第六章　SQL 進階查詢

黃仁鵬

6-1 關聯查詢

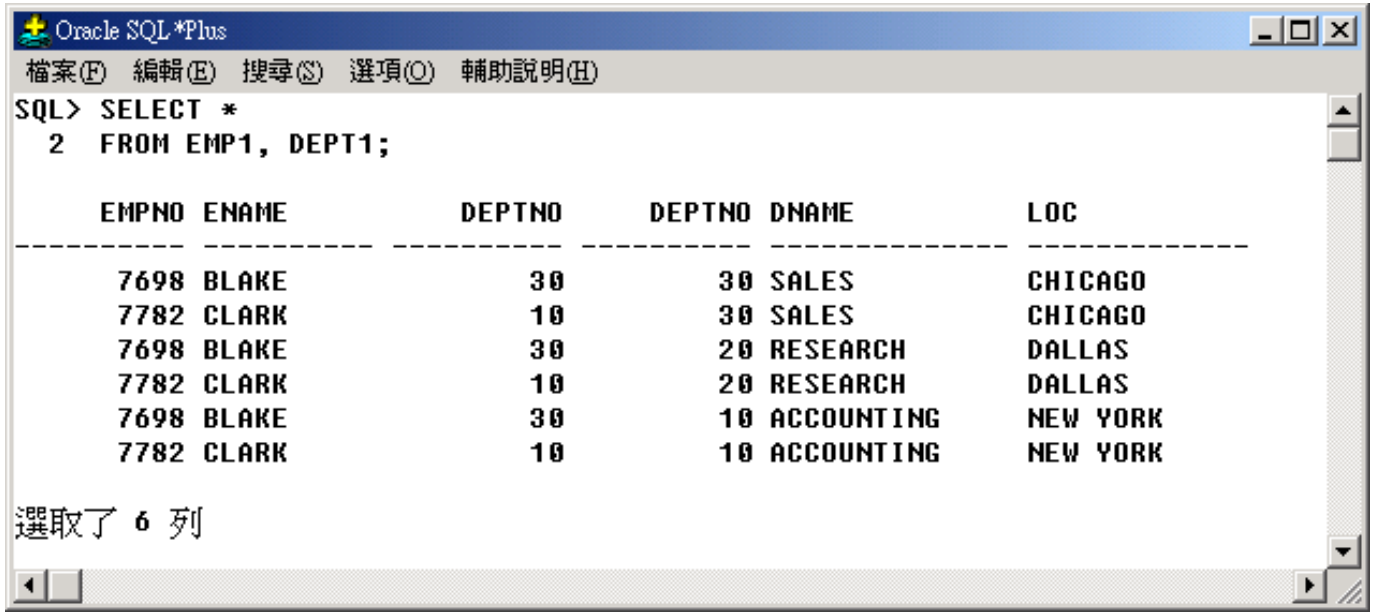
因爲資料庫設計與正規化的原因，單一資料庫表格的查詢似乎比較少，反而因爲正規化把在單一表格分割成兩個或兩個以上的表格，所以原本可以在原單一表格中查詢到的資料，現在必須透過兩個或兩個以上的表格查詢才能得到，這種跨表格的查詢稱爲關聯查詢。

6-1-1
卡笛生乘積(Cartesian Product)

卡笛生乘積實際上就是一種無條件的關聯，這種操作往往會產生大量的資料列，其結果並沒有太大的意義。因此在相關的關聯操作時，往往會加上限制條件，再進行關聯運算。

兩個表格的“卡笛生乘積”形式

SELECT <欄位名11>, <欄位名12>,…, <欄位名1m> ,
<欄位名21>, <欄位名22>,…, <欄位名2n>,
FROM <表格名1>, <表格名2>

