

Lab 2: Select, Table Joins, Unions

1. Write a SELECT statement that returns four columns from the Invoices table:
InvoiceNumber, InvoiceDate, InvoiceDueDate, and InvoiceTotal. Use AP database.
As you can see in the screenshot below, here're the four columns from the Invoices table:

The screenshot shows a SQL query editor window titled "Lab2 Query1.sql...PTHKBA0\lyq (51)". The query text is:

```
SELECT InvoiceNumber, InvoiceDate, InvoiceDueDate, InvoiceTotal
FROM Invoices;
```

Below the query editor, the "Results" tab is active, displaying a table with 14 rows of data. The columns are InvoiceNumber, InvoiceDate, InvoiceDueDate, and InvoiceTotal. The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (51) AP 00:00:00 114 rows".

	InvoiceNumber	InvoiceDate	InvoiceDueDate	InvoiceTotal
1	989319-457	2015-12-08 00:00:00	2016-01-08 00:00:00	3813.33
2	263253241	2015-12-10 00:00:00	2016-01-10 00:00:00	40.20
3	963253234	2015-12-13 00:00:00	2016-01-13 00:00:00	138.75
4	2-000-2993	2015-12-16 00:00:00	2016-01-16 00:00:00	144.70
5	963253251	2015-12-16 00:00:00	2016-01-16 00:00:00	15.50
6	963253261	2015-12-16 00:00:00	2016-01-16 00:00:00	42.75
7	963253237	2015-12-21 00:00:00	2016-01-21 00:00:00	172.50
8	125520-1	2015-12-24 00:00:00	2016-01-04 00:00:00	95.00
9	97/488	2015-12-24 00:00:00	2016-01-24 00:00:00	601.95
10	263253250	2015-12-24 00:00:00	2016-01-24 00:00:00	42.67
11	963253262	2015-12-25 00:00:00	2016-01-25 00:00:00	42.50
12	177271-001	2015-12-26 00:00:00	2016-01-16 00:00:00	662.00
13	111-92R-10096	2015-12-30 00:00:00	2016-01-20 00:00:00	16.33
14	25022117	2016-01-01 00:00:00	2016-02-10 00:00:00	6.00

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 \$300. Use AP database.

The screenshot is as below:

The screenshot shows a SQL Server Enterprise Manager window with a query executed successfully. The query is as follows:

```
SELECT InvoiceNumber AS Number, InvoiceTotal AS Total, PaymentTotal+CreditTotal AS Credits
FROM Invoices
WHERE InvoiceTotal <= 300;
```

The results are displayed in a table with 4 columns: Number, Total, and Credits. The table contains 14 rows of data, representing invoices with a total value less than or equal to 300.

	Number	Total	Credits
1	263253241	40.20	40.20
2	963253234	138.75	138.75
3	2-000-2993	144.70	144.70
4	963253251	15.50	15.50
5	963253261	42.75	42.75
6	963253237	172.50	172.50
7	125520-1	95.00	95.00
8	263253250	42.67	42.67
9	963253262	42.50	42.50
10	111-92R-10096	16.33	16.33
11	25022117	6.00	6.00
12	21-4748363	9.95	9.95
13	4-321-2596	10.00	10.00
14	963253242	104.00	104.00

At the bottom of the window, a status bar indicates: Query executed successfully. DESKTOP-PTKBA0 (15.0 RTM) DESKTOP-PTKBA0\lyq (57) AP 00:00:00 71 rows

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 “Z-A”. Use AP database.

The screenshot is as below:

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL statement:

```
SELECT VendorContactLName + ', ' + VendorContactFName  
AS 'Full Name'  
FROM Vendors  
ORDER BY VendorContactLName DESC;
```

The bottom pane shows the results of the query, sorted by last name in descending order (Z-A). The results are displayed in a table with two columns: 'Full Name' and an index column.

	Full Name
1	Yobani, Trey
2	Wood, Liam
3	Wesley, Alisha
4	Warren, Quentin
5	Walker, Jovon
6	Sydney, Deangelo
7	Story, Kirsten
8	Stevens, Wendy
9	Spivak, Harold
10	Snyder, Karen
11	Smitzen, Jeffrey
12	Smith, Sam
13	Smith, Kylie
14	Royce, Hannah

The status bar at the bottom indicates: Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (51) AP 00:00:00 122 rows

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 if 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 database.

