SCS1309 Database Management

Lab Sheet - MySQL 06

GLTN is a multinational logistics and trade facilitation network that connects manufacturers, suppliers, and logistics providers to streamline global trade and shipping operations. The platform maintains a database of shipments, transactions, customs clearances, and partner organizations.

Database Schema Overview

1. Manufactures

ManufacturerID	Name	Country	Industry
1	Electronic Engineering Services	Sri Lanka	Electronics
2	Green Fuel Automotives	Germany	Automotive

2. Suppliers

SupplierID	Name	Country	Specialization
101	Tech Electronic	China	Circuits Supply
102	Eco Plastics	India	Plastic Supply

3. Shipments

ShipID	OriginCountry	Destination Country	Status	ManufacturerID
201	Sri Lanka	China	In Transit	1
202	India	Germany	Delivered	2

4. Transaction

TransactionID	SupplierID	ManufacturerID	Amount	Currency	TransactionDate
301	101	1	5000000	RS	2025-01-05
302	102	2	350000	EUR	2025-01-15

5. CustomesClearance

ClearanceID	ShipmentID	Status	ClearanceDate
401	201	Approved	2025-01-10
402	202	Pending	NULL
403	201	Approved	2025-01-22
404	202	Approved	2025-02-03

1. GLTN wants to track shipments in customs clearance by beyond 10 days.
    - Write a SQL query to list shipments that have been pending clearance for more than 10 days.
    - Ensure that transactions related to those shipments are flagged for review.
2. GLTN's financial team wants to monitor transactions exceeding $40,000 and implement transaction control measures.
    - Write a query to retrieve transaction details where the amount exceeds $400,000.
    - Create a Transaction Review view that lists these transactions for monitoring.
3. To ensure secure data access, GLTN follows strict DCL (Data Control Language) policies for granting and revoking privileges.
    - Grant SELECT privileges on the Shipments table to a user role LogisticsManager.
    - Revoke DELETE privileges from the FinanceTeam role on the Transactions table.
4. To speed up financial analysis queries, indexing is required on high-frequency columns in the Transactions table.
    - Create an index on the TransactionDate and Amount columns.
    - Write a query to retrieve transactions for a given date range efficiently.
5. To maintain Atomicity, Consistency, Isolation, and Durability (ACID) in transactions, GLTN needs rollback mechanisms for failed transfers.
    - Implement a transaction block to transfer funds from Manufacturer 1 to Supplier 101.
    - Ensure that, in case of failure, no partial transactions occur.
6. GLTN's logistics team wants to retrieve all shipments pending clearance for a specific

manufacturer.
    - Write a query that retrieves all shipments pending clearance for ManufacturerID = 1.
    - Optimize performance by creating an index on ShipmentID in the CustomsClearance table.
7. GLTN wants to monitor changes in transaction amounts to detect unauthorized updates.
    - Implement a trigger to log changes in the Transactions table whenever the amount is updated.
8. GLTN wants to analyze supplier diversity.
    - Write a query to list suppliers that supply to more than one manufacturer.
9. GLTN wants to flag manufacturers involved in frequent transactions.
    - Write a query to list manufacturers that have engaged in more than 5 transactions.
10. GLTN wants a dynamic view that displays all transactions made in the last 60 days.
    - Write a SQL statement to create a dynamic view for recent transactions.
11. Implement Role-Based Access Control (RBAC) for Secure Access
    - Define roles for different user groups (e.g., LogisticsManager, FinanceTeam, Admin) and assign appropriate privileges.
    - Write SQL statements to create these roles and grant/revoke privileges accordingly.
12. Enforce Ownership-Based Data Access
    - Ensure that each manufacturer can access only their own transaction records.
    - Implement row-level security using views or row-based permissions in SQL.
13. Audit Changes in Privileges
    - Implement an audit log to track changes to user privileges.
    - Write a trigger to log privilege modifications in an audit table whenever a GRANT or REVOKE operation occurs.
14. Restrict Unauthorized Data Modification
    - Ensure that only finance team members can update transaction amounts.
    - Implement a trigger to prevent unauthorized updates and log any attempt.
15. Grant Restricted Access for External Auditors
    - Create a database role ExternalAuditor with read-only access to the Transactions and Shipments tables.
    - Ensure that auditors cannot access sensitive financial data, such as transaction amounts.
16. Prevent Unauthorized Manufacturer Data Deletion
    - Write a trigger to prevent manufacturers from being deleted if they have active transactions or shipments.
    - Ensure an appropriate error message is displayed upon violation.
17. Implement a Fine-Grained Privilege System
    - Set up different privilege levels for finance staff (e.g., junior staff can only view transactions, while senior staff can modify them).
    - Implement a stored procedure to enforce these privilege levels dynamically.
18. Monitor Privilege Escalation Attempts
    - Implement a trigger to detect unauthorized privilege escalation (e.g., a non-admin user attempting to grant themselves admin rights).
    - Log such incidents in a security audit table.