

Soappy

Inventory and Alert Management System for Soap Inventory

There is a handmade soap shop for tracking other features. However, today I will focus only on inventory tracking.

Introduction Efficient inventory management is critical for maintaining business operations, particularly in retail or manufacturing. This report discusses a database solution for managing soap inventory and automating low inventory alerts. Two key tables, `soap` and `low_inventory_alert`, are used to monitor stock levels and flag critical shortages.

Structure and Purpose of the Tables

1. **Soap Table:** The `soap` table stores information about available inventory for different soap products. It has the following columns:
 - `soap_id`: A unique identifier for each soap product (primary key).
 - `name`: The name of the soap product.
 - `inventory`: An integer representing the current stock level of each product.
2. This table provides a centralized system for tracking and updating soap inventory. The `soap_id` serves as a key reference for related tables.
3. **Low Inventory Alerts Table:** The `low_inventory_alert` table records alerts generated when soap inventory falls below a defined threshold (e.g., less than 10 units). Its structure includes:
 - `alert_id`: A unique identifier for each alert (primary key).
 - `soap_id`: A foreign key referencing `soap_id` in the `soap` table.
 - `alert_date`: A timestamp indicating when the alert was generated.
4. The foreign key constraint ensures data integrity between these tables, linking alerts to their corresponding soap products.

Trigger for Low Inventory Alerts To automate alerts when inventory falls below the threshold, a database trigger named `low_inventory_alert` was implemented. This trigger monitors updates to the `soap` table. If an update causes the inventory to drop below 10, a record is inserted into the `low_inventory_alert` table with the corresponding `soap_id` and the current timestamp. This automation minimizes manual oversight and ensures timely notification of low stock levels.

The trigger code is:

```
DELIMITER $$
```

```
CREATE TRIGGER low_inventory_alert
AFTER UPDATE ON soap
FOR EACH ROW
BEGIN
    IF NEW.inventory < 10 THEN
        INSERT INTO low_inventory_alert (soap_id, alert_date)
        VALUES (NEW.soap_id, NOW());
    END IF;
END$$
```

```
DELIMITER ;
```

Preventing Redundant Alerts To avoid duplicate alerts for the same soap product, an enhanced trigger checks the `low_inventory_alert` table to ensure no prior alert exists for the same `soap_id` before inserting a new one. This approach reduces redundant records and streamlines alert management.

Testing and Validation The system was tested by updating the inventory in the `soap` table and verifying the trigger's behaviour. For example:

```
UPDATE soap SET inventory = 5 WHERE soap_id = 1;
```

This query successfully generated an alert in the `low_inventory_alerts` table. Subsequent tests confirmed that alerts are only created when inventory falls below the threshold, and the foreign key constraint ensures data integrity.

Conclusion The inventory and alert management system effectively monitors stock levels and automates low inventory notifications. Utilizing database triggers and foreign key constraints ensures data consistency and timely alerts, which are critical for operational efficiency. Future improvements may include user notifications or integration with external systems to optimize inventory management further.