

SQL Test Paper

APRIL 30

Mauritius Commercial Bank
Authored by: Budree Divraj



Qu1)

Create the “XXBCM_ORDER_MGT” table using the script - DB_Prequisite.sql

```
CREATE TABLE XXBCM_ORDER_MGT
(
    ORDER_REF VARCHAR2(2000),
    ORDER_DATE VARCHAR2(2000),
    SUPPLIER_NAME VARCHAR2(2000),
    SUPP_CONTACT_NAME VARCHAR2(2000),
    SUPP_ADDRESS VARCHAR2(2000),
    SUPP_CONTACT_NUMBER VARCHAR2(2000),
    SUPP_EMAIL VARCHAR2(2000),
    ORDER_TOTAL_AMOUNT VARCHAR2(2000),
    ORDER_DESCRIPTION VARCHAR2(2000),
    ORDER_STATUS VARCHAR2(2000),
    ORDER_LINE_AMOUNT VARCHAR2(2000),
    INVOICE_REFERENCE VARCHAR2(2000),
    INVOICE_DATE VARCHAR2(2000),
    INVOICE_STATUS VARCHAR2(2000),
    INVOICE_HOLD_REASON VARCHAR2(2000),
    INVOICE_AMOUNT VARCHAR2(2000),
    INVOICE_DESCRIPTION VARCHAR2(2000)
);
```

Output: Table XXBCM_ORDER_MGT created.

Qu2)

Based on the data provided implement a database schema with necessary tables, columns, data types and constraints. Create your tables in the database with appropriate naming convention. The tables should be appropriately normalized.

```
CREATE TABLE supplier (  
    SupplierID INT PRIMARY KEY,  
    SupplierName VARCHAR(100),  
    SupplierContactName VARCHAR(100),  
    SupplierAddress VARCHAR(200),  
    SupplierContactNumber VARCHAR(50),  
    SupplierEmail VARCHAR(100)  
);
```

```
CREATE TABLE orders (  
    OrderID INT IDENTITY(1,1) PRIMARY KEY,  
    OrderRef VARCHAR(50),  
    OrderDate DATETIME,  
    SupplierID INT,  
    OrderTotalAmount DECIMAL(10, 2),  
    OrderDescription VARCHAR(200),  
    OrderStatus VARCHAR(20)  
);
```

```
CREATE TABLE orderline (  
    OrderLineID INT IDENTITY(1,1) PRIMARY KEY,  
    OrderID INT,  
    OrderLineDescription VARCHAR(200),  
    OrderLineAmount DECIMAL(10, 2),  
    FOREIGN KEY (OrderID) REFERENCES orders(OrderID)  
);
```

```
CREATE TABLE invoices (  

```

```
InvoiceID INT IDENTITY(1,1) PRIMARY KEY,  
InvoiceRef VARCHAR(20),  
InvoiceDate DATE,  
InvoiceStatus VARCHAR(20),  
InvoiceHoldReason VARCHAR(100),  
InvoiceAmount DECIMAL(10, 2),  
InvoiceDescription VARCHAR(200),  
OrderID INT,  
FOREIGN KEY (OrderID) REFERENCES orders(OrderID)  
);
```

Output:

```
+ [table] dbo.invoices  
+ [table] dbo.orderline  
+ [table] dbo.orders  
+ [table] dbo.supplier  
+ [table] dbo.XXBCM_ORDER_MGT
```

3. Develop a SQL procedure to trigger a migration process that will extract information from table "XXBCM_ORDER_MGT" and load them in tables that you created with proper data format.

