

# 關聯性與限制條件之 管理

Constraint Type

Add, Rename, Drop Constraints

Constraint Information

# 限制條件

## ■ Integrity Constraints

- Entity Integrity Constraints – 配合主鍵限制
- Referential Integrity Constraints – 配合外來鍵限制
- Domain Integrity Constraints – 配合基數限制
- 加入限制條件後，資料庫將自動檢查輸入之資料是否符合條件，輸入未符合之資料時會傳回錯誤訊息

## ■ Constraint Type

- NOT NULL – 不允許空值存在
- UNIQUE – 不允許重複值存在(但允許空值)
- PRIMARY KEY – UNIQUE and NOT NULL
- FOREIGN KEY – 存在於子表格中與父表格相關聯
- CHECK – 檢查資料是否符合定義域(基數限制)

# 建立限制條件

- `CREATE TABLE tablename`  
`(col datatype [DEFAULT expr]`  
`[CONSTRAINT col_cons cons_type, ] ...,`  
`[CONSTRAINT table_cons cons_type (cons_col)`  
`REFERENCES parent_table (pk_col)`  
`[ON DELETE option], ..., ] );`
  - `tablename, col, datatype`: 表格名稱, 欄位名稱, 資料型態
  - `expr`: 欄位預設值 (如果有設定 DEFAULT)
  - `col_cons`: column level 限制條件名稱 (inline)
  - `table_cons`: table level 限制條件名稱 (out-of-line)
  - `cons_type`: 限制條件種類
  - `cons_col`: 加上限制條件之欄位名稱
  - `parent_table`: 設定 fk 時所關聯之父表格名稱
  - `pk_col`: 設定 fk 時所關聯之父表格主鍵名稱
  - `option`: 當刪除具限制條件之 pk 資料時, 控制是否將關聯鍵之資料也刪除 (CASCADE) 或改為空值 (SET NULL)

# 建立Not Null或Unique

- SQL> CREATE TABLE dept  
2 (deptno NUMBER(2) PRIMARY KEY,  
3  dname VARCHAR2(15) UNIQUE,  
4  loc VARCHAR2(15) NOT NULL) ;
  - 建立具column level限制條件之部門資料表
  - deptno: 部門編號，設為主鍵
  - dname: 部門名稱，必須是唯一值
  - loc: 部門位置，不允許空值
  - 未手動建立限制條件名稱時，系統會自動命名如SYS\_C003123之類的名稱格式
  - 如欲加上限制條件名稱，以UNIQUE為例，可依規定設計如下  
dname VARCHAR2(15) CONSTRAINT  
dept\_dname\_un UNIQUE,

# 建立Primary Key

- SQL> CREATE TABLE dept  
2 (deptno NUMBER(2),  
3 dname VARCHAR2(15) UNIQUE,  
4 loc VARCHAR2(15) NOT NULL,  
5 CONSTRAINT dept\_deptno\_pk  
PRIMARY KEY (deptno));
  - deptno以附加條件的方式設為主鍵
  - dept\_deptno\_pk: 結合表格名稱、欄位名稱及限制鍵型態作為主鍵名稱
  - table level限制條件的方式如同在建立完表格之後再另外加限制條件，維護上比較有條理，多習慣用於主鍵、外來鍵的設定。

# 建立Foreign Key

- SQL> CREATE TABLE emp  
2 (empno NUMBER(4),  
3 ename VARCHAR2(20) NOT NULL,  
4 job VARCHAR2(20),  
5 mgr NUMBER(4),  
6 hiredate DATE,  
7 sal NUMBER(7,2),  
8 comm NUMBER(7,2),  
9 deptno NUMBER(2),  
10 CONSTRAINT emp\_empno\_pk PRIMARY KEY  
(empno),  
11 CONSTRAINT emp\_deptno\_fk FOREIGN KEY  
(deptno)  
12 REFERENCES dept(deptno));
  - 建立員工資料表(emp)，並與部門資料表(dept)之deptno建立關聯
  - 以empno為主鍵，emp\_empno\_pk為主鍵名稱
  - 以deptno為外來鍵，emp\_deptno\_fk為外來鍵名稱
  - 此範例中相關聯的表格使用相同欄位名稱之關聯鍵(deptno)，亦可使用不同名稱，只要指定所關聯的表格名稱，即可建立關聯

