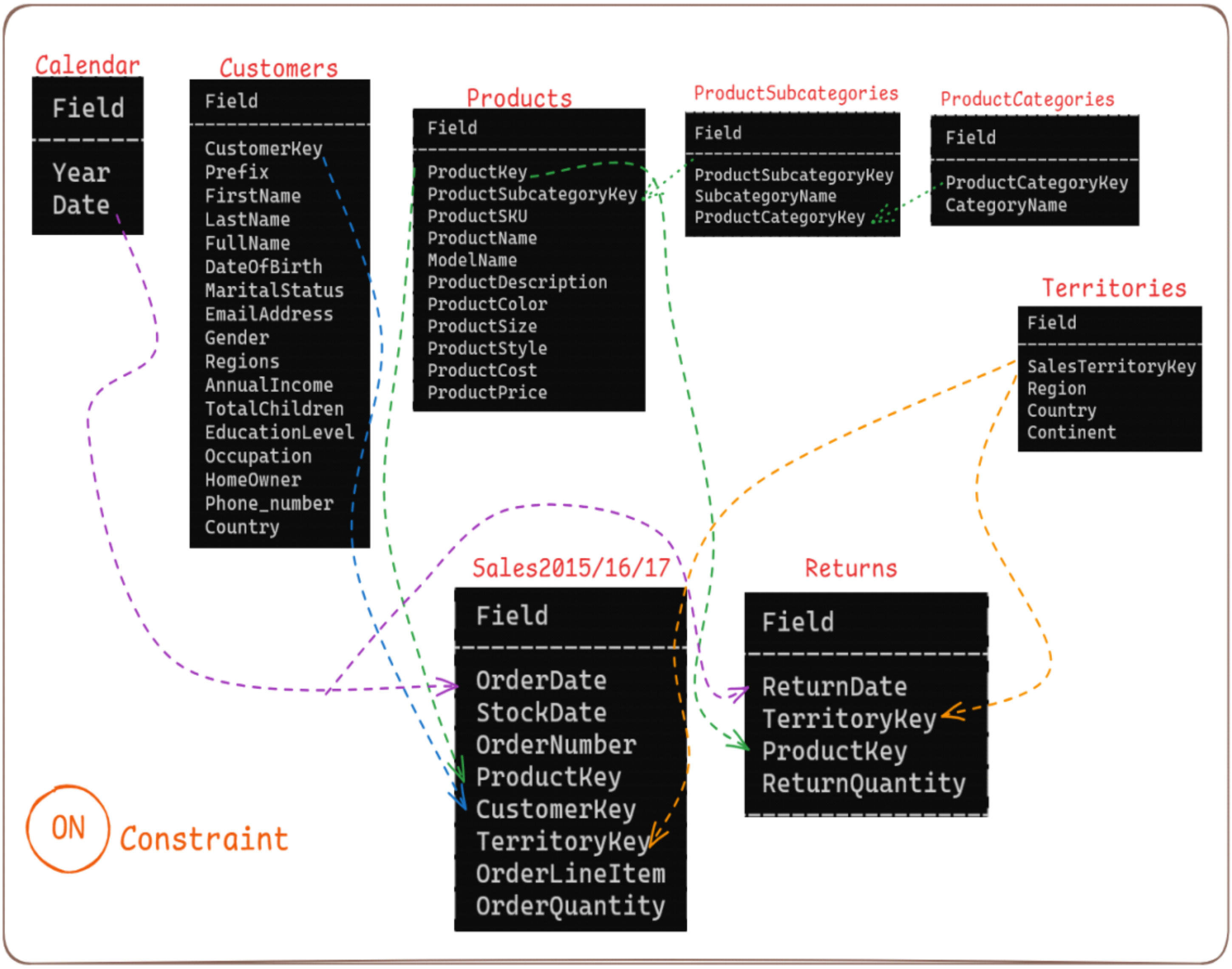
Mastering Subqueries & CTEs

Session Objectives:

- ☑ Understand Common Table Expressions (CTEs) and why we use them
- ☑ Apply subqueries in SELECT, FROM, WHERE, HAVING, and JOIN
- ☑ Use nested & correlated subqueries for advanced querying
- ☑ Optimize queries using subqueries



Calendar

Field
Year
Date

Customers

Field
CustomerKey
Prefix
FirstName
LastName
FullName
DateOfBirth
MaritalStatus
EmailAddress
Gender
Regions
AnnualIncome
TotalChildren
EducationLevel
Occupation
HomeOwner
Phone_number
Country

Products

Field
ProductKey
ProductSubcategoryKey
ProductSKU
ProductName
ModelName
ProductDescription
ProductColor
ProductSize
ProductStyle
ProductCost
ProductPrice

ProductSubcategories

Field
ProductSubcategoryKey
SubcategoryName
ProductCategoryKey

ProductCategories

Field
ProductCategoryKey
CategoryName

Territories

Field
SalesTerritoryKey
Region
Country
Continent

Sales2015/16/17

Field
OrderDate
StockDate
OrderNumber
ProductKey
CustomerKey
TerritoryKey
OrderLineItem
OrderQuantity

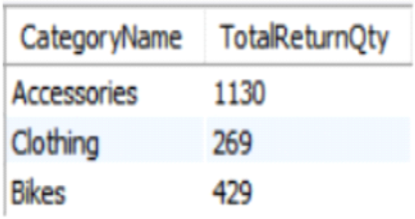
Returns

Field
ReturnDate
TerritoryKey
ProductKey
ReturnQuantity

ON Constraint

With CategoryReturns

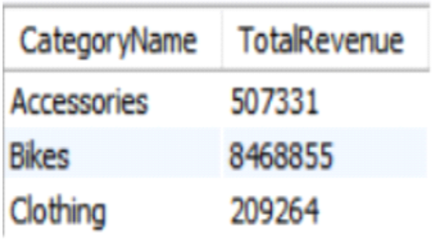
```
Select
    pc.CategoryName,
    SUM(r.ReturnQuantity) AS TotalReturnQty
FROM ProductCategories pc
JOIN ProductSubcategories ps
ON pc.ProductCategoryKey = ps.ProductCategoryKey
JOIN Products p
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN Returns r
ON r.ProductKey = p.ProductKey
GROUP BY 1;
```