SQL Procedure:

```
CREATE OR ALTER PROCEDURE MigrateFromOrderMgt
AS
BEGIN
INSERT INTO supplier (SupplierID, SupplierName, SupplierContactName, SupplierAddress,
SupplierContactNumber, SupplierEmail)
SELECT
    ROW_NUMBER() OVER (ORDER BY Supplier_Name) AS SupplierID,
    Supplier_Name,
    Supp_Contact_Name,
    Supp_Address,
    Supp_Contact_Number,
    Supp_Email
FROM XXBCM_ORDER_MGT
    WHERE Supplier_Name IS NOT NULL;

INSERT INTO orders (OrderRef, OrderDate, SupplierID, OrderTotalAmount, OrderDescription, OrderStatus)
SELECT
    ORDER_REF,
    TRY_CAST(ORDER_DATE AS DATETIME) AS OrderDate,
    (SELECT TOP 1 SupplierID FROM supplier WHERE supplier.SupplierName =
XXBCM_ORDER_MGT.SUPPLIER_NAME) AS SupplierID,
    TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) AS OrderTotalAmount,
    ORDER_DESCRIPTION,
    ORDER_STATUS
FROM
    XXBCM_ORDER_MGT
WHERE
    CHARINDEX('-', ORDER_REF) = 0;
```

```

INSERT INTO orderline (OrderID, OrderLineDescription, OrderLineAmount)
SELECT
    o.OrderID,
    o.OrderDescription,
    ISNULL(TRY_CAST(om.ORDER_LINE_AMOUNT AS DECIMAL(10, 2)), 0) AS OrderLineAmount
FROM
    XXBCM_ORDER_MGT om
JOIN
    orders o
ON
    o.OrderID = (SELECT TOP 1 OrderID FROM orders WHERE orders.OrderRef = LEFT(om.ORDER_REF,
CHARINDEX('-', om.ORDER_REF) - 1))
WHERE
    CHARINDEX('-', om.ORDER_REF) > 0;

```

```

INSERT INTO invoices (InvoiceRef, InvoiceDate, InvoiceStatus, InvoiceHoldReason, InvoiceAmount,
InvoiceDescription, OrderID)
SELECT
    INVOICE_REFERENCE,
    TRY_CAST(INVOICE_DATE AS DATETIME) AS InvoiceDate,
    INVOICE_STATUS,
    INVOICE_HOLD_REASON,
    TRY_CAST(INVOICE_AMOUNT AS DECIMAL(10, 2)) AS InvoiceAmount,
    INVOICE_DESCRIPTION,

    (
        SELECT TOP 1 OrderID
        FROM orders
        WHERE CHARINDEX('-', ORDER_REF) > 0 AND
            OrderRef = SUBSTRING(ORDER_REF, 1, CHARINDEX('-', ORDER_REF) - 1) -- Ensure valid OrderRef
    ) AS OrderID

FROM
    XXBCM_ORDER_MGT
WHERE
    INVOICE_REFERENCE IS NOT NULL;
END;

```

Qu4)

The owner wishes to have a report displaying a summary of Orders with their corresponding list of distinct invoices and their total amount to be able to reconcile his orders and payments. The report shall contain the details as per table below ordered by latest Order Date on top. Implement a Stored Procedure or Function to return the required information.

2 functions: one select from table XXBCM_ORDER_MGT and another function select from 3 tables orders, invoices and supplier joining them.

```
SELECT
CASE
WHEN ISNUMERIC(SUBSTRING(Order_REF, 3, LEN(Order_REF) - 2)) = 1
THEN CAST(SUBSTRING(Order_REF, 3, LEN(Order_REF) - 2) AS INT)
ELSE NULL
END AS OrderReference ,
CASE
WHEN TRY_CAST(ORDER_DATE AS DATETIME) IS NOT NULL
THEN
UPPER(LEFT(DATENAME(MONTH, TRY_CAST(ORDER_DATE AS DATETIME)), 3)) + '-' +
CAST(DATEPART(YEAR, TRY_CAST(ORDER_DATE AS DATETIME)) AS VARCHAR(4))
ELSE 'INVALID'
END AS OrderPeriod,

UPPER(LEFT(SUPPLIER_NAME, 1)) + LOWER(SUBSTRING(SUPPLIER_NAME, 2,
LEN(SUPPLIER_NAME))) AS SupplierName,

CASE
WHEN TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) IS NOT NULL
THEN FORMAT(TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)), 'N2')
ELSE 'Invalid Data'
END AS OrderTotalAmount,
```

ORDER_STATUS,

INVOICE_REFERENCE,

CASE
WHEN TRY_CAST(INVOICE_AMOUNT AS DECIMAL(10, 2)) IS NOT NULL
THEN FORMAT(TRY_CAST(INVOICE_AMOUNT AS DECIMAL(10, 2)), 'N2')
ELSE 'Invalid Data'
END AS InvoiceTotalAmount,