# 建立Check

- SQL> CREATE TABLE bonus  
2 (ename VARCHAR2(20),  
3 job VARCHAR2(20),  
4 sal NUMBER(7,2),  
5 comm NUMBER(7,2),  
6 CONSTRAINT bonus\_sal\_ck CHECK (sal >  
1000));
  - 建立一個bonus表格記錄員工的職稱、薪資及紅利
  - 建立check限制條件bonus\_sal\_ck，使薪資必須大於1000
  - check限制條件亦可加在欄位後面，例如  
sal NUMBER(7,2) CONSTRAINT bonus\_sal\_ck CHECK  
(sal > 1000),
  - 如未定義限制條件名稱，系統會自動取名(如SYS\_C003183)
  - 一個欄位的check限制條件可不只一個
  - 如欲修改check限制條件，須先將其刪除再作新增條件的動作

# 修改Not Null限制條件

- ALTER TABLE *table\_name*  
MODIFY *col\_name* [CONSTRAINT  
*cons\_name*] NULL|NOT NULL;
  - Not Null屬於column level限制條件，可想像為欄位屬性的一種，利用修改欄位屬性的方式新增或移除Not Null限制條件
  - *cons\_name*: 限制條件名稱，如未定義，則系統將會以SYS\_C001234之類的格式自動命名
- Example:
  - SQL> ALTER TABLE dept  
2 MODIFY loc CONSTRAINT dept\_loc\_nn  
NOT NULL ;
    - 新增not null限制條件
  - SQL> ALTER TABLE dept  
2 MODIFY loc NULL ;
    - 移除not null限制條件

# 新增Unique限制條件

- ALTER TABLE *table\_name*  
ADD [CONSTRAINT *cons\_name*] UNIQUE  
(*col\_name*);
  - *cons\_name*: 限制條件名稱，如未定義，則系統將會以SYS\_C001234之類的格式自動命名
  - *table\_name*, *col\_name*: 表格及加限制之欄位名稱
- Example:
  - SQL> ALTER TABLE dept  
2 ADD UNIQUE (dname) ;
    - 新增unique限制條件於dname欄位並由系統自動命名
  - SQL> ALTER TABLE dept  
2 ADD CONSTRAINT dept\_dname\_un UNIQUE  
(dname) ;
    - 新增自行定義之unique限制條件於dname欄位

# 新增Check

- ALTER TABLE table\_name  
ADD [CONSTRAINT cons\_name] CHECK (check condition);
  - 如未定義限制條件名稱(即未設定CONSTRAINT語句)，系統會自動取名(如SYS\_C003183)
  - 一個欄位的check限制條件可不只一個
  - 如欲修改check限制條件，須先將其刪除再作新增條件的動作
- Example:
  - SQL> ALTER TABLE bonus  
2 ADD CONSTRAINT bonus\_sal\_ck CHECK (sal > 1000));
  - 在bonus表格中新增check限制條件bonus\_sal\_ck，使薪資必須大於1000
  - 如表格內已經有資料，但違反新增之限制條件，則會出現錯誤訊息，如  
無法驗證 (S9619401.bonus\_SAL\_CK) - 違反檢查條件

# 新增Primary Key限制條件

- ALTER TABLE *table\_name*  
ADD [CONSTRAINT *cons\_name*]  
PRIMARY KEY (*col\_name*);
  - *cons\_name*: 主鍵名稱，如未定義，則系統將會以SYS\_C001234之類的格式自動命名
  - *table\_name*, *col\_name*: 表格及加限制之欄位名稱
- Example:
  - SQL> ALTER TABLE dept  
2 ADD PRIMARY KEY (deptno) ;
    - 新增PK限制條件於deptno欄位並由系統自動命名
  - SQL> ALTER TABLE dept  
2 ADD CONSTRAINT dept\_deptno\_pk  
PRIMARY KEY (deptno) ;
    - 新增自行定義之PK限制條件於deptno欄位

