CN-Lec09-Aggregate Functions

Aggregate Functions

🎯 Session Overview

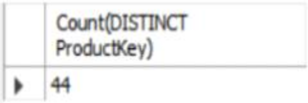
- ✅ Understand different types of aggregate functions.
- ✅ Use the GROUP BY function effectively.
- ✅ Apply the HAVING clause for advanced filtering.
- ✅ Utilise scalar functions like ROUND and ABS.

Count

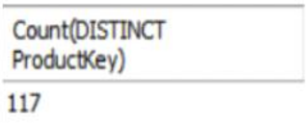
→ Count(*) -> Count all the records.
Count(Column Name) -> It is counting specific to the column
Count(DISTINCT Column Name) -> Count the distinct value in a column.

Count the number of unique products sell on 2015/16/17

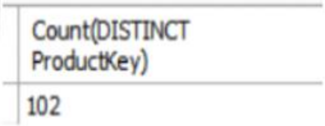
```
SELECT
        Count(DISTINCT ProductKey)
FROM `sales-2015`;

SELECT
        Count(DISTINCT ProductKey)
FROM `sales-2016`;

SELECT
        Count(DISTINCT ProductKey)
FROM `sales-2017`;
```



```
  Count(DISTINCT
  ProductKey)
▶ 44
```

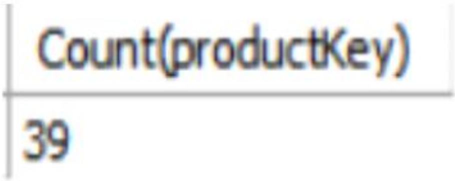


```
Count(DISTINCT
ProductKey)
117
```



```
Count(DISTINCT
ProductKey)
102
```

Count the unique products having productCost > 1000;

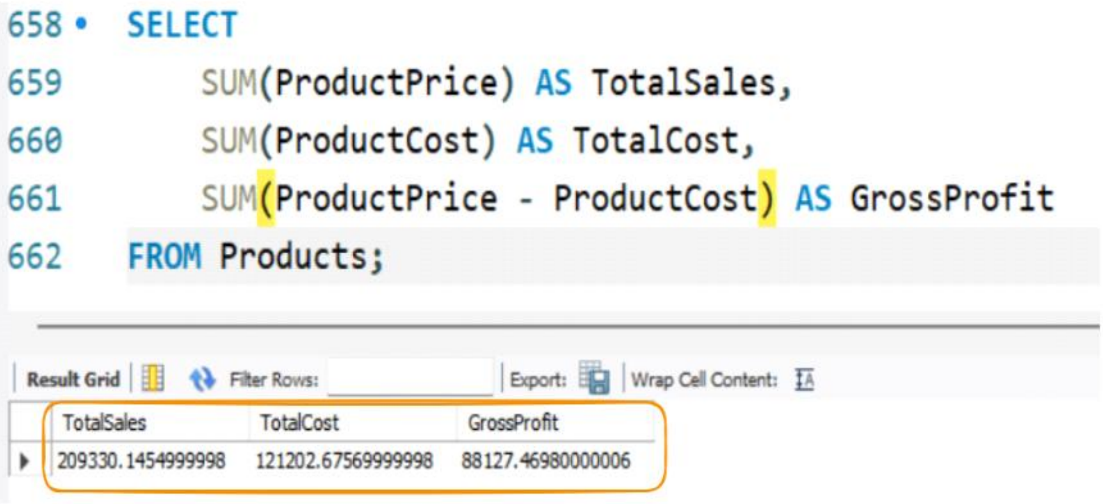


```
Count(productKey)
39
```

```
SELECT
        Count(productKey)
From Products
WHERE ProductCost > 1000;
```

