

# Carl Anastacio

## COM 231

### Activity 2

#### DDL Exercise:

##### 1. Create Product

The screenshot shows the Programiz Online SQL Editor interface. On the left, a tree view shows the 'Product' table with columns: product\_id [int], product\_name [varchar(100)], and price [decimal(10, 2)]. The main input area contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
CREATE TABLE Product (  
  product_id int primary key,  
  product_name varchar(100),  
  price decimal(10, 2)  
);
```

The 'Run SQL' button is visible. On the right, the 'Available Tables' section shows the 'Product' table with columns: product\_id, product\_name, and price. Below this, a table structure is shown with the same columns and a note that the result set is empty.

Output: SQL query successfully executed. However, the result set is empty.

Code:

```
CREATE TABLE Product (  
  product_id int primary key,  
  product_name varchar(100),  
  price decimal(10, 2)  
);
```

##### 2. Alter Table

The screenshot shows the Programiz Online SQL Editor interface. On the left, a tree view shows the 'Product' table with columns: product\_id [int], product\_name [varchar(100)], price [decimal(10, 2)], and stock\_quantity [int]. The main input area contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
ALTER TABLE Product ADD stock_quantity INT;
```

The 'Run SQL' button is visible. On the right, the 'Available Tables' section shows the 'Product' table with columns: product\_id, product\_name, price, and stock\_quantity. Below this, a table structure is shown with the same columns and a note that the result set is empty.

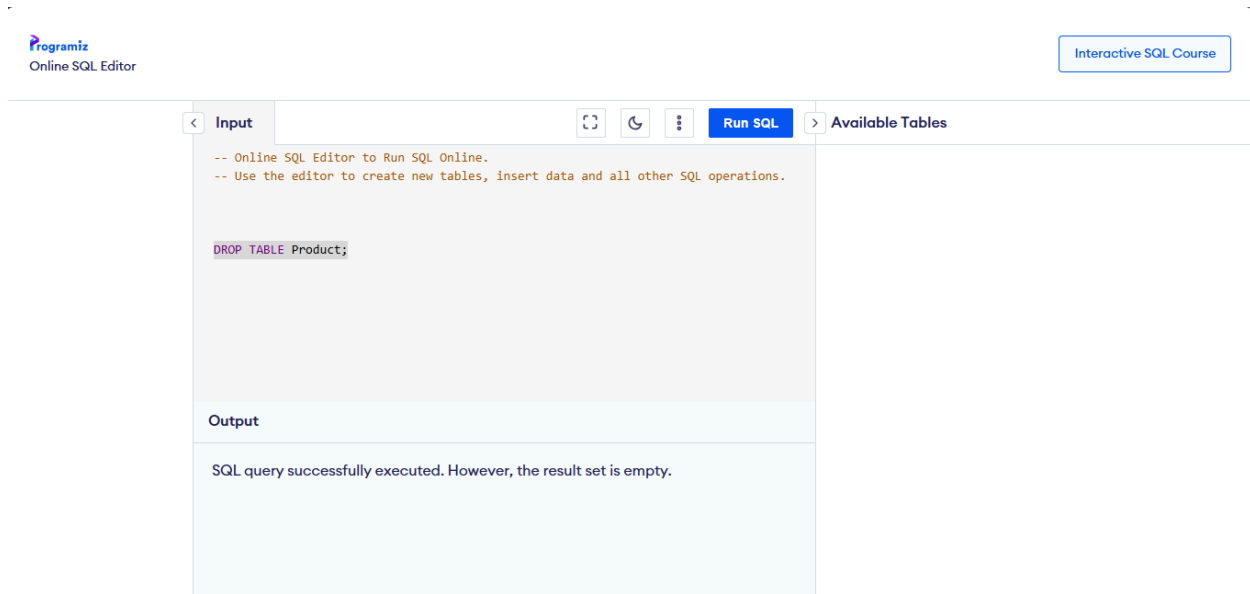
Output: SQL query successfully executed. However, the result set is empty.

