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 命令

設定查詢條件與資料排序

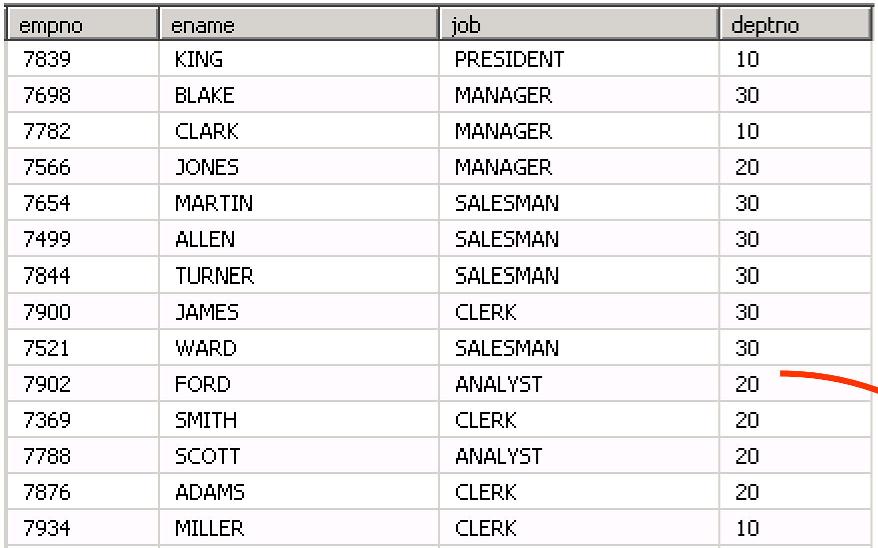
Filtering and Sorting Data

資料查詢 – SELECT 敘述

子句(Element)	Expression	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

條件查詢(限制查詢)

- 限制資料列 － 設定查詢條件
  - 依查詢條件篩選符合條件的資料列

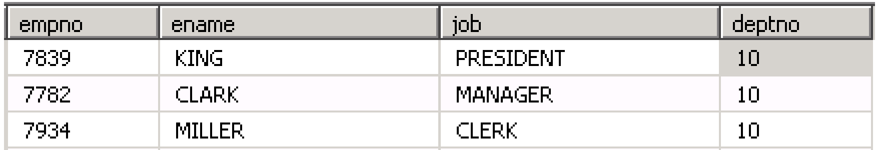


empno	ename	job	deptno
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7499	ALLEN	SALESMAN	30
7844	TURNER	SALESMAN	30
7900	JAMES	CLERK	30
7521	WARD	SALESMAN	30
7902	FORD	ANALYST	20
7369	SMITH	CLERK	20
7788	SCOTT	ANALYST	20
7876	ADAMS	CLERK	20
7934	MILLER	CLERK	10

```
Select empno, ename, job, deptno
From emp;
```

```
Select empno, ename, job, deptno
From emp
Where deptno=10;
```

只查詢部門代號為10的員工資料



empno	ename	job	deptno
7839	KING	PRESIDENT	10
7782	CLARK	MANAGER	10
7934	MILLER	CLERK	10

使用 WHERE 子句設定查詢條件

```
SELECT column,...
FROM table
WHERE conditions;
```

- 置於 FROM 子句後
- 條件子句(Conditions)
  - 邏輯值 ： 真(TRUE)、假(FALSE)、空值(NULL)
- 比較運算式/邏輯運算式/ SQL特定運算式
  - 運算元可以是欄位、運算式、函數、常數

```
mysql> SELECT empno,ename,job,deptno
    -> FROM emp
    -> WHERE deptno = 10;
+-------+--------+-----------+--------+
| empno | ename  | job       | deptno |
+-------+--------+-----------+--------+
|  7782 | CLARK  | MANAGER   |     10 |
|  7839 | KING   | PRESIDENT |     10 |
|  7934 | MILLER | CLERK     |     10 |
+-------+--------+-----------+--------+
3 rows in set (0.00 sec)
```

比較運算子(Comparison Operators)

- 比較運算子使用在二個資料項的比較大小
  - 數值資料 ： 值
  - 字串資料 ： 內碼(預設)
  - 日期時間資料 ： 世紀、年、月、日、時、分、秒

Operator	Meaning
=	等於 (Equal to)
>	大於 (Greater than)
>=	大於或等於 (Greater than or equal to)
<	小於 (Less than)
<=	小於或等於 (Less than or equal to)
<>	不等於 (Not equal to)

使用比較運算做條件查詢

- 數值資料-列出薪水大於等於3000的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE sal >= 3000;
+-------+-------+-----------+------+
| empno | ename | job       | sal  |
+-------+-------+-----------+------+
|  7839 | KING  | PRESIDENT | 5000 |
|  7902 | FORD  | ANALYST   | 3000 |
|  7788 | SCOTT | ANALYST   | 3000 |
+-------+-------+-----------+------+
```

- 日期資料- 1981-12-03 進公司的員工

