



第11章 視界與資料庫程式設計

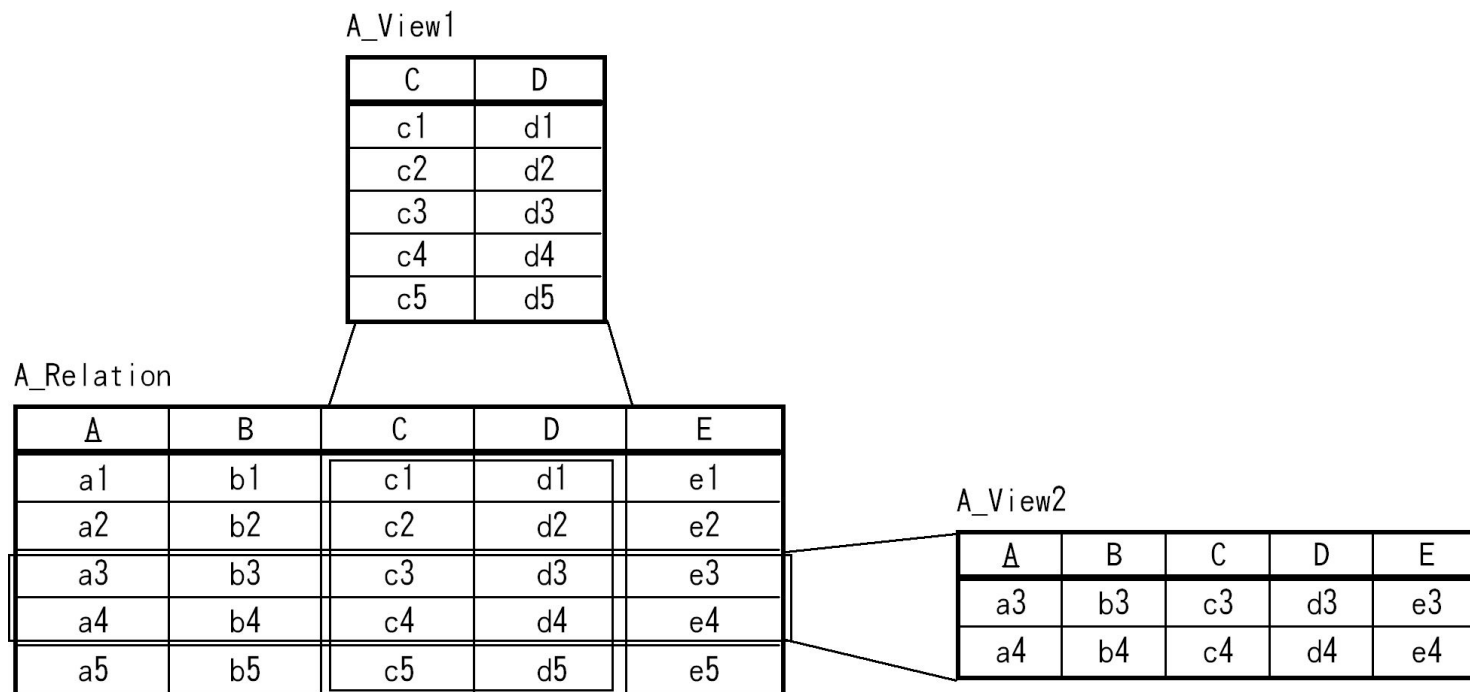
- 11-1 視界的基礎
- 11-2 建立與刪除視界
- 11-3 編輯視界的內容
- 11-4 資料庫程式設計的基礎
- 11-5 嵌入式SQL
- 11-6 動態SQL與ORM
- 11-7 Transact-SQL的預存程序





11-1-1 視界的內容-說明

- 「視界」 (Views) 相當於ANSI/SPARC三層資料系統架構中外部層 (External Level) 顯示的資料，這是從基底關聯表 (Base Relation) 導出的虛擬關聯表 (Virtual Relation)，如下圖所示：



