CSE 581 : INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

LAB : 8

DATE : 11/2/2022

Q1) Write a view named VendorsInvoices that returns four columns: VendorName, InvoiceNumber, InvoiceDate and InvoiceTotal. Then, write a SELECT statement that returns all the columns in the view, sorted by InvoiceTotal in descending order, where the first letter of the vendor's name is Q, or R.

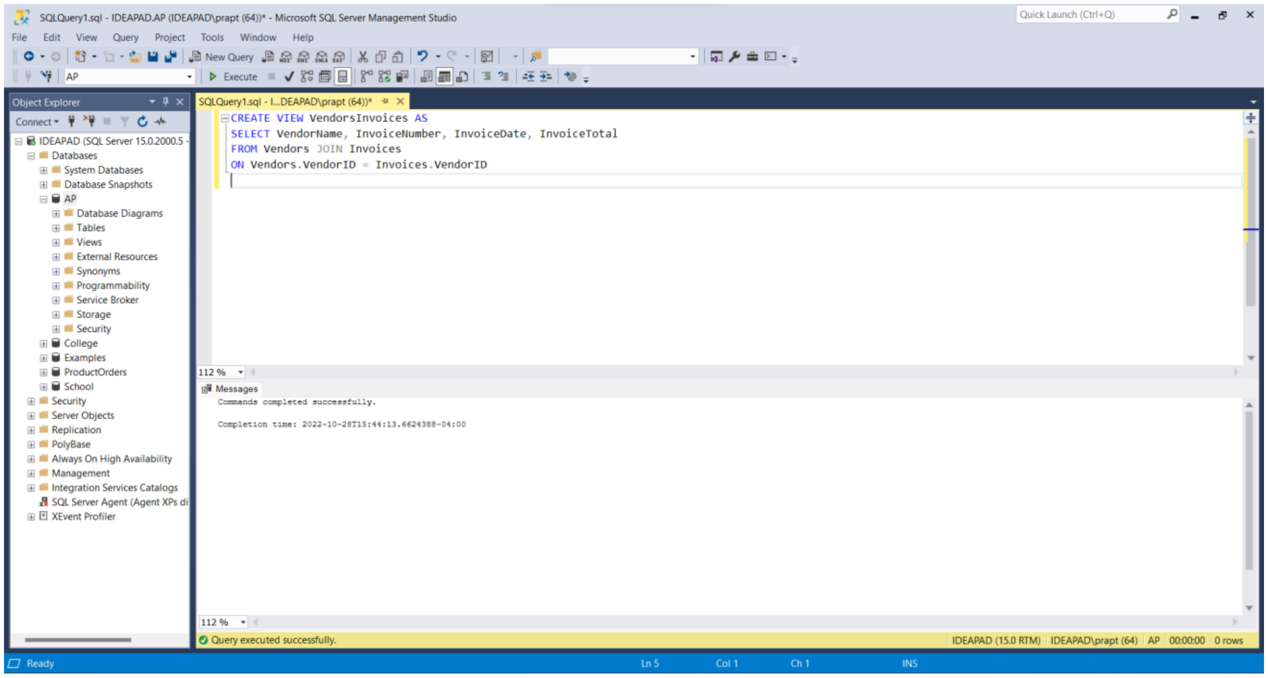
ANS:

CREATE VIEW VendorsInvoices AS

SELECT VendorName, InvoiceNumber, InvoiceDate, InvoiceTotal

FROM Vendors JOIN Invoices

ON Vendors.VendorID = Invoices.VendorID



```
CREATE VIEW VendorsInvoices AS
SELECT VendorName, InvoiceNumber, InvoiceDate, InvoiceTotal
FROM Vendors JOIN Invoices
ON Vendors.VendorID = Invoices.VendorID
```

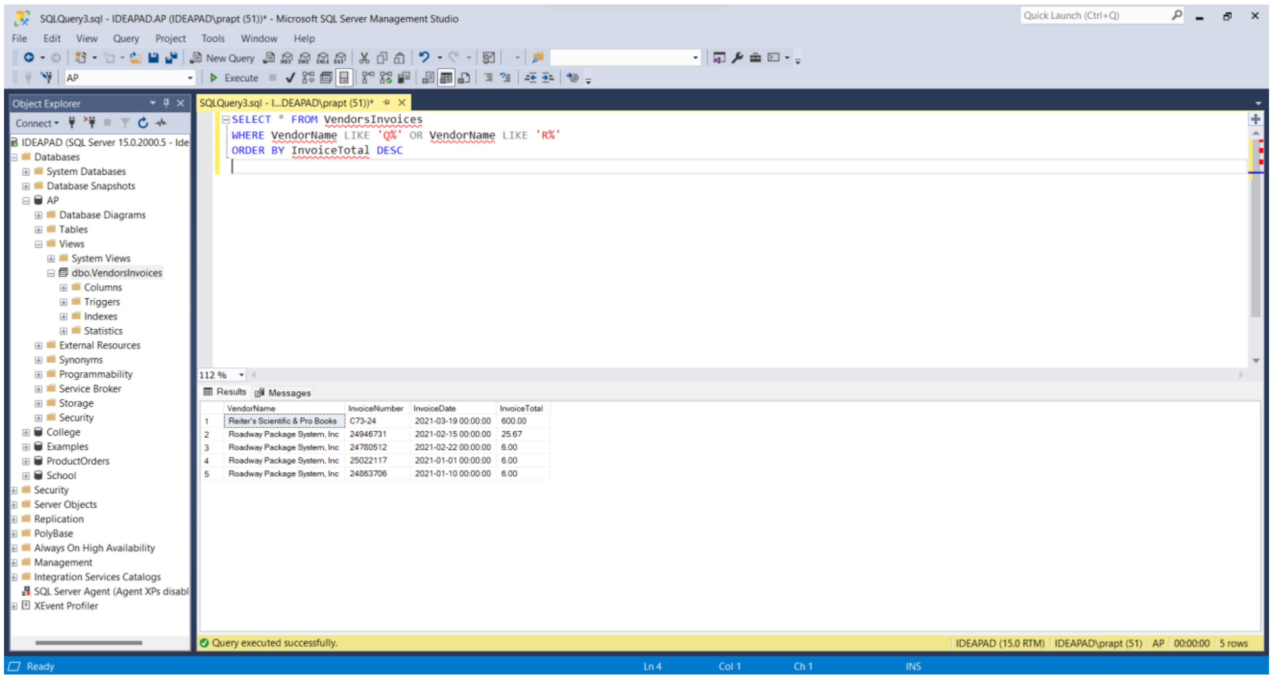
```
Commands completed successfully.

Completion time: 2022-10-28T15:44:13.6624388-04:00
```

Query executed successfully.

SELECT * FROM VendorsInvoices

WHERE VendorName LIKE 'Q%' OR VendorName LIKE 'R%'

ORDER BY InvoiceTotal DESC



```
SELECT * FROM VendorsInvoices
WHERE VendorName LIKE 'Q%' OR VendorName LIKE 'R%'
ORDER BY InvoiceTotal DESC
```

	VendorName	InvoiceNumber	InvoiceDate	InvoiceTotal
1	Reiter's Scientific & Pro Books	C73-24	2021-03-19 00:00:00	600.00
2	Roadway Package System, Inc	24946731	2021-02-15 00:00:00	25.67
3	Roadway Package System, Inc	24780512	2021-02-22 00:00:00	6.00
4	Roadway Package System, Inc	25022117	2021-01-01 00:00:00	6.00
5	Roadway Package System, Inc	24863706	2021-01-10 00:00:00	6.00

Query executed successfully. IDEAPAD (15.0 RTM) IDEAPAD\prapt (51) AP 00:00:00 5 rows

Comment: Here, CREATE VIEW is used to create a view in the database and the name VendorsInvoices is given. This view is a table that contains VendorName column from the Vendors table and the InvoiceNumber, invoiceDate, InvoiceTotal columns from the Invoices table which are selected using the SELECT statement. This is done using JOIN on Vendors and Invoices table on VendorID using ON keyword. Furthermore, SELECT * statement is used to retrieve all the columns from the view. To get the records of only those Vendors whose name starts from Q or R, WHERE clause is used. LIKE command is used in the WHERE clause to find the VendorName which starts from Q irrespective of the rest of the alphabets in the name which is represented as 'Q%' using the wildcard character % or the VendorName which starts from R irrespective of the rest of the alphabets in the name which is represented as 'R%'. Finally, the result set is sorted in descending order on InvoiceTotal.

Q2) Create a view named Top10PaidInvoices that returns three columns for each vendor: VendorName, FirstInvoiceDate (the least recent invoice date), and SumOfInvoices (the sum of the InvoiceTotal column). Return only the 10 vendors with the smallest SumOfInvoices and include only paid invoices (i.e. InvoiceTotal - CredeitTotal - PaymentTotal = 0). Then write a SELECT statement to show results of the view.

