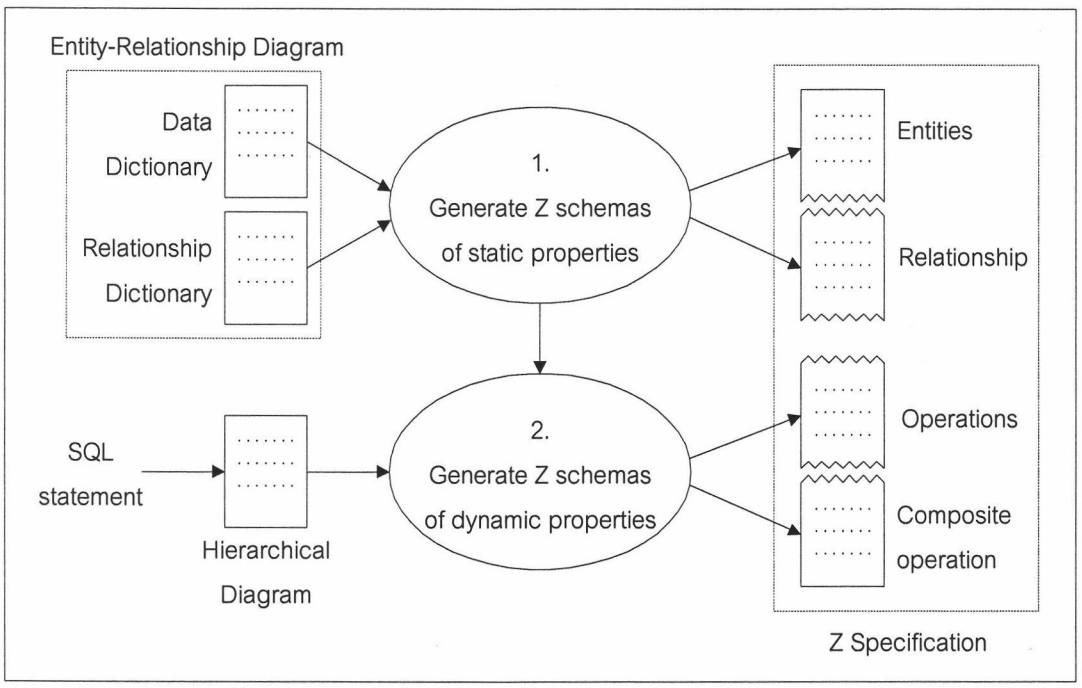
บทที่ 3

ขั้นตอนการสร้างข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ

ในบทนี้ จะกล่าวถึงขั้นตอนสร้างข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ โดยกำหนดขั้นตอนวิธีในการแปลงแผนภาพเอนทิตีและความสัมพันธ์ และแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลเป็นข้อกำหนดเซต รวมทั้งการนำเสนอกฎที่ใช้ในการแปลงแผนภาพทั้งสองเป็นข้อกำหนดเซตของโครงสร้างข้อมูลและการดำเนินการพื้นฐาน

คุณสมบัติเชิงพฤติกรรมของระบบ หมายถึง พฤติกรรมที่สามารถเกิดขึ้นของระบบใดๆ โดยอาศัยโครงสร้างข้อมูลจากแผนภาพเอนทิตีและความสัมพันธ์ เช่น การเพิ่มข้อมูล การแก้ไขข้อมูล และการค้นหาข้อมูลด้วยเงื่อนไขที่ต้องการ เป็นต้น



รูปที่ 3.1 ขั้นตอนการสร้างข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ

3.1 ขั้นตอนการสร้างข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ

ขั้นตอนการสร้างข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ ประกอบด้วย 2 ขั้นตอน ดังรูปที่ 3.1 ดังนี้

ขั้นตอนที่ 1 การสร้างเค้าร่างเซตของคุณสมบัติสถิต (Static properties)

การสร้างเค้าร่างเซตของคุณสมบัติสถิต เป็นการสร้างข้อกำหนดเซตของส่วนโครงสร้างเอนทิตีและความสัมพันธ์ โดยนำเข้าแฟ้มพจนานุกรมข้อมูลและแฟ้มพจนานุกรมความสัมพันธ์ที่เขียนขึ้นจากแผนภาพเอนทิตีและความสัมพันธ์ มาทำการแปลงเป็นข้อกำหนดเซตในส่วนของโครงสร้าง

กฎการแปลงแผนภาพเอนทิตีและความสัมพันธ์ไปเป็นข้อกำหนดเซตมีจำนวน 8 ข้อ ซึ่งจะกล่าวถึงในหัวข้อ 3.2.1 แบ่งเป็น 4 ประเภท ดังนี้

1) ชนิดข้อมูล ได้จากแฟ้มพจนานุกรมข้อมูล ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 2 ข้อ คือกฎข้อ 1-2

2) เค้าร่างเอนทิตี ได้จากแฟ้มพจนานุกรมข้อมูล ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 2 ข้อ คือกฎข้อ 3-4

3) เค้าร่างความสัมพันธ์ ได้จากแฟ้มพจนานุกรมความสัมพันธ์ ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 3 ข้อ คือกฎข้อ 5-7

4) เค้าร่างสถานะ ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 1 ข้อ คือกฎข้อ 8

ขั้นตอนที่ 2 การสร้างเค้าร่างเซตของคุณสมบัติพลวัต (Dynamic properties)

การสร้างข้อกำหนดเซตของคุณสมบัติพลวัต เป็นการสร้างข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ โดยนำเข้าแฟ้มข้อมูลแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล ซึ่งสร้างขึ้นจากการประกอบกันของการดำเนินการพื้นฐานของเครื่องมือซอฟต์แวร์ตามลำดับการทำงานของภาษาเอสคิวแอล มาทำการแปลงเป็นข้อกำหนดเซตของคุณสมบัติเชิงพฤติกรรมของระบบ

กฎการแปลงแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลไปเป็นข้อกำหนดเซตมีจำนวน 20 ข้อ ซึ่งจะกล่าวถึงในหัวข้อ 3.2.2 แบ่งเป็น 4 ประเภท ดังนี้

1) เค้าร่างการปรับปรุงข้อมูล ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 3 ข้อ คือกฎข้อ 9-11

2) เค้าร่างการค้นคืนข้อมูล ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 12 ข้อ คือกฎข้อ 12-23

3) เค้าร่างฟังก์ชัน ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 4 ข้อ คือกฎข้อ 24-27

4) เค้าร่างรวมการดำเนินการพื้นฐาน ประกอบด้วยกฎการแปลงเป็นข้อกำหนดเซตจำนวน 1 ข้อ คือกฎข้อ 28

จากขั้นตอนในการสร้างข้อกำหนดเซตทั้ง 2 ขั้นตอน จะได้แฟ้มข้อมูลผลลัพธ์ของข้อกำหนดเซตจำนวน 1 แฟ้ม ซึ่งประกอบด้วย 4 ส่วน ดังนี้

1) ข้อกำหนดเซตของโครงสร้างของเอนทิตี ซึ่งสร้างขึ้นจากแฟ้มพจนานุกรมข้อมูล

2) ข้อกำหนดเซตของความสัมพันธ์ระหว่างเอนทิตี ซึ่งสร้างขึ้นจากแฟ้มพจนานุกรมความสัมพันธ์

3) ข้อกำหนดเซตของการดำเนินการพื้นฐาน ซึ่งสร้างขึ้นจากแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล

4) ข้อกำหนดเซตของการรวมการดำเนินการพื้นฐาน ซึ่งสร้างขึ้นจากแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล

3.2 กฎการแปลงแผนภาพเอนทิตีและความสัมพันธ์ และแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลเป็นข้อกำหนดเซต

ในงานวิจัย "เครื่องมือแปลงแบบจำลองเอนทิตีและความสัมพันธ์เป็นข้อกำหนดรูปนัยในรูปสัญกรณ์เซต" [4] ได้ทำการแปลงแผนภาพเอนทิตีและความสัมพันธ์ไปเป็นข้อกำหนดเซต โดยข้อกำหนดเซตที่ได้แบ่งเป็น 2 ส่วน คือ

1) ส่วนคุณสมบัติของโครงสร้าง ประกอบด้วย การนิยามชนิดข้อมูล เค้าร่างเอนทิตี เค้าร่างเพิ่มเติมของเอนทิตี และเค้าร่างความสัมพันธ์

2) ส่วนคุณสมบัติพฤติกรรม ประกอบด้วย การเพิ่มระเบียน การแก้ไขระเบียน และการลบระเบียน

ซึ่งวิทยานิพนธ์นี้ได้ทำการปรับปรุงแก้ไขและเพิ่มคุณสมบัติพฤติกรรมชนิดใหม่ โดยกฎการแปลงแผนภาพเอนทิตีและความสัมพันธ์ และแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลเป็นข้อกำหนดเซต สามารถสรุปได้ดังนี้

3.2.1 กฎการแปลงแผนภาพเอนทิตีและความสัมพันธ์ เป็นข้อกำหนดเซตของคุณสมบัติสถิต

ข้อกำหนดเซตของคุณสมบัติสถิตของแผนภาพเอนทิตีและความสัมพันธ์ คือส่วนโครงสร้างของเอนทิตีและความสัมพันธ์ โดยนำเข้าข้อมูลจากแฟ้มพจนานุกรมข้อมูลและแฟ้มพจนานุกรมความสัมพันธ์ของแผนภาพเอนทิตีและความสัมพันธ์ และแปลงเป็นข้อกำหนดเซตตามกฎข้อที่ 1-8 ซึ่งสามารถแบ่งตามประเภทของแฟ้มข้อมูลเข้าได้ 2 ประเภท ดังนี้

3.2.1.1 กฎการแปลงแฟ้มพจนานุกรมข้อมูลเป็นข้อกำหนดเซตของคุณสมบัติสถิต

กฎการแปลงแฟ้มพจนานุกรมข้อมูลเป็นข้อกำหนดเซตของคุณสมบัติสถิต ประกอบด้วยส่วนโครงสร้างของเอนทิตี ดังกฎข้อที่ 1-4 ดังนี้

กฎข้อที่ 1 ชนิดข้อมูลของลักษณะประจำของเอนทิตีใดๆ ในแฟ้มพจนานุกรมข้อมูลของแผนภาพเอนทิตีและความสัมพันธ์ (ยกเว้นชนิดข้อมูลแบบจำนวนนับและจำนวนธรรมชาติซึ่งมีในสัญกรณ์เซต และชนิดข้อมูลแบบบูลลีนซึ่งจะกล่าวในกฎข้อที่ 2) จะถูกสร้างเป็นกีเวนเซตของชนิดข้อมูลด้วยข้อกำหนดเซต ดังนี้

$[AT_1, AT_2, \ldots, AT_n]$

โดยที่ AT คือ ชื่อชนิดข้อมูลของลักษณะประจำ

กฎข้อที่ 2 ชนิดข้อมูลแบบบูลลีนของลักษณะประจำของเอนทิตีใดๆ ในแฟ้มพจนานุกรมข้อมูลของแผนภาพเอนทิตีและความสัมพันธ์ จะถูกสร้างเป็นนิยามชนิดอิสระ (Free type definition) ด้วยข้อกำหนดเซต ดังนี้

BOOLEAN ::= True | False

โดยที่ BOOLEAN คือ ชื่อชนิดอิสระ

True | False คือ ค่าที่สามารถเป็นไปได้ของ BOOLEAN

กฎข้อที่ 3 รายละเอียดลักษณะประจำของเอนทิตีใดๆ ในแฟ้มพจนานุกรมข้อมูลของแผนภาพเอนทิตีและความสัมพันธ์ จะถูกสร้างเป็นเค้าร่างเอนทิตีด้วยข้อกำหนดเซต ดังนี้

Entity

$an_1 : AT_1$
$\vdots$
$an_n : AT_n$

CS

โดยที่ *Entity* คือ ชื่อเอนทิตี

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำ

$AT_1 \ldots AT_n$ คือ ชนิดข้อมูลของลักษณะประจำ

CS คือ ข้อบังคับของลักษณะประจำ (ถ้ามี)

กฎข้อที่ 4 เอนทิตีใดๆ ในแฟ้มพจนานุกรมข้อมูลของแผนภาพเอนทิตีและความสัมพันธ์ จะถูกสร้างเค้าร่างเพิ่มเติมของเอนทิตีในข้อกำหนดเซต โดยกำหนดเซตของสมาชิกในเอนทิตี คีย์หลัก และคีย์นอก ดังนี้

EntityExt

$entitySet : \mathbb{F}\ Entity$
$entityKey : \mathbb{F}\ (AT_1 \times AT_2 \times \ldots \times AT_n)$
$entityAtt : (AT_1 \times AT_2 \times \ldots \times AT_n) \rightarrow\!\!\!\!\!\!\rightarrow Entity$

$\forall\ Entity1, Entity2 : Entity$ ①
$\quad \mid Entity1 \in entitySet \wedge Entity2 \in entitySet \wedge Entity1 \neq Entity2$
$\forall\ EntityVar2 : Entity \bullet$ ②
$\quad (\exists\ EntityVar1 : Entity1 \bullet EntityVar2.an_i = Entity2.an_i)$
$dom\ entityAtt \subseteq entityKey$ ③

โดยที่ *EntityExt* คือ ชื่อเค้าร่างเพิ่มเติมของเอนทิตี

entitySet คือ เซตของสมาชิกของเอนทิตี

entityKey คือ เซตของคีย์หลักของเอนทิตี ซึ่งสามารถมีได้มากกว่า 1 ลักษณะประจำ

entityAtt คือ เซตซึ่งแสดงค่าสมาชิกของเอนทิตีที่สามารถระบุโดยคีย์หลัก

เพรดิเคต ① เป็นการกำหนดข้อบังคับของคีย์หลัก [4]

Entity1 คือ เอนทิตีลำดับที่ 1

Entity2 คือ เอนทิตีลำดับที่ 2

เพรดิเคต ② เป็นการกำหนดข้อบังคับของคีย์นอก [4] ซึ่งคีย์นอก 1 คีย์จะอธิบายด้วย 1 เพรดิเคต

EntityVar2 คือ เซตของเอนทิตีหลัก (Parent) ของคีย์นอก

EntityVar1 คือ เซตของเอนทิตีลูก (Child) ของคีย์นอก

เพรดิเคต ③ คือ การกำหนดโดเมนของเซต

3.2.1.2 กฎการแปลงแฟ้มพจนานุกรมความสัมพันธ์เป็นข้อกำหนดเซตของคุณสมบัติสถิต

กฎการแปลงแฟ้มพจนานุกรมความสัมพันธ์เป็นข้อกำหนดเซตของคุณสมบัติสถิต ประกอบด้วยส่วนโครงสร้างความสัมพันธ์ของเอนทิตี ดังกฎข้อที่ 5-7 ดังนี้

