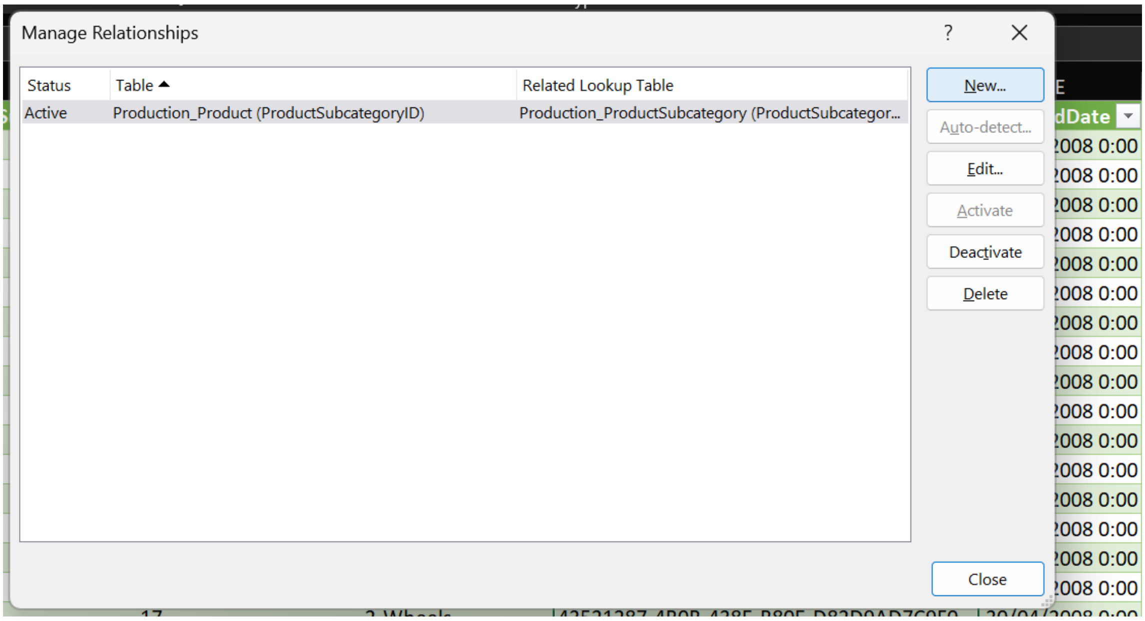
# Abdulrhman Eldeeb – 276838 Lab 1 report

In the upcoming screenshots I will be showing my steps of accomplishing my lab tasks

But there is several steps I did not take screenshot of it:

- bringing data from mySQL server to the excel: I have used ‘Data’ tab from the ribbon and then chose get data from mySQL server.

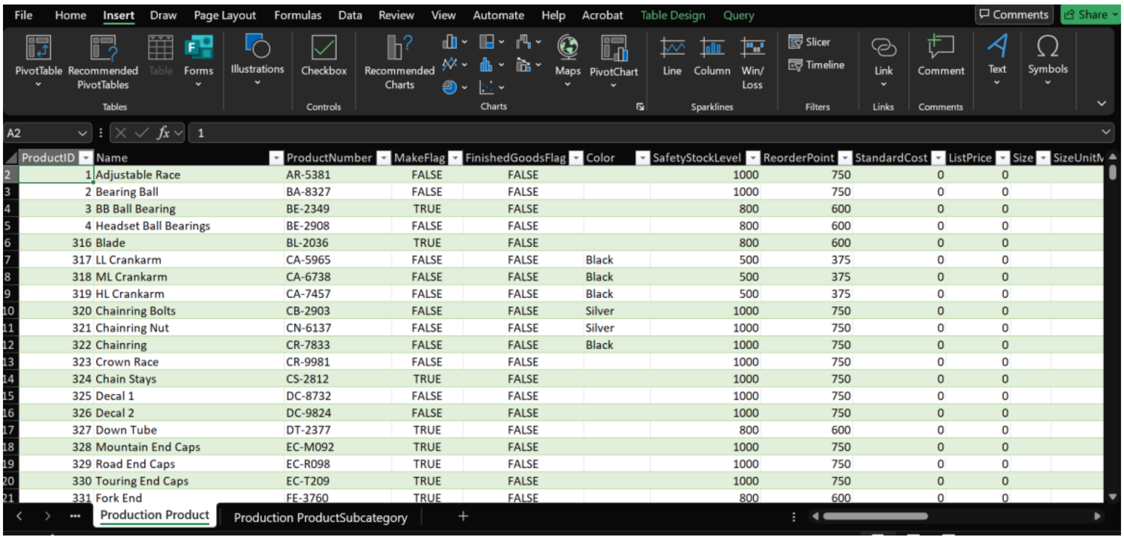
Here I have checked my relationship from the data tab as well in the beginning there was no relationships, so I have created a relationship between the two tables choosing the (ProductSubcategory) as a foreign key from the production.product and choosing the (ProductSubcategory) as a primary key from the production.productSubcategory.



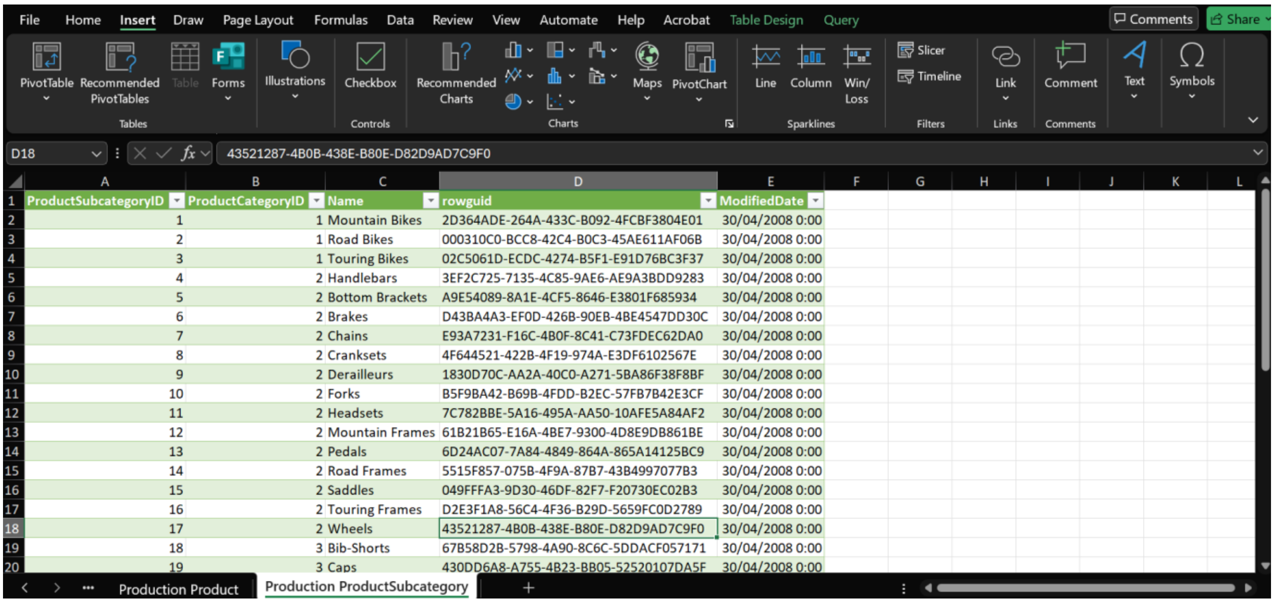
Manage Relationships

| Status | Table ▲ | Related Lookup Table |
|---|---|---|
| Active | Production_Product (ProductSubcategoryID) | Production_ProductSubcategory (ProductSubcategor... |