ANS:

CREATE VIEW Top10PaidInvoices AS

SELECT TOP 10

VendorName, MIN(InvoiceDate) AS FirstInvoiceDate, SUM(InvoiceTotal) AS SumOfInvoices

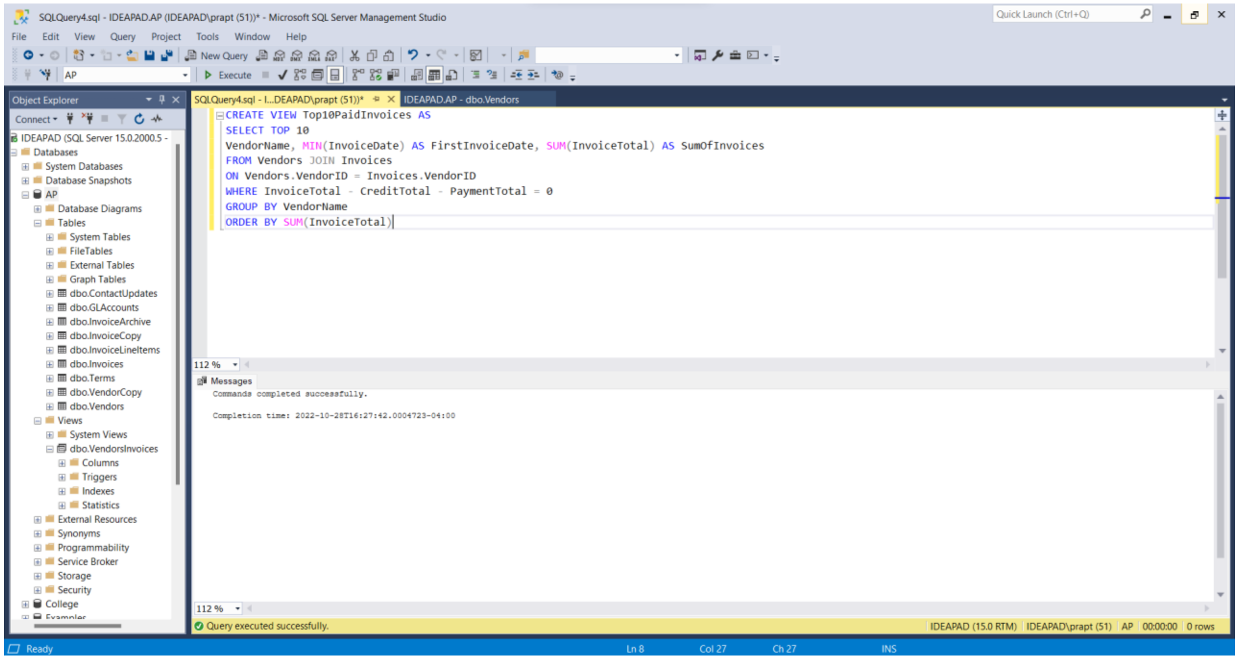
FROM Vendors JOIN Invoices

ON Vendors.VendorID = Invoices.VendorID

WHERE InvoiceTotal - CreditTotal - PaymentTotal = 0

GROUP BY VendorName

ORDER BY SUM(InvoiceTotal)

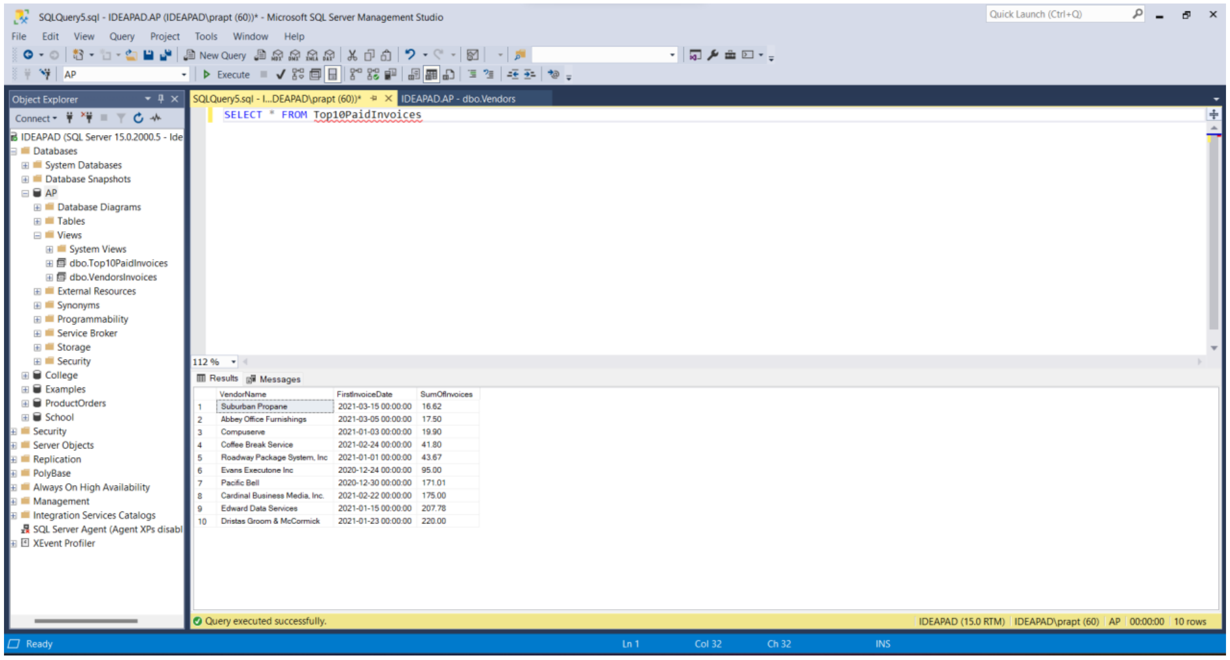


```
CREATE VIEW Top10PaidInvoices AS
SELECT TOP 10
VendorName, MIN(InvoiceDate) AS FirstInvoiceDate, SUM(InvoiceTotal) AS SumOfInvoices
FROM Vendors JOIN Invoices
ON Vendors.VendorID = Invoices.VendorID
WHERE InvoiceTotal - CreditTotal - PaymentTotal = 0
GROUP BY VendorName
ORDER BY SUM(InvoiceTotal)
```

```
Commands completed successfully.

Completion time: 2022-10-28T16:27:42.0004723-04:00
```

Query executed successfully.

SELECT * FROM Top10PaidInvoices



```
SELECT * FROM Top10PaidInvoices
```

	VendorName	FirstInvoiceDate	SumOfInvoices
1	Suburban Propane	2021-03-15 00:00:00	16.62
2	Abbey Office Furnishings	2021-03-05 00:00:00	17.50
3	Compuserve	2021-01-03 00:00:00	19.90
4	Coffee Break Service	2021-02-24 00:00:00	41.80
5	Roadway Package System, Inc	2021-01-01 00:00:00	43.67
6	Evans Executone Inc	2020-12-24 00:00:00	95.00
7	Pacific Bell	2020-12-30 00:00:00	171.01
8	Cardinal Business Media, Inc.	2021-02-22 00:00:00	175.00
9	Edward Data Services	2021-01-15 00:00:00	207.78
10	Dristas Groom & McCormick	2021-01-23 00:00:00	220.00

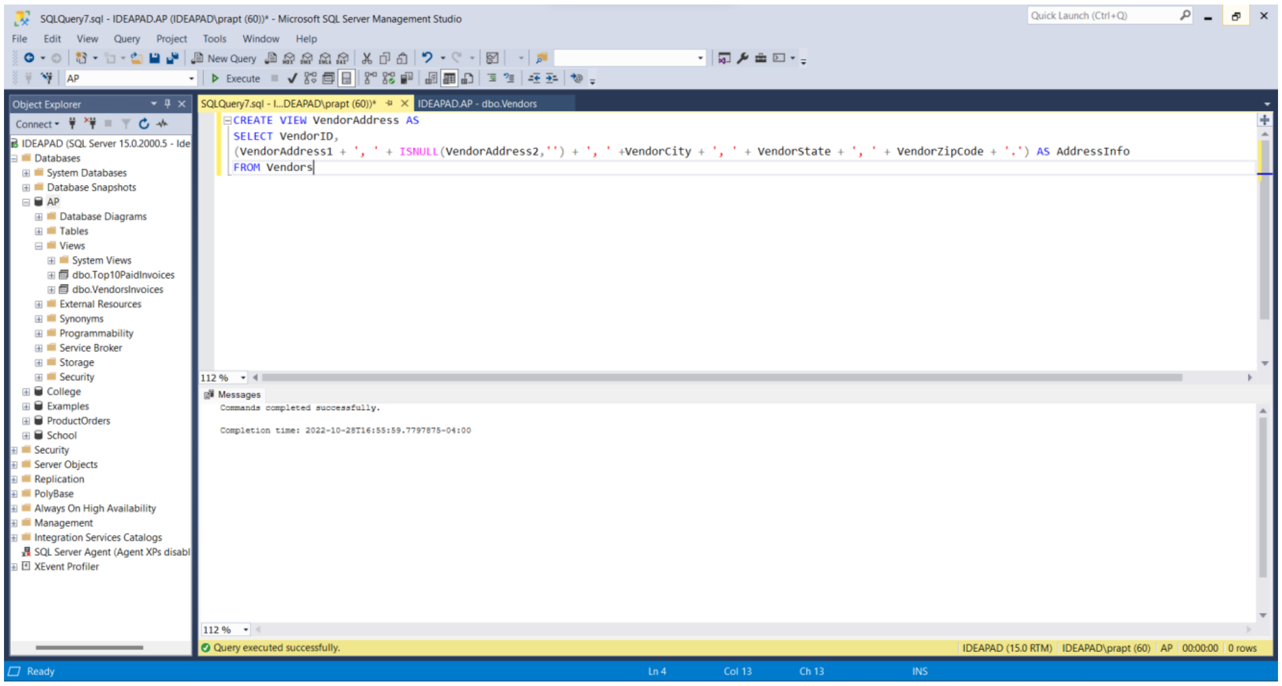
Query executed successfully.

