SQL 的五十道練習

> *篩選資料*

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

這個章節要學起來的 SQL 保留字

- `WHERE`
- `LIKE`
- `AND`
- `BETWEEN`
- `OR`
- `IN`
- `NOT`
- `IS NULL`

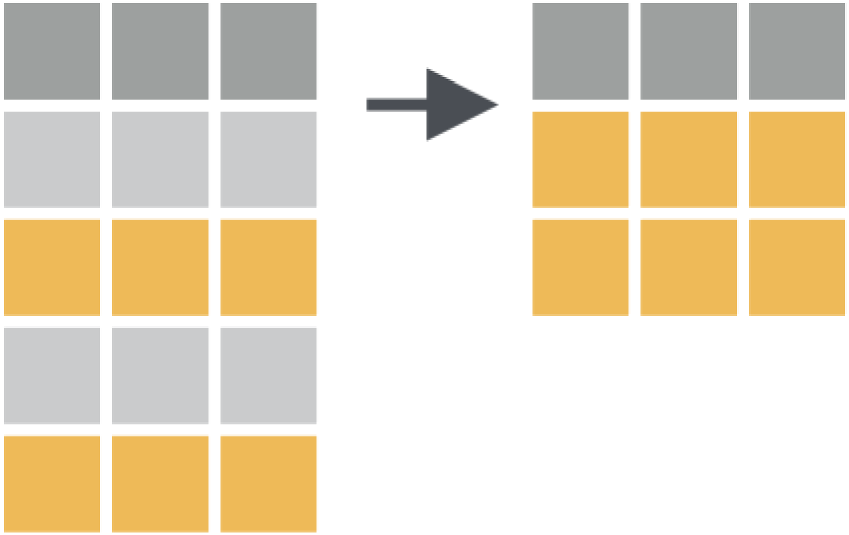
以 `WHERE` 篩選資料

在實際應用 SQL 時，常見需求是選出資料表中符合「特定條件」的觀測值，例如：

- 在 covid19 資料庫中找出台灣的資料。
- 在 imdb 資料庫中找出 1994 年上映的經典電影。
- 在 nba 資料庫中找出 Fantasy Game 想要選的球員。
- 在 twElection2020 資料庫中找出台北市的資料。

加入 `WHERE` 能夠以條件（Conditions）作為篩選觀測值的依據

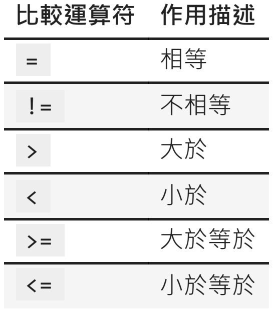
```
SELECT column_names
  FROM table_name
 WHERE conditions;
```



撰寫條件之前，我們要瞭解兩個觀念：

1. 比較運算符：能夠產生布林（Boolean）的運算符號。
2. 布林（Boolean）：用來表示比較結果的兩個值（真、假）。

基礎比較運算符



比較運算符	作用描述
`=`	相等
`!=`	不相等
`>`	大於
`<`	小於
`>=`	大於等於
`<=`	小於等於

比較結果為「真」的布林，SQLite 以 `1` 表示

In [5]:

```sql
SELECT 5566 = 5566 AS bool_true;
```

Out[5]:

bool_true
1

1 row in set (0.00 sec)

比較結果為「假」的布林，SQLite 以 `0` 表示

```
In [6]: SELECT 5566 != 5566 AS bool_false;
```

```
Out[6]: bool_false
                 0
```

1 row in set (0.00 sec)

針對資料表中的欄位使用比較運算符會在每列觀測值都對應生成一個布林

In [7]:

```
SELECT release_year = 1994 AS bool_values
  FROM movies
 LIMIT 10;
```

Out[7]:

bool_values
1
0
0
0
0
0
0
1
0
0

10 rows in set (0.00 sec)