```
mysql> SELECT empno,ename,job,deptno, hiredate
    -> FROM emp
    -> WHERE hiredate = '1981-12-03';
+-------+-------+---------+--------+---------------------+
| empno | ename | job     | deptno | hiredate            |
+-------+-------+---------+--------+---------------------+
|  7900 | JAMES | CLERK   |     30 | 1981-12-03 00:00:00 |
|  7902 | FORD  | ANALYST |     20 | 1981-12-03 00:00:00 |
+-------+-------+---------+--------+---------------------+
2 rows in set (0.00 sec)
```

使用比較運算做條件查詢

- 字串資料-列出 KING 的資料

```
mysql> SELECT empno,ename,job,deptno, hiredate, sal
    -> FROM emp
    -> WHERE ename = 'KING';
+-------+-------+-----------+--------+---------------------+---------+
| empno | ename | job       | deptno | hiredate            | sal     |
+-------+-------+-----------+--------+---------------------+---------+
|  7839 | KING  | PRESIDENT |     10 | 1981-11-17 00:00:00 | 5000.00 |
+-------+-------+-----------+--------+---------------------+---------+
```

- 相同資料型態的欄位資料

```
mysql> SELECT ename, sal, comm
    -> FROM   emp
    -> WHERE  sal<=comm;
+--------+---------+---------+
| ename  | sal     | comm    |
+--------+---------+---------+
| MARTIN | 1250.00 | 1400.00 |
+--------+---------+---------+
1 row in set (0.00 sec)
```

邏輯運算子(Logical Operators)

- 若有一個以上的條件運算必須使用邏輯運算子結合成一個運算結果

邏輯運算子	運算結果
AND	Returns TRUE if *both* component conditions are TRUE
OR	Returns TRUE if *either* component condition is TRUE
NOT	Returns TRUE if the following condition is FALSE

使用邏輯運算子 – AND

- AND 必須二個運算元都是真(TRUE)才會符合查詢條件

```
mysql> SELECT empno, ename, job, sal
    -> FROM   emp
    -> WHERE  sal>=1100 AND job='CLERK';
+-------+--------+-------+---------+
| empno | ename  | job   | sal     |
+-------+--------+-------+---------+
|  7876 | ADAMS  | CLERK | 1100.00 |
|  7934 | MILLER | CLERK | 1300.00 |
+-------+--------+-------+---------+
2 rows in set (0.00 sec)
```

使用邏輯運算子 - AND

- 列出薪水>2000且職務為manager的員工

```
mysql> SELECT empno, ename, job, sal, mgr
    -> FROM emp
    -> WHERE sal > 2000 AND job = 'MANAGER';
+-------+-------+---------+---------+------+
| empno | ename | job     | sal     | mgr  |
+-------+-------+---------+---------+------+
|  7566 | JONES | MANAGER | 2975.00 | 7839 |
|  7698 | BLAKE | MANAGER | 2850.00 | 7839 |
|  7782 | CLARK | MANAGER | 2450.00 | 7839 |
+-------+-------+---------+---------+------+
3 rows in set (0.00 sec)
```

使用邏輯運算子 - OR

- OR 只要任一個運算元是真(TRUE)就符合查詢條件

```
mysql> SELECT empno, ename, job, sal
    -> FROM   emp
    -> WHERE  sal>=1100 OR job='CLERK';
+-------+--------+-----------+---------+
| empno | ename  | job       | sal     |
+-------+--------+-----------+---------+
|  7369 | SMITH  | CLERK     |  800.00 |
|  7499 | ALLEN  | SALESMAN  | 1600.00 |
|  7521 | WARD   | SALESMAN  | 1250.00 |
|  7566 | JONES  | MANAGER   | 2975.00 |
|  7654 | MARTIN | SALESMAN  | 1250.00 |
|  7698 | BLAKE  | MANAGER   | 2850.00 |
|  7782 | CLARK  | MANAGER   | 2450.00 |
|  7788 | SCOTT  | ANALYST   | 3000.00 |
|  7839 | KING   | PRESIDENT | 5000.00 |
|  7844 | TURNER | SALESMAN  | 1500.00 |
|  7876 | ADAMS  | CLERK     | 1100.00 |
|  7900 | JAMES  | CLERK     |  950.00 |
|  7902 | FORD   | ANALYST   | 3000.00 |
|  7934 | MILLER | CLERK     | 1300.00 |
+-------+--------+-----------+---------+
14 rows in set (0.00 sec)
```

使用邏輯運算子 - OR

- 列出薪水>2000或職務為manager的員工

```
mysql> SELECT empno, ename, job, sal, mgr
    -> FROM emp
    -> WHERE sal > 2000 OR job = 'MANAGER';
+-------+-------+-----------+---------+------+
| empno | ename | job       | sal     | mgr  |
+-------+-------+-----------+---------+------+
|  7566 | JONES | MANAGER   | 2975.00 | 7839 |
|  7698 | BLAKE | MANAGER   | 2850.00 | 7839 |
|  7782 | CLARK | MANAGER   | 2450.00 | 7839 |
|  7788 | SCOTT | ANALYST   | 3000.00 | 7566 |
|  7839 | KING  | PRESIDENT | 5000.00 | NULL |
|  7902 | FORD  | ANALYST   | 3000.00 | 7566 |
+-------+-------+-----------+---------+------+
6 rows in set (0.00 sec)
```

使用邏輯運算子 - NOT

- NOT 反向運算, 只有一個運算元

