



ระบบสร้างออนไลน์จากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้

โดย

นางสาววาสนา ธิบแจ่ม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2558

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

ระบบสร้างออนไลน์จากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้

โดย

นางสาววาสนา ธิบแจ่ม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2558

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์



CUSTOMIZED ONTOLOGY GENERATION FROM
RELATIONAL DATABASE

BY

Ms. VASSANA RIBJAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2015

COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

วิทยานิพนธ์

ของ

นางสาววาสนา ธิบแจ่ม

เรื่อง

ระบบสร้างอนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

เมื่อ วันที่ 7 มกราคม พ.ศ. 2559

ประธานกรรมการสอบวิทยานิพนธ์

ปกป้อง ส่องเมือง

(ดร.ปกป้อง ส่องเมือง)

กรรมการและอาจารย์ที่ปรึกษาวิทยานิพนธ์

[Signature]

(ผู้ช่วยศาสตราจารย์ ดร.รัชฎา คงคะจันทร์)

กรรมการสอบวิทยานิพนธ์

วสัน

(ดร.วสัน ล้อมประเสริฐ)

กรรมการสอบวิทยานิพนธ์

[Signature]

(ดร.มารุต บุรณรัช)

คณบดี

[Signature]

(รองศาสตราจารย์ ปกรณ์ เสริมสุข)

หัวข้อวิทยานิพนธ์	ระบบสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้
ชื่อผู้เขียน	นางสาววาสนา ธิบแจ่ม
ชื่อปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา/คณะ/มหาวิทยาลัย	สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผู้ช่วยศาสตราจารย์ ดร.รัชฎา คงคะจันทร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม (ถ้ามี)	-
ปีการศึกษา	2558

บทคัดย่อ

งานวิจัยนี้ นำเสนอวิธีการในการ สร้างออนโทโลยี จากฐานข้อมูลเชิงสัมพันธ์ และพัฒนา ระบบ เพื่อใช้ในการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ โดยมุ่งเน้นไปที่การขยายข้อกำหนด (Constraint) ของฐานข้อมูลเชิงสัมพันธ์ เพื่อให้ได้ออนโทโลยีที่มีความถูกต้องตรงตามโครงสร้างและข้อกำหนด ของข้อมูล ซึ่งมีขั้นตอนในการดำเนินงาน คือ ดึงโครงสร้าง (Schema), ข้อกำหนด (Constraint) และข้อมูล (Data) ต่างๆ ของฐานข้อมูลเชิงสัมพันธ์ โดยใช้ MySQL DBMS แล้วทำการเปรียบเทียบโครงสร้างและข้อกำหนดต่างๆ ระหว่างฐานข้อมูลเชิงสัมพันธ์กับออนโทโลยี โดยใช้กฎและวิธีการที่ได้นำเสนอในงานวิจัยนี้ จากนั้นทำการสร้างออนโทโลยี โดยใช้ OWL API ซึ่งออนโทโลยี ที่ได้จะอยู่ในรูปแบบของภาษา OWL DL นอกจากนี้ทางผู้วิจัยได้ทำการทดสอบประสิทธิภาพในการทำงานของระบบ โดยการใช้แบบสอบถามวัดความพึงพอใจของผู้ใช้งานระบบ พบว่าระดับความพึงพอใจของผู้ใช้งานระบบอยู่ที่ 85.60%

คำสำคัญ : ฐานข้อมูลเชิงสัมพันธ์, ออนโทโลยี, โอดับบลิวแอล

Thesis Title	CUSTOMIZED ONTOLOGY GENERATION FROM RELATIONAL DATABASE
Author	Ms. Vassana Ribjam
Degree	Master of Science (Computer Science)
Major Field/Faculty/University	Department of Computer Science Faculty of Science and Technology Thammasat University
Thesis Advisor	Assistant Professor Dr. Rachada Kongkachandra
Thesis Co-Advisor (If any)	-
Academic Years	2015

ABSTRACT

In this paper we propose a method and implementation for transformation of relational database to ontology and recommend constraint for ontology. The process focused on the constraint extension of relational database, which are extract structure, constraint and data of relational database using the MySQL DBMS. Then the database schema converts to ontology using the OWL API, which is in the format of OWL DL. Finally, the conversion system was tested from a relational database to ontology and recommend constraint to the users, and 85.60% of users shown their satisfaction to the research.

Keywords : Relational Database, Ontology, OWL

กิตติกรรมประกาศ

วิทยานิพนธ์เหล่านี้สำเร็จลุล่วงได้ด้วยความสำเร็จจากความเมตตาจากผู้ช่วยศาสตราจารย์ ดร.รัชฎา คงคะจันทร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ตลอดจนคณาจารย์ และคณะกรรมการสอบ วิทยานิพนธ์ทุกท่าน ที่สละเวลาอันมีค่าในการให้คำแนะนำ ความรู้ แง่คิด และทักษะต่างๆ ที่เกี่ยวกับการทำงานวิจัย ตลอดจนการตรวจสอบความถูกต้องในการจัดทำวิทยานิพนธ์ฉบับนี้ งานวิจัยและ วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี รวมถึงขอขอบคุณเจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ ทุกท่านที่ให้ความอนุเคราะห์และอำนวยความสะดวกในการทำวิทยานิพนธ์มาโดยตลอด

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ตลอดจนญาติพี่น้องทุกคนในครอบครัว และเพื่อนๆ ที่ช่วยเหลือและให้กำลังใจอย่างยิ่งจนวิทยานิพนธ์ฉบับนี้สำเร็จสมบูรณ์ขึ้นได้ ขออัญเชิญ คุณพระศรีรัตนตรัยและสิ่งศักดิ์สิทธิ์ทั้งหลาย คู่คุ้มครองปกป้องรักษาให้ทุกท่าน มีความสุข ความเจริญ มีโภคทรัพย์และมีสุขภาพพลานามัยสมบูรณ์แข็งแรงตลอดไปด้วยเทอญ

นางสาววาสนา ธิบแจ่ม

สารบัญ

หน้า	
บทคัดย่อภาษาไทย (1)
บทคัดย่อภาษาอังกฤษ (2)
กิตติกรรมประกาศ (3)
สารบัญตาราง (10)
สารบัญภาพ (12)
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์งานวิจัย	2
1.3 ขอบเขตงานวิจัย	2
1.4 คำจำกัดความ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 รายละเอียดวิทยานิพนธ์	3
บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง	4
2.1.1 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)	4
2.1.1.1 คุณลักษณะของ Relation	5
2.1.1.2 ประเภทของ Relation	5
(1) Base Relation	5
(2) View	6

	(5)
2.1.1.3 ประเภทของคีย์ใน Relation	6
(1) Primary Key (PK) หรือคีย์หลัก	6
(2) Secondary Key หรือคีย์รอง	7
(3) Candidates Key หรือคีย์คู่แข่ง	7
(4) Alternate Key หรือคีย์สำรอง	7
(5) Foreign Key (FK) หรือคีย์นอก	7
(6) Sample Key	7
(7) Composite Key หรือคีย์ผสม	7
2.1.1.4 ค่าว่าง (Null Value)	7
2.1.1.5 ค่าเอกลักษณ์ (Unique Value)	7
2.1.1.6 การนอร์มัลไลซ์ (Normalization)	8
(1) First Normal Form (1NF)	8
(2) Second Normal Form (2NF)	9
(3) Third Normal Form (3NF)	11
2.1.1.7 Binary Relation	12
2.1.1.8 Inheritance Relation	13
2.1.2 ออนโทโลยี (Ontology)	14
2.1.2.1 องค์ประกอบของออนโทโลยี	14
(1) แนวคิด (Concepts)	14
(2) คุณสมบัติ (Property)	14
(3) ความสัมพันธ์ (Relationships)	14
(4) ข้อกำหนดการสร้างความสัมพันธ์ (Axioms)	15
(5) ตัวอย่างข้อมูล (Instance)	15
2.1.2.2 ภาษาที่ใช้อธิบายออนโทโลยี	15
(1) ลอจิกเบส (First order logic based)	15
(2) เฟรมเบส (Frame logic based)	16
(3) เว็บเบส (Web based)	16

2.1.2.3 การประยุกต์ใช้ออนโทโลยี	16
2.1.3 Web Ontology Language (OWL)	17
2.1.3.1 ประเภทของภาษา OWL	17
(1) OWL LITE	17
(2) OWL DL	17
(3) OWL FULL	17
2.1.3.2 โครงสร้างของภาษา OWL	18
(1) Namespace	18
(2) Ontology Header	18
2.1.3.3 อิลิเมนต์พื้นฐาน (Basic Element)	19
(1) คลาสและคลาสย่อย (Class and SubClass)	19
(2) คุณสมบัติของคลาส (Property)	21
(3) คุณสมบัติและชนิดข้อมูล	22
(4) ข้อกำหนดของคุณสมบัติ	23
2.2 งานวิจัยที่เกี่ยวข้อง	24
2.2.1 Rules and Implementation for Generating Ontology from Relational Database	24
2.2.2 Generating of RDF graph from a relational database using Jena API	26
2.2.3 Algorithms for Mapping RDB Schema to RDF for Facilitating Access to Deep Web	29
2.2.4 Mapping Relational Database into OWL Ontology	30
2.2.5 Automatic Mapping of Relational Database to OWL Antology	31
บทที่ 3 วิธีการดำเนินงานวิจัย	34
3.1 ขอบเขตการทดลอง	34
3.2 เครื่องมือที่ใช้ในการพัฒนา	34
3.3 สถาปัตยกรรมของระบบ	35
3.4 กระบวนการในการทำงานของระบบ	36

3.4.1 การแตกโครงสร้างของฐานข้อมูลเชิงสัมพันธ์	37
3.4.2 การเปรียบเทียบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์กับ โครงสร้างของออนโทโลยี (Mapping Rule)	37
3.4.2.1 ตรวจสอบคุณสมบัติของ Relation	38
(1) Binary Relation	38
(2) Inheritance Relation	39
(3) Normal Relation	39
3.4.2.2 ตรวจสอบคุณสมบัติ Normal Attribute	40
(1) Null	40
(2) Not Null	41
(3) Unique	42
(4) Not Null และ Unique	43
3.4.2.3 ตรวจสอบคุณสมบัติของ Attribute ที่เป็นคีย์ต่างๆ	45
(1) Primary Key เพียงหนึ่งเดียวใน Relation	45
(2) Primary Key และ Foreign Key เพียงหนึ่งเดียวใน Relation	46
(3) Foreign Key ใน Relation	48
(4) Foreign Key แบบ Inverse Reference ใน Relation	51
(5) Foreign Key และ Reference มายัง Relation ตนเอง	53
(6) Primary Key และไม่ใช่ Foreign Key (Composite Key)	55
(7) Primary Key และเป็น Foreign Key (Composite Key)	57
3.4.2.4 ตรวจสอบค่าเริ่มต้นของ Attribute ใน Relation	60
3.4.2.5 ข้อมูลในฐานข้อมูลเชิงสัมพันธ์	61
3.4.2.6 สรุปวิธีการในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี	62
3.5 การวัดผลการทดลอง	
3.5.1 วัดประสิทธิภาพการทำงานของระบบ	69
3.5.2 วัดความพึงพอใจในการใช้งานระบบ	70
3.5.2.1 เครื่องมือที่ใช้ในการรวบรวมข้อมูล	70
3.5.2.2 การเก็บรวบรวมข้อมูล	71
3.5.2.3 การวิเคราะห์ข้อมูล	71
3.5.2.4 สถิติที่ใช้ในงานวิจัย	71

บทที่ 4 ผลการวิจัยและอภิปรายผล	72
4.1 วิธีการทดลอง	72
4.1.1 การเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์	72
4.1.2 การตรวจสอบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์	73
4.1.3 การปรับแต่งโครงสร้างของออนโทโลยี	74
4.1.3.1 การปรับแต่ง Class และคุณสมบัติของ Class	74
4.1.3.2 การปรับแต่ง DatatypeProperty และคุณสมบัติของ DatatypeProperty	75
4.1.3.3 การปรับแต่ง ObjectProperty และคุณสมบัติของ ObjectProperty	76
4.1.4 การสร้างออนโทโลยี	77
4.2 ผลการทดลองและประสิทธิภาพการทำงานของระบบ	78
4.3 ผลการวิเคราะห์ความพึงพอใจของผู้ใช้งานระบบ	80
4.3.1 ข้อมูลส่วนตัวของผู้เข้าร่วมการทดลอง	80
4.3.2 ผลการวิเคราะห์ด้านความพึงพอใจ	84
4.4 อภิปรายผลการทดลอง	87
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	88
5.1 สรุปผลการวิจัย	88
5.2 ข้อเสนอแนะ	89
5.3 แนวทางการวิจัยในอนาคต	89
รายการอ้างอิง	90
ภาคผนวก	
ภาคผนวก ก แบบทดสอบถามข้อมูลผู้เข้าร่วมการทดลอง	93
ภาคผนวก ข แบบทดสอบถามประเมินความพึงพอใจของผู้เข้าร่วมการทดลอง	95

ภาคผนวก ค พจนานุกรมข้อมูลของตัวอย่างฐานข้อมูลเชิงสัมพันธ์

ประวัติผู้เขียน



สารบัญตาราง

ตารางที่ หน้า

2.1 แสดง Relation ที่เป็น Non-1NF	8
2.2 แสดง Relation ที่เป็น 1NF	9
2.3 แสดง Relation ที่เป็น Non-2NF	10
2.4 แสดง Relation นักศึกษา	10
2.5 แสดง Relation วิชา	10
2.6 แสดง Relation การลงทะเบียน	10
2.7 แสดง Relation ที่เป็น Non-3NF	11
2.8 แสดง Relation ข้อมูลลูกค้า	11
2.9 ระดับความสำคัญของลูกค้า	11
2.10 ข้อมูลสมาชิกภายใน Class Course	21
2.11 ประเภทของคุณสมบัติ	21
2.12 โครงสร้างของคุณสมบัติในรูปแบบต่างๆ	22
2.13 ชนิดข้อมูลที่อ้างอิงตาม XML Schema datatype	22
2.14 ชนิดข้อมูลที่อ้างอิงตาม Constraints	23
2.15 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ R. Yutao, J. Lihong, B. Fenglin, และ C. Hongming	25
2.16 กฎในการแปลงเป็นออนโทโลยีของ B. Jamal และ B. Mohamed	28
2.17 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ Y. M. Wondu, Farhi, M., และ Vassil T. V.	29
2.18 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ L. Haiyun และ Z. Shufeng	30
2.19 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ A. Larbi, E. H. Oussama และ B. Mohamed	32
3.1 เครื่องมือที่ใช้สำหรับงานวิจัย	34
3.2 สรุปรูปวิธีการที่ใช้ในการแปลงจากงานวิจัยก่อนหน้า	62
3.3 สรุปรูปวิธีการในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม	63

3.4 ตารางแสดงผลการทดสอบจำแนกตามกลุ่มที่มีผลการทดสอบเป็นบวก และผลการทดสอบเป็นลบ	69
4.1 ผลการแปลงฐานข้อมูลการซื้อขายสินค้าเป็นออนไลน์	79
4.2 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามเพศ	80
4.3 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามอายุ	80
4.4 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามระดับศึกษา	81
4.5 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามอาชีพ	81
4.6 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามตำแหน่ง	82
4.7 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามประสบการณ์ในการทำงาน	82
4.8 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกระดับความรู้ ทางด้านฐานข้อมูลเชิงสัมพันธ์	83
4.9 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกความรู้ทางด้านออนไลน์	83
4.10 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกระดับความรู้ทางด้านออนไลน์	83
4.11 ค่าสถิติการวิเคราะห์ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของคะแนนความพึงพอใจ	84
ค.1 โครงสร้างข้อมูลตาราง Advance	97
ค.2 โครงสร้างข้อมูลตาราง categories	97
ค.3 โครงสร้างข้อมูลตาราง clear_advance	97
ค.4 โครงสร้างข้อมูลตาราง customercustomerdemo	97
ค.5 โครงสร้างข้อมูลตาราง customerdemographics	98
ค.6 โครงสร้างข้อมูลตาราง customers	98
ค.7 โครงสร้างข้อมูลตาราง employeeterritories	98
ค.8 โครงสร้างข้อมูลตาราง territories	98
ค.9 โครงสร้างข้อมูลตาราง employees	99
ค.10 โครงสร้างข้อมูลตาราง region	99
ค.11 โครงสร้างข้อมูลตาราง shippers	99
ค.12 โครงสร้างข้อมูลตาราง suppliers	100
ค.13 โครงสร้างข้อมูลตาราง products	100
ค.14 โครงสร้างข้อมูลตาราง products_location	101
ค.15 โครงสร้างข้อมูลตาราง orders	101
ค.16 โครงสร้างข้อมูลตาราง orderdetails	101
ค.17 โครงสร้างข้อมูลตาราง shipping	102

สารบัญภาพ

ภาพที่ หน้า

2.1	โครงสร้างฐานข้อมูลเชิงสัมพันธ์	4
2.2	ลักษณะของคีย์ใน Relation	6
2.3	ตัวอย่าง Binary Relation	12
2.4	ตัวอย่าง Inheritance Relation	13
2.5	การกำหนด Namespace	18
2.6	อธิบายรายละเอียดเบื้องต้นของออนโทโลยี	19
2.7	ตัวอย่างตารางข้อมูลของระบบลงทะเบียนเรียน	19
2.8	การประกาศชื่อคลาส	20
2.9	การประกาศชื่อคลาสย่อย	20
2.10	การกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:allValuesFrom	23
2.11	การกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:someValuesFrom	24
2.12	กระบวนการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี	25
2.13	อธิบายวิธีการในการแปลง	27
3.1	ภาพรวมการทำงานของระบบ	35
3.2	ตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์	36
3.3	โครงสร้างของ Relation order	37
3.4	ตัวอย่าง Relation customercustomer_demo	38
3.5	ตัวอย่างการแปลง Binary Relation เป็นออนโทโลยี	38
3.6	ตัวอย่าง Relation products_location	39
3.7	ตัวอย่างการแปลง Inheritance Relation เป็นออนโทโลยี	39
3.8	ตัวอย่างการแปลง Normal Relation เป็นออนโทโลยี	40
3.9	ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Null เป็นออนโทโลยี	40
3.10	ตัวอย่าง Relation region	41
3.11	ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Not Null เป็นออนโทโลยี	41
3.12	ตัวอย่าง Relation customerdemographics	42
3.13	ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Unique เป็นออนโทโลยี	43
3.14	ตัวอย่าง Relation categories	44

3.15 ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Not Null และ Unique เป็นอนโทโลยี	45
3.16 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Primary Key เพียงหนึ่งเดียวใน Relation เป็นอนโทโลยี	46
3.17 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Primary Key และ Foreign Key เพียงหนึ่งเดียวใน Relation เป็นอนโทโลยี	48
3.18 ตัวอย่าง Relation products	49
3.19 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key และไม่ใช่ Primary Key ใน Relation เป็นอนโทโลยี	50
3.20 ตัวอย่าง Relation advance	51
3.21 ตัวอย่าง Relation clear_advance	51
3.22 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key แบบ Inverse Reference ใน Relation เป็นอนโทโลยี	52
3.23 ตัวอย่าง Relation employees	54
3.24 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key และ Reference มายัง Relation ตนเองใน Relation เป็นอนโทโลยี	55
3.25 ตัวอย่าง Relation shipping	56
3.26 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และไม่เป็น Foreign Key ใน Relation เป็นอนโทโลยี	57
3.27 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และเป็น Foreign Key ใน Relation เป็นอนโทโลยี	58
3.28 ตัวอย่าง Relation orderdetails	60
3.29 ตัวอย่างการแปลง Attribute ที่ค่า Default Value ใน Relation เป็นอนโทโลยี	61
3.30 ตัวอย่างข้อมูลใน Relation region	61
3.31 ตัวอย่างการแปลงข้อมูลเป็นอนโทโลยี	62
4.1 การเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์	73
4.2 ระบบแสดงโครงสร้างตาราง region ในฐานข้อมูลเชิงสัมพันธ์	73
4.3 ระบบแสดงข้อมูลที่จัดเก็บในตาราง region	73
4.4 ตัวอย่างก่อนการปรับเปลี่ยน Class advance	74
4.5 ตัวอย่างหลังการปรับเปลี่ยน Class advance	74
4.6 ตัวอย่างก่อนการปรับเปลี่ยน DatatypeProperty #advance.adv_date	75

4.7 ตัวอย่างหลังการปรับเปลี่ยน DatatypeProperty #advance.adv_date	75
4.8 ตัวอย่างก่อนการปรับเปลี่ยน ObjectProperty #shipping.ShipVia_Ref	76
4.9 ตัวอย่างหลังการปรับเปลี่ยน ObjectProperty #shipping.ShipVia_Ref	77
4.10 การกดปุ่ม Next เพื่อเข้าสู่ขั้นตอนต่อไป	77
4.11 ตัวอย่างออนโทโลยีที่อยู่ในรูปแบบของ OWL	78



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันโลกได้ก้าวสู่ยุคเทคโนโลยีสารสนเทศหรือยุคสังคมฐานความรู้ (Knowledge based society) ที่ความรู้เข้ามามีบทบาทมากในทุกๆ เรื่อง การจัดการความรู้ได้อย่างมีประสิทธิภาพ จะเป็นการเพิ่มประสิทธิภาพในการแข่งขันและเป็นการกระตุ้นการคิดค้นนวัตกรรมใหม่ๆ

ออนโทโลยี เป็นแนวคิดที่ถูกนำมาประยุกต์ใช้เพื่อการจัดการความรู้ ซึ่งสามารถจัดการ และนำเสนอความรู้ในรูปแบบของกลุ่มแนวคิดและความสัมพันธ์ระหว่างกันที่อยู่ภายใต้ขอบเขต ที่สนใจ แต่การสร้างออนโทโลยีมีปัญหาที่สำคัญคือการพึ่งพาผู้เชี่ยวชาญในการสกัดองค์ความรู้ และ จัดหมวดหมู่ประเภทของความรู้ให้เหมาะสมต่อการนำเอาความรู้ไปใช้ประโยชน์ ซึ่งในบางครั้งเป็นเรื่องยากที่จะหาผู้เชี่ยวชาญในเรื่องใดเรื่องหนึ่ง รวมถึงการใช้นุ้ชยในการสร้างออนโทโลยีก็ต้องใช้ ระยะเวลา ค่าใช้จ่าย และทรัพยากรค่อนข้างสูง นอกจากนี้ยังทำให้เกิดข้อผิดพลาดได้ง่าย ดังนั้น หากสามารถนำเทคโนโลยีคอมพิวเตอร์เข้ามาช่วยจัดการองค์ความรู้แบบอัตโนมัติได้ก็อาจจะสามารถ ลดปัญหาที่เกิดขึ้นดังกล่าวได้

โดยรูปแบบของข้อมูลที่นำมาใช้ในการสร้างออนโทโลยี สามารถแบ่งออกได้เป็น 2 รูปแบบ คือ รูปแบบของข้อมูลที่มีโครงสร้าง เช่น ข้อมูลที่จัดเก็บในฐานข้อมูลเชิงสัมพันธ์ (Relational Database) และรูปแบบของข้อมูลที่ไม่มีโครงสร้าง เช่น ข้อมูลที่จัดเก็บในเอกสารโดยทั่วไป (Full Text) เป็นต้น ซึ่งทางผู้วิจัยได้เลือกนำรูปแบบของข้อมูลที่มีโครงสร้างมาใช้ในการสร้างออนโทโลยี โดยข้อมูลจะอยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ เนื่องจากข้อมูลมีโครงสร้างอยู่ในรูปแบบของ ตารางและมีการเชื่อมโยงกันของข้อมูลอยู่แล้ว และการสร้างออนโทโลยีจากรูปแบบของข้อมูลที่ไม่มี โครงสร้างทำได้ยาก เพราะภาษามีความหลากหลาย มีความคลุมเครือ และมีกระจัดกระจาย ไม่เป็นระเบียบ

ในปัจจุบันการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ ยังมีจำนวนน้อย เนื่องจากโครงสร้างฐานข้อมูลมีความหลากหลายและมีความซับซ้อน ซึ่งส่วนใหญ่จะเน้นไปที่ โครงสร้างพื้นฐานของฐานข้อมูล ไม่ได้เน้นไปที่ข้อกำหนด (Constraints) ต่างๆ ซึ่งเป็นส่วนที่มี ความสำคัญ เพราะข้อกำหนดต่างๆ ในโครงสร้างฐานข้อมูลจะช่วยให้ข้อมูลมีความถูกต้องตาม ขอบเขตที่กำหนด และส่งผลให้มีประสิทธิภาพมากยิ่งขึ้น

ดังนั้นทางผู้วิจัยจึงได้ศึกษาค้นคว้าถึงวิธีการสร้างออนโทโลยี โดยการนำเทคนิควิทยาการคอมพิวเตอร์มาบูรณาการให้เครื่องคอมพิวเตอร์สามารถสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติได้ โดยเน้นไปที่การขยายข้อกำหนดของโครงสร้างฐานข้อมูลเชิงสัมพันธ์ เพื่อเพิ่มประสิทธิภาพของออนโทโลยีให้มีความถูกต้องมากยิ่งขึ้นตามโครงสร้างฐานข้อมูล รวมถึงช่วยลดระยะเวลา ค่าใช้จ่ายและความผิดพลาดที่อาจเกิดขึ้นจากมนุษย์ได้

1.2 วัตถุประสงค์งานวิจัย

1. เพื่อศึกษาและเปรียบเทียบหลักการในการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ
2. เพื่อนำเสนอวิธีการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ โดยเน้นไปที่การขยายข้อกำหนดต่างๆ ของโครงสร้างฐานข้อมูล
3. เพื่อเสนอแนะข้อกำหนดต่างๆ ที่ควรจะมีในโครงสร้างของออนโทโลยี โดยอยู่บนพื้นฐานของข้อกำหนดในโครงสร้างของฐานข้อมูลเชิงสัมพันธ์

1.3 ขอบเขตงานวิจัย

1. ฐานข้อมูลเชิงสัมพันธ์ต้องอยู่ในรูปนอร์มัลฟอร์มที่ 3 (Third Normal Form : 3NF)
2. ออนโทโลยีจะอยู่ในรูปแบบ OWL Ontology

1.4 คำจำกัดความ

1.4.1 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ระบบฐานข้อมูลเชิงสัมพันธ์ เป็นฐานข้อมูลที่ใช้ในโมเดลเชิงสัมพันธ์ โดยใช้หลักพื้นฐานทางคณิตศาสตร์ เป็นรูปแบบของฐานข้อมูลที่นิยมใช้กันมากในปัจจุบัน เนื่องจากเป็นการเก็บข้อมูลในรูปของตาราง (Table) ในแต่ละตารางแบ่งออกเป็นแถวๆ และในแต่ละแถวแบ่งออกเป็นคอลัมน์ (Column) ทำให้ง่ายต่อการเข้าใจและประยุกต์ใช้งาน รวมทั้งมีเครื่องมือสนับสนุนในการจัดการข้อมูลมากมาย เช่น MySQL, Oracle, MS SQL, Access เป็นต้น

1.4.2 ออนโทโลยี (Ontology)

ออนโทโลยีคือ ข้อกำหนดเกี่ยวกับแนวความคิด (Concepts) โดยแนวความคิดของออนโทโลยี คือ การบรรยายแนวความคิดของโดเมนหรือขอบเขตความสนใจใดๆ ในรูปของสิ่งต่างๆ ที่อยู่ภายในโดเมนและความสัมพันธ์ระหว่างสิ่งเหล่านั้น หรือการอธิบายความสัมพันธ์โครงสร้างความรู้ให้อยู่ในรูปแบบลำดับชั้นเชิงวัตถุ (Hierarchical Data Structure) เพื่ออธิบายขอบเขตขององค์ความรู้ที่สนใจ

