บทที่ 4

เครื่องมือซอฟต์แวร์สร้างข้อกำหนดเซด ของคุณสมบัติเชิงพฤติกรรมของระบบ

ในบทนี้ จะกล่าวถึงการพัฒนาเครื่องมือซอฟต์แวร์สร้างข้อกำหนดเซดของคุณสมบัติเชิงพฤติกรรมของระบบ โดยเครื่องมือซอฟต์แวร์สำหรับสร้างข้อกำหนดเซดนี้ ชื่อ R2Z2 ซึ่งมีขั้นตอนในการพัฒนา 3 ส่วน คือ การออกแบบข้อมูลเข้า การออกแบบข้อมูลผลลัพธ์ และการออกแบบเครื่องมือซอฟต์แวร์สร้างข้อกำหนดเซด

4.1 การออกแบบข้อมูลเข้า

ข้อมูลเข้าของเครื่องมือซอฟต์แวร์จะอยู่ในรูปแฟ้มข้อมูลตัวอักษร ประกอบด้วย 3 แฟ้มข้อมูล ดังนี้

4.1.1 แฟ้มข้อมูลพจนานุกรมข้อมูล

แฟ้มข้อมูลพจนานุกรมข้อมูลสร้างจากเอนทิตีในแผนภาพเอนทิตีและความสัมพันธ์ มีนามสกุลเป็น .txt โดยข้อมูลแต่ละตัวคั่นด้วยเครื่องหมาย ":" มีรูปแบบ คือ

EntityName:AttributeName:DataType:Key:<FK Entity-Attribute>:<CS Constraint>

โดย

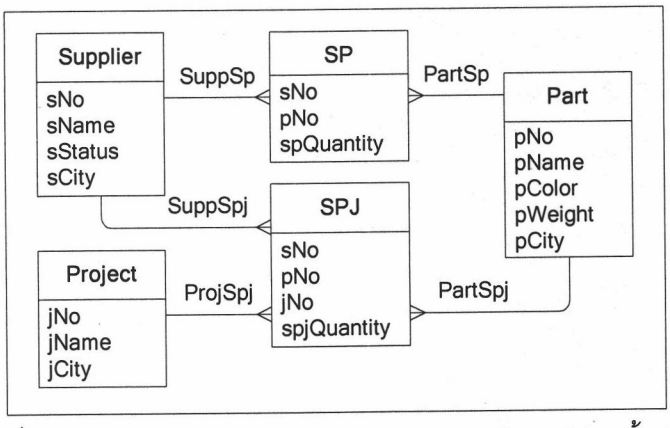
- Entity Name คือ ชื่อเอนทิตี ขึ้นต้นด้วยตัวอักษรใหญ่
- Attribute Name คือ ชื่อลักษณะประจำ ขึ้นต้นด้วยตัวอักษรเล็ก โดยในพจนานุกรมข้อมูลเดียวกันชื่อลักษณะประจำที่เหมือนกันจะต้องมีความหมายเดียวกันเสมอ
- Data Type คือ ชนิดข้อมูลของลักษณะประจำ อาจเป็นชนิดข้อมูลในสัญกรณ์เซด ได้แก่ \nat (จำนวนธรรมชาติ) \num (จำนวนนับ) หรือหากเป็นชนิดข้อมูลที่กำหนดขึ้นเองต้องเขียนด้วยตัวอักษรใหญ่ทั้งหมด
- Key คือ ชนิดของคีย์ของลักษณะประจำ โดยมีค่าเป็น PK เมื่อลักษณะประจำนั้นเป็นคีย์หลักของเอนทิตี และมีค่าเป็น NK เมื่อลักษณะประจำนั้นไม่ใช่คีย์ (non key) หรือเป็นคีย์นอก
- FK Entity-Attribute คือ คีย์นอก ถ้าลักษณะประจำนั้นเป็นคีย์นอกให้กำหนดคีย์นอกด้วยตัวอักษร FK ตามด้วยชื่อเอนทิตี คั่นด้วยเครื่องหมาย "-" และตามด้วยชื่อลักษณะประจำของคีย์นอก
- CS Constraint คือ ข้อบังคับของลักษณะประจำ ถ้าลักษณะประจำนั้นมีข้อบังคับ ให้กำหนดข้อบังคับด้วยตัวอักษร CS และตามด้วยข้อบังคับซึ่งเขียนอยู่ในรูปแทคของลาเท็กซ์ ดังรูปแบบการเขียนในตารางที่ 4.1 ในกรณีที่มีข้อบังคับมากกว่า 1 เงื่อนไข ให้คั่นด้วยเครื่องหมาย ":"

ตารางที่ 4.1 การเขียนข้อบังคับของลักษณะประจำ (Constraint)

รูปแบบการเขียน	ความหมาย
1. ขอบเขตค่าของลักษณะประจำ (สามารถใช้กับลักษณะประจำที่เป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น)	
CS > i	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่ามากกว่าค่าคงที่ที่กำหนด
CS \geq i	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่ามากกว่าหรือเท่ากับค่าคงที่ที่กำหนด
CS < i	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่าน้อยกว่าค่าคงที่ที่กำหนด
CS \leq i	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่าน้อยกว่าหรือเท่ากับค่าคงที่ที่กำหนด
CS \neq i	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่าไม่เท่ากับค่าคงที่ที่กำหนด
CS = i	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่าเท่ากับค่าคงที่ที่กำหนด
2. การกำหนด not null ให้กับลักษณะประจำ	
CS \neq \emptyset	คือ การระบุว่าลักษณะประจำนั้นต้องมีค่าเสมอ (ไม่เป็นเซตว่าง)

การเขียนเอนทิตีจะต้องเรียงลำดับให้ถูกต้อง คือ ถ้ามีการอ้างอิงเอนทิตีใดก่อน จะต้องเขียนเอนทิตีนั้นก่อนเสมอ

ตัวอย่างฐานข้อมูลจัดการชิ้นส่วน แสดงด้วยแผนภาพเอนทิตีและความสัมพันธ์ได้ดังนี้



รูปที่ 4.1 แผนภาพเอนทิตีและความสัมพันธ์ของฐานข้อมูลจัดการชิ้นส่วน

จากตัวอย่างแผนภาพเอนทิตีและความสัมพันธ์ของฐานข้อมูลจัดการชิ้นส่วนในรูปที่ 4.1 สามารถเขียนเป็นแฟ้มข้อมูลพจนานุกรมข้อมูลได้ดังนี้

```
Supplier:sNo:SNO:PK
Supplier:sName:SNAME:NK
Supplier:sStatus:\num:NK:CS \geq 0
Supplier:sCity:CITY:NK
Part:pNo:PNO:PK
Part:pName:PNAME:NK
Part:pColor:COLOR:NK
Part:pWeight:\num:NK:CS > 0
Part:pCity:CITY:NK
```

```
SP:sNo:SNO:PK:FK Supplier-sNo
SP:pNo:PNO:PK:FK Part-pNo
SP:spQuantity:\num:NK:CS \geq 0
Project:jNo:JNO:PK
Project:jName:JNAME:NK
Project:jCity:CITY:NK
SPJ:sNo:SNO:PK:FK Supplier-sNo
SPJ:pNo:PNO:PK:FK Part-pNo
SPJ:jNo:JNO:PK:FK Project-jNo
SPJ:spjQuantity:\num:NK:CS \geq 0
```

