Aggregate Function - II

MAX / MIN

```
mysql> SELECT
    ->  OrderNumber,
    ->     MAX(OrderLineItem) as MaxItem_In_Invoice
    -> FROM `sales-2017`
    -> GROUP BY 1
    -> ORDER BY 2 DESC
    -> LIMIT 5;
+-------------+--------------------+
| OrderNumber | MaxItem_In_Invoice |
+-------------+--------------------+
| SO72656     |                  8 |
| SO70714     |                  8 |
| SO72927     |                  7 |
| SO71961     |                  7 |
| SO61412     |                  7 |
+-------------+--------------------+
5 rows in set (0.05 sec)
```

```
mysql> SELECT
    ->  OrderNumber,
    ->     COUNT(OrderLineItem) as Total_Order_In_Invoice
    -> FROM `sales-2017`
    -> GROUP BY 1
    -> ORDER BY 2 DESC
    -> LIMIT 5;
+-------------+-----------------------+
| OrderNumber | Total_Order_In_Invoice |
+-------------+-----------------------+
| SO72656     |                     8 |
| SO70714     |                     8 |
| SO72927     |                     7 |
| SO71961     |                     7 |
| SO64042     |                     7 |
+-------------+-----------------------+
5 rows in set (0.06 sec)
```

```
mysql> SELECT
    ->  OrderNumber,
    ->     COUNT(OrderLineItem) as Total_Order_In_Invoice
    -> FROM `sales-2017`
    -> GROUP BY 1
    -> ORDER BY 2
    -> LIMIT 5;
+-------------+-----------------------+
| OrderNumber | Total_Order_In_Invoice |
+-------------+-----------------------+
| SO73983     |                     1 |
| SO73800     |                     1 |
| SO73834     |                     1 |
| SO74031     |                     1 |
| SO73955     |                     1 |
+-------------+-----------------------+
5 rows in set (0.06 sec)
```

Order BY can replace most of the call of Min / Max Function.

```
mysql> SELECT
    ->  OrderNumber,
    ->     MIN(OrderQuantity) as MinOrderQty
    -> FROM `sales-2017`
    -> GROUP BY 1
    -> ORDER BY 2 DESC
    -> LIMIT 10;
+-------------+-------------+
| OrderNumber | MinOrderQty |
+-------------+-------------+
| SO69113     |           3 |
| SO68777     |           3 |
| SO71107     |           3 |
| SO68952     |           3 |
| SO69670     |           3 |
| SO68220     |           3 |
| SO68991     |           3 |
| SO68953     |           3 |
| SO69122     |           3 |
| SO70021     |           3 |
+-------------+-------------+
10 rows in set (0.05 sec)
```

```
mysql> SELECT MIN(AnnualIncome) FROM Customers;
+-------------------+
| MIN(AnnualIncome) |
+-------------------+
|             10000 |
+-------------------+
1 row in set (0.05 sec)

mysql> SELECT MAX(AnnualIncome) FROM Customers;
+-------------------+
| MAX(AnnualIncome) |
+-------------------+
|            170000 |
+-------------------+
1 row in set (0.00 sec)
```

```
-- MIN / MAX
-- [5,10,15,20,25,30] -> MIN : 5 , MAX:30 , COUNT : 6
-- [1,2,3,4,5,6,7,8,9,10] -> MIN:1 , MAX : 10 , COUNT 10
```

ROUND() / ABS() / CAST()

```
mysql> SELECT
    ->  Round(ProductPrice,0) AS RoundedPrice,
    ->     ROUND(ProductCost,2) AS RoundedCost
    -> FROM Products
    -> LIMIT 10;
+--------------+-------------+
| RoundedPrice | RoundedCost |
+--------------+-------------+
|           35 |       13.09 |
|           34 |       12.03 |
|           10 |         3.4 |
|           10 |         3.4 |
|           34 |       12.03 |
|            9 |        5.71 |
|           48 |       31.72 |
|           48 |       31.72 |
|           48 |       31.72 |
|           48 |       31.72 |
+--------------+-------------+
10 rows in set (0.01 sec)
```

```
mysql> SELECT
    ->  CAST(ProductPrice AS DECIMAL(10,2)) AS RoundedPrice,
    ->     CAST(ProductCost AS DECIMAL(10,2)) AS RoundedCost
    -> FROM Products
    -> LIMIT 10;
+-------------+-------------+
| RoundedPrice | RoundedCost |
+-------------+-------------+
|        34.99 |       13.09 |
|        33.64 |       12.03 |
|         9.50 |        3.40 |
|         9.50 |        3.40 |
|        33.64 |       12.03 |
|         8.64 |        5.71 |
|        48.07 |       31.72 |
|        48.07 |       31.72 |
|        48.07 |       31.72 |
|        48.07 |       31.72 |
+-------------+-------------+
10 rows in set (0.01 sec)
```

```
mysql> SELECT ABS(-10.99);
+-------------+
| ABS(-10.99) |
+-------------+
|       10.99 |
+-------------+
1 row in set (0.01 sec)

mysql> SELECT ABS(10.99);
+------------+
| ABS(10.99) |
+------------+
|      10.99 |
+------------+
1 row in set (0.00 sec)
```

