สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ และการประยุกต์ใช้งาน GIS

Object Relational Database and GIS

T037040

โดย

นาย ศักดิ์ชาติ งามล้วน 39014510

นาย สุระศักดิ์ พณิชย์วิวัฒน์ 39014616

อาจารย์ที่ปรึกษา

รศ.ดร. สุภมิตร จิตตะยโศธร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 37040
วัน, เดือน, ปี 30 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ และการประยุกต์ใช้งาน GIS

Object Relational Database and GIS

ผู้จัดทำ

1. นาย ศักดิ์ชาติ งามล้วน 39014510
2. นาย สุระศักดิ์ พณิชย์วิวัฒน์ 39014616

อาจารย์ที่ปรึกษา

(รศ.ดร. ศุภมิตร จิตตะยโศธร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ และการประยุกต์ใช้งาน GIS

นายศักดิ์ชาติ งามล้วน 39014510
นายสุระศักดิ์ พณิชย์วิวัฒน์ 39014616
รศ.ดร. ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ในช่วงทศวรรษที่ผ่านมามีแอปพลิเคชั่นที่ใช้ข้อมูลเชิงพื้นที่หรือเชิงภูมิศาสตร์มากขึ้นเรื่อยๆ ปัจจุบันนี้ ผู้ผลิตซอฟต์แวร์ระบบจัดการฐานข้อมูล 2-3 รายได้เพิ่มโมดูลที่สนับสนุนการพัฒนาฐานข้อมูลเชิงภูมิศาสตร์ เช่น โอราเคิล, อินฟอร์มิกซ์ และ ดีบีทู

ในโครงงานนี้ เราได้ศึกษาโมดูล Spatial Cartridge ของโอราเคิล 8.05 ซึ่งใช้รูปแบบเชิงสัมพันธ์ และ โมดูล Spatial Cartridge ของโอราเคิล 8i ซึ่งใช้รูปแบบเชิงวัตถุสัมพันธ์ กรณีศึกษาของรูปแบบเชิงสัมพันธ์ได้ใช้ ระบบสารสนเทศอาคารและสถานที่ของ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งพัฒนาโดยใช้ภาษาจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Object Relational Database and GIS

Sakshart Ngamluan
Surasak Panichwiwat
Assoc. Prof. Dr. Suphamit chittayasothon
Advisor

ABSTRACT

The number of applications using spatial or geographic data has been ever increasing over the last decade. There are 2-3 DBMS vendors that add a module make the DBMS supported the development of geographic database such as Oracle, Informix and DB2.

In this project, we introduce Spatial Cartridge module of Oracle 8.0.5 that use Relational Model and Spatial Cartridge module of Oracle 8i that use Object-Relational Model. A case study of Relational Model is given using a Building and Place Information Systems of KMITL application using JAVA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานชิ้นนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี ก็เนื่องมาจากการได้รับความช่วยเหลือและการสนับสนุนจากบุคคลหลาย ๆ ท่าน

ขอขอบพระคุณ บิดา มารดา ที่ได้อบรมสั่งสอนและให้การเลี้ยงดูมาอย่างดีที่สุด รวมทั้งให้ความสนับสนุนในทุก ๆ เรื่อง

ขอขอบพระคุณ ครูบาอาจารย์ทุกท่านที่ได้อบรมสั่งสอนทั้งทางด้านวิชาการและจริยธรรม

ขอขอบพระคุณ รศ.ดร.ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษาที่คอยให้คำชี้แนะ และคำแนะนำต่างๆ

ขอขอบพระคุณ อาจารย์ บัณฑิต พัสยา ที่ช่วยให้คำปรึกษา และ ความช่วยเหลือต่างๆ เป็นอย่างดี

ขอขอบคุณ คุณนรินทร์ ธรรมาวัฒนะ และ คุณณัฐฐา บุญอยู่ สำนักวิจัยและบริการคอมพิวเตอร์ สจล. ที่คอยให้คำปรึกษา และ ความช่วยเหลือต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เนื้อหา	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ความรู้พื้นฐานของระบบสารสนเทศทางภูมิศาสตร์ (Geographic Information System : GIS)	3
2.1 องค์ประกอบพื้นฐานของระบบสารสนเทศเชิงพื้นที่	4
2.2 ขั้นตอนการสร้างฐานข้อมูลเชิงภูมิศาสตร์ (Geographic Database Creation Process)	9
2.3 ข้อมูลทางภูมิศาสตร์	10
2.4 การทำงานของระบบสารสนเทศทางภูมิศาสตร์	12
2.5 การนำระบบสารสนเทศทางภูมิศาสตร์ มาประยุกต์ใช้งานได้ด้านต่างๆ	14
2.6 ประโยชน์ของระบบ GIS	15
บทที่ 3 แนวความคิดเชิงวัตถุ (Object-Oriented Concept)	17
3.1 ออบเจกต์ (Object)	17
3.2 คลาส (Class)	17
3.3 โพลิมอร์ฟิซึม (Polymorphism)	17
3.4 อินเฮอริแตนซ์ (Inheritance)	18
3.5 เอ็นแคปซูเลชัน (Encapsulation)	18
3.6 การนำแนวความคิดเชิงวัตถุมาใช้ในโครงงาน	19
บทที่ 4 ทฤษฎีของโอราเคิล 8 ที่สนับสนุน ORDBMS	20
4.1 Built-In Datatype	21
4.2 User-Defined Datatype	23
4.3 คุณสมบัติทางออบเจกต์โอเรียนเต็ดของออบเจกต์ไทป์ ในโอราเคิล 8	28
บทที่ 5 หลักการของ Oracle8 Spatial Cartridge	29
5.1 หลักการของ Oracle8 Spatial Cartridge สำหรับโมเดลเชิงสัมพันธ์ (โอราเคิล 8.0.5)	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 หลักการของ Oracle8 Spatial Cartridge สำหรับโมเดลเชิงวัตถุสัมพันธ์ (โอราเคิล 8i) 43
5.2 ข้อเปรียบเทียบการดูแลข้อมูลเชิงพื้นที่ระหว่างโอราเคิล 8.0.5 และ โอราเคิล 8i 57
บทที่ 6 การเชื่อมต่อฐานข้อมูลด้วยเจดีบีซี (Java Database Connectivity) 58
6.1 ความหมายของเจดีบีซี 60
6.2 รูปแบบการเชื่อมต่อฐานข้อมูลของเจดีบีซี 61
6.3 โครงสร้างของเจดีบีซี 62
6.4 รูปแบบของเจดีบีซีไดร์ฟเวอร์ 63
6.5 ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล 68
บทที่ 7 การพัฒนาต้นแบบระบบสารสนเทศอาคารและสถานที่ของสจล. 70
7.1 ภาพรวมของโครงงาน 70
7.2 ฟังก์ชันการทำงานของระบบ 70
7.3 กลุ่มผู้ใช้เป้าหมาย 70
7.4 เงื่อนไขทั่วไป 70
7.5 ข้อกำหนดความต้องการของระบบ 71
7.6 ความต้องการด้านประสิทธิภาพ (Performance Requirement) 72
7.7 คุณสมบัติ (Attribute) 72
7.8 การสร้างระบบ 72
7.9 ซอฟต์แวร์ที่ใช้ 80
7.10 แนวทางการพัฒนาระบบต่อ โดยมีการจัดเก็บข้อมูลในโมเดลเชิงวัตถุสัมพันธ์ 80
บทที่ 8 บทสรุปและวิจารณ์ 81
8.1 สรุปผลการทำงานโครงการ 81
8.2 ปัญหาที่เกิดขึ้น 81
8.3 แนวทางการพัฒนาโครงงานวิจัยต่อ 82
8.4 วิจารณ์ 82
ภาคผนวก ก การแก้ไขและการนำเข้าข้อมูลแผนที่ 83
ก.1 ขั้นตอนการแก้ไขและนำเข้าข้อมูลแผนที่ชั้นต่างๆ ภายในอาคาร 83
ก.2 ขั้นตอนการแก้ไขและนำเข้าข้อมูลแผนที่สถาบัน 94
ก.3 การเพิ่มหรือลดเลเยอร์ที่แสดงในแผนที่ 95
ภาคผนวก ข คู่มือการติดตั้งและใช้งานโปรแกรม 96
ข.1 การติดตั้งโปรแกรม 96
ข.2 ส่วนประกอบของโปรแกรม 97
ข.3 การใช้งานโปรแกรม 99
บรรณานุกรม 105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
2-1 เปรียบเทียบข้อดี ข้อเสียของการจัดเก็บข้อมูลแบบเวคเตอร์และแบบตารางกริด	11
5-1 คอลัมน์ต่าง ๆ ของตาราง <layername>_sdolayer	30
5-2 คอลัมน์ต่าง ๆ ของตาราง <layername>_sdodim	31
5-3 คอลัมน์ต่าง ๆ ของตาราง <layername>_sdogeom	31
5-4 คอลัมน์ต่าง ๆ ของตาราง <layername>_sdoindex	31
5-5 ตัวอย่างข้อมูลในตาราง <layername>_sdolayer	33
5-6 ตัวอย่างข้อมูลในตาราง <layername>_sdodim	33
5-7 ตัวอย่างข้อมูลในตาราง <layername>_sdogeom	34
5-8 ตัวอย่างข้อมูลในตาราง <layer_name>_sdoindex	39
5-9 ค่าต่างๆ ที่เป็นไปได้ของ SDO_GTYPE	45
5-10 ค่าต่างๆ และความหมายของ SDO_ETYPE และ SDO_INTERPRETATION	46
5-11 คอลัมน์ต่าง ๆ ในตาราง SDO_INDEX_METADATA	50
5-12 คอลัมน์ในตารางข้อมูลอินเด็กซ์	50
5-13 ความหมายของค่า SDO_LEVEL และ SDO_NUMTILES	53
5-14 เปรียบเทียบการดูแลข้อมูลเชิงพื้นที่ระหว่างโอราเคิล 8.0.5 และ โอราเคิล 8i	57
7-1 เดต้าดิคชันนารี	75
7-2 ตารางที่ช่วยในการแม็พระหว่างอาคารกับชั้นต่าง ๆ ภายในอาคาร	77
7-3 โครงสร้างตารางที่ช่วยในการคอนฟิกระบบ	78
7-4 ข้อมูลในตาราง GISDESKTOP_CONFIG	78
7-5 ความหมายของแต่ละ TABLE_KEY	78
7-6 โครงสร้างตาราง MAINMAP	79
7-7 โครงสร้างของตาราง Kmitl_Building โดยจัดเก็บข้อมูลในโมเดลเชิงวัตถุสัมพันธ์	79
ก-1 ตารางข้อมูลแอตทริบิวต์ที่ต้องป้อนสำหรับห้อง	92
ก-2 โครงสร้างตาราง MAINMAP	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6-2 ทรีเทียร์โทเคล 62
6-3 ระดับการเชื่อมต่อของเจดีบีซีเอพีไอ 62
6-4 ประเภทของเจดีบีซีไดร์ฟเวอร์ 63
6-5 การติดต่อเจดีบีซีโดยรูปแบบที่ 1 JDBC/ODBC Bridge 64
6-6 การติดต่อเจดีบีซี โดยรูปแบบที่ 2 Native-API (Partly java) Driver 65
6-7 การติดต่อเจดีบีซีโดยรูปแบบที่ 3 Network-Protocol (All-Java) Driver 66
6-8 การติดต่อเจดีบีซีโดยรูปแบบที่ 4 Native-Protocol (All-Java) Driver 67
7-1 โครงสร้างของระบบ 73
7-2 โครงสร้างการเชื่อมต่อของระบบ 73
7-3 อีอาร์ไดอะแกรม 74
7-4 เลเยอร์ที่ใช้แสดงภาพกราฟฟิกและเลเยอร์ที่ช่วยในการคิวรี่ 76
ก-1 หน้าต่าง Layer & Linetype Properties 83
ก-2 การเลือกข้อมูลแผนที่ห้องในส่วนที่สนใจและทำการคัดลอก 84
ก-3 หน้าต่างการสร้างรูปวาดใหม่ 84
ก-4 หน้าต่างการตั้งค่าต่างๆ สำหรับรูปวาดใหม่ 85
ก-5 หน้าจอการป้อนค่าต่างๆ หลังจากนำรูปที่คัดลอกไว้มาวาง 85
ก-6 โปรแกรม FME 86
ก-7 หน้าต่าง Set Translation Parameters 86
ก-8 หน้าต่างการแปลงค่าพิกัด 87
ก-9 หน้าจอเริ่มต้นของโปรแกรม ArcView GIS 3.1 87
ก-10 หน้าต่างยืนยันการเพิ่มข้อมูลลงใน View 88
ก-11 หน้าต่างการเพิ่ม Theme 88
ก-12 การทำเครื่องหมายถูกหน้าเลเยอร์ที่ต้องการ 89
ก-13 การเลือกชนิดของรูปทรงเรขาคณิตพื้นฐาน 89
ก-14 การเริ่มต้นแก้ไขข้อมูล 89
ก-15 หน้าต่างการเปิดโปรเจ็คของโปรแกรม ArcView 90
ก-16 การเลือกเลเยอร์ให้ปุ่มนูนขึ้นและเริ่มต้นการแก้ไขข้อมูล 90
ก-17 หน้าต่างตารางของเลเยอร์ En08_03_fence 91
ก-18 หน้าต่างการเพิ่มฟิลด์ใหม่ 91
ก-19 การป้อนข้อมูลแอตทริบิวต์ 92
ก-20 หน้าต่าง Set Translation Parameters การแปลง Shape File ไปเก็บในฐานข้อมูลโอราเคิล 93
ก-21 หน้าต่างการแปลงค่าพิกัด 93
ข-1 ส่วนประกอบของโปรแกรม 97
ข-2 ส่วนแสดงรายการเลเยอร์ 98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข-3 การดูตารางข้อมูลแอตทริบิวต์ของเลเยอร์ที่สนใจ 99
ข-4 หน้าต่างผลลัพธ์ที่ได้จากการเลือกดูข้อมูลแอตทริบิวต์ของวัตถุที่สนใจ 100
ข-5 หน้าต่างการสืบค้นข้อมูล 100
ข-6 ผลลัพธ์จากการคิวรี่ 101
ข-7 ผลลัพธ์ที่ได้จากการคิวรี่ข้อมูลของห้องภายในอาคาร 102
ข-8 หน้าต่างการปรับเลือกสี 102
ข-9 หน้าต่างแสดงผลแบบ Windows 103
ข-10 หน้าต่างแสดงผลแบบ Motif 103
ข-11 หน้าต่างแสดงผลแบบ Java 104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

เนื่องจากปัจจุบันนี้มีความต้องการใช้ข้อมูลปริมาณมาก และต้องมีการจัดเก็บข้อมูลที่มีความซับซ้อนมากยิ่งขึ้น เช่น ข้อมูล,เสียง,ภาพ,วิดีโอ,ข้อมูลมัลติมีเดีย (Multimedia) และข้อมูลเชิงพื้นที่ (Spatial data) เป็นต้น การจัดการกับข้อมูลเหล่านี้ก็มีความยุ่งยากและซับซ้อนมากยิ่งขึ้นเช่นเดียวกัน ระบบฐานข้อมูลเชิงสัมพันธ์(Relational Database Management System : RDBMS) มีความยุ่งยากในการจัดการและจัดเก็บข้อมูลเหล่านี้ จึงได้ทำการศึกษาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object Relational Database Management System : ORDBMS) เนื่องจากระบบฐานข้อมูลแบบนี้มีความเหมาะสมในการรองรับข้อมูลที่มีความซับซ้อนมาก ๆ ได้ดีกว่าระบบฐานข้อมูลเชิงสัมพันธ์ ช่วยให้ง่ายในการจัดการกับข้อมูลที่มีความซับซ้อน และช่วยให้นักพัฒนาสามารถพัฒนาได้ง่ายยิ่งขึ้น

ระบบสารสนเทศทางภูมิศาสตร์ (Geographic Information System : GIS) เป็นเครื่องมือในระบบคอมพิวเตอร์สำหรับการทำแผนที่และการวิเคราะห์สิ่งต่าง ๆ ที่มีอยู่และเหตุการณ์ที่เกิดขึ้นบนโลก เทคโนโลยีของ GIS รวมเอาการทำงานของฐานข้อมูล เช่น การสอบถามและการวิเคราะห์เชิงสถิติเข้ากับมุมมองที่ไม่ซ้ำแบบและผลประโยชน์ในการวิเคราะห์เชิงภูมิศาสตร์ที่ได้รับจากแผนที่ ความสามารถเหล่านี้แยก GIS จากระบบข้อมูลแบบอื่นและทำให้มีค่าต่อสาธารณะและเอกชนอย่างกว้างขวางในการอธิบายเหตุการณ์ต่าง ๆ การคาดคะเนผลที่จะเกิดขึ้นและกลยุทธ์ในการวางแผนการทำแผนที่และวิเคราะห์ในเชิงภูมิศาสตร์ไม่ใช่เป็นของใหม่แต่ GIS สามารถทำงานเหล่านี้ได้ดีกว่าและเร็วกว่าวิธีเก่า

ในโครงงานนอกจากจะศึกษาหัวข้อดังกล่าวข้างต้นแล้ว ยังได้ทำการพัฒนาต้นแบบระบบสารสนเทศอาคารและสถานที่ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) โดยใช้โอราเคิล 8.0.5 มาจัดเก็บและจัดการข้อมูล ในโอราเคิล 8.0.5 จะมี Spatial Cartridge ซึ่งได้รวบรวมฟังก์ชัน (Function) และ โพรซิเดอร์ (Procedure) ที่สามารถจัดการ,จัดเก็บ และวิเคราะห์ กับข้อมูลเชิงพื้นที่ แต่อย่างไรก็ตามระบบที่พัฒนานี้ยังคงมีการจัดเก็บข้อมูลเชิงสัมพันธ์อยู่ ซึ่งหากต้องการที่จะพัฒนาต่อให้มีการจัดเก็บข้อมูลเชิงวัตถุสัมพันธ์ต่อ ปริญญานิพนธ์ฉบับนี้ได้เสนอแนวทางไว้แล้วในบทที่ 7

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 ศึกษาความรู้พื้นฐานของระบบสารสนเทศเชิงภูมิศาสตร์

1.2.2 ศึกษาหลักการเชิงวัตถุ (Object Oriented Concept)

1.2.3 ศึกษาคุณสมบัติความเป็นออบเจกต์ของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์(โอราเคิล 8.0.5)

1.2.4 ศึกษาการจัดเก็บข้อมูลเชิงพื้นที่โดยใช้ Spatial Cartridge ของ โอราเคิล

1.2.5 ต้นแบบระบบสารสนเทศอาคารและสถานที่ของสจล.โดยใช้ Spatial Cartridge ในโมเดลเชิงสัมพันธ์ (Relational)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.6 ศึกษาการจัดเก็บข้อมูลเชิงพื้นที่ในโมเดลเชิงวัตถุสัมพันธ์ (Object Relational) ของโอราเคิล 8i

1.2.7 เปรียบเทียบความแตกต่างของ Spatial Cartridge ในโอราเคิล 8.0.5 และ โอราเคิล 8i

1.2.8 เสนอแนวทางการพัฒนาการจัดเก็บข้อมูลในโมเดลเชิงวัตถุสัมพันธ์

1.3 ขอบเขตของงานวิจัย

ขอบเขตของโครงงานวิจัยในภาคการศึกษานี้อยู่ในวัตถุประสงค์ข้อที่ 1.2.1 –1.2.8

1.4 วิธีการดำเนินงาน

โครงงานวิจัยนี้ได้เริ่มศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลัก ๆ ดังนี้

- ความรู้พื้นฐานของระบบสารสนเทศเชิงภูมิศาสตร์
- หลักการเชิงวัตถุ (Object Oriented Concept)
- คุณสมบัติความเป็นออบเจกต์ของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์(โอราเคิล 8.0.5)
- การจัดเก็บข้อมูลเชิงพื้นที่โดยใช้ Spatial Cartridge ของ โอราเคิล 8.0.5 และ โอราเคิล 8i
- การเชื่อมต่อฐานข้อมูลโอราเคิล โดยใช้จาวา (JAVA)

จากนั้นทำการพัฒนาต้นแบบระบบสารสนเทศอาคารและสถานที่ของสจล. โดยใช้ Spatial Cartridge ในโมเดลเชิงสัมพันธ์ (Relational)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

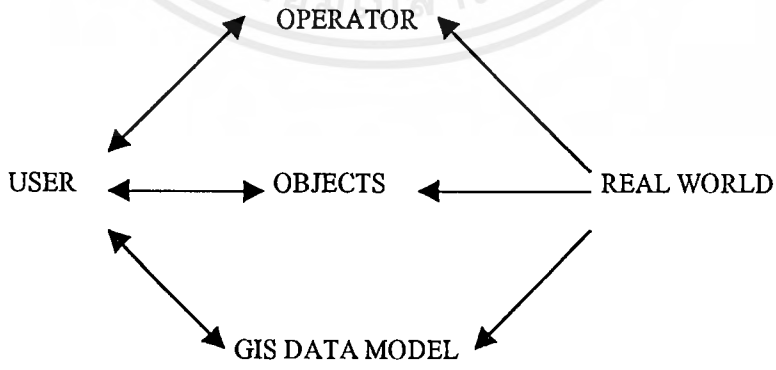
บทที่ 2
ความรู้พื้นฐานของระบบสารสนเทศทางภูมิศาสตร์
(Geographic Information Systems : GIS)

ระบบสารสนเทศทางภูมิศาสตร์ ประกอบด้วยกระบวนการต่างๆ ทางคอมพิวเตอร์ ระบบโปรแกรม สารสนเทศทางภูมิศาสตร์และบุคลากรที่ทำหน้าที่ในการจัดรวบรวมข้อมูลสารสนเทศที่ต้องการเพื่อทำการดัดแปลงเข้าจัดเก็บในระบบ การปรับปรุง การจัดการ การวิเคราะห์และการแสดงสารสนเทศเหล่านั้นในรูปแบบที่มีการอ้างอิงพิกัดทางภูมิศาสตร์ได้ตามต้องการ

การจัดการข้อมูลที่มีลักษณะเป็นสารสนเทศเชิงพื้นที่ (Spatial Information) หมายถึงข้อมูลทางพื้นที่ทุกอย่างที่มีอยู่ในระบบสารสนเทศทางภูมิศาสตร์ สามารถที่จะอ้างอิงได้กับข้อมูลบนพื้นโลกทั้งทางตรงและทางอ้อม

ข้อมูลทางภูมิศาสตร์ที่อ้างอิงกับพื้นผิวโลกโดยตรง หมายถึง ข้อมูลที่มีค่าพิกัดหรือตำแหน่งจริงบนพื้นโลกหรือในแผนที่ เช่น ตำแหน่งอาคาร ถนน ทะเลสาบ เป็นต้น สำหรับข้อมูลทางภูมิศาสตร์ ที่อ้างอิงกับข้อมูลบนพื้นผิวโลกได้โดยทางอ้อมได้แก่ ข้อมูลของบ้านเรือนที่มีอยู่เช่น บ้านเลขที่ ซอย เขต แขวง จังหวัดและรหัสไปรษณีย์ เป็นต้น

ระบบ GIS เหมือนกับการสร้างแบบจำลองของข้อมูลจริงๆ บนพื้นโลก (Data Model) โดยที่จะมีโปรแกรมเป็นผู้ประมวลผลข้อมูล (Operator) จากการประมวลผลก็จะได้ข้อมูลใหม่ที่แทนข้อมูลจริงที่อยู่บนพื้นโลก ดังนั้นในระบบ GIS จึงทำให้เราสามารถทราบถึงสิ่งต่างๆ ที่มีอยู่หรือที่จะเกิดขึ้นบนพื้นโลกจริง GIS จึงเปรียบเสมือนเครื่องมือที่ใช้ในการประมวลข้อมูลทางภูมิศาสตร์ ซึ่งอาจจะอยู่ในรูปของแผนที่หรือข้อมูลอื่นๆ เพื่อให้ได้รายละเอียดเกี่ยวกับข้อมูลที่แตกต่างออกไปเพื่อตอบสนองความต้องการในแต่ละงาน



รูปที่ 2-1 แบบจำลองของระบบ GIS

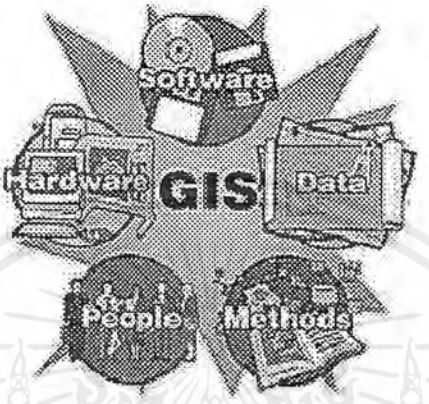
หลังจากที่ได้นำเข้าข้อมูลที่จำเป็นต่อการทำงานในแต่ละงาน และผ่านการวิเคราะห์ตามขั้นตอนของแต่ละงาน ก็จะได้ผลลัพธ์ที่เป็นข้อมูลที่มีรายละเอียดที่แตกต่างไปจากข้อมูลที่ได้นำเข้าไปแล้วซึ่งจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นข้อมูลที่แต่ละงานนั้นต้องการ จึงกล่าวได้ว่าระบบ GIS เปรียบเสมือนการนำเข้าข้อมูลต่างๆ ที่อยู่บนพื้นโลกจริงมาผสมผสานเข้าด้วยกัน

2.1 องค์ประกอบพื้นฐานของระบบสารสนเทศเชิงพื้นที่

องค์ประกอบหลักของระบบ GIS อาจจะแบ่งออกเป็น 5 ประเภทหลักๆ ได้ดังต่อไปนี้

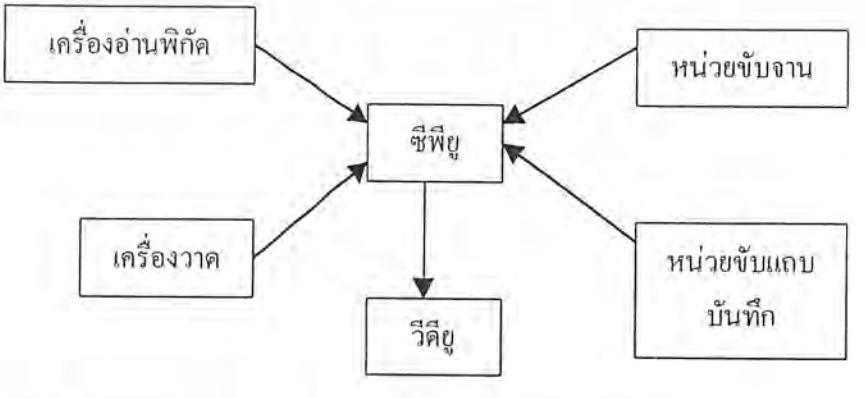


รูปที่ 2-2 ส่วนประกอบของระบบสารสนเทศทางภูมิศาสตร์

2.1.1 ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ที่ใช้ในระบบสารสนเทศทางภูมิศาสตร์ประกอบด้วย

- เครื่องคอมพิวเตอร์หรือหน่วยประมวลผลกลาง (CPU) ตั้งแต่ขนาดเล็กคือไมโครคอมพิวเตอร์จนถึงเครื่องคอมพิวเตอร์ขนาดใหญ่เมนเฟรม (Mainframe)
- อุปกรณ์นำเข้าข้อมูล (Input Device) ได้แก่หน่วยขับจาน (Disk Drive Unit) หน่วยขับแถบบันทึก (Tape Drive Unit) เครื่องอ่านพิกัด (Digitizer) เครื่องสแกนเนอร์ (Scanner)
- อุปกรณ์แสดงผล (Output Device) ได้แก่ จอภาพ (Monitor) , เครื่องวาด (Plotter) , เครื่องพิมพ์ (Printer) เป็นต้น



รูปที่ 2-3 องค์ประกอบด้านฮาร์ดแวร์ที่สำคัญของระบบสารสนเทศทางภูมิศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบของฮาร์ดแวร์ทั่วไปที่ใช้สำหรับระบบสารสนเทศทางภูมิศาสตร์แสดงอยู่ในรูปที่ 2-3 คอมพิวเตอร์หรือหน่วยประมวลผลกลางเชื่อมต่อกับหน่วยขับจาน ซึ่งมีที่ว่างสำหรับเก็บข้อมูลและโปรแกรม เครื่องอ่านพิกัด หรืออุปกรณ์อื่น ๆ ใช้สำหรับแปลงข้อมูลจากแผนที่และเอกสารเป็นรูปดิจิทัล แล้วส่งไปยังเครื่องคอมพิวเตอร์ เครื่องวาดรูป หรือเครื่องมือแสดงผลแบบอื่น ๆ เพื่อนำเสนอผลจากการประมวลผลข้อมูล ส่วนเครื่องขับแถบบันทึกใช้สำหรับการบันทึกข้อมูลหรือโปรแกรมบนแถบบันทึกแม่เหล็ก หรือสำหรับการสื่อสารกับระบบอื่น ๆ การสื่อสารระหว่างคอมพิวเตอร์สามารถทำได้โดยผ่านทางระบบเครือข่ายของสายรับส่งข้อมูลพิเศษหรือของสายโทรศัพท์ โดยใช้อุปกรณ์ที่เรียกว่าโมเดม (Modem) ผู้ใช้ควบคุมคอมพิวเตอร์และอุปกรณ์ต่อพ่วง โดยผ่านทางจอ (VDU) หรือบางครั้งเรียกว่าเครื่องปลายทาง (Terminal) เครื่องปลายทางของผู้ใช้อาจเป็นไมโครคอมพิวเตอร์เอง หรืออาจรวมฮาร์ดแวร์พิเศษเพื่อให้แสดงแผนที่ได้รวดเร็ว มีเครื่องมืออีกมากมายที่สามารถนำมาเสริมในส่วนของฮาร์ดแวร์เพื่อสนองความต้องการของผู้ใช้

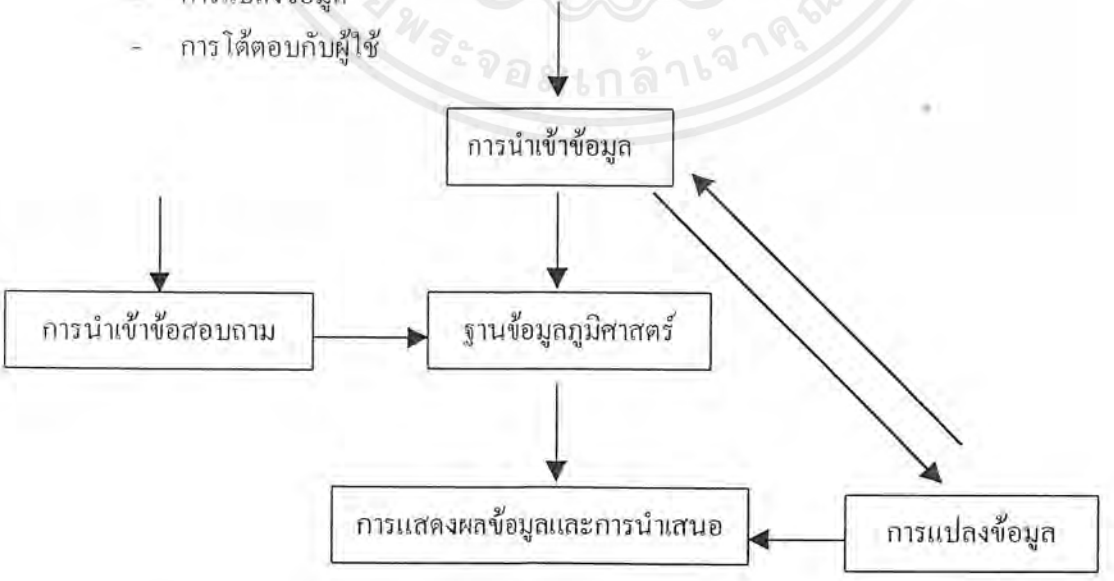
2.1.2 ซอฟต์แวร์ (Software)

เป็นโปรแกรมที่ใช้ในการจัดการข้อมูลในระบบ GIS ตามความต้องการของผู้ใช้ โปรแกรมที่ใช้กันแพร่หลายในปัจจุบันได้แก่ โปรแกรมอาร์คอินโฟ (ARC/INFO) , โปรแกรมสแปน (SPANS) , โปรแกรมพาแมพ (PAMAP) , โปรแกรมเจเนซิส (GENESIS) ฯลฯ

2.1.3 ขั้นตอนการทำงาน (Procedure)

การทำงานของโปรแกรมสำหรับระบบสารสนเทศทางภูมิศาสตร์ ประกอบด้วยโมดูล (Module) ทางเทคนิคพื้นฐาน 5 หน่วย ได้แก่