1.4.3 OWL (Web Ontology Language)

เป็นภาษาที่ใช้สำหรับอธิบายข้อมูลในเชิงออนโทโลยี และกำหนดความสัมพันธ์ระหว่างข้อมูลตามขอบเขตที่สนใจ ซึ่งพัฒนาต่อมาจากภาษา RDF (Resource Description Framework) และสืบทอดมาจากภาษา DAML (DARPA Agent Markup Language) + OIL (Ontology Interchange Language) โดยภาษา OWL ได้นำเอาคลาสและคุณสมบัติของคลาสจาก RDF มาใช้ รวมทั้งเพิ่มส่วนของการกำหนดชนิดข้อมูล การบรรยายข้อมูลเชิงตรรกะและการกำหนดขนาดข้อมูล ทำให้ข้อมูลที่ถูกแทนที่มีความหมายมากยิ่งขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบที่สามารถสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติได้
2. ได้ระบบที่สามารถปรับปรุงข้อกำหนดของออนโทโลยีได้ตามความต้องการ โดยจะอยู่บนพื้นฐานของฐานข้อมูลเชิงสัมพันธ์
3. ทำให้สามารถสร้างออนโทโลยีจากฐานข้อมูลเดิมได้ถูกต้องแม่นยำมากยิ่งขึ้น รวมถึงเป็นการลดค่าใช้จ่าย ระยะเวลา และทรัพยากรต่างๆ ที่ใช้ในการสร้างออนโทโลยี

1.6 รายละเอียดวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย บทนำในบทที่ 1 จะกล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย คำจำกัดความ และผลประโยชน์ที่คาดว่าจะได้รับ บทที่ 2 กล่าวถึงทฤษฎีต่างๆ ที่เกี่ยวข้องกับงานวิจัย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง บทที่ 3 เป็นการอธิบายวิธีการดำเนินงานวิจัย แนวคิดพื้นฐานและขั้นตอนการดำเนินงาน บทที่ 4 เป็นส่วนของผลการทดลองและการประเมินผลการทดลอง และบทที่ 5 บทสุดท้าย เป็นส่วนของการสรุปผลการทดลองของงานวิจัยรวมทั้งรายละเอียดและข้อเสนอแนะ

บทที่ 2

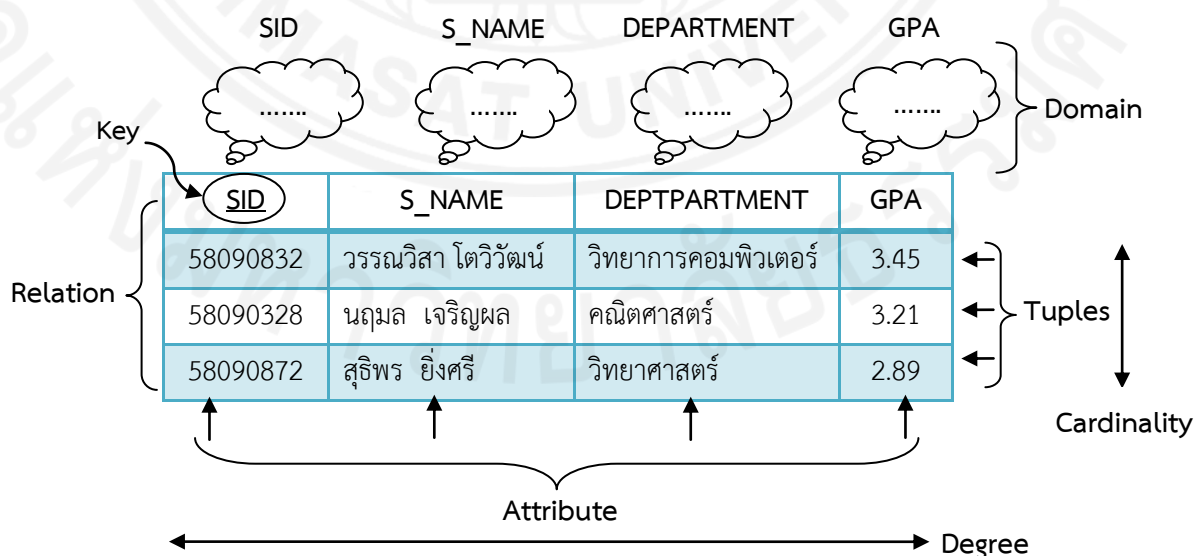
วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

วิทยานิพนธ์นี้นำเสนอวิธีการในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีแบบอัตโนมัติ โดยมุ่งเน้นไปที่การขยายข้อกำหนดของโครงสร้างฐานข้อมูล เพื่อให้สามารถแปลงเป็นออนโทโลยีได้อย่างถูกต้อง โดยมีทฤษฎีที่เกี่ยวข้องในงานวิจัยดังต่อไปนี้

2.1.1 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นฐานข้อมูลที่ใช้โมเดลเชิงสัมพันธ์ (Relational Database Model) ถูกคิดค้นและพัฒนาโดย อี เอฟ คอดด์ (E.F. Codd) โดยใช้หลักพื้นฐานทางคณิตศาสตร์ เป็นรูปแบบของฐานข้อมูลที่นิยมใช้กันมากในปัจจุบัน เนื่องจากง่ายต่อการเข้าใจและประยุกต์ใช้งาน รวมทั้งมีเครื่องมือสนับสนุนในการจัดการข้อมูลมากมาย เช่น MySQL, Oracle, MS SQL, Access เป็นต้น ซึ่งการเก็บข้อมูลในฐานข้อมูลเชิงสัมพันธ์จะอยู่ในรูปแบบของตาราง (Table) หลายๆ ตารางที่มีความสัมพันธ์กัน ในแต่ละตารางจะแบ่งออกเป็นแถวๆ และในแต่ละแถวจะแบ่งออกเป็นคอลัมน์ (Column) (การเรียนการสอนผ่านเว็บ, 2550)



ภาพที่ 2.1 โครงสร้างฐานข้อมูลเชิงสัมพันธ์

จากภาพ 2.1 โครงสร้างฐานข้อมูลเชิงสัมพันธ์ จะประกอบไปด้วย

1. Relation หมายถึง ตาราง (Table) สองมิติที่ประกอบด้วยแถว (Row) และ คอลัมน์ (Column)
2. Attribute หมายถึง คอลัมน์ (Column) หรือฟิลด์ (Field) ใน Relation
3. Tuples หมายถึง แถว (Row) หรือระเบียบข้อมูล (Record) ใน Relation
4. Degree หมายถึง จำนวน Attribute ใน Relation
5. Cardinality หมายถึง จำนวนแถวของข้อมูลหรือ Record ใน Relation
6. Domain หมายถึง การกำหนดขอบเขตของค่าของข้อมูลที่จะเป็นในแต่ละ Attribute เพื่อป้องกันไม่ให้ข้อมูลที่ผู้ใช้จัดเก็บ มีความผิดพลาดไปจากความเป็นจริงจากที่ควรจะเป็น
7. Key หมายถึง ชุดของ Attribute ที่มีจำนวนน้อยที่สุดที่ทำให้ค่าไม่ซ้ำกัน (ค่าของ Attribute อาจประกอบด้วยหนึ่ง Attribute หรือกลุ่มของ Attribute ที่ใช้สำหรับเป็นตัวบ่งบอกถึงความแตกต่างของ Tuples ใน Relation)

2.1.1.1 คุณลักษณะของ Relation

1. ช่อง (Cell) แต่ละช่องของตารางจะเก็บข้อมูลได้เพียงค่าเดียว (Single Value)
2. ชนิดข้อมูล ข้อมูลที่อยู่ในคอลัมน์เดียวกันจะต้องมีชนิดข้อมูล (Data Type) เป็นแบบเดียวกัน เช่น Attribute เงินเดือน ของทุกแถวจะต้องมีข้อมูลที่เป็นตัวเลขเท่านั้น
3. ชื่อคอลัมน์แต่ละคอลัมน์ของ Relation หนึ่งๆ จะต้องมีชื่อคอลัมน์ที่ไม่ซ้ำกัน ส่วนลำดับของคอลัมน์ก่อนและหลังไม่ถือว่าสำคัญ
4. แถวข้อมูลแต่ละแถวของทุกคอลัมน์ใน Relation หนึ่งๆ ต้องไม่ซ้ำกัน (ทุกคอลัมน์) ส่วนการเรียงลำดับของแถวไม่ถือว่าสำคัญ

2.1.1.2 ประเภทของ Relation

ในระบบจัดการฐานข้อมูลต่างๆ ไป Relation อาจจำแนกออกได้เป็น 2 ประเภท ดังนี้คือ

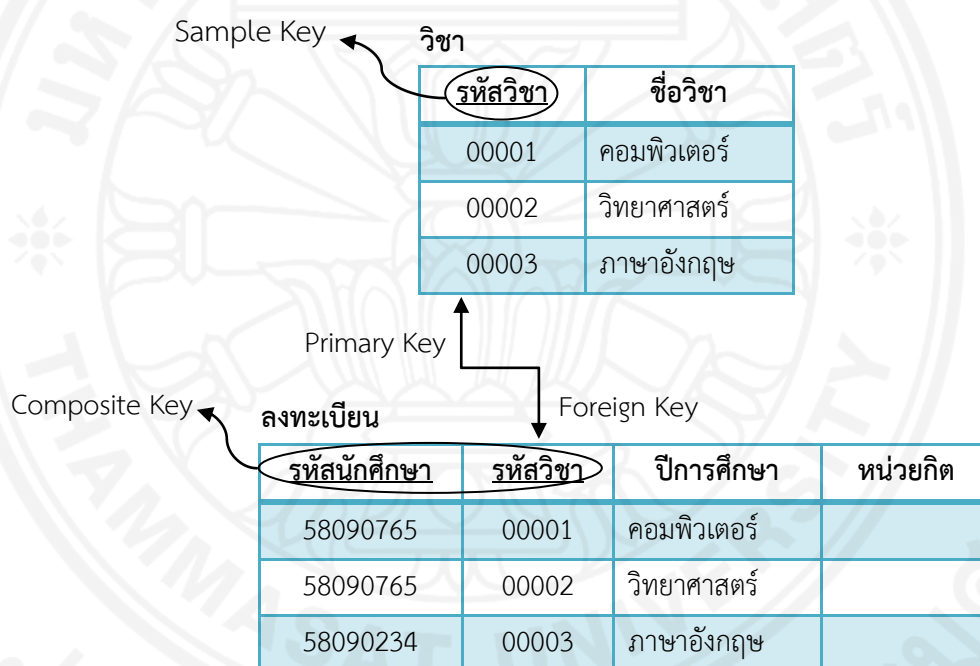
(1) Base Relation

เป็น Relation ที่ถูกกำหนดขึ้นเพื่อใช้เก็บข้อมูล และจะสามารถนำข้อมูลไปใช้ได้ ก็ต่อเมื่อมีการสร้าง Relation ในภาษาสำหรับนิยามข้อมูล (Data Definition Language หรือ DDL) ซึ่งจะเป็นตารางที่มีการจัดเก็บข้อมูลจริงไว้ในฐานข้อมูล

(2) View

เป็น Relation ที่ถูกสร้างขึ้นตามความต้องการใช้ข้อมูลของผู้ใช้ แต่ละคนเนื่องจากผู้ใช้แต่ละคนอาจมีความต้องการใช้ข้อมูลในลักษณะที่แตกต่างกัน เพื่อความสะดวกในการใช้ข้อมูลและช่วยให้การรักษาความปลอดภัยของฐานข้อมูลทำได้ง่ายขึ้น โดย View จะถูกกำหนดขึ้นมาจาก Base Relation ซึ่ง View ที่ถูกสมมติขึ้นมาจะไม่มีเก็บข้อมูลจริงๆ ในระบบฐานข้อมูล แต่เป็นข้อมูลที่แปลค่ามา (Virtual Table หรือ Derived Table) ตามความต้องการของผู้ใช้แต่ละคน

2.1.1.3 ประเภทของคีย์ใน Relation



ภาพที่ 2.2 ลักษณะของคีย์ใน Relation

จากภาพที่ 2.2 จะแสดงลักษณะของคีย์ใน Relation ซึ่งสามารถแบ่งออกได้ดังนี้

(1) Primary Key (PK) หรือคีย์หลัก

Primary Key หรือคีย์หลัก คือ Attribute ที่มีคุณสมบัติของข้อมูลที่มีค่าเป็นเอกลักษณ์ หรือไม่มีค่าซ้ำกัน มีผลทำให้สามารถเข้าถึงข้อมูลในฐานข้อมูลได้อย่างรวดเร็ว

(2) Secondary Key หรือคีย์รอง

Secondary Key หรือคีย์รอง คือ Attribute ที่ไม่ใช่คีย์หลัก แต่สามารถใช้ในการสืบค้นข้อมูลได้ โดยคีย์รองไม่จำเป็นต้องมีคุณสมบัติที่มีค่าเป็นเอกลักษณ์

(3) Candidates Key หรือคีย์คู่แข่ง

Candidates Key หรือคีย์คู่แข่ง คือในแต่ละ Relation อาจมี Attribute ที่ทำหน้าที่เป็นคีย์หลักได้มากกว่าหนึ่ง Attribute เช่น นักศึกษาแต่ละคน มี Attribute คือ รหัสนักศึกษาและรหัสประจำตัวประชาชน ซึ่งทั้งรหัสนักศึกษาและรหัสประจำตัวประชาชน สามารถเป็นคีย์หลักได้ทั้งคู่ เป็นต้น โดยการเลือกคีย์คู่แข่ง ส่วนใหญ่จะเลือกจาก Attribute ที่ชื่อสั้นที่สุดเป็นคีย์หลัก

(4) Alternate Key หรือคีย์สำรอง

Alternate Key หรือคีย์สำรอง คือ Attribute ที่เป็นคีย์คู่แข่งที่ไม่ได้ถูกเลือกให้เป็นคีย์หลัก

(5) Foreign Key (FK) หรือคีย์นอก

Foreign Key (FK) หรือคีย์นอก คือคีย์ซึ่งประกอบด้วย Attribute หรือกลุ่มของ Attribute ใน Relation หนึ่ง ซึ่งมีคุณสมบัติเป็นคีย์หลัก และไปปรากฏในอีก Relation หนึ่ง เพื่อประโยชน์ในการเชื่อมโยงข้อมูลซึ่งกันและกัน

(6) Sample Key

Sample Key คือคีย์ที่ประกอบด้วย 1 Attribute เท่านั้น

(7) Composite Key หรือคีย์ผสม

Composite Key หรือคีย์ผสม คือการนำ Attribute อย่างน้อย 2 Attribute มารวมกันเป็นคีย์หลัก เนื่องจากหากใช้ฟิลด์ใดฟิลด์หนึ่งมาเป็นคีย์หลัก จะส่งผลให้ข้อมูลในแต่ละแถวเกิดความซ้ำซ้อนกันได้

2.1.1.4 ค่าว่าง (Null Value)

ค่าว่าง คือค่าของ Attribute อาจจะเป็นค่าว่าง (Null) หรือไม่มีค่า หรือยังไม่ทราบค่าได้ เช่น GPA ของนักศึกษา ซึ่งจะสามารถทราบค่าได้ก็ต่อเมื่อนักศึกษาจบภาคการศึกษานั้นๆ แล้ว เป็นต้น

2.1.1.5 ค่าเอกลักษณ์ (Unique Value)

ค่าเอกลักษณ์ คือค่าของ Attribute ในแต่ละแถว ของ Relation นั้นๆ ต้องไม่ซ้ำซ้อนกัน เช่น Attribute รหัสนักศึกษา ซึ่งในแต่ละแถวจะต้องไม่มีรหัสนักศึกษาที่ซ้ำซ้อนกัน เป็นต้น

2.1.1.6 การนอร์มัลไลซ์ (Normalization)

คือกระบวนการปรับเปลี่ยนโครงสร้างของ Relation Schema ให้อยู่ในรูปแบบ Normal Form (NF) เพื่อลดความซ้ำซ้อนของข้อมูล และขจัดปัญหาความผิดปกติของข้อมูลที่เกิดจากการเพิ่ม/การลบ หรือการแก้ไขข้อมูล โดย Normal Form ก็คือรูปแบบโครงสร้างของ Relation ที่มีคุณสมบัติเฉพาะ ซึ่ง NF จะมีด้วยกันหลายระดับ ในระดับที่สูงก็จะมีจัดการโครงสร้างข้อมูลที่ดีกว่าในระดับต่ำกว่า (“การนอร์มัลไลซ์”, 2546) ซึ่งการนอร์มัลไลเซชัน แบ่งออกได้หลายระดับ ดังนี้

(1) First Normal Form (1NF)

First Normal Form มีคุณสมบัติคือ ทุกๆ Attribute ในแต่ละแถวจะต้องเป็น Single Value ซึ่งก็คือ ในแต่ละ Relation จะไม่มีค่าของกลุ่มข้อมูลที่ซ้ำกัน (Repeating Group) ดังตัวอย่าง

ตัวอย่าง Relation ที่เป็น Non-1NF

ตารางที่ 2.1 แสดง Relation ที่เป็น Non-1NF

รหัสนักศึกษา	ชื่อ	นามสกุล	รหัสวิชาที่ลงทะเบียน
580909443	วรรณภา	พรมจักรา	204-101
			204-204
			204-205
58090384	ณัฐนิชา	บุญมาศ	204-102
			204-204

จากตารางที่ 2.1 จะเห็นได้ว่านักศึกษาคนหนึ่ง สามารถลงทะเบียนได้มากกว่าหนึ่งวิชา ซึ่งการจัดเก็บข้อมูลในลักษณะนี้เป็นการสิ้นเปลืองโดยใช้เหตุ เนื่องจากมีค่าของกลุ่มข้อมูลที่ซ้ำซ้อนกันมากมาย

ตัวอย่าง การทำนอร์มัลไลเซชันให้อยู่ในรูป 1NF

ตารางที่ 2.2 แสดง Relation ที่เป็น 1NF

รหัสนักศึกษา	ชื่อ	นามสกุล	รหัสวิชาที่ลงทะเบียน
580909443	วรรณภา	พรมจักรา	204-101
580909443	วรรณภา	พรมจักรา	204-204
580909443	วรรณภา	พรมจักรา	204-205
58090384	ณัฐนิชา	บุญมาศ	204-102
58090384	ณัฐนิชา	บุญมาศ	204-204

จากตาราง ที่ 2.2 เป็นตัวอย่างการทำนอร์มัลไลเซชันให้อยู่ในรูป 1NF เพื่อขจัด Attribute หรือกลุ่มของ Attribute ที่ซ้ำซ้อนกัน แต่อย่างไรก็ตาม การทำนอร์มัลไลเซชัน ระดับที่ 1 (1NF) ก็ยังมีข้อด้อยในการเพิ่ม/ลบ หรือแก้ไขข้อมูลดังนี้

1. การเพิ่มข้อมูล (Insert) อาจทำให้ไม่สามารถเพิ่มข้อมูลบางอย่างได้ หรือเมื่อเพิ่มแล้วทำให้ขัดแย้งกับข้อมูลเดิม
2. การลบข้อมูล (Delete) การลบข้อมูลบางส่วนออกไป จะทำให้ลบ ข้อมูลอื่นออกไปด้วย โดยที่ไม่ได้ตั้งใจ
3. การแก้ไขข้อมูล (Update) เนื่องจากมีข้อมูลอยู่หลายแถว ทำให้ เวลาแก้ไขจะต้องแก้ไขข้อมูลทุกแถว

(2) Second Normal Form (2NF)

Second Normal Form มีคุณสมบัติ คือต้องเป็น First Normal Form (1NF) และในหนึ่ง Relation จะต้องมีการมี Attribute ที่เป็น Primary Key และทุก Non-key Attribute จะต้องขึ้นอยู่กับ Attribute ที่เป็น Primary Key นี้ ซึ่งการนอร์มัลไลเซชันระดับที่ 2 นี้ เป็นการขจัด Attribute ที่ไม่ขึ้นกับ Primary Key ออกไป เพื่อให้ Attribute อื่นทั้งหมด ขึ้นตรงกับส่วนที่เป็น Primary Key เท่านั้น

ตัวอย่าง Relation ที่เป็น Non-2NF

ตารางที่ 2.3 แสดง Relation ที่เป็น Non-2NF

รหัสนักศึกษา	ชื่อ	นามสกุล	รหัสวิชา	ชื่อวิชา	ภาคการศึกษา
580909443	วรรณภา	พรมจักรา	204-101	วิทยาศาสตร์	1/2558
580909443	วรรณภา	พรมจักรา	204-204	คอมพิวเตอร์	1/2558
580909443	วรรณภา	พรมจักรา	204-205	คณิตศาสตร์	2/2558
58090384	ณัฐนิชา	บุญมาศ	204-102	ภาษาอังกฤษ	2/2558
58090384	ณัฐนิชา	บุญมาศ	204-204	คอมพิวเตอร์	2/2558

จากตารางที่ 2.3 จะเห็นได้ว่า Attribute ชื่อ และ นามสกุล จะขึ้นอยู่กับ รหัสนักศึกษา เพียงอย่างเดียว ไม่ได้ขึ้นกับ รหัสวิชาที่ลงทะเบียน ส่วน ชื่อวิชาที่ลงทะเบียน ก็ขึ้นอยู่กับ รหัสวิชาที่ลงทะเบียน เพียงอย่างเดียว โดยไม่ขึ้นกับ รหัสนักศึกษา เช่นกัน การเก็บข้อมูลในลักษณะนี้ทำให้บางข้อมูลแถวไม่สามารถค้นหาข้อมูลโดยใช้คีย์ได้ และยังทำให้เกิดความซ้ำซ้อนของข้อมูล

ตัวอย่าง การทำนอร์มัลไลเซชันให้อยู่ในรูป 2NF

ตารางที่ 2.4 แสดง Relation นักศึกษา

รหัสนักศึกษา	ชื่อ	นามสกุล
580909443	วรรณภา	พรมจักรา
58090384	ณัฐนิชา	บุญมาศ

ตารางที่ 2.6 แสดง Relation การลงทะเบียน

รหัสนักศึกษา	รหัสวิชา	ภาคการศึกษา
580909443	204-101	1/2558
580909443	204-204	1/2558
580909443	204-205	2/2558
58090384	204-102	2/2558
58090384	204-204	2/2558

ตารางที่ 2.5 แสดง Relation วิชา

รหัสวิชา	ชื่อวิชา
204-101	วิทยาศาสตร์
204-102	ภาษาอังกฤษ
204-204	คอมพิวเตอร์
204-205	คณิตศาสตร์

จากตารางที่ 2.4 - 2.6 เป็นตัวอย่างการทำ นอร์มัลไลเซชันให้อยู่ในรูป 2NF ซึ่ง เป็นการขจัด Attribute ที่ไม่ขึ้นกับคีย์หลักออกไป เพื่อให้ Attribute อื่นทั้งหมดขึ้นตรงกับ ส่วนที่เป็นคีย์หลักทั้งหมด จะช่วยทำให้สามารถแก้ไขปัญหาการสืบค้นข้อมูลบางอย่างโดยใช้คีย์ไม่ได้ ออกไป รวมถึงลดความซ้ำซ้อนในการเก็บข้อมูลอีกด้วย

(3) Third Normal Form (3NF)

Third Normal Form มีคุณสมบัติคือต้องเป็น Second Normal Form (2NF) และในหนึ่ง Relation จะต้องไม่มี Attribute ที่ไม่ได้เป็น Primary Key ขึ้นตรงกับ Attribute อื่นที่ไม่ใช่ Primary Key (Transitively Dependency หรือการไม่ขึ้นตรงกับคีย์หลัก)

ตัวอย่าง Relation ที่เป็น Non-3NF

ตารางที่ 2.7 แสดง Relation ที่เป็น Non-3NF

รหัสลูกค้า	ชื่อลูกค้า	ระดับ	ประเภท
150001	วนิดา สุขสรร	ชั้นดี	A
150002	ชมพูนุช เป็นเอก	ชั้นกลาง	B
150003	กชينا ปานระภา	ชั้นกลาง	C

จากตารางที่ 2.7 จะเห็นได้ว่า Attribute ประเภท จะขึ้นอยู่กับ Attribute ระดับ ซึ่งเป็น Attribute อื่นที่ไม่ใช่ Primary Key จึงทำให้ Relation ข้างต้น ไม่มี คุณสมบัติ Third Normal Form (3NF)

ตัวอย่าง การทำนอร์มัลไลเซชันให้อยู่ในรูป 3NF

ตารางที่ 2.8 แสดง Relation ข้อมูลลูกค้า

รหัสลูกค้า	ชื่อลูกค้า	ระดับ
150001	วนิดา สุขสรร	ชั้นดี
150002	ชมพูนุช เป็นเอก	ชั้นกลาง
150003	กชينا ปานระภา	พอใช้

ตารางที่ 2.9 ระดับความสำคัญของลูกค้า

ระดับ	ประเภท
ชั้นดี	A
ชั้นกลาง	B
พอใช้	C

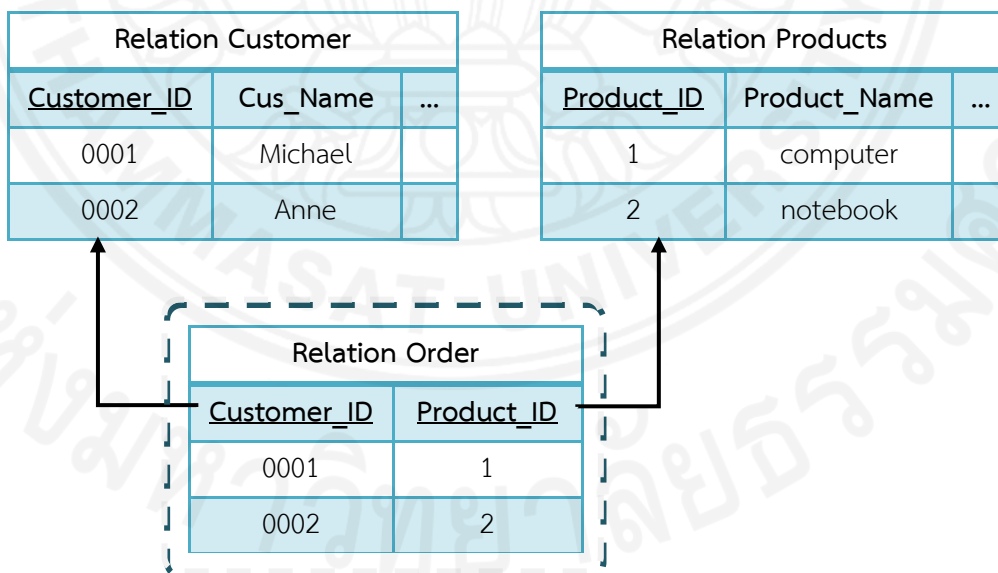
จากตารางที่ 2.8 - 2.9 เป็นตัวอย่างการทำ นอร์มัลไลเซชันให้อยู่ในรูป 3NF ซึ่งการทำ 3NF เป็นการขจัด Attribute ที่ไม่ใช่คีย์ที่ขึ้นตรงกับ Attribute อื่นที่ไม่ใช่ Primary Key ออกไป เพื่อลดความซ้ำซ้อนในการเก็บข้อมูลและทำให้อ้างถึงข้อมูลได้ง่ายขึ้น

2.1.1.7 Binary Relation

Binary Relation คือ Relation ที่อยู่ระหว่าง 2 Relation ที่มี ความสัมพันธ์กัน โดยมีลักษณะดังต่อไปนี้

1. R เป็น Binary Relation ระหว่าง 2 Relation คือ A และ B
2. $A \neq R$ และ $B \neq R$
3. R มี 2 Attribute คือ a และ b
4. a และ b เป็น Primary Key ใน R
5. a เป็น Foreign Key ที่ References ไปยัง attribute c ใน A
6. b เป็น Foreign Key ที่ References ไปยัง attribute d ใน B