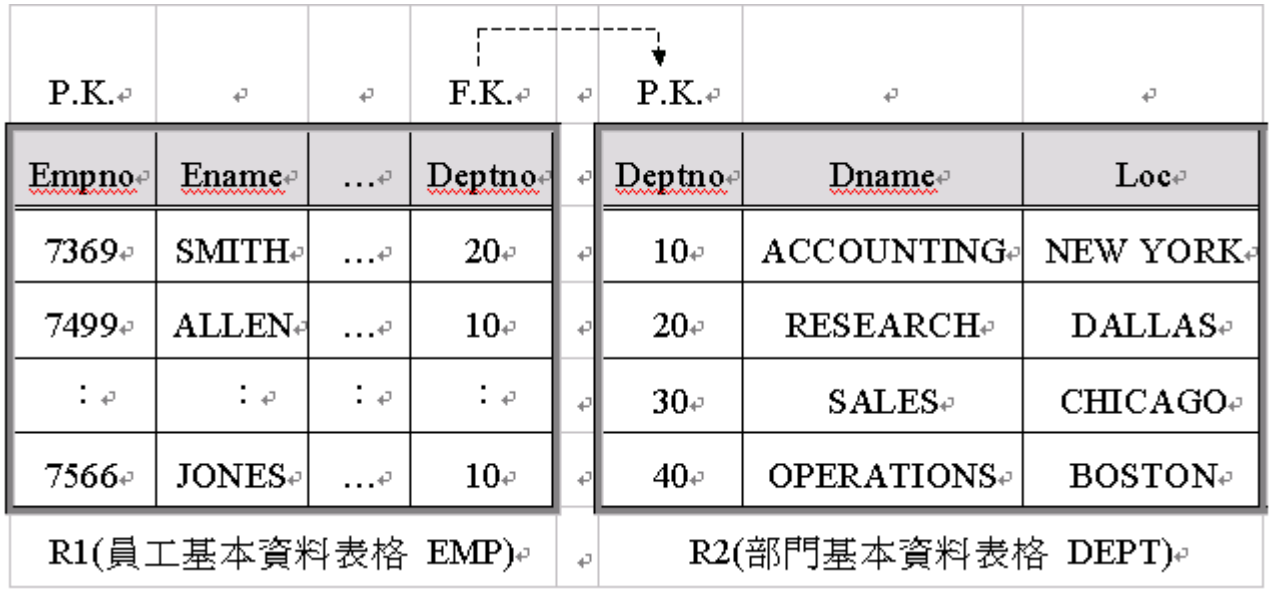


6-1-2 等值關聯(Equal Join)

SELECT …

FROM <表格名1>, <表格名2>

WHERE <表格名1.欄位名1> = <表格名2.欄位名2>;



P.K.			F.K.		P.K.		
Empno	Ename	…	Deptno		Deptno	Dname	Loc
7369	SMITH	…	20		10	ACCOUNTING	NEW YORK
7499	ALLEN	…	10		20	RESEARCH	DALLAS
:	:	:	:		30	SALES	CHICAGO
7566	JONES	…	10		40	OPERATIONS	BOSTON
R1(員工基本資料表格 EMP)					R2(部門基本資料表格 DEPT)		

6-1-3　非等值關聯

非等值關聯就是指關聯條件中不使用“ = ”運算子的關聯運算。非等值關聯能夠使用的比較運算子包括 != 、 < 、 > 、<= 、 >= 、 BETWEEN... AND 和 LIKE 等。

6-1-4 自身關聯

- 自身關聯可以把一個表格看成兩個完全相同的表格 (副本)，然後再對這兩個表格在相關欄位上進行關聯，其關聯方式與多表格關聯完全相同。

P.K. F.K.

Empno	Ename	Mgr	Sal	Deptno
7369	SMITH	7499	800	20
7499	FORD	7566	1800	10
7521	WARD	7698	1450	30
:	:	:	:	:
7566	JONES	7839	2975	10
7839	KING	-	3975	10

員工基本資料表格 EMP

- 表格自身關聯

SELECT …

FORM <表格名> <表格別名1>, <表格名> <表格 別名2>

WHERE <表格別名1.欄位名1> = <表格別名2.欄位名2>;

