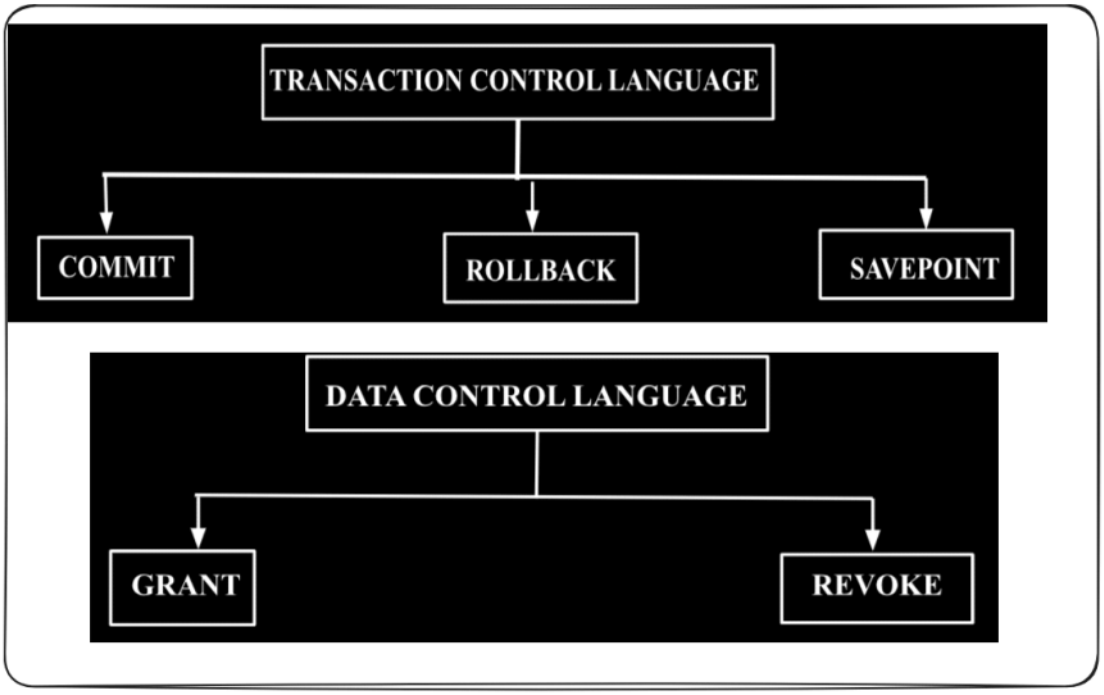
TCL & DCL Commands

🎯 Session Objectives

- ✅ Understand what TCL (Transaction Control Language) is.
- ✅ Learn how to use COMMIT, ROLLBACK, and SAVEPOINT.
- ✅ Explore how TCL enforces ACID properties (Atomicity, Consistency, Isolation, Durability).
- ✅ Understand practical usage with real SQL examples

SYNTAX:

```
GRANT privilege_name
ON object_name
TO {user_name | PUBLIC | role_name}
[WITH GRANT OPTION];
```



ACID Properties in TCL

A - Atomicity
C - Consistency
I - Isolation
D - Durability.

```
USE Weekend_analysis;

CREATE TABLE bank_transactions(
    txn_id INT AUTO_INCREMENT PRIMARY KEY,
    account_id INT,
    balance DECIMAL(10,2),
    remarks VARCHAR(100),
    txn_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

DESC bank_transactions;
```

Field	Type	Null	Key	Default	Extra
txn_id	int	NO	PRI	NULL	auto_increment
account_id	int	YES		NULL	
balance	decimal(10,2)	YES		NULL	
remarks	varchar(100)	YES		NULL	
txn_time	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

```
INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , 1000.00 , 'Opening Balance'),
(2 , 2000.00 , 'Opening Balance');

SELECT * FROM bank_transactions;
```

```
-- ATOMICITY [ALL OR NOTHING Execution]

START TRANSACTION;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , -500.00 , 'Debit For Transfer');

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES ('INVALID' , 500.00 , 'Credit For Transfer');
-- Error Code: 1366. Incorrect integer value: 'INVALID' for column 'account_id' at row 1

ROLLBACK; -- Rollback the Partial Transaction....
```

```
-- ATOMICITY [ALL OR NOTHING Execution]
START TRANSACTION;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , -500.00 , 'Debit For Transfer');
SELECT * FROM bank_transactions;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (2 , 500.00 , 'Credit For Transfer');
SELECT * FROM bank_transactions;

-- ROLLBACK;
COMMIT;
```

