Assignment 1: Write a SELECT query to retrieve all columns from a 'customers' table, and modify it to return only the customer name and email address for customers in a specific city.

```
CREATE TABLE Customer (
    customer_id NUMBER PRIMARY KEY,
    cname VARCHAR2(100),
    email VARCHAR2(100),
    phone VARCHAR2(20),
    city VARCHAR2(100),
    country VARCHAR2(100)
);
CREATE TABLE Order (
    order_id NUMBER PRIMARY KEY,
    customer_id NUMBER,
    order_date DATE,
    order_amount NUMBER,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);
```

Retrieve all columns from a 'customers' table

```
SELECT * FROM customers;
```

Modify it return only the customer name and email address for customers in a specific city

```
SELECT customer_name, email_address
FROM customers
WHERE city = 'New York';
```

Assignment 2: Craft a query using an INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region, and a LEFT JOIN to display all customers including those without orders.

INNER JOIN:

```
SELECT c.customer_id, c.cname, o.order_id, o.order_date
FROM customer c
INNER JOIN order o ON c.customer_id = o.customer_id
WHERE c.region = 'specified_region';
```

LEFT JOIN:

```
SELECT c.customer_id, c.customer_name, o.order_id, o.order_date
FROM customer c
LEFT JOIN order o ON c.customer_id = o.customer_id;
```

Assignment 3: Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

```
SELECT DISTINCT c.customer_id, c.customer_name
FROM customer c
JOIN order o ON c.customer_id = o.customer_id
WHERE o.order_amount > (
    SELECT AVG(order_amount) FROM order
)
UNION
SELECT customer_id, customer_name
FROM customer
```

```
WHERE city = 'New York';
```

Assignment 4: Compose SQL statements to BEGIN a transaction, INSERT a new record into the 'orders' table, COMMIT the transaction, then UPDATE the 'products' table, and ROLLBACK the transaction.

```
-- Insert a new record into the 'order' table

INSERT INTO order (order_id, customer_id, order_date, order_amount)
VALUES (101, 1001, SYSDATE, 500);

-- Commit the transaction
COMMIT;

-- Update the 'products' table

UPDATE products
SET stock_quantity = stock_quantity - 10
WHERE product_id = 201;

-- Rollback the transaction
ROLLBACK;
```

Assignment 5: Begin a transaction, perform a series of INSERTs into 'orders', setting a SAVEPOINT after each, rollback to the second SAVEPOINT, and COMMIT the overall transaction.

```
-- Perform the first INSERT and set the first SAVEPOINT
INSERT INTO order VALUES (1, 101, SYSDATE, 'NEW');
SAVEPOINT savepoint1;

-- Perform the second INSERT and set the second SAVEPOINT
INSERT INTO order VALUES (2, 102, SYSDATE, 'NEW');
SAVEPOINT savepoint2;

-- Perform the third INSERT and set the third SAVEPOINT
INSERT INTO order VALUES (3, 103, SYSDATE, 'NEW');
SAVEPOINT savepoint3;

-- Perform the fourth INSERT and set the fourth SAVEPOINT
INSERT INTO order VALUES (4, 104, SYSDATE, 'NEW');
SAVEPOINT savepoint4;

-- Rollback to the second SAVEPOINT
ROLLBACK TO SAVEPOINT savepoint2;

-- Commit the overall transaction
COMMIT;
```

Assignment 6: Draft a brief report on the use of transaction logs for data recovery and create a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Introduction

Transaction logs are crucial components of database management systems (DBMS). They record all the changes made to the database, ensuring data integrity and aiding in data recovery in the event of system failures. This report outlines the function of transaction logs in data recovery and presents a hypothetical scenario to illustrate their importance.

Function of Transaction Logs

Transaction logs serve several key purposes:

1. Recording Transactions: Every change in the database, including insertions, deletions, and updates, is logged with details such as the transaction ID, timestamp, and the nature of the change.

2. Ensuring Atomicity and Durability: By recording each transaction, the log ensures that transactions are either fully completed or fully rolled back, maintaining the ACID (Atomicity, Consistency, Isolation, Durability) properties.

3. Supporting Recovery Processes: In case of an unexpected shutdown or crash, transaction logs allow the DBMS to restore the database to its last consistent state by replaying or rolling back transactions.

Hypothetical Scenario

Imagine a retail company, XShop, which relies heavily on its online database for managing inventory and processing sales transactions. One day, XShop's database server experiences an unexpected shutdown due to a power failure. At the time of the shutdown, several transactions were in progress, including inventory updates and sales records.

Role of Transaction Logs in Data Recovery

1. Initial Assessment: Once the server is back online, the database administrator (DBA) reviews the transaction log. The log shows all transactions that were in progress at the time of the crash, including their start times and the operations performed.

2. Analyzing the Log: The DBA identifies transactions that were completed before the crash and those that were incomplete. For example, Transaction A (updating inventory for Product X) started at 10:01 AM and completed at 10:02 AM. Transaction B (processing a sale for Product Y) started at 10:03 AM but was interrupted at 10:04 AM.

3. Recovery Process:

Redo Completed Transactions: The DBA ensures that all changes made by completed transactions (e.g., Transaction A) are present in the database by replaying these transactions if necessary.

Undo Incomplete Transactions: For incomplete transactions (e.g., Transaction B), the DBA rolls back any partial changes to ensure the database remains consistent. This involves reversing any steps logged for the transaction.

4. Database Restoration: The transaction log helps restore the database to a state that reflects all completed transactions up to the point of the crash while excluding any incomplete ones. This ensures that inventory levels and sales records are accurate and consistent.