DYLAN MAHARAJ

Data Science

PM06
Geeks4Learning

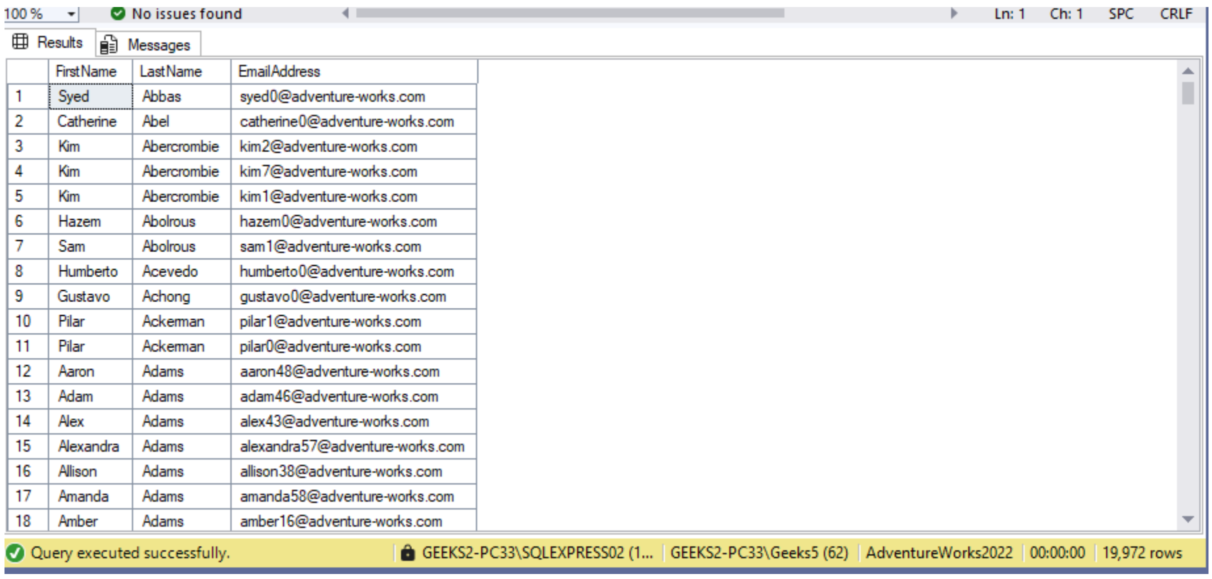
Task 1

-- 1. All customer names and emails

SELECT FirstName, LastName, EmailAddress

FROM Person.EmailAddress ea

JOIN Person.Person p ON ea.BusinessEntityID = p.BusinessEntityID;

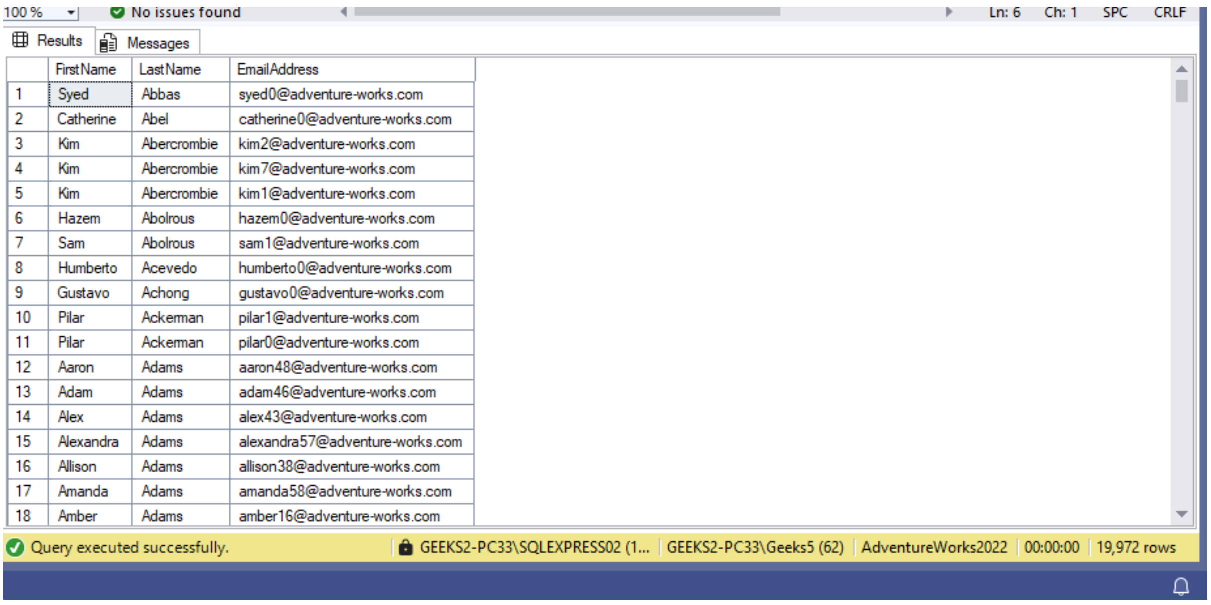


	FirstName	LastName	EmailAddress
1	Syed	Abbas	syed0@adventure-works.com
2	Catherine	Abel	catherine0@adventure-works.com
3	Kim	Abercrombie	kim2@adventure-works.com
4	Kim	Abercrombie	kim7@adventure-works.com
5	Kim	Abercrombie	kim1@adventure-works.com
6	Hazem	Abolrous	hazem0@adventure-works.com
7	Sam	Abolrous	sam1@adventure-works.com
8	Humberto	Acevedo	humberto0@adventure-works.com
9	Gustavo	Achong	gustavo0@adventure-works.com
10	Pilar	Ackerman	pilar1@adventure-works.com
11	Pilar	Ackerman	pilar0@adventure-works.com
12	Aaron	Adams	aaron48@adventure-works.com
13	Adam	Adams	adam46@adventure-works.com
14	Alex	Adams	alex43@adventure-works.com
15	Alexandra	Adams	alexandra57@adventure-works.com
16	Allison	Adams	allison38@adventure-works.com
17	Amanda	Adams	amanda58@adventure-works.com
18	Amber	Adams	amber16@adventure-works.com

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (62) | AdventureWorks2022 | 00:00:00 | 19,972 rows

-- 2. Order by last name

```
SELECT FirstName, LastName, EmailAddress

FROM Person.EmailAddress ea

JOIN Person.Person p ON ea.BusinessEntityID = p.BusinessEntityID

ORDER BY LastName;
```



100 % | No issues found | Ln: 6 | Ch: 1 | SPC | CRLF

Results | Messages

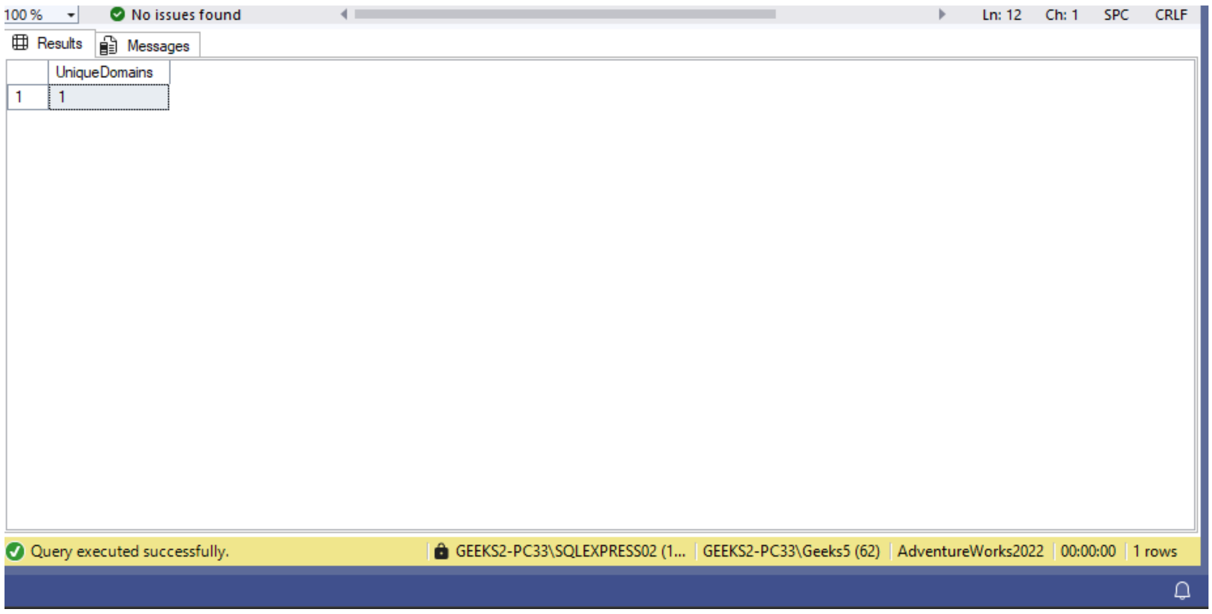
	FirstName	LastName	EmailAddress
1	Syed	Abbas	syed0@adventure-works.com
2	Catherine	Abel	catherine0@adventure-works.com
3	Kim	Abercrombie	kim2@adventure-works.com
4	Kim	Abercrombie	kim7@adventure-works.com
5	Kim	Abercrombie	kim1@adventure-works.com
6	Hazem	Abolrous	hazem0@adventure-works.com
7	Sam	Abolrous	sam1@adventure-works.com
8	Humberto	Acevedo	humberto0@adventure-works.com
9	Gustavo	Achong	gustavo0@adventure-works.com
10	Pilar	Ackerman	pilar1@adventure-works.com
11	Pilar	Ackerman	pilar0@adventure-works.com
12	Aaron	Adams	aaron48@adventure-works.com
13	Adam	Adams	adam46@adventure-works.com
14	Alex	Adams	alex43@adventure-works.com
15	Alexandra	Adams	alexandra57@adventure-works.com
16	Allison	Adams	allison38@adventure-works.com
17	Amanda	Adams	amanda58@adventure-works.com
18	Amber	Adams	amber16@adventure-works.com

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (62) | AdventureWorks2022 | 00:00:00 | 19,972 rows

-- 3. Count unique email domains

SELECT COUNT(DISTINCT SUBSTRING(EmailAddress, CHARINDEX('@', EmailAddress) + 1, LEN(EmailAddress))) AS UniqueDomains

FROM Person.EmailAddress;



```
100 %    No issues found    Ln: 12   Ch: 1   SPC   CRLF
Results   Messages
   UniqueDomains
1  1
Query executed successfully.   GEEKS2-PC33\SQLEXPRESS02 (1...  GEEKS2-PC33\Geeks5 (62)  AdventureWorks2022  00:00:00  1 rows
```

-- 4. Categorize customers

SELECT p.FirstName, p.LastName, ea.EmailAddress,

CASE

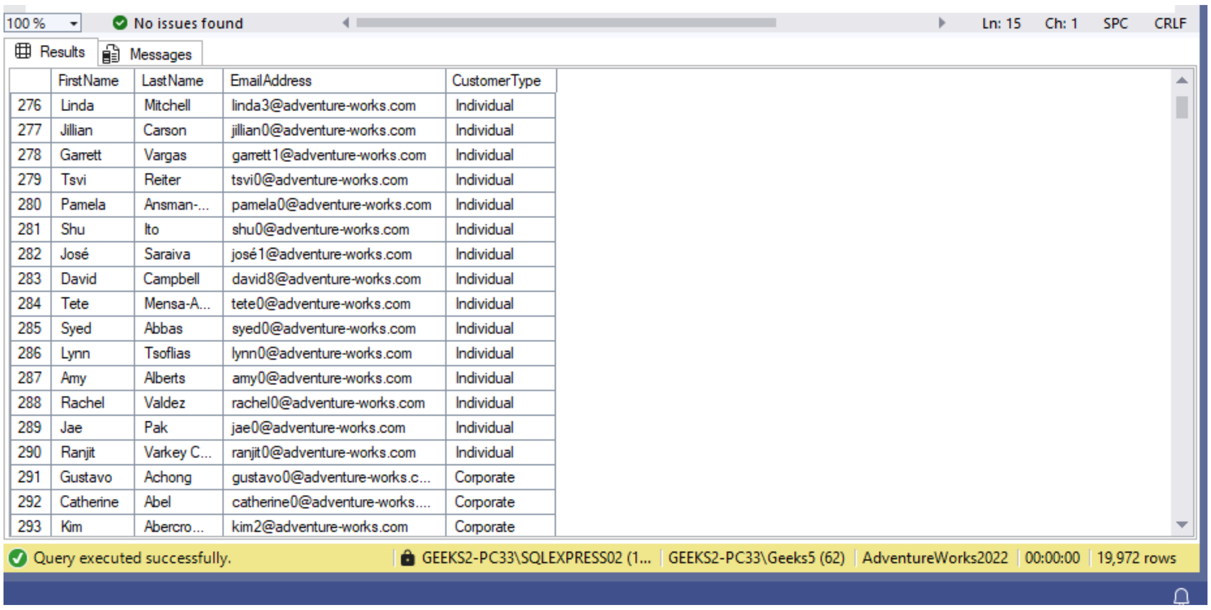
WHEN p.PersonType = 'SC' THEN 'Corporate'

ELSE 'Individual'

END AS CustomerType

FROM Person.Person p

JOIN Person.EmailAddress ea ON p.BusinessEntityID = ea.BusinessEntityID;



100 % | No issues found | Ln: 15 | Ch: 1 | SPC | CRLF

Results | Messages

	FirstName	LastName	EmailAddress	CustomerType
276	Linda	Mitchell	linda3@adventure-works.com	Individual
277	Jillian	Carson	jillian0@adventure-works.com	Individual
278	Garrett	Vargas	garrett1@adventure-works.com	Individual
279	Tsvi	Reiter	tsvi0@adventure-works.com	Individual
280	Pamela	Ansman-...	pamela0@adventure-works.com	Individual
281	Shu	Ito	shu0@adventure-works.com	Individual
282	José	Saraiva	josé1@adventure-works.com	Individual
283	David	Campbell	david8@adventure-works.com	Individual
284	Tete	Mensa-A...	tete0@adventure-works.com	Individual
285	Syed	Abbas	syed0@adventure-works.com	Individual
286	Lynn	Tsoflias	lynn0@adventure-works.com	Individual
287	Amy	Alberts	amy0@adventure-works.com	Individual
288	Rachel	Valdez	rachel0@adventure-works.com	Individual
289	Jae	Pak	jae0@adventure-works.com	Individual
290	Ranjit	Varkey C...	ranjit0@adventure-works.com	Individual
291	Gustavo	Achong	gustavo0@adventure-works.c...	Corporate
292	Catherine	Abel	catherine0@adventure-works....	Corporate
293	Kim	Abercro...	kim2@adventure-works.com	Corporate

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (62) | AdventureWorks2022 | 00:00:00 | 19,972 rows

Task 2

-- 1. Employees and Departments

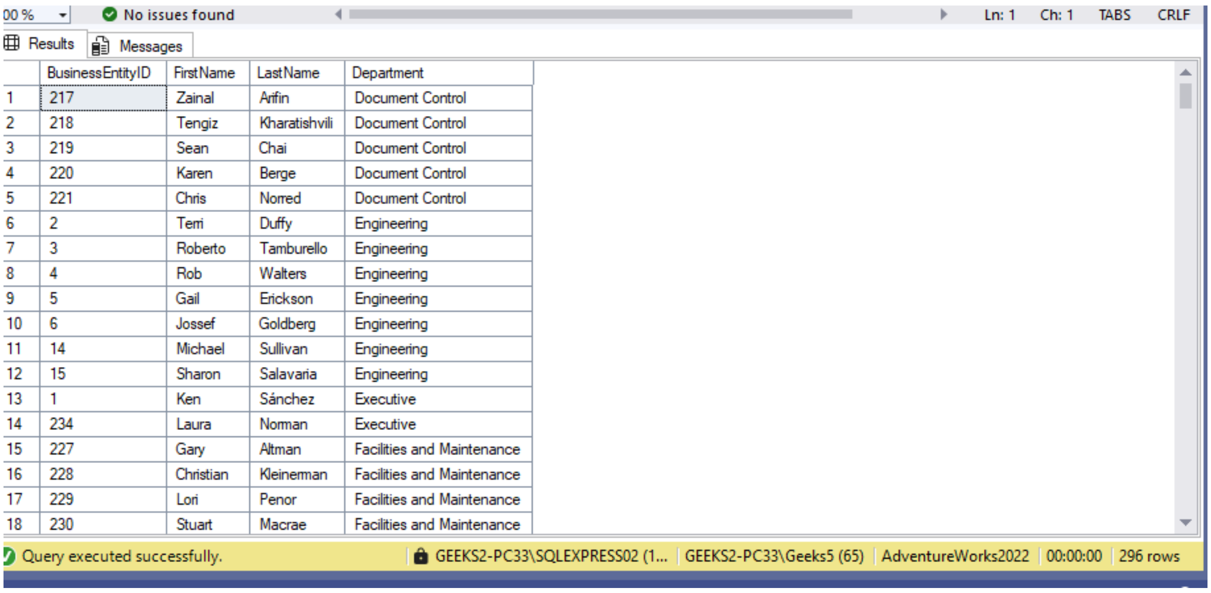
SELECT e.BusinessEntityID, p.FirstName, p.LastName, d.Name AS Department

FROM HumanResources.Employee e

JOIN Person.Person p ON e.BusinessEntityID = p.BusinessEntityID

JOIN HumanResources.EmployeeDepartmentHistory edh ON e.BusinessEntityID = edh.BusinessEntityID

JOIN HumanResources.Department d ON edh.DepartmentID = d.DepartmentID;



No issues found Ln: 1 Ch: 1 TABS CRLF

	BusinessEntityID	FirstName	LastName	Department
1	217	Zainal	Arifin	Document Control
2	218	Tengiz	Kharatishvili	Document Control
3	219	Sean	Chai	Document Control
4	220	Karen	Berge	Document Control
5	221	Chris	Norred	Document Control
6	2	Terri	Duffy	Engineering
7	3	Roberto	Tamburello	Engineering
8	4	Rob	Walters	Engineering
9	5	Gail	Erickson	Engineering
10	6	Jossef	Goldberg	Engineering
11	14	Michael	Sullivan	Engineering
12	15	Sharon	Salavaria	Engineering
13	1	Ken	Sánchez	Executive
14	234	Laura	Norman	Executive
15	227	Gary	Altman	Facilities and Maintenance
16	228	Christian	Kleinerman	Facilities and Maintenance
17	229	Lori	Penor	Facilities and Maintenance
18	230	Stuart	Macrae	Facilities and Maintenance

Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GEEKS2-PC33\Geeks5 (65) AdventureWorks2022 00:00:00 296 rows

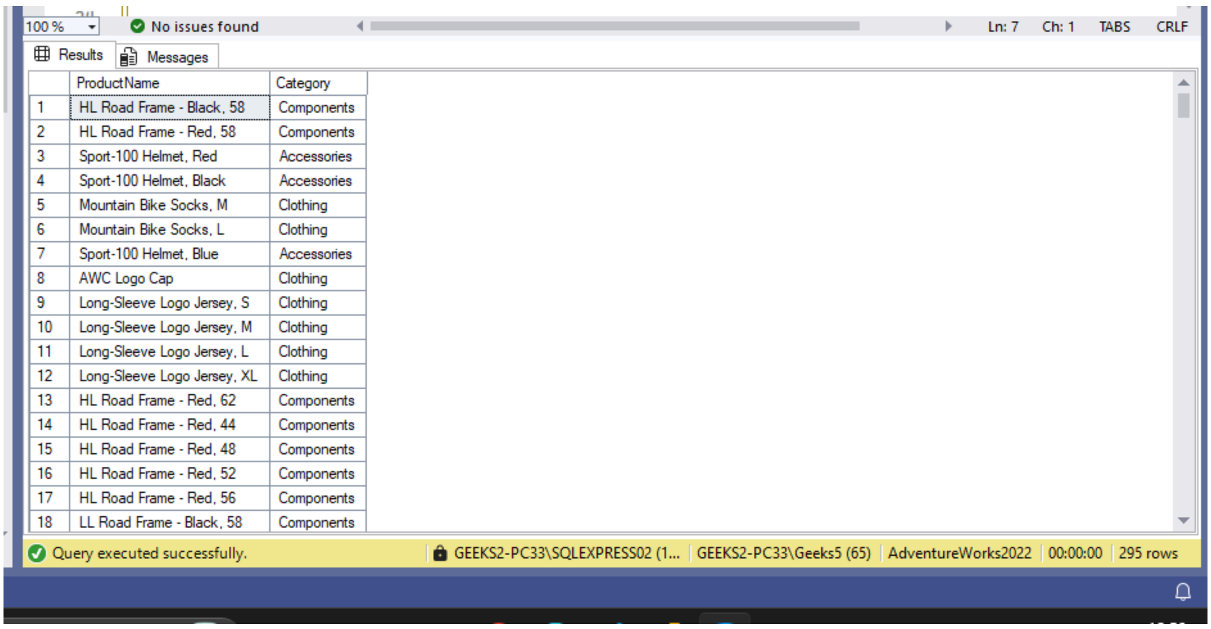
-- 2. Products and Categories

SELECT p.Name AS ProductName, pc.Name AS Category

FROM Production.Product p

JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID = ps.ProductSubcategoryID

JOIN Production.ProductCategory pc ON ps.ProductCategoryID = pc.ProductCategoryID;



	ProductName	Category
1	HL Road Frame - Black, 58	Components
2	HL Road Frame - Red, 58	Components
3	Sport-100 Helmet, Red	Accessories
4	Sport-100 Helmet, Black	Accessories
5	Mountain Bike Socks, M	Clothing
6	Mountain Bike Socks, L	Clothing
7	Sport-100 Helmet, Blue	Accessories
8	AWC Logo Cap	Clothing
9	Long-Sleeve Logo Jersey, S	Clothing
10	Long-Sleeve Logo Jersey, M	Clothing
11	Long-Sleeve Logo Jersey, L	Clothing
12	Long-Sleeve Logo Jersey, XL	Clothing
13	HL Road Frame - Red, 62	Components
14	HL Road Frame - Red, 44	Components
15	HL Road Frame - Red, 48	Components
16	HL Road Frame - Red, 52	Components
17	HL Road Frame - Red, 56	Components
18	LL Road Frame - Black, 58	Components

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (65) | AdventureWorks2022 | 00:00:00 | 295 rows

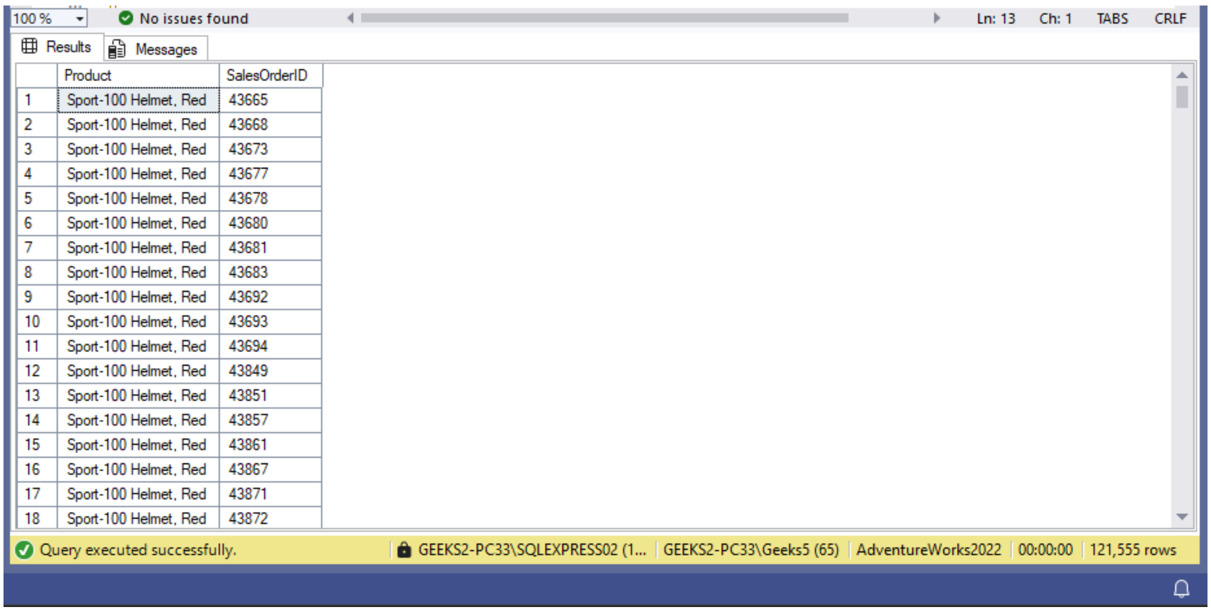
-- 3. Products with/without sales

SELECT p.Name AS Product, soh.SalesOrderID

FROM Production.Product p

LEFT JOIN Sales.SalesOrderDetail sod ON p.ProductID = sod.ProductID

LEFT JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID;



100 % No issues found Ln: 13 Ch: 1 TABS CRLF

Results Messages

	Product	SalesOrderID
1	Sport-100 Helmet, Red	43665
2	Sport-100 Helmet, Red	43668
3	Sport-100 Helmet, Red	43673
4	Sport-100 Helmet, Red	43677
5	Sport-100 Helmet, Red	43678
6	Sport-100 Helmet, Red	43680
7	Sport-100 Helmet, Red	43681
8	Sport-100 Helmet, Red	43683
9	Sport-100 Helmet, Red	43692
10	Sport-100 Helmet, Red	43693
11	Sport-100 Helmet, Red	43694
12	Sport-100 Helmet, Red	43849
13	Sport-100 Helmet, Red	43851
14	Sport-100 Helmet, Red	43857
15	Sport-100 Helmet, Red	43861
16	Sport-100 Helmet, Red	43867
17	Sport-100 Helmet, Red	43871
18	Sport-100 Helmet, Red	43872

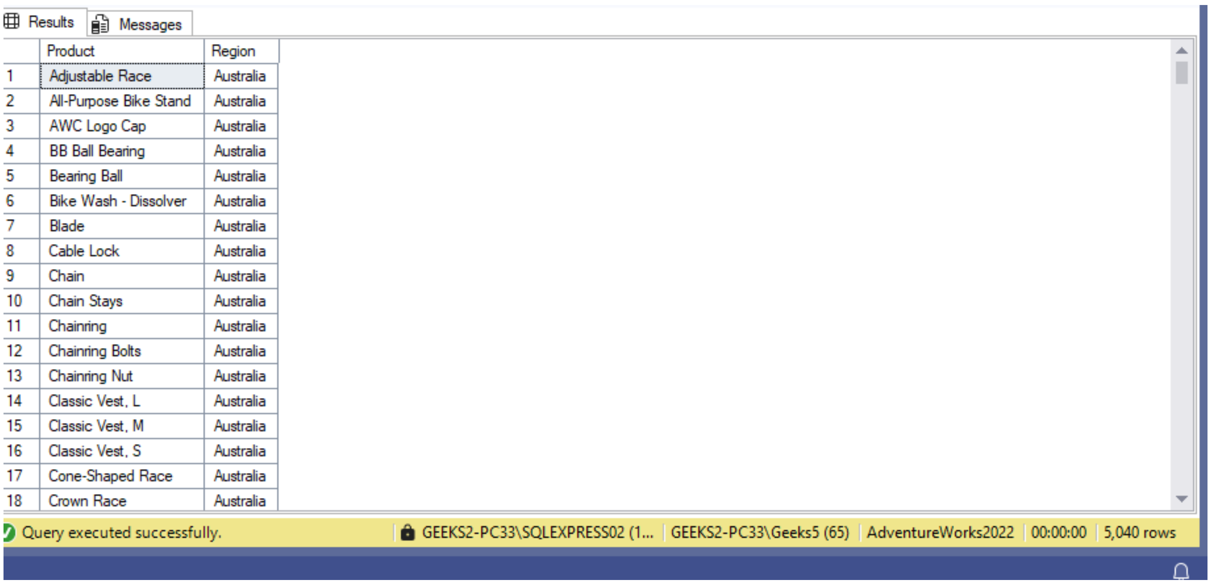
Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GEEKS2-PC33\Geeks5 (65) AdventureWorks2022 00:00:00 121,555 rows

-- 4. Product vs. Region

SELECT p.Name AS Product, sp.Name AS Region

FROM Production.Product p

CROSS JOIN Sales.SalesTerritory sp;



	Product	Region
1	Adjustable Race	Australia
2	All-Purpose Bike Stand	Australia
3	AWC Logo Cap	Australia
4	BB Ball Bearing	Australia
5	Bearing Ball	Australia
6	Bike Wash - Dissolver	Australia
7	Blade	Australia
8	Cable Lock	Australia
9	Chain	Australia
10	Chain Stays	Australia
11	Chainring	Australia
12	Chainring Bolts	Australia
13	Chainring Nut	Australia
14	Classic Vest, L	Australia
15	Classic Vest, M	Australia
16	Classic Vest, S	Australia
17	Cone-Shaped Race	Australia
18	Crown Race	Australia

```
Query executed successfully.   GEEKS2-PC33\SQLEXPRESS02 (1...   GEEKS2-PC33\Geeks5 (65)   AdventureWorks2022   00:00:00   5,040 rows
```

TASK 3

-- 1. Customers from a specific city

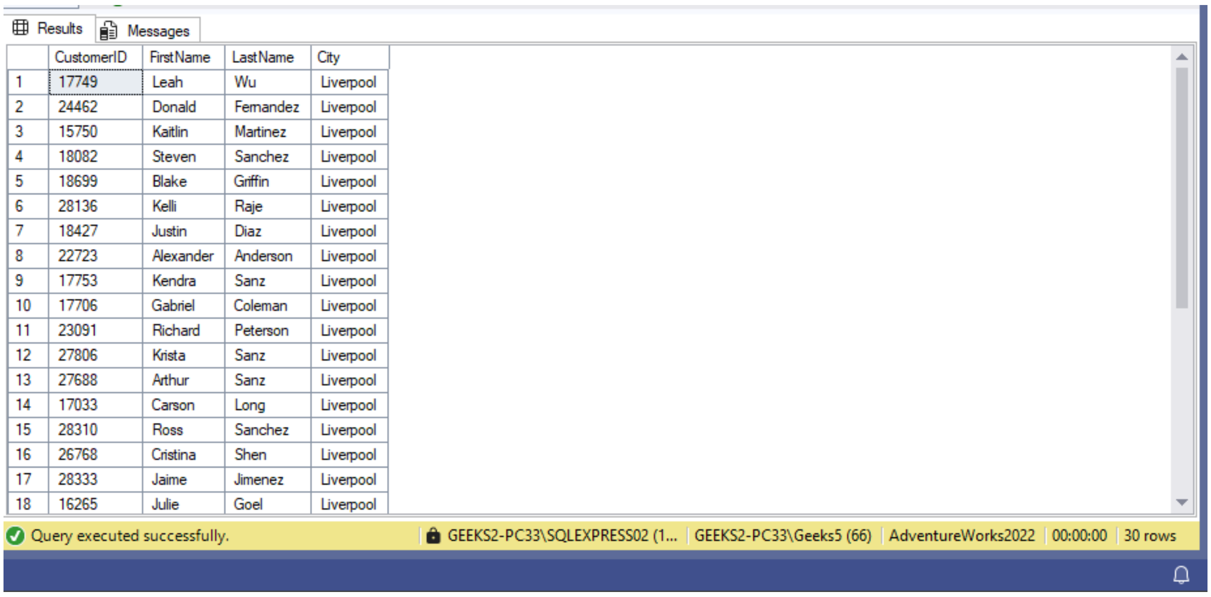
SELECT c.CustomerID, p.FirstName, p.LastName, a.City

FROM Sales.Customer c

JOIN Person.Person p ON c.PersonID = p.BusinessEntityID

JOIN Person.Address a ON c.CustomerID = a.AddressID

WHERE a.City = 'Liverpool';



	CustomerID	FirstName	LastName	City
1	17749	Leah	Wu	Liverpool
2	24462	Donald	Fernandez	Liverpool
3	15750	Kaitlin	Martinez	Liverpool
4	18082	Steven	Sanchez	Liverpool
5	18699	Blake	Griffin	Liverpool
6	28136	Kelli	Raje	Liverpool
7	18427	Justin	Diaz	Liverpool
8	22723	Alexander	Anderson	Liverpool
9	17753	Kendra	Sanz	Liverpool
10	17706	Gabriel	Coleman	Liverpool
11	23091	Richard	Peterson	Liverpool
12	27806	Krista	Sanz	Liverpool
13	27688	Arthur	Sanz	Liverpool
14	17033	Carson	Long	Liverpool
15	28310	Ross	Sanchez	Liverpool
16	26768	Cristina	Shen	Liverpool
17	28333	Jaime	Jimenez	Liverpool
18	16265	Julie	Goel	Liverpool

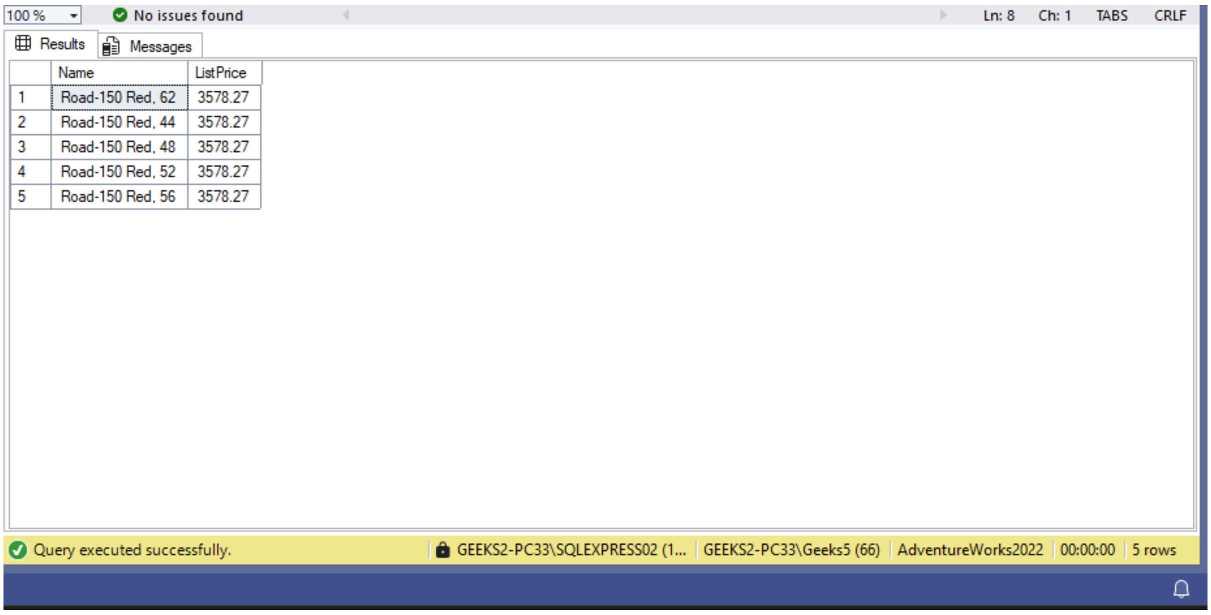
```
Query executed successfully.    GEEKS2-PC33\SQLEXPRESS02 (1...    GEEKS2-PC33\Geeks5 (66)    AdventureWorks2022    00:00:00    30 rows
```

-- 2. Top 5 products by price

SELECT TOP 5 Name, ListPrice

FROM Production.Product

ORDER BY ListPrice DESC;



100 % No issues found Ln: 8 Ch: 1 TABS CRLF

Results Messages

	Name	ListPrice
1	Road-150 Red, 62	3578.27
2	Road-150 Red, 44	3578.27
3	Road-150 Red, 48	3578.27
4	Road-150 Red, 52	3578.27
5	Road-150 Red, 56	3578.27

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (66) | AdventureWorks2022 | 00:00:00 | 5 rows

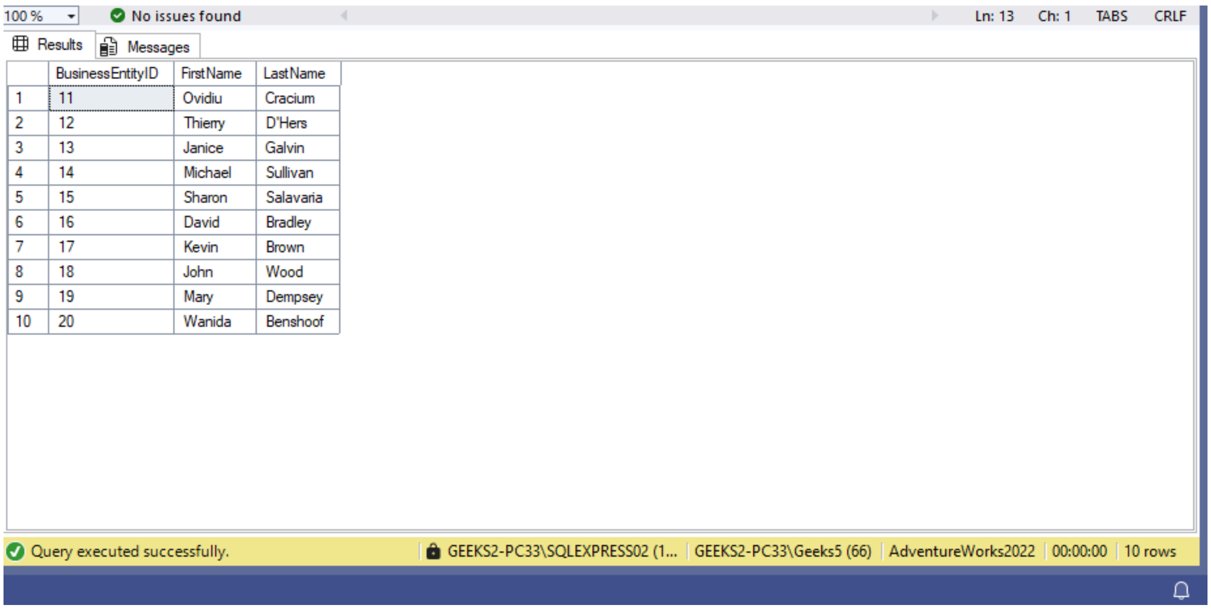
-- 3. Pagination using OFFSET-FETCH

SELECT BusinessEntityID, FirstName, LastName

FROM Person.Person

ORDER BY BusinessEntityID

OFFSET 10 ROWS FETCH NEXT 10 ROWS ONLY;



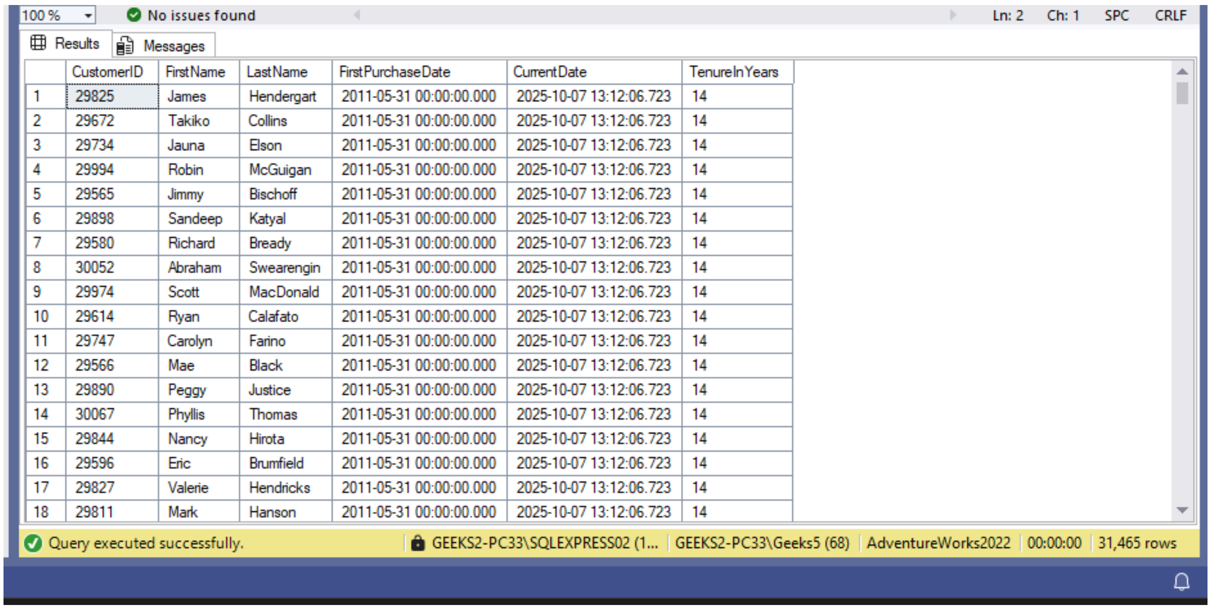
	BusinessEntityID	FirstName	LastName
1	11	Ovidiu	Cracium
2	12	Thierry	D'Hers
3	13	Janice	Galvin
4	14	Michael	Sullivan
5	15	Sharon	Salavaria
6	16	David	Bradley
7	17	Kevin	Brown
8	18	John	Wood
9	19	Mary	Dempsey
10	20	Wanida	Benshoof

```
Query executed successfully.    GEEKS2-PC33\SQLEXPRESS02 (1...  GEEKS2-PC33\Geeks5 (66)  AdventureWorks2022  00:00:00  10 rows
```

Task 4

-- 1. Retrieve current date/time and customer tenure in years

```
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName,
    soh.OrderDate AS FirstPurchaseDate,
    GETDATE() AS CurrentDate,
    DATEDIFF(YEAR, soh.OrderDate, GETDATE()) AS TenureInYears
FROM Sales.Customer c
JOIN Person.Person p ON c.PersonID = p.BusinessEntityID
JOIN Sales.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID;
```

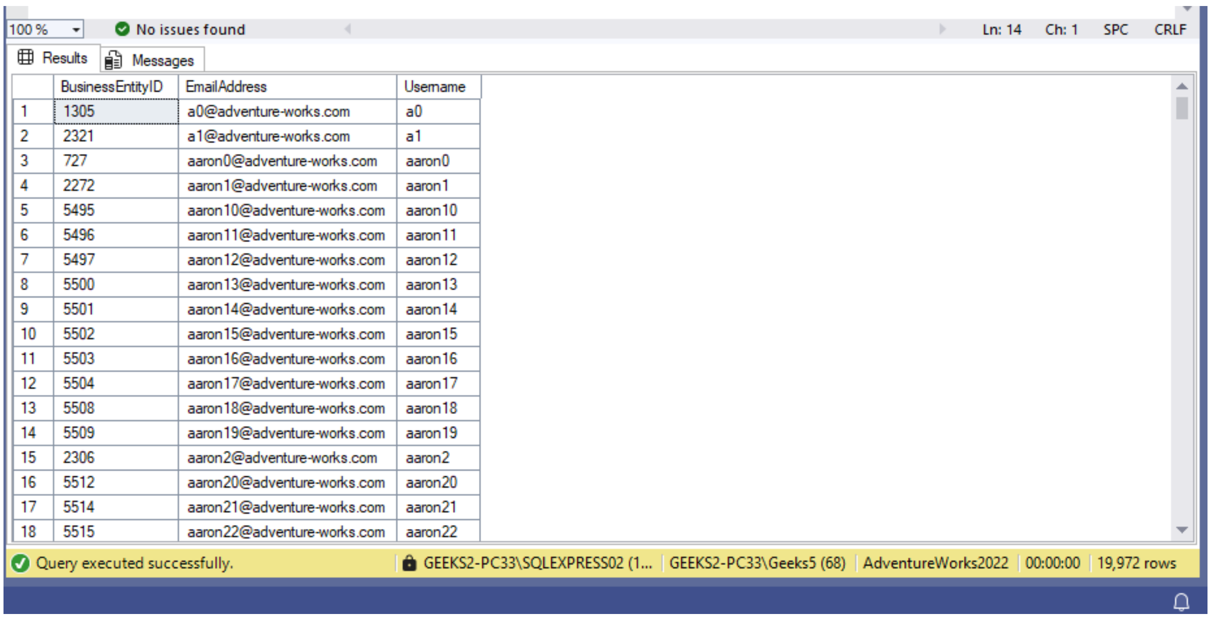


	CustomerID	FirstName	LastName	FirstPurchaseDate	CurrentDate	TenureInYears
1	29825	James	Hendergart	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
2	29672	Takiko	Collins	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
3	29734	Jauna	Elson	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
4	29994	Robin	McGuigan	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
5	29565	Jimmy	Bischoff	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
6	29898	Sandeep	Katyal	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
7	29580	Richard	Bready	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
8	30052	Abraham	Swearengin	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
9	29974	Scott	MacDonald	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
10	29614	Ryan	Calafato	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
11	29747	Carolyn	Farino	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
12	29566	Mae	Black	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
13	29890	Peggy	Justice	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
14	30067	Phyllis	Thomas	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
15	29844	Nancy	Hirota	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
16	29596	Eric	Brumfield	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
17	29827	Valerie	Hendricks	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14
18	29811	Mark	Hanson	2011-05-31 00:00:00.000	2025-10-07 13:12:06.723	14

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (68) | AdventureWorks2022 | 00:00:00 | 31,465 rows

-- 2. Extract username from email

```
SELECT
    ea.BusinessEntityID,
    ea.EmailAddress,
    LEFT(ea.EmailAddress, CHARINDEX('@', ea.EmailAddress) - 1) AS Username
FROM Person.EmailAddress ea;
```



100 % No issues found Ln: 14 Ch: 1 SPC CRLF

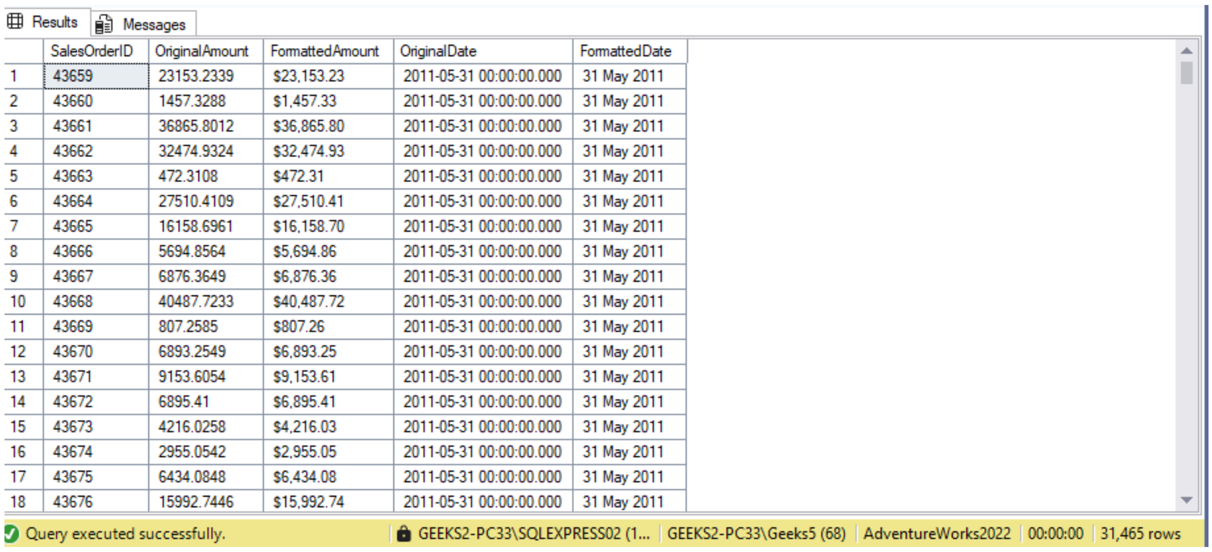
Results Messages

	BusinessEntityID	EmailAddress	Username
1	1305	a0@adventure-works.com	a0
2	2321	a1@adventure-works.com	a1
3	727	aaron0@adventure-works.com	aaron0
4	2272	aaron1@adventure-works.com	aaron1
5	5495	aaron10@adventure-works.com	aaron10
6	5496	aaron11@adventure-works.com	aaron11
7	5497	aaron12@adventure-works.com	aaron12
8	5500	aaron13@adventure-works.com	aaron13
9	5501	aaron14@adventure-works.com	aaron14
10	5502	aaron15@adventure-works.com	aaron15
11	5503	aaron16@adventure-works.com	aaron16
12	5504	aaron17@adventure-works.com	aaron17
13	5508	aaron18@adventure-works.com	aaron18
14	5509	aaron19@adventure-works.com	aaron19
15	2306	aaron2@adventure-works.com	aaron2
16	5512	aaron20@adventure-works.com	aaron20
17	5514	aaron21@adventure-works.com	aaron21
18	5515	aaron22@adventure-works.com	aaron22

Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GEEKS2-PC33\Geeks5 (68) AdventureWorks2022 00:00:00 19,972 rows

-- 3. Format numbers and dates

```
SELECT
    soh.SalesOrderID,
    soh.TotalDue AS OriginalAmount,
    FORMAT(soh.TotalDue, 'C', 'en-US') AS FormattedAmount,
    soh.OrderDate AS OriginalDate,
    FORMAT(soh.OrderDate, 'dd MMM yyyy') AS FormattedDate
FROM Sales.SalesOrderHeader soh;
```



Results | Messages

	SalesOrderID	OriginalAmount	FormattedAmount	OriginalDate	FormattedDate
1	43659	23153.2339	$23,153.23	2011-05-31 00:00:00.000	31 May 2011
2	43660	1457.3288	$1,457.33	2011-05-31 00:00:00.000	31 May 2011
3	43661	36865.8012	$36,865.80	2011-05-31 00:00:00.000	31 May 2011
4	43662	32474.9324	$32,474.93	2011-05-31 00:00:00.000	31 May 2011
5	43663	472.3108	$472.31	2011-05-31 00:00:00.000	31 May 2011
6	43664	27510.4109	$27,510.41	2011-05-31 00:00:00.000	31 May 2011
7	43665	16158.6961	$16,158.70	2011-05-31 00:00:00.000	31 May 2011
8	43666	5694.8564	$5,694.86	2011-05-31 00:00:00.000	31 May 2011
9	43667	6876.3649	$6,876.36	2011-05-31 00:00:00.000	31 May 2011
10	43668	40487.7233	$40,487.72	2011-05-31 00:00:00.000	31 May 2011
11	43669	807.2585	$807.26	2011-05-31 00:00:00.000	31 May 2011
12	43670	6893.2549	$6,893.25	2011-05-31 00:00:00.000	31 May 2011
13	43671	9153.6054	$9,153.61	2011-05-31 00:00:00.000	31 May 2011
14	43672	6895.41	$6,895.41	2011-05-31 00:00:00.000	31 May 2011
15	43673	4216.0258	$4,216.03	2011-05-31 00:00:00.000	31 May 2011
16	43674	2955.0542	$2,955.05	2011-05-31 00:00:00.000	31 May 2011
17	43675	6434.0848	$6,434.08	2011-05-31 00:00:00.000	31 May 2011
18	43676	15992.7446	$15,992.74	2011-05-31 00:00:00.000	31 May 2011

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (68) | AdventureWorks2022 | 00:00:00 | 31,465 rows

Task 5

INSERT INTO HumanResources.Employee (BusinessEntityID, NationalIDNumber, LoginID, JobTitle, BirthDate, MaritalStatus, Gender, HireDate)

VALUES (

291,

'995441256',

'adventure-works\dylanmaharaj',

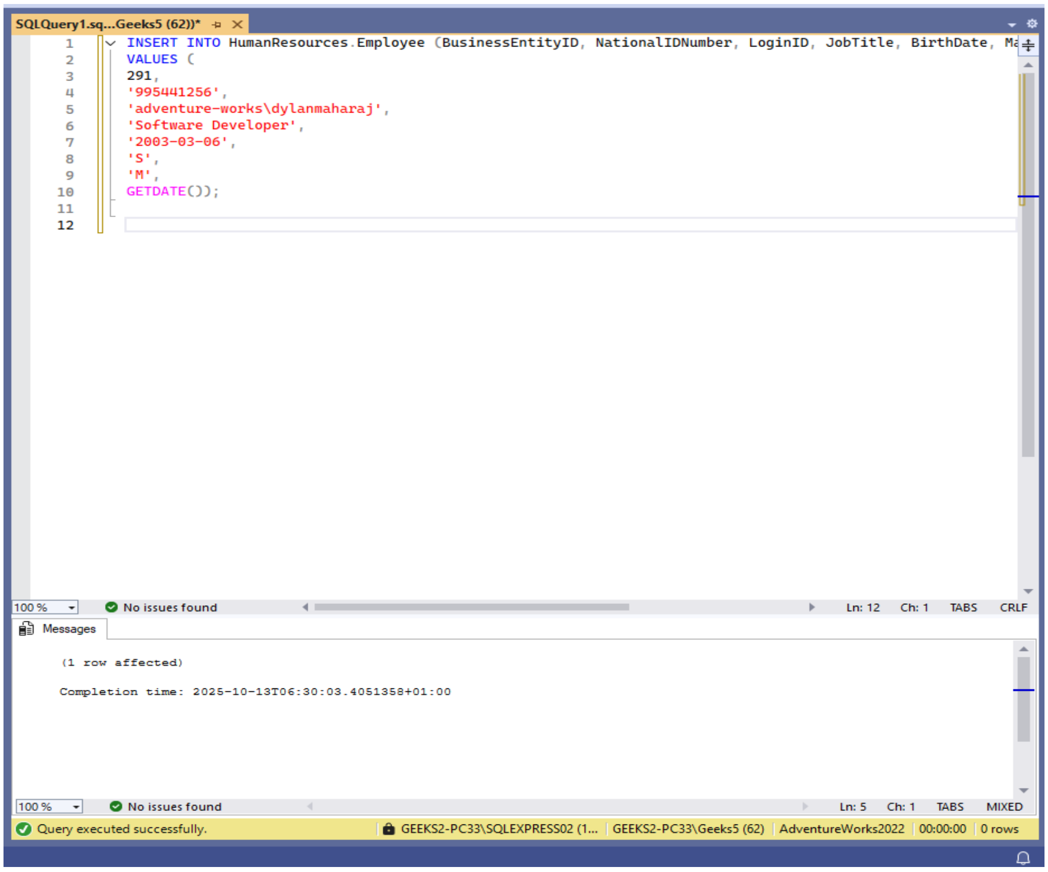
'Software Developer',

'2003-03-06',

'S',

'M',

GETDATE());



```
INSERT INTO HumanResources.Employee (BusinessEntityID, NationalIDNumber, LoginID, JobTitle, BirthDate, Ma
VALUES (
291,
'995441256',
'adventure-works\dylanmaharaj',
'Software Developer',
'2003-03-06',
'S',
'M',
GETDATE());
```

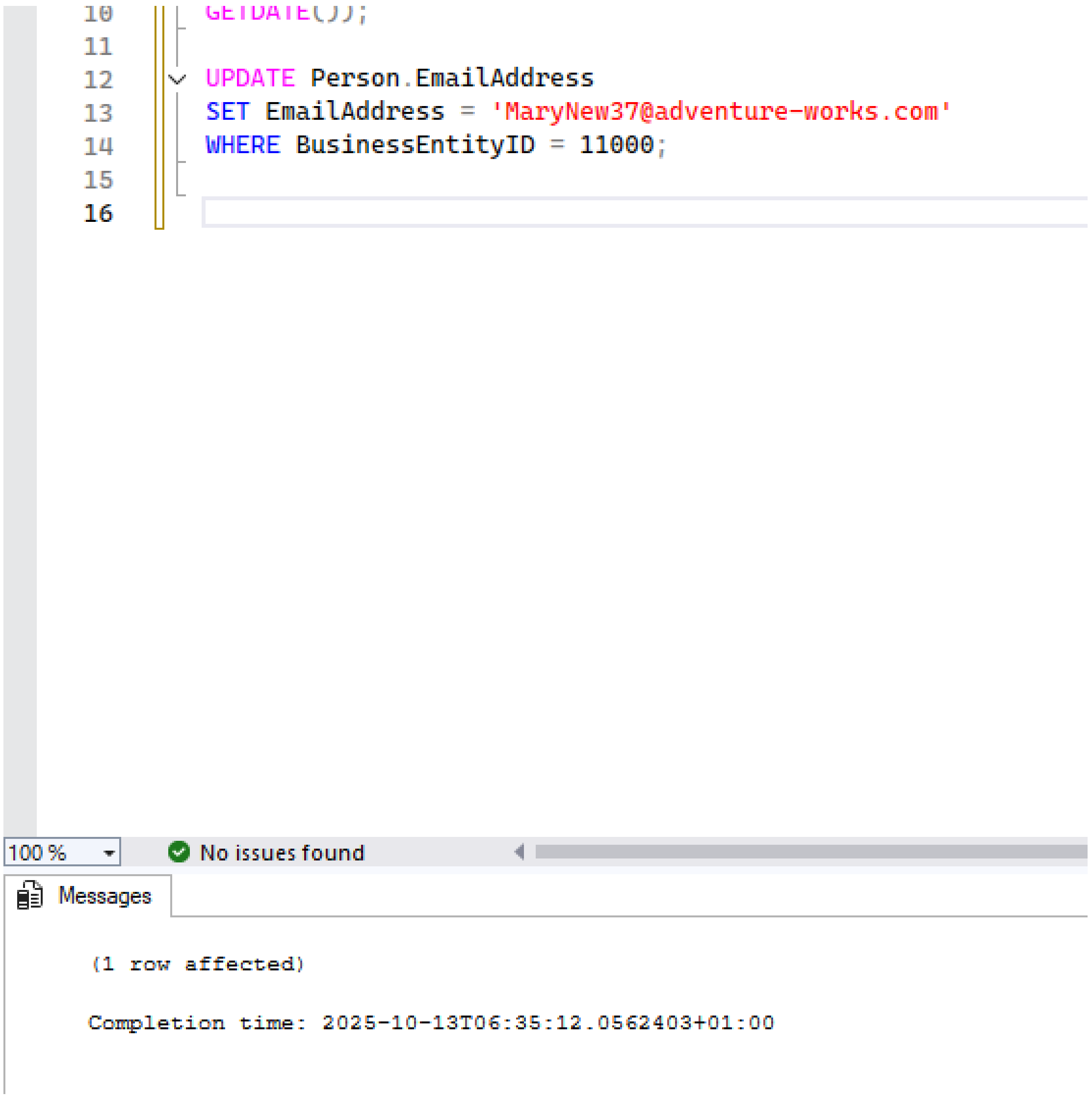
```
(1 row affected)

Completion time: 2025-10-13T06:30:03.4051358+01:00
```

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (62) | AdventureWorks2022 | 00:00:00 | 0 rows

UPDATE Person.EmailAddress

SET EmailAddress = 'MaryNew37@adventure-works.com'

WHERE BusinessEntityID = 11000;



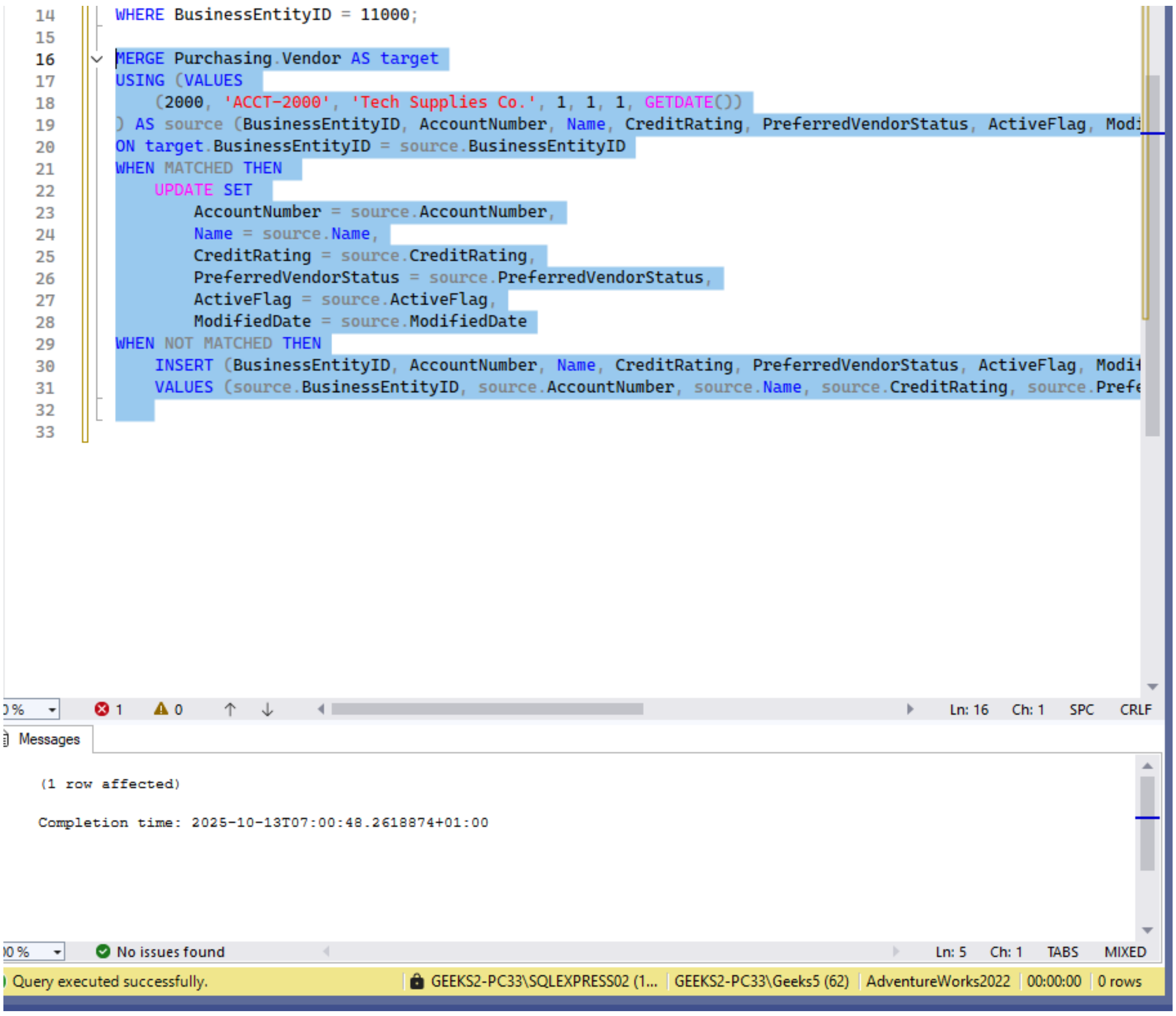
```
GETDATE());

UPDATE Person.EmailAddress
SET EmailAddress = 'MaryNew37@adventure-works.com'
WHERE BusinessEntityID = 11000;

```

100 % No issues found

Messages

```
(1 row affected)

Completion time: 2025-10-13T06:35:12.0562403+01:00
```

```
MERGE Purchasing.Vendor AS target

USING (VALUES

    (2000, 'ACCT-2000', 'Tech Supplies Co.', 1, 1, 1, GETDATE())

) AS source (BusinessEntityID, AccountNumber, Name, CreditRating, PreferredVendorStatus, ActiveFlag,
ModifiedDate)

ON target.BusinessEntityID = source.BusinessEntityID

WHEN MATCHED THEN

    UPDATE SET

        AccountNumber = source.AccountNumber,

        Name = source.Name,

        CreditRating = source.CreditRating,

        PreferredVendorStatus = source.PreferredVendorStatus,

        ActiveFlag = source.ActiveFlag,

        ModifiedDate = source.ModifiedDate

WHEN NOT MATCHED THEN

    INSERT (BusinessEntityID, AccountNumber, Name, CreditRating, PreferredVendorStatus, ActiveFlag,
ModifiedDate)

    VALUES (source.BusinessEntityID, source.AccountNumber, source.Name, source.CreditRating,
source.PreferredVendorStatus, source.ActiveFlag, source.ModifiedDate);
```



```sql
WHERE BusinessEntityID = 11000;

MERGE Purchasing.Vendor AS target
USING (VALUES
    (2000, 'ACCT-2000', 'Tech Supplies Co.', 1, 1, 1, GETDATE())
) AS source (BusinessEntityID, AccountNumber, Name, CreditRating, PreferredVendorStatus, ActiveFlag, Modi
ON target.BusinessEntityID = source.BusinessEntityID
WHEN MATCHED THEN
    UPDATE SET
        AccountNumber = source.AccountNumber,
        Name = source.Name,
        CreditRating = source.CreditRating,
        PreferredVendorStatus = source.PreferredVendorStatus,
        ActiveFlag = source.ActiveFlag,
        ModifiedDate = source.ModifiedDate
WHEN NOT MATCHED THEN
    INSERT (BusinessEntityID, AccountNumber, Name, CreditRating, PreferredVendorStatus, ActiveFlag, Modif
    VALUES (source.BusinessEntityID, source.AccountNumber, source.Name, source.CreditRating, source.Prefe
```

```
(1 row affected)

Completion time: 2025-10-13T07:00:48.2618874+01:00
```

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (62) | AdventureWorks2022 | 00:00:00 | 0 rows

Task 6

SELECT

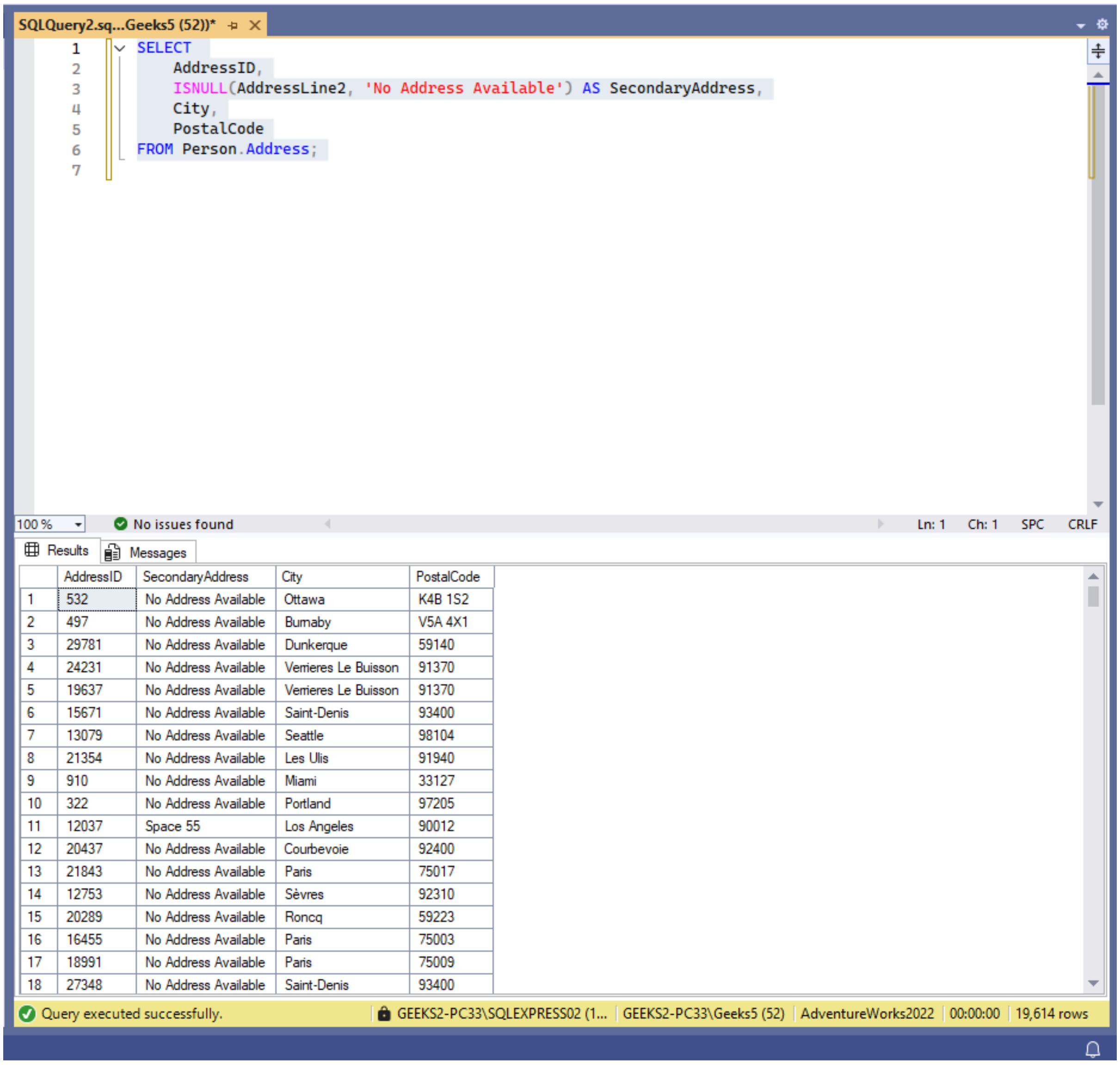
AddressID,

ISNULL(AddressLine2, 'No Address Available') AS SecondaryAddress,

City,

PostalCode

FROM Person.Address;



```sql
SELECT
    AddressID,
    ISNULL(AddressLine2, 'No Address Available') AS SecondaryAddress,
    City,
    PostalCode
FROM Person.Address;
```

	AddressID	SecondaryAddress	City	PostalCode
1	532	No Address Available	Ottawa	K4B 1S2
2	497	No Address Available	Burnaby	V5A 4X1
3	29781	No Address Available	Dunkerque	59140
4	24231	No Address Available	Verrieres Le Buisson	91370
5	19637	No Address Available	Verrieres Le Buisson	91370
6	15671	No Address Available	Saint-Denis	93400
7	13079	No Address Available	Seattle	98104
8	21354	No Address Available	Les Ulis	91940
9	910	No Address Available	Miami	33127
10	322	No Address Available	Portland	97205
11	12037	Space 55	Los Angeles	90012
12	20437	No Address Available	Courbevoie	92400
13	21843	No Address Available	Paris	75017
14	12753	No Address Available	Sèvres	92310
15	20289	No Address Available	Roncq	59223
16	16455	No Address Available	Paris	75003
17	18991	No Address Available	Paris	75009
18	27348	No Address Available	Saint-Denis	93400

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (52) | AdventureWorks2022 | 00:00:00 | 19,614 rows

SELECT

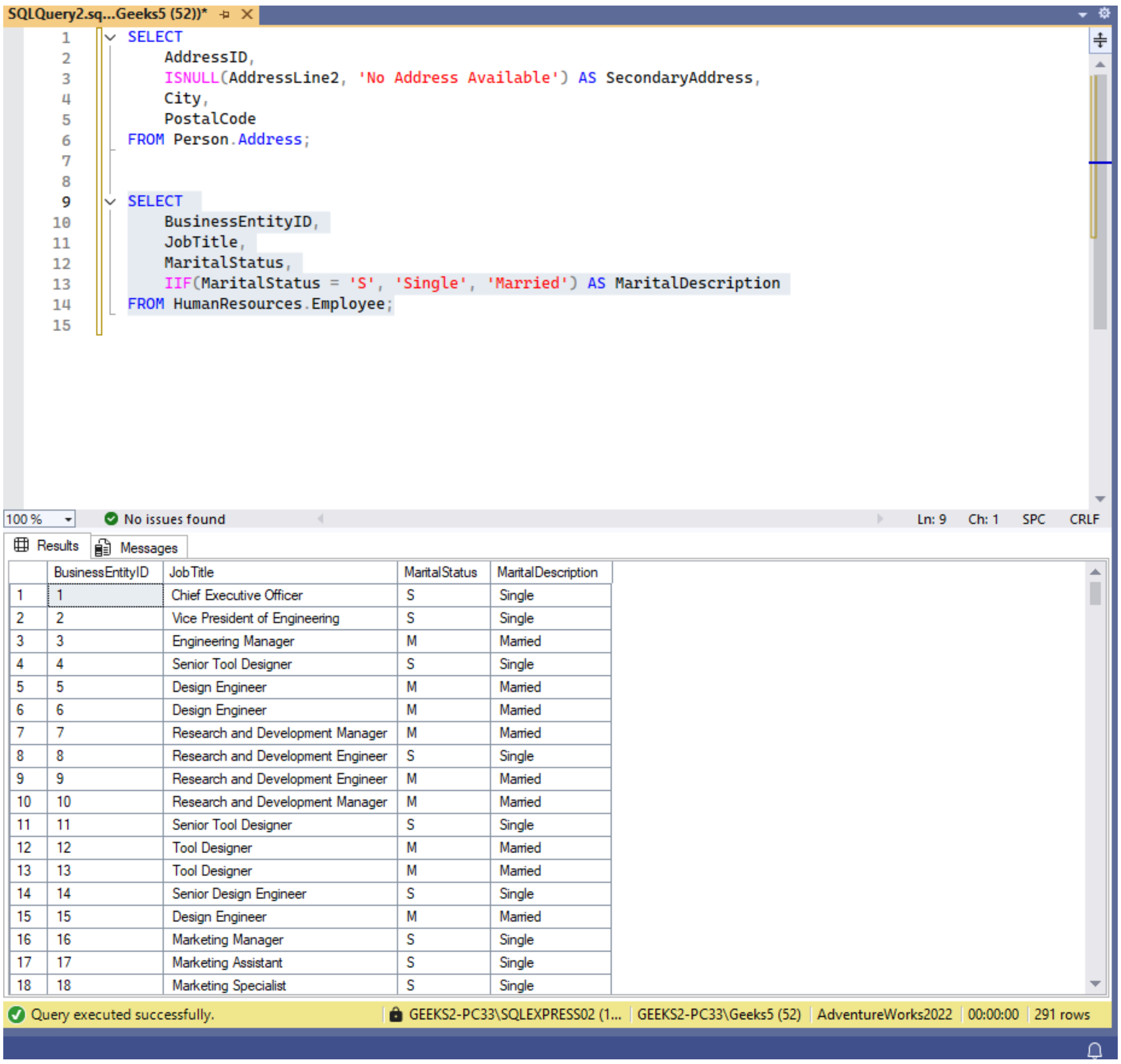
BusinessEntityID,

JobTitle,

MaritalStatus,

IIF(MaritalStatus = 'S', 'Single', 'Married') AS MaritalDescription

FROM HumanResources.Employee;



SQLQuery2.sq...Geeks5 (52))*

```
SELECT
    AddressID,
    ISNULL(AddressLine2, 'No Address Available') AS SecondaryAddress,
    City,
    PostalCode
FROM Person.Address;


SELECT
    BusinessEntityID,
    JobTitle,
    MaritalStatus,
    IIF(MaritalStatus = 'S', 'Single', 'Married') AS MaritalDescription
FROM HumanResources.Employee;
```

100 % No issues found Ln: 9 Ch: 1 SPC CRLF

Results Messages

	BusinessEntityID	JobTitle	MaritalStatus	MaritalDescription
1	1	Chief Executive Officer	S	Single
2	2	Vice President of Engineering	S	Single
3	3	Engineering Manager	M	Married
4	4	Senior Tool Designer	S	Single
5	5	Design Engineer	M	Married
6	6	Design Engineer	M	Married
7	7	Research and Development Manager	M	Married
8	8	Research and Development Engineer	S	Single
9	9	Research and Development Engineer	M	Married
10	10	Research and Development Manager	M	Married
11	11	Senior Tool Designer	S	Single
12	12	Tool Designer	M	Married
13	13	Tool Designer	M	Married
14	14	Senior Design Engineer	S	Single
15	15	Design Engineer	M	Married
16	16	Marketing Manager	S	Single
17	17	Marketing Assistant	S	Single
18	18	Marketing Specialist	S	Single

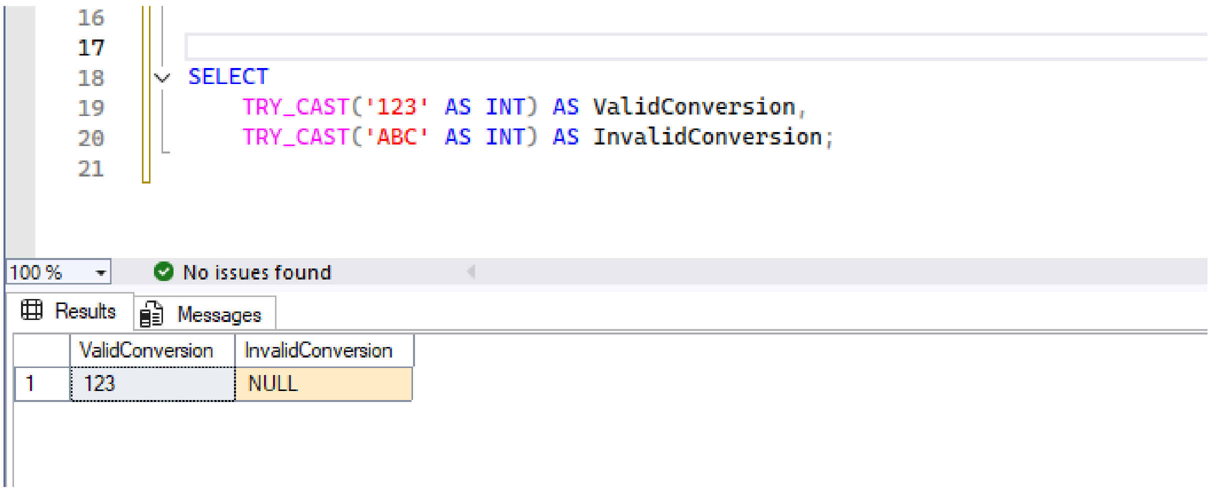
Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GEEKS2-PC33\Geeks5 (52) AdventureWorks2022 00:00:00 291 rows

-- Example of converting text data into integers safely

SELECT

TRY_CAST('123' AS INT) AS ValidConversion,

TRY_CAST('ABC' AS INT) AS InvalidConversion;



```
SELECT
    TRY_CAST('123' AS INT) AS ValidConversion,
    TRY_CAST('ABC' AS INT) AS InvalidConversion;
```

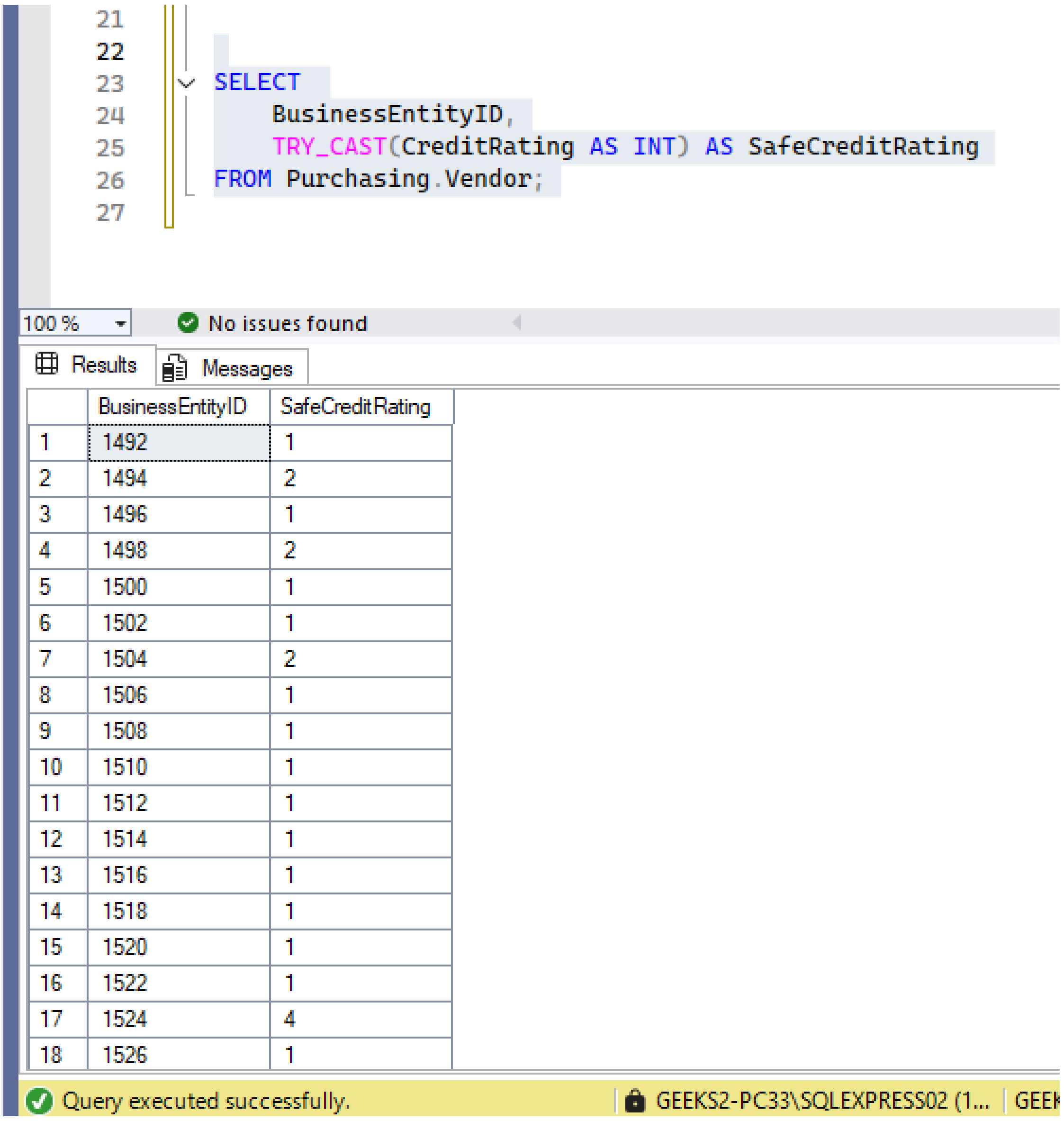
	ValidConversion	InvalidConversion
1	123	NULL

SELECT

BusinessEntityID,

TRY_CAST(CreditRating AS INT) AS SafeCreditRating

FROM Purchasing.Vendor;



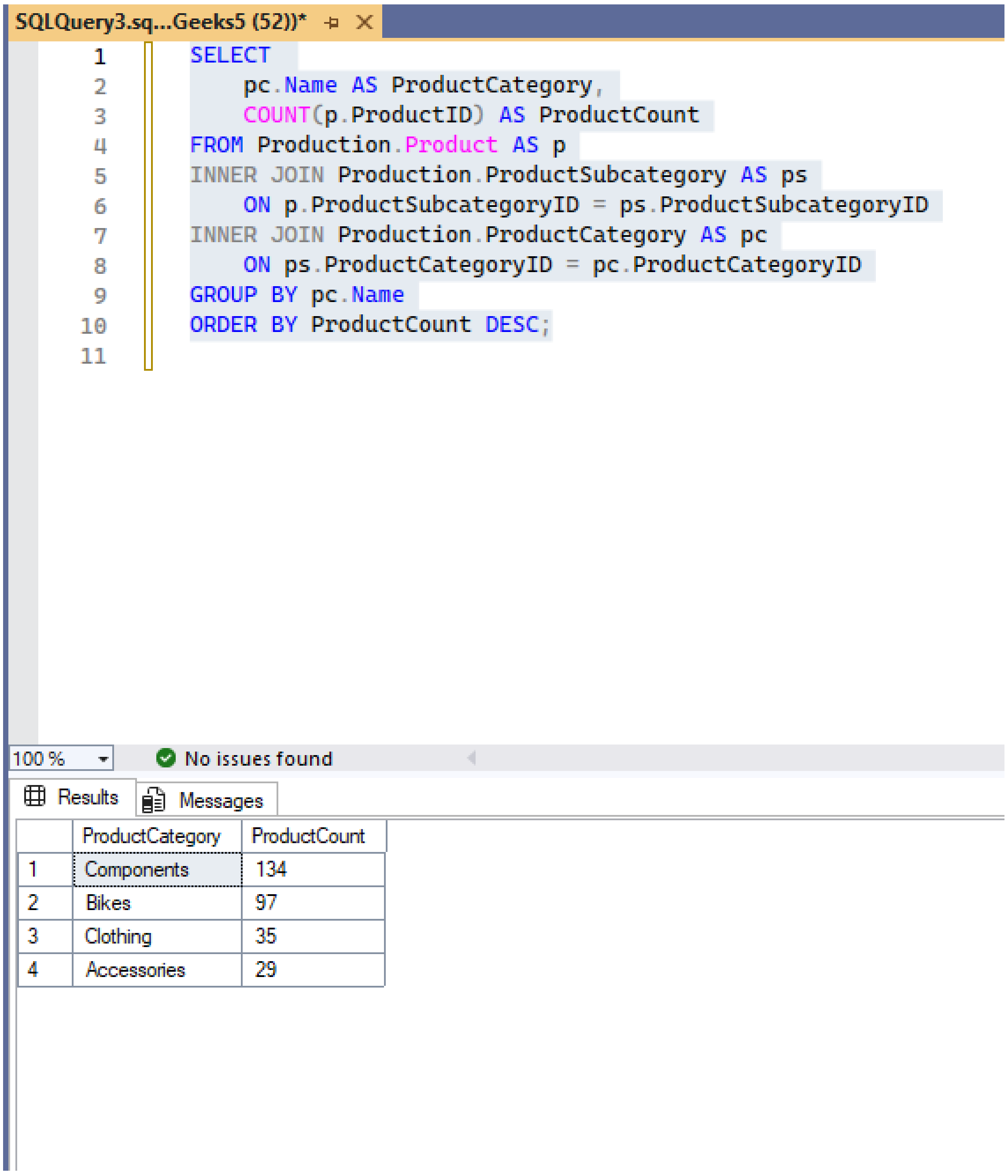
```sql

SELECT
    BusinessEntityID,
    TRY_CAST(CreditRating AS INT) AS SafeCreditRating
FROM Purchasing.Vendor;

```

	BusinessEntityID	SafeCreditRating
1	1492	1
2	1494	2
3	1496	1
4	1498	2
5	1500	1
6	1502	1
7	1504	2
8	1506	1
9	1508	1
10	1510	1
11	1512	1
12	1514	1
13	1516	1
14	1518	1
15	1520	1
16	1522	1
17	1524	4
18	1526	1

Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GEEK

Task 7

```
SELECT
    pc.Name AS ProductCategory,
    COUNT(p.ProductID) AS ProductCount
FROM Production.Product AS p
INNER JOIN Production.ProductSubcategory AS ps
    ON p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN Production.ProductCategory AS pc
    ON ps.ProductCategoryID = pc.ProductCategoryID
GROUP BY pc.Name
ORDER BY ProductCount DESC;
```


SQLQuery3.sq...Geeks5 (52))*

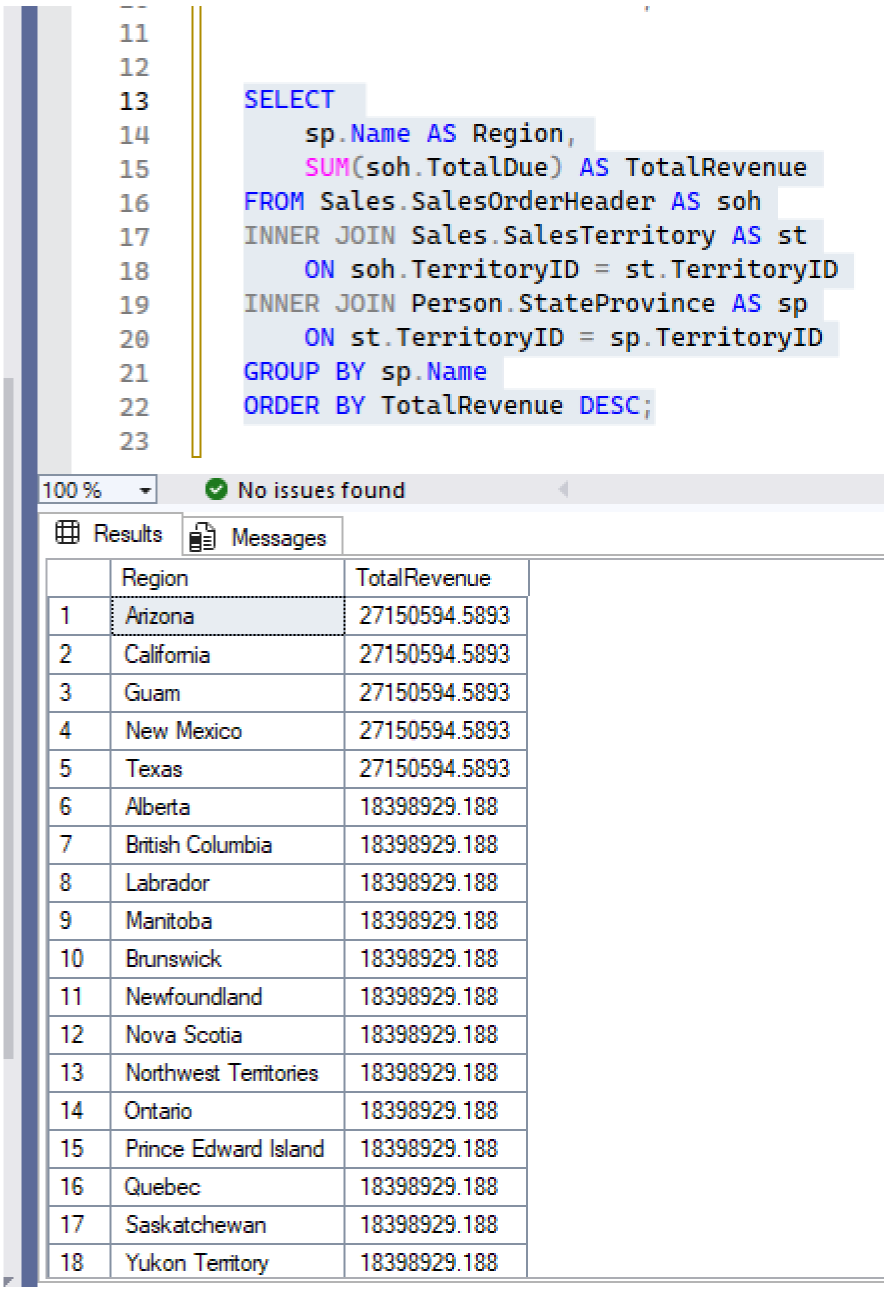
```sql
SELECT
    pc.Name AS ProductCategory,
    COUNT(p.ProductID) AS ProductCount
FROM Production.Product AS p
INNER JOIN Production.ProductSubcategory AS ps
    ON p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN Production.ProductCategory AS pc
    ON ps.ProductCategoryID = pc.ProductCategoryID
GROUP BY pc.Name
ORDER BY ProductCount DESC;
```

100 %   No issues found

Results   Messages

	ProductCategory	ProductCount
1	Components	134
2	Bikes	97
3	Clothing	35
4	Accessories	29

```
SELECT
    sp.Name AS Region,
    SUM(soh.TotalDue) AS TotalRevenue
FROM Sales.SalesOrderHeader AS soh
INNER JOIN Sales.SalesTerritory AS st
    ON soh.TerritoryID = st.TerritoryID
INNER JOIN Person.StateProvince AS sp
    ON st.TerritoryID = sp.TerritoryID
GROUP BY sp.Name
ORDER BY TotalRevenue DESC;
```


```sql
SELECT
    sp.Name AS Region,
    SUM(soh.TotalDue) AS TotalRevenue
FROM Sales.SalesOrderHeader AS soh
INNER JOIN Sales.SalesTerritory AS st
    ON soh.TerritoryID = st.TerritoryID
INNER JOIN Person.StateProvince AS sp
    ON st.TerritoryID = sp.TerritoryID
GROUP BY sp.Name
ORDER BY TotalRevenue DESC;
```

	Region	TotalRevenue
1	Arizona	27150594.5893
2	California	27150594.5893
3	Guam	27150594.5893
4	New Mexico	27150594.5893
5	Texas	27150594.5893
6	Alberta	18398929.188
7	British Columbia	18398929.188
8	Labrador	18398929.188
9	Manitoba	18398929.188
10	Brunswick	18398929.188
11	Newfoundland	18398929.188
12	Nova Scotia	18398929.188
13	Northwest Territories	18398929.188
14	Ontario	18398929.188
15	Prince Edward Island	18398929.188
16	Quebec	18398929.188
17	Saskatchewan	18398929.188
18	Yukon Territory	18398929.188

SELECT

pc.Name AS ProductCategory,

COUNT(p.ProductID) AS ProductCount

FROM Production.Product AS p

INNER JOIN Production.ProductSubcategory AS ps

ON p.ProductSubcategoryID = ps.ProductSubcategoryID

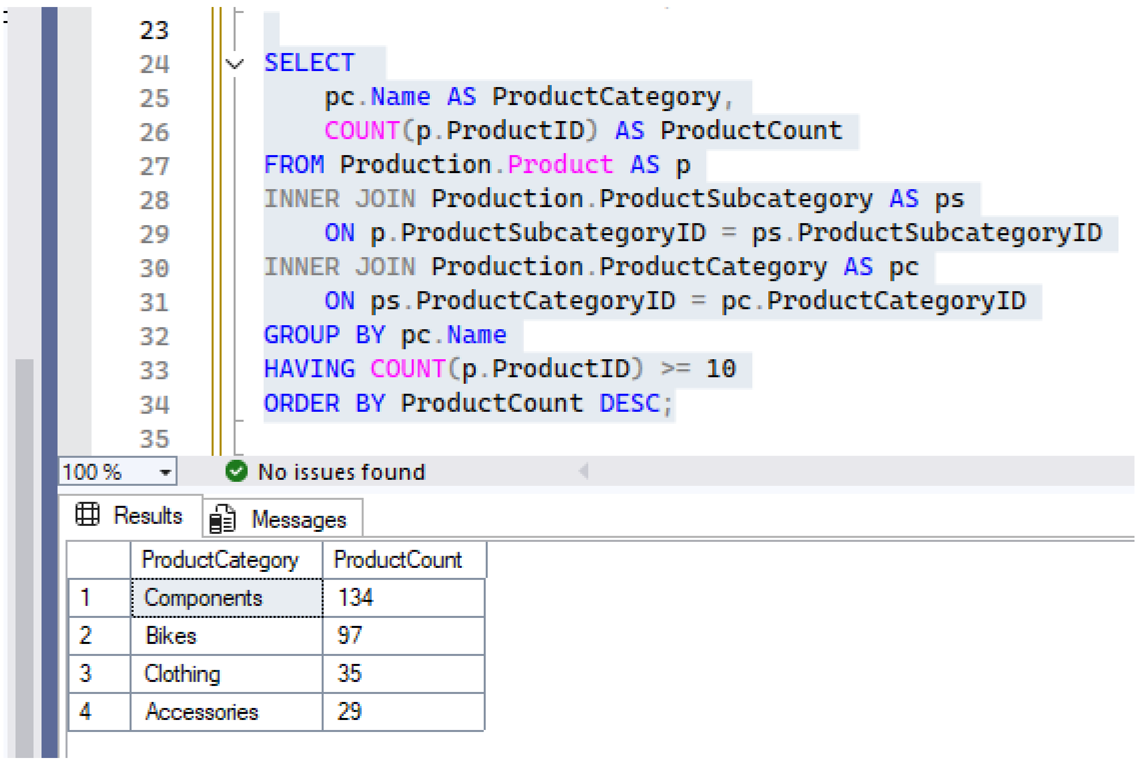
INNER JOIN Production.ProductCategory AS pc

ON ps.ProductCategoryID = pc.ProductCategoryID

GROUP BY pc.Name

HAVING COUNT(p.ProductID) >= 10

ORDER BY ProductCount DESC;

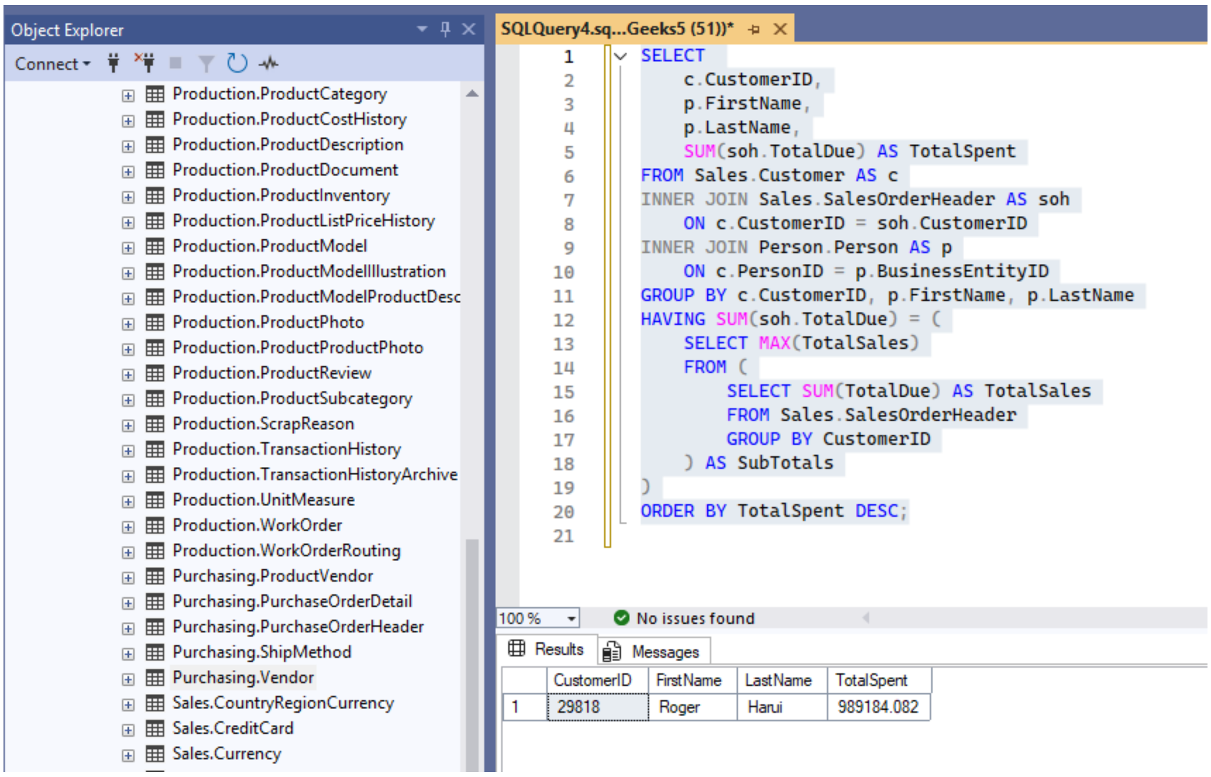


```
SELECT
    pc.Name AS ProductCategory,
    COUNT(p.ProductID) AS ProductCount
FROM Production.Product AS p
INNER JOIN Production.ProductSubcategory AS ps
    ON p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN Production.ProductCategory AS pc
    ON ps.ProductCategoryID = pc.ProductCategoryID
GROUP BY pc.Name
HAVING COUNT(p.ProductID) >= 10
ORDER BY ProductCount DESC;
```

	ProductCategory	ProductCount
1	Components	134
2	Bikes	97
3	Clothing	35
4	Accessories	29

Task 8

```
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName,
    SUM(soh.TotalDue) AS TotalSpent
FROM Sales.Customer AS c
INNER JOIN Sales.SalesOrderHeader AS soh
    ON c.CustomerID = soh.CustomerID
INNER JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID
GROUP BY c.CustomerID, p.FirstName, p.LastName
HAVING SUM(soh.TotalDue) = (
    SELECT MAX(TotalSales)
    FROM (
        SELECT SUM(TotalDue) AS TotalSales
        FROM Sales.SalesOrderHeader
        GROUP BY CustomerID
    ) AS SubTotals
)
ORDER BY TotalSpent DESC;
```

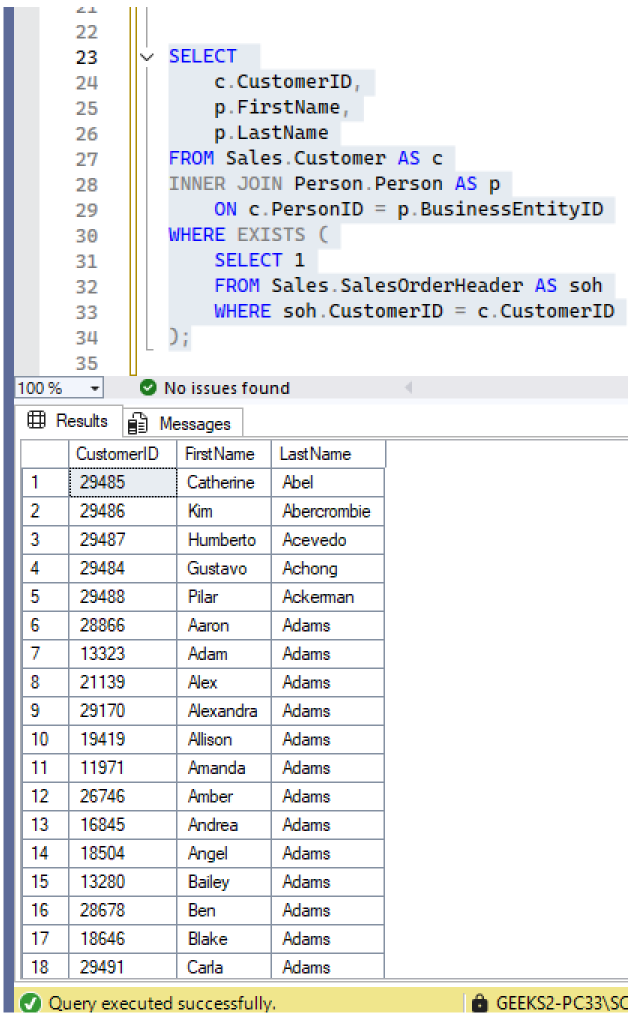


Object Explorer
Connect
Production.ProductCategory
Production.ProductCostHistory
Production.ProductDescription
Production.ProductDocument
Production.ProductInventory
Production.ProductListPriceHistory
Production.ProductModel
Production.ProductModelIllustration
Production.ProductModelProductDesc
Production.ProductPhoto
Production.ProductProductPhoto
Production.ProductReview
Production.ProductSubcategory
Production.ScrapReason
Production.TransactionHistory
Production.TransactionHistoryArchive
Production.UnitMeasure
Production.WorkOrder
Production.WorkOrderRouting
Purchasing.ProductVendor
Purchasing.PurchaseOrderDetail
Purchasing.PurchaseOrderHeader
Purchasing.ShipMethod
Purchasing.Vendor
Sales.CountryRegionCurrency
Sales.CreditCard
Sales.Currency
SQLQuery4.sq...Geeks5 (51))*
100 %
No issues found
Results
Messages

```sql
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName,
    SUM(soh.TotalDue) AS TotalSpent
FROM Sales.Customer AS c
INNER JOIN Sales.SalesOrderHeader AS soh
    ON c.CustomerID = soh.CustomerID
INNER JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID
GROUP BY c.CustomerID, p.FirstName, p.LastName
HAVING SUM(soh.TotalDue) = (
    SELECT MAX(TotalSales)
    FROM (
        SELECT SUM(TotalDue) AS TotalSales
        FROM Sales.SalesOrderHeader
        GROUP BY CustomerID
    ) AS SubTotals
)
ORDER BY TotalSpent DESC;
```

	CustomerID	FirstName	LastName	TotalSpent
1	29818	Roger	Harui	989184.082

```
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName
FROM Sales.Customer AS c
INNER JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID
WHERE EXISTS (
    SELECT 1
    FROM Sales.SalesOrderHeader AS soh
    WHERE soh.CustomerID = c.CustomerID
);
```



```
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName
FROM Sales.Customer AS c
INNER JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID
WHERE EXISTS (
    SELECT 1
    FROM Sales.SalesOrderHeader AS soh
    WHERE soh.CustomerID = c.CustomerID
);
```

100 % No issues found

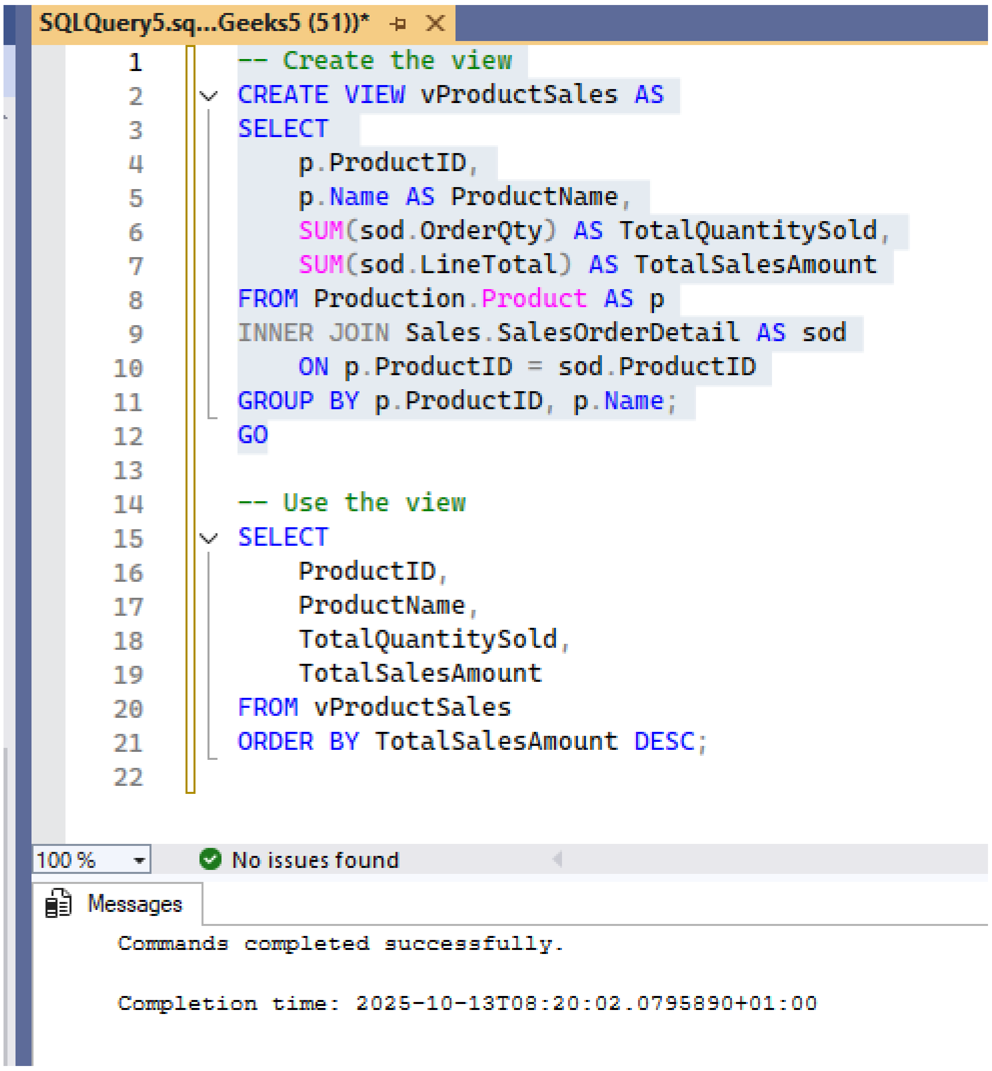
Results Messages

	CustomerID	FirstName	LastName
1	29485	Catherine	Abel
2	29486	Kim	Abercrombie
3	29487	Humberto	Acevedo
4	29484	Gustavo	Achong
5	29488	Pilar	Ackerman
6	28866	Aaron	Adams
7	13323	Adam	Adams
8	21139	Alex	Adams
9	29170	Alexandra	Adams
10	19419	Allison	Adams
11	11971	Amanda	Adams
12	26746	Amber	Adams
13	16845	Andrea	Adams
14	18504	Angel	Adams
15	13280	Bailey	Adams
16	28678	Ben	Adams
17	18646	Blake	Adams
18	29491	Carla	Adams

Query executed successfully. GEEKS2-PC33\SC

Task 9

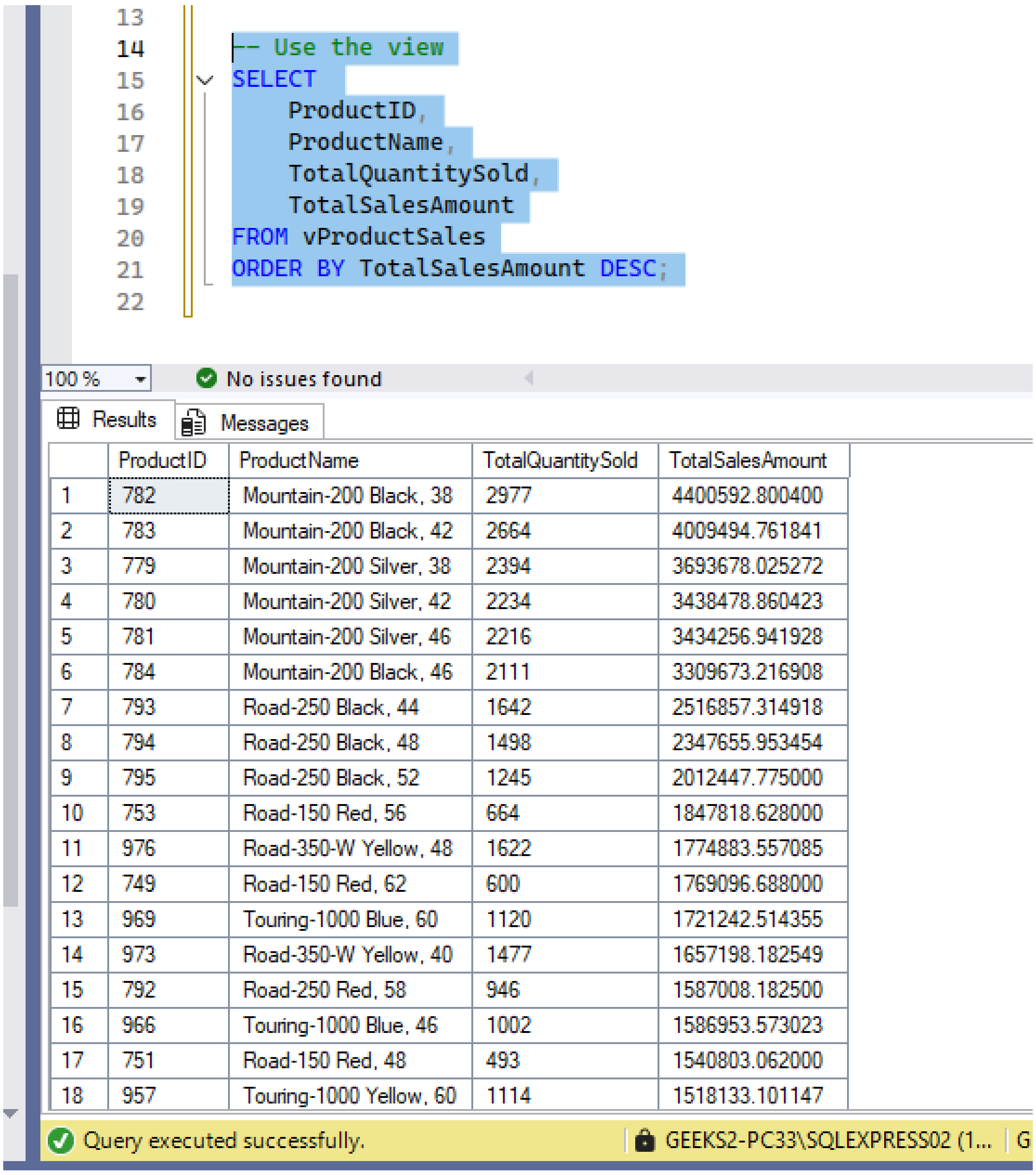
```
CREATE VIEW vProductSales AS

SELECT

    p.ProductID,

    p.Name AS ProductName,

    SUM(sod.OrderQty) AS TotalQuantitySold,

    SUM(sod.LineTotal) AS TotalSalesAmount

FROM Production.Product AS p

INNER JOIN Sales.SalesOrderDetail AS sod

    ON p.ProductID = sod.ProductID

GROUP BY p.ProductID, p.Name;

GO
```



```
-- Create the view
CREATE VIEW vProductSales AS
SELECT
    p.ProductID,
    p.Name AS ProductName,
    SUM(sod.OrderQty) AS TotalQuantitySold,
    SUM(sod.LineTotal) AS TotalSalesAmount
FROM Production.Product AS p
INNER JOIN Sales.SalesOrderDetail AS sod
    ON p.ProductID = sod.ProductID
GROUP BY p.ProductID, p.Name;
GO

-- Use the view
SELECT
    ProductID,
    ProductName,
    TotalQuantitySold,
    TotalSalesAmount
FROM vProductSales
ORDER BY TotalSalesAmount DESC;
```

```
Commands completed successfully.

Completion time: 2025-10-13T08:20:02.0795890+01:00
```

```
-- Use the view
SELECT
    ProductID,
    ProductName,
    TotalQuantitySold,
    TotalSalesAmount
FROM vProductSales
ORDER BY TotalSalesAmount DESC;
```



```
-- Use the view
SELECT
    ProductID,
    ProductName,
    TotalQuantitySold,
    TotalSalesAmount
FROM vProductSales
ORDER BY TotalSalesAmount DESC;
```

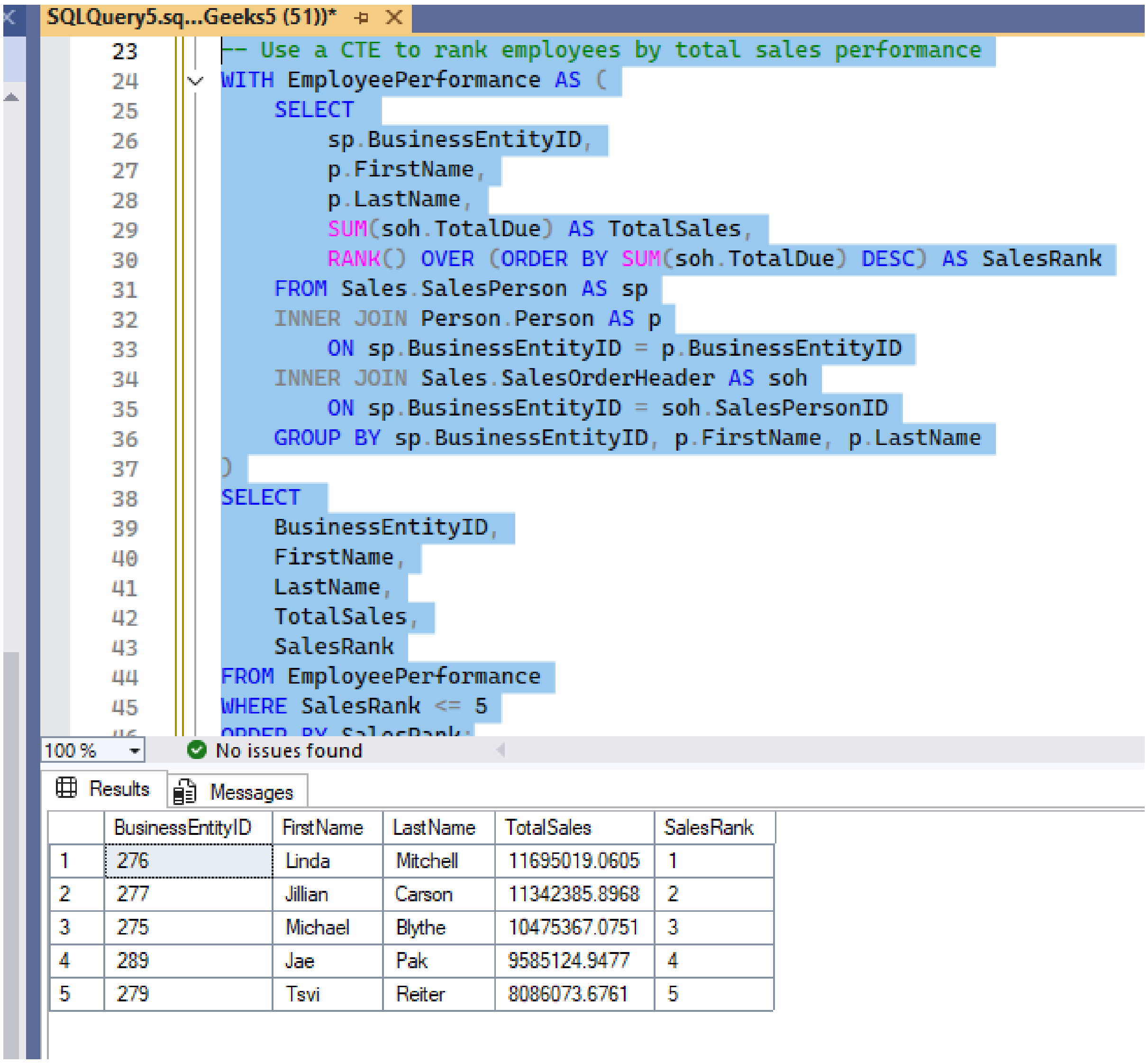
100 % — No issues found

Results | Messages

	ProductID	ProductName	TotalQuantitySold	TotalSalesAmount
1	782	Mountain-200 Black, 38	2977	4400592.800400
2	783	Mountain-200 Black, 42	2664	4009494.761841
3	779	Mountain-200 Silver, 38	2394	3693678.025272
4	780	Mountain-200 Silver, 42	2234	3438478.860423
5	781	Mountain-200 Silver, 46	2216	3434256.941928
6	784	Mountain-200 Black, 46	2111	3309673.216908
7	793	Road-250 Black, 44	1642	2516857.314918
8	794	Road-250 Black, 48	1498	2347655.953454
9	795	Road-250 Black, 52	1245	2012447.775000
10	753	Road-150 Red, 56	664	1847818.628000
11	976	Road-350-W Yellow, 48	1622	1774883.557085
12	749	Road-150 Red, 62	600	1769096.688000
13	969	Touring-1000 Blue, 60	1120	1721242.514355
14	973	Road-350-W Yellow, 40	1477	1657198.182549
15	792	Road-250 Red, 58	946	1587008.182500
16	966	Touring-1000 Blue, 46	1002	1586953.573023
17	751	Road-150 Red, 48	493	1540803.062000
18	957	Touring-1000 Yellow, 60	1114	1518133.101147

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | G

```
WITH EmployeePerformance AS (
   SELECT
      sp.BusinessEntityID,
      p.FirstName,
      p.LastName,
      SUM(soh.TotalDue) AS TotalSales,
      RANK() OVER (ORDER BY SUM(soh.TotalDue) DESC) AS SalesRank
   FROM Sales.SalesPerson AS sp
   INNER JOIN Person.Person AS p
      ON sp.BusinessEntityID = p.BusinessEntityID
   INNER JOIN Sales.SalesOrderHeader AS soh
      ON sp.BusinessEntityID = soh.SalesPersonID
   GROUP BY sp.BusinessEntityID, p.FirstName, p.LastName
)
SELECT
   BusinessEntityID,
   FirstName,
   LastName,
   TotalSales,
   SalesRank
FROM EmployeePerformance
WHERE SalesRank <= 5
ORDER BY SalesRank;
```



SQLQuery5.sq...Geeks5 (51))*

```sql
-- Use a CTE to rank employees by total sales performance
WITH EmployeePerformance AS (
    SELECT
        sp.BusinessEntityID,
        p.FirstName,
        p.LastName,
        SUM(soh.TotalDue) AS TotalSales,
        RANK() OVER (ORDER BY SUM(soh.TotalDue) DESC) AS SalesRank
    FROM Sales.SalesPerson AS sp
    INNER JOIN Person.Person AS p
        ON sp.BusinessEntityID = p.BusinessEntityID
    INNER JOIN Sales.SalesOrderHeader AS soh
        ON sp.BusinessEntityID = soh.SalesPersonID
    GROUP BY sp.BusinessEntityID, p.FirstName, p.LastName
)
SELECT
    BusinessEntityID,
    FirstName,
    LastName,
    TotalSales,
    SalesRank
FROM EmployeePerformance
WHERE SalesRank <= 5
ORDER BY SalesRank;
```

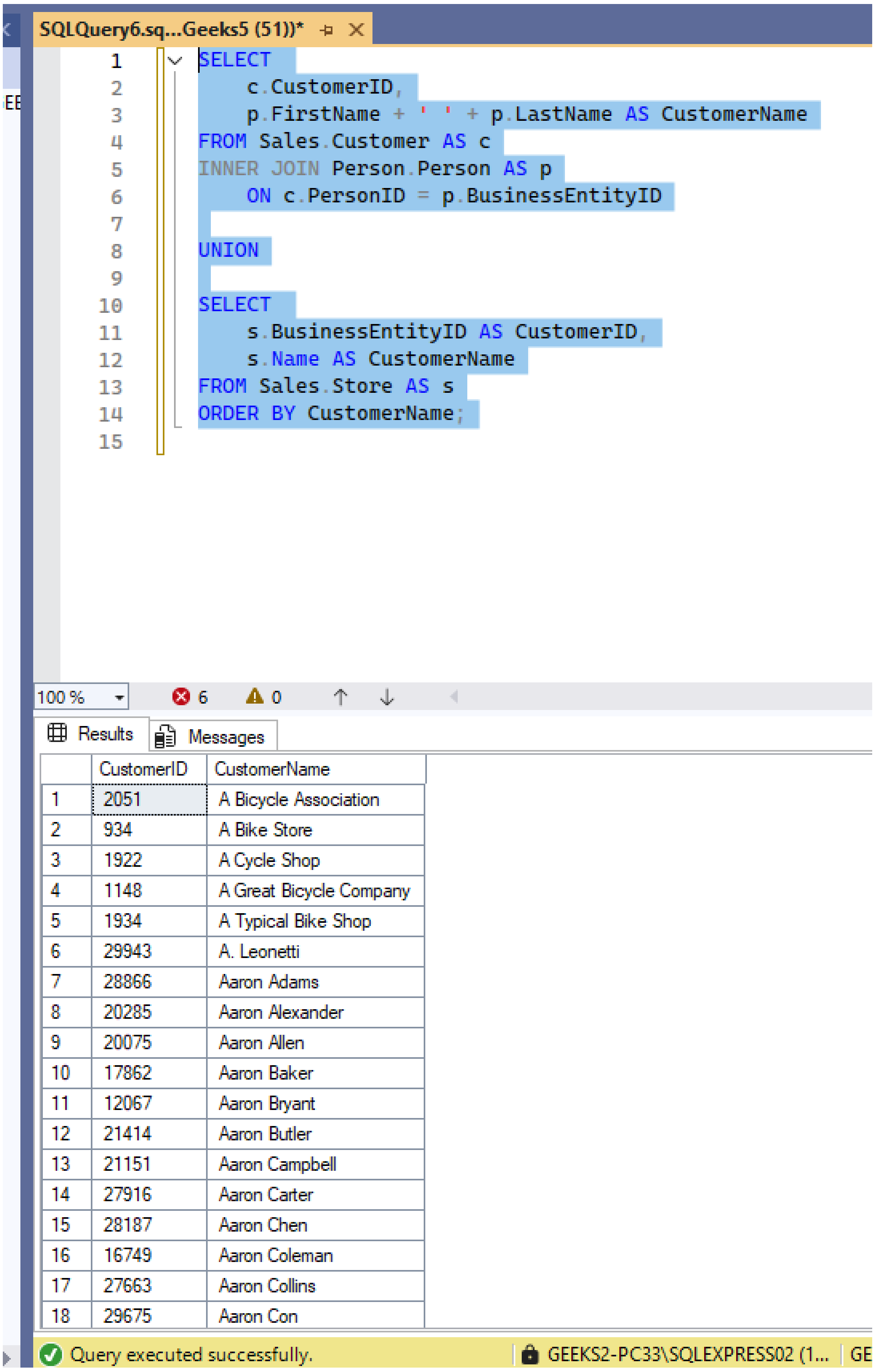
100 % No issues found

Results Messages

	BusinessEntityID	FirstName	LastName	TotalSales	SalesRank
1	276	Linda	Mitchell	11695019.0605	1
2	277	Jillian	Carson	11342385.8968	2
3	275	Michael	Blythe	10475367.0751	3
4	289	Jae	Pak	9585124.9477	4
5	279	Tsvi	Reiter	8086073.6761	5

Task 10

```
SELECT

    c.CustomerID,

    p.FirstName + ' ' + p.LastName AS CustomerName

FROM Sales.Customer AS c

INNER JOIN Person.Person AS p

    ON c.PersonID = p.BusinessEntityID


UNION


SELECT

    s.BusinessEntityID AS CustomerID,

    s.Name AS CustomerName

FROM Sales.Store AS s

ORDER BY CustomerName;
```

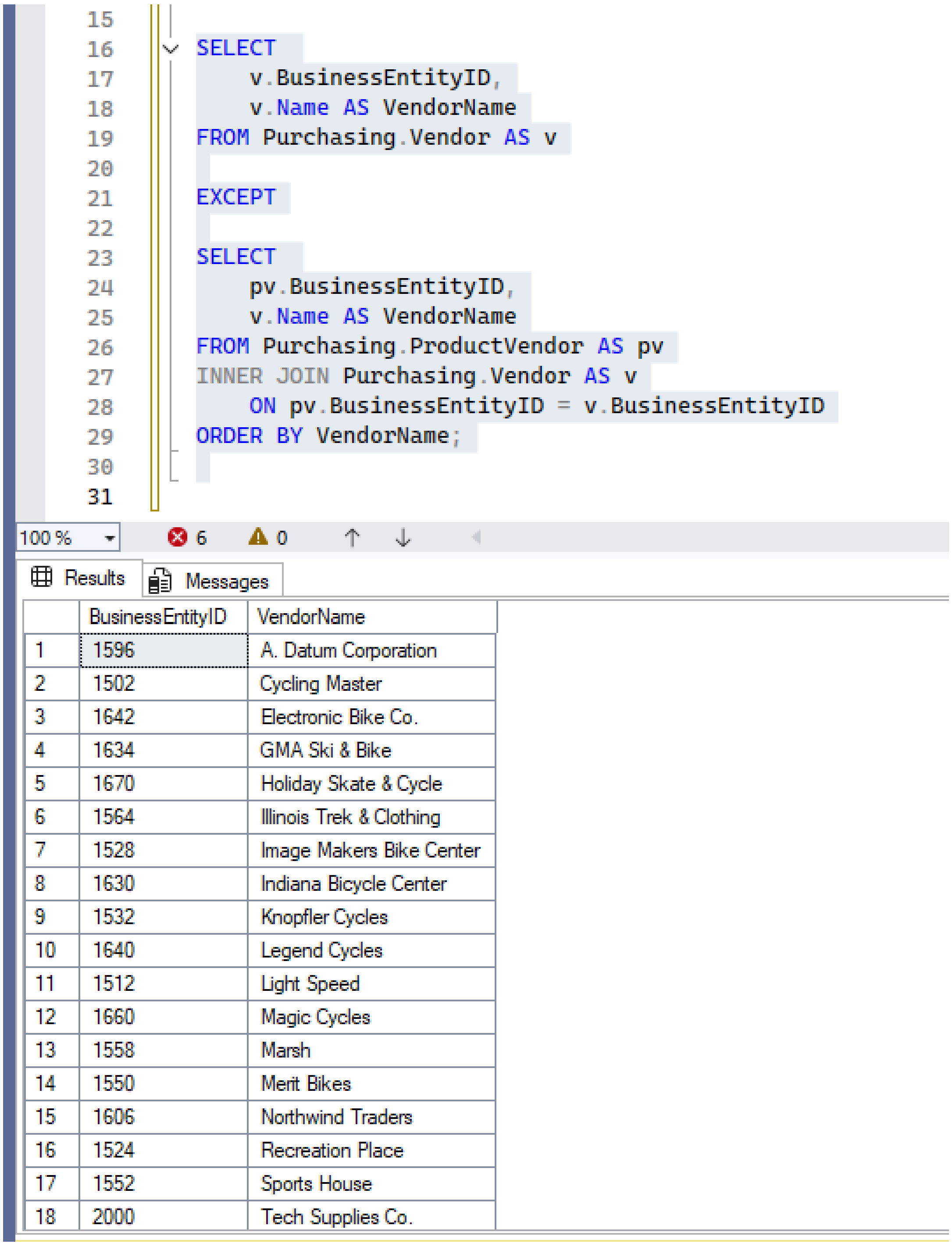


```
SELECT
    c.CustomerID,
    p.FirstName + ' ' + p.LastName AS CustomerName
FROM Sales.Customer AS c
INNER JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID

UNION

SELECT
    s.BusinessEntityID AS CustomerID,
    s.Name AS CustomerName
FROM Sales.Store AS s
ORDER BY CustomerName;
```

	CustomerID	CustomerName
1	2051	A Bicycle Association
2	934	A Bike Store
3	1922	A Cycle Shop
4	1148	A Great Bicycle Company
5	1934	A Typical Bike Shop
6	29943	A. Leonetti
7	28866	Aaron Adams
8	20285	Aaron Alexander
9	20075	Aaron Allen
10	17862	Aaron Baker
11	12067	Aaron Bryant
12	21414	Aaron Butler
13	21151	Aaron Campbell
14	27916	Aaron Carter
15	28187	Aaron Chen
16	16749	Aaron Coleman
17	27663	Aaron Collins
18	29675	Aaron Con

Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GE

```
SELECT

    v.BusinessEntityID,

    v.Name AS VendorName

FROM Purchasing.Vendor AS v


EXCEPT


SELECT

    pv.BusinessEntityID,

    v.Name AS VendorName

FROM Purchasing.ProductVendor AS pv

INNER JOIN Purchasing.Vendor AS v

    ON pv.BusinessEntityID = v.BusinessEntityID

ORDER BY VendorName;
```



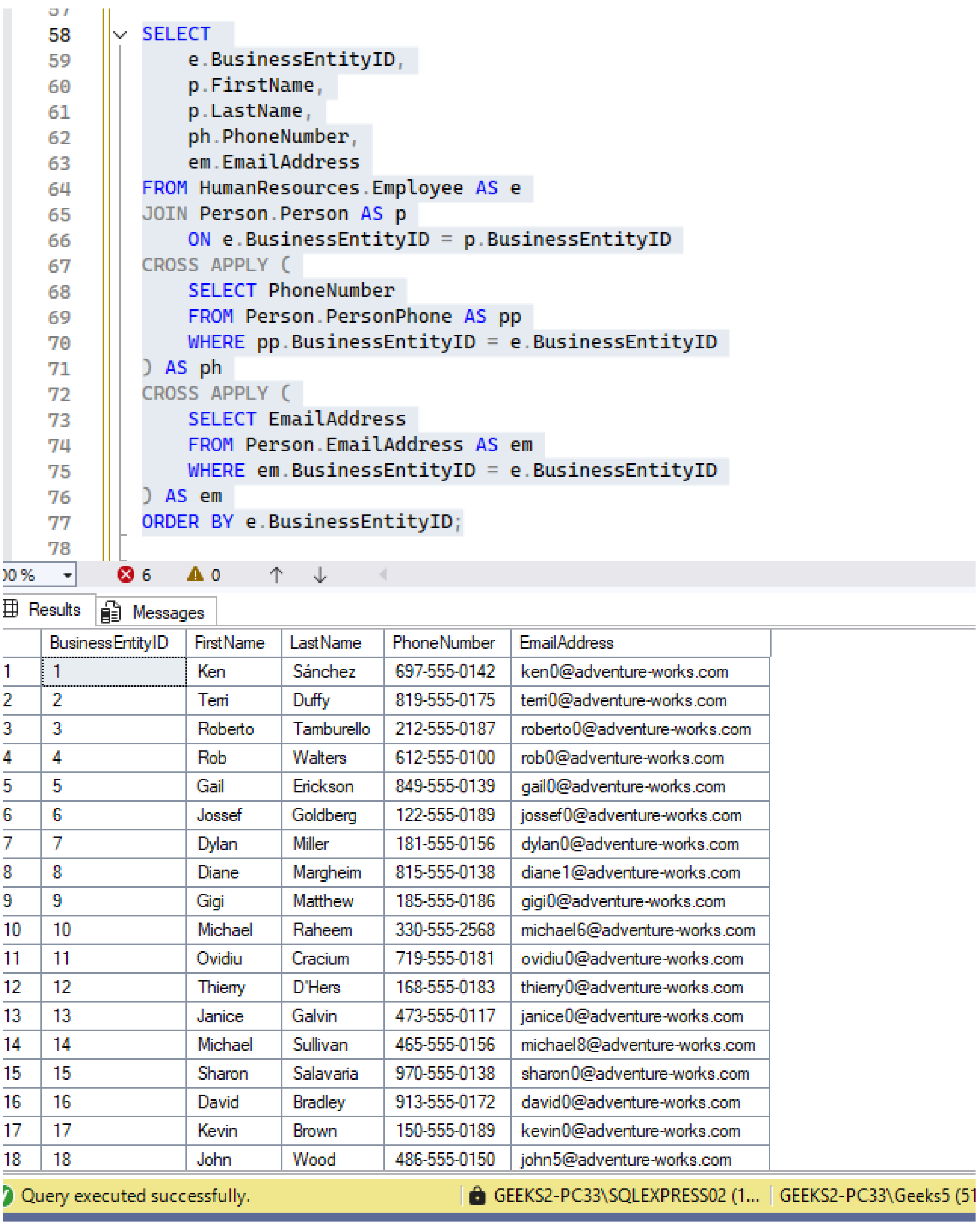
```sql
SELECT
    v.BusinessEntityID,
    v.Name AS VendorName
FROM Purchasing.Vendor AS v

EXCEPT

SELECT
    pv.BusinessEntityID,
    v.Name AS VendorName
FROM Purchasing.ProductVendor AS pv
INNER JOIN Purchasing.Vendor AS v
    ON pv.BusinessEntityID = v.BusinessEntityID
ORDER BY VendorName;
```

100 % 6 0

Results Messages

	BusinessEntityID	VendorName
1	1596	A. Datum Corporation
2	1502	Cycling Master
3	1642	Electronic Bike Co.
4	1634	GMA Ski & Bike
5	1670	Holiday Skate & Cycle
6	1564	Illinois Trek & Clothing
7	1528	Image Makers Bike Center
8	1630	Indiana Bicycle Center
9	1532	Knopfler Cycles
10	1640	Legend Cycles
11	1512	Light Speed
12	1660	Magic Cycles
13	1558	Marsh
14	1550	Merit Bikes
15	1606	Northwind Traders
16	1524	Recreation Place
17	1552	Sports House
18	2000	Tech Supplies Co.

```
SELECT
    e.BusinessEntityID,
    p.FirstName,
    p.LastName,
    ph.PhoneNumber,
    em.EmailAddress
FROM HumanResources.Employee AS e
JOIN Person.Person AS p
    ON e.BusinessEntityID = p.BusinessEntityID
CROSS APPLY (
    SELECT PhoneNumber
    FROM Person.PersonPhone AS pp
    WHERE pp.BusinessEntityID = e.BusinessEntityID
) AS ph
CROSS APPLY (
    SELECT EmailAddress
    FROM Person.EmailAddress AS em
    WHERE em.BusinessEntityID = e.BusinessEntityID
) AS em
ORDER BY e.BusinessEntityID;
```



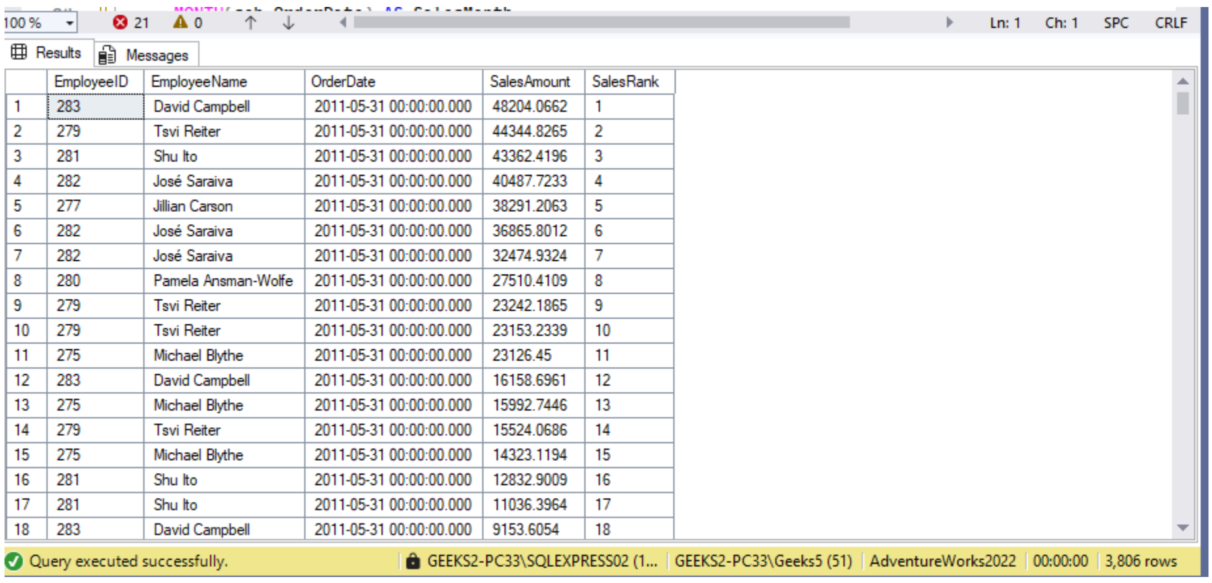
```sql
SELECT
    e.BusinessEntityID,
    p.FirstName,
    p.LastName,
    ph.PhoneNumber,
    em.EmailAddress
FROM HumanResources.Employee AS e
JOIN Person.Person AS p
    ON e.BusinessEntityID = p.BusinessEntityID
CROSS APPLY (
    SELECT PhoneNumber
    FROM Person.PersonPhone AS pp
    WHERE pp.BusinessEntityID = e.BusinessEntityID
) AS ph
CROSS APPLY (
    SELECT EmailAddress
    FROM Person.EmailAddress AS em
    WHERE em.BusinessEntityID = e.BusinessEntityID
) AS em
ORDER BY e.BusinessEntityID;
```

	BusinessEntityID	FirstName	LastName	PhoneNumber	EmailAddress
1	1	Ken	Sánchez	697-555-0142	ken0@adventure-works.com
2	2	Terri	Duffy	819-555-0175	terri0@adventure-works.com
3	3	Roberto	Tamburello	212-555-0187	roberto0@adventure-works.com
4	4	Rob	Walters	612-555-0100	rob0@adventure-works.com
5	5	Gail	Erickson	849-555-0139	gail0@adventure-works.com
6	6	Jossef	Goldberg	122-555-0189	jossef0@adventure-works.com
7	7	Dylan	Miller	181-555-0156	dylan0@adventure-works.com
8	8	Diane	Margheim	815-555-0138	diane1@adventure-works.com
9	9	Gigi	Matthew	185-555-0186	gigi0@adventure-works.com
10	10	Michael	Raheem	330-555-2568	michael6@adventure-works.com
11	11	Ovidiu	Cracium	719-555-0181	ovidiu0@adventure-works.com
12	12	Thierry	D'Hers	168-555-0183	thierry0@adventure-works.com
13	13	Janice	Galvin	473-555-0117	janice0@adventure-works.com
14	14	Michael	Sullivan	465-555-0156	michael8@adventure-works.com
15	15	Sharon	Salavaria	970-555-0138	sharon0@adventure-works.com
16	16	David	Bradley	913-555-0172	david0@adventure-works.com
17	17	Kevin	Brown	150-555-0189	kevin0@adventure-works.com
18	18	John	Wood	486-555-0150	john5@adventure-works.com

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (51

Task 11

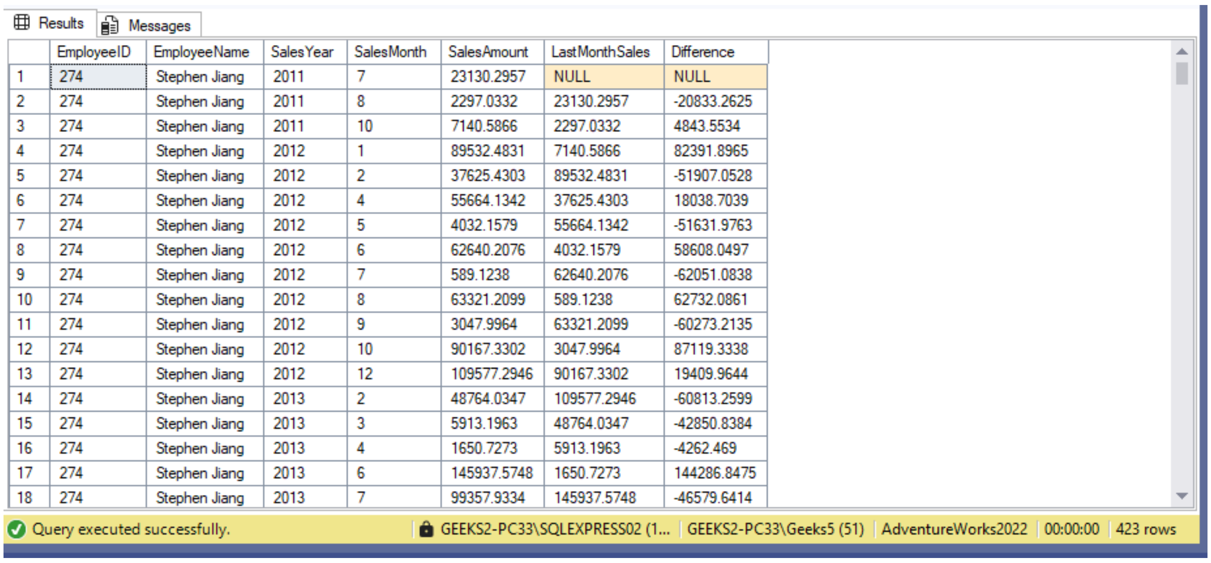
```sql
-- 1. Assign sales rank per employee
SELECT
    sp.BusinessEntityID AS EmployeeID,
    p.FirstName + ' ' + p.LastName AS EmployeeName,
    soh.OrderDate,
    soh.TotalDue AS SalesAmount,
    RANK() OVER (
        PARTITION BY YEAR(soh.OrderDate), MONTH(soh.OrderDate)
        ORDER BY soh.TotalDue DESC
    ) AS SalesRank
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesPerson AS sp
    ON soh.SalesPersonID = sp.BusinessEntityID
JOIN Person.Person AS p
    ON sp.BusinessEntityID = p.BusinessEntityID;
```



	EmployeeID	EmployeeName	OrderDate	SalesAmount	SalesRank
1	283	David Campbell	2011-05-31 00:00:00.000	48204.0662	1
2	279	Tsvi Reiter	2011-05-31 00:00:00.000	44344.8265	2
3	281	Shu Ito	2011-05-31 00:00:00.000	43362.4196	3
4	282	José Saraiva	2011-05-31 00:00:00.000	40487.7233	4
5	277	Jillian Carson	2011-05-31 00:00:00.000	38291.2063	5
6	282	José Saraiva	2011-05-31 00:00:00.000	36865.8012	6
7	282	José Saraiva	2011-05-31 00:00:00.000	32474.9324	7
8	280	Pamela Ansman-Wolfe	2011-05-31 00:00:00.000	27510.4109	8
9	279	Tsvi Reiter	2011-05-31 00:00:00.000	23242.1865	9
10	279	Tsvi Reiter	2011-05-31 00:00:00.000	23153.2339	10
11	275	Michael Blythe	2011-05-31 00:00:00.000	23126.45	11
12	283	David Campbell	2011-05-31 00:00:00.000	16158.6961	12
13	275	Michael Blythe	2011-05-31 00:00:00.000	15992.7446	13
14	279	Tsvi Reiter	2011-05-31 00:00:00.000	15524.0686	14
15	275	Michael Blythe	2011-05-31 00:00:00.000	14323.1194	15
16	281	Shu Ito	2011-05-31 00:00:00.000	12832.9009	16
17	281	Shu Ito	2011-05-31 00:00:00.000	11036.3964	17
18	283	David Campbell	2011-05-31 00:00:00.000	9153.6054	18

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (51) | AdventureWorks2022 | 00:00:00 | 3,806 rows

```
-- 2. Compare this month’s vs last month’s sales
SELECT
    sp.BusinessEntityID AS EmployeeID,
    p.FirstName + ' ' + p.LastName AS EmployeeName,
    YEAR(soh.OrderDate) AS SalesYear,
    MONTH(soh.OrderDate) AS SalesMonth,
    SUM(soh.TotalDue) AS SalesAmount,
    LAG(SUM(soh.TotalDue)) OVER (
        PARTITION BY sp.BusinessEntityID
        ORDER BY YEAR(soh.OrderDate), MONTH(soh.OrderDate)
    ) AS LastMonthSales,
    SUM(soh.TotalDue) - LAG(SUM(soh.TotalDue)) OVER (
        PARTITION BY sp.BusinessEntityID
        ORDER BY YEAR(soh.OrderDate), MONTH(soh.OrderDate)
    ) AS Difference
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesPerson AS sp
    ON soh.SalesPersonID = sp.BusinessEntityID
JOIN Person.Person AS p
    ON sp.BusinessEntityID = p.BusinessEntityID
GROUP BY sp.BusinessEntityID, p.FirstName, p.LastName, YEAR(soh.OrderDate), MONTH(soh.OrderDate)
ORDER BY sp.BusinessEntityID, SalesYear, SalesMonth;
```

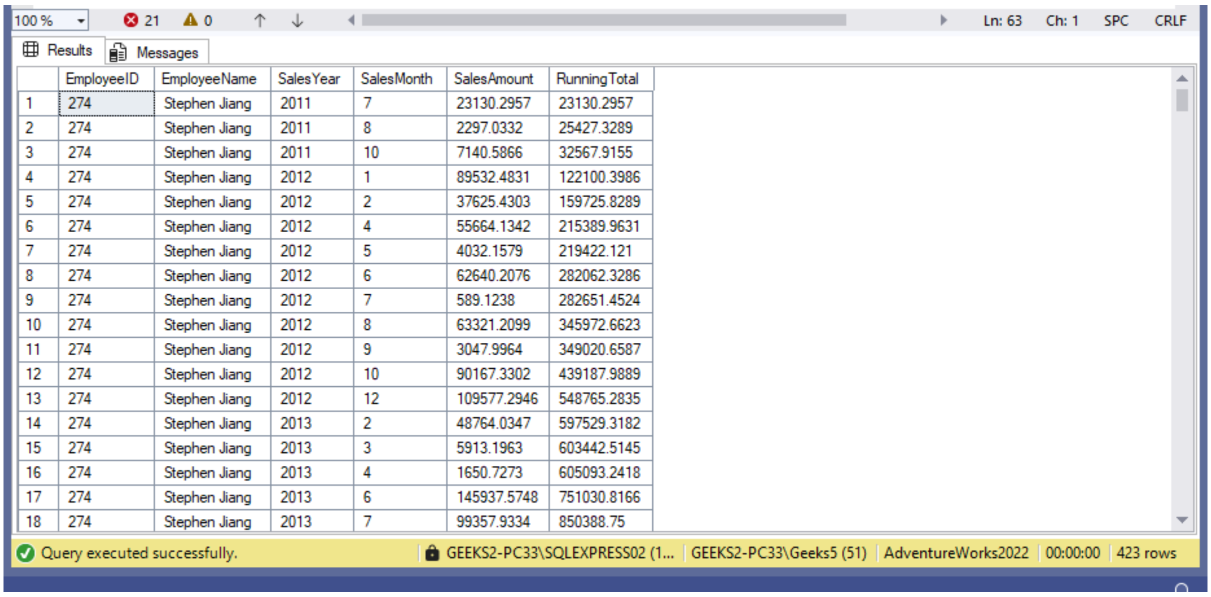


	EmployeeID	EmployeeName	SalesYear	SalesMonth	SalesAmount	LastMonthSales	Difference
1	274	Stephen Jiang	2011	7	23130.2957	NULL	NULL
2	274	Stephen Jiang	2011	8	2297.0332	23130.2957	-20833.2625
3	274	Stephen Jiang	2011	10	7140.5866	2297.0332	4843.5534
4	274	Stephen Jiang	2012	1	89532.4831	7140.5866	82391.8965
5	274	Stephen Jiang	2012	2	37625.4303	89532.4831	-51907.0528
6	274	Stephen Jiang	2012	4	55664.1342	37625.4303	18038.7039
7	274	Stephen Jiang	2012	5	4032.1579	55664.1342	-51631.9763
8	274	Stephen Jiang	2012	6	62640.2076	4032.1579	58608.0497
9	274	Stephen Jiang	2012	7	589.1238	62640.2076	-62051.0838
10	274	Stephen Jiang	2012	8	63321.2099	589.1238	62732.0861
11	274	Stephen Jiang	2012	9	3047.9964	63321.2099	-60273.2135
12	274	Stephen Jiang	2012	10	90167.3302	3047.9964	87119.3338
13	274	Stephen Jiang	2012	12	109577.2946	90167.3302	19409.9644
14	274	Stephen Jiang	2013	2	48764.0347	109577.2946	-60813.2599
15	274	Stephen Jiang	2013	3	5913.1963	48764.0347	-42850.8384
16	274	Stephen Jiang	2013	4	1650.7273	5913.1963	-4262.469
17	274	Stephen Jiang	2013	6	145937.5748	1650.7273	144286.8475
18	274	Stephen Jiang	2013	7	99357.9334	145937.5748	-46579.6414

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (51) | AdventureWorks2022 | 00:00:00 | 423 rows

-- 3. Get running total of sales per employee

```
SELECT
    sp.BusinessEntityID AS EmployeeID,
    p.FirstName + ' ' + p.LastName AS EmployeeName,
    YEAR(soh.OrderDate) AS SalesYear,
    MONTH(soh.OrderDate) AS SalesMonth,
    SUM(soh.TotalDue) AS SalesAmount,
    SUM(SUM(soh.TotalDue)) OVER (
        PARTITION BY sp.BusinessEntityID
        ORDER BY YEAR(soh.OrderDate), MONTH(soh.OrderDate)
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS RunningTotal
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesPerson AS sp
    ON soh.SalesPersonID = sp.BusinessEntityID
JOIN Person.Person AS p
    ON sp.BusinessEntityID = p.BusinessEntityID
GROUP BY sp.BusinessEntityID, p.FirstName, p.LastName, YEAR(soh.OrderDate), MONTH(soh.OrderDate)
ORDER BY sp.BusinessEntityID, SalesYear, SalesMonth;
```



100 % | 21 | 0 | Ln: 63 | Ch: 1 | SPC | CRLF

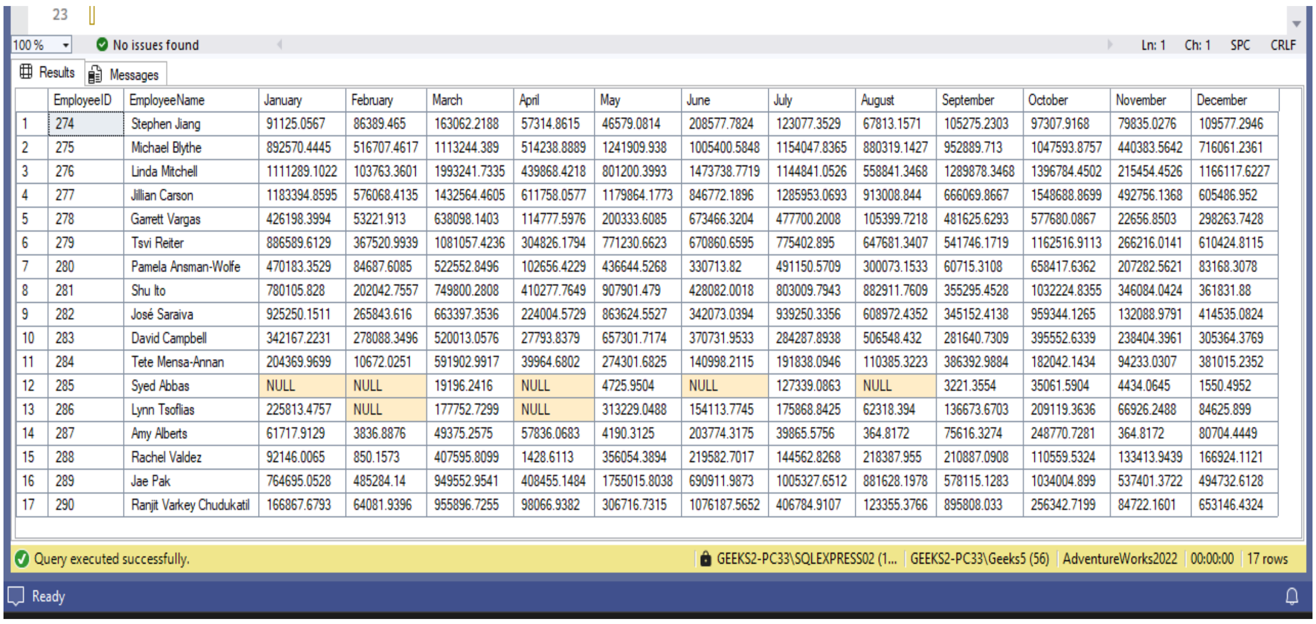
Results | Messages

	EmployeeID	EmployeeName	SalesYear	SalesMonth	SalesAmount	RunningTotal
1	274	Stephen Jiang	2011	7	23130.2957	23130.2957
2	274	Stephen Jiang	2011	8	2297.0332	25427.3289
3	274	Stephen Jiang	2011	10	7140.5866	32567.9155
4	274	Stephen Jiang	2012	1	89532.4831	122100.3986
5	274	Stephen Jiang	2012	2	37625.4303	159725.8289
6	274	Stephen Jiang	2012	4	55664.1342	215389.9631
7	274	Stephen Jiang	2012	5	4032.1579	219422.121
8	274	Stephen Jiang	2012	6	62640.2076	282062.3286
9	274	Stephen Jiang	2012	7	589.1238	282651.4524
10	274	Stephen Jiang	2012	8	63321.2099	345972.6623
11	274	Stephen Jiang	2012	9	3047.9964	349020.6587
12	274	Stephen Jiang	2012	10	90167.3302	439187.9889
13	274	Stephen Jiang	2012	12	109577.2946	548765.2835
14	274	Stephen Jiang	2013	2	48764.0347	597529.3182
15	274	Stephen Jiang	2013	3	5913.1963	603442.5145
16	274	Stephen Jiang	2013	4	1650.7273	605093.2418
17	274	Stephen Jiang	2013	6	145937.5748	751030.8166
18	274	Stephen Jiang	2013	7	99357.9334	850388.75

Query executed successfully. | GEEKS2-PC33\SQLEXPRESS02 (1... | GEEKS2-PC33\Geeks5 (51) | AdventureWorks2022 | 00:00:00 | 423 rows

Task 12

```
-- Pivot sales by month for each SalesPerson
SELECT *
FROM
(
    SELECT
        sp.BusinessEntityID AS EmployeeID,
        p.FirstName + ' ' + p.LastName AS EmployeeName,
        DATENAME(MONTH, soh.OrderDate) AS SalesMonth,
        soh.TotalDue AS SalesAmount
    FROM Sales.SalesOrderHeader AS soh
    JOIN Sales.SalesPerson AS sp
        ON soh.SalesPersonID = sp.BusinessEntityID
    JOIN Person.Person AS p
        ON sp.BusinessEntityID = p.BusinessEntityID
) AS SourceTable
PIVOT
(
    SUM(SalesAmount)
    FOR SalesMonth IN ([January],[February],[March],[April],[May],[June],
                       [July],[August],[September],[October],[November],[December])
) AS PivotTable
ORDER BY EmployeeID;
```



100 % No issues found Ln: 1 Ch: 1 SPC CRLF

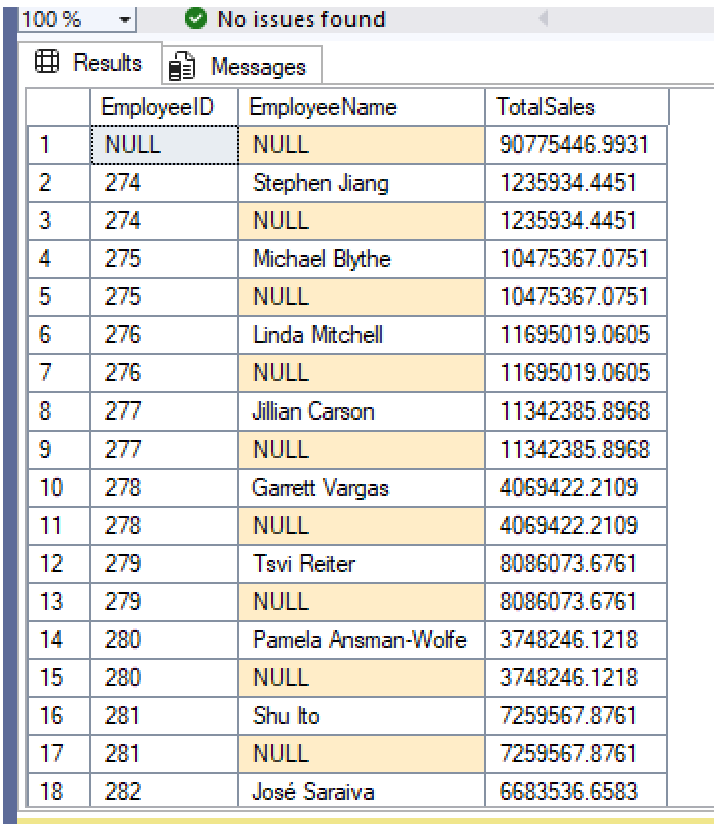
Results Messages

	EmployeeID	EmployeeName	January	February	March	April	May	June	July	August	September	October	November	December
1	274	Stephen Jiang	91125.0567	86389.465	163062.2188	57314.8615	46579.0814	208577.7824	123077.3529	67813.1571	105275.2303	97307.9168	79835.0276	109577.2946
2	275	Michael Blythe	892570.4445	516707.4617	1113244.389	514238.8889	1241909.938	1005400.5848	1154047.8365	880319.1427	952889.713	1047593.8757	440383.5642	716061.2361
3	276	Linda Mitchell	1111289.1022	103763.3601	1993241.7335	439868.4218	801200.3993	1473738.7719	1144841.0526	558841.3468	1289878.3468	1396784.4502	215454.4526	1166117.6227
4	277	Jillian Carson	1183394.8595	576068.4135	1432564.4605	611758.0577	1179864.1773	846772.1896	1285953.0693	913008.844	666069.8667	1548688.8699	492756.1368	605486.952
5	278	Garrett Vargas	426198.3994	53221.913	638098.1403	114777.5976	200333.6085	673466.3204	477700.2008	105399.7218	481625.6293	577680.0867	22656.8503	298263.7428
6	279	Tsvi Reiter	886589.6129	367520.9939	1081057.4236	304826.1794	771230.6623	670860.6595	775402.895	647681.3407	541746.1719	1162516.9113	266216.0141	610424.8115
7	280	Pamela Ansman-Wolfe	470183.3529	84687.6085	522552.8496	102656.4229	436644.5268	330713.82	491150.5709	300073.1533	60715.3108	658417.6362	207282.5621	83168.3078
8	281	Shu Ito	780105.828	202042.7557	749800.2808	410277.7649	907901.479	428082.0018	803009.7943	882911.7609	355295.4528	1032224.8355	346084.0424	361831.88
9	282	José Saraiva	925250.1511	265843.616	663397.3536	224004.5729	863624.5527	342073.0394	939250.3356	608972.4352	345152.4138	959344.1265	132088.9791	414535.0824
10	283	David Campbell	342167.2231	278088.3496	520013.0576	27793.8379	657301.7174	370731.9533	284287.8938	506548.432	281640.7309	395552.6339	238404.3961	305364.3769
11	284	Tete Mensa-Annan	204369.9699	10672.0251	591902.9917	39964.6802	274301.6825	140998.2115	191838.0946	110385.3223	386392.9884	182042.1434	94233.0307	381015.2352
12	285	Syed Abbas	NULL	NULL	19196.2416	NULL	4725.9504	NULL	127339.0863	NULL	3221.3554	35061.5904	4434.0645	1550.4952
13	286	Lynn Tsoflias	225813.4757	NULL	177752.7299	NULL	313229.0488	154113.7745	175868.8425	62318.394	136673.6703	209119.3636	66926.2488	84625.899
14	287	Amy Alberts	61717.9129	3836.8876	49375.2575	57836.0683	4190.3125	203774.3175	39865.5756	364.8172	75616.3274	248770.7281	364.8172	80704.4449
15	288	Rachel Valdez	92146.0065	850.1573	407595.8099	1428.6113	356054.3894	219582.7017	144562.8268	218387.955	210887.0908	110559.5324	133413.9439	166924.1121
16	289	Jae Pak	764695.0528	485284.14	949552.9541	408455.1484	1755015.8038	690911.9873	1005327.6512	881628.1978	578115.1283	1034004.899	537401.3722	494732.6128
17	290	Ranjit Varkey Chudukatil	166867.6793	64081.9396	955896.7255	98066.9382	306716.7315	1076187.5652	406784.9107	123355.3766	895808.033	256342.7199	84722.1601	653146.4324

Query executed successfully. GEEKS2-PC33\SQLEXPRESS02 (1... GEEKS2-PC33\Geeks5 (56) AdventureWorks2022 00:00:00 17 rows

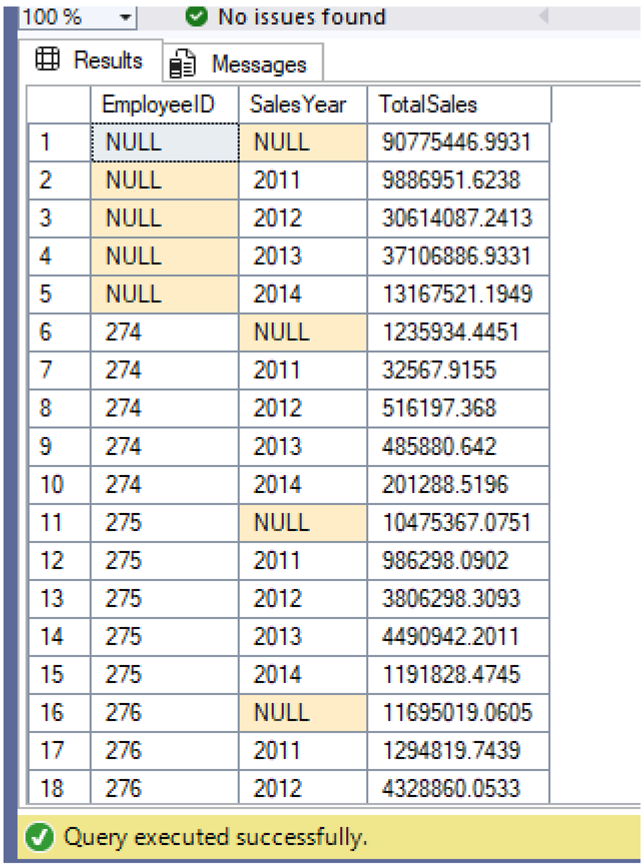
Ready

```
-- ROLLUP: subtotal by SalesPerson and Grand Total
SELECT
    sp.BusinessEntityID AS EmployeeID,
    p.FirstName + ' ' + p.LastName AS EmployeeName,
    SUM(soh.TotalDue) AS TotalSales
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesPerson AS sp
    ON soh.SalesPersonID = sp.BusinessEntityID
JOIN Person.Person AS p
    ON sp.BusinessEntityID = p.BusinessEntityID
GROUP BY ROLLUP(sp.BusinessEntityID, p.FirstName + ' ' + p.LastName)
ORDER BY EmployeeID;
```



	EmployeeID	EmployeeName	TotalSales
1	NULL	NULL	90775446.9931
2	274	Stephen Jiang	1235934.4451
3	274	NULL	1235934.4451
4	275	Michael Blythe	10475367.0751
5	275	NULL	10475367.0751
6	276	Linda Mitchell	11695019.0605
7	276	NULL	11695019.0605
8	277	Jillian Carson	11342385.8968
9	277	NULL	11342385.8968
10	278	Garrett Vargas	4069422.2109
11	278	NULL	4069422.2109
12	279	Tsvi Reiter	8086073.6761
13	279	NULL	8086073.6761
14	280	Pamela Ansman-Wolfe	3748246.1218
15	280	NULL	3748246.1218
16	281	Shu Ito	7259567.8761
17	281	NULL	7259567.8761
18	282	José Saraiva	6683536.6583

```
-- CUBE: subtotal by SalesPerson and Year

SELECT

    sp.BusinessEntityID AS EmployeeID,

    YEAR(soh.OrderDate) AS SalesYear,

    SUM(soh.TotalDue) AS TotalSales

FROM Sales.SalesOrderHeader AS soh

JOIN Sales.SalesPerson AS sp

    ON soh.SalesPersonID = sp.BusinessEntityID

GROUP BY CUBE(sp.BusinessEntityID, YEAR(soh.OrderDate))

ORDER BY EmployeeID, SalesYear;
```



100 % No issues found

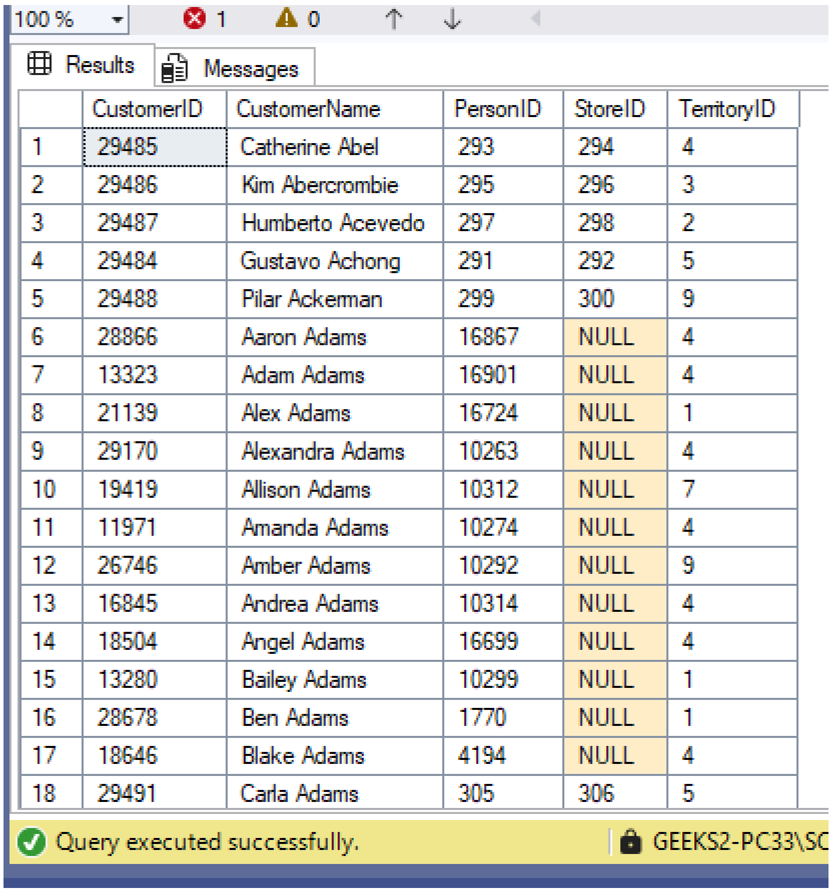
Results Messages

	EmployeeID	SalesYear	TotalSales
1	NULL	NULL	90775446.9931
2	NULL	2011	9886951.6238
3	NULL	2012	30614087.2413
4	NULL	2013	37106886.9331
5	NULL	2014	13167521.1949
6	274	NULL	1235934.4451
7	274	2011	32567.9155
8	274	2012	516197.368
9	274	2013	485880.642
10	274	2014	201288.5196
11	275	NULL	10475367.0751
12	275	2011	986298.0902
13	275	2012	3806298.3093
14	275	2013	4490942.2011
15	275	2014	1191828.4745
16	276	NULL	11695019.0605
17	276	2011	1294819.7439
18	276	2012	4328860.0533

Query executed successfully.

Task 13

EXEC dbo.GetAllCustomers;


100 %   1   0

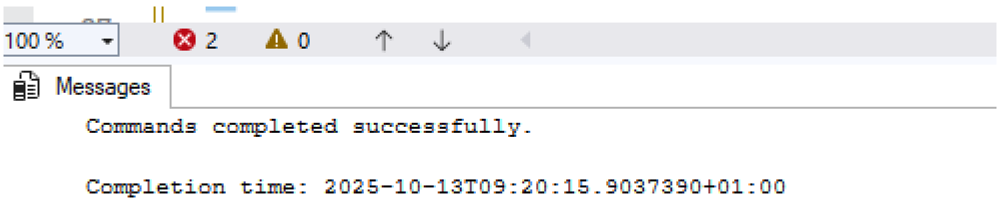
Results   Messages

	CustomerID	CustomerName	PersonID	StoreID	TerritoryID
1	29485	Catherine Abel	293	294	4
2	29486	Kim Abercrombie	295	296	3
3	29487	Humberto Acevedo	297	298	2
4	29484	Gustavo Achong	291	292	5
5	29488	Pilar Ackerman	299	300	9
6	28866	Aaron Adams	16867	NULL	4
7	13323	Adam Adams	16901	NULL	4
8	21139	Alex Adams	16724	NULL	1
9	29170	Alexandra Adams	10263	NULL	4
10	19419	Allison Adams	10312	NULL	7
11	11971	Amanda Adams	10274	NULL	4
12	26746	Amber Adams	10292	NULL	9
13	16845	Andrea Adams	10314	NULL	4
14	18504	Angel Adams	16699	NULL	4
15	13280	Bailey Adams	10299	NULL	1
16	28678	Ben Adams	1770	NULL	1
17	18646	Blake Adams	4194	NULL	4
18	29491	Carla Adams	305	306	5

Query executed successfully.   GEEKS2-PC33\SC

-- Create a new stored procedure to get top customers by region

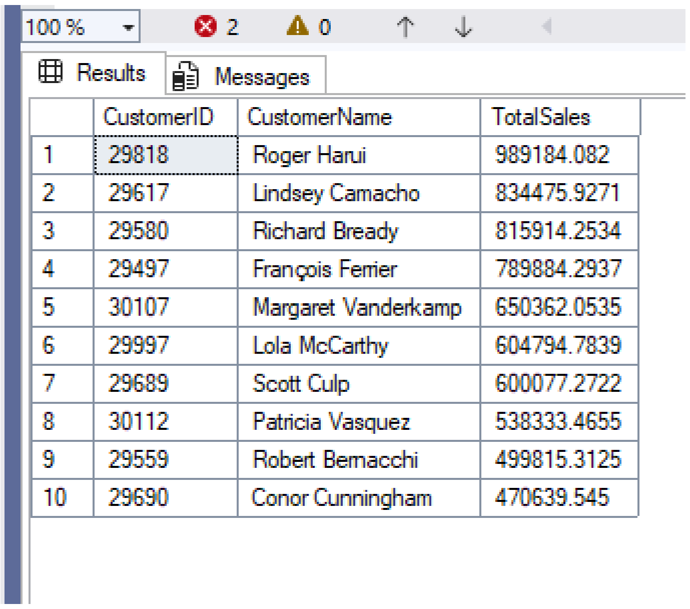
```
CREATE PROCEDURE dbo.GetTopCustomersByRegion
    @Region NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;

    SELECT TOP 10
        c.CustomerID,
        p.FirstName + ' ' + p.LastName AS CustomerName,
        SUM(soh.TotalDue) AS TotalSales
    FROM Sales.SalesOrderHeader AS soh
    JOIN Sales.Customer AS c
        ON soh.CustomerID = c.CustomerID
    JOIN Person.Person AS p
        ON c.PersonID = p.BusinessEntityID
    JOIN Sales.SalesTerritory AS t
        ON soh.TerritoryID = t.TerritoryID
    WHERE t.Name = @Region
    GROUP BY c.CustomerID, p.FirstName, p.LastName
    ORDER BY TotalSales DESC;
END;
GO
```



```
Commands completed successfully.

Completion time: 2025-10-13T09:20:15.9037390+01:00
```

-- Example execution for a specific region

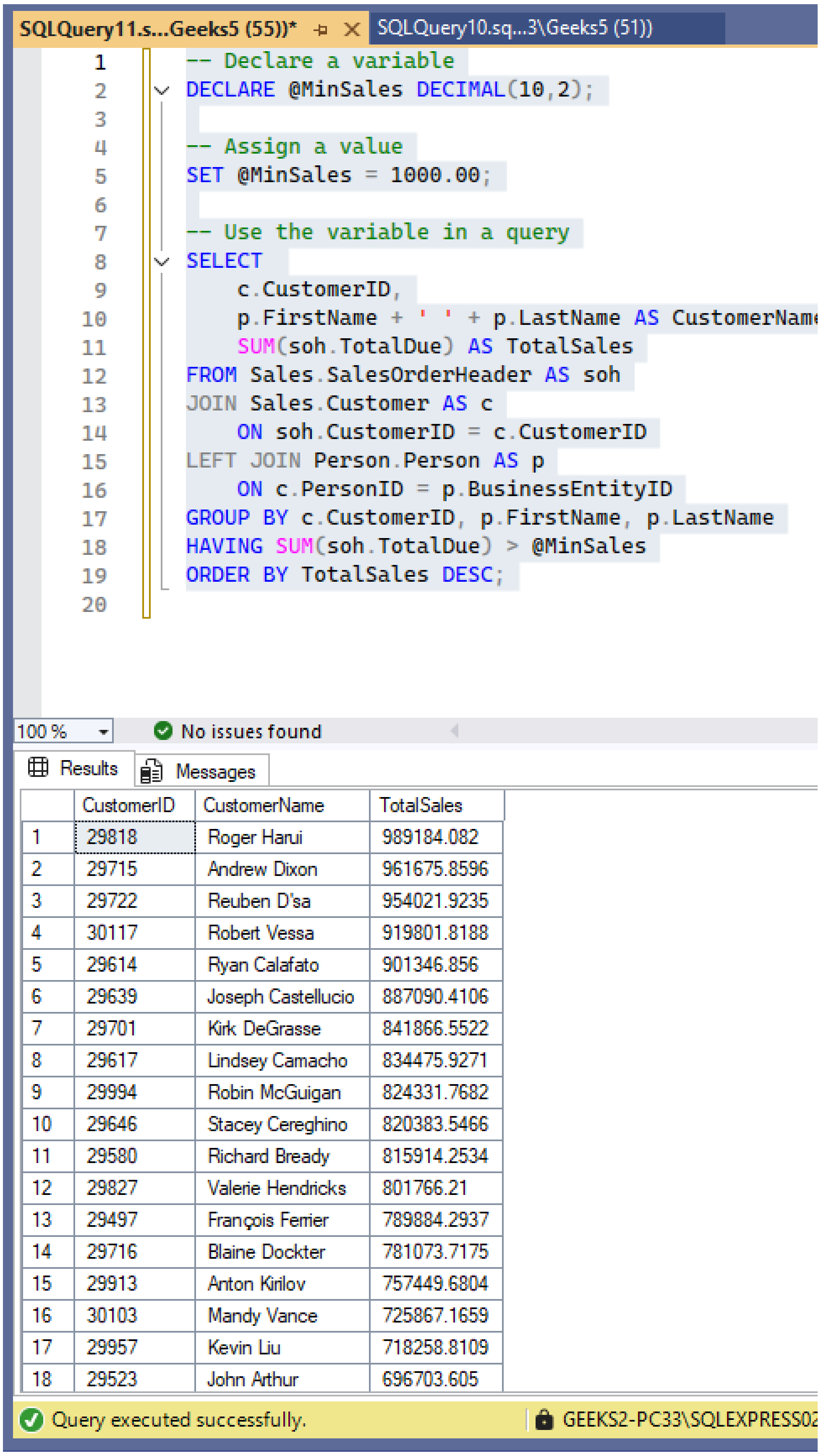
EXEC dbo.GetTopCustomersByRegion @Region = 'Northwest';



	CustomerID	CustomerName	TotalSales
1	29818	Roger Harui	989184.082
2	29617	Lindsey Camacho	834475.9271
3	29580	Richard Bready	815914.2534
4	29497	François Ferrier	789884.2937
5	30107	Margaret Vanderkamp	650362.0535
6	29997	Lola McCarthy	604794.7839
7	29689	Scott Culp	600077.2722
8	30112	Patricia Vasquez	538333.4655
9	29559	Robert Bernacchi	499815.3125
10	29690	Conor Cunningham	470639.545

Task 14

```
-- Declare a variable
DECLARE @MinSales DECIMAL(10,2);

-- Assign a value
SET @MinSales = 1000.00;

-- Use the variable in a query
SELECT
    c.CustomerID,
    p.FirstName + ' ' + p.LastName AS CustomerName,
    SUM(soh.TotalDue) AS TotalSales
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.Customer AS c
    ON soh.CustomerID = c.CustomerID
LEFT JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID
GROUP BY c.CustomerID, p.FirstName, p.LastName
HAVING SUM(soh.TotalDue) > @MinSales
ORDER BY TotalSales DESC;
```



SQLQuery11.s...Geeks5 (55))*   SQLQuery10.sq...3\Geeks5 (51))

```sql
-- Declare a variable
DECLARE @MinSales DECIMAL(10,2);

-- Assign a value
SET @MinSales = 1000.00;

-- Use the variable in a query
SELECT
    c.CustomerID,
    p.FirstName + ' ' + p.LastName AS CustomerName
    SUM(soh.TotalDue) AS TotalSales
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.Customer AS c
    ON soh.CustomerID = c.CustomerID
LEFT JOIN Person.Person AS p
    ON c.PersonID = p.BusinessEntityID
GROUP BY c.CustomerID, p.FirstName, p.LastName
HAVING SUM(soh.TotalDue) > @MinSales
ORDER BY TotalSales DESC;
```

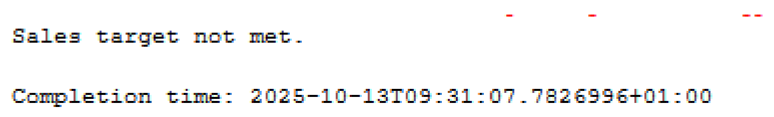
100 %   No issues found

Results   Messages

	CustomerID	CustomerName	TotalSales
1	29818	Roger Harui	989184.082
2	29715	Andrew Dixon	961675.8596
3	29722	Reuben D'sa	954021.9235
4	30117	Robert Vessa	919801.8188
5	29614	Ryan Calafato	901346.856
6	29639	Joseph Castellucio	887090.4106
7	29701	Kirk DeGrasse	841866.5522
8	29617	Lindsey Camacho	834475.9271
9	29994	Robin McGuigan	824331.7682
10	29646	Stacey Cereghino	820383.5466
11	29580	Richard Bready	815914.2534
12	29827	Valerie Hendricks	801766.21
13	29497	François Ferrier	789884.2937
14	29716	Blaine Dockter	781073.7175
15	29913	Anton Kirilov	757449.6804
16	30103	Mandy Vance	725867.1659
17	29957	Kevin Liu	718258.8109
18	29523	John Arthur	696703.605

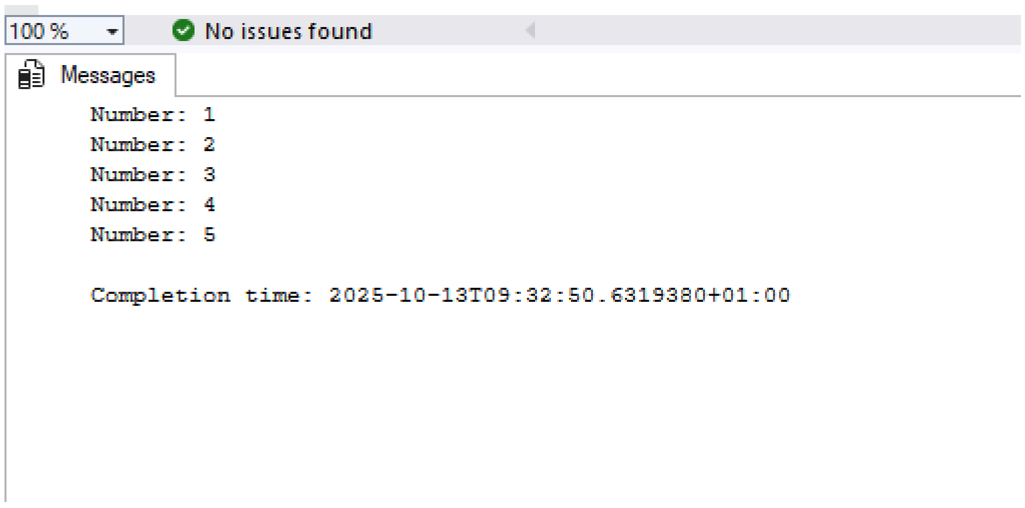
Query executed successfully.   GEEKS2-PC33\SQLEXPRESS02

```
-- 1. Declare a variable to hold total sales
DECLARE @TotalSales DECIMAL(10,2);

-- 2. Calculate total sales from all orders
SELECT @TotalSales = SUM(TotalDue)
FROM Sales.SalesOrderHeader;

-- 3. Check if total sales meet a target
IF @TotalSales >= 100000
BEGIN
    PRINT 'Sales target met!';
END
ELSE
BEGIN
    PRINT 'Sales target not met.';
END;
```



```
Sales target not met.

Completion time: 2025-10-13T09:31:07.7826996+01:00
```

```
DECLARE @i INT = 1;


WHILE @i <= 5

BEGIN

    PRINT 'Number: ' + CAST(@i AS VARCHAR(10));

    SET @i = @i + 1;

END;
```



100 %   No issues found

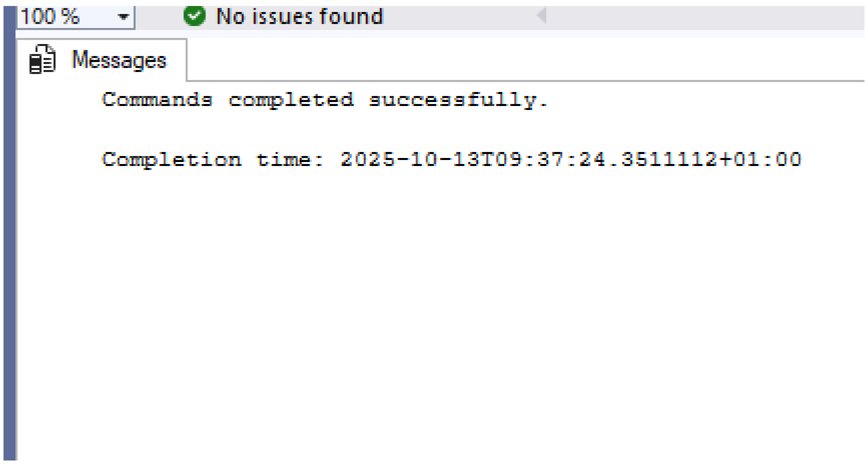
Messages

```
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5

Completion time: 2025-10-13T09:32:50.6319380+01:00
```

Task 15

-- Procedure: Calculate sales ratio with error handling

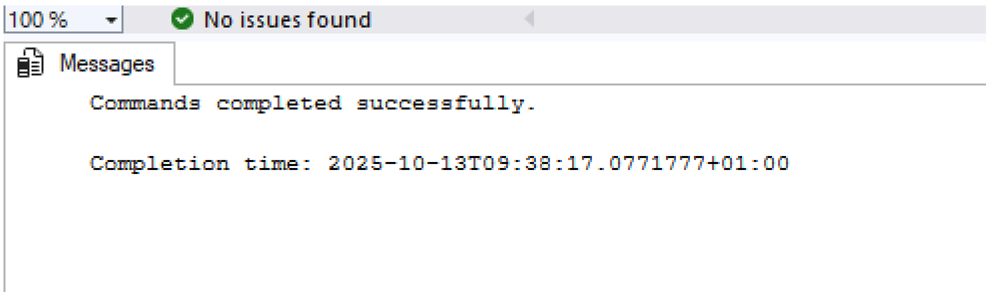
```
CREATE PROCEDURE dbo.CalculateSalesRatio
    @EmployeeID INT,
    @SalesTarget DECIMAL(10,2)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        DECLARE @TotalSales DECIMAL(10,2);
        DECLARE @Ratio DECIMAL(10,2);

        SELECT @TotalSales = SUM(TotalDue)
        FROM Sales.SalesOrderHeader
        WHERE SalesPersonID = @EmployeeID;

        SET @Ratio = @TotalSales / @SalesTarget;

        PRINT 'Sales ratio: ' + CAST(@Ratio AS VARCHAR(20));
    END TRY
    BEGIN CATCH
        PRINT 'An error occurred: ' + ERROR_MESSAGE();
    END CATCH
END;
GO
```



```
Commands completed successfully.

Completion time: 2025-10-13T09:37:24.3511112+01:00
```

-- Procedure: Raise custom error if target not met

```
CREATE PROCEDURE dbo.CheckSalesTarget
    @EmployeeID INT,
    @SalesTarget DECIMAL(10,2)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @TotalSales DECIMAL(10,2);

    SELECT @TotalSales = SUM(TotalDue)
    FROM Sales.SalesOrderHeader
    WHERE SalesPersonID = @EmployeeID;

    IF @TotalSales < @SalesTarget
        THROW 50001, 'Employee did not meet the sales target!', 1;
    ELSE
        PRINT 'Employee met the sales target!';
END;
GO
```

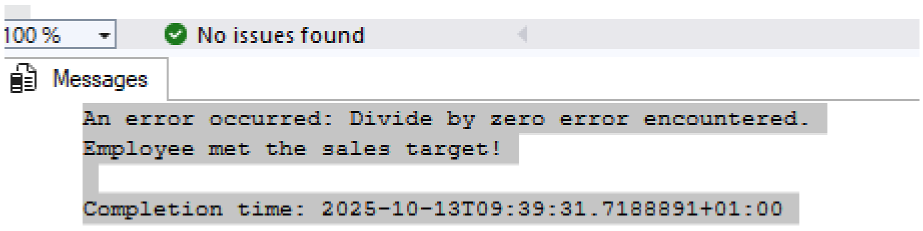


```
Commands completed successfully.

Completion time: 2025-10-13T09:38:17.0771777+01:00
```

-- Example executions

EXEC dbo.CalculateSalesRatio @EmployeeID = 276, @SalesTarget = 0;

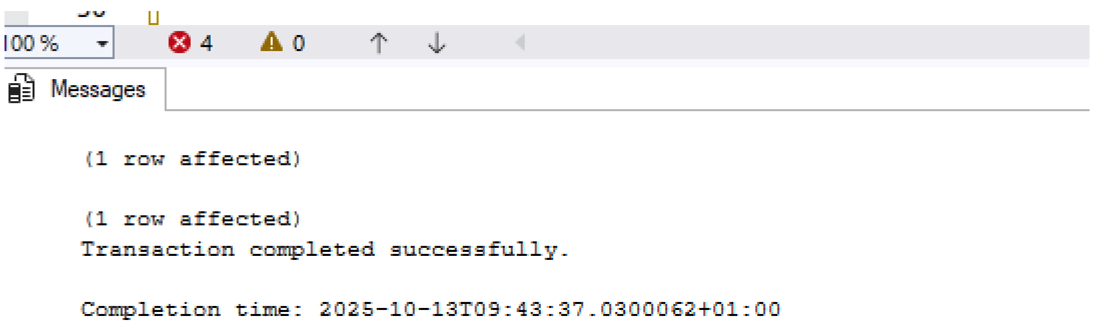
EXEC dbo.CheckSalesTarget @EmployeeID = 276, @SalesTarget = 50000;



```
An error occurred: Divide by zero error encountered.
Employee met the sales target!

Completion time: 2025-10-13T09:39:31.7188891+01:00
```

Task 16

```
-- Transaction example
SET XACT_ABORT ON;


BEGIN TRY
    BEGIN TRANSACTION;


    UPDATE Sales.Customer
    SET TerritoryID = 5
    WHERE CustomerID = 11000;


    --  log the change in a custom table
    INSERT INTO dbo.CustomerChangeLog (CustomerID, ChangeType, ChangeDate)
    VALUES (11000, 'Territory change', GETDATE());


    -- Commit transaction if both statements succeed
    COMMIT TRANSACTION;


    PRINT 'Transaction completed successfully.';
END TRY
BEGIN CATCH
    -- Rollback if any error occurs
    IF XACT_STATE() <> 0
        ROLLBACK TRANSACTION;


    PRINT 'Transaction rolled back due to error: ' + ERROR_MESSAGE();
END CATCH;
```



```
(1 row affected)

(1 row affected)
Transaction completed successfully.

Completion time: 2025-10-13T09:43:37.0300062+01:00
```