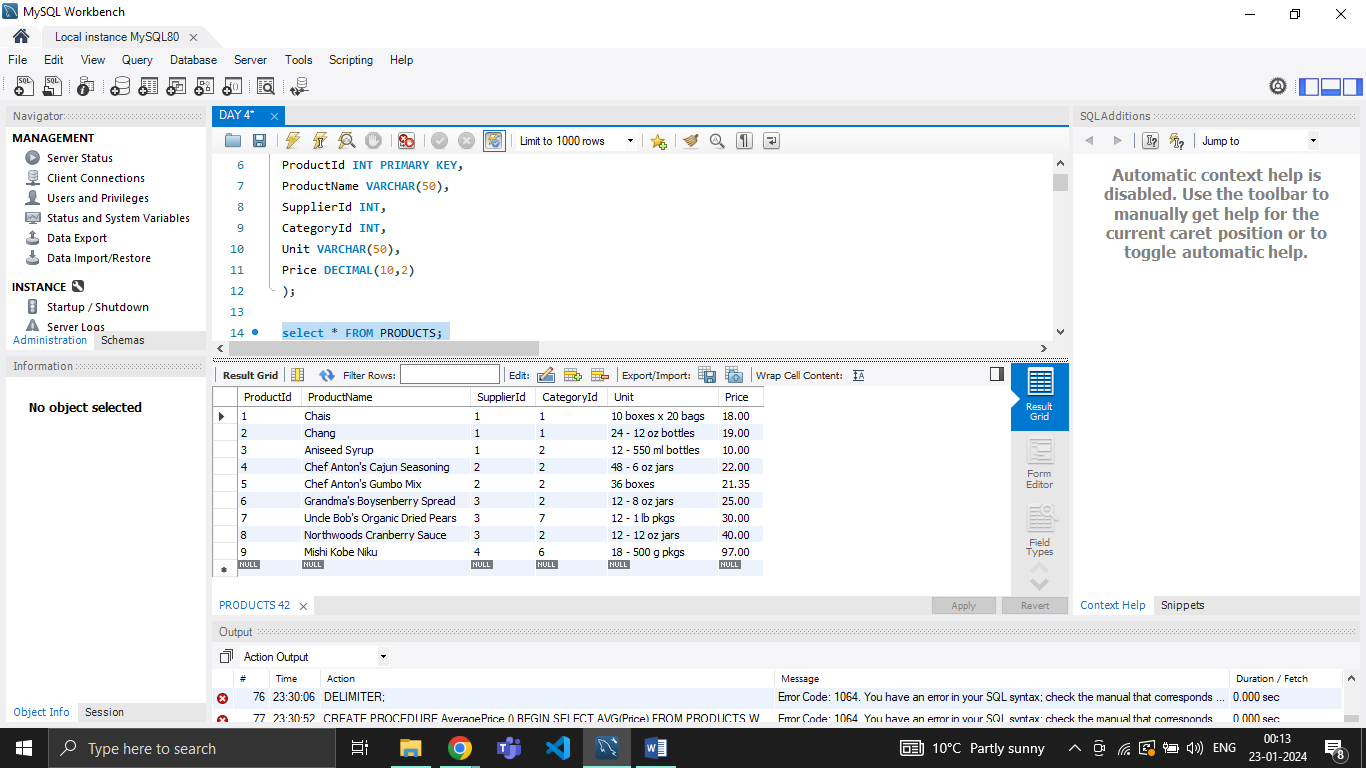
