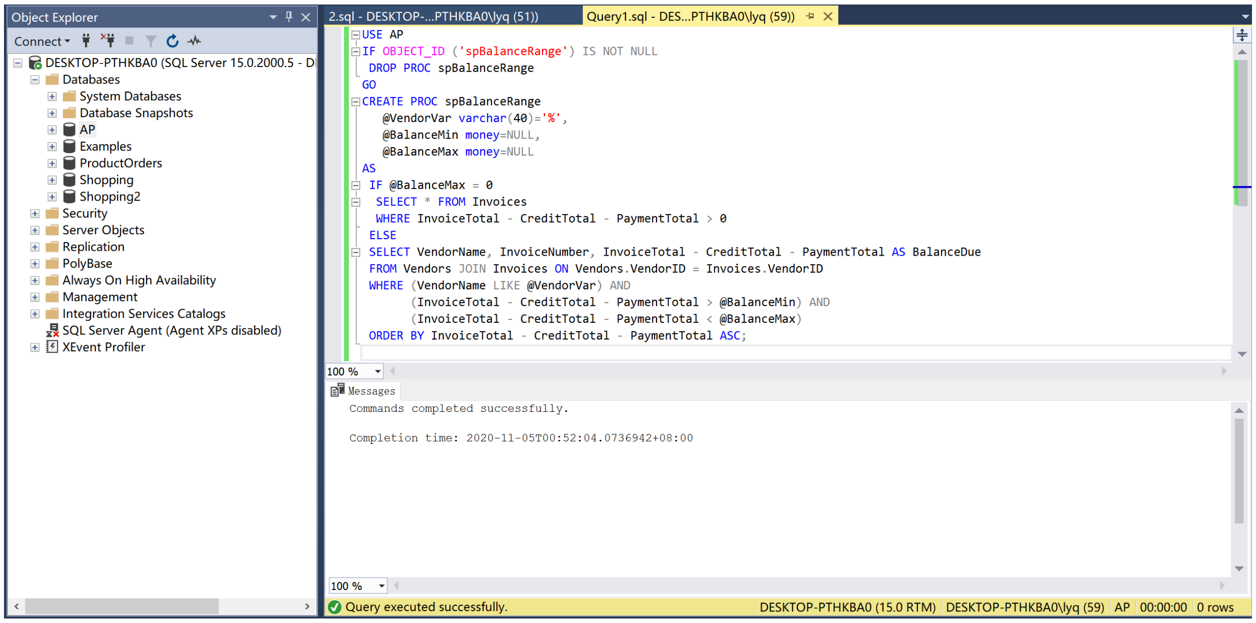
Lab 9: Stored Procedures, Functions

Use AP database throughout.
Balance column's definition: InvoiceTotal – CreditTotal – PaymentTotal
1.Create a stored procedure named spBalanceRange that accepts three optional parameters. The procedure should return a result set consisting of VendorName, InvoiceNumber, and Balance for each invoice with a balance due, sorted with smallest balance due first. The parameter @VendorVar is a mask that's used with a LIKE operator to filter by vendor name. @BalanceMin and @BalanceMax are parameters used to specify the requested range of balances due. If called with no parameters or with a maximum value of 0, the procedure should return all invoices with a balance due.

Here are my procedures as below:



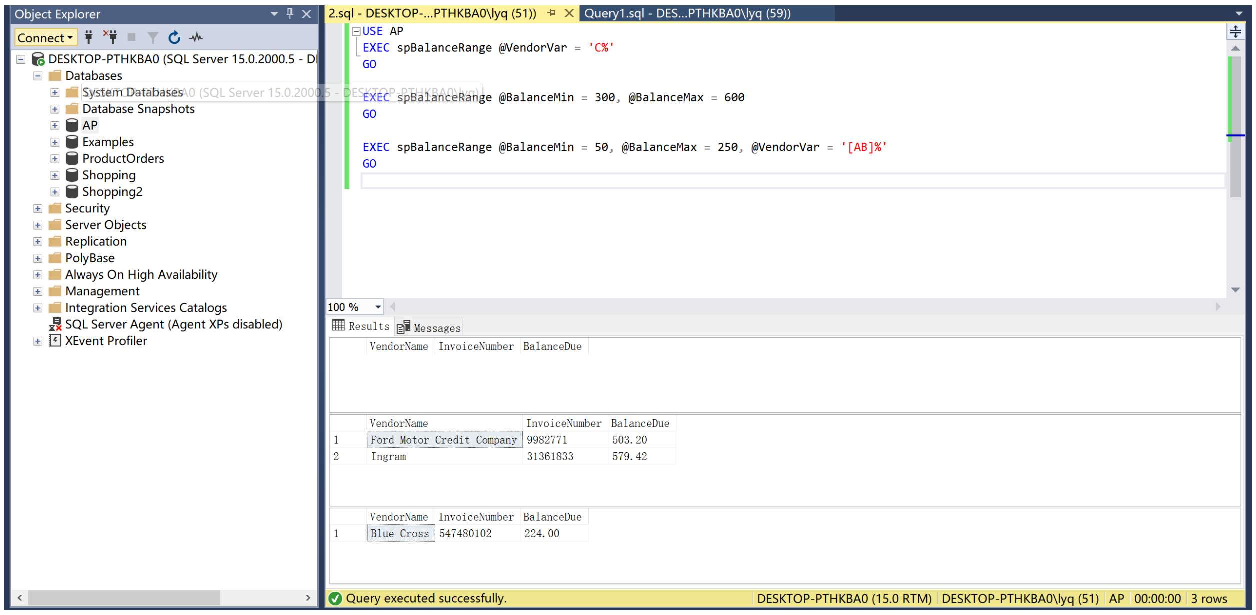
```
USE AP
IF OBJECT_ID ('spBalanceRange') IS NOT NULL
 DROP PROC spBalanceRange
GO
CREATE PROC spBalanceRange
   @VendorVar varchar(40)='%',
   @BalanceMin money=NULL,
   @BalanceMax money=NULL
AS
 IF @BalanceMax = 0
  SELECT * FROM Invoices
  WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0
 ELSE
 SELECT VendorName, InvoiceNumber, InvoiceTotal - CreditTotal - PaymentTotal AS BalanceDue
 FROM Vendors JOIN Invoices ON Vendors.VendorID = Invoices.VendorID
 WHERE (VendorName LIKE @VendorVar) AND
       (InvoiceTotal - CreditTotal - PaymentTotal > @BalanceMin) AND
       (InvoiceTotal - CreditTotal - PaymentTotal < @BalanceMax)
 ORDER BY InvoiceTotal - CreditTotal - PaymentTotal ASC;
```

```
Commands completed successfully.

Completion time: 2020-11-05T00:52:04.0736942+08:00
```

Object Explorer
Connect
DESKTOP-PTHKBA0 (SQL Server 15.0.2000.5 - D
Databases
System Databases
Database Snapshots
AP
Examples
ProductOrders
Shopping
Shopping2
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
2.sql - DESKTOP-...PTHKBA0\lyq (51))
Query1.sql - DES...PTHKBA0\lyq (59))
100 %
Messages
Query executed successfully.
DESKTOP-PTHKBA0 (15.0 RTM)
DESKTOP-PTHKBA0\lyq (59)
AP
00:00:00
0 rows

2. Code three calls to the procedure created in question 1:
a. passed by position with @VendorVar = ‘C%’ and no balance range
b. passed by name with @VendorVar omitted and a balance range from $300 to $600
c. passed by position with a balance due from $50 to $250, filtering for vendors whose names begin with A or B

The procedures of a.b.c are as below :


Object Explorer
Connect
DESKTOP-PTHKBA0 (SQL Server 15.0.2000.5 - D
Databases
System Databases
Database Snapshots
AP
Examples
ProductOrders
Shopping
Shopping2
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
2.sql - DESKTOP-...PTHKBA0\lyq (51))
Query1.sql - DES...PTHKBA0\lyq (59))
100 %
Results
Messages