Comment: Here, CREATE VIEW statement is used to create a view in the database and is named as Top10PaidInvoices using the AS keyword. This view is a table that contains VendorName column from the Vendors table. It contains the FirstinvoiceDate column that is retrieved using the aggregate function on the InvoiceDate column, SumOfInvoices column that is retrieved by using the aggregate function SUM() on the InvoiceTotal column from the Invoices table which are selected using the SELECT TOP 10 statement to retrieve the first 10 rows. This is done using JOIN on Vendors and Invoices table on VendorID using ON keyword. To include only paid invoices we use WHERE clause to find only those records where InvoiceTotal – CreditTotal – PaymentTotal = 0. Grouping is done on the VendorName column using GROUP BY keyword and the result set is sorted in ascending order on the SumOfInvoices column using ORDER BY keyword to get the smallest SumOfInvoices.

Q3) Create an updatable view named VendorAddress that returns the VendorID and AddressInfo (i.e. VendorAddress1 + ‘, ’ + VendorAddress2 + ‘, ’ + VendorCity + ‘, ’ + VendorState + ‘, ’ + VendorZipCode + ‘.’) for each vendor. Whenever VendorAddress2 is NULL, substitute the NULL with a blank space. Then write a SELECT query to examine the result set where VendorID = 5. Write a SELECT statement to verify the result.

Ans:

```
CREATE VIEW VendorAddress AS

SELECT VendorID,

(VendorAddress1 + ', ' + ISNULL(VendorAddress2,'') + ', ' +VendorCity + ', ' + VendorState + ', ' + VendorZipCode + '.') AS AddressInfo

FROM Vendors
```



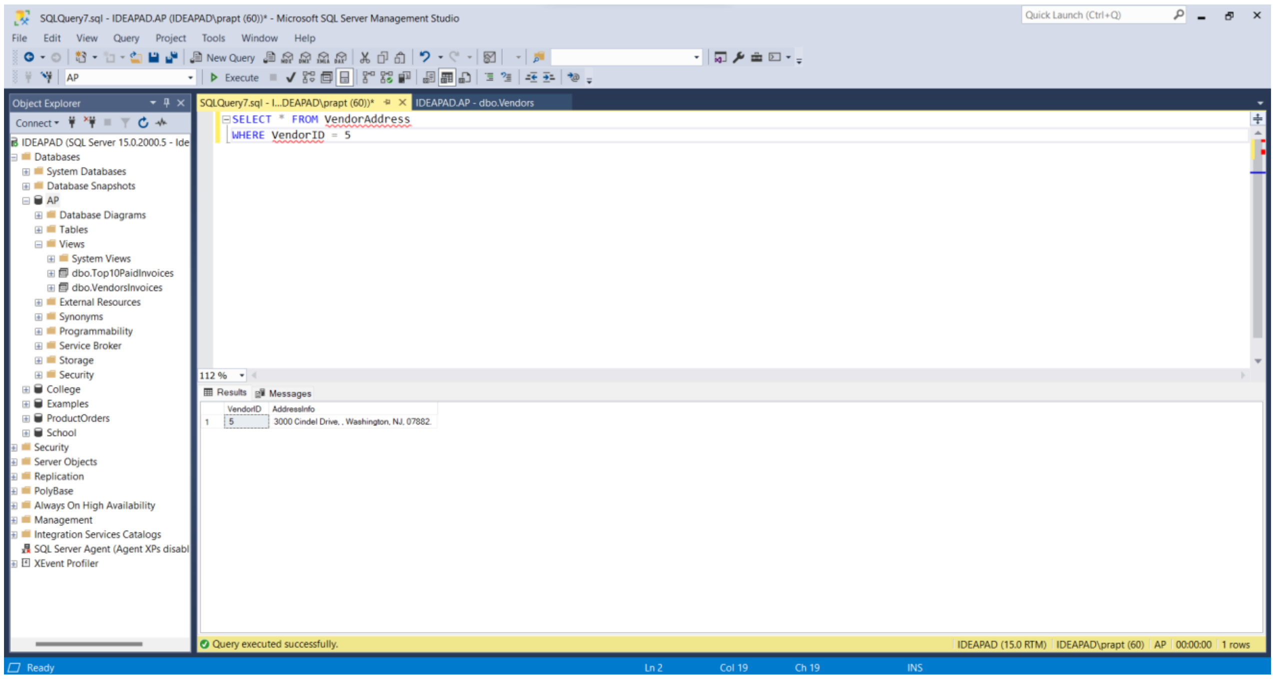
```
CREATE VIEW VendorAddress AS
SELECT VendorID,
(VendorAddress1 + ', ' + ISNULL(VendorAddress2,'') + ', ' +VendorCity + ', ' + VendorState + ', ' + VendorZipCode + '.') AS AddressInfo
FROM Vendors
```

```
Commands completed successfully.

Completion time: 2022-10-28T16:55:59.7797875-04:00
```

Query executed successfully.

SELECT * FROM VendorAddress

WHERE VendorID = 5

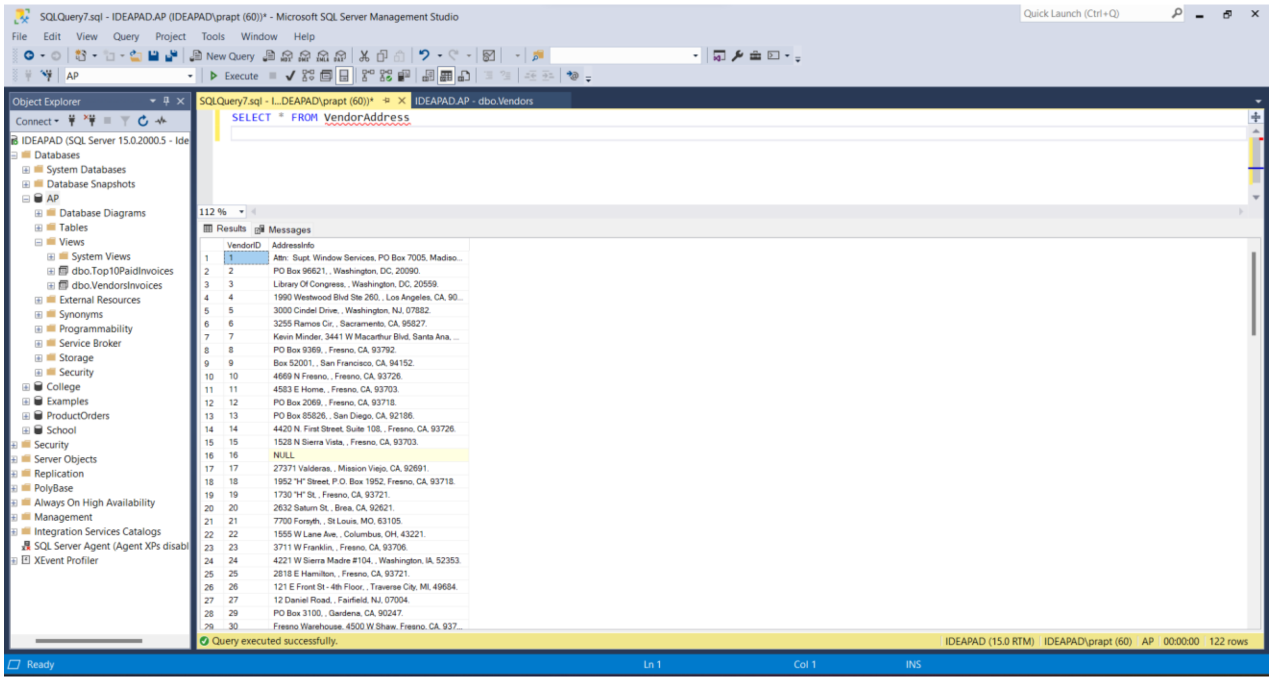


```
SELECT * FROM VendorAddress
WHERE VendorID = 5
```

	VendorID	AddressInfo
1	5	3000 Cindel Drive, , Washington, NJ, 07882.

Query executed successfully.

