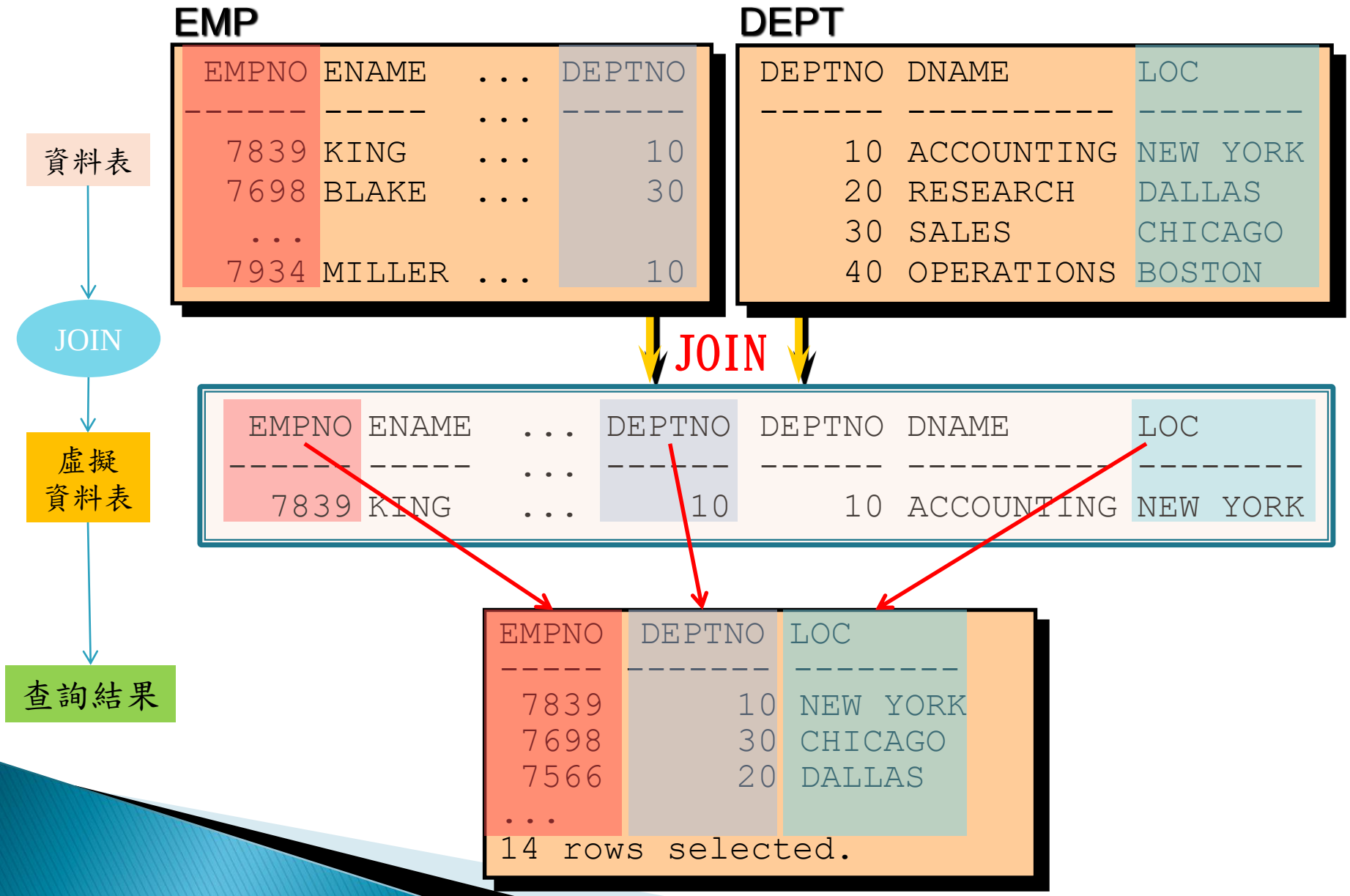
關聯式資料庫管理系統
結構化查詢語言(SQL)
使用MySQL

資料查詢 - SELECT 命令
多資料表查詢
(Join)