其中：<欄位名1> 和 <欄位名2> 標記自關聯表中的兩個不同欄位，這兩個欄位要求有相同的資料型態和寬度。

6-1-5 外部關聯

外部關聯它不僅傳回兩個或兩個以上的表格中能夠直接匹配的資料列外，還傳回關聯表格中無法直接匹配的資料列。

表格外部關聯指令

SELECT …

FROM <表格名1>, <表格名2>

WHERE { <表格名1.欄位名1> = <表格名2.欄位名2> (+) |

(+) <表格名1.欄位名1 > = <表格名2.欄位名2> } ;

Outer Join (外部關聯) 的種類

- 左外部關聯(Left Outer Join)
- 右外部關聯(Right Outer Join)
- 全外部關聯(Full Outer Join)

A1	A2	A3
A11	A21	J1
A12	A22	J2
A13	A23	J3
A14	A24	J4

R1

B1	B2	B3
J2	B21	B31
J3	B22	B32
J4	B23	B33
J5	B24	B34

R2

左外部關聯(Left Outer Join)

- 這個關聯運算會把「左方關聯表」的所有資料值輸出，如果左關聯表 R1 中找不到可匹配的 R2 值組時，則會以虛值(Null) 代替 R2 的值組再與 R1 的值關聯運算。

A1	A2	A3	B1	B2	B3
A11	A21	J1	Null	Null	Null
A12	A22	J2	J2	B21	B31
A13	A23	J3	J3	B22	B32
A14	A24	J4	J4	B23	B33

右外部關聯(Right Outer Join)

- 這個關聯運算會把「右方關聯表」的所有資料值輸出，如果右關聯表 R2 中找不到可匹配的 R1 值組時，則會以虛值(Null)代替 R1 的值組再與 R2 的值做關聯(Join)運算。

A1	A2	A3	B1	B2	B3
A12	A22	J2	J2	B21	B31
A13	A23	J3	J3	B22	B32
A14	A24	J4	J4	B23	B33
Null	Null	Null	J5	B24	B34

全外部關聯(Full Outer Join)

- 這個關聯運算會把「左方關聯表」與「右方關聯表」的所有資料值輸出，如果左關聯表 R1 或右關聯表 R2 找不到可匹配的值組做關聯運算時，則會以虛值(Null) 代替。

A1	A2	A3	B1	B2	B3
A11	A21	J1	Null	Null	Null
A12	A22	J2	J2	B21	B31
A13	A23	J3	J3	B22	B32
A14	A24	J4	J4	B23	B33
Null	Null	Null	J5	B24	B34

6-2 子查詢

所謂子查詢是指在 WHERE 子句或 HAVING 子句的條件中出現的查詢。相對地，稱包含子查詢的查詢爲父查詢或主查詢。因爲子查詢使得一系列簡單查詢可以構成複雜的查詢，因此子查詢也稱爲嵌套查詢。