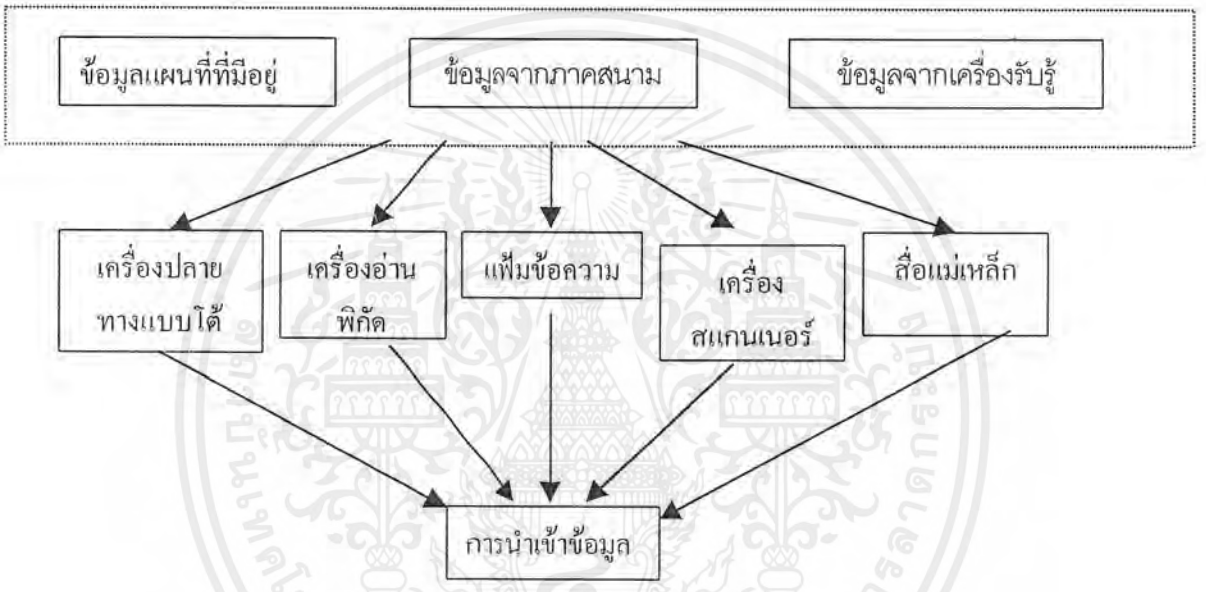
- การนำเข้าข้อมูล และตรวจสอบความถูกต้อง
- การเก็บข้อมูล และจัดการฐานข้อมูล
- การแสดงผลข้อมูลและการนำเสนอ
- การแปลงข้อมูล
- การโต้ตอบกับผู้ใช้



รูปที่ 2-4 องค์ประกอบด้านซอฟต์แวร์หลักๆ ของระบบสารสนเทศทางภูมิศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

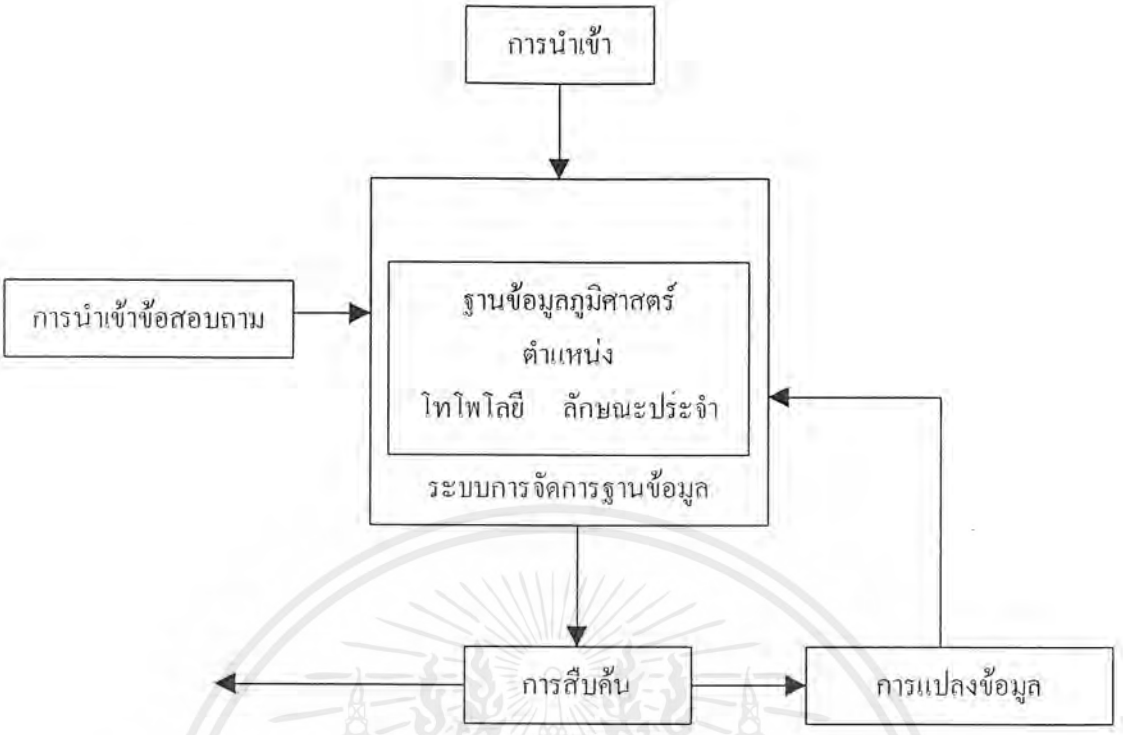
2.1.3.1 การนำเข้าข้อมูล หมายรวมถึงการแปลงข้อมูลทุกรูปแบบซึ่งอาจได้จากแผนที่ การสำรวจภาคสนาม เครื่องรับรู้ (ซึ่งรวมถึงภาพถ่ายทางอากาศ ดาวเทียม และเครื่องบันทึก) ให้อยู่ในรูปดิจิทัลที่เข้ากันได้ มีเครื่องมือทางคอมพิวเตอร์หลายอย่างซึ่งผลิตขึ้นเพื่อวัตถุประสงค์นี้ ได้แก่ เครื่องปลายทางแบบโต้ตอบ จอภาพ เครื่องอ่านพิกัด รายการข้อมูลในแฟ้มข้อความ เครื่องสแกนเนอร์ (ในดาวเทียมหรือในเครื่องบินใช้สำหรับบันทึกข้อมูลโดยตรง แต่เครื่องสแกนเนอร์ก็ใช้สำหรับแปลงข้อมูลแผนที่และภาพถ่ายได้ด้วย) หรือเครื่องมือที่ใช้ในการบันทึกข้อมูลซึ่งบรรจุอยู่ในสื่อแม่เหล็ก เช่น เทป ดรัม และจานแม่เหล็ก



รูปที่ 2-5 องค์ประกอบของข้อมูลที่เข้าไปสู่ฐานข้อมูล

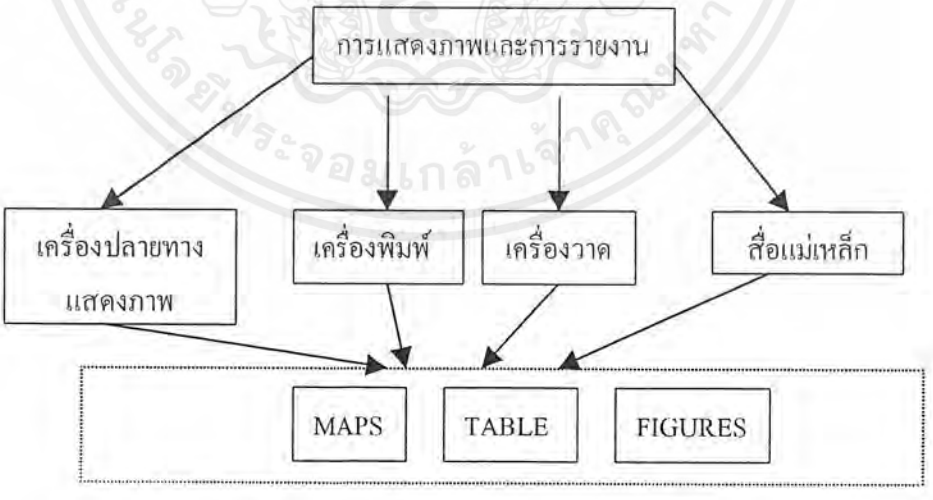
2.1.3.2 การเก็บข้อมูลไว้ในคอมพิวเตอร์และการจัดการฐานข้อมูล (รูปที่ 2-6) เป็นเรื่องของวิธีการซึ่งใช้กับข้อมูลเกี่ยวกับตำแหน่ง ความเชื่อมโยง (โทโพโลยี) และลักษณะประจำต่าง ๆ ขององค์ประกอบทางภูมิศาสตร์ (จุด เส้น พื้นที่ ซึ่งใช้แทนสิ่งต่างๆบนพื้นผิวโลก) โดยข้อมูลเหล่านี้จะถูกจัดเป็นโครงสร้างและเป็นระบบให้สอดคล้องกับการที่จะนำข้อมูลไปจัดการโดยใช้คอมพิวเตอร์และให้สอดคล้องกับทัศนะของผู้ใช้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 องค์ประกอบของฐานข้อมูลภูมิศาสตร์

2.1.3.2 การแสดงผลข้อมูลและการนำเสนอ (รูปที่ 2-7) เป็นเรื่องของการแสดงข้อมูลและการรายงานผลการวิเคราะห์ต่อผู้ใช้ ข้อมูลอาจนำเสนอในรูปของแผนที่ ตาราง และรูปกราฟและผัง โดยวิธีการต่างๆ ตั้งแต่การแสดงภาพชั่วคราวทางจอภาพ ตลอดจนการแสดงผลด้วยเครื่องพิมพ์ หรือเครื่องวาดรูปบนกระดาษหรือฟิล์ม จนถึงข้อมูลที่บันทึกในรูปดิจิทัลบนสื่อแม่เหล็ก

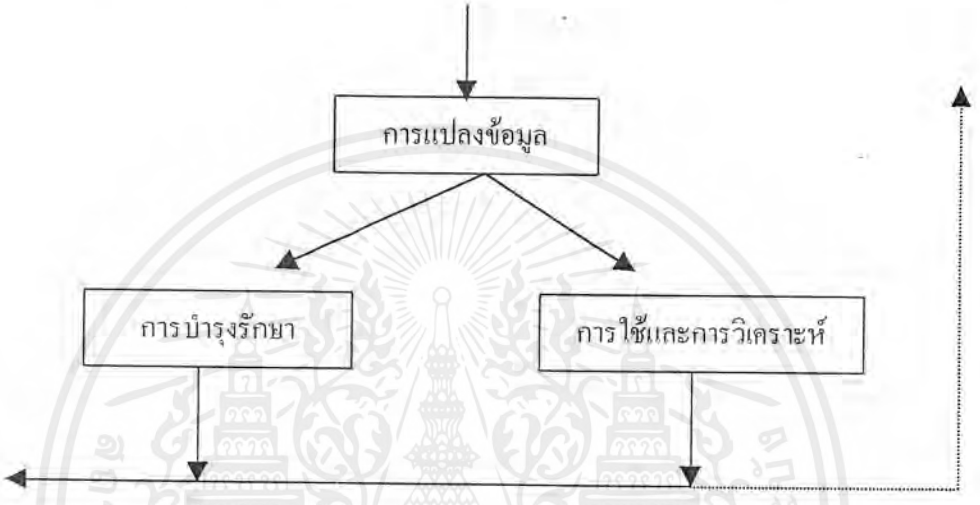


รูปที่ 2-7 ส่วนประกอบของส่วนแสดงผลและเตรียมรายงาน

2.1.3.4 การแปลงข้อมูล (รูปที่ 2-8) หมายรวมถึงวิธีการดำเนินการ 2 ประเภท ได้แก่ ประเภทแรก การแปลงเพื่อลบส่วนที่ผิดพลาดออกจากข้อมูล หรือปรับให้ทันสมัย หรือการจับคู่กับข้อมูลชุดอื่น และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทที่ 2 วิธีการวิเคราะห์หลายรูปแบบที่สามารถใช้กับข้อมูลเพื่อตอบคำถามในเรื่องของระบบสารสนเทศทางภูมิศาสตร์ การแปลงข้อมูลสามารถทำกับข้อมูลทางพื้นที่และข้อมูลที่ไม่อิงพื้นที่ โดยอาจดำเนินการร่วมกันหรือแยกกันก็ได้ วิธีการแปลงข้อมูลมีมากมายหลายแบบ เช่น การเปลี่ยนมาตราส่วน การปรับข้อมูลตามเส้นโครงแผนที่ใหม่ การสืบค้นข้อมูล การคำนวณพื้นที่และความยาวเส้นเขตวิธีการแปลงข้อมูลเหล่านี้เป็นวิธีทั่วไป การแปลงข้อมูลแบบอื่น ๆ ส่วนใหญ่มักจะเพื่อการใช้งานที่เฉพาะเจาะจงมาก และการเพิ่มส่วนนี้ไว้ใน GIS มักจะเป็นไปตามข้อเรียกร้องของผู้ใช้ระบบ



รูปที่ 2-8 การแปลงข้อมูล

2.1.4 ฐานข้อมูลเชิงภูมิศาสตร์ (Geographic Database)

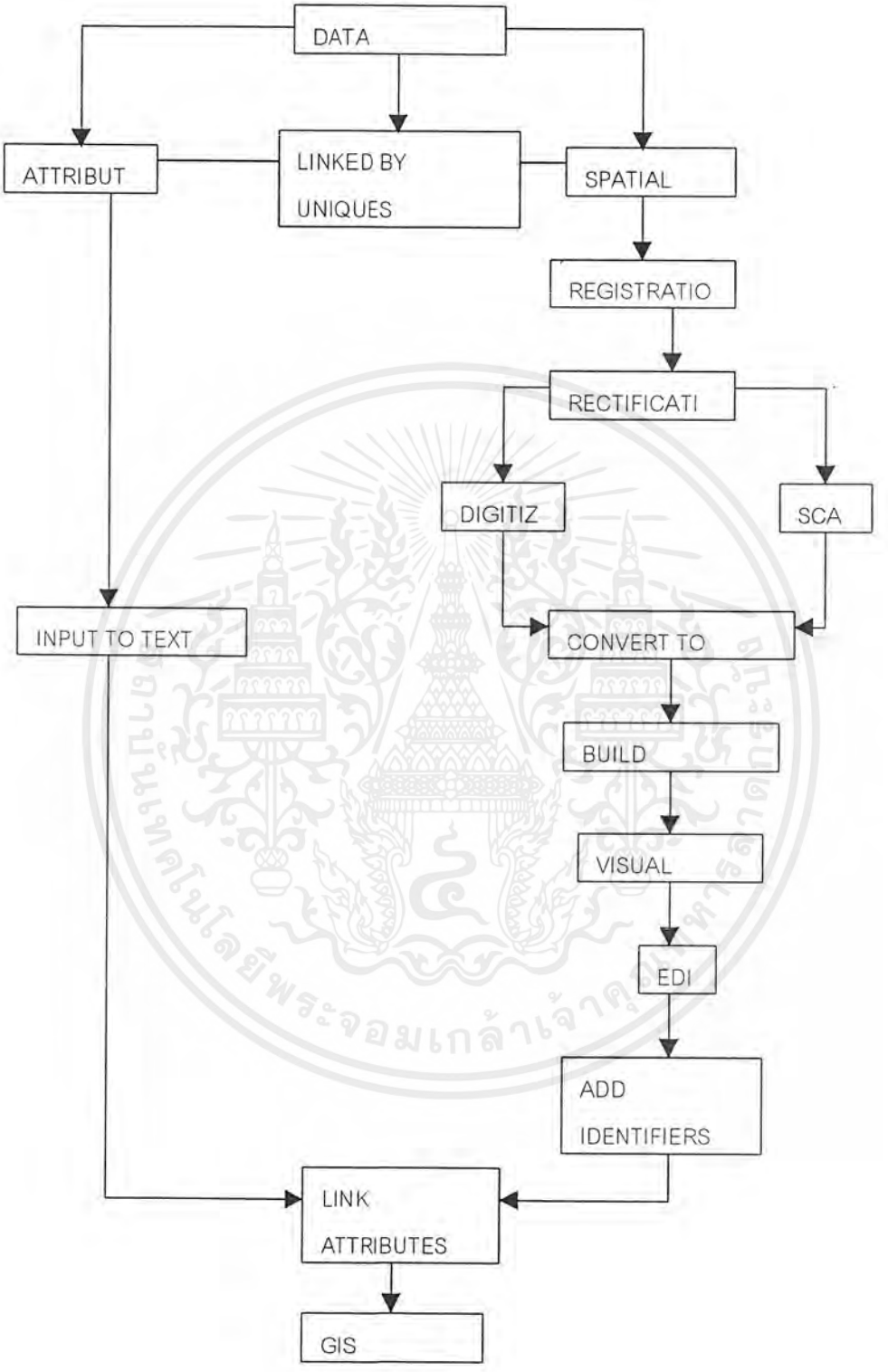
ฐานข้อมูลเชิงภูมิศาสตร์ คือ ข้อมูลต่างๆที่ใช้ในระบบการเก็บรวบรวมข้อมูลเชิงพื้นที่ (Spatial Data) ที่สัมพันธ์กับข้อมูลเชิงบรรยาย (Attribute Data)

2.1.5 การจัดการและบุคลากร (Organization and People)

บุคลากรในระบบสารสนเทศทางภูมิศาสตร์ ได้แก่ผู้บริหารฐานข้อมูล (Database Administrator : DBA) ผู้บันทึกข้อมูล (Digitizing Staff) ผู้เขียนโปรแกรมประยุกต์ (Application Programmer) ผู้ใช้ (End User) เป็นต้น บุคลากรเหล่านี้ จะทำหน้าที่การทำงานที่แตกต่างกันตามการจัดการภายในองค์กรเพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ขั้นตอนการสร้างฐานข้อมูลเชิงภูมิศาสตร์ (Geographic Database Creation Process)



รูปที่ 2-9 ขั้นตอนการสร้างฐานข้อมูลเชิงภูมิศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ข้อมูลทางภูมิศาสตร์ สามารถแบ่งออกได้เป็น 2 ส่วนหลักๆ คือ

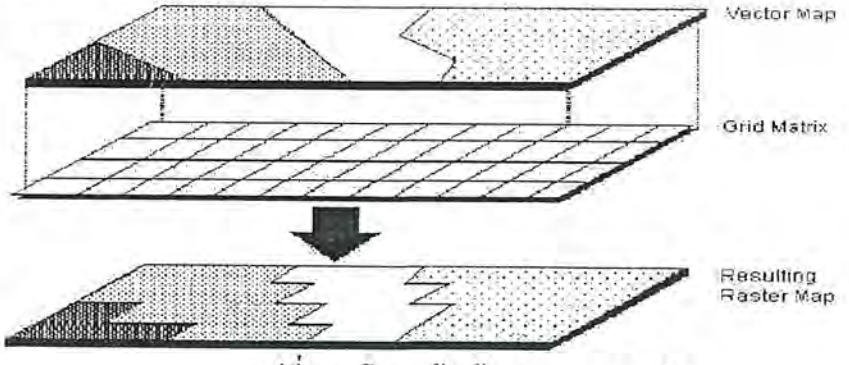
2.3.1 ข้อมูลเชิงพื้นที่ (Spatial Data) ข้อมูลเชิงพื้นที่จะเป็นข้อมูลที่เกี่ยวข้องกับตำแหน่ง ที่ตั้งของข้อมูลต่างๆ บนพื้นโลก โดยแบ่งเป็น 2 ประเภทตามโครงสร้างของข้อมูลเชิงพื้นที่ ดังนี้

2.3.1.1 โครงสร้างข้อมูลแบบเวคเตอร์ (Vector Model) ข้อมูลเชิงพื้นที่ในรูปเวคเตอร์จะเก็บอยู่ใน 4 รูปแบบ ดังต่อไปนี้

- จุด (Point) เป็นค่าพิกัดในตำแหน่งแนวนอน (X axis) และแนวตั้ง (Y axis) ไม่มีความยาวหรือพื้นที่จะใช้แสดงข้อมูลบนพื้นโลกที่เป็นลักษณะของตำแหน่งที่ตั้งเช่น ที่ตั้งของมหาวิทยาลัย เป็นต้น
- เส้น (Line) เป็นชุดของค่าพิกัดตำแหน่งที่ต่อเนื่องกันโดยมีจุดเริ่มต้นและจุดปลายมีเฉพาะความยาวไม่มีพื้นที่ จะใช้แสดงข้อมูลบนพื้นโลกที่เป็นลักษณะของเส้นเช่น เส้นทางคมนาคม , ขอบเขตการปกครอง
- พื้นที่รูปปิด (Polygon) เป็นชุดของค่าพิกัดตำแหน่งที่ต่อเนื่องกัน โดยมีจุดเริ่มต้น และจุดปลายอยู่ที่ตำแหน่งเดียวกันมีทั้งความยาว (เส้นรอบรูป) และพื้นที่ จะใช้แสดงข้อมูลที่เป็นลักษณะของพื้นที่ เช่น พื้นที่ของบริเวณแหล่งปลูกข้าว เป็นต้น
- พื้นที่ที่มีความสูงต่ำ (Surface) คือพื้นที่รูปปิดที่มีค่าพิกัดตำแหน่งในแนวดิ่ง (Vertical) มีทั้งความยาวเส้นรอบรูป พื้นที่ และความสูงต่ำของพื้นที่ มีค่า X , Y และ Z เช่นข้อมูล Slope , Aspect เป็นต้น

2.3.1.2 โครงสร้างข้อมูลแบบตารางกริด (Raster Model) ข้อมูลเชิงพื้นที่ในรูปราสเตอร์ (Raster) จะเก็บอยู่ในรูปตารางกริดย่อยๆ ต่อเนื่องกัน โดยมีคุณสมบัติดังนี้

- ตารางกริดย่อย (cell) เรียงลำดับตามแถวจากมุมบนซ้ายไปขวา
- แต่ละตารางกริดย่อยบรรจุข้อมูลได้เพียงค่าเดียว
- ตารางกริดย่อยแต่ละช่องจะมีค่าตำแหน่งพิกัด
- ตารางกริดหนึ่งชุดเรียกว่าชั้น (Layer)
- ฐานข้อมูลหนึ่งอาจจประกอบด้วยชั้นหลายๆ ชั้น เช่น ชั้นของการใช้ที่ดิน (Landuse Layer) ชั้นของเส้นค่าความสูง (Contour Layer)



รูปที่ 2-10 โครงสร้างข้อมูลแบบตารางกริด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	แบบเวคเตอร์	แบบตารางกริด
ข้อดี	- แสดงโครงสร้างข้อมูลเชิงพื้นที่ได้ดี - โครงสร้างข้อมูลกะทัดรัด - ความเชื่อมโยงทางโทโพโลยีสามารถทำได้ครบถ้วนด้วยการเชื่อมโยงแบบเครือข่าย - มีความถูกต้องในเชิงกราฟิก - สามารถทำการสืบค้น การแก้ไข และการวางนัยทั่วไปกับข้อมูลกราฟิกและลักษณะประจำได้ - เทคโนโลยีชนิดนี้มีราคาแพง โดยเฉพาะถ้าต้องใช้ฮาร์ดแวร์และซอฟต์แวร์ที่มีความซับซ้อน - การวิเคราะห์พื้นที่และการกรองรายละเอียดภายในรูปหลายเหลี่ยจาเกือบเป็นไปไม่ได้	- มีโครงสร้างข้อมูลง่าย ๆ - การวางซ้อนและการรวมข้อมูลแผนที่กับข้อมูลที่รับรู้จากระยะไกลทำได้ง่าย - การวิเคราะห์ทางพื้นที่ในแบบต่าง ๆ ทำได้ง่าย - การทดสอบด้วยการจำลองสถานการณ์ทำได้ง่าย เพราะหน่วยพื้นที่แต่ละหน่วยมีรูปร่างและขนาดเท่ากัน - เทคโนโลยีมีราคาถูกและกำลังมีการพัฒนาอย่างจริงจัง
ข้อเสีย	- โครงสร้างข้อมูลซับซ้อน - การรวมแผนที่แบบเวคเตอร์หลาย ๆ แผ่นหรือรวมแผนที่เวคเตอร์กับตารางกริดด้วยวิธีวางซ้อนมีความยุ่งยากมาก - การทดสอบด้วยการจำลองสถานการณ์ทำได้ยาก เพราะแต่ละหน่วยของแผนที่มีโครงสร้างโทโพโลยีต่างกัน - การแสดงและการเขียนเป็นแผนที่เสียค่าใช้จ่ายสูง โดยเฉพาะเมื่อต้องการแสดงสีและสัญลักษณ์ที่มีคุณภาพสูง	- ข้อมูลกราฟิกมีขนาดใหญ่ - การใช้ช่องกริดขนาดใหญ่เพื่อลดปริมาตรข้อมูลทำให้สูญเสียโครงสร้างข้อมูลเชิงพื้นที่และเป็นการสูญเสียข้อมูลสารสนเทศอย่างมาก - แผนที่ตารางกริดที่หยาบจะไม่สวยเท่าแผนที่ซึ่งเขียนด้วยเส้น - การสร้างเครือข่ายเชื่อมโยงทำได้ยาก - การแปลงเส้นโครงแผนที่ต้องใช้เวลามาก เว้นแต่จะใช้ขั้นตอนวิธีหรือฮาร์ดแวร์พิเศษ

ตารางที่ 2-1 เปรียบเทียบข้อดี ข้อเสียของการจัดเก็บข้อมูลแบบเวคเตอร์และแบบตารางกริด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ข้อมูลเชิงบรรยาย (Attribute Data) เป็นข้อมูลที่บอกเราว่าข้อมูลเชิงพื้นที่ที่เห็นอยู่นั้นมีรายละเอียดอะไรบ้างเช่น เขตบึงกุ่ม มี 3 ตำบล มีพื้นที่ทั้งหมด 69.903 ตารางกิโลเมตร มีจำนวนประชากร 226,902 คน ฯลฯ

อย่างไรก็ตามเมื่อนำเข้าข้อมูลทั้ง 2 ประเภทข้างต้นโปรแกรมทางด้าน GIS จะทำการเชื่อมโยงข้อมูลทั้งสองเข้าด้วยกันซึ่งจุดนี้จะเป็นส่วนสำคัญของโครงสร้างข้อมูลทางระบบ GIS และจากการที่ข้อมูลทั้งสองเชื่อมโยงกันอยู่นี้เองทำให้เราสามารถที่จะสอบถามข้อมูลได้จากทั้งสองทางคือเราสามารถที่จะทราบข้อมูลแอตทริบิวต์ ของข้อมูลพื้นที่ได้จากการเลือกข้อมูลพื้นที่และสามารถที่จะค้นหาตำแหน่งของข้อมูลพื้นที่ได้จากการกำหนดลักษณะของข้อมูลแอตทริบิวต์ ในการทำงานจะมีการสร้างข้อมูลแอตทริบิวต์ ที่เก็บข้อมูลในแต่ละหัวข้อแยกกันไปเป็นตารางย่อยๆ เพื่อป้องกันการซ้ำซ้อนของข้อมูลซึ่งระบบ GIS สามารถที่จะเชื่อมโยงตารางเหล่านี้เข้าด้วยกันได้ โดยใช้ข้อมูลเป็นตัวเชื่อมซึ่งจะทำให้เราสามารถที่จะทราบรายละเอียดของข้อมูลที่เราต้องการในตารางในตารางย่อยได้

2.4 การทำงานของระบบสารสนเทศทางภูมิศาสตร์

2.4.1 การนำเข้าข้อมูล

- การ Digitize จะเป็นการนำเข้าข้อมูลเชิงพื้นที่โดยอาศัยอุปกรณ์ทางคอมพิวเตอร์ที่เรียกว่า Digitizer อุปกรณ์จะทำการแปลงข้อมูลแผนที่บนกระดาษให้อยู่ในรูปแบบของข้อมูลดิจิตอล
- การนำเข้าข้อมูลจากการรังวัด เราสามารถนำเอาค่าของมุมและระยะทางที่ได้จากการทำรังวัดเข้าสู่ระบบสารสนเทศทางภูมิศาสตร์ได้ด้วยโปรแกรม COGO ของ ARC/INFO
- การนำเข้าข้อมูลแอตทริบิวต์ ต่างๆ โดยอาศัยการพิมพ์ข้อมูลเข้าที่คีย์บอร์ด

2.4.2 การจัดการข้อมูลโดยอาศัย Map Library

การจัดการข้อมูลในลักษณะนี้จะใช้กับข้อมูลที่มีขนาดใหญ่มากๆ โดยที่โปรแกรมจะจัดแบ่งข้อมูลขนาดใหญ่นั้นออกเป็นส่วนย่อยๆ เพื่อทำให้การแสดงผลมีความรวดเร็วและป้องกันการแก้ไขข้อมูลที่ซ้ำซ้อนกัน

2.4.3 การแสดงผลและการสอบถามข้อมูล

- เราสามารถที่จะทราบข้อมูลแอตทริบิวต์ของข้อมูลพื้นที่หรือตำแหน่งต่าง ๆ ได้จากการกำหนดตำแหน่งที่ต้องการแล้วสั่งให้โปรแกรมแสดงผลข้อมูลแอตทริบิวต์ ของข้อมูลพื้นที่ที่ได้เลือกไว้
- เราสามารถที่จะค้นหาตำแหน่งของข้อมูลที่ตรงตามข้อกำหนดต่าง ๆ ที่เราเลือกจากข้อมูลแอตทริบิวต์ เช่นถ้าเรามีข้อมูลเกี่ยวกับที่ดินอยู่แล้ว ต้องการทราบว่ามีบริเวณใดบ้างที่มีการใช้ที่ดินแบบทำการเกษตรเราก็ต้องทำการค้นหาข้อมูลโดยอาศัยข้อกำหนดข้างต้นแล้วทำการแสดงพื้นที่ที่เลือกได้ออกมาที่จอภาพ

2.4.4 การวิเคราะห์ข้อมูล

การวิเคราะห์ข้อมูลในระบบสารสนเทศทางภูมิศาสตร์ มีด้วยกันหลายแบบตามลักษณะของการนำไปใช้งานได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.1 การสร้างระยะห่าง (Proximity หรือ Buffer Analysis)

เป็นการสร้างขอบเขตของข้อมูลขึ้นมาตามแนวข้อมูลเดิมที่เรามีอยู่เช่น ถ้าในการวิเคราะห์เราต้องการที่จะกำหนดบริเวณที่อยู่ห่างจากถนนสายหลัก 800 เมตร ให้เป็นเขตที่มีความสะดวกในการเดินทางเข้าอยู่อาศัยสิ่งที่จะต้องทำคือสร้าง Buffer zone เป็นแนวห่างจากถนนสายหลักที่เรากำหนดเป็นระยะ 800 เมตรเป็นต้น

2.4.4.2 การเรียงซ้อนทับของข้อมูล (Overlay)

จะเป็นการรวมข้อมูลของชั้น (Layer) ต่างๆ เข้าด้วยกันเพื่อให้ได้ข้อมูลที่รายละเอียดตามที่เราต้องการออกมาเช่น ถ้าเราต้องการที่จะหาบริเวณที่มีความเหมาะสมในการอยู่อาศัยโดยมีการกำหนดเงื่อนไขว่าบริเวณที่เหมาะสมในการอยู่อาศัยจะต้องมีบริการประปา โทรศัพท์และอยู่ห่างจากถนนสายหลักภายในระยะ 800 เมตร เราก็จะทราบว่าชั้นที่จะต้องใช้ในการวิเคราะห์คือขอบเขตที่มีการบริการประปา โทรศัพท์ และอยู่ห่างจากถนนสายหลักไม่เกิน 800 เมตร (ได้จากการทำ Buffer Analysis) แล้วนำเอาข้อมูลทั้งสามชั้นมาซ้อนทับกันก็จะได้ข้อมูลใหม่ออกมาที่รวมเอาข้อมูลเชิงพื้นที่และข้อมูลเชิงบรรยายของข้อมูลทั้ง 3 ปัจจัยเข้าไว้ด้วยกันทำให้เราทราบว่ามีบริเวณใดบ้างที่มีความเหมาะสมในการอยู่อาศัย

2.4.4.3 การวิเคราะห์โครงข่ายเส้นทาง (Network Analysis)

จะเป็นการวิเคราะห์ข้อมูลที่อยู่ในรูปของเส้นต่างๆ เช่น ถนน ท่อประปา การจราจร เป็นต้น หนึ่งในการทำวิเคราะห์โครงข่ายเส้นทาง จะเป็นการค้นหาเส้นทางที่สั้นที่สุดหรือใช้ระยะเวลาในการเดินทางน้อยที่สุด จากจุดที่เรากำหนดเป็นจุดเริ่มต้นไปยังจุดหมายที่ต้องการเช่นการหาเส้นทางที่สั้นที่สุดหรือรวดเร็วที่สุด ในการเดินทางจากสถานีตำรวจไปยังจุดที่เกิดอุบัติเหตุต่างๆ เราจะเรียกการประยุกต์ใช้แบบนี้ว่า Emergency Response ซึ่งการวิเคราะห์จะให้ผลที่น่าเชื่อถือได้ก็ต่อเมื่อเรามีการเก็บข้อมูลเกี่ยวกับถนนเส้นต่างๆ อย่างละเอียดทั้งในด้านความกว้าง จำนวนรถ สัญญาณจราจร ฯลฯ เพื่อจะทำให้แบบจำลองนี้สมจริงมากที่สุด

2.4.4.4 การวิเคราะห์พื้นผิวของพื้นที่ (Surface Analysis)