CASE
WHEN INVOICE_STATUS = 'Paid' THEN 'OK'
WHEN INVOICE_STATUS = 'Pending' THEN 'To follow up'
WHEN INVOICE_STATUS IS NULL OR INVOICE_STATUS = '' THEN 'To verify'
ELSE 'To verify'
END AS Action

FROM XXBCM_ORDER_MGT;

=====

SELECT

-- Extract numeric value from Order Reference (excluding 'PO')

CASE
WHEN ISNUMERIC(SUBSTRING(o.OrderRef, 3, LEN(o.OrderRef) - 2)) = 1
THEN CAST(SUBSTRING(o.OrderRef, 3, LEN(o.OrderRef) - 2) AS INT)
ELSE NULL
END AS OrderReference,

CASE
WHEN TRY_CAST(o.OrderDate AS DATETIME) IS NOT NULL
THEN
UPPER(LEFT(DATENAME(MONTH, TRY_CAST(o.OrderDate AS DATETIME)), 3)) + '-' +
CAST(DATEPART(YEAR, TRY_CAST(o.OrderDate AS DATETIME)) AS VARCHAR(4))
ELSE 'INVALID'
END AS OrderPeriod,

UPPER(LEFT(s.SupplierName, 1)) + LOWER(SUBSTRING(s.SupplierName, 2, LEN(s.SupplierName))) AS
SupplierName,

```
CASE
  WHEN TRY_CAST(o.OrderTotalAmount AS DECIMAL(10, 2)) IS NOT NULL
  THEN FORMAT(TRY_CAST(o.OrderTotalAmount AS DECIMAL(10, 2)), 'N2')
  ELSE 'Invalid Data'
END AS OrderTotalAmount,
```

```
o.OrderStatus,
```

```
i.InvoiceRef,
```

```
CASE
  WHEN TRY_CAST(i.InvoiceAmount AS DECIMAL(10, 2)) IS NOT NULL
  THEN FORMAT(TRY_CAST(i.InvoiceAmount AS DECIMAL(10, 2)), 'N2')
  ELSE 'Invalid Data'
END AS InvoiceTotalAmount,
```

```
CASE
  WHEN i.InvoiceStatus = 'Paid' THEN 'OK'
  WHEN i.InvoiceStatus = 'Pending' THEN 'To follow up'
  WHEN i.InvoiceStatus IS NULL OR i.InvoiceStatus = '' THEN 'To verify'
  ELSE 'To verify'
END AS Action
```

```
FROM
  orders o -- Base table
INNER JOIN
  supplier s
ON
  o.SupplierID = s.SupplierID
LEFT JOIN
  invoices i
ON
  o.OrderID = i.OrderID
```

Output:

110 %

Results Messages

	OrderReference	OrderPeriod	SupplierName	OrderTotalAmount	OrderStatus	InvoiceRef	InvoiceTotalAmount	Action
1	1	JAN-2022	Pegasus ltd	Invalid Data	Closed	INV_PO001	Invalid Data	OK
2	1	JAN-2022	Pegasus ltd	Invalid Data	Closed	INV_PO001	600.00	OK
3	1	JAN-2022	Pegasus ltd	Invalid Data	Closed	INV_PO001	Invalid Data	OK
4	1	JAN-2022	Pegasus ltd	Invalid Data	Closed	INV_PO001	800.00	OK
5	1	JAN-2022	Pegasus ltd	Invalid Data	Closed	INV_PO001.1	3,000.00	OK
6	1	JAN-2022	Pegasus ltd	Invalid Data	Closed	INV_PO001.1	1,100.00	OK
7	2	JAN-2022	Mottoway corp.	750,000.00	Open	INV_PO002.1	100,000.00	OK
8	2	JAN-2022	Mottoway corp.	750,000.00	Open	INV_PO002.2	Invalid Data	OK
9	2	JAN-2022	Mottoway corp.	750,000.00	Open	INV_PO002.3	249,000.00	To follow up
10	3	JAN-2022	Disisay co. ltd.	57,300.00	Closed	INV_PO003.1	Invalid Data	OK
11	3	JAN-2022	Disisay co. ltd.	57,300.00	Closed	INV_PO003.1	20,300.00	OK
12	3	JAN-2022	Disisay co. ltd.	57,300.00	Closed	INV_PO003.1	5,000.00	OK
13	3	JAN-2022	Disisay co. ltd.	57,300.00	Closed	INV_PO003.2	Invalid Data	OK
14	4	FEB-2022	Lamboni stat inc.	6,800.00	Closed	INV_PO004	Invalid Data	OK
15	4	FEB-2022	Lamboni stat inc.	6,800.00	Closed	INV_PO004	Invalid Data	OK
16	4	FEB-2022	Lamboni stat inc.	6,800.00	Closed	INV_PO004.1	3,000.00	OK
17	5	APR-2022	Ertello ltd	21,000.00	Closed	INV_PO005.1	Invalid Data	To follow up
18	5	APR-2022	Ertello ltd	21,000.00	Closed	INV_PO005.1	5,000.00	OK
19	5	APR-2022	Ertello ltd	21,000.00	Closed	INV_PO005.2	8,000.00	OK
20	5	APR-2022	Ertello ltd	21,000.00	Closed	INV_PO005.2	Invalid Data	OK
21	6	FEB-2022	Stuffie stationery	250,000.00	Open	INV_PO006.1	24,300.00	OK
22	6	FEB-2022	Stuffie stationery	250,000.00	Open	INV_PO006.3	Invalid Data	OK

Query executed successfully.

DESKTOP-C

Qu5)

Return details for the SECOND (2nd) highest Order Total Amount from the list. Only one record is expected with the following information. Implement a Stored Procedure or Function to return the required information.

2 functions: one select from table XXBCM_ORDER_MGT and another function select from 3 tables orders, invoices and supplier joining them.