4.1.2 แฟ้มข้อมูลพจนานุกรมความสัมพันธ์

แฟ้มข้อมูลพจนานุกรมความสัมพันธ์สร้างจากความสัมพันธ์ในแผนภาพเอนทิตีและความสัมพันธ์ มีนามสกุลเป็น .txt โดยข้อมูลแต่ละตัวคั่นด้วยเครื่องหมาย ":" มีรูปแบบ คือ

```
Relation Name:Entity 1:Entity 2:Relation Type
```

โดย

- Relation Name คือ ชื่อความสัมพันธ์
- Entity 1 คือ เอนทิตีลำดับที่ 1 ของความสัมพันธ์
- Entity 2 คือ เอนทิตีลำดับที่ 2 ของความสัมพันธ์
- Relation Type คือ ชนิดของความสัมพันธ์ ค่าที่สามารถเป็นไปได้คือ 1-1 1-m และ m-1

จากตัวอย่างแผนภาพเอนทิตีและความสัมพันธ์ของฐานข้อมูลจัดการชิ้นส่วนในรูปที่ 4.1 สามารถเขียนเป็นแฟ้มข้อมูลพจนานุกรมความสัมพันธ์ได้ดังนี้

```
spSupplier:SP:Supplier:m-1
spPart:SP:Part:m-1
spjSupplier:SPJ:Supplier:m-1
spjPart:SPJ:Part:m-1
spjProject:SPJ:Project:m-1
```

4.1.3 แฟ้มข้อมูลแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล

แฟ้มข้อมูลแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล เขียนขึ้นจากแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล ซึ่งเกิดจากการนำการดำเนินการพื้นฐานที่มีอยู่ในเครื่องมือซอฟต์แวร์มาเรียงลำดับการทำงานให้สอดคล้องกับคำสั่งภาษาเอสคิวแอล 1 คำสั่ง โดยใช้แผนภาพเชิงลำดับชั้น

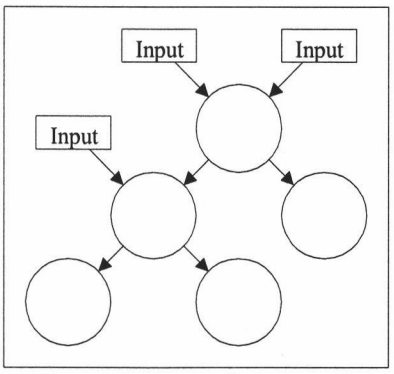
แผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล แสดงดังรูปที่ 4.2 ซึ่งมีส่วนประกอบดังนี้

1) โหนด ใช้สัญลักษณ์วงกลมแทนการดำเนินการพื้นฐานที่มีอยู่ในเครื่องมือซอฟต์แวร์ โดยระบุชื่อการดำเนินการพื้นฐานนั้นๆ ภายในวงกลม เช่น Select Project หรือ Join เป็นต้น แต่ละโหนดสามารถมีเส้น

เชื่อมเข้าสู่โหนดและออกจากโหนดได้ไม่เกิน 2 เส้น

2) เส้นเชื่อมระหว่างโหนด ใช้สัญลักษณ์เส้นตรงหัวลูกศรทึบ เพื่อแสดงลำดับของการดำเนินการ โดยแต่ละโหนดสามารถมีเส้นเชื่อมออกไปยังโหนดอื่น 2 เส้น และมีเส้นเชื่อมเข้ามาจากโหนดอื่นเข้าหาได้เพียง 1 เส้น ยกเว้นโหนดเริ่มต้นซึ่งจะไม่มีเส้นเชื่อมของโหนดอื่นเข้าหา

3) ข้อมูลเข้า ใช้สัญลักษณ์สี่เหลี่ยมผืนผ้าแทนชื่อเอนทิตีที่เป็นข้อมูลเข้าสู่โหนดนั้นๆ โดยระบุชื่อเอนทิตีนั้นๆ ภายในกรอบสี่เหลี่ยม แต่ละโหนดสามารถมีข้อมูลเข้าเพียง 1 เอนทิตี ยกเว้นโหนดเริ่มต้นซึ่งสามารถมีข้อมูลเข้าได้ 2 เอนทิตี



รูปที่ 4.2 แผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล

การดำเนินการพื้นฐานที่มีในเครื่องมือซอฟต์แวร์ ประกอบด้วย 19 การดำเนินการ ซึ่งชื่อของการดำเนินการพื้นฐานอ้างอิงมาจากคำสั่งในภาษาเอสคิวแอล และพีชคณิตความสัมพันธ์ ได้แก่

1) Insert คือ การเพิ่มข้อมูล ซึ่งตรงกับคำสั่ง insert ในภาษาเอสคิวแอล

2) Update คือ การแก้ไขข้อมูล ซึ่งตรงกับคำสั่ง update ในภาษาเอสคิวแอล

3) Delete คือ การลบข้อมูล ซึ่งตรงกับคำสั่ง delete ในภาษาเอสคิวแอล

4) Select คือ การเลือกข้อมูลจากเอนทิตี โดยเลือกเฉพาะแถวที่มีค่าของลักษณะประจำตรงตามที่กำหนด ซึ่งคือส่วนหนึ่งของคำสั่ง where ในภาษาเอสคิวแอล และตรงกับการเลือกแถว (Select หรือ Restrict) ในพีชคณิตความสัมพันธ์ ซึ่งใช้คำสั่ง where เช่นเดียวกัน ในวิทยานิพนธ์นี้จึงกำหนดให้ใช้ชื่อ Select ตามชื่อการดำเนินการในพีชคณิตความสัมพันธ์

5) Project คือ การเลือกข้อมูลจากเอนทิตี โดยเลือกเฉพาะสดมภ์ตามที่กำหนด ซึ่งตรงกับคำสั่ง select ในภาษาเอสคิวแอล และตรงกับคำสั่ง project ในพีชคณิตความสัมพันธ์ ในวิทยานิพนธ์นี้จึงกำหนดให้ใช้ชื่อ Project ตามคำสั่งในพีชคณิตความสัมพันธ์

6) Join คือ การจอยแบบธรรมชาติ เป็นการเลือกข้อมูลจาก 2 เอนทิตี เฉพาะแถวที่มีค่าของลักษณะประจำซึ่งใช้ในการจอยตรงกัน ซึ่งคือส่วนหนึ่งของคำสั่ง where ในภาษาเอสคิวแอล และตรงกับคำสั่ง join ในพีชคณิตความสัมพันธ์ ในวิทยานิพนธ์นี้จึงกำหนดให้ใช้ชื่อ Join ตามคำสั่งในพีชคณิตความสัมพันธ์

7) Union คือ การรวมข้อมูลจาก 2 เอนทิตีเข้าด้วยกัน และทั้งสองเอนทิตีต้องมีโครงสร้างเหมือนกันทุกประการ ซึ่งตรงกับคำสั่ง union ในภาษาเอสคิวแอลและพีชคณิตความสัมพันธ์

8) Intersect คือ การเลือกข้อมูลจาก 2 เอนทิตี โดยเลือกเฉพาะข้อมูลที่เหมือนกัน และทั้งสองเอนทิตีต้องมีโครงสร้างเหมือนกันทุกประการ ซึ่งตรงกับคำสั่ง intersect ในภาษาเอสคิวแอลและพีชคณิตความสัมพันธ์

