

```
1. SELECT CustLastName + '' + LEFT(CustFirstName,2) + '.' AS ContactName,  
      SUBSTRING(CustPhone,4,7) AS Phone  
   FROM ProductOrders.dbo.Customers  
 WHERE LEFT(CustPhone,3) = '559'  
 ORDER BY CustLastName ASC, LEFT(CustFirstName,1) ASC
```

The screenshot shows the Microsoft SQL Server Management Studio interface with three tabs open:

- lab3no1.sql - DESKTOP-P37904\owner (32) - Microsoft SQL Server Management Studio
- lab3no2.sql - DESKTOP-P37904\owner (36)
- lab3no3.sql - DESKTOP-P37904\owner (34)

The Object Explorer sidebar shows the database structure for "DESKTOP-P37904\SQLEXPRESS (SQL Server 14.0.1000 - DESKTOP-U)". The "Tables" node under "ProductOrders" is expanded, showing "Customers".

The lab3no3.sql tab contains the following T-SQL code:

```
SELECT CustLastName + ' ' + LEFT(CustFirstName,2) + '.' AS ContactName,
       SUBSTRING(CustPhone,4,7) AS Phone
  FROM ProductOrders.dbo.Customers
 WHERE LEFT(CustPhone,3) = '559'
 ORDER BY CustLastName ASC, LEFT(CustFirstName,1) ASC
```

The Results pane displays the query results:

ContactName	Phone
Danielle Liang	5595000
Hiroko Nakatani	5595825
Nataly Th.	556245

At the bottom, a status bar indicates: "Query executed successfully." and "DESKTOP-P37904\SQLEXPRESS... DESKTOP-P37904\owner... master 00:00:00 3 rows".

2. SELECT InvoiceNumber, (InvoiceTotal-CreditTotal-PaymentTotal) AS BalanceDue
FROM AP.dbo.Invoices
WHERE InvoiceTotal-CreditTotal-PaymentTotal>0
AND InvoiceDueDate<GETDATE()+5;

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'DESKTOP-UP37904\SQLEXPRESS' (master) - Microsoft SQL Server Management Studio. The 'File', 'Edit', 'View', 'Query', 'Project', 'Tools', 'Window', and 'Help' menus are visible at the top. The 'Quick Launch (Ctrl+Q)' button is on the far right.

The 'Object Explorer' pane on the left shows the database structure for 'master'. It includes nodes for 'Logins', 'System Catalog Views', 'System Functions', 'System Procedures', 'System Triggers', 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', 'System Types', 'System Parameters', 'System Variables', 'System Objects', 'System Databases', 'System Snapshots', 'System Snaps', 'System Services', 'System Agents', 'System Replication', 'System Security', 'System Servers', 'System Replication', 'System PolyBase', 'System Management', and 'System XEvent Profiler'.

The 'Results' pane at the bottom displays the results of the executed query:

```
SELECT InvoiceNumber, (InvoiceTotal-CreditTotal-PaymentTotal) AS BalanceDue
FROM AP.dbo.Invoices
WHERE InvoiceTotal-CreditTotal-PaymentTotal > 0
AND InvoiceDueDate = GETDATE() + 5;
```

InvoiceNumber	BalanceDue
1 39104	85.31
2 963251004	52.25
3 31361833	579.42
4 263252069	59.97
5 26325270	67.92
6 26325273	39.75
7 48080	19351.18
8 9802771	500.26
9 131116	99.36
10 0-2436	10076.06
11 547480102	224.00

The status bar at the bottom shows 'Query executed successfully.' and the connection details: DESKTOP-UP37904\SQLEXPRESS ... DESKTOP-UP37904\owner ... master 00:00:00 | 11 rows.

3. SELECT InvoiceNumber, (InvoiceTotal-CreditTotal-PaymentTotal) AS BalanceDue
FROM AP.dbo.Invoices
WHERE InvoiceTotal-CreditTotal-PaymentTotal>0
AND InvoiceDueDate<
CAST(
CAST(YEAR(GETDATE()) AS varchar)
+'-'
+CAST(MONTH(GETDATE()) + 1 AS varchar)
+'-01'
AS smalldatetime)
-1;

lab5n3.sql - DESKTOP-UP37904\SOLEXPRESS.master (DESKTOP-UP37904\owner (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

master | lab5n2.sql - DESKTOP-UP37904\owner (56) | lab5n1.sql - DESKTOP-UP37904\owner (52) | lab5n3.sql - DESKTOP-UP37904\owner (54) |

Object Explorer

Connect -> DESKTOP-UP37904\SOLEXPRESS (SQL Server 14.0.1000 - DESKTOP-UP37904)

Databases

- System Databases
- Database Snapshots
- File and Filegroups
- File and Filegroups
- Examples
- ProductOrders
- Tables
- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.Sales
- dbo.Invoices
- dbo.InvoiceDetails
- dbo.OrderDetails
- dbo.Orders

Views

- External Resources
- Synonyms
- Programmability
- Service Broker
- Storage
- Security