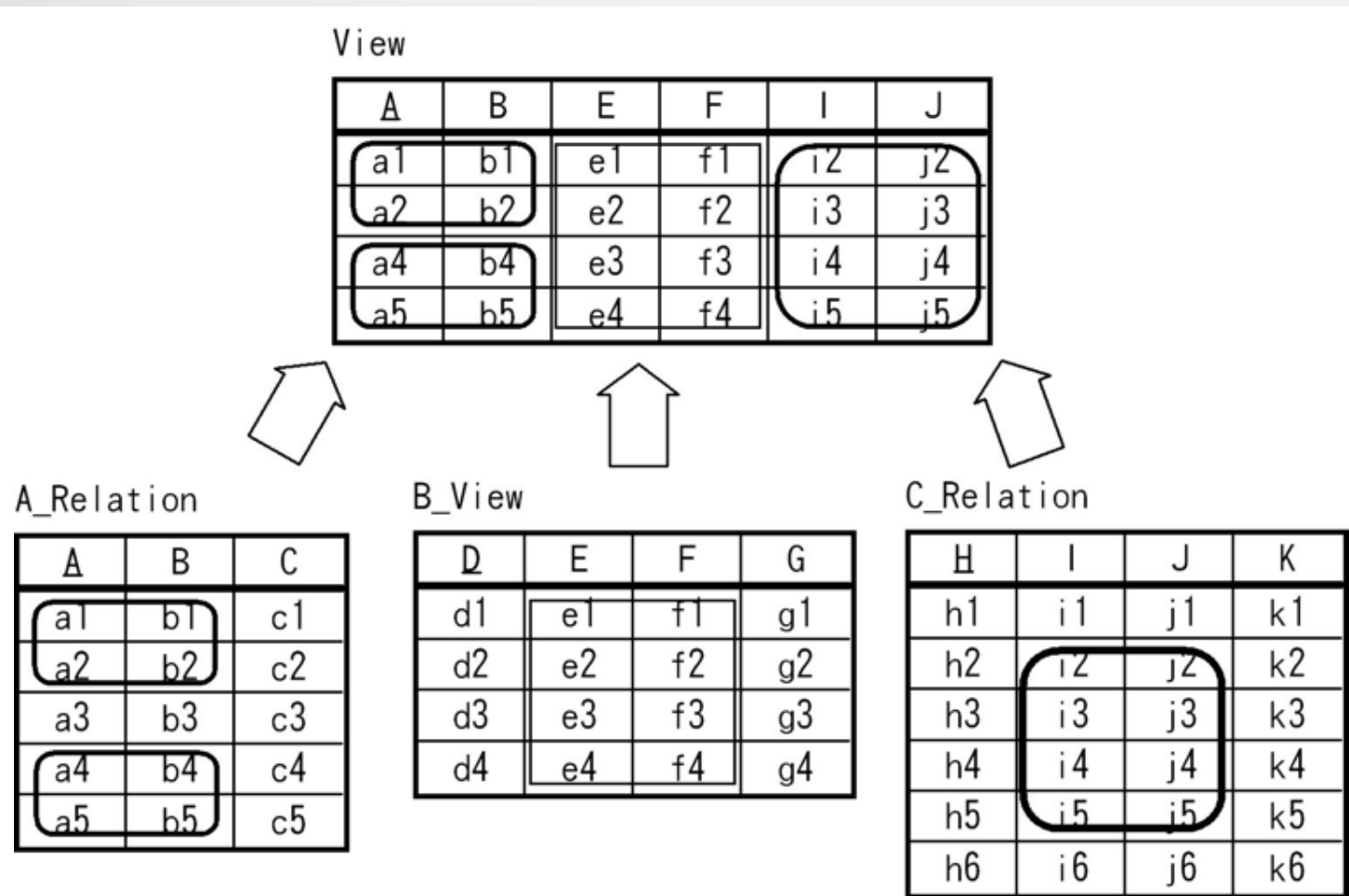


11-1-1 視界的內容-資料來源

- 視界之所以稱為虛擬關聯表，這是因為它並沒有真正將資料儲存在磁碟，而只是一些定義資料，定義從那些基底關聯表或視界挑出那些屬性或值組，**SQL**語言是使用**CREATE VIEW**指令來定義視界。
- 視界顯示的資料是從定義的基底關聯表導出，依照定義過濾掉不屬於視界的資料，如果視界是從其他視界導出，只是重複過濾一次，所以視界如同是一個從不同基底關聯表或視界抽出的資料積木，然後使用這些積木拼出所需的關聯表。



