

Assignment

# User Defined Functions

---

Harsh Siddhapura

1230169813

02/18/2024

## QUESTION - 5

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'CONNECTIONS' pane shows the server hierarchy: **SERVERS** > **localhost, <default> (sa)** > **Databases** > **AP** > **Functions** > **Aggregate Functions**. The main editor window, titled 'UserDefinedFunctions.sql - localh...AP (sa)', contains the following SQL code:

```
1  /*
2  ** Name: Harsh Siddhapura
3  ** Enrollment No.: 1230169813
4  ** Course: IFT530 - Advanced Database Management System
5  ** Date: 02/25/2024
6  */
7
8
9  -- Question - 5
10 CREATE FUNCTION fnUnpaidInvoiceID()
11 RETURNS INT
12 AS
13 BEGIN
14     DECLARE @InvoiceID INT;
15     SELECT TOP 1 @InvoiceID = InvoiceID
16     FROM Invoices
17     WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0
18     ORDER BY InvoiceDate ASC;
19     RETURN @InvoiceID;
20 END;
21 GO
22
23
24 SELECT VendorName, InvoiceNumber, InvoiceDueDate,
25        InvoiceTotal - CreditTotal - PaymentTotal AS Balance
26 FROM Vendors v
27 JOIN Invoices i
28     ON v.VendorID = i.VendorID
29 WHERE InvoiceID = dbo.fnUnpaidInvoiceID();
30 GO
31
32
33
34 -- Question - 6
35 CREATE FUNCTION fnDateRange(@StartDate DATE, @EndDate DATE)
36 RETURNS TABLE
37 AS
38 RETURN
39 (
40     SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
41     FROM Invoices
42     WHERE InvoiceDate BETWEEN @StartDate AND @EndDate
43 );
44 GO
```

The 'Messages' pane at the bottom shows the execution results:

```
3:12:29 PM      Started executing query at Line 10
                Commands completed successfully.
                Total execution time: 00:00:00.030
```

The status bar at the bottom indicates: Ln 10, Col 1 (269 selected) Spaces: 4 UTF-8 LF SQL 0 rows Choose SQL Language 00:00:00 localhost : AP

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'SERVERS' tree shows the 'Databases' folder expanded, with 'AP' selected. The 'Functions' folder under 'AP' is also expanded, showing 'dbo.fnUnpaidInvoiceID' and 'dbo.fnUnpaidInvoiceID'.

The main pane shows a T-SQL script titled 'UserDefinedFunctions.sql - localh...AP (sa)'. The script contains two functions: 'fnUnpaidInvoiceID' and 'fnDateRange'. The 'fnUnpaidInvoiceID' function is a scalar-valued function that returns the invoice ID of the top invoice with the highest balance. The 'fnDateRange' function is a table-valued function that returns a table of invoices within a specified date range.

The script is as follows:

```
1  /*
2  ** Name: Harsh Siddhapura
3  ** Enrollment No.: 1230169813
4  ** Course: IFT530 - Advanced Database Management System
5  ** Date: 02/25/2024
6  */
7
8
9  -- Question - 5
10 CREATE FUNCTION fnUnpaidInvoiceID()
11 RETURNS INT
12 AS
13 BEGIN
14     DECLARE @InvoiceID INT;
15     SELECT TOP 1 @InvoiceID = InvoiceID
16     FROM Invoices
17     WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0
18     ORDER BY InvoiceDate ASC;
19     RETURN @InvoiceID;
20 END;
21 GO
22
23
24 SELECT VendorName, InvoiceNumber, InvoiceDueDate,
25        InvoiceTotal - CreditTotal - PaymentTotal AS Balance
26 FROM Vendors v
27 JOIN Invoices i
28     ON v.VendorID = i.VendorID
29 WHERE InvoiceID = dbo.fnUnpaidInvoiceID();
30 GO
31
32
33
34 -- Question - 6
35 CREATE FUNCTION fnDateRange(@StartDate DATE, @EndDate DATE)
36 RETURNS TABLE
37 AS
38 RETURN
39 (
40     SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
41     FROM Invoices
42     WHERE InvoiceDate BETWEEN @StartDate AND @EndDate
43 );
44 GO
```

The 'Results' pane shows the output of the script, which is a single row of data:

	VendorName	InvoiceNumber	InvoiceDueDate	Balance
1	Data Reproductions Corp	39104	2023-02-09	85.31

### QUESTION - 6

The screenshot displays the SQL Server Enterprise Manager interface on the left and the SQL Server Query Editor on the right. The Enterprise Manager shows the server hierarchy for 'localhost, <default> (sa)', with the 'Functions' folder expanded under 'Databases' > 'AP'. The 'dbo.fnUnpaidInvoiceID' function is highlighted. The Query Editor shows the SQL script for creating two functions: 'fnUnpaidInvoiceID' and 'fnDateRange'.

