String and Aggregate Functions

🎯 Session Objectives:

- 🔠 Common string functions: LENGTH, TRIM, CONCAT, UPPER, LOWER, LEFT, RIGHT, REVERSE, REPLACE, etc.
- 🧹 Data Cleaning
- 🔧 Data Manipulation
- 🔍 Substring Extraction
- ✅ 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.

```
330 •  SELECT
331        Replace(ProductStyle, 0, 'N/A') as ProductStyle
332    FROM Products;
333
334
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

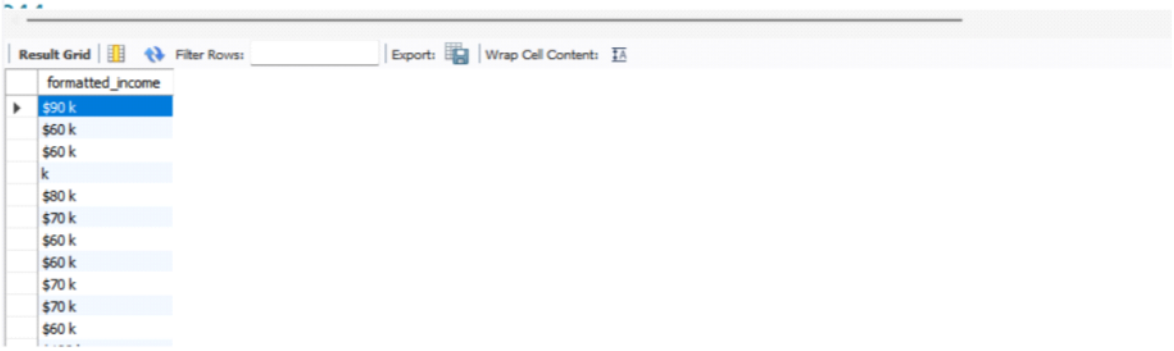
ProductStyle
N/A
N/A
U
U
N/A
U
U
U
U
U
U

```
334 •  SELECT
335        DISTINCT ProductStyle
336    FROM Products;
```

Result Grid | Filter Rows: | Export:

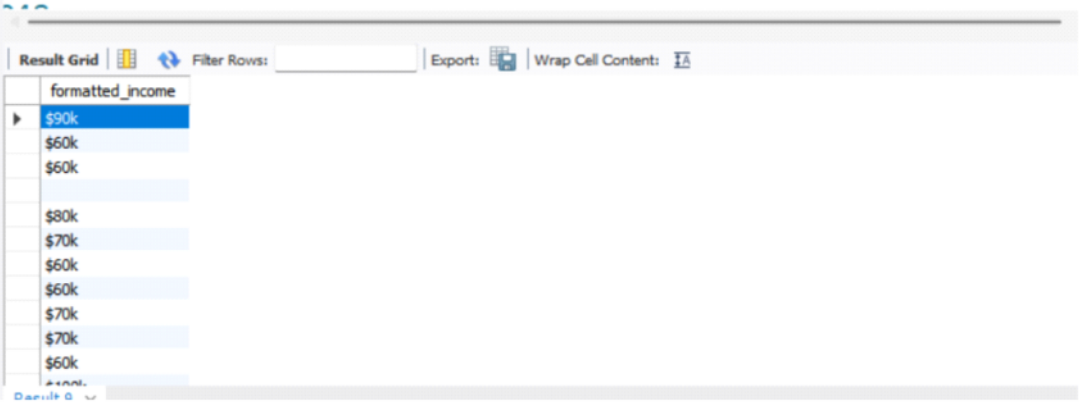
ProductStyle
0
U
W
M

```sql
SELECT
    CONCAT(REPLACE(AnnualIncome,',000',''), 'k') AS formatted_income
FROM Customers;
```



formatted_income
$90 k
$60 k
$60 k
k
$80 k
$70 k
$60 k
$60 k
$70 k
$70 k
$60 k

```sql
SELECT
    REPLACE(AnnualIncome,',000','k') AS formatted_income
FROM Customers;
```



formatted_income
$90k
$60k
$60k
$80k
$70k
$60k
$60k
$70k
$70k
$60k

```sql
-- Clean the AnnualIncome Column and transform to integer data type
SELECT * FROM Customers;

SET SQL_SAFE_UPDATES = 0;

UPDATE Customers
SET AnnualIncome = REPLACE(REPLACE(AnnualIncome , '$',''),',','');

