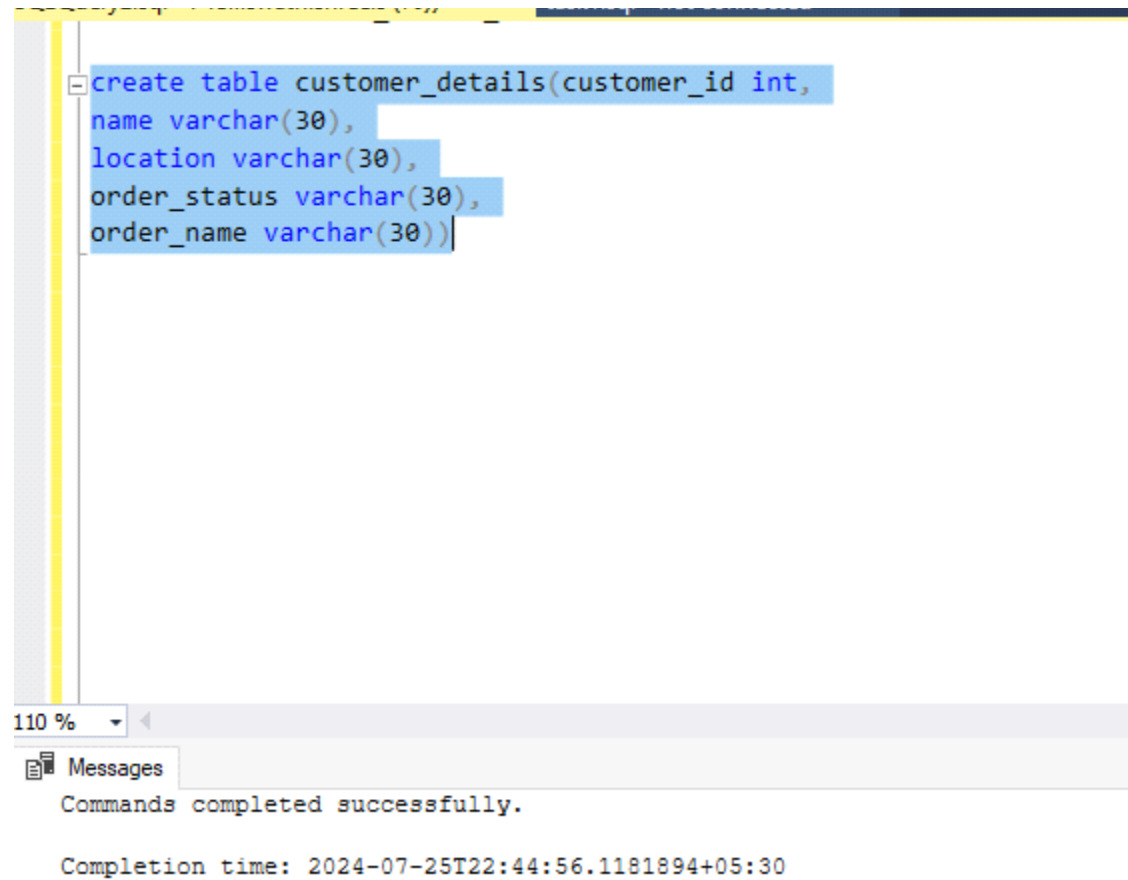


25/07/2024

Daily Task

Question 1:

Creating the table customer\_details



```
create table customer_details(customer_id int,  
name varchar(30),  
location varchar(30),  
order_status varchar(30),  
order_name varchar(30))
```

