

關聯式資料庫管理系統 結構化查詢語言(SQL) 使用MySQL

資料查詢 - SELECT 敘述
簡單查詢(Simple Queries)

資料查詢 – SELECT 敘述

子句(Element)	運算式(Expression)	用途(Role)
SELECT	<select list>	給定查詢的資料項目 Defines which columns to return
FROM	<table source>	給定資料來源 Defines table(s) to query
WHERE	<search condition>	給定查詢/過濾資料條件 Filters rows using a predicate
GROUP BY	<group by list>	資料分組設定 Arranges rows by groups
HAVING	<search condition>	給定分組資料查詢/過濾條件 Filters groups using a predicate
ORDER BY	<order by list>	給定查詢結果排序方式 Sorts the output

撰寫SQL命令

- ▶ 指令不分大小寫
- ▶ 敘述可以編寫成多行
- ▶ 習慣上每個子句以一行表示
- ▶ 關鍵字(Keywords)不可以分割成多行
- ▶ 以分號(;)作為敘述的結束符號並執行該敘述
- ▶ 使用縮排來提升敘述的可讀性

SELECT - 查詢資料的指令

▶ 語法

```
SELECT * | {[DISTINCT] column | expression [alias], ...}  
FROM table;
```

▶ 子句

- SELECT 指定檢視的資料項
 - 欄位(Column name)、常值(Literal)、運算式(expression)、函數(Function)
- FROM 指定資料的來源
 - 資料表(Table)、虛擬資料表(Virtual Table)、視觀表(View)

▶ 敘述

- 由二個以上子句所組成
- 資料項間以逗號(,)來區隔
- 以分號作為結束的符號

```
SELECT ename, job, sal  
FROM emp;
```

選取資料表中的所有(欄位)資料

```
SELECT *  
FROM TableName;
```

	說明
SELECT	選取資料項(欄位)
*	資料表中所有的欄位
FROM	指定查詢的資料來源(資料表)
TableName	資料表名稱

```
mysql> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

選取指定欄位資料

```
SELECT column,...  
FROM   TableName;
```

	說明
SELECT	選取資料項(欄位)
ColumnName	資料表中欄位名稱
FROM	指定查詢的資料來源(資料表)
TableName	資料表名稱

```
SELECT empno, ename, sal  
FROM emp;
```

顯示資料表結構(Table Structure)

- ▶ 使用 DESCRIBE 命令查詢資料表的結構(欄位資訊)

```
DESC[RIBE] tablename
```

- ▶ 查看資料表 dept 的結構(欄位資訊)

```
DESCRIBE dept
```

Field	Type	Null	Key	Default	Extra
DEPTNO	int(11)	NO	PRI	NULL	
DNAME	varchar(14)	YES		NULL	
LOC	varchar(13)	YES		NULL	

MySQL資料表欄位常用的資料型態

資料類型	資料型態	說 明
文數字資料 (字元)	CHAR(n)	固定size, 不足部份以空白填入, 顯示時將空白截斷 長度為 n 個字元, n 的範圍為 1 到 255。
	VARCHAR(n)	可變長度的字元資料, 最大長度為 n 個, n 的範圍為 1 到 255。
數值資料	DECIMAL[(p[, s])] NUMERIC[(p[, s])]	數值資料, P 為精確位數, 總長度(p)=整數+小數(s)
	INT(n)	整數
日 期	DATETIME	日期時間

欄位表頭 (Column Heading)

```
mysql> select deptno, DNAME from dept;
```

deptno	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

▶ 欄位表頭顯示的方式

- Column 大寫 則以大寫表示
- column 小寫 則以小寫表示
- * 以 DESC 命令所見為準

算術運算式 (Arithmetic Expressions)

▶ 運算式

- 運算元(operand) 運算子(operator)
- $5 + 3$

▶ 運算式中的運算元(operand)資料型態要相同

▶ 算術運算子

- 加法 $+$
- 減法 $-$
- 乘法 $*$
- 除法 $/$

算術運算

▶ 數值資料運算