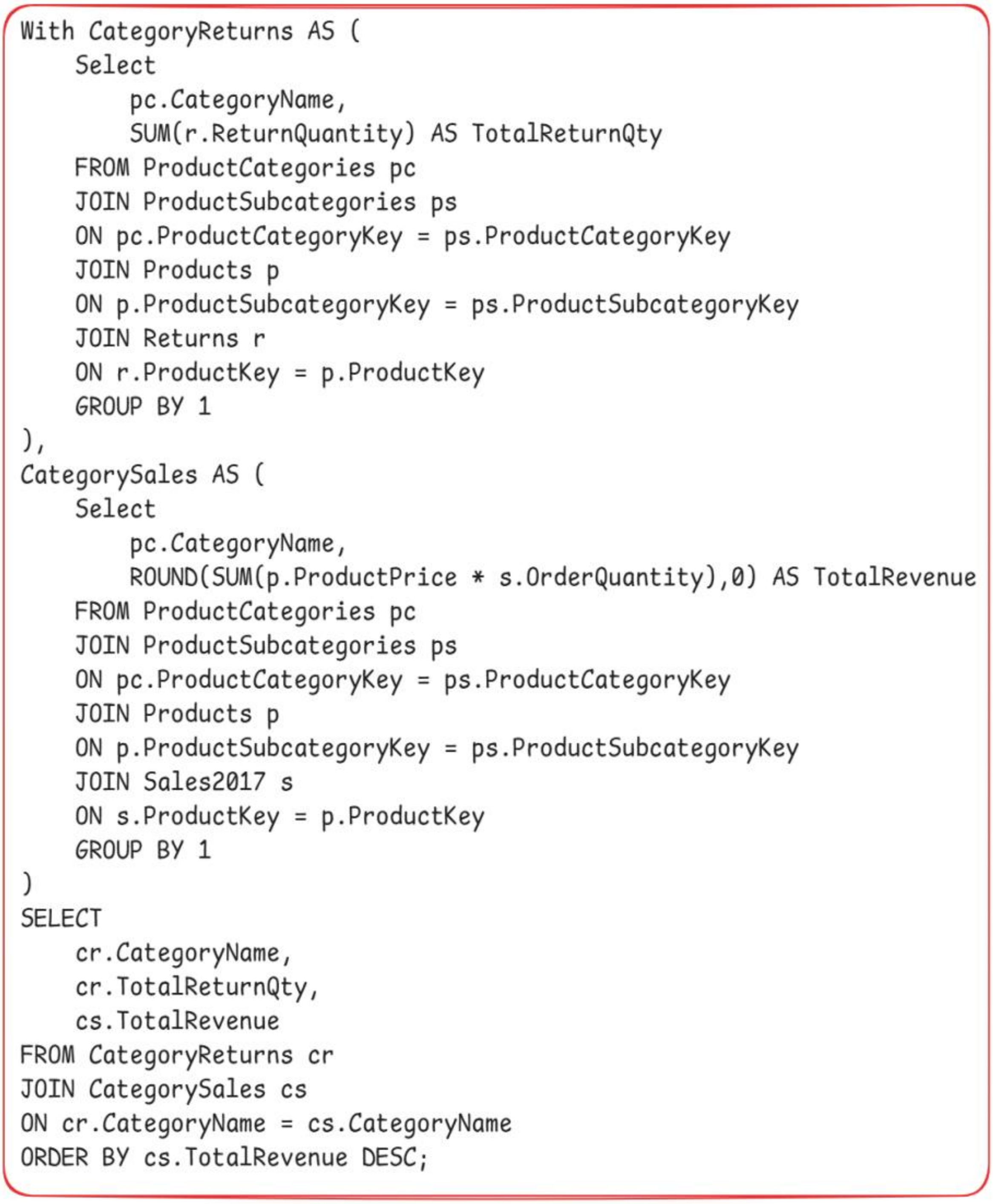
CategoryName	TotalReturnQty
Accessories	1130
Clothing	269
Bikes	429

With CategoryRevenue

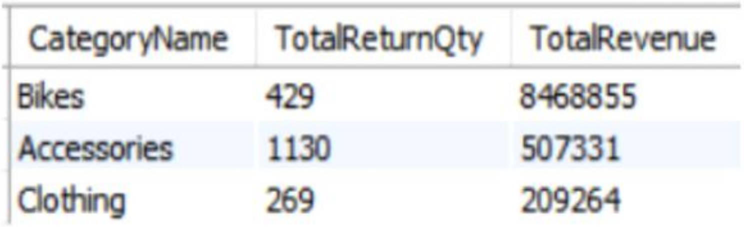
```
Select
    pc.CategoryName,
    ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue
FROM ProductCategories pc
JOIN ProductSubcategories ps
ON pc.ProductCategoryKey = ps.ProductCategoryKey
JOIN Products p
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
JOIN Sales2017 s
ON s.ProductKey = p.ProductKey
GROUP BY 1;
```