110 %

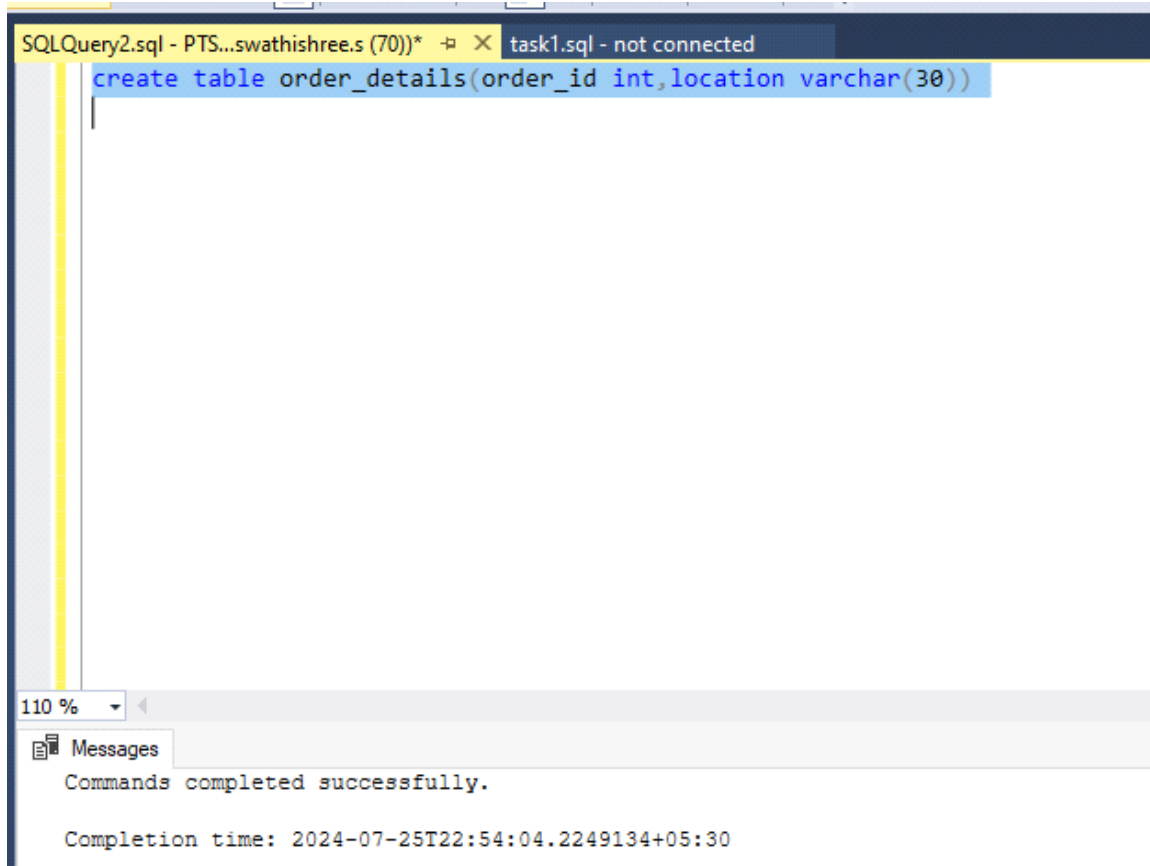
Messages

Commands completed successfully.

Completion time: 2024-07-25T22:44:56.1181894+05:30

Question 2:

Creating the table order details



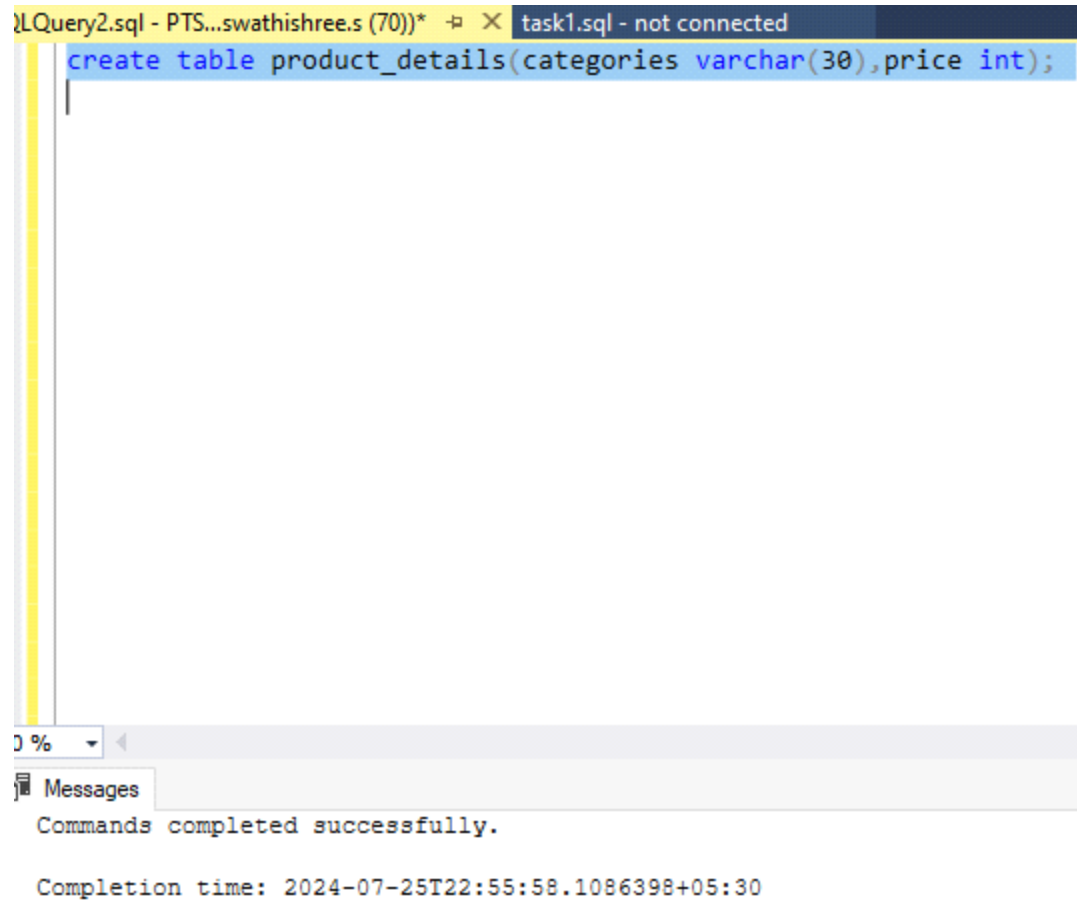
The screenshot shows a SQL query editor window with two tabs: 'SQLQuery2.sql - PTS...swathishree.s (70))' and 'task1.sql - not connected'. The active tab contains the SQL command: `create table order_details(order_id int,location varchar(30))`. Below the editor, a 'Messages' pane displays the output: 'Commands completed successfully.' and 'Completion time: 2024-07-25T22:54:04.2249134+05:30'. The editor's zoom level is set to 110%.

```
SQLQuery2.sql - PTS...swathishree.s (70))* × task1.sql - not connected
create table order_details(order_id int,location varchar(30))

110 %
Messages
Commands completed successfully.
Completion time: 2024-07-25T22:54:04.2249134+05:30
```

### Question 3

creating the table product details



The screenshot shows a SQL query editor window with two tabs: 'Query2.sql - PTS...swathishree.s (70))\*' and 'task1.sql - not connected'. The active tab contains the SQL command `create table product_details(categories varchar(30),price int);`. Below the editor, a 'Messages' pane displays the message 'Commands completed successfully.' and the completion time '2024-07-25T22:55:58.1086398+05:30'.

```
Query2.sql - PTS...swathishree.s (70))* task1.sql - not connected
create table product_details(categories varchar(30),price int);

0 %
Messages
Commands completed successfully.
Completion time: 2024-07-25T22:55:58.1086398+05:30
```

inserting the values in customer\_details table

```
SQLQuery2.sql - PTS...swathishree.s (70)) * task1.sql - not connected

insert into customer_details values(101,'swathi','Thanjavur','ordered','furniture'),
(102,'vishnu','Chennai','ordered','grossary');
insert into customer_details(customer_id,name,location,order_name)values
(103,'vani','kerala','stickers'),
(104,'subbaraj','kerala','electronics');

10 %
Messages

(2 rows affected)

Completion time: 2024-07-25T23:00:44.6685124+05:30
```