```
mysql> SELECT ename, sal, sal+3000  
-> FROM emp;
```

ename	sal	sal+3000
SMITH	800.00	3800.00
ALLEN	1600.00	4600.00
WARD	1250.00	4250.00
JONES	2975.00	5975.00
MARTIN	1250.00	4250.00
BLAKE	2850.00	5850.00
CLARK	2450.00	5450.00
SCOTT	3000.00	6000.00
KING	5000.00	8000.00
TURNER	1500.00	4500.00
ADAMS	1100.00	4100.00
JAMES	950.00	3950.00
FORD	3000.00	6000.00
MILLER	1300.00	4300.00

14 rows in set (0.00 sec)

運算子優先順序(Operator Precedence)

▶ 運算子優先順序

- $*$, $/$ 優先於 $+$, $-$
- $*$ 與 $/$ 同時出現時，由左至右
- $+$ 與 $-$ 同時出現時，由左至右

▶ 使用括號(Using Parentheses)

- 小刮弧
 - 改變運算順序
 - $3*4+2/3$
 - $3*(4+2)/3$
 - 意義更清楚
 - $3*4+2/3$
 - $(3*4)+(2/3)$

運算子優先順序(Operator Precedence)

► 使用括號改變優先順序

```
mysql> SELECT ename, sal, 12*sal+100  
-> FROM emp;
```

ename	sal	12*sal+100
SMITH	800.00	9700.00
ALLEN	1600.00	19300.00
WARD	1250.00	15100.00
JONES	2975.00	35800.00
MARTIN	1250.00	15100.00
BLAKE	2850.00	34300.00
CLARK	2450.00	29500.00
SCOTT	3000.00	36100.00
KING	5000.00	60100.00
TURNER	1500.00	18100.00
ADAMS	1100.00	13300.00
JAMES	950.00	11500.00
FORD	3000.00	36100.00
MILLER	1300.00	15700.00

14 rows in set (0.00 sec)

```
mysql> SELECT ename, sal, 12*(sal+100)  
-> FROM emp;
```

ename	sal	12*(sal+100)
SMITH	800.00	10800.00
ALLEN	1600.00	20400.00
WARD	1250.00	16200.00
JONES	2975.00	36900.00
MARTIN	1250.00	16200.00
BLAKE	2850.00	35400.00
CLARK	2450.00	30600.00
SCOTT	3000.00	37200.00
KING	5000.00	61200.00
TURNER	1500.00	19200.00
ADAMS	1100.00	14400.00
JAMES	950.00	12600.00
FORD	3000.00	37200.00
MILLER	1300.00	16800.00

14 rows in set (0.00 sec)

設定欄位別名

▶ 欄位別名(Column Alias)

- 一般欄位 (General Columns)
- 運算式欄位 (Expression Column)
- 中文及特殊字或空白使用 “ ” 刮住

Column | expression [AS] alias

```
mysql> SELECT empno “編號” , ename Name, sal AS SALARY, sal*12 "Ann Sal"  
-> FROM emp;
```

編號	Name	SALARY	Ann Sal
7839	KING	5000	60000
7698	BLAKE	2850	34200
7782	CLARK	2450	29400
7566	JONES	2975	35700
7654	MARTIN	1250	15000
7499	ALLEN	1600	19200
7844	TURNER	1500	18000
7900	JAMES	950	11400
7521	WARD	1250	15000
7902	FORD	3000	36000
7369	SMITH	800	9600
7788	SCOTT	3000	36000
7876	ADAMS	1100	13200
7934	MILLER	1300	15600

14 rows in set (0.01 sec)

空值(Null Value)

- ▶ NULL 值是在新增資料時沒有設定欄位值，資料庫管理系統便設定該欄位的內容為NULL
 - 無法使用
 - 無法指派
 - 無法運算
 - 與 0 和空白不相同
- ▶ 所有型態皆可以為空值
- ▶ 運算式中若含有null value時，其結果為null

空值-算術運算 (Null Value in Arithmetic Exprs.)

▶ 空值的運算

```
mysql> SELECT empno, ename, sal, comm, sal+comm  
-> FROM emp;
```

empno	ename	sal	comm	sal+comm
7369	SMITH	800.00	NULL	NULL
7499	ALLEN	1600.00	300.00	1900.00
7521	WARD	1250.00	500.00	1750.00
7566	JONES	2975.00	NULL	NULL
7654	MARTIN	1250.00	1400.00	2650.00
7698	BLAKE	2850.00	NULL	NULL
7782	CLARK	2450.00	NULL	NULL
7788	SCOTT	3000.00	NULL	NULL
7839	KING	5000.00	NULL	NULL
7844	TURNER	1500.00	0.00	1500.00
7876	ADAMS	1100.00	NULL	NULL
7900	JAMES	950.00	NULL	NULL
7902	FORD	3000.00	NULL	NULL
7934	MILLER	1300.00	NULL	NULL

Expression Column

14 rows in set (0.00 sec)

字串運算-字串連結(Concatenation)

- ▶ 函數：Concat(expr1, expr2, ...)
 - Expr 可以是任何資料型態