ตัวอย่าง Binary Relation



ภาพที่ 2.3 ตัวอย่าง Binary Relation

จากภาพที่ 2.3 จะเห็นได้ว่า Relation Order มีคุณสมบัติเป็น Binary Relation เนื่องจากเป็น Relation ที่อยู่ระหว่าง Relation Customer และ Relation Products โดยที่ Relation Customer และ Relation Products ไม่ได้มีคุณสมบัติเป็น Binary Relation นอกจากนี้ Relation Order ยังประกอบด้วย 2 Attribute อันได้แก่ Customer_ID และ Product_ID โดยที่ทั้ง 2 Attribute มีคุณสมบัติเป็นทั้ง Primary Key และ Foreign Key ซึ่ง References ไปยัง Relation Customer และ Products ตามลำดับ

2.1.1.8 Inheritance Relation

ถ้า 2 Relation ที่ไม่ได้เป็น Binary Relation คือ R_1 และ R_2 แล้ว Attribute A_i เป็น Primary Key และ Foreign Key ของ R_1 ซึ่ง Reference ไปยัง R_2 เพียง Relation เดียว จะนิยามได้ว่า R_1 มีคุณสมบัติเป็น Inheritance Relation

ตัวอย่าง Inheritance Relation

Relation Products			Relation Products_Location		
Field	Key	Table Ref	Field	Key	Table Ref
<u>ProductID</u>	PK		<u>ProductLocalID</u>	PK,FK	Products
ProductName			Address		
SupplierID	FK	Suppliers	PostCode		
CategoryID	FK	Categories	Country		
QuantityPerUnit					

ภาพที่ 2.4 ตัวอย่าง Inheritance Relation

จากภาพที่ 2.4 จะเห็นได้ว่า Relation Products_Location มีคุณสมบัติเป็น Inheritance Relation เนื่องจากไม่มีคุณสมบัติเป็น Binary Relation และ Attribute ProductLocalID ของ Relation เป็น Primary Key และ Foreign Key ซึ่ง Reference ไปยัง Relation Products เพียง Relation เท่านั้น

2.1.2 ออนโทโลยี (Ontology)

ออนโทโลยี (Ontology) เป็นฐานความรู้ ที่อธิบายรูปแบบโครงสร้าง ความสัมพันธ์ระหว่างข้อมูลในขอบเขตที่สนใจ (Domain) ให้ได้ใจความและมีความถูกต้องมากที่สุด ซึ่งเป็นเทคโนโลยีด้านการพัฒนาภาษาเชิงความหมาย ให้เป็นภาษาที่เครื่องคอมพิวเตอร์สามารถเข้าใจ ความหมาย (Semantic) และทำตามคำสั่งได้ ออนโทโลยีถือเป็นฐานความรู้ที่ถูกสร้างขึ้นโดยมีโดเมน ที่จำกัด และมีความสำคัญในแง่ของการเป็นศูนย์กลางความรู้ ซึ่งสามารถถูกนำมาใช้งานร่วมกันได้ นอกจากนี้คำจำกัดความของออนโทโลียังหมายถึงการอธิบายความสัมพันธ์ของ โครงสร้างความรู้ ให้อยู่ในรูปแบบ ของลำดับชั้น ในเชิงวัตถุ (Hierarchical Data Structure) เพื่ออธิบายขอบเขต ขององค์ความรู้ที่สนใจ

ออนโทโลยีเป็น อีกแนวทางหนึ่ง ที่ช่วยในการจัดการฐานความรู้ และได้มีนำมา ประยุกต์ใช้กับงาน ใน ระบบต่างๆ เช่น ระบบฐานข้อมูลทางด้านชีววิทยา , ระบบ พาณิชย อีเล็กทรอนิกส์, รวมไปถึงระบบงานต่างๆ ที่พัฒนาขึ้นบนเว็บเชิงความหมาย เพื่อช่วยในการจัดเก็บ และค้นคืนความรู้ การแลกเปลี่ยน และการนำความรู้กลับมาใช้ใหม่

2.1.2.1 องค์ประกอบของออนโทโลยี

ออนโทโลยีเป็นการแสดงโครงสร้างของแนวความคิด ที่บรรยายของเขต ขององค์ความรู้ในเรื่องใดเรื่องหนึ่ง ซึ่งประกอบไปด้วยการนิยามความหมายหรือแนวคิด (Concepts) ซึ่งเป็นพื้นฐานสำคัญในการสร้างฐานความรู้ โดยแนวคิดเหล่านี้จัดเรียงอยู่ในลำดับชั้นของการ ถ่ายทอดความสัมพันธ์ (ทิพย์สุริย์ ติรวงศ์กุล และ จันอ๊อ แซ่เจิน, 2551, น. 4) ซึ่งสามารถสรุปได้ดังนี้

(1) แนวคิด (Concepts)

แนวคิด (Concepts) คือ ขอบเขตของความรู้ในเรื่องใดเรื่องหนึ่งและ สามารถอธิบายรายละเอียดได้ เช่น People, Expertise, Science, Education เป็นต้น

(2) คุณสมบัติ (Property)

คุณสมบัติ (Property) คือ คุณสมบัติต่างๆ ที่นำมาใช้อธิบายรายละเอียด ของแนวคิด เช่น ตำแหน่งงาน (Position), สถานที่ทำงาน (Office) เป็นต้น

(3) ความสัมพันธ์ (Relationships)

ความสัมพันธ์ (Relationships) คือ รูปแบบการแสดงความสัมพันธ์ ระหว่างแนวความคิด โดยแบ่งออกเป็นแบบต่างๆ ดังนี้

1. ความสัมพันธ์แบบลำดับชั้น (Subclass of หรือ is-a hierarchy) เป็นความสัมพันธ์แบบการถ่ายทอดคุณสมบัติ โดยจะถ่ายทอดคุณสมบัติของแนวความคิดแม่ไปยังลูก เช่น Computer Science is – a Science ซึ่งสามารถอธิบายได้ว่า วิทยาการคอมพิวเตอร์ (Computer Science) เป็นสาขาของ วิทยาศาสตร์ (Science)

2. ความสัมพันธ์แบบเป็นส่วนหนึ่ง (Part - of) เป็นความสัมพันธ์แบบการเป็นส่วนประกอบ เช่น Wheel part - of Car ซึ่งสามารถอธิบายได้ว่า ล้อรถ (Wheel) เป็นส่วนหนึ่งของรถ (Car)

3. ความสัมพันธ์เชิงความหมาย (Syn - of) เป็นความสัมพันธ์ที่แสดงถึงแนวความคิดที่มีความเหมือนในเชิงความหมายต่อกัน เช่น Degree syn - of Education ซึ่งสามารถอธิบายได้ว่า ระดับการศึกษา (Degree) มีความหมายเดียวกับการศึกษา (Education) โดยสามารถใช้แทนกันได้

4. ความสัมพันธ์แบบเป็นตัวแทน (Instance-of) เป็นความสัมพันธ์ที่แสดงถึงการเป็นตัวแทน หรือสมาชิกของแนวคิด

5. นอกจากนี้ออนโทโลยี ยังประกอบไปด้วยความสัมพันธ์เชิงความหมายอื่นๆ ที่สอดคล้องกับแนวความคิด ซึ่งจะกำหนดโดยผู้เชี่ยวชาญ

(4) ข้อกำหนดการสร้างความสัมพันธ์ (Axioms)

ข้อกำหนดการสร้างความสัมพันธ์ (Axioms) คือ เงื่อนไขหรือตรรกะในการแปลงความสัมพันธ์ระหว่างแนวคิดกับแนวคิด หรือ คุณสมบัติ เพื่อการแปลงความหมายที่ต้องการ

(5) ตัวอย่างข้อมูล (Instance)

ตัวอย่างข้อมูล (Instance) คือ คำศัพท์ที่มีการกำหนดความหมายไว้ในออนโทโลยีเรื่องนั้นๆ

2.1.2.2 ภาษาที่ใช้อธิบายออนโทโลยี

ภาษาที่ใช้ในการพัฒนาออนโทโลยี สามารถแบ่งตามรูปแบบภาษาได้ดังต่อไปนี้

(1) ลอจิกเบส (First order logic based)

ลอจิกเบส (First order logic based) คือ รูปแบบของภาษาที่อธิบายออนโทโลยี โดยการใช้ตรรกะเพื่อการอนุมานความรู้ โดยส่วนประกอบคือ ภาคแสดง (Predicate), อาร์กิวเมนต์ (Argument) และตัวบ่งปริมาณ (Quantifier) คุณสมบัติของภาษาออนโทโลยีแบบลอจิกเบส คือ วากยสัมพันธ์ (Syntax) ต้องมีความชัดเจนและมีรูปแบบที่เป็นทางการ จึงจะสามารถสร้างกฎอนุมานได้ ตัวอย่างภาษา เช่น CYCL, CLASSIC, LOOM เป็นต้น

(2) เฟรมเบส (Frame logic based)

เฟรมเบส (Frame logic based) คือ รูปแบบของภาษาที่อธิบายออนโทโลยี โดยใช้หลักการของการคิดเชิงปฏิสัมพันธ์ของมนุษย์ เมื่อนึกถึงวัตถุ หรือสิ่งใดที่เชื่อมโยงกับคุณลักษณะเด่นของวัตถุหรือสิ่งนั้นด้วย ดังนั้น เฟรมเบสจึงประกอบไปด้วย เซตของคุณสมบัติ ข้อความอธิบายแนวคิด หรือคำอธิบายคุณสมบัติ โดยคุณสมบัติของภาษาออนโทโลยีที่พัฒนาแบบเฟรมเบส คือเข้าใจง่าย ตัวอย่างภาษาเช่น OKBC และ F-logic เป็นต้น

(3) เว็บเบส (Web based)

เว็บเบส (Web based) คือ รูปแบบของภาษาที่อธิบายออนโทโลยี ซึ่งพัฒนาจากภาษาที่ใช้สำหรับอธิบายทรัพยากรบนเว็บ ได้แก่ XML (Extensible Markup Language), RDF (Resource Description Framework) ซึ่งอธิบายออนโทโลยี โดยใช้พื้นฐานของลอจิกเบสและเฟรมเบส ภาษาที่พัฒนาขึ้นจึงมีหลักการและแบบแผน สามารถอ้างอิงได้และอยู่ในรูปแบบที่มนุษย์สามารถเข้าใจได้ง่าย ตัวอย่างภาษา เช่น DAML + OIL (DARPA Agent Markup Language + Ontology Interface Language) และ OWL (Web Ontology Language) เป็นต้น

2.1.2.3 การประยุกต์ใช้ออนโทโลยี

ออนโทโลยีได้ถูกนำไปประยุกต์ใช้ในหลายๆ องค์กร ในหลายๆ งาน โดยพิจารณาจากวัตถุประสงค์ของการนำไปใช้งาน ดังนี้

1. การประยุกต์ใช้ออนโทโลยี เพื่อการเข้าถึงข้อมูลที่มีโครงสร้าง หรือมีรูปแบบที่แตกต่างกัน (Common Access to Information) โดยออนโทโลยีจะทำหน้าที่จัดกลุ่มคำ หรือเตรียมคำที่มีความหมายเดียวกันหรือสามารถเข้าใจตรงกันได้ เพื่อให้สามารถทำงานร่วมกัน และนำกลับมาใช้ใหม่ได้
2. ระบบสืบค้นข้อมูล ออนโทโลยีมีส่วนช่วยในการขยายคำที่ใช้ในการสืบค้นข้อมูล โดยการพิจารณาความหมายของสิ่งต่างๆ ที่สนใจ ทำให้สามารถค้นหาและเข้าถึงข้อมูลได้ตรงตามความต้องการของผู้ใช้งาน มีความถูกต้องแม่นยำมากขึ้น และยังช่วยลดระยะเวลาในการสืบค้นอีกด้วย ตัวอย่างระบบ เช่น ระบบค้นหาการท่องเที่ยว, ระบบค้นหาพันธบัตร เป็นต้น
3. การประยุกต์ใช้ออนโทโลยี เพื่อการแปลงข้อมูลให้อยู่ในรูปแบบภาษาต่างๆ (Neutral Authoring) เพื่อให้โปรแกรมอื่นๆ สามารถนำไปใช้งานให้เกิดประโยชน์ได้
4. การประยุกต์ใช้ออนโทโลยี เพื่อใช้กำหนดรายละเอียดของซอฟต์แวร์ (Ontology as Specification) เพื่อให้สามารถออกแบบซอฟต์แวร์ในโดเมนและรวบรวมคำศัพท์สำหรับกำหนดความต้องการในการพัฒนาซอฟต์แวร์ ซึ่งประโยชน์ที่ได้คือ การทำคู่มือโปรแกรม , การบำรุงรักษาซอฟต์แวร์และการนำกลับมาใช้ใหม่

2.1.3 Web Ontology Language (OWL)

ภาษา OWL ถูกพัฒนาขึ้นโดยดับเบิลยูทีซี (W3C) จัดว่าเป็นองค์ประกอบหนึ่งในงานเว็บเชิงความหมาย (Semantic Web) ที่ใช้สำหรับบรรยายข้อมูลเชิงความหมาย ซึ่งสามารถกำหนดโครงสร้างในลักษณะลำดับชั้น และอธิบายข้อมูล (Metadata) ที่มีความสัมพันธ์ในระบบฐานข้อมูลได้ รวมทั้งสามารถรองรับการบรรยายข้อมูลเชิงตรรกะ ชนิดข้อมูล และตัวบ่งปริมาณได้ ทำให้ข้อมูลที่ถูกแทนที่นั้นมีความหมายมากยิ่งขึ้น ลักษณะการบรรยายจะอยู่ในรูปของคลาส , คุณสมบัติของคลาส และความสัมพันธ์ของคลาส เพื่ออธิบายแอนติตี้และความสัมพันธ์ต่างๆ ที่เกิดขึ้น

OWL เป็นภาษาที่พัฒนาต่อมาจากภาษา RDF (Resource Description Framework) และสืบทอดมาจากภาษา DAML (DARPA Agent Markup Language) +OIL (Ontology Interchange Language) โดยภาษา OWL มีประสิทธิภาพอย่างมากในการอธิบายเนื้อหาต่างๆ ตามขอบเขต ซึ่งคอมพิวเตอร์สามารถอ่านค่าและเข้าใจความหมายของข้อมูล

ภาษา OWL แบ่งออกเป็น 3 ประเภท คือ OWL LITE, OWL DL, OWL FULL โดยแต่ละประเภทจะถูกออกแบบมาให้เหมาะสมกับการใช้งานตามกลุ่มการใช้งาน (ศุภกฤษฎี นวัตกรรมกุล, 2556, น. 24)

2.1.3.1 ประเภทของภาษา OWL

(1) OWL LITE

ออกแบบมาเพื่อสนับสนุนการใช้งานเบื้องต้น โดยมีการกำหนดโครงสร้างในรูปแบบลำดับชั้น และมีการบังคับใช้คุณสมบัติพื้นฐาน ในการกำหนดโครงสร้างข้อมูล โดย OWL LITE ถูกออกแบบมาให้ใช้ในการพัฒนาได้ง่ายและมีการเตรียมฟังก์ชันการใช้งานต่างๆ สำหรับเริ่มใช้งานในการเขียน OWL ได้

(2) OWL DL

ออกแบบเพื่อสนับสนุนการอธิบาย Logic Business Segment โดยใน OWL DL จัดให้มีคุณสมบัติที่เหมาะสมกับการใช้งานด้านฐานข้อมูล และการแทนความรู้ ที่ตั้งอยู่บนพื้นฐานของการอธิบายด้วยเหตุผลทางตรรกะ OWL DL สามารถบรรยายข้อมูลและโครงสร้างข้อมูลในรูปแบบของโครงสร้างภาษา OWL ด้วยการกำหนดข้อจำกัดของคลาสและคุณสมบัติของคลาสได้

(3) OWL FULL

ออกแบบมาเพื่อสนับสนุนผู้ใช้งานที่ต้องการความครบถ้วนและมีโครงสร้างภาษาที่สมบูรณ์แบบ โดย OWL FULL จะมีการผสมผสานกันระหว่าง OWL และ RDF Schema ผู้ใช้งานสามารถบรรยายข้อมูลในรูปแบบ RDF Schema ได้อย่างอิสระ

ทั้ง OWL DL และ OWL FULL ต่างก็สนับสนุนเซตของภาษา OWL แต่มีข้อจำกัดของคุณลักษณะบางอย่างที่แตกต่างกันบนพื้นฐานของ RDF Schema โดย OWL FULL จะมีการผสมผสานกันระหว่าง OWL และ RDF Schema โดยไม่มีการบังคับในส่วนการแบ่งคลาส การกำหนดคุณสมบัติ และค่าของข้อมูล ส่วน OWL DL จะมีข้อบังคับในการใช้ RDF การกำหนดคลาส การกำหนดคุณสมบัติ และค่าของข้อมูลเป็นต้น

2.1.3.2 โครงสร้างของภาษา OWL

ภาษา OWL จะประกอบด้วยกลุ่มของข้อมูล Namespace, Ontology Header, Class, Property และรายละเอียดต่างๆ ของข้อมูล โดยไฟล์นามสกุลที่ใช้สำหรับการสร้างเอกสาร เป็นไฟล์นามสกุล .owl หรือ .rdf โดยโครงสร้างของภาษา OWL มีดังต่อไปนี้ (ศุภกฤษฎี นวัตกรรมกุล, 2556, น. 25-30)

(1) Namespace

การกำหนด Namespace จะประกาศไว้ที่ส่วนเริ่มต้นของเอกสาร เพื่อเป็นการกำหนดกลุ่มในการอ้างอิงข้อมูล โดยเอกสาร OWL ที่ถูกสร้างจะขึ้นอยู่กับโครงสร้างของภาษา RDF, RDFS และชนิดข้อมูลเป็นแบบโครงสร้างภาษา RDFS ซึ่งการเขียน Namespace จะประกาศไว้ภายใต้คำสั่งของ rdf:RDF syntax ดังตัวอย่างภาพที่ 2.5

ตัวอย่าง รหัสต้นฉบับการกำหนด Namespace

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#"
```

ภาพที่ 2.5 การกำหนด Namespace

(2) Ontology Header

เป็นการอธิบายรายละเอียดเบื้องต้นของออนโทโลยี ภายใต้โอลิเมนต์ <owl:Ontology rdf:about=""> ประกอบด้วยคำสั่ง <owl:versionInfo> ใช้แสดงรุ่นของข้อมูล ที่ทำการสร้าง , คำสั่ง <rdf:comment> ใช้แสดงส่วนการอธิบายหมายเหตุของข้อมูล และคำสั่ง <owl:imports rdf:resource=""> ใช้แสดงการอ้างอิงเกี่ยวกับแหล่งข้อมูล ดังตัวอย่างภาพที่ 2.6

ตัวอย่าง รหัสต้นฉบับโครงสร้าง Ontology Header

```
<owl:Ontology rdf:about="">
  <owl:versionInfo>V 1.17 2003/02/26 </owl:versionInfo>
  <rdfs:comment>An example ontology</rdfs:comment>
  <owl:imports rdf:resource="http://www.example.org/owl"/>
</owl:Ontology>
```

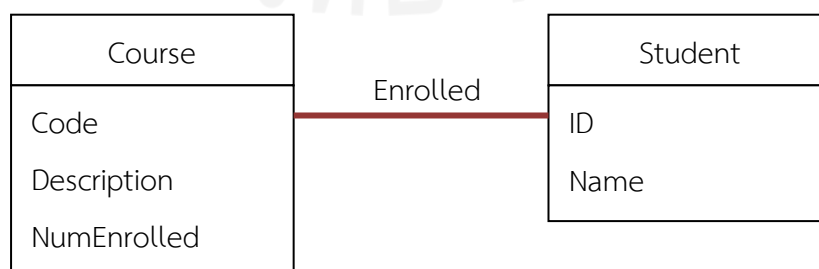
ภาพที่ 2.6 อธิบายรายละเอียดเบื้องต้นของออนโทโลยี

2.1.3.3 อลิเมนต์พื้นฐาน (Basic Element)

อิลิเมนต์พื้นฐานของ OWL Ontology ส่วนมากจะเกี่ยวข้องกับคลาส , คุณสมบัติของคลาส , ตัวแทนของข้อมูล (Instances ของคลาส) และความสัมพันธ์ระหว่างตัวแทนข้อมูลต่างๆ โดยมีรายละเอียดดังนี้

(1) คลาสและคลาสย่อย (Class and SubClass)

คลาส (Class) เป็นโครงสร้างที่ใช้กำหนดกลุ่มของข้อมูลหรือ Object ที่แหล่งของข้อมูลมีคุณลักษณะคล้ายคลึงกัน โดยทุกคลาสจะมีความสัมพันธ์กับเซตของข้อมูลที่มีลักษณะเฉพาะหรือที่เรียกว่า Class Extension ซึ่งเป็นลักษณะของการสืบทอดคลาส โดย Individual ภายใน Class Extension ถูกเรียกว่าตัวแทนข้อมูล จะมีลักษณะโครงสร้างเป็นลำดับชั้น และมีการสืบทอดคุณสมบัติของคลาส ในโครงสร้างของภาษา OWL กล่าวได้ว่า ทุกๆ สิ่งที่มีลักษณะเฉพาะอย่างทั้งหลาย จะเป็นสมาชิกของคลาส owl:Thing ซึ่งถูกกำหนดให้เป็นคลาสใหญ่ที่ครอบคลุมทุกคลาสข้อมูล ดังนั้นไม่ว่าผู้ใช้งานจะสร้างคลาสใดขึ้นมา จะเสมือนว่าเป็นสมาชิกอยู่ภายใต้คลาส owl:Thing และในทางตรงกันข้าม owl:Nothing ใช้ในการกำหนดคลาสที่ไม่มีสมาชิกภายในคลาสนั้น โดยอ้างอิงเพียงบางข้อมูลที่เกี่ยวข้องกัน เพื่อใช้เป็นตัวอย่างการแทนข้อมูลได้ ดังแสดงในภาพที่ 2.7 (Hart et al, 2004)



ภาพที่ 2.7 ตัวอย่างตารางข้อมูลของระบบลงทะเบียนเรียน

จากภาพที่ 2.7 จะแสดงตัวอย่างตารางข้อมูลของระบบลงทะเบียนเรียน โดยประกอบด้วยตารางข้อมูลที่มีความสัมพันธ์กัน 2 ตาราง คือ ตารางหลักสูตร (Course) และ ตารางข้อมูลนักศึกษา (Student) ซึ่งทั้ง 2 ตารางนี้มีความสัมพันธ์ที่มีชื่อว่าลงทะเบียน (Enrolled) โดยตารางข้อมูลหลักสูตร ประกอบด้วยคุณสมบัติ รหัสหลักสูตร (Code), คำอธิบายหลักสูตร (Description) และจำนวนผู้ลงทะเบียน (NumEnrolled) ส่วนตารางข้อมูลนักศึกษา ประกอบด้วยคุณสมบัติ รหัสนักศึกษา (ID) และชื่อนักศึกษา (Name) ซึ่งรูปแบบของการเขียนคลาสสามารถที่จะ กำหนดการประกาศสำหรับการใช้งานคลาสแบบเฉพาะเจาะจงได้ และสามารถประกาศชื่อคลาสได้ ที่ตามต้องการ ดังตัวอย่างภาพที่ 2.8

ตัวอย่าง การประกาศชื่อคลาส

```
<owl:Class rdf:ID="Course"/>
<owl:Class rdf:ID="Student"/>
```

ภาพที่ 2.8 การประกาศชื่อคลาส

จากตัวอย่าง การกำหนดชื่อคลาสจะอยู่ภายใต้ Syntax `rdf:ID=""` โดยภายใต้เครื่องหมาย " " จะแทนด้วยชื่อคลาส ซึ่งการแทนชื่อคลาสในภาษา OWL สามารถเขียนได้หลายรูปแบบ เช่น `#Course` เขียนแทนด้วย syntax `rdf:resource="#Course"` และนอกจากนี้ คลาสยังมีโครงสร้างพื้นฐาน คือ `rdfs:subClassOf` โดยจะมีความสัมพันธ์กับการเฉพาะเจาะจงกับคลาสโดยทั่วไป เช่น ถ้า X เป็น SubClass ของ Y ดังนั้น ทุกๆ Instance ของ X จะเป็น Instance ของ Y ด้วย ซึ่ง `rdfs:subClassOf` จะเป็นความสัมพันธ์แบบ Transitive คือถ้า X เป็น SubClass ของ Y และ Y เป็น SubClass ของ Z ดังนั้น X จะเป็น SubClass ของ Z ด้วย ดังภาพที่ 2.9 เป็นตัวอย่างการแทนค่าข้อมูล โดยจะกำหนดให้ Subject เป็น SubClass ของ Course

ตัวอย่าง การประกาศชื่อคลาสร้อย

```
<owl:Class rdf:ID="#Subject"/>
  <owl:subClassOf rdf:ID="#Course"/>
</owl:Class>
```

ภาพที่ 2.9 การประกาศชื่อคลาสร้อย

การนิยามคลาส จะประกอบด้วย 2 ส่วน คือ ชื่อคลาส ที่ใช้แทนการอ้างอิงการใช้งานคลาสนั้น และการแสดงรายละเอียดหรือข้อกำหนดต่างๆ ซึ่งส่วนของการอธิบายสมาชิกภายในคลาส ทำได้โดยการประกาศถึงคุณลักษณะเฉพาะของสมาชิกภายในคลาสดังตารางที่ 2.10

ตารางที่ 2.10 ข้อมูลสมาชิกภายใน Class Course

Code	Description	NumEnrolled
INFS3101	Ontology and Semantic Web	5

จากตารางที่ 2.10 แสดงข้อมูลสมาชิกภายในคลาส Course โดยข้อมูลที่ใช้แสดงถึงข้อมูลเฉพาะคือ Code ซึ่งสามารถเขียนบรรยายในรูปแบบของภาษา OWL ได้ดังนี้
`<owl:Thing rdf:ID="INFS3101 "/>`

(2) คุณสมบัติของคลาส (Property)

ภาษา OWL ได้แบ่งคุณสมบัติออกเป็น 2 ประเภท ได้แก่ Datatype Property และ Object Property ดังแสดงในตารางที่ 2.11 และในแต่ละคุณสมบัติก็มีโครงสร้างในรูปแบบต่างๆ ซึ่งสามารถแสดงได้ดังตารางที่ 2.12

ตารางที่ 2.11 ประเภทของคุณสมบัติ

ประเภทคุณสมบัติ	นิยาม
1. Datatype Property	แสดงความสัมพันธ์ระหว่าง instance ของคลาสและ RDF literals รวมถึงการแสดงชนิดข้อมูล XML Schema Datatype
2. Object Property	แสดงความสัมพันธ์ระหว่าง instance ของคลาสที่เกี่ยวข้องกันระหว่าง 2 คลาส

ตารางที่ 2.12 โครงสร้างของคุณสมบัติในรูปแบบต่างๆ

RDF Schema :	rdfs:subPropertyOf, rdfs:domain และ rdfs:range
Relations to other properties :	owl:equivalentProperty และ owl:inverseOf
Global cardinality constraints	owl:FunctionalProperty และ owl:InverseFunctionalProperty
Logical Property characteristics	owl:SymmetricProperty และ owl:TransitiveProperty

(3) คุณสมบัติและชนิดข้อมูล

โครงสร้างของภาษา OWL จะประกอบด้วยชนิดของข้อมูลที่ต้องการใช้งาน ซึ่งการประกาศการใช้งานจะอยู่ในรูปแบบคำสั่ง XML Schema Datatype โดยการประกาศใช้จะอ้างอิงตามชนิดข้อมูลที่ใช้งานจริง และมีการประกาศการใช้งาน IRI ที่มีการอ้างอิงการใช้งานชนิดข้อมูลนั้นๆ ซึ่งมีรูปแบบคำสั่งคือ <http://www.w3.org/2001/XMLSchema#NAME> โดย NAME จะแทนการอ้างอิงประเภทของชนิดข้อมูล ชนิดข้อมูลที่อ้างอิงตาม XML Schema Datatype แสดงได้ดังตารางที่ 2.13

ตารางที่ 2.13 ชนิดข้อมูลที่อ้างอิงตาม XML Schema datatype

กลุ่มชนิดข้อมูล	คำสั่งที่ใช้งาน
ประเภท string	xsd:string, xsd:normalizedString, xsd:token, xsd:language, xsd:NMTOKEN, xsd:Name, xsd:anyURI, xsd:NCName, xsd:hexBinary, xsd:base64Binary
ประเภท boolean	xsd:boolean
ประเภท numerical	xsd:decimal, xsd:float, xsd:double, xsd:integer, xsd:positiveInteger, xsd:negativeInteger, xsd:long, xsd:negativeInteger, xsd:nonNegativeInteger, xsd:int, xsd:short, xsd:byte, xsd:unsignedLong, xsd:unsignedInt, xsd:unsignedShort, xsd:unsignedByte
ประเภท time	xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay, xsd:gMonth.

(4) ข้อกำหนดของคุณสมบัติ

เพื่อจัดการความหลากหลายของข้อมูล ภาษา OWL ได้ตั้งข้อกำหนดการใช้งานคุณสมบัติและลักษณะพิเศษของการบรรยายคลาส โดยอธิบายคลาสในรูปแบบ anonymous class รวมถึงคลาสทุกคลาสที่เป็น Individual Class ได้ถูกกำหนดให้เป็นข้อจำกัดในการใช้งานคุณสมบัติ โดยโครงสร้างภาษา OWL แบ่งข้อกำหนดของคุณสมบัติเหล่านี้ ได้เป็น 2 ประเภทได้แก่ Value Constraints และ Cardinality Constraints ดังแสดงในตารางที่ 2.14

ตารางที่ 2.14 ชนิดข้อมูลที่อ้างอิงตาม Constraints

ประเภท	คำสั่งที่ใช้งาน
1. value constraints	owl:allValuesFrom, owl:someValuesFrom, owl:hasValue
2. cardinality constraints	owl:maxCardinality, owl:minCardinality, owl:cardinality

ตัวอย่าง การกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:allValuesFrom

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent" />
  <owl:allValuesFrom rdf:resource="#Human" />
</owl:Restriction>
```