- 子查詢－SELECT…FROM…WHERE

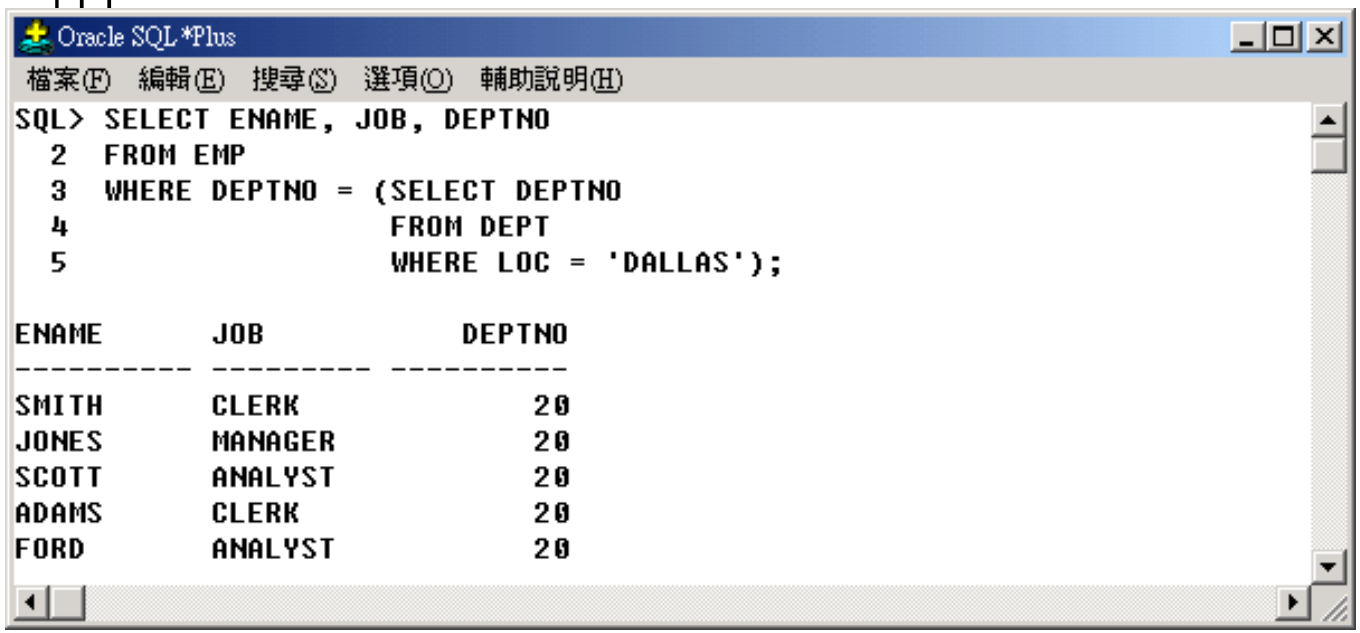
SELECT …

FROM　<表格名>

WHERE　<欄位名或欄位運算式>　<比較運算子>
(SELECT　…　FROM　<表格名>
WHERE <條件>);

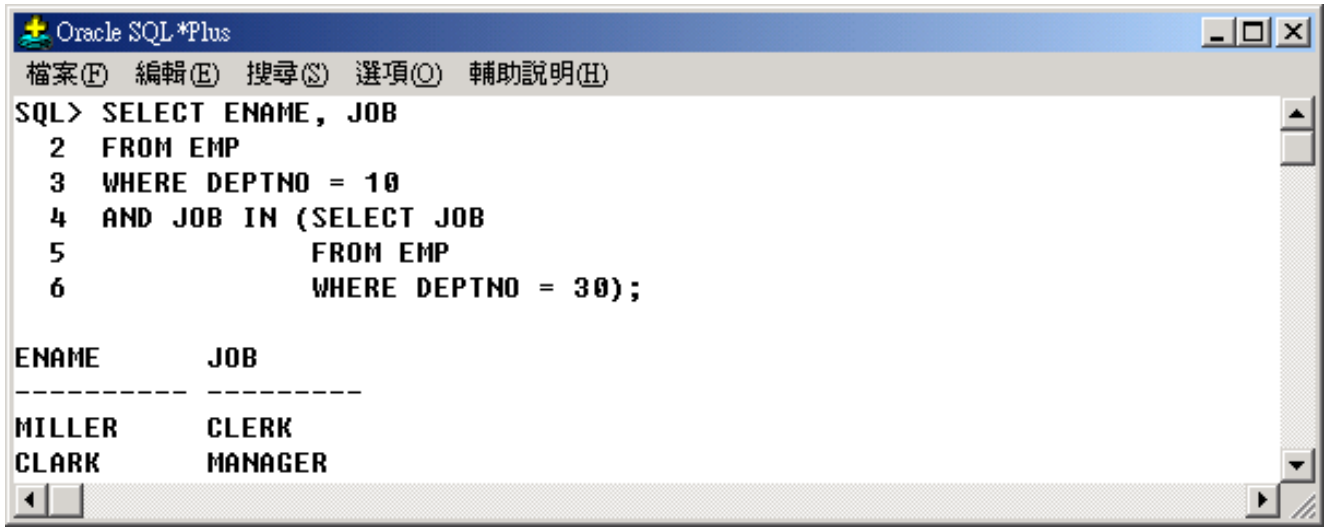
6-2-1 單一記錄值子查詢

此類子查詢只傳回單一記錄值，因此所有邏輯運算子 (如 ＞，＝，＜ 等等) 都可以用



6-2-2 多記錄值子查詢

此類子查詢傳回不是單一值而是一組資料列。這種查詢必須用多值比較運算子與主查詢相連繫。



- [NOT] IN 表示 [不] 屬於某集合中成員的關係

[NOT IN 格式]

```
SELECT ...
FROM        <表格名1>
WHERE   <表格名1.欄位名1> NOT IN
                (SELECT <表格名2.欄位名2>
                 FROM <表格名2>
                WHERE <表格名2.欄位名3> = 運算式);
```