11-1-1 視界的內容-資料來源圖例





11-1-2 視界的種類

- 列欄子集視界（**Row-and-Column Subset Views**）：從單一基底關聯表或視界導出的視界，只挑選關聯表或視界中所需的屬性和值組，換句話說。
- 合併視界（**Join Views**）：使用合併查詢從多重基底關聯表或視界所導出的視界，新視界的屬性和值組是來自多個基底關聯表或視界。
- 統計摘要視界（**Statistical Summary Views**）：一種列欄子集視界或合併視界，只是使用聚合函數產生指定欄位所需的統計資料。



11-1-3 視界的優缺點-優點1

- 達成邏輯資料獨立：視界相當於外部與概念對映（**External/Conceptual Mapping**），更改基底關聯表的綱要，只需同時更改視界的外部與概念對映的定義資料，就可以讓使用者檢視相同觀點的資料，而不會影響外部綱要。
- 增加資料安全性：視界可以隱藏和過濾資料，只讓使用者看到它允許看到的資料，增加資料的安全性。
 - 例如：在**Students**關聯表擁有生日欄位**birthday**，我們可以使用視界隱藏學生生日，只讓使用者看到其他部分的學生資料。



11-1-3 視界的優缺點-優點2

- 簡化資料查詢：將常用和複雜的查詢定義成視界，就可以簡化資料查詢，因為不再需要每次重複執行複雜的**SQL**查詢指令，直接使用現成的視界即可。
- 簡化使用者觀點：視界可以增加資料的可讀性，讓資料庫使用者專注於所需的資料。
 - 例如：替欄位更名成使用者觀點的欄位名稱。



11-1-3 視界的優缺點-缺點

- 執行效率差：視界並沒有真正儲存資料，資料是在使用時才從基底關聯表導出，因為經過轉換手續，執行效率一定比不過直接存取基底關聯表。
- 更多的操作限制：在新增、更新和刪除視界資料時，為了避免違反資料庫的完整性限制條件，在操作上有更多的限制。
- 增加管理的複雜度：視界可以一層一層的從其他視界導出。
 - 例如：View2和View4是從View1導出，View3是從View2導出，複雜的視界關聯將增加管理眾多關聯表和視界的複雜度，一不小心刪錯視界，就有可能造成嚴重後果。



11-2 建立與刪除視界

- 11-2-1 建立列欄子集視界
- 11-2-2 建立合併視界
- 11-2-3 建立統計摘要視界
- 11-2-4 從其他視界建立視界





11-2 建立與刪除視界- 建立語法

- 在SQL語言建立視界是使用CREATE VIEW指令，其基本語法如下所示：

```
CREATE VIEW view_name AS  
select_statement
```

可當一個資料表來用

- 上述語法建立名為view_name的視界，資料來源是AS之後的SELECT指令敘述。視界的欄位和資料型態都是對應SELECT指令敘述的欄位，如果使用聚合函數（Aggregate Function），請使用AS指令定義別名。



11-2 建立與刪除視界- 刪除語法

- 對於資料表中不再需要的視界，SQL語言可以使用 **DROP VIEW**指令刪除視界，其基本語法如下所示：

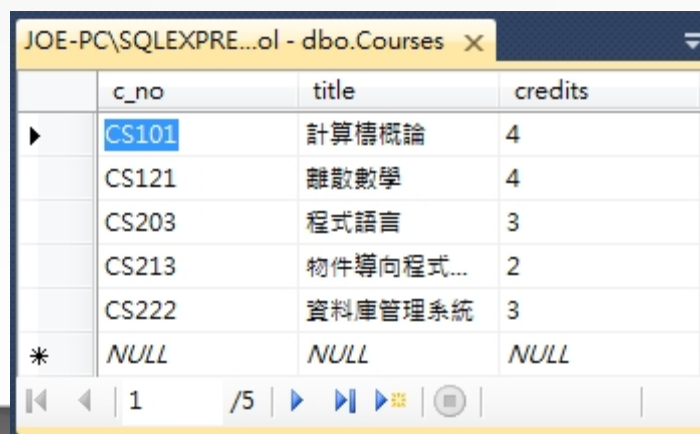
DROP VIEW view_name

- 刪除名為view_name的視界。



11-2-1 建立列欄子集視界-說明

- 列欄子集視界是指視界的內容是基底關聯表屬性集的子集合，也就是以資料表的欄位和記錄為單位，從這些欄位和記錄集合中，取出所需子集合的檢視表。
- 例如：在本節準備建立視界的基底關聯表**Courses**資料表，如下圖所示：



