

資料庫系統筆記 Ch. 5

參考書籍：

- 1. 《Databases System 7th Edition》— Ramez Elmasri, Shamkant B. Navathe.
- 2. 《數據庫系統基礎 第六版》— Ramez Elmasri, Shamkant B. Navathe，李翔鷹、劉鑛、邱海艷、陳立軍譯

筆記作者：[黃漢軒](#)

值域

值域（Domain）即為原子（Atom）值的集合，原子值的概念即為不可分割的值。

- 定義一個值域的方式通常是給出一個資料類型。
 - 例如我們可以將 age 定義為一個介於 0 到 100 的值，其值域即為 0 - 100，其值即為原子（因為不能被分割）

關聯與關聯模式

定義關聯

我們可以用以下的格式來定義關聯模式，即為 $R(A_1, A_2, \dots, A_n)$ ，其中 R 為關聯名（Relation Name），且 A_1, A_2, \dots, A_n 為屬性。

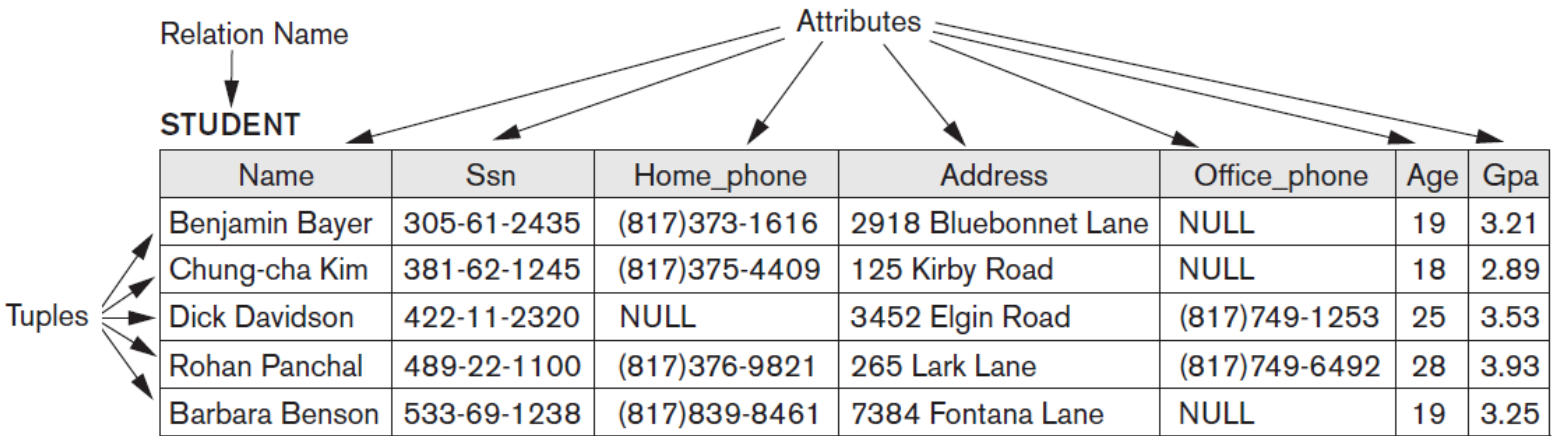
- 每個屬性（Attribute）都有他們的值域，例如 A_1 的值域即為 $\text{dom}(A_1)$

定義關聯模式

我們可以使用 $r(R)$ 來定義 R 的關聯，即為 $r = \{t_1, t_2, \dots, t_m\}$ ，其中每一個 t 即為一個元組（tuple）。

- 每個元組都有 n 個值，對應到每個屬性 A_i ，以 $t(A_i)$ 來描述對應到屬性 A_i 的值，其值屬於其值域或者 NULL。

概念上來說，他們很像一張表。



關聯的特性

元組的排序

數學上來說，元組是無序的。

但基於儲存在磁碟的關聯，在不指定排序方式的情況下，他們的排序可能會按照某種方式進行排序（例如新增時間）。

同時，我們也可以根據屬性的值升序與降序，例如上表可以根據名子或年紀來進行排序。

元組中的數值排序

我們必須保持屬性與數值的排序，這樣才能定義數值應該要映射在哪個屬性上。

但若我們可以使用另一種定義，也就是在元組中，使用對（Pair）來定義值與屬性的關聯，那這時候元組中的數值排序就不大重要了。

例如我們假設有 n 個屬性，若我們要插入一組元組，則我們定義數值為 $\langle t_1, t_2, t_3, \dots, t_n \rangle$ ，分別對應到 n 個屬性。

但若我們可以用以下的方式來定義元組的數值排序，也就是 $\langle (a_3, t_3), (a_1, t_1), (a_2, t_2) \dots (a_8, t_8) \rangle$ ，

