Window Functions-II

Session Goals

- ✅ Understand what window functions are and when to use them.
- ✅ Break down and apply the syntax of common window functions like ROW_NUMBER(), SUM(), AVG(), etc.
- ✅ Differentiate window functions from regular aggregate functions.
- ✅ Use analytical window functions to extract advanced insights.
- ✅ Compare rows without self-joins.
- ✅ Partition and rank data meaningfully.
- ✅ Detect trends, group data into tiles, and calculate change over time.

Football Analogy

Aggregate

- We wanted to know the total goals per match

Window Function [Aggregate]

- Here, we need to know each and individual player performance of a match.

-- Find the Maximum AnnualIncome based on EducationLevel & Occupation?

```
216 •  SELECT DISTINCT EducationLevel FROM Customers;
```

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

EducationLevel
Bachelors
Partial College
High School
Partial High School
Graduate Degree

```
217 •  SELECT DISTINCT Occupation FROM Customers;
```

M:N -> 5*5 = 25 unique filters

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

Occupation
Professional
Management
Skilled Manual
Clerical
Manual

```
19 •  SELECT DISTINCT EducationLevel,Occupation FROM Customers; -- 25 row(s) returned
20
```

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

EducationLevel	Occupation
Bachelors	Professional
Bachelors	Management
Bachelors	Manual
Bachelors	Clerical
Bachelors	Skilled Manual
Graduate Degree	Management
Graduate Degree	Manual
Graduate Degree	Clerical
Graduate Degree	Skilled Manual
Graduate Degree	Professional
High School	Skilled Manual
High School	Professional
High School	Manual
High School	Management
High School	Clerical
Partial College	Skilled Manual
Partial College	Clerical
Partial College	Professional
Partial College	Manual

```
SELECT
    CustomerKey,
    FullName,
    EducationLevel,
    Occupation,
    AnnualIncome,
    MAX(AnnualIncome) OVER(
        PARTITION BY EducationLevel , Occupation
    ) AS Max_income_by_edu_occ
FROM Customers;
```

CustomerKey	FullName	EducationLevel	Occupation	AnnualIncome	Max_income_by_edu_occ
12810	CHASE STEWART	Bachelors	Clerical	30000	40000
12280	THERESA ALVAREZ	Bachelors	Clerical	30000	40000
11399	BRENDA MEHTA	Bachelors	Clerical	30000	40000
11398	COLIN NATH	Bachelors	Clerical	30000	40000
12284	CRISTINA BECK	Bachelors	Clerical	30000	40000
11395	BETH GUTIERREZ	Bachelors	Clerical	30000	40000
12815	DOMINIQUE MEHTA	Bachelors	Clerical	30000	40000
12821	CESAR MCDONALD	Bachelors	Clerical	40000	40000
12471	LACEY ZENG	Bachelors	Clerical	40000	40000
12472	JEFFERY ZHANG	Bachelors	Clerical	30000	40000
12799	GEORGE CHANDRA	Bachelors	Clerical	30000	40000
11549	CRYSTAL LIANG	Bachelors	Clerical	40000	40000
12459	STEFANIE RODRIG...	Bachelors	Clerical	30000	40000
11545	REGINALD DOMIN...	Bachelors	Clerical	30000	40000

EducationLevel	Occupation	AnnualIncome	Max_income_by_edu_occ
Bachelors	Management	90000	170000
Bachelors	Management	90000	170000
Bachelors	Management	130000	170000
Bachelors	Management	70000	170000
Bachelors	Management	60000	170000
Bachelors	Management	70000	170000
Bachelors	Management	130000	170000
Bachelors	Manual	10000	10000
Bachelors	Manual	10000	10000
Bachelors	Manual	10000	10000
Bachelors	Manual	10000	10000
Bachelors	Manual	10000	10000
Bachelors	Manual	10000	10000
Bachelors	Manual	10000	10000

