

關聯式資料庫管理系統 結構化查詢語言(SQL) 使用MySQL

資料庫物件的維護
(Data Definition Language)
視觀表(View)
索引(Index)

視觀表(What Is a View?)

- ▶ 視觀表(View)是藉由SELECT查詢結果動態組合生成的虛擬資料表(Virtual Table)

EMP Table

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|-------|--------|-----------|------|-----------|------|------|--------|
| 7839 | KING | PRESIDENT | | 17-NOV-81 | 5000 | | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 28-SEP-81 | 1250 | 1400 | 30 |
| 7789 | ATKINS | SALESMAN | 7698 | 30-SEP-81 | 1600 | 300 | 30 |
| 7900 | WARD | SALESMAN | 7698 | 01-JAN-82 | 1150 | 0 | 30 |
| 7934 | MILLER | CLERK | 7782 | 23-JAN-82 | 1300 | | 10 |

EMPVU10 View

| EMPNO | ENAME | JOB |
|-------|--------|-----------|
| 7839 | KING | PRESIDENT |
| 7782 | CLARK | MANAGER |
| 7934 | MILLER | CLERK |

視觀表(View)

- ▶ 視觀表(View)物件中只有查詢定義(SELECT), 而該查詢所使用的資料表稱為基底資料表(Based Tables)

```
CREATE VIEW empvu10
```

```
AS
```

```
SELECT empno, ename, sal, job
```

```
FROM emp
```

```
WHERE deptno = 10;
```

基底資料表
(Based Tables)

- 視觀表所呈現的結果和資料表(Table)類似
 - 由 rows and columns 所組成
- Views本身並不儲存任何的資料，基底資料表(Based Tables)才是真正儲存資料的地方
- ▶ 使用上有如的資料表(table)
 - 所有的查詢語法(SELECT)都可以在View上操作
 - DML-新增、刪除、更新資料，則會受限制

資料表(table)是一種實體結構(Physical Structure)
視觀表(View)是一種虛擬結構(Virtual Structure)

視觀表(View)的特性

- ▶ 可以加強資料庫的安全性
 - View 可以將實體資料表結構隱藏起來，同時限制使用者只可以檢視及使用哪些資料表欄位。
 - 檢視表可以是唯讀的，亦即外部使用者無法直接透過 View 去修改內部資料。
- ▶ 將複雜的查詢(SELECT)包裝在 View 中，可以簡化查詢的複雜度
- ▶ 當資料表結構有變更時，只需要更改 View 的設定，而不需更改程式
- ▶ 同一個資料表可建立多個視觀表

建立視觀表(Creating a View)

▶ CREATE VIEW 命令

```
CREATE [OR REPLACE] VIEW view_name [(column[,...n ])]  
AS  
  <select_statement >  
[WITH CHECK OPTION];
```

- 如果加上「OR REPLACE」子句的意思就是若同名的 View 已經存在就覆蓋取代它
 - 如果 View 已存在，可以把 CREATE OR REPLACE VIEW 當做是 ALTER VIEW
- 如果 View 不存在，CREATE OR REPLACE VIEW 如同 CREATE VIEW
- 基底資料表(Based Tables)
 - 已存在資料表或檢視表
- 不可以使用 subquery 在FROM 子句中
- 不要有ORDER BY子句，若存在ORDER BY 也將被忽略

建立視觀表(Creating a View)

- ▶ 使用基底資料表(Based Tables)的欄位名稱

```
mysql> CREATE VIEW empvu10
-> AS
->     SELECT  *
->     FROM    emp
->     WHERE   deptno = 10;
```

```
mysql> SELECT * FROM empvu10;
```

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|-------|--------|-----------|------|---------------------|---------|------|--------|
| 7782 | CLARK | MANAGER | 7839 | 1981-06-09 00:00:00 | 2450.00 | NULL | 10 |
| 7839 | KING | PRESIDENT | NULL | 1981-11-17 00:00:00 | 5000.00 | NULL | 10 |
| 7934 | MILLER | CLERK | 7782 | 1982-01-23 00:00:00 | 1300.00 | NULL | 10 |