txn_id	account_id	balance	remarks	txn_time
1	1	1000.00	Opening Balance	2025-11-15 11:17:39
2	2	2000.00	Opening Balance	2025-11-15 11:17:39
5	1	-500.00	Debit For Transfer	2025-11-15 11:28:24
6	2	500.00	Credit For Transfer	2025-11-15 11:28:43
NULL	NULL	NULL	NULL	NULL

```sql
-- How to find the Current Balance
SELECT
    account_id,
    SUM(balance) AS remaining_balance
FROM bank_transactions
GROUP BY account_id;
```

account_id	remaining_balance
1	500.00
2	2500.00

```sql
-- CONSISTENCY [Total Amount still be Same throughout the Transactions]
START TRANSACTION;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (2 , -1500.00 , 'Debit For Transfer');
SELECT * FROM bank_transactions;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , 1500, 'Credit For Transfer');
SELECT * FROM bank_transactions;

-- ROLLBACK;
COMMIT;
```

txn_id	account_id	balance	remarks	txn_time
1	1	1000.00	Opening Balance	2025-11-15 11:17:39
2	2	2000.00	Opening Balance	2025-11-15 11:17:39
5	1	-500.00	Debit For Transfer	2025-11-15 11:28:24
6	2	500.00	Credit For Transfer	2025-11-15 11:28:43
7	2	-1500.00	Debit For Transfer	2025-11-15 11:38:38
8	1	1500.00	Credit For Transfer	2025-11-15 11:38:48
NULL	NULL	NULL	NULL	NULL

```sql
-- How to find the Current Balance
SELECT
    account_id,
    SUM(balance) AS remaining_balance
FROM bank_transactions
GROUP BY account_id;
```

account_id	remaining_balance
1	2000.00
2	1000.00

```
-- ISOLATION [Transactions Don't Interfere]

START TRANSACTION;

SELECT
    SUM(balance) AS remaining_balance
FROM bank_transactions
WHERE account_id = 1; -- Checking Balance

COMMIT; -- Don't Commit Yet.

-- PARALLEL TRANSACTIONS;
START TRANSACTION;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , 200, 'Deposit');

COMMIT;
```

remaining_balance
2200.00

txn_id	account_id	balance	remarks	txn_time
1	1	1000.00	Opening Balance	2025-11-15 11:17:39
2	2	2000.00	Opening Balance	2025-11-15 11:17:39
5	1	-500.00	Debit For Transfer	2025-11-15 11:28:24
6	2	500.00	Credit For Transfer	2025-11-15 11:28:43
7	2	-1500.00	Debit For Transfer	2025-11-15 11:38:38
8	1	1500.00	Credit For Transfer	2025-11-15 11:38:48
9	1	200.00	Deposit	2025-11-15 11:48:22
NULL	NULL	NULL	NULL	NULL

```
-- ISOLATION [Transactions Don't Interfere]

START TRANSACTION;

SELECT
    SUM(balance) AS remaining_balance
FROM bank_transactions
WHERE account_id = 1; -- Checking Balance

COMMIT; -- Don't Commit Yet.

-- PARALLEL TRANSACTIONS;
START TRANSACTION;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , 200, 'Deposit');

ROLLBACK;

-- COMMIT;

SELECT * FROM bank_transactions;
```