CustomerKey	FullName	EducationLevel	Occupation	AnnualIncome	Max_income_by_edu_occ
11465	LOUIS LUO	Bachelors	Professional	90000	90000
11900	BYRON CARLSON	Bachelors	Professional	90000	90000
11812	ANDREW RODRIG...	Bachelors	Professional	40000	90000
11761	EDGAR MEHTA	Bachelors	Professional	70000	90000
11811	ABIGAIL BROOKS	Bachelors	Professional	40000	90000
12676	MICHELE MADAN	Bachelors	Professional	90000	90000
12739	EVAN YOUNG	Bachelors	Skilled Ma...	40000	80000
12510	ROBERTO SANZ	Bachelors	Skilled Ma...	40000	80000
12758	STEPHANIE RAMIREZ	Bachelors	Skilled Ma...	30000	80000
12762	JONATHAN SMITH	Bachelors	Skilled Ma...	30000	80000
12763	CHLOE RUSSELL	Bachelors	Skilled Ma...	30000	80000
12509	MARC FERRIER	Bachelors	Skilled Ma...	40000	80000
12508	NICHOLE ANAND	Bachelors	Skilled Ma...	40000	80000
12506	PRESTON SANCHEZ	Bachelors	Skilled Ma...	40000	80000

Row_Number Index Column [starts from 1 till the partition Count]

```
-- ROW_NUMBER()
SELECT
    CustomerKey,
    FullName,
    EducationLevel,
    Occupation,
    AnnualIncome,
    ROW_NUMBER() OVER(
        PARTITION BY EducationLevel , Occupation
        ORDER BY AnnualIncome DESC
    ) AS Row_Index
FROM Customers;
```

CustomerKey	FullName	EducationLevel	Occupation	AnnualIncome	Row_Index
11367	CALVIN NARA	Bachelors	Clerical	20000	83
12247	BRANDY RAMAN	Bachelors	Clerical	20000	84
12556	MANUEL KAPOOR	Bachelors	Clerical	10000	85
12226	FAITH WARD	Bachelors	Clerical	10000	86
12557	PHILLIP LOPEZ	Bachelors	Clerical	10000	87
12555	EDGAR MALHOTRA	Bachelors	Clerical	10000	88
11244	ALEXIS COLEMAN	Bachelors	Management	170000	1
12318	KRISTINA SCHMIDT	Bachelors	Management	170000	2
12123	WESLEY LIANG	Bachelors	Management	170000	3
12645	AUDREY RUIZ	Bachelors	Management	170000	4
11422	DUSTIN DENG	Bachelors	Management	170000	5
11180	APRIL ANAND	Bachelors	Management	160000	6
12061	BRYCE BROOKS	Bachelors	Management	160000	7
12706	LOUIS LIANG	Bachelors	Management	160000	8

Lead() & Lag()

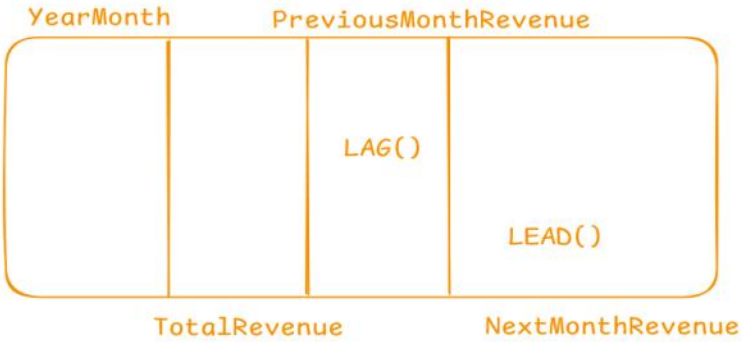
```
246     -- LEAD [Next Sale] and LAG [Past Sale]
247 •   USE Weekend_analysis;
248 •   DESC Sale;
249
```