# 新增 Foreign Key 限制條件

- ALTER TABLE *table\_name*  
ADD [CONSTRAINT *cons\_name*] FOREIGN KEY  
(*col\_name*) REFERENCES *parent\_table*  
(*ref\_col*) [ON DELETE *option*];
  - *table\_name*, *col\_name*: 表格及加限制之欄位名稱
  - *cons\_name*: 外來鍵名稱，如未定義，則系統將會以 SYS\_C001234 之類的格式自動命名
  - *parent\_name*: 所關連之父表格名稱
  - *ref\_col*: 所關連之主鍵欄位
  - *option*: 控制是否伴隨關聯欄位同步刪除
- Example:
  - SQL> ALTER TABLE emp  
2 ADD CONSTRAINT emp\_deptno\_fk FOREIGN KEY  
(deptno)  
3 REFERENCES dept (deptno);
    - 在 emp 新增自行定義之外來鍵於 deptno 欄位與 dept 產生關聯，當 dept 的關聯欄位資料被刪除時，外來鍵欄位資料不隨之異動

## ■ ON DELETE與整合性限制

■ SQL> ALTER TABLE emp

```
2 ADD CONSTRAINT emp_deptno_fk  
FOREIGN KEY (deptno)
```

```
3 REFERENCES dept (deptno)
```

```
4 ON DELETE SET NULL;
```

- 當父表格dept的關聯欄位(deptno)的資料被刪除時，子表格emp的外來鍵欄位對應的資料設為null

■ SQL> ALTER TABLE emp

```
2 ADD CONSTRAINT emp_deptno_fk  
FOREIGN KEY (deptno)
```

```
3 REFERENCES dept (deptno)
```

```
4 ON DELETE CASCADE;
```

- 當父表格dept的關聯欄位(deptno)的資料被刪除時，子表格emp的外來鍵欄位對應的資料設為也一併刪除

# 重新命名或終止限制條件

- ALTER TABLE *table\_name*  
RENAME CONSTRAINT *old\_cons* TO *new\_cons* ;
  - *old\_cons*: 原始限制條件名稱
  - *new\_cons*: 更名後之限制條件名稱
- ALTER TABLE *table\_name* ENABLE | DISABLE  
CONSTRAINT *cons\_name* [ CASCADE ] ;
  - ENABLE: 啟動限制條件
  - DISABLE: 終止限制條件
  - *cons\_name*: 限制條件名稱
  - CASCADE: 限制條件終止或啟動後亦終止或啟動關聯欄位的限制條件
- Example :
  - SQL> ALTER TABLE dept RENAME CONSTRAINT  
2 dept\_deptno\_pk TO dept\_dno\_pk ;
    - 將限制條件dept\_deptno\_pk更名為dept\_dno\_pk
  - SQL> ALTER TABLE dept DISABLE CONSTRAINT  
2 dept\_dno\_pk CASCADE ;
    - 終止限制條件dept\_dno\_pk(但未移除)，並同時終止相關聯欄位之限制條件

# 刪除限制條件

- ALTER TABLE *table\_name*  
DROP *cons\_type* [(*col\_name*)] [CASCADE] ;
  - *cons\_type*: 限制條件型態
  - *col\_name*: 如限制條件為UNIQUE，需加上欄位名稱
  - *cons\_name*: 限制條件名稱
  - CASCADE: 限制條件刪除後亦移除關聯欄位的限制條件
- ALTER TABLE *table\_name*  
DROP CONSTRAINT *cons\_name* [CASCADE] ;
  - *cons\_name*: 限制條件名稱
- Example:
  - SQL> ALTER TABLE dept  
2 DROP CONSTRAINT dept\_deptno\_pk CASCADE ;
    - 刪除主鍵dept\_deptno\_pk並移除關聯欄位之外來鍵
  - SQL> ALTER TABLE dept  
2 DROP UNIQUE dname
    - 移除dname之UNIQUE限制條件

# 查詢限制條件

- Data Dictionary – 資料字典
  - 提供資料庫管理人員查詢關聯表格屬性資訊
  - Oracle伺服器之資料字典表格分user\_, dba\_, all\_三種層級，一般使用者網要權限內可查詢user\_層級的資料字典，如
    - user\_tables, user\_constraints, user\_views, ...等
- user\_constraints –
  - 提供所有被使用者定義於關聯式資料表之限制條件資訊，如
  - SQL> SELECT constraint\_name, constraint\_type,  
table\_name, r\_constraint\_name  
2 FROM user\_constraints ;
    - 查詢所有的限制條件之名稱、型態與其存在的表格
- user\_cons\_columns
  - 提供被使用者定義限制條件之欄位資訊，如
  - SQL> SELECT constraint\_name, table\_name,  
column\_name  
2 FROM user\_cons\_columns ;
    - 查詢所有限制條件存在於表格的哪些欄位上