Code: ALTER TABLE Product ADD stock\_quantity INT;

# Carl Anastacio

## COM 231

### 3. Drop Table



The screenshot shows the Programiz Online SQL Editor interface. The top bar includes the Programiz logo, "Online SQL Editor", and a button for "Interactive SQL Course". The main workspace is divided into three sections: "Input", "Output", and "Available Tables". The "Input" section contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
DROP TABLE Product;
```

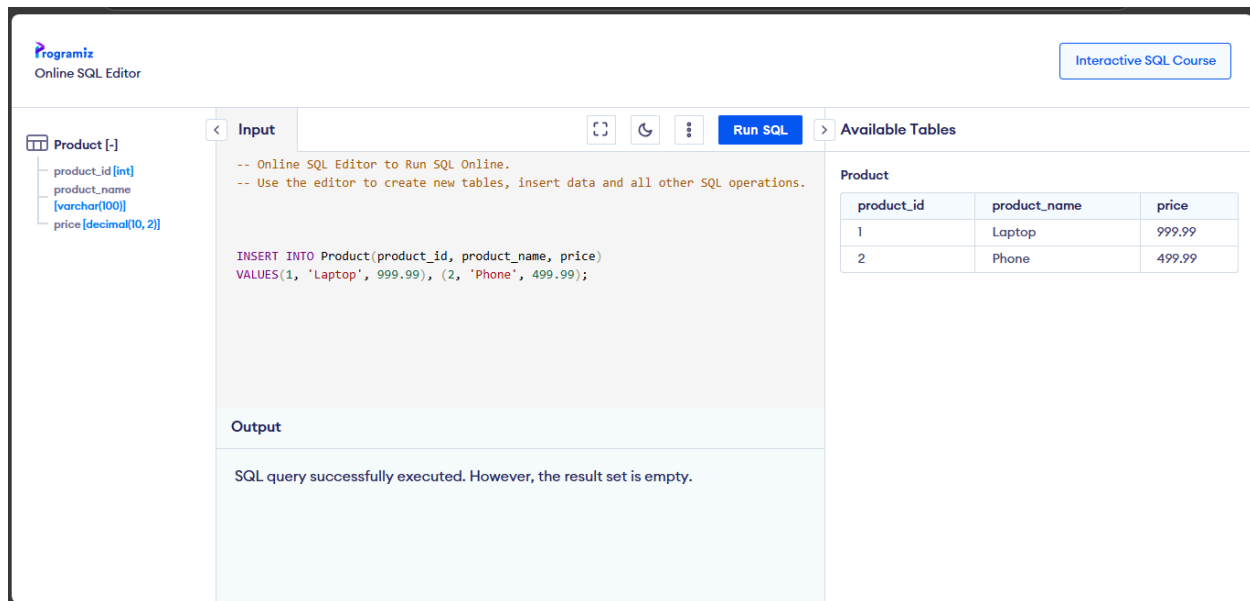
The "Output" section displays the message: "SQL query successfully executed. However, the result set is empty." The "Available Tables" section is currently empty.

Code:

```
DROP TABLE Product;
```

### DML Exercise

#### 1. Insert Table



The screenshot shows the Programiz Online SQL Editor interface. The top bar includes the Programiz logo, "Online SQL Editor", and a button for "Interactive SQL Course". The main workspace is divided into three sections: "Input", "Output", and "Available Tables". The "Input" section contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
INSERT INTO Product(product_id, product_name, price)  
VALUES(1, 'Laptop', 999.99), (2, 'Phone', 499.99);
```

The "Output" section displays the message: "SQL query successfully executed. However, the result set is empty." The "Available Tables" section shows a table named "Product" with the following data:

product_id	product_name	price
1	Laptop	999.99
2	Phone	499.99

Code:

```
INSERT INTO Product(product_id, product_name, price)  
VALUES(1, 'Laptop', 999.99), (2, 'Phone', 499.99);
```

# Carl Anastacio

## COM 231

### 2. Update Data

The screenshot shows the Programiz Online SQL Editor interface. On the left, a database schema for 'Product' is shown with columns: product\_id [int], product\_name [varchar(100)], and price [decimal(10, 2)]. The main input area contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
UPDATE Product  
SET Price = 450.00  
WHERE product_id = 2;
```

The 'Run SQL' button is visible. Below the input area, the 'Output' section displays the message: 'SQL query successfully executed. However, the result set is empty.' On the right, the 'Available Tables' section shows a table named 'Product' with the following data:

product_id	product_name	price
1	Laptop	999.99
2	Phone	450

Code:

```
UPDATE Product  
SET Price = 450.00  
WHERE product_id = 2;
```

### 3. Delete product\_id

The screenshot shows the Programiz Online SQL Editor interface. On the left, the same database schema for 'Product' is shown. The main input area contains the following SQL code:

```
DELETE FROM Product  
WHERE product_id = 2;
```

The 'Run SQL' button is visible. Below the input area, the 'Output' section displays the message: 'SQL query successfully executed. However, the result set is empty.' On the right, the 'Available Tables' section shows a table named 'Product' with the following data:

product_id	product_name	price
1	Laptop	999.99

A large red 'X' is drawn over the table, indicating that the product with id 2 has been successfully deleted.

Code:

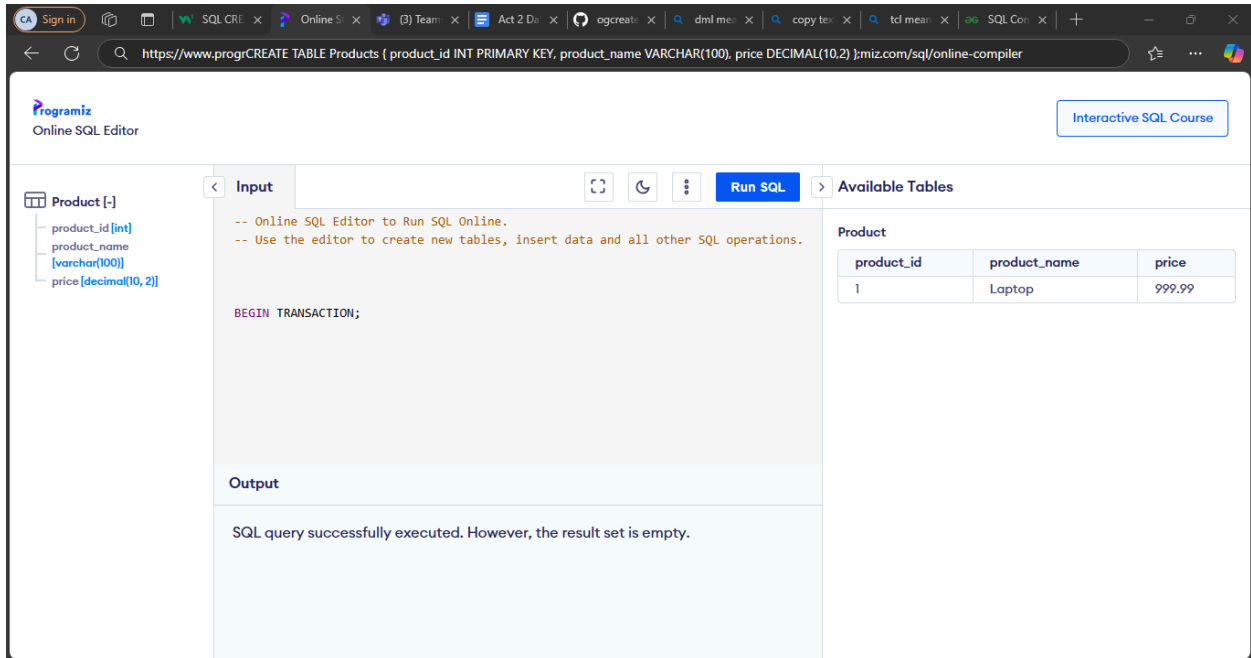
```
DELETE FROM Product  
WHERE product_id = 2;
```

# Carl Anastacio

## COM 231

### TCL Exercise

#### 1. Begin Transaction



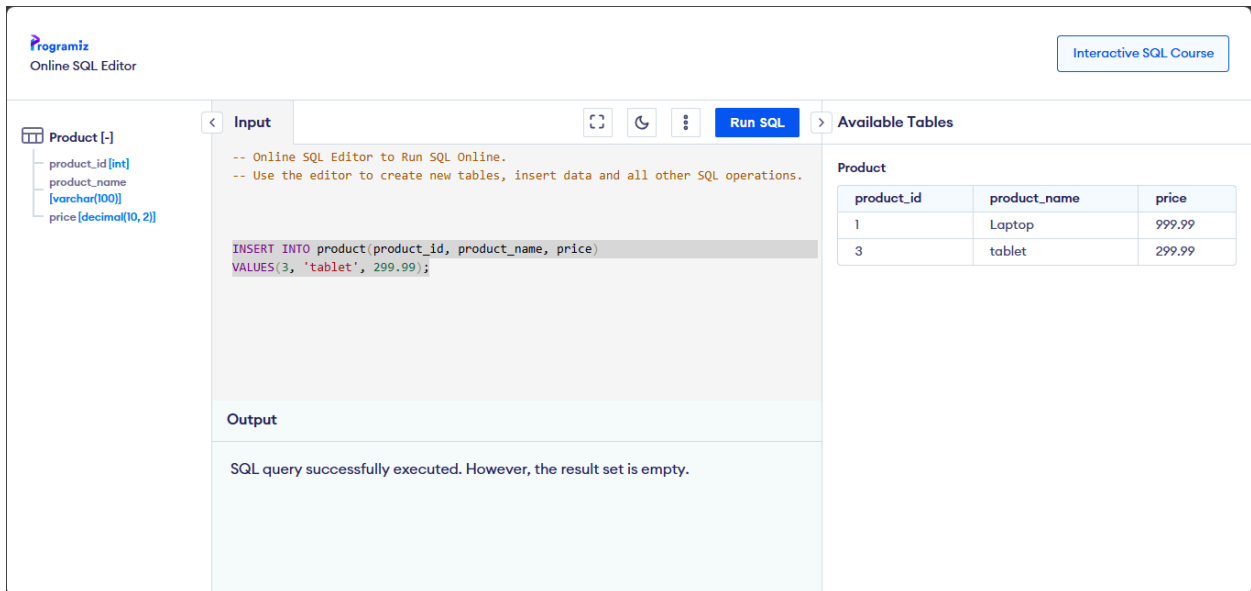
The screenshot shows the Programiz Online SQL Editor interface. On the left, a schema diagram for a table named 'Product' is visible, with columns: product\_id [int], product\_name [varchar(100)], and price [decimal(10, 2)]. The main 'Input' area contains the following SQL code:

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
BEGIN TRANSACTION;
```