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

Field	Type	Null	Key	Default	Extra
SaleID	int	YES		NULL	
Salesperson	text	YES		NULL	
SaleAmount	int	YES		NULL	
SaleDate	text	YES		NULL	

```
-- LEAD [Next Sale] and LAG [Past Sale]
USE Weekend_analysis;
DESC Sale;

SELECT
    *,
    LAG(SaleAmount) OVER(PARTITION BY Salesperson ORDER BY SaleDate) AS PreviousSale,
    LEAD(SaleAmount) OVER(PARTITION BY Salesperson ORDER BY SaleDate) AS NextSale
FROM Sale;
```

SaleID	Salesperson	SaleAmount	SaleDate	PreviousSale	NextSale
1	Alice	300	2023-01-01	NULL	200
3	Alice	200	2023-01-03	300	100
6	Alice	100	2023-01-06	200	450
8	Alice	450	2023-01-08	100	150
11	Alice	150	2023-01-11	450	350
14	Alice	350	2023-01-14	150	NULL
2	Bob	150	2023-01-02	NULL	300
5	Bob	300	2023-01-05	150	200
9	Bob	200	2023-01-09	300	250
12	Bob	250	2023-01-12	200	100
15	Bob	100	2023-01-15	250	NULL
4	Charlie	250	2023-01-04	NULL	350
7	Charlie	350	2023-01-07	250	400
10	Charlie	400	2023-01-10	350	300
13	Charlie	300	2023-01-13	400	NULL



Challenge : 1 - Find the PreviousMonthRevenue and NextMonthRevenue FROM Sales2015.

```
SELECT * FROM Sales2015;

DESC Sales2015;

SELECT
    DATE_FORMAT(s.OrderDate , '%Y-%m') AS YearMonth,
    ROUND(SUM(p.ProductPrice * s.OrderQuantity),0) AS TotalRevenue,

    LAG(ROUND(SUM(p.ProductPrice * s.OrderQuantity),0))
    OVER(ORDER BY DATE_FORMAT(s.OrderDate , '%Y-%m')) AS PreviousMonthRevenue,

    LEAD(ROUND(SUM(p.ProductPrice * s.OrderQuantity),0))
    OVER(ORDER BY DATE_FORMAT(s.OrderDate , '%Y-%m')) AS NextMonthRevenue

FROM Sales2015 s
JOIN Products p
ON p.ProductKey = s.ProductKey
GROUP BY 1;
```

YearMonth	TotalRevenue	PreviousMonthRevenue	NextMonthRevenue
2015-01	585313	NULL	532226
2015-02	532226	585313	643436
2015-03	643436	532226	653364
2015-04	653364	643436	659326
2015-05	659326	653364	669989
2015-06	669989	659326	486115
2015-07	486115	669989	536453
2015-08	536453	486115	344063
2015-09	344063	536453	404277
2015-10	404277	344063	326611
2015-11	326611	404277	563762
2015-12	563762	326611	NULL

NTILE() → Bucket(N)

```
SELECT ProductName, ProductPrice FROM Products
ORDER BY 2; -- 295 / 4 -> 73/74
```

ProductName	ProductPrice
Patch Kit/8 Patches	2.29
Road Tire Tube	3.99
Water Bottle - 30 oz.	4.99
Mountain Tire Tube	4.99
Touring Tire Tube	4.99
Bike Wash - Dissolver	7.95
AWC Logo Cap	8.6442
Road Bottle Cage	8.99
Racing Socks, M	8.99
Racing Socks, L	8.99
Mountain Bike Socks, M	9.5
Mountain Bike Socks, L	9.5
Mountain Bottle Cage	9.99
Taillights - Battery-P...	13.99
Minipump	19.99
Chain	20.24
LL Road Tire	21.49

```
SELECT
    ProductName,
    ProductPrice,
    NTILE(4) OVER(ORDER BY ProductPrice) As Price_Quartile
FROM Products;
```

