

SQL Assignment

Answers and Documentation

Software Engineer	Software Engineer - Regulatory Reporting Tech. Engineering Chapter Technology SBU February 2025
-------------------	---

Submitted by:
Mr. Kishanraj Rughoobee
on
25/03/2025

Table of contents

1	Introduction	3
2	Assigned tasks :.....	3
2.1	Create the "XXBCM_ORDER_MGT" table using the script - DB_Prequisite.sql	3
2.2	Implementation of database schema with necessary tables, columns, data types and constraints. 4	
2.3	Table Creation Scripts.....	4
2.3.1	Table Creation : XXBCM_SUPPLIER	4
2.3.2	Table Creation: XXBCM_PURCHASE_ORDER	5
2.3.3	Table Creation: XXBCM_PURCHASE_ORDER_LINES	6
2.3.4	Table Creation: XXBCM_INVOICE_PAYMENT	7
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.	8
4	Distinct Invoices and their total amount	12
5	Second highest Order Total Amount	14
6	Number of orders and total amount ordered	16

1 Introduction

BCM Ltd is a financial institution recently set up. While the company provides financial services and advices, it makes use of an off-the-shelf information system to help operate its daily activities. The system data is hosted on an Oracle database.

However, as the owner is still managing the company's internal expenses and invoices manually, he is looking forward to move to a simple information system as a way of having a better view on his Purchase Orders and payments to make a better follow up and to be able to undertake decisions that are more accurate.

All information were kept in an excel sheet but have already been loaded in the existing Oracle database in its **raw format** in table "XXBCM_ORDER_MGT". The first column in the table contains both the order reference and order lines. The table contains the list of expenses managed by the owner in terms of Purchase Orders raised as well as the respective Invoices received from his suppliers. A tracking of payments effected per invoice is also maintained in the transactions.

2 Assigned tasks :

2.1 Create the "XXBCM_ORDER_MGT" table using the script - DB_Prequisite.sql

```
CREATE TABLE [dbo].[XXBCM_ORDER_MGT](  
    [ORDER_REF] [varchar](2000) NULL,  
    [ORDER_DATE] [varchar](2000) NULL,  
    [SUPPLIER_NAME] [varchar](2000) NULL,  
    [SUPP_CONTACT_NAME] [varchar](2000) NULL,  
    [SUPP_ADDRESS] [varchar](2000) NULL,  
    [SUPP_CONTACT_NUMBER] [varchar](2000) NULL,  
    [SUPP_EMAIL] [varchar](2000) NULL,  
    [ORDER_TOTAL_AMOUNT] [varchar](2000) NULL,  
    [ORDER_DESCRIPTION] [varchar](2000) NULL,  
    [ORDER_STATUS] [varchar](2000) NULL,  
    [ORDER_LINE_AMOUNT] [varchar](2000) NULL,  
    [INVOICE_REFERENCE] [varchar](2000) NULL,  
    [INVOICE_DATE] [varchar](2000) NULL,  
    [INVOICE_STATUS] [varchar](2000) NULL,  
    [INVOICE_HOLD_REASON] [varchar](2000) NULL,
```

```

[INVOICE_AMOUNT] [varchar](2000) NULL,
[INVOICE_DESCRIPTION] [varchar](2000) NULL
) ON [PRIMARY]
GO

