CSE 581 : INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

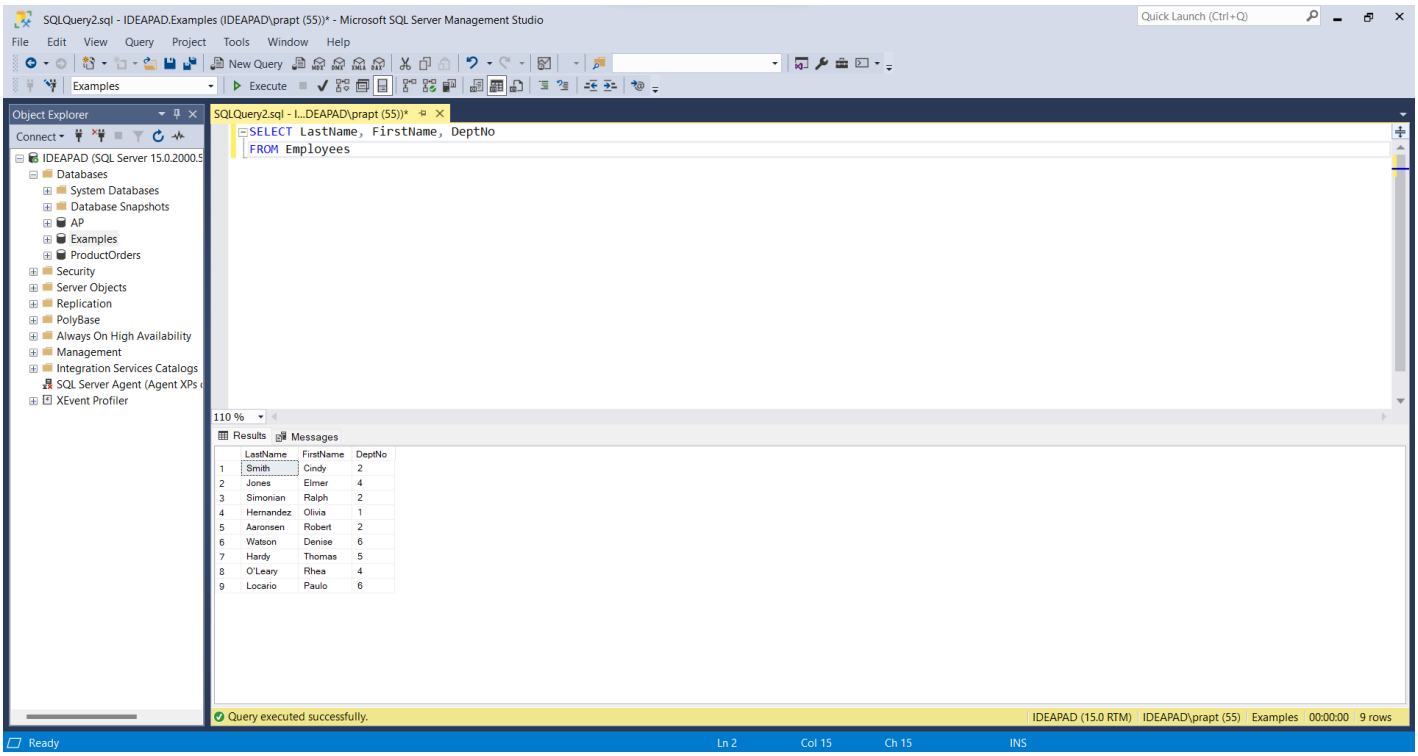
LAB : 2

DATE : 9/14/2022

Q1) Write a SELECT statement that returns three columns from the Employees table: LastName, FirstName, and DeptNo. ***Use Examples database.***

Ans: SELECT LastName, FirstName, DeptNo

FROM Employees



Comment: SELECT statement is used to retrieve three columns from the Employees table having five columns by specifying the names of the columns that have to be displayed. All 9 rows from the Employees table are displayed.

Remark: Thus, for displaying records, SELECT statement is used.