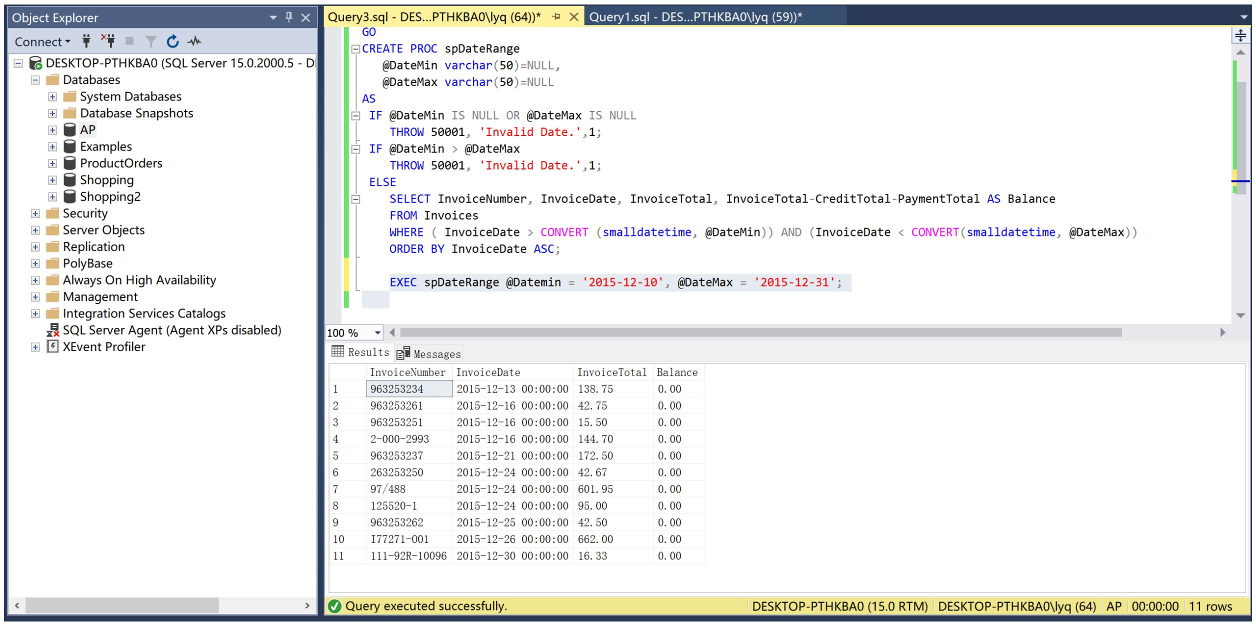
```
USE AP
EXEC spBalanceRange @VendorVar = 'C%'
GO

EXEC spBalanceRange @BalanceMin = 300, @BalanceMax = 600
GO

EXEC spBalanceRange @BalanceMin = 50, @BalanceMax = 250, @VendorVar = '[AB]%'
GO
```

```
   VendorName  InvoiceNumber  BalanceDue

   VendorName                InvoiceNumber  BalanceDue
1  Ford Motor Credit Company 9982771        503.20
2  Ingram                    31361833       579.42

   VendorName  InvoiceNumber  BalanceDue
1  Blue Cross  547480102      224.00
```

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (51) AP 00:00:00 3 rows

3. Create a stored procedure named spDateRange that accepts two parameters, @DateMin and @DateMax, with data type varchar and default value null. If called with no parameters or with null values, raise an error that describes the problem. If called with non-null values, validate the parameters. Test that the literal strings are valid dates and test that @DateMin is earlier than @DateMax. If the parameters are valid, return a result set that includes the InvoiceNumber, InvoiceDate, InvoiceTotal, and Balance for each invoice for which the InvoiceDate is within the date range, sorted with earliest invoice first.

Here's the stored procedure and there are 11 invoices meet the time range.