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

從多個資料表查詢資料(Obtaining Data from Multiple Tables)



連結多個表格的運算

- 連結查詢-用來查詢多個資料表中的資料
  - 將多個資料表中的資料利用連結運算(Join)結合成一個虛擬表格(Virtual Table)
- 向量積運算(Cartesian Product)
  - 合併二個表格中的所有資料

Name
Davis
Funk
King

×

Product
Alice Mutton
Crab Meat
Ipoh Coffee

=

Name	Product
Davis	Alice Mutton
Davis	Crab Meat
Davis	Ipoh Coffee
Funk	Alice Mutton
Funk	Crab Meat
Funk	Ipoh Coffee
King	Alice Mutton
King	Crab Meat
King	Ipoh Coffee

X 個欄位 M筆記錄 × Y 個欄位 N筆記錄 = X+Y 個欄位 M*N筆記錄

連結運算方式(Join Types)

- 交叉連結 Cross Join
- 內部連結 Inner Join（又稱一般連結）
  - 自然連結　　Natural Join
  - 相等連結　　Equal Joins
  - 不相等連結　　Non-Equal Joins
- 外部連結 Outer Join
  - 左邊外部連結　Left Outer Joins
  - 右邊外部連結　Right Outer Joins
  - 完全外部連結　Full Outer Joins(5.1才支援)
- 自我連結 Self Joins

交叉連結(Cross Join)

- 無條件連結
  - 第一個資料表所有的rows會與第二個資料表中每一個row合併(Cartesian Product)
- ANSI SQL-92 語法:

```
SELECT  ...
FROM table1 CROSS JOIN table2
...
```

- ANSI SQL-89 語法:

```
SELECT  ...
FROM table1, table2
...
```

交叉連結(Cross Join)

- 使用二個表格的資料來產生測試用資料

```
SELECT *
FROM emp e1 CROSS JOIN dept e2;
```

```
SELECT e1.ename, e2.job
FROM emp e1 CROSS JOIN emp e2;
```

不明確的欄位名稱(Ambiguous Column Names)

- 若一個以上的表格中有相同的欄位名稱
  - 附加表格名稱來分辨相同的欄位名稱

```
Table_prefixes.Column_Name
```

- 附加表格名稱可提升執行效能

```
SELECT ename, sal, deptno, dname
FROM dept JOIN emp ON dept.deptno = emp.deptno;
```

```
SELECT emp.ename, emp.sal, emp.deptno, dept.dname
FROM dept JOIN emp ON dept.deptno = emp.deptno;
```

- 表格別名(table aliase)

```
SELECT e.ename, e.sal, e.deptno, d.dname
FROM dept d JOIN emp e ON d.deptno = e.deptno;
```

內部連結(Inner Join)

- 產生只有符合條件連結(join conditions)的資料
  - 第一個資料表所有的rows會與第二個資料表中每一個row做連結條件測試,結果為真(TRUE)才會合併產生一筆新紀錄(row)
  - 如何設定連結條件
    - SQL-92 語法使用 ON 子句 (preferred)
    - SQL-89 語法使用 WHERE 子句
- 為何使用 ON 子句來設定連結條件?
  - 可以跟資料查詢條件分開來以免造成混淆

自然連結(Natural Join)

- 連結條件(Join Condition)
  - 使用表格內之同名欄位作為連結條件

```
SELECT   ...
FROM table1 NATURAL JOIN table2
...
```

  - 若欄位名稱型態不同時，則產生錯誤訊息

```
SELECT e.ename, e.job, e.sal, e.deptno, d.dname,d.loc
FROM emp e NATURAL JOIN dept d;
```

```
SELECT e.ename, e.job, e.sal, e.deptno, d.dname,d.loc
FROM emp e JOIN dept d ON(e.deptno=d.deptno);
```

自然連結(Natural Join)

