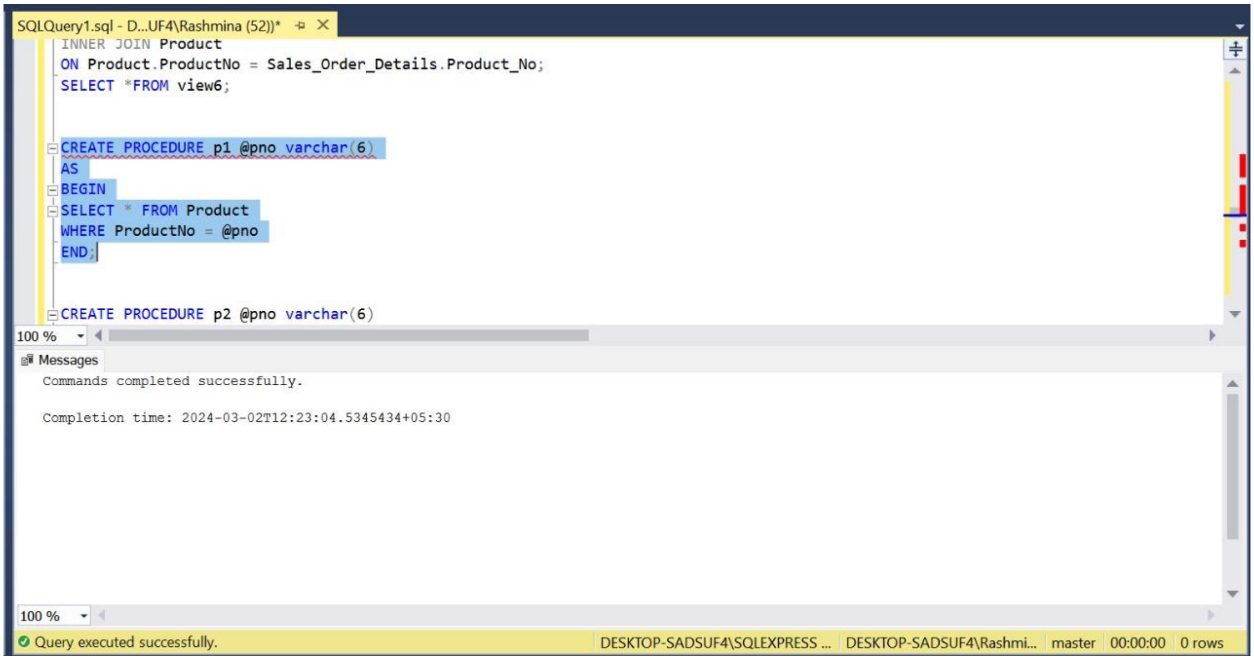
1. Write a stored procedure to display information for a given product number

```
CREATE PROCEDURE p1 @pno
varchar(6)

AS

BEGIN

SELECT * FROM Product

WHERE ProductNo = @pno

END;
```