若是在 `WHERE` 後應用比較運算符撰寫條件，會篩選出布林為真（ `1` ）的觀測值

In [8]:

```sql
SELECT release_year = 1994 AS bool_values
  FROM movies
 WHERE release_year = 1994;
```

Out[8]:

bool_values
1
1
1
1
1

5 rows in set (0.00 sec)

In [9]:

```sql
SELECT *
  FROM movies
 WHERE release_year = 1994;  -- 篩選 1994 年上映的電影
```

Out[9]:

id	title	release_year	rating	director	runtime
1	The Shawshank Redemption	1994	9.3	Frank Darabont	142
8	Pulp Fiction	1994	8.9	Quentin Tarantino	154
12	Forrest Gump	1994	8.8	Robert Zemeckis	142
31	Léon: The Professional	1994	8.5	Luc Besson	110
34	The Lion King	1994	8.5	Roger Allers	88

5 rows in set (0.00 sec)

比較運算符也可以應用在文字類型的變數

In [10]:

```sql
SELECT *
  FROM movies
 WHERE director = 'Christopher Nolan';
```

Out[10]:

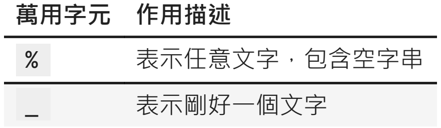
id	title	release_year	rating	director	runtime
4	The Dark Knight	2008	9	Christopher Nolan	152
13	Inception	2010	8.8	Christopher Nolan	148
29	Interstellar	2014	8.6	Christopher Nolan	169
47	The Prestige	2006	8.5	Christopher Nolan	130
54	Memento	2000	8.4	Christopher Nolan	113
71	The Dark Knight Rises	2012	8.4	Christopher Nolan	164
127	Batman Begins	2005	8.2	Christopher Nolan	140

7 rows in set (0.00 sec)

特徵比對

除了基礎比較運算符，對文字類型的變數撰寫條件時，還能夠使用具備特徵比對（Pattern matching）性質的比較運算符 `LIKE`

使用 `LIKE` 比較運算符的時候需要搭配萬用字元（Wildcards）



萬用字元	作用描述
`%`	表示任意文字，包含空字串
`_`	表示剛好一個文字

In [11]:

```
SELECT *
  FROM players
 WHERE firstName LIKE 'L%';  -- 篩選名字是 L 開頭的球員
```

Out[11]:

firstName	lastName	temporaryDisplayName	personId	teamId	jersey	pos	heightFeet	heightInches	heightMeters	weightPounds
LeBron	James	James, LeBron	2544	1610612747	23	F	6	9	2.06	250
Lou	Williams	Williams, Lou	101150	1610612737	6	G	6	1	1.85	175
LaMarcus	Aldridge	Aldridge, LaMarcus	200746	1610612751	12	C-F	6	11	2.11	250
Langston	Galloway	Galloway, Langston	204038	1610612756	2	G	6	1	1.85	200
Larry	Nance Jr.	Nance Jr., Larry	1626204	1610612739	22	F-C	6	7	2.01	245
Lonzo	Ball	Ball, Lonzo	1628366	1610612740	2	G	6	6	1.98	190
Lauri	Markkanen	Markkanen, Lauri	1628374	1610612741	24	F-C	7	0	2.13	240
Luke	Kennard	Kennard, Luke	1628379	1610612746	5	G	6	5	1.96	206
Luke	Kornet	Kornet, Luke	1628436	1610612738	40	F-C	7	2	2.18	250
Landry	Shamet	Shamet, Landry	1629013	1610612751	20	G	6	4	1.93	190
Lonnie	Walker IV	Walker IV, Lonnie	1629022	1610612759	1	G-F	6	4	1.93	204
Luka	Doncic	Doncic, Luka	1629029	1610612742	77	F-G	6	7	2.01	230
Luguentz	Dort	Dort, Luguentz	1629652	1610612760	5	G	6	3	1.9	215
Luka	Samanic	Samanic, Luka	1629677	1610612759	19	F	6	10	2.08	227
LaMelo	Ball	Ball, LaMelo	1630163	1610612766	2	G	6	6	1.98	180
Lamar	Stevens	Stevens, Lamar	1630205	1610612739	8	F	6	6	1.98	230

