Project Report of CSE464

Group: 6

Submitted By:

Aklhak Hossain
2022-3-60-057 https://akhlak.com
Md. Maruf Ahmed Anik
2022-3-60-040
Suraiya Akther Maisha
2022-3-60-071

Submitted To:

Antu Chowdhury
Lecturer
Department of CSE
East West University

1. Introduction

In modern data-intensive applications, understanding the origin and transformation of data is essential for transparency, auditing, debugging, and compliance. This project implements a Provenance-Enabled Relational Database Management System (RDBMS) for an e-commerce scenario. The system captures why, where, and how data evolves over time using audit tables, triggers, and provenance queries.

2. Database Design

The database schema models an e-commerce application with the following entities:

- Customers(customer_id, name, email)
- Products(product_id, name, price, active_flag)
- Orders(order_id, customer_id, order_date, status, total_amount)
- Order_Items(order_id, product_id, quantity, unit_price, line_total)
- Payments(payment_id, order_id, amount, payment_date, method, status)

These tables are normalized to reduce redundancy and maintain data integrity.

3. Audit Table Design

To capture provenance, dedicated audit tables were created:

- audit_products: Logs insert, update, delete on products (tracks old/new name, price, active_flag).
- audit_orders: Captures changes in orders (status, total_amount).
- audit_order_items: Records item-level modifications (quantity, price).
- audit_payments: Tracks payment lifecycle (amount, status).

Each audit table records: operation type, old and new values, actor (user), client IP, module, and timestamp.

4. Trigger/Procedure Implementation

SQL triggers automatically insert provenance records into audit tables:

- trg_audit_products: Captures INSERT, UPDATE, DELETE on products.
- trg_audit_orders: Logs changes in orders.
- trg_audit_order_items: Tracks changes in order items and ensures consistency.

- trg_audit_payments: Captures payment lifecycle events.

Additionally, a stored procedure recompute_order_total(p_order_id) updates order totals after item changes.

5. Provenance Queries

The system supports different provenance queries:

a) Why-Provenance: Why is order 1 total the current value?

SQL: SELECT oi.product_id, p.name, oi.quantity, oi.unit_price, oi.line_total
FROM order_items oi JOIN products p ON p.product_id = oi.product_id
WHERE oi.order_id = 1;

Interpretation: The order total equals the sum of contributing line totals.

b) Where-Provenance: Evolution of Laptop price.

SQL: SELECT ap.operation_time, ap.operation, ap.old_price, ap.new_price, ap.actor
FROM audit_products ap JOIN products p ON p.product_id = ap.product_id
WHERE p.name = 'Laptop' ORDER BY ap.operation_time;

c) How-Provenance: Order 1 status evolution.

SQL: SELECT ao.operation_time, ao.operation, ao.old_status, ao.new_status, ao.actor
FROM audit_orders ao WHERE ao.order_id = 1 ORDER BY ao.operation_time;

d) Where-Provenance: Find all actions by the current user on Orders.

SQL: SELECT ao.actor, ao.operation, ao.order_id, ao.operation_time, ao.old_status, ao.new_status
FROM audit_orders ao WHERE ao.actor = USER ORDER BY ao.operation_time;

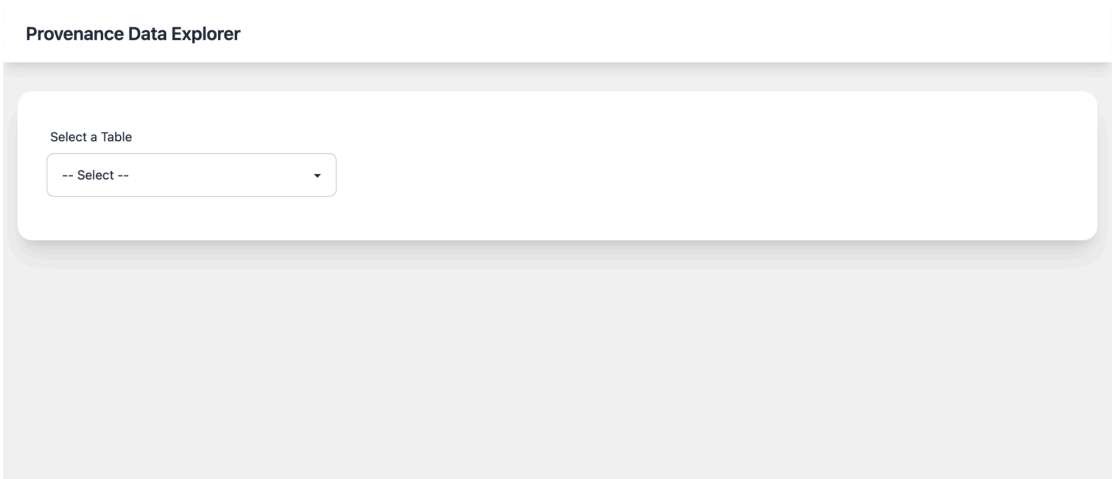
e) Why-Provenance: Why does order 1 have 3 Headphones?