9) Except คือ การเลือกข้อมูลจาก 2 เอนทิตี โดยเลือกเฉพาะข้อมูลที่อยู่ในเอนทิตีแรก แต่ไม่อยู่ในเอนทิตีหลัง และทั้งสองเอนทิตีต้องมีโครงสร้างเหมือนกันทุกประการ ซึ่งตรงกับคำสั่ง except ในภาษาเอสคิวแอล และคำสั่ง minus ในพีชคณิตความสัมพันธ์ ในวิทยานิพนธ์นี้จึงกำหนดให้ใช้ชื่อ Except ตามคำสั่งในภาษาเอสคิวแอล

10) In คือ เงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย โดยเลือกเฉพาะข้อมูลจากคิวรี่หลักที่มีค่าของลักษณะประจำบางอย่างอยู่ในคิวรี่ย่อย ซึ่งตรงกับคำสั่ง in ในภาษาเอสคิวแอล

11) NotIn คือ เงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย โดยเลือกเฉพาะข้อมูลจากคิวรี่หลักที่มีค่าของลักษณะประจำบางอย่างไม่อยู่ในคิวรี่ย่อย ซึ่งตรงกับคำสั่ง not in ในภาษาเอสคิวแอล

12) Exists คือ เงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย โดยเลือกเฉพาะข้อมูลจากคิวรี่หลักที่มีค่าของลักษณะประจำทั้งหมดอยู่ในคิวรี่ย่อย ซึ่งตรงกับคำสั่ง exists ในภาษาเอสคิวแอล

13) NotExists คือ เงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย โดยเลือกเฉพาะข้อมูลจากคิวรี่หลักที่มีค่าของลักษณะประจำทั้งหมดไม่อยู่ในคิวรี่ย่อย ซึ่งตรงกับคำสั่ง not exists ในภาษาเอสคิวแอล

14) All คือ เงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย โดยเลือกเฉพาะข้อมูลจากคิวรี่หลักที่มีค่าของลักษณะประจำบางอย่างตรงกับทั้งหมดของคิวรี่ย่อย ซึ่งตรงกับคำสั่ง all ในภาษาเอสคิวแอล

15) Any คือ เงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย โดยเลือกเฉพาะข้อมูลจากคิวรี่หลักที่มีค่าของลักษณะประจำบางอย่างตรงกับบางส่วนของคิวรี่ย่อย ซึ่งตรงกับคำสั่ง any ในภาษาเอสคิวแอล

16) Count คือ ฟังก์ชันนับจำนวนสมาชิก โดยนับจำนวนสมาชิกทั้งหมดในเอนทิตี ซึ่งตรงกับฟังก์ชัน count ในภาษาเอสคิวแอล

17) Max คือ ฟังก์ชันหาค่าสูงสุดของลักษณะประจำ โดยหาค่าสูงสุดของลักษณะประจำ และลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น ซึ่งตรงกับฟังก์ชัน max ในภาษาเอสคิวแอล

18) Min คือ ฟังก์ชันหาค่าต่ำสุดของลักษณะประจำ โดยหาค่าต่ำสุดของลักษณะประจำ และลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น ซึ่งตรงกับฟังก์ชัน min ในภาษาเอสคิวแอล

19) Sum คือ ฟังก์ชันหาค่าผลรวมของลักษณะประจำ โดยหาค่าผลรวมทั้งหมดของลักษณะประจำและลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น ซึ่งตรงกับฟังก์ชัน sum ในภาษาเอสคิวแอล

แฟ้มข้อมูลแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลมีนามสกุลเป็น .txt โดยจะต้องเริ่มต้นข้อมูลของแผนภาพด้วย “%BDDBEGIN” และจบด้วย “%BDDEND” เสมอ ส่วนข้อมูลต่างๆ ของการดำเนินการพื้นฐานคั่นด้วยเครื่องหมาย “:” มีรูปแบบ ดังนี้

```
%BDDBEGIN
Node #:Operation Name:Input 1: Input 2:Condition:Output Node:Error Node
%BDDEND
```

โดย

- Node # คือ หมายเลขประจำโหนด ต้องเป็นจำนวนธรรมชาติเริ่มจาก 1 เรียงตามลำดับ
- Operation Name คือ ชื่อการดำเนินการพื้นฐาน โดยค่าที่สามารถเป็นไปได้คือ Insert, Update, Delete, Select, Project, Join, Union, Intersect, Except, In, NotIn, Exists, NotExists, All, Any, Count, Max, Min และ Sum
- Input 1 คือ หมายเลขโหนดหรือชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1 หรือชื่อเอนทิตีที่ต้องการเพิ่ม แก้ไข หรือลบข้อมูล
- Input 2 คือ หมายเลขโหนดหรือชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2 สำหรับการดำเนินการพื้นฐานบางชนิด คือ การจอย การยูเนียน การอินเตอร์เซค การหาผลต่าง และเงื่อนไขในการทำคิวรี่ย่อย หากการดำเนินการพื้นฐานใดไม่มีข้อมูลเข้าลำดับที่ 2 ให้ใส่เครื่องหมาย -
- Condition คือ เงื่อนไขของการดำเนินการพื้นฐานนั้นๆ (ถ้ามี) ตามรายละเอียดในตารางที่ 4.2 โดยเขียนอยู่ในรูปแทคของลาเท็กซ์ หากการดำเนินการพื้นฐานใดไม่มีเงื่อนไข ให้ใส่เครื่องหมาย -
- Output Node คือ หมายเลขโหนดที่เป็นการดำเนินการพื้นฐานถัดไป หมายถึงโหนดที่ใช้โหนดนี้เป็นข้อมูลเข้า ในกรณีที่โหนดนี้เป็นโหนดสุดท้ายให้ระบุค่าเป็น Success
- Error Node คือ การดำเนินการที่ต้องทำเมื่อเกิดความผิดพลาดในการดำเนินการพื้นฐานขึ้น ซึ่งมีค่าเป็น NotFound เสมอ

การดำเนินการพื้นฐานบางชนิด จะต้องระบุเงื่อนไขของการดำเนินการพื้นฐาน โดยมีกฎเกณฑ์ในการเขียนเงื่อนไขของการดำเนินการพื้นฐาน ดังตารางที่ 4.2

ตารางที่ 4.2 การเขียนเงื่อนไขของการดำเนินการพื้นฐาน

Primitive Operation	**การเขียน** Condition
Select	ระบุเงื่อนไขการเปรียบเทียบเพื่อให้ได้แถวของสมาชิกที่ต้องการ เครื่องหมายการเปรียบเทียบที่สามารถใช้ได้คือ =, >, <, \geq, \leq และ \neq โดยพจน์แรกคือชื่อลักษณะประจำที่ต้องการเปรียบเทียบ และพจน์หลังคือค่าคงที่ที่ต้องการเปรียบเทียบซึ่งหากเป็นตัวอักษรต้องอยู่ภายในเครื่องหมาย " " และหากมีมากกว่า 1 เงื่อนไข สามารถเชื่อมด้วย *\land* (และ) หรือ *\or* (หรือ)
Project	ระบุชื่อลักษณะประจำที่ต้องการ หากมีมากกว่า 1 ลักษณะประจำ ให้คั่นด้วยช่องว่าง
Join	ระบุเงื่อนไขที่ใช้ในการจอยแบบธรรมชาติ ซึ่งใช้เครื่องหมาย "=" ในการเปรียบเทียบคีย์ของข้อมูลเข้าทั้ง 2 โดยพจน์แรกคือชื่อลักษณะประจำของข้อมูลเข้าลำดับที่ 1 และพจน์หลังคือชื่อลักษณะประจำของข้อมูลเข้าลำดับที่ 2 มีรูปแบบการเขียนคือ "ชื่อเอนทิตีหรือหมายเลขโหนด,