A 'Run SQL' button is located to the right of the input area. Below the input area is the 'Output' section, which displays the message: 'SQL query successfully executed. However, the result set is empty.' On the right side, the 'Available Tables' section shows a table named 'Product' with the following data:

product_id	product_name	price
1	Laptop	999.99

#### 2. Insert a Record



The screenshot shows the Programiz Online SQL Editor interface. The schema diagram for the 'Product' table remains the same. The 'Input' area now contains the following SQL code:

```
INSERT INTO product(product_id, product_name, price)  
VALUES(3, 'tablet', 299.99);
```

The 'Run SQL' button is still present. The 'Output' section displays the message: 'SQL query successfully executed. However, the result set is empty.' The 'Available Tables' section now shows two records in the 'Product' table:

product_id	product_name	price
1	Laptop	999.99
3	tablet	299.99

Code:

```
INSERT INTO Product(product_id, product_name, price)  
VALUES(3, 'Tablet', 299.99);
```

### 3. Rollback transaction

Programiz  
Online SQL Editor

Interactive SQL Course

Product [-]

product\_id [int]

product\_name [varchar(100)]

price [decimal(10, 2)]

<

Input

↺ ↻ ⋮

Run SQL

>

-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.

ROLLBACK;

Available Tables

Product

product_id	product_name	price
1	Laptop	999.99

Output

SQL query successfully executed. However, the result set is empty.

### 4. Check Table

Programiz  
Online SQL Editor

Interactive SQL Course

Product [-]

product\_id [int]

product\_name [varchar(100)]

price [decimal(10, 2)]

<

Input

↺ ↻ ⋮

Run SQL

>

-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.

SELECT \* FROM Product;

Available Tables

Product

product_id	product_name	price
1	Laptop	999.99

Output

product_id	product_name	price
1	Laptop	999.99

## **Carl Anastacio**

### **COM 231**

#### **Reflection Questions:**

1. What happens when you use ROLLBACK versus COMMIT?

When we use rollback it previously go back to the BEGIN TRANSACTION like its a check-point for a command and COMMIT finalizes the change that occurred in those command

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

We would use those to protect our DATABASE from unwanted access and only accessible to the people given permission with

3. How does ALTER TABLE differ from UPDATE?

Based on our activity the ALTER TABLE allows us to add column or change the specific table while UPDATE allows us to change the element inside that column or row