	c_no	title	credits
▶	CS101	計算機概論	4
	CS121	離散數學	4
	CS203	程式語言	3
	CS213	物件導向程式...	2
	CS222	資料庫管理系統	3
*	NULL	NULL	NULL



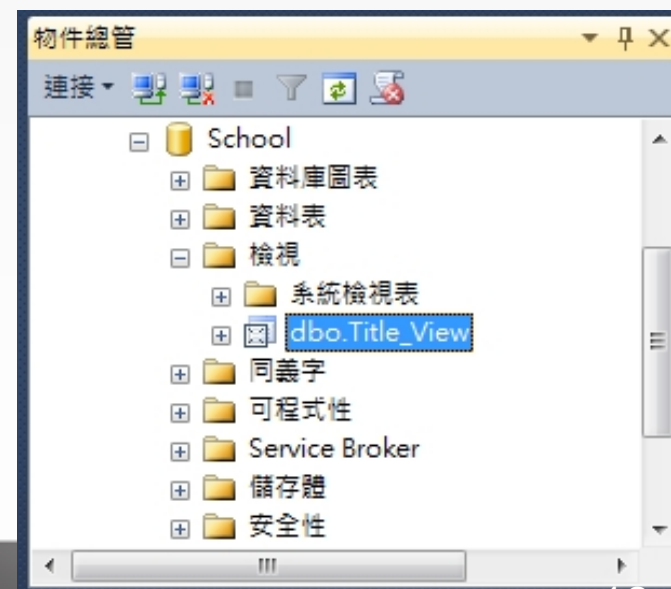
11-2-1 建立列欄子集視界- 建立欄子集視界(範例)

- 欄子集視界（Column Subset Views）是指這個視界的屬性是基底關聯表屬性集合的子集合。

SQL查詢範例：Ch11_2_1_01.sql

- 從Courses基底關聯表建立課程名稱資料的Title_View視界，如下所示：

```
CREATE VIEW Title_View AS  
SELECT c_no, title FROM Courses
```





11-2-1 建立列欄子集視界- 建立欄子集視界(結果)

SQL查詢範例：Ch11_2_1_02.sql

- 使用SELECT指令查詢Title_View視界的內容，如下所示：

SELECT * FROM Title_View

	c_no	title
1	CS101	計算機概論
2	CS121	離散數學
3	CS203	程式語言
4	CS213	物件導向...
5	CS222	資料庫管...



11-2-1 建立列欄子集視界- 建立列子集視界(範例)

- 列子集視界（Row Subset Views）是指這個視界的值組是基底關聯表值組集合的子集合。

SQL查詢範例：Ch11_2_1_04.sql

- 從Courses基底關聯表建立課程學分credits欄位大於等於3的Credits_View視界，如下所示：

```
CREATE VIEW Credits_View AS
```

```
SELECT * FROM Courses
```

```
WHERE credits >= 3
```



11-2-1 建立列欄子集視界- 建立列子集視界(結果)

SQL查詢範例：Ch11_2_1_05.sql

■ 查詢Credits_View視界的內容，如下所示：

SELECT * FROM Credits_View

	c_no	title	credits
1	CS101	計算機概論	4
2	CS121	離散數學	4
3	CS203	程式語言	3
4	CS222	資料庫管...	3



11-2-1 建立列欄子集視界- 建立列欄子集視界(範例)

- 列欄子集視界（ Row-and-Column Subset Views ）是指視界的屬性和值組都是基底關聯表屬性和值組集合的子集合。

SQL查詢範例：Ch11_2_1_06.sql

- 從Courses基底關聯表建立學分credits欄位大於等於3，而且只有c_no和title二個欄位的Major_View視界，如下所示：

```
CREATE VIEW Major_View AS  
SELECT c_no, title FROM Courses  
WHERE credits >= 3
```



