SQL 的五十道練習

> *函數*

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

什麼是函數

Function，中文翻譯為函數或者函式，在資料分析和程式語言中都扮演舉足輕重的角色

函數是預先被定義好的運算處理邏輯，透過它的作用，能夠將「輸入」對應為「輸出」，進而完成計算數值與操作文字等任務。

函數的運作有五個組成：

1. 函數的名稱。
2. 輸入。
3. 參數。
4. 運算處理邏輯。
5. 輸出。

以買珍珠奶茶為例

Source: Google Search

以一個常用的文字操作函數 `SUBSTR` 為例

In [5]:

```
SELECT 'Tony Stark' AS full_name,
       SUBSTR('Tony Stark', 1, 4) AS first_name,
       SUBSTR('Tony Stark', 6, 5) AS last_name;
```

Out[5]:

full_name	first_name	last_name
Tony Stark	Tony	Stark

1 row in set (0.00 sec)

函數可依照功能分為兩大類：

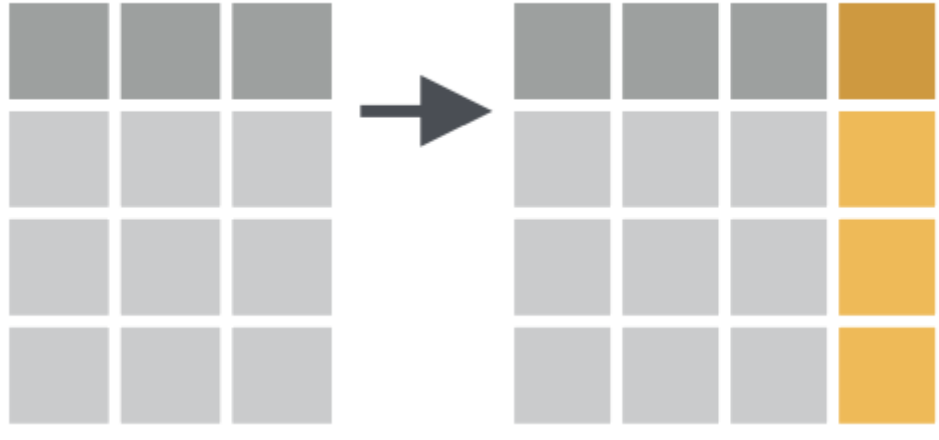
- 通用函數（Universal functions）
    - 轉換資料類型。
    - 計算數值。
    - 操作文字。
    - 操作日期時間。
- 聚合函數（Aggregate functions）

通用函數與聚合函數的不同在於其所作用的維度

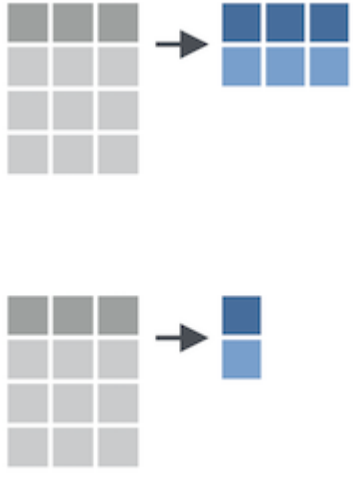
- 通用函數作用在「水平」方向。
- 聚合函數作用在「垂直」方向。

作用在「水平」方向的通用函數

效果類似「衍生計算欄位」，差別在於一個是以函數輸出衍生計算欄位，一個則是以運算符生成衍生計算欄位。



作用在「垂直」方向的聚合函數



通用函數的特徵：每個觀測值對應一個輸出

In [6]:
```
SELECT heightMeters,
       ROUND(heightMeters, 1) AS rounded_height
  FROM players
 LIMIT 5;
```

Out[6]:

heightMeters	rounded_height
2.06	2.1
2.01	2.0
2.03	2.0
2.08	2.1
1.98	2.0

5 rows in set (0.00 sec)