-- OR
/*
        UPDATE Customers
        SET AnnualIncome = REPLACE(AnnualIncome , '$' , '');

        UPDATE Customers
        SET AnnualIncome = REPLACE(AnnualIncome , ',' , '');
*/
```

CustomerKey	Prefix	FirstName	LastName	FullName	DateOfBirth	MaritalStatus	EmailAddress	Gender	Regions	AnnualIncome
11000	MR.	JON	YANG	JON YANG	04/08/1966	M	jon24@learnsector.com	M	NULL	90000
11001	MR.	EUGENE	HUANG	EUGENE HUANG	14/05/1965	S	eugene10@learnsector.com	M	NULL	60000
11002	MR.	RUBEN	TORRES	RUBEN TORRES	08/12/1965	M	ruben35@learnsector.com	M	NULL	60000
11003	MS.	CHRISTY	ZHU	CHRISTY ZHU	15/02/1968	S	christy12@learnsector.com	F	NULL	
11004	MRS.	ELIZABETH	JOHNSON	ELIZABETH JOHNSON	08/08/1968	S	elizabeth5@learnsector.com	F	NULL	80000
11005	MR.	JULIO	RUIZ	JULIO RUIZ	08/05/1965	S	julio1@learnsector.com	M	NULL	70000
11007	MR.	MARCO	MEHTA	MARCO MEHTA	05/09/1964	M	marco14@learnsector.com	M	NULL	60000
11008	MRS.	ROBIN	VERHOFF	ROBIN VERHOFF	07/07/1964	S	rob4@learnsector.com	F	NULL	60000
11009	MR.	SHANNON	CARLSON	SHANNON CARLSON	04/01/1964	S	shannon38@learnsector.com	M	NULL	70000
11010	MS.	JACQUELYN	SUAREZ	JACQUELYN SUAREZ	02/06/1964	S	jacquelyn20@learnsector.com	F	NULL	70000

```
966 •  DESC Customers;
```

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
CustomerKey	int	YES		NULL	
Prefix	text	YES		NULL	
FirstName	varchar(50)	YES		NULL	
LastName	varchar(50)	YES		NULL	
FullName	varchar(100)	YES		NULL	
DateOfBirth	text	YES		NULL	
MaritalStatus	text	YES		NULL	
EmailAddress	varchar(100)	YES		NULL	
Gender	text	YES		NULL	
Regions	varchar(50)	YES		NULL	
AnnualIncome	text	YES		NULL	
TotalChildren	int	YES		NULL	
EducationLevel	text	YES		NULL	
Occupation	text	YES		NULL	
HomeOwner	text	YES		NULL	
Phone_number	bigint	YES		NULL	

Transform the Data Type to Integer.

```
966 •  UPDATE Customers
967    SET AnnualIncome = NULL
968    WHERE AnnualIncome = '';
969
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerKey	Prefix	FirstName	LastName	FullName	DateOfBirth	MaritalStatus	EmailAddress	Gender	Regions	AnnualIncome
11000	MR.	JON	YANG	JON YANG	04/08/1966	M	jon24@learnsector.com	M	NULL	90000
11001	MR.	EUGENE	HUANG	EUGENE HUANG	14/05/1965	S	eugene10@learnsector.com	M	NULL	60000
11002	MR.	RUBEN	TORRES	RUBEN TORRES	08/12/1965	M	ruben35@learnsector.com	M	NULL	60000
11003	MS.	CHRISTY	ZHU	CHRISTY ZHU	15/02/1968	S	christy12@learnsector.com	F	NULL	NULL
11004	MRS.	ELIZABETH	JOHNSON	ELIZABETH JOHNSON	08/08/1968	S	elizabeth5@learnsector.com	F	NULL	80000
11005	MR.	JULIO	RUIZ	JULIO RUIZ	08/05/1965	S	julio1@learnsector.com	M	NULL	70000
11007	MR.	MARCO	MEHTA	MARCO MEHTA	05/09/1964	M	marco14@learnsector.com	M	NULL	60000
11008	MRS.	ROBIN	VERHOFF	ROBIN VERHOFF	07/07/1964	S	rob4@learnsector.com	F	NULL	60000

```
972 •  ALTER TABLE Customers
973    MODIFY COLUMN AnnualIncome INT;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
MaritalStatus	text	YES		NULL	
EmailAddress	varchar(100)	YES		NULL	
Gender	text	YES		NULL	
Regions	varchar(50)	YES		NULL	
AnnualIncome	int	YES		NULL	
TotalChildren	int	YES		NULL	
EducationLevel	text	YES		NULL	
Occupation	text	YES		NULL	
HomeOwner	text	YES		NULL	

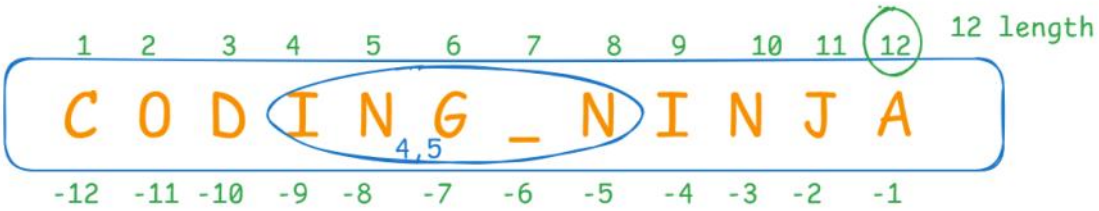
SQL - Index starts from 1.

Substring → A Part of a String.