Question 4:

Fetch all customers and display if there are any order names

SQLQuery2.sql - PTS...swathishree.s (70))\* task1.sql - not connected

```
select * from customer_details  
where order_status='ordered';
```

110 %

Results Messages

	customer_id	name	location	order_status	order_name
1	101	swathi	Thanjavur	ordered	furniture
2	102	vishnu	Chennai	ordered	grossary

Question 5:

Fetch all the order names placed from particular location

SQLQuery2.sql - PTS...swathishree.s (70))\* task1.sql - not connected

```
select * from customer_details  
where location='kerala'
```

110 %

Results Messages

	customer_id	name	location	order_status	order_name
1	103	vani	kerala	NULL	stickers
2	104	subbaraj	kerala	NULL	electronics

Inserting the datas for the product\_details table

```
SQLQuery2.sql - PTS...swathishree.s (70))* × task1.sql - not connected
insert into product_details values('laptop',100000),('laptop',101121),('speaker',1013),('mobile',10000)

10 %
Messages
(4 rows affected)
Completion time: 2024-07-25T23:06:51.5403427+05:30
```

Question 6:

what is the max price of products from particular category

SQLQuery2.sql - PTS...swathishree.s (70)) \* task1.sql - not connected

```
select categories,max(price)from product_details  
group by categories
```

110 %

Results Messages

	categories	(No column name)
1	laptop	101121
2	mobile	10000
3	speaker	1013

question 7:

Display any product with the product name as like mobile,speaker



SQLQuery2.sql - PTS...swathishree.s (70)) \* task1.sql - not connected

```
select * from product_details  
where categories in('mobile','speaker')
```

