Group By and Having in MS-SQL Server

Soyeb Ghachi

Technical Trainer | Mentor |SME

Overview

GROUP BY

-Group By using Aggregate Function

HAVING

-Group By with Having Clause

GROUP BY

GROUP BY

Introduction to SQL Server GROUP BY clause.

The GROUP BY clause allows you to arrange the rows of a query in groups. The groups are determined by the columns that you specify in the GROUP BY clause.

The following illustrates the GROUP BY clause syntax:

```
SELECT
   select_list
FROM
   table_name
GROUP BY
   column_name1,
   column_name2 ,...;
```

A) Using GROUP BY clause with the aggregate functions.

```
SELECT
    customer_id,
    YEAR (order_date) order_year,
    COUNT (order_id) order_placed
FROM
    orders
WHERE
    customer_id IN (1, 2)
GROUP BY
    customer_id,
    YEAR (order_date)
ORDER BY
    customer_id;
```

customer_id	order_year	order_placed
1	2016	1
1	2018	2
2	2017	2
2	2018	1

```
SELECT
    city,
    COUNT (customer_id) customer_count
FROM
    customers
GROUP BY
    city
ORDER BY
    city;
```

city	customer_count
Albany	3
Amarillo	5
Amityville	9
Amsterdam	5
Anaheim	11
Apple Valley	11
Astoria	12
Atwater	5
Auburn	4
Bakersfield	5

```
SELECT
    brand_name,
    MIN (list_price) min_price,
    MAX (list_price) max_price
FROM
   products p
INNER JOIN brands b ON b.brand_id = p.brand_id
WHERE
    model_year = 2018
GROUP BY
    brand_name
ORDER BY
    brand_name;
```

brand_name	min_price	max_price
Electra	269.99	2999.99
Heller	2599.00	2599.00
Strider	89.99	289.99
Surly	469.99	2499.99
Trek	159.99	11999.99

Having clause

Introduction to SQL Server HAVING clause:

The HAVING clause is often used with the GROUP BY clause to filter groups based on a specified list of conditions. The following illustrates the HAVING clause syntax:

```
SELECT
   select_list
FROM
   table_name
GROUP BY
   group_list
HAVING
   conditions;
```

A) SQL Server HAVING clause with the SUM() function.

```
SELECT
    order_id,
    SUM (
        quantity * list_price * (1 - discount)
    ) net_value
FROM
    order_items
GROUP BY
    order_id
HAVING
    SUM (
        quantity * list_price * (1 - discount)
    ) > 20000
ORDER BY
    net_value;
```

order_id	net_value
973	20177.7457
1334	20509.4254
1348	20648.9537
930	24607.0261
1364	24890.6244
1482	25365.4344
1506	25574.9555
937	27050.7182
1541	29147.0264

B) SQL Server HAVING clause with MAX and MIN functions.

```
SELECT
    category_id,
    MAX (list_price) max_list_price,
    MIN (list_price) min_list_price
FROM
    products
GROUP BY
    category_id
HAVING
    MAX (list_price) > 4000 OR MIN (list_price) < 500;
```

category_id	max_list_price	min_list_price
1	489.99	89.99
2	2599.99	416.99
3	2999.99	250.99
5	4999.99	1559.99
6	5299.99	379.99
7	11999.99	749.99

References

https://learn.microsoft.com/en-us/sql/relational-databases/performance/joins

https://www.w3schools.com/sql/sql_where.asp

https://www.sqlservertutorial.net/sql-server-basics/sql-server-where/

https://www.javatpoint.com/sql-server-joins

https://www.javatpoint.com/sql-server-joins

https://www.guru99.com/sql-server-joins.html

https://learn.microsoft.com/en-us/sql/relational-databases/performance/subqueries