```
375    -- SUBSTRING
376    -- SYNTAX : SUBSTR() OR SUBSTRING('str',start_idx,length_of_characters)
377
378 •  SELECT
379        SUBSTRING("Hello World" ,1,5) AS substr; -- having 11 length
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

substr
Hello




```
381 •  SELECT
382        SUBSTR("Coding Ninja", 4,5);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SUBSTR("Coding Ninja", 4,5)
ing N

```
384 •  SELECT
385        SUBSTRING("Hello World" ,-5,5) AS substr; -- 'World'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

substr
World

```
387 •  SELECT
388        SUBSTRING("Hello World" ,-11,50) AS substr; -- 'Hello World'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

substr
Hello World

```
390 •  SELECT
391        SUBSTRING("Hello World" ,-9,5) AS substr; -- 'llo W'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

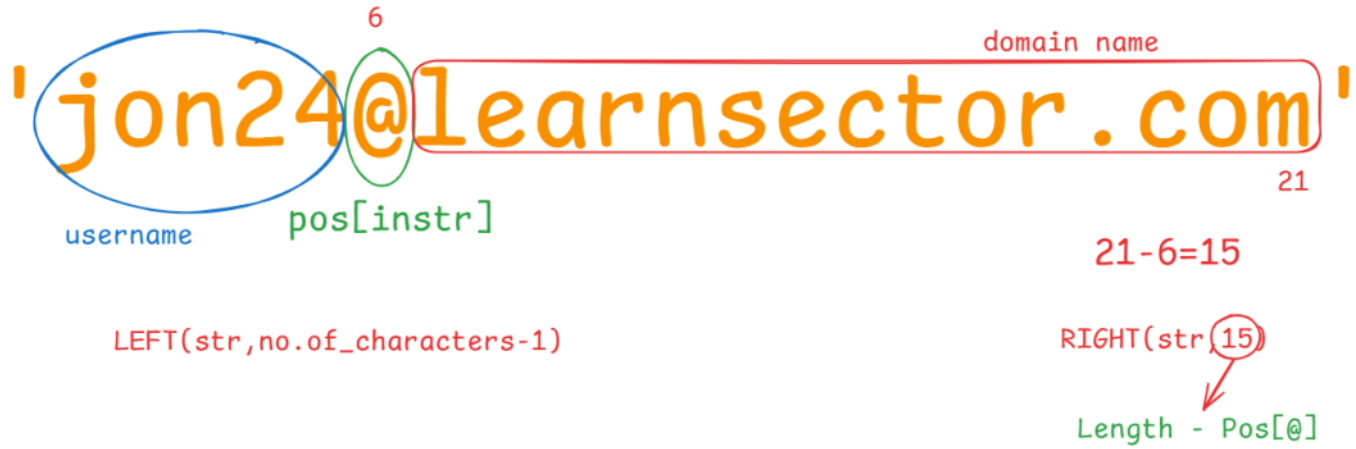
substr
llo W

```
395 •  SELECT
396        ProductKey,
397        SUBSTR(ProductSKU,1,7) AS Product_SKU,
398        ProductColor
399    FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductKey	Product_SKU	ProductColor
214	HL-U509	Red
215	HL-U509	Black
218	SO-B909	White
219	SO-B909	White
220	HL-U509	Blue
223	CA-1098	Multi
226	LJ-0192	Multi
229	LJ-0192	Multi
232	LJ-0192	Multi
235	LJ-0192	Multi
238	FR-R92R	Red
241	FR-R92R	Red

INSTR -> Find the Substring Position.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerKey	Prefix	FirstName	LastName	FullName	DateOfBirth	MaritalStatus	EmailAddress
11000	MR.	JON	YANG	JON YANG	04/08/1966	M	jon24@learnsector.com
11001	MR.	EUGENE	HUANG	EUGENE HUANG	14/05/1965	S	eugene10@learnsector.com
11002	MR.	RUBEN	TORRES	RUBEN TORRES	08/12/1965	M	ruben35@learnsector.com
11003	MS.	CHRISTY	ZHU	CHRISTY ZHU	15/02/1968	S	christy12@learnsector.com
11004	MRS.	ELIZABETH	JOHNSON	ELIZABETH JOHNSON	08/08/1968	S	elizabeth5@learnsector.com
11005	MR.	JULIO	RUIZ	JULIO RUIZ	08/05/1965	S	julio1@learnsector.com
11007	MR.	MARCO	MEHTA	MARCO MEHTA	05/09/1964	M	marco14@learnsector.com
11008	MRS.	ROBIN	VERHOFF	ROBIN VERHOFF	07/07/1964	S	rob4@learnsector.com
11009	MR.	SHANNON	CARLSON	SHANNON CARLSON	04/01/1964	S	shannon38@learnsector.com
11010	MS.	JACQUELYN	SUAREZ	JACQUELYN SUAREZ	02/06/1964	S	jacquelyn20@learnsector.com
11011	MR.	CURTIS	LU	CURTIS LU	11/04/1963	M	curtis9@learnsector.com