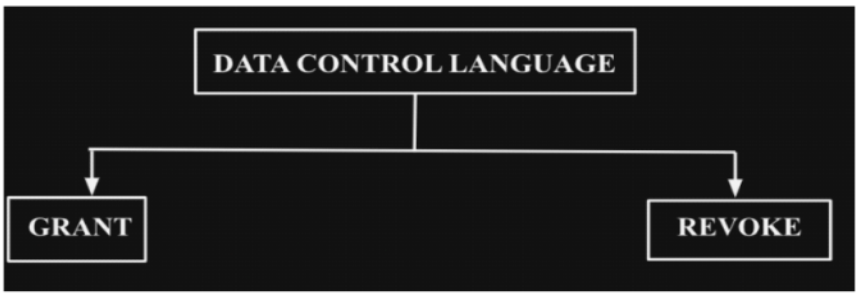
remaining_balance
1800.00

```
-- DURABALITY -- Changes Persist after Commit....
START TRANSACTION;

INSERT INTO bank_transactions (account_id, balance, remarks)
VALUES (1 , -100, 'ATM Withdrawl');

COMMIT;

SELECT * FROM bank_transactions;
```

txn_id	account_id	balance	remarks	txn_time
1	1	1000.00	Opening Balance	2025-11-15 11:17:39
2	2	2000.00	Opening Balance	2025-11-15 11:17:39
5	1	-500.00	Debit For Transfer	2025-11-15 11:28:24
6	2	500.00	Credit For Transfer	2025-11-15 11:28:43
7	2	-1500.00	Debit For Transfer	2025-11-15 11:38:38
8	1	1500.00	Credit For Transfer	2025-11-15 11:38:48
9	1	200.00	Deposit	2025-11-15 11:48:22
10	1	-500.00	Paying for Grocery	2025-11-15 11:54:30
11	1	300.00	Deposit	2025-11-15 11:54:53
12	1	-500.00	Paying for Grocery	2025-11-15 11:58:18
15	1	300.00	Deposit	2025-11-15 12:01:16
17	1	-100.00	ATM Withdrawl	2025-11-15 12:18:15
NULL	NULL	NULL	NULL	NULL

DCL - Data Control Language

DBA - Database Administrator



SYNTAX:

```
GRANT privilege_name
ON object_name
TO {user_name | PUBLIC | role_name}
[WITH GRANT OPTION];
```

https://dev.mysql.com/doc/refman/8.4/en/grant.html

```
CREATE DATABASE dcl_overview;

USE dcl_overview;

CREATE TABLE Employee_data(
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(100),
    salary DECIMAL(10,2)
);

DESC Employee_data;
```

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	NULL	
emp_name	varchar(100)	YES		NULL	
salary	decimal(10,2)	YES		NULL	

```
-- Let's Create a New User
CREATE USER 'deepak'@'localhost' IDENTIFIED BY 'deepak@123';
CREATE USER 'vaibhav'@'localhost'  IDENTIFIED BY 'vaibhav@123';
```

```
94  12:43:04  CREATE USER 'deepak'@'localhost' IDENTIFIED BY 'deepak@123'   0 row(s) affected
95  12:43:08  CREATE USER 'vaibhav'@'localhost' IDENTIFIED BY 'vaibhav@123'  0 row(s) affected
```

```
-- Grant Some Access to the Users
GRANT SELECT ON dcl_overview.employee_data TO 'deepak'@'localhost';
GRANT INSERT ON dcl_overview.employee_data TO 'vaibhav'@'localhost';
```

```
Command Prompt
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\krish>mysql -u deepak -p
'mysql' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\krish>
```

```
Command Prompt - "C:\Progr
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\krish>mysql -u deepak -p
'mysql' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\krish>"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" -u deepak -p
Enter password: *********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.42 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| dcl_overview       |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.12 sec)

mysql> USE dcl_overview;
Database changed
mysql> SHOW TABLES;
+------------------------+
| Tables_in_dcl_overview |
+------------------------+
| employee_data          |
+------------------------+
1 row in set (0.01 sec)
```

```
mysql> INSERT INTO employee_data(emp_id , emp_name, salary)
    -> VALUES(1 , 'Deepak' , 1000000.00);
ERROR 1142 (42000): INSERT command denied to user 'deepak'@'localhost' for table 'employee_data'
```