-- Find the second-highest order total amount

-- Retrieve the second-highest order total

SELECT TOP 1
OrderTotalAmount

FROM (

SELECT DISTINCT TOP 2

```

    TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) AS OrderTotalAmount
FROM
    XXBCM_ORDER_MGT
ORDER BY
    OrderTotalAmount DESC
) AS SecondHighest
ORDER BY
    OrderTotalAmount ASC; -- Get the second-highest by ordering in ascending order

-- Now retrieve the details of the record with this order total
SELECT
    -- Extract numeric value from Order Reference (without 'PO')
    CASE
        WHEN ISNUMERIC(SUBSTRING(ORDER_REF, 3, LEN(Order_REF) - 2)) = 1
        THEN CAST(SUBSTRING(Order_REF, 3, LEN(Order_REF) - 2) AS INT)
        ELSE NULL
    END AS OrderReference,

    -- Convert Order Date to "January 2022"
    CASE
        WHEN TRY_CAST(ORDER_DATE AS DATETIME) IS NOT NULL
        THEN
            DATENAME(MONTH, TRY_CAST(ORDER_DATE AS DATETIME)) + ' ' +
            CAST(DATEPART(YEAR, TRY_CAST(ORDER_DATE AS DATETIME)) AS VARCHAR(4)) -- Concatenate month
        and year
        ELSE 'INVALID' -- Handle invalid or NULL date cases
    END AS OrderDate,

    -- Convert Supplier Name to uppercase
    UPPER(SUPPLIER_NAME) AS SupplierName,

    -- Format Order Total Amount with commas and decimal places
    CASE
        WHEN TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) IS NOT NULL
        THEN FORMAT(TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)), 'N2')
        ELSE 'Invalid Data'
    END AS OrderTotalAmount,

    ORDER_STATUS,

    -- Combine invoice references for the same order (pipe-delimited)
    (SELECT STRING_AGG(INVOICE_REFERENCE, '|')
```

```
FROM XXBCM_ORDER_MGT
WHERE ORDER_REF = XXBCM_ORDER_MGT.ORDER_REF) AS InvoiceReferences
```

```
FROM
  XXBCM_ORDER_MGT
WHERE
  TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) = (
    SELECT TOP 1 OrderTotalAmount
    FROM (
      SELECT DISTINCT TOP 2
        TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) AS OrderTotalAmount
      FROM
        XXBCM_ORDER_MGT
      ORDER BY
        OrderTotalAmount DESC
    ) AS SecondHighest
    ORDER BY
      OrderTotalAmount ASC
  );
```

```
=====
```

```
-- Common Table Expression (CTE) to determine the second-highest order total
WITH SecondHighest AS (
  SELECT
    DISTINCT TOP 2
    TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2)) AS OrderTotalAmount
  FROM
    XXBCM_ORDER_MGT
  ORDER BY
    OrderTotalAmount DESC
)
-- Retrieve the second-highest unique order total
SELECT TOP 1
  OrderTotalAmount
FROM
  SecondHighest
ORDER BY
  OrderTotalAmount ASC; -- Get the second-highest by ordering in ascending order

-- Now retrieve the unique record corresponding to this order total
SELECT
  -- Extract numeric value from Order Reference (without 'PO')
```

```

CASE
  WHEN ISNUMERIC(SUBSTRING(o.OrderRef, 3, LEN(o.OrderRef) - 2)) = 1
  THEN CAST(SUBSTRING(o.OrderRef, 3, LEN(o.OrderRef) - 2) AS INT)
  ELSE NULL
END AS OrderReference,

-- Convert Order Date to "January 2022"
CASE
  WHEN TRY_CAST(o.OrderDate AS DATETIME) IS NOT NULL
  THEN
    DATENAME(MONTH, TRY_CAST(o.OrderDate AS DATETIME)) + ' ' +
    CAST(DATEPART(YEAR, TRY_CAST(o.OrderDate AS DATETIME)) AS VARCHAR(4)) -- Concatenate month
and year
  ELSE 'INVALID'
END AS OrderDate,

-- Supplier name in title case
UPPER(LEFT(s.SupplierName, 1)) + LOWER(SUBSTRING(s.SupplierName, 2, LEN(s.SupplierName))) AS
SupplierName,

-- Format Order Total Amount with commas and decimal places
CASE
  WHEN TRY_CAST(o.OrderTotalAmount AS DECIMAL(10, 2)) IS NOT NULL
  THEN FORMAT(TRY_CAST(o.OrderTotalAmount AS DECIMAL(10, 2)), 'N2')
  ELSE 'Invalid Data'
END AS OrderTotalAmount,

o.OrderStatus, -- Return order status as-is

-- Retrieve invoice references for the specific order (pipe-delimited)
(SELECT STRING_AGG(i.InvoiceRef, '|')
FROM invoices i
WHERE i.OrderID = o.OrderID) AS InvoiceReferences

FROM
  orders o -- Base table
INNER JOIN
  supplier s
ON
  o.SupplierID = s.SupplierID -- Join orders to suppliers
LEFT JOIN
  invoices i
ON

```

o.OrderID = i.OrderID -- Join orders to invoices (with potential duplicates)

WHERE

**TRY_CAST(o.OrderTotalAmount AS DECIMAL(10, 2)) = (
SELECT TOP 1 OrderTotalAmount
FROM
SecondHighest -- Subquery returning the second-highest order total
)**

GROUP BY

**o.OrderRef, -- Ensures grouping by unique order reference
o.OrderDate,
s.SupplierName,
o.OrderTotalAmount,
o.OrderID,
o.OrderStatus;**

Output:

110 %	
Results Messages	
1	OrderTotalAmount 265000.00

	OrderReference	OrderDate	SupplierName	OrderTotalAmount	OrderStatus	InvoiceReferences
1	12	August 2022	Pegasus Ltd	265,000.00	Open	INV_PO012.1 INV_PO012.2 INV_PO012.1 INV_PO012.2 INV_PO012.3 INV_PO012.1 INV_PO012.2 INV_PO012.3 INV_PO012.3

Qu6)

List all suppliers with their respective number of orders and total amount ordered from them between the period of 01 January 2022 and 31 August 2022. Output details as per below. Implement a Stored Procedure or Function to return the required information.

2 functions: one select from table XXBCM_ORDER_MGT and another function select from 3 tables orders, invoices and supplier joining them.

