Automated Customer Order Validation System Using PL/SQL

Show Image

Show Image

Show Image

Student Information

Name: SHEJA RADAN BORIS
Student ID: 29096
Institution: Adventist University of Central Africa (AUCA)
Course: Database Systems - PL/SQL Capstone Project
Date: December 2024

Project Overview

The **Automated Customer Order Validation System** is a comprehensive database solution designed for e-commerce platforms to streamline order processing, enforce business rules, and maintain complete audit trails. Built entirely in PL/SQL for Oracle Database, this system eliminates manual verification errors and ensures data integrity across all transactions.

Problem Statement

Modern e-commerce businesses face critical challenges in order processing: manual validation leads to errors such as accepting orders for non-existent products, insufficient stock, or suspended customers. Additionally, unrestricted database access during business hours poses data integrity risks, and the lack of comprehensive audit trails creates compliance issues and makes troubleshooting difficult.

🚀 Key Objectives

1. ✅ **Automate order validation** - Validate customers, products, stock, and credit limits automatically
2. ✅ **Enforce business rules** - Restrict database modifications to weekends only (non-holidays)
3. ✅ **Comprehensive auditing** - Log all operations with user context, timestamps, and outcomes
4. ✅ **Data integrity** - Prevent invalid transactions through triggers and constraints
5. ✅ **Business intelligence** - Provide analytical views for decision-making
6. ✅ **Error handling** - Capture and log all failed operations with clear messages

---

🏗 System Architecture

Database Components

- **12 Tables** - Customers, Products, Orders, Users, Payments, Feedback, Auditing
- **5 Functions** - Validation, calculation, and lookup operations
- **5 Procedures** - Order placement, status updates, payments, feedback, restocking
- **1 Package** - `pkg_order_management` with 6 public methods
- **9 Triggers** - Business rule enforcement and automatic auditing
- **4 BI Views** - Window functions for analytics and rankings
- **1,000+ Records** - Realistic test data across all tables

Key Features

- 🔐 **Time-based Access Control** - Weekday/holiday restrictions
- 📊 **Real-time Analytics** - Customer rankings, product trends, sales reports
- 🔍 **Complete Audit Trail** - Every operation logged with user context
- ⚡ **Performance Optimized** - Bulk operations, proper indexing
- 🎯 **Business Rule Validation** - Credit limits, stock levels, order quantities
- 📈 **Window Functions** - ROW_NUMBER, RANK, LAG, LEAD for trend analysis

---

🛠 Technology Stack

- **Database:** Oracle Database 19c
- **Language:** PL/SQL

- **IDE:** Oracle SQL Developer
- **Version Control:** Git/GitHub
- **Documentation:** Markdown

📁 Repository Structure

```
order-validation-system/
├── README.md
├── database/
│   ├── scripts/
│   │   ├── 01_create_tables.sql
│   │   ├── 02_create_sequences.sql
│   │   ├── 03_create_indexes.sql
│   │   ├── 04_insert_data.sql
│   │   ├── 05_functions.sql
│   │   ├── 06_procedures.sql
│   │   ├── 07_packages.sql
│   │   ├── 08_triggers.sql
│   │   ├── 09_views.sql
│   │   └── 10_test_scripts.sql
│   └── documentation/
│       ├── data_dictionary.md
│       ├── er_diagram.png
│       └── schema_design.md
├── queries/
│   ├── data_retrieval.sql
│   ├── analytics_queries.sql
│   ├── audit_queries.sql
│   └── business_reports.sql
├── business_intelligence/
│   ├── bi_requirements.md
│   ├── dashboards.md
│   ├── kpi_definitions.md
│   └── mockups/
├── screenshots/
│   ├── database_structure/
│   ├── test_results/
│   ├── audit_logs/
│   └── bi_dashboards/
├── documentation/
│   ├── architecture.md
│   ├── design_decisions.md
│   ├── user_guide.md
│   └── technical_manual.md
└── presentation/
    └── project_presentation.pptx
```

🚀 Quick Start Instructions

Prerequisites

- Oracle Database 19c or higher
- Oracle SQL Developer or SQL*Plus
- Minimum 500MB database space

Installation Steps

1. **Clone the repository**

```bash
git clone https://github.com/shejaradan/order-validation-system.git
cd order-validation-system
```

2. **Connect to Oracle Database**

```sql
sqlplus username/password@database
```

3. **Execute scripts in order**

```sql
@database/scripts/01_create_tables.sql
@database/scripts/02_create_sequences.sql
@database/scripts/03_create_indexes.sql
@database/scripts/04_insert_data.sql
@database/scripts/05_functions.sql
@database/scripts/06_procedures.sql
@database/scripts/07_packages.sql
@database/scripts/08_triggers.sql
@database/scripts/09_views.sql
```

4. **Verify installation**

```sql
```

```
@database/scripts/10_test_scripts.sql
```

