
資料庫入門

SQL語言基礎查詢

鄭安翔

ansel_cheng@hotmail.com

課程大綱

1) 資料庫基本概念

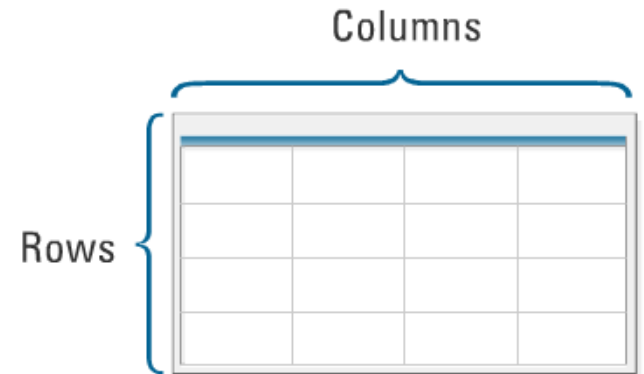
- 表格、記錄與欄位
- 結構化查詢語言
- 資料型態

2) 基本查詢敘述

表格、記錄與欄位

■ 表格

- 儲存資料的基本元件
- 欄與列所組成邏輯性的資料結構



■ 記錄

- 表格裡面儲存的每一筆資料稱為記錄(**Record**)或列(**row**)
- 每一筆記錄(**Record**) 有著相同數量與型態的欄位

■ 欄位

- 欄位資料的型態及邏輯意義相同
- 表格設計時，通常會指定一個欄位為主鍵(**primary key**)
- 主鍵不可重複，可以用來代表一筆資料

表格、記錄與欄位

LID	year	season	title
001	2001	Spring	Soccer League (Spring '01)
002	2001	Summer	Summer Soccer Fest 2001
003	2001	Fall	Fall Soccer League 2001
004	2004	Summer	The Summer of Soccer Love

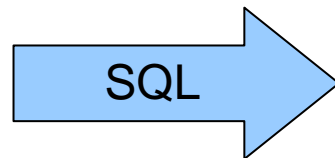
PID	name	address	city	province	postal_code
047	Steve Sterling	12 Grove Park Road	Manchester	Manchester	M4 6NF
048	Alice Hornblower	62 Woodside Lane	Reading	Berks	RG31 9TT
049	Wally Winkle	17 Chippenham Road	London	London	SW19 4FT

LID	PID
001	047
001	048
002	048
002	049
003	048

結構化查詢語言 SQL

■ Structured Query Language

- ❑ 與關聯式資料庫系統 RDBMS 溝通的標準語言
- ❑ 可進行新增(Insert)、查詢(Query)、修改(Update)、刪除>Delete)等交易(Transaction).
- ❑ 在標準SQL 語法的規範下，不同的RDBMS廠商可能會擁有自己非標準的SQL 語法產生.



SQL語法的分類

- 資料定義語言DDL(Data Definition Language)
 - 建立、修改或刪除 table、schema、domain、index、view
 - 主要指令：CREATE、ALTER、DROP
- 資料操作語言DML(Data Manipulation Language)
 - 新增、修改、刪除資料內容
 - 主要指令：INSERT、UPDATE、DELETE

SQL語法的分類

- 資料查詢語言 DQL (Data Query Language)
 - 用來查詢資料內容
 - 主要指令：**SELECT**
- 資料控制語言DCL(Data Control Language)
 - 管理資料庫的交易及安全性
 - 主要指令：**COMMIT**、**ROLLBACK**、**GRANT**、**REVOKE**

資料庫資料型態

- 資料庫中可以儲存各種不同類型的資料
 - 不同資料庫支援的資料型態不完全相同
- SQL語法中常用的資料型態
 - 整數數值
 - 實數(小數)數值
 - 文字字串：前後使用單引號或雙引號括起來
 - 日期：前後使用單引號或雙引號括起來
 - 預設格式為 '年-月-日'
 - 空值：用來表示未知或空的資料
 - NULL

常用的資料型態

	empno	ename	job	manager	hiredate	salary	comm	deptno
▶	7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
	7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
	7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
	7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
	7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	NULL
	7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
	7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	NULL	20
	7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
	7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30

課程大綱

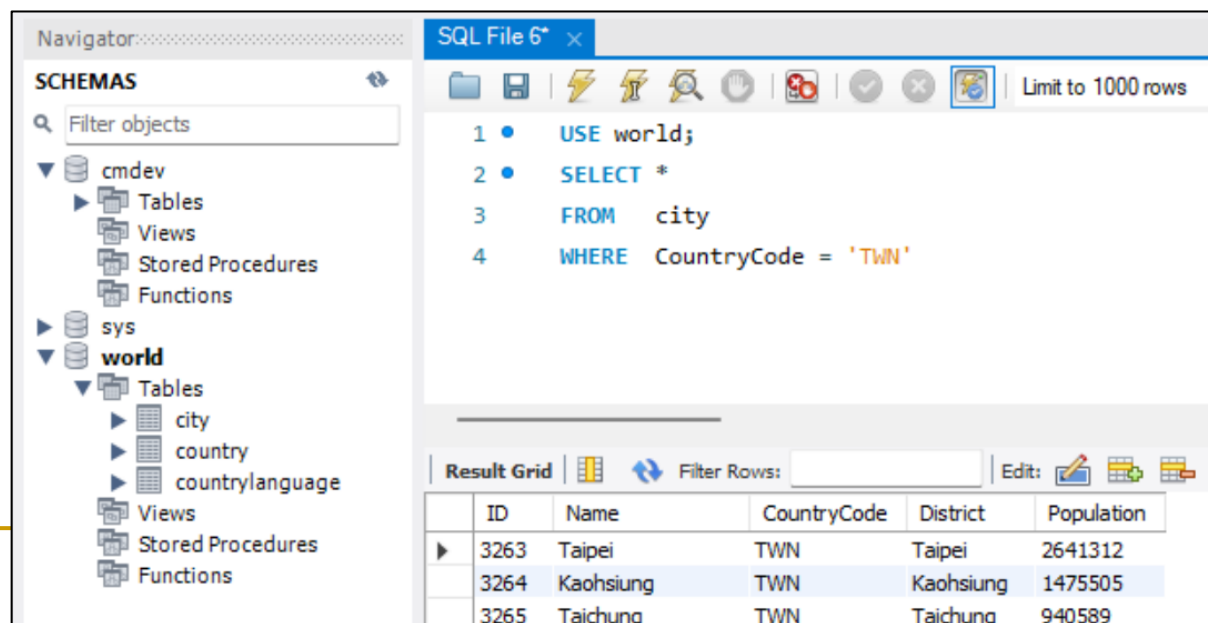
1) 資料庫結構基本概念

2) 基本查詢敘述

- 指定資料庫
- 基本查詢語法
- 查詢條件
- 排序設定
- 限制設定

指定資料庫

- 資料庫伺服器可以建立多個資料庫
 - 執行資料庫操作前，需先指定要使用的資料庫
 - 語法：**USE** 資料庫名稱;
 - **MySQL Workbench** 目前使用的資料庫名稱會變成粗體字