```sql
SELECT
    EmailAddress,
    INSTR(EmailAddress, '@') AS at_position
FROM Customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmailAddress	at_position
jon24@learnsector.com	6
eugene10@learnsector.com	9
ruben35@learnsector.com	8
christy12@learnsector.com	10
elizabeth5@learnsector.com	11
julio1@learnsector.com	7
marco14@learnsector.com	8
rob4@learnsector.com	5
shannon38@learnsector.com	10
jacquelyn20@learnsector.com	12
curtis9@learnsector.com	8
lauren41@learnsector.com	9

7 18

Mumbai#Maharastra@India

18-7 = 11 - 1 = 10

Aggregation with Group By

Summary Table

Count

[Filter (having Clause)]

```sql
-- Find the Total Customers Counts.
SELECT
    COUNT(*) AS TotalCounts
FROM Customers;

SELECT
    COUNT(CustomerKey) AS TotalCounts
FROM Customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TotalCounts
2062

```sql
SELECT DISTINCT Occupation FROM Customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation
Professional
Management
Skilled Manual
Clerical
Manual

```sql
SELECT
    COUNT(DISTINCT Occupation) AS Unique_Occupation_Count
FROM Customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Unique_Occupation_Count
5

Result Grid | Filter Rows:

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

→

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

```sql
SELECT
    Occupation,
    COUNT(CustomerKey) AS Total_People
FROM Customers
GROUP BY 1;
```

Indexing....

```sql
SELECT
    Occupation,
    COUNT(CustomerKey) AS Total_People
FROM Customers
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

```sql
SELECT
    Occupation,
    COUNT(CustomerKey) AS Total_People
FROM Customers
GROUP BY Occupation
ORDER BY Total_People DESC;
```

```sql
-- Find the TotalPrice and TotalCost of Products
SELECT
    SUM(ProductPrice) AS TotalPrice,
    SUM(ProductCost) AS TotalCost
FROM Products;
```

SUM

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TotalPrice	TotalCost
209330.1454999998	121202.67569999998

```sql
-- Find the TotalPrice and TotalCost of Products
SELECT
    CAST(SUM(ProductPrice) AS DECIMAL(8,2))AS TotalPrice,
    ROUND(SUM(ProductCost),2) AS TotalCost
FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TotalPrice	TotalCost
209330.15	121202.68

```sql
-- Find the Gross Profit of all Products;
SELECT
    ROUND(SUM(ProductPrice - ProductCost),2) AS GrossProfit
FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

