

ภาคผนวก ก

คำสั่งที่ใช้ในระบบ

คำสั่งที่ใช้ในระบบส่วนใหญ่จะเป็นคำสั่งเกี่ยวกับการจัดการข้อมูลของฐานข้อมูลในระบบ ผู้ใช้ที่ต้องการใช้งานระบบจะต้องพิมพ์คำสั่งตามรูปแบบคำสั่งที่กำหนดในระบบ

ก.1 รูปแบบคำสั่ง

ในส่วนนี้จะเป็นการอธิบายรูปแบบคำสั่งที่ใช้ในระบบทั้งหมด ซึ่งมีข้อกำหนดในการอธิบายรูปแบบคำสั่งดังนี้

อักษรตัวพิมพ์ใหญ่ หมายถึง คำหลักของคำสั่งที่ผู้ใช้ต้องพิมพ์ตามนั้น แต่จะพิมพ์ด้วยอักษรตัวพิมพ์เล็กหรืออักษรตัวพิมพ์ใหญ่ก็ได้

เครื่องหมาย [] หมายถึง ตัวเลือก (option) ที่ผู้ใช้อาจกำหนดหรือไม่กำหนดสิ่งที่อยู่ในเครื่องหมาย [] ก็ได้

เครื่องหมาย { } หมายถึง การบังคับให้ผู้ใช้กำหนดอย่างใดอย่างหนึ่งในสิ่งที่ระบุอยู่ในเครื่องหมาย { }

เครื่องหมาย ... หมายถึง ผู้ใช้สามารถระบุสิ่งที่อยู่หน้าเครื่องหมาย ... ซ้ำได้

เครื่องหมาย <> หมายถึง ตัวชี้เฉพาะที่ผู้ใช้ต้องระบุในคำสั่ง อาจเป็นชื่อของฐานข้อมูล รีเลชัน หรือแอตทริบิวต์ที่ผู้ใช้ต้องการ

1. คำสั่งเกี่ยวกับการกำหนดโครงสร้างฐานข้อมูลและชนิดข้อมูลที่จัดเก็บในฐานข้อมูล แบ่งเป็นสองส่วนคือ

1) คำสั่งเกี่ยวกับฐานข้อมูล ประกอบด้วยคำสั่งต่อไปนี้

- การสร้างฐานข้อมูลใหม่ในระบบ
- การแสดงรายชื่อฐานข้อมูล
- การลบฐานข้อมูลออกจากระบบ
- การระบุชื่อฐานข้อมูลที่ต้องการใช้งาน
- การยกเลิกการใช้งานฐานข้อมูลที่คำสั่งใช้งานอยู่

ความหมายและรูปแบบของแต่ละคำสั่งเกี่ยวกับฐานข้อมูลมีดังนี้

คำสั่ง CREATE DATABASE

ความหมาย : สร้างฐานข้อมูลใหม่ในระบบ

รูปแบบ : CREATE DATABASE <database_name>

พารามิเตอร์ : database_name คือชื่อของฐานข้อมูล

คำสั่ง DROP DATABASE

ความหมาย : ลบฐานข้อมูลออกจากระบบ

รูปแบบ : DROP DATABASE <database_name>

พารามิเตอร์ : database_name คือชื่อของฐานข้อมูล

คำสั่ง SHOW DATABASE

ความหมาย : แสดงรายชื่อฐานข้อมูลทั้งหมดในระบบ

รูปแบบ : SHOW DATABASE

คำสั่ง USE

ความหมาย : ระบุชื่อฐานข้อมูลที่ต้องการใช้งาน

รูปแบบ : USE <database_name>

พารามิเตอร์ : database_name คือชื่อของฐานข้อมูล

คำสั่ง CLOSE

ความหมาย : ยกเลิกการใช้งานฐานข้อมูลที่กำลังใช้งานอยู่

รูปแบบ : CLOSE

คำสั่งเกี่ยวกับรีเลชัน ประกอบด้วยคำสั่งต่อไปนี้

- การสร้างรีเลชันใหม่
- การลบรีเลชันออกจากฐานข้อมูล
- การแสดงรายชื่อรีเลชันทั้งหมดในฐานข้อมูล
- การแสดงโครงสร้างของรีเลชัน

ความหมายและรูปแบบของแต่ละคำสั่งเกี่ยวกับรีเลชันมีดังนี้

คำสั่ง CREATE TABLE

ความหมาย : สร้างรีเลชันใหม่ในฐานข้อมูลที่กำลังใช้งานอยู่

รูปแบบ : CREATE TABLE <table_name>

```
({<column_name> { [ CHAR (length)] | [ NUM (prec [,scale ])] [DATE] }
    [DEFAULT Value_element]
    [{PRIMARY KEY | SECONDARY KEY}] [NOT NULL]
    | Table_constraint}
[, {<column_name> Datatype [DEFAULT value_element]
    [{PRIMARY KEY | SECONDARY KEY}] [NOT NULL]
    | Table_constraint}]... )
```

โดยที่ Table_constraint เป็นการกำหนดเงื่อนไขบังคับค่าแอตทริบิว มีรูปแบบดังนี้

```
[{PRIMARY KEY | SECONDARY KEY}
    (<column_name> [,<column_name>] ... )
[FOREIGN KEY (<column_name> [,<column_name> ] ... )
    REFERENCES <table_name> (<column_name> [,column_name] ... )
    [ON DELETE] ]
```

พารามิเตอร์ : CHAR	เป็นการกำหนดชนิดข้อมูลของค่าแอตทริบิวเป็นอักขระ ขนาดความยาวเท่ากับ length
NUM	เป็นการกำหนดชนิดข้อมูลของค่าแอตทริบิวเป็นตัวเลข มีตำแหน่งเลขหน้าจุดทศนิยมเท่ากับ prec และตำแหน่ง เลขหลังจุดทศนิยมเท่ากับ scale
DATE	เป็นการกำหนดชนิดข้อมูลของแอตทริบิวเป็นวันที่
DEFAULT	เป็นการกำหนดค่าโดยปริยายให้แอตทริบิว
PRIMARY KEY	เป็นการกำหนดให้แอตทริบิวเป็นคีย์หลัก
SECONDARY KEY	เป็นการกำหนดให้แอตทริบิวเป็นคีย์รอง
NOT NULL	เป็นการกำหนดว่าไม่อนุญาตให้ค่าแอตทริบิวเป็นค่าว่าง
FOREIGN KEY	เป็นการกำหนดเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิวที่ กำหนดเป็นคีย์นอกอ้างอิงค่าแอตทริบิวจากคีย์หลักใน อีกรีเลชันหนึ่ง

REFERENCES	เป็นการกำหนดถึงแอตทริบิวที่เป็นคีย์หลักของรีเลชันที่ถูกอ้างอิงค่าแอตทริบิวจากการกำหนดคีย์นอก
ON DELETE	เป็นการกำหนดเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิวแบบลบต่อเนื่องกัน
table_name	คือชื่อของรีเลชัน
column_name	คือชื่อของแอตทริบิว
value_element	คือค่าของแอตทริบิว อาจเป็นข้อมูลที่เป็นอักขระ เลขจำนวนเต็ม เลขทศนิยม หรือวันที่ในรูปแบบ “วัน/เดือน/ปี”