SQL 查詢語法

- 查詢敘述的基本語法：

SELECT 查詢欄位

FROM 查詢表格

WHERE 查詢條件

GROUP BY 分組設定

HAVING 分組條件

ORDER BY 排序設定

LIMIT 限制設定



選擇性的條件子句

- 注意條件子句的順序，順序不對會發生錯誤

指定查詢欄位

■ 語法：

SELECT 欄位1[, 欄位2...] **FROM** 表格名;

□ 依照需求，決定查詢的欄位和順序

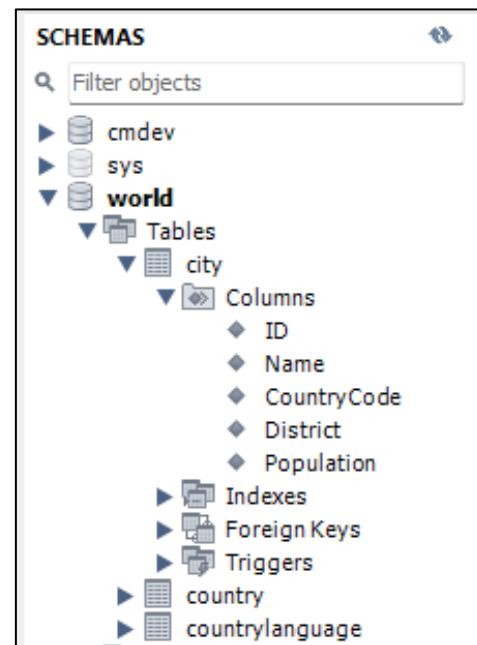
- 只取出指定欄位的資料
- 一個以上欄位以逗號隔開

□ 顯示資料時依語法中的順序顯示

- 與表格Schema欄位順序無關

□ 欄位使用 * ，表示查詢全部欄位

- 依表格Schema欄位順序顯示



指定查詢表格

- 語法：

SELECT 欄位1[, 欄位2...] **FROM** 資料庫名稱.表格名稱;

SELECT 欄位1[, 欄位2...] **FROM** 表格名稱;

- 查詢指定資料庫中的表格資料
- 查詢目前使用中的資料庫時，可直接使用表格名稱
 - 使用 **USE** 敘述轉換使用的資料庫

範例

- 查詢 cmdev資料庫 emp表格 所有資料

範例

- 查詢 cmdev資料庫 emp表格 所有資料

範例

- 查詢 world資料庫 city表格 ID, Name, District 資料

欄位顯示設定

■ 運算式

- 欄位資料查詢出來，可加上運算後顯示
- 欄位名稱以運算式顯示
- 運算式不包含欄位時，顯示運算結果

■ 別名(alias)

- 使用自訂名稱顯示查詢的欄位或運算結果
- 運算式使用別名可避免欄位名稱過長的問題
- 別名應避免空白或**SQL**保留字
 - 單引號或雙引號

欄位顯示設定

- 語法

SELECT {欄位|**運算式**} (**AS**) [**別名**] [, ...]

FROM 表格名稱

- AS 可省略

數學運算

優先順序	運算符號	用途	範例	運算結果
1	*	乘法	7*3	21
	/	真除法	7/3	2.333
	DIV	整數除法	7 DIV 3	2
	%	求餘數	7%3	1
	MOD	求餘數	7 MOD 3	1
2	+	加法	7+3	10
	-	減法	7-3	4

- 括號中的運算式優先計算

範例

- 查詢 運算式不包含欄位資料

範例

- 查詢 `cmdev.emp` 中姓名,薪水,年薪, 年薪加半個月年終

範例

- 查詢 `cmdev.emp` 中姓名(欄位以 `select` 顯示), 年薪(欄位以 `Annual Salary` 顯示)

條件查詢

- WHERE 子句設定查詢條件

- 語法：

SELECT 欄位1... FROM 表格名稱

WHERE 查詢條件;

- 定義查詢條件

- 關係運算

- 邏輯運算

- 特殊條件

關係運算

- 字串型態比大小為字典排序
- 日期型態資料
 - = 表示指定某一日期
 - < 表示指定日期之前，> 表示指定日期之後

運算符號	用途	範例
=	等於	empno = 7369
<=>	等於	ename <=> 'SMITH'
!=	不等於	job != 'SALESMAN'
<>	不等於	manager <> 7902
<	小於	hiredate < '1980-12-17'
<=	小於等於	salary <= 2500
>	大於	comm > 300
>=	大於等於	deptno >= 20

範例

- 查詢 world資料庫 city表格 台灣的城市
- 查詢 world資料庫 city表格 非台灣的城市

範例

- 查詢 world資料庫 city表格 人口少於 800人的城市
- 查詢 world資料庫 city表格 人口少於或等於 800人的城市

範例

- 查詢 emp表格 雇用時間為 1981-09-08 的資料

範例

- 查詢 emp表格 雇用時間在 1981-09-08 之前的資料
- 查詢 emp表格 雇用時間在 1981-09-08 之後的資料

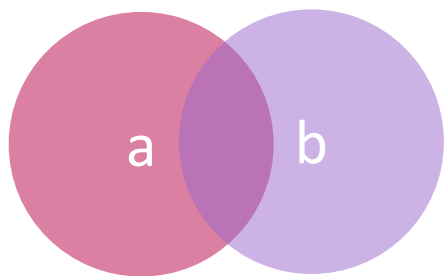
邏輯運算

- 運用邏輯運算子可以組合多個條件式，表達更複雜的邏輯條件

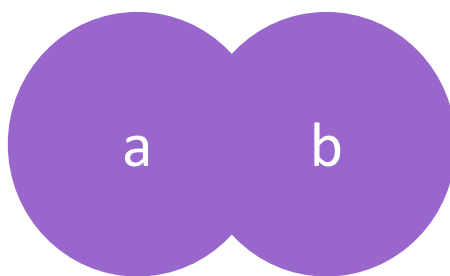
優先順序	運算符號	用途	範例
1	NOT	非	NOT CountryCode = 'TWN'
2	&&	且	CountryCode = 'TWN' AND Population < 100000
	AND		
3		或	CountryCode = 'TWN' OR CountryCode='USA'
	OR		
	XOR	互斥	