The screenshot is as below:

The screenshot shows a SQL Server Enterprise Manager interface with three query windows open. The active window, 'Lab2 Query4.sql...PTHKBA0\lyq (57)', contains the following SQL query:

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

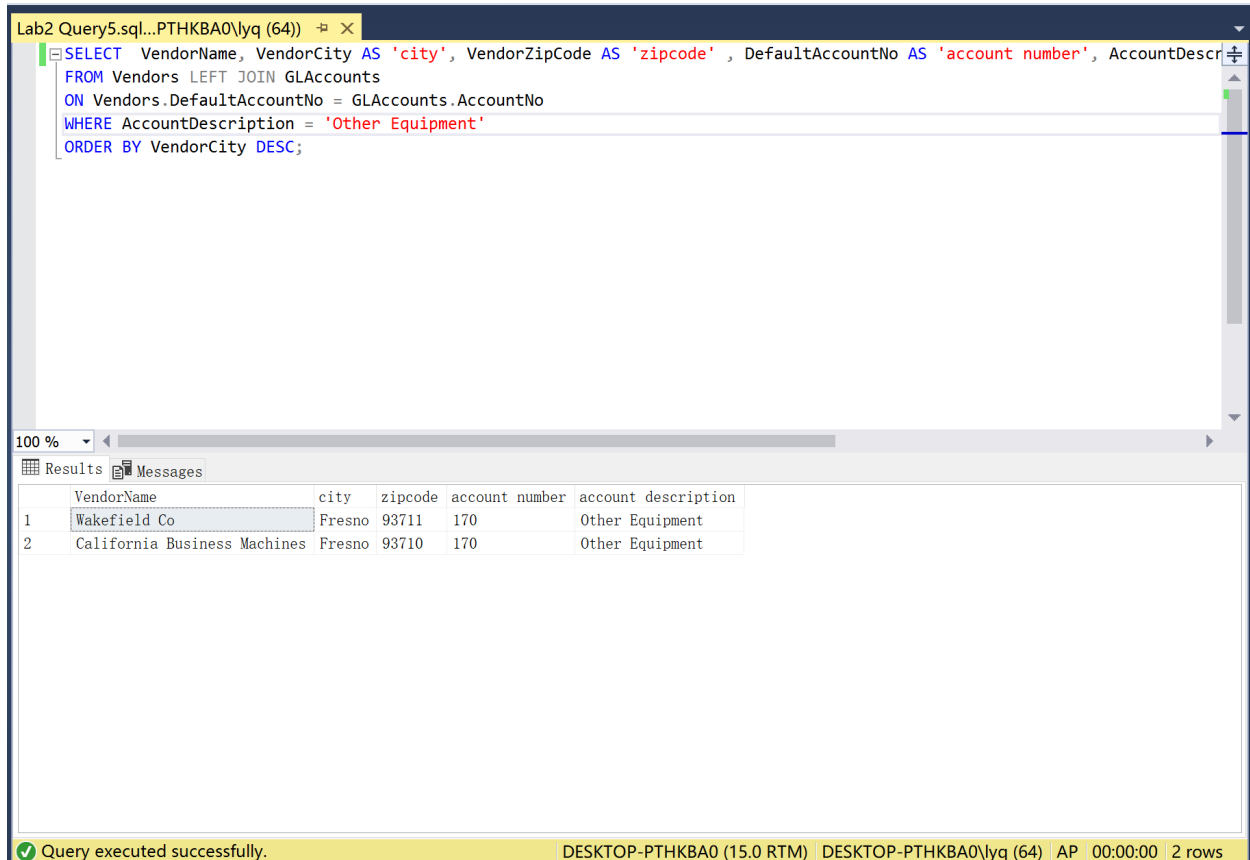
Below the query window, the 'Results' pane displays 14 rows of data from the 'Invoices' table. The columns are: InvoiceID, VendorID, InvoiceNumber, InvoiceDate, InvoiceTotal, PaymentTotal, CreditTotal, TermsID, InvoiceDueDate, and PaymentDate. The data shows various invoices with their respective totals and payment dates.

	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate	PaymentDate
1	1	122	989319-457	2015-12-08 00:00:00	3813.33	3813.33	0.00	3	2016-01-08 00:00:00	2016-01-07 00:00:00
2	2	123	263253241	2015-12-10 00:00:00	40.20	40.20	0.00	3	2016-01-10 00:00:00	2016-01-14 00:00:00
3	3	123	963253234	2015-12-13 00:00:00	138.75	138.75	0.00	3	2016-01-13 00:00:00	2016-01-09 00:00:00
4	4	123	2-000-2993	2015-12-16 00:00:00	144.70	144.70	0.00	3	2016-01-16 00:00:00	2016-01-12 00:00:00
5	5	123	963253251	2015-12-16 00:00:00	15.50	15.50	0.00	3	2016-01-16 00:00:00	2016-01-11 00:00:00
6	6	123	963253261	2015-12-16 00:00:00	42.75	42.75	0.00	3	2016-01-16 00:00:00	2016-01-21 00:00:00
7	7	123	963253237	2015-12-21 00:00:00	172.50	172.50	0.00	3	2016-01-21 00:00:00	2016-01-22 00:00:00
8	8	89	125520-1	2015-12-24 00:00:00	95.00	95.00	0.00	1	2016-01-04 00:00:00	2016-01-01 00:00:00
9	9	121	97/488	2015-12-24 00:00:00	601.95	601.95	0.00	3	2016-01-24 00:00:00	2016-01-21 00:00:00
10	10	123	263253250	2015-12-24 00:00:00	42.67	42.67	0.00	3	2016-01-24 00:00:00	2016-01-22 00:00:00
11	11	123	963253262	2015-12-25 00:00:00	42.50	42.50	0.00	3	2016-01-25 00:00:00	2016-01-20 00:00:00
12	12	96	177271-001	2015-12-26 00:00:00	662.00	662.00	0.00	2	2016-01-16 00:00:00	2016-01-13 00:00:00
13	13	95	111-92R-10096	2015-12-30 00:00:00	16.33	16.33	0.00	2	2016-01-20 00:00:00	2016-01-23 00:00:00