11-2-1 建立列欄子集視界- 建立列欄子集視界(結果)

SQL查詢範例：Ch11_2_1_07.sql

■ 查詢Major_View視界的內容，如下所示：

SELECT * FROM Major_View

	c_no	title
1	CS101	計算機概論
2	CS121	離散數學
3	CS203	程式語言
4	CS222	資料庫管理系統



11-2-2 建立合併視界-說明

- 合併視界（**Join Views**）是多個關聯表執行合併查詢所建立的視界。



11-2-2 建立合併視界-範例

SQL查詢範例：Ch11_2_2_01.sql

- 從Students、Courses、Instructors和Classes四個資料表建立合併視界Std_Class_View，可以顯示學生的選課資料，如下所示：

```
CREATE VIEW Std_Class_View AS  
SELECT Classes.sid, Students.name, Classes.eid,  
Instructors.name AS professor,  
Classes.c_no, Courses.title, Classes.room  
FROM Students, Courses, Instructors, Classes  
WHERE Students.sid = Classes.sid  
and Courses.c_no = Classes.c_no  
and Instructors.eid = Classes.eid
```



11-2-2 建立合併視界-結果

SQL查詢範例：Ch11_2_2_02.sql

■ 查詢Std_Class_View視界的內容，如下所示：

SELECT * FROM Std_Class_View

☆ view僅存放定義，平常不存在

	sid	name	eid	professor	c_no	title	room
1	S001	陳會安	E001	陳慶新	CS101	計算機概論	180-M
2	S003	張三丰	E001	陳慶新	CS213	物件導向程式設計	622-G
3	S001	陳會安	E002	楊金權	CS222	資料庫管理系統	100-M
4	S002	江小魚	E002	楊金權	CS222	資料庫管理系統	100-M
5	S003	張三丰	E002	楊金權	CS121	離散數學	221-S
6	S004	李四方	E002	楊金權	CS222	資料庫管理系統	100-M
7	S001	陳會安	E003	李鴻章	CS203	程式語言	221-S
8	S001	陳會安	E003	李鴻章	CS213	物件導向程式設計	500-K
9	S002	江小魚	E003	李鴻章	CS203	程式語言	327-S



11-2-3 建立統計摘要視界-說明

- 統計摘要視界（**Statistical Summary Views**）屬於一種列欄子集視界或合併視界，只是使用聚合函數（**Aggregate Function**）產生指定欄位所需的統計資料。



11-2-3 建立統計摘要視界-範例1

SQL查詢範例：Ch11_2_3_01.sql

- 建立Students、Courses和Classes三個資料表的統計摘要視界Credits_View，這是一個合併視界，而且使用COUNT()和SUM()聚合函數（Aggregate Function）顯示每位學生的選課數和所修的總學分，如下所示：

```
CREATE VIEW Total_Credits_View AS
```

```
SELECT Students.sid, COUNT(*) AS numofcourses,
```

```
SUM(Courses.credits) AS credits
```

```
FROM Students, Courses, Classes
```

```
WHERE Students.sid = Classes.sid
```

```
and Courses.c_no = Classes.c_no
```

```
GROUP BY Students.sid
```

join的條件一定要有

一定會有GROUP BY



11-2-3 建立統計摘要視界-結果1

SQL查詢範例：Ch11_2_3_02.sql

■ 查詢Total_Credits_View視界的內容，如下所示：

SELECT * FROM Total_Credits_View

	sid	numofcourses	credits
1	S001	4	12
2	S002	2	6
3	S003	2	6
4	S004	1	3



11-2-3 建立統計摘要視界-範例2

SQL查詢範例：Ch11_2_3_03.sql

- 請修改統計摘要視界Total_Credits_View，建立只顯示學生所修總學分超過6個學分的學生選課總數，和學分數的合併視界Top_Credits_View，如下所示：

```
CREATE VIEW Top_Credits_View AS
SELECT Students.sid, COUNT(*) AS numofcourses,
SUM(Courses.credits) AS credits
FROM Students, Courses, Classes
WHERE Students.sid = Classes.sid
    and Courses.c_no = Classes.c_no
GROUP BY Students.sid
HAVING SUM(Courses.credits) >= 6
```