ภาพที่ 2.10 การกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:allValuesFrom

จากภาพที่ 2.10 เป็นตัวอย่างการกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:allValuesFrom ซึ่งจากตัวอย่างเป็นการนิยามว่ามนุษย์ทุกคนจะต้องมีผู้ให้กำเนิด เป็นต้น ในการแทนค่าข้อกำหนดของคุณสมบัติ โดยคำสั่ง owl:allValuesFrom เป็นการจำกัดความต้องการสำหรับทุกๆ Instance ของคลาสให้มีคุณสมบัติที่เป็นลักษณะเฉพาะเท่านั้น ซึ่งค่าของคุณสมบัตินี้ทุกๆ สมาชิกภายในคลาสจะถูกกำหนดภายใต้คำสั่ง owl:allValuesFrom

ตัวอย่าง การกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:SomeValuesFrom

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent" />
  <owl:someValuesFrom rdf:resource="#Physician" />
</owl:Restriction>
```

ภาพที่ 2.11 การกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:someValuesFrom

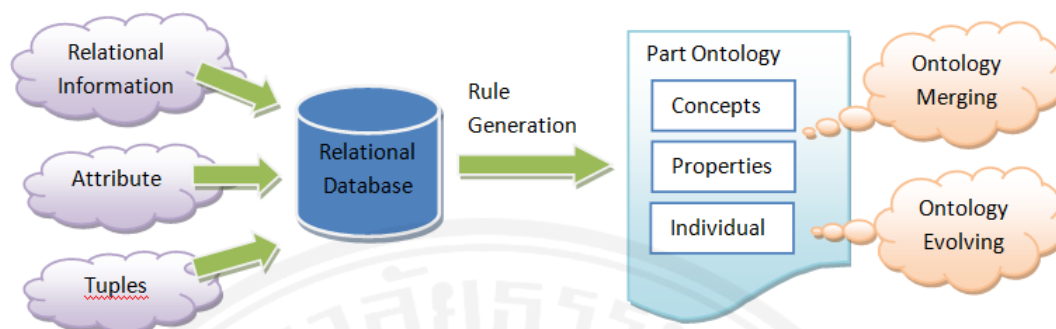
จากภาพที่ 2.11 เป็นตัวอย่างการกำหนดคุณสมบัติคลาสด้วยคำสั่ง owl:someValuesFrom ซึ่งจากตัวอย่างเป็นการนิยามว่า โดยทั่วไปผู้ให้กำเนิดอาจมีชีวิตรอยู่ทั้งสองคน หรืออาจมีชีวิตรอยู่เพียงบิดาหรือมารดา เป็นต้น ในการแทนค่าข้อกำหนดของคุณสมบัติ โดยคำสั่ง owl:someValuesFrom เป็นการกำหนดว่า โดยอย่างน้อย 1 ค่าของคุณสมบัติจะต้องเกี่ยวข้องกับ instance ของการบรรยายคลาส หรือค่าข้อมูลภายในขอบเขตของข้อมูล

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Rules and Implementation for Generating Ontology from Relational Database

Rules and Implementation for Generating Ontology from Relational Database (Yutao, Lihong, Fenglin & Hongming, 2012) เป็นงานวิจัยที่นำเสนอกฎในการแปลงจาก Relational Database เป็น Ontology และพัฒนาโปรแกรมสำหรับสร้าง ontology โดย Relational Database จะอยู่ในรูปแบบของฐานข้อมูล MySQL และมีรูปแบบโครงสร้างฐานข้อมูลเป็น Third Normal Form (3NF) ส่วน ontology จะอยู่ในรูปแบบของ OWL ซึ่งมีกระบวนการในการแปลง 2 กระบวนการดังนี้

1. Schema Transformation เป็นกระบวนการในการดึงข้อมูลโครงสร้างของ Relational Database แล้วทำการแปลงเป็น Concept และ Property ใน Ontology
2. Data Transformation เป็นกระบวนการในการดึงข้อมูลที่จัดเก็บอยู่ใน Relational Database แล้วแปลงข้อมูลทั้งหมดให้เป็น Instance ของ Ontology



ภาพที่ 2.12 กระบวนการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี. From “Rules and implementation for generating ontology from relational database,” by R. Yutao, J. Lihong, B. Fenglin, and C. Hongming, 2012, The Second International Conference on Cloud and Green Computing, p. 238.

จากภาพที่ 2.12 เป็นแสดงกระบวนการในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี โดยเริ่มจากการแบ่ง Relational Database ออกเป็น 3 ส่วน คือ Relational Information, Attribute และ Tuples จากนั้นทำการแปลงเป็นออนโทโลยี โดยใช้กฎในการแปลงที่ทางผู้วิจัยได้นำเสนอไว้ ซึ่งในส่วนของออนโทโลยีที่ได้ก็แบ่งออกเป็น 3 ส่วนเช่นกัน คือ Concepts, Properties และ Individual โดยกฎที่ใช้ในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี เป็นดังแสดงในตารางที่ 2.15

ตารางที่ 2.15 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ R. Yutao, J. Lihong, B. Fenglin, และ C. Hongming

Rule	Case	OWL Component
1	Relation มีอย่างน้อย 1 primary key และไม่มี attribute อื่นที่เป็น foreign key	Class
2	Binary Relation	Convert เป็น 2 Object Property ที่ผกผันกัน
3	Attribute	DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute

ตารางที่ 2.15 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ R. Yutao, J. Lihong, B. Fenglin, และ C. Hongming (ต่อ)

Rule	Case	OWL Component
4	Foreign Key	ObjectProperty <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Class ที่ตรงกับ Reference Relation ของ Attribute
5	Primary Key	DatatypeProperty <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute
6	ระหว่าง 2 Relation ที่ไม่ใช่ Binary Relation คือ T1 กับ T2 ถ้า a เป็น Primary Key ของ T1 และ b เป็น Primary Key กับ Foreign Key ของ T2 ซึ่งเป็น Attribute ที่ Reference ไปยัง T1	Convert Class T2 เป็น SubClassOf ของ Class T1
7	Attribute กำหนดเป็น Not Null	minCardinality มีค่าเท่ากับ 1
8	Unique Attribute	FunctionalProperty
9	Data	Individual

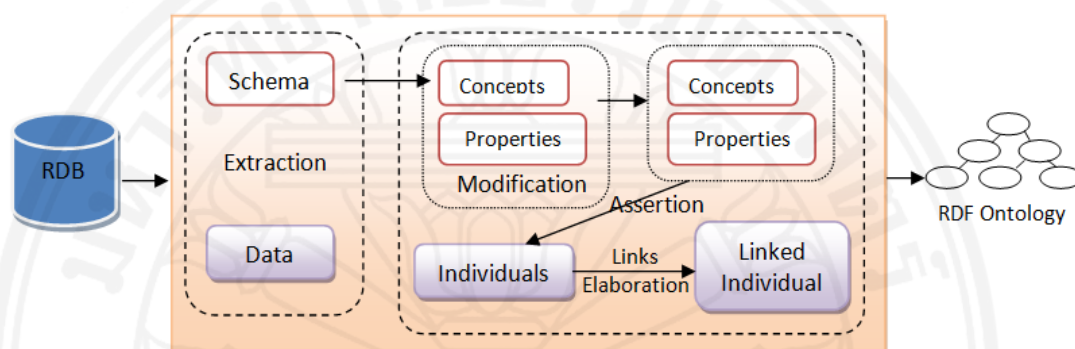
Note. From “Rules and implementation for generating ontology from relational database,” by R. Yutao, J. Lihong, B. Fenglin, and C. Hongming, 2012, The Second International Conference on Cloud and Green Computing, p. 237-244.

2.2.2 Generating of RDF graph from a relational database using

Jena API

Generating of RDF graph from a relational database using Jena API (Jamal & Mohamed, 2013) เป็นงานวิจัยที่นำเสนอวิธีการในการ Extraction Data จาก Relational Database ให้เป็นโครงสร้างข้อมูลแบบใหม่ที่อยู่ในรูปแบบของ RDF graph โดยใช้ Jena API เพื่อใช้ในการพัฒนา website ปกติ ให้เป็น Semantic Web ซึ่งมีขั้นตอนในการดำเนินงานดังนี้

1. ใช้ Jena API สร้างโครงสร้างของ Ontology จากส่วนประกอบของ Relational Database Schema
2. นำโครงสร้าง Ontology ที่ได้ไปทำการปรับโครงสร้างด้วยภาษา SPARQL
3. ทำการสร้าง Individual ของ Ontology จาก Record Data ของ Relational Database



ภาพที่ 2.13 อธิบายวิธีการในการแปลง. From “Generating of RDF graph from a relational database using Jena API,” by B. Jamal and B. Mohamed, 2013, International Journal of Engineering and Technology (IJET), 5(2), p. 1971.

จากภาพที่ 2.13 เป็นวิธีการในการแปลงจาก Relational Database เป็น RDF Ontology โดยมีวิธีการทำงานดังต่อไปนี้

1. ทำการ Extraction Schema ใน Relational Database แล้วทำการแปลงเป็น Concepts กับ Property ใน Ontology จากนั้นทำการปรับปรุง Concepts และ Property จนได้ Concepts และ Property ตามที่ต้องการ โดยกฎที่ใช้ในการแปลง เป็นดังตารางที่ 2.16
2. ทำการ Extraction Data ใน Relational Database แล้วทำการแปลงเป็น Individual ใน Ontology
3. ทำการเพิ่ม Individual ที่แปลงได้ใน Concepts และ Property ที่ทำการปรับปรุงแล้ว จากนั้นทำการ Link Elaboration Individual
4. นำ Concepts, Property และ Individual ที่ได้มาสร้างเป็น RDF Graph

ตารางที่ 2.16 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ B. Jamal และ B. Mohamed

Rule	Case	OWL Component
1	Relation	Class
2	Attribute	DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute
3	Foreign Key	ObjectProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Class ที่ตรงกับ Reference Relation ของ Attribute
4	Primary Key	1. DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute 2. InverseFunctionalProperty
5	Foreign Key ที่ไม่เป็น Primary Key	เพิ่มคุณสมบัติ FunctionalProperty
6	Relation ที่มีทุก Attribute เป็น Primary Key และ Foreign Key	Convert เป็น Object Property ที่ผกผันกันระหว่าง relation ของ attribute นั้น กับ relation ที่ attribute นั้น reference
7	Data	Individual

Note. From “Generating of RDF graph from a relational database using Jena API,” by B. Jamal and B. Mohamed, 2013, International Journal of Engineering and Technology (IJET), 5(2), p. 1970-1975.

2.2.3 Algorithms for Mapping RDB Schema to RDF for Facilitating Access to Deep Web

Algorithms for Mapping RDB Schema to RDF for Facilitating Access to Deep Web (Wondu, Farhi & Vassil, 2013) เป็นงานวิจัยที่นำเสนอ Algorithms ในการ Mapping Relational Database เป็น RDF ซึ่ง Relational Database จะอยู่ในรูปแบบของ Data Dictionary โดยกฎและวิธีการในการแปลงจาก Relational Database เป็น Ontology สามารถสรุปได้ดังตารางที่ 2.17

ตารางที่ 2.17 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ Y. M. Wondu, M. Farhi และ T. V. Vassil

Rule	Case	OWL Component
1	Relation	Class
2	Attribute	DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute
3	Foreign Key	ObjectProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Class ที่ตรงกับ Reference Relation ของ Attribute
4	Primary Key	1. maxCardinality มีค่าเท่ากับ 1 2. InverseFunctionalProperty
5	Attribute ที่เป็น Foreign Key และเป็นส่วนหนึ่งของ Primary Key	subClassOf
6	Unique Attribute	1. maxCardinality มีค่าเท่ากับ 1 2. InverseFunctionalProperty
7	Not Null Attribute	minCardinality มีค่าเท่ากับ 1
8	Check Constraints	hasValue

Note. From “Algorithms for Mapping RDB Schema to RDF for Facilitating Access to Deep Web,” by Y. M. Wondu, M. Farhi, and T. V. Vassil, 2013, The First International Conference on Building and Exploring Web Based Environments, p. 32-41.

2.2.4 Mapping Relational Database into OWL Ontology

Mapping Relational Database into OWL Ontology (Haiyun & Shufeng, 2014) เป็นงานวิจัยที่นำเสนอวิธีการสร้าง RDF graph จาก Relational Database เพื่อเพิ่มประสิทธิภาพในการพัฒนา Semantic Web โดยทำการพัฒนา Prototype ที่ใช้ในการสร้าง Ontology จาก Relational Database ซึ่ง Prototype นี้จะทำหน้าที่ Extracts Schema Metadata ของ Database แล้วทำการแปลงให้อยู่ในรูปแบบที่สะดวกต่อการ Migrate จากนั้นทำการสร้าง OWL Ontology และ Data ของ RDF โดยจะอยู่ในรูปแบบของเอกสาร OWL โดยกฎและวิธีการในการแปลงจาก Relational Database เป็น Ontology สามารถสรุปได้ดังตารางที่ 2.18

ตารางที่ 2.18 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ L. Haiyun และ Z. Shufeng

Rule	Case	OWL Component
1	Normal Relation ที่ไม่ใช่ Binary Relation	Class
2	Binary Relation	Convert เป็น 2 Object Property ที่ผกผันกัน
3	ระหว่าง 2 Relation ที่ไม่ใช่ Binary Relation คือ T1 กับ T2 ถ้า a เป็น Primary Key ของ T1 และ b เป็น Primary Key กับ Foreign Key ของ T2 ซึ่งเป็น Attribute ที่ Reference ไปยัง T1	Convert Class T2 เป็น SubclassOf ของ Class T1
4	Normal Attribute	DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute

ตารางที่ 2.18 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ L. Haiyun และ Z. Shufeng (ต่อ)

Rule	Case	OWL Component
5	Primary Key	1. maxCardinality มีค่าเท่ากับ 1 2. minCardinality มีค่าเท่ากับ 1
6	Unique Attribute	maxCardinality มีค่าเท่ากับ 1
7	Not Null Attribute	minCardinality มีค่าเท่ากับ 1
8	Data	Individual

Note. From “Mapping Relational Database into OWL Ontology,” by L. Haiyun and Z. Shufeng, 2014, International Journal of Engineering and Technology (IJET), 5(2), p. 4735-4740.

2.2.5 Automatic Mapping of Relational Database to OWL Antology

Automatic Mapping of Relational Database to OWL Antology (Larbi, Oussama & Mohamed, 2014) เป็นงานวิจัยที่วิเคราะห์และแสดงให้เห็นถึงภาพรวมของการ Migrate Relational Database เป็น Ontology รวมถึงจุดอ่อน ข้อจำกัด และข้อแตกต่างของงานวิจัยก่อนหน้ากับงานวิจัยนี้ เพื่อให้ Ontology ที่สร้างได้ครอบคลุมทั้งข้อจำกัดและรักษาข้อมูลให้มีความถูกต้องและสอดคล้องกับ Relational Database ซึ่งงานวิจัยนี้ได้นำเสนอกฎในการแปลง โดยจะเลือกจากกรณีทั้งหมดที่เป็นไปได้ในโครงสร้างของ Relational Database แล้วมาทำการแปลงเป็น Ontology ซึ่งวิธีการดำเนินงานจะแบ่งออกเป็น 3 ส่วนดังนี้

1. Mapping Relation เป็นส่วนของการแปลง Relation ต่างๆ
2. Mapping Attribute เป็นส่วนของการแปลง Attribute ต่างๆ
3. Mapping Constraint เป็นส่วนของการแปลงข้อกำหนดต่างๆ

โดยกฎและวิธีการในการแปลงจาก Relational Database เป็น Ontology สามารถสรุปได้ดังตารางที่ 2.19

ตารางที่ 2.19 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ A. Larbi,
E. H. Oussama และ B. Mohamed

Rule	Case	OWL Component
1	Normal Relation ที่ไม่ใช่ Binary Relation	Class
2	Binary Relation	Convert เป็น 2 Object Property ที่ผกผันกัน
3	ระหว่าง 2 Relation ที่ไม่ใช่ Binary Relation คือ T1 กับ T2 ถ้า a เป็น Primary Key ของ T1 และ b เป็น Primary Key กับ Foreign Key ของ T2 ซึ่งเป็น Attribute ที่ Reference ไปยัง T1	Convert Class T2 เป็น SubclassOf ของ Class T1
4	Normal Attribute	DatatypeProperty <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute
5	Primary Key	1. DatatypeProperty <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute 2. InverseFunctionalProperty
6	Foreign Key	1. ObjectProperty <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Class ที่ตรงกับ Reference Relation ของ Attribute 2. FunctionalProperty
7	Unique Attribute	maxCardinality มีค่าเท่ากับ 1
8	Not Null Attribute	minCardinality มีค่าเท่ากับ 1

ตารางที่ 2.19 กฎในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีของ A. Larbi, E. H. Oussama และ B. Mohamed (ต่อ)

Rule	Case	OWL Component
9	Unique and Not Null Attribute	1. maxCardinality มีค่าเท่ากับ 1 2. minCardinality มีค่าเท่ากับ 1
10	Data	Individual

Note. From “Automatic Mapping of Relational Databases to OWL Antology,” by A. Larbi, E. H. Oussama and B. Mohamed, 2014, International Journal of Engineering Research & Technology (IJERT), 3(4), p. 1988-1994.



บทที่ 3

วิธีการดำเนินงานวิจัย

งานวิจัยนี้นำเสนอวิธีการในการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์และพัฒนา ระบบเพื่อใช้ในการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ และปรับแต่งโครงสร้าง ของออนโทโลยีได้ โดยมุ่งเน้นไปที่การขยายข้อ กำหนด (Constraint) ของฐานข้อมูลเชิงสัมพันธ์ เพื่อให้สามารถสร้างออนโทโลยีได้อย่างถูกต้องและมีความสมบูรณ์ ตามโครงสร้างและข้อ กำหนดมาก ยิ่งขึ้น โดยมีขั้นตอนในการดำเนินงาน ดังต่อไปนี้

3.1 ขอบเขตการทดลอง

การทดลองสำหรับงานวิจัยนี้ จะใช้ฐานข้อมูลเชิงสัมพันธ์ที่ผ่านการ Normalization เป็นระดับที่ 3 (3NF) และแปลงเป็นออนโทโลยีที่อยู่ในรูปแบบของภาษา OWL

3.2 เครื่องมือที่ใช้ในการพัฒนา

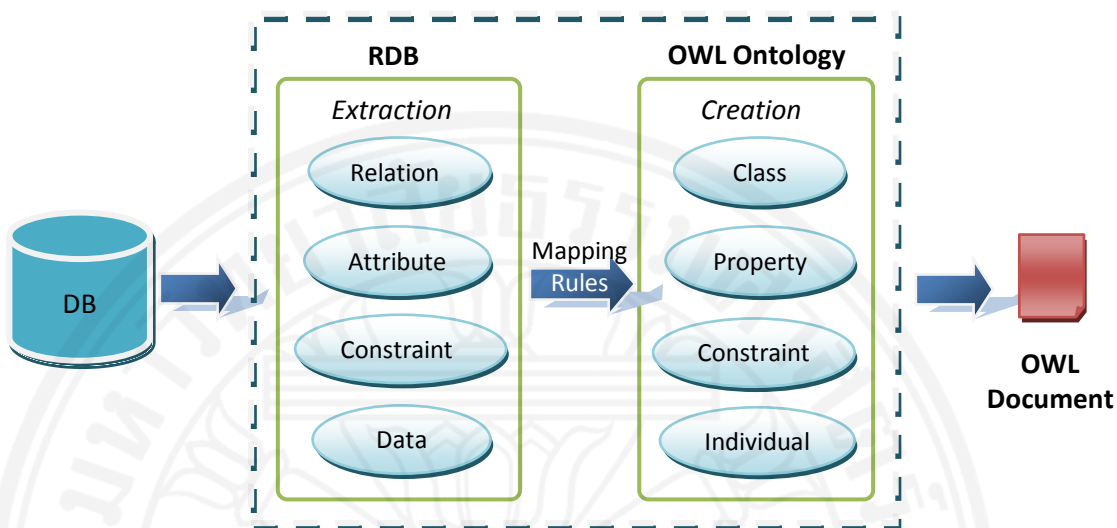
เครื่องมือและซอฟต์แวร์คอมพิวเตอร์ที่ใช้ในงานวิจัยฉบับนี้ มีรายละเอียดดังแสดงใน

ตารางที่ 3.1

ตารางที่ 3.1 เครื่องมือที่ใช้สำหรับงานวิจัย

Hardware	- Intel Core i7- 2630QM CPU @ 2.00 GHz RAM 4 GB
Operating System	Microsoft Windows 7 Ultimate
Software	- Visual Studio .Net 2010 - MySQL - Notepad - EditPlus 3 - Protégé

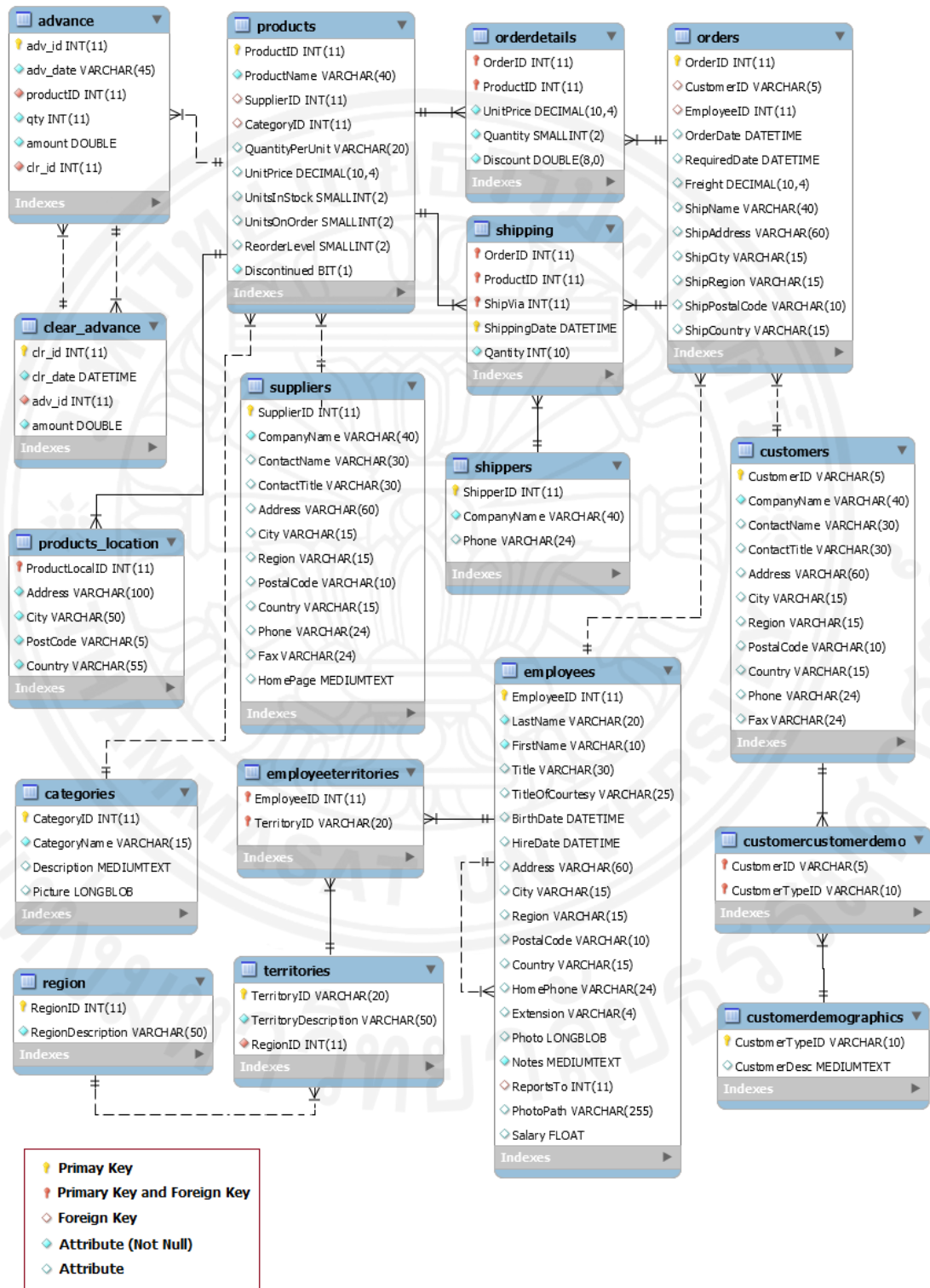
3.3 สถาปัตยกรรมของระบบ



รูปที่ 3.1 ภาพรวมการทำงานของระบบ

จากภาพที่ 3.1 แสดงภาพรวมการทำงานของระบบ โดยมีขั้นตอนในการดำเนินงานคือ เริ่มจากการเชื่อมต่อฐานข้อมูล จากนั้นทำการดึงโครงสร้าง (Schema) ข้อกำหนด (Constraint) และข้อมูล (Data) ต่างๆ ที่ จัดเก็บในฐานข้อมูลเชิงสัมพันธ์ โดยใช้ MySQL DBMS จากนั้นทำการ Mapping โครงสร้างและข้อกำหนดต่างๆ ของฐานข้อมูลเชิงสัมพันธ์กับโครงสร้างและข้อกำหนดของออนโทโลยี และทำการสร้างออนโทโลยี โดยใช้ OWL API โดยออนโทโลยีที่แปลงได้ จะอยู่ในรูปแบบของภาษา OWL DL

3.4 กระบวนการในการทำงานของระบบ



ภาพที่ 3.2 ตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์

จากภาพที่ 3.2 เป็นตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่ทางผู้วิจัยจะใช้เป็นตัวอย่าง สำหรับแสดงกระบวนการในการสร้างอนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์ โดยมีวิธีในการดำเนินงาน ดังต่อไปนี้

3.4.1 การแตกโครงสร้างของฐานข้อมูลเชิงสัมพันธ์

เป็นกระบวนการในการแตกโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ออกมา โดยจะแตกออกเป็น Relation ต่างๆ ซึ่งแต่ละ Relation จะประกอบด้วย Column, Datatype, Null, Unique, Key, Table Reference และ Default Value ซึ่งตัวอย่าง Relation ที่แตกได้ สามารถแสดงได้ดังภาพที่ 3.3

Relation :: orders						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
OrderID	int	NO		PK		
CustomerID	varchar	YES		FK	customers	
EmployeeID	int	YES		FK	employees	
OrderDate	datetime	YES				
RequiredDate	datetime	YES				
Freight	decimal	YES				0.0000
ShipName	varchar	YES				
ShipAddress	varchar	YES				
ShipCity	varchar	YES				
ShipRegion	varchar	YES				
ShipPostalCode	varchar	YES				
ShipCountry	varchar	YES				

ภาพที่ 3.3 โครงสร้างของ Relation order

3.4.2 การเปรียบเทียบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์กับโครงสร้างของอนโทโลยี (Mapping Rule)

เป็นกระบวนการในการเปรียบเทียบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์กับโครงสร้างของอนโทโลยี โดยนำแต่ละ Relation ของฐานข้อมูลเชิงสัมพันธ์ที่แตกได้มาทำการเปรียบเทียบโครงสร้างกับอนโทโลยี และสร้างเป็นอนโทโลยี โดยมีวิธีในการดำเนินงานเป็นดังต่อไปนี้

3.4.2.1 ตรวจสอบคุณสมบัติของ Relation

เป็นกระบวนการในการตรวจสอบหาคุณสมบัติ Binary Relation และ Inheritance Relation ของแต่ละ Relation ซึ่งสามารถแบ่งออกเป็นกรณีต่างๆ ได้ดังนี้

(1) Binary Relation

กรณี Relation ที่ทำการตรวจสอบมีคุณสมบัติเป็น Binary Relation เป็นจะทำการแปลงเป็น 2 Object Property ที่ผกผันกัน ในออนโทโลยี ยกตัวอย่างเช่น Relation `customercustomer_demo` ดังแสดงในภาพที่ 3.4 ซึ่งเป็น Relation ที่อยู่ระหว่าง 2 Relation `customers` และ `customerdemographics` โดยสามารถแปลงเป็นออนโทโลยี ได้ดังภาพที่ 3.5

Relation :: customercustomerdemo						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
CustomerID	varchar	NO		PK,FK	customers	
CustomerTypeID	varchar	NO		PK,FK	customerdemographics	

ภาพที่ 3.4 ตัวอย่าง Relation `customercustomer_demo`