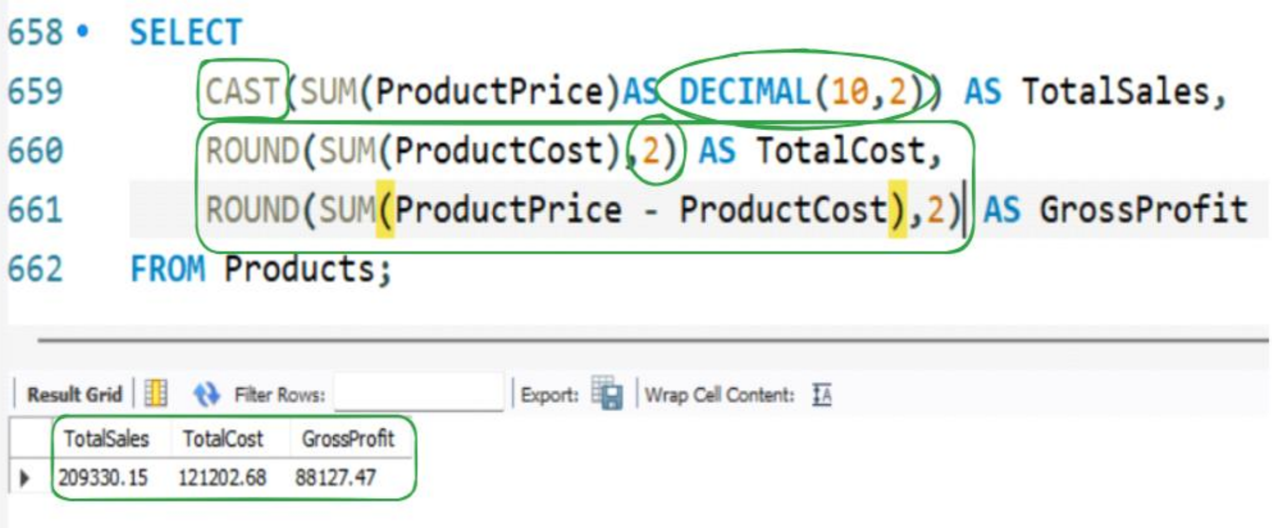
SUM/AVG() → Finding the sum/ avg of Numerical Columns.

Calculate the Sum of Product Price, Sum of Product Cost, Total Gross Profit,



```sql
SELECT
      SUM(ProductPrice) AS TotalSales,
      SUM(ProductCost) AS TotalCost,
      SUM(ProductPrice - ProductCost) AS GrossProfit
FROM Products;
```

TotalSales	TotalCost	GrossProfit
209330.1454999998	121202.67569999998	88127.46980000006



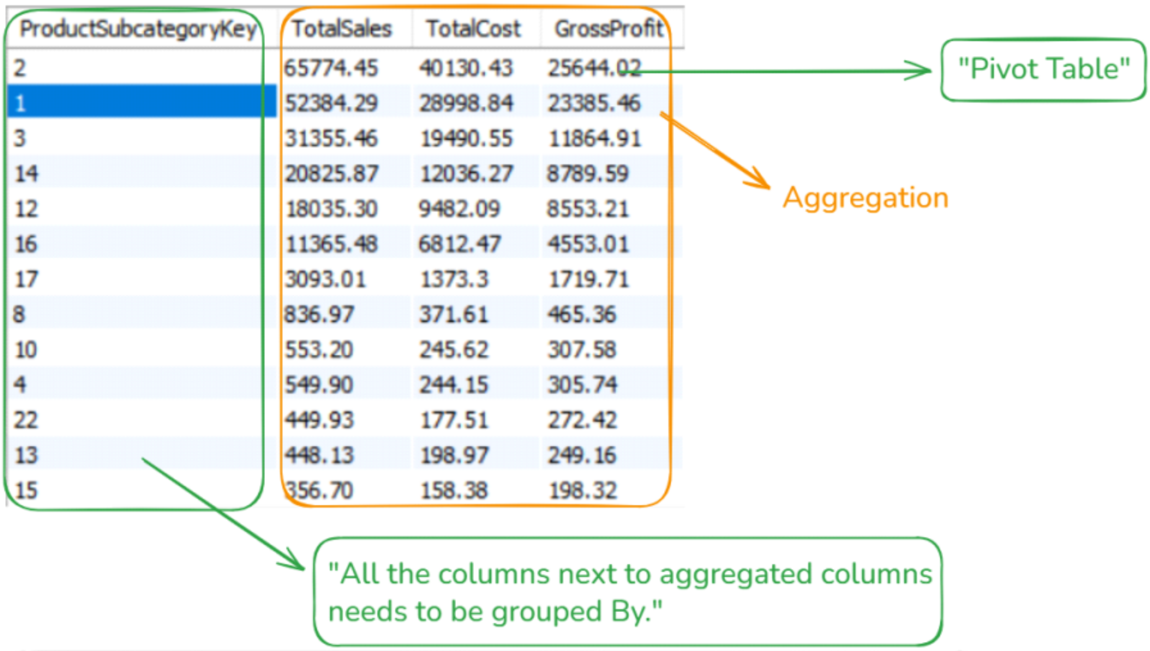
```sql
SELECT
      CAST(SUM(ProductPrice)AS DECIMAL(10,2)) AS TotalSales,
      ROUND(SUM(ProductCost),2) AS TotalCost,
      ROUND(SUM(ProductPrice - ProductCost),2) AS GrossProfit
FROM Products;
```

TotalSales	TotalCost	GrossProfit
209330.15	121202.68	88127.47

GROUP BY

"Select Group by when' you have to group categorical Value, and the corresponding column needs to be aggregated"

Find the TotalSales/Cost/Profit for Each ProductSubcategoryKey.