Ctrl + Z : Bye

```
mysql> ^Z
Bye

C:\Users\krish>"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" -u vaibhav -p
Enter password: ***********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.42 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| dcl_overview       |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.01 sec)

mysql> USE dcl_overview;
Database changed
mysql> SHOW TABLES;
+------------------------+
| Tables_in_dcl_overview |
+------------------------+
| employee_data          |
+------------------------+
1 row in set (0.01 sec)

mysql> SELECT * FROM employee_data;
ERROR 1142 (42000): SELECT command denied to user 'vaibhav'@'localhost' for table 'employee_data'
```

```
mysql> INSERT INTO Employee_data (emp_id, emp_name, salary)
    -> VALUES
    -> (101, 'Amit Verma', 45000.00),
    -> (102, 'Neha Sharma', 52000.50),
    -> (103, 'Rohit Kumar', 39000.00),
    -> (104, 'Sneha Kapoor', 61000.75),
    -> (105, 'Vikram Singh', 48000.00),
    -> (106, 'Priya Nair', 70000.00),
    -> (107, 'Rahul Joshi', 36000.00),
    -> (108, 'Ananya Gupta', 55000.25),
    -> (109, 'Manish Tiwari', 43000.80),
    -> (110, 'Pooja Mehta', 58000.00);
Query OK, 10 rows affected (0.04 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> ^Z
Bye

C:\Users\krish>"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" -u deepak -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.42 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| dcl_overview       |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.01 sec)

mysql> USE dcl_overview;
Database changed
mysql> SHOW TABLES;
+------------------------+
| Tables_in_dcl_overview |
+------------------------+
| employee_data          |
+------------------------+
1 row in set (0.01 sec)
```

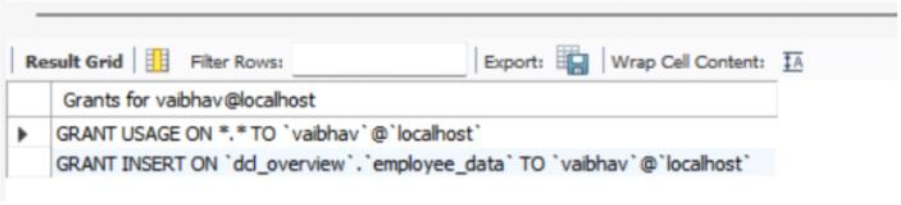
```
mysql> SELECT * FROM  employee_data;
+--------+---------------+----------+
| emp_id | emp_name      | salary   |
+--------+---------------+----------+
|    101 | Amit Verma    | 45000.00 |
|    102 | Neha Sharma   | 52000.50 |
|    103 | Rohit Kumar   | 39000.00 |
|    104 | Sneha Kapoor  | 61000.75 |
|    105 | Vikram Singh  | 48000.00 |
|    106 | Priya Nair    | 70000.00 |
|    107 | Rahul Joshi   | 36000.00 |
|    108 | Ananya Gupta  | 55000.25 |
|    109 | Manish Tiwari | 43000.80 |
|    110 | Pooja Mehta   | 58000.00 |
+--------+---------------+----------+
10 rows in set (0.00 sec)
```

```
525 •  SHOW GRANTS FOR 'deepak'@'localhost';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

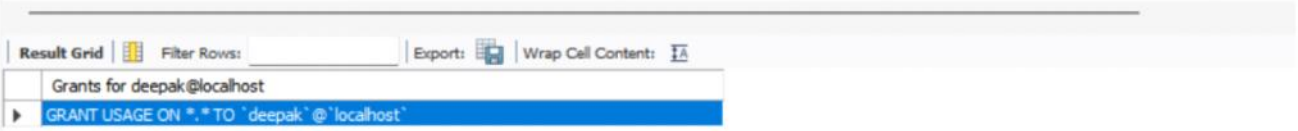
Grants for deepak@localhost
GRANT USAGE ON *.* TO `deepak`@`localhost`
GRANT SELECT ON `dcl_overview`.`employee_data` TO `deepak`@`localhost`

```
527 •  SHOW GRANTS FOR 'vaibhav'@'localhost';
```



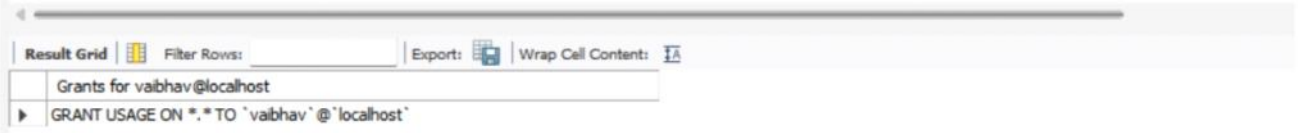
Grants for vaibhav@localhost
GRANT USAGE ON *.* TO `vaibhav`@`localhost`
GRANT INSERT ON `dd_overview`.`employee_data` TO `vaibhav`@`localhost`