จะเป็นการนำเอาข้อมูลค่าความสูงของพื้นที่มาสร้างเป็นแบบจำลองของภาพสามมิติและสามารถที่จะวาดข้อมูลอื่นๆ ของพื้นที่นั้น (รวมทั้งภาพดาวเทียม) ลงบนภาพสามมิติได้เพื่อทำให้การตีความหมายของภาพที่เห็นได้ชัดเจนยิ่งขึ้นตลอดจนการสร้างภาพ profile ของพื้นที่เพื่อให้เห็นลักษณะของพื้นที่ทางภาพตัดขวางหรือการคำนวณหาขนาด หรือปริมาตรของพื้นที่ที่จะต้องทำการขุดหรือถม หาแนวการเอียงตัวของพื้นผิว Aspect รวมทั้งการสร้างเส้นชั้นความสูงจากข้อมูลค่าความสูงที่มีอยู่

2.4.5 การทำ Image Catalog

ข้อมูลของระบบสารสนเทศทางภูมิศาสตร์โดยมากจะได้มาจากการ Digitize จากแผนที่ซึ่งข้อมูลต่างๆ เหล่านี้จะเก็บอยู่ในรูปของจุด, เส้น และ พื้นที่รูปปิด แต่ในบางกรณีที่เราอาจจะมีความจำเป็นที่จะต้องนำเอาข้อมูลภาพถ่ายดาวเทียมและผลลัพธ์ของการ Classified ข้อมูลที่ได้จากการทำงานในระบบ Remote Sensing หรือภาพสแกนของภาพถ่ายทางอากาศมาใช้งานร่วมกับข้อมูลของระบบสารสนเทศทางภูมิศาสตร์ การนำมาใช้งานร่วมกันได้แก่การสอบถามข้อมูล การวาดภาพของภาพถ่ายทางอากาศหรือภาพจากข้อมูลดาวเทียมร่วมกับข้อมูลในระบบเพื่อให้ง่ายต่อการแปลข้อมูลและให้รายละเอียดได้มากขึ้น รวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งการใช้ภาพดังกล่าวมาช่วยในการแก้ไขข้อมูลในระบบให้ถูกต้อง วิธีการวาดข้อมูลในระบบสารสนเทศทางภูมิศาสตร์ กับข้อมูลภาพทั้งสองนั้นจะเรียกว่าการทำ Register ซึ่งข้อมูลของภาพทั้งสองจะต้องมีการ Rectified มาก่อนเพื่อทำให้มีค่าพิกัดที่ถูกต้องและคล้องจองกับข้อมูลในระบบสารสนเทศทางภูมิศาสตร์ นอกจากนั้นยังมีภาพสแกนอีกแบบที่เรียกว่า การทำ Register ซึ่งเป็นข้อมูลภาพที่เราจะนำมาเชื่อมโยงกับข้อมูลประเภทต่างๆ ในชั้นข้อมูลของเราเมื่อต้องการที่จะเรียกดูภาพก็สามารถที่จะใช้คำสั่งเรียกดูได้เช่น ถ้าเรามีข้อมูลที่เกี่ยวกับตำแหน่งของเสาไฟฟ้าอยู่แล้วเราทำการสแกนภาพวาดของ Transformer เก็บไว้ในฐานข้อมูลแล้วทำการเชื่อมโยงกับข้อมูลในชั้นของเสาไฟฟ้าของเรา เราก็สามารถที่จะเรียกดูภาพของ Transformer ของเสาไฟฟ้านั้นดูได้โดยที่อาจจะมีเสาไฟฟ้าหลายเสาที่ใช้ Transformer ลักษณะเดียวกัน หรือเราอาจจะทำงานเกี่ยวกับฐานข้อมูลของเมืองที่ต้องการแสดงตำแหน่ง รูป และรายการอาหารของร้านอาหารต่างๆ ในเมืองเพื่อให้นักท่องเที่ยวมาสอบถามจากระบบของเรา ดังนั้นจึงต้องมีการเก็บตำแหน่งของข้อมูลร้านอาหารในเมือง พร้อมทั้งสแกนภาพถ่ายของร้านอาหารและรายการอาหารของแต่ละร้านลงในฐานข้อมูล จากนั้นทำการเชื่อมโยงข้อมูลของภาพทั้งสองเข้ากับข้อมูลตำแหน่งของร้านอาหารซึ่งเมื่อเราเลือกร้านอาหารและรายการอาหารออกมาได้ การทำงานในลักษณะดังกล่าวจะเรียกว่าการรวบรวมข้อมูล (Integration)

2.5 การนำระบบสารสนเทศทางภูมิศาสตร์ มาประยุกต์ใช้งานได้ด้านต่างๆ (GIS Application)

2.5.1 การประยุกต์ใช้งานด้านการคมนาคมขนส่ง

เป็นการนำเอาระบบ GIS ไปใช้ในการดูแลและจัดการซ่อมแซมถนน โดยที่จะต้องเก็บข้อมูลเส้นถนนต่างๆ ลงไปรวมทั้งแอตทริบิวต์ เกี่ยวกับการซ่อมบำรุงถนนแต่ละสายโดยเมื่อถึงเวลาที่จะทำการเลือกถนนที่จะต้องซ่อมบำรุงก็สามารถที่จะเลือกและแสดงข้อมูลของเส้นถนนที่ต้องการซ่อมบำรุงแล้วสร้างแผนที่หรือส่งรายงานให้ฝ่ายซ่อมบำรุงทราบว่าต้องซ่อมถนนเส้นใดบ้างเมื่อทำการซ่อมเสร็จก็จะต้องปรับแก้ข้อมูลถนนให้ถูกต้อง อีกตัวอย่างของการนำเอาระบบ GIS มาใช้ได้แก่ การกำหนดเส้นทางการเดินรถประจำทางใหม่ซึ่งจะต้องให้บริการประชาชนได้มากที่สุด โดยจะต้องมีการเลือกเส้นทางที่ได้มีการวางแผนไว้แล้ว ซึ่งจะมีมาตรฐานกำหนดไว้ว่าผู้ที่มาใช้บริการจะต้องอยู่ห่างจากเส้นทางไม่เกิน 1 กิโลเมตร ดังนั้นจึงต้องนำเอาข้อมูลจำนวนประชากร จากนั้นทำการเปรียบเทียบจำนวนประชากรที่เส้นทางแต่ละเส้นให้บริการได้แล้วเลือกเส้นทางที่ให้บริการมากที่สุด

2.5.2 การประยุกต์ใช้งานเกี่ยวกับการวางแผนการขยายพื้นที่ของสนามบิน

เป็นการนำเอาระบบ GIS ไปวิเคราะห์หาว่าจะเลือกแผนผังใดที่เหมาะสมต่อการขยายตัวของสนามบินโดยมีเงื่อนไขว่าจะต้องมีเสียงดังรบกวนประชาชนให้น้อยที่สุดและมีผลกระทบกับจำนวนประชากรที่ต้องย้ายออกไปให้น้อยที่สุดเมื่อนำเอาแบบแปลนต่างๆ มาวิเคราะห์กับข้อมูลประชากรแล้วก็จะเห็นได้ว่าแบบแปลนใดที่เหมาะสมที่สุดโดยพิจารณาจากผลกระทบข้างต้น

2.5.3 การประยุกต์ใช้งานเกี่ยวกับการประเมินอาชญากรรม

จะเป็นการจัดเก็บข้อมูลของอาชญากรรมประเภทต่างๆ ที่เกิดขึ้น เพื่อพิจารณาถึงความสัมพันธ์ของประเภทอาชญากรรมและช่วงเวลาที่เกิด เมื่อเห็นความสัมพันธ์ของทั้งสองสิ่งแล้วเราก็จะทราบว่าใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริเวณใดมีอาชญากรรมเกิดขึ้นบ่อยในช่วงเวลาที่กำหนดทำให้ทางตำรวจสามารถที่จะกระจายกำลังออกไปดูแลความสงบได้ถูกต้องมากยิ่งขึ้น

2.5.4 การประยุกต์ใช้งานทางด้านการวางแผนการใช้ที่ดิน

ทำการนำข้อมูลทางด้านการใช้ที่ดินเข้าสู่ระบบ แล้วก็ทำการเปรียบเทียบดูกับพื้นที่และข้อมูลอื่นๆ ดูว่ามีพื้นที่ใดบ้างที่ควรจะต้องเปลี่ยนแปลงการใช้ที่ดินเพื่อให้ผลประโยชน์มากที่สุด

2.5.5 การประยุกต์ใช้ทางด้านสิ่งแวดล้อม

การดูผลกระทบของฝนกรด (Acid rain) ที่จะมีต่อพื้นที่ป่าซึ่งเราทราบว่าฝนกรดมีผลกระทบต่อพันธ์ไม้แต่ละประเภทแตกต่างกันไป ทำให้เราสามารถที่จะวางแผนที่จะป้องกันฝนกรดได้หรือการตรวจการย้ายถิ่นฐานของหมีขั้วโลกในช่วงฤดูหนาว ซึ่งจะไม่มีแสงแดดเลยทำให้นักชีววิทยาไม่สามารถที่จะตราจสอบการย้ายถิ่นฐานของหมี ดังนั้นนักชีววิทยาจึงใช้เครื่อง GPS ผูกติดกับคอของหมีเพื่อให้ระบบ GPS ส่งค่าตำแหน่งพิกัดของหมีออกมาทุกๆ ช่วงเวลาที่กำหนดแล้วนำข้อมูลที่ได้จากระบบ GPS เข้าสู่ระบบ GIS ก็จะได้ข้อมูลการเคลื่อนย้ายถิ่นฐานของหมีได้

2.5.6 การประยุกต์ใช้งานทางด้านการแบ่งขอบเขต

ในประเทศอเมริกาจะมีการแบ่งขอบเขต สำหรับการเลือกตั้งเพื่อที่จะทำให้จำนวนประชากรของแต่ละเขตมีจำนวนเท่าๆ กัน โดยที่ทางอเมริกาจะมีการสำรวจสำมะโนประชากรทุกๆ 10 ปี ทำให้เราทราบจำนวนของประชากรที่ถูกต้องได้

2.5.7 การประยุกต์ใช้ทางด้านเกี่ยวกับการหาพื้นที่

ใช้ทางด้านเกี่ยวกับการหาพื้นที่ที่น่าจะมีผลผลิตสูงแล้วทำการสร้างระบบชลประทานจะเป็นการนำเอาข้อมูลต่างๆ ที่มีผลต่อการเจริญเติบโตของพืช ซึ่งเป็นวัตถุดิบในการผลิตมาทำการวิเคราะห์หาพื้นที่ที่น่าจะมีผลผลิตสูงเพื่อใช้ในการตัดสินใจในการสร้างระบบชลประทานเพื่อส่งเสริมผลผลิตของพื้นที่นี้ให้มากขึ้น

2.6 ประโยชน์ของระบบ GIS

2.6.1 ช่วยลดค่าใช้จ่ายในการทำงาน

เนื่องจากระบบ GIS มีการจัดเก็บข้อมูลในรูปของดิจิตอล ดังนั้นจึงเป็นการง่ายที่จะเรียกข้อมูลออกมาทำการแก้ไขตลอดจนนำข้อมูลที่มีอยู่นั้นมาสร้างแผนที่ซึ่งจะง่ายและรวดเร็ว

2.6.2 มีการจัดการระบบฐานข้อมูลที่ดี

เมื่อเราเก็บข้อมูลแผนที่ลงในฐานข้อมูล GIS แล้วเราสามารถที่จะเรียกค้นข้อมูลแผนที่ที่เราต้องการได้อย่างรวดเร็วและถ้าเราเรียกค้นข้อมูลที่ได้มาจากการวิเคราะห์ก็จะทำให้เราได้ข้อมูลที่มีรายละเอียดมากกว่าและดีกว่า

2.6.3 เป็นเครื่องมือช่วยในการตัดสินใจที่ดี

ถ้าเรานำข้อมูลที่จะใช้ในการทำงานลงไปทั้งทางด้าน พื้นที่และเชิงบรรยายแล้วทำการวิเคราะห์ข้อมูลเพื่อช่วยเราตอบปัญหาบางอย่างที่เราต้องการทราบ เราจะเห็นว่าข้อมูลระบบ GIS จะเป็นแบบจำลองของพื้นที่จริงๆ บนพื้นโลกได้เป็นอย่างดีซึ่งเราจะนำผลลัพธ์ที่ได้นี้ไปเป็นเครื่องช่วยในการตัดสินใจ อย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไรก็ตามข้อมูลที่ได้จากการวิเคราะห์จะถูกต้องเหมือนกับความเป็นจริงที่จะเกิดขึ้นบนพื้นโลกได้จะต้องอาศัยการนำเข้าข้อมูลที่มีความถูกต้องและมีความทันสมัย

2.6.4 ระบบ GIS มีการทำงานที่มีประสิทธิภาพ

ในงานบางลักษณะโดยปกติแล้วทำได้ยากยิ่งเช่น การสร้างภาพ 3 มิติของพื้นที่ หรือการสร้างและวิเคราะห์ในระบบ Network ซึ่งเมื่อมีระบบ GIS นั้นจะทำให้เราสามารถทำงานดังกล่าวได้อย่างมีประสิทธิภาพ

2.6.5 ความสามารถในการผสมผสานชุดของข้อมูล

ระบบ GIS มีความสามารถในการผสมผสานชุดของข้อมูลต่างๆ เข้าด้วยกัน การที่ข้อมูลเศรษฐกิจและสังคมกับข้อมูลทางกายภาพสามารถที่จะผสมผสานกันได้นั้นทำให้ GIS เป็นเครื่องมืออย่างหนึ่งที่มีประสิทธิภาพในระบบสารสนเทศทางภูมิศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่ 3
แนวความคิดเชิงวัตถุ (Object-Oriented Concept)

การเขียนโปรแกรมด้วยวิธี Object-Oriented นั้น คือวิธีการเขียนโปรแกรมคอมพิวเตอร์โดยใช้แนวความคิดเสมือนการมองภาพวัตถุ (Object) ในโลกแห่งความเป็นจริง เป็นการจัดโครงสร้างของโปรแกรมให้เป็นระเบียบมากยิ่งขึ้น และช่วยให้การพัฒนาโปรแกรมในรุ่นต่อๆไปทำได้ง่ายขึ้น ออบเจกต์โอเรียนเต็ด (Object-Oriented) เป็นวิธีการจัดแบ่งประเภทของวัตถุในทางนามธรรม (Abstract) ออกเป็นกลุ่มๆ (Classes) ซึ่งในแต่ละกลุ่มก็มีสถานะ (States) และพฤติกรรม (Behaviors) เฉพาะของตัวเอง เพื่อเป็นต้นแบบให้แก่ออบเจกต์ (Object) ที่จะถูกสร้างขึ้นใหม่ในอนาคต

3.1 ออบเจกต์ (Object)

ออบเจกต์ ประกอบด้วยข้อมูล (Attribute) และวิธีการจัดการ (Method) กับข้อมูลนั้น ยกตัวอย่างออบเจกต์ในโลกแห่งความเป็นจริง เช่น หลอดไฟเป็นสิ่งของที่มีสถานะ เช่น รูปทรง กำลังวัตต์ สถานะของสวิทช์ไฟเปิดปิด เป็นต้น และมีพฤติกรรม เช่น การส่องแสงสว่าง การเกิดความร้อนเป็นต้น เราสามารถจะแทนสถานะด้วยตัวแปร (Variable) และแทนพฤติกรรมด้วยเมทธอด(Method) ซึ่งเมทธอดจะทำหน้าที่เก็บซ่อนไว้ ประโยชน์การเก็บซ่อน (Encapsulation) นี้คือ ถ้าหากเราต้องเปลี่ยนแปลงข้อมูลภายในออบเจกต์ การเปลี่ยนแปลงจะไม่ส่งผลต่อออบเจกต์อื่น ๆ ที่อยู่ภายนอกเพราะข้อมูลเป็นอิสระจากกัน เราจึงสามารถเปลี่ยนแปลงได้ และประโยชน์อีกข้อหนึ่งก็คือ เป็นการจัดการกับข้อมูลภายในออบเจกต์ว่าข้อมูลส่วนนี้ต้องการที่จะเปิดเผยหรือไม่ ซึ่งสามารถกำหนดได้โดยการประกาศคุณสมบัติให้กับออกเจกต์ เช่น Private , Public , Protect ซึ่งจะได้กล่าวรายละเอียดในหัวข้อต่อไป

3.2 คลาส (Class)

คลาส คือ ต้นแบบของการกำหนดตัวแปรและเมทธอดของออบเจกต์ คลาสหนึ่งคลาสสามารถเป็นต้นแบบให้แก่ออบเจกต์หลาย ๆ ออบเจกต์ได้ เช่น รถจักรยานหลาย ๆ คันมีคุณสมบัติการเป็นจักรยานเหมือนกันมีล้อสองล้อ มีตัวถัง มีอานสำหรับนั่งเหมือนกัน เป็นต้น ที่โรงงานผลิตจักรยานก็จะมีต้นแบบพื้นฐานของรถจักรยานที่จะผลิตเช่น แบบพิมพ์เขียว (คลาส) ของรถจักรยาน เป็นต้น แต่เมื่อผลิตรถจักรยานจริง ๆ (ออบเจกต์) ออกมาเสร็จแล้ว รายละเอียดปีกย่อยจะแตกต่างกันไปแล้วแต่รุ่น เช่น สี ขนาด น้ำหนัก รูปทาง อานนั่ง เป็นต้น การที่เราสร้างออบเจกต์ขึ้นมาหนึ่งออบเจกต์หรือหลายออบเจกต์เราเรียกว่าอินสแตนซ์ (Instance) เราต้องสร้าง อินสแตนซ์ขึ้นมาจากคลาสก่อนที่เราจะมีการใช้ตัวแปรหรือเมทธอดของคลาสนั้น และ จะเรียกอินสแตนซ์ ว่าเป็นออบเจกต์เมื่อการเปลี่ยแปลงข้อมูลหรือเรียกใช้เมทธอดของอินสแตนซ์นั้นเกิดขึ้น

3.3 โพลิมอร์ฟิซึม (Polymorphism)

โพลิมอร์ฟิซึม คือ การสร้างฟังก์ชันที่มีหน้าที่แตกต่างกัน แต่เรียกใช้ได้ด้วยชื่อเดียวกัน เรียกว่า 'One interface, multiple methods' เพื่อให้ใช้งานได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
37040

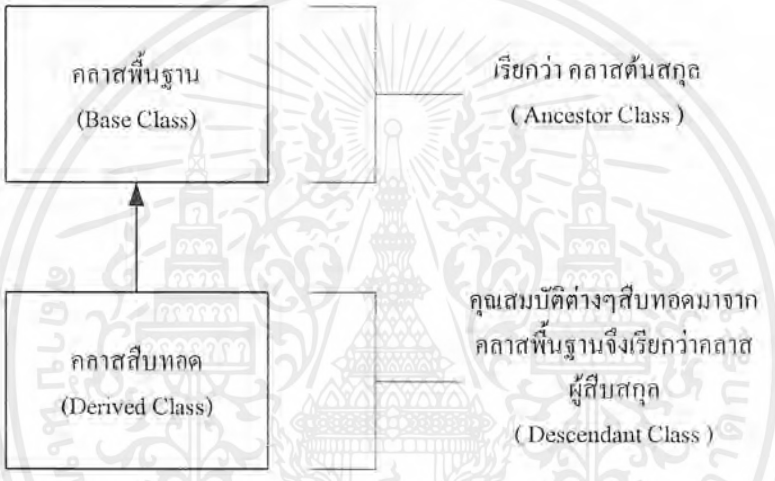
โพลิมอร์ฟิซึม แบ่งออกได้เป็น 2 ลักษณะ คือ

1. ช่วงการคอมไพล์โปรแกรม (Compile time) การตรวจสอบชนิดออบเจกต์ของฟังก์ชันที่ถูกเรียกมาทำงาน รวมทั้งเตรียมข้อมูลบางอย่างที่จำเป็นจะทำขณะคอมไพล์โปรแกรม การตัดสินใจใช้ฟังก์ชันใดขึ้นอยู่กับคุณสมบัติของพารามิเตอร์ (Parameter) ของฟังก์ชันนั้น

2. ช่วงการรันโปรแกรม (Run time) การตรวจสอบนั้นจะทำขณะที่โปรแกรมกำลังทำงาน จะทำการตัดสินใจเลือกฟังก์ชันใดฟังก์ชันหนึ่งที่มีชื่อซ้ำกันในแต่ละคลาสมาใช้งาน

3.4 อินเฮอริแตนซ์ (Inheritance)

อินเฮอริแตนซ์ หมายถึง การที่ออบเจกต์อันหนึ่งสามารถสืบทอดคุณสมบัติจากออบเจกต์อื่นนำมาใช้งานได้



รูปที่ 3-1 ความสัมพันธ์แบบอินเฮอริแตนซ์

คลาสสืบทอด จะสืบทอดแอตทริบิวต์และเมทธอดจากคลาสพื้นฐาน ทำให้สามารถใช้แอตทริบิวต์และเมทธอดเดิมของคลาสแม่ผสมกับแอตทริบิวต์และเมทธอดของคลาสลูกที่สร้างขึ้นใหม่

3.5 เอ็นแคปซูเลชัน (Encapsulation)

โครงสร้างของโปรแกรมต้องจบในตัวเองเป็นก้อน และเป็นอิสระและทำงานได้ด้วยตนเอง ทุก ๆ คลาสจะต้อง Self-Contained (มีครบในตัว) และทำงานได้อย่างเป็นเอกเทศ แต่ในขณะเดียวกันก็ต้องสามารถแยกส่วนออกมาและทำงานได้เป็นอิสระเช่นกัน เช่น ถ้าสุนัขเป็นออบเจกต์ที่เราออกแบบ และมีตัวหมัด (อีกออบเจกต์) หนึ่งซ้อนอยู่ เราสามารถเอา ตัวหมัด (ออบเจกต์) มาใช้ได้เลย เพราะ ออบเจกต์เป็น Selft-contain หรือทำงานได้อย่างเป็นอิสระ เอาหมัดไปใส่ออบเจกต์อื่น เช่น แมว ก็ได้ หมายความว่าเอาเครื่อง ฮอนด้าไปใส่ มาสด้า หรือ โตโยต้า ก็ยังทำงานได้ ปรกติ เพราะเครื่องนั้นยังไงก็ทำงานได้ด้วยตัวเอง เพราะข้อมูลและวิธีการนั้นถูกอัดเป็น capsule รวมกันอยู่ในตัวเดียวแล้ว

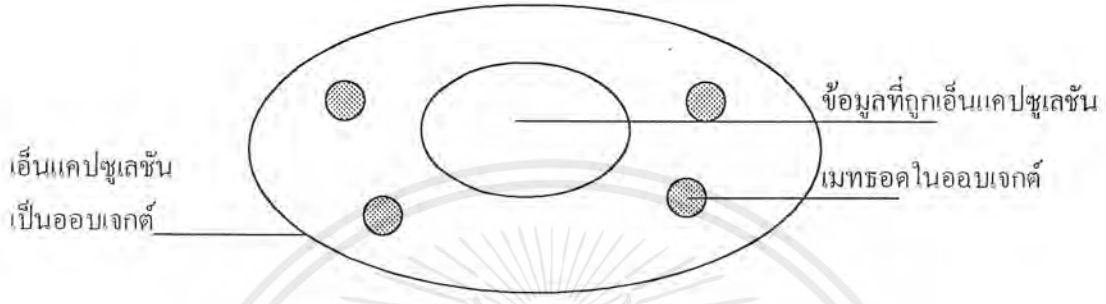
นอกจากนี้ยังมีความสามารถในการป้องกันการเข้าถึงข้อมูลภายในออบเจกต์จากฟังก์ชันภายนอก ซึ่งวิธีการในการป้องกันนั้นจะอยู่ที่ชนิดของการประกาศข้อมูลภายในคลาส ซึ่งมีอยู่สามชนิด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Private การประกาศข้อมูลแบบนี้จะมีผลทำให้ไม่มีคำสั่งใดที่จะอ้างถึงข้อมูลภายในคลาสนั้นได้เลย นอกจากเมทธอดภายในคลาสนั้นเอง ซึ่งปรกติแล้วจะมีอย่างน้อยหนึ่งเมทธอดที่ไม่ได้ประกาศให้เป็นข้อมูลชนิดนี้ เพื่อที่จะได้ใช้ในการเข้าถึงข้อมูลภายในคลาสได้

- Public การประกาศข้อมูลแบบนี้จะมีผลทำให้สามารถอ้างอิงข้อมูลภายในคลาสได้โดยอิสระ

- Protect การประกาศข้อมูลแบบนี้จะมีผลทำให้มีเฉพาะ Derived Class เท่านั้นที่จะอ้างอิงได้



รูปที่ 3-2 การเอ็นแคปซูเลชัน

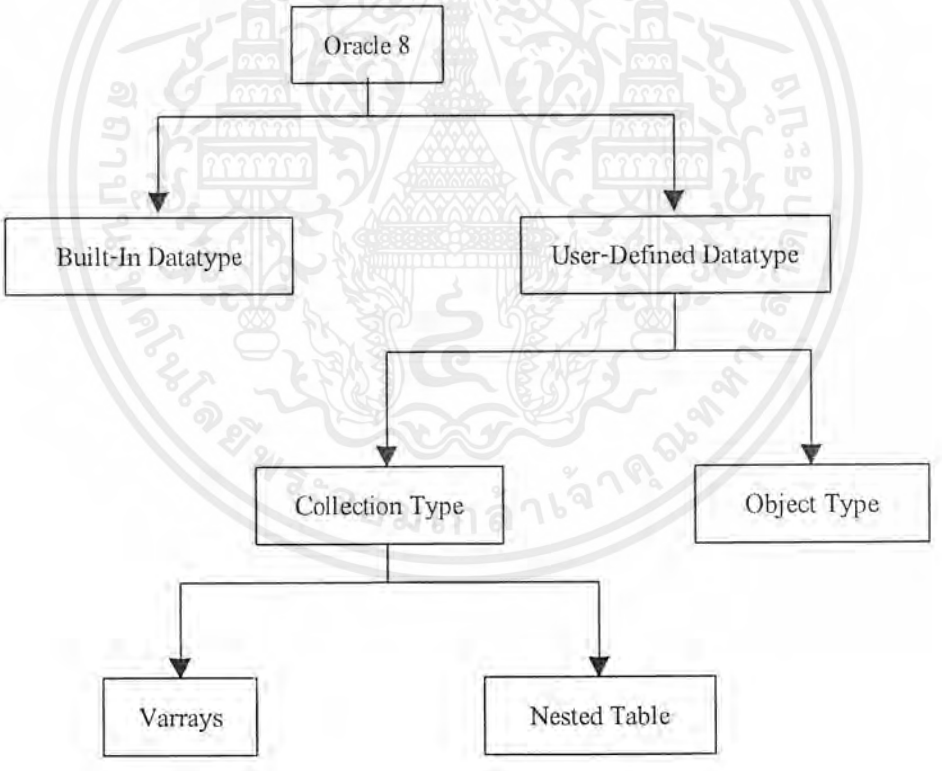
3.6 การนำแนวความคิดเชิงวัตถุมาใช้ในโครงงาน

ในโครงงานนี้ได้นำเรื่องของแนวความคิดเชิงวัตถุมาใช้ในการเขียนโปรแกรมด้วยภาษาจาวา ช่วยให้การพัฒนาโปรแกรมมีระเบียบแบบแผน ง่ายต่อการพัฒนา และง่ายต่อการดูแลแก้ไขโปรแกรมในอนาคตด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ทฤษฎีของโอราเคิล 8 ที่สนับสนุน ORDBMS

โอราเคิล 8 เป็นระบบฐานข้อมูลแบบเชิงวัตถุสัมพันธ์ (Object Relational Database Management System : ORDBMS) ซึ่งเป็นระบบฐานข้อมูลที่สามารถรองรับข้อมูลที่มีความซับซ้อนมาก ๆ เช่น ข้อมูลมัลติมีเดีย, รูปภาพ, เสียง เป็นต้น สามารถจัดการกับข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพโดยโอราเคิล8 ได้จัดเตรียมชนิดข้อมูลชนิดใหม่ขึ้นมาเพื่อรองรับกับข้อมูลที่มีความซับซ้อนมาก ๆ เหล่านี้โดยเฉพาะยอมให้มีข้อมูลชนิดที่ผู้ใช้กำหนดขึ้นมาเอง (User-Defined Datatype) ข้อมูลชนิดนี้จะมีคุณสมบัติทางออบเจกต์โอเรียนเต็ด เช่นเดียวกับออบเจกต์ในโปรแกรมภาษาที่สามารถเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ต่าง ๆ แต่คุณสมบัติบางอย่างอาจยังไม่เทียบเท่าโปรแกรมภาษาเหล่านั้น เช่น คุณสมบัติการสืบทอด(Inheritance) สำหรับโอราเคิล 8.0.5 ยังไม่สามารถทำได้



รูปที่ 4-1 โครงสร้างของชนิดของข้อมูลในโอราเคิล 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 Built-In Datatype

Built-In Datatype เป็นชนิดของข้อมูลที่มีอยู่ในโอราเคิล ได้แก่

4.1.1 ข้อมูลชนิด Character

4.1.1.1 ข้อมูลชนิด CHAR

ชนิดของข้อมูลประเภท CHAR เป็นชนิดของข้อมูลตัวอักษรที่มีขนาดของความยาวคงที่ โดยที่เมื่อมีการสร้างตารางที่มีคอลัมน์เป็นชนิดข้อมูลประเภท CHAR จะต้องมีการกำหนดความยาวข้อมูลเป็นไบต์ระหว่าง 1 และ 2000 ไบต์ (โดยจะมีค่าเริ่มต้นเป็น 1) โดยโอราเคิลจะมีการรับประกันว่า

- เมื่อมีการ Insert และ Update แถวในตาราง ค่าที่เก็บอยู่ในคอลัมน์ที่มีข้อมูลประเภท CHAR จะมีขนาดคงที่ตามที่กำหนดไว้

- ถ้าหากข้อมูลที่จะ Insert หรือ Update มีขนาดของความยาวน้อยกว่าที่กำหนดไว้ ข้อมูลจะมีการเติมช่องว่างให้ได้ขนาดตามที่กำหนดไว้

- ถ้าหากข้อมูลที่จะ Insert หรือ Update มีขนาดของความยาวมากกว่าเนื่องจากช่องว่างตรงส่วนท้ายของข้อมูล ช่องว่างเหล่านั้นจะถูกตัดออกจนได้ขนาดตามที่กำหนดไว้

- ถ้าหากข้อมูลมีขนาดของความยาวมาก จะมีการ return ค่าเป็น error ออกมา

4.1.1.2 ข้อมูลชนิด VARCHAR2

ชนิดของข้อมูลประเภท VARCHAR2 เป็นชนิดของข้อมูลตัวอักษรที่มีขนาดของความยาวเปลี่ยนแปลงได้ โดยที่เมื่อมีการสร้างตารางที่มีคอลัมน์เป็นชนิดของข้อมูลประเภท VARCHAR2 จะต้องมีการกำหนดความยาวของข้อมูลที่มากที่สุดเป็นไบต์ โดยสามารถมีค่าได้ระหว่าง 1 และ 4000 ดังนั้นข้อมูลที่เก็บไว้จะมีขนาดเท่าใดก็ได้ที่อยู่ระหว่าง 1 และ ความยาวสูงสุด (แต่จะมีการ return ค่า error ออกมาถ้าหากข้อมูลมีขนาดใหญ่กว่าความยาวสูงสุด)

4.1.1.3 ข้อมูลชนิด VARCHAR

ชนิดของข้อมูลประเภท VARCHAR เป็นชนิดของข้อมูลที่มีลักษณะเหมือนกับชนิดของข้อมูลประเภท VARCHAR2 แต่อย่างไรก็ตามเวอร์ชันของโอราเคิลในอนาคต ชนิดของข้อมูล VARCHAR จะมีการเก็บขนาดของข้อมูลที่สามารถเปลี่ยนแปลงได้

4.1.1.4 ข้อมูลชนิด NCHAR และ NVARCHAR2

ชนิดของข้อมูลประเภท NCHAR และ NVARCHAR2 เก็บข้อมูลประเภท NLS โดยที่ชนิดของข้อมูลประเภท NCHAR เก็บข้อมูลที่มีขนาดคงที่ที่สอดคล้องกับข้อมูลประเภทความยาวคงที่ (fixed-length national character set) หรือความยาวเปลี่ยนแปลงได้ (variable-length national character set) สำหรับชนิดของข้อมูลประเภท NVARCHAR2 จะเก็บข้อมูลที่มีความยาวเปลี่ยนแปลงได้ ดังนั้นเมื่อมีการสร้างตารางที่มีคอลัมน์ของข้อมูลประเภท NCHAR หรือ NVARCHAR2 จะต้องมีการกำหนดขนาดที่มากที่สุดได้ทั้งหน่วยตัวอักษร (สำหรับข้อมูลประเภทความยาวคงที่ fixed-length national character set) หรือหน่วยไบต์ (สำหรับข้อมูลประเภทความยาวเปลี่ยนแปลงได้ variable-length national character set)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขนาดของความยาวที่มากที่สุดสำหรับชนิดของข้อมูลประเภท NCHAR คือ 2000 ไบต์ หรือจำนวนของตัวอักษรสามารถเก็บได้ 2000 ไบต์