- ANSI SQL-92

```
mysql> SELECT a.empno, a.ename, a.mgr, a.sal, a.deptno, b.deptno, b.dname
    -> FROM emp a NATURAL JOIN dept b
    -> WHERE a.deptno = 10;
+-------+--------+------+------+--------+--------+------------+
| empno | ename  | mgr  | sal  | deptno | deptno | dname      |
+-------+--------+------+------+--------+--------+------------+
|  7839 | KING   | NULL | 5000 |     10 |     10 | ACCOUNTING |
|  7782 | CLARK  | 7839 | 2450 |     10 |     10 | ACCOUNTING |
|  7934 | MILLER | 7782 | 1300 |     10 |     10 | ACCOUNTING |
+-------+--------+------+------+--------+--------+------------+
```

- ANSI SQL-89

```
SELECT a.empno, a.ename, a.mgr, a.sal, a.deptno, b.deptno, b.dname
FROM emp a, dept b
WHERE a.deptno = b.deptno
      AND a.deptno = 10;
```

與上述結果相同

自然連結(Natural Join)- Using 子句

- 在相同欄位中指定欄位做為連結條件

```
mysql> SELECT a.empno, a.ename, a.mgr, a.sal, a.deptno, b.deptno, b.dname
    -> FROM emp a JOIN dept b USING(deptno)
    -> WHERE a.deptno = 10;
+-------+--------+------+------+--------+--------+------------+
| empno | ename  | mgr  | sal  | deptno | deptno | dname      |
+-------+--------+------+------+--------+--------+------------+
|  7839 | KING   | NULL | 5000 |     10 |     10 | ACCOUNTING |
|  7782 | CLARK  | 7839 | 2450 |     10 |     10 | ACCOUNTING |
|  7934 | MILLER | 7782 | 1300 |     10 |     10 | ACCOUNTING |
+-------+--------+------+------+--------+--------+------------+
3 rows in set (0.00 sec)
```

不可使用關係名稱

```
mysql> SELECT a.empno, a.ename, a.mgr, a.sal, a.deptno, b.deptno, b.dname
    -> FROM emp a JOIN dept b USING(a.deptno)
    -> WHERE a.deptno = 10;
ERROR 1064 (42000): You have an error in your SQL syntax.  Check the
manual that corresponds to your MySQL server version for the right syntax
to use near '.deptno)
 WHERE a.deptno = 10' at line 3mysql>
```

內部連結(Inner Join)語法

- 將要連結的表格名稱列在 FROM 子句中,並加入 JOIN 運算子及 ON 來設定連結條件
- 給定表格別名(Table aliases preferred)
- 表格先後順序無關

```
SELECT ...
FROM t1 JOIN t2 ON JoinCondions
...
```

```
SELECT e.ename, e.deptno, d.dname
FROM dept d JOIN emp e ON d.deptno = e.deptno;
```

```
SELECT e.ename, e.deptno, d.dname
FROM   dept d , emp e
WHERE  d.deptno = e.deptno;
```

內部連結(Inner Join)

- 使用表格全名

```
mysql> SELECT emp.empno, emp.ename, emp.mgr, emp.sal, emp.deptno,
    ->        dept.deptno, dept.dname
    -> FROM emp join dept on emp.deptno = dept.deptno;
+-------+--------+------+------+--------+--------+------------+
| empno | ename  | mgr  | sal  | deptno | deptno | dname      |
+-------+--------+------+------+--------+--------+------------+
|  7839 | KING   | NULL | 5000 |     10 |     10 | ACCOUNTING |
|  7782 | CLARK  | 7839 | 2450 |     10 |     10 | ACCOUNTING |
|  7934 | MILLER | 7782 | 1300 |     10 |     10 | ACCOUNTING |
|  7566 | JONES  | 7839 | 2975 |     20 |     20 | RESEARCH   |
|  7902 | FORD   | 7566 | 3000 |     20 |     20 | RESEARCH   |
|  7369 | SMITH  | 7902 |  800 |     20 |     20 | RESEARCH   |
|  7788 | SCOTT  | 7566 | 3000 |     20 |     20 | RESEARCH   |
|  7876 | ADAMS  | 7788 | 1100 |     20 |     20 | RESEARCH   |
|  7698 | BLAKE  | 7839 | 2850 |     30 |     30 | SALES      |
|  7654 | MARTIN | 7698 | 1250 |     30 |     30 | SALES      |
|  7499 | ALLEN  | 7698 | 1600 |     30 |     30 | SALES      |
|  7844 | TURNER | 7698 | 1500 |     30 |     30 | SALES      |
|  7900 | JAMES  | 7698 |  950 |     30 |     30 | SALES      |
|  7521 | WARD   | 7698 | 1250 |     30 |     30 | SALES      |
+-------+--------+------+------+--------+--------+------------+
```