กฎข้อที่ 5 ความสัมพันธ์ของเอนทิตีแบบหนึ่งต่อหนึ่ง ในแฟ้มพจนานุกรมความสัมพันธ์ของแผนภาพเอนทิตีและความสัมพันธ์ จะถูกสร้างเป็นเค้าร่างความสัมพันธ์แบบหนึ่งต่อหนึ่งด้วยข้อกำหนดเซต [4] ดังนี้

RelationshipEntity1Entity2

$rel : \mathbb{F}\ (Entity1 \times Entity2)$

$\forall rel1, rel2 : rel \bullet$

$second\ (rel1) = second\ (rel2) \Rightarrow first\ (rel1) = first\ (rel2) \land$

$first\ (rel1) = first\ (rel2) \Rightarrow second\ (rel1) = second\ (rel2)$

โดยที่ *RelationshipEntity1Entity2* คือ ชื่อเค้าร่างความสัมพันธ์ของ 2 เอนทิตี

rel คือ ชื่อความสัมพันธ์

เพรดิเคต เป็นการอธิบายความสัมพันธ์แบบหนึ่งต่อหนึ่งของ 2 เอนทิตี

rel1 คือ ความสัมพันธ์ลำดับที่ 1

rel2 คือ ความสัมพันธ์ลำดับที่ 2

กฎข้อที่ 6 ความสัมพันธ์ของเอนทิตีแบบหนึ่งต่อหลาย ในแฟ้มพจนานุกรมความสัมพันธ์ของแผนภาพเอนทิตีและความสัมพันธ์ จะถูกสร้างเป็นเค้าร่างความสัมพันธ์แบบหนึ่งต่อหลายด้วยข้อกำหนดเซต [4] ดังนี้

RelationshipEntity1Entity2

$rel : \mathbb{F}\ (Entity1 \times Entity2)$

$\forall rel1, rel2 : rel \bullet$

$second\ (rel1) = second\ (rel2) \Rightarrow first\ (rel1) = first\ (rel2)$

โดยที่ *RelationshipEntity1Entity2* คือ ชื่อเค้าร่างความสัมพันธ์ของ 2 เอนทิตี

rel คือ ชื่อความสัมพันธ์

เพรดิเคต เป็นการอธิบายความสัมพันธ์แบบหนึ่งต่อหลายของ 2 เอนทิตี

rel1 คือ ความสัมพันธ์ลำดับที่ 1

rel2 คือ ความสัมพันธ์ลำดับที่ 2

กฎข้อที่ 7 ความสัมพันธ์ของเอนทิตีแบบหลายต่อหนึ่ง ในแฟ้มพจนานุกรมความสัมพันธ์ของแผนภาพเอนทิตีและความสัมพันธ์ จะถูกสร้างเป็นเค้าร่างความสัมพันธ์แบบหลายต่อหนึ่งด้วยข้อกำหนดเซต [4] ดังนี้

RelationshipEntity1Entity2

$rel : \mathbb{F}\ (Entity1 \times Entity2)$

$\forall rel1, rel2 : rel \bullet$

$first\ (rel1) = first\ (rel2) \Rightarrow second\ (rel1) = second\ (rel2)$

โดยที่ *RelationshipEntity1Entity2* คือ ชื่อเค้าร่างความสัมพันธ์ของ 2 เอนทิตี

rel คือ ชื่อความสัมพันธ์

เพรดิเคต เป็นการอธิบายความสัมพันธ์แบบหลายต่อหนึ่งของ 2 เอนทิตี

rel1 คือ ความสัมพันธ์ลำดับที่ 1

rel2 คือ ความสัมพันธ์ลำดับที่ 2

กฎข้อที่ 8 ทุกแผนภาพเอนทิตีและความสัมพันธ์ จะมีเค้าร่างแสดงสถานะ ซึ่งประกอบด้วยเค้าร่างแสดงความสำเร็จ (Success) และเค้าร่างแสดงการไม่พบข้อมูล (NotFound) ดังนี้

Response := Success | NotFound

Success

r! : Response

---

r! = Success

NotFound

r! : Response

---

r! = NotFound

โดยที่ *Response* คือ ชื่อชนิดอิสระ ซึ่งมีค่าที่สามารถเป็นไปได้คือ Success และ NotFound

Success คือ ชื่อเค้าร่างแสดงความสำเร็จ

r! คือ ชื่อผลลัพธ์ของค่าความสำเร็จในเค้าร่างแสดงความสำเร็จ หรือชื่อผลลัพธ์ของค่าการไม่พบข้อมูลในเค้าร่างแสดงการไม่พบข้อมูล

NotFound คือ ชื่อเค้าร่างแสดงการไม่พบข้อมูล

เพรดิเคต คือ การแสดงค่าความสำเร็จ หรือการไม่พบข้อมูล

3.2.2 กฎการแปลงแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล เป็นข้อกำหนดเซตของคุณสมบัติพลวัต

ข้อกำหนดเซตของคุณสมบัติพลวัตของแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล คือ ข้อกำหนดเซตของการดำเนินการพื้นฐานและการรวมการดำเนินการพื้นฐานซึ่งเป็นคุณสมบัติเชิงพฤติกรรมของระบบ โดยนำเข้าข้อมูลจากแฟ้มแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลและแปลงเป็นข้อกำหนดเซต ตามกฎข้อที่ 9-28 ดังนี้

กฎข้อที่ 9 การดำเนินการพื้นฐานเพิ่มข้อมูล จะถูกสร้างเป็นเค้าร่างก่อนเพิ่มข้อมูล (Init) และเค้าร่างเพิ่มข้อมูล (Insert) ด้วยข้อกำหนดเซต โดยการเพิ่มข้อมูลจะรวมข้อมูลใหม่เข้ากับเซตของข้อมูลเดิม ดังนี้

Init

Δ *EntityExt*

$entitySet' = \varnothing$ ①

$entityKey' = \varnothing$ ②

$entityAtt' = \varnothing$ ③

Insert

Δ *EntityExt*

entityKey? : (AT_1 x AT_2 x ... x AT_n)

newValue? : Entity

---

entityKey? $\notin$ *dom entityAtt* ④

CS ⑤

entitySet' = *entitySet* $\cup$ {*newValue?*} ⑥

โดยที่ *Init* คือ ชื่อเค้าร่างก่อนการเพิ่มข้อมูล

เพรดิเคต ① เป็นการกำหนดให้เอนทิตีเป็นเซตว่าง

เพรดิเคต ② เป็นการกำหนดให้ค่าซึ่งเป็นเซตของคีย์หลักเป็นเซตว่าง

เพรดิเคต ③ เป็นการกำหนดให้ค่าของคีย์หลักซึ่งชี้ไปยังเอนทิตีเป็นเซตว่าง

Insert คือ ชื่อเค้าร่างการเพิ่มข้อมูลในเอนทิตี

Δ *EntityExt* คือ การรวมเค้าร่างเพิ่มเติมของเอนทิตี เพื่อการแก้ไขเปลี่ยนแปลง

entityKey? คือ ข้อมูลเข้า คีย์หลักที่ต้องการเพิ่มในเอนทิตี

newValue? คือ ข้อมูลเข้า ค่าข้อมูลที่ต้องการเพิ่มในเอนทิตี

เพรดิเคต ④ เป็นการตรวจสอบการไม่มีอยู่ของคีย์หลักที่ต้องการเพิ่มในเอนทิตี

เพรดิเคต ⑤ เป็นข้อบังคับของลักษณะประจำ (ถ้ามี)