Here I have my first and second tables shown in the excel sheets



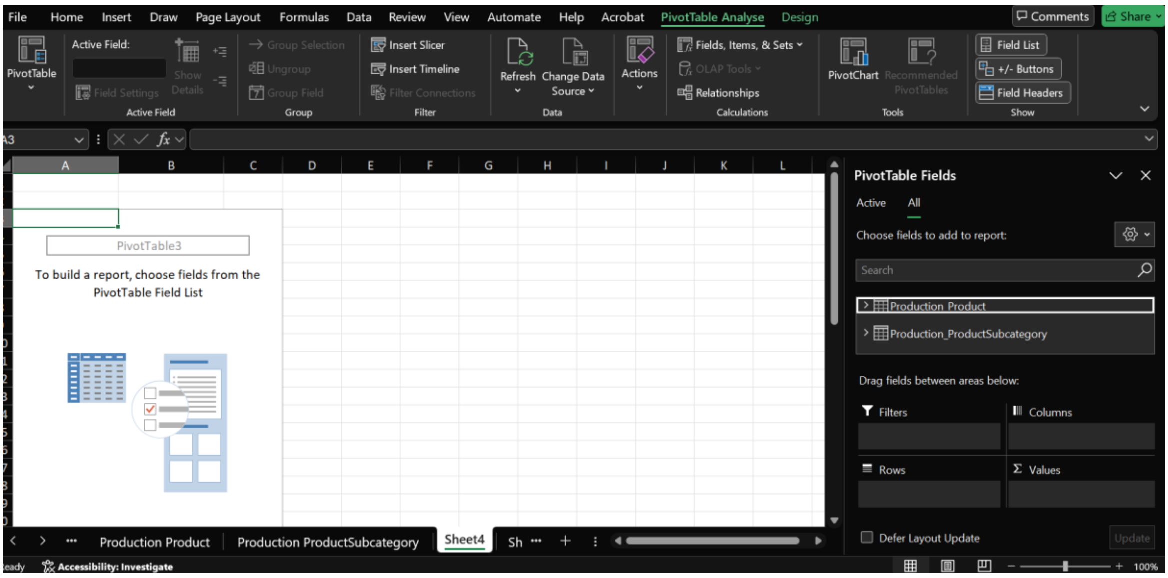
| | ProductID | Name | ProductNumber | MakeFlag | FinishedGoodsFlag | Color | SafetyStockLevel | ReorderPoint | StandardCost | ListPrice | Size | SizeUnitM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | Adjustable Race | AR-5381 | FALSE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 3 | 2 | Bearing Ball | BA-8327 | FALSE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 4 | 3 | BB Ball Bearing | BE-2349 | TRUE | FALSE | | 800 | 600 | 0 | 0 | | |
| 5 | 4 | Headset Ball Bearings | BE-2908 | FALSE | FALSE | | 800 | 600 | 0 | 0 | | |
| 6 | 316 | Blade | BL-2036 | TRUE | FALSE | | 800 | 600 | 0 | 0 | | |
| 7 | 317 | LL Crankarm | CA-5965 | FALSE | FALSE | Black | 500 | 375 | 0 | 0 | | |
| 8 | 318 | ML Crankarm | CA-6738 | FALSE | FALSE | Black | 500 | 375 | 0 | 0 | | |
| 9 | 319 | HL Crankarm | CA-7457 | FALSE | FALSE | Black | 500 | 375 | 0 | 0 | | |
| 10 | 320 | Chainring Bolts | CB-2903 | FALSE | FALSE | Silver | 1000 | 750 | 0 | 0 | | |
| 11 | 321 | Chainring Nut | CN-6137 | FALSE | FALSE | Silver | 1000 | 750 | 0 | 0 | | |
| 12 | 322 | Chainring | CR-7833 | FALSE | FALSE | Black | 1000 | 750 | 0 | 0 | | |
| 13 | 323 | Crown Race | CR-9981 | FALSE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 14 | 324 | Chain Stays | CS-2812 | TRUE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 15 | 325 | Decal 1 | DC-8732 | FALSE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 16 | 326 | Decal 2 | DC-9824 | FALSE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 17 | 327 | Down Tube | DT-2377 | TRUE | FALSE | | 800 | 600 | 0 | 0 | | |
| 18 | 328 | Mountain End Caps | EC-M092 | TRUE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 19 | 329 | Road End Caps | EC-R098 | TRUE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 20 | 330 | Touring End Caps | EC-T209 | TRUE | FALSE | | 1000 | 750 | 0 | 0 | | |
| 21 | 331 | Fork End | FE-3760 | TRUE | FALSE | | 800 | 600 | 0 | 0 | | |



D18 43521287-4B0B-438E-B80E-D82D9AD7C9F0

| ProductSubcategoryID | ProductCategoryID | Name | rowguid | ModifiedDate |
|---|---|---|---|---|
| 1 | 1 | Mountain Bikes | 2D364ADE-264A-433C-B092-4FCBF3804E01 | 30/04/2008 0:00 |
| 2 | 1 | Road Bikes | 000310C0-BCC8-42C4-B0C3-45AE611AF06B | 30/04/2008 0:00 |
| 3 | 1 | Touring Bikes | 02C5061D-ECDC-4274-B5F1-E91D76BC3F37 | 30/04/2008 0:00 |
| 4 | 2 | Handlebars | 3EF2C725-7135-4C85-9AE6-AE9A3BDD9283 | 30/04/2008 0:00 |
| 5 | 2 | Bottom Brackets | A9E54089-8A1E-4CF5-8646-E3801F685934 | 30/04/2008 0:00 |
| 6 | 2 | Brakes | D43BA4A3-EF0D-426B-90EB-4BE4547DD30C | 30/04/2008 0:00 |
| 7 | 2 | Chains | E93A7231-F16C-4B0F-8C41-C73FDEC62DA0 | 30/04/2008 0:00 |
| 8 | 2 | Cranksets | 4F644521-422B-4F19-974A-E3DF6102567E | 30/04/2008 0:00 |
| 9 | 2 | Derailleurs | 1830D70C-AA2A-40C0-A271-5BA86F38F8BF | 30/04/2008 0:00 |
| 10 | 2 | Forks | B5F9BA42-B69B-4FDD-B2EC-57FB7B42E3CF | 30/04/2008 0:00 |
| 11 | 2 | Headsets | 7C782BBE-5A16-495A-AA50-10AFE5A84AF2 | 30/04/2008 0:00 |
| 12 | 2 | Mountain Frames | 61B21B65-E16A-4BE7-9300-4D8E9DB861BE | 30/04/2008 0:00 |
| 13 | 2 | Pedals | 6D24AC07-7A84-4849-864A-865A14125BC9 | 30/04/2008 0:00 |
| 14 | 2 | Road Frames | 5515F857-075B-4F9A-87B7-43B4997077B3 | 30/04/2008 0:00 |
| 15 | 2 | Saddles | 049FFFA3-9D30-46DF-82F7-F20730EC02B3 | 30/04/2008 0:00 |
| 16 | 2 | Touring Frames | D2E3F1A8-56C4-4F36-B29D-5659FC0D2789 | 30/04/2008 0:00 |
| 17 | 2 | Wheels | 43521287-4B0B-438E-B80E-D82D9AD7C9F0 | 30/04/2008 0:00 |
| 18 | 3 | Bib-Shorts | 67B58D2B-5798-4A90-8C6C-5DDACF057171 | 30/04/2008 0:00 |
| 19 | 3 | Caps | 430DD6A8-A755-4B23-BB05-52520107DA5F | 30/04/2008 0:00 |

Then I have inserted a pivot table to a new sheet and added the  two tables to it



PivotTable3

To build a report, choose fields from the PivotTable Field List

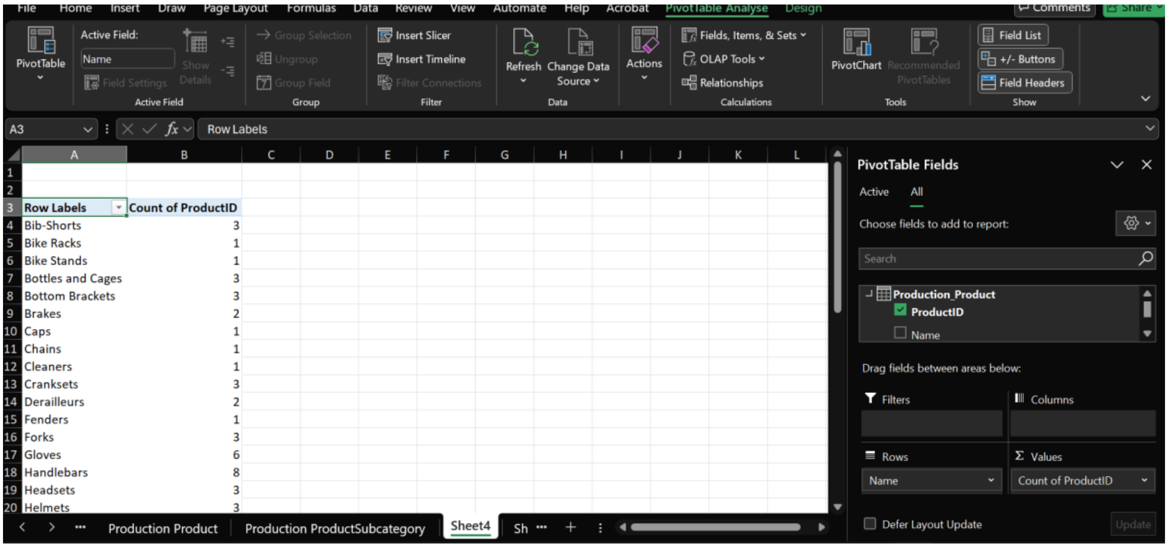
PivotTable Fields

Choose fields to add to report:

- Production_Product
- Production_ProductSubcategory

Drag fields between areas below:

I have dragged the name column from the production.productSubcategory to the rows and the productid from the  production.product to the value and set it to count



A3 Row Labels

| Row Labels | Count of ProductID |
|---|---|
| Bib-Shorts | 3 |
| Bike Racks | 1 |
| Bike Stands | 1 |
| Bottles and Cages | 3 |
| Bottom Brackets | 3 |
| Brakes | 2 |
| Caps | 1 |
| Chains | 1 |
| Cleaners | 1 |
| Cranksets | 3 |
| Derailleurs | 2 |
| Fenders | 1 |
| Forks | 3 |
| Gloves | 6 |
| Handlebars | 8 |
| Headsets | 3 |
| Helmets | 3 |

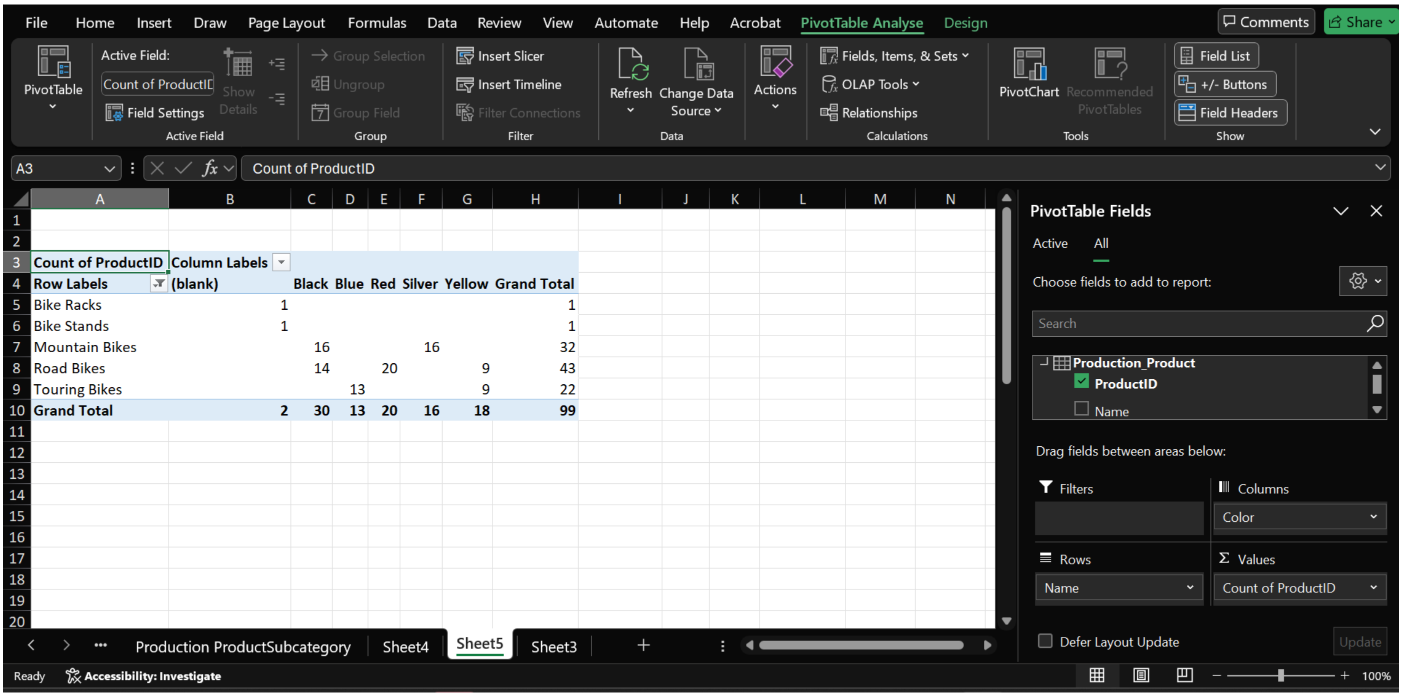
PivotTable Fields

- Production_Product
  - ☑ ProductID
  - □ Name

Rows: Name

Values: Count of ProductID

Then I have created a new pivot table in a new sheet putting the name of the subcategory in the row field and the color of the product in the columns then I have put the product ID in the Sum value and set it to count, at the end I have  used the row label filter to show only the products that have 'bike' in the name.



Active Field: Count of ProductID

A3 — Count of ProductID

| Count of ProductID | Column Labels | | | | | | |
|---|---|---|---|---|---|---|---|
| Row Labels | (blank) | Black | Blue | Red | Silver | Yellow | Grand Total |
| Bike Racks | 1 | | | | | | 1 |
| Bike Stands | 1 | | | | | | 1 |
| Mountain Bikes | | 16 | | | 16 | | 32 |
| Road Bikes | | 14 | | 20 | | 9 | 43 |
| Touring Bikes | | | 13 | | | 9 | 22 |
| Grand Total | 2 | 30 | 13 | 20 | 16 | 18 | 99 |

PivotTable Fields

Active All

Choose fields to add to report:

Search

Production_Product
☑ ProductID
□ Name

Drag fields between areas below:

Filters

Columns: Color

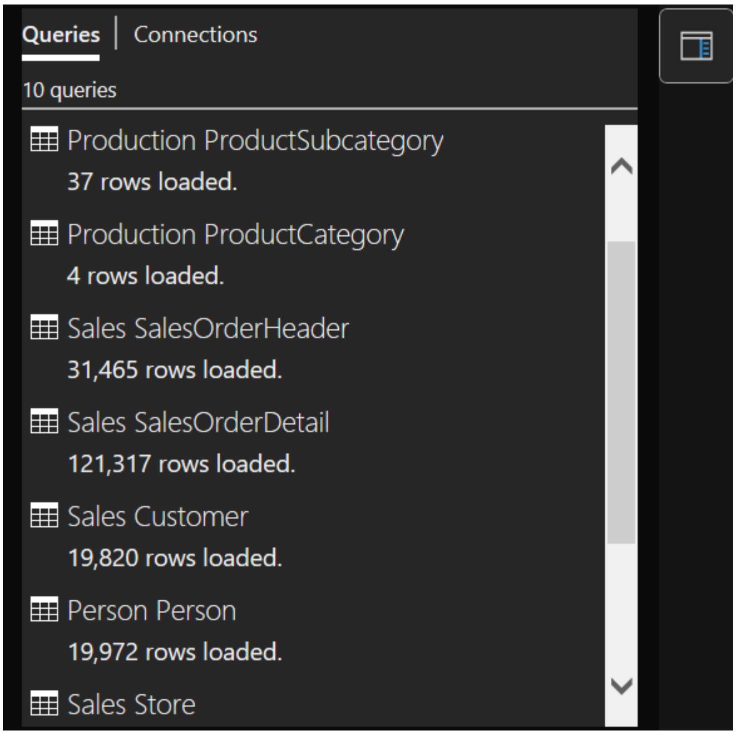
Rows: Name

Values: Count of ProductID

□ Defer Layout Update

Update

To be able to perform the other  pivot tables I have called the tables from mySQL by using get data from the ribbon



10 queries

Production ProductSubcategory
37 rows loaded.

Production ProductCategory
4 rows loaded.

Sales SalesOrderHeader
31,465 rows loaded.