Q2) Write a SELECT statement that returns two columns from the Employees table, named 'Name', and 'DeptNumber':

Name - Column alias for the concatenated format of LastName and FirstName columns

(Format: LastName followed by comma and space followed by FirstName)

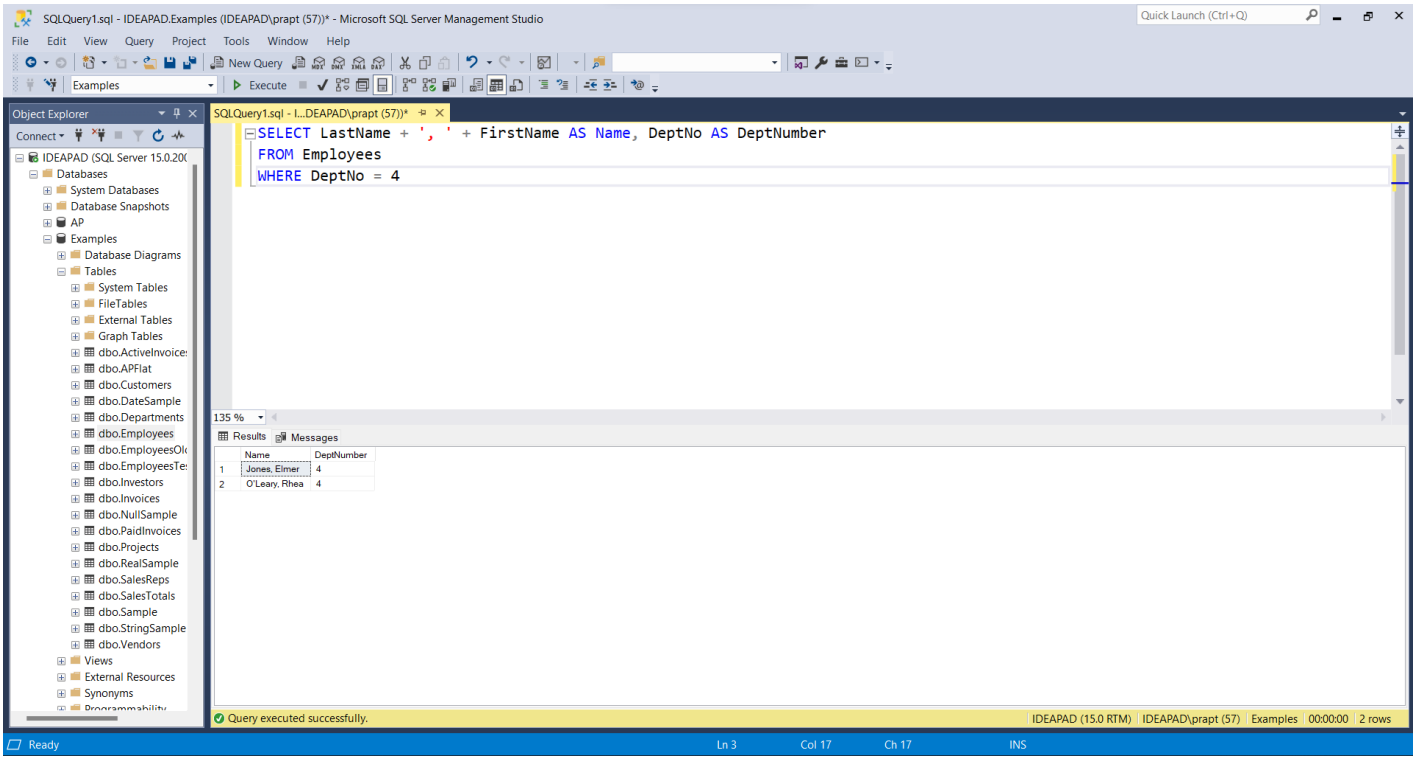
DeptNumber - Column alias for the DeptNo column

And filter for Employees with DeptNo value as 4. ***Use Examples database.***

Ans: SELECT LastName + ', ' + FirstName AS Name, DeptNo AS DeptNumber

FROM Employees

WHERE DeptNo = 4



Comment: SELECT statement is used to display two columns from which the first column is the concatenation of two columns of the Employees table namely LastName and FirstName which is done using the '+' operator and the second column is the DeptNo column. Here, while displaying data different column names are given using AS keyword. Moreover, only the records that have DeptNo = 4 are to be displayed for which WHERE keyword is used. So, only 2 rows are displayed.

