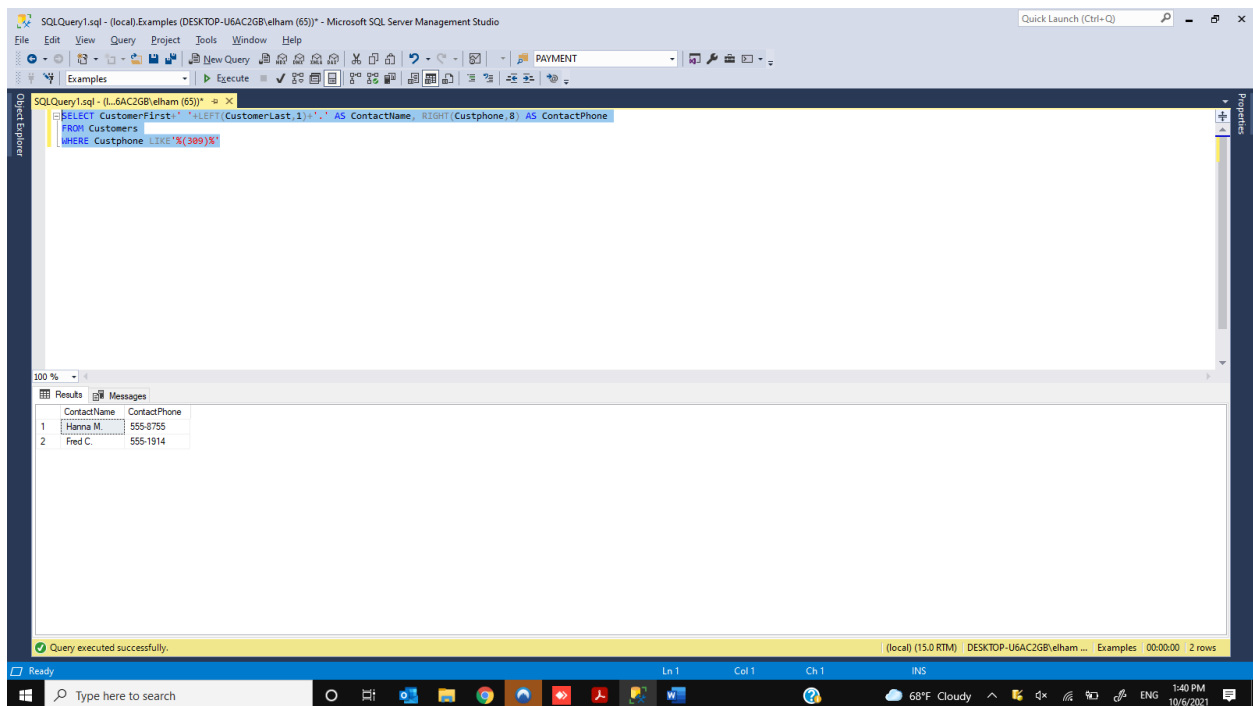


1. Write a SELECT statement that returns two columns based on the Customers table. The first column, ContactName, is the customer's name in this format: Customers first name (i.e. CustomerFirst column) followed by first letter of Customers last name (i.e. CustomerLast column) followed by a dot (for example, the format must look like, "Maria A."). The second column, ContactPhone, is the CustPhone column without the area code. Only return rows for those customers in the '309' area code. Sort the results set by Customers first name in ascending order. Use Examples database.

```
SELECT CustomerFirst+' '+LEFT(CustomerLast,1)+'.' AS ContactName, RIGHT(Custphone,8) AS ContactPhone
FROM Customers
WHERE Custphone LIKE '%(309)%';
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query in the 'SQLQuery1.sql' file. The query is as follows:

```
SELECT CustomerFirst+' '+LEFT(CustomerLast,1)+'.' AS ContactName, RIGHT(Custphone,8) AS ContactPhone
FROM Customers
WHERE Custphone LIKE '%(309)%';
```

Below the query editor, the 'Results' pane shows the output of the query. It contains two columns: 'ContactName' and 'ContactPhone'. The results are as follows:

ContactName	ContactPhone
Hanna M.	555-8755
Fred C.	555-1914

The status bar at the bottom indicates that the query was executed successfully. The taskbar at the bottom shows the system clock as 1:40 PM on 10/6/2021.

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-PaymentTotal + CreditTotal AS BalanceDue
FROM PaidInvoices
WHERE Invoicetotal - PaymentTotal + CreditTotal <> 0
AND InvoiceDueDate < GETDATE() + 10;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