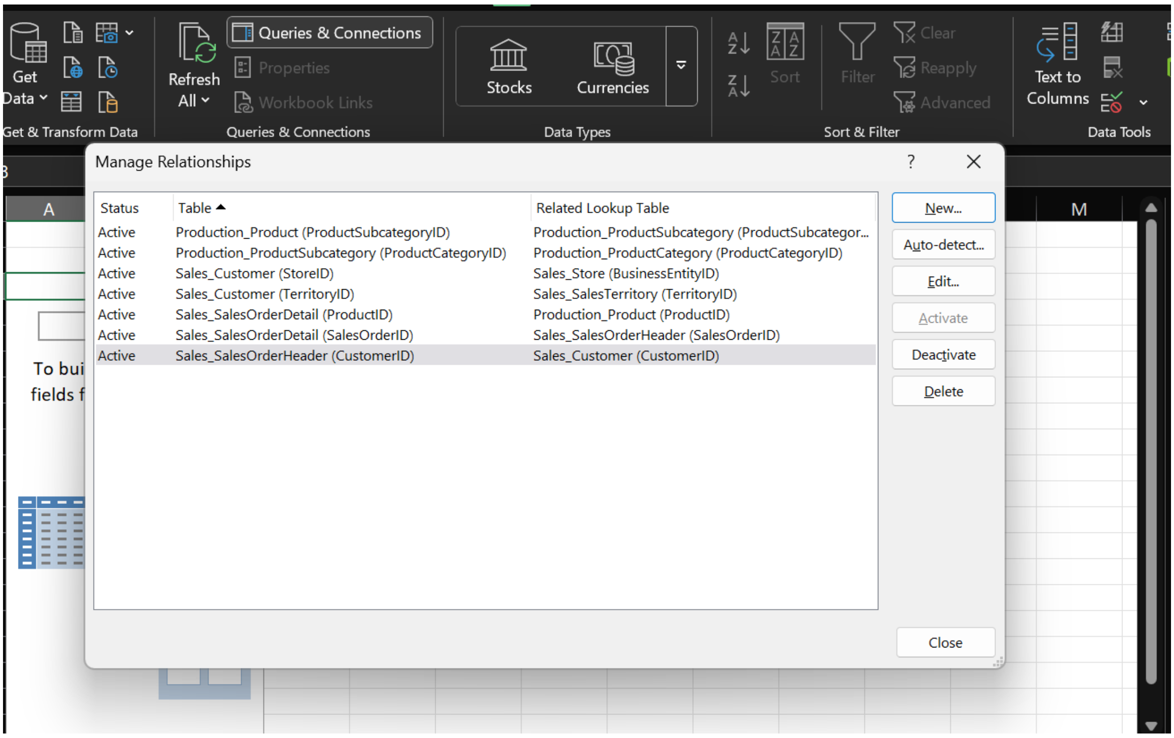
Sales SalesOrderDetail
121,317 rows loaded.

Sales Customer
19,820 rows loaded.

Person Person
19,972 rows loaded.

Sales Store

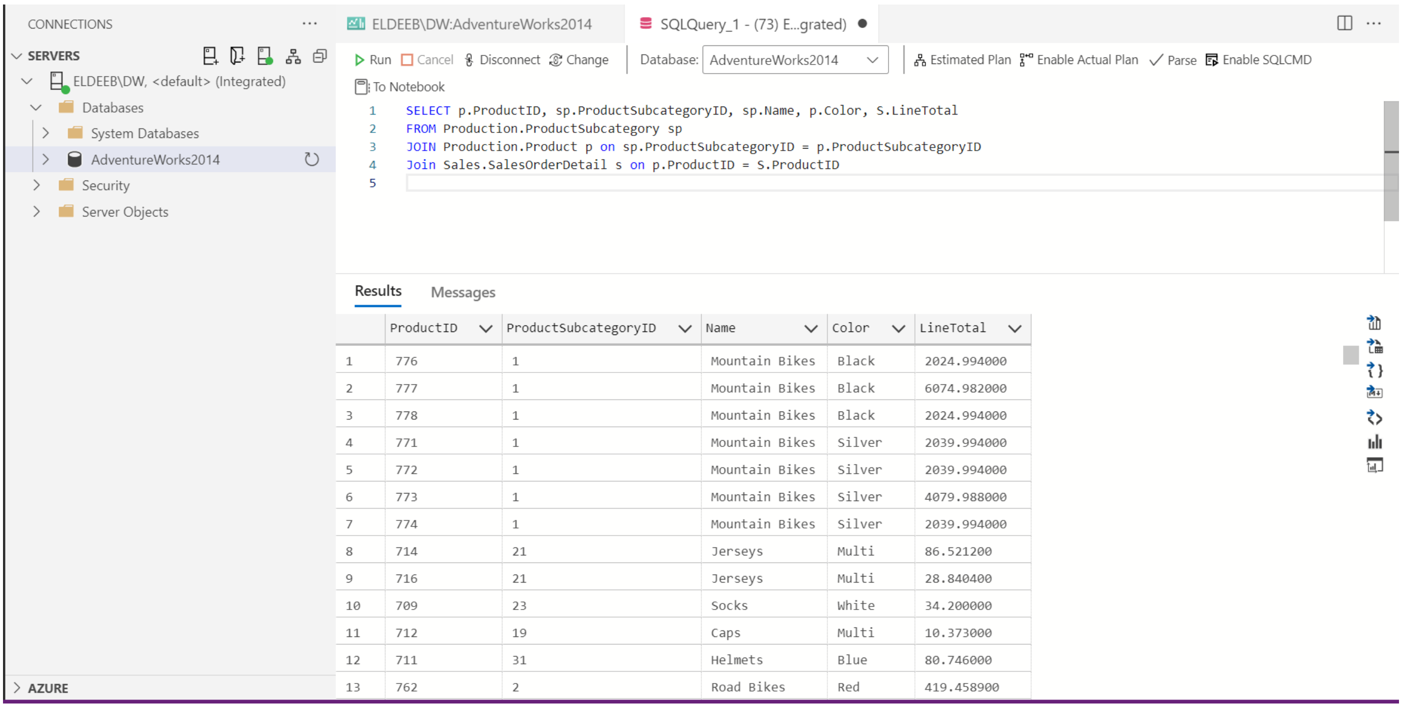
Then I defined the rest of the relationships between the tables.



Manage Relationships

| Status | Table | Related Lookup Table |
|---|---|---|
| Active | Production_Product (ProductSubcategoryID) | Production_ProductSubcategory (ProductSubcategor... |
| Active | Production_ProductSubcategory (ProductCategoryID) | Production_ProductCategory (ProductCategoryID) |
| Active | Sales_Customer (StoreID) | Sales_Store (BusinessEntityID) |
| Active | Sales_Customer (TerritoryID) | Sales_SalesTerritory (TerritoryID) |
| Active | Sales_SalesOrderDetail (ProductID) | Production_Product (ProductID) |
| Active | Sales_SalesOrderDetail (SalesOrderID) | Sales_SalesOrderHeader (SalesOrderID) |
| Active | Sales_SalesOrderHeader (CustomerID) | Sales_Customer (CustomerID) |

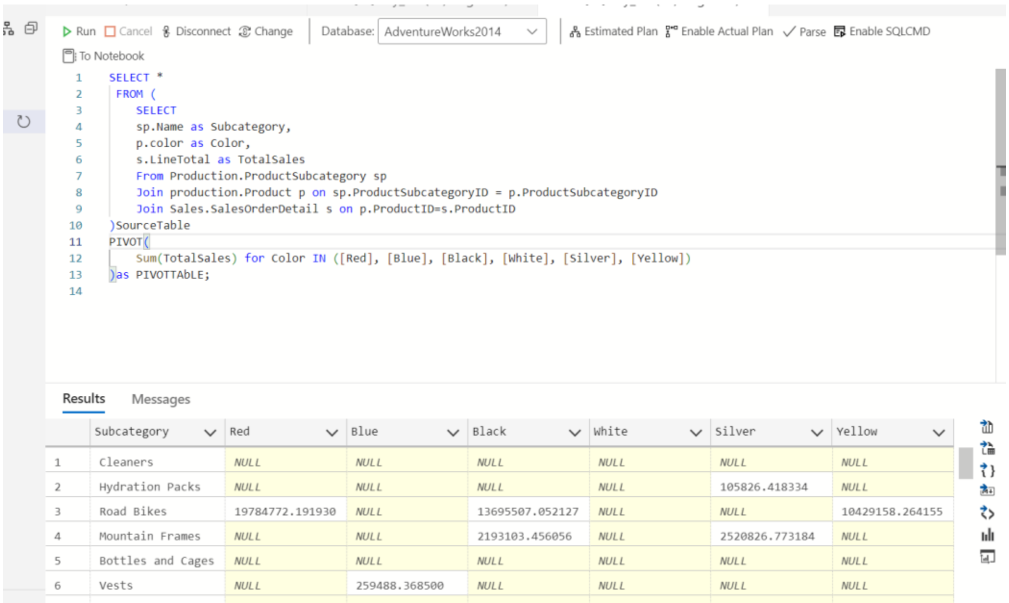
Here I opened Azure and created a base table with joined columns as it shown to retrieve the data required in the 3rd task.



```sql
SELECT p.ProductID, sp.ProductSubcategoryID, sp.Name, p.Color, S.LineTotal
FROM Production.ProductSubcategory sp
JOIN Production.Product p on sp.ProductSubcategoryID = p.ProductSubcategoryID
Join Sales.SalesOrderDetail s on p.ProductID = S.ProductID
```