|-11| = 11

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ROUND(SUM(ProductCost - ProductPrice),2) AS GrossProfit
    -> FROM Products
    -> GROUP BY ProductSubcategoryKey
    -> ORDER BY GrossProfit
    -> LIMIT 5;
```

ASC

ProductSubcategoryKey	TotalProductCost	TotalProductPrice	GrossProfit
2	40130.43	65774.45	-25644.02
1	28998.84	52384.29	-23385.46
3	19490.55	31355.46	-11864.91
14	12036.27	20825.87	-8789.59
12	9482.09	18035.3	-8553.21

5 rows in set (0.01 sec)

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> GROUP BY ProductSubcategoryKey
    -> ORDER BY GrossProfit
    -> LIMIT 5;
```

ProductSubcategoryKey	TotalProductCost	TotalProductPrice	GrossProfit
19	5.71	8.64	2.94
29	2.97	7.95	4.98
7	8.99	20.24	11.25
30	8.22	21.98	13.76
34	10.31	25	14.69

5 rows in set (0.00 sec)

ABS|-101.99| = 101.99

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> GROUP BY ProductSubcategoryKey
    -> ORDER BY GrossProfit DESC
    -> LIMIT 5;
```

ProductSubcategoryKey	TotalProductCost	TotalProductPrice	GrossProfit
2	40130.43	65774.45	25644.02
1	28998.84	52384.29	23385.46
3	19490.55	31355.46	11864.91
14	12036.27	20825.87	8789.59
12	9482.09	18035.3	8553.21

5 rows in set (0.01 sec)

GROUP_CONCAT

```
mysql> SELECT DISTINCT CategoryName FROM `product-categories`;
+--------------+
| CategoryName |
+--------------+
| Bikes        |
| Components   |
| Clothing     |
| Accessories  |
+--------------+
```

```
SELECT GROUP_CONCAT(Col_name SEPARATOR , ", ")
FROM table_name;
```

```
mysql> SELECT
    ->  GROUP_CONCAT(CategoryName) AS AllCategories
    -> FROM `product-categories`;
+-------------------------------------+
| AllCategories                       |
+-------------------------------------+
| Bikes,Components,Clothing,Accessories |
+-------------------------------------+
1 row in set (0.01 sec)
```

```
-- GROUP_CONCAT
SELECT
    GROUP_CONCAT(CategoryName) AS AllCategories
FROM `product-categories`;
```

```
mysql> SELECT
    ->  GROUP_CONCAT(CategoryName SEPARATOR " - ") AS AllCategories
    -> FROM `product-categories`;
+-------------------------------------------+
| AllCategories                             |
+-------------------------------------------+
| Bikes - Components - Clothing - Accessories |
+-------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT
    -> GROUP_CONCAT(DISTINCT EducationLevel SEPARATOR " - ") AS UniqueEducationLevel
    -> FROM Customers;
+-----------------------------------------------------------------------------------+
| UniqueEducationLevel                                                              |
+-----------------------------------------------------------------------------------+
| Bachelors - Graduate Degree - High School - Partial College - Partial High School |
+-----------------------------------------------------------------------------------+
1 row in set (0.01 sec)
```