คำสั่ง DROP TABLE

ความหมาย : ลบรีเลชันเก่าออกจากฐานข้อมูลที่กำลังใช้งานอยู่

รูปแบบ : DROP TABLE <table_name>

พารามิเตอร์ : table_name คือชื่อรีเลชัน

คำสั่ง SHOW TABLE

ความหมาย : แสดงรายชื่อรีเลชันทั้งหมดในฐานข้อมูลที่กำลังใช้งานอยู่

รูปแบบ : SHOW TABLE

คำสั่ง DESCRIBE

ความหมาย : แสดงโครงสร้างของรีเลชันที่ต้องการ

รูปแบบ : DESC[RIBE] <table_name>

พารามิเตอร์ : table_name คือชื่อของรีเลชัน

2. คำสั่งเกี่ยวกับข้อมูลในฐานข้อมูล ประกอบด้วยคำสั่งต่อไปนี้

- การเพิ่มทูปเปลใหม่ในรีเลชัน
- การปรับปรุงทูปเปลเก่าในรีเลชัน
- การลบทูปเปลเก่าในรีเลชัน
- การแสดงทูปเปลในรีเลชัน

ความหมายและรูปแบบของแต่ละคำสั่งเกี่ยวกับข้อมูลในฐานข้อมูลมีดังนี้

คำสั่ง INSERT INTO

ความหมาย : เพิ่มทึ่เปิดใหม่ในรีเลชันที่ต้องการ

รูปแบบ : INSERT INTO <table_name> [(<column_name> [, <column_name>] ...)
VALUES (value_element [,value_element] ...)

พารามิเตอร์ : table_name คือชื่อของรีเลชัน

column_name คือชื่อของแอตทริบิว

VALUES เป็นการกำหนดค่าให้แอตทริบิว

value_element คือค่าของแอตทริบิว อาจเป็นข้อมูลที่เป็นอักขระ เลข
จำนวนเต็ม เลขทศนิยม หรือวันที่ในรูปแบบ “วัน/เดือน/
ปี”

คำสั่ง UPDATE

ความหมาย : ปรับปรุงทึ่เปิดเก่าในรีเลชันที่ต้องการ

รูปแบบ : UPDATE <table_name>
SET <column_name> = value_element
[, <column_name> = value_element] ...
[WHERE [<column_name> { = | < | > | >= | != } value_element]
| [<column_name> BETWEEN value_element AND value_element]
| [<column_name> IN (value_element [,value_element] ...) }])

พารามิเตอร์ : SET เป็นการกำหนดค่าแอตทริบิว

WHERE เป็นการกำหนดเงื่อนไขในการค้นหาทึ่เปิดของรีเลชัน

BETWEEN AND เป็นการกำหนดการตรวจสอบค่าแอตทริบิวด้วยฟังก์ชัน
between and

IN เป็นการกำหนดการตรวจสอบค่าแอตทริบิวด้วยฟังก์ชัน
in

table_name คือชื่อของรีเลชัน

column_name คือชื่อของแอตทริบิว

value_element คือค่าของแอตทริบิว อาจเป็นข้อมูลที่เป็นอักขระ เลข
จำนวนเต็ม เลขทศนิยม หรือวันที่ในรูปแบบ “วัน/เดือน/
ปี”

คำสั่ง DELETE

ความหมาย : ลบทุกเปิดเก่าในรีเลชันที่ต้องการ

รูปแบบ : DELETE FROM <table_name>

[WHERE [<column_name> {= | < | <= | > | >= | != } value_element]

| [<column_name> BETWEEN value_element AND value_element]

| [<column_name> IN (value_element [,value_element] ...)] }]]

พารามิเตอร์ : table_name คือชื่อของรีเลชัน

column_name คือชื่อของแอตทริบิว

value_element คือค่าของแอตทริบิว อาจเป็นข้อมูลที่เป็นอักขระ เลข จำนวนเต็ม เลขทศนิยม หรือวันที่ในรูปแบบ “วัน/เดือน/ปี”

คำสั่ง SELECT

ความหมาย : แสดงทุกเปิดในรีเลชันที่ต้องการ

รูปแบบ : SELECT { * | <column_name> [, <column_name>] ... }

FROM <table_name>

[WHERE [<column_name> {= | < | <= | > | >= | != } value_element]

| [<column_name> BETWEEN value_element AND value_element]

| [<column_name> IN (value_element [,value_element] ...)] }]]

พารามิเตอร์ : WHERE คือการกำหนดเงื่อนไขในการค้นหาทุกเปิดของรีเลชัน

* คือการระบุทุกแอตทริบิวในรีเลชัน

table_name คือชื่อของรีเลชัน

column_name คือชื่อของแอตทริบิว

value_element คือค่าของแอตทริบิว อาจเป็นข้อมูลที่เป็นอักขระ เลข จำนวนเต็ม เลขทศนิยม หรือวันที่ในรูปแบบ “วัน/เดือน/ปี”

3. คำสั่งในการออกจากการใช้งานระบบ มีความหมายและรูปแบบของคำสั่งดังนี้

คำสั่ง Quit

ความหมาย : ออกจากโปรแกรม

รูปแบบ : QUIT

ก.2 ข้อกำหนดในการใช้คำสั่งในโปรแกรม

ในการใช้งาน โปรแกรมมีข้อกำหนดในการใช้คำสั่งดังนี้

1. คำสั่งในระบบจะต้องพิมพ์ให้ถูกต้องตามรูปแบบที่กำหนดในระบบ และต้องปิดท้ายคำสั่งด้วยเครื่องหมาย ‘;’ เสมอ

2. ผู้ใช้จะต้องใช้คำสั่ง USE เพื่อระบุชื่อฐานข้อมูลที่ต้องการใช้งานก่อนจะใช้คำสั่งอื่นๆ เพื่อทำงานเกี่ยวกับข้อมูลของฐานข้อมูลในระบบได้

3. ชื่อต่างๆที่ผู้ใช้กำหนดขึ้นจะมีความยาวได้ไม่เกิน 15 ตัวอักษร จะเป็นตัวอักษรภาษาอังกฤษ ตัวเลข หรือขีดเส้นใต้ ก็ได้ แต่ต้องขึ้นต้นด้วยตัวอักษร ไม่มีช่องว่างระหว่างชื่อ และต้องไม่ซ้ำกับคำหลักที่ใช้ในระบบ ซึ่งชื่อต่างๆ ที่ผู้ใช้สามารถกำหนดขึ้น ได้แก่