Primitive Operation	**การเขียน** Condition
	ชื่อลักษณะประจำ"
In, NotIn	ระบุชื่อลักษณะประจำที่ต้องทำเงื่อนไขอิน หรือน็อทอิน โดยต้องระบุชื่อลักษณะประจำของ คิวรี่หลักและคิวรี่ย่อยคู่กันตามลำดับและคั่นด้วยช่องว่าง มีรูปแบบการเขียนคือ "ชื่อเอนทิตีหรือหมายเลขโหนด,ชื่อลักษณะประจำ" และต้องครบตามลักษณะประจำที่เป็นเงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย
Exists, NotExists	ระบุชื่อลักษณะประจำที่ต้องการทำเงื่อนไขเอ็กสีซท์ หรือน็อทเอ็กสีซท์ โดยต้องระบุชื่อลักษณะประจำของคิวรี่หลักและคิวรี่ย่อยคู่กันตามลำดับและคั่นด้วยช่องว่าง มีรูปแบบการเขียนคือ "ชื่อเอนทิตีหรือหมายเลขโหนด,ชื่อลักษณะประจำ" และต้องครบถ้วนตามลักษณะประจำที่เป็นผลลัพธ์ของคิวรี่หลัก
All, Any	ระบุชื่อลักษณะประจำที่ต้องการทำเงื่อนไขออล หรือเงื่อนไขเอนนี่ เครื่องหมายการเปรียบเทียบที่สามารถใช้ได้คือ =, >, <, \geq, \leq และ \neq โดยต้องระบุทั้งลักษณะประจำของคิวรี่หลักและคิวรี่ย่อยคู่กันตามลำดับและคั่นด้วยช่องว่าง มีรูปแบบการเขียนคือ "ชื่อเอนทิตีหรือหมายเลขโหนด,ชื่อลักษณะประจำ" และต้องครบตามลักษณะประจำที่เป็นเงื่อนไขในการเปรียบเทียบกับคิวรี่ย่อย
FSum	ระบุชื่อลักษณะประจำที่ต้องการหาค่าผลรวม โดยลักษณะประจำนั้นจะต้องมีชนิดข้อมูลเป็นจำนวนนับหรือจำนวนธรรมชาติ

จากคำสั่งภาษาเอสคิวแอล 1 คำสั่ง นำมาสร้างเป็นแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล โดยเลือกใช้การดำเนินการพื้นฐานที่มีอยู่ และเรียงลำดับให้สอดคล้องกับการทำงานของคำสั่งภาษาเอสคิวแอล กฎเกณฑ์ในการเรียงลำดับการดำเนินการพื้นฐานสามารถแบ่งตามคำสั่งภาษาเอสคิวแอลเป็น 2 ประเภท ดังนี้

1) การปรับปรุงข้อมูล ได้แก่ การเพิ่ม การแก้ไข และการลบข้อมูล สามารถเลือกใช้การดำเนินการพื้นฐาน Insert Update หรือ Delete เพียงอย่างใดอย่างหนึ่งเท่านั้น และไม่สามารถใช้ร่วมกับการดำเนินการพื้นฐานอื่นๆ ได้

2) การค้นคืนข้อมูล โดยแบ่งตามคำสั่งภาษาเอสคิวแอลเป็น 3 ประเภท ดังนี้

2.1) กรณีคำสั่งภาษาเอสคิวแอลมี 1 คิวรี่ มีลำดับของการดำเนินการพื้นฐานดังนี้

2.1.1) การจอยแบบธรรมชาติ (Join) เป็นการจอยระหว่าง 2 เอนทิตี หากในคำสั่งเอสคิวแอลมีการจอยมากกว่า 2 เอนทิตีขึ้นไปจะต้องทำการจอยทีละคู่เสมอ กรณีที่คำสั่งเอสคิวแอลนั้นไม่มีการจอย ให้ทำข้อ 2.1.2

2.1.2) การเลือกแถว (Select) เป็นการเลือกเฉพาะแถวที่ค่าของลักษณะประจำตรงตามที่กำหนด กรณีที่คำสั่งเอสคิวแอลนั้นไม่กำหนดเงื่อนไขในการเลือกแถว ให้ทำข้อ 2.1.3

2.1.3) การเลือกสดมภ์ (Project) เป็นการเลือกเฉพาะลักษณะประจำที่กำหนด กรณีที่คำสั่งเอสคิวแอลนั้นไม่มีการกำหนดสดมภ์ที่ต้องการ ให้ทำข้อ 2.1.4

2.1.4) การดำเนินการพื้นฐานฟังก์ชัน (Count Max Min และ Sum) หากในคำสั่งเอสคิวแอลมีการเรียกใช้ฟังก์ชัน count max min หรือ sum การดำเนินการพื้นฐานฟังก์ชันนั้นจะต้องอยู่เป็นลำดับสุดท้ายเสมอ และก่อนใช้การดำเนินการพื้นฐานฟังก์ชัน Max และ Min จะต้องทำการเลือกเฉพาะสดมภ์ (Project) ที่ต้องการหาค่าสูงสุดหรือต่ำสุดก่อนเสมอ

2.2) กรณีคำสั่งภาษาเอสคิวแอลมีมากกว่า 1 คิวรี่ และเชื่อมกันด้วยคำสั่ง Union Intersect หรือ Except มีลำดับของการดำเนินการพื้นฐานดังนี้

2.2.1) ต้องทำแต่ละคิวรี่ให้เสร็จก่อนที่จะนำคิวรี่มาเชื่อมกันทีละคู่ ซึ่งการเลือกทำคิวรี่ใดก่อนหลังไม่มีความสำคัญเพราะแต่ละคิวรี่เป็นอิสระจากกัน และในแต่ละคิวรี่มีลำดับการดำเนินการพื้นฐานเช่นเดียว กับข้อ 2.1

2.2.2) การเชื่อมคิวรี่ โดยมีการดำเนินการพื้นฐานที่สามารถเลือกใช้ได้ 3 ชนิด คือ Union Intersect หรือ Except ตามคำสั่งในภาษาเอสคิวแอล

2.3) กรณีคำสั่งภาษาเอสคิวแอลประกอบด้วยคิวรี่ย่อย มีลำดับการดำเนินการพื้นฐานดังนี้

2.3.1) ต้องทำคิวรี่ย่อยที่อยู่ในสุดก่อนเสมอ แล้วจึงทำคิวรี่ที่ถัดออกมา เพราะผลลัพธ์ที่ได้จากคิวรี่ย่อยที่อยู่ในสุดจะถูกนำมาใช้ในคิวรี่ถัดออกมา และในแต่ละคิวรี่มีลำดับการดำเนินการพื้นฐานเช่นเดียวกับข้อ 2.1

2.3.2) การเชื่อมคิวรี่หลักและคิวรี่ย่อยเข้าด้วยกัน กระทำเมื่อทำคิวรี่ย่อยและคิวรี่หลักเสร็จสิ้น โดยมีการดำเนินการพื้นฐานที่สามารถเลือกใช้ได้ 6 ชนิด คือ In NotIn Exists NotExists All หรือ Any ตามคำสั่งในภาษาเอสคิวแอล