ProductSubcategoryKey	TotalSales	TotalCost	GrossProfit
2	65774.45	40130.43	25644.02
1	52384.29	28998.84	23385.46
3	31355.46	19490.55	11864.91
14	20825.87	12036.27	8789.59
12	18035.30	9482.09	8553.21
16	11365.48	6812.47	4553.01
17	3093.01	1373.3	1719.71
8	836.97	371.61	465.36
10	553.20	245.62	307.58
4	549.90	244.15	305.74
22	449.93	177.51	272.42
13	448.13	198.97	249.16
15	356.70	158.38	198.32

"Pivot Table"

Aggregation

"All the columns next to aggregated columns needs to be grouped By."

```
SELECT
    ProductSubcategoryKey,
    CAST(SUM(ProductPrice)AS DECIMAL(10,2)) AS TotalSales,
    ROUND(SUM(ProductCost),2) AS TotalCost,
    ROUND(SUM(ProductPrice - ProductCost),2) AS GrossProfit
FROM Products
GROUP BY ProductSubcategoryKey
ORDER BY GrossProfit DESC;
```

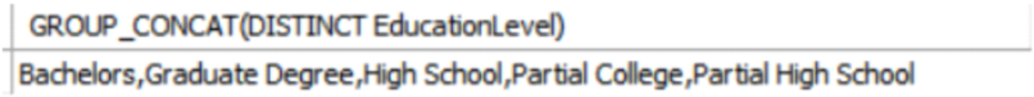
Group_CONCAT()

Syntax:

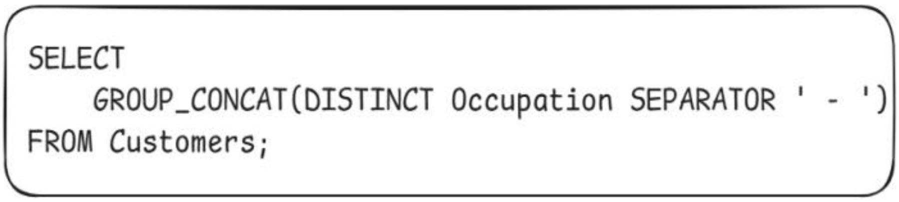
```
SELECT GROUP_CONCAT(<Column_name> SEPERATOR ',')
FROM <Table Name>
```

Group_concat - EducationLevel[,], Occupation[-], CategoryName[->], SubCategoryName[#]

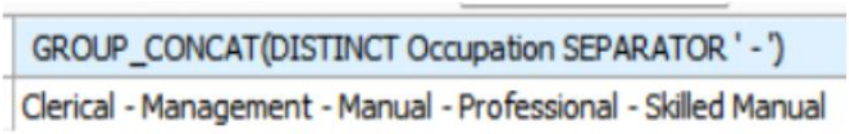
```
SELECT
    GROUP_CONCAT(DISTINCT EducationLevel)
FROM Customers;
```



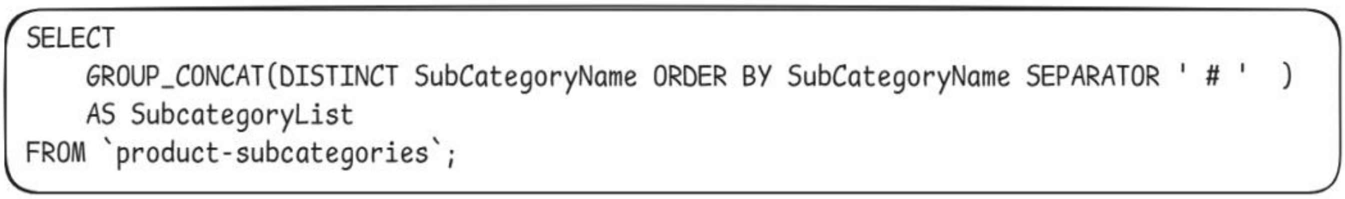
GROUP_CONCAT(DISTINCT EducationLevel)
Bachelors,Graduate Degree,High School,Partial College,Partial High School