```
mysql> SELECT
    -> GROUP_CONCAT(DISTINCT Occupation SEPARATOR " - ") AS UniqueOccupation
    -> FROM Customers;
+-----------------------------------------------------------------+
| UniqueOccupation                                                |
+-----------------------------------------------------------------+
| Clerical - Management - Manual - Professional - Skilled Manual |
+-----------------------------------------------------------------+
1 row in set (0.00 sec)
```

HAVING() - It's used to apply filter after GROUP BY.

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> GROUP BY ProductSubcategoryKey
    -> ORDER BY GrossProfit DESC
    -> LIMIT 5;
+----------------------+-----------------+-------------------+-------------+
| ProductSubcategoryKey | TotalProductCost | TotalProductPrice | GrossProfit |
+----------------------+-----------------+-------------------+-------------+
|                    2 |        40130.43 |          65774.45 |    25644.02 |
|                    1 |        28998.84 |          52384.29 |    23385.46 |
|                    3 |        19490.55 |          31355.46 |    11864.91 |
|                   14 |        12036.27 |          20825.87 |     8789.59 |
|                   12 |         9482.09 |           18035.3 |     8553.21 |
+----------------------+-----------------+-------------------+-------------+
5 rows in set (0.01 sec)
```

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> GROUP BY ProductSubcategoryKey
    -> HAVING GrossProfit > 10000
    -> ORDER BY GrossProfit DESC
    -> LIMIT 5;
+----------------------+------------------+-------------------+-------------+
| ProductSubcategoryKey | TotalProductCost | TotalProductPrice | GrossProfit |
+----------------------+------------------+-------------------+-------------+
|                    2 |         40130.43 |          65774.45 |    25644.02 |
|                    1 |         28998.84 |          52384.29 |    23385.46 |
|                    3 |         19490.55 |          31355.46 |    11864.91 |
+----------------------+------------------+-------------------+-------------+
3 rows in set (0.01 sec)
```

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> GROUP BY ProductSubcategoryKey
    -> ORDER BY GrossProfit DESC
    -> LIMIT 10;
+----------------------+------------------+-------------------+-------------+
| ProductSubcategoryKey | TotalProductCost | TotalProductPrice | GrossProfit |
+----------------------+------------------+-------------------+-------------+
|                    2 |         40130.43 |          65774.45 |    25644.02 |
|                    1 |         28998.84 |          52384.29 |    23385.46 |
|                    3 |         19490.55 |          31355.46 |    11864.91 |
|                   14 |         12036.27 |          20825.87 |     8789.59 |
|                   12 |          9482.09 |           18035.3 |     8553.21 |
|                   16 |          6812.47 |          11365.48 |     4553.01 |
|                   17 |           1373.3 |           3093.01 |     1719.71 |
|                    8 |           371.61 |            836.97 |      465.36 |
|                   10 |           245.62 |             553.2 |      307.58 |
|                    4 |           244.15 |             549.9 |      305.74 |
+----------------------+------------------+-------------------+-------------+
10 rows in set (0.00 sec)
```

GrossProfit>1000

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> WHERE ProductSubcategoryKey IN (2,1,3,14,12,16,17)
    -> GROUP BY ProductSubcategoryKey
    -> ORDER BY GrossProfit DESC
    -> LIMIT 10;
+----------------------+------------------+-------------------+-------------+
| ProductSubcategoryKey | TotalProductCost | TotalProductPrice | GrossProfit |
+----------------------+------------------+-------------------+-------------+
|                    2 |         40130.43 |          65774.45 |    25644.02 |
|                    1 |         28998.84 |          52384.29 |    23385.46 |
|                    3 |         19490.55 |          31355.46 |    11864.91 |
|                   14 |         12036.27 |          20825.87 |     8789.59 |
|                   12 |          9482.09 |           18035.3 |     8553.21 |
|                   16 |          6812.47 |          11365.48 |     4553.01 |
|                   17 |           1373.3 |           3093.01 |     1719.71 |
+----------------------+------------------+-------------------+-------------+
7 rows in set (0.00 sec)
```

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->  Round(SUM(ProductCost),2) AS TotalProductCost,
    ->  Round(SUM(ProductPrice),2) AS TotalProductPrice,
    ->     ABS(ROUND(SUM(ProductCost - ProductPrice),2)) AS GrossProfit
    -> FROM Products
    -> WHERE ProductSubcategoryKey IN (2,1,3,14,12,16,17)
    -> GROUP BY ProductSubcategoryKey
    -> HAVING GrossProfit > 5000
    -> ORDER BY GrossProfit DESC
    -> LIMIT 10;
+----------------------+-----------------+-------------------+-------------+
| ProductSubcategoryKey | TotalProductCost | TotalProductPrice | GrossProfit |
+----------------------+-----------------+-------------------+-------------+
|                    2 |        40130.43 |          65774.45 |    25644.02 |
|                    1 |        28998.84 |          52384.29 |    23385.46 |
|                    3 |        19490.55 |          31355.46 |    11864.91 |
|                   14 |        12036.27 |          20825.87 |     8789.59 |
|                   12 |         9482.09 |           18035.3 |     8553.21 |
+----------------------+-----------------+-------------------+-------------+
5 rows in set (0.00 sec)
```