GrossProfit
88127.47

```sql
SELECT
    ROUND(SUM(ProductPrice) - SUM(ProductCost) , 2) AS GrossProfit
FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

GrossProfit
88127.47

Average

```sql
-- AVERAGE [AVG]
SELECT
    AVG(ProductCost) AS AvgCost,
    AVG(ProductPrice) AS AvgPrice
FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AvgCost	AvgPrice
413.661009215017	714.4373566552895

```sql
-- AVERAGE [AVG]
SELECT
    ROUND(AVG(ProductCost),2) AS AvgCost,
    ROUND(AVG(ProductPrice),2) AS AvgPrice
FROM Products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AvgCost	AvgPrice
413.66	714.44

```sql
-- Find the Average AnnualIncome based on Gender
-- Hint : Group By['Gender'] & Agg['AnnualIncome']

SELECT
    Gender,
    AVG(AnnualIncome) AS AvgIncome
FROM Customers
GROUP BY Gender
ORDER BY AvgIncome DESC;
```

Gender	AvgIncome
F	57534.3811
M	57165.3543
NA	46666.6667

```sql
SELECT
    Gender,
    AVG(AnnualIncome) AS AvgIncome,
    COUNT(CustomerKey) AS Total_Count
FROM Customers
GROUP BY Gender
ORDER BY AvgIncome DESC;
```

Gender	AvgIncome	Total_Count
F	57534.3811	1023
M	57165.3543	1021
NA	46666.6667	18

```sql
SELECT
    Gender,
    ROUND(AVG(AnnualIncome),0) AS AvgIncome,
    COUNT(CustomerKey) AS Total_Count
FROM Customers
GROUP BY Gender
ORDER BY AvgIncome DESC;
```

Gender	AvgIncome	Total_Count
F	57534	1023
M	57165	1021
NA	46667	18

```sql
SELECT
    Occupation,
    EducationLevel,
    ROUND(AVG(AnnualIncome),0) AS AvgIncome
FROM Customers
GROUP BY Occupation , EducationLevel
ORDER BY AvgIncome DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation	EducationLevel	AvgIncome
Management	Partial High School	143333
Management	Partial College	131111
Professional	Partial High School	114545
Management	High School	107188
Management	Graduate Degree	99593
Management	Bachelors	93416
Professional	Partial College	86645
Professional	High School	76709
Skilled Manual	Partial High School	70000
Professional	Graduate Degree	67479
Professional	Bachelors	66923
Skilled Manual	Partial College	58734

Result 56

```sql
SELECT
    Occupation,
    EducationLevel,
    ROUND(AVG(AnnualIncome),0) AS AvgIncome,
    COUNT(*) AS TotalCount
FROM Customers
GROUP BY Occupation , EducationLevel
ORDER BY AvgIncome DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation	EducationLevel	AvgIncome	TotalCount
Management	Partial High School	143333	3
Management	Partial College	131111	9
Professional	Partial High School	114545	11
Management	High School	107188	32
Management	Graduate Degree	99593	123
Management	Bachelors	93416	163
Professional	Partial College	86645	156
Professional	High School	76709	79
Skilled Manual	Partial High School	70000	23
Professional	Graduate Degree	67479	119
Professional	Bachelors	66923	196
Skilled Manual	Partial College	58734	159

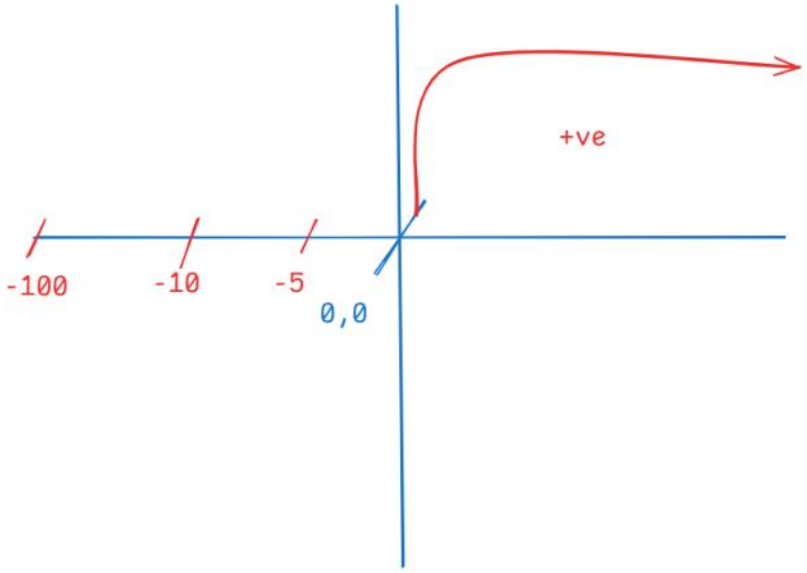
```sql
SELECT
    ProductColor,
    ROUND(AVG(ProductPrice) - AVG(ProductCost) ,2) AS AvgProfit
FROM Products
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductColor	AvgProfit
Red	535.54
Silver	446.95
Yellow	360.84
Blue	356.51
Black	316.37
Grey	73.44
Silver/Black	35.59
NA	30.99
Multi	28.36
White	5.87