```
mysql> SELECT empno, ename, job, sal, mgr
    -> FROM emp
    -> WHERE NOT(sal > 2000 OR job = 'MANAGER');
+-------+--------+----------+---------+------+
| empno | ename  | job      | sal     | mgr  |
+-------+--------+----------+---------+------+
|  7369 | SMITH  | CLERK    |  800.00 | 7902 |
|  7499 | ALLEN  | SALESMAN | 1600.00 | 7698 |
|  7521 | WARD   | SALESMAN | 1250.00 | 7698 |
|  7654 | MARTIN | SALESMAN | 1250.00 | 7698 |
|  7844 | TURNER | SALESMAN | 1500.00 | 7698 |
|  7876 | ADAMS  | CLERK    | 1100.00 | 7788 |
|  7900 | JAMES  | CLERK    |  950.00 | 7698 |
|  7934 | MILLER | CLERK    | 1300.00 | 7782 |
+-------+--------+----------+---------+------+
8 rows in set (0.00 sec)
```

SQL特定運算子

- BETWEEN 運算子：一個連續區間值的查詢
- IN 運算子：列舉值的查詢
- LIKE 運算子：萬用字元查詢
- IS NULL 運算子：空值(NULL)的查詢

Operator	Meaning
BETWEEN <low_val> AND <hi_val>	Between two values (inclusive)
IN (list)	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null value

BETWEEN AND 運算子

- 連續區間的條件判斷

```
expr BETWEEN x1 AND x2
```

  - expr 介於x1及x2之間(包含x1 & x2)
  - x1 必需小於 x2
- 可使用在數值、日期及字元資料
  - WHERE sal BETWEEN 1000 AND 3000
  - WHERE hiredate BETWEEN ’1981-01-01’ AND ’1981-06-30’
  - WHERE ename BETWEEN ’A’ AND ’E’

使用BETWEEN AND 查詢

- 查詢薪水介於 2000 到 3500 之間的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE sal BETWEEN 2000 AND 3500;
+-------+-------+---------+---------+
| empno | ename | job     | sal     |
+-------+-------+---------+---------+
|  7566 | JONES | MANAGER | 2975.00 |
|  7698 | BLAKE | MANAGER | 2850.00 |
|  7782 | CLARK | MANAGER | 2450.00 |
|  7788 | SCOTT | ANALYST | 3000.00 |
|  7902 | FORD  | ANALYST | 3000.00 |
+-------+-------+---------+---------+
5 rows in set (0.00 sec)
```

IN 運算子

- 列舉值的查詢

```
expr IN(value1, value2,...)
```

  - value 所成的串列
  - expr 和 value list 的資料型態要相同
- 可使用在數值、日期及字元資料
  - WHERE sal IN (100, 200, 300)
  - WHERE code IN (’A’,’B’,’C’)
  - WHERE hiredate IN (’1981-05-01’,’1981-10-03’)

使用IN查詢

- 列出職務為 salesman 或 manager 的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE job IN ('SALESMAN','MANAGER');
+-------+--------+----------+---------+
| empno | ename  | job      | sal     |
+-------+--------+----------+---------+
|  7499 | ALLEN  | SALESMAN | 1600.00 |
|  7521 | WARD   | SALESMAN | 1250.00 |
|  7566 | JONES  | MANAGER  | 2975.00 |
|  7654 | MARTIN | SALESMAN | 1250.00 |
|  7698 | BLAKE  | MANAGER  | 2850.00 |
|  7782 | CLARK  | MANAGER  | 2450.00 |
|  7844 | TURNER | SALESMAN | 1500.00 |
+-------+--------+----------+---------+
7 rows in set (0.00 sec)
```

LIKE 運算子

- 模糊比對-萬用字元查詢

```
expr LIKE 'pattern' ESCAPE 'character'
```

- pattern

%	百分比	任意字元
_	底線	一個字元

- ESCAPE ’character’

定義跳脫字元   Example : ESCAPE ’&’

- &%   ’%’
- &_   ’_’

使用LIKE查詢

- 列出姓名以A開頭的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE ename LIKE 'A%';
+-------+-------+----------+---------+
| empno | ename | job      | sal     |
+-------+-------+----------+---------+
|  7499 | ALLEN | SALESMAN | 1600.00 |
|  7876 | ADAMS | CLERK    | 1100.00 |
+-------+-------+----------+---------+
2 rows in set (0.00 sec)
```