- **ชื่อฐานข้อมูล** ชื่อฐานข้อมูลทั้งหมดในระบบจะต้องไม่ซ้ำกัน ใช้ในการอ้างอิงถึงฐานข้อมูล
- **ชื่อรีเลชัน** ชื่อรีเลชันในฐานข้อมูลหนึ่งๆ จะต้องไม่ซ้ำกัน ใช้ในการอ้างอิงถึงรีเลชันในฐานข้อมูล
- **ชื่อแอตทริบิว** หรือเรียกอีกอย่างว่า ชื่อคอลัมน์ชื่อแอตทริบิวในรีเลชันหนึ่งๆ จะต้องไม่ซ้ำกันใช้ในการอ้างอิงถึงแอตทริบิวของรีเลชันในฐานข้อมูล

4. ชนิดข้อมูลของแอตทริบิว ในการสร้างรีเลชันผู้ใช้สามารถกำหนดชนิดของข้อมูลเพื่อใช้จัดเก็บข้อมูลแอตทริบิวต่างๆ ในรีเลชัน ได้ดังนี้

- **CHAR** ข้อมูลชนิดอักขระ สามารถกำหนดขนาดข้อมูลเพื่อกำหนดเนื้อที่ในการจัดเก็บข้อมูล สามารถกำหนดขนาดข้อมูลชนิดอักขระได้ในช่วง 1 ถึง 255 ตัวอักษร เช่น CHAR (20) หมายถึงกำหนดชนิดข้อมูลอักขระที่มีขนาดในการจัดเก็บ 20 ไบต์ ขนาดข้อมูลที่กำหนดจะลงที่ แม้ว่าใส่ข้อมูลในแอตทริบิวไม่ครบตามขนาดที่กำหนดก็ตาม ก็จะเติมช่องว่างให้จนเต็ม
- **NUM** ข้อมูลชนิดตัวเลข แบ่งเป็น สองชนิดคือ
 - ข้อมูลชนิดตัวเลขจำนวนเต็ม จะจัดเก็บด้วยเนื้อที่ขนาด 1 2 หรือ 4 ไบต์ ขึ้นอยู่กับจำนวนตำแหน่งที่ผู้ใช้ระบุ เช่น NUM(2) จะจัดเก็บด้วยเนื้อที่ขนาด 1 ไบต์ และสามารถกำหนดค่าข้อมูลได้ในช่วง 0-99
 - ข้อมูลชนิดตัวเลขทศนิยม จะจัดเก็บด้วยเนื้อที่ขนาดคงที่ 8 ไบต์ โดยผู้ใช้สามารถระบุตำแหน่งตัวเลขทั้งหมด และจำนวนตำแหน่งสูงสุดหลังจุดทศนิยมได้ในช่วง 1 ถึง 9 เช่น NUM (5,2) จะเป็นการกำหนดข้อมูลชนิดตัวเลข

ทศนิยมมีจำนวนตำแหน่งทั้งหมดรวมจุดทศนิยมไม่เกิน 5 ตำแหน่ง และ
จำนวนตัวเลขหลังจุดทศนิยมไม่เกิน 2 ตำแหน่ง เป็นต้น

- **DATE** ข้อมูลวันที่ในรูปแบบ วัน/เดือน/ปี จัดเก็บด้วยค่าเลขจำนวนเต็มขนาด 4 ไบต์ โดยคำนวณจากสมการ วันที่ = ปี *10000+ เดือน*100+ วัน เช่น “2/2/2000” ค่าที่จัดเก็บ คือ 20000202 เป็นต้น

5. การกำหนดคีย์หลักคีย์หลัก และคีย์รอง สามารถกำหนดแอตทริบิวต์ใดๆ ในรีเลชันเป็นคีย์ได้มากกว่าหนึ่งแอตทริบิวต์ แต่ในรีเลชันหนึ่งจะกำหนดคีย์แต่ละประเภทได้เพียงคีย์ครั้งเดียวเท่านั้น เมื่อแอตทริบิวต์ใดถูกกำหนดเป็นคีย์หลักหรือคีย์รองจะถูกกำหนดว่าไม่อนุญาตให้ใส่ค่าว่างให้กับแอตทริบิวต์นั้นโดยอัตโนมัติด้วย

6. การกำหนดเงื่อนไขการตรวจสอบค่าแอตทริบิวต์ใน where จะกำหนดเป็นเงื่อนไขง่ายๆ และกำหนดได้เพียงครั้งละหนึ่งแอตทริบิวต์ ซึ่งเงื่อนไขที่สามารถกำหนดได้มีดังนี้

- การเปรียบเทียบเท่ากับ (=) เช่น age = 30
- การเปรียบเทียบมากกว่า (>) เช่น age > 30
- การเปรียบเทียบมากกว่าหรือเท่ากับ (>=) เช่น age >= 30
- การเปรียบเทียบน้อยกว่า (<) เช่น age < 30
- การเปรียบเทียบน้อยกว่าเท่ากับ (<=) เช่น age <= 30
- การเปรียบเทียบไม่เท่ากับ (!=) เช่น age != 30
- การเปรียบเทียบค่าในช่วงปิดระหว่างค่า 2 ค่า (Between .. and ..) เช่น age between 20 and 30 หมายความว่าค่าแอตทริบิวต์มีค่าได้ไม่น้อยกว่า 20 และไม่เกิน 30
- การเปรียบเทียบค่าจากรายการที่กำหนด (In) เช่น city in (“chiangmai”, “chainingrai”) หมายความว่าค่าแอตทริบิวต์ที่สามารถกำหนดได้ คือ chiangmai หรือ chainingrai

7. การออกจากโปรแกรมผู้ใช้จะต้องใช้คำสั่ง QUIT เพื่อสิ้นสุดการทำงานของโปรแกรมเท่านั้น เพราะระบบทำการบันทึกข้อมูลในบัฟเฟอร์ที่มีการเปลี่ยนแปลงกลับลงสู่ดิสก์ และปิดแฟ้มข้อมูลทั้งหมดที่กำลังเปิดใช้งานอยู่ก่อนออกจากระบบ มิฉะนั้นจะทำให้ไม่สามารถเรียกใช้งานโปรแกรมได้อีก เนื่องจากข้อมูลในแฟ้มข้อมูลอาจมีข้อผิดพลาดเกิดขึ้นและไม่สามารถเปิดใช้แฟ้มข้อมูลเดิมในระบบได้

ภาคผนวก ข

ไวยากรณ์ของคำสั่งในระบบ

เมื่อผู้ใช้ต้องการใช้งานระบบจะต้องพิมพ์คำสั่งซึ่งส่วนใหญ่จะเป็นคำสั่งที่เกี่ยวกับการจัดการข้อมูลในฐานข้อมูล และผู้ใช้จะต้องพิมพ์คำสั่งให้ถูกต้องตามไวยากรณ์และลักษณะคำสั่งที่ได้ออกแบบไว้จึงจะได้ผลลัพธ์ตามที่ผู้ใช้ต้องการ ไวยากรณ์ของคำสั่งทั้งหมดที่ใช้ในโปรแกรมมีดังนี้

```

query: /* empty */
      | TK_QUIT
      | TK_SEMICOLON
      | verb_clause end_semi

end_semi:/* empty */
        | TK_SEMICOLON

verb_clause : create
            | delete
            | describe
            | drop
            | insert
            | select
            | show
            | update
            | use
            | closedb

dbname: TK_IDENT

relname: TK_IDENT

attname: TK_IDENT

create: TK_CREATE TK_TABLE relname '(' field_list ')'
       |TK_CREATE TK_DATABASE dbname