16 rows in set (0.00 sec)

In [12]:

```
SELECT *
  FROM players
 WHERE firstName LIKE 'L_____'; -- 篩選名字是 L 開頭後面接五個字元的球員
```

Out[12]:

firstName	lastName	temporaryDisplayName	personId	teamId	jersey	pos	heightFeet	heightInches	heightMeters	weightPounds
LeBron	James	James, LeBron	2544	1610612747	23	F	6	9	2.06	250
Landry	Shamet	Shamet, Landry	1629013	1610612751	20	G	6	4	1.93	190
Lonnie	Walker IV	Walker IV, Lonnie	1629022	1610612759	1	G-F	6	4	1.93	204
LaMelo	Ball	Ball, LaMelo	1630163	1610612766	2	G	6	6	1.98	180

4 rows in set (0.00 sec)

邏輯運算符

當`WHERE`後的條件有多個的時候，必須使用邏輯運算符結合這些條件

基礎的邏輯運算符有：

- `AND` 結合條件的交集。
- `BETWEEN` 結合數值比較條件的交集。
- `OR` 結合條件的聯集。
- `IN` 結合條件的聯集。
- `NOT` 反轉真假。

使用 AND 結合兩個條件時，要兩條件皆為真才會判斷為真，其餘狀況均為假

In [13]:

```
SELECT rating >= 8.8 AS condition_1,
       rating <= 9.0 AS condition_2,
       rating >= 8.8 AND rating <= 9.0 AS condtion_1_and_condition_2
  FROM movies
 LIMIT 15;
```

Out[13]:

condition_1	condition_2	condtion_1_and_condition_2
1	0	0
1	0	0
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
0	1	0
0	1	0

15 rows in set (0.00 sec)

In [14]:

```
SELECT *
  FROM movies
 WHERE rating >= 8.8 AND
       rating <= 9.0;  -- 評等介於 8.8 與 9.0 之間的電影
```

Out[14]:

id	title	release_year	rating	director	runtime
3	The Godfather: Part II	1974	9	Francis Ford Coppola	202
4	The Dark Knight	2008	9	Christopher Nolan	152
5	12 Angry Men	1957	9	Sidney Lumet	96
6	Schindler's List	1993	8.9	Steven Spielberg	195
7	The Lord of the Rings: The Return of the King	2003	8.9	Peter Jackson	201
8	Pulp Fiction	1994	8.9	Quentin Tarantino	154
9	The Good, the Bad and the Ugly	1966	8.8	Sergio Leone	178
10	The Lord of the Rings: The Fellowship of the Ring	2001	8.8	Peter Jackson	178
11	Fight Club	1999	8.8	David Fincher	139
12	Forrest Gump	1994	8.8	Robert Zemeckis	142
13	Inception	2010	8.8	Christopher Nolan	148

11 rows in set (0.00 sec)

以 AND 結合數值比較條件時，更推薦使用 BETWEEN

In [15]:

```
SELECT *
  FROM movies
 WHERE rating BETWEEN 8.8 AND 9.0;  -- 評等介於 8.8 與 9.0 之間的電影
```