內部連結(Inner Join)-使用表格別名

- 建立並使用表格別名

```
mysql> SELECT a.empno, a.ename, a.mgr, a.sal, a.deptno, b.deptno, b.dname
    -> FROM emp a JOIN dept b ON a.deptno = b.deptno;
+-------+--------+------+------+--------+--------+------------+
| empno | ename  | mgr  | sal  | deptno | deptno | dname      |
+-------+--------+------+------+--------+--------+------------+
|  7839 | KING   | NULL | 5000 |     10 |     10 | ACCOUNTING |
|  7782 | CLARK  | 7839 | 2450 |     10 |     10 | ACCOUNTING |
|  7934 | MILLER | 7782 | 1300 |     10 |     10 | ACCOUNTING |
|  7566 | JONES  | 7839 | 2975 |     20 |     20 | RESEARCH   |
|  7902 | FORD   | 7566 | 3000 |     20 |     20 | RESEARCH   |
|  7369 | SMITH  | 7902 |  800 |     20 |     20 | RESEARCH   |
|  7788 | SCOTT  | 7566 | 3000 |     20 |     20 | RESEARCH   |
|  7876 | ADAMS  | 7788 | 1100 |     20 |     20 | RESEARCH   |
|  7698 | BLAKE  | 7839 | 2850 |     30 |     30 | SALES      |
|  7654 | MARTIN | 7698 | 1250 |     30 |     30 | SALES      |
|  7499 | ALLEN  | 7698 | 1600 |     30 |     30 | SALES      |
|  7844 | TURNER | 7698 | 1500 |     30 |     30 | SALES      |
|  7900 | JAMES  | 7698 |  950 |     30 |     30 | SALES      |
|  7521 | WARD   | 7698 | 1250 |     30 |     30 | SALES      |
+-------+--------+------+------+--------+--------+------------+
14 rows in set (0.00 sec)
```

內部連結(Inner Join)-查詢條件

- 列出員工姓名KING所在的部門資訊

```
mysql> SELECT a.empno, a.ename, a.mgr, a.sal, a.deptno, b.deptno, b.dname
    -> FROM emp a JOIN dept b ON a.deptno = b.deptno
    -> WHERE a.ename = 'KING';
+-------+-------+------+------+--------+--------+------------+
| empno | ename | mgr  | sal  | deptno | deptno | dname      |
+-------+-------+------+------+--------+--------+------------+
|  7839 | KING  | NULL | 5000 |     10 |     10 | ACCOUNTING |
+-------+-------+------+------+--------+--------+------------+
1 row in set (0.01 sec)
```

多個資料表的連結運算

CUSTOMER

NAME	CUSTID
JOCKSPORTS	100
TKB SPORT SHOP	101
VOLLYRITE	102
JUST TENNIS	103
K+T SPORTS	105
SHAPE UP	106
WOMENS SPORTS	107
...	...

9 rows selected.

ORD

CUSTID	ORDID
101	610
102	611
104	612
106	601
102	602
106	
106	
...	

21 rows s

ITEM

ORDID	ITEMID
610	3
611	1
612	1
601	1
602	1
...	