จากกฎเกณฑ์ในการเรียงลำดับการดำเนินการพื้นฐานให้สอดคล้องกับคำสั่งภาษาเอสคิวแอลดังที่กล่าวมาแล้ว สามารถแสดงตัวอย่างการเลือกใช้และจัดเรียงการดำเนินการพื้นฐานได้ ดังตารางที่ 4.3

ตารางที่ 4.3 ตัวอย่างการเลือกใช้และเรียงลำดับการดำเนินการพื้นฐานให้สอดคล้องกับภาษาเอสคิวแอล

ตัวอย่างภาษาเอสคิวแอล	การดำเนินการพื้นฐานที่ใช้	คำอธิบาย
select * from R;	1. Project (R, att_1, att_2, ..., att_n)	โดยระบุข้อมูลเข้าคือ R และลักษณะประจำทั้งหมด คือ att_1, att_2, ..., att_n
select * from R where att_i = C;	1. Select (R, att_i = C)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการคือ att_i = C
select att_1, att_2 from R where att_i = C;	1. Select (R, att_i = C) 2. Project (1, att_1, att_2)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการคือ att_i = C โดยระบุข้อมูลเข้าคือ 1 (ผลลัพธ์จาก Select) และลักษณะประจำที่ต้องการคือ att_1, att_2
select * from R_1, R_2	1. Join (R_1, R_2, $R_1.att_i$ = $R_2.att_i$)	โดยระบุข้อมูลเข้าคือ R_1, R_2 และเงื่อนไข

ตัวอย่างภาษาเอสคิวแอล	การดำเนินการพื้นฐานที่ใช้	คำอธิบาย
where $R_1.att_i = R_2.att_i$;		ของการจอยคือ $R_1.att_i = R_2.att_i$
select * from R_1, R_2 where $R_1.att_i = R_2.att_i$ and R1.att_i = C;	1. Join (R_1, R_2, $R_1.att_i = R_2.att_i$)	โดยระบุข้อมูลเข้าคือ R_1, R_2 และเงื่อนไขของการจอยคือ $R_1.att_i = R_2.att_i$
	2. Select (1, att_i = C)	โดยระบุข้อมูลเข้าคือ 1 (ผลลัพธ์จาก Join) และเงื่อนไขของแถวที่ต้องการ คือ att_i = C
select att_1, att_2 from R_1, R_2 where $R_1.att_i = R_2.att_i$ and R1.att_i = C;	1. Join (R_1, R_2, $R_1.att_i = R_2.att_i$)	โดยระบุข้อมูลเข้าคือ R_1, R_2 และเงื่อนไข ของการจอยคือ $R_1.att_i = R_2.att_i$
	2. Select (1, att_i = C)	โดยระบุข้อมูลเข้าคือ 1 (ผลลัพธ์จาก Join) และเงื่อนไขของแถวที่ต้องการ คือ att_i = C
	3. Project (2, att_1, att_2)	โดยระบุข้อมูลเข้าคือ 2 (ผลลัพธ์จาก Select) และลักษณะประจำ att_1, att_2
select * from R_1 where att_1 = C UNION select * from R_2 where att_2 = C;	1. Select (R_1, att_i = C)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการคือ att_1 = C
	2. Select (R_2, att_i = C)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการคือ att_2 = C
	3. Union (1, 2)	โดยระบุข้อมูลเข้าคือ 1, 2 (ผลลัพธ์จาก Select ทั้ง 2 อัน)
select * from R_1 where att_i IN (select * from R_2 where att_2 = C);	1. Select (R_2, att_i = C)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการคือ att_2 = C
	2. In (R_1, 1, $R_1.att_i$, 1.att_i)	โดยระบุข้อมูลเข้าคือ R_1, 1 (ผลลัพธ์จาก Select) และเงื่อนไขของ In คือ $R_1.att_i$, 1.att_i
select att_1, att_2 from R_1 where EXISTS (select * from R_2 where att_i = C);	1. Select (R_2, att_i = C)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการ คือ att_i = C
	2. Exists (R_1, 1, $R_1.att_i$, 1.att_i)	โดยระบุข้อมูลเข้าคือ R_1, 1 (ผลลัพธ์จาก Select) และเงื่อนไขของ Exists คือ $R_1.att_i$, 1.att_i
	3. Project (2, att_1, att_2)	โดยระบุข้อมูลเข้าคือ 2 (ผลลัพธ์จาก Exists) และลักษณะประจำ att_1, att_2
select * from R_1 where att_1 > ALL	1. Project (R_2, att_1)	โดยระบุข้อมูลเข้าคือ R2 และลักษณะ ประจำ att_i

ตัวอย่างภาษาเอสคิวแอล	การดำเนินการพื้นฐานที่ใช้	คำอธิบาย
(select att_2 from R_2);	2. All (R_1, 1, $R_1.att_1$, $1.att_2$)	โดยระบุข้อมูลเข้าคือ R_1, 1 (ผลลัพธ์จาก Project) และเงื่อนไขของ All คือ $R_1.att_1$, $1.att_2$
select COUNT (*) from R where att_i = C;	1. Select (R, att_i = C) 2. Count (1)	โดยระบุข้อมูลเข้าคือ R และเงื่อนไขของแถวที่ต้องการ คือ att_i = C โดยระบุข้อมูลเข้าคือ 1 (ผลลัพธ์จาก Select)
select MAX (att_i) from R;	1. Project (R, att_i) 2. Max (1)	โดยระบุข้อมูลเข้าคือ R (ผลลัพธ์จาก Project) และลักษณะประจำ att_i โดยระบุข้อมูลเข้าคือ 1 (ผลลัพธ์จาก Project)
select SUM (att_i) from R;	1. Sum (R, att_i)	โดยระบุข้อมูลเข้าคือ R และลักษณะประจำ att_i

โดย att หมายถึงชื่อลักษณะประจำ

R หมายถึงชื่อเอนทิตี

C หมายถึงค่าคงที่ใดๆ

และส่วนที่อยู่ในวงเล็บของแต่ละการดำเนินการพื้นฐานคือข้อมูลเข้าและเงื่อนไขของการดำเนินการพื้นฐานนั้นๆ

ตัวอย่างการหารายชื่อผู้จัดหาและสถานะ ซึ่งมีค่าสถานะมากกว่า 15 และอยู่ในลอนดอน จากพจนานุกรมข้อมูลของระบบจัดการชิ้นส่วนในหัวข้อ 4.1.2 เขียนเป็นภาษาเอสคิวแอล คือ

select sName, sStatus from Supplier where sStatus > 15 and sCity = “London”;

สามารถเขียนในรูปของแฟ้มข้อมูลการดำเนินการพื้นฐานได้ ดังนี้

```
# BDD1
%BDDBEGIN
1:Select:Supplier:-:sStatus > 15_\land_sCity = “London”:2:Error
2:Project:1:-:sNo sStatus:Success:Error
%BDDEND
```

โดยบรรทัดที่ขึ้นต้นด้วยเครื่องหมาย # จะไม่ถูกนำมาประมวลผลในเครื่องมือซอฟต์แวร์

4.2 การออกแบบข้อมูลผลลัพธ์

เครื่องมือซอฟต์แวร์จะอ่านข้อมูลจากแฟ้มข้อมูลเข้าทั้ง 3 แฟ้ม เพื่อนำไปสร้างแฟ้มผลลัพธ์ของข้อกำหนดเซต โดยแฟ้มข้อมูลผลลัพธ์จะเขียนอยู่ในรูปแทคของลาเท็กซ์ ประกอบไปด้วยกีเวนเซตของซนิดข้อมูลของลักษณะประจำ และเค้าร่างต่างๆ ได้แก่ เค้าร่างเอนทิตี เค้าร่างเพิ่มเติมของเอนทิตี เค้าร่าง