```
529    -- Revoking the Access of a Particular User
530 •  REVOKE SELECT ON dcl_overview.employee_data FROM 'deepak'@'localhost';
```



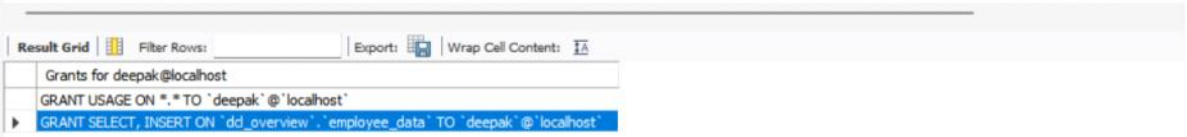
Grants for deepak@localhost
GRANT USAGE ON *.* TO `deepak`@`localhost`

```
532 •  REVOKE INSERT ON dcl_overview.employee_data FROM 'vaibhav'@'localhost';
```



Grants for vaibhav@localhost
GRANT USAGE ON *.* TO `vaibhav`@`localhost`

```
534 •  GRANT SELECT,INSERT ON dcl_overview.employee_data TO 'deepak'@'localhost';
535 •  GRANT CREATE ON dcl_overview.employee_data TO 'vaibhav'@'localhost';
```



Grants for deepak@localhost
GRANT USAGE ON *.* TO `deepak`@`localhost`
GRANT SELECT, INSERT ON `dd_overview`.`employee_data` TO `deepak`@`localhost`

Grants for vaibhav@localhost
GRANT USAGE ON *.* TO `vaibhav`@`localhost`
GRANT CREATE ON `dd_overview`.`employee_data` TO `vaibhav`@`localhost`

```
SHOW GRANTS FOR 'deepak'@'localhost';

SHOW GRANTS FOR 'vaibhav'@'localhost';

-- Revoking the Access of a Particular User
REVOKE SELECT ON dcl_overview.employee_data FROM 'deepak'@'localhost';

REVOKE INSERT ON dcl_overview.employee_data FROM 'vaibhav'@'localhost';