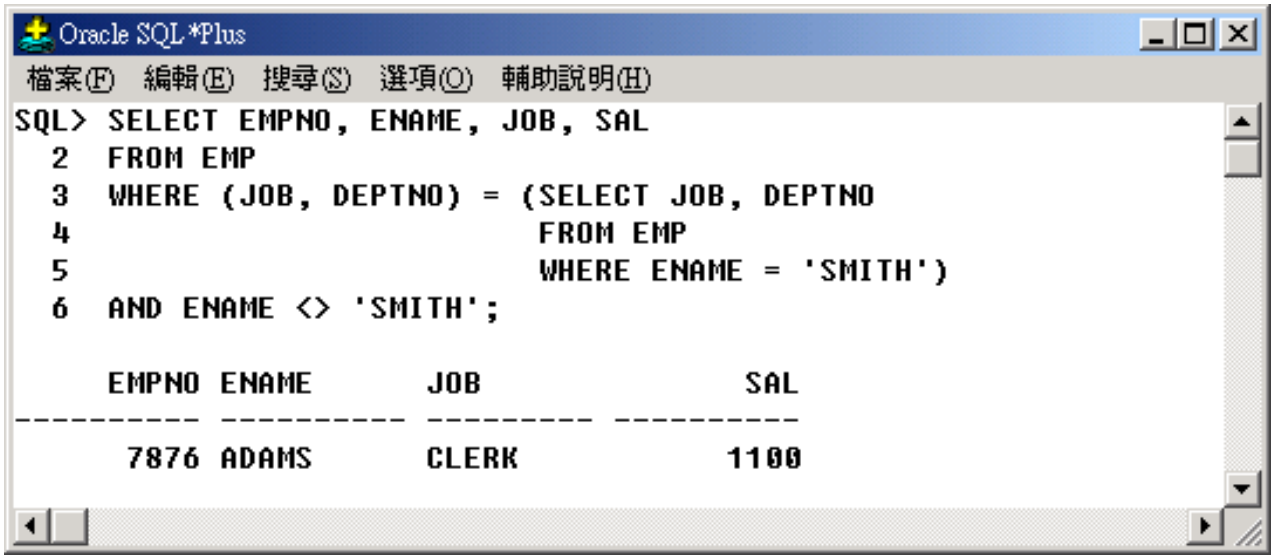
[外部關聯格式]

```
SELECT ...
FROM        <表格名1>,<表格名2>
WHERE  <表格名1.欄位名1> = <表格名2.欄位名2> (+)
 AND        <表格名2.欄位名2>  IS  NULL
AND         <表格名2.欄位名3> (+) = 運算式;
```

- [NOT] ANY：將主查詢中的一個值與子查詢傳回值中的一個值進行比較。
- [NOT] ALL：將主查詢的值與子查詢傳回值中的每個值進行比較 。
- [NOT] EXISTS：EXISTS 表示子查詢至少傳回一列時條件成立。而 NOT EXISTS 表示子查詢不傳回任何列時條件成立。