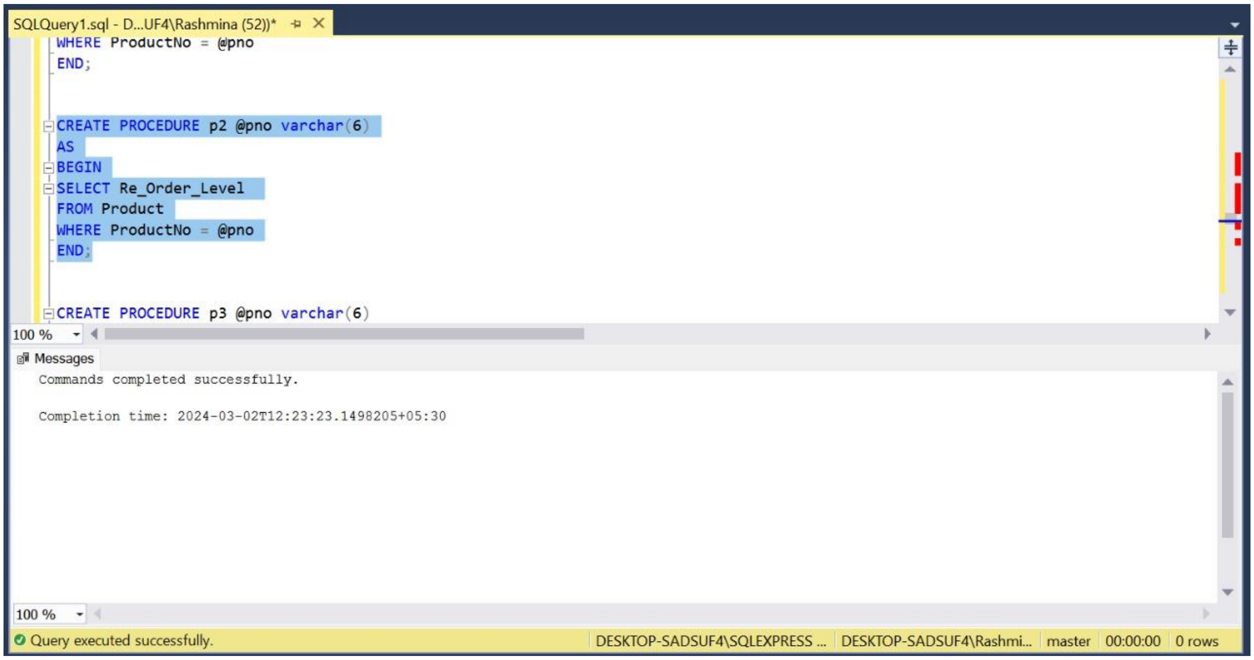
```
INNER JOIN Product
ON Product.ProductNo = Sales_Order_Details.Product_No;
SELECT *FROM view6;


CREATE PROCEDURE p1 @pno varchar(6)
AS
BEGIN
SELECT * FROM Product
WHERE ProductNo = @pno
END;


CREATE PROCEDURE p2 @pno varchar(6)
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:23:04.5345434+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows

2. Write a stored procedure to retrieve the Re_Order_Level for a given product number. Note: return the Re_Order_Level

```
CREATE PROCEDURE p2 @pno varchar(6)

AS

BEGIN

SELECT Re_Order_Level

FROM Product

WHERE ProductNo = @pno

END;
```



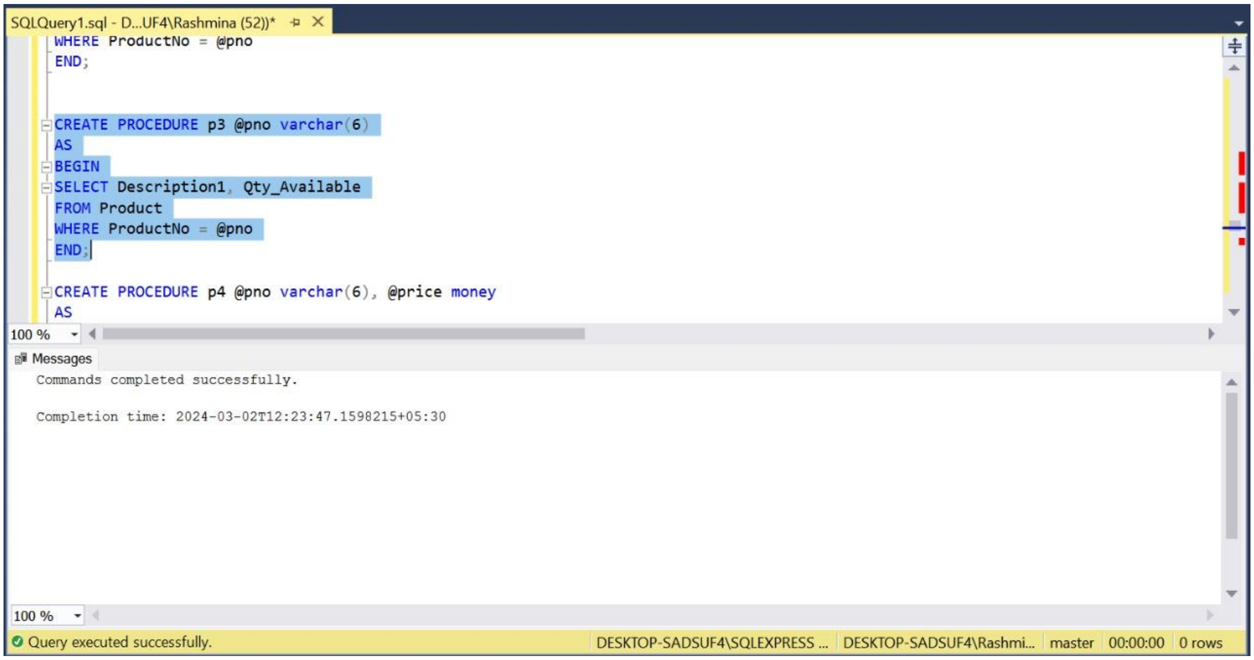
```
WHERE ProductNo = @pno
END;