**SQL Script:**

```
9  -- Question - 5
10 CREATE FUNCTION fnUnpaidInvoiceID()
11 RETURNS INT
12 AS
13 BEGIN
14     DECLARE @InvoiceID INT;
15     SELECT TOP 1 @InvoiceID = InvoiceID
16     FROM Invoices
17     WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0
18     ORDER BY InvoiceDate ASC;
19     RETURN @InvoiceID;
20 END;
21 GO
22
23
24 SELECT VendorName, InvoiceNumber, InvoiceDueDate,
25     InvoiceTotal - CreditTotal - PaymentTotal AS Balance
26 FROM Vendors v
27     JOIN Invoices i
28     ON v.VendorID = i.VendorID
29 WHERE InvoiceID = dbo.fnUnpaidInvoiceID();
30 GO
31
32
33
34 -- Question - 6
35 CREATE FUNCTION fnDateRange(@StartDate DATE, @EndDate DATE)
36 RETURNS TABLE
37 AS
38 RETURN
39 (
40     SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
41     FROM Invoices
42     WHERE InvoiceDate BETWEEN @StartDate AND @EndDate
43 );
44 GO
45
46 SELECT * FROM fnDateRange('2022-10-10', '2022-10-20');
47
48
49
50 -- Question - 7
51 SELECT DISTINCT v.VendorName, fn.InvoiceNumber, fn.InvoiceDate, fn.InvoiceTotal, fn.Balance
```

**Messages:**

3:21:13 PM Started executing query at Line 35  
Commands completed successfully.  
Total execution time: 00:00:00.030

Sun Feb 25 3:22:07 PM

Search

CONNECTIONS

SERVERS

localhost, <default> (sa)

Databases

System Databases

AP

Tables

Views

Synonyms

Programmability

Stored Procedures

Functions

Table-valued Functions

dbo.fnDateRange

Scalar-valued Functions

dbo.fnUnpaidInvoiceID

Aggregate Functions

Database Triggers

Assemblies

Types

Sequences

External Resources

Service Broker

Storage

Security

IFT530-ADBMS

Security

Server Objects

UserDefinedFunctions.sql - localh...AP (sa) X

Users > harshsiddhapura > Harsh > Education > MS\_JT > Sem-2 > IFT530 - Advance Database Management System > Assignments > A-07\_User-De

Run Cancel Disconnect Change Database: AP Estimated Plan Enable Actual Plan Parse

Enable SQLCMD To Notebook

```

/
8
9  -- Question - 5
10 CREATE FUNCTION fnUnpaidInvoiceID()
11 RETURNS INT
12 AS
13 BEGIN
14     DECLARE @InvoiceID INT;
15     SELECT TOP 1 @InvoiceID = InvoiceID
16     FROM Invoices
17     WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0
18     ORDER BY InvoiceDate ASC;
19     RETURN @InvoiceID;
20 END;
21 GO
22
23
24 SELECT VendorName, InvoiceNumber, InvoiceDueDate,
25         InvoiceTotal - CreditTotal - PaymentTotal AS Balance
26 FROM Vendors v
27 JOIN Invoices i
28     ON v.VendorID = i.VendorID
29 WHERE InvoiceID = dbo.fnUnpaidInvoiceID();
30 GO
31
32
33
34 -- Question - 6
35 CREATE FUNCTION fnDateRange(@StartDate DATE, @EndDate DATE)
36 RETURNS TABLE
37 AS
38 RETURN
39 (
40     SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
41     FROM Invoices
42     WHERE InvoiceDate BETWEEN @StartDate AND @EndDate
43 );
44 GO
45
46 SELECT * FROM fnDateRange('2022-10-10', '2022-10-20');
47
48
49

```

Results Messages

	InvoiceNumber	InvoiceDate	InvoiceTotal	Balance
1	263253241	2022-10-10	40.20	0.00
2	963253234	2022-10-13	138.75	0.00
3	2-000-2993	2022-10-16	144.70	0.00
4	963253251	2022-10-16	15.50	0.00
5	963253261	2022-10-16	42.75	0.00

