

GROUPING SETS

簡單來說：

`GROUPING SETS` 是 SQL 中的一種進階語法，用於在單個查詢中定義多個分組集合，生成靈活的彙總結果。它就像讓你一次執行多個 `GROUP BY` 查詢，並將結果合併，省時又高效。特別適用於需要多維度報表或分析的場景，例如資料倉儲或商業智慧。

詳細解釋

定義

`GROUPING SETS` 是 SQL 標準的一部分，允許在 `GROUP BY` 子句中指定多個分組組合，產生不同層次的彙總數據。它是 `GROUP BY` 的擴展，支援更複雜的數據分析需求，常見於關係型資料庫如 SQL Server、Oracle、PostgreSQL 等。

- 核心功能：**在一個查詢中生成多個分組層級的結果，而不需要多次執行 `GROUP BY`。
- 相關技術：**與 `ROLLUP` 和 `CUBE` 同屬 SQL 彙總擴展，但 `GROUPING SETS` 更靈活，可自訂分組。

語法

```
SELECT column1, column2, ..., aggregate_function(column)
FROM table_name
GROUP BY GROUPING SETS (
    (column1, column2), -- 分組 1
    (column1),          -- 分組 2
    (),                 -- 總計
    ...
);
```

- column1, column2：**用於分組的欄位。
- aggregate_function：**彙總函數，如 `SUM`、`COUNT`、`AVG`。
- GROUPING SETS：**定義多個分組組合，每個括號表示一種分組方式。
- ()：**空分組，表示對所有數據進行總計。

工作原理

- 每個分組集合獨立計算彙總結果。
- 結果集將所有分組的結果合併成一個輸出。
- 非分組欄位的值在對應分組外會顯示為 `NULL`（可用 `GROUPING` 函數辨識）。

實際例子

假設有一個銷售資料表 `Sales`：

Region	Product	Sales_Amount
North	Laptop	1000

Region	Product	Sales_Amount
North	Phone	500
South	Laptop	800
South	Phone	600

想分析：

1. 按 **Region** 和 **Product** 分組的總銷售額。
2. 按 **Region** 分組的總銷售額。
3. 按 **Product** 分組的總銷售額。
4. 整體總銷售額。

使用 **GROUPING SETS** 的查詢：

```
SELECT Region, Product, SUM(Sales_Amount) AS Total_Sales
FROM Sales
GROUP BY GROUPING SETS (
    (Region, Product), -- 按 Region 和 Product 分組
    (Region),          -- 按 Region 分組
    (Product),         -- 按 Product 分組
    ()                 -- 整體總計
);
```

結果：

Region	Product	Total_Sales
North	Laptop	1000
North	Phone	500
South	Laptop	800
South	Phone	600
North	NULL	1500
South	NULL	1400
NULL	Laptop	1800
NULL	Phone	1100
NULL	NULL	2900

解釋：

- **(Region, Product)**：細分到每個地區和產品的銷售額。
- **(Region)**：每個地區的總銷售額，**Product** 顯示為 **NULL**。

- `(Product)`：每種產品的總銷售額，`Region` 顯示為 `NULL`。
- `()`：所有數據的總銷售額，`Region` 和 `Product` 均為 `NULL`。

與 ROLLUP 和 CUBE 的比較

- **ROLLUP**：
 - 產生層次性彙總，從最細分組到總計。
 - 例子：`GROUP BY ROLLUP (Region, Product)` 產生 `(Region, Product)`、`(Region)` 和 `()`。
 - 限制：只能按層次遞減，不能自訂分組。
- **CUBE**：
 - 產生所有可能的分組組合。
 - 例子：`GROUP BY CUBE (Region, Product)` 產生 `(Region, Product)`、`(Region)`、`(Product)` 和 `()`。
 - 限制：可能產生過多組合，增加計算成本。
- **GROUPING SETS**：
 - 靈活指定所需分組組合，不受層次或全組合限制。
 - 例子：可以只選 `(Region, Product)` 和 `(Product)`，不包括其他。

進階功能：GROUPING 函數

- 作用：辨識某欄位是否因彙總而為 `NULL`。
- 語法：

```
GROUPING(column)
```

- 返回 1：該欄位因分組彙總為 `NULL`。
 - 返回 0：該欄位為實際值。
- 例子：

```
SELECT Region, Product, SUM(Sales_Amount) AS Total_Sales,  
       GROUPING(Region) AS Is_Region_Grouped,  
       GROUPING(Product) AS Is_Product_Grouped  
FROM Sales  
GROUP BY GROUPING SETS ((Region, Product), (Region), ());
```

- 當 `Region` 為 `NULL` 時，`GROUPING(Region)` 為 1。

應用場景

1. 商業報表

- 生成多維度銷售報表（如按地區、產品、時間的總計）。

2. 資料倉儲

- 在 SQL Server Analysis Services (SSAS) 或其他 OLAP 系統中，快速生成多層次彙總。

3. 數據分析

- 分析不同維度的趨勢（如客戶年齡層與購買行為）。

4. 儀表板

- 為 Power BI 或 Tableau 提供靈活的數據集。

優點

- **靈活性**：自訂分組組合，避免不必要的計算。
- **高效性**：單查詢取代多次 `GROUP BY`，減少資源消耗。
- **可讀性**：語法清晰，易於維護。

挑戰

- **學習曲線**：對 SQL 新手來說，理解 `GROUPING SETS` 和 `NULL` 值較複雜。
- **效能考量**：若分組組合過多，可能增加查詢時間。
- **支援性**：部分較舊的資料庫可能不支援（SQL Server 2008 及以上支援）。

結論

`GROUPING SETS` 是 SQL 中強大的工具，讓你能在一條查詢中靈活定義多個分組，生成多層次彙總結果，特別適合報表和分析需求。它就像一個「數據切片機」，幫你從不同角度快速檢視數據。如果想深入實際案例（例如在 SQL Server 中實現）或與其他技術（如 `ROLLUP`）比較，請告訴我，我可以再細講！