ความสัมพันธ์ของเอนทิตี เค้าร่างแสดงสถานะ เค้าร่างผลลัพธ์ของการดำเนินการพื้นฐาน เค้าร่างการดำเนินการพื้นฐาน และเค้าร่างรวมการดำเนินการพื้นฐาน โดยแฟ้มผลลัพธ์นี้จะถูกสร้างขึ้นในชื่อแฟ้มข้อมูลที่ผู้ใช้กำหนดและมีนามสกุลเป็น .zed โดยใช้กฎ 28 ข้อ ดังที่กล่าวในบทที่ 3

ตัวอย่างบางส่วนของแฟ้มข้อมูลผลลัพธ์ของฐานข้อมูลจัดการชิ้นส่วน ดังรูปที่ 4.1

```
\begin{zed}
        [SNO, SNAME, CITY, PNO, PNAME, COLOR, JNO, JNAME]
\end{zed}
\begin{zed}
        BOOLEAN ::= True | False \\
\end{zed}
\begin{schema} {Supplier}
        sNo : SNO \\
        sName : SNAME \\
        sStatus : \num \\
        sCity : CITY \\
\where
        sStatus \geq 0 \\
\end{schema}
.
.
.
\begin{schema} {SupplierExt}
        supplierSet : \finset Supplier \\
        supplierKey : \finset SNO \\
        supplierAtt : SNO \pfun Supplier \\
\where
        \forall Supplier1,Supplier2 : Supplier \\
                | Supplier1 \in supplierSet \land Supplier2 \in supplierSet \land \\
                  Supplier1 \neq Supplier2 @ Supplier1.sNo \neq Supplier2.sNo \\
        \dom supplierAtt \subseteq supplierKey \\
\end{schema}
.
.
.
\begin{schema} {RelationshipspSupplier}
        spSupplier : \finset (SP \cross Supplier)
\where
        \forall spSupplier1,spSupplier2 : spSupplier\\
```

```
                @ first~spSupplier1 = first~spSupplier2 \implies \\
                    second~spSupplier1 = second~spSupplier2 \\
\end{schema}
.
.
.
\begin{schema} {BDD1R1}
        sNo : SNO \\
        sName : SNAME \\
        sStatus : \num \\
        sCity : CITY \\
\end{schema}
\begin{schema} {BDD1Op1Select}
        input1? : \finset Supplier \\
        sCityValue? : CITY
        output! : \finset BDD1R1 \\
\where
        input1? \neq \emptyset \\
        \forall out : output!; in1 : input1? \\
                | in1.sStatus > 15 \land in1.sCity = sCityValue? @ out = in1 \\
\end{schema}
.
.
.
```

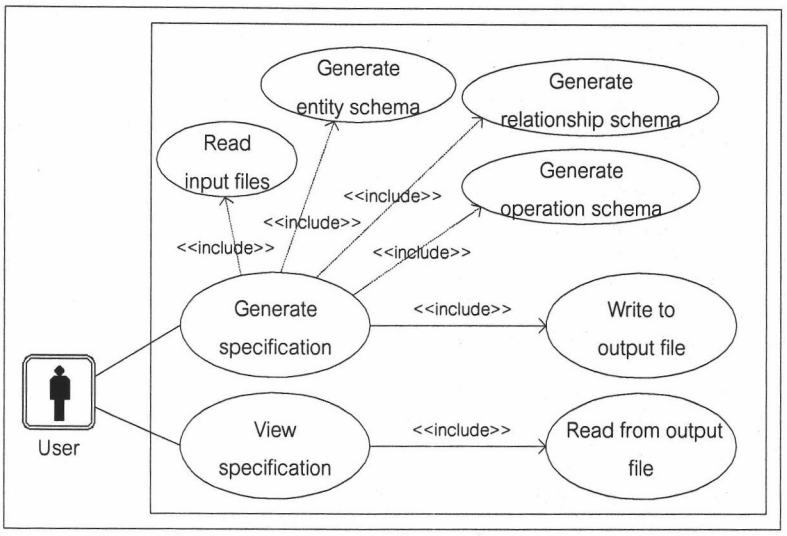
4.3 การออกแบบเครื่องมือซอฟต์แวร์

เครื่องมือซอฟต์แวร์ R2Z2 นี้พัฒนาขึ้นด้วยภาษาจาวา และใช้ระเบียบวิธีการออกแบบเชิงวัตถุ โดยใช้รูปแบบยูเอ็มแอล (UML - Unified Modeling Language) [11], [12] สามารถอธิบายขั้นตอนการทำงานของเครื่องมือซอฟต์แวร์ด้วยไดอะแกรมต่างๆ ได้แก่ ยูสเคสไดอะแกรมระดับสูง ยูสเคสไดอะแกรมของการทำงานย่อย แอคทิวิตีไดอะแกรม และคลาสไดอะแกรม

4.3.1 ยูสเคสไดอะแกรม

การทำงานของเครื่องมือซอฟต์แวร์ สามารถแสดงในรูปแบบของยูสเคสไดอะแกรมได้ดังรูปที่ 4.3 โดยสามารถแบ่งการทำงานออกเป็น 4 ยูสเคสหลัก ดังนี้

1) การสร้างข้อกำหนดเซต (Generate specification) เป็นส่วนการสร้างข้อกำหนดเซตของเค้าร่างต่างๆ ซึ่งได้จากแฟ้มข้อมูลเข้า 3 แฟ้ม คือ แฟ้มพจนานุกรมข้อมูล แฟ้มพจนานุกรมความสัมพันธ์ และแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล ยูสเคสไดอะแกรมนี้แสดงการทำงานย่อยของยูสเคสดังนี้



รูปที่ 4.3 ยูสเคสไดอะแกรมของเครื่องมือซอฟต์แวร์

1.1) การอ่านแฟ้มข้อมูลเข้า (Read input files) เป็นการอ่านข้อมูลของแฟ้มข้อมูลทั้ง 3 แฟ้ม คือ แฟ้มพจนานุกรมข้อมูล แฟ้มพจนานุกรมความสัมพันธ์ และแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล เพื่อเก็บไว้ในหน่วยความจำ

1.2) การสร้างเค้าร่างเอนทิตี (Generate entity schema) เป็นการสร้างเค้าร่างเอนทิตี และเค้าร่างเพิ่มเติมของเอนทิตี จากแฟ้มพจนานุกรมข้อมูล รายละเอียดอธิบายในหัวข้อ 4.3.2.1 แสดงกิจกรรมภายในยูสเคสย่อยนี้ด้วยแอคทิวิตีไดอะแกรม ดังรูปที่ 4.4

1.3) การสร้างเค้าร่างความสัมพันธ์ (Generate relationship schema) เป็นการสร้างเค้าร่างความสัมพันธ์ของเอนทิตี จากแฟ้มพจนานุกรมความสัมพันธ์ รายละเอียดอธิบายในหัวข้อ 4.3.2.2 และแสดงกิจกรรมภายในยูสเคสย่อยนี้ด้วยแอคทิวิตีไดอะแกรม ดังรูปที่ 4.4