- ขนาดของความยาวที่มากที่สุดสำหรับชนิดของข้อมูลประเภท NVARCHAR2 คือ 4000 ไบต์หรือจำนวนของตัวอักษรสามารถเก็บได้ 4000 ไบต์

4.1.1.5 ข้อมูลชนิด LONG

ชนิดของข้อมูลประเภท LONG จะใช้ในการเก็บข้อมูลที่มีขนาดมากถึง 2 จิกะไบต์

4.1.2 ข้อมูลชนิด NUMBER

ชนิดของข้อมูลประเภทตัวเลข จะใช้ในการเก็บตัวเลขจำนวนเต็ม (fixed numbers) และตัวเลขที่มีจุดทศนิยม (floating-point numbers) เก็บขนาดได้ถึง 21 ไบต์

4.1.3 ข้อมูลชนิด DATE

ชนิดของข้อมูลประเภทวัน จะเก็บข้อมูลที่เป็นวันและเวลา โดยจะมีการเก็บปี , เดือน , วัน , ชั่วโมง , นาที และวินาที มีขนาดคงที่ 7 ไบต์

4.1.4 ข้อมูลชนิด LOB

4.1.4.1 ข้อมูลชนิด BLOB

ชนิดของข้อมูลประเภท BLOB จะเก็บข้อมูลประเภทไบนารี โดยสามารถเก็บข้อมูลไบนารีได้ถึง 4 จิกะไบต์

4.1.4.2 ข้อมูลชนิด CLOB และ NCLOB

ชนิดของข้อมูล CLOB และ NCLOB จะเก็บข้อมูลประเภท ตัวอักษร ชนิดของข้อมูลประเภท CLOB จะเก็บข้อมูลประเภทที่เป็น Single-byte สำหรับชนิดของข้อมูลประเภท NCLOB จะใช้เก็บข้อมูลประเภท fixed-length หรือ NCHAR โดยชนิดของข้อมูลทั้งสองนี้สามารถเก็บข้อมูล 4 จิกะไบต์

4.1.4.3 ข้อมูลชนิด BFILE

ชนิดของข้อมูลประเภท BFILE จะเก็บข้อมูลประเภทไบนารีที่เป็นแฟ้มข้อมูลนอกฐานข้อมูล โดยในคอลัมน์ BFILE จะเก็บค่า file locator ที่ชี้ตำแหน่งของแฟ้มข้อมูล โดยชนิดของข้อมูลประเภทนี้สามารถเก็บข้อมูลได้ถึง 4 จิกะไบต์ (ชนิดของข้อมูลประเภทนี้จะเป็นอ่านได้อย่างเดียว read-only ไม่สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้)

4.1.5 ข้อมูลชนิด RAW และ LONG RAW

ชนิดของข้อมูล RAW จะใช้ในการเก็บข้อมูลไบนารีประเภทเปลี่ยนแปลงขนาดได้ โดยที่ขนาดที่มากที่สุดจะต้องมีการกำหนดไว้ โดยสามารถมีขนาดได้สูงสุดถึง 2000 ไบต์ สำหรับชนิดของข้อมูลประเภท LONG RAW จะใช้เก็บข้อมูลไบนารีประเภทเปลี่ยนแปลงขนาดได้ โดยสามารถมีขนาดของข้อมูลสูงสุดถึง 2^31-1 ไบต์หรือ 2 จิกะไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 ข้อมูลชนิด ROWID

ชนิดของข้อมูลประเภท ROWID จะใช้เก็บข้อมูลประเภทข้อมูลไบนารีที่บอกถึงตำแหน่งของแถว โดยที่มีขนาด 10 ไบต์สำหรับประเภท extended ROWID หรือ 6 ไบต์สำหรับประเภท restricted ROWID

4.1.7 ข้อมูลชนิด MLSLable

ชนิดของข้อมูลประเภท MLSLABEL เป็นชนิดของข้อมูลที่มีอยู่ใน Trusted Oracle โดยจะเก็บค่า Label ที่ถูกสร้างโดย multilevel secure operating Systems ซึ่ง Trusted Oracle จะใช้ Label นี้ในการควบคุมการเข้าถึงฐานข้อมูล

4.2 User-Defined Datatype

User-Defined Datatype เป็นชนิดข้อมูลที่ผู้ใช้กำหนดขึ้นมาเองซึ่งจะแบ่งออกเป็น 2 ชนิด คือ ออบเจกต์ไทป์ และ Collection Type

4.2.1 ออบเจกต์ไทป์ (Object Type)

ออบเจกต์ไทป์ในโอราเคิล8เป็นชนิดของข้อมูลที่ผู้ใช้กำหนดขึ้นมาใช้เอง(user-defined) ประกอบด้วย 3 ส่วน คือ

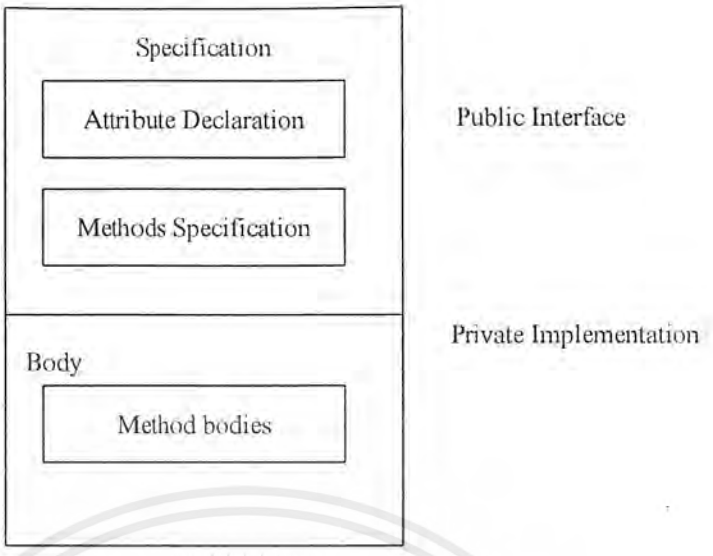
- Name เป็นชื่อที่ใช้ในการอ้างถึง ออบเจกต์ไทป์ ซึ่งจะไม่ซ้ำกันภายใน schema เดียวกัน
- แอตทริบิวต์ เป็นส่วนประกอบของออบเจกต์ไทป์ แอตทริบิวต์ มีชนิดเป็น Built-In datatype หรือ ชนิดของข้อมูลที่ผู้ใช้กำหนดขึ้นมาเอง (User-define type)
- เมทธอด เป็น ฟังก์ชัน หรือ โพรซิเดอร์ ซึ่งเขียนโดยใช้ภาษา PL/SQL และเก็บไว้ในฐานข้อมูล หรืออาจเขียนโดยใช้ภาษา C และจะถูกเก็บไว้ข้างนอกฐานข้อมูล

ออบเจกต์ไทป์ ก็คือต้นแบบ โครงสร้างของข้อมูลที่ใช้ต้นแบบนี้จะเรียกว่า ออบเจกต์

ออบเจกต์ไทป์ มีสองส่วน คือ Specification และ ส่วนของ Body ในส่วนของ Specification คือเป็นส่วนที่ผู้เขียน Application ติดต่อกับ Application ในส่วนนี้ได้กำหนดโครงสร้างข้อมูลและ เป็นส่วนที่ประกาศการกระทำ (methods) ส่วนของ Body เป็นส่วนที่สร้างเมทธอด ดังรูปที่ 4-2

ในออบเจกต์ไทป์ Specification ทุกแอตทริบิวต์จะต้องกำหนดก่อนเมทธอดเสมอ ถ้า ออบเจกต์นั้นมี เฉพาะแอตทริบิวต์ ออบเจกต์ นั้นไม่จำเป็นต้องมี Type Body เราไม่สามารถกำหนดแอตทริบิวต์ ในส่วนของ Body ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2 ส่วนประกอบของ ออบเจกต์ ใน โอราเคิล 8

ตัวอย่าง

```
create type complex as object(
        rpart real,
        ipart real,
        member function plus(x complex) return complex,
        member function less(x complex) return complex,
        member function times(x complex) return complex,
        member function divby(x complex) return complex);
create type body complex as
        member function plus(x complex) return complex is
        begin
                return complex(rpart+x.part,ipart+x.ipart);
        end plus;
        member function less(x complex) return complex is
        begin
                return complex(rpart-x,rpart,impart-x,ipart);
        end
        member function times(x Complex) return Complex is
        begin
                return Complex(rpart*x.rpart-impart*x.ipart,
                                rpart*x.ipart+ipart*x.rpart);
        end times;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
member funtion divby(x Complex) return Complex is
z REAL := x.rpart**2+x.ipart**2;
begin
        return Complex((rpart*x.rpart+ipart*x.ipart)/z
                (rpart*x.rpart-rpart*x.ipart)/z);
end divby;
end;
```

4.2.1.1 วิธีการเรียกใช้เมทธอดของออบเจกต์ไทป์

การเรียกใช้เมทธอดของออบเจกต์ไทป์เราสามารถเรียกใช้ได้แบบเดียวกับการเรียกใช้ Built-In Function ของภาษา SQL แต่เมทธอดที่สามารถเรียกใช้แบบเดียวกับการเรียกใช้ Built-In Function ของภาษา SQL จะต้องมีลักษณะดังต่อไปนี้

- เมทธอดนั้นจะต้องไม่ไปเปลี่ยนค่าข้อมูลในฐานข้อมูล (Database table) ดังนั้นในเมทธอดจะต้องไม่มีคำสั่ง INSERT,DELETE, หรือ UPDATE
- เมทธอดที่ถูกเรียกใช้จาก SELECT , VALUES หรือ SET สามารถที่จะเขียนค่าข้อมูลให้กับแอตทริบิวต์ของออบเจกต์ไทป์ได้
- เมทธอดนั้นไม่สามารถที่จะเรียกใช้โปรแกรมย่อย(subprogram) อื่น ๆ ที่ไม่มีลักษณะตามที่กล่าวมา

เพื่อที่จะทำให้เมทธอดมีลักษณะตามที่ได้กล่าวมาแล้วเราจะต้องใช้ pragma (compiler directive) RESTRICT_REFERENCES ในการกำหนดลักษณะของ method

pragma เป็นตัวบอก PL/SQL Compiler จะไม่ยอมรับเมทธอดที่มีการเขียนหรืออ่านข้อมูลในฐานข้อมูล (database table) หรือแอตทริบิวต์ของออบเจกต์หรือทั้งสองอย่าง การใช้ pragma RESTRICT_REFERENCES มีลักษณะดังนี้

```
PRAGMA RESTRICT_REFERENCES (
        Method_name,WNDS[,WNPS][,RNDS][,RNPS] );
```

- WNDS (writes no database state) คือไม่มีการเปลี่ยนแปลงข้อมูลในฐานข้อมูล (Database table) ไม่มีคำสั่ง INSERT ,DELETE หรือ UPDATE
- RNDS (read no database state) คือไม่มีการดึงข้อมูลมาจากฐานข้อมูล ซึ่งก็คือจะไม่มีคำสั่ง SELECT
- WNPS (writes no package state) คือไม่มีการเปลี่ยนแปลงค่าของตัวแปรใน แพ็คเกจ ซึ่งในที่นี้ก็คือ แอตทริบิวต์ ของ ออบเจกต์
- RNPS (read no package state) คือไม่มีการอ้างถึงค่าของตัวแปรในแพ็คเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2 Object Type Constructor Methods

ทุก ๆ ออบเจกต์ไทป์ระบบจะกำหนดให้มี Constructor ซึ่งเมทธอดนั้นจะทำการสร้าง ออบเจกต์ใหม่ขึ้นมาตามที่ได้กำหนดไว้ในส่วน Specification ของออบเจกต์ไทป์ชื่อของ constructor methods ก็คือชื่อของออบเจกต์ไทป์ นั้น

4.2.1.3 Comparision Methods

เป็นเมทธอดที่ทำหน้าที่ในการเปรียบเทียบระหว่างออบเจกต์ซึ่งโอราเคิลจะช่วยในการเปรียบเทียบข้อมูล 2 ข้อมูลซึ่งมีชนิดเป็น built-in datatype ตัวอย่างเช่นเปรียบเทียบตัวเลข 2 จำนวนและบอกว่าข้อมูลหนึ่งมากกว่า, เท่ากับ หรือน้อยกว่าอีกข้อมูลหนึ่ง โอราเคิลไม่สามารถที่จะเปรียบเทียบข้อมูลซึ่งมีชนิดเป็นข้อมูลที่กำหนดขึ้นมาเอง (User-defined type) โดยตัวเองได้ ดังนั้น โอราเคิลได้เตรียมวิธีในการบอกความสัมพันธ์ระหว่าง ออบเจกต์ ซึ่งมีชนิดเป็นข้อมูลที่กำหนดขึ้นมาเอง (User-defined type) ไว้ 2 วิธี คือ

- แม็พเมทธอด (Map Method) วิธีการนี้จะใช้ความสามารถของโอราเคิลในการเปรียบเทียบข้อมูลที่มีชนิดเป็น built-in datatype ตัวอย่างเช่น เรากำหนดออบเจกต์ไทป์สี่เหลี่ยมซึ่งมีแอตทริบิวต์เป็นความสูงและความกว้าง เราสามารถกำหนด แม็พเมทธอด พื้นที่ (area) ซึ่งส่งค่าตัวเลขที่เป็นผลคูณของความกว้าง จากนั้น โอราเคิล ก็สามารถที่จะเปรียบเทียบสี่เหลี่ยม 2 รูปโดยใช้การเปรียบเทียบพื้นที่ของสี่เหลี่ยม

ตัวอย่างการสร้างแม็พเมทธอด

```
create type purchase_order_t as object(
        Pono     number,
        Custref REF      customer_info_t,
        Orderdate        date,
        Shipdate         date,
        Map member function ret_value return number,
        Pragma RESTRICT_REFERENCES(
                Ret_value,WNDS,RNDS,WNPS,RNPS)
);
create type body purchase_order_t as
        map member function ret_value return number is
        begin
                return pono;
        end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออเดอร์เมทธอด (Order Method) ใช้วิธีการเปรียบภายในเมทธอดและส่งค่าซึ่งจะบอกความสัมพันธ์ระหว่าง ออบเจกต์ กลับออกมา เช่น ส่งค่ากลับออกมาเป็น –1 ถ้า ออบเจกต์ แรกเล็ก จะส่งค่า 0 ถ้าเท่ากัน หรือส่งค่ากลับออกมาเป็น 1 ถ้าใหญ่กว่า

ตัวอย่างการสร้าง ออเดอร์เมทธอด

```
create type customer_info_t as object (
        Custno number,
        Custname varchar2(200),
        Address address_t,
        Phone_list phone_list_t,
        Order member function cust_order( x in customer_info_t ) return integer,
        Pragma restrict_references( cust_order,WNDS,RNDS,WNPS,RNPS)
);
```

สมมติว่าเรากำหนดออบเจกต์ไทป์ Address ขึ้นมาซึ่งประกอบด้วย street,city,state และ zip คำว่ามากกว่า หรือ น้อยกว่า จะใช้กับ Address ไม่ได้แต่เราต้องการ การทำงานที่ซับซ้อนยิ่งขึ้นเพื่อให้รู้ว่า Addresses นั้นเท่านั้น ในการกำหนดออบเจกต์ไทป์ เราสามารถที่จะมีแม็พเมทธอด หรือ ออเดอร์เมทธอดก็ได้ แต่ไม่สามารถมีทั้งสองอย่างในออบเจกต์ไทป์เดียวกันได้ ถ้าออบเจกต์ไทป์ไม่มีเมทธอดที่ใช้ในการเปรียบเทียบ (comparision method) โอราเคิลไม่สามารถจะบอกความสัมพันธ์ระหว่างออบเจกต์ได้ว่าเล็กกว่า

หรือใหญ่กว่าโอราเคิลจะเปรียบเทียบ 2 ออบเจกต์ ที่ไม่มีเมทธอดที่ใช้ในการเปรียบเทียบ โดยการเปรียบเทียบแอตทริบิวต์ที่เหมือนกัน

4.2.2 Collection type

Collection type มีลักษณะที่เป็นหน่วยของข้อมูล (data unit) ที่มีจำนวนอิเลเมนต์ไม่จำกัดและอีเลเมนต์ทั้งหมดมีชนิดข้อมูลเหมือนกัน Collection type ได้แก่ array type และ table type ซึ่งหน่วยของข้อมูลจะเรียกว่า varrays และ nestable table

Collection type จะมี constructor methods ชื่อของ constructor methods ก็คือชื่อของ collection types นั้น ๆ

4.2.2.1 Varrays

มีลักษณะเป็นเซ็ตของข้อมูลแบบมีลำดับ (Array)โดยสมาชิกทุกตัวจะเป็นข้อมูลชนิดเดียวกัน สมาชิกแต่ละตัวจะมีตัวชี้ (Index) ซึ่งเป็นหมายเลขที่สอดคล้องกับตำแหน่งของสมาชิกในอาเรย์จำนวนของสมาชิกในอาเรย์จำนวนของสมาชิกในอาเรย์ก็คือขนาดของอาเรย์โอราเคิล ยอมให้ขนาดของอาเรย์เปลี่ยนแปลงได้ ซึ่งนี่ก็คือสาเหตุที่ถูกเรียกว่า varrays เราจะต้องระบุขนาดที่ใหญ่ที่สุดเมื่อเรามีการกำหนดข้อมูลชนิดอาเรย์

ตัวอย่างการกำหนดข้อมูลชนิดอาเรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

create type prices as varray(10) of number(12,2);

varray ชนิด prices จะมีจำนวนสมาชิกได้ไม่เกิน 10 และสมาชิกแต่ละตัวมีชนิดเป็น number(12,2) การสร้างข้อมูลชนิดอาเรย์จะไม่มีการจองเนื้อที่ แต่จะเป็นการกำหนดชนิดของข้อมูล ซึ่งเราสามารถนำไปใช้เป็น

- ชนิดข้อมูลของคอลัมน์ใน relational table
- แอตทริบิวต์ของออบเจกต์ไทป์
-เป็นตัวแปร PL/SQL , พารามิเตอร์ หรือ ชนิดข้อมูลที่ส่งค่ากลับมาจาก ฟังก์ชัน

4.2.2.2 Nested table

Nested Table เป็นเซ็ตของข้อมูลแบบที่ไม่เรียงลำดับ ซึ่งสมาชิกทุก ๆ ตัว จะเป็นข้อมูลชนิดเดียวกันและ Nested Table จะมีเพียง คอลัมน์ (column) เดียวและ คอลัมน์ จะเป็นข้อมูลชนิด built-in datatype หรือ เป็นออบเจกต์ไทป์ เราสามารถมองเป็นเหมือนกับเป็นตาราง ที่มีหลาย ๆ คอลัมน์ ได้ซึ่ง คอลัมน์ ก็คือ แอตทริบิวต์ ของออบเจกต์ไทป์ ตัวอย่างเช่น purchase order คำสั่งต่อไปนี้เป็นการกำหนดข้อมูลชนิดตาราง (table type) ของ lineitem

Create type lineitem_table as table of lineitem;

การสร้างข้อมูลชนิดตาราง จะไม่มีการจองเนื้อที่ แต่จะเป็นการกำหนดชนิดของข้อมูลซึ่งเราสามารถนำไปใช้เป็น

- ชนิดข้อมูลของคอลัมน์ใน relational table
- แอตทริบิวต์ของออบเจกต์ไทป์
- เป็นตัวแปร PL/SQL ,พารามิเตอร์ หรือ ชนิดของข้อมูลที่ส่งค่ากลับมาจาก ฟังก์ชัน

ตัวอย่างของการสร้างตารางซึ่งมีชนิดของข้อมูลเป็น nested table

create table purchase_order_table of purchase_order

nested table lineitem store as lineitem_table;

4.3 คุณสมบัติทางออบเจกต์โอเรียนเต็ดของออบเจกต์ไทป์ ในโอราเคิล 8

- การปกป้องข้อมูล (Encapsulation) ออบเจกต์ไทป์ ในโอราเคิล 8 จะแบ่งการ ปกป้องข้อมูล ออกเป็น 2 ส่วนคือในส่วนของ Specification จะมีการปกป้องแบบ Public และในส่วนของ Body จะมีการปกป้องเป็นแบบ private

- โพลิมอร์ฟิซึม (polymorphism) ออบเจกต์ไทป์ ในโอราเคิล 8 สามารถมีเมทธอดที่ชื่อซ้ำกันแต่มีการทำงานที่แตกต่างกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
หลักการของ Oracle8 Spatial Cartridge

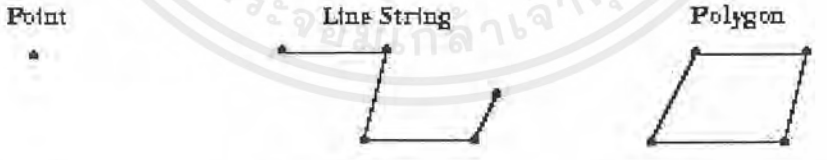
Oracle8 Spatial Cartridge เก็บรวบรวมกลุ่มของฟังก์ชัน และโพรซิเดอร์ซึ่งทำให้ข้อมูลเชิงพื้นที่สามารถที่จะจัดเก็บ , การเข้าถึงข้อมูล และการวิเคราะห์ข้อมูลได้อย่างรวดเร็ว และมีประสิทธิภาพในระบบฐานข้อมูลโอราเคิล8

Spatial Cartridge มีการออกแบบมาเพื่อใช้จัดเก็บ , ดึงข้อมูล และเพื่อการปรับปรุงแก้ไข (Manipulation) ข้อมูลเชิงพื้นที่ให้ง่ายยิ่งขึ้น อย่างเช่น ระบบสารสนเทศทางภูมิศาสตร์ (Geometric Information System : GIS) เมื่อข้อมูลนั้นได้มีการจัดเก็บลงในระบบฐานข้อมูลโอราเคิล8 การปรับปรุงแก้ไขข้อมูล และการดึงข้อมูลออกมาประมวลผลก็สามารถที่จะทำได้ง่าย ตัวอย่างของข้อมูลเชิงพื้นที่ที่เราพบเห็นกันอยู่บ่อยๆ ได้แก่ แผนที่ถนน ซึ่งแผนที่ถนนมีลักษณะเป็น 2 มิติที่ประกอบด้วย จุด , เส้น และพื้นที่รูปปิด สามารถใช้เป็นสัญลักษณ์แทนเมือง , ถนน และเขตการปกครอง อย่างเช่น รัฐ หรือ จังหวัด เป็นต้น ตำแหน่งของเมือง ถนน และเขตการปกครอง ซึ่งมีอยู่จริงบนโลก ถูกเปลี่ยนมาเป็น การแสดงผลในลักษณะ 2 มิติ หรือ เป็นข้อมูลบนแผ่นกระดาษ โดยที่ตำแหน่งและระยะทางของวัตถุจะต้องมีความสัมพันธ์กับตำแหน่งและระยะทางจริงๆ บนพื้นโลก ข้อมูลซึ่งอ้างถึงตำแหน่ง (ละติจูด,ลองจิจูด , ความสูง และความลึก) ของวัตถุที่เราต้องการบนพื้นโลกก็คือข้อมูลเชิงพื้นที่ เมื่อแผนที่ถูกนำมาแสดง ข้อมูลเชิงพื้นที่จะถูกใช้เป็นสิ่งที่ใช้อ้างถึงตำแหน่งของวัตถุบนแผนที่ 2 มิติ

ข้อมูลเชิงพื้นที่ชนิดอื่นที่สามารถจัดเก็บโดยใช้ Spatial Cartrigde รวมไปถึงข้อมูลที่ได้จาก Computer Aided Design (CAD) และระบบ Computer Aided Manufacturing (CAM)

5.1 หลักการของ Oracle8 Spatial Cartridge สำหรับโมเดลเชิงสัมพันธ์ (โอราเคิล 8.0.5)

5.1.1 ชนิดของรูปเรขาคณิต (Geometric Typess)



รูปที่ 5-1 ชนิดของรูปเรขาคณิตสำหรับโมเดลเชิงสัมพันธ์และโมเดลเชิงวัตถุสัมพันธ์

สำหรับโมเดลเชิงสัมพันธ์ Spatial Cartridge สามารถที่จะรองรับข้อมูลที่เป็นรูปเรขาคณิตได้ 3 ชนิดดังนี้

1. จุด และ กลุ่มของจุด 2 มิติ (2D Point and Point Cluster) ประกอบด้วยจุดพิกัด X และ Y โดยปกติจะสอดคล้องกับละติจูด(Latitude) และ ลองจิจูด (Longitude)
2. เส้น 2 มิติ (2D Line String) จะประกอบด้วย 1 คู่ของจุดหรือมากกว่าซึ่งประกอบกันเป็นเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. พื้นที่รูปปิด 2 มิติ (2D Polygon) ประกอบด้วยการเชื่อมต่อกันของเส้นหลายๆ เส้น รวมกันเป็นพื้นที่ปิด

5.1.2 รูปแบบของข้อมูล (Data Model)

รูปแบบข้อมูลของ Spatial Cartridge มีลักษณะเป็นลำดับชั้นของโครงสร้างที่ประกอบด้วย อีเลเมนต์ (Element) , รูปเรขาคณิต (Gemoetries) และเลเยอร์ (Layers) ซึ่งมีความสอดคล้องกับการกำหนดข้อมูลเชิงพื้นที่ เลเยอร์ของ Spatial Cartridge จะประกอบด้วยรูปทรงเรขาคณิต (วัตถุที่มีรูปทรงเป็นเรขาคณิต) ตัวอย่างเช่น จุดอาจจะใช้แทนตำแหน่งของอาคาร เส้นใช้แทนถนนหรือสายการบิน และ พื้นที่รูปปิดหมายถึง รัฐ , เมือง , เขตการปกครอง เป็นต้น

5.1.2.1 อีเลเมนต์ (Element)

อีเลเมนต์เป็นรูปทรงเรขาคณิตพื้นฐานที่ใช้สำหรับ Spatial Cartridge ชนิดของอีเลเมนต์ของ Spatial Cartridge ได้แก่ จุด , เส้น และพื้นที่รูปปิด

- ข้อมูลของจุดประกอบด้วยคู่อันดับ 1 คู่ (Coordinate)
- ข้อมูลของเส้นประกอบด้วยคู่อันดับ 2 คู่
- ข้อมูลของพื้นที่รูปปิดประกอบด้วยคู่ของคู่อันดับซึ่งแต่ละคู่จะหมายถึงเส้นซึ่งเป็นด้านของพื้นที่รูปปิด

5.1.2.2 รูปทรงเรขาคณิต (Geometry)

รูปทรงเรขาคณิตเป็นสิ่งที่ใช้แทนข้อมูลเชิงพื้นที่ของผู้ใช้ มีลักษณะเป็นเซ็ตแบบเรียงลำดับของอีเลเมนต์ วัตถุรูปทรงเรขาคณิตแต่วัตถุจะต้องมีสิ่งที่ใช้ระบุถึงวัตถุนั้นๆ ซึ่งจะต้องไม่ซ้ำกัน โดยใช้ตัวเลขในการระบุซึ่งเรียกว่า Geometry identifier (GID) ซึ่งจะใช้ในการเชื่อมโยงวัตถุนั้นๆ เข้ากับข้อมูลที่ใช้บอกคุณสมบัติ (แอตทริบิวต์) ของวัตถุนั้น แต่สำหรับโมเดลเชิงวัตถุสัมพันธ์ไม่จำเป็นต้องมี

5.1.2.3 เลเยอร์ของข้อมูลเชิงพื้นที่

เลเยอร์ของข้อมูลเชิงพื้นที่เป็นที่รวบรวมรูปทรงเรขาคณิตที่มีลักษณะของคุณสมบัติเหมือนกัน ตัวอย่างเช่น เลเยอร์ของระบบสารสนเทศทางภูมิศาสตร์อาจจะเป็นลักษณะของภูมิประเทศ ในขณะที่อีกเลเยอร์หนึ่งอาจเป็นความหนาแน่นของประชากรแต่ละเลเยอร์ของข้อมูลจะถูกเก็บในฐานข้อมูลภายในตารางที่ถูกกำหนดไว้

5.1.3 โครงสร้างของฐานข้อมูล

Spatial Cartridge จะใช้ตาราง 4 ตารางสำหรับเก็บข้อมูลเชิงพื้นที่ และเก็บอินเด็กซ์ของข้อมูล โดยในแต่ละเลเยอร์ของ Spatial Cartridge นี้จะประกอบด้วย 4 ตารางดังกล่าวนี้ สำหรับรายละเอียดของแต่ละตารางจะเป็นดังนี้

SDO_ORDCNT	SDO_LEVEL	SDO_NUMTILES	SDO_COORDSYS
Number	Number	Number	Varchar

ตารางที่ 5-1 คอลัมน์ต่าง ๆ ของตาราง <layername>_sdolayer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDO_DIMNUM	SDO_LB	SDO_UB	SDO_TOLERANCE	SDO_DIMNAME
Number	Number	Number	Number	Varchar

ตารางที่ 5-2 คอลัมน์ต่าง ๆ ของตาราง<layername>_sdodim

sdo_gid	sdo_eseq	Sdo_etype	sdo_seq	Sdo_x1	sdo_y1	...	sdo_xn	sdo_yn
Number	Number	Number	Number	Number	Number	...	Number	Number

ตารางที่ 5-3 คอลัมน์ต่าง ๆ ของตาราง<layername>_sdogeom

sdo_gid	Sdo_code	Sdo_maxcode	sdo_groupcode	sdo_meta
Number	Raw	Raw	Raw	Raw

ตารางที่ 5-4 คอลัมน์ต่าง ๆ ของตาราง<layername>_sdoindex

คอลัมน์ของแต่ละตารางถูกกำหนดไว้ดังนี้

ตาราง <layername>_SDOLAYER

- **SDO_ORDCNT** คอลัมน์ SDO_ORDCNT เป็นจำนวนของจุดพิกัดที่เก็บใน 1 แถว (row) ของตาราง <layername>_SDOGEOM ซึ่งค่าของคอลัมน์นี้จะเป็นจำนวนของคอลัมน์ที่เก็บข้อมูลเกี่ยวกับจุดพิกัดของตาราง <layername>_SDOGEOM ตัวอย่างเช่น ถ้าในตาราง <layername>_SDOGEOM เก็บข้อมูลจุดพิกัดได้ 4 จุด ค่าของคอลัมน์นี้ก็จะเป็น 8 เนื่องจากใน 1 จุดจะมีการเก็บข้อมูลเป็น 2 คอลัมน์สำหรับเก็บค่า X และ Y
- **SDO_LEVEL** คอลัมน์ SDO_LEVEL เป็นคอลัมน์ที่ใช้เก็บจำนวนครั้งของการแบ่งชั้นข้อมูลออกเป็นช่องสี่เหลี่ยมย่อยๆ ในระหว่างการทำ อินเด็กซ์ ของข้อมูลเชิงพื้นที่
- **SDO_NUMTILES** คอลัมน์เก็บจำนวนของช่องสี่เหลี่ยมที่เปลี่ยนแปลงขนาดได้ (variable size tile) ที่ถูกใช้เพื่อแบ่งแต่ละวัตถุในตาราง <layername>_SDOGEOM ค่าในคอลัมน์นี้จะมีค่าเป็น NULL เมื่อใช้ช่องสี่เหลี่ยมในการแบ่งที่มีขนาดคงที่
- **SDO_COORDSYS** คอลัมน์นี้เป็นออปชั่น ใช้บอกชื่อของระบบโคออร์ดิเนตโดยใช้มาตรฐานตัวอย่างเช่น POSC หรือ OGIS

ตาราง <layername>_SDODIM

- **SDO_DIMNUM** คอลัมน์นี้เก็บหมายเลขมิติเริ่มต้นที่ หมายเลข 1 และ เพิ่มขึ้นเรื่อยๆ
- **SDO_LB** คอลัมน์นี้เก็บขอบเขตล่างของพิกัดในมิตินี้ ตัวอย่างเช่น มิติเป็นละติจูด ขอบเขตล่างควรจะเป็น –90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **SDO_UB** คอลัมน์นี้เป็นขอบเขตบนของพิกัดในมิตินี้ ตัวอย่างเช่น ถ้ามิติเป็นลองจิจูด ขอบเขตบนจะมีค่าเป็น 180
- **SDO_TOLERANCE** คอลัมน์นี้เก็บระยะห่างระหว่างจุดสองจุดที่สามารถพิจารณาเป็นจุดเดียวกันได้ เนื่องจากการปัดเศษ ค่านี้ต้องมากกว่า 0 ตัวอย่างเช่นอาจจะมีค่าเป็น
- **SDO_DIMNAME** คอลัมน์นี้เก็บชื่อของมิติ เช่น ลองจิจูด , ละติจูด , X หรือ Y

ตาราง <layername>_SDOGEOM

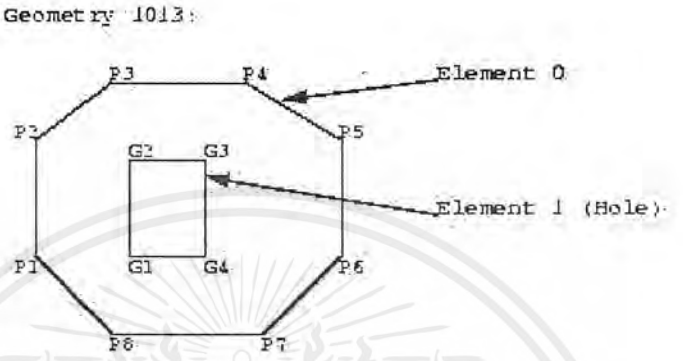
- **SDO_GID** คอลัมน์นี้เก็บหมายเลขที่ไม่ซ้ำซึ่งเป็นตัวบ่งชี้รูปทรงเรขาคณิตในเลเยอร์
- **SDO_ESEQ** คอลัมน์นี้เก็บลำดับหมายเลขของแต่ละอีเลเมนต์ (Element Sequence Number) ในรูปทรงเรขาคณิต
- **SDO_ETYPE** คอลัมน์นี้เก็บชนิดของรูปทรงเรขาคณิตพื้นฐานของอีเลเมนต์ ซึ่งจะมี 3 ชนิดคือ SDO_GEOM.POINT_TYPE (ชนิดจุด), SDO_GEOM.LINESRING_TYPE(ชนิดเส้น), SDO_GEOM.POLYGON_TYPE(ชนิดพื้นที่รูปปิด) ซึ่งอาจแทนค่าได้ด้วยค่า 1,2,3 ตามลำดับ ซึ่งหากมีค่าเป็นศูนย์จะถูกเพิกเฉย
- **SDO_SEQ (Sequence Number)** คอลัมน์นี้เก็บหมายเลขลำดับของแต่ละแถวของข้อมูลที่ทำให้เกิดอีเลเมนต์
- **SDO_X1** ค่า X ของคู่ลำดับที่ 1
- **SDO_Y1** ค่า Y ของคู่ลำดับที่ 2
- **SDO_Xn** ค่า X ของคู่ลำดับที่ n
- **SDO_Yn** ค่า Y ของคู่ลำดับที่ n

ตาราง <layername>_SDOINDEX

- **SDO_GID** คอลัมน์นี้เก็บตัวเลขที่ไม่ซ้ำบ่งชี้รูปทรงเรขาคณิตในเลเยอร์ เป็น FOREIGN KEY กลับไปยังตาราง <layername>_SDOGEOM
- **SDO_CODE** คอลัมน์นี้เก็บหมายเลขของช่องสี่เหลี่ยมย่อย (tile) ที่ปกคลุม SDO_GID จำนวนของไบต์ที่ต้องใช้สำหรับคอลัมน์ SDO_CODE และ SDO_MAXCODE ขึ้นอยู่กับระดับของการแบ่งช่องสี่เหลี่ยมย่อยซึ่งเราสามารถใช้ฟังก์ชัน SDO_ADMIN.SD_CODE_SIZE เพื่อหาขนาดที่ต้องการในแต่ละเลเยอร์ได้ จำนวนไบต์สูงสุดที่เป็นไปได้คือขนาด 255 ไบต์
- **SDO_MAXCODE** คอลัมน์นี้เก็บค่าของช่องสี่เหลี่ยมที่สามารถเปลี่ยนแปลงขนาดได้ซึ่งเป็นช่องสี่เหลี่ยมย่อยที่เล็กที่สุดในควอทแดรท์ปัจจุบัน คอลัมน์นี้จะไม่ถูกใช้สำหรับ fixed-size tile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **SDO_GROUPCODE** คอลัมน์นี้เก็บส่วนหน้า(prefix) ของ SDO_CODE แทนค่า variable-size tile ที่ระดับ <layername>_SDOLAYER ,SDO_LAYER ซึ่งบรรจุหรือเท่ากับช่องสี่เหลี่ยมที่ถูกแทนโดย SDO_CODE คอลัมน์นี้ไม่ถูกใช้สำหรับ fixed-size tile
- **SDO_META** คอลัมน์นี้ไม่จำเป็นสำหรับการเรียกข้อมูลเชิงพื้นที่ แต่จะจัดเตรียมข้อมูลที่จำเป็นในการหาขอบเขตของช่องสี่เหลี่ยม(tile)



รูปที่ 5-2 ตัวอย่างรูปทรงเรขาคณิต

SDO_ORDCNT (number)
4

ตารางที่ 5-5 ตัวอย่างข้อมูลในตาราง <layername>_sdolayer

SDO_DIMNUM (number)	SDO_LB (number)	SDO_UB (number)	SDO_TOLERANCE (number)	SDO_DIMNAME (varchar)
1	0	100	.05	X axis
2	0	100	.05	Y axis

ตารางที่ 5-6 ตัวอย่างข้อมูลในตาราง <layername>_sdodim

SDO_GID (number)	SDO_ESEQ (number)	SDO_ETYPE (number)	SDO_SEQ (number)	SDO_X1 (number)	SDO_Y1 (number)	SDO_X2 (number)	SDO_Y2 (number)
1013	0	3	0	P1(X)	P1(Y)	P2(X)	P2(Y)
1013	0	3	1	P2(X)	P2(Y)	P3(X)	P3(Y)
1013	0	3	2	P3(X)	P3(Y)	P4(X)	P4(Y)
1013	0	3	3	P4(X)	P4(Y)	P5(X)	P5(Y)
1013	0	3	4	P5(X)	P5(Y)	P6(X)	P6(Y)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1013	0	3	5	P6(X)	P6(Y)	P7(X)	P7(Y)
1013	0	3	6	P7(X)	P7(Y)	P8(X)	P8(Y)
1013	0	3	7	P8(X)	P8(Y)	P1(X)	P1(Y)
1013	1	3	0	G1(X)	G1(Y)	G2(X)	G2(Y)
1013	1	3	1	G2(X)	G2(Y)	G3(X)	G3(Y)
1013	1	3	2	G3(X)	G3(Y)	G4(X)	G4(Y)
1013	1	3	3	G4(X)	G4(Y)	G1(X)	G1(Y)

ตารางที่ 5-7 ตัวอย่างข้อมูลในตาราง<layername>_sdogeom

5.1.4 การป้อนข้อมูลเชิงพื้นที่ลงในฐานข้อมูล

5.1.4.1 รูปแบบของการป้อนข้อมูล

มี 2 ขั้นตอนที่เกี่ยวข้องกับการนำข้อมูลเข้าสู่ระบบฐานข้อมูลเชิงพื้นที่ (Spatial Database)

1. การนำข้อมูลเชิงพื้นที่ไปเก็บในตาราง
2. การสร้างหรือการเปลี่ยนแปลงอินเด็กซ์ของตาราง

5.1.4.2 กระบวนการการป้อนข้อมูล (Load Process)

กระบวนการในการป้อนข้อมูลสามารถแบ่งออกได้เป็น 2 วิธีคือ

- *การป้อนข้อมูลจำนวนมาก (Bulk Loading)* กระบวนการนี้จะใช้ในการป้อนข้อมูลจำนวนมากๆ ลงในฐานข้อมูล และใช้ SQL *Loader ในการป้อนข้อมูล
- *การป้อนข้อมูลโดยวิธี Transactional Inserts* กระบวนการนี้ใช้ในการป้อนข้อมูลจำนวนน้อยๆ ลงในฐานข้อมูล และใช้คำสั่ง Insert ในภาษา SQL

5.1.4.2.1 การป้อนข้อมูลจำนวนมาก (Bulk Loading) สามารถใช้ในการนำเข้าข้อมูลที่มีอยู่เดิมหรือข้อมูลที่เป็นแอสกี (ASCII) ได้ การป้อนข้อมูลจำนวนมากจะทำให้สำเร็จได้ก็โดยการใช้โปรแกรม SQL *Loader ดังตัวอย่างต่อไปนี้จะแสดงให้เห็นรูปแบบไฟล์ที่ใช้ในการนำข้อมูลเข้าสู่ตาราง SDOGEOM ซึ่งอยู่ในเลเยอร์ ROADS เราสามารถเลือกรูปแบบของข้อมูลแอสกี โดยการเขียน SQL *Loader ไฟล์เพื่อใช้ในการนำข้อมูลเข้าสู่ตาราง

สมมติว่าข้อมูลแอสกี ประกอบด้วยไฟล์ซึ่งมีการแบ่งข้อมูลเป็นคอลัมน์ และแบ่งแยกข้อมูลออกเป็นแถวๆ ซึ่งถูกกำหนดโดยรูปแบบของตารางดังต่อไปนี้