```

field_list : field_list_item

| field_list ',' field_list_item

field_list_item : field_spec

| TK_PRIMARY TK_KEY '(' key_list ')'

| TK_SECONDARY TK_KEY '(' key_list ')'

| TK_FOREIGN TK_KEY '(' key_list ') references

field_spec : atname type opt_attr

type : TK_NUM '(' precision ')'

| TK_NUM '(' precision ',' scale ')'

| TK_CHAR char_len

| TK_DATE

precision : TK_INTEGER

scale : TK_INTEGER

char_len : /* empty */

| '(' TK_INTEGER ')'

opt_attr : /* empty */

| opt_attr_list

opt_attr_list : opt_attr_list attribute

| attribute

attribute : TK_NOT TK_NULL

| TK_DEFAULT TK_STRING

| TK_PRIMARY TK_KEY

| TK_SECONDARY TK_KEY

simple_expr : atname function_expr

function_expr : relational_op value1

| TK_BETWEEN value TK_AND value

| TK_IN '(' values ')'

relational_op : TK_EQ

| TK_GE

| TK_GT

| TK_LE

| TK_LT

| TK_NE

key_list : key_list ',' atname

| atname

references : TK_REFERENCE relname '(' key_list ')' op_delete_on

op_delete_on : /* emptye */

| TK_ON TK_DELETE

insert : TK_INSERT TK_INTRO relname insert_field TK_VALUES '(' values ')'

insert_field: /* empty */ { }

| '(' key_list ')'

values : values ',' value

| value

value: TK_INTEGER

| TK_REAL

| TK_STRING

value1: atname

| value

drop : TK_DROP TK_TABLE relname

| TK_DROP TK_DATABASE dbname

delete : TK_DELETE TK_FROM relname where_clause

where_clause: /* empty */ { }

| TK_WHERE simple_expr { }

update : TK_UPDATE relname TK_SET update_list where_clause

update_list : update_list ',' attname TK_EQ value

| attname TK_EQ value

show : TK_SHOW show_param

show_param : TK_DATABASES

| TK_TABLES

describe : describe_command relname

describe_command : TK_DESC

| TK_DESCRIBE

use : TK_USE dbname

closedb : TK_CLOSE

select : TK_SELECT selected TK_FROM relname where_clause

selected : '*'

| selected_list

selected_list : selected_list ',' attname

| attname

ไวยากรณ์ของคำสั่งที่ใช้ในโปรแกรมข้างต้น ได้มีการกำหนดไว้ในแฟ้มข้อมูลชื่อ grammar เพื่อเป็นข้อมูลนำเข้าของโปรแกรม YACC เมื่อต้องการใช้โปรแกรม YACC จะต้องใช้คำสั่ง

```
yacc -d grammar
```

จะได้คำสั่ง #define ที่กำหนดรหัสให้กับสัญลักษณ์สิ้นสุดเก็บอยู่ในแฟ้มข้อมูลชื่อ y.tab.h และชุดคำสั่งย่อย yyparse() ที่ได้จาก yacc จะเก็บอยู่ในแฟ้มข้อมูลชื่อ y.tab.c ซึ่งสามารถนำไปใช้ด้วยตัวแปลภาษาซีร่วมกับโปรแกรมต้นฉบับอื่น ๆ ในระบบ ซึ่งการทำงานของโปรแกรม y.tab.c นี้จะต้องใช้คำหลักที่กำหนดในแฟ้มข้อมูลชื่อ lex.h ด้วย

ถ้าต้องการเพิ่มคำสั่งใหม่เพื่อใช้งานระบบก็สามารถทำได้โดยการกำหนดกฎเพิ่มเติมในแฟ้มข้อมูลชื่อ `grammar` และกรณีต้องการเพิ่มคำหลักเพื่อใช้ในการตรวจสอบคำสั่งให้กำหนดเพิ่มในแฟ้มข้อมูลชื่อ `lex.h` และกำหนด `%token` ของคำหลักเพิ่มเติมในแฟ้มข้อมูลชื่อ `gramma` ด้วย

สัญลักษณ์ตัวหนา เป็นไปตามภาพไอ-อาร์ ประกอบด้วย

1. ชนิดของนิพจน์ไอ (operator types) ประกอบด้วย



แทนนิพจน์คณิตศาสตร์ (arithmetic operator types)



แทนสิ่งซึ่งถือถึงความสัมพันธ์หรือค่าที่มีความสัมพันธ์กัน
รวมไปถึงของไอ (relational operator types)

2. ชนิดของความสัมพันธ์ไอ (relationship types) ประกอบด้วย



แทนความสัมพันธ์ซึ่งใช้รวมกับคำสั่ง ไอหรือสิ่งความสัมพันธ์กับ
กับตนเอง



แทนความสัมพันธ์ซึ่งรวมกับหลายๆ ไอ โดยสิ่งซึ่งความสัมพันธ์
เกิดขึ้นสอง



แทนความสัมพันธ์ซึ่งรวมกับหลายๆ ไอโดยให้ความสัมพันธ์กับ
หรือ ไม่ก็ได้



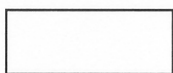
แทนความสัมพันธ์ซึ่งรวมกับหลายๆ ไอโดยให้ความสัมพันธ์กับ
ขึ้นหรือไม่ได้

ภาคผนวก ค

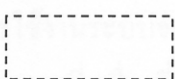
สัญลักษณ์ต่างๆ ที่ใช้ในแผนภาพโอ-อาร์

สัญลักษณ์ต่างๆ ที่ใช้ในแผนภาพโอ-อาร์ ประกอบด้วย

1. ชนิดของสิ่งที่สนใจ (object types) ประกอบด้วย



แทนสิ่งชนิดอิสระ (independent object type)



แทนสิ่งที่เกิดจากความสัมพันธ์หรือสิ่งที่มีความสัมพันธ์กัน
ระหว่างสองสิ่ง (association object type)

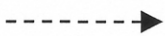
2. ชนิดของความสัมพันธ์ (relationship types) ประกอบด้วย



แทนความสัมพันธ์ร่วมแบบค่าเดียว โดยต้องมีค่าความสัมพันธ์เกิดขึ้นเสมอ



แทนความสัมพันธ์ร่วมแบบหลายค่า โดยต้องมีค่าความสัมพันธ์เกิดขึ้นเสมอ



แทนความสัมพันธ์ร่วมแบบค่าเดียว โดยอาจมีความสัมพันธ์เกิดขึ้นหรือไม่ก็ได้



แทนความสัมพันธ์ร่วมแบบหลายค่า โดยอาจมีความสัมพันธ์เกิดขึ้นหรือไม่ก็ได้

ภาคผนวก ง

การใช้งานระบบ

ระบบที่พัฒนาขึ้นนี้ พัฒนาภายใต้ระบบปฏิบัติการลินุกซ์ และผู้ใช้สามารถใช้งานระบบที่พัฒนาขึ้นนี้โดยผ่านการพิมพ์คำสั่งที่ได้กำหนดไว้ และต้องพิมพ์คำสั่งให้ถูกต้องตามรูปแบบของคำสั่งดังที่ได้อธิบายไว้แล้วในภาคผนวก ก