เพรดิเคต ⑥ เป็นการนำค่าข้อมูลเข้ามายูเนียนกับเซตของข้อมูลเดิมของเอนทิตีนั้น [4]

กฎข้อที่ 10 การดำเนินการพื้นฐานของการแก้ไขข้อมูล จะถูกสร้างเป็นเค้าร่างการแก้ไขข้อมูล (Update) ด้วยข้อกำหนดเซต โดยระบุค่าคีย์หลักและข้อมูลใหม่ที่ต้องการแก้ไข ซึ่งจะนำไปทำการแทนที่ข้อมูลในเซตข้อมูลเดิมด้วยคีย์หลักที่กำหนด ดังนี้

Update

Δ *EntityExt*

entityKey? : (AT_1 x AT_2 x ... x AT_n)

newValue? : Entity

---

entityKey? $\in$ *dom entityAtt* ①

CS ②

entityAtt' = *entityAtt* $\oplus$ { *entityKey?* $\mapsto$ *newValue?* } ③

โดยที่ *Update* คือ ชื่อเค้าร่างการแก้ไขข้อมูลในเอนทิตี

Δ *EntityExt* คือ การรวมเค้าร่างเพิ่มเติมของเอนทิตี เพื่อการแก้ไขเปลี่ยนแปลง

entityKey? คือ ข้อมูลเข้า คีย์หลักที่ต้องการแก้ไขของเอนทิตี

newValue? คือ ข้อมูลเข้า ค่าข้อมูลใหม่ที่ต้องการแก้ไข

เพรดิเคต ① เป็นการตรวจสอบการมีอยู่ของคีย์หลักที่ต้องการแก้ไข

เพรดิเคต ② CS เป็นข้อบังคับของลักษณะประจำ (ถ้ามี)

เพรดิเคต ③ เป็นการแทนค่าข้อมูลใหม่ลงในข้อมูลเดิมตามค่าของคีย์หลักที่กำหนด

กฎข้อที่ 11 การดำเนินการพื้นฐานของการลบข้อมูล จะถูกสร้างเป็นเค้าร่างการลบข้อมูล (Delete) ด้วยข้อกำหนดเซต โดยระบุค่าคีย์หลักและข้อมูลที่ต้องการลบ ซึ่งจะนำเซตของข้อมูลเดิมลบด้วยเซตของข้อมูลที่ต้องการลบ หากข้อมูลที่ต้องการลบเป็นคีย์นอกของเอนทิตีอื่น ข้อมูลของคีย์นอกในเอนทิตีนั้นจะถูกลบออกด้วย ดังนี้

Delete

Δ *EntityExt*
entityKey? : $(AT_1 \times AT_2 \times ... \times AT_n)$
delEntity? : *Entity*
$DeleteEntity_i$

$entityKey? \in dom\ entityAtt$ ①
$delEntity_i?.an_{pk} = delEntity?.an_{fk}$ ②
$entitySet' = entitySet \setminus \{delEntity?\}$ ③

โดยที่ *Delete* คือ ชื่อเค้าร่างการลบข้อมูลในเอนทิตี

Δ *EntityExt* คือ การรวมเค้าร่างเพิ่มเติมของเอนทิตี เพื่อการแก้ไขเปลี่ยนแปลง

entityKey? คือ ข้อมูลเข้าซึ่งเป็นคีย์หลักที่ต้องการลบจากเอนทิตี

delEntity? คือ ข้อมูลเข้าซึ่งเป็นค่าที่ต้องการลบจากเอนทิตี

$DeleteEntity_i$ คือ ชื่อเค้าร่างการลบข้อมูลของเอนทิตีที่เป็นลูก (ถ้ามี)

เพรดิเคต ① เป็นการตรวจสอบการมีอยู่ของคีย์หลักที่ต้องการลบ

เพรดิเคต ② เป็นการตรวจสอบค่าคีย์หลักของเอนทิตีลูก (ถ้ามี) กับค่าคีย์นอกที่ต้องการลบ โดย $delEntity_i?.an_{pk}$ คือคีย์หลักของเอนทิตีลูก และ $delEntity?.an_{fk}$ คือคีย์นอกของเอนทิตีที่ต้องการลบ

เพรดิเคต ③ เป็นการลบค่าข้อมูลออกจากข้อมูลของเอนทิตี [4]

กฎข้อที่ 12 การดำเนินการพื้นฐานของการเลือกแถว เป็นการเลือกแถวที่ต้องการจากเอนทิตีด้วยค่าของลักษณะประจำตามที่กำหนด จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างการเลือกแถว (Select) ด้วยข้อกำหนดเซต โดยเลือกเฉพาะสมาชิกของเซตที่มีค่าของลักษณะประจำตรงตามค่าที่กำหนด ดังนี้

R

$an_1 : AT_1$
...
$an_n : AT_n$

Select

input1? : $\mathbb{F}$ *Entity*
$an_iValue? : AT_i$
output! : $\mathbb{F}$ *R*

$input1? \neq \varnothing$ ①
$\forall out : output!; in1 : input1? \mid in1.an_i = an_iValue1? \bullet out = in1$ ②

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างการเลือกแถว ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเค้าร่างเอนทิตีที่เป็นข้อมูลเข้า (*input1?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี

Select คือ ชื่อเค้าร่างการเลือกแถว

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการทำการเลือกแถว

an_n*Value?* คือ ข้อมูลเข้าซึ่งเป็นค่าลักษณะประจำที่เป็นเงื่อนไขในการเลือกแถว

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างการเลือกแถว

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นข้อมูลเข้าต้องมีข้อมูลอยู่

เพรดิเคต ② กำหนดให้ข้อมูลผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นข้อมูลเข้า ($out = in1$) โดยเลือกเฉพาะข้อมูลที่มีค่าของลักษณะประจำตรงตามค่าที่กำหนด ($in1.an_i = an_i Value1?$)

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้า

กฎข้อที่ 13 การดำเนินการพื้นฐานของการเลือกสดมภ์ เป็นการเลือกสดมภ์ที่ต้องการจากเอนทิตี จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างการเลือกสดมภ์ (Project) ด้วยข้อกำหนดเซต โดยเลือกเฉพาะลักษณะประจำที่กำหนด ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

Project

$input1? : \mathbb{F}\ Entity$

$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①

$\forall out : output!; in1 : input1?$ ②

$\bullet\ out.an_1 = in1.an_1 \wedge$

$\ldots \quad \wedge$

$out.an_n = in1.an_n$

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างการเลือกสดมภ์ ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำตามสดมภ์ที่ต้องการ

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี

Project คือ ชื่อเค้าร่างการเลือกสดมภ์

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการทำการเลือกสดมภ์

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างการเลือกสดมภ์

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นข้อมูลเข้าต้องมีข้อมูลอยู่

เพรดิเคต ② กำหนดให้ข้อมูลผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นข้อมูลเข้า ($out.an_1 = in1.an_1 \land ... \land out.an_n = in1.an_n$) โดยเลือกเฉพาะลักษณะประจำตามที่ระบุไว้