$|-99.99| = 99.99$ abs()

-5 > -110 > -1111 > -111111




```sql
SELECT
    ProductColor,
    ROUND(AVG(ProductCost) - AVG(ProductPrice) ,2) AS AvgProfit
FROM Products
GROUP BY 1
ORDER BY 2 DESC;
```

ABS()

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductColor	AvgProfit
White	-5.87
Multi	-28.36
NA	-30.99
Silver/Black	-35.59
Grey	-73.44
Black	-316.37
Blue	-356.51
Yellow	-360.84
Silver	-446.95
Red	-535.54

```
209 •  SELECT
210        ProductColor,
211        ABS(ROUND(AVG(ProductCost) - AVG(ProductPrice) ,2)) AS AvgProfit
212    FROM Products
213    GROUP BY 1
214    ORDER BY 2 DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductColor	AvgProfit
Red	535.54
Silver	446.95
Yellow	360.84
Blue	356.51
Black	316.37
Grey	73.44
Silver/Black	35.59
NA	30.99
Multi	28.36
White	5.87

MAX()/MIN()

```
221 •  SELECT DISTINCT MaritalStatus FROM Customers; # 2
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MaritalStatus
M
S

```
223 •  SELECT
224        MaritalStatus,
225        Gender,
226        MAX(TotalChildren) AS Max_Children
227    FROM Customers
228    GROUP BY 1,2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MaritalStatus	Gender	Max_Children
M	M	5
S	M	5
S	F	5
M	F	5
M	NA	2
S	NA	5

```
SELECT
    ProductName,
    MAX(ProductPrice) AS Max_Price
FROM Products
GROUP BY ProductName
ORDER BY Max_Price DESC
LIMIT 5;
```

ProductName	Max_Price
Road-150 Red, 62	3578.27
Road-150 Red, 44	3578.27
Road-150 Red, 48	3578.27
Road-150 Red, 52	3578.27
Road-150 Red, 56	3578.27

```
SELECT
    EducationLevel,
    Occupation,
    MAX(AnnualIncome) AS Max_Income,
    MIN(AnnualIncome) AS Min_Income
FROM Customers
GROUP BY 1,2;
```

EducationLevel	Occupation	Max_Income	Min_Income
Bachelors	Professional	90000	40000
Bachelors	Management	170000	40000
Partial College	Skilled Manual	90000	20000
High School	Skilled Manual	80000	10000
Partial College	Clerical	40000	30000
Partial High School	Clerical	40000	10000
Graduate Degree	Management	170000	50000
Partial College	Professional	170000	40000
High School	Professional	170000	30000
Partial High School	Skilled Manual	90000	30000
Graduate Degree	Manual	20000	10000
Graduate Degree	Clerical	40000	10000
Bachelors	Manual	10000	10000
Partial College	Manual	20000	10000
Bachelors	Clerical	40000	10000
High School	Manual	30000	10000
Partial College	Management	170000	100000
High School	Management	170000	80000
Partial High School	Professional	120000	100000
Partial High School	Management	160000	120000
Bachelors	Skilled Manual	80000	30000
Partial High School	Manual	20000	10000
Graduate Degree	Skilled Manual	80000	40000
High School	Clerical	30000	30000
Graduate Degree	Professional	130000	60000

GROUP_CONCAT

```
259 •  SELECT * FROM ProductCategories;
260
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