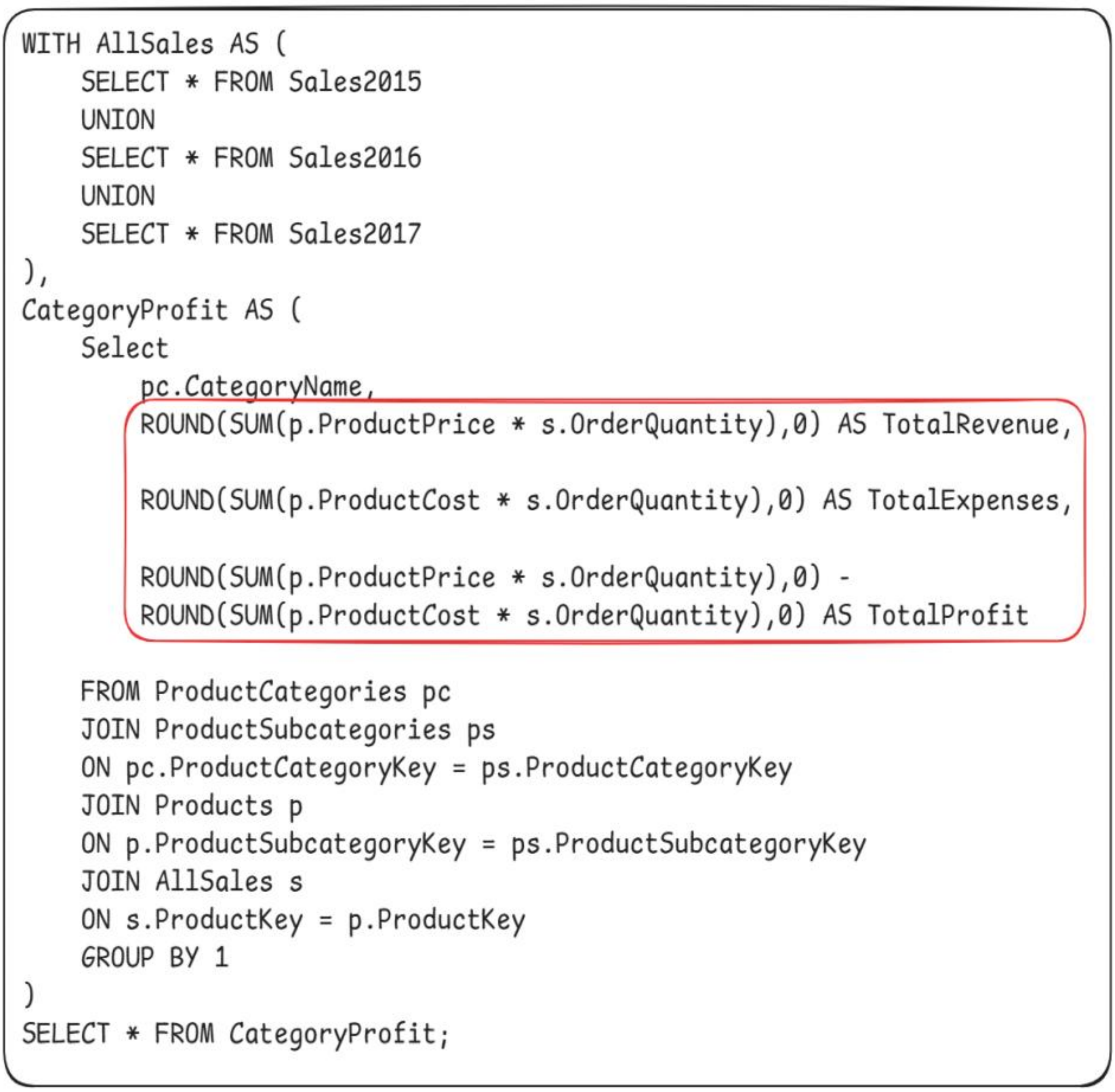
CategoryName	TotalRevenue
Accessories	507331
Bikes	8468855
Clothing	209264



```
With CategoryReturns AS (
    Select
        pc.CategoryName,
        SUM(r.ReturnQuantity) AS TotalReturnQty
    FROM ProductCategories pc
    JOIN ProductSubcategories ps
    ON pc.ProductCategoryKey = ps.ProductCategoryKey
    JOIN Products p
    ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
    JOIN Returns r
    ON r.ProductKey = p.ProductKey
    GROUP BY 1
),
CategorySales AS (
    Select
        pc.CategoryName,
        ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue
    FROM ProductCategories pc
    JOIN ProductSubcategories ps
    ON pc.ProductCategoryKey = ps.ProductCategoryKey
    JOIN Products p
    ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
    JOIN Sales2017 s
    ON s.ProductKey = p.ProductKey
    GROUP BY 1
)
SELECT
    cr.CategoryName,
    cr.TotalReturnQty,
    cs.TotalRevenue
FROM CategoryReturns cr
JOIN CategorySales cs
ON cr.CategoryName = cs.CategoryName
ORDER BY cs.TotalRevenue DESC;
```



CategoryName	TotalReturnQty	TotalRevenue
Bikes	429	8468855
Accessories	1130	507331
Clothing	269	209264

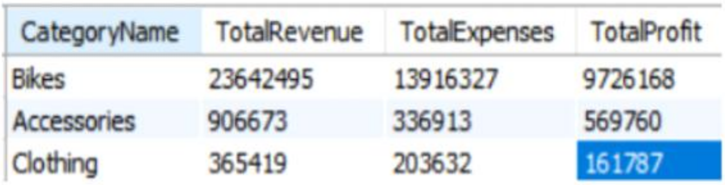


```
WITH AllSales AS (
    SELECT * FROM Sales2015
    UNION
    SELECT * FROM Sales2016
    UNION
    SELECT * FROM Sales2017
),
CategoryProfit AS (
    Select
        pc.CategoryName,
        ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue,

        ROUND(SUM(p.ProductCost * s.OrderQuantity),0) AS TotalExpenses,

        ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) -
        ROUND(SUM(p.ProductCost * s.OrderQuantity),0) AS TotalProfit

    FROM ProductCategories pc
    JOIN ProductSubcategories ps
    ON pc.ProductCategoryKey = ps.ProductCategoryKey
    JOIN Products p
    ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
    JOIN AllSales s
    ON s.ProductKey = p.ProductKey
    GROUP BY 1
)
SELECT * FROM CategoryProfit;
```