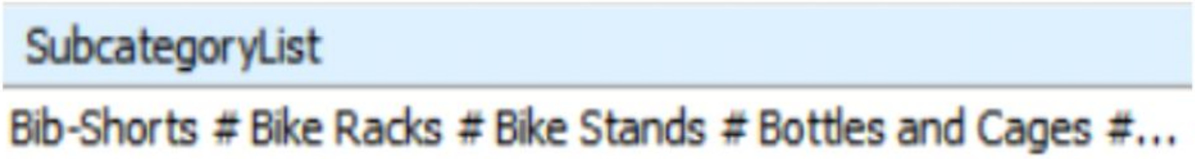
```
SELECT
    GROUP_CONCAT(DISTINCT Occupation SEPARATOR ' - ')
FROM Customers;
```



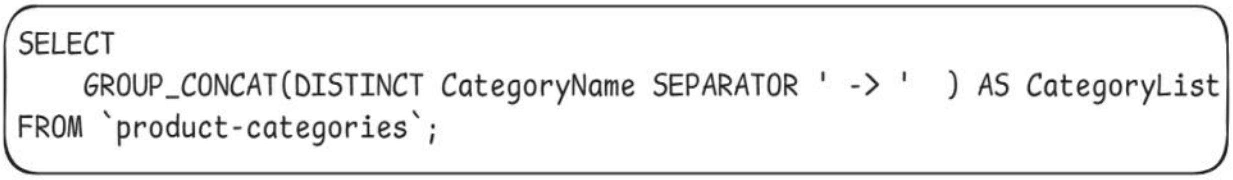
GROUP_CONCAT(DISTINCT Occupation SEPARATOR ' - ')
Clerical - Management - Manual - Professional - Skilled Manual



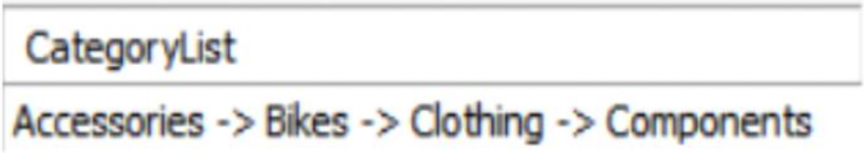
```
SELECT
    GROUP_CONCAT(DISTINCT SubCategoryName ORDER BY SubCategoryName SEPARATOR ' # '   )
    AS SubcategoryList
FROM `product-subcategories`;
```



SubcategoryList
Bib-Shorts # Bike Racks # Bike Stands # Bottles and Cages #...



```
SELECT
    GROUP_CONCAT(DISTINCT CategoryName SEPARATOR ' -> '   ) AS CategoryList
FROM `product-categories`;
```



CategoryList
Accessories -> Bikes -> Clothing -> Components

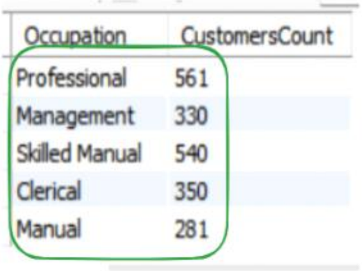
GROUP BY Challenge

Based on Each Education Level Find the total Customer.

Based on Each Occupation Find the total Customer.

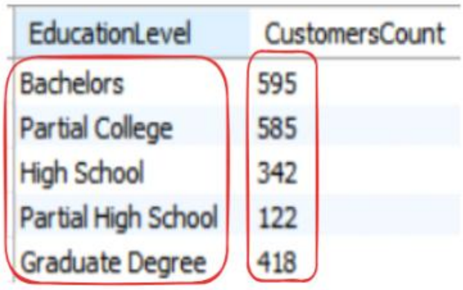
Based on Each Occupation, Education Level Find the total Customer.

```
SELECT
    Occupation,
    COUNT(CustomerKey) AS CustomersCount
FROM Customers
GROUP BY Occupation;
```