64 rows selected.

- Join N tables
  Needs (N-1) join conditions

多個資料表的連結運算

- 語法:

```
SELECT ...
FROM t1 JOIN t2 ON t1.columnA = t2.columnA
        JOIN t3 ON t2.columnB = t3.columnB
        JOIN t4 ON t3.columnB = t4.columnB
...
```

```
SELECT c.name, c.custid, o.ordid, i.itemid
FROM customer c JOIN ord o ON c.custid=o.custid
                JOIN item i ON o.ordid=i.ordid;
```

```
SELECT  c.name, c.custid, o.ordid, i.itemid
FROM    customer c,  ord o, item i
WHERE   c.custid=o.custid AND o.ordid=i.ordid ;
```

多個資料表的連結運算

- 多個資料表的連結(計算員工業績)


```
mysql> SELECT a.empno, a.ename, sum(c.total) total
    -> FROM emp a JOIN customer b ON a.empno = b.repid
    ->            JOIN ord c ON b.custid = c.custid
    -> GROUP BY a.empno, a.ename;
+-------+--------+----------+
| empno | ename  | total    |
+-------+--------+----------+
|  7499 | ALLEN  |  7870.80 |
|  7654 | MARTIN | 27775.50 |
|  7844 | TURNER | 58050.90 |
+-------+--------+----------+
3 rows in set (0.08 sec)
```

不相等的連結(Non-Equi joins)

- 連結條件不是使用等號(=)
- 列出所有員工薪資的等級

EMP

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950

...
14 rows selected.

SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

"salary in the EMP table is between low salary and high salary in the SALGRADE table"

不相等的連結(Non-Equi joins)

- 連結條件的運算子(Operators)
  - >= 或 <=
  - BETWEEN AND

```
SELECT a.empno, a.ename, a.sal, b.grade
FROM emp a join salgrade b on
     (a.sal BETWEEN b.losal AND b.hisal);
```

- 若符合條件的資料超過一筆以上時，資料可能會不正確。

不相等的連結(Non-Equi joins)

- 列出所有員工薪資的等級

```
mysql> SELECT a.empno, a.ename, a.sal, b.grade
    -> FROM emp a JOIN salgrade b ON a.sal BETWEEN b.losal AND b.hisal;
+-------+--------+------+-------+
| empno | ename  | sal  | grade |
+-------+--------+------+-------+
|  7900 | JAMES  |  950 |     1 |
|  7369 | SMITH  |  800 |     1 |
|  7876 | ADAMS  | 1100 |     1 |
|  7654 | MARTIN | 1250 |     2 |
|  7521 | WARD   | 1250 |     2 |
|  7934 | MILLER | 1300 |     2 |
|  7499 | ALLEN  | 1600 |     3 |
|  7844 | TURNER | 1500 |     3 |
|  7698 | BLAKE  | 2850 |     4 |
|  7782 | CLARK  | 2450 |     4 |
|  7566 | JONES  | 2975 |     4 |
|  7902 | FORD   | 3000 |     4 |
|  7788 | SCOTT  | 3000 |     4 |
|  7839 | KING   | 5000 |     5 |
+-------+--------+------+-------+
14 rows in set (0.18 sec)
```

外部連結(Outer Join)