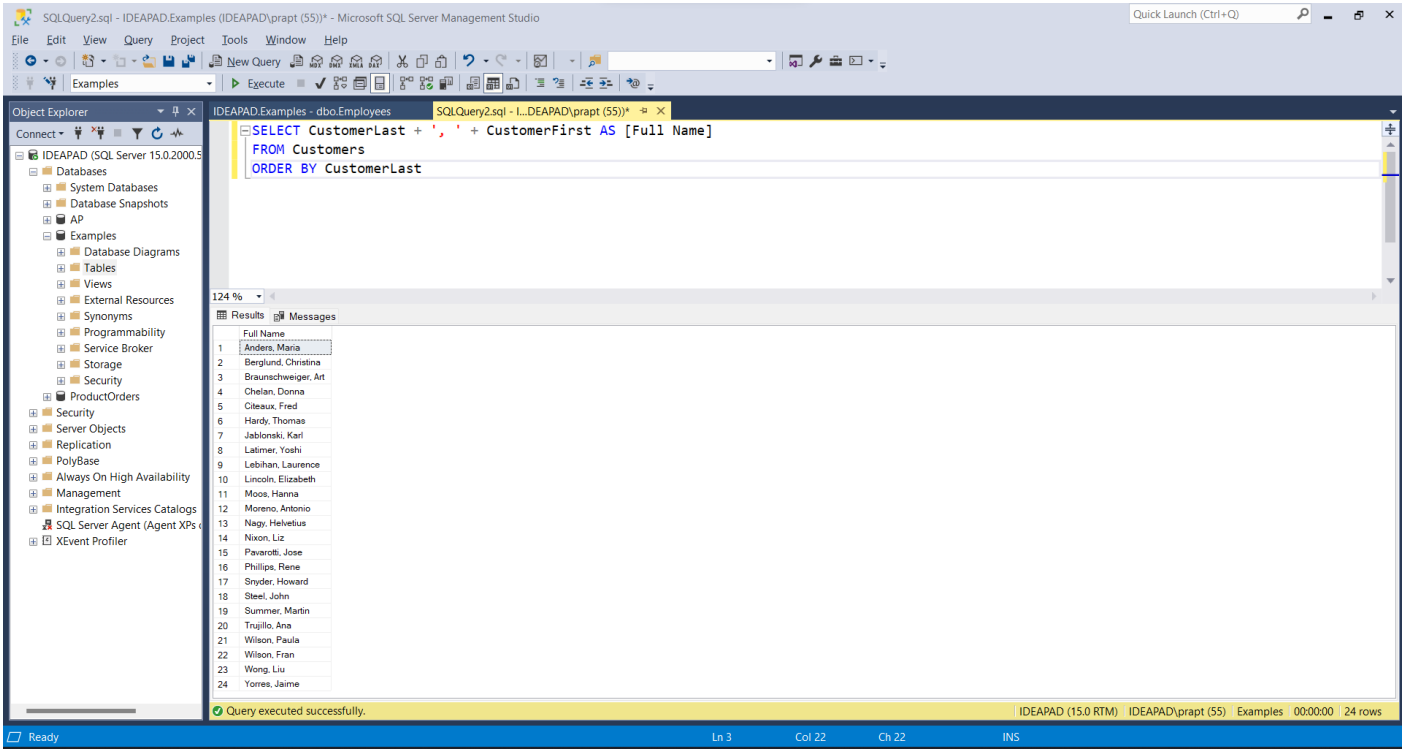
Remark: AS keyword is used for aliasing. WHERE clause is used to apply a condition to display a particular set of records. '+' operator is used for concatenation.

Q3) Write a SELECT statement that returns one column from the Customers table named "Full Name". Create this column from the CustomerFirst and CustomerLast columns. Format it as follows: CustomerLast, comma, space, CustomerFirst. Sort the result set by CustomerLast from "A-Z". ***Use Examples database.***

Ans: SELECT CustomerLast + ', ' + CustomerFirst AS [Full Name]

FROM Customers

ORDER BY CustomerLast



Comment: SELECT statement is used to display one column that is created by concatenation of two columns of the Customers table using the '+' operator into a single column using column alias using AS keyword and named "FullName". ORDER BY keyword is used to sort by CustomerLast in ascending order.

Remark: ORDER BY keyword is used to sort the data of the result set in any particular manner. ORDER BY can be done on column name, alias, column number and in different oder like ascending, descending.

Q4) 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 non-null value if there is no

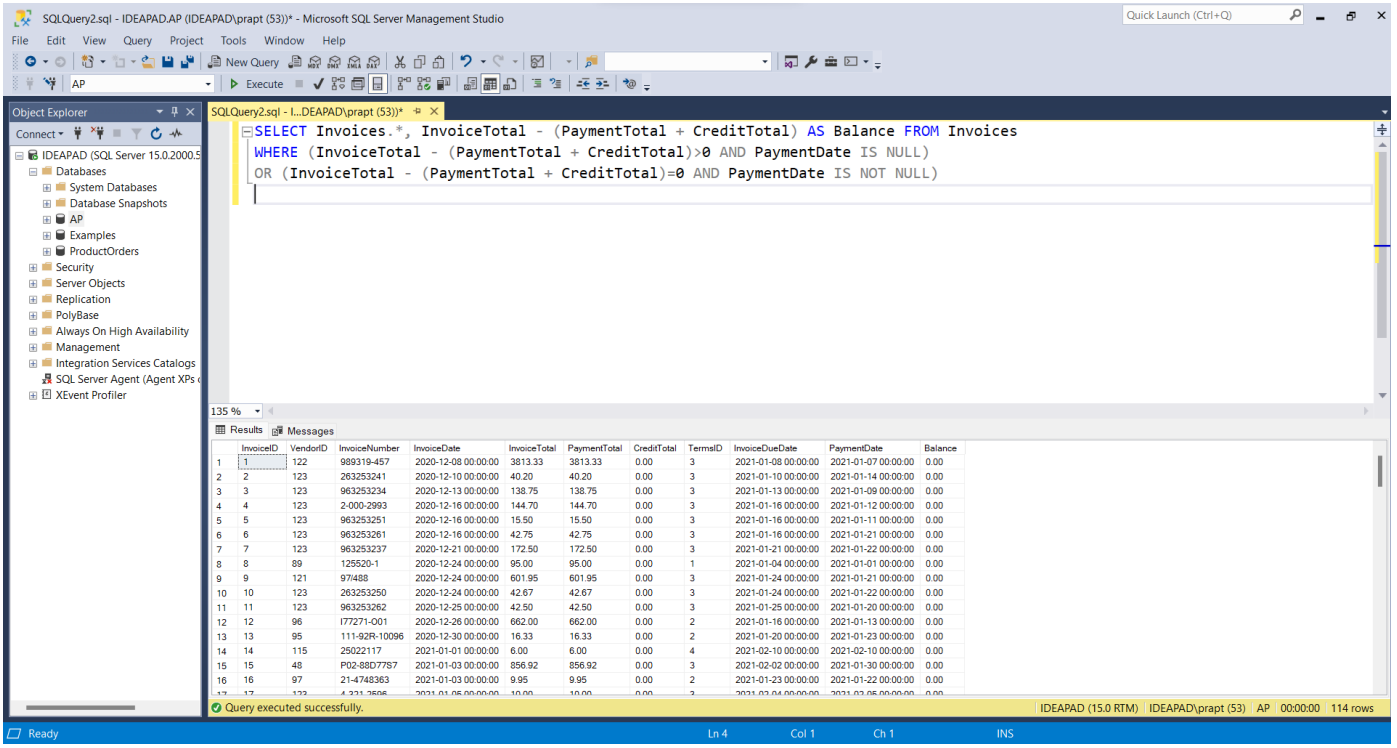
balance due, and a null value if there is balance due. Code a compound condition in the WHERE clause that tests for these conditions. (Balance: InvoiceTotal minus the sum of PaymentTotal and CreditTotal). ***Use AP database.***

ANS: SELECT Invoices.*, InvoiceTotal - (PaymentTotal + CreditTotal) AS Balance

FROM Invoices

WHERE (InvoiceTotal - (PaymentTotal + CreditTotal)>0 AND PaymentDate IS NULL)

OR (InvoiceTotal - (PaymentTotal + CreditTotal)=0 AND PaymentDate IS NOT NULL)



Comment: Here, the SELECT statement is used to display all columns from the Invoices table in addition to the balance column. From this table the records that are to be displayed using WHERE clause is those that have balance greater than zero which makes the payment date to be null (thus AND keyword is used) OR those that have balance as zero which makes the payment date to be not null (thus AND keyword is used). These two conditions are then compounded by OR keyword as these records is to be displayed.

Remark: AND is used when two conditions need to be met simultaneously. OR is used when either of the conditions must be met. IS NULL and IS NOT NULL are used to check for the records that are null or not null respectively.

Q5) Write a SELECT statement that returns five columns: CustLastName, CustCity, CustState, OrderDate and ShippedDate from the Customers table and Orders table. The result set should have one row for each customer, with the city, order date and shipped date for that customer's ID. Filter for Customers whose CustState is 'MA' and ShippedDate is null. Sort the result set by CustLastName from A to Z. ***Use ProductOrders database.***

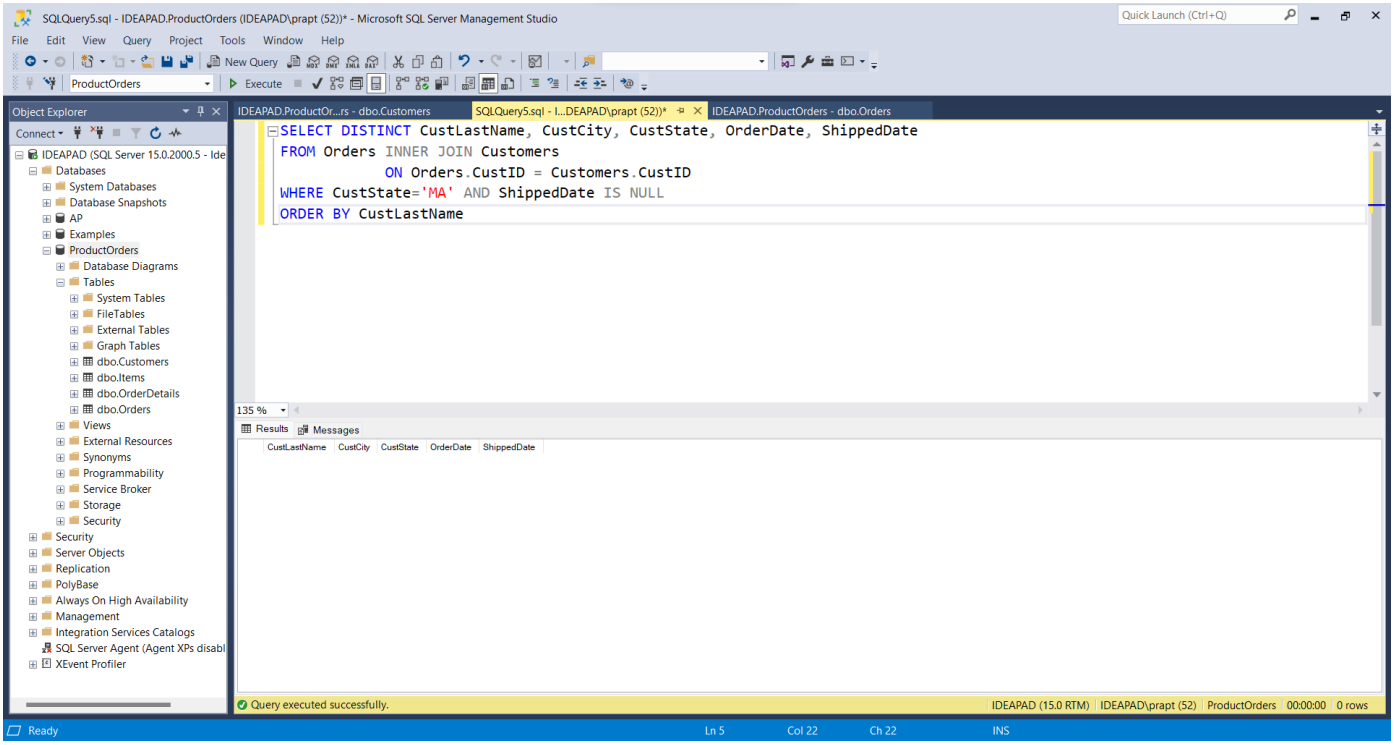
ANS: SELECT DISTINCT CustLastName, CustCity, CustState, OrderDate, ShippedDate

FROM Orders INNER JOIN Customers

ON Orders.CustID = Customers.CustID

WHERE CustState='MA' AND ShippedDate IS NULL

ORDER BY CustLastName



Comment: Here, the SELECT statement displays five columns, three of which belong to Customers table and two belong to Orders table. INNER JOIN is used to display those records from these tables having the common column name CustID. Inner join gives the rows as an intersection of tables. So further applying a WHERE clause to retrieve the records whose CustState is MA as well as ShippedDate is null, AND keyword is used which gives 0 rows as there exists no record with CustState as MA. The output is sorted in ascending order on CustLastName using ORDER BY.

Remark: DISTINCT keyword is used to display only unique values I.e., remove the duplicate values from displaying in the result set. INNER JOIN is used to join or combine two tables based on the common column in them (intersection). ON keyword is used to apply the conditions for joining two tables.

Q6) Write a SELECT statement that returns two columns: VendorName and FullName (A concatenation of VendorContactLName and VendorContactFName, with a space in between). The result set should have one row for each vendor whose contact has the same first name (i.e., VendorContactFName) as another vendor's contact. Sort the final result set by FullName column from Z to A. ***Use AP database.***

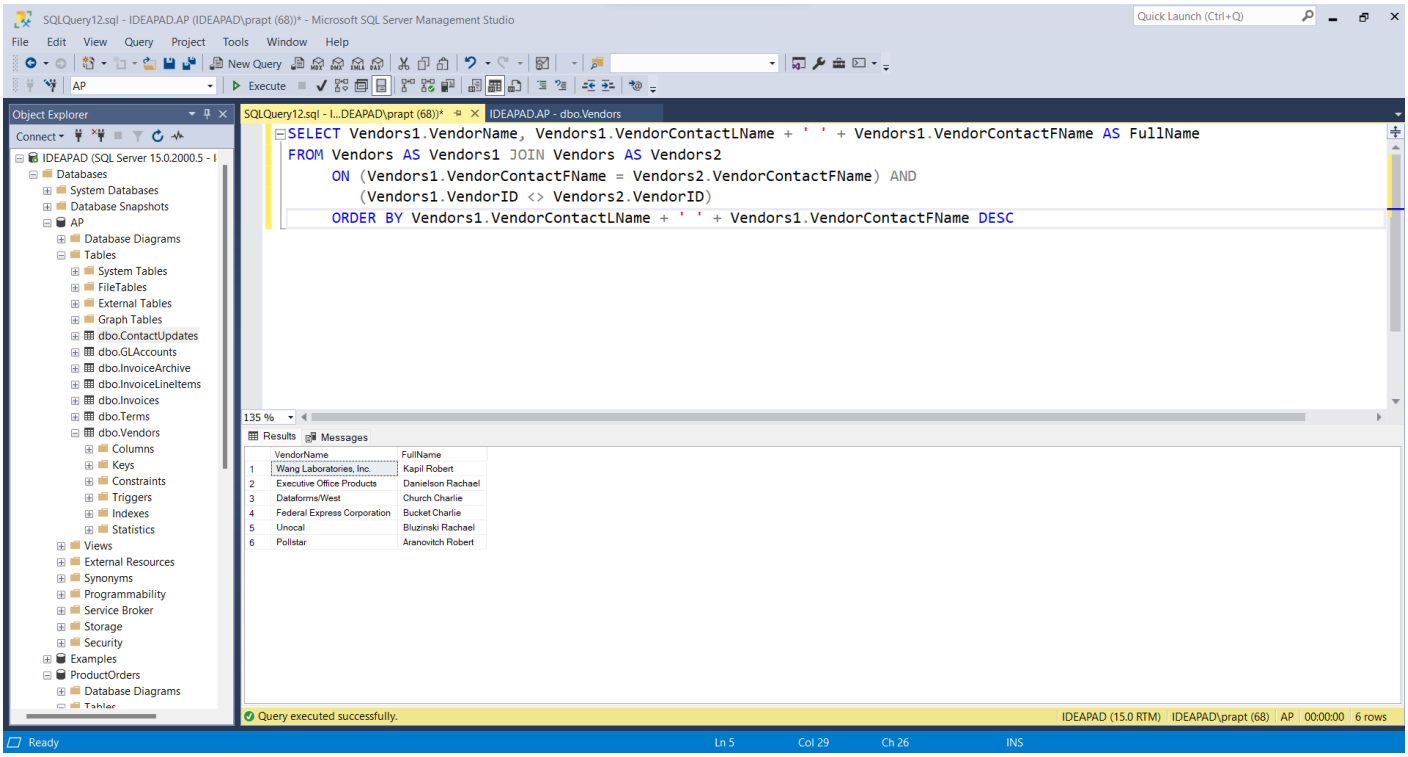
Ans: SELECT Vendors1.VendorName, Vendors1.VendorContactLName + ' ' + Vendors1.VendorContactFName AS FullName

From Vendors AS Vendors1 JOIN Vendors AS Vendors2

ON (Vendors1.VendorContactFName = Vendors2.VendorContactFName) AND

(Vendors1.VendorID<>Vendors2.VendorID)

