

SQL 的五十道練習

分組與聚合結果篩選

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

這個章節要學起來的 SQL 保留字

- GROUP BY
- HAVING

以 GROUP BY 分組

分組 **GROUP BY** 的功能可以視為 **DISTINCT** 與 **ORDER BY** 兩者同時作用

```
SELECT column_names  
FROM table_name  
GROUP BY column_names;
```

DISTINCT 與 ORDER BY 兩者同時作用

In [5]:

```
SELECT DISTINCT pos AS distinct_pos  
FROM players  
ORDER BY distinct_pos;
```

Out[5]:

| distinct_pos |
|--------------|
| C |
| C-F |
| F |
| F-C |
| F-G |
| G |
| G-F |

7 rows in set (0.00 sec)

使用分組 GROUP BY

In [6]:

```
SELECT pos AS distinct_pos  
FROM players  
GROUP BY pos;
```

Out[6]:

| distinct_pos |
|--------------|
| C |
| C-F |
| F |
| F-C |
| F-G |
| G |
| G-F |

7 rows in set (0.00 sec)

在「函數」的章節，我們介紹過一種「用來彙總資訊」的函數，稱為聚合函數（Aggregate functions）

單獨使用聚合函數的時候，是將一整欄變數的資訊彙總後輸出

```
In [7]: SELECT AVG(heightMeters) AS height_meters_avg  -- 所有球員的平均身高
        FROM players;
```

```
Out[7]:
```

| <u>height_meters_avg</u> |
|--------------------------|
| 1.989173553719 |

1 row in set (0.00 sec)

假如現在希望計算不同 `pos`（鋒衛位置）的球員平均身高，現在我們會怎麼做？

- 先知道有哪些鋒衛位置。
- 篩選不同鋒衛位置的球員，一一計算平均身高。

In [8]: `SELECT DISTINCT pos -- 先知道有哪些鋒衛位置`
`FROM players;`

Out[8]:

| pos |
|-----|
| F |
| C-F |
| G-F |
| G |
| F-G |
| C |
| F-C |

7 rows in set (0.00 sec)

篩選不同鋒衛位置的球員，一一計算平均身高

```
SELECT AVG(heightMeters) AS forward_avg_height_meters
  FROM players
 WHERE pos = 'F';
SELECT AVG(heightMeters) AS center_forward_avg_height_meters
  FROM players
 WHERE pos = 'C-F';
SELECT AVG(heightMeters) AS guard_forward_avg_height_meters
  FROM players
 WHERE pos = 'G-F';
-- 繼續篩選剩餘的四個鋒衛位置...
```

這不是一個聰明的辦法，萬一用來當作篩選條件欄位有很多的獨一值該怎麼辦呢？

```
In [9]: SELECT COUNT(DISTINCT country) AS number_of_dist_countries -- 球員的國籍  
        FROM players;
```

```
Out[9]: 

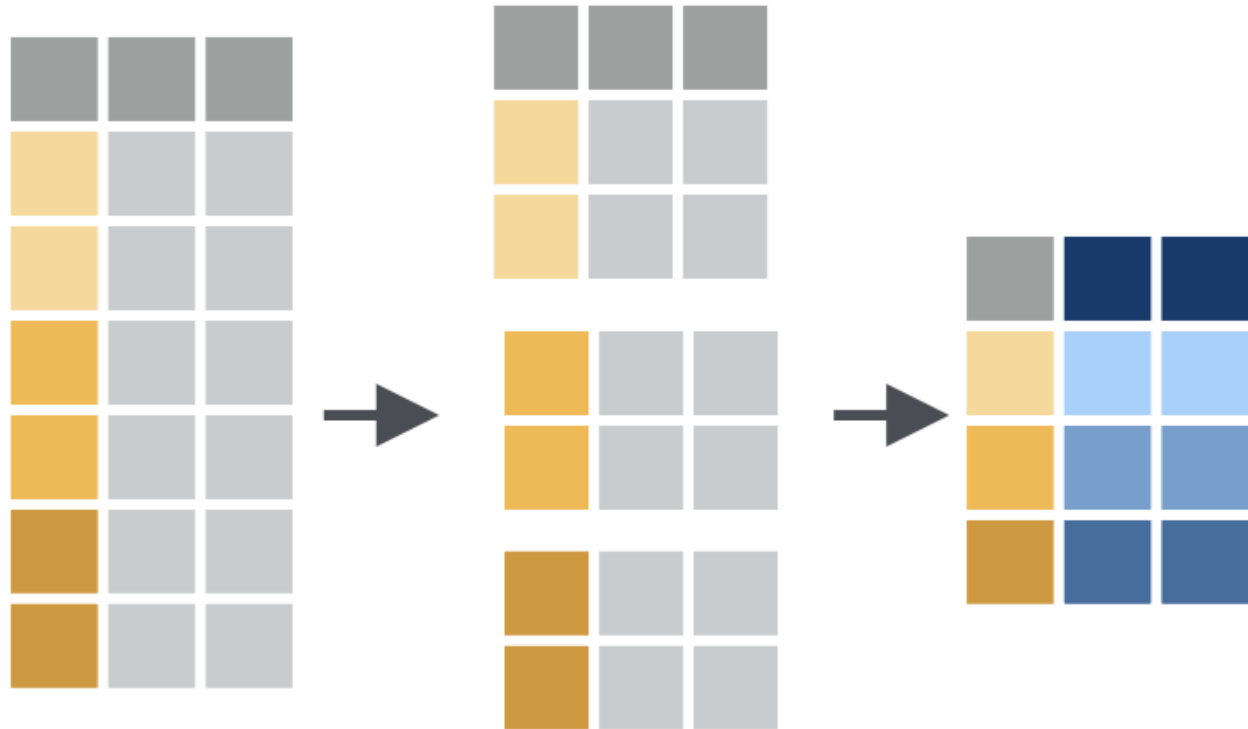
| number_of_dist_countries |
|--------------------------|
| 43                       |


