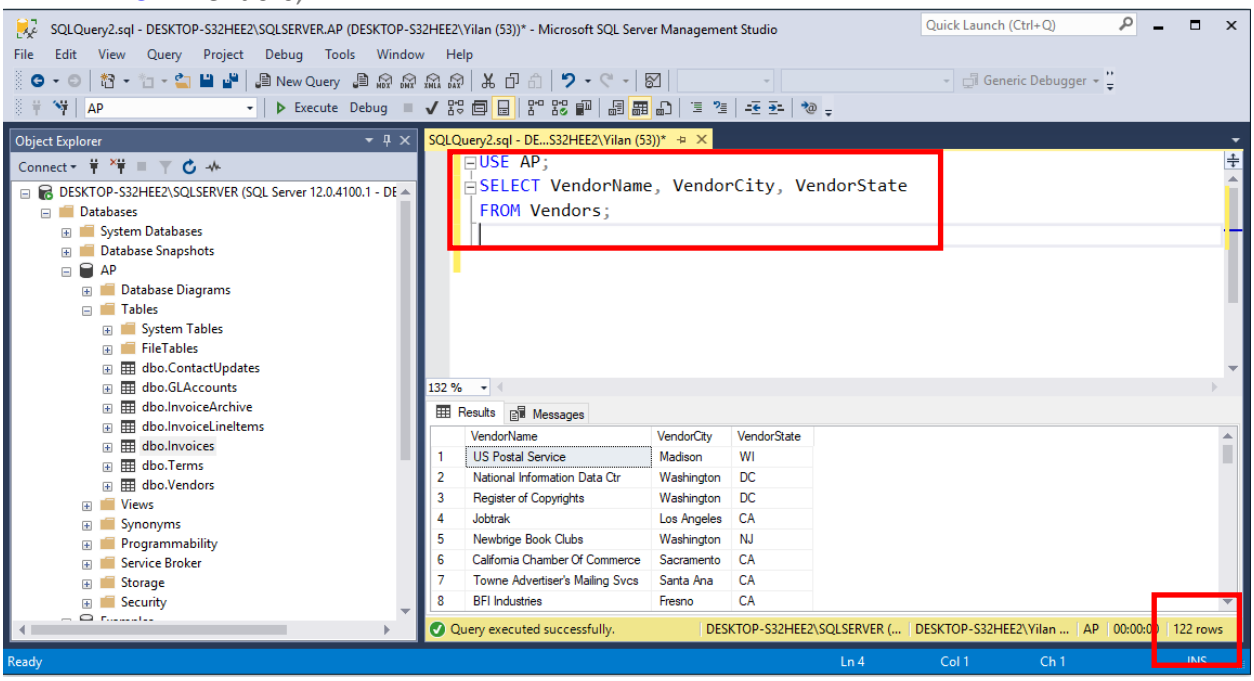
Lab 2: Select, Table Joins, Unions Solution

1. Write a SELECT statement that returns three columns from the Vendors table: VendorName VendorCity and VendorState.

```
USE AP;
SELECT VendorName, VendorCity, VendorState
FROM Vendors;
```