```

Sample Data has been inserted to the table: **[XXBCM_ORDER_MGT]**

ORDER_REF	ORDER_DATE	SUPPLIER_NAME	SUPP_CONTACT	SUPP_ADDRESS	SUPP_CONTACT...	SUPP_EMAIL	ORDER_TOTAL...	ORDER_DESCR...	ORDER_STATUS	ORDER_LINE...	INVOICE_REFE...	INVOICE_DATE	INVOICE_STAT...	INVOICE_HOL...
P0001	03-JAN-2022	PEGASUS LTD	Georges Neeroo	40, Ferney Way...	461 5841, 57412...	geonee@pegas...	10000	Vehicle Spare p...	Closed	NULL	NULL	NULL	NULL	NULL
P0001-1	03-JAN-2022	PEGASUS LTD	Georges Neeroo	40, Ferney Way...	461 5841, 57412...	geonee@pegas...	NULL	Back lights	Received	2500	INV_PO001	28-FEB-2022	Paid	NULL
P0001-2	03-JAN-2022	PEGASUS LTD	Georges Neeroo	40, Ferney Way...	461 5841, 57412...	geonee@pegas...	NULL	Spark plug	Received	600	INV_PO001	28-FEB-2022	Paid	NULL
P0002	10-JAN-2022	MOTTOWAY C...	Stevens Seernah	8312, Cite la Cu...	5794 2513	steseer@motto...	750000	Purchase of Car	Open	NULL	NULL	NULL	NULL	NULL
P0002-1	10-JAN-2022	MOTTOWAY C...	Stevens Seernah	8312, Cite la Cu...	5794 2513	steseer@motto...	NULL	Purchase of Car...	Received	100000	INV_PO002.1	01-MAR-2022	Paid	NULL
P0003	24-JAN-2022	DIGISAY CO. LTD.	Berry Parker	68, Marock Lan...	57841266, 6028...	bparker@intnet...	57300	Computer scre...	Closed	NULL	NULL	NULL	NULL	NULL
P0003-1	24-JAN-2022	DIGISAY CO. LTD.	Berry Parker	68, Marock Lan...	57841266, 6028...	bparker@intnet...	NULL	Screens 22"	Received	22000	INV_PO003.1	22-MAR-2022	Paid	NULL
P0003-2	24-JAN-2022	DIGISAY CO. LTD.	Berry Parker	68, Marock Lan...	57841266, 6028...	bparker@intnet...	NULL	Screens 35"	Received	20300	INV_PO003.1	22-MAR-2022	Paid	NULL
P0003-3	24-JAN-2022	DIGISAY CO. LTD.	Berry Parker	68, Marock Lan...	57841266, 6028...	bparker@intnet...	NULL	Screens 15"	Received	5000	INV_PO003.1	22-MAR-2022	Paid	NULL
P0001-3	03-JAN-2022	PEGASUS LTD	Georges Neeroo	40, Ferney Way...	461 5841, 57412...	geonee@pegas...	NULL	Fuel filter	Received	2000	INV_PO001	28-FEB-2022	Paid	NULL
P0004	07-FEB-2022	LAMBONI STAT...	Frederic Pey	9, Lamu Buildin...	52557435	manustat@gm...	6800	Purchase of Sta...	Closed	NULL	NULL	NULL	NULL	NULL
P0004-1	07-FEB-2022	LAMBONI STAT...	Frederic Pey	9, Lamu Buildin...	52557435	manustat@gm...	2000	Pen Parker blue	Received	2000	INV_PO004	15-FEB-2022	Paid	NULL
P0004-2	07-FEB-2022	LAMBONI STAT...	Frederic Pey	9, Lamu Buildin...	52557435	manustat@gm...	2000	Pencil HB boxes	Received	1200	INV_PO004	15-FEB-2022	Paid	NULL
P0001-4	03-JAN-2022	PEGASUS LTD	Georges Neeroo	40, Ferney Way...	461 5841, 57412...	geonee@pegas...	NULL	Air filter	Received	800	INV_PO001	28-FEB-2022	Paid	NULL
P0002-1	10-JAN-2022	MOTTOWAY C...	Stevens Seernah	8312, Cite la Cu...	5794 2513	steseer@motto...	NULL	Purchase of Car...	Received	300000	INV_PO002.2	02-MAY-2022	Paid	NULL
P0005	07-APR-2022	EMTELLO LTD	Megan Hemblv	998, Bistrop Str...	2420641, 57841...	mhem@my-t...	21000	International M...	Closed	NULL	NULL	NULL	NULL	NULL
P0005-1	07-APR-2022	EMTELLO LTD	Megan Hemblv	998, Bistrop Str...	2420641, 57841...	mhem@my-t...	NULL	National oeo n...	Received	3000	INV_PO005.1	18-MAY-2022	Pending	Damaged maa...