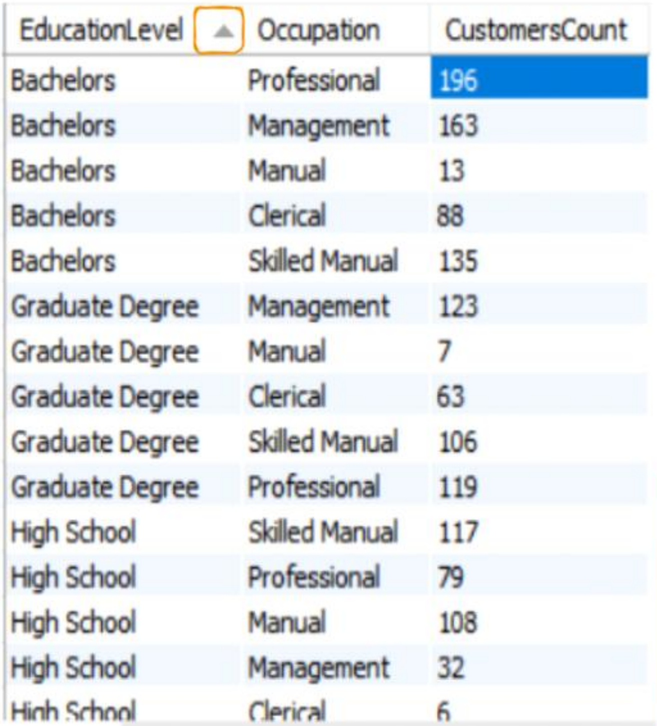


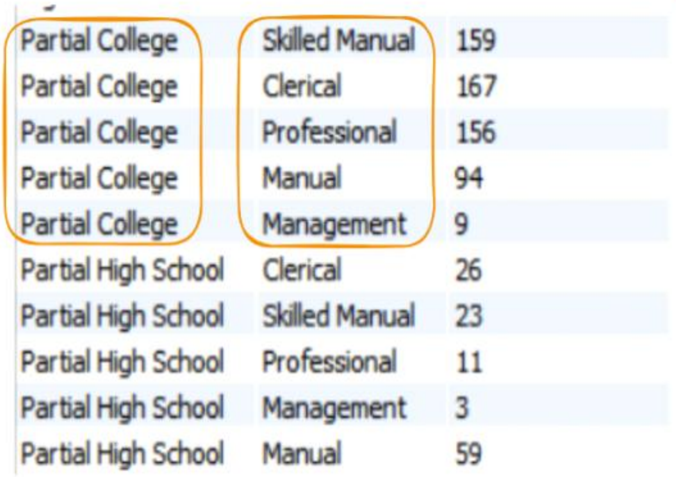
Occupation	CustomersCount
Professional	561
Management	330
Skilled Manual	540
Clerical	350
Manual	281

```
SELECT
    EducationLevel,
    COUNT(CustomerKey) AS CustomersCount
FROM Customers
GROUP BY EducationLevel;
```



EducationLevel	CustomersCount
Bachelors	595
Partial College	585
High School	342
Partial High School	122
Graduate Degree	418





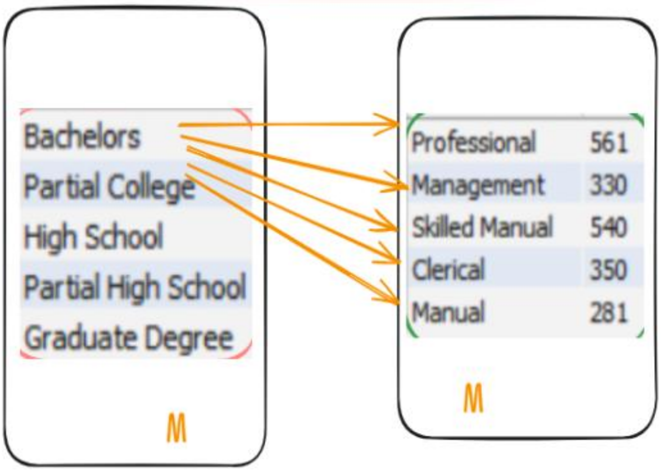
EducationLevel	Occupation	CustomersCount
Bachelors	Professional	196
Bachelors	Management	163
Bachelors	Manual	13
Bachelors	Clerical	88
Bachelors	Skilled Manual	135
Graduate Degree	Management	123
Graduate Degree	Manual	7
Graduate Degree	Clerical	63
Graduate Degree	Skilled Manual	106
Graduate Degree	Professional	119
High School	Skilled Manual	117
High School	Professional	79
High School	Manual	108
High School	Management	32
High School	Clerical	6
Partial College	Skilled Manual	159
Partial College	Clerical	167
Partial College	Professional	156
Partial College	Manual	94
Partial College	Management	9
Partial High School	Clerical	26
Partial High School	Skilled Manual	23
Partial High School	Professional	11
Partial High School	Management	3
Partial High School	Manual	59

5 education

5 Occupation

25

```
SELECT
    EducationLevel,
    Occupation,
    COUNT(CustomerKey) AS CustomersCount
FROM Customers
GROUP BY 1,2;
```

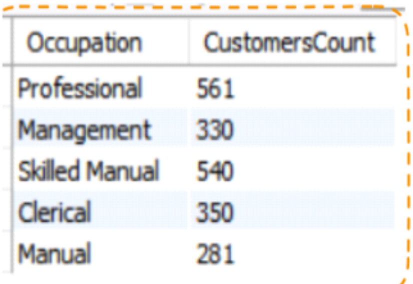


WHERE CLAUSE VS HAVING CLAUSE

Applying a filter on
Original Table

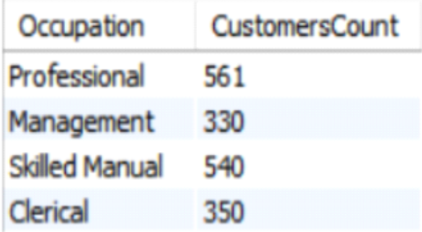
Applying a Filter on
Grouped Table/ Summarize Table.