Out[15]:

id	title	release_year	rating	director	runtime
3	The Godfather: Part II	1974	9	Francis Ford Coppola	202
4	The Dark Knight	2008	9	Christopher Nolan	152
5	12 Angry Men	1957	9	Sidney Lumet	96
6	Schindler's List	1993	8.9	Steven Spielberg	195
7	The Lord of the Rings: The Return of the King	2003	8.9	Peter Jackson	201
8	Pulp Fiction	1994	8.9	Quentin Tarantino	154
9	The Good, the Bad and the Ugly	1966	8.8	Sergio Leone	178
10	The Lord of the Rings: The Fellowship of the Ring	2001	8.8	Peter Jackson	178
11	Fight Club	1999	8.8	David Fincher	139
12	Forrest Gump	1994	8.8	Robert Zemeckis	142
13	Inception	2010	8.8	Christopher Nolan	148

11 rows in set (0.00 sec)

使用 OR 結合兩個條件時，要兩者皆為假才為假，其餘狀況均為真

In [16]:

```
SELECT divName = 'Atlantic' AS condition_1,
       divName = 'Pacific' AS condition_2,
       divName = 'Atlantic' OR divName = 'Pacific' AS condition_1_or_condition_2
  FROM teams
 LIMIT 10;
```

Out[16]:

condition_1	condition_2	condition_1_or_condition_2
0	0	0
1	0	1
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
0	0	0
0	1	1

10 rows in set (0.00 sec)

In [17]:

```
SELECT *
  FROM teams
 WHERE divName = 'Atlantic' OR
       divName = 'Pacific';  -- 分組為 Atlantic 或 Pacific 的球隊
```

Out[17]:

isNBAFranchise	isAllStar	city	altCityName	fullName	tricode	teamId	nickname	urlName	teamShortName	confName	c
1	0	Boston	Boston	Boston Celtics	BOS	1610612738	Celtics	celtics	Boston	East	
1	0	Golden State	Golden State	Golden State Warriors	GSW	1610612744	Warriors	warriors	Golden State	West	
1	0	LA	LA Clippers	LA Clippers	LAC	1610612746	Clippers	clippers	LA Clippers	West	
1	0	Los Angeles	Los Angeles Lakers	Los Angeles Lakers	LAL	1610612747	Lakers	lakers	L.A. Lakers	West	
1	0	Brooklyn	Brooklyn	Brooklyn Nets	BKN	1610612751	Nets	nets	Brooklyn	East	
1	0	New York	New York	New York Knicks	NYK	1610612752	Knicks	knicks	New York	East	
1	0	Philadelphia	Philadelphia	Philadelphia 76ers	PHI	1610612755	76ers	sixers	Philadelphia	East	
1	0	Phoenix	Phoenix	Phoenix Suns	PHX	1610612756	Suns	suns	Phoenix	West	
1	0	Sacramento	Sacramento	Sacramento Kings	SAC	1610612758	Kings	kings	Sacramento	West	
1	0	Toronto	Toronto	Toronto Raptors	TOR	1610612761	Raptors	raptors	Toronto	East	

10 rows in set (0.00 sec)

以 OR 結合比較條件時，更推薦使用 IN

In [18]:

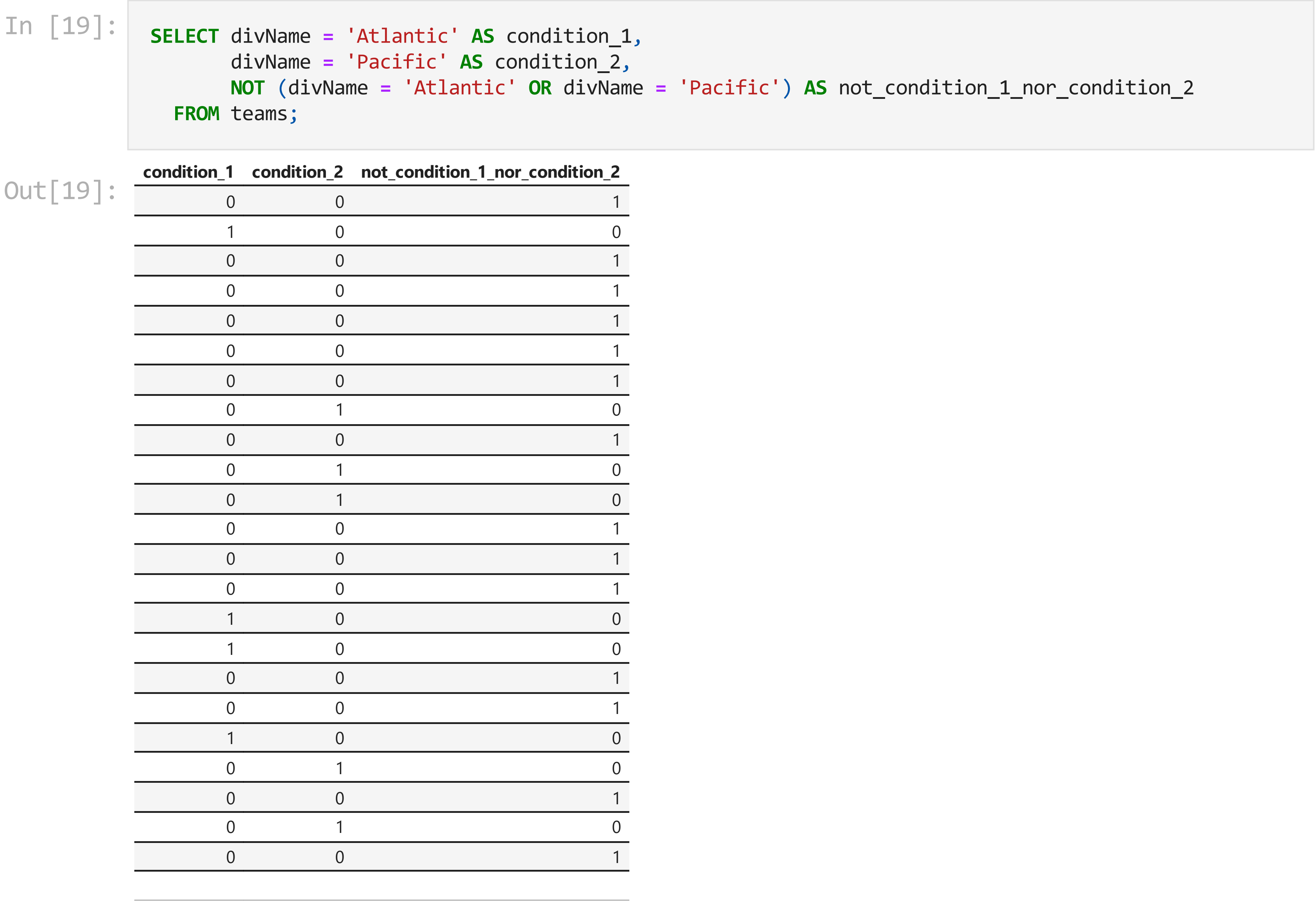
```
SELECT *
  FROM teams
 WHERE divName IN ('Atlantic', 'Pacific');  -- 分組為 Atlantic 或 Pacific 的球隊
```

Out[18]:

isNBAFranchise	isAllStar	city	altCityName	fullName	tricode	teamId	nickname	urlName	teamShortName	confName
1	0	Boston	Boston	Boston Celtics	BOS	1610612738	Celtics	celtics	Boston	East
1	0	Golden State	Golden State	Golden State Warriors	GSW	1610612744	Warriors	warriors	Golden State	West
1	0	LA	LA Clippers	LA Clippers	LAC	1610612746	Clippers	clippers	LA Clippers	West
1	0	Los Angeles	Los Angeles Lakers	Los Angeles Lakers	LAL	1610612747	Lakers	lakers	L.A. Lakers	West
1	0	Brooklyn	Brooklyn	Brooklyn Nets	BKN	1610612751	Nets	nets	Brooklyn	East
1	0	New York	New York	New York Knicks	NYK	1610612752	Knicks	knicks	New York	East
1	0	Philadelphia	Philadelphia	Philadelphia 76ers	PHI	1610612755	76ers	sixers	Philadelphia	East
1	0	Phoenix	Phoenix	Phoenix Suns	PHX	1610612756	Suns	suns	Phoenix	West
1	0	Sacramento	Sacramento	Sacramento Kings	SAC	1610612758	Kings	kings	Sacramento	West
1	0	Toronto	Toronto	Toronto Raptors	TOR	1610612761	Raptors	raptors	Toronto	East