ProductName	ProductPrice	Price_Quartile
Patch Kit/8 Patches	2.29	1
Road Tire Tube	3.99	1
Water Bottle - 30 oz.	4.99	1
Mountain Tire Tube	4.99	1
Touring Tire Tube	4.99	1
Bike Wash - Dissolver	7.95	1
AWC Logo Cap	8.6442	1
Road Bottle Cage	8.99	1
Racing Socks, M	8.99	1
Racing Socks, L	8.99	1
Mountain Bike Socks, M	9.5	1
Mountain Bike Socks, L	9.5	1
Mountain Bottle Cage	9.99	1
Taillights - Battery-P...	13.99	1
Minipump	19.99	1
Chain	20.24	1
LL Road Tire	21.49	1

ProductName	ProductPrice	Price_Quartile
Women's Mountain S...	69.99	2
Women's Mountain S...	69.99	2
Women's Mountain S...	69.99	2
Women's Tights, S	74.99	2
Women's Tights, M	74.99	2
Women's Tights, L	74.99	2
HL Mountain Pedal	80.99	2
HL Road Pedal	80.99	2
Touring Pedal	80.99	2
LL Road Front Wheel	85.565	2
LL Mountain Rear W...	87.745	2
Men's Bib-Shorts, S	89.99	2
Men's Bib-Shorts, M	89.99	2
Men's Bib-Shorts, L	89.99	2
Front Derailleur	91.49	2
HL Touring Handlebars	91.57	2
ML Bottom Bracket	101.24	2

ProductName	ProductPrice	Price_Quartile
Mountain-500 Silver,...	564.99	3
Mountain-500 Silver,...	564.99	3
ML Road Frame - Re...	594.83	3
ML Road Frame - Re...	594.83	3
ML Road Frame - Re...	594.83	3
ML Road Frame - Re...	594.83	3
ML Road Frame - Re...	594.83	3
Road-650 Red, 58	699.0982	3
Road-650 Red, 60	699.0982	3
Road-650 Red, 62	699.0982	3
Road-650 Red, 44	699.0982	3
Road-650 Red, 48	699.0982	3
Road-650 Red, 52	699.0982	3
Road-650 Black, 58	699.0982	3
Road-650 Black, 60	699.0982	3
Road-650 Black, 62	699.0982	3
Road-650 Black. 44	699.0982	3

ProductName	ProductPrice	Price_Quartile
Touring-1000 Blue, 54	2384.07	4
Touring-1000 Blue, 60	2384.07	4
Road-250 Red, 44	2443.35	4
Road-250 Red, 48	2443.35	4
Road-250 Red, 52	2443.35	4
Mountain-100 Black,...	3374.99	4
Mountain-100 Black,...	3374.99	4
Mountain-100 Black,...	3374.99	4
Mountain-100 Black,...	3374.99	4
Mountain-100 Silver,...	3399.99	4
Mountain-100 Silver,...	3399.99	4
Mountain-100 Silver,...	3399.99	4
Mountain-100 Silver,...	3399.99	4
Road-150 Red, 62	3578.27	4
Road-150 Red, 44	3578.27	4
Road-150 Red, 48	3578.27	4
Road-150 Red, 52	3578.27	4

First_Value()

Syntax:

```
SELECT
  window_function(...) OVER (
    PARTITION BY column_name
    ORDER BY column_name
    ROWS/RANGE ...
  ) AS result_column
FROM table_name;
```

UNBOUNDED

```
-- ====================== First_value() ======================
-- Find the Customers who purchased each products based on the earliest date....
```