2.2 Implementation of database schema with necessary tables, columns, data types and constraints.

The following new tables have been created:

- **XXBCM_SUPPLIER**
- **XXBCM_PURCHASE_ORDER**
- **XXBCM_PURCHASE_ORDER_LINES**
- **XXBCM_INVOICE_PAYMENT**

2.3 Table Creation Scripts

2.3.1 Table Creation : XXBCM_SUPPLIER

```


CREATE TABLE [dbo].[XXBCM_SUPPLIER](
[SUPPLIER_ID] [int] IDENTITY(1,1) NOT NULL,
[SUPPLIER_NAME] [varchar](2000) NULL,
[SUPPLIER_CONTACT_NAME] [varchar](2000) NULL,
[SUPPLIER_ADDRESS] [varchar](2000) NULL,
[SUPPLIER_CONTACT_NUMBER] [varchar](2000) NULL,
[SUPPLIER_EMAIL] [varchar](2000) NULL,
CONSTRAINT [PK_XXBCM_SUPPLIER] PRIMARY KEY CLUSTERED
(
[SUPPLIER_ID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
```

```
) ON [PRIMARY]
```


```
GO
```

	Column Name	Data Type	Allow Nulls
	SUPPLIER_ID	int	<input type="checkbox"/>
	SUPPLIER_NAME	varchar(2000)	<input checked="" type="checkbox"/>
	SUPPLIER_CONTACT_NAME	varchar(2000)	<input checked="" type="checkbox"/>
	SUPPLIER_ADDRESS	varchar(2000)	<input checked="" type="checkbox"/>
	SUPPLIER_CONTACT_NUMBER	varchar(2000)	<input checked="" type="checkbox"/>
	SUPPLIER_EMAIL	varchar(2000)	<input checked="" type="checkbox"/>

Primary Key: **SUPPLIER_ID (set to Auto Increment)**

2.3.2 Table Creation: XXBCM_PURCHASE_ORDER



```
CREATE TABLE [dbo].[XXBCM_PURCHASE_ORDER](
    [ORDER_ID] [int] IDENTITY(1,1) NOT NULL,
    [SUPPLIER_NAME] [varchar](2000) NULL,
    [ORDER_REF] [varchar](2000) NULL,
    [ORDER_DATE] [smalldatetime] NULL,
    [ORDER_TOTAL_AMOUNT] [float] NULL,
    [ORDER_DESCRIPTION] [varchar](2000) NULL,
    [ORDER_STATUS] [varchar](2000) NULL,
    CONSTRAINT [PK_XXBCM_PURCHASE_ORDER] PRIMARY KEY CLUSTERED
(
    [ORDER_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

	Column Name	Data Type	Allow Nulls
	ORDER_ID	int	<input type="checkbox"/>
	SUPPLIER_N	varchar(2000)	<input checked="" type="checkbox"/>
	ORDER_REF	varchar(2000)	<input checked="" type="checkbox"/>
	ORDER_DATE	smalldatetime	<input checked="" type="checkbox"/>
	ORDER_TOTAL_AMOUNT	float	<input checked="" type="checkbox"/>
	ORDER_DESCRIPTION	varchar(2000)	<input checked="" type="checkbox"/>
	ORDER_STATUS	varchar(2000)	<input checked="" type="checkbox"/>

Primary Key: **ORDER_ID (set to Auto Increment)**

2.3.3 Table Creation: XXBCM_PURCHASE_ORDER_LINES

```
CREATE TABLE [dbo].[XXBCM_PURCHASE_ORDER_LINES](
    [POL_ID] [int] IDENTITY(1,1) NOT NULL,
    [PO_ID] [int] NULL,
    [ORDER_REF] [varchar](2000) NULL,
    [ORDER_DESCRIPTION] [varchar](2000) NULL,
    [ORDER_LINE_AMOUNT] [float] NULL,
    CONSTRAINT [PK_XXBCM_PURCHASE_ORDER_LINES] PRIMARY KEY CLUSTERED
(
    [POL_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

	Column Name	Data Type	Allow Nulls
	POL_ID	int	<input type="checkbox"/>
	PO_ID	int	<input checked="" type="checkbox"/>
	ORDER_REF	varchar(2000)	<input checked="" type="checkbox"/>
	ORDER_DESCRIPTION	varchar(2000)	<input checked="" type="checkbox"/>
	ORDER_LINE_AMOUNT	float	<input checked="" type="checkbox"/>