```
geometry rows : GID , ESEQ , ETYPE , SEQ , LON1 , LAT1 , LON2 , LAT2
LOAD DATA INFILE *
INTO TABLE ROADS_SDOGEOM
FIELDS TERMINATED BY WHITESPACE TRAILING NULLCOLS
(SDO_GID INTEGER EXTERNAL ,
SDO_ESEQ INTEGER EXTERNAL ,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SDO_ETYPE INTEGER EXTERNAL ,
SDO_SEQ INTEGER EXTERNAL ,
SDO_X1 FLOAT EXTERNAL ,
SDO_Y1 FLOAT EXTERNAL ,
SDO_X2 FLOAT EXTERNAL ,
SDO_Y2 FLOAT EXTERNAL )

BEGINDATA
1 0 3 0 –122.401200 37.805200 -122.401900 37.805200
1 0 3 1 –122.401900 37.805200 -122.402400 37.805500
1 0 3 2 –122.402400 37.805500 -122.403100 37.806000
1 0 3 3 –122.403100 37.806000 -122.404400 37.806800
1 0 3 4 –122.404400 37.806800 -122.401200 37.805200
1 1 3 0 –122.405900 37.806600 -122.407549 37.806394
1 1 3 1 –122.407549 37.806394 -122.408300 37.806300
1 1 3 2 –122.408300 37.806300 -122.409100 37.806200
1 1 3 3 –122.409100 37.806200 -122.405900 37.806600
2 0 3 0 –122.410800 37.806000 -122.412300 37.805800
2 0 3 1 –122.412300 37.805800 -122.414100 37.805600
2 0 3 2 –122.414100 37.805600 -122.412300 37.805800
2 0 3 3 –122.412300 37.805800 -122.410800 37.806000
3 0 1 0 –122.567474 37.643564
3 0 1 1 –122.345345 37.345345
```

5.1.4.2.2 การป้อนข้อมูลโดยวิธี Transactional Inserts โดยใช้ SQL

Spatial Cartridge ใช้ตารางมาตรฐานของโอราเคิล8 ดังนั้นจึงสามารถที่จะเข้าถึงข้อมูลโดยใช้ SQL มาตรฐาน ดังตัวอย่างเป็นการป้อนข้อมูลสำหรับ รูปทรงเรขาคณิต (GID 17) ซึ่งประกอบด้วยพื้นที่รูปปิดที่มีด้าน 4 ด้าน และให้สังเกตุว่าจุดพิกัดจุดแรกของพื้นที่รูปปิดเป็น (5,20) และจะวนมาจบที่จุดเริ่มต้นเป็นการครบรอบของพื้นที่รูปปิด

```
INSERT INTO SAMPLE_SDOGEOM
                    ( SDO_GID , SDO_ESEQ , SDO_ETYPE , SDO_SEQ ,
                      SDO_X1 , SDO_Y1 , SDO_X2 , SDO_Y2 , SDO_X3 , SDO_Y3 ,
                      SDO_X4 , SDO_Y4 , SDO_X5 , SDO_Y5)
VALUES ( 17 , 0 , 3 , 0 , 5 , 20 , 5 , 30 , 10 , 30 , 20 , 5 , 20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง INSERT ของ SQL จะทำการใส่ข้อมูลลงฐานข้อมูลทีละ 1 แถว ต่อการเรียกใช้ 1 ครั้ง ในตัวอย่างที่ผ่านมา ตารางดังกล่าวมีจำนวนคอลัมน์เพียงพอที่จะเก็บพื้นที่รูปปิดได้ภายในแถวเดียว อย่างไรก็ตามถ้าตารางของเรามีจำนวนคอลัมน์ไม่เพียงพอ เราก็ต้องทำการป้อนข้อมูลโดยใช้คำสั่ง INSERT ของ SQL หลายๆ ครั้ง ในการนำข้อมูลรูปทรงเรขาคณิตขนาดใหญ่เข้าสู่ฐานข้อมูล ทำได้โดยการใส่ SDO_GID , SDO_ESEQ และ SDO_TYPE และเพิ่มค่า SDO_SEQ ขึ้นไปเรื่อยดังตัวอย่าง

```
INSERT INTO SAMPLE2_SDOGEOM
                    ( SDO_GID , SDO_ESEQ , SDO_ETYPE , SDO_SEQ ,
                      SDO_X1 , SDO_Y1 , SDO_X2 , SDO_Y2 , SDO_X3 , SDO_Y3 ,
                      SDO_X4 , SDO_Y4 , SDO_X5 , SDO_Y5)
VALUES ( 18 , 0 , 3 , 0 , 1 , 15 , 1 , 16 , 2 , 17 , 3 , 17 , 4 , 18 );

INSERT INTO SAMPLE2_SDOGEOM
                    ( SDO_GID , SDO_ESEQ , SDO_ETYPE , SDO_SEQ ,
                      SDO_X1 , SDO_Y1 , SDO_X2 , SDO_Y2 , SDO_X3 , SDO_Y3 ,
                      SDO_X4 , SDO_Y4 , SDO_X5 , SDO_Y5)
VALUES ( 18 , 0 , 3 , 1 , 4 , 18 , 5 , 18 , 6 , 19 , 7 , 18 , 6 , 17);

INSERT INTO SAMPLE2_SDOGEOM
                    ( SDO_GID , SDO_ESEQ , SDO_ETYPE , SDO_SEQ ,
                      SDO_X1 , SDO_Y1 , SDO_X2 , SDO_Y2 , SDO_X3 , SDO_Y3 ,
                      SDO_X4 , SDO_Y4 , SDO_X5 , SDO_Y5)
VALUES ( 18 , 0 , 3 , 2 , 6 , 17 , 7 , 16 , 7 , 15 , 6 , 14 , 7 , 13 );

INSERT INTO SAMPLE2_SDOGEOM
                    ( SDO_GID , SDO_ESEQ , SDO_ETYPE , SDO_SEQ ,
                      SDO_X1 , SDO_Y1 , SDO_X2 , SDO_Y2 , SDO_X3 , SDO_Y3 ,
                      SDO_X4 , SDO_Y4 , SDO_X5 , SDO_Y5)
VALUES ( 18 , 0 , 3 , 3 , 7 , 13 , 6 , 12 , 5 , 13 , 4 , 13 , 3 , 14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INSERT INTO SAMPLE2_SDOGEOM
                    ( SDO_GID , SDO_ESEQ , SDO_ETYPE , SDO_SEQ ,
                      SDO_X1 , SDO_Y1 , SDO_X2 , SDO_Y2 , SDO_X3 , SDO_Y3 )
VALUES ( 18 , 0 , 3 , 4 , 3 , 14 , 2 , 14 , 1 , 15 );
```

5.1.4.2.3 การป้อนข้อมูลโดยวิธี Transactional Inserts โดยใช้ฟังก์ชันของ Spatial Geometry

Spatial Cartridge ได้จัดเตรียมฟังก์ชันไว้ 2 ฟังก์ชันเพื่อใช้อำนวยความสะดวกในป้อนข้อมูลลงในตารางที่เก็บข้อมูลเชิงพื้นที่ ข้อดีของการใช้ฟังก์ชันนี้ก็คือเหมาะสำหรับการป้อนข้อมูลของอีเลเมนต์ที่มีหลายๆ จุดพิกัดได้โดยไม่ต้องสนใจว่าใน 1 แถว ของตารางที่ใช้เก็บข้อมูลจะมีจำนวนคอลัมน์ที่ใช้เก็บจุดพิกัดอยู่เท่าไหร่เนื่องจากฟังก์ชันนี้จะทำการตรวจสอบเองโดยอัตโนมัติ

ขั้นตอนในการนำข้อมูลเข้าสู่ตารางสำหรับเก็บข้อมูลเชิงพื้นที่

1. ทำการกำหนดค่าเริ่มต้นให้กับอีเลเมนต์ที่ต้องการจะนำไปเก็บในตาราง แต่ในขั้นตอนนี้ไม่ได้ใส่ข้อมูลจุดพิกัดสำหรับอีเลเมนต์นี้ลงในตารางพารามิเตอร์ที่ส่งเข้าไปในฟังก์ชัน SDO_GEOM.INIT_ELEMENT() ซึ่งใช้ในการกำหนดค่าเริ่มต้นให้กับอีเลเมนต์นี้

- ชื่อของเลเยอร์ตัวอย่างเช่น ROADS
- GID ซึ่งเป็นหมายเลขที่ใช้ในการระบุถึงรูปทรงเรขาคณิตนั้นๆ

ฟังก์ชัน SDO_GEOM.INIT_ELEMENT() จะส่งค่าเลขลำดับของอีเลเมนต์มาให้ ซึ่งค่าของเลขลำดับนี้จะใช้พารามิเตอร์ส่งให้กับโพรซิเดอร์ SDO_GEOM.ADD_NODES()

2.ทำการใส่ค่าจุดพิกัดของอีเลเมนต์นี้โดยใช้โพรซิเดอร์ SDO_GEOM.ADD_NODES() ซึ่งโพรซิเดอร์นี้มีพารามิเตอร์ดังนี้

- ชื่อของเลเยอร์
- GID
- เลขลำดับของอีเลเมนต์
- ชนิดของอีเลเมนต์ (element type)
- ค่าจุดพิกัดของอีเลเมนต์

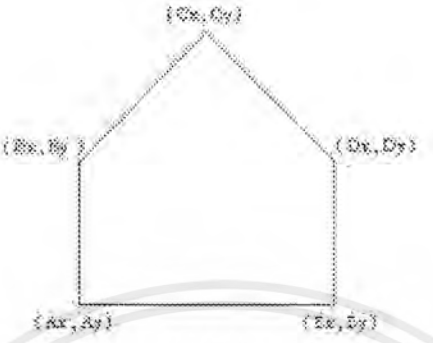
ตัวอย่างต่อไปนี้เป็นพื้นที่รูปปิดแบบง่ายๆ , มีหมายเลข GID เป็น 1234 ประกอบด้วยด้านของรูป 5ด้านขั้นตอนแรกจะเป็นการกำหนดค่าเริ่มต้นให้กับอีเลเมนต์โดยจะมีการใช้ฟังก์ชันที่ชื่อว่า SDO_GEOM.INIT_ELEMENT() ในการกำหนดค่าเริ่มต้นให้กับอีเลเมนต์

```
elem_value := sdo_geom.init_element('ROADS' , 1234 );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเป็นการป้อนข้อมูลของพื้นที่รูปปิดโดยใช้โพรซิเดอร์ SDO_GEOM.ADD_NODES()

```
sdo_geom.add_nodes('ROADS' , 1234 , elem_value , sdo_geom.polygon_type,
Ax , Ay , Bx , By , Cx , Cy , Dx , Dy , Ex , Ey , Ax , Ay);
```



รูปที่ 5-3 ข้อมูลชนิด พื้นที่รูปปิด

5.1.5 วิธีการทำอินเด็กซ์ (Indexing Method)

อินเด็กซ์เชิงพื้นที่ (Spatial Index) จัดเป็น logical index ซึ่งตรงกันข้ามกับ physical index รายการข้อมูลในอินเด็กซ์เชิงพื้นที่จะขึ้นอยู่กับตำแหน่งของวัตถุรูปทรงเรขาคณิตในเลเยอร์ แต่จะไม่ขึ้นอยู่กับตำแหน่งที่ถูกเก็บลงบนดิสก์ นั่นหมายความว่าตารางที่บรรจุข้อมูลเชิงพื้นที่สามารถโยกย้ายหรือแตกย่อยได้ โดยที่อินเด็กซ์เชิงพื้นที่ไม่จำเป็นต้องถูกสร้างใหม่

Spatial Cartridge มี 2 วิธีการ ในการสร้างอินเด็กซ์เชิงพื้นที่ได้แก่

1. การแบ่งช่องสี่เหลี่ยมขนาดคงที่ (fixed-size tiling)
2. การแบ่งช่องสี่เหลี่ยมที่มีขนาดเปลี่ยนแปลงได้ (variable-size tiling)

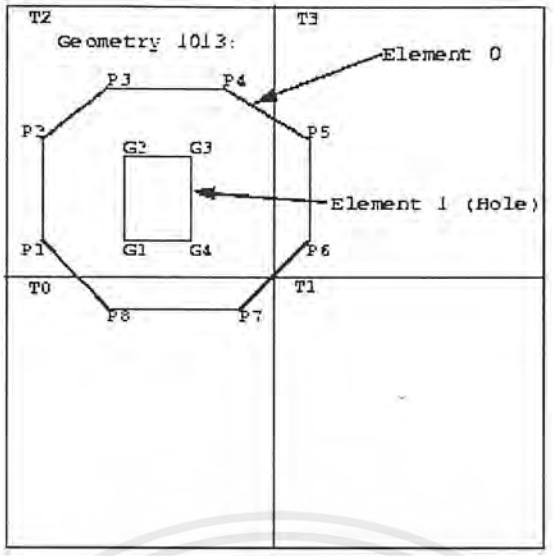
ในการทำอินเด็กซ์เชิงพื้นที่ Object space(เลเยอร์ที่ทุกๆวัตถุรูปทรงเรขาคณิตตั้งอยู่) จะถูกนำไปผ่านกระบวนการที่เรียกว่า "tessellation" ซึ่งจะทำการแบ่งช่องสี่เหลี่ยม(tile) ให้ครอบคลุมทุกๆอีเลเมนต์ ซึ่งสามารถใช้ได้ทั้ง fixed-size tile และ variable-size tile

จำนวนของช่องสี่เหลี่ยมที่ถูกใช้เพื่อปกคลุมอีเลเมนต์เป็นพารามิเตอร์ที่ผู้ใช้สามารถปรับได้ การใช้ช่องสี่เหลี่ยมที่มีขนาดเล็กจะละเอียดกว่าการใช้ช่องสี่เหลี่ยมที่มีขนาดใหญ่

5.1.5.1 การแบ่งช่องสี่เหลี่ยมในเลเยอร์ (tessellation of a layer)

การแบ่งช่องสี่เหลี่ยม(tessellation)คือ กระบวนการในการหาว่า มีช่องสี่เหลี่ยมใดที่ปกคลุมอีเลเมนต์ที่กำหนด ในการทำกระบวนการนี้พื้นที่ที่วัตถุอยู่ในเลเยอร์(Object space) จะถูกแบ่งเป็นช่องสี่เหลี่ยมที่มีขนาดเท่าๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 การแบ่งเลเยอร์

จากรูปเลเยอร์นี้จะถูกทำการแบ่งเลเยอร์(tessellation) ออกเป็นช่องสี่เหลี่ยมเท่าๆกันสี่ช่อง หลังการทำข้อมูลในตาราง <layername>_SDOINDEX จะมีข้อมูลดังตารางข้างล่าง จากตารางจะสังเกตได้ว่ารูปทรงเรขาคณิตหมายเลข SDO_GID=1013 มีอีเลเมนต์อยู่ในช่องสี่เหลี่ยมหมายเลข T0,T2,T3

SDO_GID <number>	SDO_CODE <raw>
1013	T0
1013	T2
1013	T3

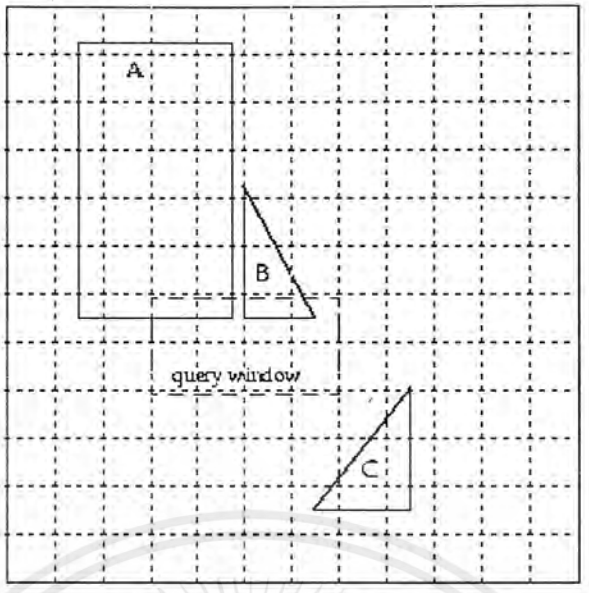
ตารางที่ 5-8 ตัวอย่างข้อมูลในตาราง <layer_name>_sdoindex

5.1.5.2 การทำอินเด็กซ์เชิงพื้นที่โดยใช้ช่องสี่เหลี่ยมขนาดคงที่ (Fixed-size tile spatial indexing)

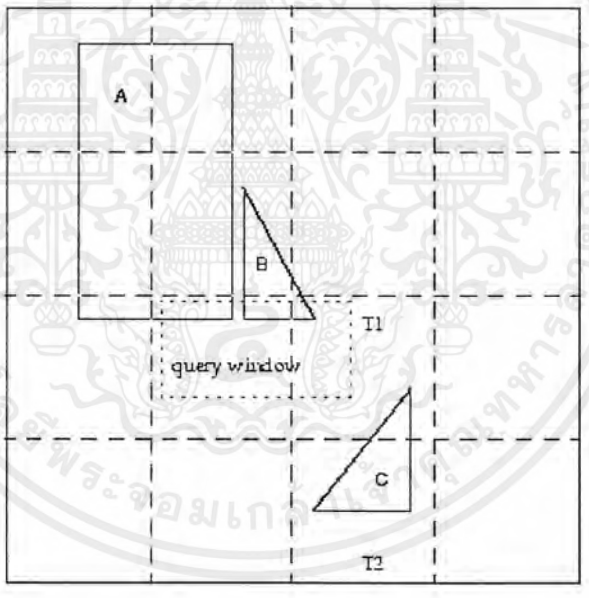
วิธีนี้เป็นวิธีที่แนะนำให้ใช้ ขนาดของช่องสี่เหลี่ยมที่ใช้ในการแบ่งพื้นที่เลเยอร์จะมีขนาดเท่ากัน จึงทำให้คำสั่งภาษา SQL สามารถใช้ตัวดำเนินการเท่ากับ(=) เพื่อเปรียบเทียบช่องสี่เหลี่ยมในการ join ได้ ซึ่งจะได้ประสิทธิภาพดี

ถ้าหากเราใช้ช่องสี่เหลี่ยมขนาดเล็กเพื่อปกคลุมรูปทรงเรขาคณิตที่มีขนาดเล็ก และใช้ช่องสี่เหลี่ยมขนาดเดียวกันนี้กับรูปทรงเรขาคณิตขนาดใหญ่ จะทำให้ต้องใช้จำนวนช่องสี่เหลี่ยมมาก ส่งผลให้ขนาดของตารางอินเด็กซ์มีขนาดใหญ่มากขึ้น แต่ถึงอย่างไรก็ตามการใช้ช่องสี่เหลี่ยมที่มีขนาดใหญ่จำนวนช่องจะลดลงแต่ก็จะไม่เหมาะกับรูปทรงเรขาคณิตที่มีขนาดเล็กดังรูปที่ 5-5 ฟังก์ชัน SDO_TUNE.ESTIMATE_TILEING_LEVEL() จะใช้ช่วยในการหาระดับของการทำการแบ่งที่เหมาะสม ซึ่งจะได้จำนวนช่องสี่เหลี่ยมที่พอเหมาะสำหรับชุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-5 การแบ่งพื้นที่เลเยอร์ที่ใช้ช่องสี่เหลี่ยมขนาดเล็ก



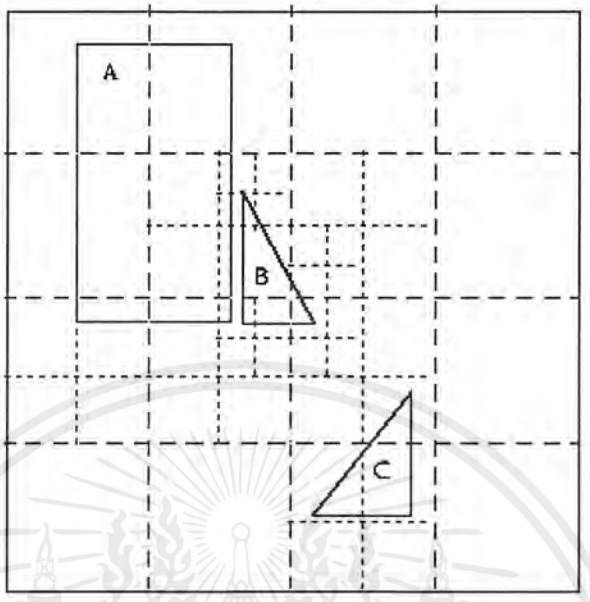
รูปที่ 5-6 การแบ่งพื้นที่เลเยอร์ที่ใช้ช่องสี่เหลี่ยมขนาดใหญ่

5.1.5.3 การทำอินเด็กซ์เชิงพื้นที่โดยใช้ช่องสี่เหลี่ยมที่มีขนาดเปลี่ยนแปลงได้ (variable-size tile spatial indexing)

โอราเคิลเองไม่แนะนำให้ใช้วิธีนี้ (มีไว้เพื่อทดลองเท่านั้น) วิธีนี้จะใช้ช่องสี่เหลี่ยมที่มีขนาดแตกต่างกันปกคลุมทรงเรขาคณิต ขนาดของช่องสี่เหลี่ยมจะขึ้นอยู่กับขนาดและรูปร่างของรูปทรงเรขาคณิตที่กำลังถูกทำอินเด็กซ์ รูปทรงที่มีขนาดใหญ่ควรใช้ช่องสี่เหลี่ยมที่มีขนาดใหญ่ และรูปทรงที่มีขนาดเล็กก็ควรใช้ช่องสี่เหลี่ยมที่มีขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนของช่องสี่เหลี่ยมที่ต้องใช้ในการปกคลุมรูปทรงเรขาคณิตถูกควบคุมโดยคอลัมน์ SDO_NUMTILES ในตาราง <layername>_SDOLAYER



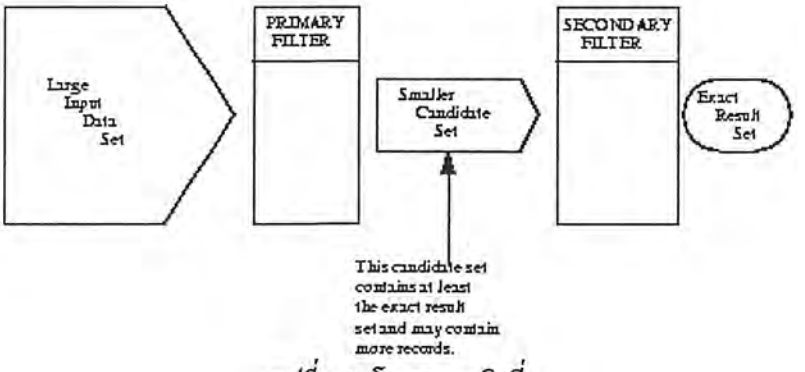
รูปที่ 5-7 การแบ่งพื้นที่เลเยอร์โดยใช้วิธี variable-size tiling

5.1.6 การคิวรีข้อมูลเชิงพื้นที่ (Querying Spatial Data)

การคิวรี่จะแบ่งการทำงานออกเป็น 2 ส่วน

5.1.6.1 Primary filter เป็นการคิวรี่ข้อมูลเชิงพื้นที่แบบคร่าวๆ เพื่อให้ได้ผลลัพธ์ออกมาในจำนวนที่น้อยลง เพื่อส่งต่อให้แก่ Secondary filter เหตุที่ต้องทำ Primary filter ก็เพื่อลดข้อมูลที่ Secondary filter จะต้องพิจารณาให้น้อยลง

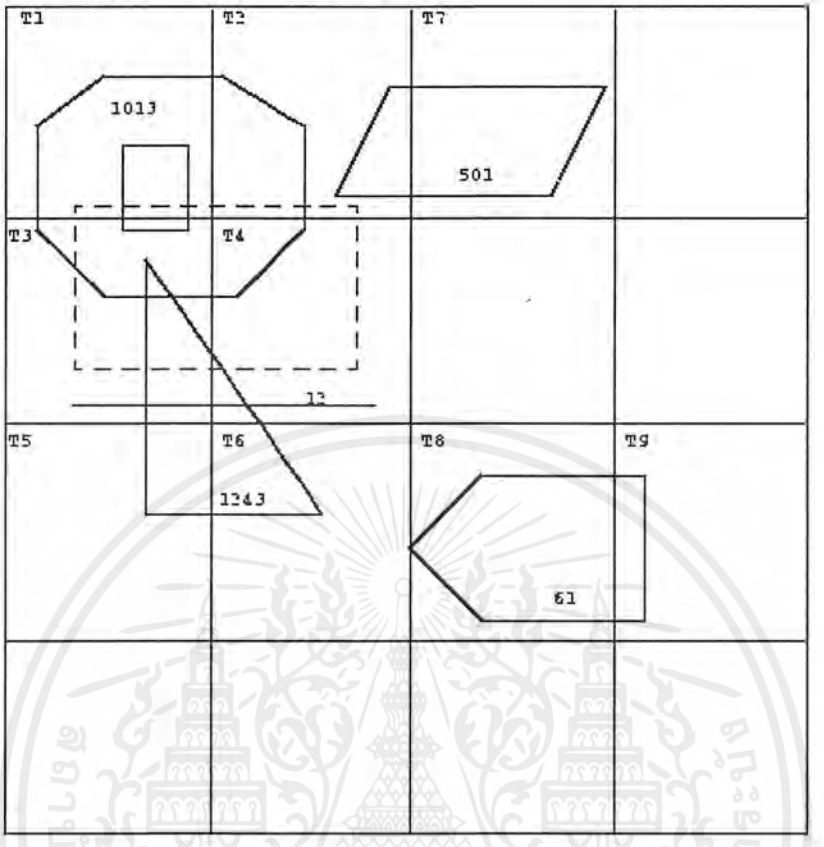
5.1.6.2 Secondary filter เป็นการคิวรี่ข้อมูลจากผลลัพธ์ที่มาจาก Primary filter การทำ Secondary filter จะมีการเรียกใช้โอเปอเรชันการคำนวณ ซึ่งมีค่าใช้จ่าย (Cost) สูง ถ้าใช้กับข้อมูลปริมาณมาก แต่เนื่องจากเราประยุกต์ใช้กับข้อมูลปริมาณที่น้อยลงที่ได้มาจาก Primary filter ทำให้ค่าใช้จ่ายน้อยลง



รูปที่ 5-8 โมเดลการคิวรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคิวรี่ข้อมูลเชิงพื้นที่เป็นการหาว่ามีรูปทรงเรขาคณิตใดบ้างที่อยู่ภายในหน้าต่างคิวรี่ (Query window) ดังเส้นประในรูป หน้าต่างนี้เรียกว่า Fence



รูปที่ 5-9 ตัวอย่างหน้าต่างคิวรี่

ถ้าหน้าต่างคิวรี่ยังไม่มีเก็บในฐานข้อมูล ขั้นแรกจะต้องสร้างเลเยอร์สำหรับหน้าต่างคิวรี่ขึ้นมาก่อน โดยใช้คำสั่งดังนี้

SQL> EXECUTE MDSYS.SDO_WINDOW.CREATE_WINDOW_LAYER (fencelayer, DIMNUM1, LB1,UB1, TOLERANCE1, DIMNAME1, DIMNUM2, LB2, UB2, TOLERANCE2, DIMNAME2);

จากนั้น INSERT คู่ลำดับของหน้าต่างคิวรี่ลงในตาราง

SQL> EXECUTE DBMS_OUTPUT.PUTLINE(MDSYS.SDO_WINDOW.BUILD_WINDOW_FIXED (comp_user, fencelayer, SDO_ETYPE, TILE_SIZE, X1,Y1, X2,Y2, X3,Y3, X4,Y4, X1,Y1));

นำค่า SDO_LEVEL จากตาราง <fencelayer>_SDOLAYER มาใส่ที่ TILE_SIZE ซึ่งเป็นพารามิเตอร์หนึ่งของโพรซิเดอร์ SDO_WINDOW.BUILD_WINDOW_FIXED()

จากนั้นทำ primary filter โดยการหาว่ามีรูปทรงเรขาคณิตใดบ้างที่อยู่ในช่องสีเหลี่ยมเดียวกันกับหน้าต่างคิวรี่ โดยใช้คำสั่งภาษา SQL ดังนี้