| | ProductID | ProductSubcategoryID | Name | Color | LineTotal |
|---|---|---|---|---|---|
| 1 | 776 | 1 | Mountain Bikes | Black | 2024.994000 |
| 2 | 777 | 1 | Mountain Bikes | Black | 6074.982000 |
| 3 | 778 | 1 | Mountain Bikes | Black | 2024.994000 |
| 4 | 771 | 1 | Mountain Bikes | Silver | 2039.994000 |
| 5 | 772 | 1 | Mountain Bikes | Silver | 2039.994000 |
| 6 | 773 | 1 | Mountain Bikes | Silver | 4079.988000 |
| 7 | 774 | 1 | Mountain Bikes | Silver | 2039.994000 |
| 8 | 714 | 21 | Jerseys | Multi | 86.521200 |
| 9 | 716 | 21 | Jerseys | Multi | 28.840400 |
| 10 | 709 | 23 | Socks | White | 34.200000 |
| 11 | 712 | 19 | Caps | Multi | 10.373000 |
| 12 | 711 | 31 | Helmets | Blue | 80.746000 |
| 13 | 762 | 2 | Road Bikes | Red | 419.458900 |

Then I have use pivot table to calculate the total sales of the subcategory items based on the color.

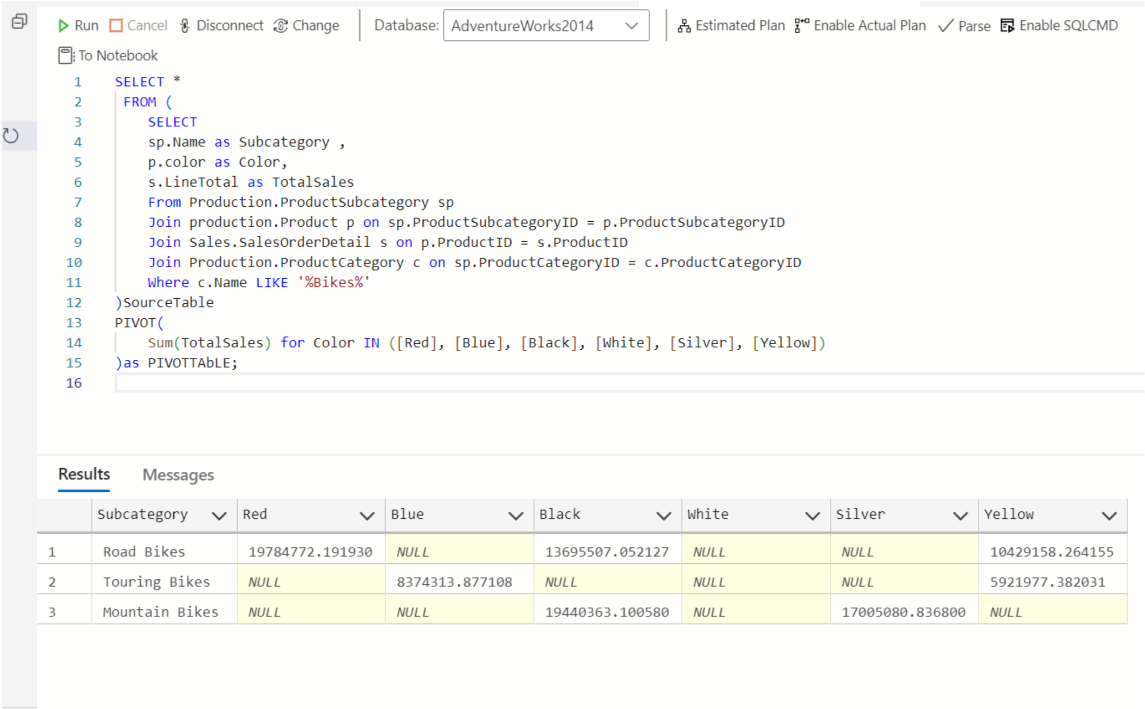


Database: AdventureWorks2014

```sql
SELECT *
FROM (
    SELECT
    sp.Name as Subcategory,
    p.color as Color,
    s.LineTotal as TotalSales
    From Production.ProductSubcategory sp
    Join production.Product p on sp.ProductSubcategoryID = p.ProductSubcategoryID
    Join Sales.SalesOrderDetail s on p.ProductID=s.ProductID
)SourceTable
PIVOT(
    Sum(TotalSales) for Color IN ([Red], [Blue], [Black], [White], [Silver], [Yellow])
)as PIVOTTAbLE;
```

| | Subcategory | Red | Blue | Black | White | Silver | Yellow |
|---|---|---|---|---|---|---|---|
| 1 | Cleaners | NULL | NULL | NULL | NULL | NULL | NULL |
| 2 | Hydration Packs | NULL | NULL | NULL | NULL | 105826.418334 | NULL |
| 3 | Road Bikes | 19784772.191930 | NULL | 13695507.052127 | NULL | NULL | 10429158.264155 |
| 4 | Mountain Frames | NULL | NULL | 2193103.456056 | NULL | 2520826.773184 | NULL |
| 5 | Bottles and Cages | NULL | NULL | NULL | NULL | NULL | NULL |
| 6 | Vests | NULL | 259488.368500 | NULL | NULL | NULL | NULL |

I have added another join statement to include the Production.ProductCategory table and be able to filter the results and get only Bikes category, and I included a WHERE.... LIKE to filter the results.

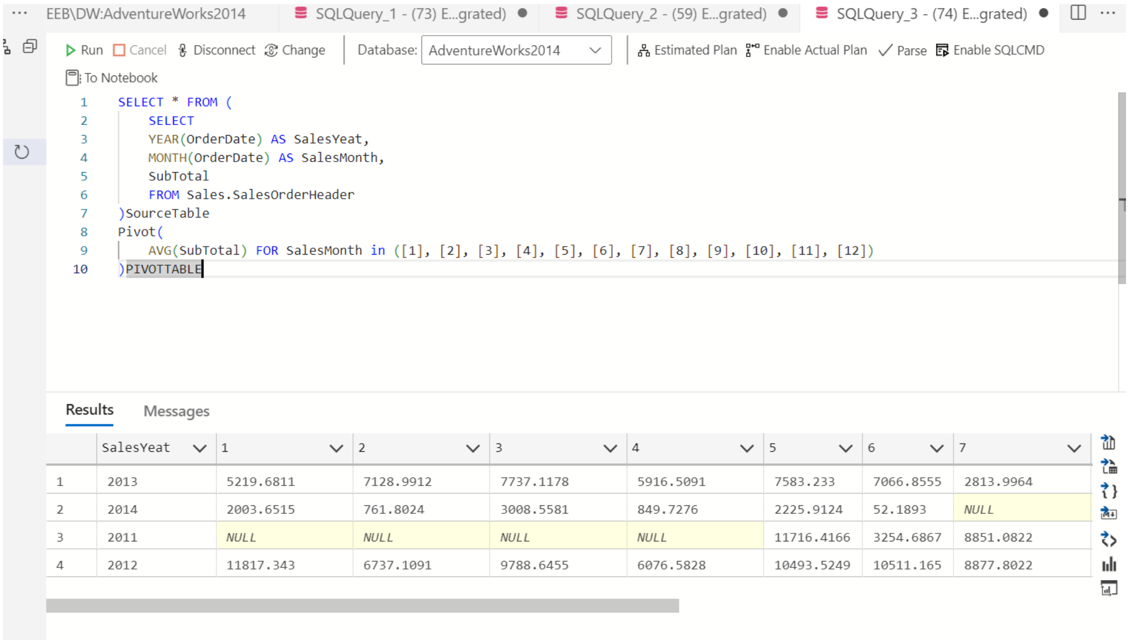


Database: AdventureWorks2014

```sql
SELECT *
 FROM (
    SELECT
    sp.Name as Subcategory ,
    p.color as Color,
    s.LineTotal as TotalSales
    From Production.ProductSubcategory sp
    Join production.Product p on sp.ProductSubcategoryID = p.ProductSubcategoryID
    Join Sales.SalesOrderDetail s on p.ProductID = s.ProductID
    Join Production.ProductCategory c on sp.ProductCategoryID = c.ProductCategoryID
    Where c.Name LIKE '%Bikes%'
)SourceTable
PIVOT(
    Sum(TotalSales) for Color IN ([Red], [Blue], [Black], [White], [Silver], [Yellow])
)as PIVOTTAbLE;
```