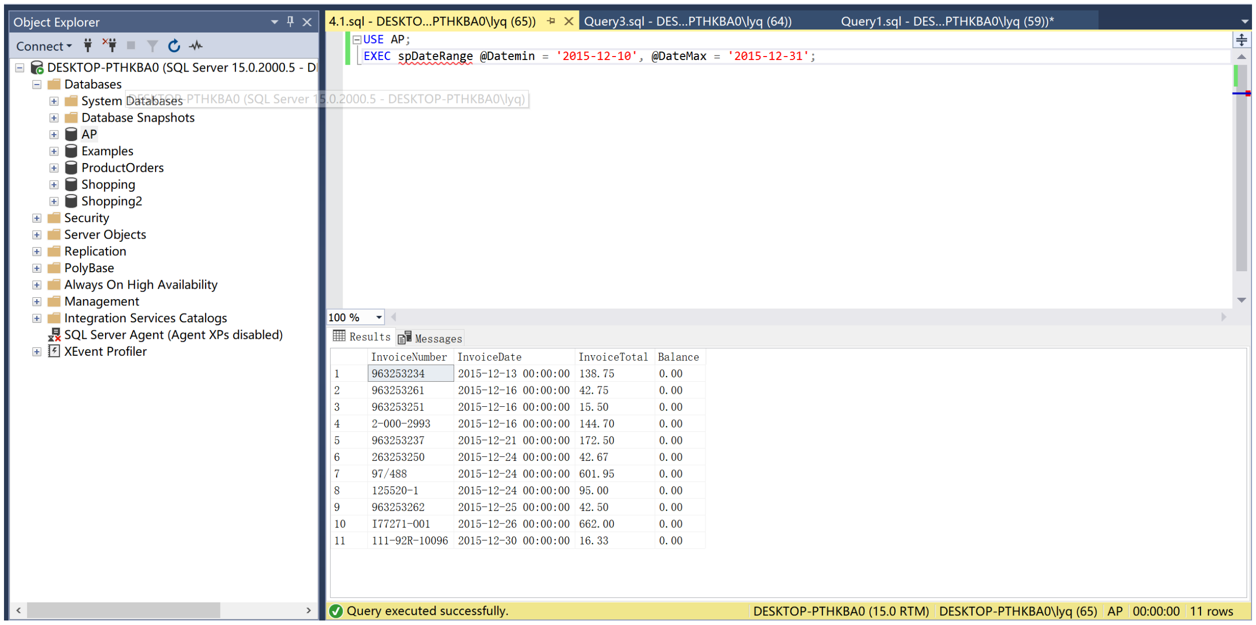
```
GO
CREATE PROC spDateRange
   @DateMin varchar(50)=NULL,
   @DateMax varchar(50)=NULL
AS
 IF @DateMin IS NULL OR @DateMax IS NULL
    THROW 50001, 'Invalid Date.',1;
 IF @DateMin > @DateMax
    THROW 50001, 'Invalid Date.',1;
 ELSE
    SELECT InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal-CreditTotal-PaymentTotal AS Balance
    FROM Invoices
    WHERE ( InvoiceDate > CONVERT (smalldatetime, @DateMin)) AND (InvoiceDate < CONVERT(smalldatetime, @DateMax))
    ORDER BY InvoiceDate ASC;

    EXEC spDateRange @Datemin = '2015-12-10', @DateMax = '2015-12-31';
```

	InvoiceNumber	InvoiceDate	InvoiceTotal	Balance
1	963253234	2015-12-13 00:00:00	138.75	0.00
2	963253261	2015-12-16 00:00:00	42.75	0.00
3	963253251	2015-12-16 00:00:00	15.50	0.00
4	2-000-2993	2015-12-16 00:00:00	144.70	0.00
5	963253237	2015-12-21 00:00:00	172.50	0.00
6	263253250	2015-12-24 00:00:00	42.67	0.00
7	97/488	2015-12-24 00:00:00	601.95	0.00
8	125520-1	2015-12-24 00:00:00	95.00	0.00
9	963253262	2015-12-25 00:00:00	42.50	0.00
10	I77271-001	2015-12-26 00:00:00	662.00	0.00
11	111-92R-10096	2015-12-30 00:00:00	16.33	0.00

Query executed successfully.   DESKTOP-PTHKBA0 (15.0 RTM)   DESKTOP-PTHKBA0\lyq (64)   AP   00:00:00   11 rows

Object Explorer
Connect
DESKTOP-PTHKBA0 (SQL Server 15.0.2000.5 - D
Databases
System Databases
Database Snapshots
AP
Examples
ProductOrders
Shopping
Shopping2
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
Query3.sql - DES...PTHKBA0\lyq (64))*
Query1.sql - DES...PTHKBA0\lyq (59))*
100 %
Results
Messages

4. Code (1) A call to the stored procedure created in question 3 that returns invoices with an InvoiceDate between December 10 and December 31, 2015,