使用LIKE查詢

- 列出姓名以N結尾的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE ename LIKE '%N';
+-------+--------+----------+---------+
| empno | ename  | job      | sal     |
+-------+--------+----------+---------+
|  7499 | ALLEN  | SALESMAN | 1600.00 |
|  7654 | MARTIN | SALESMAN | 1250.00 |
+-------+--------+----------+---------+
2 rows in set (0.00 sec)
```

使用LIKE查詢

- 列出姓名中含有T的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE ename LIKE '%T%';
+-------+--------+----------+---------+
| empno | ename  | job      | sal     |
+-------+--------+----------+---------+
|  7369 | SMITH  | CLERK    |  800.00 |
|  7654 | MARTIN | SALESMAN | 1250.00 |
|  7788 | SCOTT  | ANALYST  | 3000.00 |
|  7844 | TURNER | SALESMAN | 1500.00 |
+-------+--------+----------+---------+
4 rows in set (0.00 sec)
```

使用LIKE查詢

- 列出姓名第二個字為A的員工

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE ename LIKE '_A%';
+-------+--------+----------+---------+
| empno | ename  | job      | sal     |
+-------+--------+----------+---------+
|  7521 | WARD   | SALESMAN | 1250.00 |
|  7654 | MARTIN | SALESMAN | 1250.00 |
|  7900 | JAMES  | CLERK    |  950.00 |
+-------+--------+----------+---------+
3 rows in set (0.00 sec)
```

使用LIKE ESCAPE查詢

- 搜尋濃度3%的溶濟

```
mysql> SELECT *
    -> FROM liquorS
    -> WHERE content LIKE '%3&%%' ESCAPE '&';
```

```
mysql> SELECT *
    -> FROM liquorS
    -> WHERE content LIKE '%3\%%' ESCAPE ‘\';
```

IS NULL 運算子

- 空值運算子(判斷資料是否為NULL)

```
expr IS NULL
```

  - NULL 專用的運算子
  - (NULL = NULL) → NULL
  - NULL 與空白和0不相同
  - 任何型態欄位皆可以為 NULL value

使用IS NULL查詢

- 找出公司老闆的資料(mgr 是 NULL的員工)

```
mysql> SELECT empno, ename, job, sal, mgr
    -> FROM emp
    -> WHERE mgr IS NULL;
+-------+-------+-----------+---------+------+
| empno | ename | job       | sal     | mgr  |
+-------+-------+-----------+---------+------+
|  7839 | KING  | PRESIDENT | 5000.00 | NULL |
+-------+-------+-----------+---------+------+
1 row in set (0.00 sec)
```

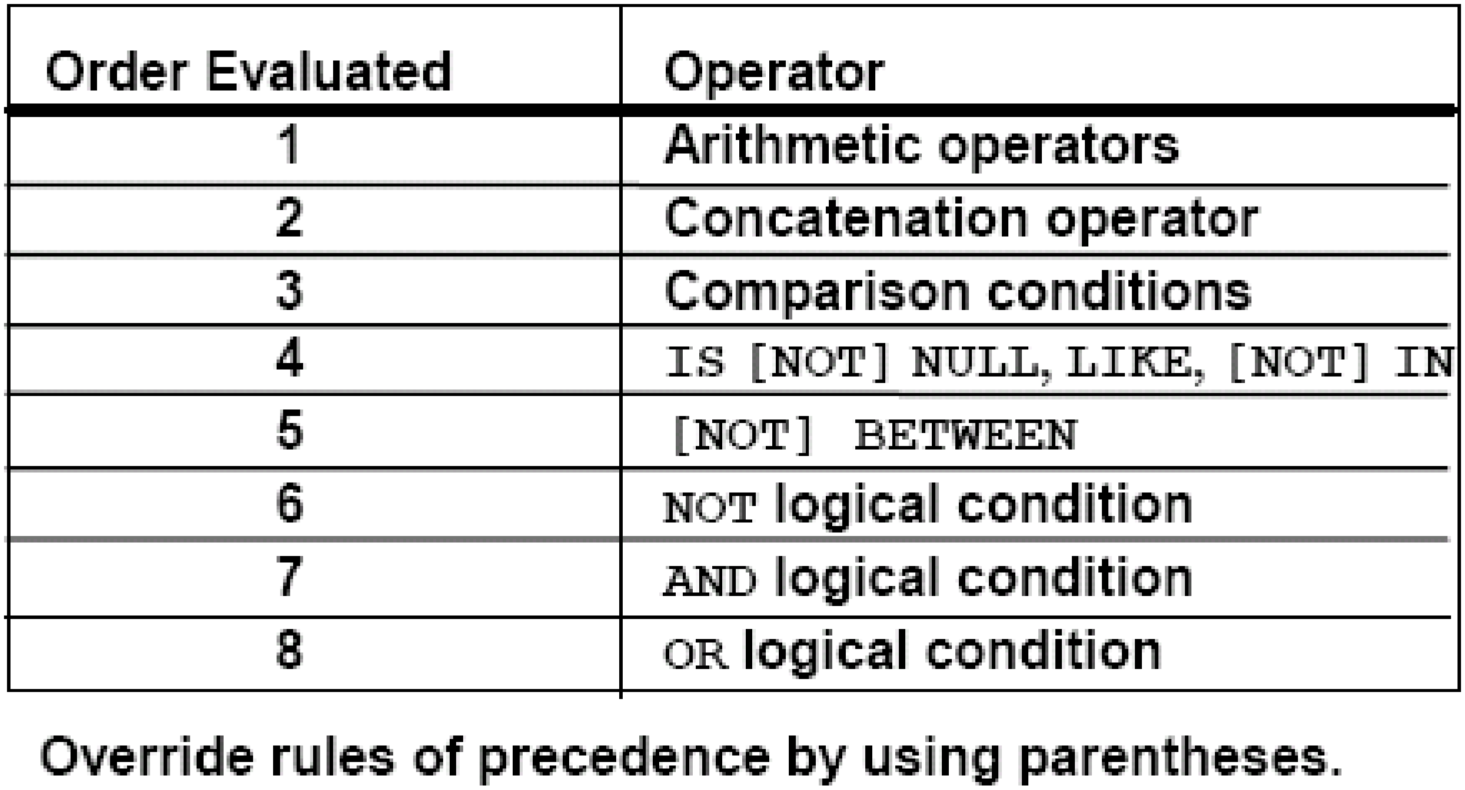
使用NOT查詢

- 列出除了manager & salesman以外的員工