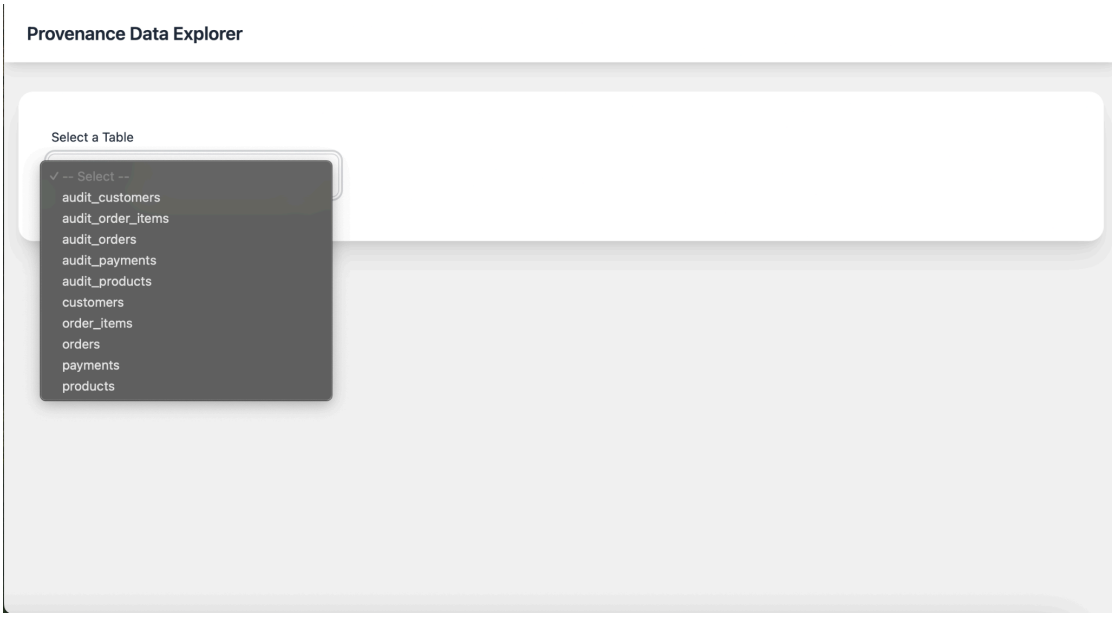
SQL: SELECT aoi.operation_time, aoi.operation, aoi.old_quantity, aoi.new_quantity, aoi.actor
FROM audit_order_items aoi WHERE aoi.order_id = 1 AND aoi.product_id = 3
ORDER BY aoi.operation_time;