建立視觀表 - 設定欄位名稱

```
mysql> CREATE VIEW salvu20 (EMPLOYEE_NO, EMPLOYEE, ANNUAL_SAL)
-> AS
-> SELECT empno, ename NAME, sal*12
-> FROM emp
-> WHERE deptno = 20;
```

```
mysql> desc salvu20;
```

| Field | Type | Null | Key | Default | Extra |
|-------------|--------------|------|-----|---------|-------|
| EMPLOYEE_NO | int(11) | NO | | NULL | |
| EMPLOYEE | varchar(10) | YES | | NULL | |
| ANNUAL_SAL | decimal(9,2) | YES | | NULL | |

3 rows in set (0.00 sec)

```
mysql> select * from salvu20;
```

| EMPLOYEE_NO | EMPLOYEE | ANNUAL_SAL |
|-------------|----------|------------|
| 7369 | SMITH | 9600.00 |
| 7566 | JONES | 35700.00 |
| 7788 | SCOTT | 36000.00 |
| 7876 | ADAMS | 13200.00 |
| 7902 | FORD | 36000.00 |

5 rows in set (0.00 sec)

建立視觀表 - 使用別名當欄位名稱

```
mysql> CREATE VIEW salvu30
-> AS
-> SELECT empno EMPLOYEE_NUMBER, ename NAME, sal SALARY
-> FROM emp
-> WHERE deptno = 30;
```

```
mysql> desc salvu30;
```

| Field | Type | Null | Key | Default | Extra |
|-----------------|--------------|------|-----|---------|-------|
| EMPLOYEE_NUMBER | int(11) | NO | | NULL | |
| NAME | varchar(10) | YES | | NULL | |
| SALARY | decimal(7,2) | YES | | NULL | |

```
mysql> select * from salvu30;
```

| EMPLOYEE_NUMBER | NAME | SALARY |
|-----------------|--------|---------|
| 7499 | ALLEN | 1600.00 |
| 7521 | WARD | 1250.00 |
| 7654 | MARTIN | 1250.00 |
| 7698 | BLAKE | 2850.00 |
| 7844 | TURNER | 1500.00 |
| 7900 | JAMES | 950.00 |

視觀表類別(View types)

- ▶ 簡單視觀表(Simple Views)
 - 只有一個基底資料表(Based Table)-沒有 JOIN
 - 沒有使用群組函數或做資料分組
 - 可更新的檢視表(Updateable View)
- ▶ 複雜視觀表(Complex Views) :
 - 一個以上的基底資料表(Based Table)- JOIN
 - 有使用群組函數或做資料分組
 - 不可更新的檢視表

建立視觀表 - 複雜視觀表(Complex Views)

```
mysql> CREATE VIEW dept_sum_vu(name, minsal, maxsal, avgsal)
-> AS
-> SELECT d.dname, MIN(e.sal), MAX(e.sal), AVG(e.sal)
-> FROM emp e JOIN dept d ON(e.deptno = d.deptno)
-> GROUP BY d.dname;
```

```
mysql> desc dept_sum_vu;
```

| Field | Type | Null | Key | Default | Extra |
|--------|---------------|------|-----|---------|-------|
| name | varchar(14) | YES | | NULL | |
| minsal | decimal(7,2) | YES | | NULL | |
| maxsal | decimal(7,2) | YES | | NULL | |
| avgsal | decimal(11,6) | YES | | NULL | |

```
mysql> select * from dept_sum_vu;
```

| name | minsal | maxsal | avgsal |
|------------|---------|---------|-------------|
| ACCOUNTING | 1300.00 | 5000.00 | 2916.666667 |
| RESEARCH | 800.00 | 3000.00 | 2175.000000 |
| SALES | 950.00 | 2850.00 | 1566.666667 |

視觀表(View)的使用

- ▶ View 就像是一個Table, 大部份使用Table可以完成的工作，也可以透過View來完成
- ▶ 視觀表(View)除了可以查詢外也可以透過它來做資料維護
 - 使用View來新增、修改或刪除資料
- ▶ 可更新的視觀表(Updateable View)
 - 不可以包含計算或函數的欄位
 - 只有一個基底資料表(Based Tables)