Grouped
Table



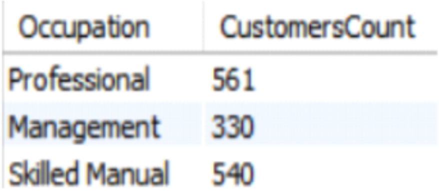
Occupation	CustomersCount
Professional	561
Management	330
Skilled Manual	540
Clerical	350
Manual	281

```
SELECT
    Occupation,
    COUNT(CustomerKey) AS CustomersCount
FROM Customers
GROUP BY Occupation
Having CustomersCount > 300;
```



Occupation	CustomersCount
Professional	561
Management	330
Skilled Manual	540
Clerical	350

```
SELECT
    Occupation,
    COUNT(CustomerKey) AS CustomersCount
FROM Customers
WHERE Occupation NOT LIKE "Clerical"
GROUP BY Occupation
Having CustomersCount > 300;
```



Occupation	CustomersCount
Professional	561
Management	330
Skilled Manual	540

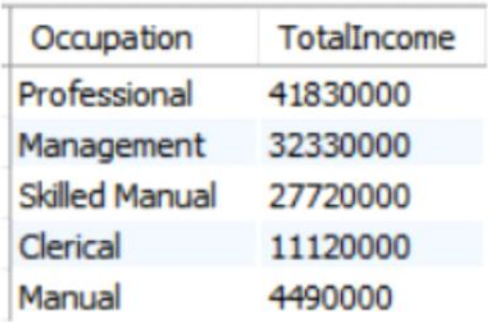
#Challenges

|-99| = 99 [ABS]

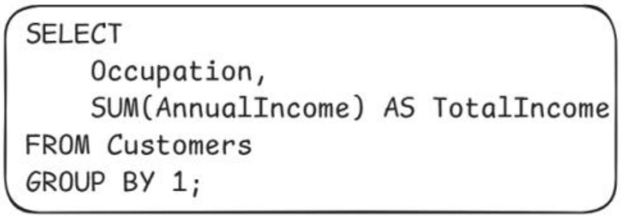
Find the Total AnnualIncome based on Occupation / Education / Gender.

Find the Total AnnualIncome based on Occupation - Education where Gender = "Male".

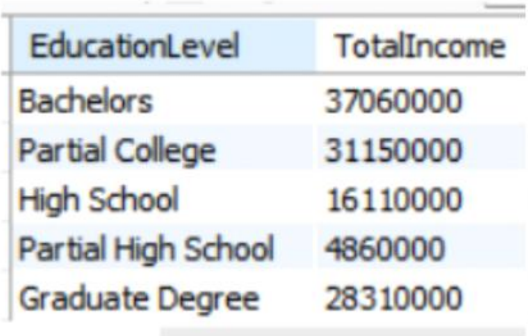
Find the AvgCost ,AvgPrice, ABS[(AvgCost-AvgPrice)] AS AvgProfit having AvgProfit > 100 based on each instance of ProductSubcategoryKey.



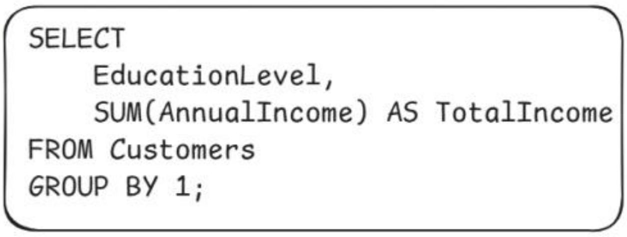
Occupation	TotalIncome
Professional	41830000
Management	32330000
Skilled Manual	27720000
Clerical	11120000
Manual	4490000



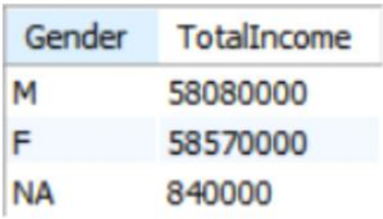
```
SELECT
    Occupation,
    SUM(AnnualIncome) AS TotalIncome
FROM Customers
GROUP BY 1;
```