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

無40部門之員工

```
+--------+-------+--------+
| deptno | empno | ename  |
+--------+-------+--------+
|     10 |  7839 | KING   |
|     10 |  7934 | MILLER |
|     10 |  7782 | CLARK  |
|     20 |  7566 | JONES  |
|     20 |  7876 | ADAMS  |
|     20 |  7788 | SCOTT  |
|     20 |  7369 | SMITH  |
|     20 |  7902 | FORD   |
|     30 |  7521 | WARD   |
|     30 |  7698 | BLAKE  |
|     30 |  7844 | TURNER |
|     30 |  7499 | ALLEN  |
|     30 |  7654 | MARTIN |
|     30 |  7900 | JAMES  |
+--------+-------+--------+
```

```
mysql> SELECT a.deptno, a.dname, b.empno, b.ename
    -> FROM dept a JOIN emp b
    ->      ON a.deptno = b.deptno;
+--------+------------+-------+--------+
| deptno | dname      | empno | ename  |
+--------+------------+-------+--------+
|     10 | ACCOUNTING |  7839 | KING   |
|     10 | ACCOUNTING |  7782 | CLARK  |
|     10 | ACCOUNTING |  7934 | MILLER |
|     20 | RESEARCH   |  7566 | JONES  |
|     20 | RESEARCH   |  7902 | FORD   |
|     20 | RESEARCH   |  7369 | SMITH  |
|     20 | RESEARCH   |  7788 | SCOTT  |
|     20 | RESEARCH   |  7876 | ADAMS  |
|     30 | SALES      |  7698 | BLAKE  |
|     30 | SALES      |  7654 | MARTIN |
|     30 | SALES      |  7499 | ALLEN  |
|     30 | SALES      |  7844 | TURNER |
|     30 | SALES      |  7900 | JAMES  |
|     30 | SALES      |  7521 | WARD   |
+--------+------------+-------+--------+
14 rows in set (0.00 sec)
```

此連結為內部連結，40部門無法顯示

外部連結(Outer Join)

- 外部連結

```
SELECT alias.column, alias.column,...
FROM table1 [AS] alias1
          {[LEFT|RIGHT|FULL [OUTER]] JOIN }
          table2 [AS] alias2
       ON alias1.column = alias2.column
WHERE ...;
```

  - 指定一資料表，當連結條件符合時則顯示，若與第二個資料表都不符合時則以null value顯示
  - OUTER可以省略不寫
  - Types
    - LEFT OUTER JOIN
    - RIGHT OUTER JOIN
    - FULL OUTER JOIN

外部連結(Outer Join)

- 列出所有部門下的員工

```
mysql> SELECT a.deptno, a.dname, b.empno, b.ename
    -> FROM dept a LEFT OUTER JOIN emp b
    ->       ON a.deptno = b.deptno;
+--------+------------+-------+--------+
| deptno | dname      | empno | ename  |
+--------+------------+-------+--------+
|     10 | ACCOUNTING |  7839 | KING   |
|     10 | ACCOUNTING |  7782 | CLARK  |
|     10 | ACCOUNTING |  7934 | MILLER |
|     20 | RESEARCH   |  7566 | JONES  |
|     20 | RESEARCH   |  7902 | FORD   |
|     20 | RESEARCH   |  7369 | SMITH  |
|     20 | RESEARCH   |  7788 | SCOTT  |
|     20 | RESEARCH   |  7876 | ADAMS  |
|     30 | SALES      |  7698 | BLAKE  |
|     30 | SALES      |  7654 | MARTIN |
|     30 | SALES      |  7499 | ALLEN  |
|     30 | SALES      |  7844 | TURNER |
|     30 | SALES      |  7900 | JAMES  |
|     30 | SALES      |  7521 | WARD   |
|     40 | OPERATIONS |  NULL | NULL   |
+--------+------------+-------+--------+
15 rows in set (0.00 sec)
```

外部連結(Outer Join)

- 列出沒有員工的部門

```
mysql> SELECT e.ename, e.deptno, d.dname, d.loc
    -> FROM   emp e RIGHT OUTER JOIN dept d
    ->             ON  e.deptno = d.deptno
    -> WHERE  e.empno is NULL;
+-------+--------+------------+--------+
| ename | deptno | dname      | loc    |
+-------+--------+------------+--------+
| NULL  |   NULL | OPERATIONS | BOSTON |
+-------+--------+------------+--------+
1 row in set (0.01 sec)
```

自我連結(Self Joins)