CategoryName	TotalRevenue	TotalExpenses	TotalProfit
Bikes	23642495	13916327	9726168
Accessories	906673	336913	569760
Clothing	365419	203632	161787

```
WITH AllSales AS (
    SELECT * FROM Sales2015
    UNION
    SELECT * FROM Sales2016
    UNION
    SELECT * FROM Sales2017
),
CategoryProfit AS (
    Select
        pc.CategoryName,
        ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue,
        ROUND(SUM(p.ProductCost * s.OrderQuantity),0) AS TotalExpenses
    FROM ProductCategories pc
    JOIN ProductSubcategories ps
    ON pc.ProductCategoryKey = ps.ProductCategoryKey
    JOIN Products p
    ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
    JOIN AllSales s
    ON s.ProductKey = p.ProductKey
    GROUP BY 1
)
SELECT
    *,
    (TotalRevenue - TotalExpenses) AS TotalProfit
FROM CategoryProfit;
```

SUBQUERIES

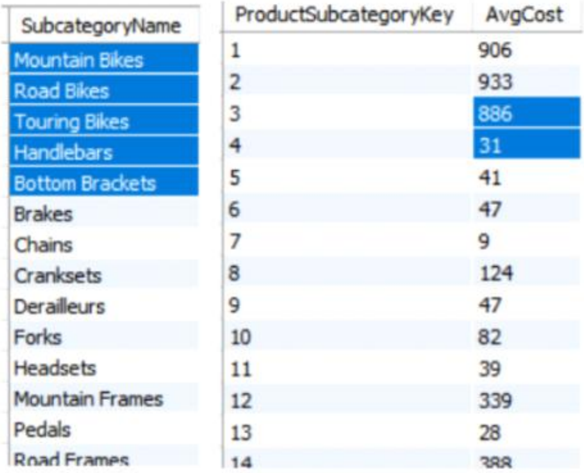
A Subquery is a query nested inside another query [Inner Query].
It returns data used by other query.

-- You can use that subquery in : [SELECT , FROM , WHERE , HAVING, JOIN]

Subquery in Select Clause

```
SELECT
    ProductSubcategoryKey,
    ROUND(AVG(ProductCost),0) AS AvgCost
FROM Products
GROUP BY 1;
```

```
SELECT
    SubcategoryName
FROM ProductSubcategories;
```

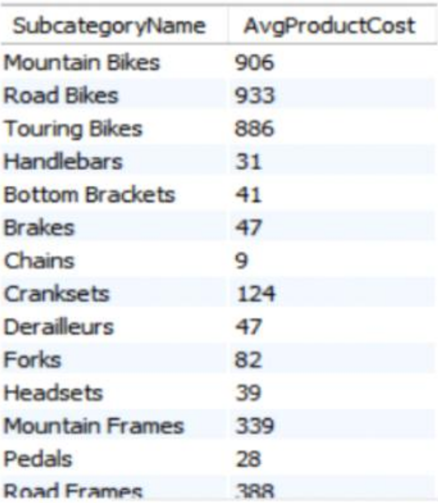


SubcategoryName
Mountain Bikes
Road Bikes
Touring Bikes
Handlebars
Bottom Brackets
Brakes
Chains
Cranksets
Derailleurs
Forks
Headsets
Mountain Frames
Pedals
Road Frames

ProductSubcategoryKey	AvgCost
1	906
2	933
3	886
4	31
5	41
6	47
7	9
8	124
9	47
10	82
11	39
12	339
13	28
14	388

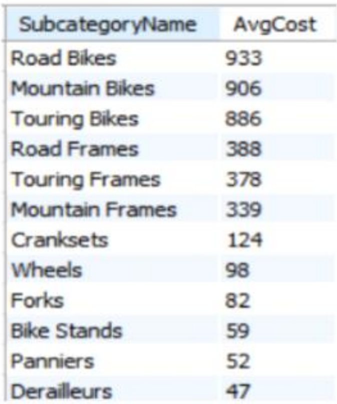
Show Each subcategories with its Average Product Cost

```
SELECT
    ps.SubcategoryName,
    (
        SELECT
            ROUND(AVG(ProductCost),0) AS AvgCost
        FROM Products p
        WHERE p.ProductSubcategoryKey = ps.ProductSubcategoryKey
    ) AS AvgProductCost
FROM ProductSubcategories ps;
```



SubcategoryName	AvgProductCost
Mountain Bikes	906
Road Bikes	933
Touring Bikes	886
Handlebars	31
Bottom Brackets	41
Brakes	47
Chains	9
Cranksets	124
Derailleurs	47
Forks	82
Headsets	39
Mountain Frames	339
Pedals	28
Road Frames	388

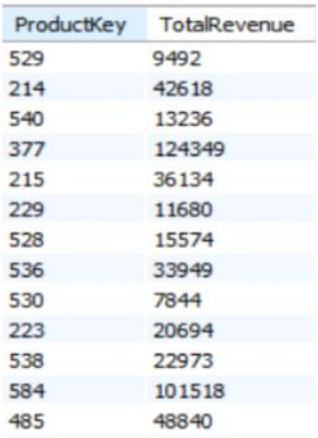
```
SELECT
    ps.SubcategoryName,
    ROUND(AVG(ProductCost),0) AS AvgCost
FROM Products p
JOIN
ProductSubcategories ps
ON p.ProductSubcategoryKey = ps.ProductSubcategoryKey
GROUP BY 1;
```