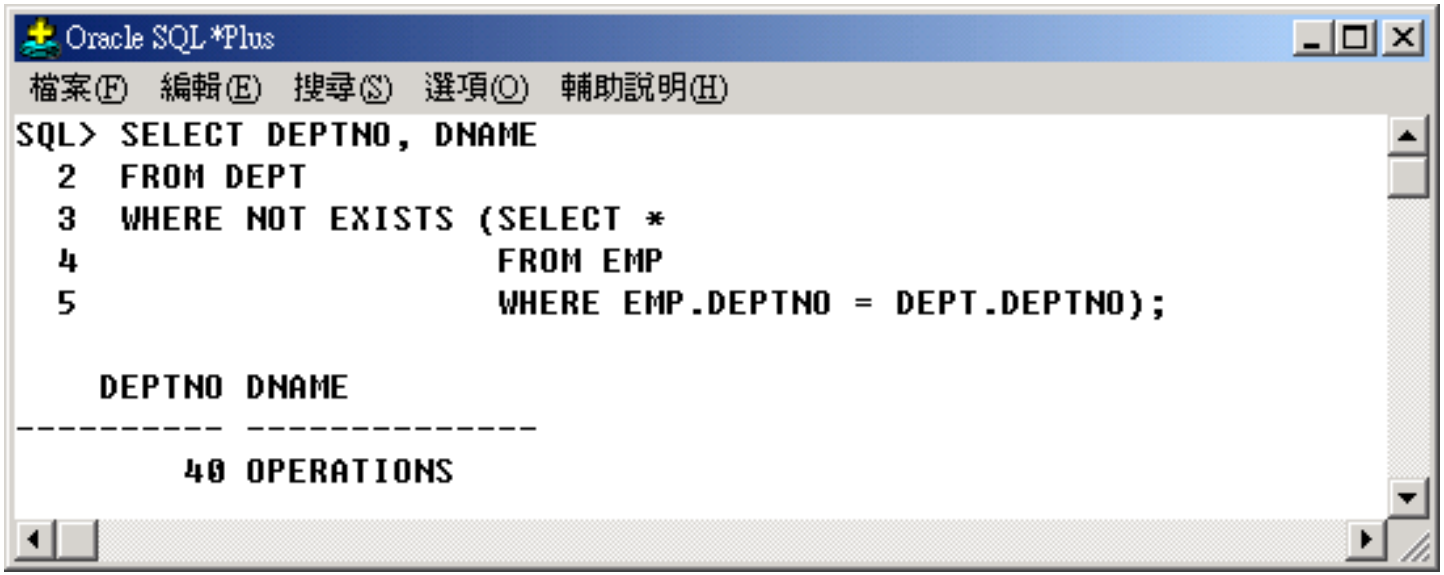
6-2-3 多欄位子查詢

子查詢中不但可以查出一個欄位的值，還可以查多個欄位。子查詢傳回欄位的個數及型態必須要與主查詢欄位的個數和型態匹配



6-2-4 相關子查詢

相關子查詢是指子查詢的 WHERE 條件子句中有引用主查詢的查詢列。反之我們稱之不相關子查詢。



6-2-5 子查詢的其他用法

- 在 UPDATE 敘述中提供更新的值，或是爲 WHERE 子句中提供進行比較的值。

```
UPDATE <表格名>
SET (<欄位名1>, <欄位名2>, ...) =
 (SELECT <欄位名1>, <欄位名2>,...
  FROM  <表格名>
   WHERE  <條件> )
WHERE  <欄位名或欄位運算式> <比較運算子> (SELECT <欄位名>
                                          FROM  <表格名>
                                          WHERE  <條件>);
```

- 在 DELETE 敘述的 WHERE 子句中提供進行比較的值。

```
DELETE FROM   <表格名>
WHERE   <欄位名或欄位運算式> <比較運算子> (SELETE <欄位名>
                                          FROM   <表格名>
                                          WHERE  <條件>);
```

- 在 INSERT 敘述中提供插入的資料列

INSERT INTO <表格名> (<欄位名>, <欄位名>, …)
SELECT (<欄位名>, <欄位名>, …)
FROM <表格名>
WHERE <欄位名或欄位運算式> <比較運算子>(SELECT <欄位名>
FROM <表格名>
WHERE<條件>);

- 在 CREATE TABLE 命令中提供插入的資料列

CREATE TABLE <新表格名> **AS**
SELECT <欄位名>, <欄位名>, …
FROM <表格名>
WHERE <欄位名或欄位運算式> <比較運算子> (SELECT <欄位名>
FROM <表格名>
WHERE <條件>);

6-3 集合查詢

每一個查詢都能得到一組的記錄列。若欲將多個查詢的結果合併爲一個結果，可以利用集合運算來達成。ORACLE 系統提供了聯集(UNION)、交集(INTERSECT) 與差集(MINUS) 三種集合運算。

6-3-1 聯集(UNION)

聯集(UNION) 是將兩個或兩個以上的查詢結果合併成一個新的結果，若有列資料分屬於兩個或兩個以上的查詢結果，則只取其中一列資料。

6-3-2　交集(INTERSECT)

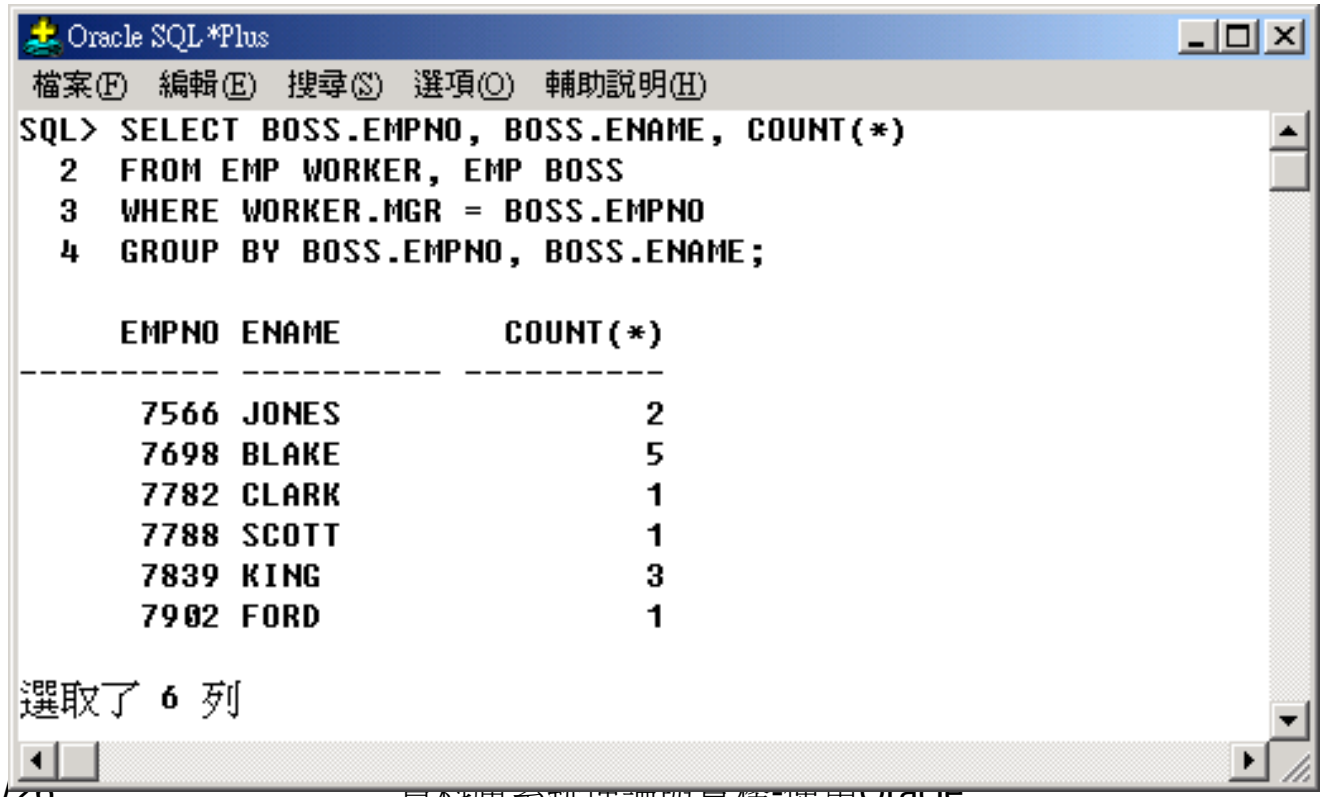
交集(INTERSECT) 是將兩個或兩個以上的查詢結果中相同共有的資料列組成新的結果。