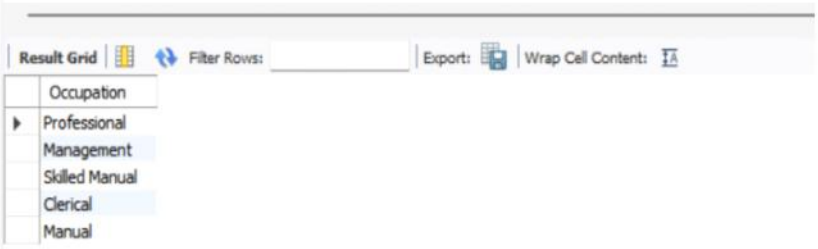
ProductCategoryKey	CategoryName
1	Bikes
2	Components
3	Clothing
4	Accessories

```
261 •  SELECT GROUP_CONCAT(CategoryName) AS Categories
262    FROM ProductCategories;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

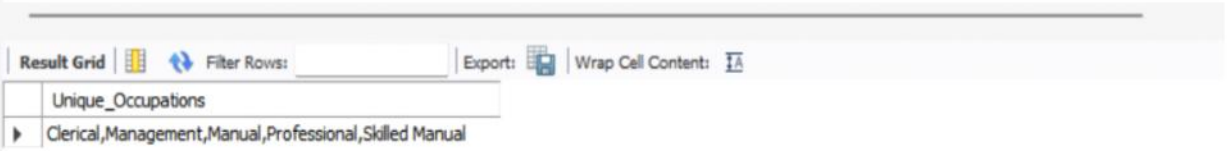
Cateogries
Bikes,Components,Clothing,Accessories

```sql
-- Occupation - 5 unique
SELECT DISTINCT Occupation FROM Customers;
```



Occupation
Professional
Management
Skilled Manual
Clerical
Manual

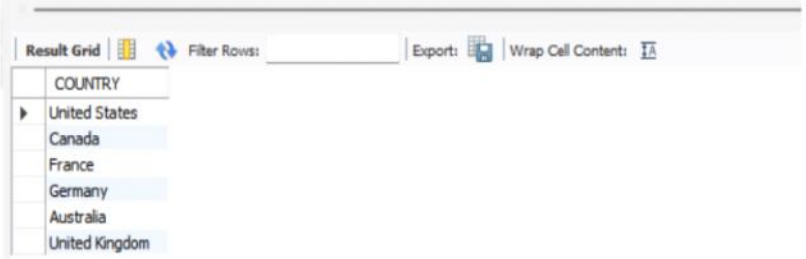
```sql
SELECT
    GROUP_CONCAT(DISTINCT Occupation) AS Unique_Occupations
FROM Customers;
```



Unique_Occupations
Clerical,Management,Manual,Professional,Skilled Manual

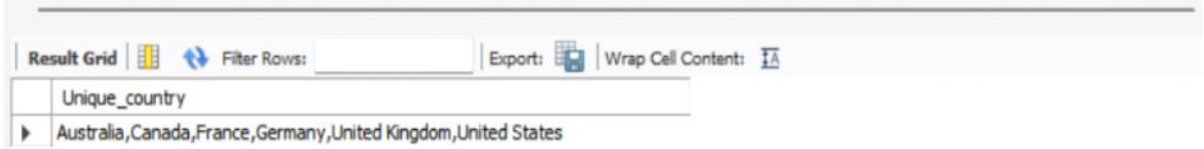
```sql
-- Territories
SELECT  * FROM Territories;

SELECT DISTINCT COUNTRY FROM Territories;
```



COUNTRY
United States
Canada
France
Germany
Australia
United Kingdom

```sql
SELECT GROUP_CONCAT(DISTINCT Country) AS Unique_country
FROM Territories;
```



Unique_country
Australia,Canada,France,Germany,United Kingdom,United States

Having Clause

Filtering the summary table [Group By]

```
SELECT
    EducationLevel,
    Occupation,
    MAX(AnnualIncome) AS Max_Income,
    MIN(AnnualIncome) AS Min_Income
FROM Customers
WHERE CLAUSE <ORIGINAL TABLE FILTER>
GROUP BY 1,2
HAVING CLAUSE <SUMMARY TABLE FILTER>;
```