視觀表(View) - 查詢

▶ 查詢功能與資料表完全一樣

```
mysql> SELECT COUNT(*), SUM(SALARY) SUM_SAL_30, AVG(SALARY) AVG_SAL_30
-> FROM SALVU30;
```

| COUNT(*) | SUM_SAL_30 | AVG_SAL_30 |
|----------|------------|-------------|
| 6 | 9400.00 | 1566.666667 |

```
1 row in set (0.00 sec)
```

```
mysql> SELECT NAME, AVGSAL
-> FROM DEPT_SUM_VU
-> WHERE AVGSAL>2000;
```

| NAME | AVGSAL |
|------------|-------------|
| ACCOUNTING | 2916.666667 |
| RESEARCH | 2175.000000 |

```
2 rows in set (0.00 sec)
```

可更新的視觀表(Updateable View)

- ▶ 查詢VIEW是否可執行DML命令 ?
 - INFORMATION_SCHEMA.VIEWS

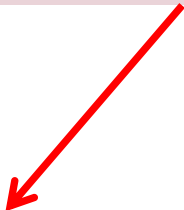
```
mysql> select table_name, is_updatable  
-> from information_schema.views;
```

| table_name | is_updatable |
|----------------------------|--------------|
| dept_sum_vu | NO |
| empvu10 | YES |
| salvu20 | YES |
| salvu30 | YES |
| actor_info | NO |
| customer_list | YES |
| film_list | NO |
| nicer_but_slower_film_list | NO |
| sales_by_film_category | NO |
| sales_by_store | NO |
| staff_list | YES |

視觀表(View) - 資料更新

```
mysql> UPDATE salvu30
-> SET SALARY = 2000
-> WHERE EMPLOYEE_NUMBER=7499;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

VIEW



```
mysql> SELECT * FROM SALVU30;
+-----+-----+-----+
| EMPLOYEE_NUMBER | NAME   | SALARY |
+-----+-----+-----+
| 7499 | ALLEN | 2000.00 |
| 7521 | WARD  | 1250.00 |
| 7654 | MARTIN | 1250.00 |
| 7698 | BLAKE | 2850.00 |
| 7844 | TURNER | 1500.00 |
| 7900 | JAMES | 950.00 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> SELECT EMPNO, ENAME, JOB, SAL
-> FROM EMP
-> WHERE EMPNO=7499;
+-----+-----+-----+-----+
| EMPNO | ENAME | JOB      | SAL      |
+-----+-----+-----+-----+
| 7499 | ALLEN | SALESMAN | 2000.00 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Based Table



視觀表(View) - 資料更新

```
mysql> UPDATE SALVU20  
-> SET EMPLOYEE='MARY'  
-> WHERE EMPLOYEE_NO=7902;  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

非計算欄位
可更新

```
mysql> SELECT * FROM SALVU20;  
+-----+-----+-----+  
| EMPLOYEE_NO | EMPLOYEE | ANNUAL_SAL |  
+-----+-----+-----+  
|          7369 | SMITH    |    9600.00 |  
|          7566 | JONES    |   35700.00 |  
|          7788 | SCOTT    |   36000.00 |  
|          7876 | ADAMS    |   13200.00 |  
|          7902 | MARY     |   36000.00 |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

計算欄位
不可更新

```
mysql> UPDATE SALVU20  
-> SET ANNUAL_SAL=48000  
-> WHERE EMPLOYEE_NO=7902;  
ERROR 1348 (HY000): Column 'ANNUAL_SAL' is not updatable
```


視觀表(View) - 新增資料

```
mysql> DESC EMP;
```

| Field | Type | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| EMPNO | int(11) | NO | PRI | NULL | |
| ENAME | varchar(10) | YES | | NULL | |
| JOB | varchar(9) | YES | | NULL | |
| MGR | int(11) | YES | MUL | NULL | |
| HIREDATE | datetime | YES | | NULL | |

```
mysql> INSERT INTO EMPVU10(EMPNO, ENAME, SAL, DEPTNO)
-> VALUES (9700, 'KEN', 3800, 10);
Query OK, 1 row affected (0.04 sec)
```

```
mysql> SELECT * FROM EMPVU10;
```

