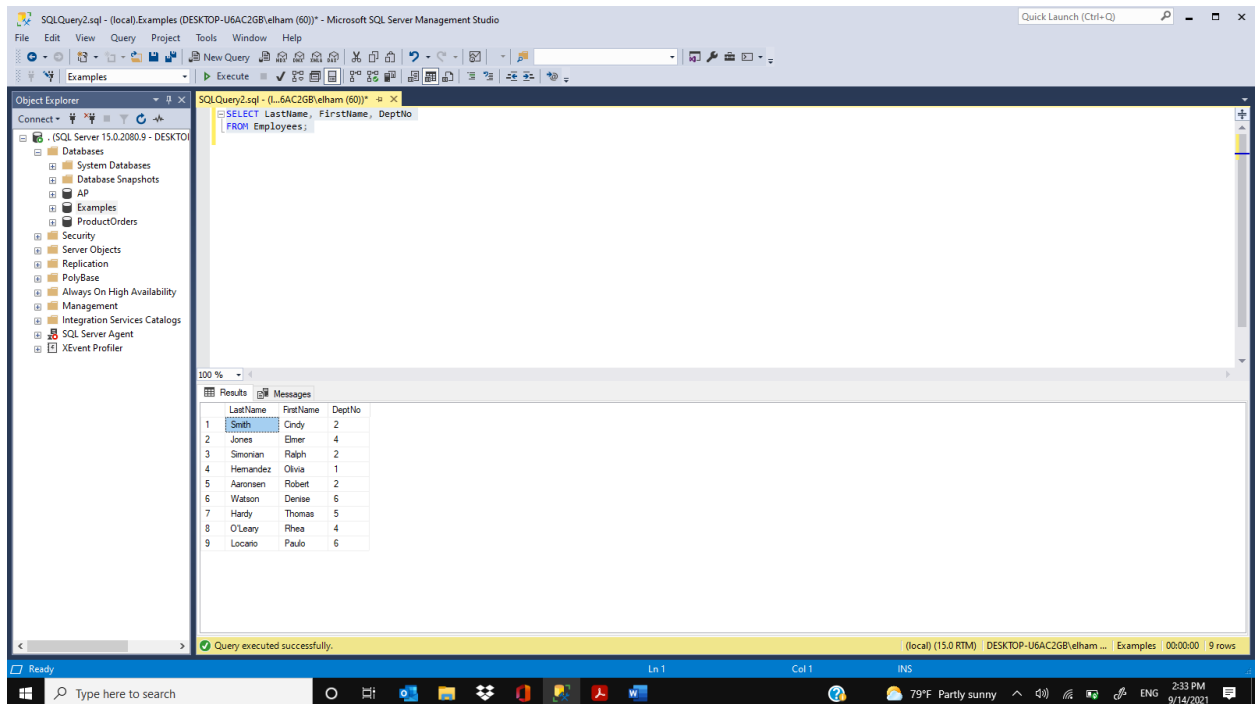


Q1: Write a SELECT statement that returns three columns from the Employees table: LastName, FirstName and DeptNo. Use Examples database.

A1: Using Example database and running the following script we get the following table containing first name and last name of employees and the department number they are working in.

```
SELECT LastName, FirstName, DeptNo
FROM Employees;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left displays the 'Examples' database. The 'Query Editor' in the center contains the SQL query: `SELECT LastName, FirstName, DeptNo FROM Employees;`. The 'Results' pane at the bottom shows the output of the query, which is a table with three columns: LastName, FirstName, and DeptNo. The table contains 9 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and '9 rows'.

	LastName	FirstName	DeptNo
1	Smith	Cindy	2
2	Jones	Elmer	4
3	Simonian	Ralph	2
4	Hernandez	Olivia	1
5	Aaronsen	Robert	2
6	Watson	Denise	6
7	Hardy	Thomas	5
8	O'Leary	Rhea	4
9	Locato	Paulo	6

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 followed by FirstName)

DeptNumber Column alias for the DeptNo column

And filter for Customers with DeptNo value as 2. Use Examples database.

A2: here we can see names of people who are working at department 2.