這樣我們可以賦予每個值對應到的屬性，此時元組中的數值該怎麼排序就不太重要了。

元組中的值與 NULL

- 元組中的每個值都是原子值，多值屬性在關聯模型中只能拆成多個屬性來存放數值，為了保持元組中值是原子值的概念。
- NULL 值通常表示這個值不適用／未知的情況，
例如一個學生的家用電話號碼存放在資料庫中可能是 NULL，因為該學生沒有家用電話。

關聯的解釋

- 關聯模式與元組上的解釋
 - 關聯模式可以當成一種斷言（Assertion），例如我們考慮一個學生會有學號與名稱， $\text{Student}(\text{Name}, \text{StudentID})$ 即為一個關聯模式，其 Name 與 StudentID 即為這個學生的屬性。
 - 我們可以把元組考慮成一個事實，例如一個學生叫做 Uriah，學號是 109590031，那麼他的事實就是名子 Uriah (Name=Uriah)，且學號為 109590031 (StudentID=109590031)。

關聯模式的表示方法

- 有 n 個屬性的關聯模式 R ，表示為 $R(A_1, A_2, \dots, A_n)$ 。
- 大小字母 Q 、 R 、 S 代表關聯名。
- 小寫字母 q 、 r 、 s 代表關聯狀態。
- 字母 t 、 u 、 v 代表元組。
- 使用關聯名來限定屬性，例如 Student.Name 代表 Student 關聯底下的 Name 屬性。
- 關聯 $r(R)$ 的 n 元元組 t 表示成 $t = \langle v_1, v_2, \dots, v_n \rangle$ ，其中 v_i 對應到 A_i 的值。
- $t[A_i]$ 對應到 t 上 A_i 的值的值，例如 $t = \langle \text{Uriah}, 19 \rangle$ 時， $t[\text{Name}] = \text{Uriah}$ 且 $t[\text{Age}] = 19$ 。
- $t[A_u, A_w, \dots, A_z]$ 對應到 t 上 A_u, A_w, \dots, A_z 的值，例如 $t[\text{Name}, \text{Age}] = \langle \text{Uriah}, 19 \rangle$ 。

關聯模型約束與關聯資料庫模式

- 資料庫約束主要可以分成以下三類：
 - 基於固有模型的約束（Inherent model-based constraint），或稱為隱式約束。
 - 基於模式的約束（Schema-based constraint），或稱為顯式約束。
 - 不能在資料模型上直接表示的約束，通常稱為基於應用的約束（Application-based constraint）。
- 我們使用依賴來評測資料庫的好壞，分別分為：
 - 資料依賴（Data dependency）
 - 函數依賴（Function dependency）
 - 多值依賴（Multivalue dependency）
- 依賴也會在規範化（Normalization）的過程中使用。
- 基於模式的約束大概分為以下幾類：
 - 值域約束：需要符合屬性上對於值域的約束，例如數值只能放數值，不能放字串。
 - 鍵約束
 - NULL 值約束：是否允許 NULL，不允許則需要加上 NOT NULL 的限制。
 - 類型完整性約束
 - 參考完整性約束

鍵約束

- 考慮一個關聯上，我們使用一個屬性的子集合，來唯一化所有的元組，稱為關聯鍵（Super Key）
- 比關聯鍵還要來得嚴格且好用的稱為鍵（Key），
鍵沒有冗餘的屬性，且必須要符合最小關聯鍵（Mininum Superkey），以及任意兩個元組中，他們的鍵不能重複。
- 一般情況下，關係模式可以有多個鍵，每個鍵稱為候選鍵（Candidate Key），從候選鍵選出一個主鍵（Primary Key）。
其餘的候選鍵可以冠上唯一的特性，也就是唯一鍵（Unique Key）
- 由上面幾種鍵，我們就可以約束資料的唯一性。

