

ชนิดข้อมูลที่ซับซ้อนและแนวทางเชิงวัตถุ

ในหัวข้อนี้จะกล่าวถึงส่วนขยายของคำสั่ง SQL ที่ยอมให้มีชนิดข้อมูลที่ซับซ้อน รวมถึงการรีเลชันเชิงกลุ่ม และรูปแบบเชิงวัตถุ โดยรูปแบบของคำสั่งจะอ้างอิงมาตรฐาน SQL-3 ซึ่งคำสั่งจะอยู่ในรูปแบบคำสั่งของ Illustra database system โดย Illustra นี้เป็นเวอร์ชันทางการค้าของ Postgres database system ที่มีการพัฒนาที่มหาวิทยาลัยแห่งแคลิฟอร์เนีย ที่ Berkley ซึ่งเป็นระบบจัดการฐานข้อมูลที่ขยายความสามารถเชิงวัตถุเข้าไปในโมเดลเชิงสัมพันธ์

1. Structured and collection Types

พิจารณาคำสั่งต่อไปนี้ เป็นการกำหนดรีเลชัน employee ด้วยแอตทริบิวต์เชิงซ้อน

```
create type mystring char varying
```

```
create type mytraining
```

```
(tno integer,
```

```
tdate date)
```

```
create type emp
```

```
(empno as integer,
```

```
name as mystring,
```

```
children setof(mystring),
```

```
training setof(mytraining))
```

```
create table employee of type emp
```

คำสั่งแรกเป็นการกำหนดชนิดข้อมูล mystring ซึ่งเป็นชนิดสตริงแบบ variable length character คำสั่งที่สองเป็นการกำหนดชนิดข้อมูล mytraining ซึ่งประกอบไปด้วย tno และ tdate และคำสั่งที่สามเป็นการกำหนดชนิดข้อมูล emp ซึ่งประกอบไปด้วย empno name กลุ่มของ children และ กลุ่มของ training และคำสั่งสุดท้ายเป็นการสร้างตาราง employee ซึ่งจากคำสั่งดังกล่าว จะพบว่ามีแตกต่างจาก การสร้างตารางในระบบฐานข้อมูลเชิงสัมพันธ์ เนื่องจาก เรายอมให้แอตทริบิวต์มีลักษณะเป็นเซตได้ และด้วยคุณสมบัตินี้จะยอมให้เราสามารถ

กำหนดชนิดของข้อมูลเป็นแอตทริบิวต์แบบผสม(composite attribute) และแอตทริบิวต์แบบหลายค่า(multivalued attribute)ได้โดยตรง

เราสามารถสร้างตาราง employee ได้โดยตรง โดยไม่ต้องสร้างชนิดข้อมูลของ emp ขึ้นมาก่อนก็ได้

```
create table employee  
(empno as integer,  
name as mystring,  
childname setof(mystring),  
training setof(mytraining))
```

โดยทั่วไปแล้ว ลักษณะชนิดข้อมูลเชิงซ้อนนี้ จะสนับสนุนชนิดข้อมูลแบบคอลเลกชัน (collection : คือข้อมูลที่ไม่มีลำดับ และสามารถมีข้อมูลได้หลาย ๆ ค่าข้อมูล) เช่นอะเรย์(array) และมัลติเซต(multiset) ตัวอย่างเช่น สมมุติว่าเราเพิ่มแอตทริบิวต์ telephone ให้กับรีเลชัน employee ซึ่งพนักงานแต่ละคนสามารถมีโทรศัพท์ได้หลายหมายเลข เราจะเพิ่มแอตทริบิวต์ telephone ดังนี้

```
telephone mystring[3]
```