```
mysql> SELECT empno, ename, job, sal, deptno
    -> FROM emp
    -> WHERE job NOT IN ('SALESMAN','MANAGER');
+-------+--------+-----------+---------+--------+
| empno | ename  | job       | sal     | deptno |
+-------+--------+-----------+---------+--------+
|  7369 | SMITH  | CLERK     |  800.00 |     20 |
|  7788 | SCOTT  | ANALYST   | 3000.00 |     20 |
|  7839 | KING   | PRESIDENT | 5000.00 |     10 |
|  7876 | ADAMS  | CLERK     | 1100.00 |     20 |
|  7900 | JAMES  | CLERK     |  950.00 |     30 |
|  7902 | FORD   | ANALYST   | 3000.00 |     20 |
|  7934 | MILLER | CLERK     | 1300.00 |     10 |
+-------+--------+-----------+---------+--------+
7 rows in set (0.00 sec)
```

NOT BETWEEN .. AND ..
NOT IN
NOT LIKE
IS NOT NULL

運算子執行優先順序



Order Evaluated	Operator
1	Arithmetic operators
2	Concatenation operator
3	Comparison conditions
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	NOT logical condition
7	AND logical condition
8	OR logical condition

Override rules of precedence by using parentheses.

- 可使用小括號改變執行運算順序

運算子執行優先順序

- AND ➔ OR

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE job = 'MANAGER'
    ->     OR job = 'SALESMAN'
    ->     AND sal < 1500;
+-------+--------+----------+------+
| empno | ename  | job      | sal  |
+-------+--------+----------+------+
|  7698 | BLAKE  | MANAGER  | 2850 |
|  7782 | CLARK  | MANAGER  | 2450 |
|  7566 | JONES  | MANAGER  | 2975 |
|  7654 | MARTIN | SALESMAN | 1250 |
|  7521 | WARD   | SALESMAN | 1250 |
+-------+--------+----------+------+
5 rows in set (0.00 sec)
```

1.AND

2.OR

運算子執行優先順序

- 利用小括號改變運算順序

```
mysql> SELECT empno, ename, job, sal
    -> FROM emp
    -> WHERE(job = 'MANAGER'
    ->    OR job = 'SALESMAN') 1.OR
    ->    AND sal < 1500;       2.AND
+-------+--------+----------+------+
| empno | ename  | job      | sal  |
+-------+--------+----------+------+
|  7654 | MARTIN | SALESMAN | 1250 |
|  7521 | WARD   | SALESMAN | 1250 |
+-------+--------+----------+------+
2 rows in set (0.00 sec)
```

CASE 運算式

- 列舉式 CASE (Simple CASE)
  - 與一串列舉的值做比較
  - 傳回第一個(值相等)的回傳值
  - 若都不相等則傳回 ELSE 的回傳值,若無給定 ELSE 則傳回 NULL 值
- 條件式CASE (Searched CASE)
  - 與一串列舉的條件做比較
  - 傳回第一個(條件運算結果=真)的回傳值
  - 若都不符合則傳回 ELSE 的回傳值,若無給定 ELSE 則傳回 NULL 值