```
SELECT DISTINCT A.SDO_GID
FROM <layer1>_SDOINDEX A, <fencelayer>_SDOINDEX B
WHERE          A.SDO_CODE = B.SDO_CODE
 AND B.SDO_GID = {GID returned from SDO_WINDOW.BUILD_WINDOW_FIXED};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่ได้จากการทำ primary filter จะได้ {1013,501,1243,12}

จากนั้นทำ secondary filter โดยใช้ผลลัพธ์จาก primary filter ดังนี้

```
SELECT SDO_X1, SDO_Y1, SDO_X2, SDO_Y2, SDO_X3, SDO_Y3, SDO_X4, SDO_Y4
FROM <layer1>_SDOGEOM,
(
SELECT SDO_GID GID1
  FROM (
      SELECT DISTINCT A.SDO_GID
        FROM <layer1>_SDOINDEX A,
          <fencelayer>_SDOINDEX B
        WHERE A.SDO_CODE = B.SDO_CODE
        AND B.SDO_GID = {GID returned from SDO_WINDOW.BUILD_WINDOW_FIXED}
     )
  WHERE SDO_GEOM.RELATE('<layer1>', SDO_GID, 'ANYINTERACT', '<fence>', 1) =
'TRUE'
  )
WHERE SDO_GID = GID1;
```

จะได้ GID ที่ซ้อนทับกันกับหน้าต่างดิวรี่ ดังนี้ {1213,1013}

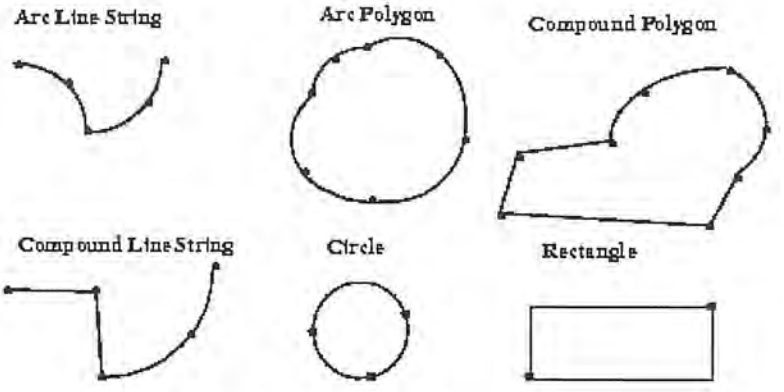
5.2 หลักการของ Oracle8 Spatial Cartridge สำหรับโมเดลเชิงวัตถุสัมพันธ์ (โอราเคิล8i)

5.2.1 ชนิดของรูปเรขาคณิต

สำหรับโมเดลเชิงวัตถุสัมพันธ์ สามารถรองรับข้อมูลรูปเรขาคณิตได้ทั้ง 3 ชนิดข้างต้นดังรูปที่5-1 รวมทั้งยังสามารถรองรับรูปทรงเรขาคณิตเพิ่มเติมอีกดังนี้

1. เส้นโค้ง 2 มิติ (2-D Arc Line String) ประกอบด้วย 1 คู่ของจุดหรือมากกว่า ระหว่างจุด 2 จุด จะต่อกันด้วยเส้นโค้ง
2. พื้นที่รูปปิดโค้ง 2 มิติ (2-D Arc Polygon) ประกอบด้วยการเชื่อมต่อกันของเส้นโค้งหลายๆ เส้น รวมกันเป็นพื้นที่ปิด
3. พื้นที่รูปปิดผสม 2 มิติ (2-D Compound Polygon) ประกอบด้วยการเชื่อมต่อกันของเส้นตรงและเส้นโค้งหลายๆ เส้น รวมกันเป็นพื้นที่ปิด
4. เส้นโค้งผสม 2 มิติ (2-D Compound Line String) ประกอบด้วย 1 คู่ของจุดหรือมากกว่าซึ่งระหว่างจุด 2 จุดจะต่อกันด้วยเส้นตรงหรือเส้นโค้ง
5. วงกลม 2 มิติ (2-D Circle)
6. สี่เหลี่ยม 2 มิติ (2-D Rectangle)ประกอบด้วยจุดมุมซ้ายล่าง และมุมขวาบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-10 ชนิดของรูปเรขาคณิตสำหรับโมเดลเชิงวัตถุสัมพันธ์ที่เพิ่มเติมเข้ามา

5.2.2 รูปแบบของข้อมูล

มีรูปแบบเหมือนกับของโมเดลเชิงสัมพันธ์ คือ ประกอบไปด้วยอีเลเมนต์ , รูปทรงเรขาคณิต และเลเยอร์

5.2.3 โครงสร้างฐานข้อมูล

รูปทรงเรขาคณิตจะถูกเก็บเป็นออบเจกต์ในคอลัมน์ที่มีชนิดเป็นออบเจกต์ไทป์ SDO_GEOMETRY ในตารางที่ผู้ใช้กำหนดเอง แต่ละรูปทรงเรขาคณิตจะเก็บเพียงแถวเดียว แทนที่จะเก็บเป็นหลายๆ แถวและคอลัมน์เหมือนในตาราง <layername>_SDOGEOM ของโมเดลเชิงสัมพันธ์ ชื่อตารางไม่จำเป็นต้องมี "_SDOGEOM" ต่อท้าย เพราะออบเจกต์ไทป์ SDO_GEOMETRY ไม่มีแอตทริบิวต์ SDO_GID ตารางใดๆ ที่มีคอลัมน์ที่เป็นชนิด SDO_GEOMETRY จะต้องมีคอลัมน์หรือกลุ่มของคอลัมน์ที่เป็นไพรมารีคีย์ที่ไม่ซ้ำค่า (Unique primary key) ตารางนี้เรียกว่าเป็นตารางรูปทรงเรขาคณิต (Geometry Table)

ในโอราเคิล 8i Spatial ได้กำหนดออบเจ็กต์ไทป์ SDO_GEOMETRY ดังนี้

```
CREATE TYPE sdo_geometry AS OBJECT (
SDO_GTYPE NUMBER,
SDO_SRID NUMBER,
SDO_POINT SDO_POINT_TYPE,
SDO_ELEM_INFO MDSYS.SDO_ELEM_INFO_ARRAY,
SDO_ORDINATES MDSYS.SDO_ORDINATE_ARRAY);
```

แอตทริบิวต์ต่างๆ ของ SDO_GEOMETRY มีความหมายดังนี้

- **SDO_GTYPE** บ่งบอกชนิดของรูปทรงเรขาคณิต ซึ่งมีค่าต่างๆ ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า	ชนิดของรูปทรงเรขาคณิต	คำอธิบาย
0	UNKNOWN_GEOMETRY	Spatial ไม่สนใจรูปทรงเรขาคณิตนี้
1	POINT	รูปทรงเรขาคณิตที่ประกอบด้วยจุด 1 จุด
2	LINESTRING	รูปทรงเรขาคณิตที่ประกอบด้วยเส้น 1 เส้น
3	POLYGON	รูปทรงเรขาคณิตที่ประกอบด้วยพื้นที่รูปปิด 1 รูป (อาจมีรูหรือไม่มีรูก็ได้)
4	Collection	รูปทรงเรขาคณิตที่ประกอบด้วยชุดของอีเลเมนต์ที่ต่างกัน
5	MULTIPOINT	รูปทรงเรขาคณิตที่ประกอบด้วยจุดหลายจุด
6	MULTILINESTRING	รูปทรงเรขาคณิตที่ประกอบด้วยเส้นหลายๆเส้น
7	MULTIPOLYGON	รูปทรงเรขาคณิตที่ประกอบด้วยพื้นที่รูปปิดหลายๆรูป (มีขอบนอกมากกว่าหนึ่ง)

ตารางที่ 5-9 ค่าต่างๆ ที่เป็นไปได้ของ SDO_GTYPE

สำหรับค่า 8-99 ถูกสงวนไว้ใช้งานในอนาคต

- **SDO_SRID** ถูกสงวนสำหรับใช้งานในอนาคต
- **SDO_POINT** เป็นออบเจกต์ไทป์ที่ประกอบด้วยแอตทริบิวต์ X,Y,Z ที่มีชนิดเป็น NUMBER
- **SDO_ELEM_INFO** เป็นอาเรย์ของตัวเลขซึ่งเปลี่ยนขนาดได้ แอตทริบิวต์นี้จะทำให้ทราบถึงวิธีการตีความคู่ลำดับที่ถูกเก็บในแอตทริบิวต์ SDO_ORDINATES

ตัวเลข 3 ตัวแต่ละชุดจะเก็บข้อมูลเกี่ยวกับรูปทรงเรขาคณิตหนึ่งๆ รูปทรงหนึ่งอาจจะประกอบด้วยหลายอีเลเมนต์ ถ้ารูปทรงเรขาคณิตมี 1 อีเลเมนต์แล้วอาเรย์ SDO_ELEM_INFO จะมีตัวเลข 3 ตัว ถ้ามี 2 อีเลเมนต์แล้วจะมีเลข 6 ตัว เป็นต้น แต่ละชุดของตัวเลข 3 ตัวมีความหมายดังนี้

- SDO_STARTING_OFFSET บ่งชี้ถึงตำแหน่งภายในอาเรย์ SDO_ORDINATES ซึ่งเป็นที่เก็บคู่ลำดับของอีเลเมนต์ ค่าของตำแหน่งเริ่มต้นที่ 1 ไม่ใช่0 ดังนั้นคู่ลำดับแรกสำหรับอีเลเมนต์แรกจะอยู่ที่ SDO_GEOMETRY..DO_ORDINATES(1)
- SDO_ETYPE บ่งบอกถึงชนิดของอีเลเมนต์ มีค่าได้ตั้งแต่ 0-5 โดยที่ 1,2,3 เป็นอีเลเมนต์อย่างง่าย ส่วน 4 และ 5 เป็นอีเลเมนต์ประกอบ อีเลเมนต์ของอีเลเมนต์ประกอบจะต้องต่อเนื่องกัน จุดสุดท้ายของอีเลเมนต์ย่อยหนึ่งต้องเป็นจุดเริ่มต้นของอีเลเมนต์ย่อยต่อไป
- SDO_INTERPRETATION มีความหมาย 2 อย่างขึ้นอยู่กับว่า SDO_ETYPE เป็นอีเลเมนต์ประกอบหรือไม่

ถ้า SDO_ETYPE เป็นอีเลเมนต์ประกอบ (4 หรือ 5) ฟิลด์นี้จะบอกว่ามีตัวเลข 3 ตัวกี่ชุดที่เป็นส่วนประกอบของอีเลเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า SDO_ETYPE ไม่ใช่อีเลเมนต์ประกอบ (1,2 หรือ 3) ฟิลด์นี้จะบอกถึงว่าลำดับของคู่ลำดับสำหรับอีเลเมนต์นี้จะถูกตีความว่าอย่างไร เช่น เส้นหรือพื้นที่รูปปิดอาจถูกสร้างมาจากลำดับของเส้นตรงหรือเส้นโค้ง

ค่าของ SDO_ETYPE และ SDO_INTERPRETATION มีความหมายดังตาราง

SDO_ETYPE	SDO_INTERPRETATION	ความหมาย
0	0	ไม่รองรับอีเลเมนต์ชนิดนี้
1	1	จุด
1	N>1	กลุ่มของจุด N จุด
2	1	เส้น ซึ่งแต่ละจุดเชื่อมต่อด้วยเส้นตรง
2	2	เส้น ซึ่งแต่ละจุดเชื่อมต่อด้วยเส้นโค้ง
3	1	พื้นที่รูปปิด ซึ่งแต่ละจุดเชื่อมต่อด้วยเส้นตรง
3	2	พื้นที่รูปปิด ซึ่งแต่ละจุดเชื่อมต่อด้วยเส้นโค้ง เส้นโค้งแต่ละเส้นจะใช้จุด 3 จุดในการอ้างอิง ได้แก่จุดเริ่มต้นของเส้นโค้ง จุดใดๆ บนเส้นโค้งและจุดสุดท้ายของเส้นโค้ง
3	3	สี่เหลี่ยม ใช้เฉพาะจุดมุมล่างซ้าย และมุมบนขวาในการอธิบายรูป
3	4	วงกลม ใช้จุด 3 จุดในการอธิบายรูป
4	N>1	เส้น ซึ่งแต่ละจุดถูกเชื่อมต่อด้วยเส้นตรงหรือเส้นโค้ง ค่า N บอกถึงจำนวนอีเลเมนต์ย่อยที่ประกอบกันเป็นเส้น
5	N>1	พื้นที่รูปปิด ซึ่งประกอบด้วยเส้นตรงและเส้นโค้งต่อกัน ค่า N บอกถึงจำนวนของอีเลเมนต์ย่อยที่ประกอบกันเป็นพื้นที่รูปปิด

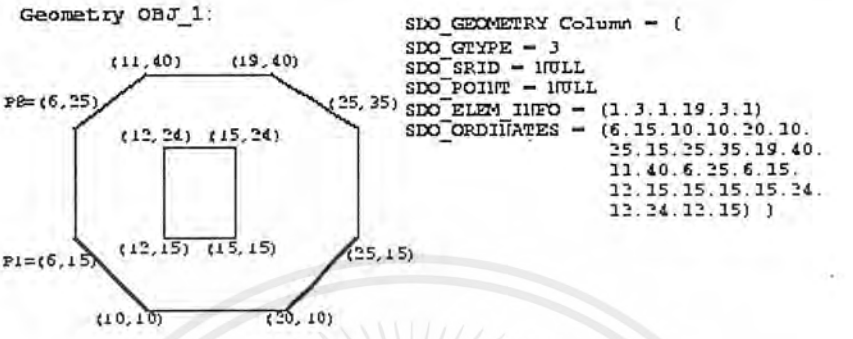
ตารางที่ 5-10 ค่าต่างๆ และความหมายของ SDO_ETYPE และ SDO_INTERPRETATION

- **SDO_ORDINATES** เป็นอาเรย์ (1048576) ของ NUMBER ที่เปลี่ยนขนาดได้ ซึ่งเก็บคู่ลำดับของรูปทรง อาเรย์นี้ต้องถูกใช้ร่วมกันกับอาเรย์ SDO_ELEM_INFO ค่าในอาเรย์จะถูกเก็บเรียงตามมิติ ตัวอย่างเช่น พื้นที่รูปปิด 2 มิติ ซึ่งประกอบไปด้วยขอบ 4 จุด จะถูกเก็บเป็น { X1,Y1,X2,Y2,X3,Y3,X4, Y4,X1,Y1 } ถ้าเป็นพื้นที่รูปปิด 3 มิติ จะเก็บเป็น { X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,X1, Y1,Z1 }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

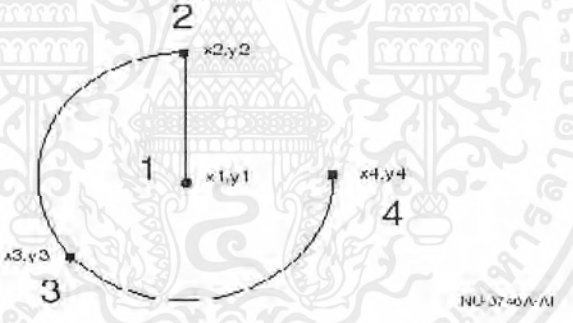
ค่าของ SDO_ORDINATES จะต้องถูกต้องและไม่ว่างเปล่า ค่า SDO_ORDINATES(1) เป็นคู่ลำดับแรกของจุดแรกในอีเลเมนต์แรก

ตัวอย่างที่ 1 รูปทรงเรขาคณิตที่ประกอบด้วย 2 อีเลเมนต์ โดยอีเลเมนต์พื้นที่รูปปิดที่อยู่ด้านในจะเป็นรู



รูปที่ 5-11 พื้นที่รูปปิดซึ่งมีรูภายใน

ตัวอย่างที่ 2 เป็นรูปทรงเรขาคณิตที่ประกอบด้วย เส้นตรง 1 เส้น และเส้นโค้งส่วนของวงกลม 1 เส้น โดยใช้จุด 4 จุด จุด 1และ 2 อธิบายถึงเส้นตรง ส่วนจุด 2,3 และ 4 อธิบายถึงเส้นโค้ง



รูปที่ 5-12 รูปทรงเรขาคณิต

อาเรย์ SDO_ELEM_INFO เก็บ {(1,4,2,),(1,2,1),(3,2,2)} และ SDO_ORDINATES เก็บ (X1,Y1, X2,Y2,X3,Y3,X4,Y4)

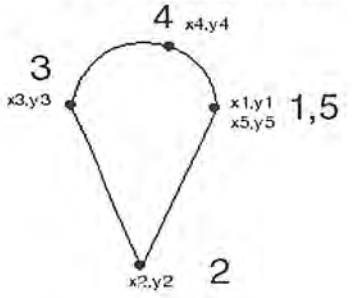
ตัวเลขชุดแรก (1,4,2) บ่งบอกว่าอีเลเมนต์นี้ประกอบไปด้วยเส้นที่เป็นเส้นตรงและเส้นโค้ง ซึ่งแต่ละเส้นถูกอธิบายโดยตัวเลข 3 ตัว อีก 2 ชุดที่เหลือ

ตัวเลขชุดที่ 2 (1,2,1) บ่งบอกว่าเส้นนั้นสร้างมาจากเส้นตรง ซึ่งคู่ลำดับเริ่มต้นที่ตำแหน่งที่ 1 ของ SDO_ORDINATES และสิ้นสุดที่ตำแหน่งเริ่มต้นของตัวแรกชุดถัดไป ซึ่งในที่นี้คือ 3 นั่นคือจุดสิ้นสุดจะอยู่ที่ตำแหน่งที่ 3 ของ SDO_ORDINATES นั่นคือคู่ลำดับ X2,Y2

ตัวเลขชุดที่ 3 (3,2,2) บ่งบอกว่าเส้นนี้เกิดจากเส้นโค้ง ซึ่งเริ่มต้นที่ตำแหน่งที่ 3 และสิ้นสุดที่ตำแหน่งเริ่มต้นของตัวเลขชุดถัดไป แต่ในกรณีนี้เป็นตัวเลขชุดสุดท้าย ดังนั้นจะไปสิ้นสุดที่คู่ลำดับสุดท้ายของ SDO_ORDINATES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3 เป็นรูปพื้นที่รูปปิด ซึ่งเกิดจากเส้นตรงและเส้นโค้ง และใช้จุด 5 จุดในการอธิบายรูปนี้ จุด 1,2 และ 3 อธิบายถึงเส้นตรง ส่วนจุด 3,4 และ 5 อธิบายถึงเส้นโค้ง โดยที่จุด 1 และ 5 เป็นจุดเดียวกัน



รูปที่ 5-13 รูปทรงเรขาคณิตพื้นที่รูปปิด

อาเรย์ SDO_ELEM_INFO จะเก็บ {(1,5,2),(1,2,1),(5,2,2)}

ตัวเลขชุดแรก (1,5,2) บ่งบอกว่ารูปทรงนี้เป็นพื้นที่รูปปิด ซึ่งเกิดจากเส้นตรงและเส้นโค้ง

ตัวเลขชุดที่ 2 (1,2,1) บ่งบอกว่าเส้นนี้ประกอบด้วยส่วนของเส้นตรง ซึ่งเริ่มต้นที่ตำแหน่งที่ 1 และสิ้นสุดที่ตำแหน่งที่ 5 (จากตัวเลขชุดที่3) คือที่จุด X3,Y3

ตัวเลขชุดที่ 3 (5,2,2) อธิบายถึงเส้นโค้งซึ่งมีจุดเริ่มต้นอยู่ที่ตำแหน่งที่ 5 และสิ้นสุดที่จุด X5,Y5

โครงสร้างของ Geometry Metadata

Geometry Metadata อธิบายถึงมิติ ขอบเขตบน ขอบเขตล่าง และค่าTolerance ในแต่ละมิติ โดยจะเก็บในตารางที่ชื่อ SDO_GEOM_METADATA ซึ่งถูกสร้างดังนี้

```
Create Table SDO_GEOM_METADATA (
 TABLE_NAME   VARCHAR2(30),
 COLUMN_NAME  VARCHAR2(30),
 DIMINFO     MDSYS.SDO_DIM_ARRAY);
```

ตารางนี้ควรจะถูกสร้างขึ้นมาสำหรับผู้ใช้ทุกคนที่มีตารางซึ่งมีคอลัมน์ที่มีชนิดเป็น SDO_GEOMETRY ตัวอย่างเช่น สุรศักดิ์ (Surasak) มีตาราง Roads,Parks,Rivers ที่มีคอลัมน์ theGeometry เป็นชนิดSDO_GEOMETRY แล้วจะต้องมีข้อมูล 3 แถวอยู่ในตาราง SURASAK.SDO_GEOM_METADATA ผู้ใช้หรือแอปพลิเคชั่นชันมีหน้าที่รับผิดชอบดูแลข้อมูลในตารางนี้

คอลัมน์ SDO_GEOM_METADATA.TABLE_NAME จะเก็บชื่อของตารางเช่น Roads,Paarks ซึ่งมีคอลัมน์ที่มีชนิดเป็น SDO_GEOMETRY ซึ่งชื่อคอลัมน์นี้จะถูกเก็บใน SDO_GEOM_METADATA. COLUMN_NAME สำหรับคอลัมน์ SDO_GEOM_METADATA.DIMINFO เป็นอาเรย์ของออบเจกต์ไทป์ เรียงตามมิติ

```
Create Type SDO_DIM_ARRAY as VARRAY(4) of SDO_DIM_ELEMENT;
```

และ SDO_DIM_ELEMENT ถูกกำหนดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Create Type SDO_DIM_ELEMENT as OBJECT (
 SDO_DIMNAME VARCHAR2(64),
 SDO_LB NUMBER NOT NULL,
 SDO_UB NUMBER NOT NULL,
 SDO_TOLERANCE NUMBER NOT NULL);
```

SDO_DIM_ARRAY มีขนาดเป็น N ถ้ามี N มิติ ดังนั้น SDO_GEOM_METADATA.DIMINFO จะเก็บ 2 อินสแตนซ์ของ SDO_DIM_ELEMENT สำหรับรูปทรง 2 มิติ และแต่ละอินสแตนซ์ของ SDO_DIM_ELEMENT จะต้องไม่ว่างเปล่า

สำหรับตารางอินเด็กซ์เชิงพื้นที่จะมีชื่อว่า SDO_INDEX_METADATA 1 ตารางสำหรับผู้ใช้แต่ละคน

ชื่อคอลัมน์	ชนิดข้อมูล	จุดประสงค์
SDO_LEVEL	NUMBER	ระดับของการแบ่งออบเจกต์ออกเป็นช่องสี่เหลี่ยมย่อยๆ
SDO_NUMTILES	NUMBER	จำนวนช่องสี่เหลี่ยมต่อออบเจกต์ที่ถูกใช้ในการประมาณรูปร่าง
SDO_MAXLEVEL	NUMBER	ค่าสูงสุดสำหรับการแบ่งระดับช่องสี่เหลี่ยม มักสูงกว่า SDO_LEVEL
SDO_COMMIT_INTERVAL	NUMBER	จำนวนของรูปทรงเรขาคณิต (แถว) ที่ถูกประมวลผลระหว่างการสร้างอินเด็กซ์ ก่อนการยืนยันการเพิ่มข้อมูลลงในตาราง SDOINDEX
SDO_INDEX_TABLE	VARCHAR2	ชื่อของตาราง SDOINDEX
SDO_TABLESPACE	VARCHAR2	เหมือนกันกับคำสั่งพื้นฐาน CREATE TABLE Tablespace ในการสร้างตาราง SDOINDEX
SDO_INITIAL_EXTENT	NUMBER	เหมือนกับคำสั่ง CREATE TABLE
SDO_NEXT_EXTENT	NUMBER	เหมือนกับคำสั่ง CREATE TABLE
SDO_PCTINCREASE	NUMBER	เหมือนกับคำสั่ง CREATE TABLE
SDO_MIN_EXTENTS	NUMBER	เหมือนกับคำสั่ง CREATE TABLE
SDO_MAX_EXTENTS	NUMBER	เหมือนกับคำสั่ง CREATE TABLE
SDO_FIXED_METADATA	RAW	ถ้าสามารถประยุกต์ใช้ได้ คอลัมน์นี้จะเก็บส่วนของ Metadata ของ SDO_GROUPCODE หรือ SDO_CODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		สำหรับอินเด็กซ์ที่ถูกฟิกซ์ระดับ
SDO_INDEX_COORDSYS	VARCHAR2	จองไว้สำหรับการใช้งานในอนาคต
SDO_INDEX_PRIMARY	NUMBER	บ่งบอกว่าเป็น Primary หรือ Secondary Index , 1 = primary, 2 = secondary.
SDO_INDEX_OWNER	VARCHAR2	เจ้าของอินเด็กซ์
SDO_INDEX_NAME	VARCHAR2	ชื่อของอินเด็กซ์
SDO_TSNAME	VARCHAR2	ชื่อโครงสร้างของ SDO_INDEX_TABLE
SDO_COLUMN_NAME	VARCHAR2	ชื่อคอลัมน์ที่อินเด็กซ์ถูกสร้าง

ตารางที่ 5-11 คอลัมน์ต่าง ๆ ในตาราง SDO_INDEX_METADATA

สำหรับตารางข้อมูลอินเด็กซ์ประกอบไปด้วยคอลัมน์ต่อไปนี้บางคอลัมน์หรือทั้งหมด

ชื่อคอลัมน์	ชนิดข้อมูล	จุดประสงค์
SDO_ROWID	RAW	หมายเลขของแถวในตารางที่เก็บออบเจกต์
SDO_CODE	RAW	อินเด็กซ์สำหรับออบเจกต์ในแถวซึ่งถูกระบุดโดย SDO_ROWID
SDO_MAXCODE	RAW	ค่าสูงสุดของ SDO_CODE
SDO_GROUPCODE	RAW	การป้อนอินเด็กซ์ที่ระดับ SDO_LEVEL.
SDO_META	RAW	ส่วนของMetadata ของ SDO_CODE สำหรับไฮบริดอินเด็กซ์

ตารางที่ 5-12 คอลัมน์ในตารางข้อมูลอินเด็กซ์

5.2.4 การป้อนข้อมูลเชิงพื้นที่ลงในฐานข้อมูล

มีขั้นตอนดังนี้

1. นำข้อมูลเชิงพื้นที่ไปเก็บในคอลัมน์ที่มีชนิดเป็น SDO_GEOMETRY
2. สร้างอินเด็กซ์เชิงพื้นที่บนคอลัมน์ที่มีชนิดเป็น SDO_GEOMETRY

กระบวนการป้อนข้อมูล แบ่งได้เป็น 2 วิธี คือ การป้อนข้อมูลจำนวนมาก และ การป้อนโดยวิธี Transaction Inserts เช่นเดียวกับกระบวนการป้อนข้อมูลของโมเดลเชิงสัมพันธ์

5.2.4.1 การป้อนข้อมูลปริมาณมาก ตัวอย่างการป้อนข้อมูลออบเจกต์ SDO_GEOMETRY ปริมาณมากๆ โดยสมมติให้ตาราง POLY_4PT ถูกสร้างดังนี้

```
CREATE TABLE POLY_4PT (GID     VARCHAR2(32),
                       GEOMETRY  MDSYS.SDO_GEOMETRY);
```

สมมติว่าข้อมูลแอสกี ประกอบด้วยไฟล์ซึ่งมีการแบ่งข้อมูลเป็นคอลัมน์ และแบ่งแยกข้อมูลออกเป็นแถวๆ ซึ่งถูกกำหนดโดยรูปแบบของตารางดังต่อไปนี้

geometry rows : GID , GEOMETRY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง ไฟล์ควบคุมสำหรับการป้อนข้อมูลจำนวนมาก

```
LOAD DATA INFILE *
INTO TABLE POLY_4PT
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
TRAILING NULLCOLS
(
gid    char(6),
geometry COLUMN OBJECT
begindata
1,3,,pt,,,,1,3,1; -122.4215,37.7862, -122.422,37.7869, -122.421,37.789,
-122.42,37.7866, -122.4215,37.7862:
2,3,,pt,,,,1,3,1; -122.4019,37.8052, -122.4027,37.8055, -122.4031,37.806,
-122.4012,37.8052, -122.4019,37.8052:
( sdo_gtype INTEGER EXTERNAL,
sdo_srid INTEGER EXTERNAL,
  isnull FILLER CHAR,
  SDO_POINT COLUMN OBJECT  NULLIF geometry.isnull="pt"
  ( X INTEGER EXTERNAL,
Y INTEGER EXTERNAL,
Z INTEGER EXTERNAL),
SDO_ELEM_INFO VARRAY terminated by ';'
(SDO_ORDINATES char(38)),
SDO_ORDINATES VARRAY terminated by ':'
   (SDO_ORDINATES char(38))))
3,3,,pt,,,,1,3,1; -122.426,37.803, -122.4242,37.8053, -122.42355,37.8044,
-122.4235,37.8025, -122.426,37.803:
```

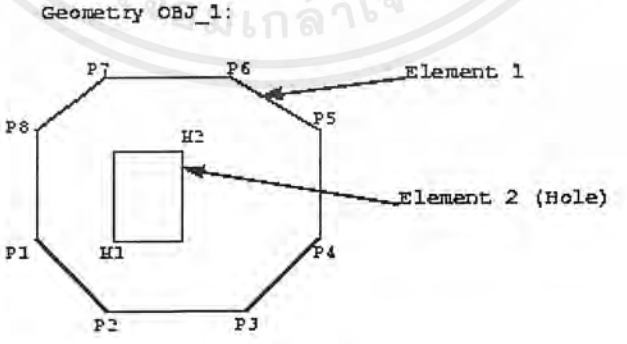
ตัวอย่างไฟล์ควบคุมสำหรับการป้อนข้อมูลจุดจำนวนมาก