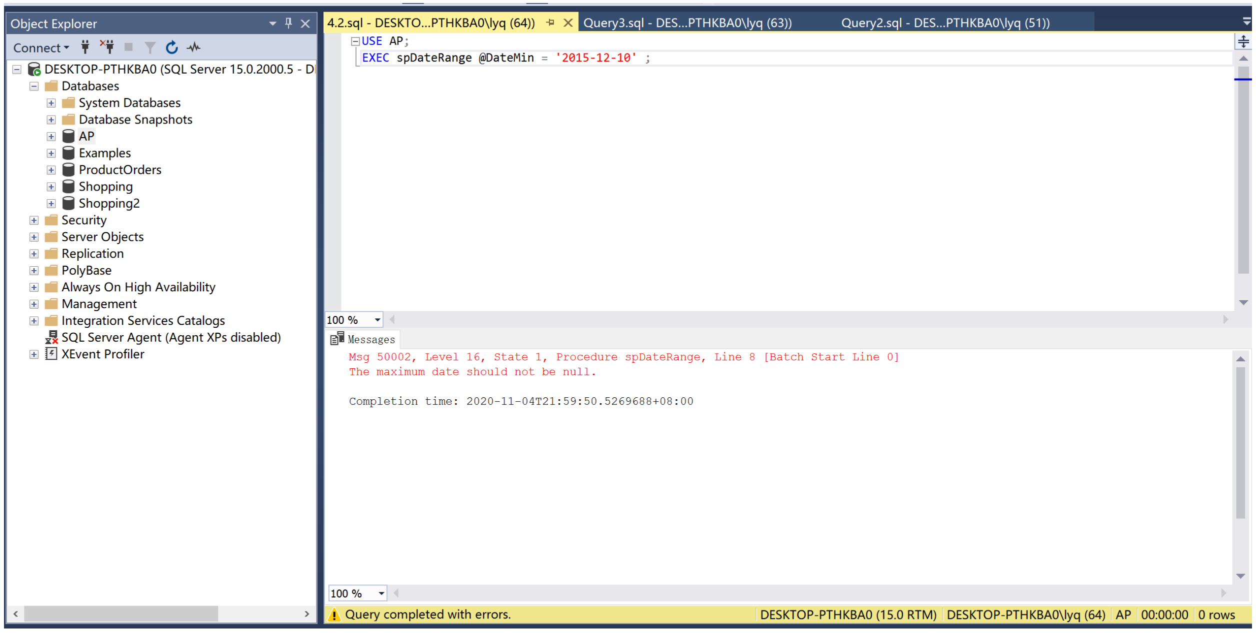
```
USE AP;
EXEC spDateRange @Datemin = '2015-12-10', @DateMax = '2015-12-31';
```

```
     InvoiceNumber  InvoiceDate          InvoiceTotal  Balance
1    963253234      2015-12-13 00:00:00  138.75        0.00
2    963253261      2015-12-16 00:00:00  42.75         0.00
3    963253251      2015-12-16 00:00:00  15.50         0.00
4    2-000-2993     2015-12-16 00:00:00  144.70        0.00
5    963253237      2015-12-21 00:00:00  172.50        0.00
6    263253250      2015-12-24 00:00:00  42.67         0.00
7    97/488         2015-12-24 00:00:00  601.95        0.00
8    125520-1       2015-12-24 00:00:00  95.00         0.00
9    963253262      2015-12-25 00:00:00  42.50         0.00
10   I77271-001     2015-12-26 00:00:00  662.00        0.00
11   111-92R-10096  2015-12-30 00:00:00  16.33         0.00
```

Query executed successfully.   DESKTOP-PTHKBA0 (15.0 RTM)   DESKTOP-PTHKBA0\lyq (65)   AP   00:00:00   11 rows

(2) A call to the stored procedure again that returns invoices with an @DateMin is December 10. These calls should also catch any errors that are raised by the procedure and print the error number and description.

The error number is 50002 and the description is also as below.