GRANT SELECT,INSERT ON dcl_overview.employee_data TO 'deepak'@'localhost';
GRANT CREATE ON dcl_overview.employee_data TO 'vaibhav'@'localhost';
```

Case Study - Retail Analytics - I

Background

Retail companies face challenges including stagnant growth, unclear customer segmentation, and inventory inefficiencies. This case study examines how SQL-based data analysis can identify product performance trends, segment customers, and reveal behavioural patterns to inform marketing and inventory decisions.

Business Problems

➡️ Product Performance Variability: Identify best and worst-selling products.

➡️ Customer Segmentation: Group customers by purchasing patterns.

➡️ Customer Behaviour Analysis: Understand repeat purchase habits and loyalty signals.

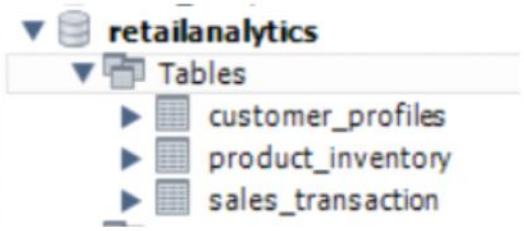
Objectives:

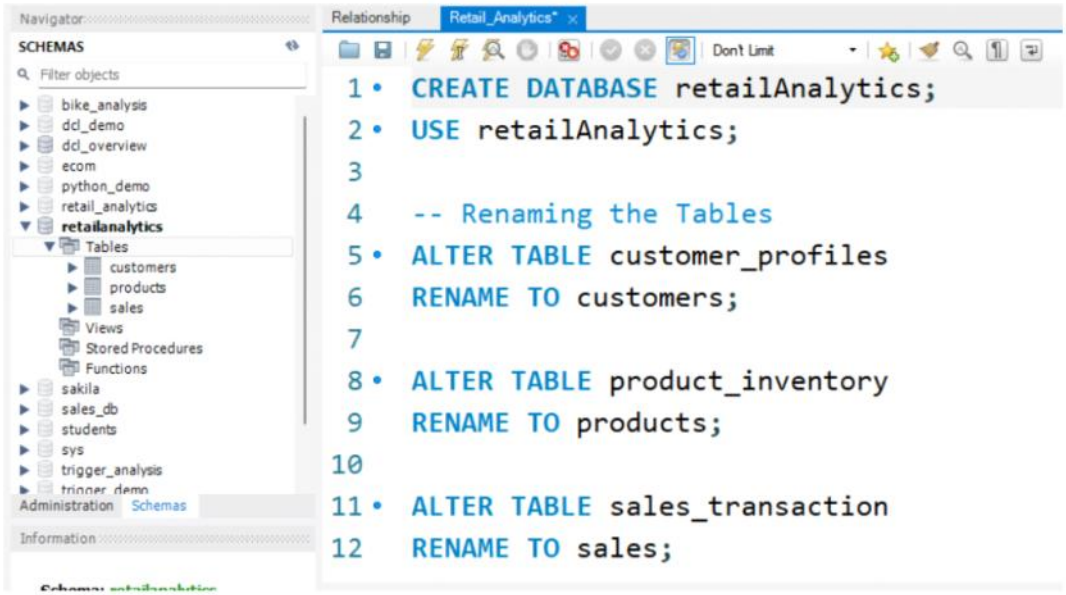
- *To utilize SQL queries for data cleaning and exploratory data analysis to ensure data quality and gain initial insights.*
- *To identify high and low sales products to optimize inventory and tailor marketing efforts.*
- *To segment customers based on their purchasing behavior for targeted marketing campaigns. Create Customer segments -*

Total Quantity of Products Purchased	Customer Segment
0	No Orders
1-10	Low
10-30	Mid
>30	High Value

- *To analyze customer behavior for insights on repeat purchases and loyalty, informing customer retention strategies.*

Reading out the dataset






```
14 • DESC customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
i»¿CustomerID	int	YES		NULL	
Age	int	YES		NULL	
Gender	text	YES		NULL	
Location	text	YES		NULL	
JoinDate	text	YES		NULL	

```
15 • DESC products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

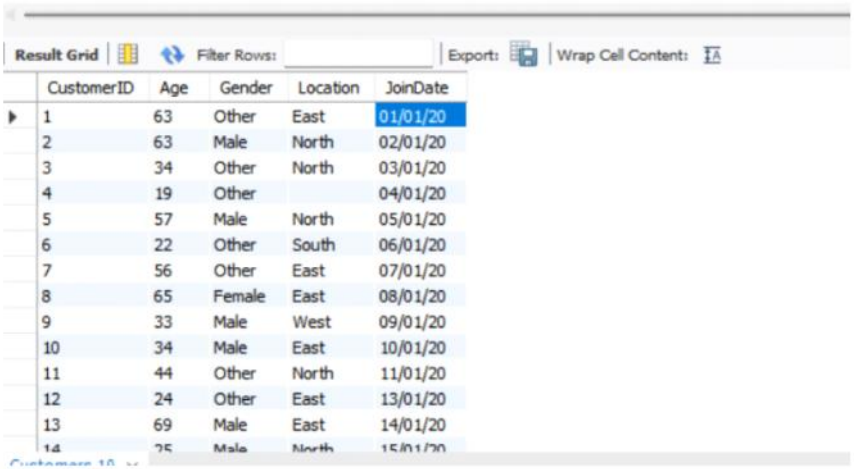
Field	Type	Null	Key	Default	Extra
i»¿ProductID	int	YES		NULL	
ProductName	text	YES		NULL	
Category	text	YES		NULL	
StockLevel	int	YES		NULL	
Price	double	YES		NULL	