邏輯運算

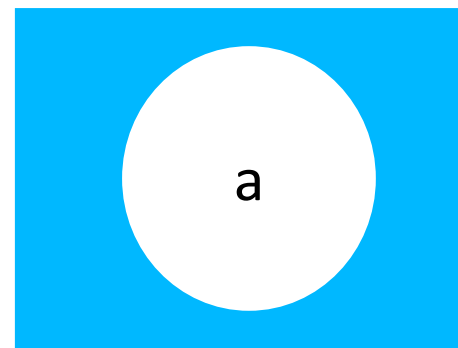
- 「AND」 且 $(a \geq 10) \text{ AND } (a \leq 20)$
- 「OR」 或 $(a = 5) \text{ OR } (a = 10)$
- 「NOT」 非 $\text{NOT}(a = 200)$



A and B



A OR B



NOT a

範例

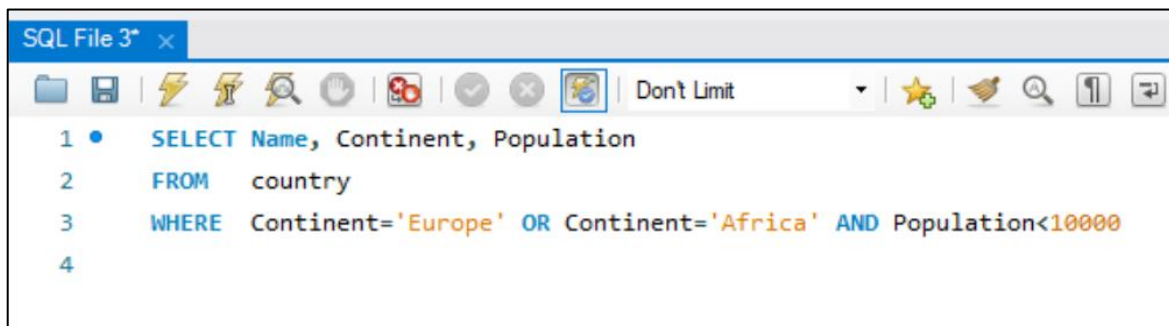
- 查詢 world資料庫 city表格 非台灣的城市
 - 使用 NOT

範例

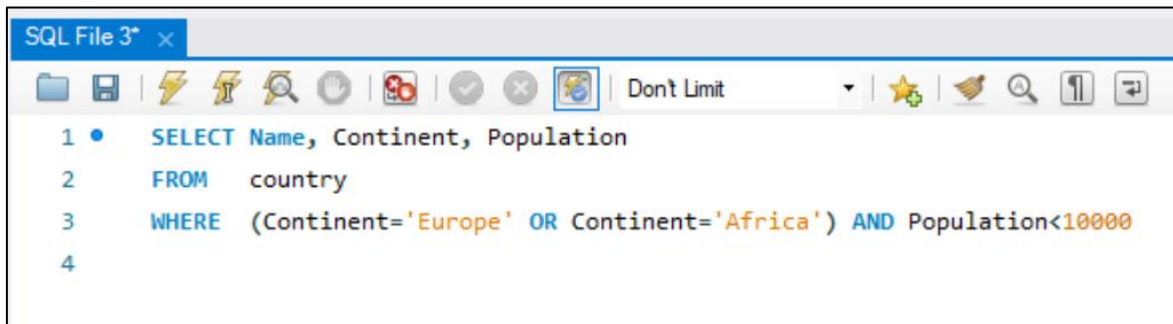
- 查詢 city 表格中 台灣人口少於 100000 人的城市
- 查詢 city 表格中 台灣或美國的城市

範例

- 這兩個查詢一樣嗎？



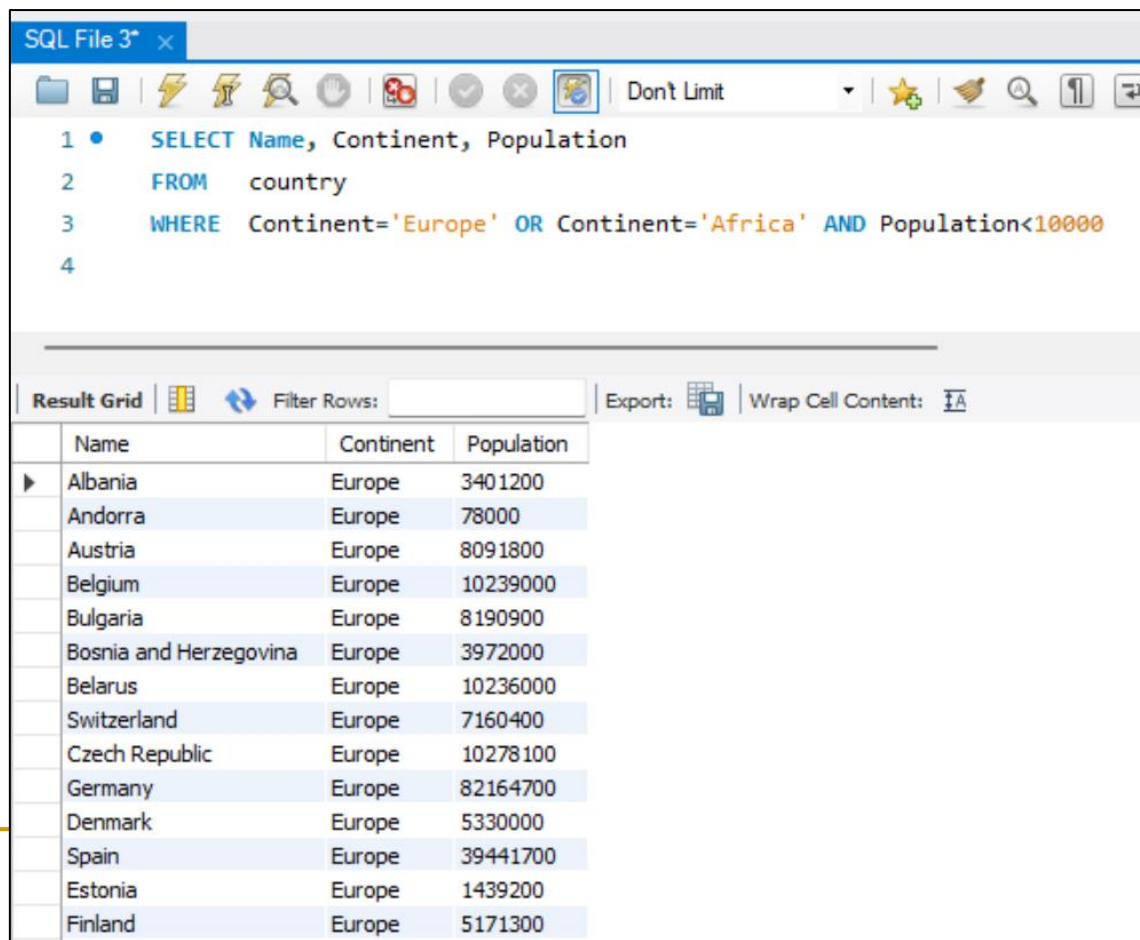
```
SQL File 3* x
1 • SELECT Name, Continent, Population
2   FROM country
3   WHERE Continent='Europe' OR Continent='Africa' AND Population<10000
4
```



```
SQL File 3* x
1 • SELECT Name, Continent, Population
2   FROM country
3   WHERE (Continent='Europe' OR Continent='Africa') AND Population<10000
4
```

