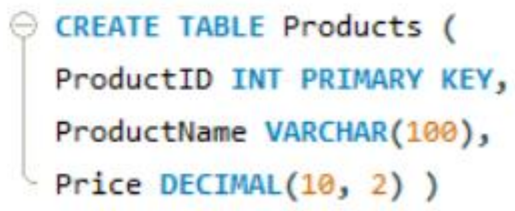
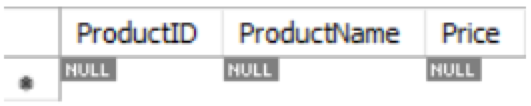
Alvhin C. Solo **COM232** **April 2, 2025**

DDL Exercise

1.Create a table named Products:

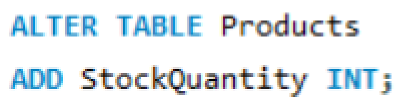


```sql
CREATE TABLE Products (
ProductID INT PRIMARY KEY,
ProductName VARCHAR(100),
Price DECIMAL(10, 2) )
```

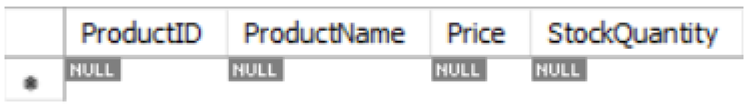


	ProductID	ProductName	Price
*	NULL	NULL	NULL

2. Add a new column StockQuantity;

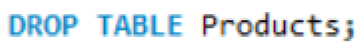


```sql
ALTER TABLE Products
ADD StockQuantity INT;
```

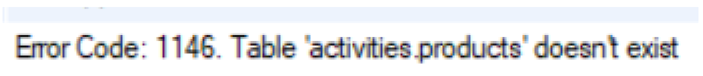


	ProductID	ProductName	Price	StockQuantity
*	NULL	NULL	NULL	NULL

3. Remove the Products table:



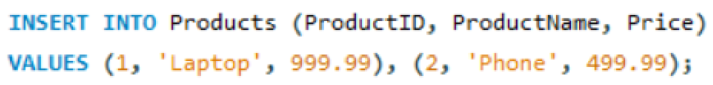
```sql
DROP TABLE Products;
```



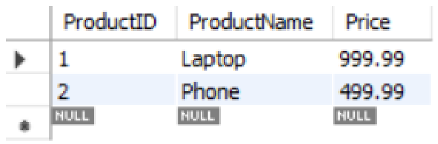
```
Error Code: 1146. Table 'activities.products' doesn't exist
```

DML Exercise

1.Insert two records into Products

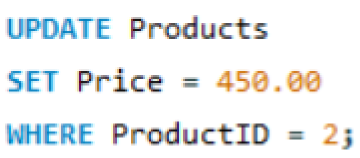


```sql
INSERT INTO Products (ProductID, ProductName, Price)
VALUES (1, 'Laptop', 999.99), (2, 'Phone', 499.99);
```

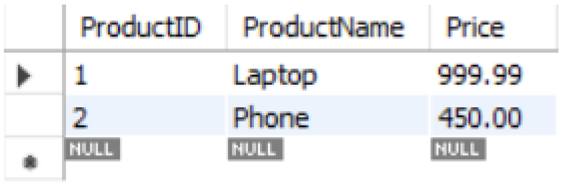


	ProductID	ProductName	Price
▶	1	Laptop	999.99
	2	Phone	499.99
*	NULL	NULL	NULL

2. Update the price of the Phone

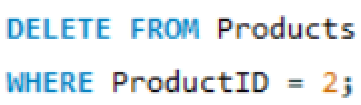


```sql
UPDATE Products
SET Price = 450.00
WHERE ProductID = 2;
```

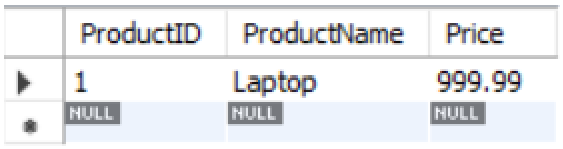


	ProductID	ProductName	Price
▶	1	Laptop	999.99
	2	Phone	450.00
*	NULL	NULL	NULL

3.Delete the Phone record



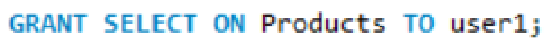
```sql
DELETE FROM Products
WHERE ProductID = 2;
```