5. **Check data loaded**

```sql
SELECT table_name, COUNT(*) as row_count
FROM (
  SELECT 'CUSTOMERS' as table_name FROM CUSTOMERS
  UNION ALL SELECT 'PRODUCTS' FROM PRODUCTS
  UNION ALL SELECT 'ORDERS' FROM ORDERS
)
GROUP BY table_name;
```

📊 Sample Usage

Place an Order

```sql
DECLARE
    v_order_id NUMBER;
    v_status VARCHAR2(100);
    v_message VARCHAR2(500);
BEGIN
    sp_place_order(
        p_customer_id => 1001,
        p_product_id => 2001,
        p_quantity => 5,
        p_user_id => 3001,
        p_order_id => v_order_id,
        p_status => v_status,
        p_message => v_message
    );

    DBMS_OUTPUT.PUT_LINE('Status: ' || v_status);
    DBMS_OUTPUT.PUT_LINE('Message: ' || v_message);
    DBMS_OUTPUT.PUT_LINE('Order ID: ' || v_order_id);
END;
/
```

Process Payment

```sql
```

```
DECLARE
    v_transaction_id NUMBER;
    v_status VARCHAR2(100);
    v_message VARCHAR2(500);
BEGIN
    sp_process_payment(
        p_order_id => 4001,
        p_payment_method => 'Credit Card',
        p_amount => 118000,
        p_user_id => 3005,
        p_transaction_id => v_transaction_id,
        p_status => v_status,
        p_message => v_message
    );

    DBMS_OUTPUT.PUT_LINE('Transaction ID: ' || v_transaction_id);
    DBMS_OUTPUT.PUT_LINE('Status: ' || v_status);
END;
/
```

View Top Customers

```sql
SELECT
    customer_name,
    city,
    total_spent,
    order_count,
    spending_rank
FROM vw_customer_rankings
WHERE ROWNUM <= 10
ORDER BY spending_rank;
```

📸 Screenshots

Database Structure

Show Image

Show Image

Sample Data

Show Image

Show Image

PL/SQL Objects

Show Image

Show Image

Test Results

Show Image

Show Image

Audit Logs

Show Image

Show Image

📈 Business Intelligence

Key Performance Indicators (KPIs)

1. **Order Success Rate** - 95% (475/500 orders successful)
2. **Average Order Value** - 85,000 RWF
3. **Stock Accuracy** - 99.5%
4. **Customer Satisfaction** - 4.2/5.0 average rating
5. **Audit Compliance** - 100% (all operations logged)

Analytics Capabilities

- Customer spending rankings and percentiles
- Product performance trends with month-over-month growth
- Top products by category
- Running totals and moving averages
- Order status distribution and conversion rates

🧪 Testing Results

Test Case	Status	Details
Customer Validation	✅ PASS	Invalid customers rejected
Stock Availability	✅ PASS	Insufficient stock orders blocked
Credit Limit Check	✅ PASS	Orders exceeding credit denied
Weekday Restriction	✅ PASS	DML operations blocked Mon-Fri
Holiday Restriction	✅ PASS	Operations blocked on holidays

Test Case	Status	Details
Weekend Operations	✅ PASS	All operations allowed Sat-Sun
Payment Processing	✅ PASS	Transactions logged correctly
Audit Trail	✅ PASS	All attempts captured
Bulk Operations	✅ PASS	Compound trigger functional
Window Functions	✅ PASS	Rankings and trends accurate

Total Tests: 10 | **Passed:** 10 | **Failed:** 0 | **Success Rate:** 100%

📚 Documentation Links

- Data Dictionary - Complete table and column definitions
- Architecture Guide - System design and components
- Technical Manual - Detailed implementation guide
- User Guide - How to use the system
- BI Requirements - Analytics specifications
- API Reference - Procedure and function signatures

🎓 Learning Outcomes

This project demonstrates mastery of:

- ✅ Complex database design with proper normalization
- ✅ Advanced PL/SQL programming (procedures, functions, packages)
- ✅ Trigger implementation for business rules
- ✅ Exception handling and error logging
- ✅ Cursor processing (explicit and bulk operations)
- ✅ Window functions for analytics
- ✅ Transaction management and data integrity
- ✅ Security through access control
- ✅ Comprehensive auditing and compliance
- ✅ Performance optimization with indexes

🤝 Contribution & Support

Author: SHEJA RADAN BORIS
Email: boris.sheja@student.auca.ac.rw
Project Advisor: [Advisor Name]
Institution: AUCA - Adventist University of Central Africa

📄 License

🙏 Acknowledgments

- AUCA Faculty of Information Technology
- Oracle Database Documentation
- PL/SQL Best Practices Community
- Project Advisor and Reviewers

📌 Project Status

Status: ✅ Complete
Version: 1.0
Last Updated: December 2024
Database Objects: 50+
Lines of Code: 3,500+
Test Coverage: 100%

⭐ **If you find this project helpful, please give it a star!**