```
SELECT LastName+', '+FirstName AS Name
,DeptNo AS DeptNumber
FROM Employees
WHERE DeptNo=2;
```

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

```
SELECT LastName+', '+FirstName AS Name
,DeptNo AS DeptNumber
FROM Employees
WHERE DeptNo=2;
```

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

	Name	DeptNumber
1	Smith, Cindy	2
2	Simonian, Ralph	2
3	Aaronsen, Robert	2

The status bar at the bottom indicates that the query was executed successfully and that there are 3 rows of data.

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: CustomerFirst, space, CustomerLast. Sort the result set by CustomerFirst from "A-Z". Use Examples database.

A3: Here we can see the names of customers in A-Z order.

```
SELECT CustomerFirst+', '+CustomerLast AS [Full Name]
FROM Customers
ORDER BY CustomerFirst;
```

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

```
SELECT CustomerFirst+', '+CustomerLast AS [Full Name]
FROM Customers
ORDER BY CustomerFirst;
```

The Results pane shows the output of the query, which is a list of customer names sorted alphabetically by their first name. The results are as follows:

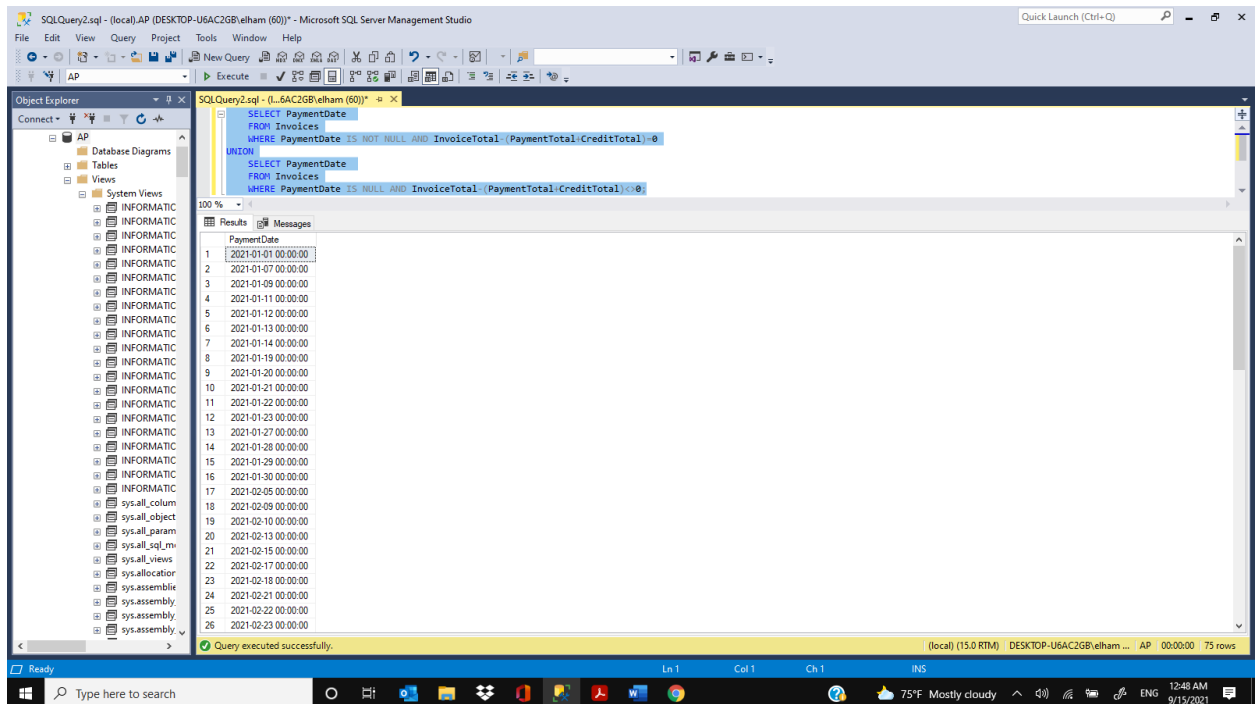
Full Name
Aria, Trujillo
Antonio, Moreno
Art, Braunschweiler
Christina, Berglund
Donna, Chen
Elizabeth, Lincoln
Fran, Wilson
Fred, Citeaux
Hanna, Moos
Helvetius, Nagy
Howard, Snyder
Jaime, Yones
John, Steel
Jose, Pavarotti
Karl, Jablonski
Laurence, Leblanc
Liu, Wong
Liz, Neison
Maria, Anders
Martin, Summer
Paula, Wilson
Rene, Phillips
Thomas, Hardy
Yoshi, Latimer

The status bar at the bottom indicates that the query was executed successfully and returned 24 rows.

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.

A4: Here we have all the dates in which either there is no payment due, or the payment date is not entered.

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



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 'CA' and ShippedDate is null. Sort the result set by CustLastName from Z to A. Use ProductOrders database.

A5: By entering the following script we will get the name, city and order date of the customer who lives in California and her package has not shipped yet (no date for shipment).

