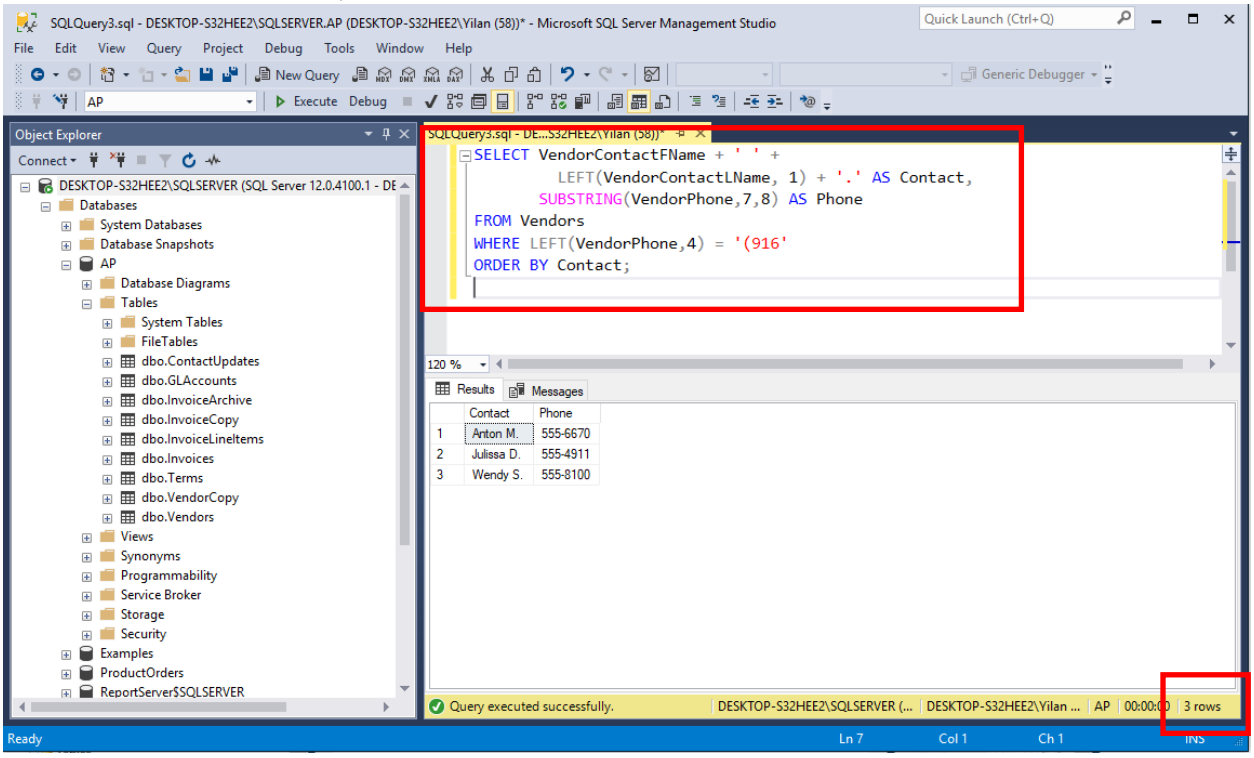
Lab 5: Subqueries Functions Solution

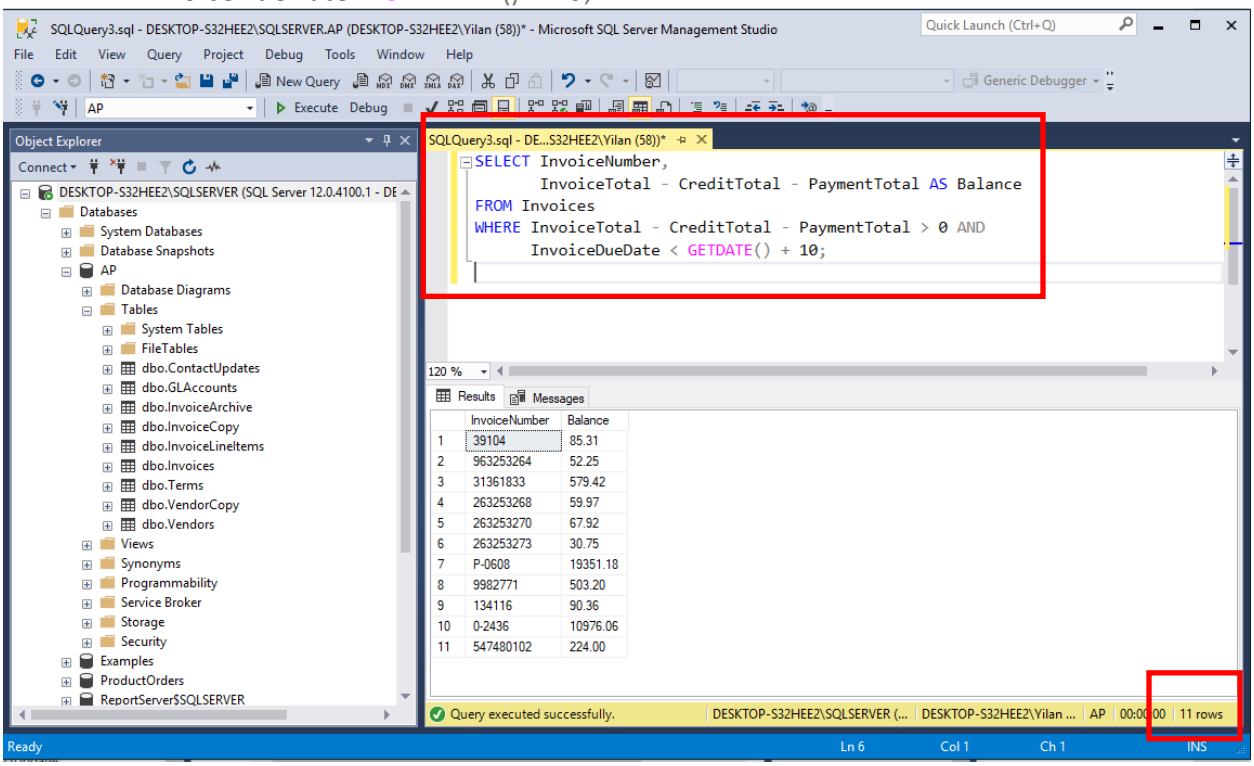
1. Write a SELECT statement that returns two columns based on the Vendors table. The first column, Contact, is the vendor contact name in this format: first name followed by last initial (for example, "John S.") The second column, Phone, is the VendorPhone column without the area code. Only return rows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