Challenge 1

Find the total income of customers grouped by gender, but only include those with a total income greater than 1,000,000.

```
mysql> SELECT
    ->  Gender,
    ->     SUM(AnnualIncome) AS TotalIncome
    -> FROM Customers
    -> GROUP BY Gender
    -> HAVING TotalIncome > 100000
    -> ORDER BY 2 DESC;
+--------+-------------+
| Gender | TotalIncome |
+--------+-------------+
| F      |    58570000 |
| M      |    58080000 |
| NA     |      840000 |
+--------+-------------+
3 rows in set (0.01 sec)
```

Challenge 2

Find the minimum and maximum income for each educational level, but only include groups where the difference between the maximum and minimum income is more than 50,000.

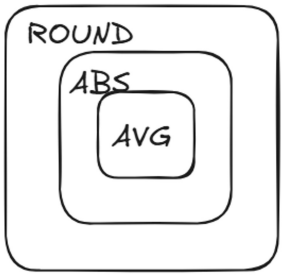
```
mysql> SELECT
    ->  EducationLevel,
    ->     MIN(AnnualIncome) AS Min_Income,
    ->     MAX(AnnualIncome) AS Max_Income,
    ->     ABS(MIN(AnnualIncome) - MAX(AnnualIncome)) AS Abs_Income
    -> FROM Customers
    -> GROUP BY 1
    -> HAVING Abs_Income > 50000
    -> ORDER BY 4 DESC;
+--------------------+------------+------------+------------+
| EducationLevel     | Min_Income | Max_Income | Abs_Income |
+--------------------+------------+------------+------------+
| Bachelors          |      10000 |     170000 |     160000 |
| Partial College    |      10000 |     170000 |     160000 |
| High School        |      10000 |     170000 |     160000 |
| Graduate Degree    |      10000 |     170000 |     160000 |
| Partial High School |      10000 |     160000 |     150000 |
+--------------------+------------+------------+------------+
5 rows in set (0.01 sec)
```

Challenge 3

Find the average cost, price, and price difference for each product subcategory where the average price difference is more than 20, and order the results in descending order of the price difference.

```
mysql> DESC Products;
+-----------------------+
| Field                 |
+-----------------------+
| ProductKey            |
| ProductSubcategoryKey |
| ProductSKU            |
| ProductName           |
| ModelName             |
| ProductDescription    |
| ProductColor          |
| ProductSize           |
| ProductStyle          |
| ProductCost           |
| ProductPrice          |
```

```
mysql> SELECT
    ->  ProductSubcategoryKey,
    ->     ROUND(AVG(ProductCost),2) AS AvgCost,
    ->     ROUND(AVG(ProductPrice),2) AS AvgPrice,
    ->     ROUND(ABS(AVG(ProductCost - ProductPrice)),2) AS Avg_Price_Diff
    -> FROM Products
    -> GROUP BY 1
    -> HAVING Avg_Price_Diff > 20
    -> ORDER BY 4 DESC;
+----------------------+---------+----------+----------------+
| ProductSubcategoryKey | AvgCost | AvgPrice | Avg_Price_Diff |
+----------------------+---------+----------+----------------+
|                    1 |  906.21 |  1637.01 |          730.8 |
|                    2 |  933.27 |  1529.64 |         596.37 |
|                    3 |  885.93 |  1425.25 |         539.31 |
|                   12 |  338.65 |   644.12 |         305.47 |
|                   14 |  388.27 |    671.8 |         283.54 |
|                   16 |  378.47 |   631.42 |         252.95 |
|                    8 |  123.87 |   278.99 |         155.12 |
|                   17 |   98.09 |   220.93 |         122.84 |
|                   10 |   81.87 |    184.4 |         102.53 |
|                   27 |   59.47 |      159 |          99.53 |
|                   26 |   44.88 |      120 |          75.12 |
|                   35 |   51.56 |      125 |          73.44 |
|                    6 |   47.29 |    106.5 |          59.21 |
|                    9 |   47.27 |   106.48 |           59.2 |
|                   18 |   37.12 |    89.99 |          52.87 |
|                    5 |   40.95 |    92.24 |          51.29 |
|                   11 |   38.66 |    87.07 |          48.41 |
|                   24 |   30.93 |    74.99 |          44.06 |
|                   25 |   23.75 |     63.5 |          39.75 |
|                   22 |   25.36 |    64.28 |          38.92 |
|                    4 |   30.52 |    68.74 |          38.22 |
|                   13 |   28.42 |    64.02 |          35.59 |
|                   32 |   20.57 |    54.99 |          34.42 |
|                   15 |    17.6 |    39.63 |          22.04 |
|                   31 |   12.38 |    34.09 |          21.71 |
+----------------------+---------+----------+----------------+
25 rows in set (0.01 sec)
```