ก.1 การเริ่มเข้าสู่การทำงานของระบบ

การใช้งานระบบจะต้องเข้าสู่การทำงานภายใต้ระบบปฏิบัติการลินุกซ์ก่อน โดยจะเข้าสู่การทำงานของระบบที่เครื่องที่มีระบบปฏิบัติการลินุกซ์โดยตรง หรือจะใช้วิธีการเชื่อมต่อผ่านเครื่องคอมพิวเตอร์อื่นๆ โดยผ่านเครือข่ายด้วยโปรแกรม Telnet เพื่อติดต่อไปยังเครื่องที่ทำงานภายใต้ระบบปฏิบัติการลินุกซ์ สำหรับเครื่องที่ใช้ในการพัฒนาระบบได้กำหนดหมายเลข IP (Internet Protocol) ไว้เป็น 172.28.80.59

เมื่อจะเข้าสู่การทำงานภายใต้ระบบปฏิบัติการลินุกซ์ ผู้ใช้จะต้องใส่ login และ password ดังตัวอย่างต่อไปนี้

```
Login : benjamas
```

```
Password :
```

จากนั้นจึงพิมพ์คำสั่งเพื่อใช้งานโปรแกรมระบบที่พัฒนาขึ้น ดังนี้

```
[LinuxM105]$ myrdbms<Enter>
```

เมื่อ [LinuxM105]\$ เป็น Linux prompt ของเครื่อง โปรแกรม myrdbms จะเริ่มทำงานโดยจะแสดงข้อความ myrdbms>> ซึ่งเป็น prompt ของระบบที่พัฒนาขึ้น

```
myrdbms>>
```

ค.2 ตัวอย่างการใช้งานระบบ

การใช้งานระบบผู้ใช้จะต้องพิมพ์คำสั่งให้ถูกต้องตามรูปแบบของคำสั่ง โดยที่ทุกคำสั่งจะต้องปิดท้ายคำสั่งด้วยเครื่องหมาย ';' เสมอ เมื่อผู้ใช้พิมพ์คำสั่งเรียบร้อยแล้วให้กดปุ่ม Enter ระบบจะแสดงผลการทำงานของคำสั่งที่ผู้ใช้พิมพ์เข้าไป แต่ถ้าผู้ใช้พิมพ์คำสั่งไม่ถูกต้องจะแสดงข้อความแจ้งให้ผู้ใช้ทราบ เช่น ถ้าผู้ใช้พิมพ์คำสั่งสร้างรีเลชันด้วยคำสั่งต่อไปนี้

```
create table (dept_id num(1) primary key, dept_name char(20) not null); <Enter>
```

เนื่องจากตัวอย่างคำสั่งสร้างรีเลชันดังกล่าวไม่ได้ระบุชื่อรีเลชัน จึงมีรูปแบบของคำสั่งไม่ถูกต้องตามที่กำหนดไว้ ดังนั้นระบบจะแสดงข้อความแจ้งข้อผิดพลาดพร้อมกับระบุตำแหน่งที่ผิดพลาดของคำสั่งให้ด้วย ผลลัพธ์ของคำสั่งเป็นดังนี้

```
create table (dept_id num(1) primary key, dept_name char(20) not null);
```

^

```
ERROR : Syntax error near '(dept_id ...'
```

หรือในกรณีผู้ใช้พิมพ์คำสั่ง โดยไม่ได้ปิดท้ายคำสั่งด้วยเครื่องหมาย ';' ดังตัวอย่างคำสั่งต่อไปนี้

```
create table dept(dept_id num(1) primary key, dept_name char(20) not null);<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
create table dept(dept_id num(1) primary key, dept_name char(20) not null
```

^

```
ERROR : ';' expected
```


ต่อไปนี้เป็นตัวอย่างการใช้คำสั่งต่างๆ เพื่อสร้างฐานข้อมูลใหม่ในระบบชื่อ personal โดยในฐานข้อมูลประกอบด้วยข้อมูล 2 รีเลชัน คือ

- 1) รีเลชันชื่อ **dept** เป็นรีเลชันที่จัดเก็บรหัสหน่วยงาน (ID) และชื่อหน่วยงาน (NAME) โดยที่รหัสและชื่อหน่วยงานในรีเลชันจะไม่ซ้ำกันและไม่เป็นค่าว่าง และรีเลชัน dept ประกอบด้วยข้อมูลต่อไปนี้

ID	NAME
1	Compse
2	Math
3	Physic
4	Chem

- 2) รีเลชันชื่อ **person** เป็นรีเลชันที่จัดเก็บรหัสพนักงาน (ID) ชื่อพนักงาน (NAME) วันเดือนปีเกิดของพนักงาน (BDATE) เพศของพนักงาน (SEX) และรหัสหน่วยงานที่พนักงานสังกัดอยู่ (DEPT) โดยที่รหัสพนักงานในรีเลชันจะไม่ซ้ำกันและไม่เป็นค่าว่าง ชื่อพนักงานต้องไม่เป็นค่าว่าง

ID	NAME	BDATE	SEX	DEPT
1	matinee	1/11/2517	F	1
2	benjamas	8/10/2516	F	1
3	sunti	24/8/2521	M	2
4	noppamas	6/11/2521	F	2

โดยที่กำหนดให้ค่าของแอตทริบิวต์ชื่อ dept ในรีเลชัน person อ้างอิงรหัสหน่วยงานจากค่าของแอตทริบิวต์ id ในรีเลชัน dept ของฐานข้อมูล personal

ตัวอย่างลำดับขั้นตอนในการสร้างฐานข้อมูลข้างต้นสามารถทำได้ดังนี้

1. การสร้างชื่อฐานข้อมูลใหม่

เริ่มต้นด้วยการสร้างฐานข้อมูลใหม่ในระบบชื่อ personal คำสั่งที่ใช้คือคำสั่ง create database ดังนี้

```
create database personal;<Enter>
```

ผลลัพธ์ของคำสั่งเพื่อแจ้งให้ทราบว่าได้สร้างฐานข้อมูลชื่อ personal ในระบบเรียบร้อยแล้วดังนี้

```
Database 'PERSONAL' created
```

ในกรณีที่ผู้ใช้ต้องการดูรายชื่อฐานข้อมูลทั้งหมดในระบบ ก็สามารถใช้คำสั่ง show databases ดังนี้

```
show databases; <Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
+-----+-----+-----+-----+
|DbName  |NoRel  |DbCreate  |LstConsId |
+-----+-----+-----+-----+
|PERSONAL|      0| 25/2/2002|          0|
+-----+-----+-----+-----+
1 row(s) selected
```