```
16 • DESC sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

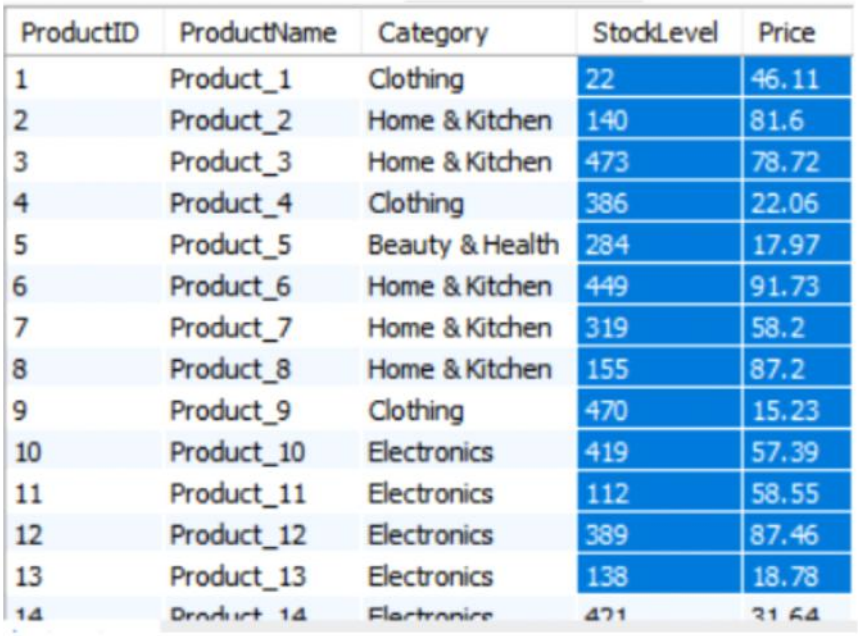
Field	Type	Null	Key	Default	Extra
i»¿TransactionID	int	YES		NULL	
CustomerID	int	YES		NULL	
ProductID	int	YES		NULL	
QuantityPurcha...	int	YES		NULL	
TransactionDate	text	YES		NULL	
Price	double	YES		NULL	

```
35    -- Reading out all the dataset ....
36 •  SELECT * FROM Customers;
```



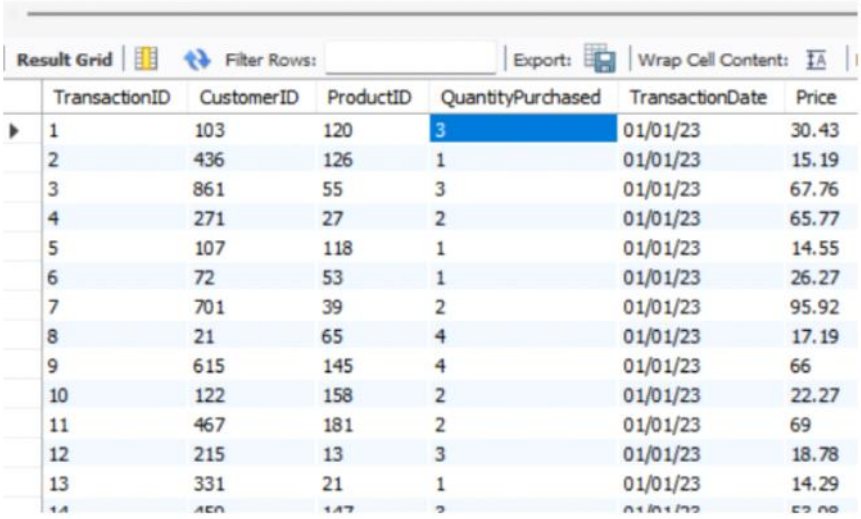
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	Age	Gender	Location	JoinDate
1	63	Other	East	01/01/20
2	63	Male	North	02/01/20
3	34	Other	North	03/01/20
4	19	Other		04/01/20
5	57	Male	North	05/01/20
6	22	Other	South	06/01/20
7	56	Other	East	07/01/20
8	65	Female	East	08/01/20
9	33	Male	West	09/01/20
10	34	Male	East	10/01/20
11	44	Other	North	11/01/20
12	24	Other	East	13/01/20
13	69	Male	East	14/01/20
14	25	Male	North	15/01/20