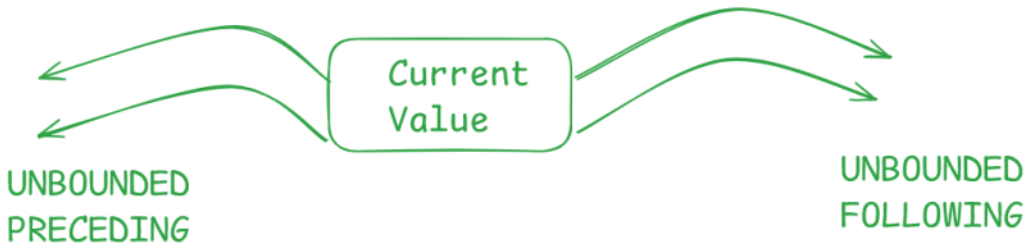
```
SELECT * FROM Sales2015;

SELECT
    s.ProductKey,
    c.FullName,
    s.OrderDate,
    FIRST_VALUE(c.FullName) OVER (PARTITION BY s.ProductKey  ORDER BY s.OrderDate)
    AS FirstPurchase
FROM Sales2015 s
JOIN Customers c
ON c.CustomerKey = s.CustomerKey;
```

ProductKey	FullName	OrderDate	FirstPurchase
310	JARED COOK	2015-01-23	JARED COOK
310	TIFFANY LI	2015-02-07	JARED COOK
310	TREVOR BRYANT	2015-02-07	JARED COOK
310	NATALIE ADAMS		D COOK
310	BRADLEY NARA		JARED COOK

NATALIE ADAMS

ProductKey	FullName	OrderDate	FirstPurchase
310	SIERRA PARKER	2015-06-30	JARED COOK
311	MEREDITH ALVAREZ	2015-01-10	MEREDITH ALVAREZ
311	NICHOLAS THOM...	2015-02-14	MEREDITH ALVAREZ
311	DONNA XIE	2015-02-18	MEREDITH ALVAREZ
311	STEVE WAGNER	2015-02-28	MEREDITH ALVAREZ

LAST_VALUE()




```
-- -- ====================== Last_value() ======================
-- Find the Last Region where each products were sold based on the Latest OrderDate....
```

```
SELECT
    s.ProductKey,
    t.Region,
    s.OrderDate,

    LAST_VALUE(t.Region) OVER (PARTITION BY s.ProductKey  ORDER BY s.OrderDate
        RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
    ) AS last_region_product_sold

FROM Sales2015 s
JOIN Territories t
ON t.SalesTerritoryKey = s.TerritoryKey;
```

ProductKey	Region	OrderDate	last_region_product_sold
311	Germany	2015-06-21	Germany
311	Northwest	2015-06-23	Germany
311	Germany	2015-06-24	Germany
311	Southwest	2015-06-25	Germany
311	Northwest	2015-06-25	Germany
311	Northwest	2015-06-27	Germany
311	Southwest	2015-06-27	Germany
311	Germany	2015-06-30	Germany
312	Southwest	2015-01-01	Australia
312	United Kin...	2015-01-02	Australia
312	Australia	2015-01-02	Australia
312	Australia	2015-01-03	Australia
312	France	2015-01-04	Australia
312	Northwest	2015-01-04	Australia
312	Canada	2015-01-06	Australia
312	Southwest	2015-01-06	Australia
312	Australia	2015-01-11	Australia
312	Southwest	2015-01-12	Australia

Nth_VALUE()

```
-- ====================== Nth_value() ======================
-- Find the 5th Customers who purchased each products based on the earliest date....
```

```
SELECT * FROM Sales2015;

SELECT
    s.ProductKey,
    c.FullName,
    s.OrderDate,
    NTH_VALUE(c.FullName , 5) OVER (PARTITION BY s.ProductKey  ORDER BY s.OrderDate)
    AS FifthPurchaser
FROM Sales2015 s
JOIN Customers c
ON c.CustomerKey = s.CustomerKey;
```