2. ระบุชื่อฐานข้อมูลที่ต้องการใช้งาน

ก่อนที่จะทำการสร้างรีเลชันใหม่ไว้ในฐานข้อมูล จะต้องระบุชื่อฐานข้อมูลที่เป็นเจ้าของรีเลชันใหม่ที่ต้องการสร้างขึ้น ด้วยคำสั่ง use ต่อไปนี้

```
use personal;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
Database 'PERSONAL' used
```

3. การสร้างรีเลชัน

ขั้นตอนต่อไปนี้เป็นกรการสร้างรีเลชันไว้ในฐานข้อมูลที่กำลังใช้งาน โดยจะสร้างรีเลชันชื่อ dept และรีเลชันชื่อ person ไว้ในฐานข้อมูล personal

- การสร้างรีเลชัน ชื่อ dept โดยจะกำหนดให้แอตทริบิวหรือฟิลด์ชื่อ id เป็นคีย์หลัก และแอตทริบิวชื่อ name เป็นคีย์รองของรีเลชัน ดังนั้นรหัสและชื่อหน่วยงานในรีเลชัน dept จะไม่ซ้ำกันและไม่เป็นค่าว่าง คำสั่งที่ใช้สร้างรีเลชัน คือ คำสั่ง create table ดังนี้

```
create table dept(id num(1) primary key, name char(20) secondary key);<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
Table 'DEPT' created
```

ในกรณีที่ผู้ใช้ต้องการดูรายละเอียดโครงสร้างของรีเลชัน dept ที่สร้างขึ้น จะใช้คำสั่ง describe ดังนี้

```
desc dept;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```

+-----+-----+-----+-----+-----+-----+-----+
|AttName |Domain |Pos  |Len  |Precision |Scale |Nullable |DefaVal |
+-----+-----+-----+-----+-----+-----+-----+
|ID       |I      |    0|    1|          1|    0|    0|N     |NULL   |
|NAME     |C      |    1|   20|          0|    0|    0|N     |NULL   |
+-----+-----+-----+-----+-----+-----+
2 row(s) selected

```

- การสร้างรีเลชันชื่อ **person** ไว้ในฐานข้อมูลที่กำลังใช้งานอยู่ (personal) โดยที่รหัสพนักงานในรีเลชันจะไม่ซ้ำกันและไม่เป็นค่าว่าง ชื่อพนักงานต้องไม่เป็นค่าว่าง และจะกำหนดให้ข้อมูลเพศของพนักงานมีค่าโดยปริยาย (default) เป็น 'F' และค่าของแอตทริบิวต์ dept จะต้องอ้างอิงรหัสหน่วยงานมาจากค่าของแอตทริบิวต์ id ในรีเลชัน dept ของฐานข้อมูลเดียวกัน คำสั่ง create table ที่ใช้มีรายละเอียดดังนี้

```
create table person(id num(4) ,name char(20) not null , bdate date,sex char default
"F" , dept num(1), primary key (id), foreign key(dept) references dept(id));<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
Table 'PERSON' created
```

กรณีที่ผู้ใช้ต้องการดูรายละเอียดโครงสร้างของรีเลชัน person ที่สร้างขึ้น ให้ใช้คำสั่ง describe ดังนี้

```
desc person;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|AttName |Domain |Pos  |Len  |Precision |Scale |Nullable |DefaVal |
+-----+-----+-----+-----+-----+-----+-----+-----+
|ID       |I      |    0|    2|          4|    0|    N    |NULL   |
|NAME     |C      |    2|   20|          0|    0|    N    |NULL   |
|BDATE    |D      |   22|    4|         10|    0|    Y    |NULL   |
|SEX      |C      |   26|    1|          0|    0|    Y    |F      |
|DEPT     |I      |   27|    1|          1|    0|    Y    |NULL   |
+-----+-----+-----+-----+-----+-----+-----+
5 row(s) selected
```

แต่ถ้าผู้ใช้ต้องการดูรายชื่อรีเลชันทั้งหมดในฐานข้อมูลที่กำลังใช้งานอยู่ (personal) ก็สามารถใช้คำสั่ง show tables ดังนี้

```
show tables;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
+-----+-----+-----+-----+-----+
|RelName |NoAtt |TopSize |RelCreate |RelLstUpd |
+-----+-----+-----+-----+-----+
|DEPT    |      2|      21| 25/2/2002| 25/2/2002|
|PERSON  |      5|      28| 25/2/2002| 25/2/2002|
+-----+-----+-----+-----+-----+
2 row(s) selected
```

4. การเพิ่มข้อมูลของรีเลชัน

การเพิ่มข้อมูลหรือทูปเปิดใหม่ในรีเลชัน ระบบจะตรวจสอบคุณลักษณะข้อมูลและเงื่อนไขบังคับของข้อมูลตามที่กำหนดไว้ในขั้นตอนการสร้างรีเลชัน ดังนั้นผู้ใช้จะต้องกำหนดข้อมูลให้ถูกต้องตามชนิดชนิด ขนาด หรือเงื่อนไขบังคับต่างๆ ให้ถูกต้องด้วย มิฉะนั้นระบบจะไม่ทำการเพิ่มทูปเปิดใหม่ของรีเลชันให้ สำหรับการเพิ่มทูปเปิดใหม่ของรีเลชันสามารถทำได้โดยใช้คำสั่ง insert into ต่อไปนี้จะเป็นตัวอย่างคำสั่งการเพิ่มทูปเปิดในรีเลชัน dept และ person

- การเพิ่มทูปเปิดของรีเลชัน dept ตามข้อมูลต่างๆ ที่กำหนดไว้ข้างต้น สามารถทำได้ดังนี้

```
insert into dept (id,name) values (1,"compse");<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
1 row(s) inserted
```

ผู้ใช้สามารถเพิ่มทูปเปิดในรีเลชันโดยไม่ระบุชื่อแอตทริบิวของรีเลชันได้ โดยข้อมูลที่กำหนดไว้หลังคำสั่ง values จะถูกกำหนดให้กับแต่ละแอตทริบิวของรีเลชันตามลำดับ ดังตัวอย่างคำสั่งต่อไปนี้

```
insert into dept values (2,"math");<Enter>
```

นั่นคือข้อมูล 2 จะถูกกำหนดให้กับแอตทริบิวต์ id และข้อมูล "math" จะถูกกำหนดให้กับแอตทริบิวต์ name ของรีเลชัน dept ตามลำดับ

สำหรับทิวเปิลใหม่อีก 2 ทิวเปิลของรีเลชัน dept ผู้ใช้สามารถเพิ่มเข้าไปในรีเลชัน dept ตามลำดับคำสั่งดังนี้

```
insert into dept values (3,"physic");<Enter>
```

```
insert into dept values (4,"chem");<Enter>
```

- การเพิ่มทิวเปิลของรีเลชัน **person** ตามข้อมูลที่กำหนดไว้ข้างต้น สามารถทำได้ด้วยคำสั่ง insert into เช่นเดียวกับการเพิ่มทิวเปิลในรีเลชัน dept ตามลำดับคำสั่งต่อไปนี้

```
insert into person values (1,"matinee","1/11/2517","F",1);<Enter>
```

```
insert into person values (2,"benjamas","8/10/2516","F",1);<Enter>
```

```
insert into person values (3,"sunti","24/8/2521","M",2);<Enter>
```

```
insert into person (id,name, bdate,dept) values(4,"noppamas","6/11/2521",2);<Enter>
```