1.4) การสร้างเค้าร่างการดำเนินการ (Generate operation schema) เป็นการสร้างเค้าร่างผลลัพธ์ และเค้าร่างการดำเนินการพื้นฐาน จากแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล รายละเอียดอธิบายในหัวข้อ 4.3.2.3 และแสดงกิจกรรมภายในยูสเคสย่อยนี้ได้ด้วยแอคทิวิตีไดอะแกรม ดังรูปที่ 4.4

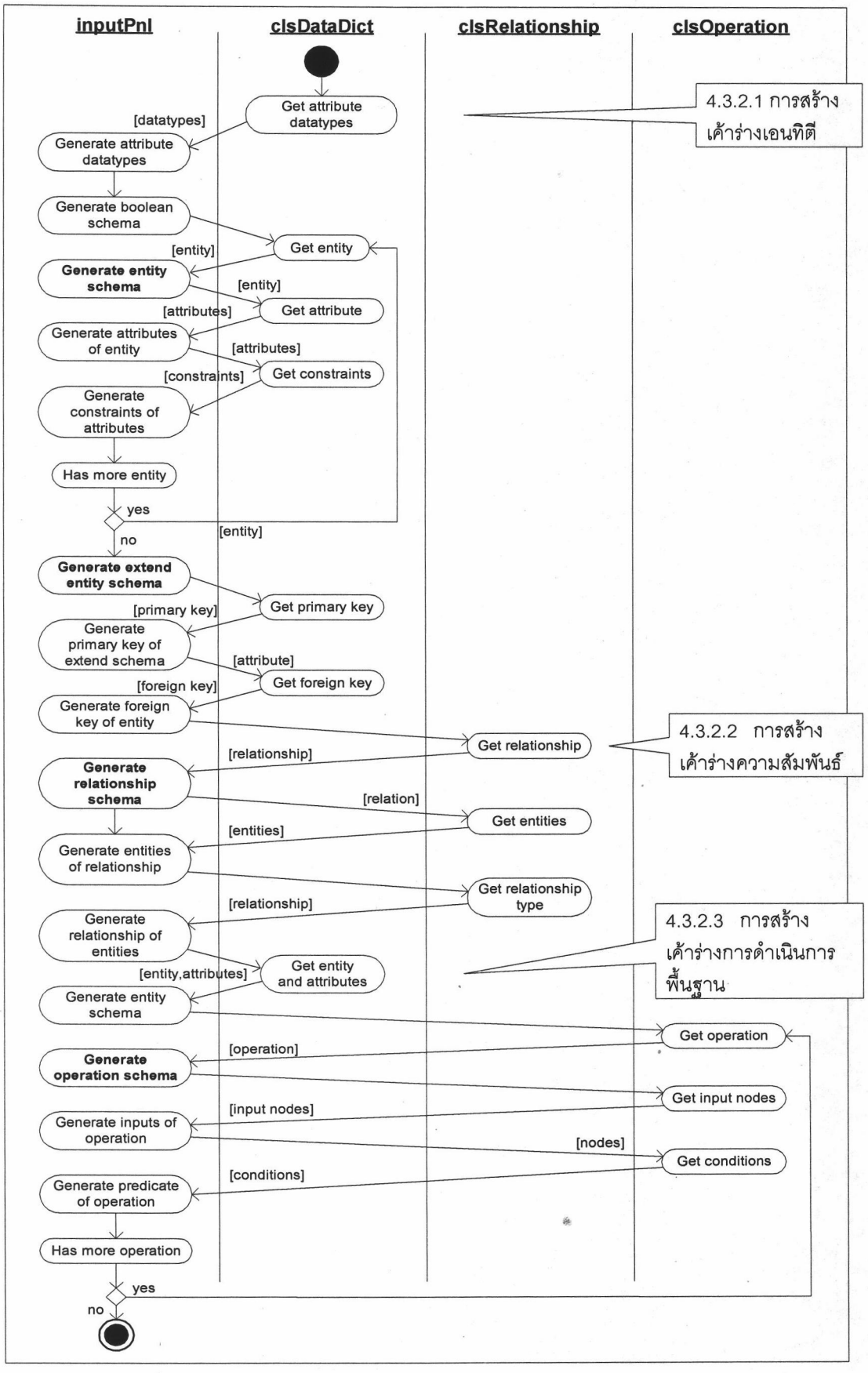
2) การบันทึกข้อกำหนดเซตลงแฟ้มผลลัพธ์ (Write to output file) เป็นส่วนการบันทึกข้อกำหนดเซตที่สร้างขึ้นจากข้อ 1) ลงแฟ้มผลลัพธ์ ตามชื่อแฟ้มผลลัพธ์ที่ผู้ใช้กำหนด

3) การเรียกดูแฟ้มผลลัพธ์ของข้อกำหนด (View specification) เป็นส่วนที่ผู้ใช้เรียกดูแฟ้มผลลัพธ์ของข้อกำหนดเซตที่ถูกสร้างขึ้นล่าสุด

4) การอ่านแฟ้มผลลัพธ์ (Read from output file) เป็นส่วนการอ่านแฟ้มผลลัพธ์ของข้อกำหนดเซตที่ถูกสร้างขึ้นล่าสุด

4.3.2 แอคทิวิตีไดอะแกรม

เป็นการอธิบายรายละเอียดของกิจกรรมภายในยูสเคสย่อยของยูสเคสสร้างข้อกำหนดเซต ซึ่งเป็นรายละเอียดของการสร้างข้อกำหนดเซตของเค้าร่างจากแฟ้มข้อมูลเข้า 3 ชนิด คือ เค้าร่างเอนทิตีสร้างจากแฟ้มพจนานุกรมข้อมูล เค้าร่างความสัมพันธ์สร้างจากแฟ้มพจนานุกรมความสัมพันธ์ และเค้าร่างการดำเนินการพื้นฐานสร้างจากแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล ดังรูปที่ 4.4



รูปที่ 4.4 แอคทิวิตีไดอะแกรมแสดงการทำงานของยูสเคสการสร้างข้อกำหนดเซต

4.3.2.1 แอคทิวิตีไดอะแกรมของยูสเคสย่อยการสร้างเค้าร่างเอนทิตี

เป็นการสร้างเค้าร่างเอนทิตีจากแฟ้มพจนานุกรมข้อมูล รายละเอียดกิจกรรมภายในยูสเคสย่อยนี้แสดงด้วยแอคทิวิตีไดอะแกรม ดังรูปที่ 4.4 โดยคลาส inputPnl ทำการแปลงแฟ้มพจนานุกรมข้อมูลเป็นข้อกำหนดเซต และคลาส clsDataDic ทำการอ่านข้อมูลของแฟ้มพจนานุกรมข้อมูลที่เก็บไว้ในหน่วยความจำ

การทำงานเริ่มจากการอ่านค่าชนิดข้อมูลทั้งหมดของลักษณะประจำ เพื่อนำไปสร้างเป็น กีเวนเซตของชนิดข้อมูล และชนิดอิสระของชนิดข้อมูลแบบบูลลีน ตามด้วยการอ่านชื่อเอนทิตี ชื่อลักษณะประจำของเอนทิตี ข้อบังคับของลักษณะประจำ เพื่อนำไปสร้างเป็นข้อกำหนดเซตของเค้าร่างเอนทิตี ลักษณะประจำของเอนทิตี และข้อบังคับของลักษณะประจำตามลำดับ โดยทำงานเช่นนี้จนกระทั่งจบแฟ้มข้อมูล จากนั้นจึงอ่านข้อมูลของคีย์หลักและคีย์นอกของเอนทิตี เพื่อนำไปสร้างเป็นข้อกำหนดเซตของเค้าร่างเพิ่มเติมของเอนทิตี