聚合函數的特徵：整欄變數對應一個輸出

In [7]:
```
SELECT AVG(heightMeters) AS avg_height
  FROM players;
```

Out[7]:

avg_height
1.989173553719

1 row in set (0.00 sec)

通用函數

常見的通用函數又可以細分為四種類型

- 轉換資料類型。
- 計算數值。
- 操作文字。
- 操作日期時間。

通用函數：轉換資料類型

使用 `CAST` 函數可以將查詢結果的資料類型轉換為指定資料類型

```
CAST(data AS data_type)
```

面對兩個整數相除所衍生的欄位依然會以整數類型存在的情況

舉例來說，計算球員的生涯場均得分要以 `points` 、 `gamesPlayed` 相除。

In [8]:
```
SELECT points / gamesPlayed AS points_per_game -- 可以應用 CAST 函數在分子或分母
  FROM career_summaries
 LIMIT 5;
```

Out[8]:

points_per_game
27
23
7
16
11

5 rows in set (0.00 sec)

使用 `COALESCE` 函數可以將空值（或稱遺漏值）轉換為指定常數

```
COALESCE(NULL, replacement)
```

舉例來說，在 `covid19` 資料庫的 `lookup_table` 資料表中有許多國家沒有詳細到有州、省、縣、郡的資訊

In [9]:

```
SELECT * -- 可以應用 COALESCE 函數在 Province_State 或 Admin2
  FROM lookup_table
 LIMIT 5;
```

Out[9]:

UID	Combined_Key	iso2	iso3	Country_Region	Province_State	Admin2	Lat	Long_	Population
4	Afghanistan	AF	AFG	Afghanistan	NULL	NULL	33.93911	67.709953	38928341
8	Albania	AL	ALB	Albania	NULL	NULL	41.1533	20.1683	2877800
12	Algeria	DZ	DZA	Algeria	NULL	NULL	28.0339	1.6596	43851043
16	American Samoa, US	AS	ASM	US	American Samoa	NULL	-14.271	-170.132	55641
20	Andorra	AD	AND	Andorra	NULL	NULL	42.5063	1.5218	77265

5 rows in set (0.00 sec)

通用函數：計算數值

使用 `ROUND` 函數可以調整查詢結果的小數點位數

```
ROUND(REAL, n_digits)
```

In [10]:

```
SELECT CAST(points AS REAL) / gamesPlayed AS points_per_game -- 可以應用 ROUND 函數
  FROM career_summaries
 LIMIT 5;
```

Out[10]:

points_per_game
27.0160796324655
23.1900684931507
7.6025641025641
16.3336206896552
11.7212276214834

5 rows in set (0.00 sec)

通用函數：操作文字

使用 `LENGTH` 函數可以計算文字中有幾個字元，包含空格、標點符號

```
LENGTH(TEXT)
```

In [11]:

```
SELECT firstName AS length_of_first_name, -- 可以應用 LENGTH 函數
       lastName AS length_of_last_name    -- 可以應用 LENGTH 函數
  FROM players
 LIMIT 5;
```

Out[11]:

length_of_first_name	length_of_last_name
LeBron	James
Carmelo	Anthony
Udonis	Haslem
Dwight	Howard
Andre	Iguodala

5 rows in set (0.00 sec)

使用 SUBSTR 函數可利用位置將文字中的指定段落擷取出來

```
SUBSTR(TEXT, start, length)
```

In [12]:

```
SELECT city -- 可以應用 SUBSTR 函數
  FROM teams
 LIMIT 3;
```

Out[12]:

city
Atlanta
Boston
Cleveland

3 rows in set (0.00 sec)

使用 `LOWER` 與 `UPPER` 函數可以調整英文的大小寫

```
LOWER(TEXT)
UPPER(TEXT)
```

In [13]:

```
SELECT SUBSTR(city, 1, 3) AS upper_tricode, -- 可以應用 UPPER 函數
       SUBSTR(city, 1, 3) AS lower_tricode  -- 可以應用 LOWER 函數
  FROM teams
 LIMIT 3;
```