สำหรับการเพิ่มทิวเปิลของรีเลชัน person ในคำสั่งสุดท้ายข้างต้น จะสังเกตว่าไม่ได้ระบุข้อมูลเพศของพนักงานไว้ในคำสั่ง แต่ในขั้นตอนการสร้างรีเลชัน person ได้กำหนดค่าโดย

ปรียาขของแอดทริบิว sex ไว้เป็น "F" ดังนั้นค่าของแอดทริบิว sex ของทุเปิดนี้จะมีค่าเป็น "F" โดยอัตโนมัติ

5. การดูข้อมูลของรีเลชัน

การดูข้อมูลต่าง ๆ ที่จัดเก็บในแต่ละรีเลชันสามารถทำได้ด้วยการใช้ คำสั่ง select โดยลักษณะการใช้คำสั่ง select แบ่งเป็น 2 ประเภท

- การดูทุกทุเปิดของรีเลชัน จะใช้คำสั่ง select โดยไม่มีการกำหนดเงื่อนไขของ where ตัวอย่างเช่น
 - การดูข้อมูลแต่ละทุเปิดของรีเลชัน dept ด้วยคำสั่ง select ต่อไปนี้

```
select * from dept;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
+-----+-----+
| ID   | NAME   |
+-----+-----+
|    1 | compsc |
|    2 | math   |
|    3 | physic |
|    4 | chem   |
+-----+-----+
4 row(s) selected
```

- การดูข้อมูลแต่ละทุเปิดของรีเลชัน person ด้วยคำสั่ง select ต่อไปนี้

```
select * from person;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
+-----+-----+-----+-----+-----+
| ID   | NAME      | BDATE      | SEX | DEPT |
+-----+-----+-----+-----+-----+
|    1 | matinee   | 1/11/2517 | F   |    1 |
|    2 | benjamas  | 8/10/2516 | F   |    1 |
|    3 | sunti     | 24/8/2521 | M   |    2 |
|    4 | noppamas  | 6/11/2521 | F   |    2 |
+-----+-----+-----+-----+-----+
4 row(s) selected
```

- การดูข้อมูลบางทิวเป็ลของรีเลชันจะใช้คำสั่ง `select` โดยกำหนดเงื่อนไขของ `where` ตัวอย่างเช่น
 - การดูข้อมูลแต่ละทิวเป็ลของรีเลชัน `person` เฉพาะพนักงานที่มีวันเดือนปีเกิดอยู่ในช่วงเดือนพฤศจิกายน ปี พ.ศ 2521 ด้วยคำสั่ง `select` ต่อไปนี้

```
select * from person where bdate between "1/11/2521" and "30/11/2521";<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

ID	NAME	BDATE	SEX	DEPT
4	noppamas	6/11/2521	F	2

1 row(s) selected

- การขอข้อมูลของรีเลชันอาจจะระบุชื่อแอตทริบิวต์ที่ต้องการ เช่น ถ้าต้องการดูข้อมูลจากรีเลชัน `person` โดยแสดงเฉพาะชื่อพนักงานที่เป็นเพศหญิง ด้วยคำสั่ง `select` ต่อไปนี้

```
select name from person where sex = "F";<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

NAME
matinee
benjamas
noppamas

3 row(s) selected

6. การปรับปรุงแก้ไขข้อมูลของรีเลชัน

การปรับปรุงแก้ไขข้อมูลของรีเลชันจะใช้คำสั่ง update ซึ่งการปรับปรุงแก้ไขข้อมูลของรีเลชันจะมีการตรวจสอบชนิดข้อมูล ขนาดและเงื่อนไขบังคับของรีเลชันที่กำหนดไว้ในขั้นตอนการสร้างรีเลชันก่อนทำการแก้ไขปรับปรุงข้อมูลให้ ดังตัวอย่างคำสั่งต่อไปนี้

- ถ้าต้องการปรับปรุงวันเดือนปีเกิดของพนักงานชื่อ "sunti" ในรีเลชัน person ให้เป็นค่าวันเดือนปีเกิด "8/9/2521" ด้วยคำสั่งต่อไปนี้

```
update person set bdate = "8/9/2521" where name = "sunti";<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
1 row(s) matched 1 row (s)updated
```

จากผลลัพธ์ที่ได้แสดงว่าข้อมูลวันเดือนปีเกิดของพนักงานที่ชื่อ "sunti" ในรีเลชัน person ได้ถูกปรับปรุงเป็นข้อมูลวันเดือนปีเกิดค่าใหม่เรียบร้อยแล้ว ดังนั้นถ้าผู้ใช้เรียกดูข้อมูลในรีเลชัน person ใหม่ด้วยคำสั่ง select * from person; จะได้ผลลัพธ์ดังนี้

ID	NAME	BDATE	SEX	DEPT
1	matinee	1/11/2517	F	1
2	benjamas	8/10/2516	F	1
3	sunti	8/9/2521	M	2
4	noppamas	6/11/2521	F	2

4 row(s) selected

- ถ้าต้องการปรับปรุงรหัสหน่วยงานของหน่วยงาน compsc ในรีเลชัน dept ใหม่ โดยเปลี่ยนจากรหัส 1 เป็น 5 ด้วยคำสั่งต่อไปนี้

```
update dept set id = 5 where name="compsc";<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
1 row(s) matched 1 row(s) updated
```

จากผลลัพธ์ที่ได้แสดงว่าข้อมูลรหัสหน่วยงานของรีเลชัน dept ได้ถูกปรับปรุงเป็นค่ารหัสใหม่เรียบร้อยแล้ว ดังนั้นถ้าผู้ใช้เรียกดูข้อมูลในรีเลชัน dept ใหม่ด้วยคำสั่ง `select * from person;` จะได้ผลลัพธ์ดังนี้

```
+-----+
| ID   | NAME   |
+-----+
|     5| compsc |
|     2| math   |
|     3| physic |
|     4| chem   |
+-----+
4 row(s) selected
```

เนื่องจากรหัสหน่วยงานเป็นคีย์หลักในรีเลชัน dept และมีการกำหนดเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิวต์ของรหัสหน่วยงานในรีเลชัน person โดยอ้างอิงค่าแอตทริบิวต์จากรหัสหน่วยงานของรีเลชัน dept ไว้ในขั้นตอนการสร้างรีเลชัน person ดังนั้นเมื่อรหัสหน่วยงานของรีเลชัน dept มีการปรับปรุงเปลี่ยนแปลง ระบบจะทำการปรับปรุงค่ารหัสหน่วยงานที่ตรงกันในรีเลชัน person ให้โดยอัตโนมัติด้วย ถ้าผู้ใช้ข้อมูลในรีเลชัน person ใหม่จะพบว่าข้อมูลรหัสหน่วยงานของรีเลชัน person ได้ถูกปรับปรุงเช่นเดียวกัน และถ้าใช้คำสั่ง `select * from person;` เพื่อดูข้อมูลในรีเลชัน person จะได้ผลลัพธ์ดังนี้

```
+-----+-----+-----+-----+
| ID   | NAME     | BDATE   | SEX  | DEPT |
+-----+-----+-----+-----+
|     1| matinee  | 1/11/2517| F    | 5    |
|     2| benjamas | 8/10/2516| F    | 5    |
|     3| sunti    | 8/9/2521| M    | 2    |
|     4| noppamas | 6/11/2521| F    | 2    |
+-----+-----+-----+-----+
4 row(s) selected
```