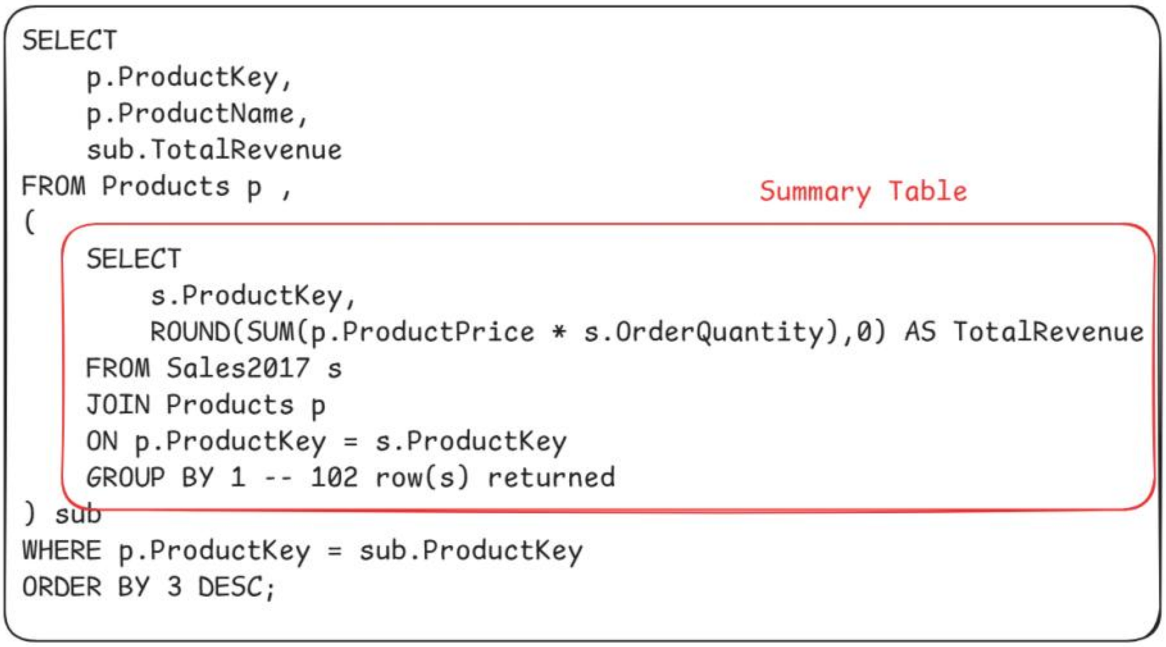
SubcategoryName	AvgCost
Road Bikes	933
Mountain Bikes	906
Touring Bikes	886
Road Frames	388
Touring Frames	378
Mountain Frames	339
Cranksets	124
Wheels	98
Forks	82
Bike Stands	59
Panniers	52
Derailleurs	47

Subquery in FROM Clause

```
SELECT
    p.ProductKey,
    ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue
FROM Sales2017 s
JOIN Products p
ON p.ProductKey = s.ProductKey
GROUP BY 1;
```

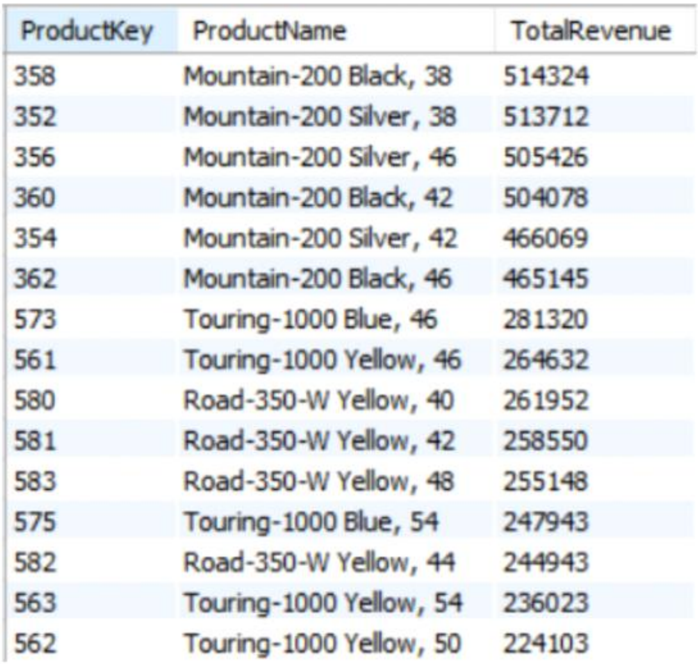


ProductKey	TotalRevenue
529	9492
214	42618
540	13236
377	124349
215	36134
229	11680
528	15574
536	33949
530	7844
223	20694
538	22973
584	101518
485	48840



```
SELECT
    p.ProductKey,
    p.ProductName,
    sub.TotalRevenue
FROM Products p ,
(
    SELECT
        s.ProductKey,
        ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue
    FROM Sales2017 s
    JOIN Products p
    ON p.ProductKey = s.ProductKey
    GROUP BY 1 -- 102 row(s) returned
) sub
WHERE p.ProductKey = sub.ProductKey
ORDER BY 3 DESC;
```