CREATE PROCEDURE p2 @pno varchar(6)
AS
BEGIN
SELECT Re_Order_Level
FROM Product
WHERE ProductNo = @pno
END;


CREATE PROCEDURE p3 @pno varchar(6)
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:23:23.1498205+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows

3. Write a stored procedure to retrieve the Description and Qty_Available for a given product number. Note: retrieve the Description and Qty_Available as output parameters.

```
CREATE PROCEDURE p3 @pno
varchar(6)

AS

BEGIN

SELECT Description, Qty_Available

FROM Product

WHERE ProductNo = @pno

END;
```



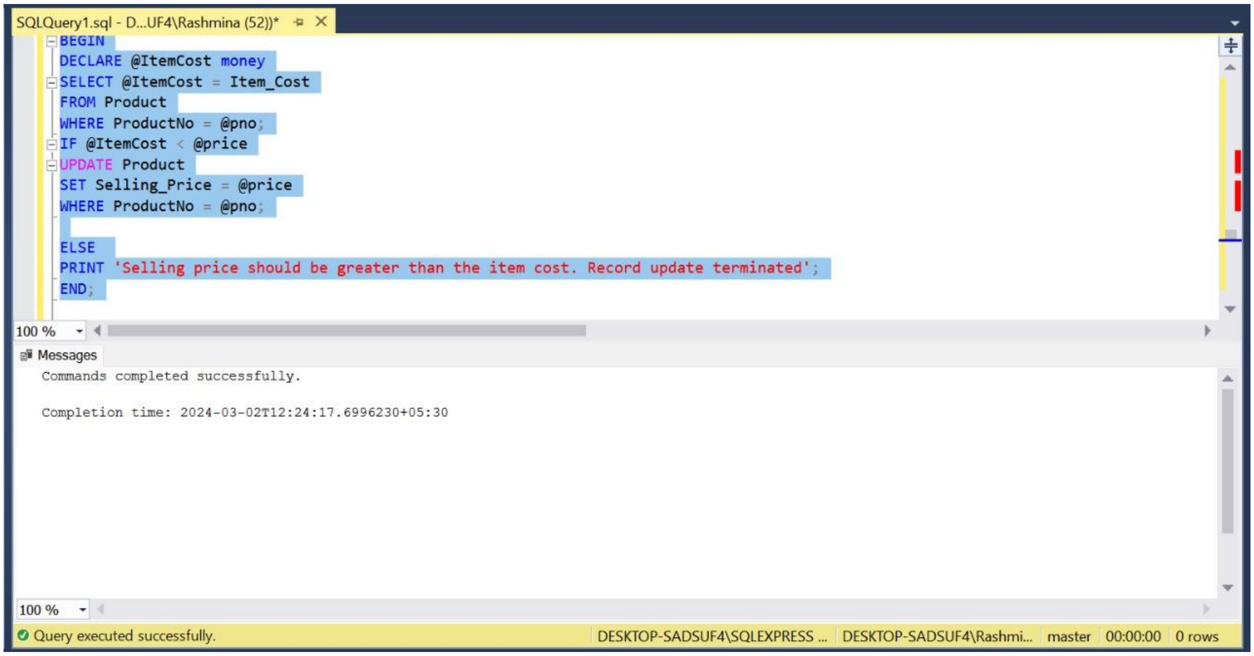
```
WHERE ProductNo = @pno
END;


CREATE PROCEDURE p3 @pno varchar(6)
AS
BEGIN
SELECT Description1, Qty_Available
FROM Product
WHERE ProductNo = @pno
END;

CREATE PROCEDURE p4 @pno varchar(6), @price money
AS
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:23:47.1598215+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows

4. Write a stored procedure to update the Selling_Price for a given ProductNo. Note: Selling_Price should be higher than the Item_Cost, otherwise display an error message called “Selling price should be greater than the item cost. Record update terminated”.