| | Subcategory | Red | Blue | Black | White | Silver | Yellow |
|---|---|---|---|---|---|---|---|
| 1 | Road Bikes | 19784772.191930 | NULL | 13695507.052127 | NULL | NULL | 10429158.264155 |
| 2 | Touring Bikes | NULL | 8374313.877108 | NULL | NULL | NULL | 5921977.382031 |
| 3 | Mountain Bikes | NULL | NULL | 19440363.100580 | NULL | 17005080.836800 | NULL |

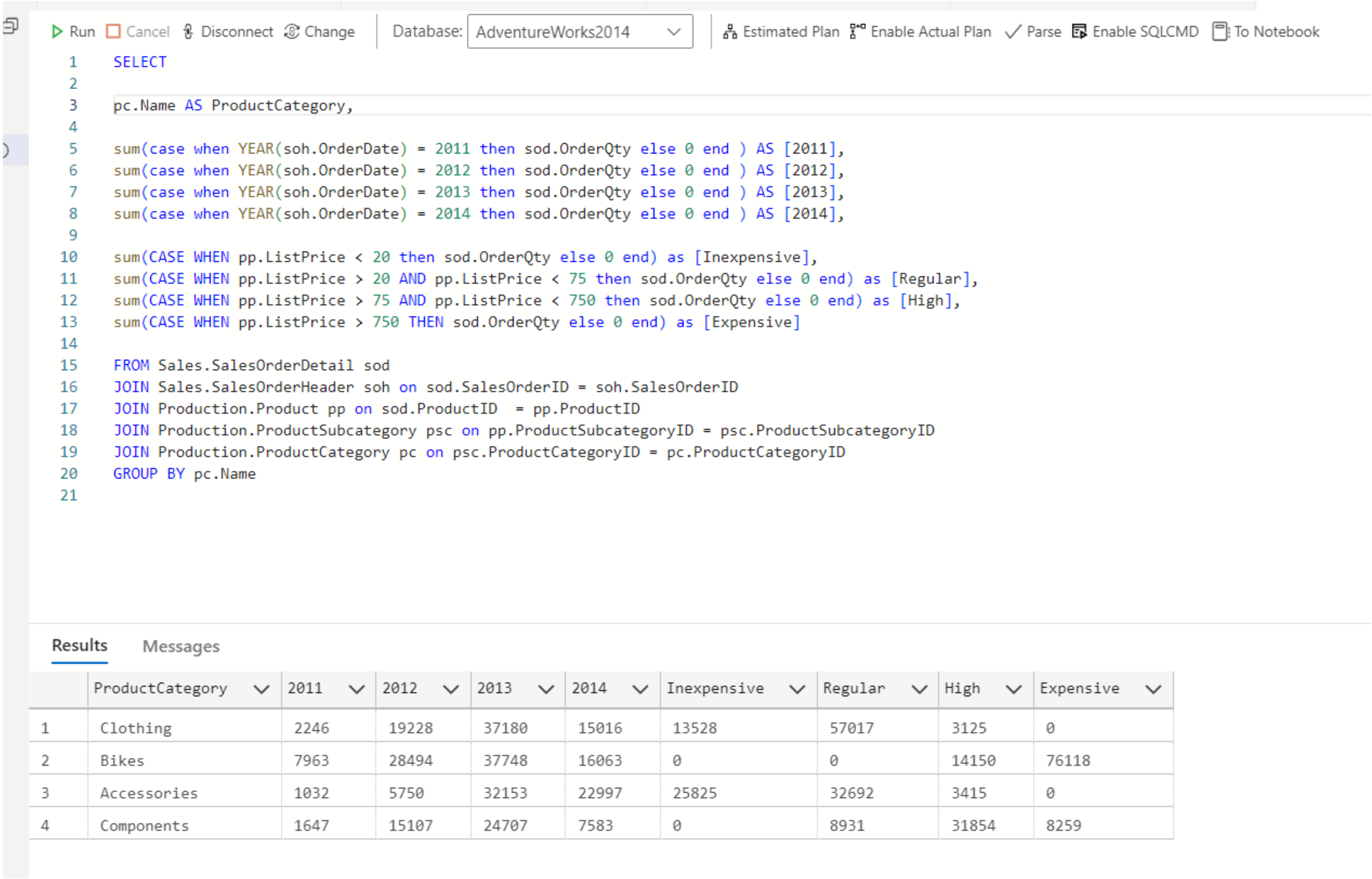
In the third subtask I have created a pivoted table from the columns in the table Sales.SalesOrderHeader to show the average of the total sales per every month among the yrs.



```sql
SELECT * FROM (
    SELECT
    YEAR(OrderDate) AS SalesYeat,
    MONTH(OrderDate) AS SalesMonth,
    SubTotal
    FROM Sales.SalesOrderHeader
)SourceTable
Pivot(
    AVG(SubTotal) FOR SalesMonth in ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12])
)PIVOTTABLE
```

| | SalesYeat | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 1 | 2013 | 5219.6811 | 7128.9912 | 7737.1178 | 5916.5091 | 7583.233 | 7066.8555 | 2813.9964 |
| 2 | 2014 | 2003.6515 | 761.8024 | 3008.5581 | 849.7276 | 2225.9124 | 52.1893 | NULL |
| 3 | 2011 | NULL | NULL | NULL | NULL | 11716.4166 | 3254.6867 | 8851.0822 |
| 4 | 2012 | 11817.343 | 6737.1091 | 9788.6455 | 6076.5828 | 10493.5249 | 10511.165 | 8877.8022 |

