**Class Task:**

Try to create a similar example where:

* You add a new product to a product table.
* Add an inventory entry and set a savepoint.
* Deduct from inventory on order placement.
* Rollback if the inventory quantity goes negative, ensuring inventory consistency.

**Tasks:**

Task 1:

Create a new table named book\_inventory with columns for book\_id, book\_name, quantity, and price.

Insert three different book records with initial quantities.

Without committing the transaction, reduce the quantity of one book and create a savepoint named

quantity\_update.

Task 2:

In the staff table, add a new staff member with an initial salary.

Increase their salary by 12% and create a savepoint named salary\_boost.

Further increase the salary by 8%. Roll back the transaction to the salary\_boost savepoint to undo the

second increase.

Task 3:

Use the vendors and supplies tables.

Insert a new vendor into the vendors table.

Then, insert a supply record for the vendor in the supplies table.

Use transaction control to ensure that both the vendor and supply records are inserted only if both

statements succeed; otherwise, roll back the changes.

Task 4:

Enable AUTOCOMMIT mode in your SQL environment.

Insert a row in the payments table with payment\_id, vendor\_id, and amount.

After the insertion, verify if the row has been committed automatically.

Disable AUTOCOMMIT afterward.

Task 5:

Using the account\_transactions table, simulate a transaction where multiple withdrawals and deposits are

made on an account.

Set multiple savepoints after each withdrawal or deposit operation.

Rollback to a specific savepoint to undo one of the deposits.

**The End 😊**