SELECT * FROM VendorAddress



```
SELECT * FROM VendorAddress
```

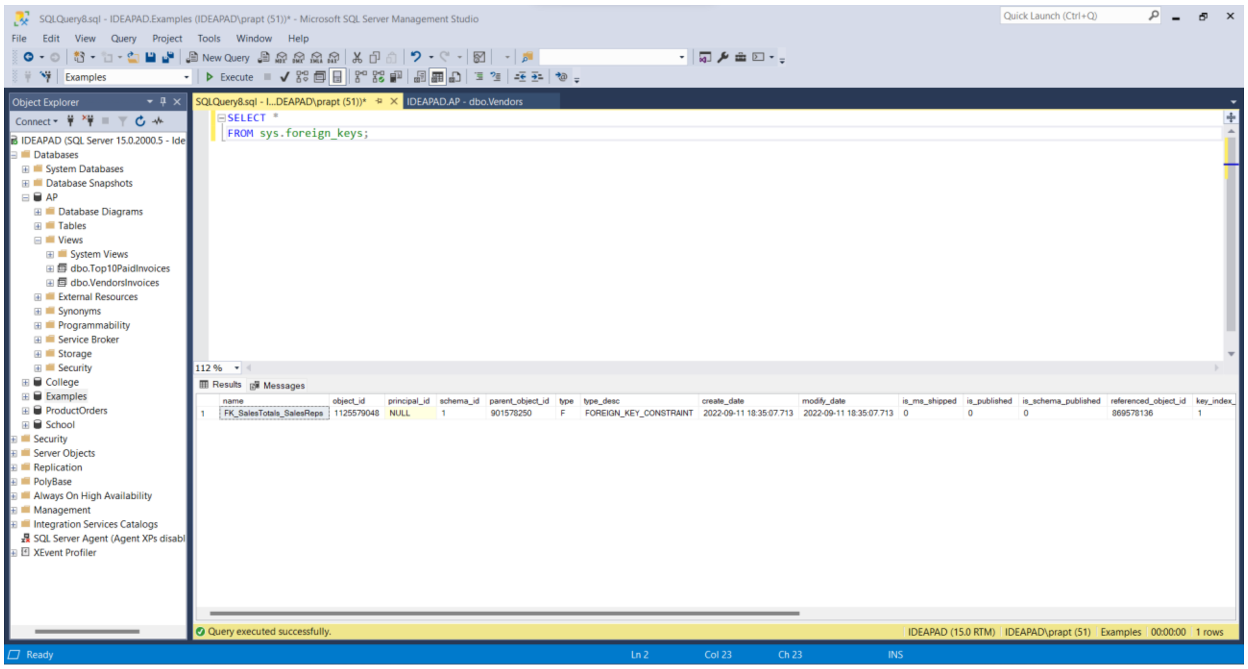
	VendorID	AddressInfo
1	1	Attn: Supt. Window Services, PO Box 7005, Madiso...
2	2	PO Box 96621, , Washington, DC, 20090.
3	3	Library Of Congress, , Washington, DC, 20559.
4	4	1990 Westwood Blvd Ste 260, , Los Angeles, CA, 90...
5	5	3000 Cindel Drive, , Washington, NJ, 07882.
6	6	3255 Ramos Cir, , Sacramento, CA, 95827.
7	7	Kevin Minder, 3441 W Macarthur Blvd, Santa Ana, ...
8	8	PO Box 9369, , Fresno, CA, 93792.
9	9	Box 52001, , San Francisco, CA, 94152.
10	10	4669 N Fresno, , Fresno, CA, 93726.
11	11	4583 E Home, , Fresno, CA, 93703.
12	12	PO Box 2069, , Fresno, CA, 93718.
13	13	PO Box 85826, , San Diego, CA, 92186.
14	14	4420 N. First Street, Suite 108, , Fresno, CA, 93726.
15	15	1528 N Sierra Vista, , Fresno, CA, 93703.
16	16	NULL
17	17	27371 Valderas, , Mission Viejo, CA, 92691.
18	18	1952 "H" Street, P.O. Box 1952, Fresno, CA, 93718.
19	19	1730 "H" St, , Fresno, CA, 93721.
20	20	2632 Saturn St, , Brea, CA, 92621.
21	21	7700 Forsyth, , St Louis, MO, 63105.
22	22	1555 W Lane Ave, , Columbus, OH, 43221.
23	23	3711 W Franklin, , Fresno, CA, 93706.
24	24	4221 W Sierra Madre #104, , Washington, IA, 52353.
25	25	2818 E Hamilton, , Fresno, CA, 93721.
26	26	121 E Front St - 4th Floor, , Traverse City, MI, 49684.
27	27	12 Daniel Road, , Fairfield, NJ, 07004.
28	29	PO Box 3100, , Gardena, CA, 90247.
29	30	Fresno Warehouse, 4500 W Shaw, Fresno, CA, 937...

Query executed successfully. IDEAPAD (15.0 RTM) IDEAPAD\prapt (60) AP 00:00:00 122 rows

Comment: Here, CREATE VIEW statement is used to create a view in the database and is named as VendorAddress using the AS keyword. This view is a table that contains VendorID column, concatenation VendorAddress1, VendorAddress2 or blank space if the VendorAdress2 is null which is done using ISNULL() function, VendorCity, VendorState, VendorZipCode. Then, SELECT * statement is used to retrieve data from the VendorAddress view.

Q4) Write a SELECT statement that selects all the columns for the catalog view that returns information about foreign keys in ***the Examples database***. How many foreign key(s) is/are defined in the Examples database and what is/are they?

Ans: SELECT * FROM sys.foreign_keys;



```
SELECT *
FROM sys.foreign_keys;
```

	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published	referenced_object_id	key_index_
1	FK_SalesTotals_SalesReps	1125579048	NULL	1	901578250	F	FOREIGN_KEY_CONSTRAINT	2022-09-11 18:35:07.713	2022-09-11 18:35:07.713	0	0	0	869578136	1

```
Query executed successfully.
```

Comment: Here, SELECT * statement is used to select all the columns for the catalog view that returns all the foreign keys in the Examples database. There is one foreign key.

Q5) Write a script that declares and sets a variable named @TotalBalanceDue, which is equal to the total outstanding balance due. What is the datatype of the variable @TotalBalanceDue? If that balance due is less than $50,000.00, the script should return a result set consisting of VendorName, InvoiceNumber, InvoiceDueDate, and Balance for each invoice with a balance due, sorted with the newest due date first. Then also return the value of @TotalBalanceDue in the format of "Balance due is …". If the total outstanding balance due is more than $50,000.00, the script should return the message "Balance due is more than $50,000.00".

Ans:

```
DECLARE @TotalInvoiceDue money;

SELECT @TotalInvoiceDue = SUM(InvoiceTotal - CreditTotal - PaymentTotal)

FROM Invoices

WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0;

IF @TotalInvoiceDue < 50000

BEGIN

PRINT 'Balance due is ' + CAST(@TotalInvoiceDue as NVARCHAR(9))+ '.'
```

SELECT VendorName, InvoiceNumber, InvoiceDueDate, InvoiceTotal - CreditTotal - PaymentTotal AS Balance

FROM Invoices JOIN Vendors

ON Invoices.VendorID = Vendors.VendorID

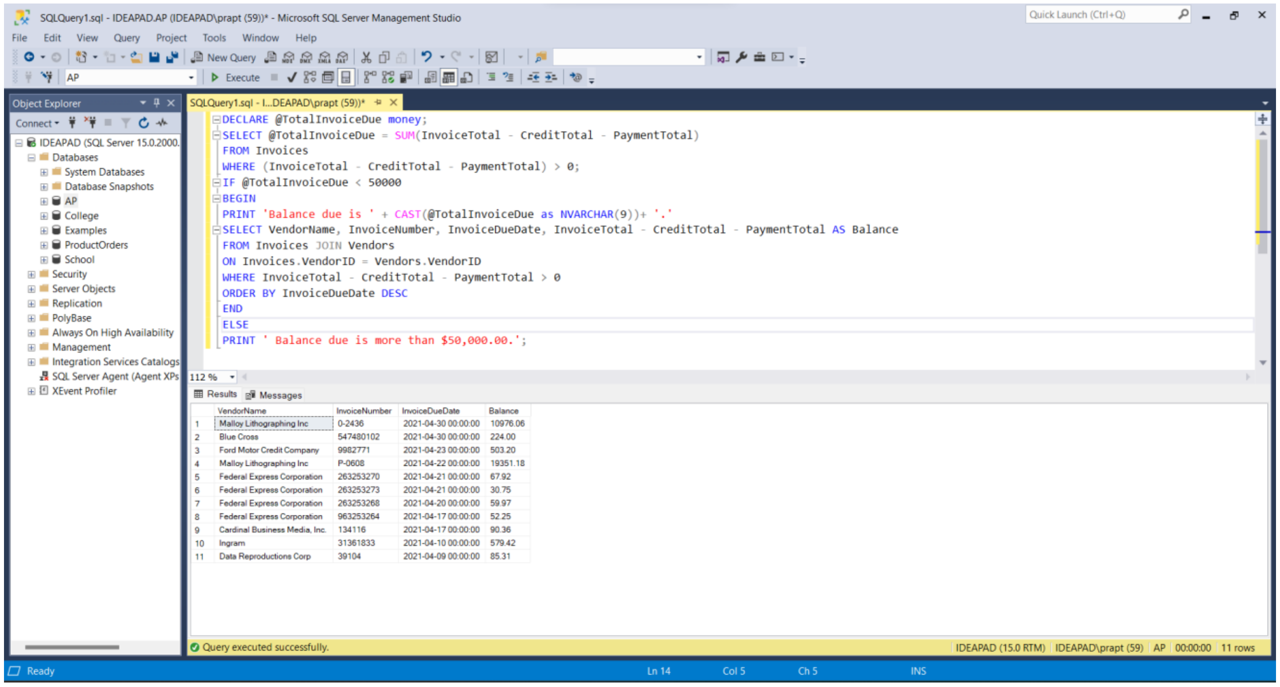
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0