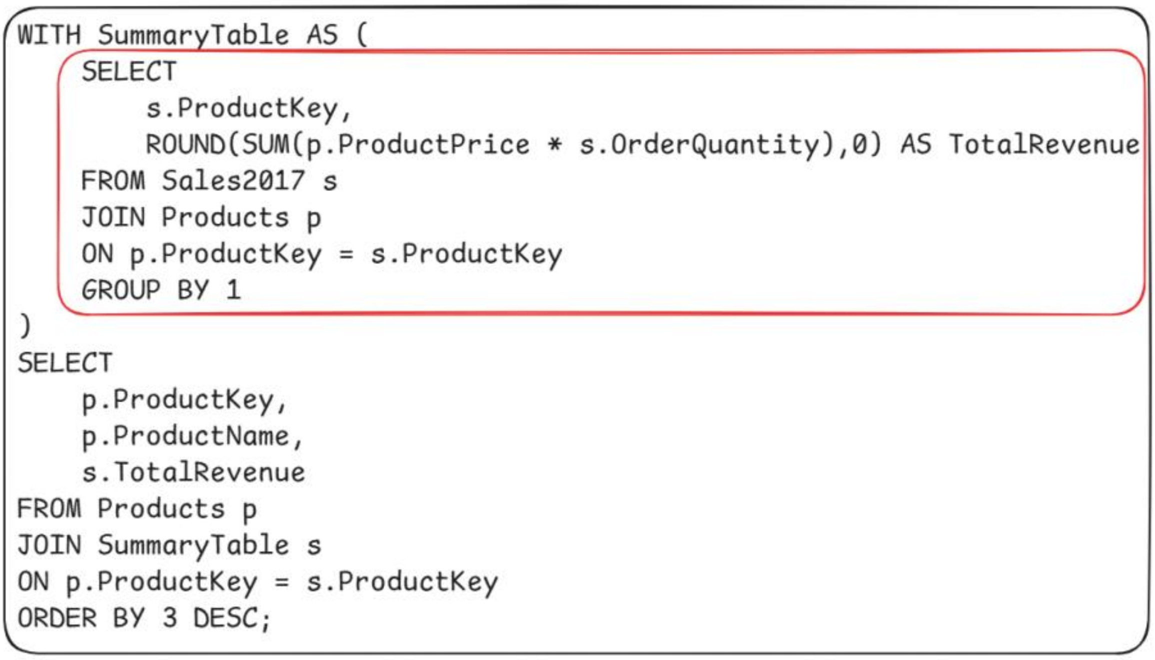
Summary Table



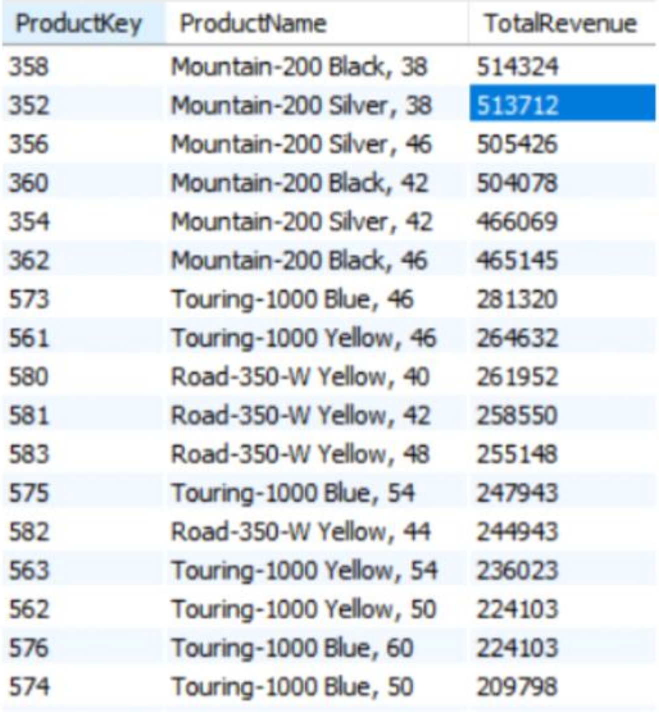
ProductKey	ProductName	TotalRevenue
358	Mountain-200 Black, 38	514324
352	Mountain-200 Silver, 38	513712
356	Mountain-200 Silver, 46	505426
360	Mountain-200 Black, 42	504078
354	Mountain-200 Silver, 42	466069
362	Mountain-200 Black, 46	465145
573	Touring-1000 Blue, 46	281320
561	Touring-1000 Yellow, 46	264632
580	Road-350-W Yellow, 40	261952
581	Road-350-W Yellow, 42	258550
583	Road-350-W Yellow, 48	255148
575	Touring-1000 Blue, 54	247943
582	Road-350-W Yellow, 44	244943
563	Touring-1000 Yellow, 54	236023
562	Touring-1000 Yellow, 50	224103

Note:

Both The code with CTE or subquery is equivalent.



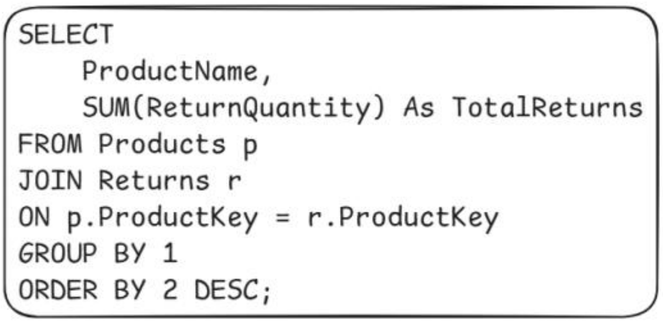
```
WITH SummaryTable AS (
    SELECT
        s.ProductKey,
        ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue
    FROM Sales2017 s
    JOIN Products p
    ON p.ProductKey = s.ProductKey
    GROUP BY 1
)
SELECT
    p.ProductKey,
    p.ProductName,
    s.TotalRevenue
FROM Products p
JOIN SummaryTable s
ON p.ProductKey = s.ProductKey
ORDER BY 3 DESC;
```



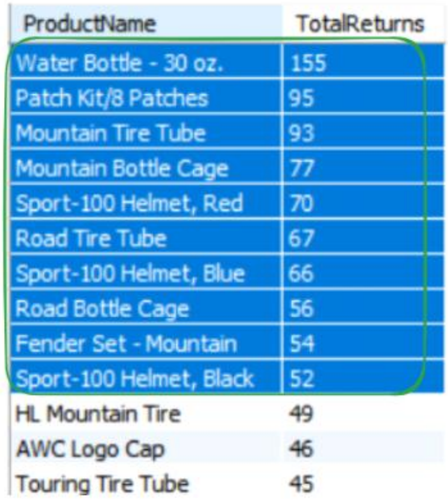
ProductKey	ProductName	TotalRevenue
358	Mountain-200 Black, 38	514324
352	Mountain-200 Silver, 38	513712
356	Mountain-200 Silver, 46	505426
360	Mountain-200 Black, 42	504078
354	Mountain-200 Silver, 42	466069
362	Mountain-200 Black, 46	465145
573	Touring-1000 Blue, 46	281320
561	Touring-1000 Yellow, 46	264632
580	Road-350-W Yellow, 40	261952
581	Road-350-W Yellow, 42	258550
583	Road-350-W Yellow, 48	255148
575	Touring-1000 Blue, 54	247943
582	Road-350-W Yellow, 44	244943
563	Touring-1000 Yellow, 54	236023
562	Touring-1000 Yellow, 50	224103
576	Touring-1000 Blue, 60	224103
574	Touring-1000 Blue, 50	209798