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้า

กฎข้อที่ 14 การดำเนินการพื้นฐานของการจอยแบบธรรมชาติ เป็นการรวมข้อมูลของ 2 เอนทิตีที่มีลักษณะประจำร่วมกัน (common attribute) เข้าด้วยกัน จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างการจอยแบบธรรมชาติ (Join) ด้วยข้อกำหนดเซต โดยเลือกเฉพาะสมาชิกจากทั้ง 2 เอนทิตีที่ค่าของลักษณะประจำร่วมกันมีค่าเท่ากัน ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

Join

$input1? : \mathbb{F}\ Entity1$

$input2? : \mathbb{F}\ Entity2$

$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①

$input2? \neq \varnothing$ ②

$\forall\ out : output!; in1 : input1?; in2 : input2$ ③

$\mid in1.an_{fk} = in2.an_{pk}$

$\bullet\ out.an_1 = in1.an_1 \land$

$...\qquad\qquad \land$

$out.an_n = in2.an_n$

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างการจอยแบบธรรมชาติ ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตีรวมกัน

$an_1 ... an_n$ คือ ชื่อลักษณะประจำของเอนทิตี

Join คือ ชื่อเค้าร่างการจอยแบบธรรมชาติ

input1? คือ ข้อมูลเข้า ชื่อเอนทิตีที่ต้องการจอยลำดับที่ 1

input2? คือ ข้อมูลเข้า ชื่อเอนทิตีที่ต้องการจอยลำดับที่ 2

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างการจอยแบบธรรมชาติ

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1 ต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2 ต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้ข้อมูลผลลัพธ์ มีค่าเท่ากับเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี ($out.an_1 = in1.an_1 \land ... \land out.an_n = in2.an_n$) โดยเลือกเฉพาะข้อมูลที่ค่าของคีย์นอกเท่ากับค่าของคีย์หลัก ($in1.an_{fk} = in2.an_{pk}$)

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1

in2 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2

กฎข้อที่ 15 การดำเนินการพื้นฐานของการยูเนียน เป็นการรวมสมาชิกของ 2 เอนทิตีที่มีลักษณะประจำเหมือนกันทุกประการ จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างยูเนียน (Union) ด้วยข้อกำหนดเซต โดยการยูเนียนทั้ง 2 เอนทิตีเข้าด้วยกัน ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

Union

Ξ *Entity1*

Ξ *Entity2*

input1? : $\mathbb{F}$ *Entity1*

input2? : $\mathbb{F}$ *Entity2*

output! : $\mathbb{F}$ *R*

θ *Entity1* = θ *Entity2* ①

$\forall$ *out* : $\mathbb{F}$ *output!*; *in1* : *input1?*; *in2* : *input2* ②

- *out* = {*in1*} $\cup$ {*in2*}

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างยูเนียน ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี (*input1?* และ *input2?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี

Union คือ ชื่อเค้าร่างการยูเนียน

Ξ *Entity1* คือ การรวมเค้าร่างของเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1 โดยไม่มีการเปลี่ยนแปลงข้อมูล

Ξ *Entity2* คือ การรวมเค้าร่างของเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2 โดยไม่มีการเปลี่ยนแปลงข้อมูล

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการยูเนียนลำดับที่ 1

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการยูเนียนลำดับที่ 2

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างยูเนียน

เพรดิเคต ① เป็นการตรวจสอบความเหมือนกันของโครงสร้างทั้ง 2 เอนทิตี

เพรดิเคต ② กำหนดให้เอนทิตีผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี ทำการยูเนียนกัน (*out* = {*in1*} $\cup$ {*in2* })

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1

in2 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2

กฎข้อที่ 16 การดำเนินการพื้นฐานของการอินเตอร์เซค เป็นการเลือกสมาชิกที่ซ้ำกันของ 2 เอนทิตีที่มีลักษณะประจำเหมือนกันทุกประการ จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างอินเตอร์เซค (Intersect) ด้วยข้อกำหนดเซต โดยการอินเตอร์เซคทั้ง 2 เอนทิตีเข้าด้วยกัน ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

Intersect

Ξ *Entity1*

Ξ *Entity2*

input1? : $\mathbb{F}$ *Entity1*

input2? : $\mathbb{F}$ *Entity2*

output! : $\mathbb{F}$ *R*

θ *Entity1* = θ *Entity2* ①

$\forall$ *out* : $\mathbb{F}$ *output!; in1* : *input1?; in2* : *input2* ②

- *out* = {*in1*} $\cap$ {*in2*}

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างอินเตอร์เซค ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี (*input1? และ input2?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี

Intersect คือ ชื่อเค้าร่างอินเตอร์เซค

Ξ *Entity1* คือ การรวมเค้าร่างของเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1 โดยไม่มีการเปลี่ยนแปลงข้อมูล

Ξ *Entity2* คือ การรวมเค้าร่างของเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2 โดยไม่มีการเปลี่ยนแปลงข้อมูล

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการอินเตอร์เซคลำดับที่ 1

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการอินเตอร์เซคลำดับที่ 2

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างอินเตอร์เซค

เพรดิเคต ① เป็นการตรวจสอบความเหมือนกันของโครงสร้างทั้ง 2 เอนทิตี

เพรดิเคต ② กำหนดให้เอนทิตีผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี ทำการอินเตอร์เซคกัน (*out* = {*in1*} $\cap$ {*in2*})

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1

in2 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2

กฎข้อที่ 17 การดำเนินการพื้นฐานของการหาผลต่าง เป็นการหาผลต่างของ 2 เอนทิตีที่มีลักษณะประจำเหมือนกัน จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างการหาผลต่าง (Except) ด้วยข้อกำหนดเซต โดยการลบเซตของสมาชิกของเอนทิตีลำดับที่ 2 ออกจากเซตของเอนทิตีลำดับแรก ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

Except

Ξ *Entity1*

Ξ *Entity2*

input1? : $\mathbb{F}$ *Entity1*

input2? : $\mathbb{F}$ *Entity2*

output! : $\mathbb{F}$ *R*

θ *Entity1* = θ *Entity2* ①

$\forall$ *out* : $\mathbb{F}$ *output!*; *in1* : *input1?*; *in2* : *input2* ②

• *out* = {*in1*} \ {*in2*}

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างการหาผลต่าง ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี (*input1?* และ *input2?*)

an_1 ... an_n คือ ชื่อลักษณะประจำของเอนทิตี

Except คือ ชื่อเค้าร่างการหาผลต่าง

Ξ *Entity1* คือ การรวมเค้าร่างของเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1 โดยไม่มีการเปลี่ยนแปลงข้อมูล

Ξ *Entity2* คือ การรวมเค้าร่างของเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2 โดยไม่มีการเปลี่ยนแปลงข้อมูล

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการหาผลต่างลำดับที่ 1 (ตัวตั้ง)

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่ต้องการหาผลต่างลำดับที่ 2 (ตัวลบ)

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างการหาผลต่าง

เพรดิเคต ① เป็นการตรวจสอบความเหมือนกันของโครงสร้างทั้ง 2 เอนทิตี

เพรดิเคต ② กำหนดให้ข้อมูลผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นข้อมูลเข้าทั้ง 2 เอนทิตี ทำการลบกัน (*out* = {*in1*} \ {*in2* })