ORDER BY InvoiceDueDate DESC

END

ELSE

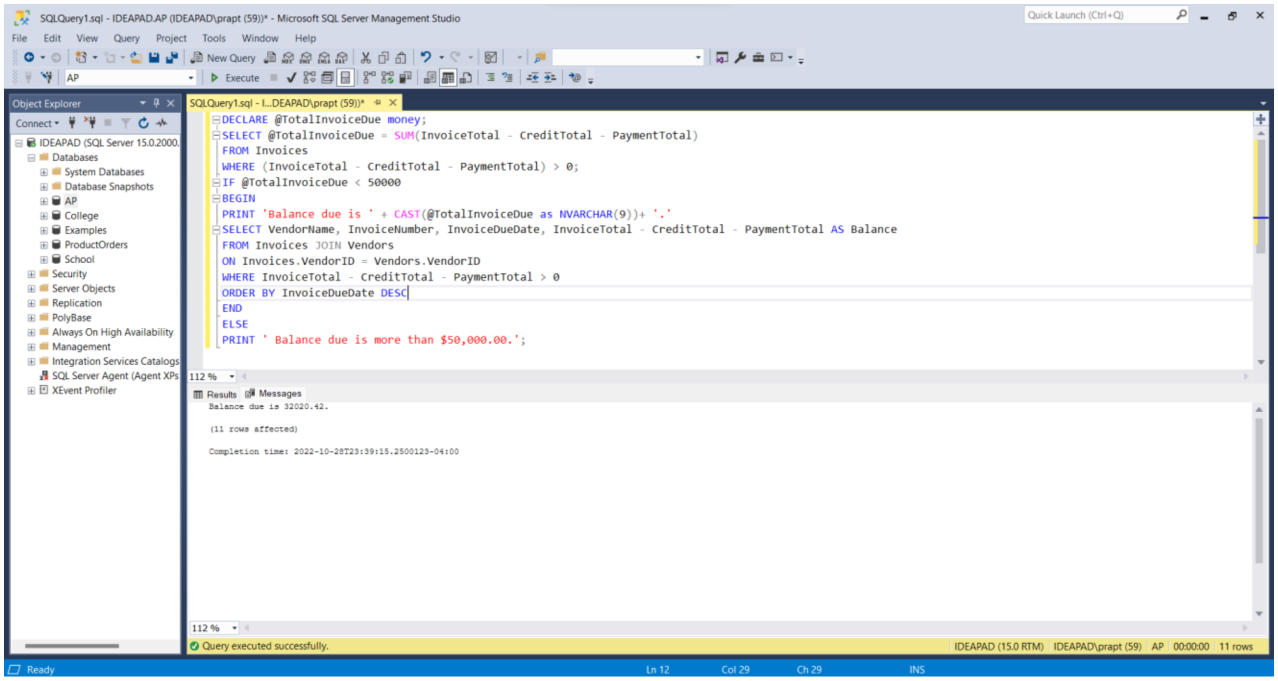
PRINT ' Balance due is more than $50,000.00.';



```sql
DECLARE @TotalInvoiceDue money;
SELECT @TotalInvoiceDue = SUM(InvoiceTotal - CreditTotal - PaymentTotal)
FROM Invoices
WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0;
IF @TotalInvoiceDue < 50000
BEGIN
PRINT 'Balance due is ' + CAST(@TotalInvoiceDue as NVARCHAR(9))+ '.'
SELECT VendorName, InvoiceNumber, InvoiceDueDate, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
FROM Invoices JOIN Vendors
ON Invoices.VendorID = Vendors.VendorID
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0
ORDER BY InvoiceDueDate DESC
END
ELSE
PRINT ' Balance due is more than $50,000.00.';
```

	VendorName	InvoiceNumber	InvoiceDueDate	Balance
1	Malloy Lithographing Inc	0-2436	2021-04-30 00:00:00	10976.06
2	Blue Cross	547480102	2021-04-30 00:00:00	224.00
3	Ford Motor Credit Company	9982771	2021-04-23 00:00:00	503.20
4	Malloy Lithographing Inc	P-0608	2021-04-22 00:00:00	19351.18
5	Federal Express Corporation	263253270	2021-04-21 00:00:00	67.92
6	Federal Express Corporation	263253273	2021-04-21 00:00:00	30.75
7	Federal Express Corporation	263253268	2021-04-20 00:00:00	59.97
8	Federal Express Corporation	963253264	2021-04-17 00:00:00	52.25
9	Cardinal Business Media, Inc.	134116	2021-04-17 00:00:00	90.36
10	Ingram	31361833	2021-04-10 00:00:00	579.42
11	Data Reproductions Corp	39104	2021-04-09 00:00:00	85.31

Query executed successfully.



```
DECLARE @TotalInvoiceDue money;
SELECT @TotalInvoiceDue = SUM(InvoiceTotal - CreditTotal - PaymentTotal)
FROM Invoices
WHERE (InvoiceTotal - CreditTotal - PaymentTotal) > 0;
IF @TotalInvoiceDue < 50000
BEGIN
PRINT 'Balance due is ' + CAST(@TotalInvoiceDue as NVARCHAR(9))+ '.'
SELECT VendorName, InvoiceNumber, InvoiceDueDate, InvoiceTotal - CreditTotal - PaymentTotal AS Balance
FROM Invoices JOIN Vendors
ON Invoices.VendorID = Vendors.VendorID
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0
ORDER BY InvoiceDueDate DESC
END
ELSE
PRINT ' Balance due is more than $50,000.00.';
```

```
Balance due is 32020.42.

(11 rows affected)

Completion time: 2022-10-28T23:39:15.2500123-04:00
```

Query executed successfully.

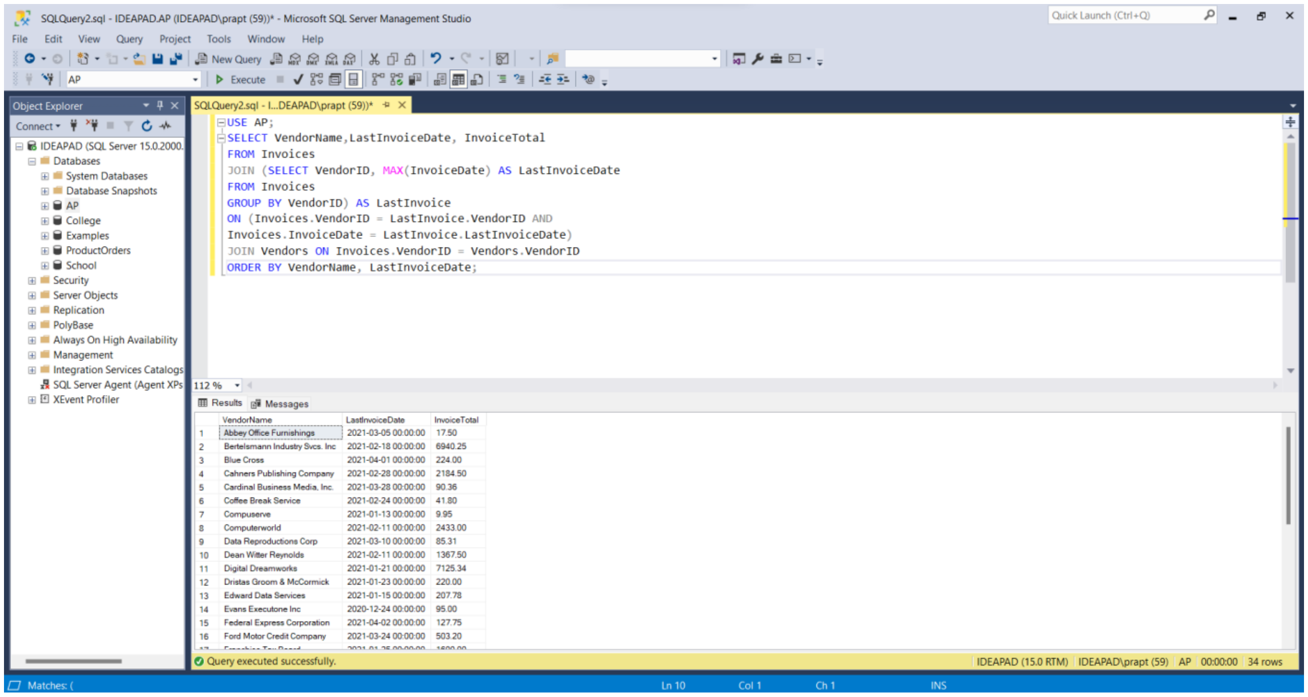
Comment: DECLARE statement is used to declare a variable after (@) sign. Datatype of that variable is also declared. SELECT statement is used to assign a value to the variable declared using the DECLARE statement. Here, a variable named TotalInvoiceDue is declared using the DECLARE statement with money datatype. The calculated value of InvoiceTotal – CreditTotal – PaymentTotal from the Invoices table where the sum is greater than 0 obtained by the SUM() aggregate function and the WHERE clause is assigned to the variable using SELECT statement. IF…ELSE is used to return result sets if the balance due is less than $50,000 else return something else. To return the result set for which the balance due is less than $50,000, check the condition in IF statement. If it is true, then the block starts with BEGIN statement for multiple SELECT and PRINT statements to be executed. First, using SELECT statement a result set where, VendorName, InvoiceNumber, InvoiceDueDate and Balance columns from the JOIN of the Invoices and Vendors table such that Balance due is greater than zero which is ordered by the newest due date first is returned. PRINT statement is used to print messages in the messages window. PRINT statement is used in the IF block along with the SELECT statement to print the message that contains the amount of the Balance that is due using CAST and concatenation. This blocks ends with END. ELSE block contains the PRINT statement that prints the message that the Balance due is greater than $50,000.00.