```
CREATE PROCEDURE p4 @pno
varchar(6), @price money

AS

BEGIN

DECLARE @ItemCost money

SELECT @ItemCost = Item_Cost

FROM Product

WHERE ProductNo = @pno;

IF @item < @price

            UPDATE Product

            SET Selling_Price = @price

            WHERE ProductNo = @pno;


ELSE

            PRINT ‘Selling price should be
greater than the item cost. Record
update terminated’;

END;
```



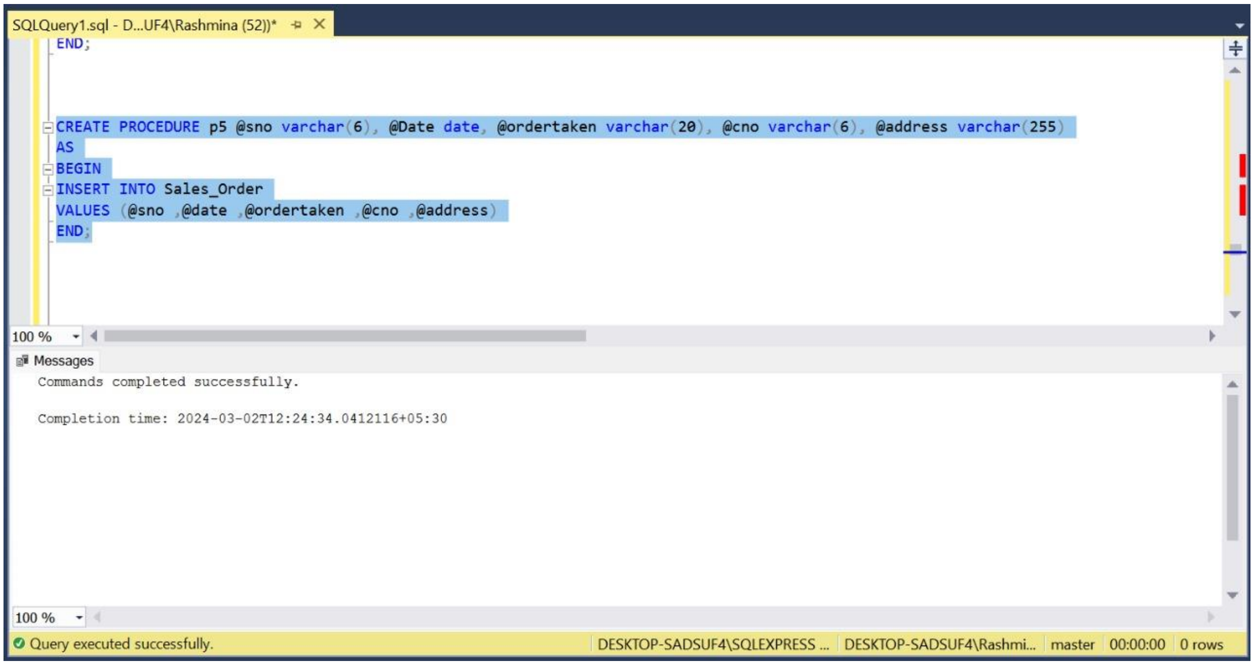
```
BEGIN
DECLARE @ItemCost money
SELECT @ItemCost = Item_Cost
FROM Product
WHERE ProductNo = @pno;
IF @ItemCost < @price
UPDATE Product
SET Selling_Price = @price
WHERE ProductNo = @pno;

ELSE
PRINT 'Selling price should be greater than the item cost. Record update terminated';
END;
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:24:17.6996230+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows

5. Write a stored procedure to insert a record to the Sales_Order table

```
CREATE PROCEDURE p5 @sno
varchar(6), @Date date, @ordertaken
varchar(20), @cno varchar(6),
@address varchar(255)

AS

BEGIN

INSERT INTO Sales_Order

VALUES (@sno ,@date ,@ordertaken
,@cno ,@address)

END;
```