範例

- 查詢 歐洲或(非洲人口少於 10000人)的城市
 - AND優先於OR



The screenshot shows a SQL IDE window titled "SQL File 3* x". The query editor contains the following SQL code:

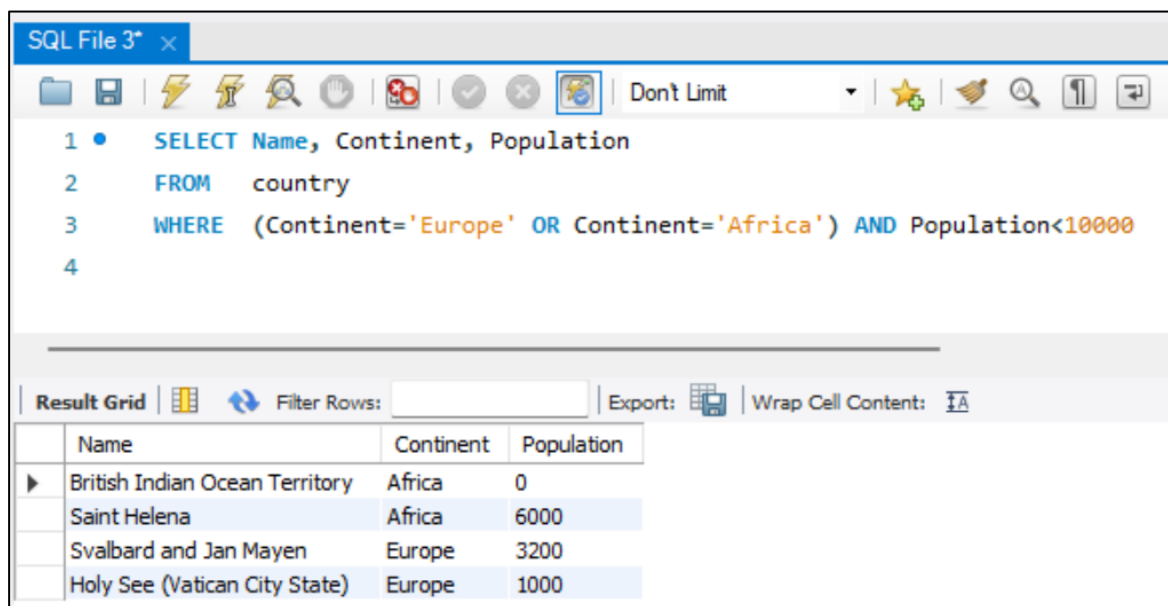
```
1 • SELECT Name, Continent, Population
2 FROM country
3 WHERE Continent='Europe' OR Continent='Africa' AND Population<10000
4
```

Below the query editor is the "Result Grid" section, which displays the results of the query. The grid has four columns: Name, Continent, and Population. The results show a list of countries from Europe, all with populations less than 10,000.

	Name	Continent	Population
▶	Albania	Europe	3401200
	Andorra	Europe	78000
	Austria	Europe	8091800
	Belgium	Europe	10239000
	Bulgaria	Europe	8190900
	Bosnia and Herzegovina	Europe	3972000
	Belarus	Europe	10236000
	Switzerland	Europe	7160400
	Czech Republic	Europe	10278100
	Germany	Europe	82164700
	Denmark	Europe	5330000
	Spain	Europe	39441700
	Estonia	Europe	1439200
	Finland	Europe	5171300