Primary Key: **POL_ID (set to Auto Increment)**


Foreign Key: **PO_ID**

2.3.4 Table Creation: **XXBCM_INVOICE_PAYMENT**

```

CREATE TABLE [dbo].[XXBCM_INVOICE_PAYMENT](
    [INVOICE_ID] [int] IDENTITY(1,1) NOT NULL,
    [INVOICE_DATE] [varchar](2000) NULL,
    [INVOICE_REFERENCE] [varchar](2000) NULL,
    [INVOICE_STATUS] [varchar](2000) NULL,
    [INVOICE_HOLD_REASON] [varchar](2000) NULL,
    [INVOICE_AMOUNT] [float] NULL,
    [INVOICE_DESCRIPTION] [varchar](2000) NULL,
    CONSTRAINT [PK_XXBCM_INVOICE_PAYMENT] PRIMARY KEY CLUSTERED
(
    [INVOICE_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

```

	Column Name	Data Type	Allow Nulls
	INVOICE_ID	int	<input type="checkbox"/>
	INVOICE_DATE	varchar(2000)	<input checked="" type="checkbox"/>
	INVOICE_REFERENCE	varchar(2000)	<input checked="" type="checkbox"/>
	INVOICE_STATUS	varchar(2000)	<input checked="" type="checkbox"/>
	INVOICE_HOLD_REASON	varchar(2000)	<input checked="" type="checkbox"/>
	INVOICE_AMOUNT	float	<input checked="" type="checkbox"/>
	INVOICE_DESCRIPTION	varchar(2000)	<input checked="" type="checkbox"/>

Primary Key: **INVOICE_ID (set to Auto Increment)**

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.

In this part, I have analysed that the loaded data on the table XXBCM_ORDER_MGT have inconsistent data types. For example, for numerical values like 5000, S000 has been loaded. Among them there are also values like o instead of 0, l instead of 1.

To proceed with the population of tables process, I have implemented a CleanData stored procedure which is called first before the table population process to ensure that the data which will be inserted will be valid and consistent.

```
CREATE PROCEDURE [dbo].[CleanData]
AS
BEGIN
    -- Update ORDER_LINE_AMOUNT: Trim spaces, remove unwanted characters
    UPDATE XXBCM_ORDER_MGT
    SET ORDER_LINE_AMOUNT = TRIM(REPLACE(REPLACE(REPLACE(REPLACE(ORDER_LINE_AMOUNT,
    ',', ''), 'o', '0'), 'l', '1'), 'S', '5'))
    WHERE ORDER_LINE_AMOUNT IS NOT NULL;

    -- Update ORDER_LINE_AMOUNT: Convert empty strings to NULL
    UPDATE XXBCM_ORDER_MGT
    SET ORDER_LINE_AMOUNT = NULL
    WHERE ORDER_LINE_AMOUNT = '';

    -- Update ORDER_LINE_AMOUNT: Convert VARCHAR to FLOAT, handle non-numeric data
    UPDATE XXBCM_ORDER_MGT
    SET ORDER_LINE_AMOUNT = TRY_CAST(ORDER_LINE_AMOUNT AS FLOAT)
```



```
WHERE ISNUMERIC(ORDER_LINE_AMOUNT) = 1;
```

```
-- Update SUPP_CONTACT_NUMBER: Trim spaces, remove unwanted characters
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET SUPP_CONTACT_NUMBER = TRIM(REPLACE(REPLACE(REPLACE(REPLACE(SUPP_CONTACT_NUMBER,  
'o','0'),'l','1'),'s','5'),' ',''))
```

```
WHERE SUPP_CONTACT_NUMBER IS NOT NULL;
```

```
-- Update SUPP_CONTACT_NUMBER: Convert empty strings to NULL
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET SUPP_CONTACT_NUMBER = NULL
```

```
WHERE SUPP_CONTACT_NUMBER = '';
```

```
-- Update ORDER_TOTAL_AMOUNT: Trim spaces, remove unwanted characters
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET ORDER_TOTAL_AMOUNT = TRIM(REPLACE(REPLACE(REPLACE(REPLACE(ORDER_TOTAL_AMOUNT,  
' ',''),'o','0'),'l','1'),'s','5'))
```

```
WHERE ORDER_TOTAL_AMOUNT IS NOT NULL;
```

```
-- Update ORDER_TOTAL_AMOUNT: Convert empty strings to NULL
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET ORDER_TOTAL_AMOUNT = NULL
```

```
WHERE ORDER_TOTAL_AMOUNT = '';
```

```
-- Update ORDER_TOTAL_AMOUNT: Convert VARCHAR to FLOAT, handle non-numeric data
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET ORDER_TOTAL_AMOUNT = TRY_CAST(ORDER_TOTAL_AMOUNT AS FLOAT)
```

```
WHERE ISNUMERIC(ORDER_TOTAL_AMOUNT) = 1;
```

```
-- Update INVOICE_AMOUNT: Trim spaces, remove unwanted characters
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET INVOICE_AMOUNT = TRIM(REPLACE(REPLACE(REPLACE(REPLACE(INVOICE_AMOUNT,  
' ',''),'o','0'),'l','1'),'S','5'))
```

```
WHERE INVOICE_AMOUNT IS NOT NULL;
```

```
-- Update INVOICE_AMOUNT: Convert empty strings to NULL
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET INVOICE_AMOUNT = NULL
```

```
WHERE INVOICE_AMOUNT = '';
```

```
-- Update INVOICE_AMOUNT: Convert VARCHAR to FLOAT, handle non-numeric data
```

```
UPDATE XXBCM_ORDER_MGT
```

```
SET INVOICE_AMOUNT = TRY_CAST(INVOICE_AMOUNT AS FLOAT)
```

```
WHERE ISNUMERIC(INVOICE_AMOUNT) = 1;
```

```
PRINT 'Data cleaning completed successfully!';
```

```
END
```

```
GO
```