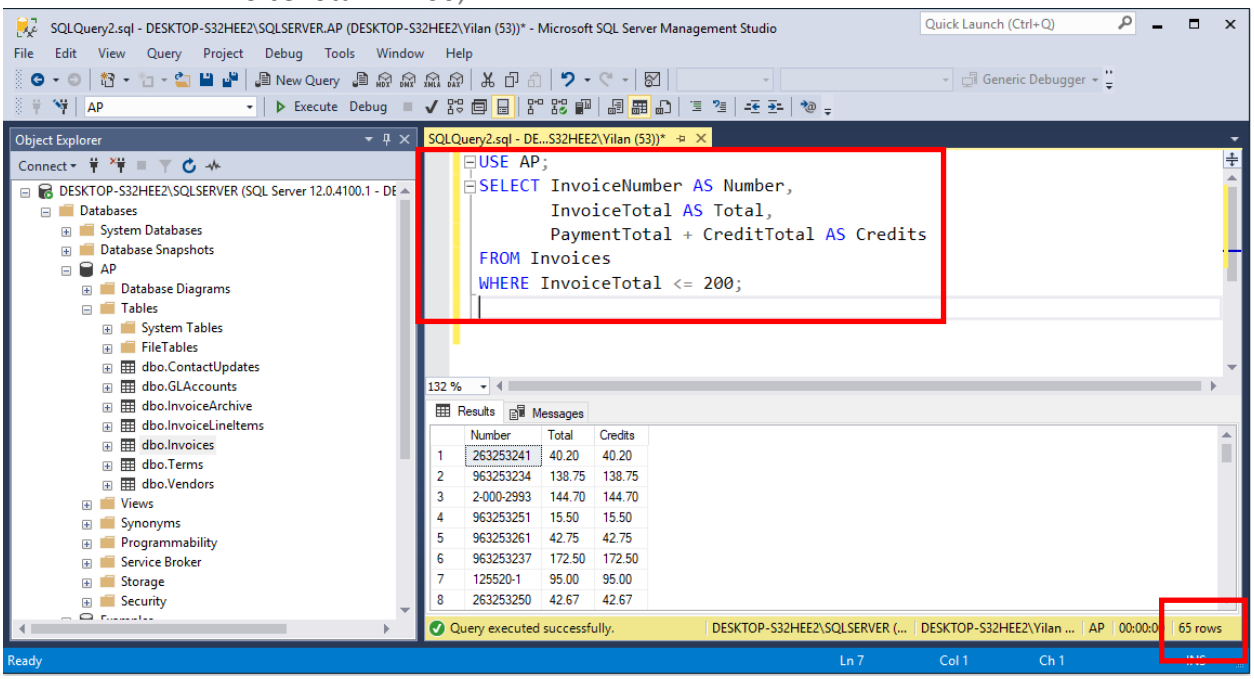


2. Write a SELECT statement that returns three columns from the Invoices table, named Number, Total, and Credits:

Number	Column alias for the InvoiceNumber column
Total	Column alias for the InvoiceTotal column
Credits	Column alias for the sum of the PaymentTotal and CreditTotal columns

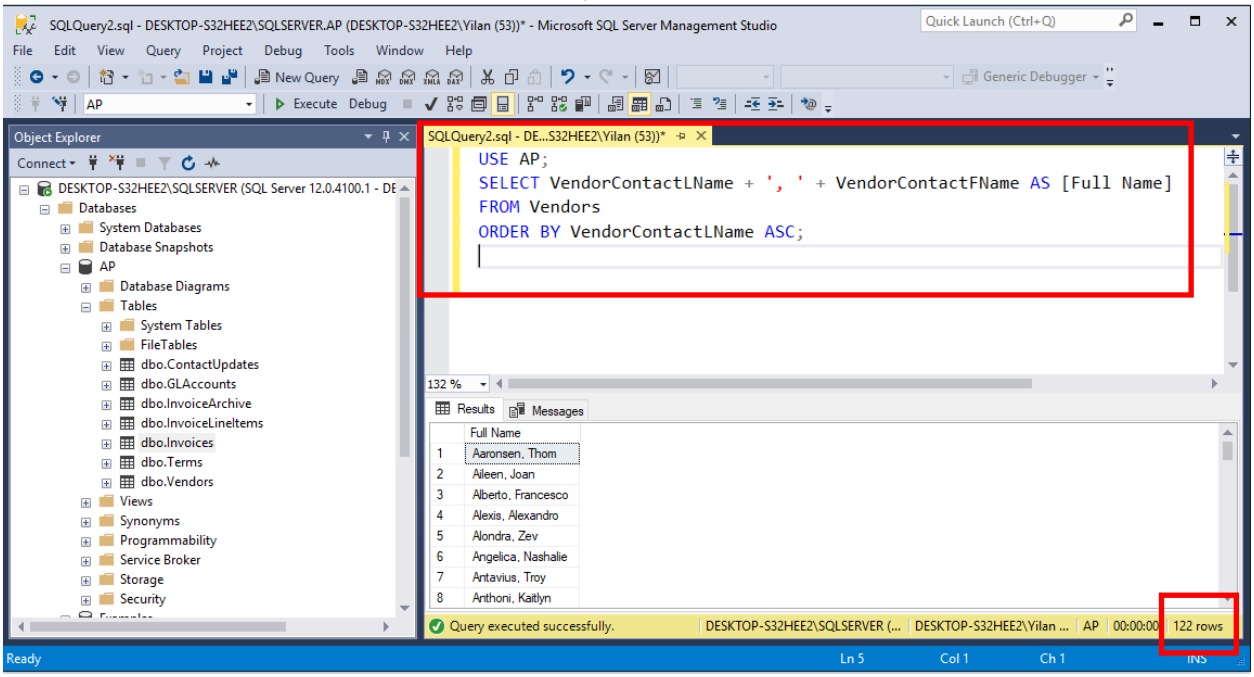
And filter for invoices with an InvoiceTotal that's less or equal to $200.

```
USE AP;
SELECT InvoiceNumber AS Number,
    InvoiceTotal AS Total,
    PaymentTotal + CreditTotal AS Credits
FROM Invoices
WHERE InvoiceTotal <= 200;
```



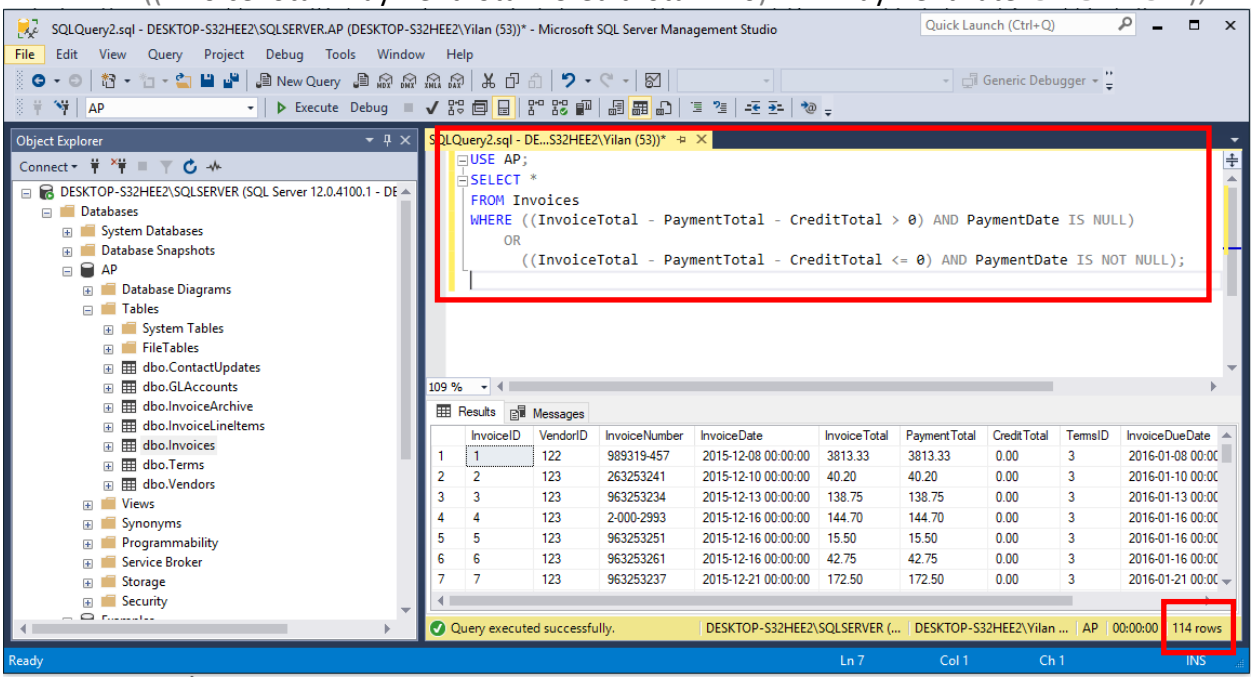
3. Write a SELECT statement that returns one column from the Vendors table named "Full Name". Create this column from the VendorContactFName and VendorContactLName columns. Format it as follows: last name, comma, first name. Sort the result set by last name from "A-Z".

```
USE AP;
SELECT VendorContactLName + ', ' + VendorContactFName AS [Full Name]
FROM Vendors
ORDER BY VendorContactLName ASC;
```



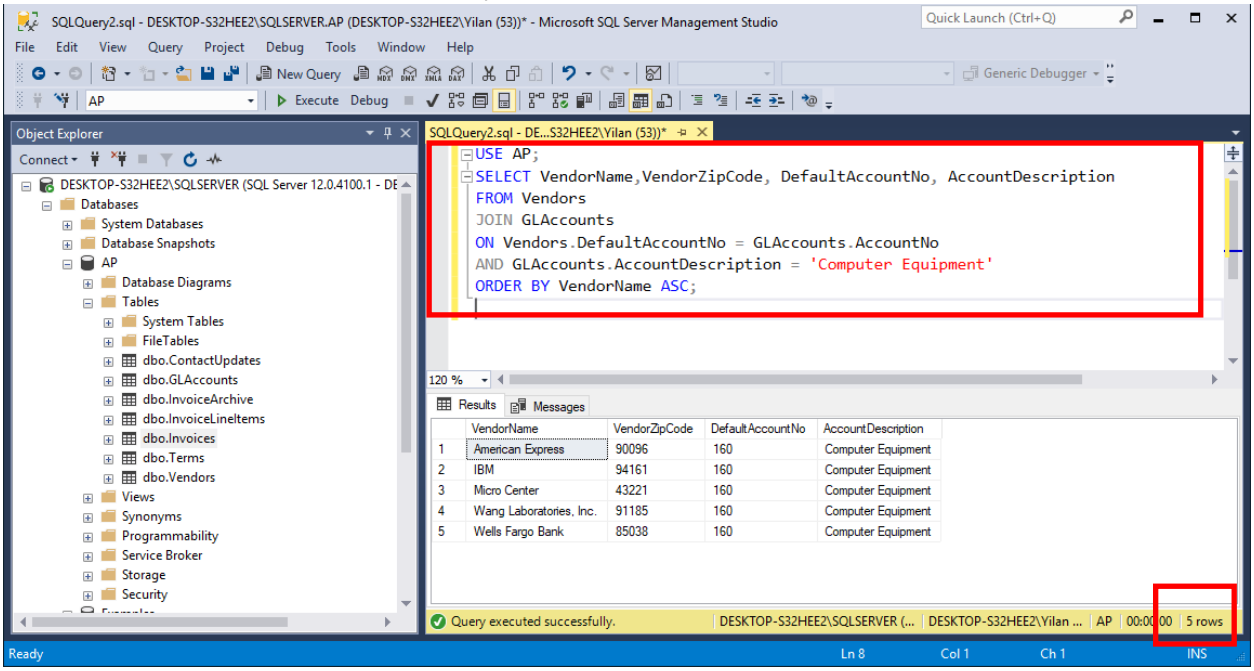
4. Write a SELECT statement that determines whether the PaymentDate column of the Invoices table has any valid values. To be valid, PaymentDate must be a null value if there is a balance due and a non-null value of there is no balance due. Code a compound condition in the WHERE clause that tests for these conditions. (Balance: InvoiceTotal minus the sum of PaymentTotal and CreditTotal)

```
USE AP;
SELECT *
FROM Invoices
WHERE ((InvoiceTotal - PaymentTotal - CreditTotal > 0) AND PaymentDate IS NULL)
        OR
    ((InvoiceTotal - PaymentTotal - CreditTotal <= 0) AND PaymentDate IS NOT NULL);
```