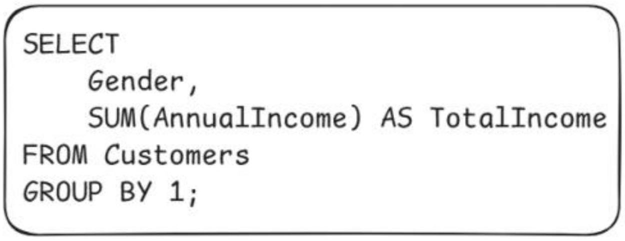
EducationLevel	TotalIncome
Bachelors	37060000
Partial College	31150000
High School	16110000
Partial High School	4860000
Graduate Degree	28310000



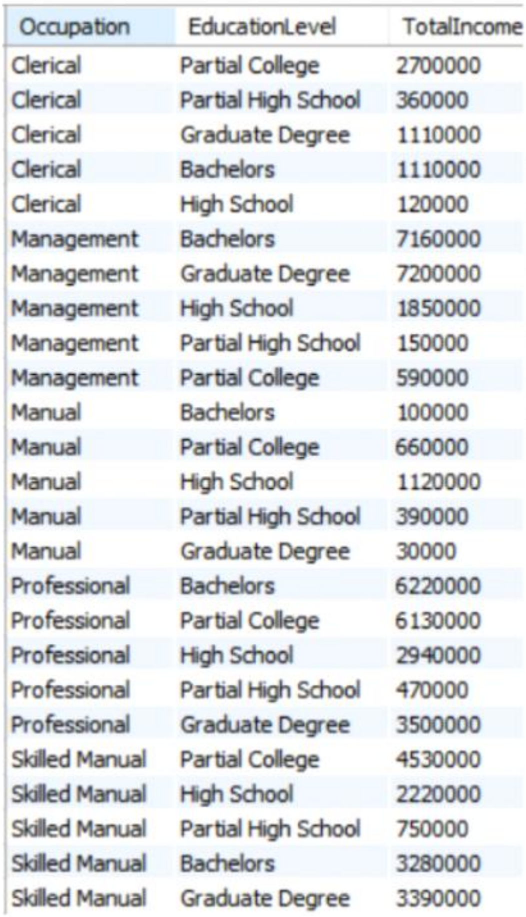
```
SELECT
    EducationLevel,
    SUM(AnnualIncome) AS TotalIncome
FROM Customers
GROUP BY 1;
```



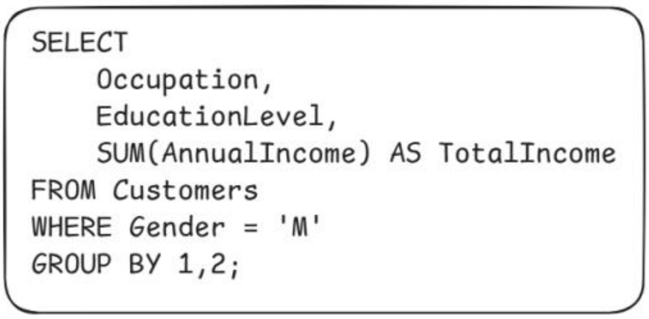
Gender	TotalIncome
M	58080000
F	58570000
NA	840000



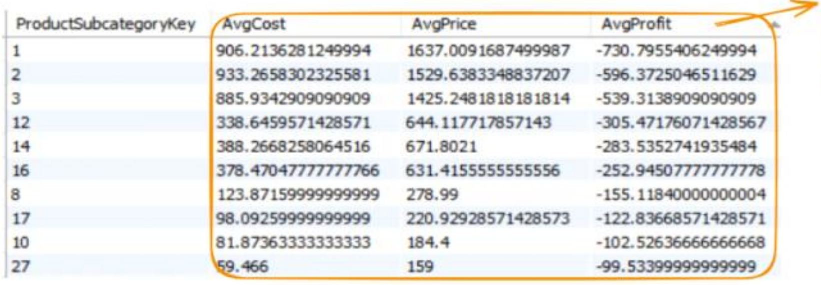
```
SELECT
    Gender,
    SUM(AnnualIncome) AS TotalIncome
FROM Customers
GROUP BY 1;
```



Occupation	EducationLevel	TotalIncome
Clerical	Partial College	2700000
Clerical	Partial High School	360000
Clerical	Graduate Degree	1110000
Clerical	Bachelors	1110000
Clerical	High School	120000
Management	Bachelors	7160000
Management	Graduate Degree	7200000
Management	High School	1850000
Management	Partial High School	150000
Management	Partial College	590000
Manual	Bachelors	100000
Manual	Partial College	660000
Manual	High School	1120000
Manual	Partial High School	390000
Manual	Graduate Degree	30000
Professional	Bachelors	6220000
Professional	Partial College	6130000
Professional	High School	2940000
Professional	Partial High School	470000
Professional	Graduate Degree	3500000
Skilled Manual	Partial College	4530000
Skilled Manual	High School	2220000
Skilled Manual	Partial High School	750000
Skilled Manual	Bachelors	3280000
Skilled Manual	Graduate Degree	3390000