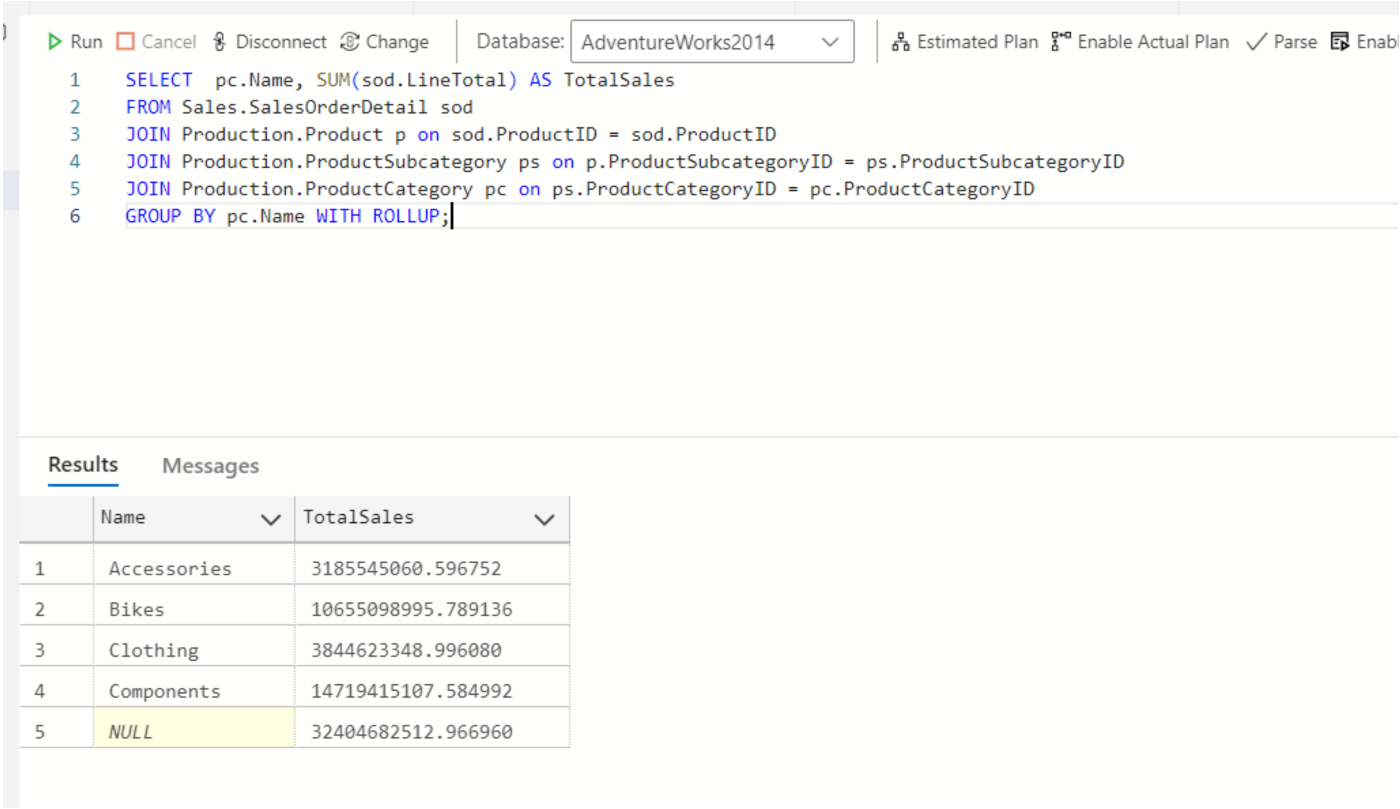
At the last task of the case statements I have used the case statements to put the years and priceCategory (its created by the case statements as well ) in columns, and put the categories in the rews.



```sql
SELECT

pc.Name AS ProductCategory,

sum(case when YEAR(soh.OrderDate) = 2011 then sod.OrderQty else 0 end ) AS [2011],
sum(case when YEAR(soh.OrderDate) = 2012 then sod.OrderQty else 0 end ) AS [2012],
sum(case when YEAR(soh.OrderDate) = 2013 then sod.OrderQty else 0 end ) AS [2013],
sum(case when YEAR(soh.OrderDate) = 2014 then sod.OrderQty else 0 end ) AS [2014],

sum(CASE WHEN pp.ListPrice < 20 then sod.OrderQty else 0 end) as [Inexpensive],
sum(CASE WHEN pp.ListPrice > 20 AND pp.ListPrice < 75 then sod.OrderQty else 0 end) as [Regular],
sum(CASE WHEN pp.ListPrice > 75 AND pp.ListPrice < 750 then sod.OrderQty else 0 end) as [High],
sum(CASE WHEN pp.ListPrice > 750 THEN sod.OrderQty else 0 end) as [Expensive]

FROM Sales.SalesOrderDetail sod
JOIN Sales.SalesOrderHeader soh on sod.SalesOrderID = soh.SalesOrderID
JOIN Production.Product pp on sod.ProductID  = pp.ProductID
JOIN Production.ProductSubcategory psc on pp.ProductSubcategoryID = psc.ProductSubcategoryID
JOIN Production.ProductCategory pc on psc.ProductCategoryID = pc.ProductCategoryID
GROUP BY pc.Name
```

| | ProductCategory | 2011 | 2012 | 2013 | 2014 | Inexpensive | Regular | High | Expensive |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Clothing | 2246 | 19228 | 37180 | 15016 | 13528 | 57017 | 3125 | 0 |
| 2 | Bikes | 7963 | 28494 | 37748 | 16063 | 0 | 0 | 14150 | 76118 |
| 3 | Accessories | 1032 | 5750 | 32153 | 22997 | 25825 | 32692 | 3415 | 0 |
| 4 | Components | 1647 | 15107 | 24707 | 7583 | 0 | 8931 | 31854 | 8259 |

To perform the first subtask in the fifth task I have created a table with the LinePrice summed and category name and did a join to put them in the same table. At the end I performed a group by with rollup to show the total of LinePrice.

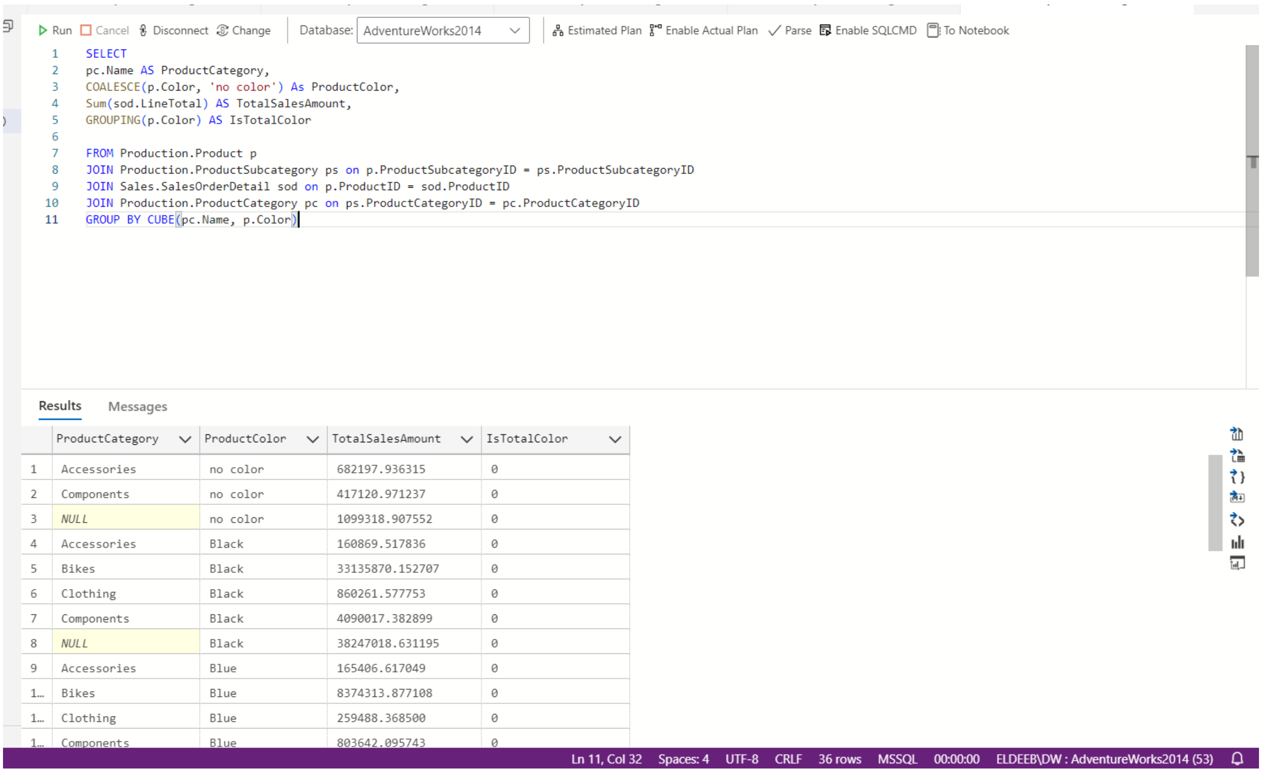


Database: AdventureWorks2014

```sql
SELECT  pc.Name, SUM(sod.LineTotal) AS TotalSales
FROM Sales.SalesOrderDetail sod
JOIN Production.Product p on sod.ProductID = sod.ProductID
JOIN Production.ProductSubcategory ps on p.ProductSubcategoryID = ps.ProductSubcategoryID
JOIN Production.ProductCategory pc on ps.ProductCategoryID = pc.ProductCategoryID
GROUP BY pc.Name WITH ROLLUP;
```

| | Name | TotalSales |
|---|---|---|
| 1 | Accessories | 3185545060.596752 |
| 2 | Bikes | 10655098995.789136 |
| 3 | Clothing | 3844623348.996080 |
| 4 | Components | 14719415107.584992 |
| 5 | *NULL* | 32404682512.966960 |

In the next task I have created a table that contains the categories with the color and the summation of the line total amount using grouping function to get the total line amount for the categories each color it have, and I used CUBE function to get all the possibilities with a summation of the all categories for the same category in the row after.