關聯資料庫與關聯資料庫模式

關聯資料庫模式（Relational Database Schema）是關聯模式的集合，如下圖陳列了許多的屬性。

其中畫底線的屬性，代表該屬性是主鍵。

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

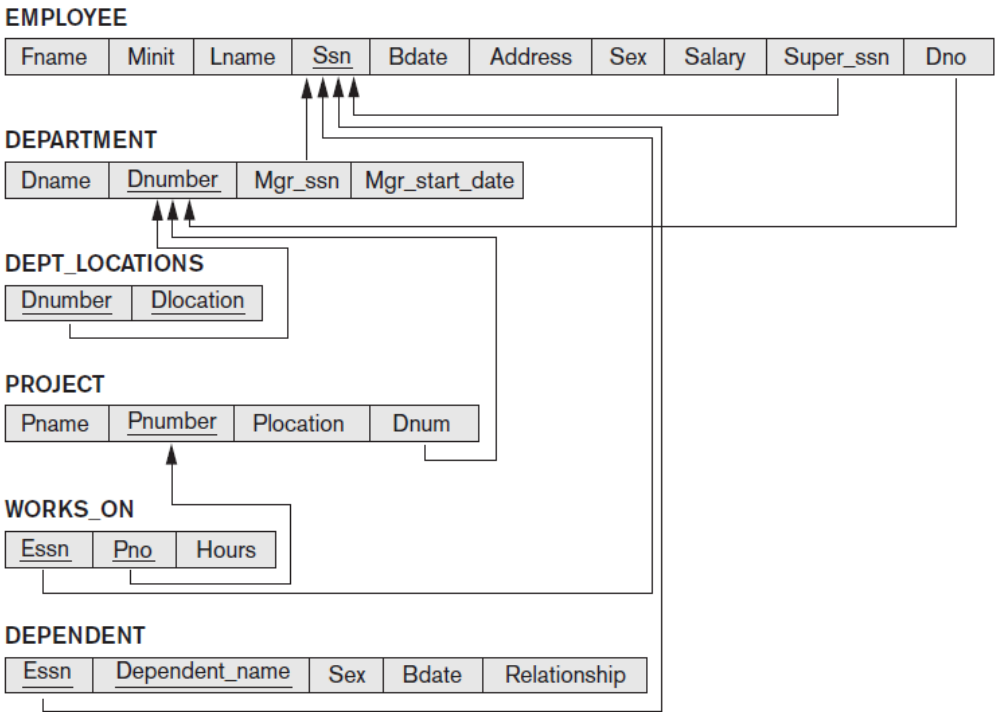
<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Figure 5.5
Schema diagram for the
COMPANY relational
database schema.

類型完整性約束、參考完整性約束與外來鍵

- 類型完整性約束規定主鍵不能是 NULL，否則不能由其他關聯參考。
- 參考完整性約束規定兩個關係之間的一致性，我們可以用外來鍵（Foreign Key）來指定兩個關係之間的一致性。
 - 概念上即為某張表 t_1 的 **FK** 存在於另一張表 t_2 的 **PK**，則滿足參考完整性約束，滿足 t_1 參考了 t_2 的元組。

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



更新操作與處理違例約束

檢索（Retrieve）用來查詢資料，更新（Update）用來更新資料，以下討論更新操作會造成違反的例子。

在關聯資料庫中，更新主要有三個動作：插入（Insert）操作、刪除（Delete）操作與更新（Update）操作。

以下的範例以這個資料表來示範。

EMPLOYEE									
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS	
Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON		
Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT			
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT				
Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

插入操作

- 插入 `<'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4>` 至 `EMPLOYEE` 内
 - 造成的問題：類型完整性約束，主鍵不應為 NULL。
 - 修復的方式：給予一組符合值域約束的值。
- 插入 `<'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4>` 至 `EMPLOYEE` 内
 - 造成的問題：鍵約束，`Ssn` 這個鍵已存在 `'999887777'` 這個值。
 - 修復的方式：可以改成另一個新的 `Ssn` 值。
- 插入 `<'Cecilia', 'J', 'Kolonsky', '677678989', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 7>` 至 `EMPLOYEE` 内
 - 造成的問題：參考完整性約束，沒有 7 這個值在 `DEPARTMENT` 内。
 - 修復的方式：插入新值到 `DEPARTMENT` 内，或者將 7 改成 `DEPARTMENT` 内現有的值。
- 插入 `<'Cecilia', 'J', 'Kolonsky', '677678989', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 7>` 至 `EMPLOYEE` 内
 - 這個動作符合所有約束，故可以成功插入。

在插入動作破壞約束時，默認是拒絕（Reject）該操作，但你也可以透過修復來讓該操作符合所有約束。

刪除操作

- 刪除 `WORKS_ON` 表內的 `Essn='999887777'` 並且 `Pno=10` 的元組
 - 這個動作符合所有約束，故可以成功刪除。
- 刪除 `EMPLOYEE` 表內的 `Ssn='999887777'` 的元組
 - 這個動作不符合參考約束，在刪除時因為 `WORKS_ON` 關聯的關係，故會造成參考約束的問題。
 - 修復的方式：將所有遞迴的元組全部刪除。
- 刪除 `EMPLOYEE` 表內的 `Ssn='3333445555'` 的元組
 - 這個動作不符合參考約束，在刪除時因為 `WORKS_ON`、`DEPARTMENT` 等表關聯的關係，故會造成更嚴重的參考完整性約束。
 - 修復的方式：將所有遞迴的元組全部刪除。

在刪除動作破壞約束時，默認是拒絕（Reject）該操作，但你也可以透過修復來讓該操作符合所有約束。

更新操作

- 更新 `EMPLOYEE` 表內的 `Ssn='999887777'` 的元組從 `Dno` 的 1 更新至 7
 - 這個動作不符合參考完整性約束。
- 更新 `EMPLOYEE` 表內的 `Ssn='999887777'` 的元組的 `Ssn` 從 `9999887777` 更新至 `987654321`
 - 這個動作不符合參考鍵約束，且同時破壞參考完整性約束。

交易的概念

基本上，應用程式都會運行著一個或多個交易，交易的概念即為在資料庫上做一系列的操作，例如更新資料、刪除資料或插入資料。

交易的最終目的是在最終結果時，能讓資料庫保持有效狀態。