11-2-3 建立統計摘要視界-結果2

SQL查詢範例：Ch11_2_3_04.sql

■ 查詢Top_Credits_View視界的內容，如下所示：

SELECT * FROM Top_Credits_View

	sid	numofcourses	credits
1	S001	4	12
2	S002	2	6
3	S003	2	6



11-2-4 從其他視界建立視界- 說明

- 視界不只可以從基底關聯表導出，如果有已經存在的視界，我們也可以從現有視界來建立新視界。



11-2-4 從其他視界建立視界- 範例

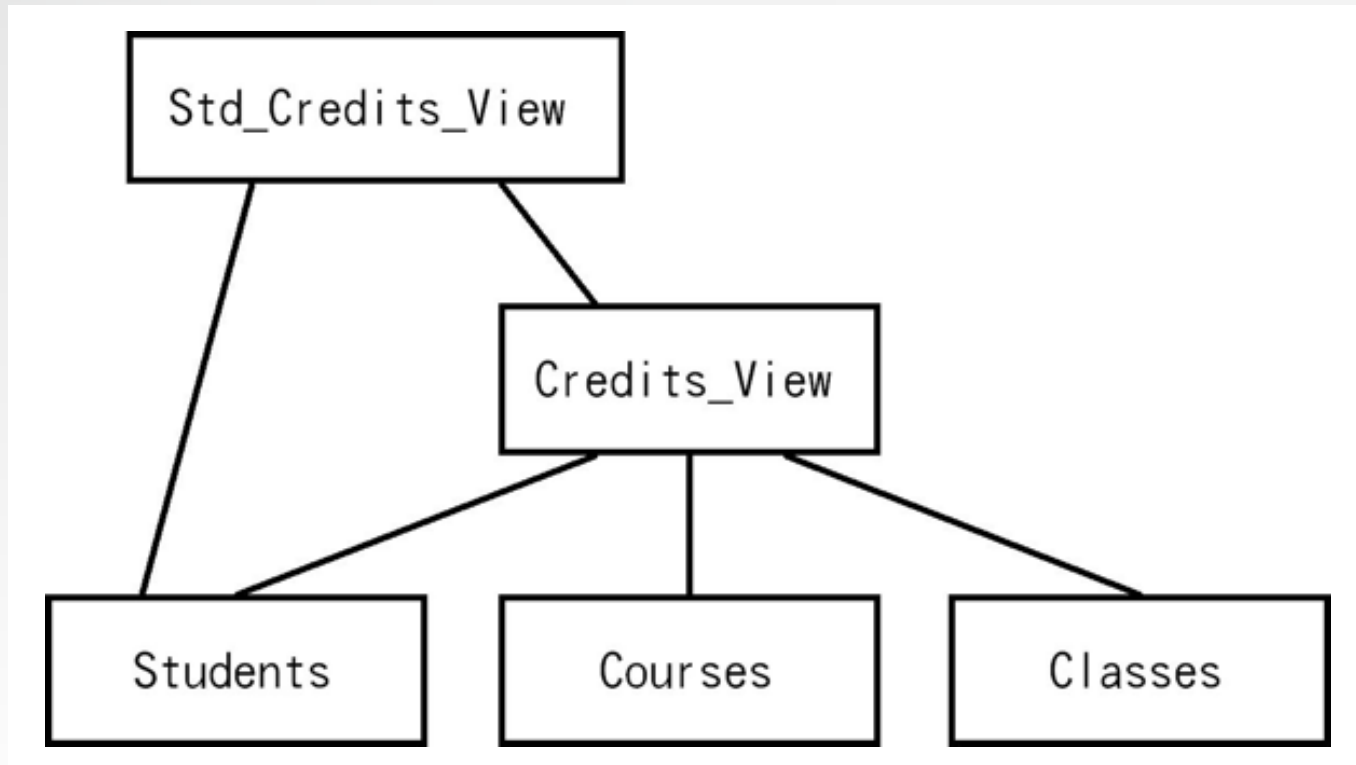
SQL查詢範例：Ch11_2_4_01.sql

- 在上一節的Total_Credits_View視界只顯示學號，我們可以再次使用此視界和Students基底關聯表，建立合併視界Std_Credits_View顯示學生姓名name和電話tel欄位的詳細資料，如下所示：

```
CREATE VIEW Std_Credits_View AS  
SELECT Total_Credits_View.*, Students.name, Students.tel  
FROM Students, Total_Credits_View  
WHERE Students.sid = Total_Credits_View.sid
```