範例

- 查詢 (歐洲或非洲國家中)人口少於 10000人的城市



The screenshot shows a SQL IDE window titled "SQL File 3* x". The query editor contains the following SQL code:

```
1 • SELECT Name, Continent, Population
2 FROM country
3 WHERE (Continent='Europe' OR Continent='Africa') AND Population<10000
4
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are as follows:

	Name	Continent	Population
▶	British Indian Ocean Territory	Africa	0
	Saint Helena	Africa	6000
	Svalbard and Jan Mayen	Europe	3200
	Holy See (Vatican City State)	Europe	1000

特殊運算

運算符號	用途	範例
BETWEEN 開始值 AND 結束值	在開始值與結 束值範圍中	salary BETWEEN 1000 AND 2000
IN 集合	包含成員	CountryCode IN ('TWN', 'USA', 'JPN')
IS	判斷是NULL	deptno IS NULL
IS NOT	判斷不是NULL	deptno IS NOT NULL
LIKE	判斷字串樣式	ename LIKE 'S%'

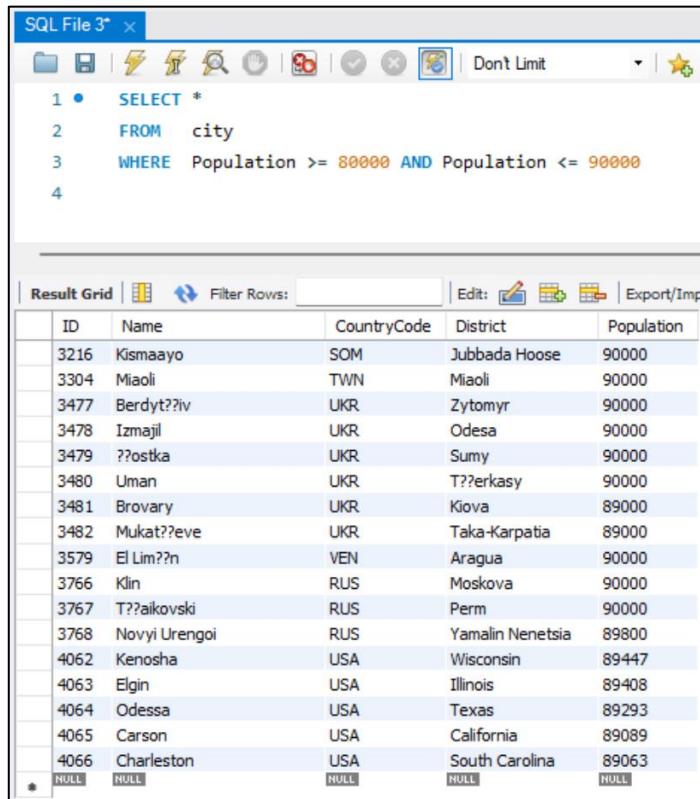
- BETWEEN AND 語法包含指定的開始值和結束值

salary BETWEEN 1000 AND 2000

salary >= 1000 AND salary <= 2000

範例

■ 使用 BETWEEN ... AND ... 改寫



The screenshot shows a SQL IDE window titled "SQL File 3*" with a toolbar and a query editor. The query is:

```
1 SELECT *
2 FROM city
3 WHERE Population >= 80000 AND Population <= 90000
4
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has columns: ID, Name, CountryCode, District, and Population. The results are as follows:

ID	Name	CountryCode	District	Population
3216	Kismaayo	SOM	Jubbada Hoose	90000
3304	Miaoli	TWN	Miaoli	90000
3477	Berdyt??iv	UKR	Zytomyr	90000
3478	Izmajil	UKR	Odesa	90000
3479	??ostka	UKR	Sumy	90000
3480	Uman	UKR	T??erkasy	90000
3481	Brovary	UKR	Kiova	89000
3482	Mukat??eve	UKR	Taka-Karpatia	89000
3579	El Lim??n	VEN	Aragua	90000
3766	Klin	RUS	Moskova	90000
3767	T??alkovski	RUS	Perm	90000
3768	Novyi Urengoi	RUS	Yamalin Nenetsia	89800
4062	Kenosha	USA	Wisconsin	89447
4063	Elgin	USA	Illinois	89408
4064	Odessa	USA	Texas	89293
4065	Carson	USA	California	89089
4066	Charleston	USA	South Carolina	89063
*	NULL	NULL	NULL	NULL

範例

■ 這兩個查詢一樣嗎？

SQL File 3*

```
1 SELECT *
2 FROM city
3 WHERE Population > 80000 AND Population < 90000
4
```

Result Grid

ID	Name	CountryCode	District	Population
2444	al-Zawiya	LBY	al-Zawiya	89338
2452	Luxembourg [Luxemburg/...	LUX	Luxembourg	80700
2686	El Fuerte	MEX	Sinaloa	89556
2687	Tierra Blanca	MEX	Veracruz	89143
2737	Masaya	NIC	Masaya	88971
2879	Wazirabad	PAK	Punjab	89700
2880	Nowshera	PAK	Nothwest Border...	89400
3159	Velbert	DEU	Nordrhein-Westf...	89881
3160	Esslingen am Neckar	DEU	Baden-W??rttem...	89667
3481	Brovary	UKR	Kiova	89000
3482	Mukat??eve	UKR	Taka-Karpatia	89000
3768	Novyi Urengoi	RUS	Yamalin Nenetsia	89800
4062	Kenosha	USA	Wisconsin	89447
4063	Elgin	USA	Illinois	89408
4064	Odessa	USA	Texas	89293
4065	Carson	USA	California	89089
4066	Charleston	USA	South Carolina	89063
NULL	NULL	NULL	NULL	NULL

SQL File 3*

```
1 SELECT *
2 FROM city
3 WHERE Population BETWEEN 80000 AND 90000
4
```

Result Grid

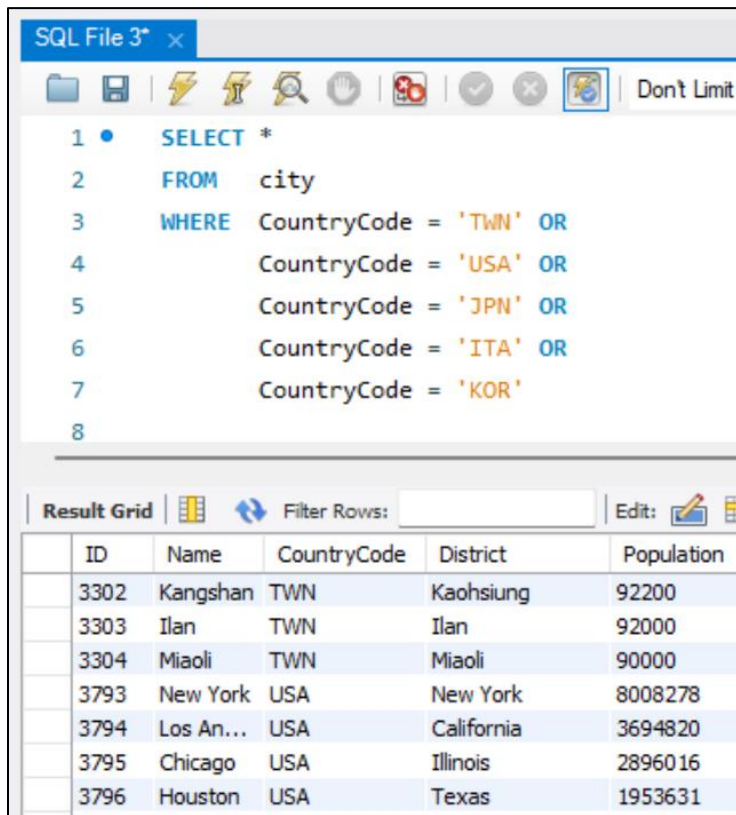
ID	Name	CountryCode	District	Population
3216	Kismaayo	SOM	Jubbada Hoose	90000
3304	Miaoli	TWN	Miaoli	90000
3477	Berdyt??iv	UKR	Zytomyr	90000
3478	Izmajil	UKR	Odesa	90000
3479	??ostka	UKR	Sumy	90000
3480	Uman	UKR	T??erkasy	90000
3481	Brovary	UKR	Kiova	89000
3482	Mukat??eve	UKR	Taka-Karpatia	89000
3579	El Lim??n	VEN	Aragua	90000
3766	Klin	RUS	Moskova	90000
3767	T??aikovski	RUS	Perm	90000
3768	Novyi Urengoi	RUS	Yamalin Nenetsia	89800
4062	Kenosha	USA	Wisconsin	89447
4063	Elgin	USA	Illinois	89408
4064	Odessa	USA	Texas	89293
4065	Carson	USA	California	89089
4066	Charleston	USA	South Carolina	89063
NULL	NULL	NULL	NULL	NULL