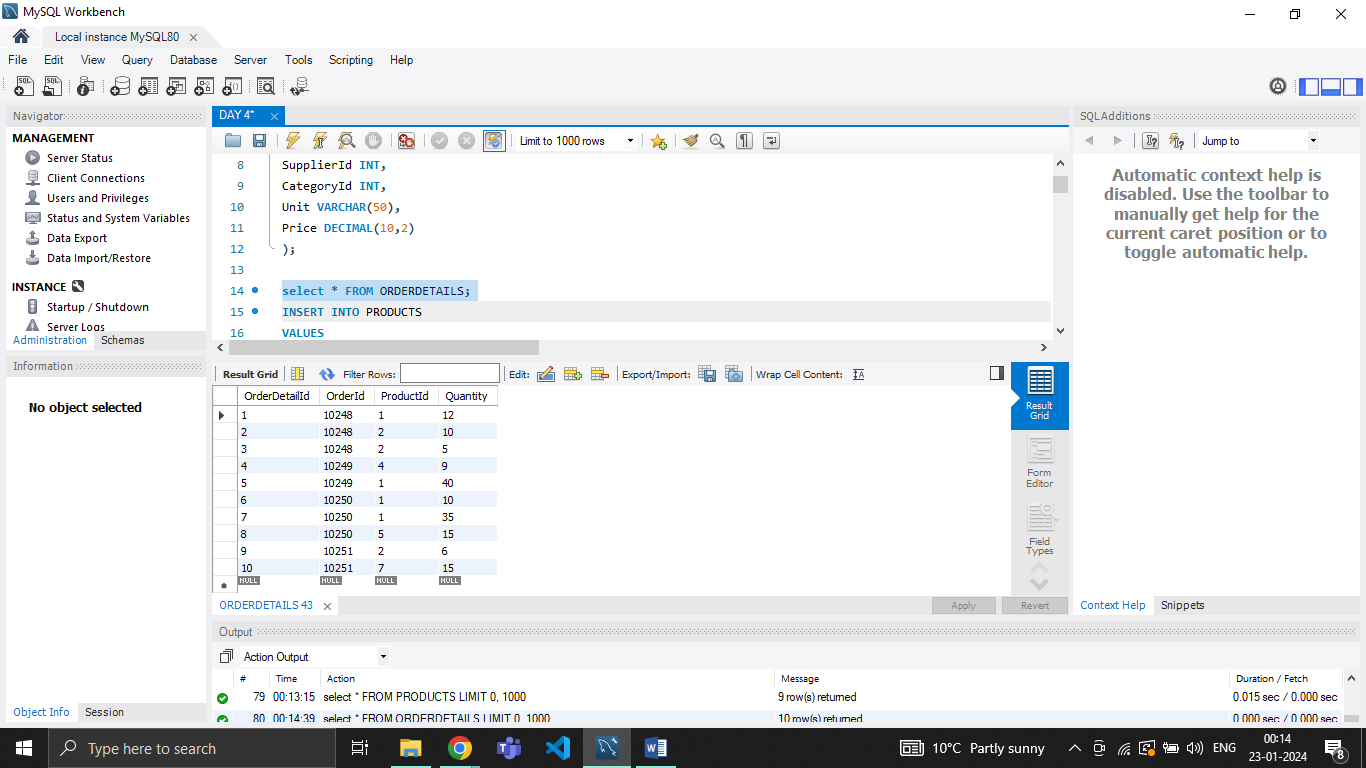
**VAISHALI BOKADIYA**