ORDER BY Vendors1.VendorContactLName + ' ' + Vendors1.VendorContactFName DESC

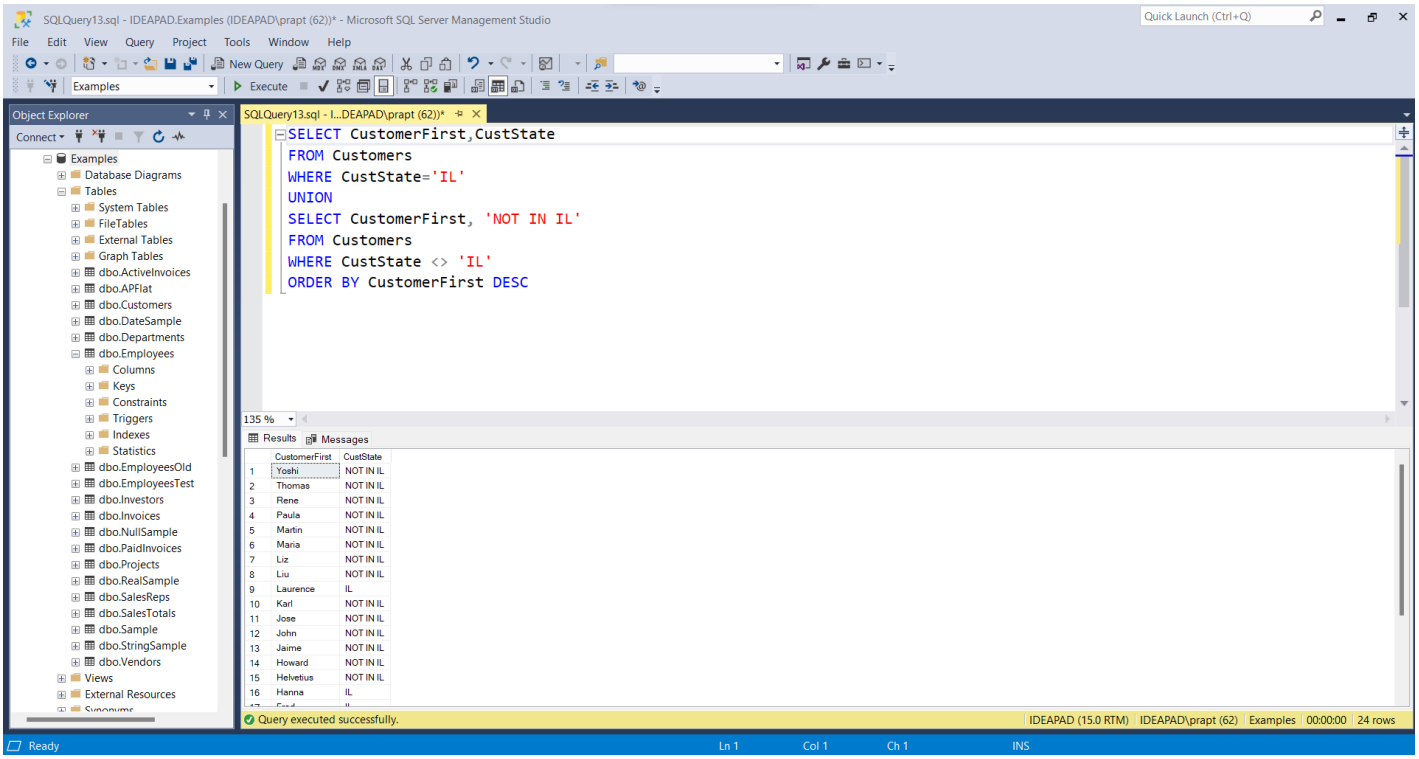


Comment: Here, SELECT statement is used to display VendorName and FullName columns from the Vendors table. Fullname column is the concatenation of VendorContactLName and VendorContactFName done using a '+' operator. JOIN(Self) is used to return the Vendors from VendorContactFName in common with other Vendors. To join the Vendors table with itself, table aliases for the same table are used which are Vendors1 and Vendors2 which are given using the AS keyword. ON keyword is used to specify the join conditions on the records that have VendorCOntactFName same, but the VendorID not same which is done using AND keyword. Lastly, the ORDER BY keyword is used to sort the records based on the FullName column in descending order.

Remark: JOIN keyword is used to combine columns, in this case two columns from the same table, thus called a SELF JOIN.

Q7) Use the UNION operator to generate a result set consisting of two columns from the Customers table: CustomerFirst and CustState. If the customer is in Illinois, the CustState value should be "IL"; otherwise, the CustState value should be "Not in IL". Sort the final result set by CustomerFirst from Z-A. **Use Examples database.**

Ans:
```
SELECT CustomerFirst, CustState
    FROM Customers
    WHERE CustState = 'IL'
UNION
    SELECT CustomerFirst, 'NOT IN IL'
    FROM Customers
    WHERE CustState <> 'IL'
    ORDER BY CustomerFirst DESC
```



Comment: Here, SELECT statements are used to display two columns CustomerFirst and CustState from the Customers table with two different WHERE clauses: one retrieving records that has CustState as 'IL' and other retrieving records that does not have CustState as 'IL'. UNION operator is used to combine the results of these two SELECT statements as both have same number of columns, with same data types and in same order. ORDER BY is used to sort the records by the column CustomerFirst in descending order.

Remark: UNION operator is used to combine two or more select statements in same order, having same columns, having same data type.

Q8) 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 in descending order. **Use AP database.** (HINT: Join GLAccounts table and InvoiceLineItems table.)

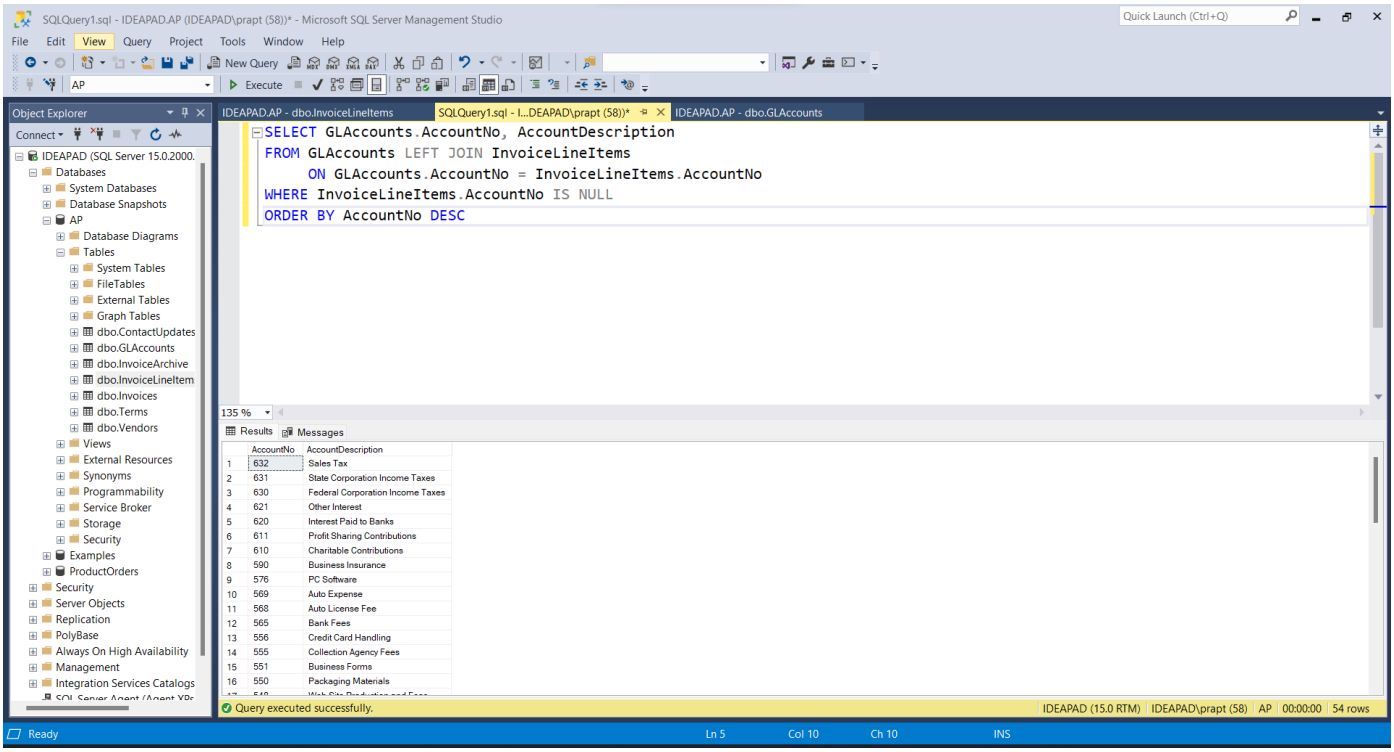
Ans: SELECT GLAccounts.AccountNo, AccountDescription

FROM GLAccounts LEFT JOIN InvoiceLineItems

ON GLAccounts.AccountNo = InvoiceListItems.AccountNo

WHERE InvoiceLineItems.AccountNo IS NULL

ORDER BY AccountNo DESC



Comment: SELECT statement is used to display columns from GLAccounts table. As the result set should have one row for each account number that has never been used which means that AccountNo data in InvoiceLineItems must be null for that record, LEFT JOIN is used to retrieve the records that are in GLAccounts or in both GLAccounts and InvoiceLineItems. WHERE clause is used for the condition that AccountNo in InvoiceLineItems is null. The records are sorted using ORDER BY keyword on the AccountNo column in descending order.

Remark: LEFT JOIN is used to join the whole first table with the part of the second table that intersects with the first table.