Security

- Server Objects
- Replication
- PolyBase
- Management
- xEvent Profiler

master | New Query | Execute | Undo | Redo | Save | Open | Print | Find | Replace | Refresh | Close |

l5n3.sql -> [Lichen Liang]

In this question, we want to find the last day of each month, for each InvoiceDueDate in the table, we want to calculate its last day of that month, and the date must be before the last day of the month. The InvoiceDueDate has the form yyyy-mm-dd, so we want to generate a date in this form. This requires concatenation and CAST function. To find the last day for each month, which differs from month to month, then it's easier to find the 1st day for the next month, then -1 from it. For example, August 8 of 2008 is expressed as 2008-08-08, we want to set the condition to be before current year(2008), current month(08), last day of the month(29). We need to get 2008-08-31. In order to get to next month, we need month +1. The first day of the month is fixed, so we can concatenate string '-01' where '-' is part of the for To get the year, use CAST(YEAR(GETDATE()) AS varchar) cast it as variable character. To get the month and +1 for next month, use CAST(MONTH(GETDATE()) + 1 AS varchar), also cast it as variable character. Cast this to smalldatetime data type, which will give the form yyyy-mm-dd Finally -1 to get to the previous day/last day of the month. we want the day before the date we got, so < operator.

```
/*SELECT InvoiceNumber, (InvoiceTotal-CreditTotal+PaymentTotal) AS BalanceDue
FROM AP.dbo.Invoices
WHERE InvoiceTotal>CreditTotal+PaymentTotal
AND InvoiceDueDate<
    CAST(
        CAST(YEAR(GETDATE()) AS varchar)
        +CAST(MONTH(GETDATE()) AS varchar)
        +CAST(MONTH(GETDATE()) + 1 AS varchar)
        +'-'+'01'
        AS smalldatetime)
    -1;
```

Results | Messages

InvoiceNumber	BalanceDue
1 38104	85.31
2 96305264	52.25
3 31516133	579.42
4 262525263	59.97
5 262525270	67.92
6 262525273	30.75
7 P-02525273	1851.18
8 96305271	50.50
9 134116	98.38
10 03436	10976.06
11 547408102	224.00

Query executed successfully.

DESKTOP-UP37904\SOLEXPRESS ... DESKTOP-UP37904\owner ... master 00:05:00 | 11 rows

4. SELECT InvoiceNumber,
 (InvoiceTotal-CreditTotal-PaymentTotal) AS BalanceDue,
 RANK() OVER (ORDER BY InvoiceTotal-CreditTotal-PaymentTotal
DESC) AS BalanceRank
FROM AP.dbo.Invoices
WHERE InvoiceTotal-CreditTotal-PaymentTotal>0
 AND InvoiceDueDate<GETDATE()+5;

The screenshot shows a Microsoft SQL Server Management Studio window with the following details:

- Title Bar:** lab5no4.sql - DESKTOP-UP3790A\SQLEXPRESS.master (DESKTOP-UP3790A\owner (56)) - Microsoft SQL Server Management Studio
- Object Explorer:** Shows the database structure for DESKTOP-UP3790A\SQLEXPRESS (SQL Server 14.0.1000 - DESKTOP-UP3790A). It includes nodes for Databases, System Databases, AP, Exemplars, ProductOrders, ProductOrderDetails, Tables, Views, Security, Server Objects, Replication, PolyBase, Management, and XEvent Profiler.
- Query Editor:** Contains a T-SQL script named lab5no4.sql. The script uses the RANK() function to add a new column BalanceRank to the AP.dbo.Invoices table. The script is as follows:

```

--Question 3
--Get new column BalanceRank, using RANK() function, where balance is InvoiceTotal-CreditTotal-PaymentTotal.
--using the RANK() OVER we will get a rank, use ORDER BY get order of balance due in descending(DESC), and the column name BalanceRank
--*/
SELECT InvoiceNumber,
       (InvoiceTotal CreditTotal PaymentTotal) AS BalanceDue,
       RANK() OVER (ORDER BY InvoiceTotal CreditTotal PaymentTotal DESC) AS BalanceRank
  FROM AP.dbo.Invoices
 WHERE InvoiceTotal CreditTotal PaymentTotal >
      AND InvoiceDueDate < GETDATE() )$;

```
- Results Grid:** Displays the results of the executed query. The grid has columns: InvoiceNumber, BalanceDue, and BalanceRank. The data is as follows:

	InvoiceNumber	BalanceDue	BalanceRank
1	498901	13351.18	
2	52436	10976.06	2
3	31301833	579.42	3
4	0902771	503.20	4
5	547480102	224.00	5
6	134116	90.36	6
7	39104	85.31	7
8	263253270	67.92	8
9	263253268	59.97	9
10	963253264	52.25	10
11	263253273	30.75	11
- Status Bar:** Shows "Query executed successfully." and other status information like Line 9, Col 46, Ch 46, INS.

Remarks

In this lab we further practiced using functions, such as date/time, string, ranking, etc. Since there are so many functions there is no way to practice them all in one lab, I think question 3 is a very good practice in general. This lab is very efficient practice for lecture. The challenge would be more complicated and specific requirements that need more functions.