At the bottom of the interface, a status bar indicates: 'Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (57) AP 00:00:00 114 rows'.

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

The screenshot is as below:



The screenshot displays a SQL query window titled 'Lab2 Query5.sql...PTHKBA0\lyq (64)'. The query is as follows:

```
SELECT VendorName, VendorCity AS 'city', VendorZipCode AS 'zipcode', DefaultAccountNo AS 'account number', AccountDescription
FROM Vendors LEFT JOIN GLAccounts
ON Vendors.DefaultAccountNo = GLAccounts.AccountNo
WHERE AccountDescription = 'Other Equipment'
ORDER BY VendorCity DESC;
```

Below the query window, the 'Results' tab is active, showing a table with 2 rows and 6 columns: VendorName, city, zipcode, account number, and account description. The data is as follows:

	VendorName	city	zipcode	account number	account description
1	Wakefield Co	Fresno	93711	170	Other Equipment
2	California Business Machines	Fresno	93710	170	Other Equipment

At the bottom of the window, a status bar indicates 'Query executed successfully.' and 'DESKTOP-PTHKBA0 (15.0 RTM) | DESKTOP-PTHKBA0\lyq (64) | AP | 00:00:00 | 2 rows'.

6. Write a SELECT statement that returns two columns from ContactUpdates table: FirstName and Full Name (A concatenation of FirstName and LastName, 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 FirstName from A to Z. Are there such vendors? Use AP database.

The screenshot is as below and as we could see there are no such vendors.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL statement:

```
SELECT DISTINCT ContactUpdates1.FirstName, ContactUpdates1.FirstName + ',' + ContactUpdates1.LastName AS 'Full Name'
FROM ContactUpdates AS ContactUpdates1 JOIN ContactUpdates AS ContactUpdates2
ON (ContactUpdates1.LastName = ContactUpdates2.LastName)
AND (ContactUpdates1.VendorID <> ContactUpdates2.VendorID)
ORDER BY ContactUpdates1.FirstName;
```

The bottom pane shows the 'Results' tab, which is currently empty, indicating that no rows were returned by the query. The status bar at the bottom of the window confirms this, stating: 'Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (52) AP 00:00:00 0 rows'.

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 California, the VendorState value should be “CA”; otherwise, the vendorState value should be “Not in CA”. Sort the final result set by VendorName from A to Z. Use AP database.

The screenshot is as below:

```
SELECT VendorName, VendorState
FROM Vendors
WHERE VendorState IN('CA')
UNION
SELECT VendorName, 'Not in CA' AS VendorState
FROM Vendors
WHERE VendorState <> ('CA')
ORDER BY VendorName;
```

	VendorName	VendorState
1	Abbey Office Furnishings	CA
2	American Booksellers Assoc	Not in CA
3	American Express	CA
4	ASC Signs	CA
5	Ascom Hasler Mailing Systems	Not in CA
6	AT&T	Not in CA
7	Aztek Label	CA
8	Baker & Taylor Books	Not in CA
9	Bertelsmann Industry Svcs. Inc	CA
10	BFI Industries	CA
11	Bill Jones	CA
12	Bill Marvin Electric Inc	CA
13	Blanchard & Johnson Associates	CA
14	Blue Cross	CA

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (72) AP 00:00:00 122 rows

8. Write a SELECT statement that returns a column from the OrderDetails table: ItemID. The result set should have one row for each item whose order was placed but not yet shipped (i.e. ShippedDate in Orders table has null value). Use ProductOrders database.

The screenshot is as below:

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL statement:

```
SELECT ItemID
FROM OrderDetails JOIN Orders
ON OrderDetails.OrderID = Orders.OrderID
WHERE ShippedDate IS NULL;
```

The bottom pane shows the 'Results' tab with a table containing 7 rows of data. The first row is highlighted.

	ItemID
1	6
2	7
3	1
4	2
5	3
6	5
7	9

The status bar at the bottom indicates: 'Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (65) ProductOrders 00:00:00 7 rows'.