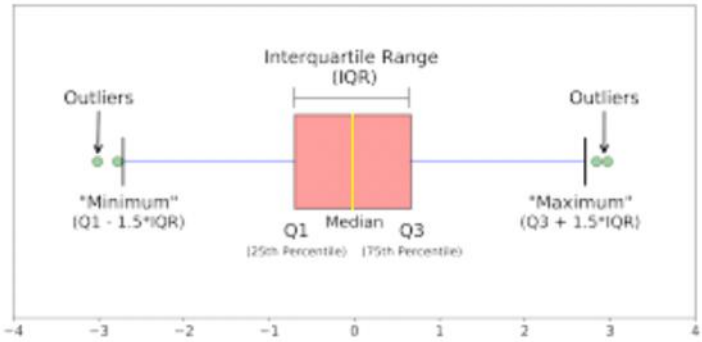
ProductKey	FullName	OrderDate	FifthPurchaser
310	JARED COOK	2015-01-23	NULL
310	TIFFANY LI	2015-02-07	NULL
310	TREVOR BRYANT	2015-02-07	NULL
310	NATALIE ADAMS	2015-02-10	NULL
310	BRADLEY NARA	2015-02-14	BRADLEY NARA
310	BRANDY SANCHEZ	2015-02-24	BRADLEY NARA
310	ROBERTO SANZ	2015-02-27	BRADLEY NARA
310	NINA YUAN	2015-03-02	BRADLEY NARA
310	DEVIN NELSON	2015-03-05	BRADLEY NARA
310	JULIA COLEMAN	2015-03-05	BRADLEY NARA
310	VICTORIA STEW...	2015-03-07	BRADLEY NARA
310	JADE BAILEY	2015-03-12	BRADLEY NARA

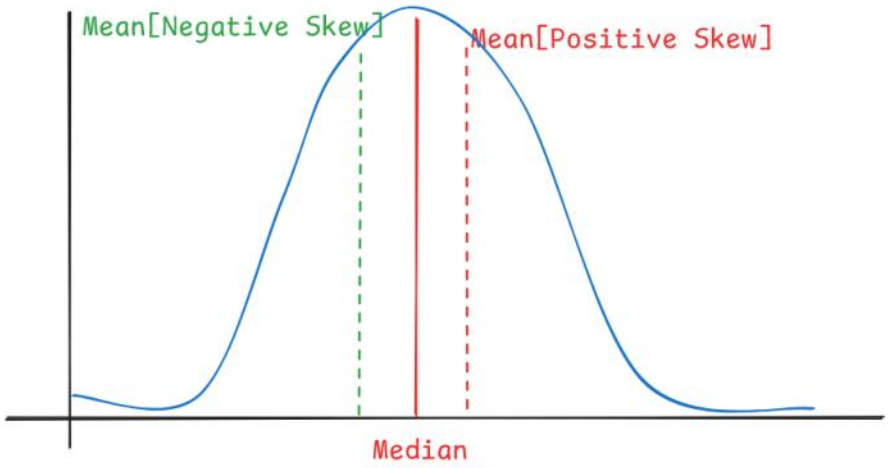
```
-- ====================== Nth_value() ======================
-- Find the 5th Customers who purchased each products based on the earliest date....

SELECT * FROM Sales2015;

SELECT
    s.ProductKey,
    c.FullName,
    s.OrderDate,
    NTH_VALUE(c.FullName , 5) OVER (PARTITION BY s.ProductKey  ORDER BY s.OrderDate
        RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
    ) AS FifthPurchaser
FROM Sales2015 s
JOIN Customers c
ON c.CustomerKey = s.CustomerKey;
```