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|-------|--------|-----------|------|---------------------|---------|------|--------|
| 7782 | CLARK | MANAGER | 7839 | 1981-06-09 00:00:00 | 2450.00 | NULL | 10 |
| 7839 | KING | PRESIDENT | NULL | 1981-11-17 00:00:00 | 5000.00 | NULL | 10 |
| 7934 | MILLER | CLERK | 7782 | 1982-01-23 00:00:00 | 1300.00 | NULL | 10 |
| 9700 | KEN | NULL | NULL | NULL | 3800.00 | NULL | 10 |

4 rows in set (0.00 sec)

```
mysql> DESC EMPVU10;
+-----+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| EMPNO | int(11) | NO | PRI | NULL | |
| ENAME | varchar(10) | YES | | NULL | |
| JOB | varchar(9) | YES | | NULL | |
| MGR | int(11) | YES | MUL | NULL | |
| HIREDATE | datetime | YES | | NULL | |
| COMM | decimal(7,2) | YES | | NULL | |
| DEPTNO | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

視觀表(View) - 新增資料

```
mysql> DESC EMP;
```

| Field | Type | Null | Key | Default | Extra |
|----------|--------------|------|-----|---------|-------|
| EMPNO | int(11) | NO | PRI | NULL | |
| ENAME | varchar(10) | YES | | NULL | |
| JOB | varchar(9) | YES | | NULL | |
| MGR | int(11) | YES | MUL | NULL | |
| HIREDATE | datetime | YES | | NULL | |
| SAL | decimal(7,2) | YES | | NULL | |
| COMM | decimal(7,2) | YES | | NULL | |

```
mysql> INSERT INTO SALVU20
```

```
-> VALUES (9800, 'JACKSON', 24000);
```

ERROR 1471 (HY000): The target table SALVU20 of the INSERT is not insertable-into

```
mysql> DESC SALVU20;
```

| Field | Type | Null | Key | Default | Extra |
|-------------|--------------|------|-----|---------|-------|
| EMPLOYEE_NO | int(11) | NO | | NULL | |
| EMPLOYEE | varchar(10) | YES | | NULL | |
| ANNUAL_SAL | decimal(9,2) | YES | | NULL | |

3 rows in set (0.00 sec)

有計算欄位
不可新增資料

視觀表(View) - 刪除資料

```
mysql> DELETE FROM EMPVU10  
-> WHERE EMPNO=9700;  
Query OK, 1 row affected (0.10 sec)
```

```
mysql> DELETE FROM SALVU30  
-> WHERE NAME='JAMES';  
Query OK, 1 row affected (0.03 sec)
```

```
mysql> SELECT * FROM SALVU20;  
+-----+-----+-----+  
| EMPLOYEE_NO | EMPLOYEE | ANNUAL_SAL |  
+-----+-----+-----+  
|          7369 | SMITH    |    9600.00 |  
|          7566 | JONES    |   35700.00 |  
|          7788 | SCOTT    |   36000.00 |  
|          7876 | ADAMS    |   13200.00 |  
|          7902 | MARY     |   36000.00 |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

違反基底資料表
的資料檢查條件

```
mysql> DELETE FROM SALVU20  
-> WHERE EMPLOYEE_NO='7902';  
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`exp`.`emp`, CONSTRAINT `EMP_MGR_FK` FOREIGN KEY (`MGR`) REFERENCES `emp` (`EMPNO`))
```

```
mysql> DELETE FROM SALVU20  
-> WHERE EMPLOYEE_NO=7369;  
Query OK, 1 row affected (0.04 sec)
```

複雜視觀表(Complex Views)

```
mysql> select * from dept_sum_vu;
```

| name | minsal | maxsal | avgsal |
|------------|---------|---------|-------------|
| ACCOUNTING | 1300.00 | 5000.00 | 2916.666667 |
| RESEARCH | 1100.00 | 3000.00 | 2518.750000 |
| SALES | 1250.00 | 2850.00 | 1770.000000 |