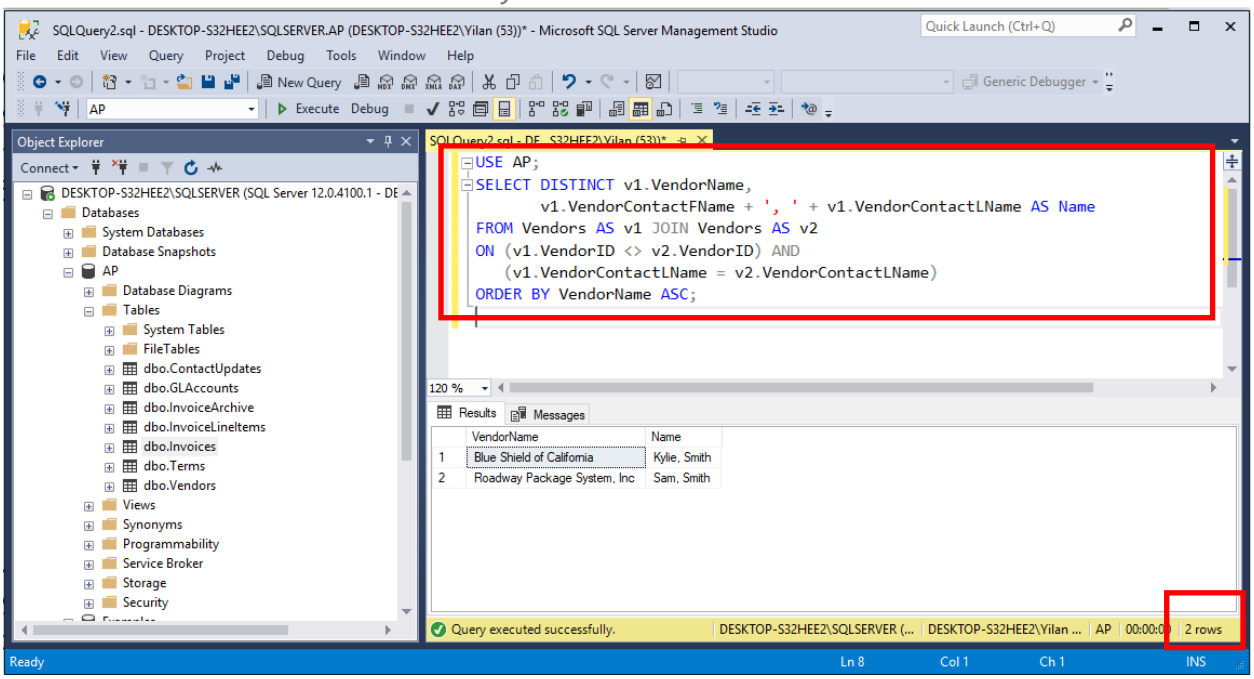
5. Write a SELECT statement that returns four columns: VendorName, VendorZipCode, DefaultAccountNo and AccountDescription from the Vendors table and GLAccounts table. The result set should have one row for each vendor, with the zipcode, account number and account description for that vendor's default account number. And filter for Vendors whose AccountDescription is Computer Equipment and sort the result set by VendorName from A to Z.

```
USE AP;
SELECT VendorName,VendorZipCode, DefaultAccountNo, AccountDescription
FROM Vendors
JOIN GLAccounts
ON Vendors.DefaultAccountNo = GLAccounts.AccountNo
AND GLAccounts.AccountDescription = 'Computer Equipment'
ORDER BY VendorName ASC;
```



