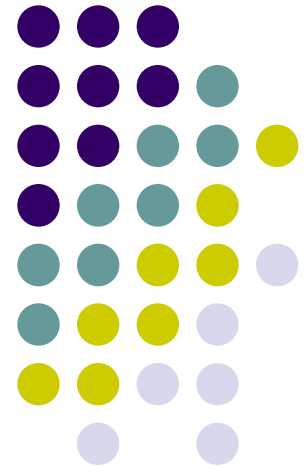


第 8 章

SQL-99：綱要定義、限制、
查詢與視界





CONTENTS

- Create Table
- Drop Table
- Alter Table
- SQL的修改命令
 - INSERT, DELETE, UPDATE
- CREAT ASSERTION
- VIEW



變更資料定義、限制或綱要

- 用來建立 (CREATE)、刪除 (DROP) 和修改 (ALTER) 資料庫中表格 (關聯) 的描述



CREATE TABLE

- 用來建立一個新的關聯
 - 指定關聯的名稱
 - 指定每個屬性及它們的資料型態
 - INTEGER
 - FLOAT
 - DECIMAL(i,j)
 - CHAR(n)
 - VARCHAR(n)



DEP-EMP Database

EMPLOYEE

| | | | | | | | | | |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

DEPARTMENT

| | | | |
|-------|----------------|---------|----------------|
| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|-------|----------------|---------|----------------|

DEPT_LOCATIONS

| | |
|----------------|------------------|
| <u>Dnumber</u> | <u>Dlocation</u> |
|----------------|------------------|

PROJECT

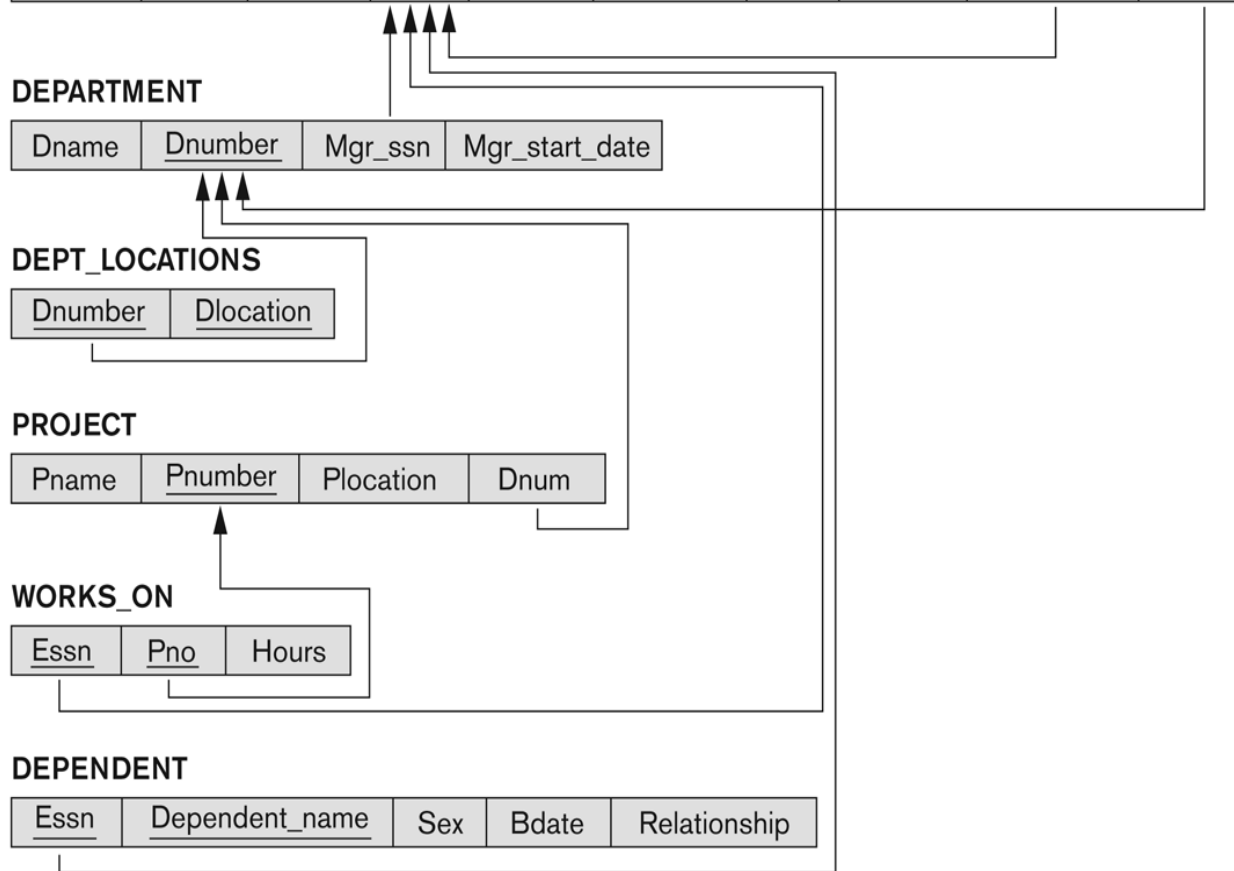
| | | | |
|-------|----------------|-----------|------|
| Pname | <u>Pnumber</u> | Plocation | Dnum |
|-------|----------------|-----------|------|