列舉式 CASE (Simple CASE)

```
CASE expr
  WHEN v1 THEN r1
  [WHEN v2 THEN r2]
  ...
  [ELSE r]
END
```

```
mysql> SELECT empno, ename,sal, job,
    ->         CASE job
    ->           WHEN 'PRESIDENT' THEN sal*1.5
    ->           WHEN 'MANAGER' THEN sal*1.3
    ->           WHEN 'ANALYST' THEN sal*1.2
    ->           ELSE sal
    ->         END sal
    ->    FROM emp;
+-------+--------+------+-----------+--------+
| empno | ename  | sal  | job       | sal    |
+-------+--------+------+-----------+--------+
|  7839 | KING   | 5000 | PRESIDENT | 7500.0 |
|  7698 | BLAKE  | 2850 | MANAGER   | 3705.0 |
|  7782 | CLARK  | 2450 | MANAGER   | 3185.0 |
|  7566 | JONES  | 2975 | MANAGER   | 3867.5 |
|  7654 | MARTIN | 1250 | SALESMAN  | 1250.0 |
|  7499 | ALLEN  | 1600 | SALESMAN  | 1600.0 |
|  7844 | TURNER | 1500 | SALESMAN  | 1500.0 |
|  7900 | JAMES  |  950 | CLERK     |  950.0 |
|  7521 | WARD   | 1250 | SALESMAN  | 1250.0 |
|  7902 | FORD   | 3000 | ANALYST   | 3600.0 |
|  7369 | SMITH  |  800 | CLERK     |  800.0 |
|  7788 | SCOTT  | 3000 | ANALYST   | 3600.0 |
|  7876 | ADAMS  | 1100 | CLERK     | 1100.0 |
|  7934 | MILLER | 1300 | CLERK     | 1300.0 |
+-------+--------+------+-----------+--------+
14 rows in set (0.00 sec)
```

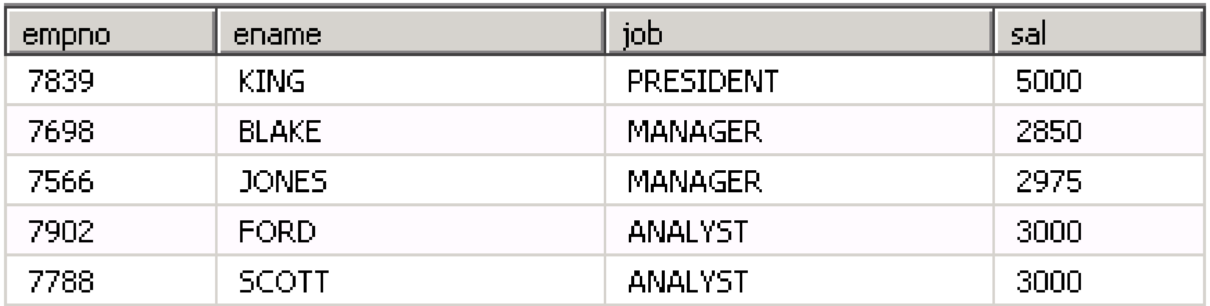
條件式CASE (Searched CASE)

```
CASE
   WHEN condition1 THEN r1
  [WHEN condition2 THEN r2]
  ...
  [ELSE r]
END
```

```
mysql> SELECT empno, ename, sal,
    ->        CASE
    ->          WHEN sal BETWEEN 0    AND 1000 THEN 'A'
    ->          WHEN sal BETWEEN 1001 AND 2000 THEN 'B'
    ->          WHEN sal BETWEEN 2001 AND 3000 THEN 'C'
    ->          WHEN sal BETWEEN 3001 AND 4000 THEN 'D'
    ->          ELSE 'E'
    ->        END sal
    ->   FROM emp;
+-------+--------+------+------+
| empno | ename  | sal  | sal  |
+-------+--------+------+------+
|  7839 | KING   | 5000 | E    |
|  7698 | BLAKE  | 2850 | C    |
|  7782 | CLARK  | 2450 | C    |
|  7566 | JONES  | 2975 | C    |
|  7654 | MARTIN | 1250 | B    |
|  7499 | ALLEN  | 1600 | B    |
|  7844 | TURNER | 1500 | B    |
|  7900 | JAMES  |  950 | A    |
|  7521 | WARD   | 1250 | B    |
|  7902 | FORD   | 3000 | C    |
|  7369 | SMITH  |  800 | A    |
|  7788 | SCOTT  | 3000 | C    |
|  7876 | ADAMS  | 1100 | B    |
|  7934 | MILLER | 1300 | B    |
+-------+--------+------+------+
14 rows in set (0.00 sec)
```

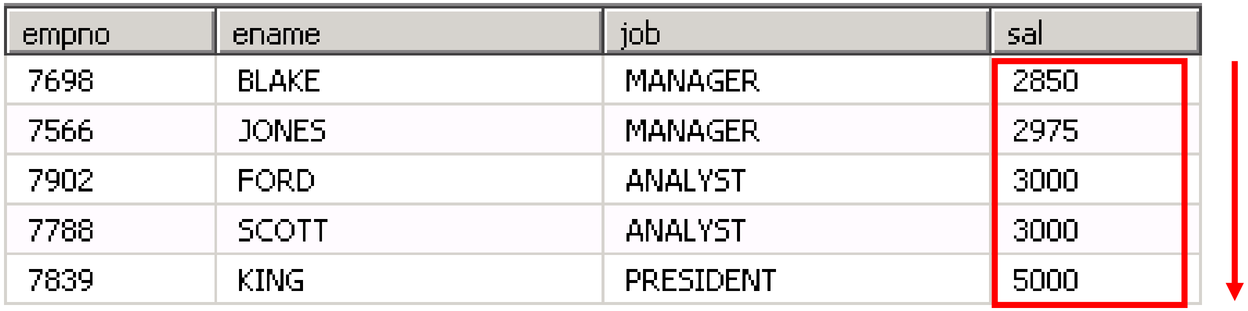
資料列排序

- 資料未排序前



empno	ename	job	sal
7839	KING	PRESIDENT	5000
7698	BLAKE	MANAGER	2850
7566	JONES	MANAGER	2975
7902	FORD	ANALYST	3000
7788	SCOTT	ANALYST	3000