3 rows in set (0.00 sec)

NO DML !

```
mysql> INSERT INTO dept_sum_vu
```

```
-> VALUES('EDUCATION',1200,3000,2800);
```

```
ERROR 1471 (HY000): The target table dept_sum_vu of the INSERT is not insertable-into
```

```
mysql> UPDATE dept_sum_vu
```

```
-> SET MAXSAL=4800
```

```
-> WHERE NAME='SALES';
```

```
ERROR 1288 (HY000): The target table dept_sum_vu of the UPDATE is not updatable
```

```
mysql> DELETE FROM dept_sum_vu
```

```
-> WHERE NAME='RESEARCH';
```

```
ERROR 1288 (HY000): The target table dept_sum_vu of the DELETE is not updatable
```

修改視觀表 (Altering the Definition of a View)

- ▶ 修改已建立的視觀表內容
 - 直接修改內容，不需要刪除後再重建
 - 保留原有的權限管控內容
 - 取代整個原有的內容

```
ALTER VIEW view_name [( column [ ,...n ])]  
AS  
    <select_statement>  
[ WITH CHECK OPTION ];
```

修改視觀表(Altering the Definition of a View)

- ▶ 修改視觀表內容，增加 WITH CHECK OPTION 選項

```
mysql> ALTER VIEW salvu30
-> AS
-> SELECT empno EMPLOYEE_NUMBER, ename NAME, sal SALARY, deptno
-> FROM emp
-> WHERE deptno = 30
-> WITH CHECK OPTION;
```

```
mysql> SELECT * FROM SALVU30;
```

| EMPLOYEE_NUMBER | NAME | SALARY | deptno |
|-----------------|--------|---------|--------|
| 7499 | ALLEN | 2000.00 | 30 |
| 7521 | WARD | 1250.00 | 30 |
| 7654 | MARTIN | 1250.00 | 30 |
| 7698 | BLAKE | 2850.00 | 30 |
| 7844 | TURNER | 1500.00 | 30 |

WITH CHECK OPTION 選項

```
mysql> SELECT * FROM SALVU30;
```

| EMPLOYEE_NUMBER | NAME | SALARY | deptno |
|-----------------|--------|---------|--------|
| 7499 | ALLEN | 2000.00 | 30 |
| 7521 | WARD | 1250.00 | 30 |
| 7654 | MARTIN | 1250.00 | 30 |
| 7698 | BLAKE | 2850.00 | 30 |
| 7844 | TURNER | 1500.00 | 30 |

```
mysql> DELETE FROM salvu30 WHERE EMPLOYEE_NUMBER=7521;  
Query OK, 1 row affected (0.04 sec)
```

```
mysql> UPDATE salvu30  
-> SET SALARY = 2000  
-> WHERE EMPLOYEE_NUMBER=7499;  
Query OK, 0 rows affected (0.03 sec)  
Rows matched: 1 Changed: 0 Warnings: 0
```

```
mysql> UPDATE salvu30  
-> SET deptno = 10  
-> WHERE EMPLOYEE_NUMBER=7499;  
ERROR 1369 (HY000): CHECK OPTION failed 'exp.salvu30'
```

資料更新時不可違反WHERE子句的條件

...
WHERE deptno = 30
WITH CHECK OPTION;

刪除視觀表 (Removing a View)

- ▶ 從資料庫中刪除視觀表
 - 不會刪除任何資料

```
DROP VIEW view_name;
```

```
DROP VIEW dept_sum_vu;
```

內嵌視觀表(Inline Views)

- ▶ 內嵌視觀表(Inline Views)是在FROM子句內的SELECT命令，用來定義外部 SELECT 命令的一個資料來源(Table value sub-query)

```
mysql> SELECT  a.ename, a.sal, a.deptno, b.avgsal
-> FROM      emp a Join (SELECT  deptno, avg(sal) avgsal
->                                FROM      emp
->                                GROUP BY deptno) b
->                                ON (a.deptno = b.deptno)
-> WHERE      a.sal < b.avgsal;
```

| ename | sal | deptno | avgsal |
|--------|---------|--------|-------------|
| CLARK | 2450.00 | 10 | 2916.666667 |
| MILLER | 1300.00 | 10 | 2916.666667 |
| ADAMS | 1100.00 | 20 | 2518.750000 |
| MARTIN | 1250.00 | 30 | 1900.000000 |
| TURNER | 1500.00 | 30 | 1900.000000 |