```
LOAD DATA INFILE *
INTO TABLE POINT
REPLACE
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
TRAILING NULLCOLS
(
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gid    char(6),
geometry COLUMN OBJEC
( sdo_gtype INTEGER EXTERNAL,
 sdo_srid INTEGER EXTERNAL,
 SDO_POINT COLUMN OBJECT
 ( X INTEGER EXTERNAL,
   Y INTEGER EXTERNAL,
   Z INTEGER EXTERNAL),
   is_null1 FILLER CHAR,
 SDO_ELEM_INFO VARRAY terminated by ';' NULLIF geometry.is_null1="v1"
      (SDO_ORDINATES char(38)),
   is_null2 FILLER CHAR,
 SDO_ORDINATES VARRAY terminated by ':'
 NULLIF geometry.is_null2="v2"
   (SDO_ORDINATES char(38))))
begindata
1,1,,-122.4215,37.7862,,v1,;v2,:
2,1,,-122.4019,37.8052,,v1,;v2,:
3,1,,-122.426,37.803,,v1,;v2,:
4,1,,-122.4171,37.8034,,v1,;v2,:
5,1,,-122.416151,37.8027228,,v1,;v2,:
```

5.2.4.2 การป้อนข้อมูลโดยวิธี Transactional Inserts โดยใช้ SQL



รูปที่ 5-14 พื้นที่รูปที่มีรูภายใน

รูปนี้มี 2 อีเลเมนต์ โดยที่

Element 1= [P1(6,15), P2(10,10), P3(20,10), P4(25,15), P5(25,35), P6(19,40), P7 (11,40), P8(6,25), P1(6,15)]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Element 2= [H1(12,15), H2(15,24)]

ตัวอย่างนี้สมมติให้ตาราง PARKS ถูกสร้างดังนี้

```
CREATE TABLE PARKS (NAME VARCHAR2(32),
                    SHAPE MDSYS.SDO_GEOMETRY);
```

คำสั่งภาษา SQL ที่ใช้ในการป้อนข้อมูลคือ

```
INSERT INTO PARKS
 VALUES ('OBJ_1', MDSYS.SDO_GEOMETRY(3, NULL,NULL,
          MDSYS.SDO_ELEM_INFO_ARRAY(1,3,1, 19,3,3),
          MDSYS.SDO_ORDINATE_ARRAY(6,15, 10,10, 20,10, 25,15, 25,35,
     19,40, 11,40, 6,25, 6,15, 12,15, 15,24)));
```

สมมติให้มี 2 มิติ คือ X และ Y ขอบเขต 0-100 และค่า tolerance ของทั้ง 2 มิติเป็น 0.005 มีคำสั่งในการป้อนข้อมูลลงในตาราง SDO_GEOM_METADATA ดังนี้

```
INSERT INTO SDO_GEOM_METADATA
 VALUES ('PARKS', 'SHAPE',
    MDSYS.SDO_DIM_ARRAY(MDSYS.SDO_DIM_ELEMENT('X', 0, 100, 0.005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0, 100, 0.005)));
```

5.2.5 วิธีการทำอินเด็กซ์

หลังจากป้อนข้อมูลเชิงพื้นที่เข้าสู่ตารางแล้ว จะต้องทำการสร้างอินเด็กซ์เชิงพื้นที่ เพื่อให้การเข้าถึงข้อมูลมีประสิทธิภาพมากขึ้น มี 2 วิธีในการทำอินเด็กซ์เชิงพื้นที่ คือ

1. การแบ่งช่องสี่เหลี่ยมขนาดคงที่ (Fixed-size tile)
2. การแบ่งช่องสี่เหลี่ยมแบบไฮบริด (Hybrid)

SDO_LEVEL	SDO_NUMTILES	การกระทำ
ไม่ระบุ	ไม่ระบุ	เกิดข้อผิดพลาด
>=1	ไม่ระบุ	แบ่งช่องสี่เหลี่ยมขนาดคงที่
>=1	>=1	แบ่งช่องสี่เหลี่ยมขนาดคงที่และขนาดเปลี่ยนแปลงได้ โดยที่ SDO_LEVEL บอกถึงขนาดของช่องสี่เหลี่ยมขนาดคงที่ ส่วน SDO_NUMTILES บอกถึงจำนวนของช่องสี่เหลี่ยมที่มีขนาดเปลี่ยนแปลงได้
ไม่ระบุ	>=1	ไม่รองรับ

ตารางที่ 5-13 ความหมายของค่า SDO_LEVEL และ SDO_NUMTILES

ใช้คำสั่ง CREATE INDEX ในการสร้างอินเด็กซ์ และเมื่อสร้างเสร็จแล้วก็ต้องทำการ Commit ถ้าจะทำการดรอปอินเด็กซ์ทำได้โดยใช้ DROP INDEX <index_name> [FORCE]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5.1 การทำอินเด็กซ์เชิงพื้นที่โดยการแบ่งช่องสี่เหลี่ยมขนาดคงที่

สำหรับโมเดลเชิงวัตถุสัมพันธ์ โอราเคิล8i จะแนะนำให้ใช้การทำอินเด็กซ์เชิงพื้นที่โดยใช้วิธีการแบ่งช่องสี่เหลี่ยมแบบไฮบริด เพราะการแบ่งช่องสี่เหลี่ยมแบบคงที่เป็นส่วนหนึ่งของแบบไฮบริด

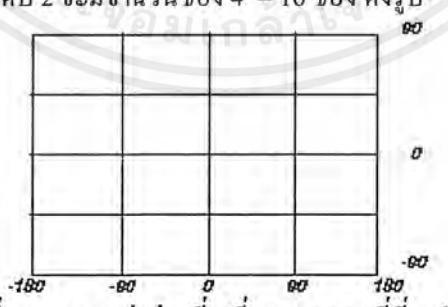
แต่ถึงแม้วิธีนี้จะไม่ใช่วิธีที่ดีกว่าแบบไฮบริด ก็สามารถนำมาใช้ได้เช่นกัน ในการทำอินเด็กซ์แบบนี้จะไม่สนใจ SDO_NUMTILES และตั้งค่า SDO_LEVEL เป็นระดับของการแบ่งช่องสี่เหลี่ยมตามต้องการ ความสัมพันธ์ระหว่างระดับการแบ่งช่องสี่เหลี่ยมกับขนาดของช่องสี่เหลี่ยมจะขึ้นอยู่กับขอบเขตบนและขอบเขตล่างของแต่ละมิติที่ถูกเก็บไว้ในคอลัมน์ DIMINFO ของตาราง SDO_GEOM_METADATA ตัวอย่างเช่นลองจิจูดควรเป็น –180 ถึง 180 และ ละติจูด –90 ถึง 90 ดังรูป

รูปที่ 5-15 ขอบเขตอย่างง่ายๆ

ถ้าค่าในคอลัมน์ SDO_LEVEL เป็น 1 ช่องสี่เหลี่ยมจะถูกแบ่งครึ่งทั้งแนวตั้งและแนวนอนดังรูป

รูปที่ 5-16 การแบ่งช่องสี่เหลี่ยมขนาดคงที่ที่ระดับ 1

จำนวนช่องสี่เหลี่ยมจะเป็น 4^n ช่อง เมื่อ n แทนระดับของการแบ่งช่องสี่เหลี่ยม ซึ่งถูกเก็บในคอลัมน์ SDO_LEVEL เช่นที่ระดับ 2 จะมีจำนวนช่อง $4^2 = 16$ ช่อง ดังรูป



รูปที่ 5-17 การแบ่งช่องสี่เหลี่ยมขนาดคงที่ที่ระดับ 2

ขนาดของช่องสี่เหลี่ยมสามารถหาได้โดยใช้สูตร

ความยาว = (ขอบเขตบน – ขอบเขตล่าง)/ 2^{SDO_LEVEL}

ความยาวทางแกน X = (180 - (-180)) / 2^2

= (360) / 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

= 90

ความยาวทางแกน Y = (90 - (-90)) / 2^2

= (180) / 4

= 45

ที่ระดับ 2 ขนาดช่องสี่เหลี่ยมเป็น 90*45 ยิ่งระดับการแบ่งเพิ่มมากขึ้นเท่าไร ขนาดช่องสี่เหลี่ยมก็จะยิ่งเล็กลง ฟังก์ชัน SDO_TUNE.ESTIMATE_TILEING_LEVEL() จะใช้ช่วยในการหาระดับของการทำการแบ่งที่เหมาะสมซึ่งจะได้จำนวนช่องสี่เหลี่ยมที่พอเหมาะสำหรับชุดข้อมูล

ตัวอย่าง การสร้างอินเด็กซ์บนตาราง ROADS ซึ่งมี SDO_GEOM_METADATA สำหรับคอลัมน์ ROADS.GEOMETRY ใช้คำสั่งดังต่อไปนี้

```
CREATE INDEX ROADS_FIXED ON ROADS(GEOMETRY)
INDEXTYPE IS MDSYS.SPATIAL_INDEX  PARAMETERS('SDO_LEVEL = 8');
```

5.2.5.1 การทำอินเด็กซ์เชิงพื้นที่โดยการแบ่งช่องสี่เหลี่ยมแบบไฮบริด

ใช้ทั้งการแบ่งช่องสี่เหลี่ยมขนาดคงที่และการแบ่งช่องสี่เหลี่ยมที่มีขนาดเปลี่ยนแปลงได้ สำหรับแต่ละรูปทรงเรขาคณิตจะมีกลุ่มของช่องสี่เหลี่ยมขนาดคงที่ และกลุ่มของช่องสี่เหลี่ยมที่มีขนาดเปลี่ยนแปลงได้ ปกคลุมรูปทรงเรขาคณิตทั้งหมด การแบ่งช่องสี่เหลี่ยมแบบไฮบริดนี้ ค่า SDO_LEVEL และ SDO_NUMTILES จะต้องมีค่ามากกว่าหรือเท่ากับ 1

SDO_NUMTILES เป็นค่าที่บอกถึงจำนวนของช่องสี่เหลี่ยมที่มีขนาดเปลี่ยนแปลงได้ ซึ่งถูกใช้ในการครอบคลุมรูปทรงเรขาคณิตที่กำลังถูกทำอินเด็กซ์ ค่านี้มักจะมีค่าน้อย สำหรับจุด ค่า SDO_NUMTILES มักจะเป็น 1 แต่ถ้าเป็นรูปทรงอื่นๆ ค่าอาจจะอยู่ประมาณ 8 ยิ่งมีค่ามากเท่าไรก็ยิ่งละเอียดมากขึ้น ช่วยในการปรับปรุงการทำ Primary Filter แต่เป็นการเพิ่มจำนวนข้อมูลอินเด็กซ์ต่อรูปทรงเรขาคณิต

SDO_LEVEL บ่งบอกถึงขนาดของช่องสี่เหลี่ยมขนาดคงที่ การตั้งค่า SDO_LEVEL ให้เหมาะสมนั้นเป็นเรื่องของศิลปะมากกว่าวิทยาศาสตร์ วิธีการหนึ่งคือการใช้ฟังก์ชัน SDO_TUNE.ESTIMATE_TILINGLEVEL() เพื่อหาค่า SDO_LEVEL ที่เหมาะสม

ตัวอย่าง การสร้างอินเด็กซ์แบบไฮบริด

```
CREATE INDEX ROADS_FIXED ON ROADS(GEOMETRY)
INDEXTYPE IS MDSYS.SPATIAL_INDEX PARAMETERS('SDO_LEVEL = 6,
SDO_NUMTILES=12');
```

5.2.6 การคิวรีข้อมูลเชิงพื้นที่

จากตัวอย่างเดียวกันกับโมเดลเชิงสัมพันธ์ในหัวข้อ 5.1.6 ในโอราเคิล8i ได้จัดเตรียมโอเปอเรเตอร์ใหม่ SDO_FILTER() เพื่อช่วยในการทำ primary filter โดยมีรูปแบบการใช้งานดังนี้

```
SDO_FILTER(geometry1 MDSYS.SDO_GEOMETRY, geometry2 MDSYS.SDO_GEOMETRY,
params VARCHAR2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

- geometry1 เป็นคอลัมน์ของไทป์ MDSYS.SDO_GEOMETRY และต้องถูกทำอินเด็กซ์
- geometry2 เป็นออบเจกต์ไทป์ MDSYS.SDO_GEOMETRY geometry2 อาจจะมาจากตารางหรือไม่ก็ได้ ถ้ามาจากตารางก็อาจถูกทำอินเด็กซ์หรือไม่ก็ได้
- params เป็นสตริงที่บ่งบอกถึงพฤติกรรมของโอเปอเรเตอร์

ตัวอย่างต่อไปนี้เป็นการทำ primary filter ซึ่งได้ผลลัพธ์เป็นรูปทรงเรขาคณิตที่อยู่ช่องสี่เหลี่ยมเดียวกันกับหน้าต่างคิวรี่ ได้แก่ {1013,501,1243,12}

ตัวอย่าง การทำ primary filter โดยใช้หน้าต่างคิวรี่ชั่วคราว โดยไม่ต้อง INSERT หน้าต่างคิวรี่ลงในตาราง หน้าต่างคิวรี่จะถูกทำอินเด็กซ์ในหน่วยความจำ และช่วยให้ประสิทธิภาพดีขึ้น

```
SELECT A.Feature_ID FROM TARGET A
WHERE mdsys.sdo_filter(A.shape, mdsys.sdo_geometry(3,NULL,NULL,
                        mdsys.sdo_elem_info(1,3,3),
                        mdsys.sdo_ordinates(x1,y1, x2,y2)),
                        'querytype=window') = 'TRUE';
```

(x1,y1) และ (x2,y2) เป็นจุดมุมล่างซ้ายและมุมบนขวาของหน้าต่างคิวรี่

ตัวอย่าง การทำ primary filter โดยใช้หน้าต่างคิวรี่ที่ถูกเก็บไว้ในฐานข้อมูล โดยสมมติว่าได้ INSERT หน้าต่างคิวรี่ลงในตารางที่ชื่อว่า WINDOWS ด้วย ID 'WIN_1'

```
SELECT A.Feature_ID FROM TARGET A, WINDOWS B
WHERE B.ID= 'WIN_1' AND
 mdsys.sdo_filter(A.shape, B.shape,'querytype=window') = `TRUE';
```

โอเปอเรเตอร์ SDO_RELATE() จะทำทั้ง primary filter และ secondary filter มีรูปแบบของคำสั่งดังนี้

```
SDO_RELATE(geometry1 MDSYS.SDO_GEOMETRY, geometry2 MDSYS.SDO_GEOMETRY,
params VARCHAR2)
```

เมื่อ

- geometry1 เป็นคอลัมน์ของไทป์ MDSYS.SDO_GEOMETRY ในตาราง T1 และถูกทำอินเด็กซ์
- geometry2 เป็นคอลัมน์ของไทป์ MDSYS.SDO_GEOMETRY ในตาราง T2 อาจจะถูกทำอินเด็กซ์หรือไม่ก็ได้ T2 อาจเป็นตารางเดียวกันกับ T1 ก็ได้
- params เป็นสตริงซึ่งบ่งบอกถึงพฤติกรรมของโอเปอเรเตอร์

ตัวอย่างต่อไปนี้เป็นการทำทั้ง primary filter และ secondary filter ซึ่งจะทำให้ได้รูปทรงเรขาคณิตที่ซ้อนทับกับหน้าต่างคิวรี่ ได้แก่ {1243,1013}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง การทำ secondary filter โดยใช้หน้าต่างคิวรี่ชั่วคราว ไม่ต้อง INSERT หน้าต่างคิวรี่ลงในตาราง หน้าต่างคิวรี่จะถูกทำอินเด็กซ์ และช่วยให้ประสิทธิภาพดีขึ้น

```
SELECT A.Feature_ID FROM TARGET A
  WHERE mdsys.sdo_relate(A.shape, mdsys.sdo_geometry(3,NULL,NULL,
                         mdsys.sdo_elem_info(1,3,3),
                         mdsys.sdo_ordinates(x1,y1, x2,y2)),
                         'mask=anyinteract querytype=window') = 'TRUE';
```

ตัวอย่าง การทำ secondary filter โดยใช้หน้าต่างคิวรี่ที่ถูกเก็บไว้ในฐานข้อมูล โดยสมมติว่าได้ INSERT หน้าต่างคิวรี่ลงในตารางที่ชื่อว่า WINDOWS ด้วย ID 'WIN_1'

```
SELECT A.Feature_ID FROM TARGET A, WINDOWS B
 WHERE B.ID= 'WIN_1' AND
       mdsys.sdo_relate(A.shape, B.shape,
       'mask=anyinteract querytype=window') = 'TRUE';
```

5.3 ข้อเปรียบเทียบการดูแลข้อมูลเชิงพื้นที่ระหว่างโอราเคิล 8.0.5 และ โอราเคิล 8i

โอราเคิล 8.0.5	โอราเคิล 8I
รองรับรูปทรงเรขาคณิตพื้นฐาน 3 ชนิด ได้แก่ จุด เส้น และ พื้นที่รูปปิด	รองรับรูปทรงเรขาคณิตพื้นฐาน 3 ชนิด และรองรับรูปทรงเรขาคณิตที่ประกอบขึ้นมาจากเส้นโค้ง หรือเส้นตรงรวมกัน
จัดเก็บข้อมูลรูปทรงเรขาคณิตในตารางที่ Spatial Cartridge กำหนด	จัดเก็บข้อมูลรูปทรงเรขาคณิตในออบเจกต์ไทป์ SDO_GEOMETRY
บางคอลัมน์ในตาราง SDO_GEOM ซึ่งเก็บค่าจุดพิกัดของรูปทรงเรขาคณิตจะมีค่าเป็น Null	จัดเก็บค่าจุดพิกัดของรูปทรงเรขาคณิตในออบเจกต์ไทป์ SDO_GEOMETRY ค่าจุดพิกัดจะไม่มีค่า Null
สำหรับรูปทรงเรขาคณิตที่เป็นเส้นโค้ง จะมีการจัดเก็บจุดพิกัดต่างๆ มากมายหลายจุด เปลืองพื้นที่ในการจัดเก็บ	สำหรับรูปทรงเรขาคณิตที่เป็นเส้นโค้งจะมีการเก็บจุดพิกัดเท่าที่จำเป็น โดยจะเก็บน้อยกว่าโมเดลเชิงสัมพันธ์ ช่วยประหยัดพื้นที่การจัดเก็บ
การคิวรี่ข้อมูลเชิงพื้นที่มี 2 ขั้นตอนคือทำ Primary filter และ Secondary filter	การคิวรี่ข้อมูลเชิงพื้นที่มี 2 ขั้นตอนเช่นเดียวกันกับโมเดลเชิงสัมพันธ์ แต่รายละเอียดการเรียกใช้ฟังก์ชันของ Spatial Cartridge ต่างกัน

ตารางที่ 5-14 เปรียบเทียบการดูแลข้อมูลเชิงพื้นที่ระหว่างโอราเคิล 8.0.5 และ โอราเคิล 8i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การเชื่อมต่อฐานข้อมูลด้วยเจดีบีซี
(Java Database Connectivity)

ภาษาจาวาเป็นภาษาคอมพิวเตอร์ที่ได้รับการพัฒนาอย่างรวดเร็วและต่อเนื่อง นอกเหนือจากการสร้างกราฟิกหรือองค์ประกอบที่ใช้ในการประดับโฮมเพจ (Homepage) ยังมีการพัฒนาองค์ประกอบใหม่ ๆ ในการปฏิบัติงานด้านต่าง ๆ ได้ถูกเพิ่มเติมประสิทธิภาพให้แก่จาวา คือ เจดีบีซี (JDBC : Jaba Database Connectivity) ซึ่งทำหน้าที่อำนวยความสะดวกในการเชื่อมต่อและสื่อสารกับระบบฐานข้อมูล โดยพื้นฐานของภาษาจาวาที่ได้รับการออกแบบมาเพื่อปฏิบัติงานกับระบบเครือข่ายคอมพิวเตอร์โดยตรง องค์ประกอบนี้จึงทำให้จาวาเหมาะสมเป็นอย่างยิ่งในการเป็นเครื่องมือสำหรับสร้างโปรแกรมประยุกต์เพื่อเข้าถึงข้อมูลแบบไคล์เอนท์เซิร์ฟเวอร์ (Client-Server) โดยมีเจดีบีซีเป็นองค์ประกอบในการเชื่อมต่อ ในการถ่ายทอดคำสั่งการเรียกค้นข้อมูลจากไคล์เอนท์ไปสู่เซิร์ฟเวอร์ และรับข้อมูลผลลัพธ์จากเซิร์ฟเวอร์กลับคืนสู่ไคล์เอนท์

ระบบไคล์เอนท์เซิร์ฟเวอร์ในปัจจุบันมีการเพิ่มขึ้นของค่าใช้จ่ายของบุคลากรที่มาทำหน้าที่ดูแลบำรุงรักษาเดต้าเบสเซิร์ฟเวอร์ (Database Server) ซึ่งได้แก่บุคลากรด้านการจัดการและด้านบริการสนับสนุนสำหรับเครือข่ายขนาดเล็ก (โดยทั่วไปมีผู้ใช้น้อยกว่า 20 คน) โดยปกติผู้จัดการเครือข่าย (Network Adminnistrator) เพียงคนเดียวสามารถรับภาระหน้าที่ดูแลบำรุงรักษาเดต้าเบสเซิร์ฟเวอร์ ควบคุมการเข้าใช้เดต้าเบสเซิร์ฟเวอร์จากบรรดาผู้ใช้ทั้งหลาย และให้บริการสนับสนุนฟรอนต์เอนด์ (Front End) เมื่อจำนวนผู้ใช้ฐานข้อมูลเพิ่มจำนวนมากขึ้น หรือตัวฐานข้อมูลเองมีขนาดใหญ่ขึ้น โดยปกติแล้วก็เริ่มจำเป็นที่ต้องจ้างผู้จัดการฐานข้อมูลเพิ่มขึ้นอีกคนเพื่อมาทำหน้าที่รันระบบจัดการฐานข้อมูลและให้บริการสนับสนุนในส่วนฟรอนต์เอนด์ด้วย นอกจากนี้แล้วในกรณีที่ตะบบจัดการฐานข้อมูลรันอยู่บนระบบปฏิบัติการที่บุคลากรด้านบริการสนับสนุนไม่คุ้นเคยนั้นก็ต้องมีการอบรม ซึ่งเป็นค่าใช้จ่ายที่ต้องเพิ่มเข้าไปในค่าใช้จ่ายเริ่มแรกด้วย

นอกจากนี้แล้ว ค่าใช้จ่ายด้านฮาร์ดแวร์ก็เพิ่มขึ้นด้วย ในขณะที่มีฐานข้อมูลไคล์เอนท์เซิร์ฟเวอร์เป็นจำนวนมากที่รันอยู่ภายใต้ระบบปฏิบัติการที่นิยมใช้กัน เช่น เน็ตแวร์ (Netware), โอเอสทู (OS2) และยูนิกซ์ (Unix) เป็นต้น และบริษัทผู้ขายส่วนใหญ่ก็ประกาศว่าระบบจัดการฐานข้อมูลของตนเองนั้นสามารถรันอยู่บนฮาร์ดแวร์เดียวกับที่ซอฟต์แวร์ไฟล์เซิร์ฟเวอร์ (File Server) รันอยู่ได้ก็ตาม แต่ว่าโดยปกติแล้วถ้าต้องการสมรรถนะที่สูงและความถูกต้องของข้อมูลก็สูงก็สมควรให้เดต้าเบสเซิร์ฟเวอร์แยกไปรันบนคอมพิวเตอร์เฉพาะกิจของฐานข้อมูลเอง สิ่งนี้หมายความว่าต้องทำการซื้อคอมพิวเตอร์เพิ่มอีกเครื่องเพื่อมาทำเป็นเดต้าเบสเซิร์ฟเวอร์โดยเฉพาะ คอมพิวเตอร์นี้ต้องมีกำลังและประสิทธิภาพสูง มีหน่วยความจำเป็นจำนวนมากและมีฮาร์ดดิสก์ (Harddisk) ขนาดใหญ่ พร้อมทั้งอุปกรณ์สนับสนุน เช่น เครื่องป้องกันไฟฟ้า (UPS) เพื่อป้องกันเซิร์ฟเวอร์เสียหายจากเหตุการณ์ไฟตก หรือไฟดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าใช้จ่ายโดยรวมของซอฟต์แวร์ระบบไคล์เอนท์เซิร์ฟเวอร์ ปกติแล้วสูงกว่าระบบจัดการฐานข้อมูลแบบผู้ใช้หลายคนบนพีซี ค่าใช้จ่ายต่อหนึ่งเซิร์ฟเวอร์สำหรับฐานข้อมูลไคล์เอนท์เซิร์ฟเวอร์ที่สามารถบริการผู้ใช้ได้แบบไม่จำกัดจำนวนมีราคาสูง บวกกับค่าใช้จ่ายแยกออกไปของฟรอนต์เอนด์ หรือเครื่องมือพัฒนาซอฟต์แวร์ พร้อมทั้งค่าใช้จ่ายบุคลากรสำหรับการฝึกอบรมโปรแกรมเมอร์ในระบบใหม่ กล่าวได้ว่าค่าใช้จ่ายของระบบไคล์เอนท์เซิร์ฟเวอร์สูงกว่าของระบบจัดการฐานข้อมูลบนพีซีมาก

ยังมีเรื่องที่น่ายุ่งยากอีกเรื่องหนึ่งคือ ระบบไคล์เอนท์เซิร์ฟเวอร์นั้นมีส่วนประกอบต่าง ๆ เป็นจำนวนมาก ตามหลักที่กล่าวว่า ยิ่งระบบใดมีส่วนประกอบเป็นจำนวนมาก โอกาสที่ส่วนประกอบต่าง ๆ ในระบบนั้นจะเสียหายก็มีมากตามไปด้วย เมื่อเกิดเหตุการณ์ร้ายแรงและระบบเกิดพังเสียหายก็เป็นการยากที่จะชี้ชัดถึงปัญหาที่แท้จริงได้ ระยะเวลาที่ใช้ในการทำให้ทุกสิ่งทุกอย่างกลับสู่สภาพปกติและเริ่มทำงานได้อีกก็เป็นเวลาที่ยาวนาน สิ่งต่าง ๆ ที่เกิดขึ้นเช่นนี้มาจากการขาดประสบการณ์และความชำนาญของบุคลากรด้านบริการสนับสนุนและโปรแกรมเมอร์ ทั้งนี้เนื่องจากไคล์เอนท์เซิร์ฟเวอร์เป็นเทคโนโลยีใหม่ เมื่อเวลาผ่านไปและระบบไคล์เอนท์เซิร์ฟเวอร์เริ่มเป็นที่นิยมใช้มากขึ้น ปัญหาที่ได้กล่าวมาแล้วก็จะลดน้อยลง

องค์ประกอบด้านฮาร์ดแวร์และซอฟต์แวร์รวมกัน ซึ่งระบบจัดการฐานข้อมูลไคล์เอนท์เซิร์ฟเวอร์รันอยู่ในแพล็ตฟอร์มต่าง ๆ เช่น พีซี, ยูนิกซ์, ซุปเปอร์เซิร์ฟเวอร์, มินิคอมพิวเตอร์, เมนเฟรม ซึ่งพีซีเป็นแพล็ตฟอร์มที่นิยมใช้กันมากที่สุดสำหรับไคล์เอนท์เซิร์ฟเวอร์

บรรดาระบบฮาร์ดแวร์ต่าง ๆ ทั้งหลายก็มีความแตกต่างกันทั้งในเรื่องคุณลักษณะและความสามารถ อย่างไรก็ตามก็มีคุณลักษณะร่วมที่จำเป็นสำหรับระบบปฏิบัติการ โดยปกติแล้วแอปพลิเคชั่นถูกเขียนให้รันภายใต้ระบบปฏิบัติการใดระบบหนึ่งโดยเฉพาะ ซอฟต์แวร์ระบบปฏิบัติการที่นิยมใช้กัน เช่น เอ็มเอสดอส (MS-Dos), พีซีดอส (PC-Dos), โอเอสทู, ยูนิกซ์, วีเอ็มเอส (VMS) ของดิจิทัล, เอ็มวีเอสเอ็กซ์เอ (MVS/XA) ที่รันบนไอบีเอ็มเมนเฟรม

คุณลักษณะเบื้องต้นของระบบปฏิบัติการที่จำเป็นสำหรับระบบจัดการฐานข้อมูลไคล์เอนท์เซิร์ฟเวอร์ คือความสามารถในการทำมัลติทาสค์กิง (Multitasking) หรือการรันแอปพลิเคชั่นหลาย ๆ ตัวพร้อมกัน มัลติทาสค์กิง ทำให้ซอฟต์แวร์ระบบจัดการฐานข้อมูลสามารถบริการสอบถามต่าง ๆ และคำขอต่าง ๆ ของบรรดาผู้ใช้ทั้งหลายได้อย่างถูกต้อง โดยปราศจากการรบกวนซึ่งกันและกัน มัลติทาสค์กิงทำงานโดยแบ่งเวลาประมวลผลของซีพียูออกเป็นช่วง ๆ (Time slices) ให้แก่บรรดางานต่าง ๆ และบรรดาโพรเซสต่าง ๆ

นอกจากนี้บางระบบปฏิบัติการยังสามารถเป็นระบบมัลติยูสเซอร์ (Multi-User) ได้ (คือยอมให้ผู้ใช้หลายคนทำงานที่แตกต่างกันได้พร้อมกัน) โดยเฉพาะเมื่อมีการใช้เทอร์มินอลธรรมดาทั้งหลายเป็นทางเข้าสู่ฐานข้อมูลโอเอสแบบมัลติยูสเซอร์ไม่ได้ ก่อให้เกิดข้อดีและข้อเสียโดยเฉพาะเมื่อถูกใช้เป็นแพล็ตฟอร์มสำหรับระบบจัดการฐานข้อมูลไคล์เอนท์เซิร์ฟเวอร์

การเพิ่มความสามารถด้านมัลติเทรด (Multithread) เข้าไปในซอฟต์แวร์โอเอสมัลติทาสค์แบบพรีเอมทีฟ (Preemtive) ความสามารถมัลติเทรดทำให้แอปพลิเคชั่นทำมัลติทาสค์ภายในตัวเองได้ ตัวอย่างเช่น ระบบจัดการฐานข้อมูลผู้ใช้คนเดียวแบบมัลติเทรดสามารถทำการเริ่มต้นใหม่ (โพรเซสหรือทาสค์) ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานออกรายงานที่ซับซ้อนโดยทำอยู่ในเบื้องหลัง (Background) ในขณะที่ผู้ใช้กำลังทำการสอบถามซึ่งทำอยู่ในเบื้องหน้า (Foreground) มัลติเทรดเป็นปัจจัยที่สำคัญสำหรับการออกแบบระบบจัดการฐานข้อมูลไคล์เอนท์เซิร์ฟเวอร์ที่ซับซ้อน มัลติเทรดทำให้แอปพลิเคชั่นสามารถควบคุมให้ทาสค์ใหม่เริ่มทำงานหรือหยุดทำงานได้ แอปพลิเคชั่นสามารถถูกออกแบบให้มีความชาญฉลาดภายในที่ตัดสินได้ว่าโพรเซสใดหรือทาสค์ใดมีลำดับความสำคัญและควรได้รับเวลาซีพียูมากที่สุด ตัวอย่างเช่น ระบบจัดการฐานข้อมูลให้การแก้ไขข้อมูลมีลำดับความสำคัญสูงกว่าการสอบถามข้อมูล เป็นต้น

เนื่องจากจุดด้อยในเรื่องการไม่มีแพล็ตฟอร์มใดที่สามารถสนองความต้องการได้ทุกประการหรือเหมาะสมในทุกสถานการณ์ในระบบไคล์เอนท์เซิร์ฟเวอร์ ในปัจจุบันมีการประยุกต์ใช้ระบบที่เรียกว่าเน็ตเวิร์คคอมพิวติ้ง (Network Computing) เพื่อช่วยแก้ไขในปัญหาของระบบไคล์เอนท์เซิร์ฟเวอร์ ในเรื่องการใช้บุคลากรในการดูแลระบบน้อยลง คือจะไม่ต้องไปดูแลในส่วนของไคล์เอนท์ที่ในอนาคตจะมีจำนวนเพิ่มขึ้นเรื่อย ๆ การง่ายในการปรับปรุงระบบไม่ต้องคอยตามแก้ไขที่ทุกเครื่องไคล์เอนท์ และจากปัญหาการทำงานไม่ครบทุกแพล็ตฟอร์ม ในปัจจุบันมีภาษาคอมพิวเตอร์หนึ่งที่สามารถเขียนโปรแกรมคอมพิวเตอร์โดยไม่คำนึงถึงแพล็ตฟอร์ม ประกอบกับความสามารถทางด้านระบบเน็ตเวิร์คคอมพิวติ้ง ที่ภาษานี้สามารถนำมาประยุกต์ใช้ได้อย่างดี ภาษาดังกล่าวคือ ภาษาจาวา

6.1 ความหมายของเจดีบีซี

เจดีบีซี (JDBC : Java Database Connectivity) ถูกพัฒนาขึ้นโดยแผนกจาวาซอฟต์ (JavaSoft Department) ของบริษัทซันไมโครซิสเต็ม (Sun Micro Systems) ซึ่งก็คือฟังก์ชันมาตรฐานหรือจาวาเอพีไอ (API : Application Programming Interface) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล นักพัฒนาสามารถใช้เจดีบีซี เอพีไอ และยังประกอบด้วยแพ็คเกจอื่น ๆ ด้วย ซึ่งนำเสนอในรูปแบบฟังก์ชันพิเศษหรือเอพีไอระดับสูง เพื่อเขียนโปรแกรมอิสระในการกำหนดระบบจัดการฐานข้อมูล (database connectivity mechanism) ในการเริ่มใช้งาน โดยทั่วไปการใช้ SQL database ในการติดต่อกับเฟรมเวิร์คเพื่อที่จะจัดมาตรฐานในการติดต่อในส่วนบนสุดของชนิดต่าง ๆ ของโมดูลการเชื่อมต่อฐานข้อมูล (database connectivity modules) ซึ่งก็คือมาตรฐานของ ANSI SQL-2 Entry level database เพราะว่าฐานข้อมูลเชิงสัมพันธ์เกือบจะทั้งหมดในปัจจุบันใช้มาตรฐานของ SQL-2 Entry level