- 資料依薪資由低至高排列



empno	ename	job	sal
7698	BLAKE	MANAGER	2850
7566	JONES	MANAGER	2975
7902	FORD	ANALYST	3000
7788	SCOTT	ANALYST	3000
7839	KING	PRESIDENT	5000

ORDER BY 子句

- 使用 ORDER BY 子句來做資料排序

```
SELECT column,...
FROM table
[WHERE conditions]
ORDER BY {column|alias|expression|position [ASC|DESC],...};
```

  - 放置於SELECT敘述的最後一行
- 依指定欄位或運算式的資料值來排序
- 排序方式
  - ASC 升冪 Ascending 由小到大 [預設]
  - DESC 降冪 Descending 由大到小
- 若有空值時,升冪在最前面,降冪在最下面
- 排序資料
  - 欄位/別名/運算式/位置

使用欄位值排序 － 由小至大

- 依 select list 中的欄位

```
mysql> SELECT empno, ename, sal
    -> FROM emp
    -> WHERE deptno = 10
    -> ORDER BY sal;
+-------+--------+------+
| empno | ename  | sal  |
+-------+--------+------+
|  7934 | MILLER | 1300 |
|  7782 | CLARK  | 2450 |
|  7839 | KING   | 5000 |
+-------+--------+------+
3 rows in set (0.01 sec)
```

使用欄位值排序 － 由大至小

- 降冪 Descending

```
mysql> SELECT empno, ename, sal
    -> FROM emp
    -> WHERE deptno = 10
    -> ORDER BY sal DESC;
+-------+--------+------+
| empno | ename  | sal  |
+-------+--------+------+
|  7839 | KING   | 5000 |
|  7782 | CLARK  | 2450 |
|  7934 | MILLER | 1300 |
+-------+--------+------+
3 rows in set (0.00 sec)
```

使用欄位值排序 － 由小至大

- 不在 select list 的欄位

```
mysql> SELECT empno, ename, sal
    -> FROM emp
    -> WHERE deptno = 10
    -> ORDER BY hiredate;
+-------+--------+------+
| empno | ename  | sal  |
+-------+--------+------+
|  7782 | CLARK  | 2450 |
|  7839 | KING   | 5000 |
|  7934 | MILLER | 1300 |
+-------+--------+------+
3 rows in set (0.00 sec)
```

使用欄位別名排序

- 依欄位別名排序

```
mysql> SELECT empno, ename, sal*12 annsal
    -> FROM emp
    -> WHERE deptno = 10
    -> ORDER BY annsal;
+-------+--------+--------+
| empno | ename  | annsal |
+-------+--------+--------+
|  7934 | MILLER |  15600 |
|  7782 | CLARK  |  29400 |
|  7839 | KING   |  60000 |
+-------+--------+--------+
3 rows in set (0.00 sec)
```

使用運算式排序

- 依運算式排序

```
mysql> SELECT empno, ename, sal+comm bonus
    -> FROM emp
    -> WHERE deptno = 30
    -> ORDER BY sal+comm;
+-------+--------+-------+
| empno | ename  | bonus |
+-------+--------+-------+
|  7698 | BLAKE  |  NULL |
|  7900 | JAMES  |  NULL |
|  7844 | TURNER |  1500 |
|  7521 | WARD   |  1750 |
|  7499 | ALLEN  |  1900 |
|  7654 | MARTIN |  2650 |
+-------+--------+-------+
6 rows in set (0.01 sec)
```

使用資料項位置排序

- 依資料項在list中的位置順序

```
mysql> SELECT empno, ename, sal
    -> FROM emp
    -> WHERE deptno = 10
    -> ORDER BY 3;
+-------+--------+------+
| empno | ename  | sal  |
+-------+--------+------+
|  7934 | MILLER | 1300 |
|  7782 | CLARK  | 2450 |
|  7839 | KING   | 5000 |
+-------+--------+------+
3 rows in set (0.00 sec)
```

註：不可以為 0 或 大於 column list 的總欄位數

使用資料項位置排序

- 依欄位在list中的位置順序

```
mysql> SELECT *
    -> FROM emp
    -> WHERE deptno = 20
    -> ORDER BY 3;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7902 | FORD  | ANALYST | 7566 | 1981-10-03 | 3000 | NULL |     20 |
|  7788 | SCOTT | ANALYST | 7566 | 1982-10-09 | 3000 | NULL |     20 |
|  7369 | SMITH | CLERK   | 7902 | 1980-10-17 |  800 | NULL |     20 |
|  7876 | ADAMS | CLERK   | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
5 rows in set (0.01 sec)
```

