SQL Transactions

Task 1: Transactional Batch Operation

Create a table called ORDERS with the following structure:

```
CREATE TABLE ORDERS (
   ORDER_ID INT PRIMARY KEY,
   CUSTOMER_NAME VARCHAR(50),
   PRODUCT_NAME VARCHAR(50),
   QUANTITY INT,
   PRICE DECIMAL(10,2)
);
```

Insert the following records within a transaction:

(101, 'John', 'Mouse', 2, 500.00)
(102, 'Emily', 'Keyboard', 1, 1500.00)
(103, 'Raj', 'Monitor', 1, 7500.00)
(104, 'Ali', 'Laptop', 1, 55000.00)

After insertion, use COMMIT to save the transaction.

Verify with: SELECT * FROM ORDERS;

Task 2: Mistaken Deletion and Recovery

Start a transaction and delete orders where PRICE > 5000. However, suppose this was a mistake. Use ROLLBACK to undo the deletion.

Display the table again and verify the records are intact.

Task 3: Using Savepoints in Complex Updates

You are asked to reduce the price of all products as part of a clearance sale:

1. Start a transaction.
2. Create a SAVEPOINT before each update:
   - Reduce price of all items by 10%.

- Further reduce price of products with QUANTITY = 1 by an additional 5%.

3. After executing all updates, realize the second reduction was too much.
4. Use ROLLBACK TO SAVEPOINT to undo only the second change.
5. Use COMMIT to finalize the first update.

Show the final result of the ORDERS table.

Task 4: Read-Only Transaction Check

Try the following using SET TRANSACTION READ ONLY:

1. Start a read-only transaction.
2. Attempt to DELETE or UPDATE a record from ORDERS.
3. Observe and explain the result.
4. End the transaction.