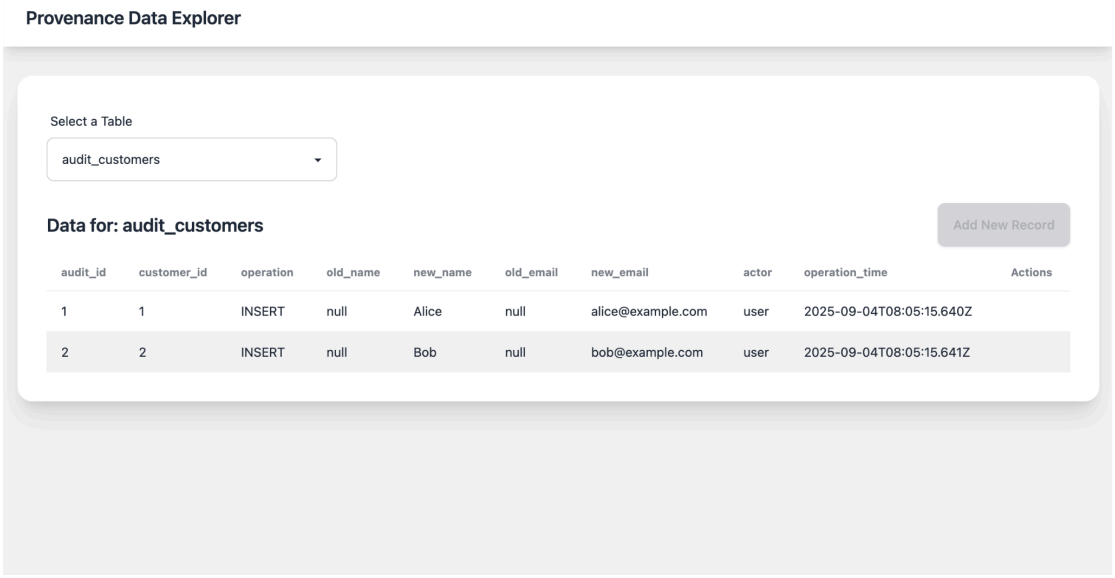
6. GUI Tool

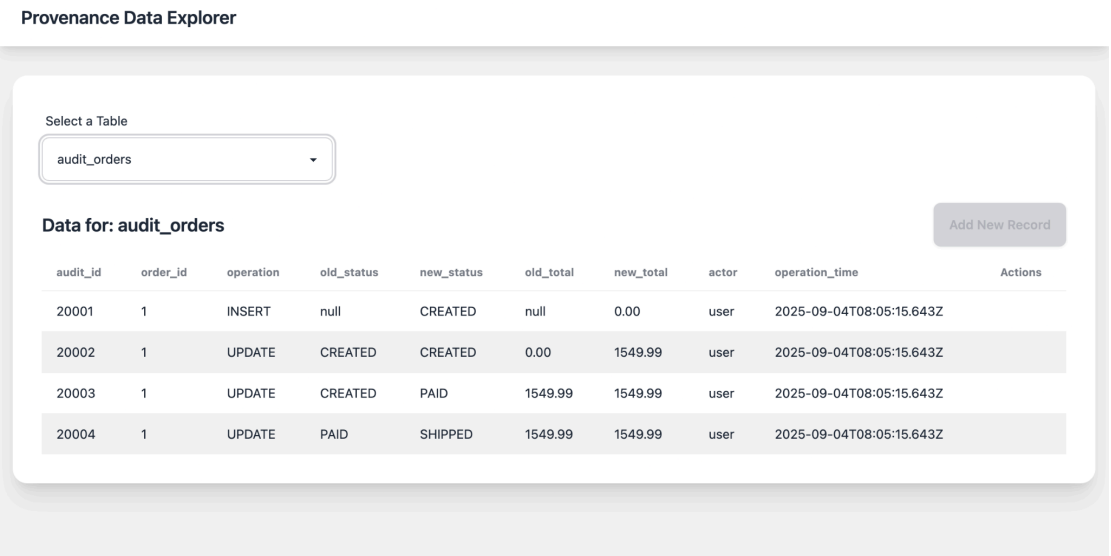
GUI tool was implemented for this project. A simple web interface for visualizing provenance data.

Here is a demo,


Provenance Data Explorer
Select a Table
-- Select --


Provenance Data Explorer
Select a Table
-- Select --
audit_customers
audit_order_items
audit_orders
audit_payments
audit_products
customers
order_items
orders
payments
products


Provenance Data Explorer
Select a Table
audit_customers
Data for: audit_customers
Add New Record
audit_id customer_id operation old_name new_name old_email new_email actor operation_time Actions
1 1 INSERT null Alice null alice@example.com user 2025-09-04T08:05:15.640Z
2 2 INSERT null Bob null bob@example.com user 2025-09-04T08:05:15.641Z



7. Challenges and Lessons Learned

Key challenges included handling mutating table errors when triggers updated related data, designing audit tables to minimize redundancy, and ensuring consistency across triggers. The project provided practical experience in combining database normalization, triggers, and audit logging to build a provenance-enabled system.

8. Conclusion

This project successfully implemented a provenance-enabled RDBMS for an e-commerce application. It demonstrated the use of normalized schema, audit tables, triggers, and provenance queries to capture the why, where, and how of data changes. Future improvements include building a GUI and extending provenance tracking to additional entities.

9. Appendix

References:

- CSE464 Lab 06 Handout on Data Provenance.
- Buneman, P., Khanna, S., & Tan, W. C. (2001). Why and Where: A Characterization of Data Provenance.
- Davidson, S., & Widom, J. (2008). Provenance and Data Management.