```

<owl:Class rdf:about="customers"></owl:Class>
<owl:ObjectProperty rdf:about="#customers_customerdemographics">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="customers"/>
  <rdfs:range rdf:resource="customerdemographics"/>
</owl:ObjectProperty>

<owl:Class rdf:about="customerdemographics"></owl:Class>
<owl:ObjectProperty rdf:about="#customerdemographics_customers">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="customerdemographics"/>
  <rdfs:range rdf:resource="customers"/>
  <owl:inverseOf rdf:resource="#customers_customerdemographics"/>
</owl:ObjectProperty>

```

ภาพที่ 3.5 ตัวอย่างการแปลง Binary Relation เป็นออนโทโลยี

(2) Inheritance Relation

กรณี Relation ที่ทำการตรวจสอบมีคุณสมบัติเป็น Inheritance Relation และไม่มีคุณสมบัติเป็น Binary Relation จะทำการแปลงเป็น Relation นั้นเป็น Class โดยชื่อ Relation จะเป็นชื่อของ Class และมีคุณสมบัติเป็น SubClassOf ในออนโทโลยี โดย SubClassOf คือ Class ที่เป็น Reference Relation ของ Relation นั้น ตัวอย่างเช่น Relation products_location ซึ่งมี Reference Relation เป็น Relation products ดังภาพที่ 3.6 ดังนั้นจะทำการแปลง Relation products_location เป็น Class products_location และมี SubClassOf เป็น Class products ดังแสดงตัวอย่างการแปลงในภาพที่ 3.7

Relation :: products_location						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
ProductLocalID	int	NO		PK,FK	products	
Address	varchar	NO				
City	varchar	NO				
PostCode	varchar	NO				
Country	varchar	NO				

ภาพที่ 3.6 ตัวอย่าง Relation products_location

```
<owl:Class rdf:about="products_location">
<rdfs:subClassOf rdf:resource="products"/>
</owl:Class>
```

ภาพที่ 3.7 ตัวอย่างการแปลง Inheritance Relation เป็นออนโทโลยี

(3) Normal Relation

กรณี Relation ที่ทำการตรวจสอบไม่มีคุณสมบัติเป็น Binary Relation และ Inheritance Relation จะถือว่า Relation นั้นเป็น Normal Relation ซึ่งจะทำการแปลงเป็น Relation นั้นเป็น Class โดยชื่อ Relation จะเป็นชื่อของ Class ตัวอย่างเช่น Relation categories ดังภาพที่ 3.4 ซึ่งสามารถแสดงตัวอย่างการแปลงดังภาพที่ 3.8

```
<owl:Class rdf:about="categories"></owl:Class>
```

ภาพที่ 3.8 ตัวอย่างการแปลง Normal Relation เป็นออนโทโลยี

3.4.2.2 ตรวจสอบคุณสมบัติ Normal Attribute

เป็นกระบวนการในการตรวจสอบแต่ละ Attribute ใน Relation ถ้า Attribute ใดใน Relation ไม่มีคุณสมบัติเป็น Primary Key หรือ Foreign Key จะถือว่า Attribute นั้นเป็น Normal Attribute และทำการตรวจสอบคุณสมบัติเพิ่มเติมของ Attribute ดังนี้

(1) Null

กรณีที่ Normal Attribute มีการระบุคุณสมบัติเพิ่มเติมเป็น Null จะทำการแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมคือ กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute และ Range เป็น Datatype ใน XSD Datatype ที่ตรงกับ Attribute ตัวอย่างเช่น Relation categories ซึ่งมี Normal Attribute ที่มีการกำหนดคุณสมบัติดังกล่าว คือ Description และ Picture ดังภาพที่ 3.4 และสามารถแปลงเป็นออนโทโลยีได้ดังภาพที่ 3.9

```
<owl:Class rdf:about="categories"></owl:Class>
<owl:DatatypeProperty rdf:about="#categories.Description">
  <rdfs:domain rdf:resource="categories"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#categories.Picture">
  <rdfs:domain rdf:resource="categories"/>
  <rdfs:range rdf:resource="&xsd:byte"/>
</owl:DatatypeProperty>
```

ภาพที่ 3.9 ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Null เป็นออนโทโลยี

(2) Not Null

กรณีที่มี Normal Attribute มีการระบุคุณสมบัติเพิ่มเติมเป็น Not Null และไม่เป็น Unique จะทำการแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมคือ กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute, Range เป็น Datatype ใน XSD Datatype ที่ตรงกับ Attribute และกำหนด Restricted เป็น minCardinality = 1 ตัวอย่างเช่น Relation region ซึ่งมี Normal Attribute ที่มีการกำหนดคุณสมบัติดังกล่าว คือ RegionDescription ดังภาพที่ 3.10 และสามารถแปลงเป็นออนโทโลยีได้ ดังภาพที่ 3.11

Relation :: region						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
RegionID	int	NO		PK		
RegionDescription	varchar	NO				

ภาพที่ 3.10 ตัวอย่าง Relation region

```

<owl:Class rdf:about="region">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#region. RegionDescription"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</owl:Class>
<owl:DatatypeProperty rdf:about="#region.RegionDescription">
  <rdfs:domain rdf:resource="region"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

ภาพที่ 3.11 ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Not Null เป็นออนโทโลยี

(3) Unique

กรณีที่มี Normal Attribute มีการระบุคุณสมบัติเพิ่มเติมเป็น Unique และไม่เป็น Not Null ตัวอย่างเช่น Relation customerdemographics ซึ่งมี Normal Attribute ที่มีการกำหนดคุณสมบัติดังกล่าว คือ CustomerDesc ดังแสดงในภาพที่ 3.12 สามารถทำการแปลงเป็นออนโทโลยีได้ โดยมีรายละเอียดดังนี้

Relation :: customerdemographics						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
CustomerTypeID	varchar	NO		PK		
CustomerDesc	mediumtext	YES	YES			

ภาพที่ 3.12 ตัวอย่าง Relation customerdemographics

1. Normal Attribute นั้น จะแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Datatype ใน XSD Datatype

ที่ตรงกับ Attribute

- กำหนด Restricted เป็น maxCardinality = 1

2. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Characteristic เป็น InverseFunctionalProperty
- กำหนด Restricted เป็น maxCardinality = 1

ซึ่งสามารถแปลง Normal Attribute ที่กำหนดคุณสมบัติเป็น Unique เป็นออนโทโลยีได้ดังภาพที่ 3.13

```

<owl:Class rdf:about="customerdemographics">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#customerdemographics.CustomerDesc"/>
    <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:maxCardinality>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#customerdemographics.CustomerDesc_Ref"/>
  <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1
</owl:maxCardinality>
</owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#customerdemographics.CustomerDesc_Ref">
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="customerdemographics"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#customerdemographics.CustomerDesc">
  <rdfs:domain rdf:resource="customerdemographics"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

ภาพที่ 3.13 ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Unique เป็นอ็อนโทโลยี

(4) Not Null และ Unique

กรณี Normal Attribute มีการระบุคุณสมบัติเพิ่มเติมเป็น Not Null และ Unique ตัวอย่างเช่น Relation categories ซึ่งมี Normal Attribute ที่มีการกำหนดคุณสมบัติดังกล่าว คือ CategoryNO ดังแสดงในภาพที่ 3.14 สามารถทำการแปลงเป็นอ็อนโทโลยีได้ โดยมีรายละเอียดดังนี้

Relation :: categories						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
CategoryID	int	NO		PK		
CategoryName	varchar	NO				
Description	mediumtext	YES				
Picture	longblob	YES				
CategoryNO	varchar	NO	YES			

ภาพที่ 3.12 ตัวอย่าง Relation categories

1. Normal Attribute นั้น จะแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Datatype ใน XSD Datatype

ที่ตรงกับ Attribute

- กำหนด Restricted เป็น Cardinality = 1

2. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Characteristic เป็น InverseFunctionalProperty
- กำหนด Restricted เป็น Cardinality = 1

ซึ่งสามารถแปลง Normal Attribute ที่กำหนดคุณสมบัติเป็น Not Null และ Unique เป็นออนโทโลยีได้ดังภาพที่ 3.15

```

<owl:Class rdf:about="categories">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#categories.CategoryNO"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#categories.CategoryNO_Ref "/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#categories.CategoryNO_Ref">
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="categories"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#categories.CategoryNO">
  <rdfs:domain rdf:resource="categories"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

ภาพที่ 3.15 ตัวอย่างการแปลง Normal Attribute ที่มีคุณสมบัติ Not Null และ Unique เป็นออนโทโลยี

3.4.2.3 ตรวจสอบคุณสมบัติของ Attribute ที่เป็นคีย์ต่างๆ

เป็นกระบวนการในการตรวจสอบคุณสมบัติของ Attribute ที่เป็นคีย์ต่างๆ ใน Relation ที่ไม่มีคุณสมบัติเป็น Binary Relation โดยสามารถแบ่งลักษณะของคีย์ที่ใน Relation ได้ดังต่อไปนี้

(1) Primary Key เพียงหนึ่งใน Relation

กรณีที่ Attribute ใน Relation มีคุณสมบัติเป็น Primary Key เพียงหนึ่งเดียวและไม่เป็น Foreign Key ตัวอย่างเช่น Relation categories ซึ่งมี Attribute ที่มีคุณสมบัติดังกล่าว คือ CategoryID ดังแสดงในภาพที่ 3.14 สามารถทำการแปลงเป็นออนโทโลยี โดยมีรายละเอียดดังนี้

1. Normal Attribute นั้น จะแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Datatype ใน XSD Datatype

ที่ตรงกับ Attribute

- Restricted เป็น Cardinality = 1

2. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือชื่อ Relation ของ Attribute ส่วน AttributeName คือชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Characteristic เป็น InverseFunctionalProperty
- กำหนด Restricted เป็น Cardinality = 1

ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็น Primary Key เพียงหนึ่งเดียวใน Relation เป็นออนโทโลยีได้ดังภาพที่ 3.16

```
<owl:Class rdf:about="categories">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#categories.CategoryID"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#categories.CategoryID_Ref"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</owl:Class>
```

ภาพที่ 3.16 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Primary Key เพียงหนึ่งเดียวใน Relation เป็นออนโทโลยี


```

<owl:ObjectProperty rdf:about="#categories.CategoryID_Ref">
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="categories"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#categories.CategoryID">
  <rdfs:domain rdf:resource="categories"/>
  <rdfs:range rdf:resource="xsd:integer"/>
</owl:DatatypeProperty>

```

ภาพที่ 3.16 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Primary Key เพียงหนึ่งเดียวใน Relation เป็นออนโทโลยี (ต่อ)

(2) Primary Key และ Foreign Key เพียงหนึ่งเดียวใน Relation

กรณีที่ Attribute ใน Relation มีคุณสมบัติเป็น Primary Key และ Foreign Key เพียงหนึ่งเดียว ตัวอย่างเช่น Relation products_location ซึ่งมี Attribute ที่มีคุณสมบัติดังกล่าว คือ ProductLocalID ดังแสดงในภาพที่ 3.6 สามารถทำการแปลงเป็นออนโทโลยี โดยมีรายละเอียดดังนี้

1. Normal Attribute นั้น จะแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Datatype ใน XSD Datatype

ที่ตรงกับ Attribute

- กำหนด Restricted เป็น Cardinality = 1

2. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute

- กำหนด Range เป็น Class ที่ตรงกับ Reference Relation ของ Attribute และกำหนดคุณสมบัติเป็น allValuesFrom

- กำหนด Characteristic เป็น InverseFunctionalProperty และ FunctionalProperty

- กำหนด Restricted เป็น Cardinality = 1 ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็น Primary Key และ Foreign Key เพียงหนึ่งเดียวใน Relation เป็นออนโทโลยีได้ดังภาพที่ 3.17

```

<owl:Class rdf:about="products_location">
  <rdfs:subClassOf rdf:resource="products"/>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#products_location.ProductLocalID"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#products_location.ProductLocalID_Ref"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#products_location.ProductLocalID_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:range rdf:resource="products"/>
  <rdfs:domain rdf:resource="products_location"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#products_location.ProductLocalID">
  <rdfs:domain rdf:resource="products_location"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

ภาพที่ 3.17 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Primary Key และ Foreign Key เพียงหนึ่งเดียวใน Relation เป็นออนโทโลยี

(3) Foreign Key ใน Relation

กรณีที่มี Attribute ใน Relation มีคุณสมบัติเป็น Foreign Key และไม่
เป็น Primary Key ตัวอย่างเช่น Relation products ซึ่งมี Attribute ที่มีคุณสมบัติดังกล่าว คือ
SupplierID และ CategoryID ดังแสดงในภาพที่ 3.18 สามารถทำการแปลงเป็นออนโทโลยี
โดยมีรายละเอียดดังนี้

Relation :: products						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
ProductID	int	NO		PK		
ProductName	varchar	NO				
SupplierID	int	YES		FK	suppliers	
CategoryID	int	YES		FK	categories	
QuantityPerUnit	varchar	YES				
UnitPrice	decimal	YES				0.0000
UnitsInStock	smallint	YES				0
UnitsOnOrder	smallint	YES				0
ReorderLevel	smallint	YES				0
Discontinued	bit	NO				b'0'

ภาพที่ 3.18 ตัวอย่าง Relation products

1. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ
ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือ
ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนด
คุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Class ที่ตรงกับ Reference Relation ของ
Attribute และกำหนดเป็นคุณสมบัติ allValuesFrom ในกรณีที่ Attribute นั้นระบุเป็น Not Null
แต่ในกรณีที่ Attribute นั้นระบุเป็น Null จะกำหนดคุณสมบัติเป็น someValueFrom
- กำหนด Characteristic เป็น FunctionalProperty
- กำหนด Restricted เป็น minCardinality = 1

ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็น Foreign Key และไม่ใช่ Primary Key ใน Relation เป็นอ็อนโทโลยีได้ดังภาพที่ 3.19

```

<owl:Class rdf:about="products">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#products.SupplierID_Ref"/>
    <owl:someValuesFrom rdf:resource="SomeValuesFrom"/>
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:minCardinality>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#products.CategoryID_Ref"/>
  <owl:someValuesFrom rdf:resource="SomeValuesFrom"/>
  <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
</owl:minCardinality>
</owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#products.SupplierID_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="products"/>
  <rdfs:range rdf:resource="suppliers"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#products.CategoryID_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="categories"/>
  <rdfs:domain rdf:resource="products"/>
</owl:ObjectProperty>

```

ภาพที่ 3.19 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key และไม่ใช่ Primary Key ใน Relation เป็นอ็อนโทโลยี

(4) Foreign Key แบบ Inverse Reference ใน Relation

กรณีที่มี Attribute ใน Relation มีคุณสมบัติเป็น Foreign Key ซึ่ง Reference ไปยังอีก Relation หนึ่งและมี Attribute ของ Relation นั้นก็ Reference มายัง Attribute ดังกล่าว หรือลักษณะความสัมพันธ์แบบ many - to - many ตัวอย่างเช่น Relation advance และ Relation clear_advance ซึ่งมี Attribute ที่เป็น Foreign Key และทำการ Reference ถึงกันและกัน คือ clr_id ใน Relation advance และ adv_id ใน Relation clear_advance ดังแสดงในภาพที่ 3.20 และ 3.21 ตามลำดับ สามารถทำการแปลงเป็นอโนโทโลยี โดยมีรายละเอียดดังนี้

Relation :: advance						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
adv_id	int	NO		PK		
adv_date	varchar	NO				
productID	int	NO		FK	products	
qty	int	NO				
amount	double	NO				
clr_id	int	NO		FK	clear_advance	

ภาพที่ 3.20 ตัวอย่าง Relation advance

Relation :: clear_advance						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
clr_id	int	NO		PK		
clr_date	datetime	NO				
adv_id	int	NO		FK	advance	
amount	double	NO				

ภาพที่ 3.21 ตัวอย่าง Relation clear_advance

1. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนด คุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Class ที่ตรงกับ Reference Relation ของ Attribute และกำหนดเป็นคุณสมบัติ allValuesFrom ในกรณีที่ Attribute นั้นระบุเป็น Not Null แต่ในกรณีที่ Attribute นั้นระบุเป็น Null จะกำหนดคุณสมบัติเป็น someValueFrom

- กำหนด Characteristic เป็น FunctionalProperty
- กำหนด Restricted เป็น minCardinality = 1
- กำหนดคุณสมบัติ InverseOf

ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็น Foreign Key แบบ Inverse Reference ใน Relation เป็นออนโทโลยีได้ดังภาพที่ 3.22

```

<owl:Class rdf:about="advance">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#advance.clr_id_Ref"/>
    <owl:allValuesFrom rdf:resource="AllValuesFrom"/>
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:minCardinality>
</owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#advance.clr_id_Ref">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="advance"/>
  <rdfs:range rdf:resource="clear_advance"/>
  <owl:inverseOf rdf:resource="#clear_advance.adv_id_Ref"/>
</owl:ObjectProperty>

```

ภาพที่ 3.22 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key แบบ Inverse Reference ใน Relation เป็นออนโทโลยี

```

<owl:Class rdf:about="clear_advance">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#clear_advance.adv_id_Ref"/>
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:minCardinality>
    <owl:allValuesFrom rdf:resource="AllValuesFrom"/>
  </owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#clear_advance.adv_id_Ref">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="advance"/>
  <rdfs:domain rdf:resource="clear_advance"/>
  <owl:inverseOf rdf:resource="#advance.clr_id_Ref"/>
</owl:ObjectProperty>

```

ภาพที่ 3.22 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key แบบ Inverse Reference ใน Relation เป็นออนโทโลยี (ต่อ)

(5) Foreign Key และ Reference มายัง Relation ตนเอง

กรณีที่ Attribute ใน Relation มีคุณสมบัติเป็น Foreign Key และ Reference มายัง Relation ตนเอง และไม่เป็น Primary Key ตัวอย่างเช่น Relation employees ซึ่งมี Attribute ที่มีคุณสมบัติดังกล่าว คือ ReportsTo ดังแสดงในภาพที่ 3.23 สามารถทำการแปลงเป็นออนโทโลยี โดยมีรายละเอียดดังนี้

Relation :: employees						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
EmployeeID	int	NO		PK		
LastName	varchar	NO				
FirstName	varchar	NO				
Title	varchar	YES				
TitleOfCourte	varchar	YES				
BirthDate	datetime	YES				
HireDate	datetime	YES				
Address	varchar	YES				
City	varchar	YES				
Region	varchar	YES				
PostalCode	varchar	YES				
Country	varchar	YES				
HomePhone	varchar	YES				
Extension	varchar	YES				
Photo	longblob	YES				
Notes	mediumtext	NO				
ReportsTo	int	YES		FK	employees	
PhotoPath	varchar	YES				
Salary	float	YES				

ภาพที่ 3.23 ตัวอย่าง Relation employees

1. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Class ที่ตรงกับ Reference Relation ของ

Attribute และกำหนดเป็นคุณสมบัติ allValuesFrom ในกรณีที่ Attribute นั้นระบุเป็น Not Null แต่ในกรณีที่ Attribute นั้นระบุเป็น Null จะกำหนดคุณสมบัติเป็น someValueFrom

- กำหนด Characteristic เป็น FunctionalProperty และ TransitiveProperty

- กำหนด Restricted เป็น minCardinality = 1 ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็น Foreign Key แบบ Inverse Reference ใน Relation เป็นออนโทโลยีได้ดังภาพที่ 3.24

```

<owl:Class rdf:about="employees">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#employees.ReportsTo_Ref"/>
    <owl:someValuesFrom rdf:resource="SomeValuesFrom"/>
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#employees.ReportsTo_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdf:type rdf:resource="owl#TransitiveProperty"/>
  <rdfs:range rdf:resource="employees"/>
  <rdfs:domain rdf:resource="employees"/>
</owl:ObjectProperty>

```

ภาพที่ 3.24 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติ Foreign Key และ Reference มายัง Relation ตนเองใน Relation เป็นออนโทโลยี

(6) Primary Key และไม่ใช่ Foreign Key (Composite Key)

กรณีที่ Attribute ใน Relation มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และไม่เป็น Foreign Key ตัวอย่างเช่น Relation shipping ซึ่งมี Attribute ที่มีคุณสมบัติดังกล่าว คือ ShippingDate ดังแสดงในภาพที่ 3.25 สามารถทำการแปลงเป็นออนโทโลยี โดยมีรายละเอียดดังนี้

Relation :: shipping						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
OrderID	int	NO		PK,FK	orders	
ProductID	int	NO		PK,FK	products	
ShipVia	int	NO		PK,FK	shippers	
ShippingDate	datetime	NO		PK		
Quantity	int	NO				

ภาพที่ 3.25 ตัวอย่าง Relation shipping

1. Normal Attribute นั้น จะแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Datatype ใน XSD Datatype

ที่ตรงกับ Attribute

- กำหนด Restricted เป็น minCardinality = 1

2. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Characteristic เป็น InverseFunctionalProperty
- กำหนด Restricted เป็น minCardinality = 1

ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และไม่เป็น Foreign Key ใน Relation เป็นออนโทโลยีได้ดังภาพที่ 3.26

```

<owl:Class rdf:about="shipping">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#shipping.ShippingDate_Ref"
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:minCardinality>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#shipping.ShippingDate"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger ">1</owl:cardinality>
  </owl:Restriction>
</owl:Class>
<owl:DatatypeProperty rdf:about="#shipping.ShippingDate">
  <rdfs:domain rdf:resource="shipping"/>
  <rdfs:range rdf:resource="&xsd;dateTime"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:about="#shipping.ShippingDate_Ref">
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="shipping"/>
</owl:ObjectProperty>

```

ภาพที่ 3.26 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และไม่เป็น Foreign Key ใน Relation เป็นอ็อนโทโลยี

(7) Primary Key และเป็น Foreign Key (Composite Key)

กรณีที่ Attribute ใน Relation มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และเป็น Foreign Key ตัวอย่างเช่น Relation shipping ซึ่งมี Attribute ที่มีคุณสมบัติดังกล่าว คือ OrderID, ProductID และ ShipVia ดังแสดงในภาพที่ 3.25 สามารถทำการแปลงเป็นอ็อนโทโลยี โดยมีรายละเอียดดังนี้

1. Normal Attribute นั้น จะแปลงเป็น DatatypeProperty โดยชื่อของ DatatypeProperty จะอยู่ในรูปแบบ #RelationName.AttributeName ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ DatatypeProperty เพิ่มเติมดังต่อไปนี้

- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Datatype ใน XSD Datatype

ที่ตรงกับ Attribute

- กำหนด Restricted เป็น minCardinality = 1
- กำหนด Domain เป็น Class ที่ตรงกับ Relation ของ Attribute
- กำหนด Range เป็น Class ที่ตรงกับ Reference Relation ของ

Attribute และกำหนดเป็นคุณสมบัติ allValuesFrom ในกรณีที่ Attribute นั้นระบุเป็น Not Null แต่ในกรณีที่ Attribute นั้นระบุเป็น Null จะกำหนดคุณสมบัติเป็น someValueFrom

● กำหนด Characteristic เป็น InverseFunctionalProperty และ FunctionalProperty

- กำหนด Restricted เป็น minCardinality = 1

2. Normal Attribute นั้น จะแปลงเป็น ObjectProperty โดยชื่อของ ObjectProperty จะอยู่ในรูปแบบ #RelationName.AttributeName_Ref ซึ่ง RelationName คือ ชื่อ Relation ของ Attribute ส่วน AttributeName คือ ชื่อของ Attribute นอกจากนี้ยังกำหนดคุณสมบัติของ ObjectProperty เพิ่มเติมดังต่อไปนี้ ซึ่งสามารถแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และไม่เป็น Foreign Key ใน Relation เป็นออนโทโลยีได้ดังภาพที่ 3.27

```
<owl:Class rdf:about="shipping">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#shipping.OrderID_Ref"/>
    <owl:allValuesFrom rdf:resource="AllValuesFrom"/>
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:minCardinality>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#shipping.ProductID_Ref"/>
  <owl:allValuesFrom rdf:resource="AllValuesFrom"/>
</owl:Restriction>
```

ภาพที่ 3.27 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และเป็น Foreign Key ใน Relation เป็นออนโทโลยี

```

    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:minCardinality>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#shipping.ShipVia_Ref"/>
  <owl:allValuesFrom rdf:resource="AllValuesFrom"/>
  <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
  </owl:minCardinality>
</owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:about="#shipping.OrderID_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:range rdf:resource="orders"/>
  <rdfs:domain rdf:resource="shipping"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#shipping.ProductID_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:range rdf:resource="products"/>
  <rdfs:domain rdf:resource="shipping"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#shipping.ShipVia_Ref">
  <rdf:type rdf:resource="owl#FunctionalProperty"/>
  <rdf:type rdf:resource="owl#InverseFunctionalProperty"/>
  <rdfs:range rdf:resource="shippers"/>
  <rdfs:domain rdf:resource="shipping"/>
</owl:ObjectProperty>

```

ภาพที่ 3.27 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และเป็น Foreign Key ใน Relation เป็นออนโทโลยี (ต่อ)