```
282 •  SELECT
283        Occupation,
284        EducationLevel,
285        ROUND(AVG(AnnualIncome),0) AS AvgIncome,
286        COUNT(*) AS TotalCount
287    FROM Customers
288    GROUP BY Occupation , EducationLevel
289    HAVING AvgIncome > 50000
290    ORDER BY AvgIncome DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation	EducationLevel	AvgIncome	TotalCount
Skilled Manual	Partial College	58734	159
Skilled Manual	Graduate Degree	56415	106
Skilled Manual	Bachelors	47612	135
Skilled Manual	High School	38534	117
Clerical	Partial College	35422	167
Clerical	Graduate Degree	31270	63
Clerical	High School	30000	6
Clerical	Bachelors	27955	88
Clerical	Partial High School	24231	26
Manual	High School	18241	108
Manual	Partial High School	15763	59
Manual	Partial College	15000	94
Manual	Graduate Degree	11429	7
Manual	Bachelors	10000	13

BEFORE

```sql
SELECT
    Occupation,
    EducationLevel,
    ROUND(AVG(AnnualIncome),0) AS AvgIncome,
    COUNT(*) AS TotalCount
FROM Customers
GROUP BY Occupation , EducationLevel
HAVING AvgIncome > 50000
ORDER BY AvgIncome DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation	EducationLevel	AvgIncome	TotalCount
Management	Partial High School	143333	3
Management	Partial College	131111	9
Professional	Partial High School	114545	11
Management	High School	107188	32
Management	Graduate Degree	99593	123
Management	Bachelors	93416	163
Professional	Partial College	86645	156
Professional	High School	76709	79
Skilled Manual	Partial High School	70000	23
Professional	Graduate Degree	67479	119
Professional	Bachelors	66923	196
Skilled Manual	Partial College	58734	159
Skilled Manual	Graduate Degree	56415	106

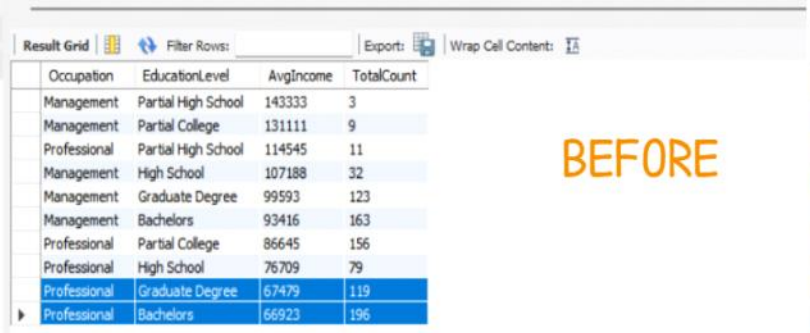
```sql
SELECT
    Occupation,
    EducationLevel,
    ROUND(AVG(AnnualIncome),0) AS AvgIncome,
    COUNT(*) AS TotalCount
FROM Customers
WHERE Occupation IN ('Management','Professional')
GROUP BY Occupation , EducationLevel
ORDER BY AvgIncome DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation	EducationLevel	AvgIncome	TotalCount
Management	Partial High School	143333	3
Management	Partial College	131111	9
Professional	Partial High School	114545	11
Management	High School	107188	32
Management	Graduate Degree	99593	123
Management	Bachelors	93416	163
Professional	Partial College	86645	156
Professional	High School	76709	79
Professional	Graduate Degree	67479	119
Professional	Bachelors	66923	196

```sql
SELECT
    Occupation,
    EducationLevel,
    ROUND(AVG(AnnualIncome),0) AS AvgIncome,
    COUNT(*) AS TotalCount
FROM Customers
WHERE Occupation IN ('Management','Professional')
GROUP BY Occupation , EducationLevel
HAVING AvgIncome > 75000
ORDER BY AvgIncome DESC;
```

Where Clause applies filter on Original Table. Whereas Having Clause works on Summary Table and called after group by and before order by.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Occupation	EducationLevel	AvgIncome	TotalCount
Management	Partial High School	143333	3
Management	Partial College	131111	9
Professional	Partial High School	114545	11
Management	High School	107188	32
Management	Graduate Degree	99593	123
Management	Bachelors	93416	163
Professional	Partial College	86645	156
Professional	High School	76709	79
Professional	Graduate Degree	67479	119
Professional	Bachelors	66923	196

BEFORE

AFTER

Occupation	EducationLevel	AvgIncome	TotalCount
Management	Partial High School	143333	3
Management	Partial College	131111	9
Professional	Partial High School	114545	11
Management	High School	107188	32
Management	Graduate Degree	99593	123
Management	Bachelors	93416	163
Professional	Partial College	86645	156
Professional	High School	76709	79