SUBQUERY IN WHERE CLAUSE

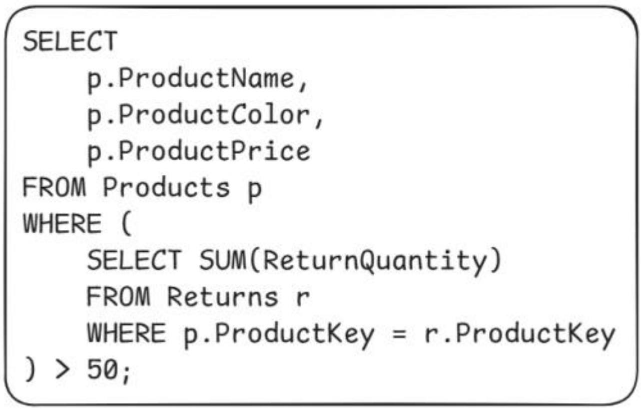
Products that are returned more than 50 times.



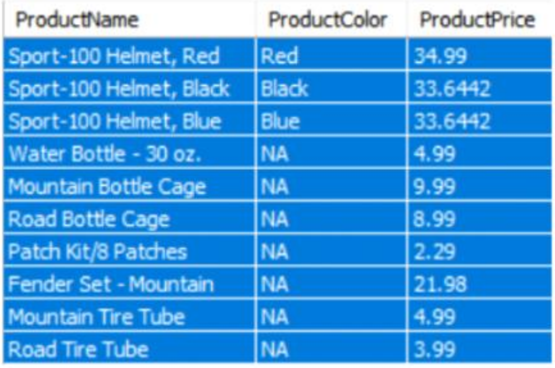
```
SELECT
    ProductName,
    SUM(ReturnQuantity) As TotalReturns
FROM Products p
JOIN Returns r
ON p.ProductKey = r.ProductKey
GROUP BY 1
ORDER BY 2 DESC;
```



ProductName	TotalReturns
Water Bottle - 30 oz.	155
Patch Kit/8 Patches	95
Mountain Tire Tube	93
Mountain Bottle Cage	77
Sport-100 Helmet, Red	70
Road Tire Tube	67
Sport-100 Helmet, Blue	66
Road Bottle Cage	56
Fender Set - Mountain	54
Sport-100 Helmet, Black	52
HL Mountain Tire	49
AWC Logo Cap	46
Touring Tire Tube	45



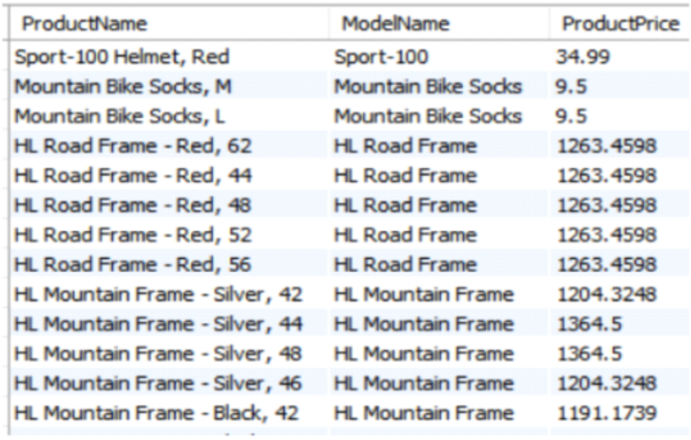
```
SELECT
    p.ProductName,
    p.ProductColor,
    p.ProductPrice
FROM Products p
WHERE (
    SELECT SUM(ReturnQuantity)
    FROM Returns r
    WHERE p.ProductKey = r.ProductKey
) > 50;
```



ProductName	ProductColor	ProductPrice
Sport-100 Helmet, Red	Red	34.99
Sport-100 Helmet, Black	Black	33.6442
Sport-100 Helmet, Blue	Blue	33.6442
Water Bottle - 30 oz.	NA	4.99
Mountain Bottle Cage	NA	9.99
Road Bottle Cage	NA	8.99
Patch Kit/8 Patches	NA	2.29
Fender Set - Mountain	NA	21.98
Mountain Tire Tube	NA	4.99
Road Tire Tube	NA	3.99

Find the Products priced above their average p.price in productsubcategory

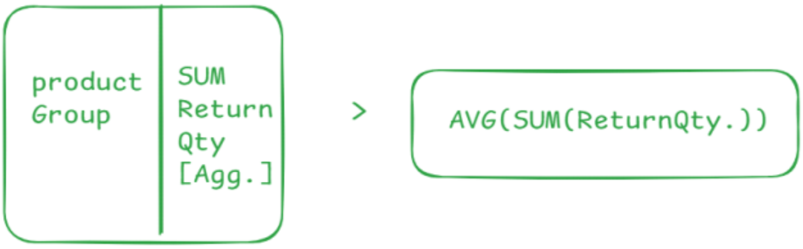
```
SELECT
    p.ProductName,
    p.ModelName,
    p.ProductPrice
FROM Products p
WHERE p.ProductPrice > (
    SELECT AVG(p2.ProductPrice)
    FROM Products p2
    WHERE p2.ProductSubcategoryKey = p.ProductSubcategoryKey
); -- 115 row(s) returned
```