11-2-4 從其他視界建立視界-圖例





11-2-4 從其他視界建立視界- 結果

SQL查詢範例：Ch11_2_4_02.sql

■ 查詢Std_Credits_View視界的內容，如下所示：

SELECT * FROM Std_Credits_View

	sid	numofcourses	credits	name	tel
1	S001	4	12	陳會安	02-22222222
2	S002	2	6	江小魚	03-33333333
3	S003	2	6	張三丰	04-44444444
4	S004	1	3	李四方	05-55555555



11-3 編輯視界的內容

- 11-3-1 視界編輯的基礎
- 11-3-2 從視界新增資料表的記錄
- 11-3-3 從視界更新資料表的記錄
- 11-3-4 從視界刪除資料表的記錄





11-3-1 視界編輯的基礎-限制條件

- 視界需要包含資料表的主鍵。
- 在CREATE VIEW指令的select_statement指令不可包含DISTINCT、聚合函數、GROUP BY和HAVING子句，如果有，視界就只能查詢，換句話說，統計摘要視界擁有聚合函數，所以只能查詢，而不能新增、更新和刪除記錄。
- 因為視界是從基底關聯表導出，所以新增、更新和刪除操作仍然需要遵守其來源基底關聯表的完整性限制條件。



11-3-1 視界編輯的基礎- WITH CHECK OPTION(語法)

- WITH CHECK OPTION指令是CREATE VIEW指令的選項，在建立視界時加上此選項，表示新增、更新和刪除記錄時，需要檢查完整性限制條件，如果不符合條件，就顯示錯誤訊息。
- WITH CHECK OPTION指令的基本語法如下所示：

```
CREATE VIEW view_name AS  
select_statement  
WITH CHECK OPTION
```



11-3-1 視界編輯的基礎- WITH CHECK OPTION(範例1)

SQL查詢範例：Ch11_3_1_01.sql

- 使用Students資料表建立學生生日資料的 Birthday_View視界，而且使用WITH CHECK OPTION選項指令，如下所示：

```
CREATE VIEW Birthday_View AS  
SELECT sid, name, birthday FROM Students  
WITH CHECK OPTION
```

- 上述SQL指令建立名為Birthday_View的視界，此視界滿足前述限制條件。



11-3-1 視界編輯的基礎- WITH CHECK OPTION(範例2)

SQL查詢範例：Ch11_3_1_02.sql

- 建立學生生日資料的NP_Birthday_View視界，此視界不含資料表主鍵sid，同時新增一筆測試記錄S007，如下所示：

```
CREATE VIEW Birthday_View AS  
SELECT sid, name, birthday FROM Students  
WITH CHECK OPTION  
GO  
INSERT INTO Students VALUES  
('S007','江峰','07-77777777','1965/05/23')
```

- 上述SQL指令建立的NP_Birthday_View視界因為沒有主鍵，所以不滿足前述限制條件。



11-3-2 從視界新增資料表的記錄- Birthday_View視界

SQL查詢範例：Ch11_3_2_01.sql

- 在Birthday_View視界新增一筆學生記錄，如下所示：

```
INSERT INTO Birthday_View  
VALUES ('S006', '江峰' , '1966-10-01' )
```

	sid	name	tel	birthday
1	S001	陳會安	02-22222222	1967-09-03
2	S002	江小魚	03-33333333	1978-02-02
3	S003	張三丰	04-44444444	1982-03-03
4	S004	李四方	05-55555555	1981-04-04
5	S005	陳允傑	02-22222222	1966-09-03
6	S006	江峰	NULL	1966-10-01
7	S007	江峰	07-77777777	1965-05-23