範例

- emp表格 雇用時間在 1981-01-01~1981-06-30 之間的資料

範例

■ 使用 IN (...) 改寫



The screenshot shows an SQL IDE window titled "SQL File 3* x". The query editor contains the following SQL code:

```
1 • SELECT *
2 FROM city
3 WHERE CountryCode = 'TWN' OR
4        CountryCode = 'USA' OR
5        CountryCode = 'JPN' OR
6        CountryCode = 'ITA' OR
7        CountryCode = 'KOR'
8
```

Below the query editor is a "Result Grid" tab. It shows a table with 6 columns: ID, Name, CountryCode, District, and Population. The table contains 7 rows of data. The "Filter Rows:" field is empty, and the "Edit:" button is visible.

ID	Name	CountryCode	District	Population
3302	Kangshan	TWN	Kaohsiung	92200
3303	Ilan	TWN	Ilan	92000
3304	Miaoli	TWN	Miaoli	90000
3793	New York	USA	New York	8008278
3794	Los An...	USA	California	3694820
3795	Chicago	USA	Illinois	2896016
3796	Houston	USA	Texas	1953631

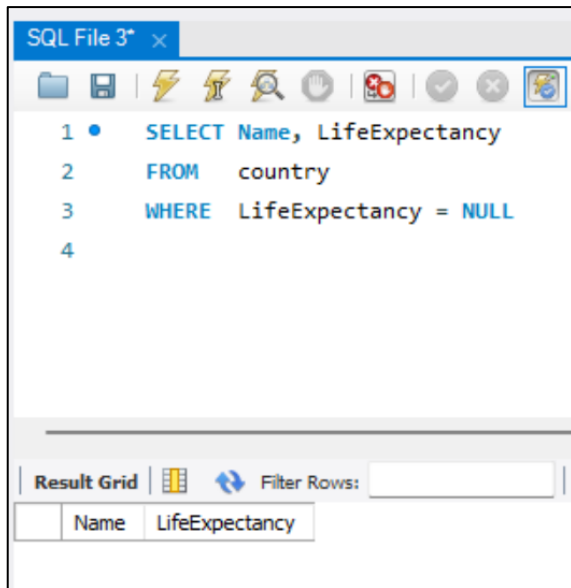
NULL值判斷

■ NULL 值

- 非必要欄位沒有值時，以 **NULL** 值表示
- 查詢欄位**NULL**的資料，不可使用 **=** 判斷
 - 使用 **IS NULL** 來取出該欄位沒有值的資料
 - 也可以使用 **<=>** 判斷
`deptno <=> NULL`
- 查詢欄位不為**NULL**的資料，不可使用 **!=** 或 **<>**
 - 使用 **IS NOT NULL** 來取出該欄位有值的資料

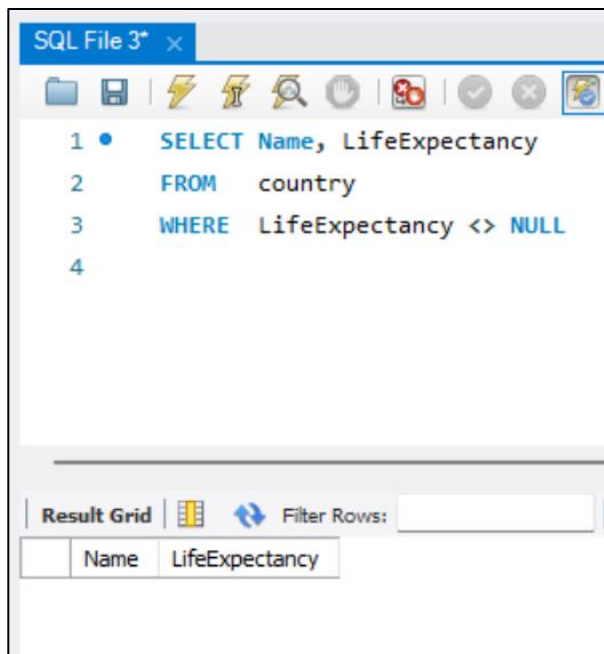
範例

- 查詢欄位NULL的資料，不可使用 = 判斷



範例

- 查詢欄位不為NULL的資料，不可使用 != 或 <>

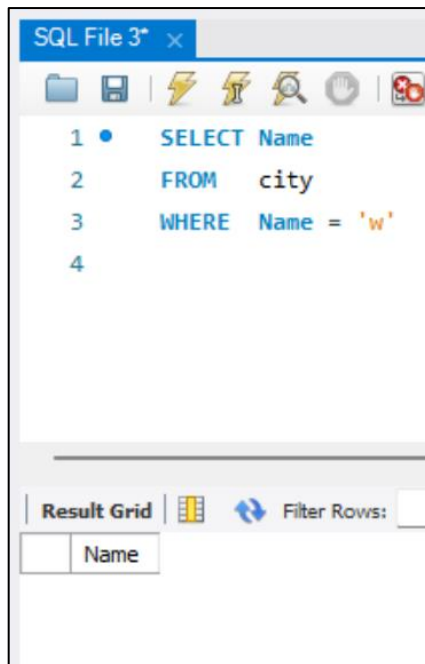


字串樣式比對

- 判斷欄位資料是否符合某種字串樣式
 - 模糊的條件比對
 - 使用 **LIKE** 取代 **=**
- **%** 表示任意字串
 - 0 到多個任何字元
 - **name LIKE 'S%'**：找出名字以**S**開頭的資料
- **_** 表示任意一個字元
 - **name LIKE '_O_'**：
 - 名字總共有四個字母
 - 第二個字母為**O**