Q6) Explain the execution result generated by the following script. Then Write a script that generates the same result set but uses a temporary table in place of the derived table. Make sure your script tests for the existence of any objects it creates.

USE AP;

```
 SELECT VendorName,LastInvoiceDate, InvoiceTotal
 FROM Invoices
JOIN (SELECT VendorID, MAX(InvoiceDate) AS LastInvoiceDate
FROM Invoices
GROUP BY VendorID) AS LastInvoice
ON (Invoices.VendorID = LastInvoice.VendorID AND
 Invoices.InvoiceDate = LastInvoice.LastInvoiceDate)
 JOIN Vendors ON Invoices.VendorID = Vendors.VendorID
 ORDER BY VendorName, LastInvoiceDate;
```

Comment: Here, USE AP is used to use the database AP I.e, perform the query in AP database. Derived table named LastInvoice is created using the SELECT statement that retrieves the VendorID and the maximum of the InvoiceDate as LastInvoiceDate using the MAX() function from Invoices table and grouped by VendorID using the GROUP BY keyword. SELECT statement is used to retrieve the VendorName, LastInvoiceDate, InvoiceTotal from the Invoices, LastInvoice(derived table) and Vendors tables for which JOIN is used on these three tables.



```
USE AP;
SELECT VendorName,LastInvoiceDate, InvoiceTotal
FROM Invoices
JOIN (SELECT VendorID, MAX(InvoiceDate) AS LastInvoiceDate
FROM Invoices
GROUP BY VendorID) AS LastInvoice
ON (Invoices.VendorID = LastInvoice.VendorID AND
Invoices.InvoiceDate = LastInvoice.LastInvoiceDate)
JOIN Vendors ON Invoices.VendorID = Vendors.VendorID
ORDER BY VendorName, LastInvoiceDate;
```

	VendorName	LastInvoiceDate	InvoiceTotal
1	Abbey Office Furnishings	2021-03-05 00:00:00	17.50
2	Bertelsmann Industry Svcs. Inc	2021-02-18 00:00:00	6940.25
3	Blue Cross	2021-04-01 00:00:00	224.00
4	Cahners Publishing Company	2021-02-28 00:00:00	2184.50
5	Cardinal Business Media, Inc.	2021-03-28 00:00:00	90.36
6	Coffee Break Service	2021-02-24 00:00:00	41.80
7	Compuserve	2021-01-13 00:00:00	9.95
8	Computerworld	2021-02-11 00:00:00	2433.00
9	Data Reproductions Corp	2021-03-10 00:00:00	85.31
10	Dean Witter Reynolds	2021-02-11 00:00:00	1367.50
11	Digital Dreamworks	2021-01-21 00:00:00	7125.34
12	Dristas Groom & McCormick	2021-01-23 00:00:00	220.00
13	Edward Data Services	2021-01-15 00:00:00	207.78
14	Evans Executone Inc	2020-12-24 00:00:00	95.00
15	Federal Express Corporation	2021-04-02 00:00:00	127.75
16	Ford Motor Credit Company	2021-03-24 00:00:00	503.20

Query executed successfully.   IDEAPAD (15.0 RTM) | IDEAPAD\prapt (59) | AP | 00:00:00 | 34 rows

ANS:

USE AP;

IF OBJECT_ID('tempdb..#LastInvoice') IS NOT NULL

DROP TABLE #LastInvoice;

SELECT VendorID, MAX(InvoiceDate) AS LastInvoiceDate

INTO #LastInvoice

FROM Invoices

GROUP BY VendorID;

SELECT VendorName, LastInvoiceDate, InvoiceTotal

FROM Invoices JOIN #LastInvoice

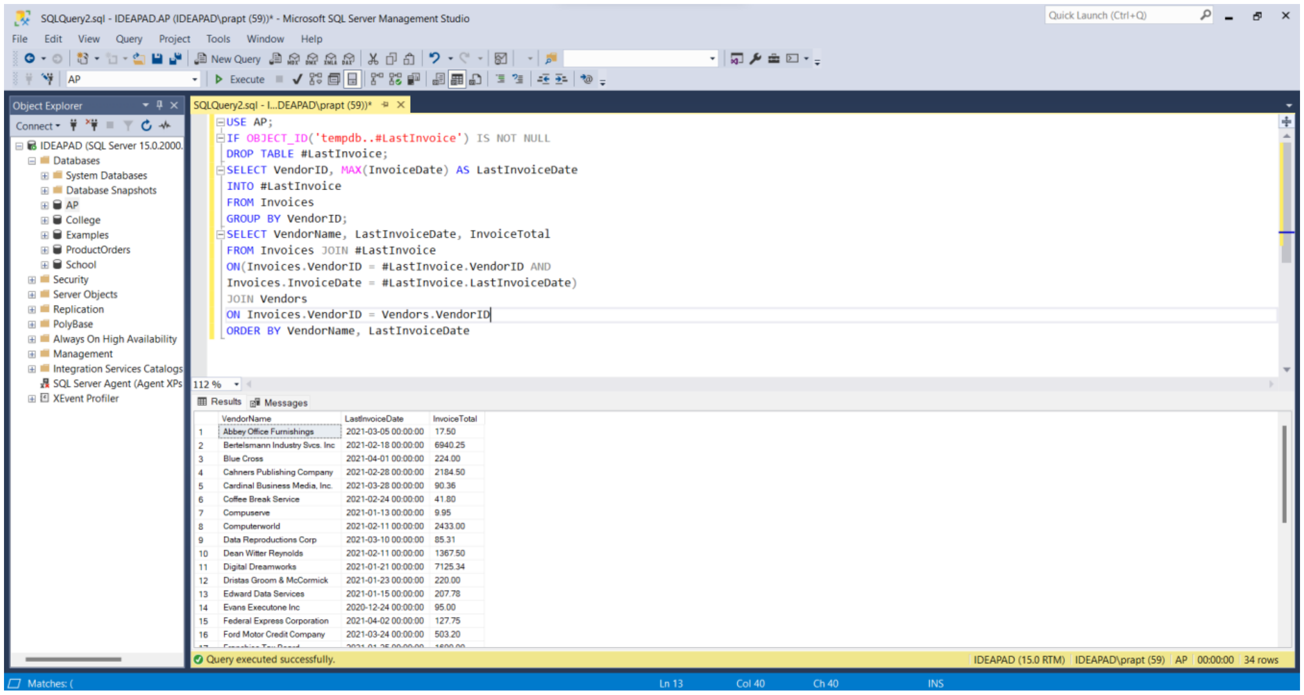
ON(Invoices.VendorID = #LastInvoice.VendorID AND

Invoices.InvoiceDate = #LastInvoice.LastInvoiceDate)

JOIN Vendors

ON Invoices.VendorID = Vendors.VendorID

ORDER BY VendorName, LastInvoiceDate



```
USE AP;
IF OBJECT_ID('tempdb..#LastInvoice') IS NOT NULL
DROP TABLE #LastInvoice;
SELECT VendorID, MAX(InvoiceDate) AS LastInvoiceDate
INTO #LastInvoice
FROM Invoices
GROUP BY VendorID;
SELECT VendorName, LastInvoiceDate, InvoiceTotal
FROM Invoices JOIN #LastInvoice
ON(Invoices.VendorID = #LastInvoice.VendorID AND
Invoices.InvoiceDate = #LastInvoice.LastInvoiceDate)
JOIN Vendors
ON Invoices.VendorID = Vendors.VendorID
ORDER BY VendorName, LastInvoiceDate
```