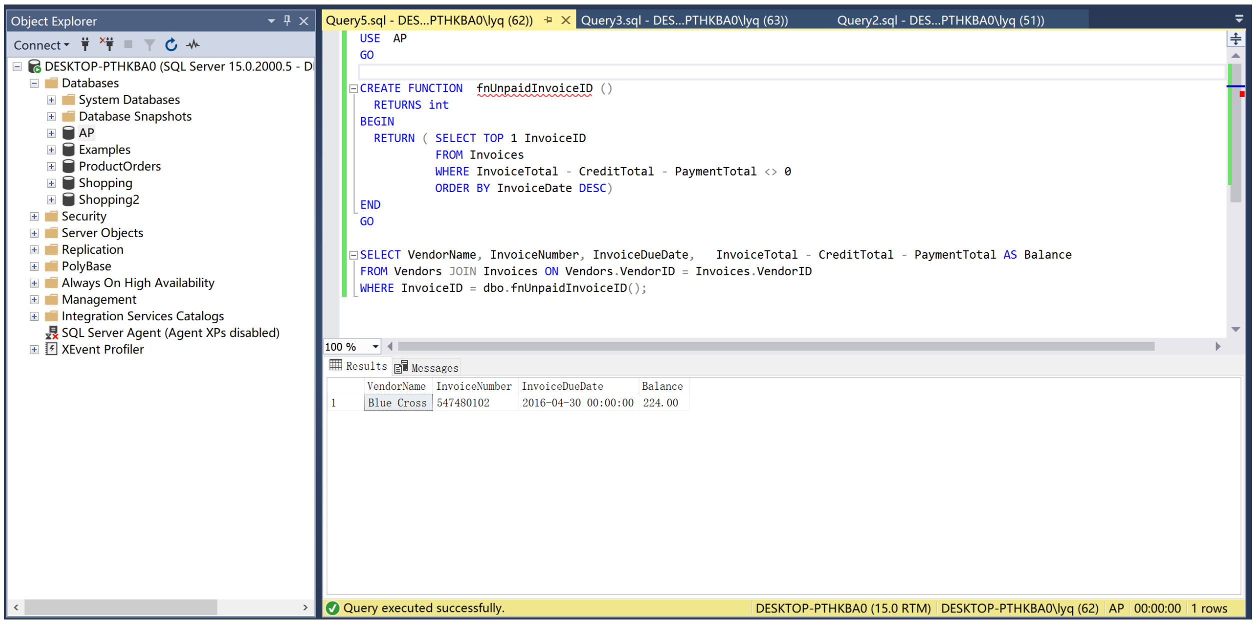


```
USE AP;
EXEC spDateRange @DateMin = '2015-12-10' ;
```

```
Msg 50002, Level 16, State 1, Procedure spDateRange, Line 8 [Batch Start Line 0]
The maximum date should not be null.

Completion time: 2020-11-04T21:59:50.5269688+08:00
```

Query completed with errors.   DESKTOP-PTHKBA0 (15.0 RTM)   DESKTOP-PTHKBA0\lyq (64)   AP   00:00:00   0 rows

5. Create a scalar-valued function named fnUnpaidInvoiceID that returns the InvoiceID of the latest invoice with an unpaid balance. Test the function in the following SELECT statement:
SELECT VendorName, InvoiceNumber, InvoiceDueDate, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
FROM Vendors JOIN Invoices
ON Vendors.VendorID = Invoices.VendorID WHERE InvoiceID = dbo.fnUnpaidInvoiceID();