Challenge 4

Find the total products, total cost, total price, and rounded profit for each product key and subcategory key. Order the results by profit in descending order.

```
mysql> DESC Products;
+-----------------------+
| Field                 |
+-----------------------+
| ProductKey            |
| ProductSubcategoryKey |
| ProductSKU            |
| ProductName           |
| ModelName             |
| ProductDescription    |
| ProductColor          |
| ProductSize           |
| ProductStyle          |
| ProductCost           |
| ProductPrice          |
+-----------------------+
```

```sql
SELECT
    ProductKey,
    ProductSubcategoryKey,
    COUNT(ProductKey) As TotalProducts,
    SUM(ProductCost) AS TotalCost,
    SUM(ProductPrice) AS TotalPrice,
    ROUND(ABS(SUM(ProductCost - ProductPrice)),2) AS RoundedProfit
FROM Products
GROUP BY 1,2
ORDER BY 6 DESC
LIMIT 10;
```

```
mysql> SELECT
    ->  ProductKey,
    ->     ProductSubcategoryKey,
    ->     COUNT(ProductKey) As TotalProducts,
    ->     SUM(ProductCost) AS TotalCost,
    ->     SUM(ProductPrice) AS TotalPrice,
    ->     ROUND(ABS(SUM(ProductCost - ProductPrice)),2) AS RoundedProfit
    -> FROM Products
    -> GROUP BY 1,2
    -> ORDER BY 6 DESC
    -> LIMIT 10;
+------------+----------------------+---------------+-----------+------------+---------------+
| ProductKey | ProductSubcategoryKey | TotalProducts | TotalCost | TotalPrice | RoundedProfit |
+------------+----------------------+---------------+-----------+------------+---------------+
|        346 |                    1 |             1 | 1912.1544 |    3399.99 |       1487.84 |
|        347 |                    1 |             1 | 1912.1544 |    3399.99 |       1487.84 |
|        344 |                    1 |             1 | 1912.1544 |    3399.99 |       1487.84 |
|        345 |                    1 |             1 | 1912.1544 |    3399.99 |       1487.84 |
|        349 |                    1 |             1 | 1898.0944 |    3374.99 |        1476.9 |
|        351 |                    1 |             1 | 1898.0944 |    3374.99 |        1476.9 |
|        350 |                    1 |             1 | 1898.0944 |    3374.99 |        1476.9 |
|        348 |                    1 |             1 | 1898.0944 |    3374.99 |        1476.9 |
|        314 |                    2 |             1 | 2171.2942 |    3578.27 |       1406.98 |
|        312 |                    2 |             1 | 2171.2942 |    3578.27 |       1406.98 |
+------------+----------------------+---------------+-----------+------------+---------------+
10 rows in set (0.00 sec)
```

Challenge 5

Find the total absolute profit for each product subcategory, only including those with a profit difference greater than 5,000. Order the results by total profit in descending order.

```sql
SELECT
    ProductSubcategoryKey,
    ROUND(ABS(SUM(ProductCost - ProductPrice)),2) AS RoundedProfit
FROM Products
GROUP BY 1
Having RoundedProfit > 5000
ORDER BY 2 DESC
LIMIT 10;
```

```
mysql> SELECT
    ->     ProductSubcategoryKey,
    ->     ROUND(ABS(SUM(ProductCost - ProductPrice)),2) AS RoundedProfit
    -> FROM Products
    -> GROUP BY 1
    -> Having RoundedProfit > 5000
    -> ORDER BY 2 DESC
    -> LIMIT 10;
+----------------------+---------------+
| ProductSubcategoryKey | RoundedProfit |
+----------------------+---------------+
|                    2 |      25644.02 |
|                    1 |      23385.46 |
|                    3 |      11864.91 |
|                   14 |       8789.59 |
|                   12 |       8553.21 |
+----------------------+---------------+
5 rows in set (0.00 sec)
```