範例

■ 字串樣式比對

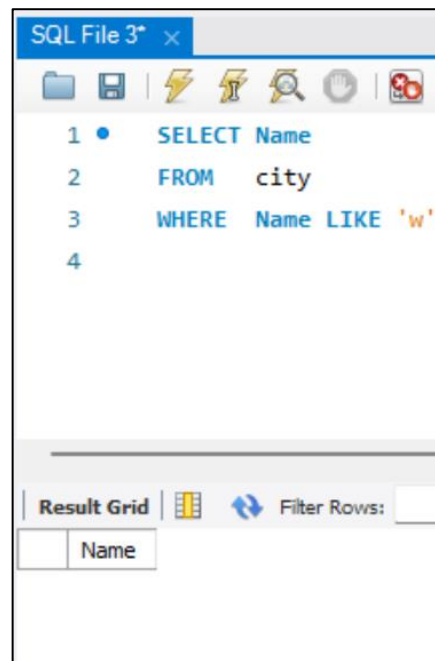


SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name = 'w'
4
```

Result Grid | Filter Rows:

Name



SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name LIKE 'w'
4
```

Result Grid | Filter Rows:

Name

範例

■ 下列樣式比對的意義？

SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name LIKE 'w%'
4
```

Result Grid | Filter Rows:

Name
Willemstad
Wollongong
Wolverhampton
Walsall
West Bromwich
Watford
Worthing

SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name LIKE '%w'
4
```

Result Grid | Filter Rows:

Name
Glasgow
Lucknow
Krak??w
Wroclaw
Rzesz??w
Chorz??w
Tarn??w
Ch??rjew
Moscow

SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name LIKE '%w%'
4
```

Result Grid | Filter Rows:

Name
Zwolle
Willemstad
Almirante Brown
Newcastle
Wollongong
Townsville
Nawabganj
Bridgetown
Antwerpen
Francistown

範例

■ 下列樣式比對的意義？

SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name LIKE 'w____'
4
```

Result Grid | Filter Rows:

	Name
▶	Welkom
	Wardha
	Wuzhou
	Weinan
	Weihai
	Wonsan
	Witten
	Warren

SQL File 3* x

```
1 • SELECT Name
2 FROM city
3 WHERE Name LIKE '____w'
4
```

Result Grid | Filter Rows:

	Name
▶	Moscow

範例

■ 下列樣式比對的意義？

SQL File 3* x

1 • SELECT Name
2 FROM city
3 WHERE Name LIKE '____w%'
4

Result Grid | Filter Rows:

Name
▶ Mallawi
al-Hawamidiya
Chatsworth
Banyuwangi
Bhusawal
Khandwa
Navadwip
al-Diwaniya
Kashiwa
Kishiwada
Okinawa

SQL File 3* x

1 • SELECT Name
2 FROM city
3 WHERE Name LIKE '_____%'
4

Result Grid | Filter Rows: | Export: | Wrap Cell Co

Name
▶ San Fernando del Valle de Cata
Santo Domingo de los Colorados
Castell??n de la Plana [Castell
Thiruvananthapuram (Trivandrum
Luxembourg [Luxemburg/L??tzebuerg]

資料排序

- 查詢傳回資料預設依資料新增到表格的順序顯示
- 設定資料排序的規則語法：

ORDER BY {名稱|運算式|索引} [ASC|DESC] (...)

- 排序基準可以是欄位名稱/別名、運算式或欄位索引
- 多重基準以逗號，隔開
 - 前面基準的值相同，比較下一個基準排序
- 排序方法
 - 由小到大遞增(ASC)，預設可省略
 - 由大到小遞減(DESC)

範例

SQL File 3* x

```
1 • SELECT  CountryCode, Name
2 FROM      city
3 ORDER BY  CountryCode
4
```

Result Grid | Filter Rows:

	CountryCode	Name
▶	ABW	Oranjestad
	AFG	Kabul
	AFG	Qandahar
	AFG	Herat
	AFG	Mazar-e-Sharif
	AGO	Luanda
	AGO	Huambo
	AGO	Lobito
	AGO	Benguela
	AGO	Namibe
	AIA	South Hill
	AIA	The Valley
	ALB	Tirana
	AND	Andorra la Vella
	ANT	Willemstad
	ARE	Dubai
	ARE	Abu Dhabi

SQL File 3* x

```
1 • SELECT  CountryCode, Name
2 FROM      city
3 ORDER BY  CountryCode ASC
4
```

Result Grid | Filter Rows:

	CountryCode	Name
▶	ABW	Oranjestad
	AFG	Kabul
	AFG	Qandahar
	AFG	Herat
	AFG	Mazar-e-Sharif
	AGO	Luanda
	AGO	Huambo
	AGO	Lobito
	AGO	Benguela
	AGO	Namibe
	AIA	South Hill
	AIA	The Valley
	ALB	Tirana
	AND	Andorra la Vella
	ANT	Willemstad
	ARE	Dubai
	ARE	Abu Dhabi