This stored procedure will be automatically executed during the migration process. For the migration process, the following stored procedure has been created:

```
CREATE PROCEDURE [dbo].[MIGRATION]
```

```
AS
```

```
BEGIN
```

```
EXEC CleanData – Executing the cleaning up of data
```

```
-- Supplier
```

```
INSERT INTO
```

```
dbo.XXBCM_SUPPLIER(SUPPLIER_NAME,SUPPLIER_CONTACT_NAME,SUPPLIER_ADDRESS,SUPPLIER_CONTACT_NUMBER,SUPPLIER_EMAIL)
```

```
SELECT
```

```
SUPPLIER_NAME,SUPP_CONTACT_NAME,SUPP_ADDRESS,SUPP_CONTACT_NUMBER,SUPP_EMAIL
```

```
FROM
```

```
dbo.XXBCM_ORDER_MGT
```

```
-- Purchase Order
```

```
INSERT INTO
```

```
dbo.XXBCM_PURCHASE_ORDER(ORDER_REF,ORDER_DATE,ORDER_TOTAL_AMOUNT,ORDER_DESCRIPTION  
,ORDER_STATUS,SUPPLIER_NAME)
```

```
SELECT
```

```
ORDER_REF,ORDER_DATE,ORDER_TOTAL_AMOUNT,ORDER_DESCRIPTION,ORDER_STATUS,SUPPLIER_NAM  
E
```

```
FROM
```

```
dbo.XXBCM_ORDER_MGT
```

```
where ORDER_REF NOT LIKE '%-%'
```

```
-- PO LINES
```

```
INSERT INTO
```

```
dbo.XXBCM_PURCHASE_ORDER_LINES(ORDER_REF,ORDER_DESCRIPTION,ORDER_LINE_AMOUNT)
```

```
SELECT
```

```
ORDER_REF,ORDER_DESCRIPTION, TRY_CAST(ORDER_LINE_AMOUNT AS FLOAT)
```

```
FROM
```

```
dbo.XXBCM_ORDER_MGT
```

```
WHERE ORDER_LINE_AMOUNT IS NOT NULL
```

```
--Update ORDER ID from PO HEADER
```

```
UPDATE I
```

```
SET I.PO_ID = p.ORDER_ID
```

```
FROM XXBCM_PURCHASE_ORDER_LINES I
```

```
JOIN XXBCM_PURCHASE_ORDER p ON left(I.ORDER_REF,5) = p.ORDER_REF
```

```
-- INVOICE PAYMENT
```

```
INSERT INTO
```

```
dbo.XXBCM_INVOICE_PAYMENT(INVOICE_DATE,INVOICE_REFERENCE,INVOICE_STATUS,INVOICE_HOLD_REASON,INVOICE_AMOUNT,INVOICE_DESCRIPTION)
```

```
SELECT
```

```
INVOICE_DATE,INVOICE_REFERENCE,INVOICE_STATUS,INVOICE_HOLD_REASON,TRY_CAST(INVOICE_AMOUNT AS FLOAT),INVOICE_DESCRIPTION
```

```
FROM
```

```
dbo.XXBCM_ORDER_MGT
```

```
WHERE INVOICE_REFERENCE IS NOT NULL
```

```
END
```

```
GO
```