```

<owl:DatatypeProperty rdf:about="#shipping.OrderID">
  <rdfs:domain rdf:resource="shipping"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#shipping.ProductID">
  <rdfs:domain rdf:resource="shipping"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#shipping.ShipVia">
  <rdfs:domain rdf:resource="shipping"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

ภาพที่ 3.27 ตัวอย่างการแปลง Attribute ที่มีคุณสมบัติเป็นส่วนหนึ่งของ Primary Key และเป็น Foreign Key ใน Relation เป็นออนโทโลยี (ต่อ)

3.4.2.4 ตรวจสอบค่าเริ่มต้นของ Attribute ใน Relation

เป็นกระบวนการในการตรวจสอบค่าเริ่มต้น (Default Value) ของแต่ละ Attribute ใน Relation ถ้า Attribute ใดมีค่า Default Value จะแปลงเป็น hasValue ในออนโทโลยี ซึ่งจะมีค่าเท่ากับค่า Default Value ตัวอย่างเช่น Relation orderdetails ซึ่งมี Attribute ที่มีค่า Default Value คือ UnitPrice, Quantity และ Discount ดังแสดงในภาพที่ 3.28 โดยสามารถแปลงเป็นออนโทโลยีได้ดังภาพที่ 3.29

Relation :: orderdetails						
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value
OrderID	int	NO		PK,FK	orders	
ProductID	int	NO		PK,FK	products	
UnitPrice	decimal	NO				0.0000
Quantity	smallint	NO				1
Discount	double	NO				0

ภาพที่ 3.28 ตัวอย่าง Relation orderdetails

```

<owl:Class rdf:about="orderdetails">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#orderdetails.UnitPrice"/>
    <owl:hasValue rdf:datatype="&xsd:string">0.0000</owl:hasValue>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#orderdetails.Quantity"/>
    <owl:hasValue rdf:datatype="&xsd:string">1</owl:hasValue>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#orderdetails.Discount"/>
    <owl:hasValue rdf:datatype="&xsd:string">0</owl:hasValue>
  </owl:Restriction>
</owl:Class>

```

ภาพที่ 3.29 ตัวอย่างการแปลง Attribute ที่ค่า Default Value ใน Relation เป็นออนโทโลยี

3.4.2.5 ข้อมูลในฐานะข้อมูลเชิงสัมพันธ์

เป็นกระบวนการในการดึงข้อมูลที่จัดเก็บในแต่ละ Relation ของฐานข้อมูลเชิงสัมพันธ์ จะแปลงเป็น Individual ในออนโทโลยี ดังแสดงในภาพที่ 3.30 เป็นตัวอย่างข้อมูลใน Relation region และภาพที่ 3.31 เป็นตัวอย่างการแปลงเป็นออนโทโลยี

Relation :: region	
RegionID	RegionDescription
1	Eastern
2	Westerns
3	Northern
4	Southern

ภาพที่ 3.30 ตัวอย่างข้อมูลใน Relation region

```

<owl:NamedIndividual rdf:about="2">
  <rdf:type rdf:resource="region"/>
  <p1:region.RegionID rdf:datatype="&xsd:string">2</p1:region.RegionID>
  <p1:region.RegionDescription
rdf:datatype="&xsd:string">Westerns</p1:region.RegionDescription>
</owl:NamedIndividual>

```

ภาพที่ 3.31 ตัวอย่างการแปลงข้อมูลเป็นออนโทโลยี

3.4.2.6 สรุปวิธีการในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี

จากวิธีการในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยีที่ได้ นำเสนอไปข้างต้น สามารถแบ่งออกได้เป็น 2 ส่วน คือ ส่วนที่ 1 จะเป็นวิธีการในการแปลงจากงานวิจัยก่อนหน้า และส่วนที่ 2 จะเป็นวิธีการที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม ซึ่งสามารถสรุปวิธีการที่ใช้ในการแปลงดังตารางที่ 3.2 และ 3.3 ตามลำดับ

ตารางที่ 3.2 สรุปวิธีการที่ใช้ในการแปลงจากงานวิจัยก่อนหน้า

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี
1	Normal Relation	Class
2	Binary Relation	Convert เป็น 2 Object Property ที่ผกผันกัน
3	Inheritance Relation	SubClassOf
4	Normal Attribute ที่กำหนดคุณสมบัติเป็น Null	DatatypeProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute
5	Attribute ที่มีคุณสมบัติเป็น Foreign Key	ObjectProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Class ที่ตรงกับ Reference Relation ของ Attribute
6	ข้อมูล	Individual

ตารางที่ 3.3 สรุปวิธีการในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	ข้อดี
1	Normal Attribute ที่กำหนดคุณสมบัติเป็น Not Null	DatatypeProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute minCardinality = 1 	เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ <ul style="list-style-type: none"> minCardinality = 1 เพื่อกำหนดว่าจะต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1
2	Normal Attribute ที่กำหนดคุณสมบัติเป็น Unique	1. DatatypeProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute maxCardinality = 1 2. ObjectProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Class ที่ตรงกับ Reference Relation ของ Attribute InverseFunctionalProperty maxCardinality = 1 	เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ <ul style="list-style-type: none"> maxCardinality = 1 เพื่อกำหนดว่าข้อมูลต้องไม่ซ้ำซ้อนกัน InverseFunctionalProperty เพื่อบอกว่า Property มี Subject เพียงค่าเดียว ถ้าหากมี Subject หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Subject เหล่านั้นคือ Subject เดียวกัน

ตารางที่ 3.3 สรุปวิธีการในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม (ต่อ)

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	ข้อดี
3	Normal Attribute ที่กำหนดคุณสมบัติเป็น Unique และ Not Null	1. DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute ● Cardinality = 1 2. ObjectProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Class ที่ตรงกับ Reference Relation ของ Attribute ● InverseFunctionalProperty ● Cardinality = 1 	เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ <ul style="list-style-type: none"> ● Cardinality = 1 เพื่อกำหนดว่าต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 และข้อมูลต้องไม่ซ้ำซ้อนกัน ● InverseFunctionalProperty เพื่อบอกว่า Property มี Subject เพียงค่าเดียว ถ้าหากมี Subject หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Subject เหล่านั้นคือ Subject เดียวกัน
4	Attribute ที่มีคุณสมบัติเป็น Primary Key เพียงหนึ่งเดียว	1. DatatypeProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute ● Cardinality = 1 2. ObjectProperty <ul style="list-style-type: none"> ● Domain : Class ที่ตรงกับ Relation ของ Attribute ● Range : Class ที่ตรงกับ Reference Relation ของ Attribute ● InverseFunctionalProperty ● Cardinality = 1 	เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ <ul style="list-style-type: none"> ● Cardinality = 1 เพื่อกำหนดว่าต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 และข้อมูลต้องไม่ซ้ำซ้อนกัน ● InverseFunctionalProperty เพื่อบอกว่า Property มี Subject เพียงค่าเดียว ถ้าหากมี Subject หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Subject เหล่านั้นคือ Subject เดียวกัน

ตารางที่ 3.3 สรุปกฎในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม (ต่อ)

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	ข้อดี
5	Attribute ที่มีคุณสมบัติเป็น Primary Key และ Foreign Key เพียงหนึ่งเดียว	<p>1. DatatypeProperty</p> <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute • Cardinality = 1 <p>2. ObjectProperty</p> <ul style="list-style-type: none"> • Domain : Class ที่ตรงกับ Relation ของ Attribute • Range : Class ที่ตรงกับ Reference Relation ของ Attribute • InverseFunctionalProperty • FunctionalProperty • Cardinality = 1 	<p>เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ</p> <ul style="list-style-type: none"> • Cardinality = 1 เพื่อกำหนดว่าต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 และข้อมูลต้องไม่ซ้ำซ้อนกัน • InverseFunctionalProperty เพื่อบอกว่า Property มี Subject เพียงค่าเดียว ถ้าหากมี Subject หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Subject เหล่านั้นคือ Subject เดียวกัน (ตรงกับ Primary Key) • FunctionalProperty เพื่อบอกว่า Property มี Object ได้เพียงค่าเดียว หากมี Object หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Object เหล่านั้นคือ Object เดียวกัน (ตรงกับ Foreign Key)

ตารางที่ 3.3 สรุปกฎในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม (ต่อ)

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	ข้อดี
6	Attribute ที่มีคุณสมบัติ Foreign Key และ Inverse Reference	ObjectProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Class ที่ตรงกับ Reference Relation ของ Attribute FunctionalProperty minCardinality = 1 InverseOf 	เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ <ul style="list-style-type: none"> minCardinality = 1 เพื่อกำหนดว่าจะต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 FunctionalProperty เพื่อบอกว่า Property มี Object ได้เพียงค่าเดียว หากมี Object หลายค่าระบบจะอนุมานโดยอัตโนมัติว่า Object เหล่านั้นคือ Object เดียวกัน (ตรงกับ Foreign Key) InverseOf เพื่อระบุว่า Property 2 Object สามารถ Inverse ข้อมูลระหว่างกันได้
7	Attribute ที่มีคุณสมบัติ เป็น Foreign Key และ Reference มายัง Relation ตนเอง	ObjectProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Class ที่ตรงกับ Reference Relation ของ Attribute FunctionalProperty TransitiveProperty minCardinality = 1 	เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ <ul style="list-style-type: none"> minCardinality = 1 เพื่อกำหนดว่าจะต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 FunctionalProperty เพื่อบอกว่า Property มี Object ได้เพียงค่าเดียว หากมี Object หลายค่าระบบจะอนุมานโดยอัตโนมัติว่า Object เหล่านั้นคือ Object เดียวกัน (ตรงกับ Foreign Key) TransitiveProperty เพื่อบอกว่า Property มีคุณสมบัติการสืบทอด ซึ่งจะช่วยในการอนุมานผลลัพธ์

ตารางที่ 3.3 สรุปกฎในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม (ต่อ)

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	ข้อดี
8	Attribute ที่มีคุณสมบัติเป็น Primary Key (Composite Key)	<ol style="list-style-type: none"> DatatypeProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute minCardinality = 1 ObjectProperty <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Class ที่ตรงกับ Reference Relation ของ Attribute InverseFunctionalProperty minCardinality = 1 	<p>เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ</p> <ul style="list-style-type: none"> minCardinality = 1 เพื่อกำหนดว่าจะต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 InverseFunctionalProperty เพื่อบอกว่า Property มี Subject เพียงค่าเดียว ถ้าหากมี Subject หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Subject เหล่านั้นคือ Subject เดียวกัน (ตรงกับ Primary Key)

ตารางที่ 3.3 สรุปกฎในการแปลงที่งานวิจัยนี้ได้นำเสนอเพิ่มเติม (ต่อ)

ลำดับ	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	ข้อดี
9	Attribute ที่มีคุณสมบัติเป็น Primary Key และ Foreign Key (Composite Key)	<p>1. DatatypeProperty</p> <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Datatype ใน XSD Datatype ที่ตรงกับ Attribute minCardinality = 1 <p>2. ObjectProperty</p> <ul style="list-style-type: none"> Domain : Class ที่ตรงกับ Relation ของ Attribute Range : Class ที่ตรงกับ Reference Relation ของ Attribute InverseFunctionalProperty FunctionalProperty minCardinality = 1 	<p>เป็นการกำหนดคุณสมบัติของ Property ให้ตรงกับ Attribute คือ</p> <ul style="list-style-type: none"> minCardinality = 1 เพื่อกำหนดว่าจะต้องมีจำนวนสมาชิกอย่างน้อยที่สุดคือ 1 InverseFunctionalProperty เพื่อบอกว่า Property นี้ Subject เพียงค่าเดียว ถ้าหากมี Subject หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Subject เหล่านั้นคือ Subject เดียวกัน ซึ่งตรงกับคุณสมบัติ Primary Key FunctionalProperty เพื่อบอกว่า Property นี้มี Object ได้เพียงค่าเดียว หากมี Object หลายค่า ระบบจะอนุมานโดยอัตโนมัติว่า Object เหล่านั้นคือ Object เดียวกัน ซึ่งตรงกับคุณสมบัติ Foreign Key
10	Attribute ที่กำหนด Default Value	hasValue	เป็นการกำหนดค่าเริ่มต้นให้กับออนโทโลยี

3.5 การวัดผลการทดลอง

3.5.1 วัดประสิทธิภาพ

ข้อมูลที่ใช้ในการทดสอบ จะใช้ฐานข้อมูลของการซื้อขายสินค้า มาทำการทดสอบการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี โดยมีตารางทั้งหมด 17 ตารางที่มีโครงสร้างแตกต่างกัน โดยวิธีในการวัดผลการทดลอง ซึ่งจะวัดจากจำนวนความถูกต้องตามกฎต่างๆ ที่ได้นำเสนอในงานวิจัยนี้

ตารางที่ 3.4 ตารางแสดงผลการทดสอบจำแนกตามกลุ่มที่มีผลการทดสอบเป็นบวก และผลการทดสอบเป็นลบ

		Condition as determined by Gold standard		
		<i>positive</i>	<i>negative</i>	
Test outcome	<i>Positive</i>	True positive (TP)	False positive (FP)	Positive predictive value or Precision
	<i>Negative</i>	False negative (FN)	True negative (TN)	Negative predictive value
		Sensitivity or Recall	Specificity (or its complement, Fall-Out)	Accuracy

หมายเหตุ. จาก “การแยกภาพตัวอักษรลายมือเขียนภาษาไทยแบบอัตโนมัติ,” โดย วิเชษฐุรจัน เอี่ยมสำอางค์, 2556, น. 72.

จากตาราง 3.4 ผลการทดสอบจะแบ่งออกเป็นสองกลุ่มได้แก่ กลุ่มแรก คือ ผลการทดสอบเป็นบวก (Positive) หมายความว่ามีการแปลงมีความถูกต้องตามกฎ และกลุ่มที่สอง คือ ผลการทดสอบเป็นลบ (Negative) หมายความว่ามีการแปลงไม่ถูกต้องตามกฎ ซึ่งในตารางจะให้ความหมายดังนี้

True Positive = ระบบสามารถแปลงได้ถูกต้องและมนุษย์แปลงได้ถูกต้อง

False Positive = ระบบสามารถแปลงได้ถูกต้องแต่มนุษย์ไม่สามารถแปลงได้

ถูกต้อง

True Negative = ระบบและมนุษย์ไม่สามารถแปลงได้ถูกต้อง

False Negative = ระบบไม่สามารถแปลงได้ถูกต้องแต่มนุษย์สามารถแปลงได้ถูกต้อง

โดยค่าที่ใช้วัดความถูกต้องคือ Recall เป็นการหาค่าความแม่นยำของระบบการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี

$$\text{Recall} = \frac{\text{Number of True Positive}}{\text{Number of True Positive} + \text{Number of False Negative}}$$

ในการทดลองนี้ผลที่ได้ จะเป็นคำตอบผลลัพธ์ของระบบการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี โดยเปรียบเทียบกับมนุษย์ที่สามารถแปลงได้ถูกต้องเสมอ (True Positive และ False Negative) เท่านั้น

3.5.2 วัดความพึงพอใจ

3.5.2.1 เครื่องมือที่ใช้ในการรวบรวมข้อมูล

เครื่องมือที่ใช้วัดผลในงานวิจัย จะใช้เป็นแบบสอบถามความพึงพอใจในการใช้งานระบบ ซึ่งมีทั้งหมด 3 ตอนดังนี้

ตอนที่ 1 เป็นข้อมูลส่วนตัวของผู้เข้าร่วมการทดลอง ได้แก่ เพศ อายุ ระดับการศึกษา อาชีพ ตำแหน่งงาน ประสบการณ์ในการทำงาน มีความรู้ทางด้านฐานข้อมูลเชิงสัมพันธ์หรือไม่ และมีความรู้ทางด้านออนโทโลยีหรือไม่

ตอนที่ 2 เป็นการวัดระดับความพึงพอใจในการใช้งานระบบ โดยจะแบ่งระดับความพึงพอใจเป็น 5 ระดับ ดังนี้

ความพึงพอใจในระดับมากที่สุด ระดับคะแนน	5
ความพึงพอใจในระดับมาก ระดับคะแนน	4
ความพึงพอใจในระดับปานกลาง ระดับคะแนน	3
ความพึงพอใจในระดับน้อย ระดับคะแนน	2
ความพึงพอใจในระดับน้อยมาก ระดับคะแนน	1

ตอนที่ 3 เป็นคำถามปลายเปิดเกี่ยวกับปัญหาที่พบและข้อเสนอแนะเพิ่มเติมในการใช้งานระบบการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้

3.5.2.2 การเก็บรวบรวมข้อมูล

ในการวิจัย ผู้วิจัยได้ดำเนินการเก็บรวบรวมข้อมูลโดยใช้แบบสอบถาม เป็นเครื่องมือในการเก็บรวบรวมข้อมูล โดยผู้วิจัยแจกแบบสอบถามให้กับบุคคลที่มีความรู้ในด้าน ฐานข้อมูล ได้รับการศึกษาหรือทำงานทางด้านเทคโนโลยีสารสนเทศ หรือในสาขาที่เกี่ยวข้องกับ ทางด้านฐานข้อมูล จำนวน 100 ชุด และได้รับแบบสอบถามกลับมาทั้งหมด 72 ชุด

3.5.2.3 การวิเคราะห์ข้อมูล

ตอนที่ 1 แบบสอบถามเกี่ยวกับข้อมูลส่วนตัวของผู้เข้าร่วมการทดลอง วิเคราะห์ข้อมูลโดยการแจกแจงความถี่และหาค่าร้อยละ

ตอนที่ 2 แบบสอบถามเกี่ยวกับความพึงพอใจของผู้ใช้งานระบบ วิเคราะห์ข้อมูลโดยหาค่าเฉลี่ย (\bar{X}) ของระดับความพึงพอใจและค่าเบี่ยงเบนมาตรฐาน (SD.) จากนั้น นำไปหาค่าเฉลี่ยที่ไปเปรียบเทียบกับที่กำหนดไว้ แล้วแปลความหมายดังนี้

เกณฑ์ในการแปลความหมายของค่าเฉลี่ย

คะแนนเฉลี่ยระหว่าง	4.51 – 5.00	แปลความหมายว่า	มากที่สุด
คะแนนเฉลี่ยระหว่าง	3.51 – 4.50	แปลความหมายว่า	มาก
คะแนนเฉลี่ยระหว่าง	2.51 – 3.50	แปลความหมายว่า	ปานกลาง
คะแนนเฉลี่ยระหว่าง	1.51 – 2.50	แปลความหมายว่า	น้อย
คะแนนเฉลี่ยระหว่าง	1.00 – 1.50	แปลความหมายว่า	น้อยที่สุด

ตอนที่ 3 แบบสอบถามเกี่ยวกับปัญหาที่พบและข้อเสนอแนะอื่นๆ ที่ ต้องการอธิบายเพิ่มเติม

3.5.2.4 สถิติที่ใช้ในงานวิจัย

การวิเคราะห์ข้อมูลที่ได้จากแบบสอบถาม จะนำข้อมูลที่รวบรวมได้ไป ประมวลผลด้วยระบบคอมพิวเตอร์ โดยใช้ Excel มาใช้ในการคำนวณเชิงสถิติ โดยใช้สถิติดังต่อไปนี้ การวิเคราะห์ข้อมูลเบื้องต้นจะใช้การแจกแจงความถี่ (Frequency Distribution) และค่าร้อยละ (Percentage) ค่าเฉลี่ย (\bar{X}) และส่วนเบี่ยงเบนมาตรฐาน (SD.)

บทที่ 4

ผลการวิจัยและอภิปรายผล

ในบทนี้จะกล่าวถึงวิธีการทดลอง ผลการทดลอง และประสิทธิภาพในการทำงานของระบบการสร้างอนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์ แบบปรับแต่งได้ รวมถึงการวิเคราะห์ความแตกต่างของเปอร์เซ็นต์ความพึงพอใจของผู้ใช้งานระบบ เพื่อศึกษาความพึงพอใจของผู้ใช้งานที่มีต่อระบบ

4.1 วิธีการทดลอง

การทดลองในงานวิจัยนี้ได้นำฐานข้อมูลเกี่ยวกับการซื้อขายสินค้า ซึ่งจะอยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ โดยฐานข้อมูลจะประกอบไปด้วยด้วยตารางที่มีโครงสร้างแตกต่างกันจำนวน 17 ตาราง ดังแสดงในภาพที่ 3.2 จากนั้นนำฐานข้อมูลเหล่านั้น เข้าสู่ขั้นตอนการแปลงเป็นอนโทโลยี โดยเริ่มจากการให้ระบบทำการเชื่อมต่อกับฐานข้อมูล จากนั้นจะเข้าสู่ขั้นตอนของการตรวจสอบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ โดยมนุษย์จะเป็นผู้ทำการตรวจสอบ เมื่อทำการตรวจสอบเรียบร้อยแล้ว จะเข้าสู่ขั้นตอนของการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นอนโทโลยี โดยใช้กฎ ที่ได้นำเสนอในบทที่ 3 ซึ่งในขั้นตอนนี้ ระบบจะทำการพิจารณาแต่ละโครงสร้างของตารางในฐานข้อมูลเชิงสัมพันธ์ เปรียบเทียบกับโครงสร้างที่มีคุณสมบัติตรงกับอนโทโลยี แล้วทำการแปลงเป็นโครงสร้างของอนโทโลยี จากนั้นจะเข้าสู่ขั้นตอนของการปรับแต่งอนโทโลยีที่สร้างได้จากระบบ โดยในขั้นตอนนี้จะใช้นักวิจัยเป็นผู้ปรับแต่งโครงสร้างของอนโทโลยี เมื่อทำการปรับแต่งโครงสร้างของอนโทโลยีเรียบร้อยแล้ว จะเข้าสู่ขั้นตอนสุดท้ายคือขั้นตอนของการสร้างเป็นอนโทโลยี โดยจะอยู่ในรูปแบบของ OWL ซึ่งขั้นตอนนี้ระบบจะเป็นผู้สร้างอนโทโลยี และผู้ใช้งานระบบสามารถบันทึกอนโทโลยีที่สร้างได้เป็นไฟล์เอกสาร (.owl) โดยวิธีการทดลองสามารถแบ่งออกเป็น 4 ขั้นตอน ดังนี้

4.1.1 การเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์

เป็นขั้นตอนของการเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์ โดยผู้ใช้งานระบบต้องทำการระบุ IP ของ Server, Port, Username, Password และ Schema ที่ใช้สำหรับการเชื่อมต่อฐานข้อมูล แล้วทำการกด Connected จากนั้นระบบจะทำการเชื่อมต่อฐานข้อมูล ดังแสดงตัวอย่างการเชื่อมต่อฐานข้อมูลในภาพที่ 4.1

ภาพที่ 4.1 การเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์

4.1.2 การตรวจสอบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์

เป็นขั้นตอนของการตรวจสอบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ โดยระบบจะทำการแสดงโครงสร้างตารางทั้งหมดในฐานข้อมูลที่ได้ทำการเชื่อมต่อ รวมถึงแสดงข้อมูลต่างๆ ที่จัดเก็บในฐานข้อมูล จากนั้นให้ผู้ใช้งานระบบจะทำการตรวจสอบโครงสร้างของฐานข้อมูล เมื่อทำการตรวจสอบเรียบร้อยแล้ว ให้ทำการกดปุ่ม Next จากนั้น ระบบจะทำการแปลงโครงสร้าง และข้อมูลในฐานข้อมูลเชิงสัมพันธ์เป็นโครงสร้าง และข้อมูลใน ออนโทโลยี โดยใช้กฎที่ได้นำเสนอในบทที่ 3 ดังแสดงในภาพที่ 4.2 และภาพที่ 4.3

Relational Database

Next »

Relation :: region							[view data]	13
Column	Data Type	NULL	UNIQUE	Key	Table Ref	Defual Value		
RegionID	int	NO		PK				
RegionDescription	varchar	NO						

ภาพที่ 4.2 ระบบแสดงโครงสร้างตาราง region ในฐานข้อมูลเชิงสัมพันธ์

Relation :: region	
RegionID	RegionDescription
1	Eastern
2	Westerns
3	Northern
4	Southern

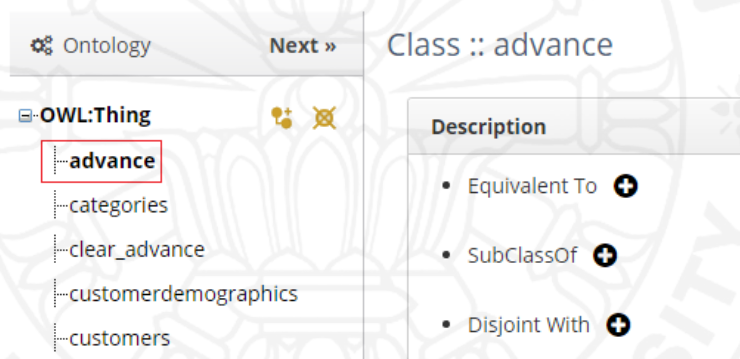
ภาพที่ 4.3 ระบบแสดงข้อมูลที่จัดเก็บในตาราง region

4.1.3 การปรับแต่งโครงสร้างของออนโทโลยี

เป็นขั้นตอนที่ผู้ใช้งานระบบทำการตรวจสอบและปรับแต่งโครงสร้างของออนโทโลยี ภายหลังจากที่ระบบสร้างขึ้นมาโดยอัตโนมัติ ซึ่งมีรายละเอียดดังนี้

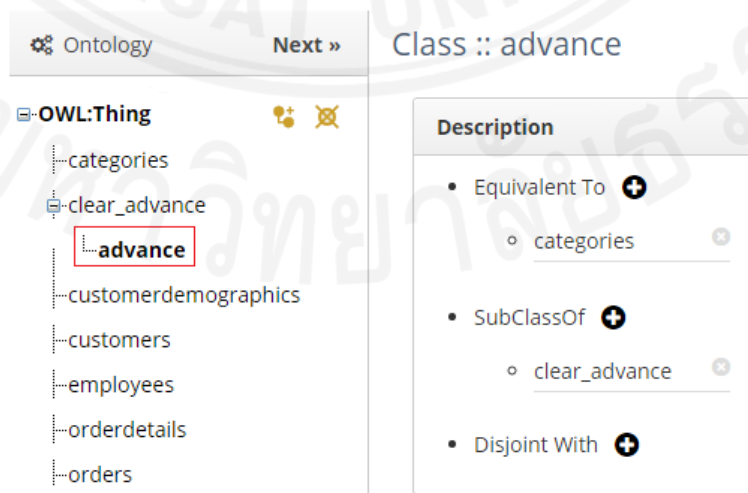
4.1.3.1 การปรับแต่ง Class และคุณสมบัติของ Class

ผู้ใช้งานระบบสามารถเพิ่มหรือลด Class รวมถึงคุณสมบัติของ Class ในออนโทโลยีได้ โดยคุณสมบัติของ Class ที่สามารถปรับแต่งได้ คือ Equivalent To, SubClassOf และ Disjoint With ตัวอย่างเช่น การปรับแต่ง Class advance ซึ่งจากเดิมระบบไม่ได้สร้างคุณสมบัติใดเลยให้กับ Class advance ดังแสดงในภาพที่ 4.4 แต่ผู้ใช้งานระบบสามารถปรับแต่งคุณสมบัติ Equivalent To ให้คุณสมบัติเท่ากับ Class categories และเป็น SubClassOf ของ Class clear_advance ดังแสดงในภาพที่ 4.5