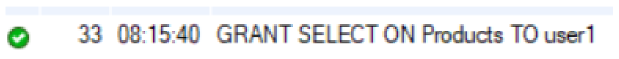
	ProductID	ProductName	Price
▶	1	Laptop	999.99
*	NULL	NULL	NULL

DCL Exercise

1.Grant SELECT permission to user user1:

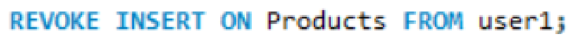


```
GRANT SELECT ON Products TO user1;
```

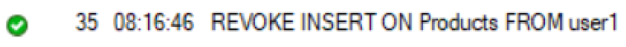


```
33  08:15:40  GRANT SELECT ON Products TO user1
```

2. Revoke INSERT permission from user1:



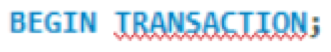
```
REVOKE INSERT ON Products FROM user1;
```



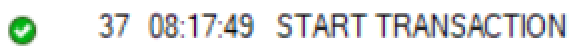
```
35  08:16:46  REVOKE INSERT ON Products FROM user1
```

TCL Exercise

1.Start a transaction:

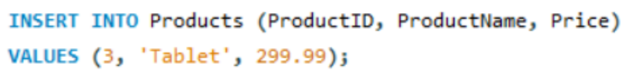


```
BEGIN TRANSACTION;
```

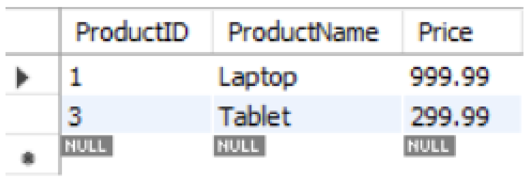


```
37  08:17:49  START TRANSACTION
```

2. Insert a record:

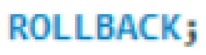


```
INSERT INTO Products (ProductID, ProductName, Price)
VALUES (3, 'Tablet', 299.99);
```

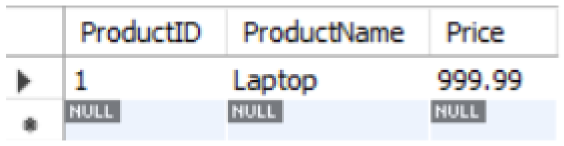


	ProductID	ProductName	Price
▶	1	Laptop	999.99
	3	Tablet	299.99
*	NULL	NULL	NULL

3. Rollback the transaction:

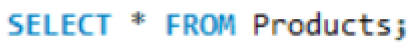


```
ROLLBACK;
```

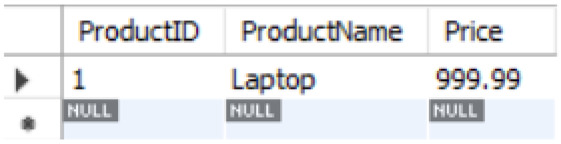


	ProductID	ProductName	Price
▶	1	Laptop	999.99
*	NULL	NULL	NULL

4. Check if the Tablet record still exists:



```
SELECT * FROM Products;
```



	ProductID	ProductName	Price
▶	1	Laptop	999.99
*	NULL	NULL	NULL

Reflection Questions

1.What happens when you use ROLLBACK versus COMMIT?

The **COMMIT** command is used to permanently save all the changes made during a transaction. Once a COMMIT is executed, the modifications cannot be undone, ensuring that the database reflects the final changes. On the other hand, **ROLLBACK** is used to undo any changes made during the transaction, restoring the database to its previous state before the transaction began. This is particularly useful when an error occurs or when the transaction needs to be canceled to maintain data integrity.

2.Why would you use GRANT and REVOKE commands in a real-world database?

In a real-world database, the **GRANT** and **REVOKE** commands play a crucial role in managing user permissions and access control. The **GRANT** command allows database administrators to assign specific privileges, such as reading, writing, or modifying data, to users or roles. This ensures that only authorized individuals can perform certain actions on the database. Conversely, the **REVOKE** command is used to remove those privileges when a user no longer requires access. These commands help enforce security policies, prevent unauthorized data manipulation, and protect sensitive information from breaches or misuse.

3.How does ALTER TABLE differ from UPDATE?

The **ALTER TABLE** command is used to modify the structure of a database table, allowing changes such as adding or deleting columns, modifying column data types, or setting constraints. This command is useful for adapting the database schema to evolving requirements without affecting existing data. In contrast, the **UPDATE** command is used to modify the actual data within a table without altering its structure. It allows specific records to be changed based on a given condition. While **ALTER TABLE** changes how data is stored, **UPDATE** changes the stored data itself.