```
SELECT DISTINCT CustLastName, CustCity, CustState, OrderDate, ShippedDate
FROM Customers JOIN Orders
ON Customers.CustID=Orders.CustID
WHERE CustState='CA' AND ShippedDate IS NULL
ORDER BY CustLastName;
```

The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows the database structure for 'ProductOrders'. The 'Query Editor' in the center contains the following SQL script:

```
SELECT DISTINCT CustLastName, CustCity, CustState, OrderDate, ShippedDate
FROM Customers JOIN Orders
ON Customers.CustID=Orders.CustID
WHERE CustState='CA' AND ShippedDate IS NULL
ORDER BY CustLastName;
```

The 'Results' pane at the bottom shows a single row of data:

	CustLastName	CustCity	CustState	OrderDate	ShippedDate
1	Lacy	Los Angeles	CA	2021-04-02 00:00:00.000	NULL

The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

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.

A6: Here we get the vendors and person of contact.

```
SELECT DISTINCT VendorName, VendorContactLName+' '+VendorContactFName AS FullName
FROM Vendors
ORDER BY FullName DESC;
```

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

```
SELECT DISTINCT VendorName, VendorContactLName+' '+VendorContactFName AS FullName
FROM Vendors
ORDER BY FullName DESC;
```

The Results pane displays the output of the query, showing two columns: VendorName and FullName. The results are sorted by FullName in descending order. The first row is highlighted.

VendorName	FullName
Ingram	Yobani Trey
The Windows Deck	Wood Liam
AT&T	Wiesley Alpha
Valprint	Warner Quentin
Publishers Marketing Assoc	Walker Jovon
Grayfit	Sydney Deangelo
American Express	Story Kirsten
Kent H Landsberg Co	Stevens Wendy
Suburban Propane	Spivak Harold
Ford Motor Credit Company	Snyder Karen
Coffee Break Service	Smitzen Jeffrey
Roadway Package System, Inc	Smith Sam
Blue Shield of California	Smith Kylie
Evans Executive Inc	Royce Hannah
Wakefield Co	Rothman Nathanael
Zylka Design	Ronaldsen Jaime
California Business Machines	Rohansen Anders
Reiter's Scientific & Pro Books	Rodolfo Carlen
Malloy Lithographing Inc	Regging Abe
Vision Envelope & Printing	Raven Jamari
Crown Printing	Randrup Leanni
Omg Information Services	Randall Yash
Excedata Inc	Quintin Marvin

The status bar at the bottom indicates that the query was executed successfully, returning 122 rows.

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.

A7: Here we can see which customer is in Illinois and who is not