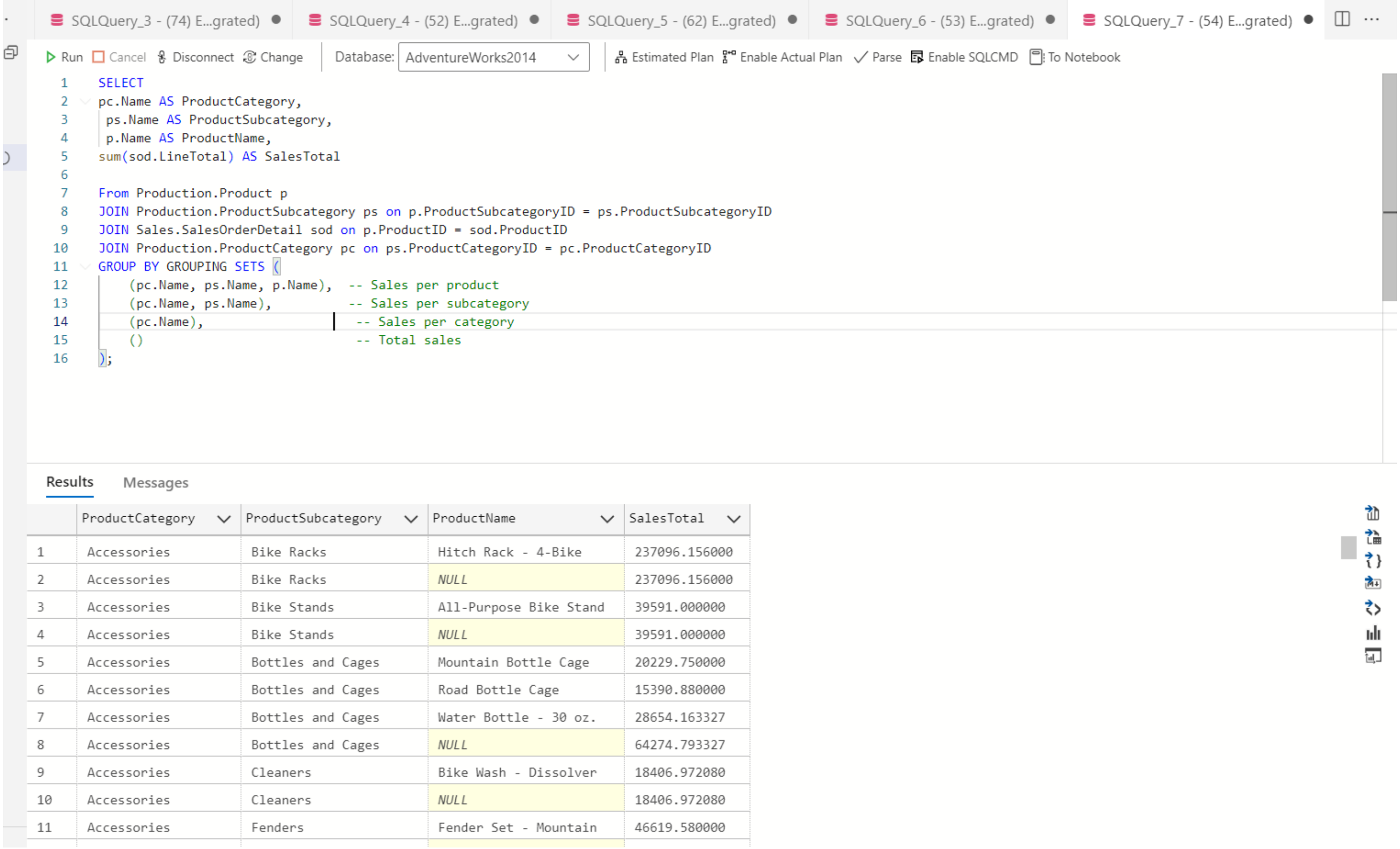
Database: AdventureWorks2014

```sql
SELECT
pc.Name AS ProductCategory,
COALESCE(p.Color, 'no color') As ProductColor,
Sum(sod.LineTotal) AS TotalSalesAmount,
GROUPING(p.Color) AS IsTotalColor

FROM Production.Product p
JOIN Production.ProductSubcategory ps on p.ProductSubcategoryID = ps.ProductSubcategoryID
JOIN Sales.SalesOrderDetail sod on p.ProductID = sod.ProductID
JOIN Production.ProductCategory pc on ps.ProductCategoryID = pc.ProductCategoryID
GROUP BY CUBE(pc.Name, p.Color)
```

| | ProductCategory | ProductColor | TotalSalesAmount | IsTotalColor |
|---|---|---|---|---|
| 1 | Accessories | no color | 682197.936315 | 0 |
| 2 | Components | no color | 417120.971237 | 0 |
| 3 | *NULL* | no color | 1099318.907552 | 0 |
| 4 | Accessories | Black | 160869.517836 | 0 |
| 5 | Bikes | Black | 33135870.152707 | 0 |
| 6 | Clothing | Black | 860261.577753 | 0 |
| 7 | Components | Black | 4090017.382899 | 0 |
| 8 | *NULL* | Black | 38247018.631195 | 0 |
| 9 | Accessories | Blue | 165406.617049 | 0 |
| 1... | Bikes | Blue | 8374313.877108 | 0 |
| 1... | Clothing | Blue | 259488.368500 | 0 |
| 1... | Components | Blue | 803642.095743 | 0 |

Ln 11, Col 32   Spaces: 4   UTF-8   CRLF   36 rows   MSSQL   00:00:00   ELDEEB\DW : AdventureWorks2014 (53)

In the last subtask I have created a table that shows the total amount of sales for each product, subcategory and category with all possibilities with showing the grand total of sales for the subcategory and category at the end, I have used Grouping Sets function to control the null values sequence.



SQLQuery_3 - (74) E...grated)
SQLQuery_4 - (52) E...grated)
SQLQuery_5 - (62) E...grated)
SQLQuery_6 - (53) E...grated)
SQLQuery_7 - (54) E...grated)
Run
Cancel
Disconnect
Change
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
Results
Messages

Database: AdventureWorks2014

```
SELECT
pc.Name AS ProductCategory,
 ps.Name AS ProductSubcategory,
 p.Name AS ProductName,
sum(sod.LineTotal) AS SalesTotal

From Production.Product p
JOIN Production.ProductSubcategory ps on p.ProductSubcategoryID = ps.ProductSubcategoryID
JOIN Sales.SalesOrderDetail sod on p.ProductID = sod.ProductID
JOIN Production.ProductCategory pc on ps.ProductCategoryID = pc.ProductCategoryID
GROUP BY GROUPING SETS (
    (pc.Name, ps.Name, p.Name),  -- Sales per product
    (pc.Name, ps.Name),          -- Sales per subcategory
    (pc.Name),                    -- Sales per category
    ()                            -- Total sales
);
```

| | ProductCategory | ProductSubcategory | ProductName | SalesTotal |
|---|---|---|---|---|
| 1 | Accessories | Bike Racks | Hitch Rack - 4-Bike | 237096.156000 |
| 2 | Accessories | Bike Racks | *NULL* | 237096.156000 |
| 3 | Accessories | Bike Stands | All-Purpose Bike Stand | 39591.000000 |
| 4 | Accessories | Bike Stands | *NULL* | 39591.000000 |
| 5 | Accessories | Bottles and Cages | Mountain Bottle Cage | 20229.750000 |
| 6 | Accessories | Bottles and Cages | Road Bottle Cage | 15390.880000 |
| 7 | Accessories | Bottles and Cages | Water Bottle - 30 oz. | 28654.163327 |
| 8 | Accessories | Bottles and Cages | *NULL* | 64274.793327 |
| 9 | Accessories | Cleaners | Bike Wash - Dissolver | 18406.972080 |
| 10 | Accessories | Cleaners | *NULL* | 18406.972080 |
| 11 | Accessories | Fenders | Fender Set - Mountain | 46619.580000 |