索引(Index)

▶ 什麼是索引

- 是一種檔案(大都是B-Tree結構), 鍵值具有排序的特性
- 用來加快資料的搜尋
- 沒有索引, 搜尋資料以循序的方式進行(Full Table Scan)

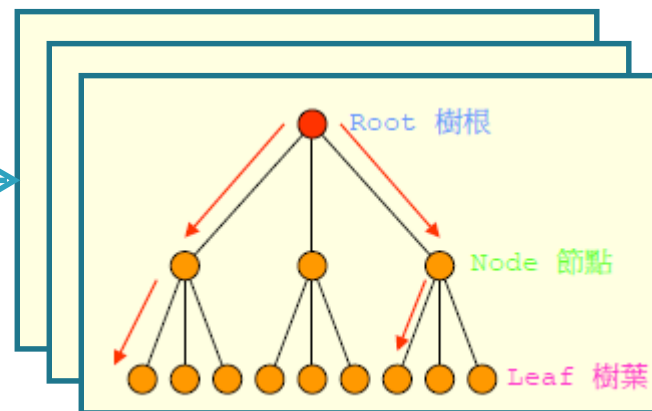
▶ 索引分為三種

- 主鍵索引 (primary key)
- 唯一索引 (unique index)
- 非唯一索引(non-unique index)

資料表(table)

| EMPLOYEE_ID | LAST_NAME | FIRST_NAME | SALARY | COMMISSION_PCT | DEPARTMENT_ID |
|-------------|-----------|------------|--------|----------------|---------------|
| 100 | King | Steven | 24000 | | 90 |
| 101 | Kochhar | Neena | 17000 | | 90 |
| 102 | De Haan | Lex | 17000 | | 90 |
| 103 | Hunold | Alexander | 9000 | | 60 |
| 104 | Ernst | Bruce | 6000 | | 60 |
| 107 | Lorentz | Diana | 4200 | | 60 |
| 124 | Mourgos | Kevin | 5800 | | 50 |
| 141 | Rajs | Trenna | 3500 | | 50 |
| 142 | Davies | Curtis | 3100 | | 50 |
| 143 | Matos | Randall | 2600 | | 50 |
| 144 | Vargas | Peter | 2500 | | 50 |
| 149 | Zlotkey | Eleni | 10500 | .2 | 80 |
| 174 | Abel | Ellen | 11000 | .3 | 80 |
| 176 | Taylor | Jonathon | 8600 | .2 | 80 |
| 178 | Grant | Kimberely | 7000 | .15 | |
| 200 | Whalen | Jennifer | 4400 | | 10 |
| 201 | Hartstein | Michael | 13000 | | 20 |
| 202 | Fay | Pat | 6000 | | 20 |
| 205 | Higgins | Shelley | 12000 | | 110 |
| 206 | Gietz | William | 8300 | | 110 |

索引(Index)



索引(Index)

- ▶ 索引(Index)可以由一個欄位或多個欄位所構成
- ▶ 建立PK, UK時, 系統會自動建立索引(Index)
- ▶ 使用者可以自行建立索引(Index)
- ▶ 於MySQL中建立FK時, 需建立索引(Index), 以方便搜尋
- ▶ PK, UK被捨棄時, 索引(Index)一併刪除
- ▶ FK被捨棄時, 並不會連同索引(Index)一併刪除
- ▶ 刪除資料表時, 所有的索引(Index)一併刪除
- ▶ 索引(Index)的使用與維護由系統負責

建立索引(Creating an Index)

▶ 建立索引

```
CREATE [UNIQUE] INDEX index  
ON TableName(column[length] [ASC|DESC],.);
```