SQL File 3* x

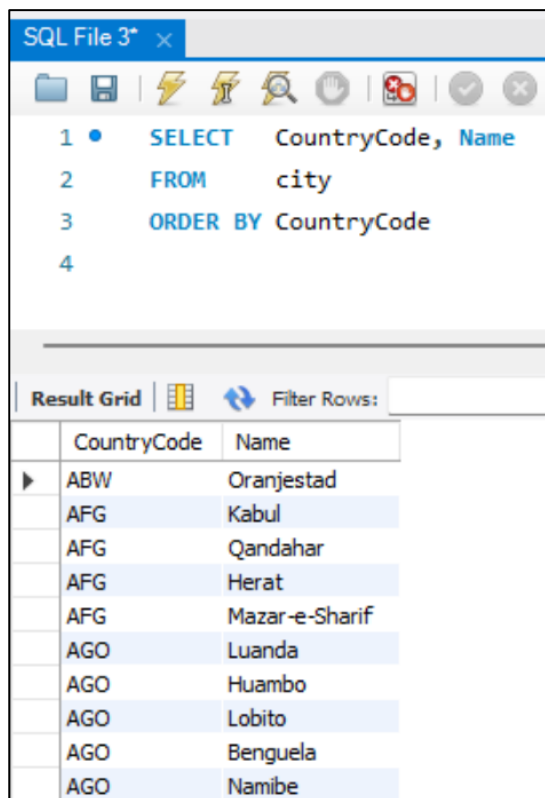
```
1 • SELECT  CountryCode, Name
2 FROM      city
3 ORDER BY  CountryCode DESC
4
```

Result Grid | Filter Rows:

	CountryCode	Name
▶	ZWE	Harare
	ZWE	Bulawayo
	ZWE	Chitungwiza
	ZWE	Mount Darwin
	ZWE	Mutare
	ZWE	Gweru
	ZMB	Lusaka
	ZMB	Ndola
	ZMB	Kitwe
	ZMB	Kabwe
	ZMB	Chingola
	ZMB	Mufulira
	ZMB	Luanshya
	ZAF	Cape Town
	ZAF	Soweto
	ZAF	Johannesburg
	ZAF	Port Elizabeth

範例

- 先依照CountryCode欄位排序；CountryCode 一樣的，再依照Name欄位排序
- 先依照CountryCode欄位升冪排序；CountryCode 一樣的，再依照Name欄位降冪排序



The screenshot shows a SQL IDE window titled "SQL File 3* x". The query editor contains the following SQL statement:

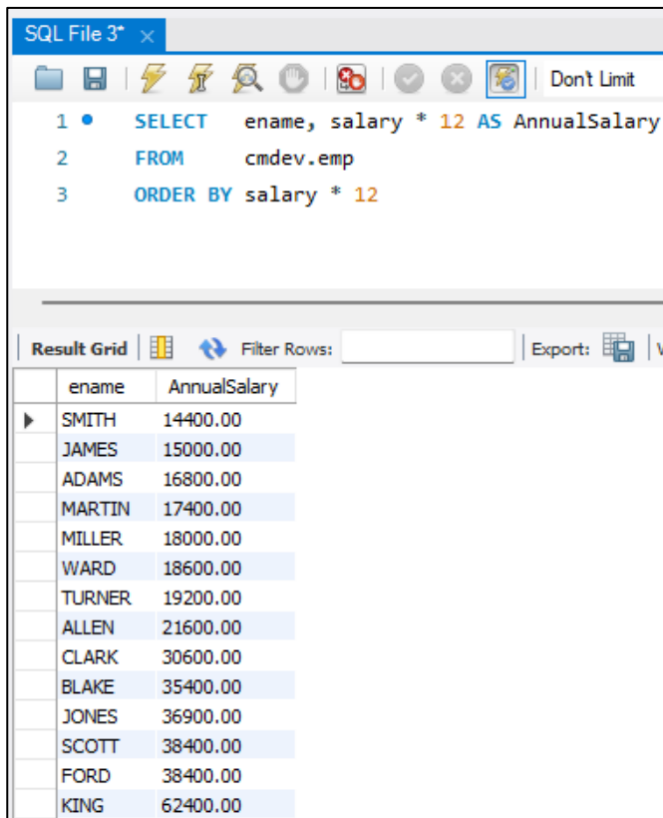
```
1 SELECT  CountryCode, Name
2 FROM    city
3 ORDER BY CountryCode
4
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are sorted by CountryCode, and for each country, the cities are listed in ascending order of Name. The data is as follows:

CountryCode	Name
ABW	Oranjestad
AFG	Kabul
AFG	Qandahar
AFG	Herat
AFG	Mazar-e-Sharif
AGO	Luanda
AGO	Huambo
AGO	Lobito
AGO	Benguela
AGO	Namibe

範例

- 使用別名或欄位索引修改排序條件



The screenshot shows a SQL query editor window titled "SQL File 3+ x". The query is as follows:

```
1 • SELECT   ename, salary * 12 AS AnnualSalary
2   FROM     cmdev.emp
3   ORDER BY salary * 12
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has two columns: "ename" and "AnnualSalary". The results are sorted by "AnnualSalary" in ascending order.

ename	AnnualSalary
SMITH	14400.00
JAMES	15000.00
ADAMS	16800.00
MARTIN	17400.00
MILLER	18000.00
WARD	18600.00
TURNER	19200.00
ALLEN	21600.00
CLARK	30600.00
BLAKE	35400.00
JONES	36900.00
SCOTT	38400.00
FORD	38400.00
KING	62400.00

限制查詢數量

- 查詢結果數量很多，限制取得指定筆數資料
- 語法：
 - LIMIT N(資料筆數)
 - 傳回前N筆資料
 - LIMIT M(開始列) N(資料筆數)
 - 由第M列資料開始，到(M+N-1)列資料
 - 共傳回N筆資料
- 搭配ORDER BY可查詢排名前幾名資料

■ 查詢1~5, 6~10 筆員工資料

■ 查詢1~5, 6~10 筆員工資料

■ 薪水最高及最低的三個員工

排除重複資料

- 查詢結果欄位中重複資料處理
- 語法：

SELECT [ALL | DISTINCT] 欄位...

- **ALL** 傳回所有資料不論是否重複
 - 預設，一般省略
- **DISTINCT** 排除重複的資料
 - 相同的資料只會傳回第一筆

範例

- 查詢 country 表格中 continent 欄位有哪些不同的值

SQL File 3* x

1 • SELECT Continent FROM world.country

2

Result Grid | Filter Rows:

Continent
North America
Asia
Africa
North America
Europe
Europe
North America
Asia
South America
Asia
Oceania
Antarctica
Antarctica
North America
Oceania
Europe

SQL File 3* x

1 • SELECT ALL Continent FROM world.country

2

Result Grid | Filter Rows: | Export:

Continent
North America
Asia
Africa
North America
Europe
Europe
North America
Asia
South America
Asia
Oceania
Antarctica
Antarctica
North America
Oceania
Europe