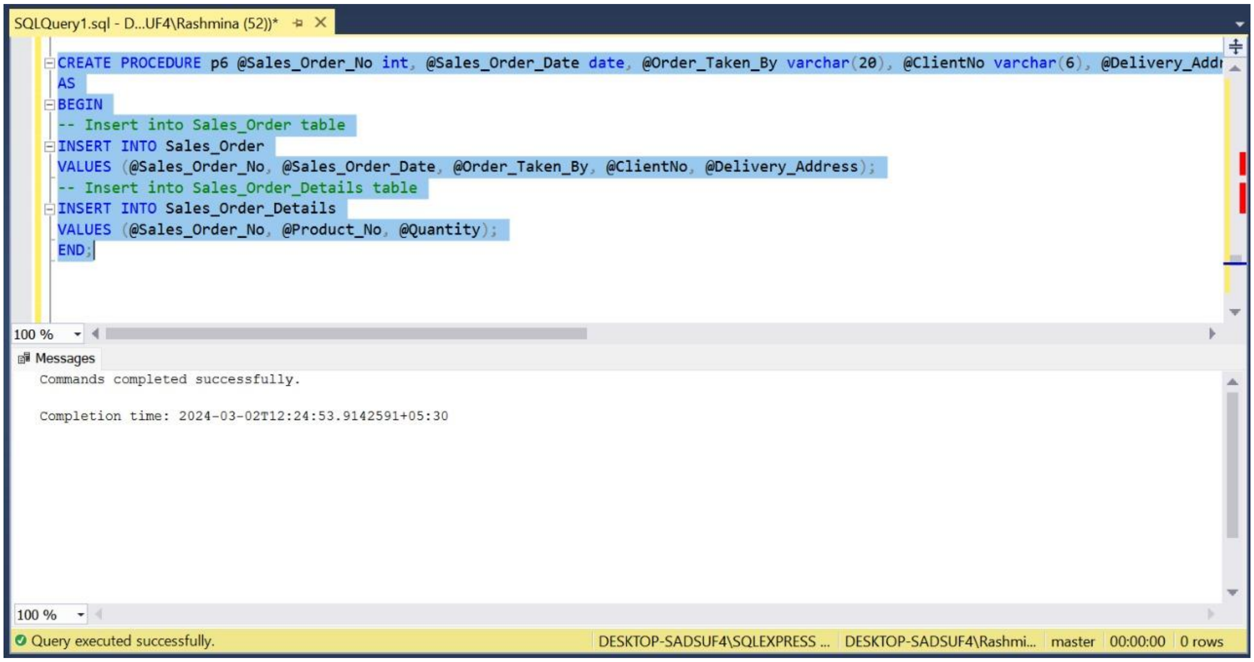
```
END;


CREATE PROCEDURE p5 @sno varchar(6), @Date date, @ordertaken varchar(20), @cno varchar(6), @address varchar(255)
AS
BEGIN
INSERT INTO Sales_Order
VALUES (@sno ,@date ,@ordertaken ,@cno ,@address)
END;
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:24:34.0412116+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows

6. Write a stored procedure to insert a record to the Sales_Order and Sales_Order_Details tables. Hint: The user will send the relevant data as input parameters as follows, Sales_Order_No, Sales_Order_Date, Order_Taken_By, ClientNo, Delivery_Address, Product_No, and Quantity.

```
CREATE PROCEDURE p6
@Sales_Order_No int,
@Sales_Order_Date date,
@Order_Taken_By varchar(20),
@ClientNo varchar(6),
@Delivery_Address varchar(255),
@Product_No varchar(6), @Quantity
int

AS

BEGIN

-- Insert into Sales_Order table

INSERT INTO Sales_Order

VALUES (@Sales_Order_No,
@Sales_Order_Date,
@Order_Taken_By, @ClientNo,
@Delivery_Address);

-- Insert into Sales_Order_Details
table

INSERT INTO Sales_Order_Details

VALUES (@Sales_Order_No,
@Product_No, @Quantity);