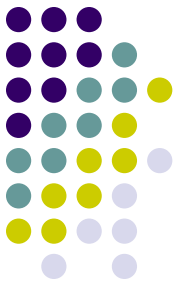
WORKS_ON

| | | |
|-------------|------------|-------|
| <u>Essn</u> | <u>Pno</u> | Hours |
|-------------|------------|-------|

DEPENDENT

| | | | | |
|-------------|-----------------------|-----|-------|--------------|
| <u>Essn</u> | <u>Dependent_name</u> | Sex | Bdate | Relationship |
|-------------|-----------------------|-----|-------|--------------|

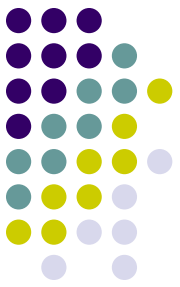




CREATE TABLE

- 在屬性上還可能會指定限制
 - 如NOT NULL
 - 預設是 NULL

```
CREATE TABLE DEPARTMENT (  
    DNAME          VARCHAR(10)    NOT NULL,  
    DNUMBER        INTEGER        NOT NULL,  
    MGRSSN         CHAR(9) ,  
    MGRSTARTDATE   CHAR(9)      ) ;
```



CREATE TABLE

- 屬性可以指定限制
- 預設是 NULL
- 可以指定預設值
 - `DNO` `INTEGER` `DEFAULT 1,`
- 限制定義域
 - `Dnumber INT NOT NULL CHECK (Dnumber > 0 AND Dnumber < 21);`
 - `CREATE DOMAIN D_NUM AS INTEGER CHECK (D_NUM > 0 AND D_NUM < 21)`

CREATE TABLE



- 指定主鍵屬性、替代鍵和參考完整性限制 (外來鍵)
- 鍵值屬性是用PRIMARY KEY和UNIQUE子句來指定

```
CREATE TABLE DEPARTMENT (  
    DNAME          VARCHAR(10) NOT NULL,  
    DNUMBER        INTEGER      NOT NULL,  
    MGRSSN         CHAR(9) ,  
    MGRSTARTDATE  CHAR(9) ,  
    PRIMARY KEY (DNUMBER) ,  
    UNIQUE (DNAME) ,  
    FOREIGN KEY (MGRSSN) REFERENCES EMPLOYEE ) ;
```



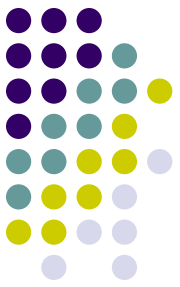


DROP TABLE

- 用來刪除綱要裡的關聯 (基底表格) 及其定義
- 之後在任何查詢或其他命令中就不能再用到這個關聯，因為它的定義資訊已經不存在了
- 範例：

DROP TABLE DEPENDENT ;





ALTER TABLE

- 用來變更基底關聯的定義
 - 假如是新增一個屬性，則在剛執行完此命令時，關聯中所有值組的這個新屬性其值都是NULL，因此這時不允許有NOT NULL限制
- 範例：
**ALTER TABLE EMPLOYEE ADD JOB
VARCHAR(12);**
- 之後資料庫使用者還是必須替每筆EMPLOYEE值組輸入新的屬性JOB的值
 - 這可以使用UPDATE命令來做