Before

ภาพที่ 4.4 ตัวอย่างก่อนการปรับเปลี่ยน Class advance



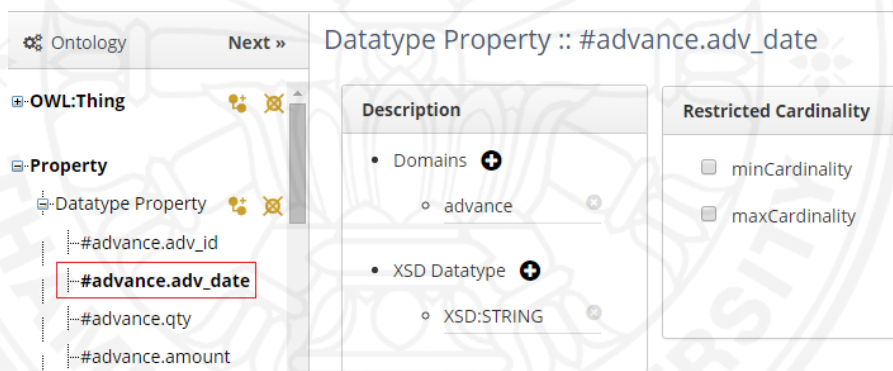
After

ภาพที่ 4.5 ตัวอย่างหลังการปรับเปลี่ยน Class advance

4.1.3.2 การปรับแต่ง DatatypeProperty และคุณสมบัติของ

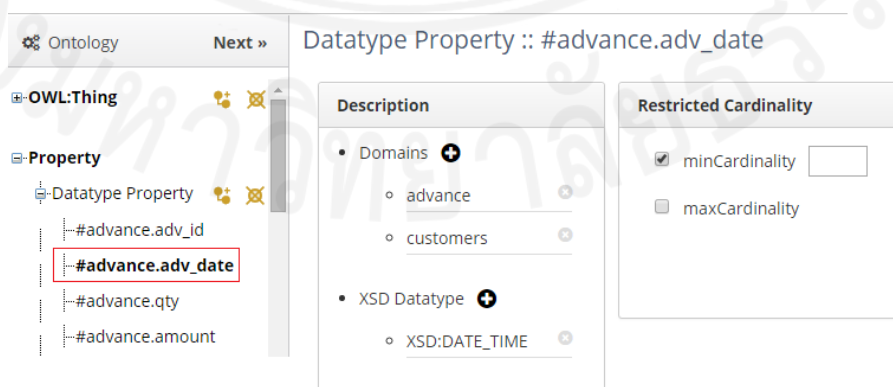
DatatypeProperty

ผู้ใช้งานระบบสามารถเพิ่มหรือลด DatatypeProperty รวมถึงคุณสมบัติของ DatatypeProperty ในออนโทโลยีได้ โดยคุณสมบัติของ DatatypeProperty ที่สามารถปรับแต่งได้ คือ Domain, XSD Datatype และ Restricted Cardinality ตัวอย่างเช่น การปรับแต่ง DatatypeProperty #advance.adv_date ซึ่งจากเดิมระบบได้สร้างคุณสมบัติ Domain มี Class เป็น advance, XSD Datatype เป็น XSD:STRING และไม่ได้กำหนดคุณสมบัติ Restricted Cardinality ให้กับ DatatypeProperty #advance.adv_date ดังแสดงในภาพที่ 4.6 แต่ผู้ใช้งานระบบสามารถเพิ่ม Class customers ให้กับคุณสมบัติ Domain หรือเปลี่ยน XSD Datatype เป็น XSD:DATE_TIME หรืออาจจะเพิ่มคุณสมบัติ Restricted Cardinality เป็น minCardinality ให้กับ DatatypeProperty #advance.adv_date ดังแสดงในภาพที่ 4.7



Before

ภาพที่ 4.6 ตัวอย่างก่อนการปรับเปลี่ยน DatatypeProperty #advance.adv_date



After

ภาพที่ 4.7 ตัวอย่างหลังการปรับเปลี่ยน DatatypeProperty #advance.adv_date

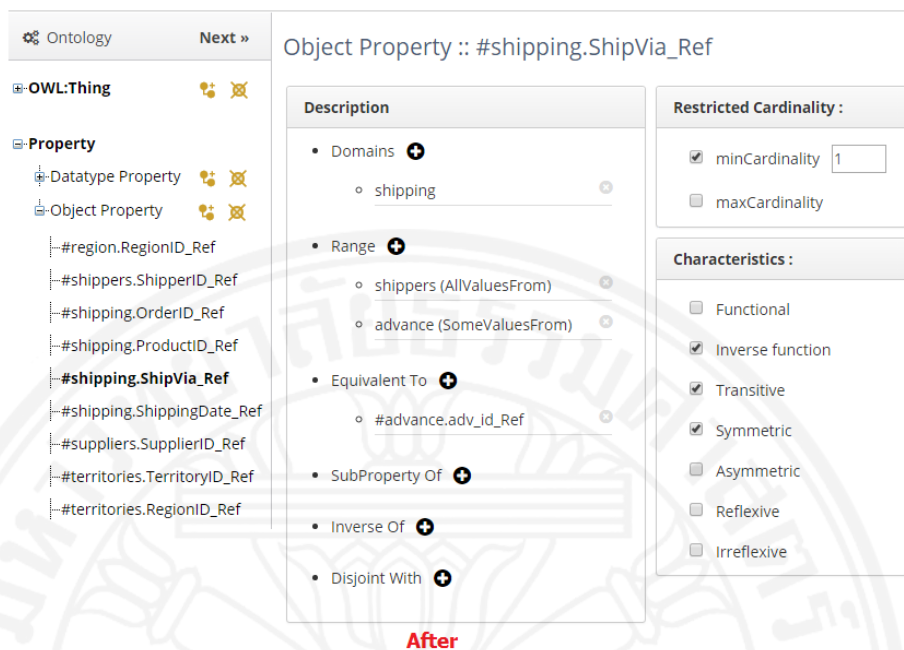
4.1.3.3 การปรับแต่ง ObjectProperty และคุณสมบัติของ ObjectProperty

ผู้ใช้งานระบบสามารถเพิ่มหรือลด ObjectProperty รวมถึงคุณสมบัติของ ObjectProperty ในออนโทโลยีได้ โดยคุณสมบัติของ ObjectProperty ที่สามารถปรับแต่งได้ คือ Domain, Range, Equivalent To, SubProperty Of, Inverse Of, Disjoint With, Restricted Cardinality และ Characteristics ตัวอย่างเช่น การปรับแต่ง ObjectProperty #shipping.ShipVia_Ref ซึ่งจากเดิมระบบได้สร้างเฉพาะคุณสมบัติ Domain มี Class เป็น shipping, Range เป็น Class shippers, Restricted Cardinality เป็น minCardinality เป็น 1 และ Characteristics เป็น Functional และ Inverse Functional ให้กับ ObjectProperty #shipping.ShipVia_Ref ดังแสดงในภาพที่ 4.8 แต่ผู้ใช้งานระบบสามารถเพิ่ม Class advance ให้กับคุณสมบัติ Range หรือเพิ่ม Equivalent To เป็น #advance.adv_id_Ref หรือเปลี่ยนคุณสมบัติ Characteristics เป็น Inverse Functional, Transitive และ Symmetric ให้กับ ObjectProperty #shipping.ShipVia_Ref ดังแสดงในภาพที่ 4.9

Object Property :: #shipping.ShipVia_Ref	
Description <ul style="list-style-type: none"> Domains + <ul style="list-style-type: none"> shipping Range + <ul style="list-style-type: none"> shippers (AllValuesFrom) Equivalent To + SubProperty Of + Inverse Of + Disjoint With + 	Restricted Cardinality : <ul style="list-style-type: none"> <input checked="" type="checkbox"/> minCardinality 1 <input type="checkbox"/> maxCardinality
Characteristics : <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Functional <input checked="" type="checkbox"/> Inverse function <input type="checkbox"/> Transitive <input type="checkbox"/> Symmetric <input type="checkbox"/> Asymmetric <input type="checkbox"/> Reflexive <input type="checkbox"/> Irreflexive 	

Before

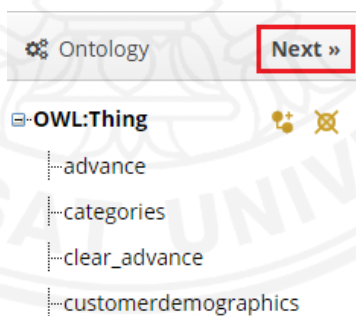
ภาพที่ 4.8 ตัวอย่างก่อนการปรับเปลี่ยน ObjectProperty #shipping.ShipVia_Ref



After

ภาพที่ 4.9 ตัวอย่างหลังการปรับเปลี่ยน ObjectProperty #shipping.ShipVia_Ref

เมื่อผู้ใช้งานระบบทำการปรับแต่งโครงสร้างของออนโทโลยีเรียบร้อยแล้ว ให้กดปุ่ม Next ดังแสดงในภาพที่ 4.10 เพื่อเข้าสู่ขั้นตอนของการสร้างไฟล์ออนโทโลยี

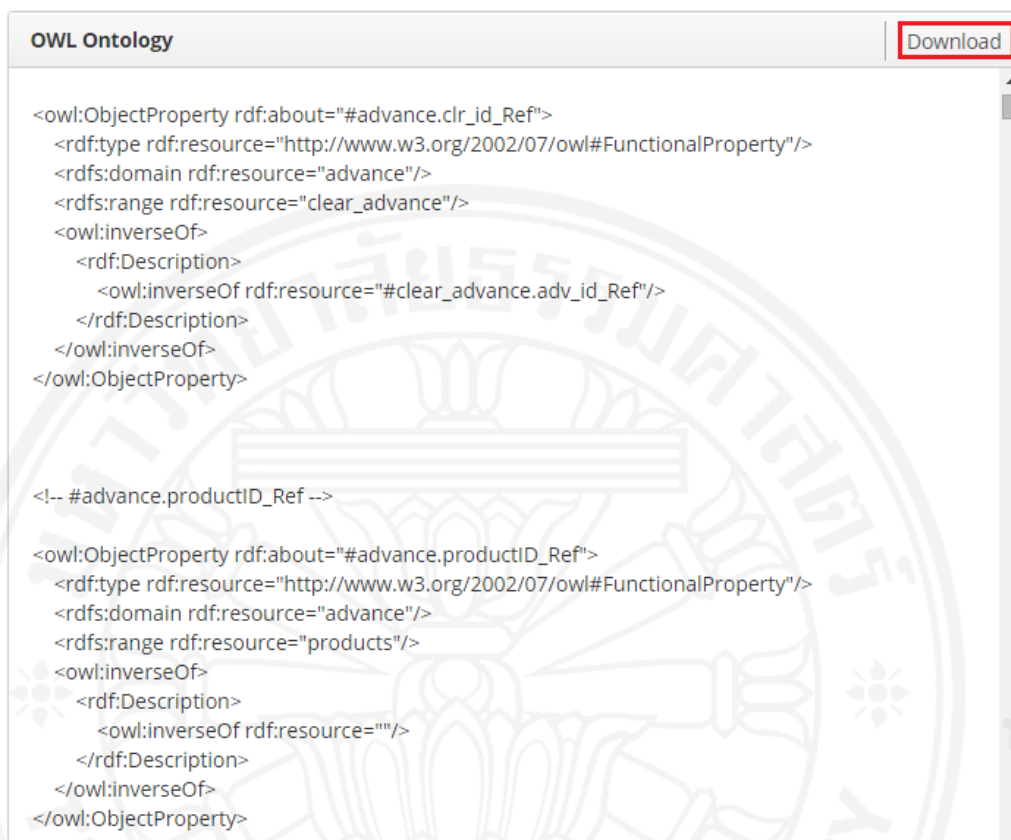


ภาพที่ 4.10 การกดปุ่ม Next เพื่อเข้าสู่ขั้นตอนต่อไป

4.1.4 การสร้างออนโทโลยี

เป็นขั้นตอนที่ระบบจะทำการสร้างออนโทโลยี โดยจะอยู่ในรูปแบบของ OWL ซึ่งผู้ใช้งานระบบสามารถบันทึกออนโทโลยีที่สร้างได้ ออกมาเป็นไฟล์เอกสาร (.owl) โดยกดที่ปุ่ม Download ดังแสดงในภาพที่ 4.11

Generate Ontology



```

<owl:ObjectProperty rdf:about="#advance.clr_id_Ref">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="advance"/>
  <rdfs:range rdf:resource="clear_advance"/>
  <owl:inverseOf>
    <rdf:Description>
      <owl:inverseOf rdf:resource="#clear_advance.adv_id_Ref"/>
    </rdf:Description>
  </owl:inverseOf>
</owl:ObjectProperty>

<!-- #advance.productID_Ref -->

<owl:ObjectProperty rdf:about="#advance.productID_Ref">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="advance"/>
  <rdfs:range rdf:resource="products"/>
  <owl:inverseOf>
    <rdf:Description>
      <owl:inverseOf rdf:resource=""/>
    </rdf:Description>
  </owl:inverseOf>
</owl:ObjectProperty>

```

ภาพที่ 4.11 ตัวอย่างออนโทโลยีที่อยู่ในรูปแบบของ OWL

4.2 ผลการทดลองและประสิทธิภาพการทำงานของระบบ

ผลการทดลองในการแปลงจากฐานข้อมูลเชิงสัมพันธ์เป็นออนโทโลยี ของฐานข้อมูล การซื้อขายสินค้า จะเป็นผลการแปลงที่ได้จากระบบ ซึ่งจะไม่มีการปรับแต่งเพิ่มเติมจากผู้ใช้งานระบบ เปรียบเทียบกับผลการแปลงที่ได้จากมนุษย์ ซึ่ง เป็นผู้เชี่ยวชาญและมีประสบการณ์ทำงานด้าน ฐานข้อมูลและออนโทโลยี จำนวน 5 ท่าน โดยในการแปลงจะเป็นไปตามกฎที่ได้นำเสนอในบทที่ 3 ถ้าผลการแปลงตรงตามกฎ จะได้คะแนนเป็น 1 แต่ถ้าผลการแปลงไม่ตรงตามกฎ จะได้คะแนนเป็น 0 ซึ่งสามารถแสดงผลการแปลงได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการแปลงฐานข้อมูลการซื้อขายสินค้าเป็นออนโทโลยี

กฎการแปลง	ฐานข้อมูลเชิงสัมพันธ์	ออนโทโลยี	
		ระบบ	มนุษย์
1. Normal Relation	17	15	15
2. Binary Relation	2	2	2
3. Inheritance Relation	1	1	1
4. Normal Attribute ที่กำหนดคุณสมบัติเป็น Null	51	51	51
5. Normal Attribute ที่กำหนดคุณสมบัติ เป็น Not Null	25	25	25
6. Normal Attribute ที่กำหนดคุณสมบัติ เป็น Unique	1	1	1
7. Normal Attribute ที่กำหนดคุณสมบัติ เป็น Unique และ Not Null	1	1	1
8. Attribute ที่มีคุณสมบัติเป็น Primary Key เพียงหนึ่งเดียว	12	12	12
9. Attribute ที่มีคุณสมบัติเป็น Primary Key และ Foreign Key เพียงหนึ่งเดียว	1	1	1
10. Attribute ที่มีคุณสมบัติเป็น Foreign Key	8	8	8
11. Attribute ที่มีคุณสมบัติ Foreign Key และ Inverse Reference	2	2	2
12. Attribute ที่มีคุณสมบัติเป็น Foreign Key และ Reference มายัง Relation ตนเอง	1	1	1
13. Attribute ที่มีคุณสมบัติเป็น Primary Key (Composite Key)	1	1	1
14. Attribute ที่มีคุณสมบัติเป็น Primary Key และ Foreign Key (Composite Key)	5	5	5
15. Attribute ที่กำหนด Default Value	9	9	9
รวม	137	135	135

จากตารางที่ 4.1 จะได้ว่าระบบสามารถแปลงได้ถูกต้องตามกฎ จำนวน 135 คะแนน และมนุษย์สามารถแปลงได้ถูกต้องตามกฎ จำนวน 135 คะแนน เมื่อนำผลการทดลองที่ได้มา คำนวณหาประสิทธิภาพการทำงานของระบบ จะได้ค่า Recall เท่ากับ 1 หรือคิดเป็นร้อยละ 100 ซึ่งหมายความว่าระบบสามารถแปลงได้ถูกต้องตามกฎที่ได้นำเสนอคิดเป็นร้อยละ 100

4.3 ผลการวิเคราะห์ความพึงพอใจของผู้ใช้งานระบบ

4.3.1 ข้อมูลส่วนตัวของผู้เข้าร่วมการทดลอง

งานวิจัยนี้มีผู้เข้าร่วมการทดลองเป็นจำนวนทั้งหมด 72 คน เป็นชาย จำนวน 47 คน คิดเป็นร้อยละ 65.28 และเป็นหญิง จำนวน 25 คน คิดเป็นร้อยละ 34.72 โดยมีอายุน้อยที่สุดคือ 22 ปี มีอายุมากที่สุดคือ 38 ปี และมีอายุเฉลี่ยของผู้เข้าร่วมการทดลองคือ 28.3 ปี ดังแสดงในตารางที่ 4.2 และ 4.3

ตารางที่ 4.2 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามเพศ

เพศ	จำนวน (คน)	ร้อยละ (%)
ชาย	47	65.28
หญิง	25	34.72
รวม	72	100

ตารางที่ 4.3 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามอายุ

อายุ (ปี)	จำนวน (คน)	อายุน้อยที่สุด (ปี)	อายุมากที่สุด (ปี)	อายุเฉลี่ย (ปี)
	72	22	38	28.3

งานวิจัยนี้ผู้เข้าร่วมการทดลองส่วนใหญ่มีระดับการศึกษาปริญญาตรี จำนวน 63 คน คิดเป็นร้อยละ 87.50 ระดับการศึกษาปริญญาโท จำนวน 9 คน คิดเป็นร้อยละ 12.50 ดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามระดับศึกษา

ระดับการศึกษา	จำนวน (คน)	ร้อยละ (%)
ต่ำกว่าปริญญาตรี	0	0
ปริญญาตรี	63	87.50
ปริญญาโท	9	12.50
รวม	72	100

ผู้เข้าร่วมการทดลองส่วนใหญ่มีอาชีพเป็นพนักงานบริษัทเอกชน จำนวน 38 คน คิดเป็นร้อยละ 52.78 มีอาชีพข้าราชการ /รัฐวิสาหกิจ จำนวน 17 คน คิดเป็นร้อยละ 23.61 มีอาชีพธุรกิจส่วนตัว จำนวน 15 คน คิดเป็นร้อยละ 20.83 และมีอาชีพนักศึกษา จำนวน 2 คน คิดเป็นร้อยละ 2.78 ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามอาชีพ

อาชีพ	จำนวน (คน)	ร้อยละ (%)
พนักงานบริษัทเอกชน	38	52.78
ข้าราชการ/รัฐวิสาหกิจ	17	23.61
ธุรกิจส่วนตัว	15	20.83
นักศึกษา	2	2.78
รวม	72	100

ผู้เข้าร่วมการทดลองส่วนใหญ่มีตำแหน่งเป็น Programmer จำนวน 23 คน คิดเป็นร้อยละ 31.94 มีตำแหน่งเป็น System Analyst (SA) จำนวน 15 คน คิดเป็นร้อยละ 20.83 มีตำแหน่งเป็น Database Administrator จำนวน 9 คน คิดเป็นร้อยละ 12.5 มีตำแหน่งเป็นผู้ดูแลระบบ จำนวน 8 คน คิดเป็นร้อยละ 11.11 มีตำแหน่งเป็นนักวิชาการคอมพิวเตอร์ จำนวน 17 คน คิดเป็นร้อยละ 23.61 และมีตำแหน่งงานอื่นๆ จำนวน 2 คน คิดเป็นร้อยละ 2.78 ดังแสดงในตารางที่ 4.6

ตารางที่ 4.6 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามตำแหน่ง

ตำแหน่งงาน	จำนวน (คน)	ร้อยละ (%)
นักวิชาการคอมพิวเตอร์	17	23.61
ผู้ดูแลระบบ	8	11.11
System Analyst (SA)	15	20.83
Programmer	23	31.94
Database Administrator	9	12.5
อื่นๆ	2	2.78
รวม	72	100

ผู้เข้าร่วมการทดลองส่วนใหญ่มีประสบการณ์ทำงาน 2 ถึง 5 ปี จำนวน 34 คน คิดเป็นร้อยละ 47.22 มีประสบการณ์งานน้อยกว่า 2 ปี จำนวน 26 คน คิดเป็นร้อยละ 36.11 และมีประสบการณ์ทำงานมากกว่า 5 ปีขึ้นไป จำนวน 12 คน คิดเป็นร้อยละ 16.67 ดังแสดงในตารางที่ 4.7

ตารางที่ 4.7 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกตามประสบการณ์ในการทำงาน

ประสบการณ์ในการทำงาน	จำนวน (คน)	ร้อยละ (%)
น้อยกว่า 2 ปี	26	36.11
2 – 5 ปี	34	47.22
5 ปีขึ้นไป	12	16.67
รวม	72	100

ผู้เข้าร่วมการทดลองมีความรู้ทางด้านฐานข้อมูลเชิงสัมพันธ์ จำนวน 72 คน คิดเป็นร้อยละ 100 และส่วนใหญ่มีความรู้ในระดับดี จำนวน 37 คน คิดเป็นร้อยละ 51.39 มีความรู้ในระดับดีมาก จำนวน 9 คน คิดเป็นร้อยละ 12.5 และมีความรู้ในระดับพอใช้ จำนวน 26 คน คิดเป็นร้อยละ 36.11 ดังแสดงในตารางที่ 4.8

ตารางที่ 4.8 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกระดับความรู้
ทางด้านฐานข้อมูลเชิงสัมพันธ์

ระดับความรู้	จำนวน (คน)	ร้อยละ (%)
ดีมาก	37	51.39
ดี	9	12.5
พอใช้	26	36.11
แย่	0	0
แย่มาก	0	0
รวม	72	100

ผู้เข้าร่วมการทดลองมีความรู้ทางด้านออนโทโลยี จำนวน 42 คน คิดเป็นร้อยละ 58.33 และไม่มีความรู้ทางด้านออนโทโลยี จำนวน 30 คน คิดเป็นร้อยละ 41.67 โดยผู้ที่มีความรู้ใน
ด้านออนโทโลยีระดับดี จำนวน 23 คน คิดเป็นร้อยละ 54.76 และมีความรู้ในระดับพอใช้ จำนวน
19 คน คิดเป็นร้อยละ 45.24 ดังแสดงในตารางที่ 4.9 และ 4.10

ตารางที่ 4.9 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกความรู้ทางด้านออนโทโลยี

ความรู้ทางด้านออนโทโลยี	จำนวน (คน)	ร้อยละ (%)
มีความรู้	42	58.33
ไม่มีความรู้	30	41.67
รวม	72	100

ตารางที่ 4.10 จำนวนและร้อยละของผู้เข้าร่วมการทดลอง จำแนกระดับความรู้
ทางด้านออนโทโลยี

ระดับความรู้	จำนวน (คน)	ร้อยละ (%)
ดีมาก	0	0
ดี	23	54.76
พอใช้	19	45.24
แย่	0	0
แย่มาก	0	0
รวม	42	100

4.3.2 ผลการวิเคราะห์ด้านความพึงพอใจ

ผลการวิเคราะห์ข้อมูลความพึงพอใจของผู้เข้าร่วมการทดลองในการใช้งานระบบการสร้างออนไลน์จากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้ สามารถแสดงได้ดังตารางที่ 4.11 โดยคะแนนเฉลี่ยที่ได้ จะนำมาแปลความหมายตามเกณฑ์ดังต่อไปนี้

คะแนนเฉลี่ยระหว่าง	4.51 – 5.00	แปลความหมายว่า	มากที่สุด
คะแนนเฉลี่ยระหว่าง	3.51 – 4.50	แปลความหมายว่า	มาก
คะแนนเฉลี่ยระหว่าง	2.51 – 3.50	แปลความหมายว่า	ปานกลาง
คะแนนเฉลี่ยระหว่าง	1.51 – 2.50	แปลความหมายว่า	น้อย
คะแนนเฉลี่ยระหว่าง	1.00 – 1.50	แปลความหมายว่า	น้อยที่สุด

ตารางที่ 4.11 ค่าสถิติการวิเคราะห์ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของคะแนนความพึงพอใจ

รายการประเมิน	ระดับความพึงพอใจ						
	5	4	3	2	1	\bar{X}	SD
1. ความสามารถของระบบในด้านการเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์	27	44	1	0	0	4.36	0.51
2. ความสามารถของระบบในด้านการสร้าง Class แบบอัตโนมัติ	34	37	1	0	0	4.49	0.53
3. ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Class แบบอัตโนมัติ	30	42	0	0	0	4.41	0.50
4. ความสามารถของระบบในด้านการสร้าง Datatype Property แบบอัตโนมัติ	27	31	14	0	0	4.18	0.74
5. ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Datatype Property แบบอัตโนมัติ	35	27	10	0	0	3.94	0.72
6. ความสามารถของระบบในด้านการกำหนด Constraint ของ Datatype Property แบบอัตโนมัติ	32	25	15	0	0	4.24	0.78
7. ความสามารถของระบบในด้านการสร้าง Object Property แบบอัตโนมัติ	18	43	11	0	0	4.10	0.63

ตารางที่ 4.11 ค่าสถิติการวิเคราะห์ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของ
คะแนนความพึงพอใจ (ต่อ)

รายการประเมิน	ระดับความพึงพอใจ							
	5	4	3	2	1	\bar{x}	SD	
8. ความสามารถของระบบในด้านการกำหนด คุณสมบัติของ Object Property แบบอัตโนมัติ	37	31	4	0	0	4.46	0.60	
9. ความสามารถของระบบในด้านการกำหนด Constraint ของ Object Property แบบอัตโนมัติ	23	28	21	0	0	4.08	0.79	
10. ความสามารถของระบบในด้านการสร้าง ออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์ แบบอัตโนมัติ	46	26	0	0	0	4.64	0.48	
11. ความสามารถของระบบในด้าน การตรวจสอบและปรับปรุงโครงสร้าง ของออนโทโลยี	39	31	2	0	0	4.51	0.56	
12. ความสามารถของระบบในการสร้างไฟล์ ออนโทโลยี (File .OWL)	43	29	0	0	0	4.60	0.49	
13. ความง่ายต่อการใช้งานระบบ	24	41	7	0	0	4.24	0.62	
14. ความเหมาะสมของเนื้อหา	16	52	4	0	0	4.11	0.50	
15. ประสิทธิภาพโดยรวมของระบบ	13	54	5	0	0	3.89	0.49	
16. ความรวดเร็วในการตอบสนองของระบบ	15	34	23	0	0	3.89	0.72	
17. ความน่าเชื่อถือได้ของระบบ	39	32	1	0	0	4.53	0.53	
18. ความสวยงาม ความทันสมัยของระบบ	52	18	2	0	0	4.69	0.52	
						รวม	4.28	0.60
						คิดเป็นร้อยละ	85.6	

จากตารางที่ 4.11 ผลสถิติการวิเคราะห์ค่าเฉลี่ยคะแนนความพึงพอใจจากผู้เข้าร่วมการทดลอง จำนวน 72 คน มีค่าเฉลี่ยคะแนนความพึงพอใจอยู่ที่ 4.28 คิดเป็นร้อยละ 85.6 ซึ่งสามารถแปลความหมายได้ว่ามีความพึงพอใจในระดับมาก และมีค่าเบี่ยงเบนมาตรฐานเฉลี่ยอยู่ที่ 0.60 โดยคะแนนความพึงพอใจเฉลี่ยสามารถเรียงลำดับจากมากไปน้อย แยกตามรายการประเมินแต่ละข้อได้ดังต่อไปนี้