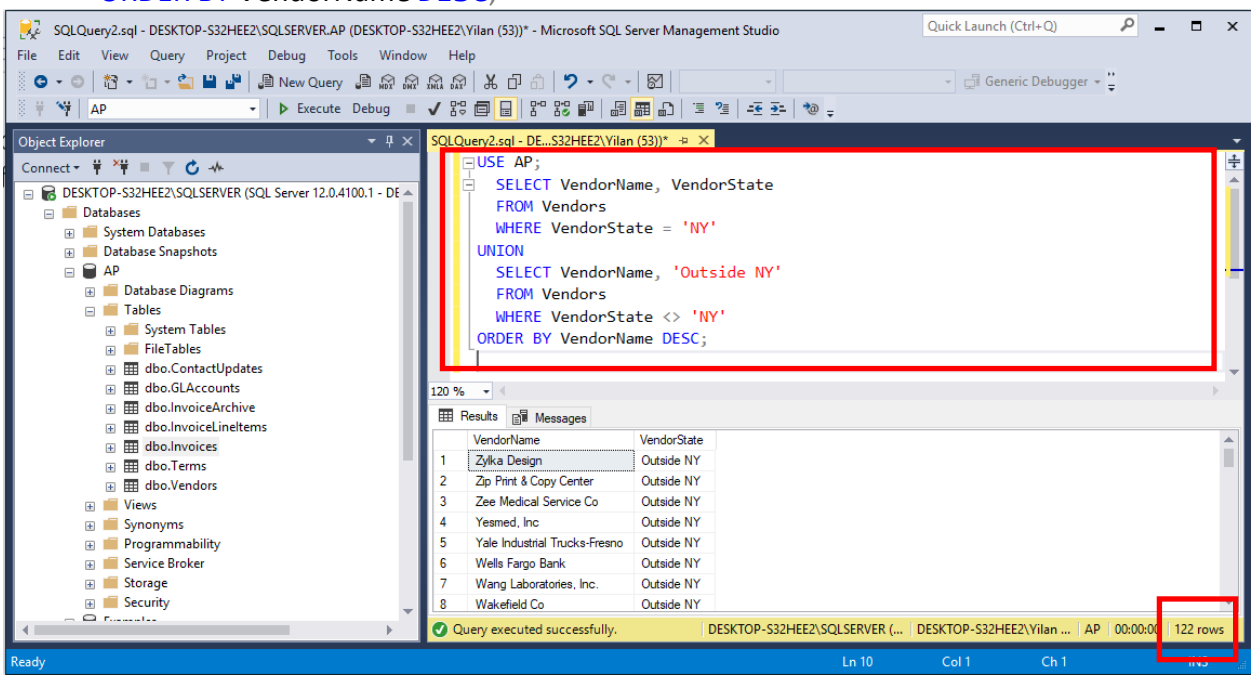
6. Write a SELECT statement that returns two columns: VendorName and Name (A concatenation of VendorContactFName and VendorContactLName, with a comma in between). The result set should have one row for each vendor whose contact has the same last name as another vendor's contact. Sort the final result set by VendorName from A to Z.

```
USE AP;
SELECT DISTINCT v1.VendorName,
    v1.VendorContactFName + ', ' + v1.VendorContactLName AS Name
FROM Vendors AS v1 JOIN Vendors AS v2
ON (v1.VendorID <> v2.VendorID) AND
  (v1.VendorContactLName = v2.VendorContactLName)
ORDER BY  VendorName  ASC;
```



7. Use the UNION operator to generate a result set consisting of two columns from the Vendors table: VendorName and VendorState. If the vendor is in New York, the VendorState value should be "NY"; otherwise, the vendorState value should be "Outside NY". Sort the final result set by VendorName from Z to A.

```
USE AP;
 SELECT VendorName, VendorState
 FROM Vendors
 WHERE VendorState = 'NY'
UNION
 SELECT VendorName, 'Outside NY'
 FROM Vendors
 WHERE VendorState <> 'NY'
ORDER BY VendorName DESC;
```



8. Write a SELECT statement that returns two columns from the GLAccounts table: AccountNo and AccountDescription. The result set should have one row for each account number that has never been used (i.e. AccountNo in InvoiceLineItems table has null value). Sort the final result set by AccountNo. (HINT: join GLAccounts table and InvoiceLineItems table.)

```
USE AP;
SELECT GLAccounts.AccountNo, AccountDescription
FROM GLAccounts LEFT JOIN InvoiceLineItems
  ON GLAccounts.AccountNo = InvoiceLineItems.AccountNo
WHERE InvoiceLineItems.AccountNo IS NULL
ORDER BY GLAccounts.AccountNo;
```