4 Distinct Invoices and their total amount

The following stored procedure has been created.

- **Task No.4 - InvoicesAndTotals**

```
CREATE PROCEDURE [dbo].[Task No.4 - InvoicesAndTotals]
```

```
AS
```

```
BEGIN
```

```
SELECT DISTINCT
```

```
    CAST(SUBSTRING(t1.ORDER_REF, PATINDEX('%[0-9]%', t1.ORDER_REF),
```

```
    LEN(t1.ORDER_REF)) AS INT) AS [Order Reference],
```

```
    {fn CONCAT(UPPER(LEFT(t1.SUPPLIER_NAME, 1)), LOWER(SUBSTRING(t1.SUPPLIER_NAME, 2, LEN(t1.SUPPLIER_NAME)))) } AS [Supplier Name],
```

```
    FORMAT(t1.ORDER_TOTAL_AMOUNT, 'N2') AS [Order Total Amount],
```

```
    t1.ORDER_STATUS AS [Order Status],
```

```
    CAST(SUBSTRING(t2.INVOICE_REFERENCE, PATINDEX('%[0-9]%',
```

```
t2.INVOICE_REFERENCE), LEN(t2.INVOICE_REFERENCE)) AS DECIMAL) AS [Invoice Reference],  
FORMAT(SUM(t2.INVOICE_AMOUNT), 'N2') AS [Invoice Total Amount],  
CASE WHEN t2.INVOICE_STATUS = 'Paid' THEN 'OK' WHEN t2.INVOICE_STATUS = 'Pending' THEN 'To  
follow up' ELSE 'To Verify' END AS Action  
FROM dbo.XXBCM_PURCHASE_ORDER AS t1 INNER JOIN  
    dbo.XXBCM_INVOICE_PAYMENT AS t2 ON CAST(SUBSTRING(t1.ORDER_REF, PATINDEX('%[0-9]%',  
t1.ORDER_REF), LEN(t1.ORDER_REF)) AS INT) = CAST(SUBSTRING(t2.INVOICE_REFERENCE, PATINDEX('%[0-  
9]%',  
    t2.INVOICE_REFERENCE), LEN(t2.INVOICE_REFERENCE)) AS DECIMAL)  
GROUP BY CAST(SUBSTRING(t1.ORDER_REF, PATINDEX('%[0-9]%', t1.ORDER_REF),  
LEN(t1.ORDER_REF)) AS INT), t1.SUPPLIER_NAME, t1.ORDER_DATE, t1.ORDER_TOTAL_AMOUNT,  
t1.ORDER_STATUS, t2.INVOICE_STATUS,  
    CAST(SUBSTRING(t2.INVOICE_REFERENCE, PATINDEX('%[0-9]%', t2.INVOICE_REFERENCE),  
LEN(t2.INVOICE_REFERENCE)) AS DECIMAL)  
END  
GO
```

Once executed, the following results are displayed:

```

USE [MCB_Assignment]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[Task No.4 - InvoicesAndTotals]

SELECT  'Return Value' = @return_value

GO