	VendorName	LastInvoiceDate	InvoiceTotal
1	Abbey Office Furnishings	2021-03-05 00:00:00	17.50
2	Bertelsmann Industry Svcs. Inc	2021-02-18 00:00:00	6940.25
3	Blue Cross	2021-04-01 00:00:00	224.00
4	Cahners Publishing Company	2021-02-28 00:00:00	2184.50
5	Cardinal Business Media, Inc.	2021-03-28 00:00:00	90.36
6	Coffee Break Service	2021-02-24 00:00:00	41.80
7	Compuserve	2021-01-13 00:00:00	9.95
8	Computerworld	2021-02-11 00:00:00	2433.00
9	Data Reproductions Corp	2021-03-10 00:00:00	85.31
10	Dean Witter Reynolds	2021-02-11 00:00:00	1367.50
11	Digital Dreamworks	2021-01-21 00:00:00	7125.34
12	Dristas Groom & McCormick	2021-01-23 00:00:00	220.00
13	Edward Data Services	2021-01-15 00:00:00	207.78
14	Evans Executone Inc	2020-12-24 00:00:00	95.00
15	Federal Express Corporation	2021-04-02 00:00:00	127.75
16	Ford Motor Credit Company	2021-03-24 00:00:00	503.20

Query executed successfully. IDEAPAD (15.0 RTM) | IDEAPAD\prapt (59) | AP | 00:00:00 | 34 rows

Comment: Here, Here, USE AP is used to use the database AP I.e, perform the query in AP database. IF statement is used to check whether there is any temporary table already existing which is not null and if it does then DROP TABLE is used to drop the temporary table. New temporary table named #LastInvoice is created in which the result set containing VendorName, Max(InvoiceDate) as LastInvoiceDate is stored using SELECT INTO from the Invoices table and grouped by VendorID. Finally, SELECT statement is used to select the VendorName, LastInvoiceDate, InvoiceTotal columns from the JOIN of three tables namely Invoices, Vendors and the temporary table LastInvoice.

Q7) Write a script that generates the date and invoice total of the latest invoice issued by each vendor, using a view instead of a derived table. Also write the script that creates the view, then use SELECT statement to show result of the view. Make sure that your script tests for the existence of the view. The view doesn't need to be redefined each time the script is executed.

Ans: USE AP;

IF OBJECT_ID('Latest_Invoices') IS NOT NULL

DROP VIEW Latest_Invoices;

GO

CREATE VIEW Latest_Invoices

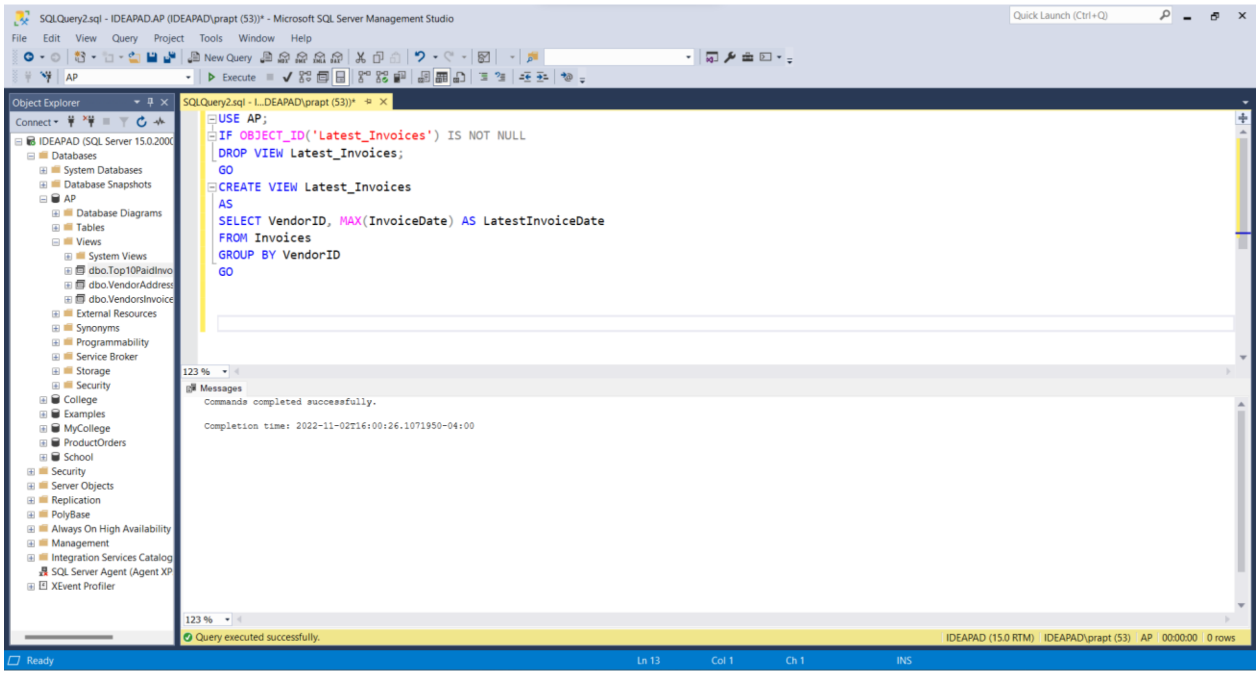
AS

SELECT VendorID, MAX(InvoiceDate) AS LatestInvoiceDate

FROM Invoices

GROUP BY VendorID

GO

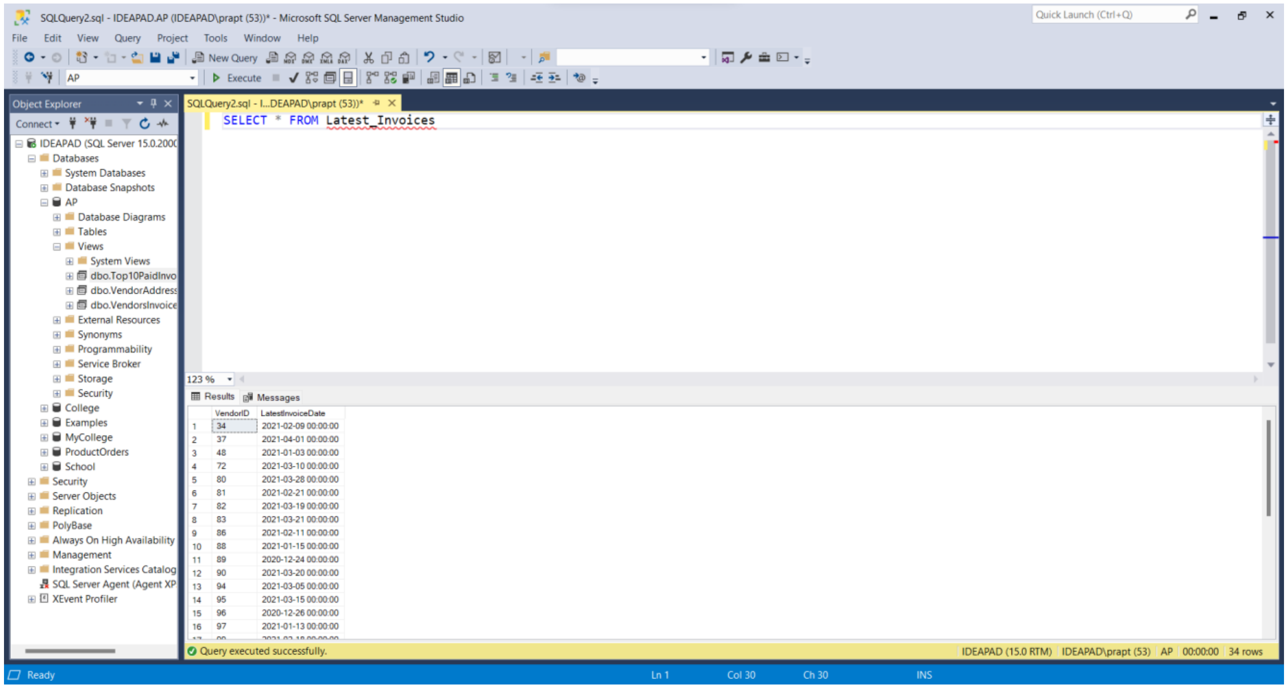


```
USE AP;
IF OBJECT_ID('Latest_Invoices') IS NOT NULL
DROP VIEW Latest_Invoices;
GO
CREATE VIEW Latest_Invoices
AS
SELECT VendorID, MAX(InvoiceDate) AS LatestInvoiceDate
FROM Invoices
GROUP BY VendorID
GO
```

```
Commands completed successfully.

Completion time: 2022-11-02T16:00:26.1071950-04:00
```

Query executed successfully.