SELECT

SUPPLIER_NAME AS SupplierName,

SUPP_CONTACT_NAME AS SupplierContactName,

CASE

WHEN CHARINDEX(',', SUPP_CONTACT_NUMBER) > 0

THEN

CASE

WHEN LEN(SUBSTRING(SUPP_CONTACT_NUMBER, 1, CHARINDEX(',', SUPP_CONTACT_NUMBER) - 1)) =

7

THEN

SUBSTRING(SUPP_CONTACT_NUMBER, 1, CHARINDEX(',', SUPP_CONTACT_NUMBER) - 1)

ELSE

SUBSTRING(SUPP_CONTACT_NUMBER, 2, CHARINDEX(',', SUPP_CONTACT_NUMBER) - 2)

END

ELSE

SUPP_CONTACT_NUMBER -- If there's no comma, return the full contact number

END AS SupplierContactNo1,

CASE

WHEN CHARINDEX(',', SUPP_CONTACT_NUMBER) > 0

THEN

CASE

WHEN LEN(SUBSTRING(SUPP_CONTACT_NUMBER, CHARINDEX(',', SUPP_CONTACT_NUMBER) + 1, LEN(SUPP_CONTACT_NUMBER))) = 7

```

        THEN
            SUBSTRING(SUPP_CONTACT_NUMBER, CHARINDEX(',', SUPP_CONTACT_NUMBER) + 1,
LEN(SUPP_CONTACT_NUMBER))
        ELSE
            SUBSTRING(SUPP_CONTACT_NUMBER, CHARINDEX(',', SUPP_CONTACT_NUMBER) + 2,
LEN(SUPP_CONTACT_NUMBER))
        END
    ELSE
        NULL -- If there's no second contact number, return NULL
    END AS SupplierContactNo2,

    -- Total number of orders for the supplier
    COUNT(DISTINCT ORDER_REF) AS TotalOrders,

    -- Total amount ordered from the supplier
    FORMAT(SUM(TRY_CAST(ORDER_TOTAL_AMOUNT AS DECIMAL(10, 2))), '##,###,###.00') AS
OrderTotalAmount
FROM
    XXBCM_ORDER_MGT
WHERE
    -- Convert ORDER_DATE to DATETIME and check the date range
    TRY_CAST(ORDER_DATE AS DATETIME) BETWEEN TRY_CAST('2022-01-01' AS DATETIME) AND
TRY_CAST('2022-08-31' AS DATETIME)
GROUP BY
    SUPPLIER_NAME, SUPP_CONTACT_NAME, SUPP_CONTACT_NUMBER
ORDER BY
    SupplierName; -- Order by Supplier Name

=====

SELECT
    -- Supplier Name
    s.SupplierName AS SupplierName,

    -- Supplier Contact Name
    s.SupplierContactName AS SupplierContactName,

    -- Format the first contact number ("999-9999" or "5999-9999")
    CASE
        WHEN CHARINDEX(',', s.SupplierContactNumber) > 0
        THEN
            CASE
                WHEN LEN(SUBSTRING(s.SupplierContactNumber, 1, CHARINDEX(',', s.SupplierContactNumber) - 1)) = 7

```

```

    THEN
        SUBSTRING(s.SupplierContactNumber, 1, CHARINDEX(',', s.SupplierContactNumber) - 1)
    ELSE
        SUBSTRING(s.SupplierContactNumber, 2, CHARINDEX(',', s.SupplierContactNumber) - 2)
    END
ELSE
    s.SupplierContactNumber -- If there's no comma, return the full contact number
END AS SupplierContactNo1,

-- Format the second contact number ("999-9999" or "5999-9999")
CASE
    WHEN CHARINDEX(',', s.SupplierContactNumber) > 0
    THEN
        CASE
            WHEN LEN(SUBSTRING(s.SupplierContactNumber, CHARINDEX(',', s.SupplierContactNumber) + 1,
LEN(s.SupplierContactNumber))) = 7
            THEN
                SUBSTRING(s.SupplierContactNumber, CHARINDEX(',', s.SupplierContactNumber) + 1,
LEN(s.SupplierContactNumber))
            ELSE
                SUBSTRING(s.SupplierContactNumber, CHARINDEX(',', s.SupplierContactNumber) + 2,
LEN(s.SupplierContactNumber))
            END
        ELSE
            NULL -- If there's no second contact number, return NULL
        END AS SupplierContactNo2,

-- Total number of orders for the supplier
COUNT(DISTINCT o.OrderRef) AS TotalOrders,

-- Total amount ordered from the supplier
FORMAT(SUM(TRY_CAST(o.OrderTotalAmount AS DECIMAL(10, 2))), '##,###,###.00') AS
OrderTotalAmount,

-- Pipe-delimited invoice references
(SELECT STRING_AGG(i.InvoiceRef, '|')
FROM invoices i
WHERE i.OrderID = o.OrderID) AS InvoiceReferences,

-- Add the OrderID to GROUP BY
o.OrderID

FROM