```

	Order Reference	Supplier Name	Order Total Amount	Order Status	Invoice Reference	Invoice Total Amount	Action
1	1	Pegasus ltd	10,000.00	Closed	1	10,000.00	OK
2	2	Mottoway corp.	750,000.00	Open	2	249,000.00	To follow up
3	2	Mottoway corp.	750,000.00	Open	2	400,000.00	OK
4	3	Digisay co. ltd.	57,300.00	Closed	3	57,300.00	OK
5	4	Lamboni stat inc.	6,800.00	Closed	4	6,200.00	OK
6	5	Emtello ltd	21,000.00	Closed	5	18,000.00	OK
7	5	Emtello ltd	21,000.00	Closed	5	3,000.00	To follow up
8	6	Stuffie stationery	250,000.00	Open	6	104,200.00	OK
9	7	Safedest taxi services	26,700.00	Closed	7	148,000.00	OK
10	8	Jinfex computers	85,200.00	Open	8	85,200.00	OK
11	9	Fireland bros.	36,800.00	Open	9	22,500.00	OK
12	10	Foxy electronics	182,700.00	Closed	10	182,700.00	OK
13	11	Lamboni stat inc.	43,200.00	Closed	11	43,200.00	OK
14	12	Pegasus ltd	265,000.00	Open	12	241,220.00	OK
15	13	Mottoway corp.	5,819,630.00	Closed	13	5,819,630.00	OK
16	14	Digisay co. ltd.	400,120.00	Open	14	204,720.00	OK
17	14	Digisay co. ltd.	400,120.00	Open	14	90,800.00	To follow up

5 Second highest Order Total Amount

- Task No.5 - Second Highest Order

To get this result, the following stored procedures has been created:

```
CREATE PROCEDURE [dbo].[Task No.5 - Second Highest Order]
```

```
AS
```

```
BEGIN
```

```
SELECT DISTINCT
```

```
    CAST(SUBSTRING(t1.ORDER_REF, PATINDEX('%[0-9]%', t1.ORDER_REF), LEN(t1.ORDER_REF)) AS
INT) AS [Order Reference],
```

```
    FORMAT(t1.ORDER_DATE, 'dd MMMM, yyyy') AS [Order Date],
```

```
    UPPER(t1.SUPPLIER_NAME) AS [Supplier Name],
```

```
    format(t1.ORDER_TOTAL_AMOUNT, 'N2') AS [Order Total Amount],
```

```

t1.ORDER_STATUS AS [Order Status],

STRING_AGG(t2.INVOICE_REFERENCE, ' | ') AS [Invoice References]

FROM dbo.XXBCM_PURCHASE_ORDER AS t1

INNER JOIN

    dbo.XXBCM_INVOICE_PAYMENT AS t2 ON CAST(SUBSTRING(t1.ORDER_REF,
PATINDEX('%[0-9]%', t1.ORDER_REF), LEN(t1.ORDER_REF)) AS INT) =
CAST(SUBSTRING(t2.INVOICE_REFERENCE, PATINDEX('%[0-9]%',

    t2.INVOICE_REFERENCE), LEN(t2.INVOICE_REFERENCE)) AS DECIMAL)

INNER JOIN

    dbo.[Second Highest Order Total Amount] ON t1.ORDER_TOTAL_AMOUNT = dbo.[Second Highest
Order Total Amount].ORDER_TOTAL_AMOUNT --a view created [Second Highest Order Total Amount] to
split and modularise the process

GROUP BY CAST(SUBSTRING(t1.ORDER_REF, PATINDEX('%[0-9]%', t1.ORDER_REF),

    LEN(t1.ORDER_REF)) AS INT),

    t1.SUPPLIER_NAME,

    t1.ORDER_DATE,

    t1.ORDER_TOTAL_AMOUNT,

    t1.ORDER_STATUS

END

GO

```

Upon executing, this Stored Procedure, the results are as follows:

```

DECLARE @return_value int

EXEC @return_value = [dbo].[Task No.5 - Second Highest Order]

SELECT 'Return Value' = @return_value

GO