```
SELECT CustomerFirst, CustState
FROM Customers
WHERE CustState='IL'
```

UNION

```
SELECT CustomerFirst, 'NOT IN IL' AS CustState
FROM Customers
WHERE CustState<>'IL'
ORDER BY CustomerFirst DESC;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the 'Examples' database structure. The Query Editor in the center contains the following SQL query:

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

The Results pane at the bottom displays the output of the query, showing a list of customers and their states, sorted by CustomerFirst in descending order.

CustomerFirst	CustState
Yoshi	NOT IN IL
Thomas	NOT IN IL
Pierre	NOT IN IL
Paula	NOT IN IL
Martin	NOT IN IL
Maria	NOT IN IL
Liz	NOT IN IL
Liu	NOT IN IL
Laurence	IL
Karl	NOT IN IL
Jose	NOT IN IL
John	NOT IN IL
Jane	NOT IN IL
Howard	NOT IN IL
Helvetius	NOT IN IL
Hanna	IL
Fred	IL
Fran	NOT IN IL
Elizabeth	NOT IN IL
Donna	NOT IN IL
Christina	NOT IN IL
Art	NOT IN IL
Antonio	NOT IN IL

The status bar at the bottom indicates that the query was executed successfully, returning 24 rows.

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.)

A8: AccountNo cannot be a null value (in table properties it said not null)!

```
SELECT *
FROM GLAccounts JOIN InvoiceLineItems
ON GLAccounts.AccountNo=InvoiceLineItems.AccountNo
ORDER BY 1 DESC;
```

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

```
SELECT *
FROM GLAccounts JOIN InvoiceLineItems
ON GLAccounts.AccountNo=InvoiceLineItems.AccountNo
ORDER BY 1 DESC;
```

The query has been executed successfully, and the results are displayed in the Results pane. The results table has 7 columns: AccountNo, AccountDescription, InvoiceID, InvoiceSequence, AccountNo, InvoiceLineItemAmount, and InvoiceLineItemDescription. The data is sorted by AccountNo in descending order.

AccountNo	AccountDescription	InvoiceID	InvoiceSequence	AccountNo	InvoiceLineItemAmount	InvoiceLineItemDescription
591	Accounting	33	1	591	220.00	Form 571-L
589	Outside Services	21	1	589	4801.26	Office lease
589	Outside Services	31	1	589	7125.34	Web site design
589	Outside Services	57	1	589	1367.50	401K Contributions
582	Travel and Accommodations	105	1	582	503.20	Bronco lease
580	Meals	12	1	580	50.00	DiCicco's
574	Business Licenses and Taxes	15	1	574	856.92	Property Taxes
572	Books, Dues, and Subscriptions	16	1	572	9.95	Monthly access fee
572	Books, Dues, and Subscriptions	23	1	572	9.95	Monthly access fee
572	Books, Dues, and Subscriptions	56	1	572	2433.00	Card deck
572	Books, Dues, and Subscriptions	47	1	572	1575.00	Catalog ad
572	Books, Dues, and Subscriptions	95	1	572	600.00	Books for research
572	Books, Dues, and Subscriptions	98	1	572	579.42	Catalog ad
570	Office Supplies	74	1	570	41.80	Coffee
570	Office Supplies	12	2	570	75.60	Kirk's
570	Office Supplies	12	3	570	58.40	Office Max
553	Freight	17	1	553	10.00	Address correction
553	Freight	18	1	553	104.00	Freight
553	Freight	1	1	553	3813.33	Freight
553	Freight	2	1	553	40.20	Freight
553	Freight	3	1	553	138.75	Freight
553	Freight	4	1	553	144.70	Int'l shipment
553	Freight	5	1	553	15.50	Freight

Thank you so much for your time.  
Sincerely,  
Seyed