```

```

orders o -- Base table
INNER JOIN
  supplier s
ON
  o.SupplierID = s.SupplierID -- Join orders to suppliers
LEFT JOIN
  invoices i
ON
  o.OrderID = i.OrderID -- Join orders to invoices

WHERE
  -- Convert ORDER_DATE to DATETIME and check the date range
  TRY_CAST(o.OrderDate AS DATETIME) BETWEEN TRY_CAST('2022-01-01' AS DATETIME) AND
  TRY_CAST('2022-08-31' AS DATETIME)

GROUP BY
  s.SupplierName,
  s.SupplierContactName,
  s.SupplierContactNumber,
  o.OrderRef,
  o.OrderTotalAmount,
  o.OrderDate,
  o.OrderStatus,
  o.OrderID -- Include OrderID in GROUP BY to avoid the error

ORDER BY
  s.SupplierName; -- Order by Supplier Name

```

Output:

	SupplierName	SupplierContactName	SupplierContactNo1	SupplierContactNo2	TotalOrders	OrderTotalAmount	InvoiceReferences	OrderID
1	DIGISAY CO. LTD.	Berry Parker	7841266	602801o	1	229,200.00	INV_PO003.1 INV_PO003.1 INV_PO003.1 INV_PO003.2	3
2	EMTELLO LTD	Megan Hembly	2420641	57841698.	1	84,000.00	INV_PO005.1 INV_PO005.1 INV_PO005.2 INV_PO005.2	5
3	FIRELAND BROS.	Amelia Bridney	5948 0015	NULL	1	36,800.00	INV_PO009.1	9
4	FOXY ELECTRONICS	Reddy Floyd	5284 5412	NULL	1	NULL	INV_PO010.1 INV_PO010.1 INV_PO010.1 INV_PO010.1 INV...	10
5	JINFIX COMPUTERS	Jordan Liu Min	8412556	2195412	1	426,000.00	INV_PO008.1 INV_PO008.1 INV_PO008.2 INV_PO008.2 INV...	8
6	LAMBONI STAT INC.	Frederic Pey	52557435.	NULL	1	20,400.00	INV_PO004 INV_PO004 INV_PO004.1	4
7	LAMBONI STAT INC.	Frederic Pey	52557435.	NULL	1	86,400.00	INV_PO011.1 INV_PO011.1	11
8	MOTTOWAY CORP.	Stevens Seernah	5794 2513	NULL	1	2,250,000.00	INV_PO002.1 INV_PO002.2 INV_PO002.3	2
9	MOTTOWAY CORP.	Stevens Seernah	5794 2513	NULL	1	NULL	INV_PO013.1 INV_PO013.1 INV_PO013.1 INV_PO013.1	13
10	PEGASUS LTD	Georges Neeroo	61 5841	57412545	1	NULL	INV_PO001 INV_PO001 INV_PO001 INV_PO001 INV_PO00...	1
11	PEGASUS LTD	Georges Neeroo	61 5841	57412545	1	2,385,000.00	INV_PO012.1 INV_PO012.2 INV_PO012.1 INV_PO012.2 INV...	12
12	SAFEST TAXI SERVICES	Steeve Narsimullu	874 1002	217 4512	1	106,800.00	INV_PO007.1 INV_PO007.1 INV_PO007.1 INV_PO007.1	7
13	STUFFIE STATIONERY	Zenhir Belali	6547416	NULL	1	1,500,000.00	INV_PO006.1 INV_PO006.3 INV_PO006.4 INV_PO006.5 INV...	6

Material Used: Microsoft SQL Server Management Studio