ProductID	ProductName	Category	StockLevel	Price
1	Product_1	Clothing	22	46.11
2	Product_2	Home & Kitchen	140	81.6
3	Product_3	Home & Kitchen	473	78.72
4	Product_4	Clothing	386	22.06
5	Product_5	Beauty & Health	284	17.97
6	Product_6	Home & Kitchen	449	91.73
7	Product_7	Home & Kitchen	319	58.2
8	Product_8	Home & Kitchen	155	87.2
9	Product_9	Clothing	470	15.23
10	Product_10	Electronics	419	57.39
11	Product_11	Electronics	112	58.55
12	Product_12	Electronics	389	87.46
13	Product_13	Electronics	138	18.78
14	Product_14	Electronics	471	31.64

```
40 •  SELECT * FROM Sales;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TransactionID	CustomerID	ProductID	QuantityPurchased	TransactionDate	Price
1	103	120	3	01/01/23	30.43
2	436	126	1	01/01/23	15.19
3	861	55	3	01/01/23	67.76
4	271	27	2	01/01/23	65.77
5	107	118	1	01/01/23	14.55
6	72	53	1	01/01/23	26.27
7	701	39	2	01/01/23	95.92
8	21	65	4	01/01/23	17.19
9	615	145	4	01/01/23	66
10	122	158	2	01/01/23	22.27
11	467	181	2	01/01/23	69
12	215	13	3	01/01/23	18.78
13	331	21	1	01/01/23	14.29
14	450	147	3	01/01/23	53.08

Challenge- 1

Remove Duplicates

Easy • Score 40/40 • Average time to solve is 10m

Problem statement

Send feedback

Write a query to identify the number of duplicates in "sales_transaction" table. Also, create a separate table containing the unique values and remove the the original table from the databases and replace the name of the new table with the original name.

Hint:

- *Use the "Sales_transaction" table.*
- *There will be two resulting tables in the output. First, the table where the count of duplicates will be identified and in the second table we can check if the duplicates were removed or not by selecting the whole table.*

Output format:

```
+---------------+-----------+
| TransactionID | count(*) |
+---------------+-----------+
| Transaction 1 |    NUM    |
| Transaction 2 |    NUM    |
+---------------+-----------+

+---------------+------------+-----------+-------------------+-----------------+-------+
| TransactionID | CustomerID | ProductID | QuantityPurchased | TransactionDate | Price |
+---------------+------------+-----------+-------------------+-----------------+-------+
| Transaction 1 | Customer 1 | Product 1 |        NUM        |       TEXT      |DECINUM|
| Transaction 2 | Customer 2 | Product 2 |        NUM        |       TEXT      |DECINUM|
| Transaction 3 | Customer 3 | Product 3 |        NUM        |       TEXT      |DECINUM|
| Transaction 4 | Customer 4 | Product 4 |        NUM        |       TEXT      |DECINUM|
+-------------+----------+----------+----------------+-------------+-------+
```

Note: *The NUM in the output format denotes a numerical value and transaction 1 means transaction with unique id 1.*

```
40 •  SELECT * FROM Sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fe

TransactionID	CustomerID	ProductID	QuantityPurchased	TransactionDate	Price
4990	474	103	3	27/07/23	93.81
4991	13	47	2	27/07/23	56.53
4992	184	39	1	27/07/23	95.92
4993	111	29	2	28/07/23	51.74
4994	927	54	4	28/07/23	82.01
4995	508	90	2	28/07/23	39.33
4996	290	85	2	28/07/23	99.81
4997	295	17	3	28/07/23	94.5
4998	451	80	1	28/07/23	58.28
4999	904	188	2	28/07/23	36.22
4999	904	188	2	28/07/23	36.22
5000	215	159	2	28/07/23	17.42
5000	215	159	2	28/07/23	17.42

5002 row(s) returned

```sql
SELECT
    TransactionID,
    COUNT(*)
FROM Sales
GROUP BY TransactionID
HAVING COUNT(*) > 1;
```

Result Grid | Filter Rows: | Export:

TransactionID	COUNT(*)
4999	2
5000	2

```sql
-- Challenge 1:
SELECT
    TransactionID,
    COUNT(*)
FROM Sales
GROUP BY TransactionID
HAVING COUNT(*) > 1;

CREATE TABLE sales_unique AS
SELECT DISTINCT * FROM sales; -- 5000 row(s) affected Records: 5000

DROP TABLE sales;
ALTER TABLE sales_unique RENAME TO sales;
```