ProductKey	FullName	OrderDate	FifthPurchaser
310	JARED COOK	2015-01-23	BRADLEY NARA
310	TIFFANY LI	2015-02-07	BRADLEY NARA
310	TREVOR BRYANT	2015-02-07	BRADLEY NARA
310	NATALIE ADAMS	2015-02-10	BRADLEY NARA
310	BRADLEY NARA	2015-02-14	BRADLEY NARA
310	BRANDY SANCHEZ	2015-02-24	BRADLEY NARA
310	ROBERTO SANZ	2015-02-27	BRADLEY NARA
310	NINA YUAN	2015-03-02	BRADLEY NARA
310	DEVIN NELSON	2015-03-05	BRADLEY NARA
310	JULIA COLEMAN	2015-03-05	BRADLEY NARA
310	VICTORIA STEW...	2015-03-07	BRADLEY NARA
310	JADE BAILEY	2015-03-12	BRADLEY NARA
310	AIDAN ROSS	2015-03-15	BRADLEY NARA
310	JONATHAN PHILL...	2015-03-19	BRADLEY NARA

ProductKey	FullName	OrderDate	FifthPurchaser
311	JORDYN BRYANT	2015-06-23	KYLE FOSTER
311	JAMIE ZENG	2015-06-25	KYLE FOSTER
311	THOMAS ZHANG	2015-06-25	KYLE FOSTER
311	BENJAMIN MOORE	2015-06-27	KYLE FOSTER
311	BRYANT SRINI	2015-06-27	KYLE FOSTER
312	AMANDA PEREZ	2015-01-04	JONATHAN HILL
312	VERONICA SUBRAM	2015-01-12	JONATHAN HILL
312	KRISTI PEREZ	2015-02-11	JONATHAN HILL
312	JASMINE TORRES	2015-02-15	JONATHAN HILL
312	JONATHAN HILL	2015-02-20	JONATHAN HILL
312	JACQUELINE PO...	2015-02-24	JONATHAN HILL
312	LUIS WANG	2015-02-28	JONATHAN HILL
312	CHRISTIAN THO...	2015-03-03	JONATHAN HILL
312	HALEY RICHARDS...	2015-03-04	JONATHAN HILL

Challenge - 1






```
-- Find the Products which are either below the lower bound[Low Outlier] and
   above the upper bound[High Outlier]

-- Outlier Detection [High Outliers] [Positive Skewed]
```

```sql
WITH Product_Stats AS(
    SELECT
        ProductPrice,
        NTILE(4) OVER (ORDER BY ProductPrice) AS Price_quartile
    FROM Products
),
Quartiles AS (
    SELECT
        MAX(CASE WHEN Price_quartile = 1 THEN ProductPrice END) AS Q1,
        MAX(CASE WHEN Price_quartile = 3 THEN ProductPrice END) AS Q3
    FROM Product_Stats
),
iqr_bounds AS(
    SELECT
    Q1,
    Q3,
    Q3 - Q1 AS IQR,
    Q1 - (1.5 * (Q3 - Q1)) AS Lower_Bound,
    Q3 + (1.5 * (Q3 - Q1)) AS Upper_Bound
    FROM Quartiles
)
SELECT
    p.ProductKey,
    p.ProductName,
    p.ProductPrice
FROM Products p
JOIN iqr_bounds iqr
ON p.ProductPrice < iqr.Lower_Bound OR p.ProductPrice > iqr.Upper_Bound
ORDER BY ProductPrice; -- 18 row(s) returned
```

low_outliers

high_outliers

ProductKey	ProductName	ProductPrice
368	Road-250 Red, 44	2443.35
369	Road-250 Red, 48	2443.35
370	Road-250 Red, 52	2443.35
348	Mountain-100 Black, 38	3374.99
349	Mountain-100 Black, 42	3374.99
350	Mountain-100 Black, 44	3374.99
351	Mountain-100 Black, 48	3374.99
344	Mountain-100 Silver, 38	3399.99
345	Mountain-100 Silver, 42	3399.99
346	Mountain-100 Silver, 44	3399.99
347	Mountain-100 Silver, 48	3399.99
310	Road-150 Red, 62	3578.27
311	Road-150 Red, 44	3578.27
312	Road-150 Red, 48	3578.27
313	Road-150 Red, 52	3578.27
314	Road-150 Red, 56	3578.27
607	Tesla ModelX	9999.99
608	Tesla ModelY	9999.99