```
mysql> SELECT concat(deptno,dname) Department
        -> FROM dept;
+-----+
| Department |
+-----+
| 10ACCOUNTING |
| 20RESEARCH   |
| 30SALES      |
| 40OPERATIONS |
+-----+
4 rows in set (0.00 sec)
```

常值(Literal)

- ▶ 不變的資料值，不儲存於資料庫內部
- ▶ 類型分為
 - 數值常值:任何數值
 - 5 -24 3.65 -8
 - 字串常值:於一對單引號內所輸入的文字
 - 'book' '中央大學'
 - 日期常值:於一對單引號內輸入的日期資料(需符合日期格式)
 - '19951015'

```
mysql> SELECT concat(ename,' job is ', job) Jobs  
-> FROM emp;
```

```
+-----+  
| Jobs  
+-----+  
| SMITH job is CLERK  
| ALLEN job is SALESMAN  
| WARD job is SALESMAN  
| JONES job is MANAGER  
| MARTIN job is SALESMAN  
| .. ..  
| FORD job is ANALYST  
| MILLER job is CLERK  
+-----+
```

重複資料列與排除

► 重複資料列(Duplicate Rows)

```
SELECT DISTINCT <column list>  
FROM <table or view>
```

```
mysql> SELECT job FROM emp;
```

job
PRESIDENT
MANAGER
MANAGER
MANAGER
SALESMAN
SALESMAN
SALESMAN
CLERK
SALESMAN
ANALYST
CLERK
ANALYST
CLERK
CLERK

14 rows in set (0.00 sec)

```
mysql> SELECT DISTINCT job FROM emp;
```

job
PRESIDENT
MANAGER
SALESMAN
CLERK
ANALYST

5 rows in set (0.01 sec)

設定查詢結果傳回的資料筆數

- ▶ SELECT 命令可以使用 LIMIT 子句來設定查詢結果傳回的資料筆數

```
SELECT [DISTINCT] <column list>  
FROM <table or view>  
LIMIT offset, count;
```

- offset : 傳回資料的開始位置, 由 0 起算, ≥ 0
- count : 傳回最大的資料筆數
- LIMIT 後面只加一個整數, 代表要傳回幾筆資料
 - 例如: limit 5 代表從第一筆開始最多傳回5筆
- OFFSET 可以把它想成要略過筆數
 - 以offset=2為例, 可想成在找到的資料筆數中略過前二筆, 即是由第三筆開始回傳
- 使用LIMIT語句時通常會伴隨著「ORDER BY」

設定查詢結果傳回的資料筆數

▶ 範例

```
mysql> select ename, sal, job  
-> from emp;
```

ename	sal	job
ALLEN	2000.00	SALESMAN
JONES	2975.00	MANAGER
MARTIN	1250.00	SALESMAN
BLAKE	2850.00	MANAGER
CLARK	2450.00	MANAGER
SCOTT	3000.00	ANALYST
KING	5000.00	PRESIDENT
TURNER	1500.00	SALESMAN
ADAMS	1100.00	CLERK
MARY	3000.00	ANALYST
MILLER	1300.00	CLERK

11 rows in set (0.00 sec)

```
mysql> select ename, sal, job  
-> from emp  
-> limit 5;
```

ename	sal	job
ALLEN	2000.00	SALESMAN
JONES	2975.00	MANAGER
MARTIN	1250.00	SALESMAN
BLAKE	2850.00	MANAGER
CLARK	2450.00	MANAGER

5 rows in set (0.00 sec)

```
mysql> select ename, sal, job  
-> from emp  
-> limit 3,5;
```

ename	sal	job
BLAKE	2850.00	MANAGER
CLARK	2450.00	MANAGER
SCOTT	3000.00	ANALYST
KING	5000.00	PRESIDENT
TURNER	1500.00	SALESMAN

5 rows in set (0.00 sec)

作業練習

1. 建立一個查詢來顯示部門(dept)資料表中的所有資料。
2. 建立一個查詢來顯示每一位員工的姓名(name)、職稱(job)、進公司日期(hire date)及員工編號(employee number)，並將員工編號顯示在最前面。
3. 建立一個查詢來顯示所有員工所擔任的職稱有哪些？（重複的資料只顯示一次）
4. 建立一個查詢來顯示每一位員工的姓名(name)、職稱(job)、進公司日期(hire date)及員工編號(employee number)，並將員工編號顯示在最前面。並將資料表頭重新命名為：Emp#, Employee, Job, Hire Date。
5. 建立一個查詢將姓名(name)及職稱(job)串接為一個資料項（資料中間利用一個空白和一個逗號做區隔），並將表頭重新命名為 Employee and Title。