```
SELECT InvoiceNumber, Invoicetotal-PaymentTotal+CreditTotal AS BalanceDue
FROM PaidInvoices
WHERE Invoicetotal - PaymentTotal + CreditTotal <> 0
AND InvoiceDueDate < GETDATE() + 10
```

The Object Explorer on the left shows the database structure, including tables like Customers, Departments, Employees, and Invoices. The Results pane at the bottom displays the output of the query:

	InvoiceNumber	BalanceDue
1	97522	400.00
2	02060	4591.90

The status bar at the bottom indicates that the query was executed successfully and returned 2 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 - PaymentTotal + CreditTotal AS BalanceDue
FROM PaidInvoices
WHERE Invoicetotal - PaymentTotal + CreditTotal <> 0
AND InvoiceDueDate < EOMONTH(GETDATE());
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
SELECT InvoiceNumber, Invoicetotal - PaymentTotal + CreditTotal AS BalanceDue
FROM PaidInvoices
WHERE Invoicetotal - PaymentTotal + CreditTotal <> 0
AND InvoiceDueDate < EOMONTH(GETDATE());
```

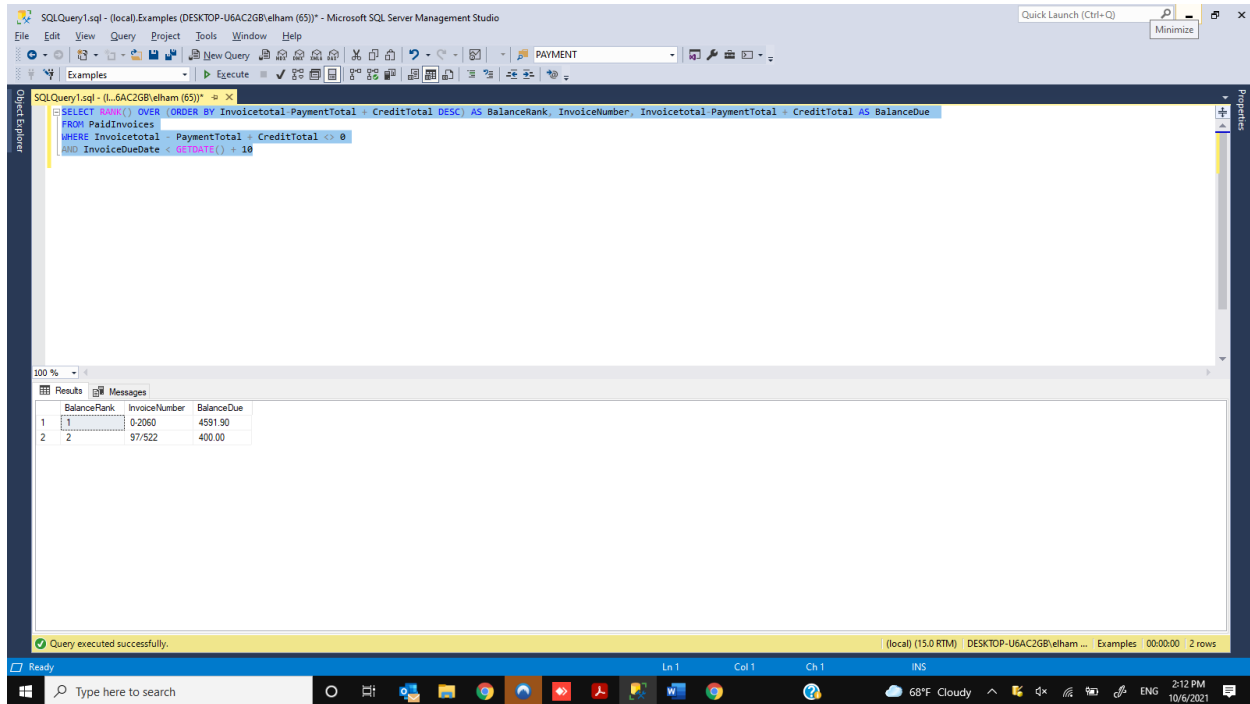
The query has been executed successfully, and the results are displayed in the Results pane. The results show two rows of data:

InvoiceNumber	BalanceDue
1	400.00
2	4591.90

The status bar at the bottom indicates that the query was executed successfully and returned 2 rows.

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 descending order.

```
SELECT InvoiceNumber, Invoicetotal-PaymentTotal + CreditTotal AS BalanceDue
FROM PaidInvoices
WHERE Invoicetotal - PaymentTotal + CreditTotal <> 0
AND InvoiceDueDate < GETDATE() + 10;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
SELECT RANK() OVER (ORDER BY Invoicetotal-PaymentTotal + CreditTotal DESC) AS BalanceRank, InvoiceNumber, Invoicetotal-PaymentTotal + CreditTotal AS BalanceDue
FROM PaidInvoices
WHERE Invoicetotal - PaymentTotal + CreditTotal <> 0
AND InvoiceDueDate < GETDATE() + 10
```

The Results pane shows the output of the query, which is a table with three columns: BalanceRank, InvoiceNumber, and BalanceDue. The results are as follows:

BalanceRank	InvoiceNumber	BalanceDue
1	0-2060	4591.90
2	97/522	400.00

The status bar at the bottom indicates that the query was executed successfully, returning 2 rows.

Thank you for your time.

Sincerely,

Seyed Alireza Zarrin Mehr