CREATE SCHEMA



- 指定名稱建立一個新的資料庫綱要
 - **CREATE SCHEMA COMPANY
AUTHORIZATION Jsmith;**





參考完整性限制的選項

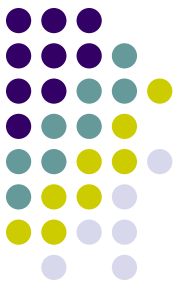
- 可針對參考完整性限制 (外來鍵) 指定RESTRICT、CASCADE、SET NULL或SET DEFAULT

```
CREATE TABLE DEPARTMENT (  
    DNAME          VARCHAR(10)          NOT NULL,  
    DNUMBER        INTEGER             NOT NULL,  
    MGRSSN         CHAR(9) ,  
    MGRSTARTDATE  CHAR(9) ,  
    PRIMARY KEY (DNUMBER) ,  
    UNIQUE (DNAME) ,  
    FOREIGN KEY (MGRSSN) REFERENCES EMPLOYEE  
    ON DELETE SET DEFAULT ON UPDATE CASCADE) ;
```



參考完整性限制的選項 (續)

```
CREATE TABLE EMPLOYEE (  
    ENAME          VARCHAR(30) NOT NULL,  
    ESSN           CHAR(9) ,  
    BDATE          DATE ,  
    DNO            INTEGER  DEFAULT 1 ,  
    SUPERSSN       CHAR(9) ,  
    PRIMARY KEY   (ESSN) ,  
    FOREIGN KEY   (DNO) REFERENCES DEPARTMENT  
        ON DELETE SET DEFAULT  
        ON UPDATE CASCADE ,  
    FOREIGN KEY   (SUPERSSN) REFERENCES EMPLOYEE ON  
    DELETE SET NULL ON UPDATE CASCADE) ;
```



SQL2與SQL-99的新增資料型態

包括DATE、TIME和TIMESTAMP資料型態

- **DATE** :
 - 由年-月-日所組成，格式為yyyy-mm-dd
- **TIME** :
 - 由時:分:秒所組成，格式為hh:mm:ss
- **TIME(i)** :
 - 由時:分:秒加上代表幾分之一秒的數字 i 所組成
 - 格式為hh:mm:ss:ii...i

SQL2與SQL-99的新增資料型態



- **TIMESTAMP (時間戳記) :**
 - 由DATE和TIME元件所組成
- **INTERVAL (期間) :**
 - 指定一個相對值，而不是絕對值
 - 可能是YEAR/MONTH期間或DAY/TIME期間
 - 當它相加或相減另一個絕對值，結果可能是正數或負數，結果也會是絕對值





SQL的修改命令

- 在SQL中有3個命令可以用來修改資料庫資料內容：
 - **INSERT**
 - **DELETE** 和
 - **UPDATE**



INSERT

- 最簡單的INSERT形式是在關聯中加入一筆值組
- 在指定值組內欄位的值時，必須以**CREATE TABLE**命令指定的屬性順序來排列
- 範例：

```
U1: INSERT INTO      EMPLOYEE
      VALUES ('Richard','K','Marini', '653298653', '30-DEC-52',
      '98 Oak Forest,Katy,TX', 'M', 37000,'987654321', 4 )
```



INSERT (續)

- 另一種形式是允許使用者明確的指定對應到 **INSERT** 命令裡的數值的屬性名稱
 - 可以有 **NULL** 值的屬性則可以不指定資料值
 - 範例：輸入一筆只指定 **FNAME**、**LNAME** 和 **SSN** 屬性的新 **EMPLOYEE** 值組

U1A:

```
INSERT INTO    EMPLOYEE (FNAME, LNAME,SSN)
              VALUES ('Richard', 'Marini', '653298653')
```



INSERT (續)

- 注意：只有以DDL命令指定的限制，DBMS才會在變更資料庫時自動遵守
- INSERT命令有種變化形式，就是在一個查詢中插入多個值組到關聯



INSERT (續)

- 範例：假設要建立具有每個部門的名稱、員工人數與總薪資的暫存表格
 - 首先使用U3A建立DEPTS_INFO表格，然後再使用U3B從資料庫中擷取摘要資訊

```
U3A:CREATE TABLE DEPTS_INFO
      (DEPT_NAME          VARCHAR(10),
       NO_OF_EMPS        INTEGER,
       TOTAL_SAL         INTEGER);
```

```
U3B:INSERT INTO DEPTS_INFO (DEPT_NAME,
      NO_OF_EMPS, TOTAL_SAL)
      SELECT      DNAME, COUNT (*), SUM (SALARY)
      FROM        DEPARTMENT, EMPLOYEE
      WHERE       DNUMBER=DNO
      GROUP BY   DNAME ;
```