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 1

in2 คือ ชื่อเอนทิตีที่เป็นข้อมูลเข้าลำดับที่ 2

กฎข้อที่ 18 การดำเนินการพื้นฐานของเงื่อนไขอิน เป็นการเลือกสมาชิกในเอนทิตีที่เป็นคิวรี่หลัก โดยเลือกเฉพาะที่มีลักษณะประจำที่กำหนดอยู่ในเอนทิตีที่เป็นคิวรี่ย่อย (subquery) จะถูกสร้างเป็นเค้าร่างผลลัพธ์

(R) และเค้าร่างอิน (In) ด้วยข้อกำหนดเซต โดยการเลือกเฉพาะสมาชิกในเอนทิตีลำดับที่ 1 ที่มีค่าลักษณะประจำที่กำหนดเป็นสมาชิกของเอนทิตีลำดับที่ 2 ดังนี้

$$
\begin{array}{|l}
\hline
R \\
an_1 : AT_1 \\
\ldots \\
an_n : AT_n \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
In \\
input1? : \mathbb{F}\ Entity1 \\
input2? : \mathbb{F}\ Entity2 \\
output! : \mathbb{F}\ R \\
\hline
input1? \neq \varnothing \quad ① \\
input2? \neq \varnothing \quad ② \\
\forall\ out : output!; in1 : input1?; in2 : input2 \quad ③ \\
\quad \mid in1.an_i \in in2.an_i \\
\quad \bullet\ out = in1 \\
\hline
\end{array}
$$

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างอิน ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นคิวรี่หลัก (*input1?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี ซึ่งเหมือนเอนทิตีที่เป็นคิวรี่หลัก

In คือ ชื่อเค้าร่างอิน

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่หลักของเงื่อนไขอิน

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่ย่อยของเงื่อนไขอิน

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างอิน

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่หลักต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่ย่อยต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้ข้อมูลผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นคิวรี่หลัก ($out = in1$) โดยเลือกเฉพาะข้อมูลที่ค่าของลักษณะประจำที่กำหนดเป็นสมาชิกของเอนทิตีที่เป็นคิวรี่ย่อย ($in1.an_i \in in2.an_i$)

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

in2 คือ ชื่อเอนทิตีที่เป็นคิวรี่ย่อย

กฎข้อที่ 19 การดำเนินการพื้นฐานของเงื่อนไขน็อทอิน เป็นการเลือกสมาชิกในเอนทิตีที่เป็นคิวรี่หลักโดยเลือกเฉพาะที่ลักษณะประจำที่กำหนดไม่อยู่ในเอนทิตีที่เป็นคิวรี่ย่อย จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างน็อทอิน (NotIn) ด้วยข้อกำหนดเซต โดยการเลือกเฉพาะสมาชิกในเอนทิตีลำดับที่ 1 ที่ค่าลักษณะประจำที่กำหนดไม่เป็นสมาชิกของเอนทิตีลำดับที่ 2 ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

NotIn

$input1? : \mathbb{F}\ Entity1$

$input2? : \mathbb{F}\ Entity2$

$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①

$input2? \neq \varnothing$ ②

$\forall\ out : output!; in1 : input1?; in2 : input2$ ③

$\quad | \ in1.an_i \notin in2.an_i$

$\quad \bullet\ out = in1$

โดยที่ R คือ เค้าร่างผลลัพธ์ของเค้าร่างน็อทอิน ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นคิวรี่หลัก (*input1?*)

an_1 ... an_n คือ ชื่อลักษณะประจำของเอนทิตี ซึ่งเหมือนเอนทิตีที่เป็นคิวรี่หลัก

NotIn คือ ชื่อเค้าร่างน็อทอิน

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่หลักของเงื่อนไขน็อทอิน

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่ย่อยของเงื่อนไขน็อทอิน

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างน็อทอิน

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่หลัก ต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่ย่อย ต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้ข้อมูลผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นคิวรี่หลัก ($out = in1$) โดยเลือกเฉพาะข้อมูลที่ค่าของลักษณะประจำที่กำหนดไม่เป็นสมาชิกของเอนทิตีที่เป็นคิวรี่ย่อย ($in1.an_i \notin in2.an_i$)

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

in2 คือ ชีอเอนทิตีที่เป็นคิวรี่ย่อย

กฎข้อที่ 20 การดำเนินการพื้นฐานของเงื่อนไขเอ็กสีซท์ เป็นการเลือกสมาชิกในเอนทิตีที่เป็นคิวรี่หลักและมีอยู่ในเอนทิตีที่เป็นคิวรี่ย่อย จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างเอ็กสีซท์ (Exists) ด้วยข้อกำหนดเซต โดยการเลือกเฉพาะสมาชิกในเอนทิตีลำดับที่ 2 ซึ่งมีค่าลักษณะประจำครอบคลุมค่าลักษณะประจำทั้งหมดของสมาชิกในเอนทิตีลำดับที่ 1 ดังนี้

R

$an_1 : AT_1$

...

$an_n : AT_n$

Exists

$input1? : \mathbb{F}\ Entity1$
$input2? : \mathbb{F}\ Entity2$
$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①
$input2? \neq \varnothing$ ②
$\forall\ out : output!; in1 : input1?; in2 : input2$ ③
$\quad \mid in1.an_1 \in in2.an_1 \land$
$\quad\quad \ldots \quad\quad \land$
$\quad\quad in1.an_n \in in2.an_n$
$\quad \bullet\ out = in2$

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างเอ็กสีซท์ ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นคิวรี่ย่อย (*input2?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี ซึ่งเหมือนเอนทิตีที่เป็นคิวรี่ย่อย

Exists คือ ชื่อเค้าร่างเอ็กสีซท์

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่หลักของเงื่อนไขเอ็กสีซท์

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่ย่อยของเงื่อนไขเอ็กสีซท์

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างเอ็กสีซท์

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่หลัก ต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่ย่อย ต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้ข้อมูลผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นคิวรี่ย่อย ($out = in2$) โดยเลือกเฉพาะข้อมูลที่สมาชิกของคิวรี่หลักมีอยู่ในคิวรี่ย่อย ($in1.an_1 \in in2.an_1 \land \ldots \land in1.an_n \in in2.an_n$)

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

in2 คือ ชีอเอนทิตีที่เป็นคิวรี่ย่อย

กฎข้อที่ 21 การดำเนินการพื้นฐานของเงื่อนไขน็อทเอ็กสีซท์ เป็นการเลือกสมาชิกในเอนทิตีที่เป็นคิวรี่หลักแต่ไม่มีอยู่ในเอนทิตีที่เป็นคิวรี่ย่อย จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างน็อทเอ็กสีซท์ (NotExists) ด้วยข้อกำหนดเซต โดยการเลือกเฉพาะสมาชิกในเอนทิตีลำดับที่ 1 ซึ่งค่าลักษณะประจำทุกค่าไม่เป็นสมาชิกของเอนทิตีลำดับที่ 2 ดังนี้

R

$an_1 : AT_1$
...
$an_n : AT_n$

NotExists

$input1? : \mathbb{F}\ Entity1$
$input2? : \mathbb{F}\ Entity2$
$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①
$input2? \neq \varnothing$ ②
$\forall\ out : output!; in1 : input1?; in2 : input2$ ③
$\quad | \ in1.an_1 \notin in2.an_1 \land$
$\quad\quad ... \quad\quad \land$
$\quad\quad in1.an_n \notin in2.an_n$
$\quad \bullet\ out = in1$

โดยที่

R คือ เค้าร่างผลลัพธ์ของเค้าร่างน็อทเอ็กสีซท์ ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นคิวรี่หลัก (*input1?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี ซึ่งเหมือนเอนทิตีที่เป็นคิวรี่หลัก