```

1 row in set (0.00 sec)

結合聚合函數與 **GROUP BY** 可以便捷地完成分組聚合

```
SELECT AGGREGATE_FUNCTION(column_names) AS alias  
FROM table_name  
GROUP BY column_names;
```



In [10]:

```
SELECT pos,  
       ROUND(AVG(heightMeters), 2) AS avg_height_meters  
FROM players  
GROUP BY pos; -- 計算不同 pos ( 鋒衛位置 ) 的球員平均身高
```

Out[10]:

| pos | avg_height_meters |
|-----|-------------------|
| C | 2.12 |
| C-F | 2.1 |
| F | 2.02 |
| F-C | 2.08 |
| F-G | 2.0 |
| G | 1.91 |
| G-F | 1.98 |

7 rows in set (0.00 sec)

In [11]:

```
SELECT country,  
       ROUND(AVG(heightMeters), 2) AS avg_height_meters  
FROM players  
GROUP BY country -- 計算不同國籍球員平均身高  
LIMIT 5;
```

Out[11]:

| country | avg_height_meters |
|-----------|-------------------|
| Angola | 2.06 |
| Argentina | 1.78 |
| Australia | 1.98 |
| Austria | 2.16 |
| Bahamas | 2.02 |

5 rows in set (0.00 sec)

GROUP BY 可以加入不只一個變數

In [12]:

```
SELECT pos,  
       country,  
       ROUND(AVG(heightMeters), 2) AS avg_height_meters  
FROM players  
GROUP BY pos,  
         country  
LIMIT 5;
```

Out[12]:

| pos | country | avg_height_meters |
|-----|------------------------|-------------------|
| C | Australia | 2.08 |
| C | Austria | 2.16 |
| C | Bahamas | 2.11 |
| C | Bosnia and Herzegovina | 2.11 |
| C | Canada | 2.06 |

5 rows in set (0.00 sec)

以 **HAVING** 篩選分組聚合結果

用來篩選資料的兩種方式：

1. 作用在「觀測值」的 `WHERE` 。
2. 作用在「分組聚合結果」的 `HAVING` 。

作用在「分組聚合結果」的 **HAVING**

```
SELECT AGGREGATE_FUNCTION(column_names) AS alias  
FROM table_name  
GROUP BY column_names  
HAVING conditions;
```

In [13]:

```
SELECT pos,  
       ROUND(AVG(heightMeters), 2) AS avg_height_meters  
FROM players  
GROUP BY pos  
HAVING AVG(heightMeters) >= 2;  -- 篩選平均身高大於 2 公尺的 pos ( 鋒衛位置 )
```

Out[13]:

| pos | avg_height_meters |
|-----|-------------------|
| C | 2.12 |
| C-F | 2.1 |
| F | 2.02 |
| F-C | 2.08 |
| F-G | 2.0 |

5 rows in set (0.00 sec)

In [14]:

```
SELECT pos,  
       ROUND(AVG(heightMeters), 2) AS avg_height_meters  
FROM players  
WHERE heightMeters >= 2 -- 篩選身高大於 2 公尺的球員  
GROUP BY pos;
```

Out[14]:

| pos | avg_height_meters |
|-----|-------------------|
| C | 2.12 |
| C-F | 2.1 |
| F | 2.04 |
| F-C | 2.08 |
| F-G | 2.03 |
| G | 2.02 |
| G-F | 2.03 |

7 rows in set (0.00 sec)

重點統整

- 分組 `GROUP BY` 可以視為 `DISTINCT` 與 `ORDER BY` 兩者同時作用。
- 結合聚合函數與 `GROUP BY` 可以便捷地完成分組聚合。
- 以 `HAVING` 篩選分組聚合結果。

/*截至目前學起來的 SQL 有哪些？

SQL 寫作順序必須遵從標準 SQL 的規定。*/

```
SELECT column_names      -- 選擇哪些欄位
  FROM table_name        -- 從哪個資料庫的資料表
 WHERE conditions        -- 篩選哪些觀測值
 GROUP BY column_names   -- 指定依照哪個變數分組
HAVING conditions       -- 篩選哪些分組聚合的結果
 ORDER BY column_names   -- 指定依照哪個變數排序
LIMIT m;               -- 查詢結果顯示前 m 列就好
```