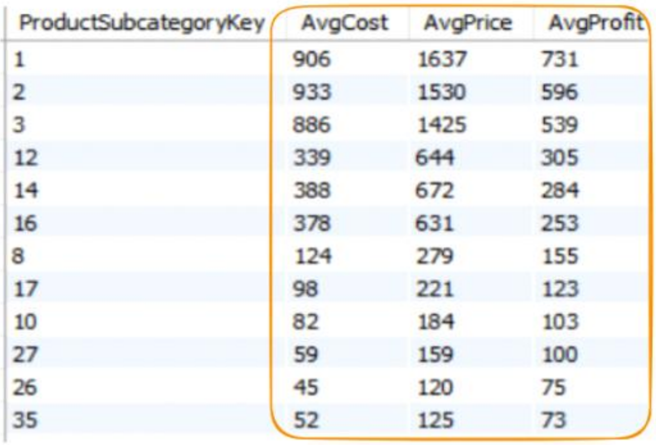
```
SELECT
    Occupation,
    EducationLevel,
    SUM(AnnualIncome) AS TotalIncome
FROM Customers
WHERE Gender = 'M'
GROUP BY 1,2;
```



ProductSubcategoryKey	AvgCost	AvgPrice	AvgProfit
1	906.2136281249994	1637.0091687499987	-730.7955406249994
2	933.2658302325581	1529.6383348837207	-596.3725046511629
3	885.9342909090909	1425.2481818181814	-539.3138909090909
12	338.6459571428571	644.117717857143	-305.47176071428567
14	388.2668258064516	671.8021	-283.5352741935484
16	378.47047777777766	631.4155555555556	-252.94507777777778
8	123.87159999999999	278.99	-155.11840000000004
17	98.09259999999999	220.92928571428573	-122.83668571428571
10	81.87363333333333	184.4	-102.52636666666668
27	59.466	159	-99.53399999999999

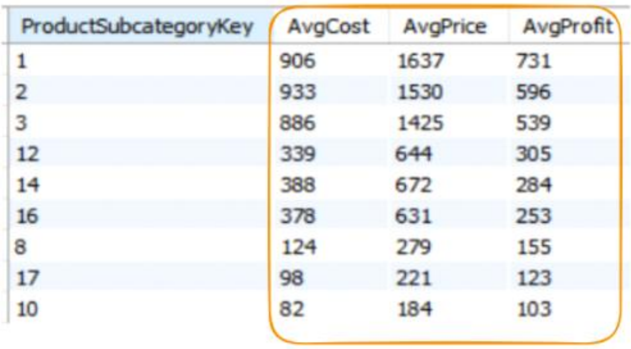
-ve profit - fix by ABS
too much value in decimal - Round

```
SELECT
        ProductSubcategoryKey,
    AVG(ProductCost) AS AvgCost,
    AVG(ProductPrice) AS AvgPrice,
    AVG(ProductCost - ProductPrice) AS AvgProfit
FROM Products
GROUP BY 1;
```



ProductSubcategoryKey	AvgCost	AvgPrice	AvgProfit
1	906	1637	731
2	933	1530	596
3	886	1425	539
12	339	644	305
14	388	672	284
16	378	631	253
8	124	279	155
17	98	221	123
10	82	184	103
27	59	159	100
26	45	120	75
35	52	125	73

```
SELECT
    ProductSubcategoryKey,
    ROUND(AVG(ProductCost),0) AS AvgCost,
    ROUND(AVG(ProductPrice),0) AS AvgPrice,
    ROUND(ABS(AVG(ProductCost - ProductPrice)),0) AS AvgProfit
FROM Products
GROUP BY 1;
```



ProductSubcategoryKey	AvgCost	AvgPrice	AvgProfit
1	906	1637	731
2	933	1530	596
3	886	1425	539
12	339	644	305
14	388	672	284
16	378	631	253
8	124	279	155
17	98	221	123
10	82	184	103

```
SELECT
    ProductSubcategoryKey,
    ROUND(AVG(ProductCost),0) AS AvgCost,
    ROUND(AVG(ProductPrice),0) AS AvgPrice,
    ROUND(ABS(AVG(ProductCost - ProductPrice)),0) AS AvgProfit
FROM Products
GROUP BY 1
HAVING AvgProfit > 100;
```