4.3.2.2 แอคทิวิตีไดอะแกรมของยูสเคสย่อยการสร้างเค้าร่างความสัมพันธ์

เป็นการสร้างเค้าร่างความสัมพันธ์จากแฟ้มพจนานุกรมความสัมพันธ์ รายละเอียดกิจกรรมภายยูสเคสย่อยนี้แสดงด้วยแอคทิวิตีไดอะแกรม ดังรูปที่ 4.4 โดยคลาส inputPnl ทำการแปลงแฟ้มพจนานุกรมความสัมพันธ์เป็นข้อกำหนดเซต และคลาส clsRelationship ทำการอ่านข้อมูลของแฟ้มพจนานุกรมความสัมพันธ์ที่เก็บไว้ในหน่วยความจำ

โดยเริ่มจากการอ่านค่าชื่อความสัมพันธ์ เพื่อนำไปสร้างข้อกำหนดเซตของเค้าร่างความสัมพันธ์ ตามด้วยการอ่านชื่อเอนทิตีที่เกี่ยวข้องในความสัมพันธ์ และชนิดของความสัมพันธ์ เพื่อนำไปข้อกำหนดเซตของการประกาศตัวแปรเอนทิตีในเค้าร่างความสัมพันธ์ และข้อบังคับของความสัมพันธ์ตามลำดับ

4.3.2.3 แอคทิวิตีไดอะแกรมของยูสเคสย่อยการสร้างเค้าร่างการดำเนินการ

เป็นการสร้างเค้าร่างการดำเนินการพื้นฐาน จากแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล รายละเอียดกิจกรรมภายยูสเคสย่อยนี้แสดงด้วยแอคทิวิตีไดอะแกรม ดังรูปที่ 4.4 โดยคลาส inputPnl ทำการแปลงแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลเป็นข้อกำหนดเซต และคลาส clsOperation ทำการอ่านข้อมูลของแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลที่เก็บไว้ในหน่วยความจำ

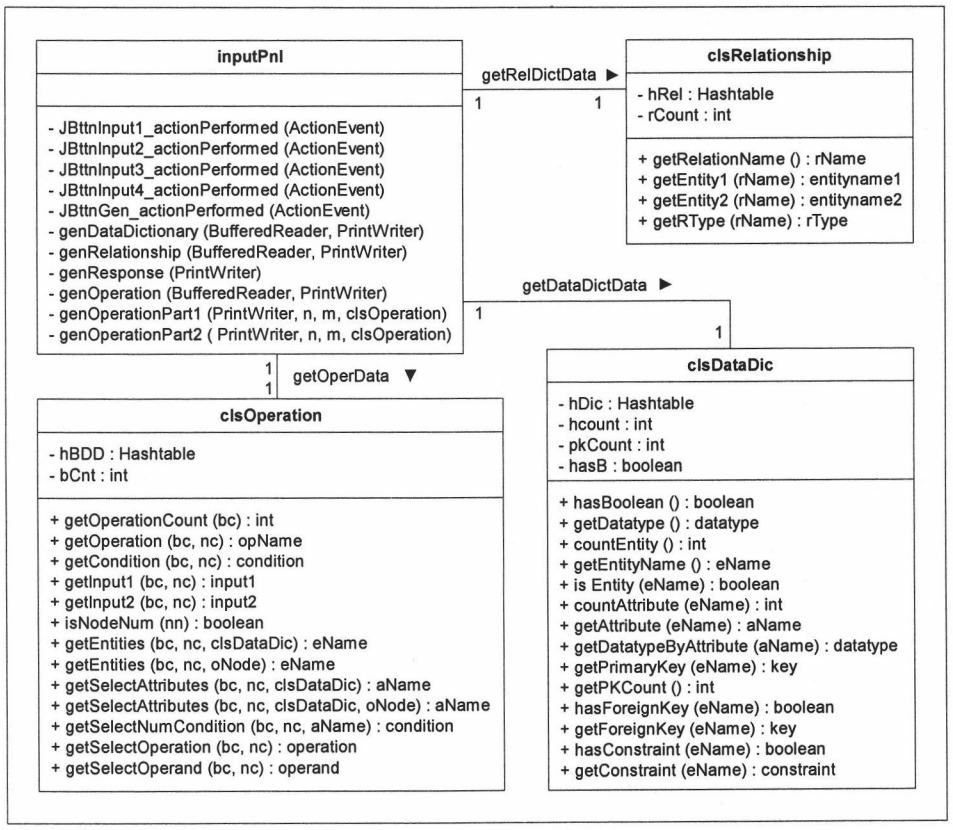
โดยเริ่มจากการอ่านข้อมูลหมายเลขและชื่อการดำเนินการพื้นฐาน เพื่อนำไปสร้างข้อกำหนดเซตของเค้าร่างการดำเนินการพื้นฐาน ตามด้วยการอ่านข้อมูลเข้า และเงื่อนไข เพื่อนำไปสร้างข้อกำหนดเซตในส่วนของการประกาศตัวแปรและข้อบังคับของการดำเนินการพื้นฐานนั้นๆ ตามลำดับ

4.3.3 คลาสไดอะแกรม

เป็นการอธิบายรายละเอียดของคลาสใช้ในเครื่องมือซอฟต์แวร์นี้ ซึ่ง 4 คลาสหลัก ดังรูปที่ 4.5 โดยรายละเอียดของคลาสต่างๆ มีดังนี้

1) คลาส inputPnl เป็นคลาสที่ใช้ในการสร้างข้อกำหนดเซต ทั้งในส่วนของกีเวนเซต และส่วนต่างๆ ของเค้าร่าง ได้แก่ ส่วนโครงสร้างของเค้าร่าง ส่วนการประกาศตัวแปร และส่วนการสร้างเพรดิเคต

2) คลาส clsDataDic เป็นคลาสสำหรับจัดการข้อมูลของพจนานุกรมข้อมูล โดยทำหน้าที่ในการเก็บข้อมูลที่มาจากแฟ้มพจนานุกรมข้อมูล และการเรียกใช้ข้อมูลเหล่านั้น



รูปที่ 4.5 คลาสไดอะแกรมของเครื่องมือซอฟต์แวร์

3) คลาส clsRelationship เป็นคลาสสำหรับจัดการข้อมูลของพจนานุกรมความสัมพันธ์ โดยทำหน้าที่ในการเก็บข้อมูลที่มาจากแฟ้มพจนานุกรมความสัมพันธ์ และการเรียกใช้ข้อมูลเหล่านั้น

4) คลาส clsOperation เป็นคลาสสำหรับจัดการข้อมูลของแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล โดยทำหน้าในการเก็บข้อมูลที่มาจากแฟ้มข้อมูลแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล และการเรียกใช้ข้อมูลเหล่านั้น

4.4 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือซอฟต์แวร์

4.4.1 เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนา

- คอมพิวเตอร์พีซี Pentium III 450 เมกกะเฮิร์ซ
- หน่วยความจำหลัก 512 เมกะไบต์
- ฮาร์ดดิสก์ความจุ 30 กิกะไบต์

4.4.2 ระบบปฏิบัติการและซอฟต์แวร์ที่ใช้ในการพัฒนา

- ระบบปฏิบัติการไมโครซอฟท์วินโดว์ส 2000
- Java API j2sdk 1.4.2
- โปรแกรม EditPlus 2.10a