ซึ่งหมายถึงอะเรย์ของหมายเลขโทรศัพท์ของพนักงานแต่ละคนนั่นเอง ซึ่งถ้าเราต้องการทราบว่าหมายเลขโทรศัพท์หมายเลขที่หนึ่งคือหมายเลขอะไร ก็สามารถบอกได้ทันที เพราะมีการอ้างถึงข้อมูลแบบอะเรย์ แต่ถ้าเรากำหนดข้อมูลโทรศัพท์นี้เป็น setof(mystring) เราจะไม่สามารถบอกได้เลยว่าหมายเลขโทรศัพท์ใดเป็นลำดับที่เท่าไร

2. Inheritance

การสืบทอดก็สามารถดำเนินการกับชนิดของข้อมูลได้เช่นกัน ตัวอย่างเช่น สมมุติว่าเราได้กำหนดชนิดข้อมูลของบุคคลดังนี้

```
create type person  
( name mystring,
```

social-security integer)

จากนั้นเราต้องการเก็บข้อมูลเกี่ยวกับนักเรียน และครู ซึ่งทั้งนักเรียนและครู ต่างก็เป็นบุคคลด้วยกันทั้งคู่ ดังนั้นเราสามารถใช้อุณหภูมิการสืบทอดมากำหนดชนิดข้อมูลนักเรียน และครูได้ดังนี้

create type student

(degree mystring,

department mystring)

under person

create type teacher

(salary integer,

department mystring)

under person

ทั้งนักเรียนและครูต่างสืบทอดคุณสมบัติของ person คือ name และ social-security กล่าวได้ว่า student และ teacher เป็น subtype ของ person และ person เป็น supertype ของ student และ teacher

สมมุติว่า เราต้องการเก็บข้อมูลของผู้ช่วยสอน(teacher assistant) ซึ่งเป็นทั้งนักเรียนและครูในเวลาเดียวกัน และอาจจะอยู่ต่างแผนกกัน แนวคิดนี้เรียกว่า multiple inherit ถ้าระบบสนับสนุน multiple inherit เราสามารถกำหนดชนิดข้อมูล teaching assistant ได้ดังนี้

create type teachingassistant

under student, teacher

teachingassistant จะสืบทอดคุณสมบัติทั้งหมดของ student และ teacher จะพบว่าแอตทริบิวต์ social-security name และ department ปรากฏอยู่ในทั้ง student และ teacher ซึ่งแอตทริบิวต์ social-security และ name ได้ถูกสืบทอดมาจาก person เหมือนกัน ดังนั้นจึงไม่เกิดความขัดแย้งจากการสืบทอด

ทอดจาก student และ teacher แต่ แอดทริบิวต์ department อาจเกิดความขัดแย้งได้เนื่องจาก teaching assistant อาจจะเป็นนักเรียนในแผนกหนึ่ง แต่อาจารย์จะอยู่อีกแผนกหนึ่ง ดังนั้นในการหลีกเลี่ยงความขัดแย้งนี้ เราจะใช้วิธีการเปลี่ยนชื่อ ดังนี้

create type teachingassistant

under student with (department as student-department) ,

teacher with (department as teacher-department)

3. Reference Type

ภาษาเชิงวัตถุสนับสนุนการอ้างอิงอ็อบเจกต์ โดยที่แอดทริบิวต์หนึ่ง ๆ ของ ชนิดข้อมูลสามารถที่จะอ้างอิงไปอ็อบเจกต์หนึ่งได้ ตัวอย่างเช่นการอ้างอิงไปที่รีเลชัน person ของแอดทริบิวต์ childname สามารถเขียนใหม่ได้ดังนี้

childname setof(ref(person))

หมายถึงแอดทริบิวต์เป็นเซตของการอ้างอิงไปที่อ็อบเจกต์ person เราสามารถจัดการกับการอ้างอิงไปที่ทูเปิลของรีเลชัน person โดยการใช้คีย์หลักของรีเลชัน person มาเป็นข้อมูลใน childname เพื่อทำหน้าที่ foreign key หรืออีกวิธีหนึ่งคือ ใช้ tuple identifier ของแต่ละทูเปิลเป็นตัวอ้างอิงแทน

ที่มา <http://www.srisangworn.go.th/home/databaselearnx/ms2t2-12.htm>