SQL 的五十道練習

> *子查詢*

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

什麼是子查詢

在「函數」的章節中，我們看過函數包含著其他函數的使用方法

前一個函數的輸出，成為下一個函數的輸入。

In [5]:
```
SELECT ROUND(AVG(rating)) AS avg_rating
  FROM movies;
```

Out[5]:

avg_rating
8.0

1 row in set (0.00 sec)

如果一段 SQL 中包含著另外一段的 SQL，這樣的 SQL 結構就被稱為子查詢（Subquery）

常見的子查詢應用情境

在什麼樣的情境下我們會想使用子查詢呢？

1. 查詢的篩選條件必須要先做一個查詢才能得知。
2. 查詢的計算內容必須要先做一個查詢才能得知。
3. 查詢所需要的資料來自不同資料表。

情境一：查詢的篩選條件必須要先做一個查詢才能得知

我們想知道 `imdb` 資料庫的 `movies` 資料表中片長最短的電影是哪一部。

```
SELECT *
  FROM movies
 WHERE runtime = MIN(runtime); -- 這個查詢會得到錯誤，因為聚合函數不能夠寫在 WHERE 後
```

我們要分兩次查詢來完成

- 先查詢「最短」的片長是幾分鐘。
- 再依據前一個查詢結果作為篩選條件。

In [6]:

```sql
SELECT MIN(runtime) AS minimum_runtime  -- 先查詢「最短」的片長是幾分鐘。
  FROM movies;
```

Out[6]:

minimum_runtime
45

1 row in set (0.00 sec)

In [7]:

```sql
SELECT *
  FROM movies
 WHERE runtime = 45;  -- 再依據前一個查詢結果作為篩選條件。
```

Out[7]:

id	title	release_year	rating	director	runtime
195	Sherlock Jr.	1924	8.2	Buster Keaton	45

1 row in set (0.00 sec)

面對情境一，我們可以應用子查詢將一段 SQL 查詢結果作為條件

```
SELECT column_names
  FROM table_name
 WHERE conditions (Another SELECT statement);
```

In [8]:

```sql
SELECT MIN(runtime) AS minimum_runtime  -- 先查詢「最短」的片長是幾分鐘。
  FROM movies;

SELECT *
  FROM movies
 WHERE runtime = 45;  -- 再依據前一個查詢結果作為篩選條件。
```

Out[8]:

minimum_runtime
45

1 row in set (0.00 sec)

情境二：查詢的計算內容必須要先做一個查詢才能得知

我們想知道 `imdb` 資料庫的 `movies` 資料表中，在千禧年之後上映的電影佔比為多少？

我們要分兩次查詢來完成

- 先查詢在千禧年之後上映的電影有幾部。
- 再依據前一個查詢結果作為計算內容。

In [9]:

```sql
SELECT COUNT(*) AS number_of_movies
  FROM movies
 WHERE release_year >= 2000;  -- 先查詢在千禧年之後上映的電影有幾部。
```

Out[9]:

number_of_movies
101

1 row in set (0.00 sec)

In [10]:

```sql
SELECT 101 / CAST(COUNT(*) AS REAL) AS millennium_percentage  -- 再依據前一個查詢結果作為計算內容。
  FROM movies;
```

Out[10]:

millennium_percentage
0.404

1 row in set (0.00 sec)

面對情境二，我們可以應用子查詢將一段 SQL 查詢結果作為計算內容

```
SELECT (Another SELECT statement) AS alias
  FROM table_name;
```

In [11]:

```sql
SELECT COUNT(*) AS number_of_movies
  FROM movies
 WHERE release_year >= 2000;  -- 先查詢在千禧年之後上映的電影有幾部。

SELECT 101 / CAST(COUNT(*) AS REAL) AS millennium_percentage  -- 再依據前一個查詢結果作為計算內容。
  FROM movies;
```

Out[11]:

number_of_movies
101

1 row in set (0.00 sec)

情境三：查詢所需要的資料來自不同的資料表

我們想知道在 `imdb` 資料庫中，哪幾部電影 Tom Hanks 有演出？

我們要分三次查詢來完成

- 先從 `actors` 資料表查詢 Tom Hanks 的演員編號是多少。
- 再依據前一個查詢結果去 `casting` 資料表查詢。
- 再依據前一個查詢結果去 `movies` 資料表查詢。

In [12]:

```sql
SELECT id
  FROM actors
 WHERE name = 'Tom Hanks';  -- 先從 actors 資料表查詢 Tom Hanks 的演員編號是多少。
```

Out[12]:

id
2865

1 row in set (0.00 sec)

In [13]:

```sql
SELECT movie_id
  FROM casting
 WHERE actor_id = 2865;  -- 再依據前一個查詢結果去 casting 資料表查詢。
```

Out[13]:

movie_id
12
26
27
82
112
189

6 rows in set (0.00 sec)

In [14]:

```
SELECT *
  FROM movies
 WHERE id IN (12, 26, 27, 82, 112, 189);  -- 再依據前一個查詢結果去 movies 資料表查詢。
```

Out[14]:

id	title	release_year	rating	director	runtime
12	Forrest Gump	1994	8.8	Robert Zemeckis	142
26	Saving Private Ryan	1998	8.6	Steven Spielberg	169
27	The Green Mile	1999	8.6	Frank Darabont	189
82	Toy Story	1995	8.3	John Lasseter	81
112	Toy Story 3	2010	8.2	Lee Unkrich	103
189	Catch Me If You Can	2002	8.1	Steven Spielberg	141

6 rows in set (0.00 sec)

面對情境三，我們可以應用子查詢將一段 SQL 查詢結果作為條件

```
SELECT column_names
  FROM table_name
 WHERE conditions (Another SELECT statement);
```

In [15]:

```sql
SELECT id
  FROM actors
 WHERE name = 'Tom Hanks';  -- 先從 actors 資料表查詢 Tom Hanks 的演員編號是多少。

SELECT movie_id
  FROM casting
 WHERE actor_id = 2865;  -- 再依據前一個查詢結果去 casting 資料表查詢。

SELECT *
  FROM movies
 WHERE id IN (12, 26, 27, 82, 112, 189);  -- 再依據前一個查詢結果去 movies 資料表查詢。
```

Out[15]:

id
2865

1 row in set (0.00 sec)

也可以應用子查詢將一段 SQL 查詢結果作為資料表

```sql
SELECT column_names
  FROM (SELECT column_names FROM table_name) AS alias;
```

In [18]:

```
SELECT *
  FROM (SELECT release_year,
               MAX(rating) AS max_rating
          FROM movies
         GROUP BY release_year) AS max_rating_each_year
 ORDER BY release_year DESC
 LIMIT 10;
```

Out[18]:

release_year	max_rating
2021	8.3
2020	8.5
2019	8.6
2018	8.5
2017	8.4
2016	8.4
2015	8.2
2014	8.6
2013	8.4
2012	8.4

10 rows in set (0.00 sec)

重點統整

- 在一段 SQL 中包含著另外一段 SQL 的結構稱為子查詢（Subquery）。
- 子查詢常見的應用情境有：
    - 查詢的篩選條件必須要先做一個查詢才能得知。
    - 查詢的計算內容必須要先做一個查詢才能得知。
    - 查詢所需要的資料來自不同資料表。

重點統整（續）

- 面對不同情境的子查詢型態有：
    - 應用子查詢將一段 SQL 查詢結果作為**條件**。
    - 應用子查詢將一段 SQL 查詢結果作為**計算內容**。
    - 應用子查詢將一段 SQL 查詢結果作為**資料表**。

```sql
/*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
```