10 rows in set (0.00 sec)

使用 NOT 將條件的比較結果反轉，亦即真假互換



In [19]:

```
SELECT divName = 'Atlantic' AS condition_1,
       divName = 'Pacific' AS condition_2,
       NOT (divName = 'Atlantic' OR divName = 'Pacific') AS not_condition_1_nor_condition_2
  FROM teams;
```

Out[19]:

condition_1	condition_2	not_condition_1_nor_condition_2
0	0	1
1	0	0
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	1	0
0	0	1
0	1	0
0	1	0
0	0	1
0	0	1
0	0	1
1	0	0
1	0	0
0	0	1
0	0	1
1	0	0
0	1	0
0	0	1
0	1	0
0	0	1

0	0	1
1	0	0
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1

30 rows in set (0.00 sec)

In [20]:

```
SELECT *
  FROM teams
 WHERE divName NOT IN ('Atlantic', 'Pacific');
```

Out[20]:

isNBAFranchise	isAllStar	city	altCityName	fullName	tricode	teamId	nickname	urlName	teamShortName	conf
1	0	Atlanta	Atlanta	Atlanta Hawks	ATL	1610612737	Hawks	hawks	Atlanta	
1	0	Cleveland	Cleveland	Cleveland Cavaliers	CLE	1610612739	Cavaliers	cavaliers	Cleveland	
1	0	New Orleans	New Orleans	New Orleans Pelicans	NOP	1610612740	Pelicans	pelicans	New Orleans	
1	0	Chicago	Chicago	Chicago Bulls	CHI	1610612741	Bulls	bulls	Chicago	
1	0	Dallas	Dallas	Dallas Mavericks	DAL	1610612742	Mavericks	mavericks	Dallas	
1	0	Denver	Denver	Denver Nuggets	DEN	1610612743	Nuggets	nuggets	Denver	
1	0	Houston	Houston	Houston Rockets	HOU	1610612745	Rockets	rockets	Houston	
1	0	Miami	Miami	Miami Heat	MIA	1610612748	Heat	heat	Miami	
1	0	Milwaukee	Milwaukee	Milwaukee Bucks	MIL	1610612749	Bucks	bucks	Milwaukee	
1	0	Minnesota	Minnesota	Minnesota Timberwolves	MIN	1610612750	Timberwolves	timberwolves	Minnesota	
1	0	Orlando	Orlando	Orlando Magic	ORL	1610612753	Magic	magic	Orlando	
1	0	Indiana	Indiana	Indiana Pacers	IND	1610612754	Pacers	pacers	Indiana	
1	0	Portland	Portland	Portland Trail Blazers	POR	1610612757	Trail Blazers	blazers	Portland	
1	0	San Antonio	San Antonio	San Antonio Spurs	SAS	1610612759	Spurs	spurs	San Antonio	
1	0	Oklahoma City	Oklahoma City	Oklahoma City Thunder	OKC	1610612760	Thunder	thunder	Oklahoma City	
1	0	Utah	Utah	Utah Jazz	UTA	1610612762	Jazz	jazz	Utah	
1	0	Memphis	Memphis	Memphis Grizzlies	MEM	1610612763	Grizzlies	grizzlies	Memphis	

1	0	Washington	Washington	Washington Wizards	WAS	1610612764	Wizards	wizards	Washington
1	0	Detroit	Detroit	Detroit Pistons	DET	1610612765	Pistons	pistons	Detroit
1	0	Charlotte	Charlotte	Charlotte Hornets	CHA	1610612766	Hornets	hornets	Charlotte