```

Order Reference	Order Date	Supplier Name	Order Total Amount	Order Status	Invoice References
2	10 January, 2022	MOTTOWAY CORP.	750,000.00	Open	INV_PO002.1 INV_PO002.2 INV_PO002.3

6 Number of orders and total amount ordered

- Task No.6 - Number of orders and total amount ordered

The following stored procedure has been created:

```
CREATE PROCEDURE [dbo].[Task No.6 - Number of orders and total amount ordered]
```

```
AS
```

```
BEGIN
```

```
SELECT
```

```
    dbo.All_Orders_01012022To31082022.SUPPLIER_NAME AS [Supplier Name],
```

```
    dbo.All_Orders_01012022To31082022.SUPPLIER_CONTACT_NAME AS [Supplier Contact Name],
```

```
    dbo.SplittedContact.[1] AS [Supplier Contact No.1], --Pivot [SplittedContact] created in a view
```

```
    dbo.SplittedContact.[2] AS [Supplier Contact No.2],
```

```
    dbo.All_Orders_01012022To31082022.[Total Orders], -- a view created  
[All_Orders_01012022To31082022] to split and modularise the process
```

```
    FORMAT(SUM(dbo.All_Orders_01012022To31082022.ORDER_TOTAL_AMOUNT), 'N2') AS [Order  
Total Amount]
```

```
FROM dbo.All_Orders_01012022To31082022
```

```
    INNER JOIN
```

```
    dbo.SplittedContact ON dbo.All_Orders_01012022To31082022.SUPPLIER_CONTACT_NUMBER =  
dbo.SplittedContact.SUPPLIER_CONTACT_NUMBER
```

```
GROUP BY
```

```
    dbo.All_Orders_01012022To31082022.SUPPLIER_NAME,
```

```
    dbo.All_Orders_01012022To31082022.SUPPLIER_CONTACT_NAME,
```

```
    dbo.All_Orders_01012022To31082022.SUPPLIER_CONTACT_NUMBER,
```

```
    dbo.All_Orders_01012022To31082022.[Total Orders],
```

```
    dbo.SplittedContact.[1], dbo.SplittedContact.[2]
```

```
END
```

```
GO
```


Upon execution, the following results are displayed:

```

DECLARE @return_value int

EXEC @return_value = [dbo].[Task No.6 - Number of orders and total amount ordered]

SELECT 'Return Value' = @return_value

GO

```

100 %

Results Messages

	Supplier Name	Supplier Contact Name	Supplier Contact No.1	Supplier Contact No.2	Total Orders	Order Total Amount
1	DIGISAY CO. LTD.	Berry Parker	57841266	6028010	26	57,300.00
2	EMTELLO LTD	Megan Hembly	2420641	57841698	5	21,000.00
3	FIRELAND BROS.	Amelia Bridney	5948 0015	5948 0015	2	36,800.00
4	FOXY ELECTRONICS	Reddy Floyd	5284 5412	5284 5412	6	182,700.00
5	JINFIX COMPUTERS	Jordan Liu Min	58412556	2195412	6	85,200.00
6	LAMBONI STAT INC.	Frederic Pey	52557435	52557435	8	50,000.00
7	MOTTOWAY CORP.	Stevens Seemah	5794 2513	5794 2513	9	6,569,630.00
8	PEGASUS LTD	Georges Neeroo	461 5841	57412545	17	275,000.00
9	SAFEDEST TAXI SERVICES	Steeve Narsimullu	5874 1002	217 4512	6	26,700.00
10	STUFFIE STATIONERY	Zenhir Belall	6547416	6547416	7	250,000.00

Notes:

Performance:

```
CREATE INDEX idx_ORDER_ID ON XXBCM_PURCHASE_ORDER(ORDER_ID)
```

For Error Handling:

CleanData stored procedure has been implemented to clean data as per required data types.

Tools used:

Microsoft SQL Server (Tables, Views, Stored Procedures, Built-in Functions)

Database Version:

Microsoft SQL Server 2019

Data Cleansing :

We have assumed that the data on the table: XXBCM_ORDER_MGT which had values:

S which means 5

O which means 0

I which means 1