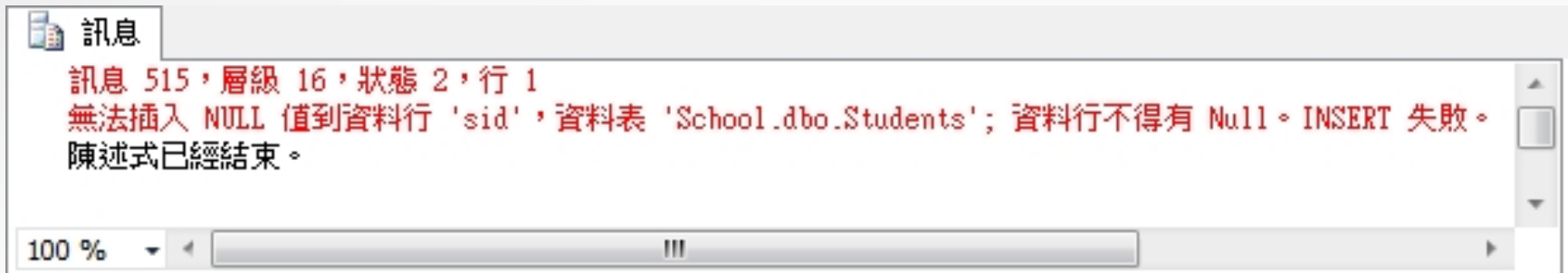


11-3-2 從視界新增資料表的記錄- No_Birthday_View視界

SQL查詢範例：Ch11_3_2_03.sql

- 在No_Birthday_View視界新增一筆學生記錄，如下所示：

```
INSERT INTO NP_Birthday_View  
VALUES ('江峰年' , '1966-10-01' )
```





11-3-3 從視界更新資料表的記錄- Birthday_View視界

SQL查詢範例：Ch11_3_3_01.sql

- 在Birthday_View視界將學號S006學生的birthday改為1966-02-01，如下所示：

UPDATE Birthday_View

SET birthday = '1966-02-01' WHERE sid = 'S006'

	sid	name	tel	birthday
1	S001	陳會安	02-22222222	1967-09-03
2	S002	江小魚	03-33333333	1978-02-02
3	S003	張三丰	04-44444444	1982-03-03
4	S004	李四方	05-55555555	1981-04-04
5	S005	陳允傑	02-22222222	1966-09-03
6	S006	江峰	NULL	1966-02-01
7	S007	江峰	07-77777777	1965-05-23



11-3-3 從視界更新資料表的記錄- No_Birthday_View視界

SQL查詢範例：Ch11_3_3_03.sql

- 在NP_Birthday_View視界將學生姓名'江峰'的 birthday 改為1966-02-01，如下所示：

```
UPDATE NP_Birthday_View
```

```
SET birthday = '1966-02-01' WHERE name = '江峰'
```

- 當執行上述SQL指令，SQL Server仍然會更新 Students 資料表的記錄S006和S007。理論上，資料庫管理系統應該避免在沒有主鍵的視界執行更新操作。



11-3-4 從視界刪除資料表的記錄- Birthday_View視界

SQL查詢範例：Ch11_3_4_01.sql

- 在Birthday_View視界刪除學號S006的學生資料，如下所示：

```
DELETE FROM Birthday_View  
WHERE sid = 'S006'
```

	sid	name	tel	birthday
1	S001	陳會安	02-22222222	1967-09-03
2	S002	江小魚	03-33333333	1978-02-02
3	S003	張三丰	04-44444444	1982-03-03
4	S004	李四方	05-55555555	1981-04-04
5	S005	陳允傑	02-22222222	1966-09-03
6	S007	江峰	07-77777777	1966-02-01



11-3-4 從視界刪除資料表的記錄- No_Birthday_View視界

SQL查詢範例：Ch11_3_4_03.sql

- 在NP_Birthday_View視界刪除學生江峰，如下所示：

```
DELETE NP_Birthday_View  
WHERE name = '江峰'
```

- 當執行上述SQL指令，SQL Server仍然會刪除Students資料表的記錄S007。理論上，資料庫管理系統應該避免在沒有主鍵的視界執行刪除操作。



資料庫程式設計部分省略