END;
```



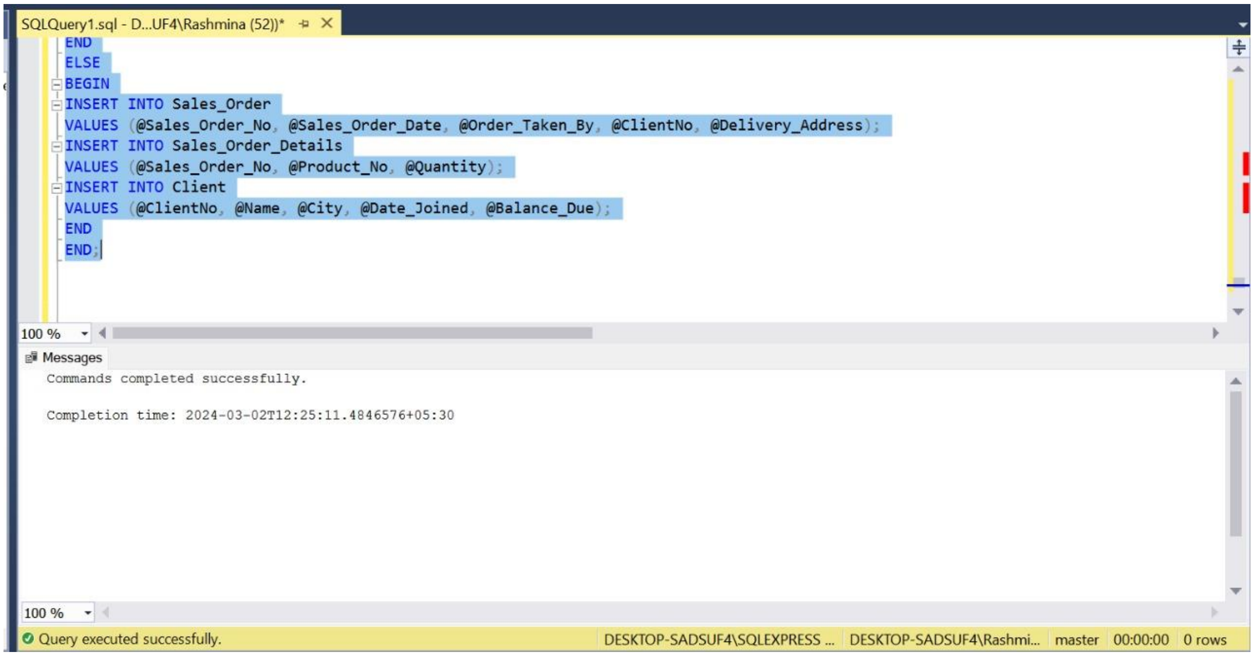
```
CREATE PROCEDURE p6 @Sales_Order_No int, @Sales_Order_Date date, @Order_Taken_By varchar(20), @ClientNo varchar(6), @Delivery_Addr
AS
BEGIN
-- Insert into Sales_Order table
INSERT INTO Sales_Order
VALUES (@Sales_Order_No, @Sales_Order_Date, @Order_Taken_By, @ClientNo, @Delivery_Address);
-- Insert into Sales_Order_Details table
INSERT INTO Sales_Order_Details
VALUES (@Sales_Order_No, @Product_No, @Quantity);
END;
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:24:53.9142591+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows

7. Write a stored procedure to insert a record to the Client, Sales_Order, and Sales_Ord

```sql
CREATE PROCEDURE p7 @ClientNo varchar(6), @Name varchar(20), @City varchar(20), @Date_Joined datetime,
@Balance_Due money, @Sales_Order_No int, @Sales_Order_Date date, @Order_Taken_By varchar(20), @Delivery_Address
varchar(255), @Product_No varchar(6),  @Quantity int

AS

BEGIN

DECLARE @cno varchar(6);

SELECT @cno =ClientNo

FROM Client;

IF @ClientNo = @cno

BEGIN

        INSERT INTO Sales_Order

        VALUES (@Sales_Order_No, @Sales_Order_Date, @Order_Taken_By, @ClientNo, @Delivery_Address);

        INSERT INTO Sales_Order_Details

        VALUES (@Sales_Order_No, @Product_No, @Quantity);

END

ELSE

BEGIN

        INSERT INTO Sales_Order

        VALUES (@Sales_Order_No, @Sales_Order_Date, @Order_Taken_By, @ClientNo, @Delivery_Address);

        INSERT INTO Sales_Order_Details

        VALUES (@Sales_Order_No, @Product_No, @Quantity);

        INSERT INTO Client

        VALUES (@ClientNo, @Name, @City, @Date_Joined, @Balance_Due);

END

END;
```



```
END
ELSE
BEGIN
INSERT INTO Sales_Order
VALUES (@Sales_Order_No, @Sales_Order_Date, @Order_Taken_By, @ClientNo, @Delivery_Address);
INSERT INTO Sales_Order_Details
VALUES (@Sales_Order_No, @Product_No, @Quantity);
INSERT INTO Client
VALUES (@ClientNo, @Name, @City, @Date_Joined, @Balance_Due);
END
END;
```

```
Commands completed successfully.

Completion time: 2024-03-02T12:25:11.4846576+05:30
```

Query executed successfully. | DESKTOP-SADSUF4\SQLEXPRESS ... | DESKTOP-SADSUF4\Rashmi... | master | 00:00:00 | 0 rows