DELETE

- 用來將值組從關聯中移除
 - 包括**WHERE**子句來選取要被刪除的值組
 - 一次只能從一個表格中刪除值組 (除非在參考完整性限制上有指定**CASCADE**)
 - 假如沒有指定**WHERE**子句，則會將值組全部刪除，最後把這個表格當作一個空表格留在資料庫中
 - 被刪除的值組個數是根據符合**WHERE**子句條件的個數
 - 一定會遵守參考完整性限制

DELETE (續)



- 範例：

U4A:DELETE FROM
WHERE

EMPLOYEE
LNAME='Brown'

U4B:DELETE FROM
WHERE

EMPLOYEE
SSN='123456789'

U4C: DELETE FROM
WHERE

EMPLOYEE
DNO IN
(SELECT DNUMBER
FROM DEPARTMENT
WHERE
DNAME='Research')

U4D: DELETE FROM

EMPLOYEE



UPDATE

- 用來修改一或多個被選取值組的屬性值
- **WHERE**子句從關聯中選取要修改的值組
- **SET**子句是用來指定被修改的屬性和新數值
- 每個命令修改同一個關聯裡的值組
- 一定會遵守參考完整性限制

UPDATE (續)



- 範例：將10號計劃的位置與控管部門編號分別改成 'Bellaire' 和5

```
U5: UPDATE    PROJECT
      SET PLOCATION = 'Bellaire', DNUM = 5
      WHERE     PNUMBER=10
```




UPDATE (續)

- 範例：將所有在 'Research' 部門工作的員工加薪10%

```
U6: UPDATE      EMPLOYEE
      SET        SALARY = SALARY *1.1
      WHERE      DNO IN (SELECT  DNUMBER
                          FROM    DEPARTMENT
                          WHERE    DNAME='Research')
```

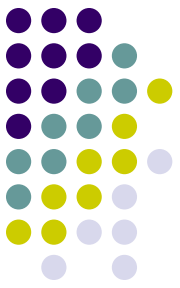
- 因為修改後的薪資值是依據每個值組原先的薪資值，所以需要參考**SALARY**屬性兩次
 - U6中右邊的**SALARY**屬性的參考動作是參考在修改前的舊**SALARY**值
 - 左邊的是參考在修改之後的新**SALARY**值





使用宣示敘述指定一般的限制

- 一般性的限制：那些不屬於基本SQL分類的限制
- 機制：**CREATE ASSERTION**
 - 元件包括：
 - 限制名稱
 - 之後的是CHECK關鍵字
 - 緊接著是一個條件

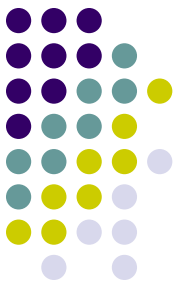


宣示敘述範例

- 指定「員工的薪資不得超過其工作部門經理的薪資」這個限制

```
CREATE ASSERTION SALARY_CONSTRAINT
CHECK
(NOT EXISTS (SELECT *
              FROM EMPLOYEE E, EMPLOYEE M,
                   DEPARTMENT D
              WHERE E.SALARY > M.SALARY AND
                    E.DNO=D.NUMBER AND
                    D.MGRSSN=M.SSN) )
```

限制名稱
CHECK
條件



使用一般的宣示敘述

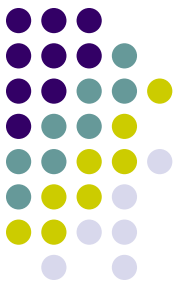
- 指定查詢選出那些會違反要求條件的值組，並在查詢中加入NOT EXISTS子句
- 查詢的結果必須是空集合
 - 假如查詢結果不是空集合，那就代表違反宣示敘述





SQL中的視界

- 視界 (**view**) 是指一個從其他表格所衍生出來的「**虛擬表格**」(**virtual table**)
- 這會限制可能應用在視界上的更新運算動作
 - 因為這個表格可能在實際上並不存在
- 在視界上進行查詢時並不會有任何限制
- 方便用來表達某些運算



視界的規格

- **SQL命令：CREATE VIEW**
 - 一個表格 (視界) 名稱
 - 屬性名稱清單和 (例如視界的屬性全都不是由函數或算術運算子產生的，就不需要指定視界的屬性名稱，它們將會與定義表格裡的屬性名稱相同)
 - 指定視界內容的查詢