- 可以指定某一個欄位為建立索引的欄位
- 字串型態欄位的部份資料建立索引
- 索引資料，可由小到大，或由大到小排列
- $\text{Index}(A, B) \neq \text{Index}(A) + \text{Index}(B)$
- $\text{Index}(A, B) \neq \text{Index}(B, A)$
- $\text{Index}(A) \leftarrow \text{Index}(A, B)$

▶ 查看索引

```
SHOW INDEX FROM TableName;
```

建立索引(Creating an Index)

```
mysql> SHOW INDEX FROM emp;
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | .. |
|-------|------------|---------------|--------------|-------------|----|
| emp | 0 | PRIMARY | 1 | EMPNO | .. |
| emp | 1 | EMP_DEPTNO_FK | 1 | DEPTNO | .. |
| emp | 1 | EMP_MGR_FK | 1 | MGR | .. |

3 rows in set (0.02 sec)

```
mysql> CREATE INDEX ind_emp_ename ON emp(ename);  
Query OK, 0 rows affected (0.23 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> SHOW INDEX FROM emp;
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | |
|-------|------------|---------------|--------------|-------------|--|
| emp | 0 | PRIMARY | 1 | EMPNO | |
| emp | 1 | EMP_DEPTNO_FK | 1 | DEPTNO | |
| emp | 1 | EMP_MGR_FK | 1 | MGR | |
| emp | 1 | ind_emp_ename | 1 | ENAME | |

4 rows in set (0.00 sec)

建立索引 - 複合索引鍵

```
mysql> CREATE INDEX ind_emp_dept_job ON emp(deptno, job);  
Query OK, 0 rows affected (0.25 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> SHOW INDEX FROM emp;
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name |
|-------|------------|------------------|--------------|-------------|
| emp | 0 | PRIMARY | 1 | EMPNO |
| emp | 1 | EMP_MGR_FK | 1 | MGR |
| emp | 1 | ind_emp_ename | 1 | ENAME |
| emp | 1 | ind_emp_dept_job | 1 | DEPTNO |
| emp | 1 | ind_emp_dept_job | 2 | JOB |

```
5 rows in set (0.00 sec)
```


删除索引(Dropping an Index)

```
ALTER TABLE table DROP INDEX IndexName;  
or  
DROP INDEX IndexName ON table;
```

```
mysql> ALTER TABLE emp DROP INDEX ind_emp_ename;  
Query OK, 0 rows affected (0.11 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DROP INDEX ind_emp_dept_job ON emp;  
ERROR 1553 (HY000): Cannot drop index  
'ind_emp_dept_job': needed in a foreign key constraint
```

建立索引的考量(When to create an Index)

▶ 適合建立索引的時機

- 經常出現在 WHERE 句子中的欄位
- 多個資料表連結時(PK與FK之關係)
- 經常使用 MAX(), MIN() Group Functions
- 值域分布很廣的欄位
- 空值多的欄位
- 查詢比低於2%時(筆數/總比數)

▶ 不適合建立索引的時機

- 不經常出現在WHERE句子中的欄位
- 使用像%string%的搜尋
- 查詢比大於5%時(筆數/總比數)
- 資料表異動頻繁時

作業練習

1. 使用EMP資料表中的員工編號(empno), 姓名(ename)及部門編號(deptno)來建立一個EMP_VU view, 並將姓名(ename)欄位改以EMPLOYEE。
2. 顯示EMP_VU view中的資料內容.
3. 使用EMP_VU view來顯示所有員工之姓名及部門編號
4. 新建一個名為 DEPT20的view, 包含在部門20的所有員工之員工編號, 員工姓名, 及部門編號。將View中的資料項目命名為 EMPLOYEE_ID, EMPLOYEE, and DEPARTMENT_ID. 並設定不允許使用者透過 DEPT20 來更改源公所屬的部門編號。
5. 顯示DEPT20 view的欄位定義資料(結構)及其所有資料內容.
6. 試試看利用DEPT20 view將 Smith 轉調到部門30.
7. 新建一個名為 SALARY_VU的view, 包含所有員工之姓名, 部門名稱, 薪資和薪資等級。將View中的資料項目分別命名為 Employee, Department, Salary, Grade。