- 使用同一表格做連結運算
- 使用表格別名來建立二個相同資料表的連結運算

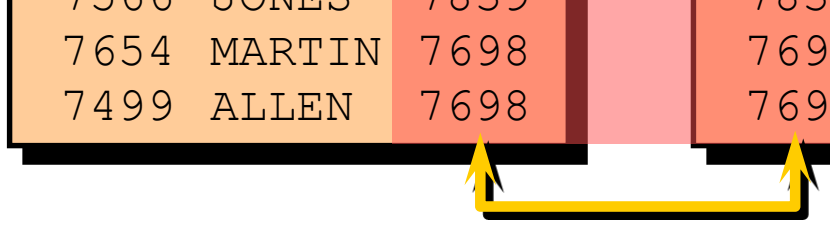
```
SELECT ...
FROM Table1 A1 JOIN Table1 A2 ON A1.Col1 = A2.Col2
...
```

EMP (WORKER)

EMPNO	ENAME	MGR
7839	KING	
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7654	MARTIN	7698
7499	ALLEN	7698

EMP (MANAGER)

EMPNO	ENAME
7839	KING
7839	KING
7839	KING
7698	BLAKE
7698	BLAKE



"MGR in the WORKER table is equal to EMPNO in the MANAGER table"

自我連結(Self Joins)

- 列出員工與他的主管的資訊

empno	ename	mgr
7839	KING	NULL
7788	SCOTT	7566
7902	FORD	7566
7521	WARD	7698
7900	JAMES	7698
7844	TURNER	7698
7499	ALLEN	7698
7654	MARTIN	7698
7934	MILLER	7782
7876	ADAMS	7788
7566	JONES	7839
7782	CLARK	7839
7698	BLAKE	7839
7369	SMITH	7902

員工

WORK FOR

empno	ename
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

主管

```
mysql> SELECT a.empno, a.ename,
    ->        a.mgr, b.ename
    -> FROM emp a join emp b
    ->       ON a.mgr = b.empno
    -> ORDER BY a.mgr;
+-------+--------+------+-------+
| empno | ename  | mgr  | ename |
+-------+--------+------+-------+
|  7788 | SCOTT  | 7566 | JONES |
|  7902 | FORD   | 7566 | JONES |
|  7521 | WARD   | 7698 | BLAKE |
|  7900 | JAMES  | 7698 | BLAKE |
|  7844 | TURNER | 7698 | BLAKE |
|  7499 | ALLEN  | 7698 | BLAKE |
|  7654 | MARTIN | 7698 | BLAKE |
|  7934 | MILLER | 7782 | CLARK |
|  7876 | ADAMS  | 7788 | SCOTT |
|  7566 | JONES  | 7839 | KING  |
|  7782 | CLARK  | 7839 | KING  |
|  7698 | BLAKE  | 7839 | KING  |
|  7369 | SMITH  | 7902 | FORD  |
+-------+--------+------+-------+
13 rows in set (0.00 sec)
```

你將發現資料少了一列， Why?

作業練習

1. 顯示所有員工之姓名,所屬部門編號,部門名稱及部門所在地點。
2. 顯示所有有賺取佣金的員工之姓名,佣金金額,部門名稱及部門所在地點。
3. 顯示姓名中包含有”A”的員工之姓名及部門名稱。
4. 顯示所有在”DALLAS”工作的員工之姓名,職稱,部門編號及部門名稱
5. 顯示出表頭名為: Employee, Emp#, Manager, Mgr#，分別表示所有員工之姓名,員工編號,主管姓字, 主管的員工編號。
6. 顯示出SALGRADE資料表的結構，並建立一查詢顯示所有員工之姓名,職稱,部門名稱,薪資及薪資等級。
7. 顯示出表頭名為: Employee, Emp Hiredate, Manager, Mgr Hiredate的資料項，來顯示所有比他的主管還要早進公司的員工之姓名,進公司日期和主管之姓名及進公司日期。
8. 顯示出表頭名為: dname, loc, Number of People, Salary的資料來顯示所有部門之部門名稱,部門所在地點,部門員工數量及部門員工的平均薪資，平均薪資四捨五入取到小數第二位。