1. ความสวยงาม ความทันสมัยของระบบ (\bar{X} = 4.69 หรือคิดเป็น 93.8%)
2. ความสามารถของระบบในด้านการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ (\bar{X} = 4.64 หรือคิดเป็น 92.8%)
3. ความสามารถของระบบในการสร้างไฟล์ออนโทโลยี (File .OWL) (\bar{X} = 4.60 หรือคิดเป็น 92%)
4. ความน่าเชื่อถือได้ของระบบ (\bar{X} = 4.53 หรือคิดเป็น 90.6%)
5. ความสามารถของระบบในด้านการตรวจสอบและปรับปรุงโครงสร้างของออนโทโลยี (\bar{X} = 4.51 หรือคิดเป็น 90.2%)
6. ความสามารถของระบบในด้านการสร้าง Class แบบอัตโนมัติ (\bar{X} = 4.49 หรือคิดเป็น 89.8%)
7. ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Object Property แบบอัตโนมัติ (\bar{X} = 4.46 หรือคิดเป็น 89.4%)
8. ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Class แบบอัตโนมัติ (\bar{X} = 4.41 หรือคิดเป็น 88.2%)
9. ความสามารถของระบบในด้านการเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์ (\bar{X} = 4.36 หรือคิดเป็น 87.2%)
10. ความสามารถของระบบในด้านการกำหนด Constraint ของ Datatype Property แบบอัตโนมัติ (\bar{X} = 4.24 หรือคิดเป็น 84.8%)
11. ความง่ายต่อการใช้งานระบบ (\bar{X} = 4.24 หรือคิดเป็น 84.8%)
12. ความสามารถของระบบในด้านการสร้าง Datatype Property แบบอัตโนมัติ (\bar{X} = 4.18 หรือคิดเป็น 83.6%)
13. ความเหมาะสมของเนื้อหา (\bar{X} = 4.11 หรือคิดเป็น 82.2%)
14. ความสามารถของระบบในด้านการสร้าง Object Property แบบอัตโนมัติ (\bar{X} = 4.10 หรือคิดเป็น 82%)
15. ความสามารถของระบบในด้านการกำหนด Constraint ของ Object Property แบบอัตโนมัติ (\bar{X} = 4.08 หรือคิดเป็น 81.86%)

16. ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Datatype Property แบบอัตโนมัติ (\bar{X} = 3.94 หรือคิดเป็น 78.8%)
17. ประสิทธิภาพโดยรวมของระบบ (\bar{X} = 3.89 หรือคิดเป็น 77.8%)
18. ความรวดเร็วในการตอบสนองของระบบ (\bar{X} = 3.89 หรือคิดเป็น 77.8%)

ทางผู้เข้าร่วมการทดลองได้พบปัญหาและมีเสนอแนะเพิ่มเติมดังสรุปได้ดังนี้

1. ในกรณีที่มีการดึงข้อมูลที่มีปริมาณมาก ระบบตอบสนองช้าและมีอาการค้าง
2. น่าจะมีคำแนะนำภาษาไทยประกอบการใช้งาน
3. น่าจะเพิ่มให้สามารถปรับปรุงในส่วนของคุณสมบัติของข้อมูลในออนไลน์ได้
4. Datatype ของ Attribute บางชนิด เช่น Blob, Byte ฯลฯ ไม่สามารถ

แปลงได้

4.4 อภิปรายผลการทดลอง

จากตารางที่ 4.1 จะพบว่าระบบนี้สามารถสร้างออนไลน์จากฐานข้อมูลเชิงสัมพันธ์ได้อย่างถูกต้องตามกฎหมายที่ได้แนะนำเสนอและถูกต้องตามที่มนุษย์สร้าง คิดเป็น 100% ซึ่งแสดงให้เห็นว่าสามารถนำระบบนี้ไปใช้สร้างเป็นออนไลน์พื้นฐานจากฐานข้อมูลเชิงสัมพันธ์แทนมนุษย์ได้ ทำให้ช่วยประหยัดทั้งเวลาและค่าใช้จ่ายต่างๆ รวมถึงลดข้อผิดพลาดที่อาจเกิดจากมนุษย์ได้

จากตารางที่ 11 จะพบว่าคะแนนเฉลี่ยความพึงพอใจของผู้เข้าร่วมการทดลองทั้งหมด 72 คน เป็น 4.28 (85.6%) โดยมีความพึงพอใจมากที่สุด 3 ลำดับแรก โดยเรียงจากมากไปน้อย คือ ความสวยงาม ความทันสมัยของระบบ (93.8%), ความสามารถของระบบในด้านการสร้างออนไลน์จากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ (92.8%) และ ความสามารถของระบบในการสร้างไฟล์ออนไลน์ (92%) และความพึงพอใจน้อยที่สุด 3 ลำดับแรก คือ ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Datatype Property แบบอัตโนมัติ (78.8%), ประสิทธิภาพโดยรวมของระบบ (77.8%) และความรวดเร็วในการตอบสนองของระบบ (77.8%)

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ระบบสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์ แบบปรับแต่งได้ เป็นระบบที่สามารถสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติได้ โดยจะเน้นไปที่โครงสร้างและข้อกำหนดของฐานข้อมูลเชิงสัมพันธ์และออนโทโลยี นอกจากนี้ผู้ใช้งานระบบสามารถที่จะตรวจสอบและปรับแต่งโครงสร้างของออนโทโลยีก่อนการสร้างออนโทโลยีที่อยู่ในรูปแบบของ OWL ได้ ซึ่งจะช่วยลดระยะเวลา แรงงาน และทรัพยากรที่ใช้ในการสร้างออนโทโลยี และช่วยเพิ่มประสิทธิภาพทำให้มีความถูกต้องมากยิ่งขึ้น รวมถึงช่วยลดข้อผิดพลาดที่อาจเกิดจากมนุษย์ได้

แม้ว่าวิธีการในการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์ ได้มีการนำเสนอในงานวิจัยก่อนหน้านี้ แต่งานวิจัยส่วนใหญ่จะเน้นไปที่โครงสร้างของฐานข้อมูลเชิงสัมพันธ์และออนโทโลยี แต่ไม่ได้เน้นไปที่ข้อกำหนดของฐานข้อมูลเชิงสัมพันธ์และออนโทโลยี ซึ่งเป็นสิ่งสำคัญที่จะช่วยให้ออนโทโลยีที่สร้างได้มีความถูกต้องมากยิ่งขึ้นตามโครงสร้างและข้อกำหนดของฐานข้อมูลเชิงสัมพันธ์ นอกจากนี้บางข้อกำหนดยังช่วยในการอนุมานคำตอบให้กับออนโทโลยีอีกด้วย

การทดลองในงานวิจัยที่ได้นำฐานข้อมูลการซื้อขายสินค้า ซึ่งอยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ โดยประกอบด้วยตารางทั้งหมด 17 ตาราง ที่มีโครงสร้างแตกต่างกัน เมื่อนำเข้าสู่ขั้นตอนในการแปลงเป็นออนโทโลยี แล้วนำมาวัดประสิทธิภาพการทำงานของระบบ โดยออนโทโลยีที่ได้ต้องไม่ผ่านการปรับแต่งมาจากผู้ใช้งานระบบ พบว่า ระบบสามารถสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์ได้อย่างถูกต้องตามกฎที่ได้นำเสนอและถูกต้องตามที่มนุษย์สร้างจากโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ โดยมีความถูกต้อง คิดเป็นร้อยละ 100

จากการเก็บรวบรวมข้อมูลของผู้เข้าร่วมการทดลอง แล้วนำข้อมูลที่ได้มาทำการวิเคราะห์ เพื่อศึกษาความพึงพอใจในการใช้งานระบบการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบปรับแต่งได้ โดยใช้ค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน

ผลลัพธ์จากการวิเคราะห์ข้อมูลส่วนตัวของผู้เข้าร่วมการทดลอง จำนวน 72 คน พบว่าส่วนใหญ่เป็นผู้ชาย คิดเป็นร้อยละ 65.28 มีอายุเฉลี่ย 28.3 ปี ส่วนใหญ่มีวุฒิการศึกษายู่ในระดับ

ปริญญาตรี คิดเป็นร้อยละ 87.50 มีอาชีพเป็นพนักงานบริษัทเอกชน คิดเป็นร้อยละ 52.78 มีตำแหน่งงานเป็น Programmer ร้อยละ 31.94 ส่วนใหญ่มีประสบการณ์ทำงาน 2 ถึง 5 ปี คิดเป็นร้อยละ 47.22 ผู้เข้าร่วมการทดลองทั้งหมดมีความรู้ทางด้านฐานข้อมูลเชิงสัมพันธ์ มีความรู้ในระดับดีมากโดยส่วนใหญ่ คิดเป็นร้อยละ 51.39 ส่วนใหญ่มีความรู้ทางด้านออนโทโลยี คิดเป็นร้อยละ 58.33 และส่วนใหญ่มีความรู้ในระดับดี คิดเป็นร้อยละ 54.76

ผลลัพธ์จากการวิเคราะห์ข้อมูลความพึงพอใจในการทำงานของระบบและการใช้งานระบบภาพรวมมีความพึงพอใจอยู่ในระดับพึงพอใจมาก เฉลี่ยคิดเป็นร้อยละ 85.6 โดยส่วนที่มีความพึงพอใจมาก 3 ลำดับแรก คือ ความสวยงาม ความทันสมัย คิดเป็นร้อยละ 93.8 ความสามารถของระบบในด้านการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ คิดเป็นร้อยละ 92.8 และ ความสามารถของระบบในการสร้างไฟล์ออนโทโลยี (File .OWL) คิดเป็นร้อยละ 92 ตามลำดับ ส่วนที่จะต้องปรับปรุงในการดำเนินงาน คือ ประสิทธิภาพโดยรวมของระบบ คิดเป็นร้อยละ 77.8 และความรวดเร็วในการตอบสนองของระบบ คิดเป็นร้อยละ 77.8

5.2 ข้อเสนอแนะ

จากการที่ได้ทดสอบในงานวิจัย พบว่ายังมีปัญหาที่ยังไม่ได้รับการวิจัยคือในส่วนของการแมพโครงสร้างประเภทของชนิดข้อมูล (Datatype) ที่ยังมีบางชนิดไม่สามารถแมพได้ และถ้าฐานข้อมูลมีขนาดใหญ่และมีความซับซ้อนมาก ก็ทำให้เกิดปัญหาด้านความเร็วในการประมวลผลข้อมูล แต่อย่างไรก็ตามผลการวิจัยก็ออกมาอยู่ในเกณฑ์ที่ดี สามารถนำไปพัฒนาต่อยอดได้

5.3 แนวทางการวิจัยในอนาคต

แนวทางในการพัฒนาระบบและงานวิจัยที่สามารถต่อยอดได้ในอนาคต อาจจะพัฒนาในส่วนของวิธีการในการแปลงข้อมูลที่จัดเก็บในฐานข้อมูลเป็นข้อมูลออนโทโลยี ให้มีความเชื่อมโยงกันของข้อมูลในรูปแบบอื่นๆ ที่ไม่ได้อ้างอิงจากโครงสร้างของฐานข้อมูล หรือเป็นการพัฒนาเพื่อเพิ่มศักยภาพของออนโทโลยีที่สร้างได้ให้มีความฉลาดมากยิ่งขึ้น เช่น เพิ่มเติมการตั้งคำถามเพื่ออนุมานผลลัพธ์ หรือเป็นการพัฒนาต่อยอดในส่วนของกรนำออนโทโลยีไปใช้ประโยชน์ในด้านอื่นๆ

รายการอ้างอิง

บทความวารสาร

- ศุภกฤษฎี นิวัฒนากุล. (2556). การเข้าถึงความรู้ทางการเกษตรด้วยเทคโนโลยีเว็บเชิงความหมาย. นครราชสีมา:มหาวิทยาลัยสุรนารี.
- ทิพย์สุรีย์ ตีรวงศ์กุล, และ จินอ๊ี้ แซ่เฉิน (2551). ระบบสืบค้นสมุนไพรรไทย. กรุงเทพฯ: กระทรวงวิทยาศาสตร์และเทคโนโลยี
- วิเชษฐ์รุจน์ เอี่ยมสำอางค์. (2556). การแยกภาพตัวอักษรลายมือเขียนภาษาไทยแบบอัตโนมัติ. กรุงเทพมหานคร:มหาวิทยาลัยธรรมศาสตร์.

สื่ออิเล็กทรอนิกส์

- การเรียนการสอนผ่านเว็บ. (2550). หน่วยที่ 3 ระบบฐานข้อมูลเชิงสัมพันธ์.
สืบค้นจาก <http://203.172.182.81/wbidatabase/unit3/unit3.php>
- หน่วยที่ 4 ฐานข้อมูลเชิงสัมพันธ์. (2548). สืบค้นจาก <http://www.sttc.ac.th/~computerbc/backup/elearning/database/chapter4.pdf>
- การนอร์มัลไลซ์. (2546). สืบค้นจาก http://www.sut.ac.th/ist/Courses/204204/Lecture/204204_47_07.pdf

Articles

- Haiyun, L., & Shufeng, Z. (2014). Mapping Relational Database into OWL Ontology. *International Journal of Engineering and Technology (IJET)*, 5(6), 4735-4740.
- Jamal, B., & Mohamed, B. (2013). Generating of RDF graph from a relational database using Jena API. *International Journal of Engineering and Technology (IJET)*, 5(2), 1970-1975.
- Larbi, A., Oussama, E. H., & Mohamed, B. (2014). Automatic Mapping of Relational Database to OWL Ontology. *International Journal of Engineering Research & Technology (IJERT)*, 3(4), 1988-1994.

Wondu, Y. M., Farhi, M., & Vassil T. V. (2013). Algorithms for mapping RDB schema to RDF for facilitating access to deep web. *The First International Conference on Building and Exploring Web Based Environments*, 32-41.

Yutao, R., Lihong, J., Fenglin, B., & Hongming, C. (2012). Rules and implementation for generating ontology from relational database. *The Second International Conference on Cloud and Green Computing*, 237-243. DOI: 978-0-7695-4864-7/12.





ภาคผนวก

ภาคผนวก ก

แบบทดสอบถามข้อมูลผู้เข้าร่วมการทดลอง

คำชี้แจง โปรดทำเครื่องหมาย ✓ ลงในช่อง และ/หรือเติมข้อความที่ตรงตามความเป็นจริง
ลงในช่องว่างให้สมบูรณ์

1. เพศ ชาย หญิง

2. อายุ _____ ปี

3. การศึกษาสูงสุดของท่าน

ต่ำกว่าปริญญาตรี

ปริญญาตรี

ปริญญาโท

ปริญญาตรี

อื่นๆ ระบุ _____

4. อาชีพ

ข้าราชการ/รัฐวิสาหกิจ

พนักงานบริษัทเอกชน

ค้าขาย/ธุรกิจส่วนตัว

นักศึกษา

อื่นๆ ระบุ _____

5. ตำแหน่งงาน

นักวิชาการคอมพิวเตอร์

ผู้ดูแลระบบ

System Analyst (SA)

Programmer

Database Administrator

อื่นๆ ระบุ _____

6. ประสบการณ์ในการทำงาน

น้อยกว่า 2 ปี

2 - 5 ปี

มากกว่า 5 ปีขึ้นไป

7. ท่านมีความรู้ทางด้านฐานข้อมูลเชิงสัมพันธ์หรือไม่

มีความรู้

ไม่มีความรู้

หากท่านมีความรู้ทางด้านฐานข้อมูลเชิงสัมพันธ์ ท่านคิดว่าความรู้ของท่านอยู่ในระดับใด

เยี่ยมมาก เย่ พอใช้ ดี ต่ำมาก

8. ท่านมีความรู้ทางด้านออนไลน์หรือไม่

มีความรู้ ไม่มีความรู้

หากท่านมีความรู้ทางด้านออนไลน์ ท่านคิดว่าความรู้ของท่านอยู่ในระดับใด

เยี่ยมมาก เย่ พอใช้ ดี ต่ำมาก



ภาคผนวก ข

แบบทดสอบถามประเมินความพึงพอใจของผู้เข้าร่วมการทดลอง

คำชี้แจง โปรดทำเครื่องหมาย ✓ ลงในช่องแบบสอบถามที่ตรงกับระดับความคิดเห็นของท่านมากที่สุด โดยตัวเลขของระดับความพึงพอใจแต่ละด้านมีความหมายดังนี้

- 5 หมายถึง ความพึงพอใจในระดับมากที่สุด
 4 หมายถึง ความพึงพอใจในระดับมาก
 3 หมายถึง ความพึงพอใจในระดับปานกลาง
 2 หมายถึง ความพึงพอใจในระดับน้อย
 1 หมายถึง ความพึงพอใจในระดับน้อยที่สุด

ข้อ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
1	ความสามารถของระบบในด้านการเชื่อมต่อฐานข้อมูลเชิงสัมพันธ์					
2	ความสามารถของระบบในด้านการสร้าง Class แบบอัตโนมัติ					
3	ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Class แบบอัตโนมัติ					
4	ความสามารถของระบบในด้านการสร้าง Datatype Property แบบอัตโนมัติ					
5	ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Datatype Property แบบอัตโนมัติ					
6	ความสามารถของระบบในด้านการกำหนด Constraint ของ Datatype Property แบบอัตโนมัติ					
7	ความสามารถของระบบในด้านการสร้าง Object Property แบบอัตโนมัติ					
8	ความสามารถของระบบในด้านการกำหนดคุณสมบัติของ Object Property แบบอัตโนมัติ					

ข้อ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
9	ความสามารถของระบบในด้านการกำหนด Constraint ของ Object Property แบบอัตโนมัติ					
10	ความสามารถของระบบในด้านการสร้างออนโทโลยีจากฐานข้อมูลเชิงสัมพันธ์แบบอัตโนมัติ					
11	ความสามารถของระบบในด้านการตรวจสอบและปรับปรุงโครงสร้างของออนโทโลยี					
12	ความสามารถของระบบในการสร้างไฟล์ออนโทโลยี (File .OWL)					
13	ความง่ายต่อการใช้งานระบบ					
14	ความเหมาะสมของเนื้อหา					
15	ประสิทธิภาพโดยรวมของระบบ					
16	ความรวดเร็วในการตอบสนองของระบบ					
17	ความน่าเชื่อถือได้ของระบบ					
18	ความสวยงาม ความทันสมัยของระบบ					

ปัญหาที่พบ

ข้อเสนอแนะอื่นๆ เพิ่มเติม

ภาคผนวก ค

พจนานุกรมข้อมูลของตัวอย่างฐานข้อมูลเชิงสัมพันธ์

ตารางที่ ค.1 โครงสร้างข้อมูลตาราง Advance

Column	Datatype	Null	KEY	Table Ref	Description
adv_id	int	NO	PK		รหัส Advance
adv_date	varchar	NO			วันที่เบิก Advance
productID	int	NO			รหัสสินค้า
qty	int	NO			จำนวน
amount	double	NO			จำนวนเงินที่เบิก
clr_id	int	NO			รหัสการเคลียร์ Advane

ตารางที่ ค.2 โครงสร้างข้อมูลตาราง categories

Column	Datatype	Null	KEY	Table Ref	Description
CategoryID	int	NO	PK		รหัสกลุ่มสินค้า
CategoryName	varchar	NO			ชื่อกลุ่มสินค้า
Description	mediumtext	YES			รายละเอียด
Picture	longblob	YES			รูปภาพ

ตารางที่ ค.3 โครงสร้างข้อมูลตาราง clear_advance

Column	Datatype	Null	KEY	Table Ref	Description
clr_id	int	NO	PK		รหัสการเคลียร์ Advane
clr_date	datetime	NO			วันที่เคลียร์ Advance
adv_id	int	NO	FK	advance	รหัส Advance
amount	double	NO			จำนวนเงิน

ตารางที่ ค.4 โครงสร้างข้อมูลตาราง customercustomerdemo

Column	Datatype	Null	KEY	Table Ref	Description
CustomerID	varchar	NO	PK,FK	customers	รหัสลูกค้า
CustomerTypeID	varchar	NO	PK,FK	customerdemographics	รหัสกลุ่มลูกค้า

ตารางที่ ค.5 โครงสร้างข้อมูลตาราง customerdemographics

Column	Datatype	Null	KEY	Table Ref	Description
CustomerTypeID	varchar	NO	PK		รหัสกลุ่มลูกค้า
CustomerDesc	mediumtext	YES			ชื่อกลุ่มลูกค้า

ตารางที่ ค.6 โครงสร้างข้อมูลตาราง customers

Column	Datatype	Null	KEY	Table Ref	Description
CustomerID	varchar	NO	PK		รหัสลูกค้า
CompanyName	varchar	NO			ชื่อลูกค้า
ContactName	varchar	YES			ผู้ติดต่อ
ContactTitle	varchar	YES			ตำแหน่งชื่อผู้ติดต่อ
Address	varchar	YES			ที่อยู่
City	varchar	YES			อำเภอ
Region	varchar	YES			จังหวัด
PostalCode	varchar	YES			รหัสไปรษณีย์
Country	varchar	YES			ประเทศ
Phone	varchar	YES			เบอร์โทรศัพท์
Fax	varchar	YES			เบอร์ Fax

ตารางที่ ค.7 โครงสร้างข้อมูลตาราง employeeterritories

Column	Datatype	Null	KEY	Table Ref	Description
EmployeeID	int	NO	PK,FK	employees	รหัสพนักงาน
TerritoryID	varchar	NO	PK,FK	territories	รหัสสาขา

ตารางที่ ค.8 โครงสร้างข้อมูลตาราง territories

Column	Datatype	Null	KEY	Table Ref	Description
TerritoryID	varchar	NO	PK		รหัสสาขา
TerritoryDescription	varchar	NO			ชื่อสาขา
RegionID	int	NO	FK	region	รหัสภูมิภาค

ตารางที่ ค.9 โครงสร้างข้อมูลตาราง employees

Column	Datatype	Null	KEY	Table Ref	Description
EmployeeID	int	NO	PK		รหัสพนักงาน
LastName	varchar	NO			นามสกุลพนักงาน
FirstName	varchar	NO			ชื่อพนักงาน
Title	varchar	YES			คำนำหน้าชื่อพนักงาน
TitleOfCourtesy	varchar	YES			ชื่อทางการ
BirthDate	datetime	YES			วันเกิด
HireDate	datetime	YES			-
Address	varchar	YES			ที่อยู่
City	varchar	YES			อำเภอ
Region	varchar	YES			จังหวัด
PostalCode	varchar	YES			รหัสไปรษณีย์
Country	varchar	YES			ประเทศ
HomePhone	varchar	YES			เบอร์โทรศัพท์บ้าน
Extension	varchar	YES			รายละเอียดเพิ่มเติม
Photo	longblob	YES			ภาพถ่าย
Notes	mediumtext	YES			Note
ReportsTo	int	YES	FK	employees	หัวหน้างาน
PhotoPath	varchar	YES			ที่เก็บภาพถ่าย
Salary	float	YES			เงินเดือน

ตารางที่ ค.10 โครงสร้างข้อมูลตาราง region

Column	Datatype	Null	KEY	Table Ref	Description
RegionID	int	NO	PK		รหัสภูมิภาค
RegionDescription	varchar	NO			ชื่อภูมิภาค

ตารางที่ ค.11 โครงสร้างข้อมูลตาราง shippers

Column	Datatype	Null	KEY	Table Ref	Description
ShipperID	int	NO	PK		รหัสบริษัทขนส่ง
CompanyName	varchar	NO			ชื่อบริษัทขนส่ง
Phone	varchar	YES			เบอร์โทรศัพท์

ตารางที่ ค.12 โครงสร้างข้อมูลตาราง suppliers

Column	Datatype	Null	KEY	Table Ref	Description
SupplierID	int	NO	PK		รหัสบริษัทผู้จำหน่าย
CompanyName	varchar	NO			ชื่อบริษัทผู้จำหน่าย
ContactName	varchar	YES			ผู้ติดต่อ
ContactTitle	varchar	YES			คำนำหน้าชื่อผู้ติดต่อ
Address	varchar	YES			ที่อยู่
City	varchar	YES			อำเภอ
Region	varchar	YES			จังหวัด
PostalCode	varchar	YES			รหัสไปรษณีย์
Country	varchar	YES			ประเทศ
Phone	varchar	YES			เบอร์โทรศัพท์
Fax	varchar	YES			เบอร์ Fax
HomePage	mediumtext	YES			HomePage

ตารางที่ ค.13 โครงสร้างข้อมูลตาราง products

Column	Datatype	Null	KEY	Table Ref	Description
ProductID	int	NO	PK		รหัสสินค้า
ProductName	varchar	NO			ชื่อสินค้า
SupplierID	int	YES	FK	suppliers	รหัสบริษัทผู้จำหน่าย
CategoryID	int	YES	FK	categories	รหัสกลุ่มสินค้า
QuantityPerUnit	varchar	YES			จำนวนต่อหน่วย
UnitPrice	decimal	YES			ราคาต่อหน่วย
UnitsInStock	smallint	YES			จำนวนใน Stock
UnitsOnOrder	smallint	YES			จำนวนที่สั่งซื้อ
ReorderLevel	smallint	YES			-
Discontinued	bi	NO			ส่วนลด

ตารางที่ ค.14 โครงสร้างข้อมูลตาราง products_location

Column	Datatype	Null	KEY	Table Ref	Description
ProductLocalID	int	NO	PK,FK	products	รหัสสินค้า
Address	varchar	NO			ที่อยู่
City	varchar	NO			อำเภอ
PostCode	varchar	NO			รหัสไปรษณีย์
Country	varchar	NO			ประเทศ

ตารางที่ ค.15 โครงสร้างข้อมูลตาราง orders

Column	Datatype	Null	KEY	Table Ref	Description
OrderID	int	NO	PK		รหัสการสั่งซื้อ
CustomerID	varchar	YES	FK	customers	รหัสลูกค้า
EmployeeID	int	YES	FK	employees	รหัสพนักงาน
OrderDate	datetime	YES			วันที่สั่งซื้อ
RequiredDate	datetime	YES			วันที่ต้องการสินค้า
Freight	decimal	YES			เลขที่จัดส่ง
ShipName	varchar	YES			ชื่อผู้รับสินค้า
ShipAddress	varchar	YES			ที่อยู่จัดส่ง
ShipCity	varchar	YES			อำเภอจัดส่ง
ShipRegion	varchar	YES			จังหวัดจัดส่ง
ShipPostalCode	varchar	YES			รหัสไปรษณีย์จัดส่ง
ShipCountry	mediumtext	YES			ประเทศจัดส่ง

ตารางที่ ค.16 โครงสร้างข้อมูลตาราง orderdetails

Column	Datatype	Null	KEY	Table Ref	Description
OrderID	int	NO	PK,FK	orders	รหัสการสั่งซื้อ
ProductID	int	NO	PK,FK	products	รหัสสินค้า
UnitPrice	decimal	NO			ราคา
Quantity	smallint	NO			จำนวน
Discount	double	NO			ส่วนลด

ตารางที่ ค.17 โครงสร้างข้อมูลตาราง shipping

Column	Datatype	Null	KEY	Table Ref	Description
OrderID	int	NO	PK,FK	orders	รหัสการสั่งซื้อ
ProductID	int	NO	PK,FK	products	รหัสสินค้า
ShipVia	int	NO	PK,FK	shippers	รหัสบริษัทขนส่ง
ShippingDate	datetime	NO	PK		วันที่ส่งสินค้า
Qantity	int	NO			จำนวน



ประวัติผู้เขียน

ชื่อ นางสาววาสนา ธิบแจ่ม

วันเดือนปีเกิด 2 สิงหาคม 2529

วุฒิการศึกษา ปีการศึกษา: 2551 เทคโนโลยีบัณฑิต
(ตั้งแต่ระดับปริญญาตรี) (เทคโนโลยีสารสนเทศ)

ตำแหน่ง มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
โปรแกรมเมอร์
บริษัทสุพรีมไฮเทคร่า จำกัด

ผลงานทางวิชาการ

วาสนา ธิบแจ่ม และ รัชฎา คงคะจันทร์. (กันยายน 2558). The Constraint Extension and Recommend for Generate from Relational Database. งานประชุมวิชาการนานาชาติ 2015 International Symposium on Multimedia and Communication Technology (ISMAT), พระนครศรีอยุธยา.