7. การลบข้อมูลของรีเลชัน

การลบข้อมูลหรือทูปเปลของรีเลชันจะใช้คำสั่ง delete ซึ่งการลบข้อมูลของรีเลชันจะมีการตรวจสอบเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิวของรีเลชันที่กำหนดไว้ในขั้นตอนการสร้างรีเลชันก่อนที่จะทำการลบทูปเปลของรีเลชันนั้นๆ ด้วย ดังตัวอย่างคำสั่งต่อไปนี้

- ถ้าต้องการลบพนักงานชื่อ "sunti" ในรีเลชัน person ด้วยคำสั่งต่อไปนี้

```
delete from person where name = "sunti";<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
1 row(s) deleted
```

จากผลลัพธ์ที่ได้แสดงว่าข้อมูลพนักงานที่ชื่อ "sunti" ในรีเลชัน person ได้ถูกลบออกจากฐานข้อมูลเรียบร้อยแล้ว ดังนั้นถ้าเรียกดูข้อมูลในรีเลชัน person ใหม่ด้วยคำสั่ง select * from person; จะได้ผลลัพธ์ดังนี้

ID	NAME	BDATE	SEX	DEPT
1	matinee	1/11/2517	F	5
2	benjamas	8/10/2516	F	5
4	noppamas	6/11/2521	F	2

3 row(s) selected

- ถ้าต้องการหน่วยงานรหัส 1 ออกจากรีเลชัน dept ด้วยคำสั่งต่อไปนี้

```
delete from dept where id = 5;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
ERROR : Integrity constraint violated -cascade on delete ='N'
```

จากผลลัพธ์ที่ได้แสดงว่าไม่สามารถลบข้อมูลหน่วยงานที่มีรหัสหน่วยงานเท่ากับ “5” ในรีเลชัน dept ได้ เนื่องจากมีแอตทริบิวของรีเลชันอื่นในฐานข้อมูลเดียวกัน (รีเลชัน person) ถูกกำหนดให้มีการอ้างอิงค่าแอตทริบิวจากคีย์หลักของรีเลชัน dept ไว้ และไม่ได้กำหนดคุณสมบัติ on delete ของการกำหนดเงื่อนไขการอ้างอิงค่าแอตทริบิวไว้ตั้งแต่ขั้นตอนการสร้างรีเลชัน person จึงทำให้ไม่สามารถลบทุเป็ดในรีเลชัน dept ได้

8. การลบรีเลชัน

การลบรีเลชันเดิมออกจากฐานข้อมูลจะมีการตรวจสอบเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิวของรีเลชันที่กำหนดไว้ในขั้นตอนการสร้างรีเลชันก่อนที่จะทำการลบรีเลชันด้วย และในการลบรีเลชันจะทำการลบข้อมูลอื่นๆ ที่เกี่ยวข้องกับรีเลชันให้ทั้งหมดด้วย ได้แก่ ทุเป็ดทั้งหมดของรีเลชัน ข้อมูลแอตทริบิวของรีเลชัน และข้อมูลดัชนี รวมทั้งเงื่อนไขบังคับของรีเลชันให้ด้วย ดังตัวอย่างคำสั่งต่อไปนี้

- ถ้าต้องการรีเลชัน dept ด้วยคำสั่งต่อไปนี้

```
drop table dept;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
ERROR : Integrity constraint violated
```

จากผลลัพธ์ที่ได้แสดงว่าไม่สามารถลบรีเลชัน dept ได้ เนื่องจากมีแอตทริบิวของรีเลชัน person ถูกกำหนดให้มีการอ้างอิงค่าแอตทริบิวจากคีย์หลักของรีเลชัน dept ไว้ และไม่ได้กำหนดคุณสมบัติ on delete ไว้ในขั้นตอนการสร้างรีเลชัน person จึงทำให้ไม่สามารถลบรีเลชัน dept ได้

- ถ้าต้องการรีเลชัน person ด้วยคำสั่งต่อไปนี้

```
drop table person;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
Table 'PERSON' dropped
```

จากผลลัพธ์ที่ได้แสดงว่ารีเลชัน person ได้ถูกลบออกจากฐานข้อมูลเรียบร้อยแล้ว รวมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับรีเลชัน person ด้วย

9. การลบฐานข้อมูล

การลบฐานข้อมูลออกจากระบบ จะทำการลบทุกรีเลชันและข้อมูลที่เกี่ยวข้องกับรีเลชันทั้งหมดในฐานข้อมูลที่ต้องการลบให้ด้วย ได้แก่ ทุเป็ดทั้งหมดของรีเลชัน ข้อมูลแอตทริบิวของรีเลชัน และข้อมูลดัชนีของรีเลชัน รวมทั้งเงื่อนไขบังคับของรีเลชันให้ด้วย ซึ่งฐานข้อมูลที่ต้องการลบจะต้องไม่ใช่ฐานข้อมูลที่กำลังใช้งานอยู่ มิฉะนั้นจะต้องใช้คำสั่ง close เพื่อยกเลิกการใช้งานฐานข้อมูลนั้นก่อน ดังตัวอย่างคำสั่งต่อไปนี้

```
drop database personal;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
ERROR : Can't drop database ('PERSONAL' in used)
```

จากผลลัพธ์ที่ได้แสดงว่าไม่สามารถลบฐานข้อมูล personal ได้ เนื่องจากอยู่ในระหว่างใช้งาน ซึ่งต้องใช้คำสั่ง close ดังนี้

```
close;<Enter>
```

ผลลัพธ์ของคำสั่งเป็นดังนี้

```
Database 'PERSONAL' closed
```

จากผลลัพธ์แสดงว่าได้ยกเลิกการใช้งานฐานข้อมูล personal เรียบร้อยแล้ว ซึ่งถ้าใช้คำสั่ง drop database personal; ใหม่อีกครั้ง ฐานข้อมูล personal จะถูกลบออกจากระบบ โดยผลลัพธ์ที่ได้คือ

```
Database 'PERSONAL' dropped
```

10. การออกจากระบบ

เมื่อต้องการออกจากระบบหรือหยุดการทำงานของโปรแกรม จะต้องใช้คำสั่ง quit เพื่อให้ระบบทำการปิดเพิ่มข้อมูลที่เปิดใช้งานในระบบด้วย มิฉะนั้นเพิ่มข้อมูลจะไม่สามารถใช้งานได้ อีกในการทำงานของระบบครั้งต่อไป