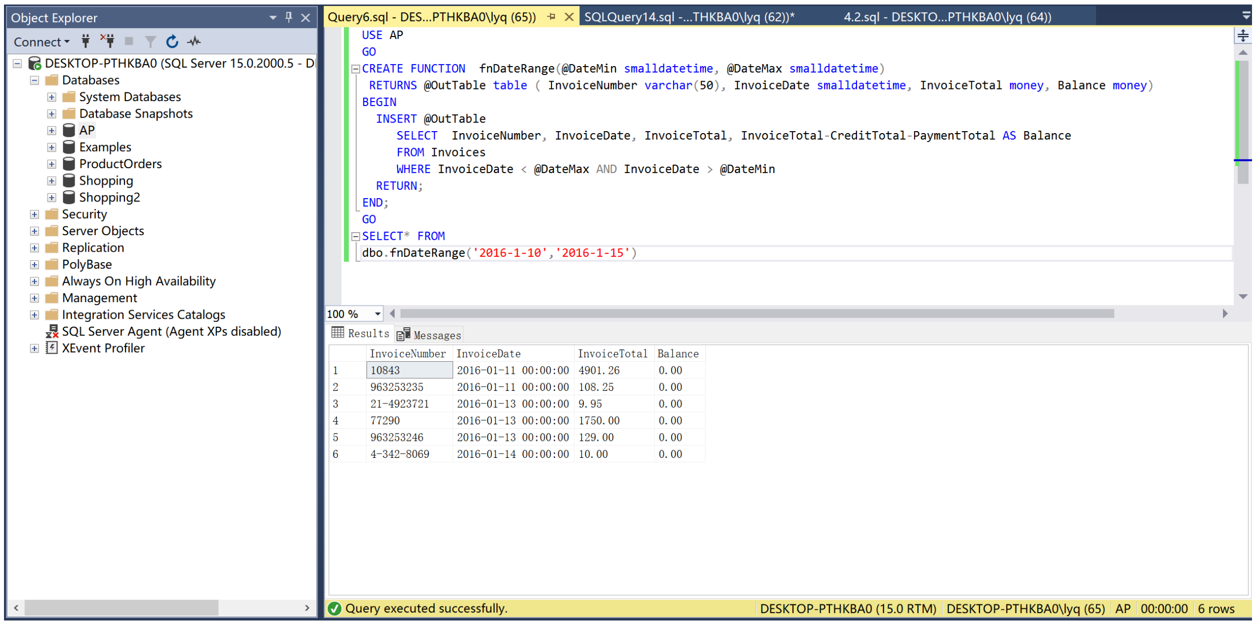

Object Explorer
Connect
DESKTOP-PTHKBA0 (SQL Server 15.0.2000.5 - D
Databases
System Databases
Database Snapshots
AP
Examples
ProductOrders
Shopping
Shopping2
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
Query5.sql - DES...PTHKBA0\lyq (62))
Query3.sql - DES...PTHKBA0\lyq (63))
Query2.sql - DES...PTHKBA0\lyq (51))
100 %
Results
Messages

```
USE  AP
GO

CREATE FUNCTION  fnUnpaidInvoiceID ()
  RETURNS int
BEGIN
  RETURN ( SELECT TOP 1 InvoiceID
           FROM Invoices
           WHERE InvoiceTotal - CreditTotal - PaymentTotal <> 0
           ORDER BY InvoiceDate DESC)
END
GO

SELECT VendorName, InvoiceNumber, InvoiceDueDate,   InvoiceTotal - CreditTotal - PaymentTotal AS Balance
FROM Vendors JOIN Invoices ON Vendors.VendorID = Invoices.VendorID
WHERE InvoiceID = dbo.fnUnpaidInvoiceID();
```

	VendorName	InvoiceNumber	InvoiceDueDate	Balance
1	Blue Cross	547480102	2016-04-30 00:00:00	224.00

Query executed successfully.   DESKTOP-PTHKBA0 (15.0 RTM)   DESKTOP-PTHKBA0\lyq (62)   AP   00:00:00   1 rows

6. Create a table-valued function named fnDateRange, similar to the stored procedure of question 3. The function requires two parameters of data type smalldatetime. Don't validate the parameters. Return a result set that includes the InvoiceNumber, InvoiceDate, InvoiceTotal, and Balance for each invoice for which the InvoiceDate is within the date range. Invoke the function from within a SELECT statement to return those invoices with InvoiceDate between January 10 and January 15, 2016.