NotExists คือ ชื่อเค้าร่างน็อทเอ็กสีซท์

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่หลักของเงื่อนไขน็อทเอ็กสีซท์

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่ย่อยของเงื่อนไขน็อทเอ็กสีซท์

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างน็อทเอ็กสีซท์

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่หลัก ต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่ย่อย ต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้เอนทิตีผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นคิวรี่หลัก ($out = in1$) โดยเลือกเฉพาะข้อมูลที่ไม่เป็นสมาชิกของเอนทิตีที่เป็นคิวรี่ย่อย ($in1.an_1 \notin in2.an_1 \land \ldots \land in1.an_n \notin in2.an_n$)

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

in2 คือ ชีอเอนทิตีที่เป็นคิวรี่ย่อย

กฎข้อที่ 22 การดำเนินการพื้นฐานของเงื่อนไขออล เป็นการเลือกสมาชิกในเอนทิตีที่เป็นคิวรี่หลักที่มีลักษณะประจำบางอย่างมีค่าตรงกับทั้งหมดของเอนทิตีที่เป็นคิวรี่ย่อย จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างออล (All) ด้วยข้อกำหนดเซต โดยการเลือกเฉพาะสมาชิกในเอนทิตีลำดับที่ 1 ซึ่งมีค่าลักษณะประจำที่กำหนดตรงกับเซตของสมาชิกทั้งหมดของเอนทิตีลำดับที่ 2 ดังนี้

R

$an_1 : AT_1$
...
$an_n : AT_n$

All

$input1? : \mathbb{F}\ Entity1$
$input2? : \mathbb{F}\ Entity2$
$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①
$input2? \neq \varnothing$ ②
$\forall\ out : output!; in1 : input1?$ ③
$\quad | (\forall\ i1 : input1?; i2 : input2? | i1.an_i \square i2.an_i \bullet true)$
$\quad \bullet\ out = in1$

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างออล ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่คิวรี่หลัก (*input1?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี ซึ่งเหมือนเอนทิตีที่เป็นคิวรี่หลัก

All คือ ชื่อเค้าร่างออล

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่หลักของเงื่อนไขออล

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่ย่อยของเงื่อนไขออล

output! คือ ข้อมูลผลลัพธ์ของเค้าร่างออล

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่หลัก ต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่ย่อย ต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้เอนทิตีผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นคิวรี่หลัก (*out = in1*) โดยเลือกเฉพาะข้อมูลที่ลักษณะประจำที่กำหนดมีค่าตรงกับทั้งหมดของสมาชิกของเอนทิตีที่เป็นคิวรี่ย่อย ($\forall\ i1 : input1?; i2 : input2? | i1.an_i \square i2.an_i \bullet true$) โดยเครื่องหมาย $\square$ หมายถึงเครื่องหมายการเปรียบเทียบที่สามารถเป็นไปได้ คือ >, <, =, >=, <= และ $\neq$

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

i1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

i2 คือ ชื่อเอนทิตีที่เป็นคิวรี่ย่อย

กฎข้อที่ 23 การดำเนินการพื้นฐานของเงื่อนไขเอนนี่ เป็นการเลือกสมาชิกในเอนทิตีที่เป็นคิวรี่หลักที่มีลักษณะประจำบางอย่างมีค่าตรงกับบางส่วนของเอนทิตีที่เป็นคิวรี่ย่อย จะถูกสร้างเป็นเค้าร่างผลลัพธ์ (R) และเค้าร่างเอนนี่ (Any) ด้วยข้อกำหนดเซต โดยการเลือกเฉพาะสมาชิกในเอนทิตีลำดับที่ 1 ซึ่งมีค่าลักษณะประจำที่กำหนดตรงกับบางส่วนของเซตของสมาชิกของเอนทิตีลำดับที่ 2 ดังนี้

R

$an_1 : AT_1$
...
$an_n : AT_n$

Any

$input1? : \mathbb{F}\ Entity1$
$input2? : \mathbb{F}\ Entity2$
$output! : \mathbb{F}\ R$

$input1? \neq \varnothing$ ①
$input2? \neq \varnothing$ ②
$\forall\ out : output!; in1 : input1?$ ③
$\quad | (\exists\ i1 : input1?; i2 : input2? | i1.an_i \square i2.an_i \bullet true)$
$\quad \bullet\ out = in1$

โดยที่ *R* คือ เค้าร่างผลลัพธ์ของเค้าร่างเอนนี่ ซึ่งมีสถานะเทียบเท่าเอนทิตี โดยมีลักษณะประจำเหมือนเอนทิตีที่เป็นคิวรี่หลัก (*input1?*)

$an_1 \ldots an_n$ คือ ชื่อลักษณะประจำของเอนทิตี ซึ่งเหมือนเอนทิตีที่เป็นคิวรี่หลัก

Any คือ ชื่อเค้าร่างเอนนี่

input1? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่หลักของเงื่อนไขเอนนี่

input2? คือ ข้อมูลเข้าซึ่งเป็นชื่อเอนทิตีที่เป็นคิวรี่ย่อยของเงื่อนไขเอนนี่

output! คือ ข้อมูลผลลัพธ์ ชื่อเค้าร่างผลลัพธ์ของเค้าร่างเอนนี่

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่หลัก ต้องมีข้อมูลอยู่

เพรดิเคต ② เป็นการตรวจสอบว่าเอนทิตีที่เป็นคิวรี่ย่อย ต้องมีข้อมูลอยู่

เพรดิเคต ③ กำหนดให้เอนทิตีผลลัพธ์มีค่าเท่ากับเอนทิตีที่เป็นคิวรี่หลัก (*out = in1*) โดยเลือกเฉพาะข้อมูลที่ค่าของลักษณะประจำที่กำหนดมีค่าตรงกับบางส่วนของสมาชิกของเอนทิตีที่เป็นคิวรี่ย่อย $(\exists\ i1 : input1?; i2 : input2? | i1.an_n \square i2.an_n \bullet true)$ โดยเครื่องหมาย $\square$ หมายถึงเครื่องหมายการเปรียบเทียบที่สามารถเป็นไปได้ คือ >, <, =, >=, <= และ $\neq$

out คือ ชื่อข้อมูลผลลัพธ์

in1 คือ ชื่อเอนทิตีที่เป็นคิวรี่หลัก

i1 คือ ชีอเอนทิตีที่เป็นคิวรี่หลัก

i2 คือ ชีอเอนทิตีที่เป็นคิวรี่ย่อย

กฎข้อที่ 24 การดำเนินการพื้นฐานของฟังก์ชันนับจำนวนสมาชิก เป็นฟังก์ชั่นนับจำนวนสมาชิกในเอนทิตี จะถูกสร้างเป็นเค้าร่างฟังก์ชันนับจำนวนสมาชิก (Count) ด้วยข้อกำหนดเซต โดยการนับจำนวนสมาชิกในเอนทิตีที่กำหนด ดังนี้