6-3-3　差集(MINUS)

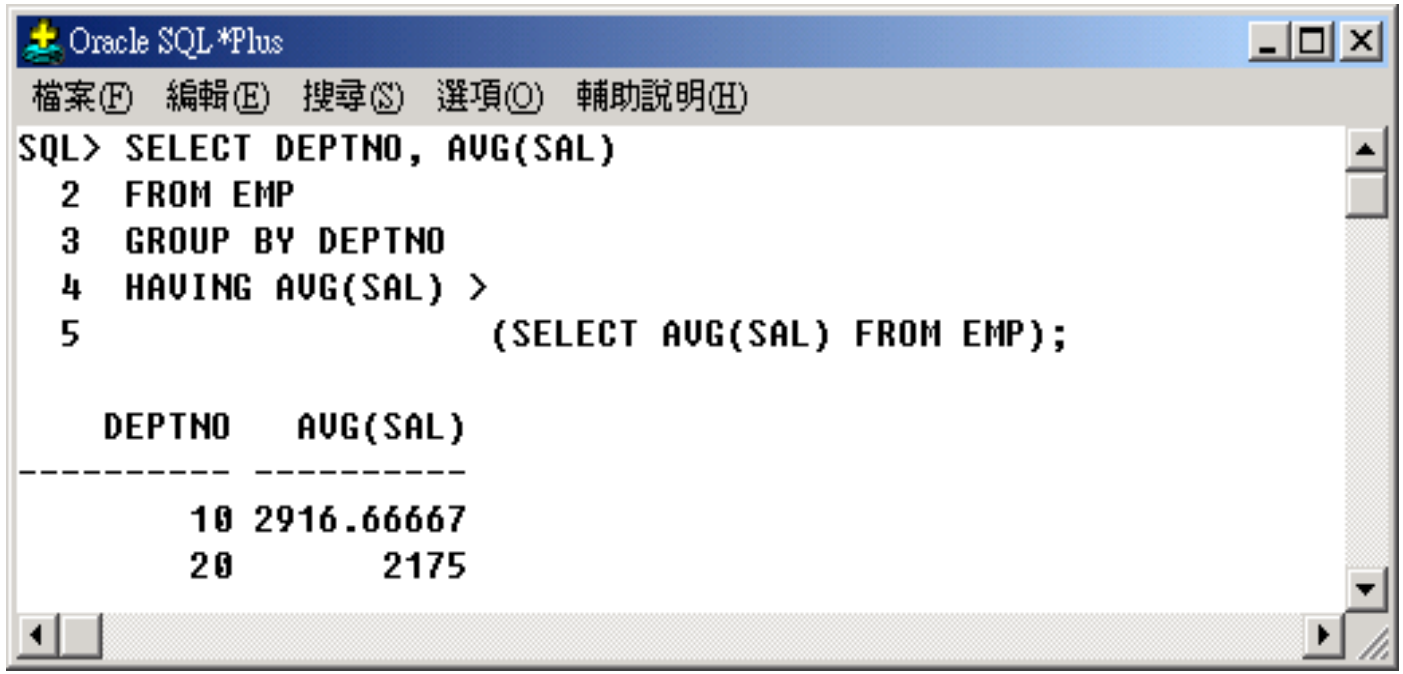
差集(MINUS) 是將一個查詢結果的資料列去掉屬於另一查詢結果的資料列所得的結果。

6-4 分組查詢

- 計算每位上司所直接管理的員工有幾位。(這裡所謂的”上司”，並非職稱(JOB)爲’MANAGER’的員工)



- 查詢那些部門，其部門的平均薪資超過全部員工的平均薪資。



- 查詢各部門非上司員工平均薪資超過 1000 元的部門資料，並列出各的部門代碼、非上司員工薪資總和、非上司員工平均薪資與各部門人數。(這裡所謂的”上司”，並非職稱(JOB)爲’MANAGER’的員工)