使用資料項位置排序

- 超出欄位的總數

```
mysql> SELECT *
    ->   FROM emp
    ->  WHERE deptno = 20
    ->  ORDER BY 9;
ERROR 1054 (42S22): Unknown column '9' in 'order clause'
```

```
mysql> SELECT empno, ename, sal
    ->   FROM emp
    ->  WHERE deptno = 10
    ->  ORDER BY 4;
ERROR 1054 (42S22): Unknown column '4' in 'order clause'
```

使用多個資料項排序

- 多個欄位的排序
  - 每一個欄位皆可指定降冪或升冪

```
mysql> SELECT *
    -> FROM emp
    -> ORDER BY deptno, job, 6 DESC, 1;
+-------+--------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-----------+------+------------+------+------+--------+
|  7934 | MILLER | CLERK     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-18 | 5000 | NULL |     10 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-10-09 | 3000 | NULL |     20 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-10-03 | 3000 | NULL |     20 |
|  7876 | ADAMS  | CLERK     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7369 | SMITH  | CLERK     | 7902 | 1980-10-17 |  800 | NULL |     20 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-10-03 |  950 | NULL |     30 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7499 | ALLEN  | SALESMAN  | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7844 | TURNER | SALESMAN  | 7698 | 1981-09-08 | 1500 |    0 |     30 |
|  7521 | WARD   | SALESMAN  | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7654 | MARTIN | SALESMAN  | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
+-------+--------+-----------+------+------------+------+------+--------+
14 rows in set (0.00 sec)
```

資料排行榜

- ORDER BY 加 LIMIT 子句

```
mysql> select ename, sal, job
    -> from emp
    -> order by sal desc;
+--------+---------+-----------+
| ename  | sal     | job       |
+--------+---------+-----------+
| KING   | 5000.00 | PRESIDENT |
| SCOTT  | 3000.00 | ANALYST   |
| MARY   | 3000.00 | ANALYST   |
| JONES  | 2975.00 | MANAGER   |
| BLAKE  | 2850.00 | MANAGER   |
| CLARK  | 2450.00 | MANAGER   |
| ALLEN  | 2000.00 | SALESMAN  |
| TURNER | 1500.00 | SALESMAN  |
| MILLER | 1300.00 | CLERK     |
| MARTIN | 1250.00 | SALESMAN  |
| ADAMS  | 1100.00 | CLERK     |
+--------+---------+-----------+
11 rows in set (0.00 sec)
```

薪資最高前5名

```
mysql> select ename, sal, job
    -> from emp
    -> order by sal desc
    -> limit 5;
+-------+---------+-----------+
| ename | sal     | job       |
+-------+---------+-----------+
| KING  | 5000.00 | PRESIDENT |
| MARY  | 3000.00 | ANALYST   |
| SCOTT | 3000.00 | ANALYST   |
| JONES | 2975.00 | MANAGER   |
| BLAKE | 2850.00 | MANAGER   |
+-------+---------+-----------+
5 rows in set (0.00 sec)
```

薪資最低前5名

```
mysql> select ename, sal, job
    -> from emp
    -> order by sal asc
    -> limit 5;
+--------+---------+----------+
| ename  | sal     | job      |
+--------+---------+----------+
| ADAMS  | 1100.00 | CLERK    |
| MARTIN | 1250.00 | SALESMAN |
| MILLER | 1300.00 | CLERK    |
| TURNER | 1500.00 | SALESMAN |
| ALLEN  | 2000.00 | SALESMAN |
+--------+---------+----------+
5 rows in set (0.00 sec)
```

作業練習

建立查詢指令以顯示下列各題描述之資料:

1.顯示出所有員工薪資超過2850元的員工之姓名和薪資。

2.顯示員工編號為7566員工的姓名和他所屬的部門編號。

3.顯示薪資不介於1500~2850元之間的所有員工之姓名和薪資。

4.顯示於1981年2月20日和1981年5月1日間進入公司的員工之姓名,職稱和進公司日期，並依進公司日期由小到大排序。

5.顯示部門10和30所有員工之姓名和他所屬的部門編號，並依名字依英文字母順序排序。

6.顯示薪資超過1500 “且” 在10 “或” 30部門工作員工之姓名和薪資，把分別把表頭命名為Employee和 Monthly Salary。

7.顯示於1982年進公司的所有員工之姓名,職稱和進公司日期。

8.顯示沒有主管的員工之姓名和職稱。

9.顯示所有有賺取佣金的員工之姓名,薪資和佣金，並以薪資和佣金作降冪排列。

10.顯示所有名字裡第三個英文字母為A的員工之姓名與職稱。

11.顯示名字裡有兩個 “L” 且在30部門工作或經理是7782的員工之姓名,經理員工編號及所屬的部門編號。

12.顯示職稱為Clerk或Analyst且薪水不等於1000,3000,5000的員工之姓名,職稱和薪資。

13.顯示佣金比薪水的1.1倍還多的員工之姓名,薪資和佣金。