



Lab 5: Subqueries Functions Solution

1. Write a SELECT statement that returns two columns based on the Vendors table. The first column, Contact, is the vendor contact name in this format: first name followed by last initial (for example, "John S.") The second column, Phone, is the VendorPhone column without the area code. Only return rows for those vendors in the 916 area code. Sort the results set by first name, then last name.

```
SELECT VendorContactFName + ' ' +  
    LEFT(VendorContactLName, 1) + '.' AS Contact,  
    SUBSTRING(VendorPhone,7,8) AS Phone  
FROM Vendors  
WHERE LEFT(VendorPhone,4) = '916'  
ORDER BY Contact;
```

A screenshot of the Microsoft SQL Server Management Studio interface. The top menu bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The Object Explorer on the left shows a tree view of the database structure, including Databases, System Databases, Database Snapshots, AP, Database Diagrams, Tables, Views, Synonyms, Programmability, Service Broker, Storage, Security, Examples, Product Orders, and ReportServer\$SQLSERVER. The central query editor window displays the SQL query from the previous block, which is highlighted with a red rectangle. Below the query editor, the Results pane shows the output of the query, which is a table with two columns: Contact and Phone. The table contains three rows of data, which are also highlighted with a red rectangle. The status bar at the bottom indicates that the query was executed successfully and shows the execution time as 00:00:00.3 rows.

Contact	Phone
Anton M.	555-6670
Julissa D.	555-4911
Wendy S.	555-8100

2. Write a SELECT statement that returns the InvoiceNumber and balance due for every invoice with a non-zero balance and an InvoiceDueDate that's less than 10 days from today (i.e. InvoiceDueDate < today's date + 10).

```
SELECT InvoiceNumber,  
       InvoiceTotal - CreditTotal - PaymentTotal AS Balance  
FROM Invoices  
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0 AND  
       InvoiceDueDate < GETDATE() + 10;
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-S32HEE2\SQLSERVER (SQL Server 12.0.4100.1 - DE...)'.

The SQL Query window shows the following query:

```
SELECT InvoiceNumber,  
       InvoiceTotal - CreditTotal - PaymentTotal AS Balance  
FROM Invoices  
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0 AND  
       InvoiceDueDate < GETDATE() + 10;
```

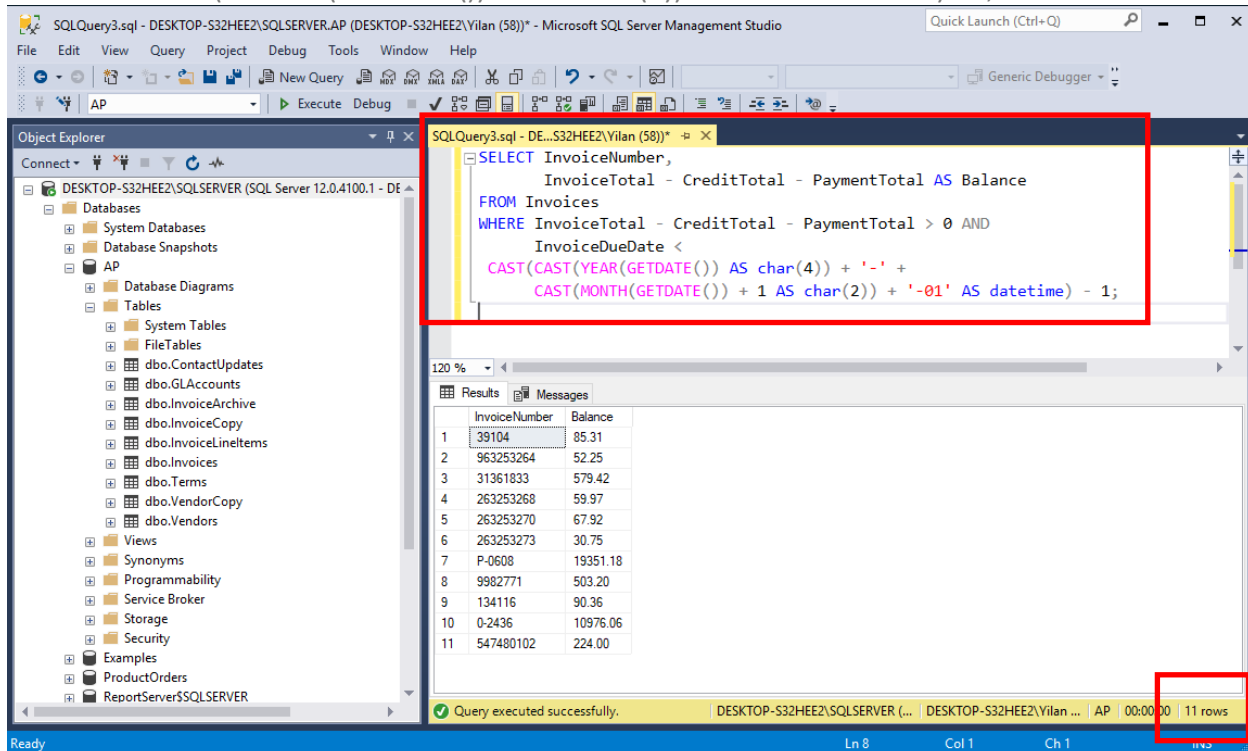
The Results window displays the following data:

	InvoiceNumber	Balance
1	39104	85.31
2	963253264	52.25
3	31361833	579.42
4	263253268	59.97
5	263253270	67.92
6	263253273	30.75
7	P-0608	19351.18
8	9982771	503.20
9	134116	90.36
10	0-2436	10976.06
11	547480102	224.00

The status bar at the bottom indicates: Query executed successfully. DESKTOP-S32HEE2\SQLSERVER (... DESKTOP-S32HEE2\Yilan ... AP 00:00:00 11 rows

3. Modify the search expression for InvoiceDueDate from the solution for question 2. Rather than 10 days from today, return invoices due before the last day of the current month.

```
SELECT InvoiceNumber,  
       InvoiceTotal - CreditTotal - PaymentTotal AS Balance  
FROM Invoices  
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0 AND  
       InvoiceDueDate <  
       CAST(CAST(YEAR(GETDATE()) AS char(4)) + '-' +  
            CAST(MONTH(GETDATE()) + 1 AS char(2)) + '-01' AS datetime) - 1;
```



4. Add a column to the query described in question 2 that uses the RANK() function to return a column named BalanceRank that ranks the balance due in ascending order.

```
SELECT InvoiceNumber,  
       InvoiceTotal - CreditTotal - PaymentTotal AS Balance,  
       RANK() OVER (ORDER BY InvoiceTotal - CreditTotal -  
                       PaymentTotal ASC) AS BalanceRank  
FROM Invoices  
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0 AND  
       InvoiceDueDate < GETDATE() + 10;
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'AP' database selected. The right pane shows a query window with the following SQL query:

```
SELECT InvoiceNumber,  
       InvoiceTotal - CreditTotal - PaymentTotal AS Balance,  
       RANK() OVER (ORDER BY InvoiceTotal - CreditTotal -  
                       PaymentTotal ASC) AS BalanceRank  
FROM Invoices  
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0 AND  
       InvoiceDueDate < GETDATE() + 10;
```

The query results are displayed in a table with the following columns: InvoiceNumber, Balance, and BalanceRank. The results are sorted by BalanceRank in ascending order.

InvoiceNumber	Balance	BalanceRank
263253273	30.75	1
963253264	52.25	2
263253268	59.97	3
263253270	67.92	4
39104	85.31	5
134116	90.36	6
547480102	224.00	7
9982771	503.20	8
31361833	579.42	9
0-2436	10976.06	10
P-0608	19351.18	11

The status bar at the bottom indicates that the query executed successfully and returned 11 rows.