SQL視界的範例

- 指定另一個不同的WORKS_ON表格

```
CREATE VIEW WORKS_ON_NEW AS  
SELECT FNAME, LNAME, PNAME, HOURS  
FROM EMPLOYEE, PROJECT, WORKS_ON  
WHERE SSN=ESSN AND PNO=PNUMBER  
GROUP BY PNAME;
```



使用虛擬表格

- 我們可以在新建立的視界上指定SQL查詢：

```
SELECT FNAME, LNAME  
FROM WORKS_ON_NEW  
WHERE PNAME='Seena';
```

- 等到不需要它的時候，再刪除該視界：

```
DROP WORKS_ON_NEW;
```




有效率的視界查詢

- 查詢修改：
 - 將視界查詢修改為針對基底表格的查詢
- 缺點：
 - 對於複雜的視界執行時會變得很沒有效率
 - 特別是假如在一段很短的時間內，有很多個查詢同時在使用此視界時



有效率的視界查詢 (續)

- 視界具體化 (view materialization) :
 - 在實體上真正建立一個暫存的視界表格並維護它
- 假設：
 - 視界的其他查詢也會跟著使用和維護這個暫存表格
- 注意：
 - 在基底表格更新時，必須維護基底表格與視界之間的一致
- 策略：
 - 遞增更新



視界的更新

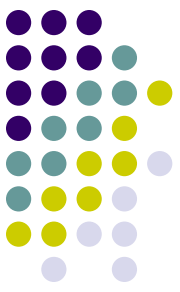
- 假如視界是定義在單一表格上，而且不含有聚合函數：
 - 則它的更新動作可以直接對應成基底表格的更新動作
- 如果視界是合併的：
 - 更新動作可能對映成基底關聯的更新動作
 - 但不一定可以



無法更新的視界

- 有定義使用群組化或聚合函數的視界不能執行更新動作
- 假如視界是定義在使用合併的多個表格上，通常不能執行更新動作
- **WITH CHECK OPTION**：假如視界是需要被更新的，必須在視界定義的結尾加上它
 - 這樣系統就會去檢查視界是否可進行更新，並且去規劃出視界更新時的執行策略





學習評量 (問題回顧)

- 8.1 請列出SQL所允許的合法的屬性資料型態。
- 8.2 SQL是用什麼方式來實作第5章所描述的實體完整性與參考完整性限制？而參考所觸發的動作是什麼？
- 8.3 請描述SQL擷取查詢中的6個子句的語法。並分別說明這6個子句中分別可以指定什麼種類的元件？這6個子句中哪些是必要的？那些是選用的？
- 8.4 請說明在理論上SQL查詢中的6個子句的執行順序。
- 8.5 請說明SQL在執行比較運算時如何處理NULL值。假如在SQL查詢中用到聚合函數時，是如何處理NULL值？如果在群組化屬性中有NULL時又是如何處理？



學生與課程資訊的資料庫

STUDENT

| Name | Student_number | Class | Major |
|-------|----------------|-------|-------|
| Smith | 17 | 1 | CS |
| Brown | 8 | 2 | CS |

COURSE

| Course_name | Course_number | Credit_hours | Department |
|---------------------------|---------------|--------------|------------|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

SECTION

| Section_identifier | Course_number | Semester | Year | Instructor |
|--------------------|---------------|----------|------|------------|
| 85 | MATH2410 | Fall | 04 | King |
| 92 | CS1310 | Fall | 04 | Anderson |
| 102 | CS3320 | Spring | 05 | Knuth |
| 112 | MATH2410 | Fall | 05 | Chang |
| 119 | CS1310 | Fall | 05 | Anderson |
| 135 | CS3380 | Fall | 05 | Stone |

GRADE_REPORT

| Student_number | Section_identifier | Grade |
|----------------|--------------------|-------|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

PREREQUISITE

| Course_number | Prerequisite_number |
|---------------|---------------------|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |



學習評量 (習題)

8.1 上頁學生生與課程資訊的資料庫，那些欄位是有保持參考完整性的限制？

試撰寫適當的SQL DDL敘述來定義資料庫。

8.5 針對學生生與課程資訊的資料庫，替下列工作撰寫SQL敘述。

(a) 插入一位新學生 <'Johnson', 25, 1, 'MATH'> in

(b) 將學生 'Smith' 的班級改為 2.

(c) 插入一筆新課程

<'Knowledge Engineering','COSC4390', 3,'COSC'>.

(d) 刪除一位名叫 'Smith' 且學號為17的學生