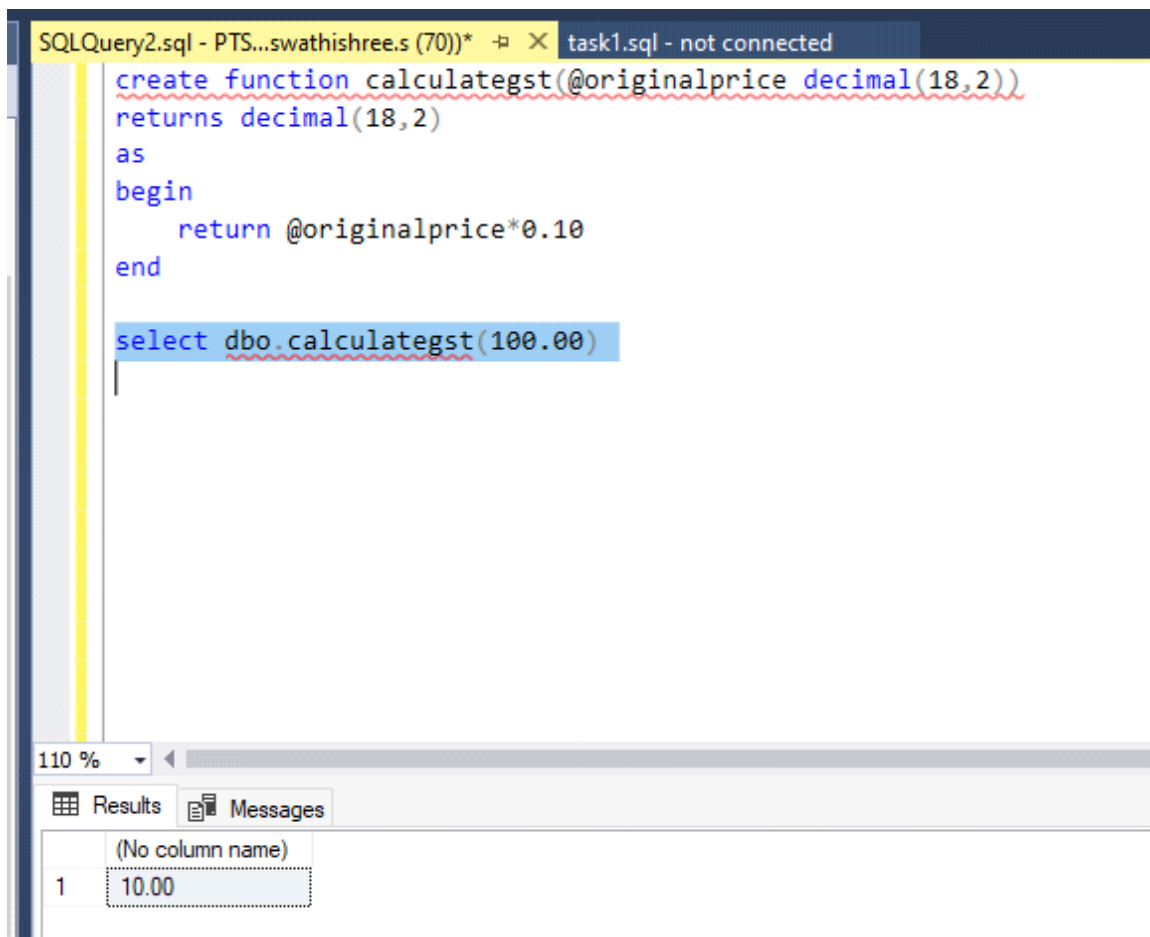
110 %

Results Messages

	categories	price
1	speaker	1013
2	mobile	10000

Question 8:

Create a function that calculate 10% gst from original price



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query in a script editor. The query defines a scalar function named `calculategst` that takes an `@originalprice` of type `decimal(18,2)` as input and returns a `decimal(18,2)` value. The function's logic is to return `@originalprice * 0.10`. Below the function definition, a `select` statement is used to call the function with the value `100.00`.

```
create function calculategst(@originalprice decimal(18,2))
returns decimal(18,2)
as
begin
    return @originalprice*0.10
end

select dbo.calculategst(100.00)
```

The bottom pane shows the results of the query. It includes a zoom level of 110% and tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row with the value 10.00. The column header is '(No column name)'.

	(No column name)
1	10.00

Question 9:

Create stored procedure that increases all the product price by 100

QLQuery2.sql - PTS...swathishree.s (70))<sup>\*</sup> task1.sql - not connected

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

10 %

Messages

Commands completed successfully.

Completion time: 2024-07-25T23:21:34.4432985+05:30

SQLQuery2.sql - PTS...swathishree.s (70))\* × task1.sql - not connected

```
INSERT INTO Products (ProductID, ProductName, Price)
VALUES
(1, 'Product A', 50.00),
(2, 'Product B', 75.00),
(3, 'Product C', 100.00);
```

110 %

Messages

(3 rows affected)

Completion time: 2024-07-25T23:22:46.1088799+05:30

SQLQuery2.sql - PTS...swathishree.s (70))\* task1.sql - not connected

```
CREATE PROCEDURE IncreaseAllPricesBy100
AS
BEGIN
    -- Update the Prices
    UPDATE Products
    SET Price = Price + 100;

    -- Return the updated prices for verification
    SELECT ProductID, ProductName, Price
    FROM Products;
END;

EXEC IncreaseAllPricesBy100;

SELECT * FROM Products;
```

110 %

Results Messages

	ProductID	ProductName	Price
1	1	Product A	150.00
2	2	Product B	175.00
3	3	Product C	200.00