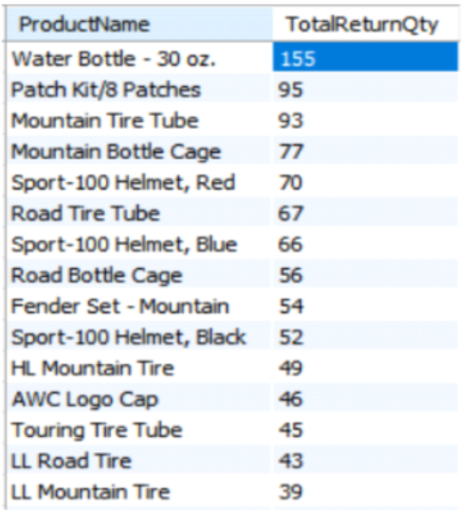
ProductName	ModelName	ProductPrice
Sport-100 Helmet, Red	Sport-100	34.99
Mountain Bike Socks, M	Mountain Bike Socks	9.5
Mountain Bike Socks, L	Mountain Bike Socks	9.5
HL Road Frame - Red, 62	HL Road Frame	1263.4598
HL Road Frame - Red, 44	HL Road Frame	1263.4598
HL Road Frame - Red, 48	HL Road Frame	1263.4598
HL Road Frame - Red, 52	HL Road Frame	1263.4598
HL Road Frame - Red, 56	HL Road Frame	1263.4598
HL Mountain Frame - Silver, 42	HL Mountain Frame	1204.3248
HL Mountain Frame - Silver, 44	HL Mountain Frame	1364.5
HL Mountain Frame - Silver, 48	HL Mountain Frame	1364.5
HL Mountain Frame - Silver, 46	HL Mountain Frame	1204.3248
HL Mountain Frame - Black, 42	HL Mountain Frame	1191.1739

SUBQUERY IN HAVING CLAUSE

Find the Products returned more than the average return quantity

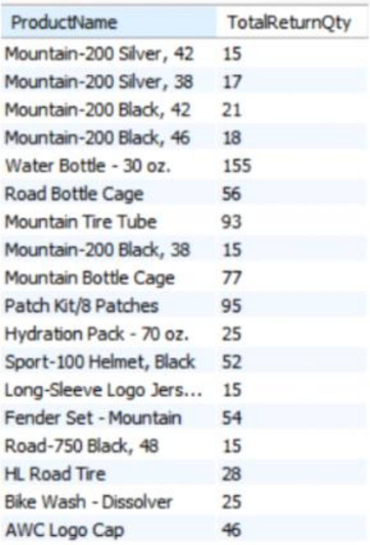


```
SELECT
    p.ProductName,
    SUM(r.ReturnQuantity) as TotalReturnQty
FROM Products p
JOIN Returns r
ON r.ProductKey = p.ProductKey
GROUP BY 1;
```



ProductName	TotalReturnQty
Water Bottle - 30 oz.	155
Patch Kit/8 Patches	95
Mountain Tire Tube	93
Mountain Bottle Cage	77
Sport-100 Helmet, Red	70
Road Tire Tube	67
Sport-100 Helmet, Blue	66
Road Bottle Cage	56
Fender Set - Mountain	54
Sport-100 Helmet, Black	52
HL Mountain Tire	49
AWC Logo Cap	46
Touring Tire Tube	45
LL Road Tire	43
LL Mountain Tire	39

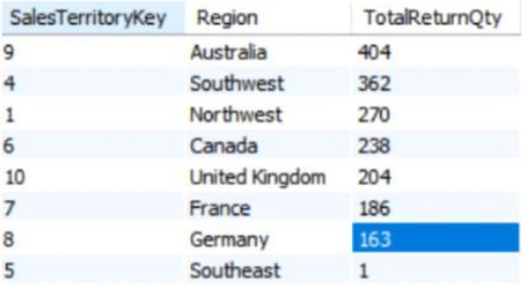
```
SELECT
    p.ProductName,
    SUM(r.ReturnQuantity) as TotalReturnQty
FROM Products p
JOIN Returns r
ON r.ProductKey = p.ProductKey
GROUP BY 1
HAVING SUM(r.ReturnQuantity) > (
    SELECT
        AVG(TotalReturnQty)
    FROM (
        SELECT
            SUM(ReturnQuantity) as TotalReturnQty
        FROM Returns
        GROUP BY ProductKey
    ) sub
); -- 34 row(s) returned
```



ProductName	TotalReturnQty
Mountain-200 Silver, 42	15
Mountain-200 Silver, 38	17
Mountain-200 Black, 42	21
Mountain-200 Black, 46	18
Water Bottle - 30 oz.	155
Road Bottle Cage	56
Mountain Tire Tube	93
Mountain-200 Black, 38	15
Mountain Bottle Cage	77
Patch Kit/8 Patches	95
Hydration Pack - 70 oz.	25
Sport-100 Helmet, Black	52
Long-Sleeve Logo Jers...	15
Fender Set - Mountain	54
Road-750 Black, 48	15
HL Road Tire	28
Bike Wash - Dissolver	25
AWC Logo Cap	46

SUBQUERY IN JOIN CLAUSE

```
SELECT
    t.SalesTerritoryKey,
    t.Region,
    sub.TotalReturnQty
FROM Territories t
JOIN(
    SELECT
        r.TerritoryKey,
        SUM(ReturnQuantity) AS TotalReturnQty
    FROM Returns r
    GROUP BY 1
) sub
ON t.SalesTerritoryKey = sub.TerritoryKey
ORDER BY 3 DESC;
```



SalesTerritoryKey	Region	TotalReturnQty
9	Australia	404
4	Southwest	362
1	Northwest	270
6	Canada	238
10	United Kingdom	204
7	France	186
8	Germany	163
5	Southeast	1

```
WITH TerritoryReturns AS (
    SELECT
        r.TerritoryKey,
    SUM(ReturnQuantity) AS TotalReturnQty
    FROM Returns r
    GROUP BY 1
)
SELECT
    t.SalesTerritoryKey,
    t.Region,
    tr.TotalReturnQty
FROM Territories t
JOIN TerritoryReturns tr
ON t.SalesTerritoryKey = tr.TerritoryKey;
```