20 rows in set (0.00 sec)

遺漏值的比較運算符

`NULL` 遺漏值（或稱空值）不適用基礎比較運算符

```
In [21]: SELECT *
           FROM lookup_table
          WHERE Province_State = NULL AND
                Admin2 = NULL;
```

```
Out[21]: Empty set (0.00 sec)
```

要判斷是否為遺漏值，必須使用 `IS NULL` 作為比較運算符

In [22]:

```
SELECT *
  FROM lookup_table
 WHERE Province_State IS NULL AND
       Admin2 IS NULL
 LIMIT 10;
```

Out[22]:

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
20	Andorra	AD	AND	Andorra	NULL	NULL	42.5063	1.5218	77265
24	Angola	AO	AGO	Angola	NULL	NULL	-11.2027	17.8739	32866268
28	Antigua and Barbuda	AG	ATG	Antigua and Barbuda	NULL	NULL	17.0608	-61.7964	97928
32	Argentina	AR	ARG	Argentina	NULL	NULL	-38.4161	-63.6167	45195777
51	Armenia	AM	ARM	Armenia	NULL	NULL	40.0691	45.0382	2963234
40	Austria	AT	AUT	Austria	NULL	NULL	47.5162	14.5501	9006400
31	Azerbaijan	AZ	AZE	Azerbaijan	NULL	NULL	40.1431	47.5769	10139175

10 rows in set (0.00 sec)

In [23]:

```
SELECT *
  FROM lookup_table
 WHERE Province_State IS NOT NULL AND
       Admin2 IS NOT NULL
 LIMIT 10;
```

Out[23]:

UID	Combined_Key	iso2	iso3	Country_Region	Province_State	Admin2	Lat	Long_	Population
535	Bonaire, Sint Eustatius and Saba, Netherlands	BQ	BES	Netherlands	Sint Eustatius and Saba	Bonaire	12.1784	-68.2385	26221
654	Saint Helena, Ascension and Tristan da Cunha, United Kingdom	SH	SHN	United Kingdom	Ascension and Tristan da Cunha	Saint Helena	-7.9467	-14.3559	5661
63072001	Adjuntas, Puerto Rico, US	PR	PRI	US	Puerto Rico	Adjuntas	18.180117	-66.754367	17363
63072003	Aguada, Puerto Rico, US	PR	PRI	US	Puerto Rico	Aguada	18.360255	-67.175131	36694
63072005	Aguadilla, Puerto Rico, US	PR	PRI	US	Puerto Rico	Aguadilla	18.459681	-67.120815	50265
63072007	Aguas Buenas, Puerto Rico, US	PR	PRI	US	Puerto Rico	Aguas Buenas	18.251619	-66.126806	24814
63072009	Aibonito, Puerto Rico, US	PR	PRI	US	Puerto Rico	Aibonito	18.131361	-66.264131	22108
63072011	Anasco, Puerto Rico, US	PR	PRI	US	Puerto Rico	Anasco	18.287985	-67.120611	26161
63072013	Arecibo, Puerto Rico, US	PR	PRI	US	Puerto Rico	Arecibo	18.406631	-66.675077	81966
63072015	Arroyo, Puerto Rico, US	PR	PRI	US	Puerto Rico	Arroyo	17.998457	-66.056546	17238

10 rows in set (0.00 sec)

重點統整

- 加入 `WHERE` 能夠以條件作為篩選觀測值的依據。
- 對變數應用比較運算符生成條件。
- 對文字類型的變數撰寫條件，能夠使用特徵比對的運算符 `LIKE` 搭配萬用字元。
- 當 `WHERE` 後的條件有多個的時候，必須使用邏輯運算符結合這些條件。
- 判斷是否為遺漏值，必須使用 `IS NULL` 作為比較運算符。

```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
 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
```