Out[13]:

upper_tricode	lower_tricode
Atl	Atl
Bos	Bos
Cle	Cle

3 rows in set (0.00 sec)

通用函數：操作日期時間

標準的日期、時間與日期時間格式

- 以 ISO8601 格式為標準
- 日期 `YYYY-MM-DD`
- 時間 `HH:MM:SS`
- 日期時間 `YYYY-MM-DD HH:MM:SS`

使用 `STRFTIME` 函數調整日期、時間或日期時間的顯示格式

```
STRFTIME(format, DATE/TIME/DATETIME)
```

常見的日期與日期時間格式參數

- `%d`：二位數的日（01-31）
- `%j`：一年中的第幾天（001-366）
- `%m`：二位數的月（01-12）
- `%w`：一星期中的第幾天（0-6）
- `%W`：一年中的第幾週（00-53）
- `%Y`：四位數的年（0000-9999）

In [14]:

```
SELECT Last_Update,
       STRFTIME('%d', Last_Update) AS day_part,
       STRFTIME('%j', Last_Update) AS year_day_format,
       STRFTIME('%m', Last_Update) AS month_part,
       STRFTIME('%w', Last_Update) AS weekday,
       STRFTIME('%W', Last_Update) AS nth_week,
       STRFTIME('%Y', Last_Update) AS year_part
  FROM daily_report
 LIMIT 1;
```

Out[14]:

Last_Update	day_part	year_day_format	month_part	weekday	nth_week	year_part
2021-04-01 04:27:05	01	091	04	4	13	2021

1 row in set (0.00 sec)

SQLite 通用函數與操作日期時間函數的官方文件

- Built-In Scalar SQL Functions
- Date And Time Functions

聚合函數

常見的聚合函數

- `AVG(column_name)`：計算變數的平均數
- `COUNT(column_name)`：計算變數的「非」遺漏值數
- `COUNT(*)`：計算資料表的觀測值數
- `MAX(column_name)`：計算變數的最大值
- `MIN(column_name)`：計算變數的最小值
- `SUM(column_name)`：計算變數的加總

In [15]:

```sql
SELECT AVG(Confirmed) AS avg_confirmed
  FROM daily_report;
```

Out[15]:

avg_confirmed
32359.3908565687

1 row in set (0.01 sec)

In [16]:

```
SELECT COUNT(Province_State) AS number_of_states
  FROM lookup_table;
```

Out[16]:

number_of_states
3981

1 row in set (0.01 sec)

In [17]:

```sql
SELECT COUNT(*) AS number_of_rows
  FROM lookup_table;
```

Out[17]:

number_of_rows
4175

1 row in set (0.00 sec)

In [18]:

```
SELECT MAX(Confirmed) AS max_confirmed
  FROM daily_report;
```

Out[18]:

max_confirmed
4611392

1 row in set (0.00 sec)

In [19]:

```
SELECT MIN(Confirmed) AS min_confirmed
  FROM daily_report;
```

Out[19]:

min_confirmed
0

1 row in set (0.00 sec)

SQLite 聚合函數的官方文件

Built-in Aggregate Functions

重點統整

- 函數是預先被定義好的運算處理邏輯，能夠將「輸入」對應為「輸出」。
- 函數依照功能區分有兩大類，兩者的差別在於作用的維度不同。
    - 通用函數作用在「水平」方向。
    - 聚合函數則作用在「垂直」方向。

重點統整（續）

- 通用函數（Universal functions）
    - 轉換資料類型。
    - 計算數值。
    - 操作文字。
    - 操作日期時間。
- 聚合函數（Aggregate functions）

```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
 LIMIT m;            -- 查詢結果顯示前 m 列就好
```