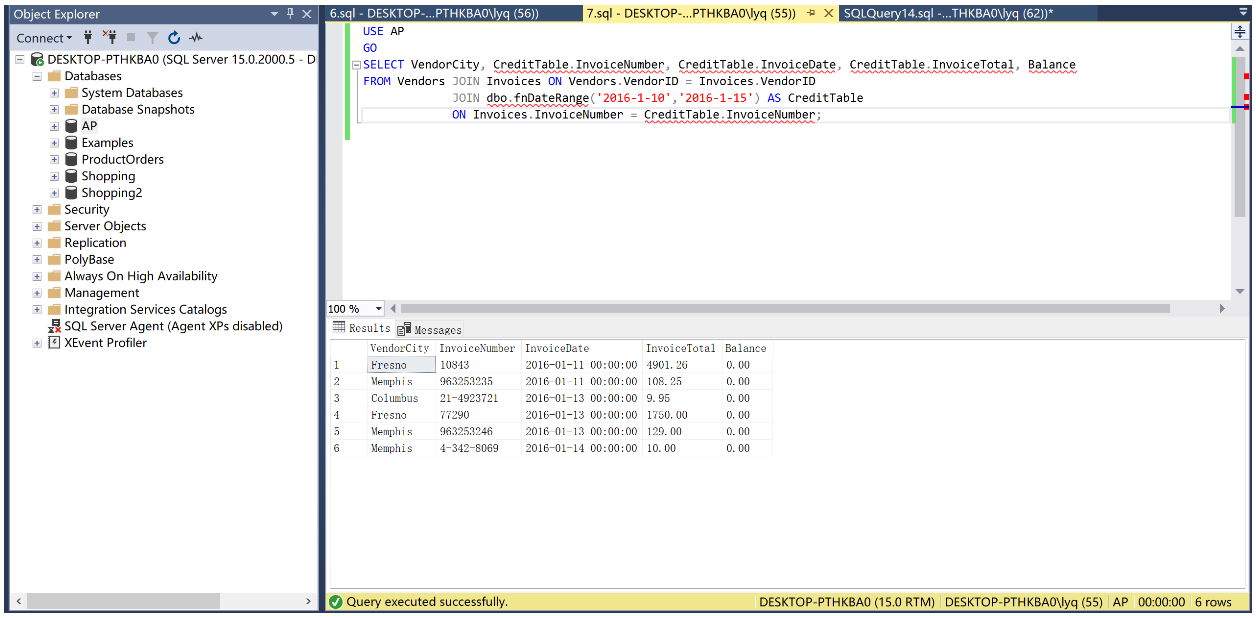


```
USE AP
GO
CREATE FUNCTION  fnDateRange(@DateMin smalldatetime, @DateMax smalldatetime)
 RETURNS @OutTable table ( InvoiceNumber varchar(50), InvoiceDate smalldatetime, InvoiceTotal money, Balance money)
BEGIN
  INSERT @OutTable
     SELECT  InvoiceNumber, InvoiceDate, InvoiceTotal, InvoiceTotal-CreditTotal-PaymentTotal AS Balance
     FROM Invoices
     WHERE InvoiceDate < @DateMax AND InvoiceDate > @DateMin
  RETURN;
END;
GO
SELECT* FROM
dbo.fnDateRange('2016-1-10','2016-1-15')
```

```
   InvoiceNumber InvoiceDate          InvoiceTotal Balance
1  10843         2016-01-11 00:00:00  4901.26      0.00
2  963253235     2016-01-11 00:00:00  108.25       0.00
3  21-4923721    2016-01-13 00:00:00  9.95         0.00
4  77290         2016-01-13 00:00:00  1750.00      0.00
5  963253246     2016-01-13 00:00:00  129.00       0.00
6  4-342-8069    2016-01-14 00:00:00  10.00        0.00
```

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (65) AP 00:00:00 6 rows

7. Use the function you created in question 6 in a SELECT statement that returns five columns: VendorCity and the four columns returned by the function.



```
USE AP
GO
SELECT VendorCity, CreditTable.InvoiceNumber, CreditTable.InvoiceDate, CreditTable.InvoiceTotal, Balance
FROM Vendors JOIN Invoices ON Vendors.VendorID = Invoices.VendorID
             JOIN dbo.fnDateRange('2016-1-10','2016-1-15') AS CreditTable
             ON Invoices.InvoiceNumber = CreditTable.InvoiceNumber;
```

```
   VendorCity InvoiceNumber InvoiceDate          InvoiceTotal Balance
1  Fresno     10843         2016-01-11 00:00:00  4901.26      0.00
2  Memphis    963253235     2016-01-11 00:00:00  108.25       0.00
3  Columbus   21-4923721    2016-01-13 00:00:00  9.95         0.00
4  Fresno     77290         2016-01-13 00:00:00  1750.00      0.00
5  Memphis    963253246     2016-01-13 00:00:00  129.00       0.00
6  Memphis    4-342-8069    2016-01-14 00:00:00  10.00        0.00
```

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (55) AP 00:00:00 6 rows