Count

$input1? : \mathbb{F}\ Entity$
$output! : \mathbb{N}$

$input1? \neq \varnothing$ ①
$output! = \#\ input1?$ ②

โดยที่ *Count* คือ ชื่อเค้าร่างฟังก์ชันนับจำนวนสมาชิก

input1? คือ ข้อมูลเข้า ชื่อเอนทิตีที่ต้องการนับจำนวนสมาชิก

output! คือ ข้อมูลผลลัพธ์ของการนับจำนวนสมาชิก

เพรดิเคต ① เป็นการตรวจสอบว่าเอนทิตีที่เป็นข้อมูลเข้าต้องมีข้อมูลอยู่

เพรดิเคต ② กำหนดให้ผลที่ได้จากการนับจำนวนสมาชิก มีค่าเท่ากับจำนวนสมาชิกของข้อมูลเข้า

กฎข้อที่ 25 การดำเนินการพื้นฐานของฟังก์ชั่นหาค่าสูงสุด เป็นฟังก์ชั่นหาค่าสูงสุดของลักษณะประจำในเอนทิตี จะถูกสร้างเป็นเค้าร่างฟังก์ชันหาค่าสูงสุด (Max) ด้วยข้อกำหนดเซต โดยการหาค่าสูงสุดของลักษณะประจำที่กำหนด โดยลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับ ดังนี้

Max

$$\Xi\ Entity$$
$$output!\ : \mathbb{Z}$$

$$\forall\ in1 : \mathbb{F}\ Entity \mid in1 \neq \varnothing \bullet\ output! = max\ \{an_i\}$$

โดยที่ *Max* คือ ชื่อเค้าร่างฟังก์ชันหาค่าสูงสุด

Ξ *Entity* คือ การรวมเค้าร่างเอนทิตีที่ต้องการหาค่าสูงสุดของลักษณะประจำ

output! คือ ข้อมูลผลลัพธ์ของการหาค่าสูงสุดของลักษณะประจำ

เพรดิเคต เป็นการกำหนดให้ผลที่ได้จากการหาค่าสูงสุด มีค่าเท่ากับค่าสูงสุดของลักษณะประจำที่กำหนด โดยลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น โดย an_i คือชื่อลักษณะประจำที่ต้องการหาค่าสูงสุด

กฎข้อที่ 26 การดำเนินการพื้นฐานของฟังก์ชั่นหาค่าต่ำสุด เป็นฟังก์ชั่นหาค่าต่ำสุดของลักษณะประจำบางอย่างในเอนทิตี จะถูกสร้างเป็นเค้าร่างฟังก์ชันหาค่าต่ำสุด (Min) ด้วยข้อกำหนดเซต โดยการหาค่าต่ำสุดของลักษณะประจำที่กำหนด โดยลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับ ดังนี้

ดังนี้

Min

$$\Xi\ Entity$$
$$output!\ : \mathbb{Z}$$

$$\forall\ in1 : \mathbb{F}\ Entity \mid in1 \neq \varnothing \bullet\ output! = min\ \{an_i\}$$

โดยที่ *Min* คือ ชื่อเค้าร่างฟังก์ชันหาค่าต่ำสุด

Ξ *Entity* คือ การรวมเค้าร่างเอนทิตีที่ต้องการหาค่าต่ำสุดของลักษณะประจำ

output! คือ ข้อมูลผลลัพธ์ที่ได้จากการหาค่าต่ำสุดของลักษณะประจำ

เพรดิเคต เป็นการกำหนดให้ผลที่ได้จากการหาค่าต่ำสุด มีค่าเท่ากับค่าต่ำสุดของลักษณะประจำที่กำหนด โดยลักษณะประจำนั้นต้องมีชนิดข้อมูลเป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น โดย an_i คือชื่อลักษณะประจำที่ต้องการหาค่าต่ำสุด

กฎข้อที่ 27 การดำเนินการพื้นฐานของฟังก์ชันหาค่าผลรวม เป็นฟังก์ชันหาค่าผลรวมของลักษณะประจำบางอย่างในเอนทิตี จะถูกสร้างเป็นเค้าร่างการหาค่าผลรวม (Sum) ด้วยข้อกำหนดเซต โดยสร้างฟังก์ชันเป็นข้อเท็จจริง (Axiomatic Definition) ลักษณะประจำที่ต้องการหาค่าผลรวมต้องมีชนิดข้อมูลเป็นจำนวนนับ ดังนี้

$$an_i == \mathbb{Z}$$

$$s : \mathbb{Z} \rightarrow an_i$$

$$Sum : \mathbb{Z} \rightarrow an_i$$

$$\forall n : \mathbb{N} \bullet n > \# s \Rightarrow Sum\ n = 0 \land n \leqslant \# s \Rightarrow Sum\ n = (Sum\ n + 1) + s\ (n)$$

โดยที่ an_i คือ การประกาศตัวแปรลักษณะประจำที่ต้องการหาค่าผลรวมเป็นจำนวนนับ โดยจะต้องเป็นจำนวนนับหรือจำนวนธรรมชาติเท่านั้น

s คือ การกำหนดข้อเท็จจริงซึ่งเป็นเซตของคู่ลำดับของจำนวนนับและค่าของลักษณะประจำที่ต้องการหาค่าผลรวม

Sum คือ การกำหนดข้อเท็จจริงเป็นฟังก์ชันหาค่าผลรวม

เพรดิเคต เป็นการกำหนดให้ผลที่ได้จากการหาค่าผลรวม มีค่าเท่ากับค่าผลรวมทั้งหมดของลักษณะประจำที่กำหนด โดยทำการเรียกตัวเอง (Recursive) เพื่อบวกสะสมค่าจนครบทั้งหมด

กฎข้อที่ 28 การสร้างนิยามเค้าร่างรวมการดำเนินการพื้นฐาน เป็นการสร้างเค้าร่างรวมการดำเนินการพื้นฐาน โดยการนำชื่อเค้าร่างการดำเนินการพื้นฐานทั้งหมดที่สร้างขึ้นจากแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอลมาเรียงต่อกันตามลำดับ รวมทั้งเปลี่ยนชื่อตัวแปรข้อมูลเข้าและข้อมูลผลลัพธ์ทุกตัวไม่ให้ซ้ำกัน ดังนี้

$BDD_i \mathrel{\widehat{=}} Op_1\ [op_1 input1? / input1?, op_n input2? / input2?, op_1 output! / output!];$

...

$Op_n\ [op_n input1? / input1?, op_n input2? / input2?, op_n output! / output!]$

โดยที่ BDD_i คือ ชื่อเค้าร่างรวมการดำเนินการพื้นฐาน

$Op_1 \ldots Op_n$ คือ ชื่อเค้าร่างการดำเนินการพื้นฐานทั้งหมดที่ปรากฏในแผนภาพเชิงลำดับชั้นของภาษาเอสคิวแอล โดยแต่ละชื่อของเค้าร่างคั่นด้วยเครื่องหมาย ";" รวมทั้งเปลี่ยนชื่อตัวแปรที่เป็นข้อมูลเข้า เช่น *input1? input2?* และข้อมูลผลลัพธ์ เช่น *output!* ของแต่ละเค้าร่างเพื่อไม่ให้ซ้ำกัน โดยอยู่ภายในเครื่องหมาย "[" และ "]" ตัวอย่างเช่น การเปลี่ยนชื่อตัวแปร *input1?* ให้เป็น *op1input1?* เขียนได้ดังนี้ *op1input1? / input1?* โดยระบุชื่อตัวแปรที่ต้องการเปลี่ยนใหม่ คั่นด้วยเครื่องหมาย "/" แล้วตามด้วยชื่อตัวแปรเดิม หากมีตัวแปรที่ต้องเปลี่ยนชื่อมากกว่า 1 ตัวแปร ให้คั่นด้วยเครื่องหมาย ","