SELECT * FROM Latest_Invoices



```
SELECT * FROM Latest_Invoices
```

	VendorID	LatestInvoiceDate
1	34	2021-02-09 00:00:00
2	37	2021-04-01 00:00:00
3	48	2021-01-03 00:00:00
4	72	2021-03-10 00:00:00
5	80	2021-03-28 00:00:00
6	81	2021-02-21 00:00:00
7	82	2021-03-19 00:00:00
8	83	2021-03-21 00:00:00
9	86	2021-02-11 00:00:00
10	88	2021-01-15 00:00:00
11	89	2020-12-24 00:00:00
12	90	2021-03-20 00:00:00
13	94	2021-03-05 00:00:00
14	95	2021-03-15 00:00:00
15	96	2020-12-26 00:00:00
16	97	2021-01-13 00:00:00

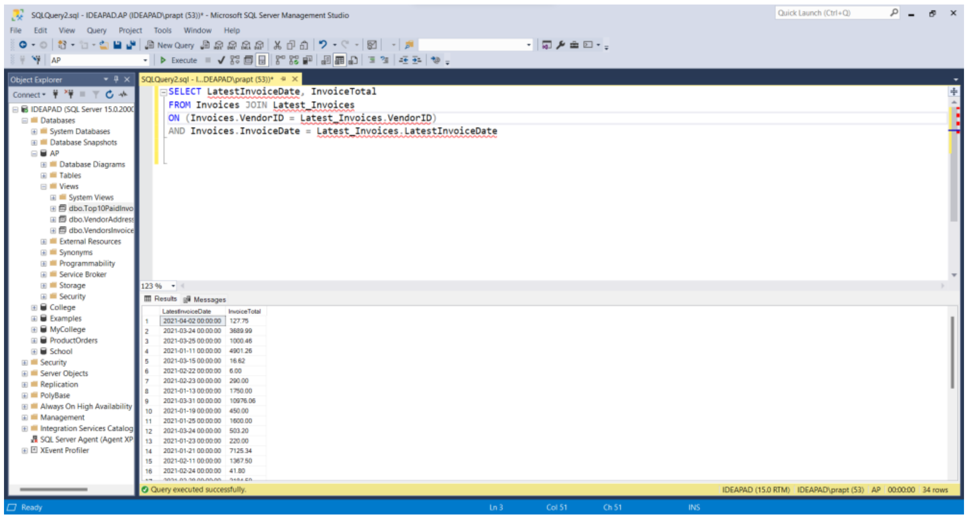
Query executed successfully. 34 rows

SELECT LatestInvoiceDate, InvoiceTotal

FROM Invoices JOIN Latest_Invoices

ON (Invoices.VendorID = Latest_Invoices.VendorID)

AND Invoices.InvoiceDate = Latest_Invoices.LatestInvoiceDate



```
SELECT LatestInvoiceDate, InvoiceTotal
FROM Invoices JOIN Latest_Invoices
ON (Invoices.VendorID = Latest_Invoices.VendorID)
AND Invoices.InvoiceDate = Latest_Invoices.LatestInvoiceDate
```

	LatestInvoiceDate	InvoiceTotal
1	2021-04-02 00:00:00	127.75
2	2021-03-24 00:00:00	3689.99
3	2021-03-25 00:00:00	1000.46
4	2021-01-11 00:00:00	4901.26
5	2021-03-15 00:00:00	16.62
6	2021-02-22 00:00:00	6.00
7	2021-02-23 00:00:00	290.00
8	2021-01-13 00:00:00	1750.00
9	2021-03-31 00:00:00	10976.06
10	2021-01-19 00:00:00	450.00
11	2021-01-25 00:00:00	1600.00
12	2021-03-24 00:00:00	503.20
13	2021-01-23 00:00:00	220.00
14	2021-01-21 00:00:00	7125.34
15	2021-02-11 00:00:00	1367.50
16	2021-02-24 00:00:00	41.80

Query executed successfully. IDEAPAD (15.0 RTM) IDEAPAD\prapt (53) AP 00:00:00 34 rows

Comment: Here, USE AP is used to use the database AP I.e, perform the query in AP database. A view is created using CREATE VIEW statement and this view is a table that contains VendorID column, and minimum of the InvoiceDate column that is done using MIN() aggregate function and named as LatestInvoiceDate using AS keyword from the Invoices table and is grouped by VendorID using GROUP BY. We get 34 rows as ouput by executing the SELECT * statement on Latest_Invoices. SELECT statement is used to retrieve the ouput where InvoiceTotal column and LatestInvoiceDate column are taken from Invoices and Latest_Invoices respectively using JOIN keyword on the table Invoices and the view Latest_Invoices.

Q8) Write a script that uses dynamic SQL to return a single column that represents the number of rows in the first table in the current database. The script should automatically choose the table that appears first alphabetically, and it should exclude tables named dtproperties and sysdiagrams. Name the column CountOfTable, where Table is the chosen table name. Show results for AP database.

Ans:

DECLARE @TableName varchar(128);

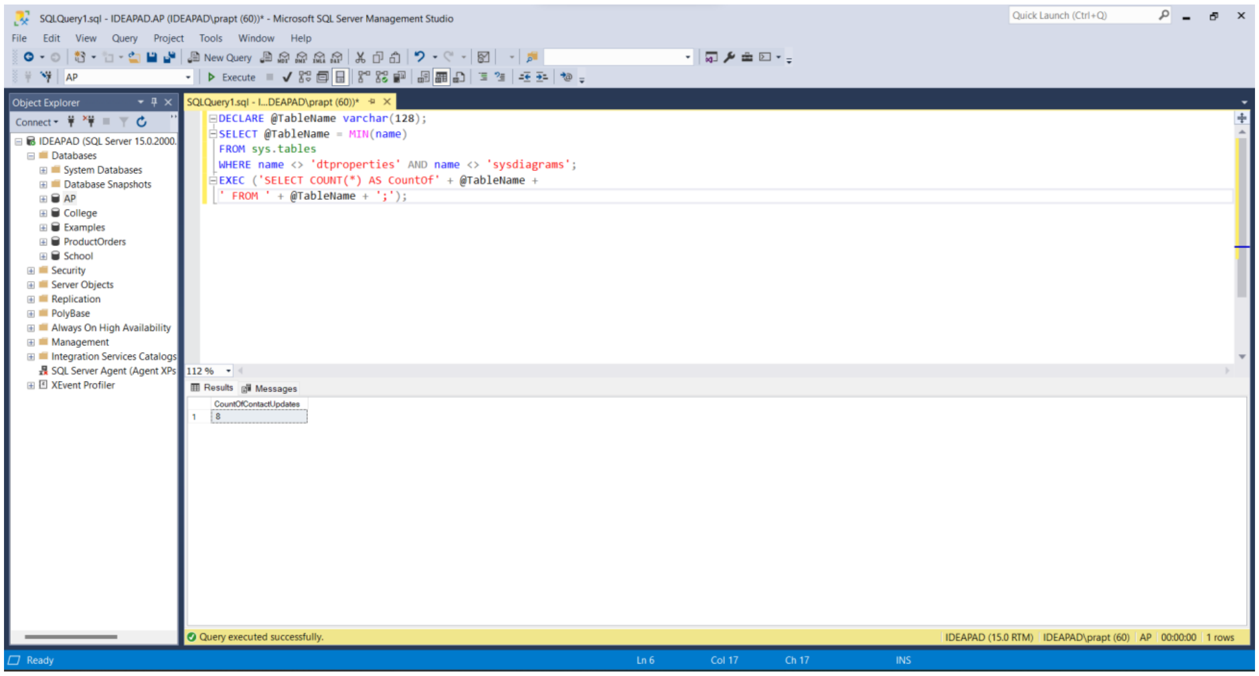
SELECT @TableName = MIN(name)

FROM sys.tables

WHERE name <> 'dtproperties' AND name <> 'sysdiagrams';

EXEC ('SELECT COUNT(*) AS CountOf' + @TableName +

' FROM ' + @TableName + ';');


SQLQuery1.sql - IDEAPAD.AP (IDEAPAD\prapt (60))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
IDEAPAD (SQL Server 15.0.2000.
Databases
System Databases
Database Snapshots
AP
College
Examples
ProductOrders
School
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs
XEvent Profiler
SQLQuery1.sql - I...DEAPAD\prapt (60))*
112 %
Results
Messages
Query executed successfully.
IDEAPAD (15.0 RTM) IDEAPAD\prapt (60) AP 00:00:00 1 rows
Ready Ln 6 Col 17 Ch 17 INS

```sql
DECLARE @TableName varchar(128);
SELECT @TableName = MIN(name)
FROM sys.tables
WHERE name <> 'dtproperties' AND name <> 'sysdiagrams';
EXEC ('SELECT COUNT(*) AS CountOf' + @TableName +
' FROM ' + @TableName + ';');
```

	CountOfContactUpdates
1	8

Comment: Here, a variable is declared named TableName with the datatype varchar(128) and using the SELECT statement a value is assigned to the variable that is minimum of the name column of the sys.tables table using the MIN() aggregate function. The records are selected such that the name is not dtproperties and sysdiagrams which is done using the WHERE clause. EXEC command is used to execute the string enclosed in () brackets which returns the count of the number of rows.

Remarks: Concepts related to creating views, retrieving data records from the views, temporary tables, scripts are implemented in this lab.