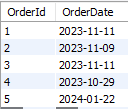
**DAY 4 ASSESSMENT**

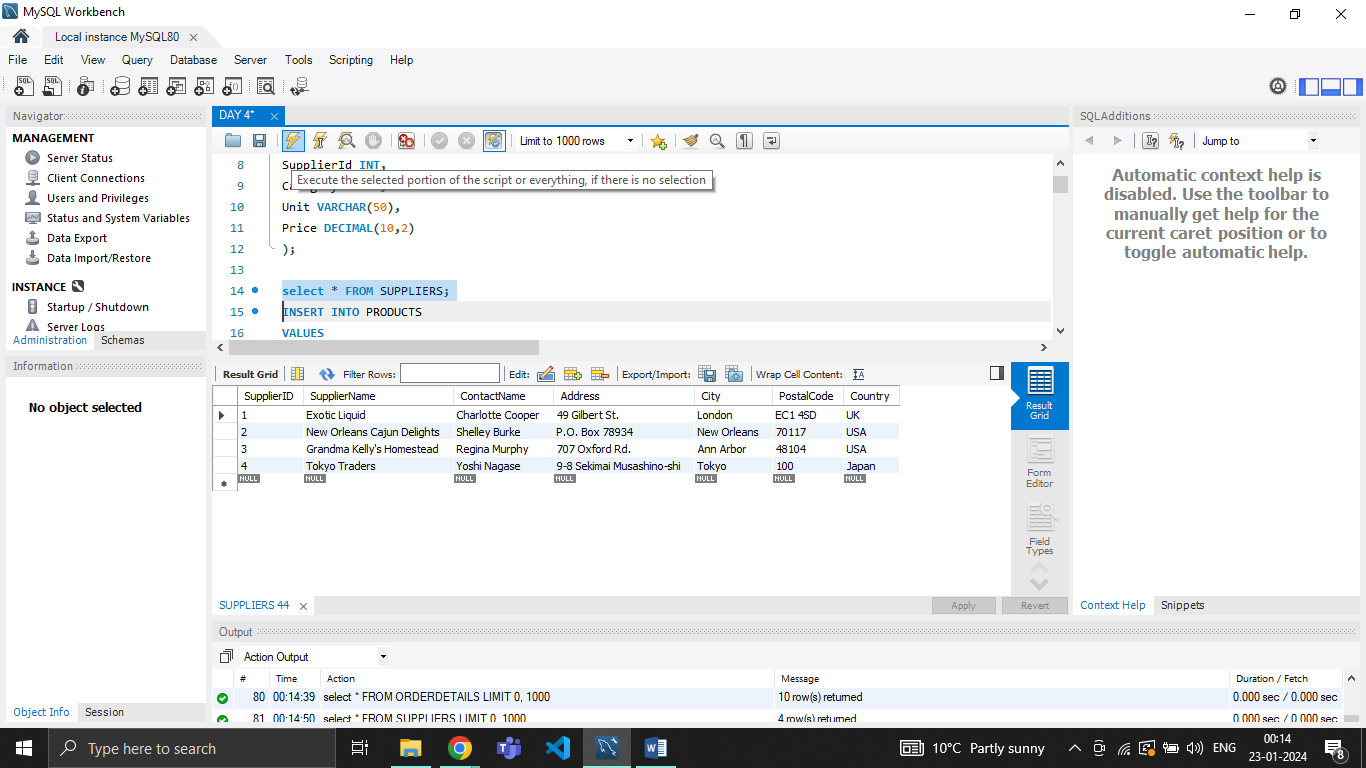
**SQL PRACTICE**

**DATABASE:**

****

****



****

**EXIST, ALL, ANY:**

**EXIST:**

SELECT SupplierName

FROM Suppliers

WHERE EXISTS

(SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);

OUTPUT:



**ANY:**

SELECT ProductName

FROM Products

WHERE ProductId = ANY

(SELECT ProductId

FROM OrderDetails

WHERE Quantity = 10);

OUTPUT:



**ALL:**

SELECT ProductName

FROM Products

WHERE Price > ALL

(SELECT Quantity

FROM OrderDetails);

OUTPUT:



**STRING FUNCTIONS:**

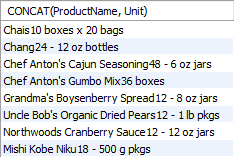
**CONCAT():**

SELECT CONCAT(ProductName, Unit)

FROM Products

WHERE Price > 15;

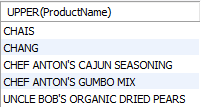
OUTPUT:



**UPPER():**

SELECT UPPER(ProductName) FROM PRODUCTS WHERE ProductId IN (SELECT ProductId FROM ORDERDETAILS);

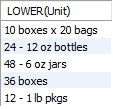
OUTPUT:



**LOWER():**

SELECT LOWER(Unit) FROM PRODUCTS WHERE ProductId IN (SELECT ProductId FROM ORDERDETAILS);

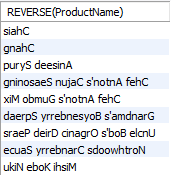
OUTPUT:



**REVERSE():**

SELECT REVERSE(ProductName) FROM PRODUCTS;

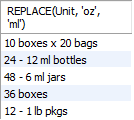
OUTPUT:



**REPLACE():**

SELECT REPLACE(Unit, 'oz', 'ml') FROM PRODUCTS WHERE ProductId IN (SELECT ProductId FROM ORDERDETAILS);

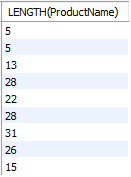
OUTPUT:



**LEN():**

SELECT LENGTH(ProductName) FROM PRODUCTS;

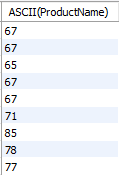
OUTPUT:



**ASCII():**

SELECT ASCII(ProductName) FROM PRODUCTS;

OUTPUT:



**MATHEMATICAL FUNCTIONS:**

**ABS():**

SELECT ABS(-66);

OUTPUT:



**SIN():**

SELECT SIN(1);

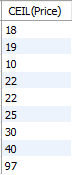
OUTPUT:



**CEIL():**

SELECT CEIL(Price) FROM PRODUCTS;

OUTPUT:



**EXP():**

SELECT EXP(5);

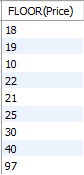
OUTPUT:



**FLOOR():**

SELECT FLOOR(Price) FROM PRODUCTS;

OUTPUT:



**LOG():**

SELECT LOG(100);

OUTPUT:



**AGGREGATE FUNCTIONS:**

**SUM():**

SELECT SUM(Quantity) FROM OrderDetails JOIN Products ON Products.ProductId=OrderDetails.ProductId WHERE Price>20;

OUTPUT:



**COUNT():**

SELECT COUNT(OrderDetailId) FROM OrderDetails JOIN Products ON Products.ProductId=OrderDetails.ProductId WHERE Price>20;

OUTPUT:



**AVG():**

SELECT AVG(Quantity) FROM OrderDetails JOIN Products ON Products.ProductId=OrderDetails.ProductId WHERE Price>20;

OUTPUT:



**MIN():**

SELECT MIN(Price) FROM PRODUCTS;

OUTPUT:



**MAX():**

SELECT MAX(Price) FROM PRODUCTS;

OUTPUT:



**REMAINING JOINS:**

**SELF JOIN:**

SELECT A.ProductId, B.ProductId, b.Price FROM PRODUCTS AS A

JOIN PRODUCTS AS B ON A.ProductId<>B.ProductId;

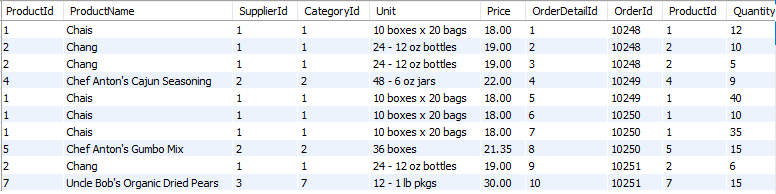
OUTPUT:

|  |  |  |
| --- | --- | --- |
| A.ProductId | B.ProductId | Price |
| 9 | 1 | 18.00 |
| 8 | 1 | 18.00 |
| 7 | 1 | 18.00 |
| 6 | 1 | 18.00 |
| 5 | 1 | 18.00 |
| 4 | 1 | 18.00 |
| 3 | 1 | 18.00 |
| 2 | 1 | 18.00 |
| 9 | 2 | 19.00 |
| 8 | 2 | 19.00 |
| 7 | 2 | 19.00 |
| 6 | 2 | 19.00 |
| 5 | 2 | 19.00 |
| 4 | 2 | 19.00 |
| 3 | 2 | 19.00 |
| 1 | 2 | 19.00 |
| 9 | 3 | 10.00 |
| 8 | 3 | 10.00 |
| 7 | 3 | 10.00 |
| 6 | 3 | 10.00 |
| 5 | 3 | 10.00 |
| 4 | 3 | 10.00 |
| 2 | 3 | 10.00 |
| 1 | 3 | 10.00 |
| 9 | 4 | 22.00 |
| 8 | 4 | 22.00 |
| 7 | 4 | 22.00 |
| 6 | 4 | 22.00 |
| 5 | 4 | 22.00 |
| 3 | 4 | 22.00 |
| 2 | 4 | 22.00 |
| 1 | 4 | 22.00 |
| 9 | 5 | 21.35 |
| 8 | 5 | 21.35 |
| 7 | 5 | 21.35 |
| 6 | 5 | 21.35 |
| 4 | 5 | 21.35 |
| 3 | 5 | 21.35 |
| 2 | 5 | 21.35 |
| 1 | 5 | 21.35 |
| 9 | 6 | 25.00 |
| 8 | 6 | 25.00 |
| 7 | 6 | 25.00 |
| 5 | 6 | 25.00 |
| 4 | 6 | 25.00 |
| 3 | 6 | 25.00 |
| 2 | 6 | 25.00 |
| 1 | 6 | 25.00 |
| 9 | 7 | 30.00 |
| 8 | 7 | 30.00 |
| 6 | 7 | 30.00 |
| 5 | 7 | 30.00 |
| 4 | 7 | 30.00 |
| 3 | 7 | 30.00 |
| 2 | 7 | 30.00 |
| 1 | 7 | 30.00 |
| 9 | 8 | 40.00 |
| 7 | 8 | 40.00 |
| 6 | 8 | 40.00 |
| 5 | 8 | 40.00 |
| 4 | 8 | 40.00 |
| 3 | 8 | 40.00 |
| 2 | 8 | 40.00 |
| 1 | 8 | 40.00 |
| 8 | 9 | 97.00 |
| 7 | 9 | 97.00 |
| 6 | 9 | 97.00 |
| 5 | 9 | 97.00 |
| 4 | 9 | 97.00 |
| 3 | 9 | 97.00 |
| 2 | 9 | 97.00 |
| 1 | 9 | 97.00 |

**CROSS JOIN:**

SELECT \* FROM PRODUCTS

CROSS JOIN ORDERDETAILS ON PRODUCTS.ProductId=ORDERDETAILS.ProductId;

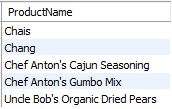


**OTHERS:**

**SUBQUERY:**

SELECT ProductName FROM PRODUCTS WHERE ProductId IN (SELECT ProductId FROM ORDERDETAILS);

OUTPUT:



**NESTED SUBQUERY:**

SELECT CategoryId, AVG(Price) FROM PRODUCTS

GROUP BY CategoryId

HAVING AVG(Price)<(

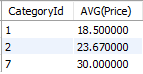
SELECT MAX(Price) FROM PRODUCTS WHERE SupplierId IN (

SELECT SupplierId FROM SUPPLIERS WHERE City='Tokyo'

)

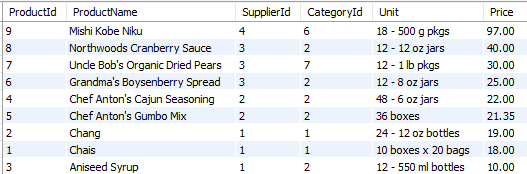
);

OUTPUT:



**ORDER BY:**

SELECT \* FROM PRODUCTS ORDER BY PRICE DESC;



**DATE FUNCTIONS:**

CREATE TABLE ORDERS(

OrderId INT PRIMARY KEY,

OrderDate date

);

INSERT INTO ORDERS

VALUES

(1, '2023-11-11'),

(2, '2023-11-09'),

(3, '2023-11-11'),

(4, '2023-10-29');

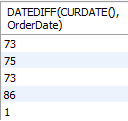
**GETDATE():**

INSERT INTO ORDERS VALUES (5, CURDATE());

**DATEDIFF():**

SELECT DATEDIFF(CURDATE(), OrderDate) FROM ORDERS;

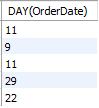
OUTPUT:



**DAY():**

SELECT DAY(OrderDate) FROM ORDERS;

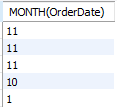
OUTPUT:



**MONTH():**

SELECT MONTH(OrderDate) FROM ORDERS;

OUTPUT:



**YEAR():**

SELECT YEAR(OrderDate) FROM ORDERS;

OUTPUT:

