```
-- 1
-- Write a query that returns all orders placed on the last day of
-- activity that can be found in the Orders table
-- Tables involved: TSQLV4 database, Orders table

-- --Desired output
-- orderid     orderdate   custid      empid
-- ----------- ----------- ----------- -----------
-- 11077       2016-05-06  65          1
-- 11076       2016-05-06  9           4
-- 11075       2016-05-06  68          8
-- 11074       2016-05-06  73          7

-- (4 row(s) affected)

USE TSQLV4;

SELECT orderid, orderdate, custid, empid FROM Sales.Orders
WHERE orderdate =
    (SELECT MAX(O.orderdate) FROM Sales.Orders AS O)
ORDER BY orderid DESC;


USE Northwinds2022TSQLV7;
Go
SELECT OrderId, OrderDate, CustomerId, EmployeeId FROM Sales.[Order]
WHERE orderdate =
    (SELECT MAX(O.OrderDate) FROM Sales.[Order] AS O)
ORDER BY OrderId DESC;

```

Results Messages

	orderid	orderdate	custid	empid
1	11077	2016-05-06	65	1
2	11076	2016-05-06	9	4
3	11075	2016-05-06	68	8
4	11074	2016-05-06	73	7

	OrderId	OrderDate	CustomerId	EmployeeId
1	11077	2016-05-06	65	1
2	11076	2016-05-06	9	4
3	11075	2016-05-06	68	8
4	11074	2016-05-06	73	7

```sql
-- 2 (Optional, Advanced)
-- Write a query that returns all orders placed
-- by the customer(s) who placed the highest number of orders
-- * Note: there may be more than one customer
--   with the same number of orders
-- Tables involved: TSQLV4 database, Orders table

USE TSQLV4;

SELECT custid, orderid, orderdate, empid
FROM Sales.Orders
WHERE custid IN (SELECT TOP (1) WITH TIES O.custid
FROM Sales.Orders AS O
GROUP BY O.custid
ORDER BY COUNT(*) DESC);

USE Northwinds2022TSQLV7;
GO
SELECT CustomerId, orderid, orderdate, EmployeeId
FROM Sales.[Order]
WHERE CustomerId IN (SELECT TOP (1) WITH TIES O.CustomerId
FROM Sales.[Order] AS O
GROUP BY O.CustomerId
ORDER BY COUNT(*) DESC);
```

Results Messages

	custid	orderid	orderdate	empid
1	71	10324	2014-10-08	9
2	71	10393	2014-12-25	1
3	71	10398	2014-12-30	2
4	71	10440	2015-02-10	4
5	71	10452	2015-02-20	8
6	71	10510	2015-04-18	6
7	71	10555	2015-06-02	6
8	71	10603	2015-07-18	8
9	71	10607	2015-07-22	5
10	71	10612	2015-07-28	1

	CustomerId	orderid	orderdate	EmployeeId
1	71	10324	2014-10-08	9
2	71	10393	2014-12-25	1
3	71	10398	2014-12-30	2
4	71	10440	2015-02-10	4
5	71	10452	2015-02-20	8
6	71	10510	2015-04-18	6
7	71	10555	2015-06-02	6
8	71	10603	2015-07-18	8
9	71	10607	2015-07-22	5
10	71	10612	2015-07-28	1

```
-- 3
-- Write a query that returns employees
-- who did not place orders on or after May 1st, 2016
-- Tables involved: TSQLV4 database, Employees and Orders tables

-- -- Desired output:
-- empid       FirstName  lastname
-- ----------- ---------- --------------------
-- 3           Judy       Lew
-- 5           Sven       Mortensen
-- 6           Paul       Suurs
-- 9           Patricia   Doyle

-- (4 row(s) affected)

USE TSQLV4

SELECT empid, FirstName, lastname
FROM HR.Employees
WHERE empid NOT IN (SELECT O.empid
    FROM Sales.Orders AS O
    WHERE O.orderdate >= '20160501');

USE Northwinds2022TSQLV7
GO
SELECT EmployeeId, EmployeeFirstName, EmployeeLastName
FROM HumanResources.Employee
WHERE EmployeeId NOT IN (SELECT O.EmployeeId
    FROM Sales.[Order] AS O
    WHERE O.OrderDate >= '20160501');
```

Results | Messages

	empid	FirstName	lastname
1	3	Judy	Lew
2	5	Sven	Mortensen
3	6	Paul	Suurs
4	9	Patricia	Doyle

	EmployeeId	EmployeeFirstName	EmployeeLastName
1	3	Judy	Lew
2	5	Sven	Mortensen
3	6	Paul	Suurs
4	9	Patricia	Doyle

```sql
-- 4
-- Write a query that returns
-- countries where there are customers but not employees
-- Tables involved: TSQLV4 database, Customers and Employees tables

-- -- Desired output:
-- country
-- ---------------
-- Argentina
-- Austria
-- Belgium
-- Brazil
-- Canada
-- Denmark
-- Finland
-- France
-- Germany
-- Ireland
-- Italy
-- Mexico
-- Norway
-- Poland
-- Portugal
-- Spain
-- Sweden
-- Switzerland
-- Venezuela

-- (19 row(s) affected)

USE TSQLV4;

SELECT DISTINCT country
FROM Sales.Customers
WHERE country NOT IN (SELECT E.country
    FROM HR.Employees AS E);

USE Northwinds2022TSQLV7;
GO
SELECT DISTINCT CustomerCountry
FROM Sales.Customer
WHERE CustomerCountry NOT IN (SELECT E.EmployeeCountry
    FROM HumanResources.Employee AS E);
```

Results Messages

	country
1	Argentina
2	Austria
3	Belgium
4	Brazil
5	Canada
6	Denmark
7	Finland
8	France
9	Germany
10	Ireland

	CustomerCountry
1	Argentina
2	Austria
3	Belgium
4	Brazil

```sql
-- 5
-- Write a query that returns for each customer
-- all orders placed on the customer's last day of activity
-- Tables involved: TSQLV4 database, Orders table

-- -- Desired output:
-- custid      orderid     orderdate   empid
-- ----------- ----------- ----------- -----------
-- 1           11011       2016-04-09  3
-- 2           10926       2016-03-04  4
-- 3           10856       2016-01-28  3
-- 4           11016       2016-04-10  9
-- 5           10924       2016-03-04  3
-- ...
-- 87          11025       2016-04-15  6
-- 88          10935       2016-03-09  4
-- 89          11066       2016-05-01  7
-- 90          11005       2016-04-07  2
-- 91          11044       2016-04-23  4

-- (90 row(s) affected)

USE TSQLV4;

SELECT custid, orderid, orderdate, empid
FROM Sales.Orders AS O1
WHERE orderdate =(SELECT MAX(O2.orderdate)
    FROM Sales.Orders AS O2
    WHERE O2.custid = O1.custid)
ORDER BY custid;

USE Northwinds2022TSQLV7;
GO
SELECT CustomerId, orderid, orderdate, EmployeeId
FROM Sales.[Order] AS O1
WHERE orderdate =(SELECT MAX(O2.orderdate)
    FROM Sales.[Order] AS O2
    WHERE O2.CustomerId = O1.CustomerId)
ORDER BY CustomerId;
```

Results | Messages

	custid	orderid	orderdate	empid
1	1	11011	2016-04-09	3
2	2	10926	2016-03-04	4
3	3	10856	2016-01-28	3
4	4	11016	2016-04-10	9
5	5	10924	2016-03-04	3
6	6	11058	2016-04-29	9
7	7	10826	2016-01-12	6
8	8	10970	2016-03-24	9
9	9	11076	2016-05-06	4
10	10	11048	2016-04-24	7

	CustomerId	orderid	orderdate	EmployeeId
1	1	11011	2016-04-09	3
2	2	10926	2016-03-04	4
3	3	10856	2016-01-28	3
4	4	11016	2016-04-10	9
5	5	10924	2016-03-04	3
6	6	11058	2016-04-29	9

```sql
-- -- 6
-- -- Write a query that returns customers
-- -- who placed orders in 2015 but not in 2016
-- -- Tables involved: TSQLV4 database, Customers and Orders tables

-- -- Desired output:
-- custid      companyname
-- ----------- ----------------------------------------
-- 21          Customer KIDPX
-- 23          Customer WVFAF
-- 33          Customer FVXPQ
-- 36          Customer LVJSO
-- 43          Customer UISOJ
-- 51          Customer PVDZC
-- 85          Customer ENQZT

-- (7 row(s) affected)

USE TSQLV4;

SELECT custid, companyname
FROM Sales.Customers AS C
WHERE EXISTS
(SELECT *
FROM Sales.Orders AS O
WHERE O.custid = C.custid
AND O.orderdate >= '20150101'
AND O.orderdate < '20160101') AND NOT EXISTS
(SELECT *
FROM Sales.Orders AS O
WHERE O.custid = C.custid
AND O.orderdate >= '20160101' AND O.orderdate < '20170101');


USE Northwinds2022TSQLV7;

GO
SELECT CustomerId, CustomerCompanyName
FROM Sales.Customer AS C
WHERE EXISTS
(SELECT *
FROM Sales.[Order] AS O
WHERE O.CustomerId = C.CustomerId
AND O.orderdate >= '20150101'
AND O.orderdate < '20160101') AND NOT EXISTS
(SELECT *
FROM Sales.[Order] AS O
WHERE O.CustomerId = C.CustomerId
AND O.orderdate >= '20160101' AND O.orderdate < '20170101');
```

Results | Messages

	custid	companyname
1	21	Customer KIDPX
2	23	Customer WVFAF
3	33	Customer FVXPQ
4	36	Customer LVJSO
5	43	Customer UISOJ
6	51	Customer PVDZC
7	85	Customer ENQZT

	CustomerId	CustomerCompanyName
1	21	Customer KIDPX
2	23	Customer WVFAF

```
-- 7 (Optional, Advanced)
-- Write a query that returns customers
-- who ordered product 12
-- Tables involved: TSQLV4 database,
-- Customers, Orders and OrderDetails tables

-- -- Desired output:
-- custid      companyname
-- ----------- ----------------------------------------
-- 48          Customer DVFMB
-- 39          Customer GLLAG
-- 71          Customer LCOUJ
-- 65          Customer NYUHS
-- 44          Customer OXFRU
-- 51          Customer PVDZC
-- 86          Customer SNXOJ
-- 20          Customer THHDP
-- 90          Customer XBBVR
-- 46          Customer XPNIK
-- 31          Customer YJCBX
-- 87          Customer ZHYOS

-- (12 row(s) affected)

USE TSQLV4;

SELECT custid, companyname
FROM Sales.Customers AS C
WHERE EXISTS
(SELECT *
FROM Sales.Orders AS O
WHERE O.custid = C.custid AND EXISTS
(SELECT *
FROM Sales.OrderDetails AS OD
WHERE OD.orderid = O.orderid
AND OD.ProductID = 12))
ORDER BY companyname ASC;

USE Northwinds2022TSQLV7;
GO
SELECT CustomerId, CustomerCompanyName
FROM Sales.Customer AS C
WHERE EXISTS
(SELECT *
FROM Sales.[Order] AS O
WHERE O.CustomerId = C.CustomerId AND EXISTS
(SELECT *
FROM Sales.OrderDetail AS OD
WHERE OD.orderid = O.orderid
AND OD.ProductID = 12))
ORDER BY CustomerCompanyName ASC;
```

Results Messages

	custid	companyname
1	48	Customer DVFMB
2	39	Customer GLLAG
3	71	Customer LCOUJ
4	65	Customer NYUHS
5	44	Customer OXFRU
6	51	Customer PVDZC
7	86	Customer SNXOJ
8	20	Customer THHDP
9	90	Customer XBBVR
10	46	Customer XPNIK
11	31	Customer YJCBX
12	87	Customer ZHYOS