AZURE

Ln 47, Col 1 (55 selected) Spaces: 4 UTF-8 LF SQL 5 rows Choose SQL Language 00:00:00 localhost: AP

## QUESTION - 7

Users > harshsiddhapura > Harsh > Education > MS\_IT > Sem-2 > IFT530 - Advance Database Management System > Assignments > A-07\_User-De

Run Cancel Disconnect Change Database: AP Estimated Plan Enable Actual Plan Parse

Enable SQLCMD To Notebook

```

22
23
24 SELECT VendorName, InvoiceNumber, InvoiceDueDate,
25         InvoiceTotal - CreditTotal - PaymentTotal AS Balance
26 FROM Vendors v
27      JOIN Invoices i
28         ON v.VendorID = i.VendorID
29 WHERE InvoiceID = dbo.fnUnpaidInvoiceID();
30 GO
31
32
33
34 -- Question - 6
35 CREATE FUNCTION fnDateRange(@StartDate DATE, @EndDate DATE)
36 RETURNS TABLE
37 AS
38 RETURN
39 (
40     SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
41     FROM Invoices
42     WHERE InvoiceDate BETWEEN @StartDate AND @EndDate
43 );
44 GO
45
46 SELECT * FROM fnDateRange('2022-10-10', '2022-10-20');
47
48
49
50 -- Question - 7
51 SELECT DISTINCT v.VendorName, fn.InvoiceNumber, fn.InvoiceDate, fn.InvoiceTotal, fn.Balance
52 FROM Vendors v
53      JOIN Invoices i ON v.VendorID = i.VendorID
54      CROSS APPLY fnDateRange('2022-10-10', '2022-10-20') AS fn
55 WHERE i.InvoiceNumber = fn.InvoiceNumber;
56

```

**Results** Messages

	VendorName	InvoiceNumber	InvoiceDate	InvoiceTotal	Balance
1	Federal Express Corporation	2-000-2993	2022-10-16	144.70	0.00
2	Federal Express Corporation	263253241	2022-10-10	40.20	0.00
3	Federal Express Corporation	963253234	2022-10-13	138.75	0.00
4	Federal Express Corporation	963253251	2022-10-16	15.50	0.00
5	Federal Express Corporation	963253261	2022-10-16	42.75	0.00

Ln 51, Col 1 (250 selected) Spaces: 4 UTF-8 LF SQL 5 rows Choose SQL Language 00:00:00 localhost : AP

## SCRIPT

/\*

**\*\* Name: Harsh Siddhapura**

**\*\* Enrollment No.: 1230169813**

**\*\* Course: IFT530 - Advanced Database Management System**

**\*\* Date: 02/25/2024**

\*/

**-- Question - 5**

**CREATE FUNCTION fnUnpaidInvoiceID()**

**RETURNS INT**

**AS**

**BEGIN**

**DECLARE @InvoiceID INT;**

**SELECT TOP 1 @InvoiceID = InvoiceID**

**FROM Invoices**

**WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0**

**ORDER BY InvoiceDate ASC;**

**RETURN @InvoiceID;**

**END;**

**GO**

**SELECT VendorName, InvoiceNumber, InvoiceDueDate,**

**InvoiceTotal - CreditTotal - PaymentTotal AS Balance**

**FROM Vendors v**

**JOIN Invoices i**

**ON v.VendorID = i.VendorID**

**WHERE InvoiceID = dbo.fnUnpaidInvoiceID();**

**GO**

**-- Question - 6**

**CREATE FUNCTION fnDateRange(@StartDate DATE, @EndDate DATE)**

**RETURNS TABLE**

**AS**

**RETURN**

**(**

**SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal - CreditTotal - PaymentTotal AS  
Balance**

**FROM Invoices**

**WHERE InvoiceDate BETWEEN @StartDate AND @EndDate**

**);**

**GO**

**SELECT \* FROM fnDateRange('2022-10-10', '2022-10-20');**

**-- Question - 7**

**SELECT DISTINCT v.VendorName, fn.InvoiceNumber, fn.InvoiceDate, fn.InvoiceTotal, fn.Balance**

**FROM Vendors v**

**JOIN Invoices i ON v.VendorID = i.VendorID**

**CROSS APPLY fnDateRange('2022-10-10', '2022-10-20') AS fn**

**WHERE i.InvoiceNumber = fn.InvoiceNumber;**