Gandikota Murali

Gandikotamurali951@gmail.com

Day – 9 Assignment

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.

To retrieve all columns from a 'customers' table, using SELECT query

```
SELECT * FROM customers;
```

```
+-------------+---------------+---------------+------+
| customer_id | customer_name | email         | city |
+-------------+---------------+---------------+------+
|           1 | Murali        | xyz@gmail.com | hyd  |
|           2 | krish         | abc@gmail.com | bang |
|           3 | ramu          | mno@gmail.com | pune |
|           4 | mukesh        | efg@gmail.com | chennai|
+-------------+---------------+---------------+------+
```

```
SELECT customer_name, email  FROM customers  WHERE city = 'specific_city';
```

```
---------------+---------------+
| customer_name | email         |
+---------------+---------------+
| Murali     | xyz@gmail.com |
| krish      | abc@gmail.com |
+---------------+---------------+
```

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.

```
SELECT * FROM customers LEFT JOIN orders ON

C.customer_id = orders.customer_id INNER JOIN C ON

customers.city = 'specified_city';

+-------------+---------------+---------------+------++------------+--------+|

| customer_id | customer_name | email         | city |order_item | price

+-------------+---------------+---------------+------++------------+--------+

|           1 | Murali        | xyz@gmail.com | hyd  | mobile     |  15000 |

|           2 | krish      | abc@gmail.com | bang  | laptop     |  65000 |

|           3 | ramu          | mno@gmail.com | pune  |watch     | 13000|

|           4 | mukesh        | efg@gmail.com | chennai|mac_book |140000|

+-------------+---------------+---------------+------++------------+--------+
```

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.

```
// Subquery to find customers who have placed orders above the average order value:

SELECT customer_id FROM orders GROUP BY customer_id

HAVING SUM(order_amount) > (SELECT AVG(order_amount)

FROM orders);

// UNION query to combine two SELECT statements with the same number of columns:

SELECT column1, column2, column3 FROM table1

UNION SELECT column1, column2, column3 FROM table2;
```

+-------------+---------------+---------------+------+

| customer_name | city |order_item | price

+-------------+---------------+---------------+------+

| Murali | hyd | mobile | 15000 |

| krish | bang | laptop | 65000 |

| ramu | pune |watch | 13000|

| mukesh |chennai|mac_book |140000|

+-------------+---------------+---------------+------+

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.

BEGIN TRANSACTION :

INSERT INTO orders (customer_id, customer_name, order_date, city, order_item,price)

VALUES (5,'2022-10-10',’Mumbai’. 300.00);

+-------------+---------------+---------------+------++------------+--------+|

| customer_id | customer_name | order_date | city |order_item | price

+-------------+---------------+---------------+------++------------+--------+

| 1 | Murali | 2023-03-15 | hyd | mobile | 15000 |

| 2 | krish | 2023-05-03 | bang | laptop | 65000 |

| 3 | ram | 2022-12-30 | pune |watch | 13000|

| 4 | mukesh | 2022-11-13 | chennai|mac_book |140000|

| 5 |Null | 2022-10-10 | Mumbai | book |300|

+-------------+---------------+---------------+------++------------+--------+

COMMIT;

update customers set customerid=5 where customername='sravya';

ROLLBACK;

```
+-------------+---------------+---------------+------++------------+--------+|
| customer_id | customer_name | email         | city |order_item | pr
+-------------+---------------+---------------+------++------------+--------+
|           1 | Murali        | xyz@gmail.com | hyd  | mobile     |  15000 |
|           2 | krish         | abc@gmail.com | bang  | laptop     |  65000 |
|           3 | ramu      | mno@gmail.com | pune  |watch     | 13000|
|           4 | mukesh        | efg@gmail.com | chennai|mac_book |140000|
|           5 |sravya        | pno@gmail.com | Mumbai | books    |300|
+-------------+---------------+---------------+------++------------+--------+
```

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

```
BEGIN;
INSERT INTO orders (id, customer_id, total) VALUES (1, 1001, 25);
SAVEPOINT sp1;
 INSERT INTO orders (id, customer_id, total) VALUES (2, 1002, 50);
SAVEPOINT sp2;
 INSERT INTO orders (id, customer_id, total) VALUES (3, 1003, 75);
SAVEPOINT sp3;
ROLLBACK TO SAVEPOINT sp2;
 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.

Transaction logs are crucial components of database management systems that record all changes made to a database. These logs serve as a reliable source of information for recovering data in the event of system failures or unexpected shutdowns.

1. Data Integrity: Transaction logs ensure data integrity by recording every transaction before it is committed to the database. This allows for rollbacks or recovery to a specific point in time.

2. Recovery Point: They provide a recovery point in case of system failures, allowing databases to be restored to a consistent state prior to the failure.

3. Performance Monitoring: Transaction logs also aid in performance monitoring and troubleshooting, as they track changes and can identify potential issues.

Hypothetical Scenario:

Imagine a scenario where a large e-commerce company experiences an unexpected server shutdown during a peak shopping period, resulting in potential data loss and customer disruption. However, due to the implementation of transaction logs, the company's database administrator can initiate a successful data recovery process.

Scenario Details:

1. Unexpected Shutdown: The e-commerce platform experiences a sudden server shutdown due to a power outage.

2. Data Loss Concerns: Concerns arise about potential data loss, including ongoing transactions and customer orders that were being processed.

3. Transaction Logs Utilization: The database administrator leverages transaction logs to restore the database to its state just before the shutdown.

4. Recovery Process: By analysing the transaction logs, the administrator identifies the last committed transactions before the shutdown.

5. Database Restoration: Using this information, the administrator restores the database to the point just before the unexpected shutdown, ensuring minimal data loss and maintaining data consistency.

6. Customer Impact Mitigation: The quick recovery minimizes disruption for customers, allowing them to resume their transactions seamlessly.

Conclusion:

Transaction logs play a vital role in data recovery, especially in scenarios of unexpected shutdowns or system failures. By maintaining a record of all database transactions, transaction logs enable organizations to restore data integrity and minimize downtime, ultimately ensuring business continuity and customer satisfaction.