	CustomerId	CustomerCompanyName
1	48	Customer DVFMB
2	39	Customer GLLAG
3	71	Customer LCOUJ
4	65	Customer NYUHS
5	44	Customer OXFRU
6	51	Customer PVDZC
7	86	Customer SNXOJ
8	20	Customer THHDP
9	90	Customer XBBVR
10	46	Customer XPNIK
11	31	Customer YJCBX
12	87	Customer ZHYOS

```sql
-- 8 (Optional, Advanced)
-- Write a query that calculates a running total qty
-- for each customer and month using subqueries
-- Tables involved: TSQLV4 database, Sales.CustOrders view

-- -- Desired output:
-- custid      ordermonth              qty         runqty
-- ----------- ----------------------- ----------- -----------
-- 1           2015-08-01 00:00:00.000 38          38
-- 1           2015-10-01 00:00:00.000 41          79
-- 1           2016-01-01 00:00:00.000 17          96
-- 1           2016-03-01 00:00:00.000 18          114
-- 1           2016-04-01 00:00:00.000 60          174
-- 2           2014-09-01 00:00:00.000 6           6
-- 2           2015-08-01 00:00:00.000 18          24
-- 2           2015-11-01 00:00:00.000 10          34
-- 2           2016-03-01 00:00:00.000 29          63
-- 3           2014-11-01 00:00:00.000 24          24
-- 3           2015-04-01 00:00:00.000 30          54
-- 3           2015-05-01 00:00:00.000 80          134
-- 3           2015-06-01 00:00:00.000 83          217
-- 3           2015-09-01 00:00:00.000 102         319
-- 3           2016-01-01 00:00:00.000 40          359
-- ...

-- (636 row(s) affected)

USE TSQLV4;

SELECT custid, ordermonth, qty, (SELECT SUM(O2.qty)
FROM Sales.CustOrders AS O2
    WHERE O2.custid = O1.custid
    AND O2.ordermonth <= O1.ordermonth) AS runqty
FROM Sales.CustOrders AS O1
ORDER BY custid, ordermonth;
-------------------------------------------------------
-- I couldn't figure out how to create the view for this exercise to work
-- with Northwinds and Northwinds did not have the view already made,
-- but this is what I tried to create the view :

-- error message is:

-- Started executing query at Line 370
-- Msg 1088, Level 16, State 18, Procedure CustOrder, Line 5
-- Cannot find the object "Order" because it does not exist or you do not have permissions.

-- USE Northwinds2022TSQLV7;
-- GO
-- SET ANSI_NULLS ON
-- GO
-- SET QUOTED_IDENTIFIER ON
-- GO

-- CREATE VIEW [Sales].[CustOrder]
--   WITH SCHEMABINDING
-- AS

-- SELECT
--   O.CustomerId,
--   DATEADD(month, DATEDIFF(month, CAST('19000101' AS DATE), O.OrderDate), CAST('19000101' AS DATE)) AS OrderMonth,
--   SUM(OD.Quantity) AS qty
-- FROM Sales.[Order] AS O
--   JOIN Sales.OrderDetail AS OD
--     ON OD.OrderId = O.OrderId
-- GROUP BY CustomerId, DATEADD(month, DATEDIFF(month, CAST('19000101' AS DATE), O.OrderDate), CAST('19000101' AS DATE));
-- GO
```

```
414
415
416   -- 9
417   -- Explain the difference between IN and EXISTS
418
419   -- The IN predicate uses three-valued logic, the EXISTS predicate uses two-valued logic, meaning
420   -- EXISTS does not recognize the 'unknown' category which is important when your data has Null values.
421   -- In the absense of a value, ie a null value, NOT IN, the negation of IN, returns unknown,
422   -- where NOT EXISTS returns TRUE.
423
424
425
426   -- 10 (Optional, Advanced)
```

```
426  -- 10 (Optional, Advanced)
427  -- Write a query that returns for each order the number of days that past
428  -- since the same customer's previous order. To determine recency among orders,
429  -- use orderdate as the primary sort element and orderid as the tiebreaker.
430  -- Tables involved: TSQLV4 database, Sales.Orders table
431
432  -- Desired output:
433  -- --custid      orderdate  orderid     diff
434  -- ----------- ---------- ----------- -----------
435  -- 1           2015-08-25 10643       NULL
436  -- 1           2015-10-03 10692       39
437  -- 1           2015-10-13 10702       10
438  -- 1           2016-01-15 10835       94
439  -- 1           2016-03-16 10952       61
440  -- 1           2016-04-09 11011       24
441  -- 2           2014-09-18 10308       NULL
442  -- 2           2015-08-08 10625       324
443  -- 2           2015-11-28 10759       112
444  -- 2           2016-03-04 10926       97
445  -- ...
446
447  -- (830 row(s) affected)
448
449  USE TSQLV4;
450
451   SELECT custid, orderdate, orderid,
452      DATEDIFF(day,
453  (SELECT TOP (1) O2.orderdate
454  FROM Sales.Orders AS O2
455  WHERE O2.custid = O1.custid
456  AND ( O2.orderdate = O1.orderdate AND O2.orderid < O1.orderid OR O2.orderdate < O1.orderdate )
457  ORDER BY O2.orderdate DESC, O2.orderid DESC), orderdate) AS diff
458  FROM Sales.Orders AS O1
459  ORDER BY custid, orderdate, orderid;
460
461  USE Northwinds2022TSQLV7;
462  GO
463   SELECT CustomerId, orderdate, orderid,
464      DATEDIFF(day,
465  (SELECT TOP (1) O2.orderdate
466  FROM Sales.[Order] AS O2
467  WHERE O2.CustomerId = O1.CustomerId
468  AND ( O2.orderdate = O1.orderdate AND O2.orderid < O1.orderid OR O2.orderdate < O1.orderdate )
469  ORDER BY O2.orderdate DESC, O2.orderid DESC), orderdate) AS diff
470  FROM Sales.[Order] AS O1
471  ORDER BY CustomerId, orderdate, orderid;
```

Results Messages

	custid	orderdate	orderid	diff
1	1	2015-08-25	10643	*NULL*
2	1	2015-10-03	10692	39
3	1	2015-10-13	10702	10
4	1	2016-01-15	10835	94
5	1	2016-03-16	10952	61
6	1	2016-04-09	11011	24
7	2	2014-09-18	10308	*NULL*
8	2	2015-08-08	10625	324
9	2	2015-11-28	10759	112
10	2	2016-03-04	10926	97

	CustomerId	orderdate	orderid	diff
1	1	2015-08-25	10643	*NULL*
2	1	2015-10-03	10692	39