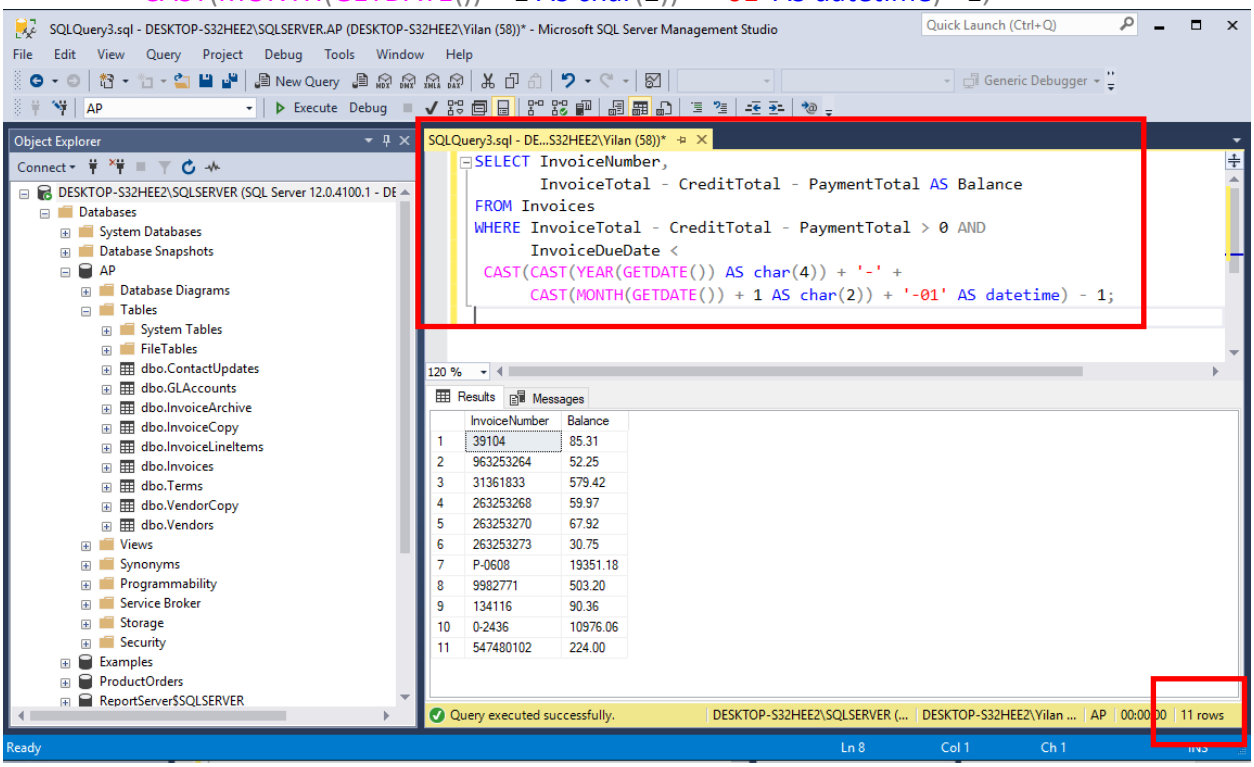
2. Write a SELECT statement that returns the InvoiceNumber and balance due for every invoice with a non-zero balance and an InvoiceDueDate that's less than 10 days from today (i.e. InvoiceDueDate < today's date + 10).

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



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;
```