foreign key constraint	Rejected invalid data	Data rejected	PASS
TC_DB_008	Test trigger functionality	No duplicate entries	Duplicate entries found	FAIL
TC_DB_009	Verify query performance	Query executes in <1 sec	Query executes in >2 sec	FAIL

4. Issues Identified

The highlights the defects identified and severity

Issue ID	Description	Severity	Steps to Reproduce
DB_001	Trigger causes duplicate entries	Critical	1. Insert data into the loans table. 2. Check for duplicate entries in the table.
DB_002	Redundant Queries	Low	1.Check Performance under high load
DB_003	Missing index on payments table	High	1. Query payments for a specific client ID. 2. Observe slow execution time.
DB_004	Constraint allows invalid data	Low	1. Insert invalid loan status.

5. Key Observations

1. Core database operations performed as expected, with a high pass rate for test cases.
2. Foreign key constraints and data validation rules were adhered to effectively.
3. A critical issue with trigger functionality caused duplicate entries, requiring immediate resolution.

6. Recommendations

1. **Fix Identified Issues**
    - Address the critical trigger issue and add an index to the payments table to improve performance.
2. **Optimize Database Queries**
    - Conduct a detailed analysis of slow redundant queries and optimize their execution plans.
3. **Enhance Testing Coverage**
    - Include additional test cases to cover edge scenarios, scalability testing, and further validate constraints.
4. **Monitor Database Performance**
    - Implement monitoring tools to track query execution times and resource usage in real-time.

7. Conclusion

The testing confirmed the functionality and reliability of most database operations. However, targeted improvements are required to address the identified performance bottlenecks and critical bugs. With these enhancements, the database will be well-prepared for production deployment.