เจดีบีซีสร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับโอดีบีซีของบริษัทไมโครซอฟต์ (Microsoft) ซึ่งในปัจจุบันได้ถือว่าเป็นมาตรฐานของการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล และระบบเครือข่ายท้องถิ่น (LAN) แต่ตามหลักการทำงานของทั้งเจดีบีซีและโอดีบีซีตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-Level Interface ของระบบเอ็กซ์วินโดวส์ (X-Windows) และเจดีบีซีไดร์ฟเวอร์ (JDBC Driver) ต้องเข้ากันได้ดีกับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) และต้องผ่านการทดสอบ (Conformance test) ซึ่งจาวาซอฟต์เป็นผู้กำหนดขึ้น

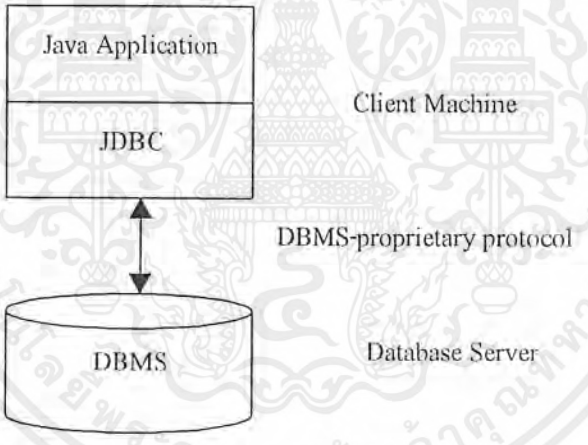
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 รูปแบบการเชื่อมต่อฐานข้อมูลของเจดีบีซี

เจดีบีซี เอพีไอ สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบทูเทียร์โมเดล และ ทรีเทียร์โมเดล

6.2.1 ทูเทียร์โมเดล (Two-Tier Model)

จาวาแอปเพล็ตหรือจาวาแอปพลิเคชั่นจะติดต่อกับฐานข้อมูลโดยตรงจึงมีความจำเป็นที่โปรแกรมจาวาต้องการเจดีบีซีไดร์ฟเวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ 6-1 คำสั่งในการเรียกค้นข้อมูลในรูปของภาษา SQL จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูลก็จะถูกส่งกลับมาสู่ผู้ใช้ ฐานข้อมูลดังกล่าวนี้ส่วนมากจะติดตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้สามารถเชื่อมต่อผ่านระบบเครือข่าย รูปแบบทูเทียร์นี้ใช้หลักการทำงานเช่นเดียวกันกับรูปแบบไคล์เอนท์เซิร์ฟเวอร์ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือไคล์เอนท์ และเครื่องคอมพิวเตอร์ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครือข่ายคอมพิวเตอร์ที่นิยมใช้ในแบบทูเทียร์มักจะเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร



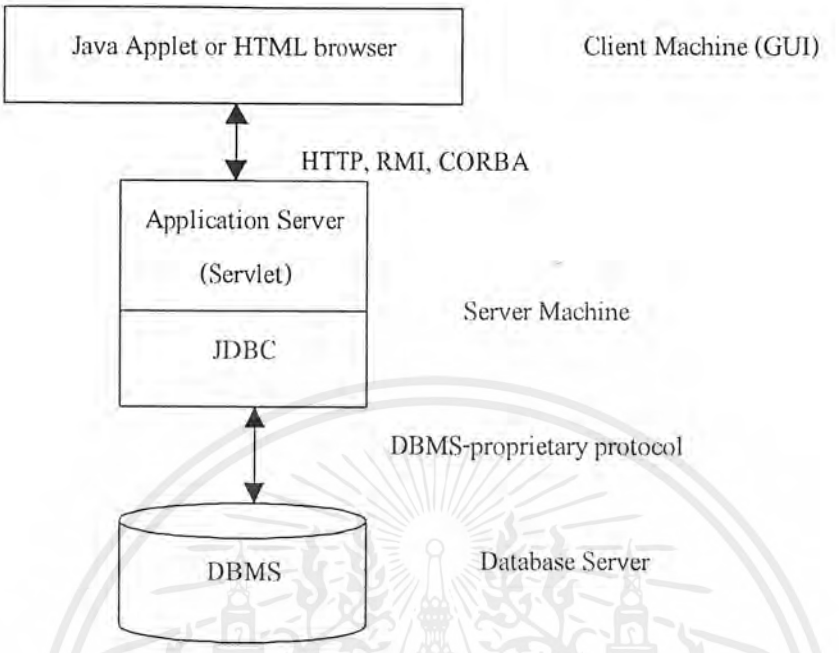
รูปที่ 6-1 ทูเทียร์โมเดล

6.2.2 ทรีเทียร์โมเดล (Three-Tier Model)

คำสั่งต่าง ๆ จากผู้ใช้จะถูกส่งไปให้กับมิดเดิลเทียร์ (Middle Tier) หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้นมิดเดิลเทียร์จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับมิดเดิลเทียร์ และส่งต่อไปให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเตอร์เน็ต (Internet) ซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ (Web-Server) ซึ่งทำหน้าที่เป็นมิดเดิลเทียร์ ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคล์เอนท์และเซิร์ฟเวอร์ฐานข้อมูลสามารถพูดคุยกันได้ การปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

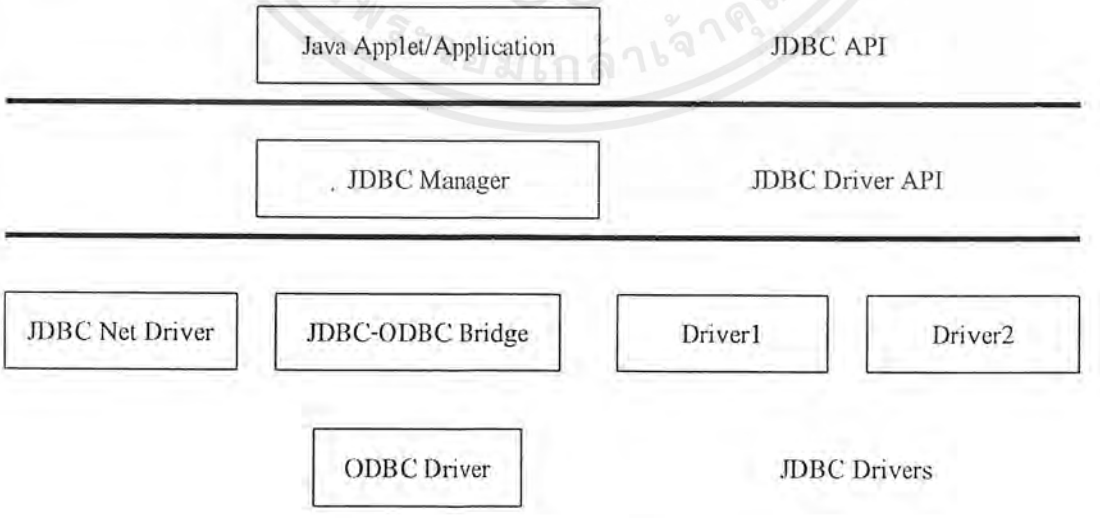
เปลี่ยนระบบคอมพิวเตอร์ของไคล์เอนท์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนฐานข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น



รูปที่ 6-2 ทรีเทียร์โทเดล

6.3 โครงสร้างของเจดีบีซี

โครงสร้างการเชื่อมต่อภายในเจดีบีซีประกอบด้วย 3 ระบบหลักคือ เจดีบีซีเอพีไอ , เจดีบีซีไดร์ฟเวอร์เอพีไอ และ เจดีบีซีไดร์ฟเวอร์ ดังรูปที่ 6-3 ระดับบนสุด เจดีบีซีเอพีไอเป็นระดับของฟังก์ชันเอพีไอที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับกลาง เจดีบีซีไดร์ฟเวอร์ มีไดร์ฟเวอร์ที่ต่างกันอยู่ 4 ชนิดที่เหมาะสม



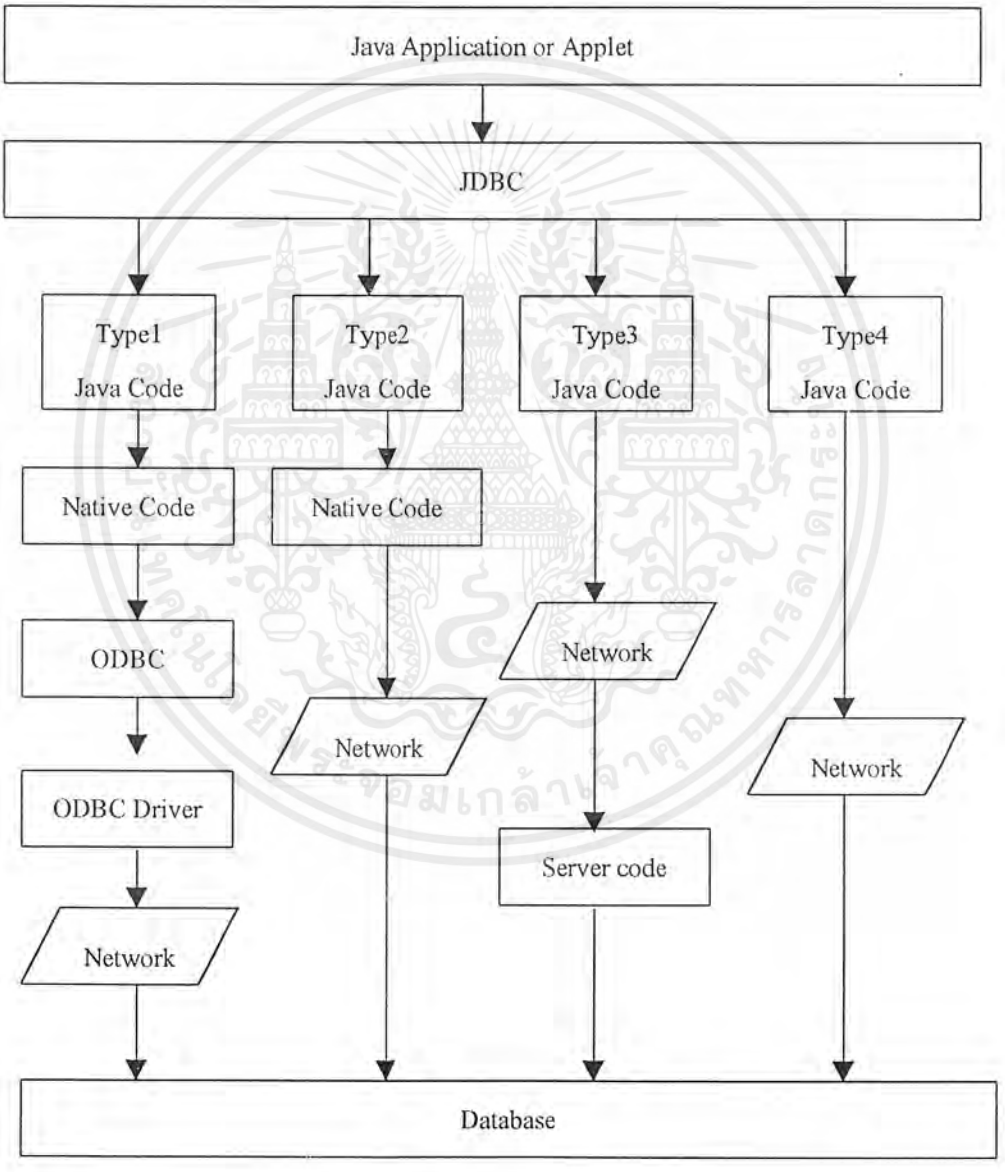
รูปที่ 6-3 ระดับการเชื่อมต่อของเจดีบีซีเอพีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 รูปแบบของเจดีบีซีไดร์ฟเวอร์

เจดีบีซีไดร์ฟเวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชันเอพีไอต่าง ๆ ของเจดีบีซี ถูกจัดแบ่งเป็นทั้งหมด 4 ชนิด คือ

1. JDBC-ODBC Bridge
2. Native-API (Partly-Java) Drivers
3. Network-Protocol (All-Java) Drivers
4. Native-Protocol (All-Java) Drivers



รูปที่ 6-4 ประเภทของเจดีบีซีไดร์ฟเวอร์

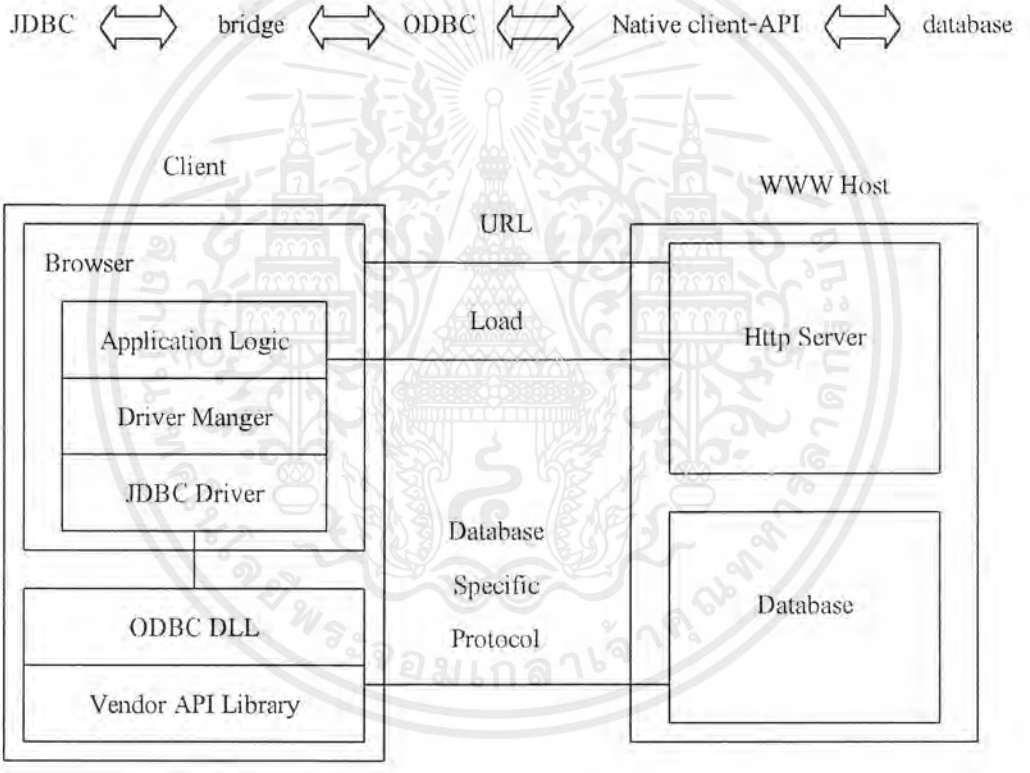
รายละเอียดการทำงานของไดร์ฟเวอร์แต่ละชนิด อธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.1 JDBC/ODBC Bridge

JDBC/ODBC Bridge ถูกพัฒนาโดยจาวาซอฟต์และอินเตอร์โซลว์ (Intersolv) ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลได้โดยป่านการทำงานของโอดีบีซี โดยนำข้อดีของ ODBC-enableed data sources ที่มีใช้อยู่โดยทั่วไปอย่างมากมาย ฝั่งไคล์เอนท์ จาวาแอปเพล็ตหรือจาวาแอปพลิเคชั่นจะถูกเขียนโดยใช้เจดีบีซีเอพีไอ Bridge จะทำการแปลงการเรียกใช้เจดีบีซีไปยังโอดีบีซี และส่งค่าโอดีบีซีไดร์ฟเวอร์ที่เหมาะสมสำหรับแบ็คเอนด์เดต้าเบส (Back-end database)

ข้อดีของ Bridge ทำให้แอปพลิเคชั่นสามารถติดต่อกับฐานข้อมูลได้อย่างง่ายดายโดยจากตัวแทนจำหน่าย (Vendor) ที่มากมาย โดยเลือกโอดีบีซีไดร์ฟเวอร์ที่เหมาะสม อย่างไรก็ตามการติดต่อฐานข้อมูลประเภทนี้ต้องพิจารณาค่าใช้จ่าย (Overhead) และความซับซ้อน (Complexity) เพราะว่าการเรียกใช้จะมีลำดับการเรียกใช้ดังนี้คือ



รูปที่ 6-5การติดต่อเจดีบีซีโดยรูปแบบที่ 1 JDBC/ODBC Bridge

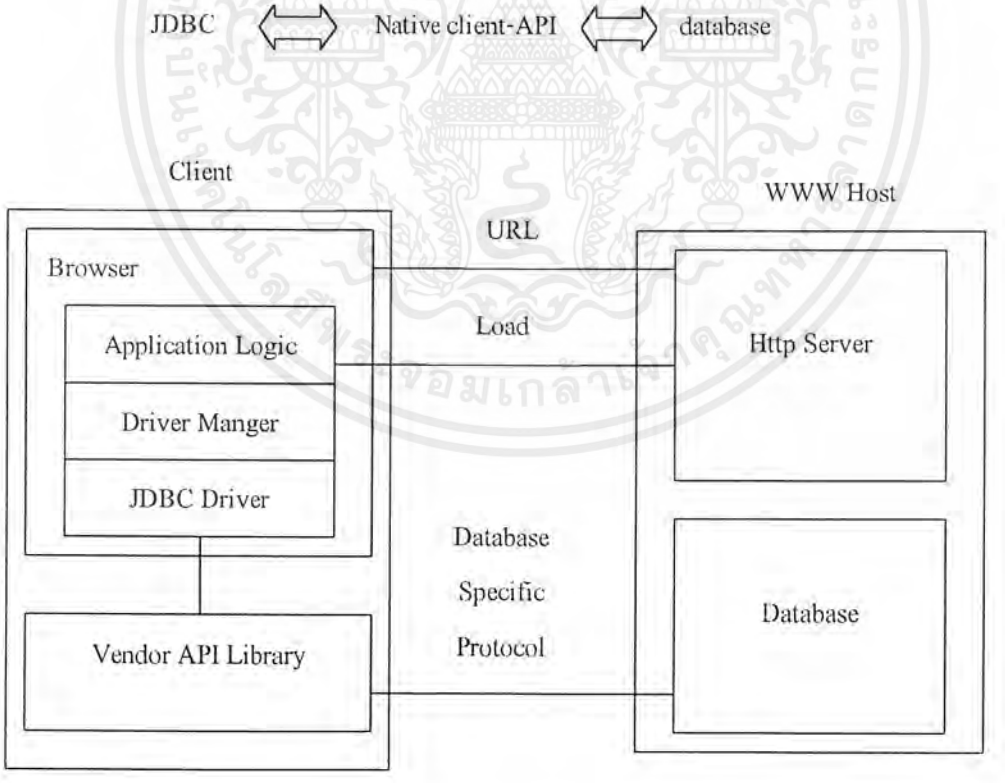
การติดต่อโดยใช้ประเภทนี้ ชนิดของไดร์ฟเวอร์จะไม่อนุญาตสำหรับการย้ายข้อมูลแบบจัสท์อินไทม์ (Just-In-Time) ของจาวาแอปเพล็ต เนทีฟโค้ด (Native Code) ต้องมีการเตรียมการติดตั้งบนเครื่องไคล์เอนท์โดยตรงเพราะว่า JDBC-ODBC Bridge เป็นไดร์ฟเวอร์ที่ถูกเขียนมาจากภาษาซี/ซีพลัสพลัส (C/C++) ในบางส่วนจึงทำให้ไม่สามารถดาวน์โหลด (Download) ข้ามเครือข่ายอินเตอร์เน็ตและแปล (Interpret) บนเว็บบราวเซอร์ได้ เพราะภาษาซี/ซีพลัสพลัส ต้องรันเฉพาะแต่ละระบบเท่านั้น ดังนั้นโอดีบีซีไดร์ฟเวอร์ในรูปแบบของไลบรารีไฟล์ (DLL : Dynamic Link Library) และ JDBC-ODBC Bridge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดร์ฟเวอร์ต้องได้รับการติดตั้งบนไคล์เอนท์แต่ละเครื่องเพื่อให้สามารถติดต่อกับฐานข้อมูลที่รู้จักได้ มีความต้องการในการเตรียมการติดตั้งซอฟต์แวร์โดยมีการจัดการรูปแบบเหมือนแอปพลิเคชั่นประเภทไคล์เอนท์เซิร์ฟเวอร์ ดังนั้น JDBC/ODBC Bridge จะไม่สามารถแก้ไขปัญหาของโปรแกรมไคล์เอนท์ ดังกล่าวได้จึงเหมาะสำหรับการใช้งานกับระบบทรีเทียร์ โดยไดร์ฟเวอร์ทั้งสองจะได้รับการติดตั้งอยู่ที่มิดเดิลเทียร์หรือเครื่องคอมพิวเตอร์ตัวกลาง เช่น เว็บเซิร์ฟเวอร์ เป็นต้น

6.4.2 Native-API (Partly-Java) Drivers

Native-API (Partly-Java) Driver เป็นทูเทียร์ที่ซึ่งเจดีบีซีไดร์ฟเวอร์ต้องการไลบรารีของตัวแทนจำหน่าย (Vendor Library) ในการแปลงฟังก์ชันของเจดีบีซีไปยังคุณลักษณะของการใช้ภาษาในการสืบค้นข้อมูล (Query) เรียกใช้ระบบจัดการฐานข้อมูล ยกตัวอย่างเช่น ไลบรารีสำหรับไซเบส (Sybase) คือ dblib, สำหรับโอราเคิล คือ ocilib ซึ่งไดร์ฟเวอร์เหล่านี้จะถูกใช้เพื่อเขียนในการรวมของภาษาจาวาและซี/ซีพลัสพลัส การติดต่อโดยเจดีบีซีประเภทนี้จะเหมือนกับ JDBC/ODBC Bridge คือต้องการโค้ดซึ่งก็คือไลบรารีของตัวแทนจำหน่าย เพื่อที่จะใช้ในการติดตั้งบนเครื่องไคล์เอนท์ ดังนั้นต้องมีการบำรุงรักษา (Maintenance) ปัญหาเหมือนกับบริดจ์ อย่างไรก็ตามไดร์ฟเวอร์ประเภทนี้จะเร็วกว่าประเภทแรกเพราะว่ามีเลเยอร์พิเศษของการแปลงเป็นโอดีบีซีถูกกำจัดออกไป



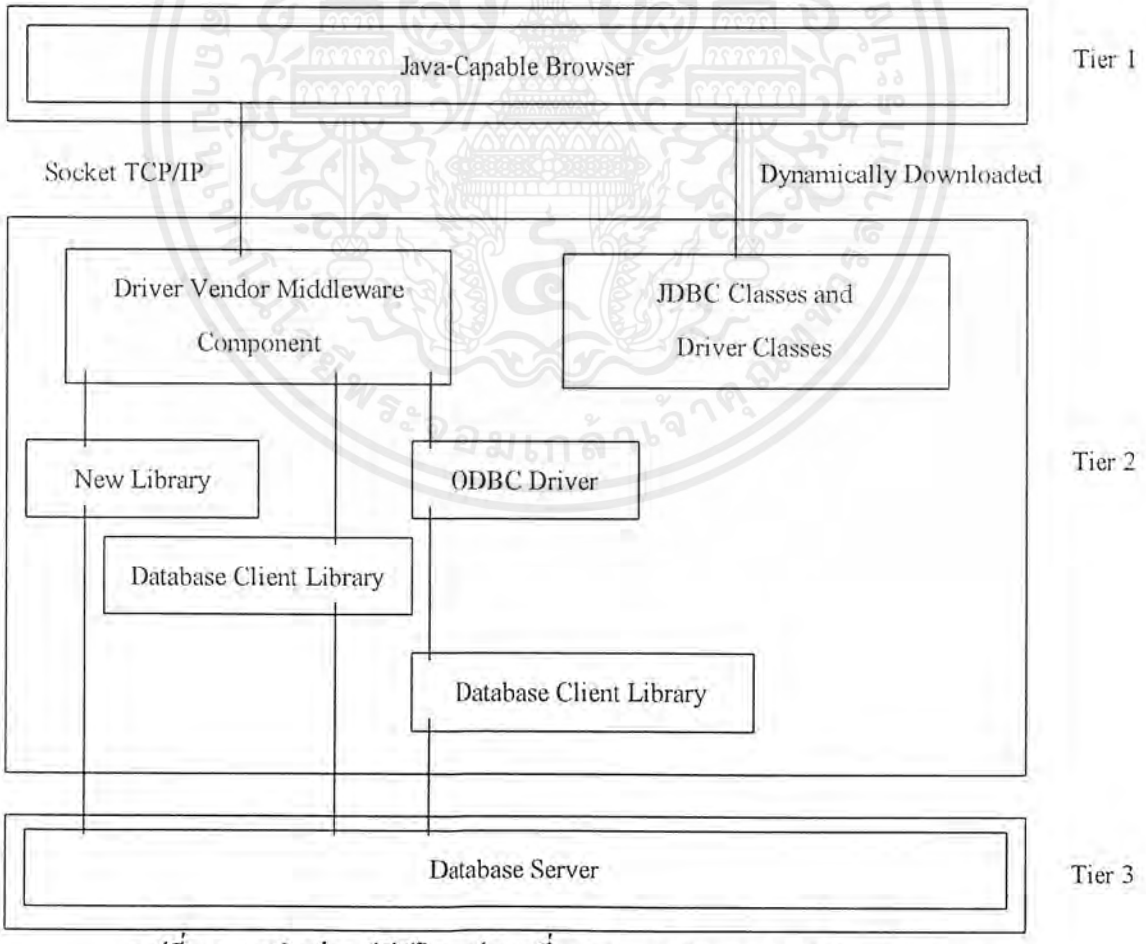
รูปที่ 6-6 การติดต่อเจดีบีซี โดยรูปแบบที่ 2 Native-API (Partly java) Driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของไดร์ฟเวอร์ชนิดนี้คือ สามารถใช้ Native-API Driver ติดต่อกับฐานข้อมูลนั้น ๆ โดยตรง โดยผ่านโปรโตคอลเดิมที่ใช้อยู่ก่อนแล้ว ทำให้เหมาะสมกับการเชื่อมต่อฐานข้อมูลแบบทูเทียร์ไคล์เอนท์เซิร์ฟเวอร์ แต่ก็มีข้อเสียที่คล้ายคลึงกับ JDBC-ODBC Bridge คือต้องติดตั้งไดร์เวอร์ (DLL) และโปรโตคอลไดร์ฟเวอร์ก่อนเช่น SQL-NET, INET, Open Client ฯลฯ บนเครื่องคอมพิวเตอร์นั้นๆ ที่สำคัญส่วนหนึ่งของ Native-API Driver ถูกเขียนขึ้นมาจากภาษาซีพลัสพลัส (Partly-java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเตอร์เน็ตได้และไม่สามารถเชื่อมโยงฐานข้อมูลข้ามชนิดกันได้

6.4.3 Network-Protocol (All-Java) Drivers

Network-Protocol (All –Java) Driver ประเภทนี้ ไดร์ฟเวอร์จะทำการแปลงการเรียกใช้เจดีบีซีให้อยู่ในรูปของเน็ตเวิร์คโปรโตคอลร่วม (DBMS-independent network protocal) ซึ่งหลังจากนั้นจะถูกแปลงให้อยู่ในรูปเฉพาะของแต่ละฐานข้อมูล (database-specific API) บนเซิร์ฟเวอร์นั้น ๆ รูปแบบการเชื่อมต่อฐานข้อมูลจะเป็นลักษณะทรีเทียร์ เจดีบีซีไดร์ฟเวอร์ (ปกติ 200 KB หรือน้อยกว่านั้น) ทำการเอกซีคิวท์บนไคล์เอนท์ และถูกใช้เครื่องมือทางตรรกส่งคำสั่ง SQL ไปยังเครือข่ายและไปยังเจดีบีซีเซิร์ฟเวอร์ เมื่อได้รับข้อมูลกลับมาจากเซิร์ฟเวอร์ และถูกจัดการการเชื่อมต่อ ไดร์ฟเวอร์ประเภทนี้อนุญาตสำหรับการใช้จัสท์อินไทม์บนเครื่องไคล์เอนท์



รูปที่ 6-7 การติดต่อเจดีบีซีโดยรูปแบบที่ 3 Network-Protocol (All-Java) Driver

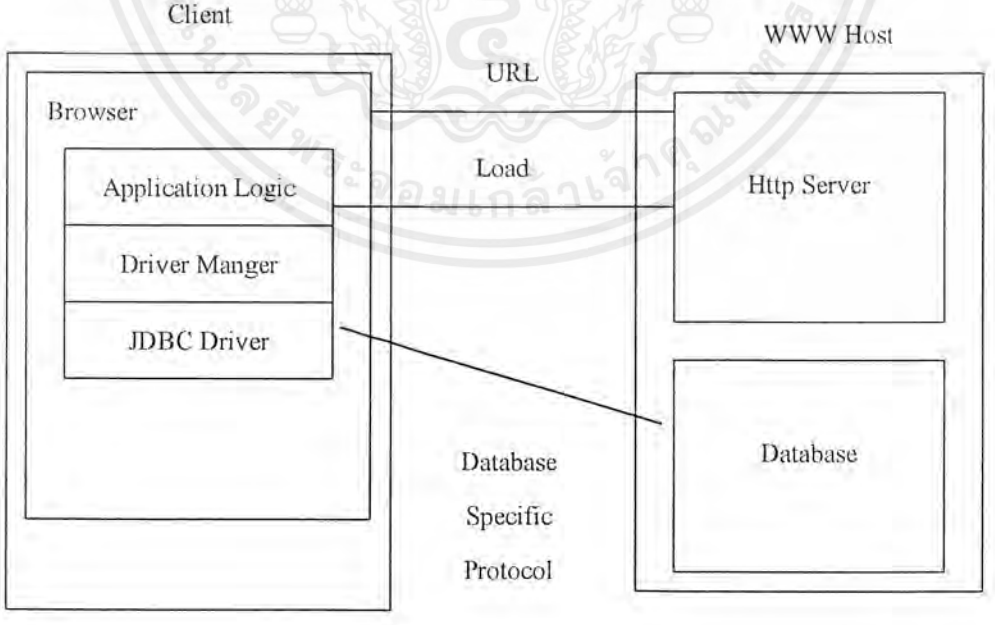
ยกตัวอย่างของ dbAnywhere,SequelLink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Middleware server component สามารถอธิบายเหมือนเนทีฟคอมโพเนนท์ หรือถูกเขียนด้วยจาวา การอธิบายเนทีฟในการเชื่อมต่อไปยังเดต้าเบสเซิร์ฟเวอร์ โดยใช้ไลบรารีของตัวแทนจำหน่าย หรือ โอดีบีซี, Symantc's db Aywre และ Intrsolv's SeqLink ทั้งคู่ถูกจัดให้อยู่ในประเภทนี้ , SequeLink ไม่ต้องการเดต้าเบสไคล์เอนท์ไลบรารีที่จะใช้ในการติดตั้งลงบนเซิร์ฟเวอร์ แต่จะใช้ไลบรารีของตนเอง เซิร์ฟเวอร์ต้องมีการติดตั้งคอนฟิกกูเรชัน (Configuration) สำหรับฐานข้อมูลที่จะใช้ติดต่อ เช่น ติดตั้งหมายเลขของพอร์ต , ตัวแปรของสิ่งแวดล้อมเฉพาะของฐานข้อมูลสำหรับอินสแตนซ์ และตัวแปรอื่น ๆ ที่เซิร์ฟเวอร์อาจจะต้องการ ถ้าเซิร์ฟเวอร์ Middleware ถูกเขียนด้วยภาษาจาวา สามารถใช้ JDBC-compliant driver ในการติดต่อกับระบบจัดการฐานข้อมูลโดย Database Proprietary Protocol ของตัวแทนจำหน่ายไม่ต้องการที่จะคอนฟิกที่เซิร์ฟเวอร์สำหรับฐานข้อมูล

6.4.4 Native-Protocol (All-Java) Drivers

Native-Protocol (All-Java) Drivers จะแปลงการเรียกใช้คำสั่งของเจอีบีซีให้อยู่ในรูปแบบของเน็ตเวิร์คโปรโตคอลเฉพาะของฐานข้อมูลนั้นโดยตรง โดยคุณสมบัติของตัวแทนจำหน่ายฐานข้อมูล ไดร์ฟเวอร์เหล่านี้สามารถเขียนในภาษาจาวาและสามารถติดต่อกับแอปเพล็ตแบบจัสท์อินไทม์ เพราะว่าไดร์ฟเวอร์เหล่านี้จะแปลงเจดีบีซีตรงไปยังเนทีฟโปรโตคอลโดยปราศจากการใช้โอดีบีซีหรือเนทีฟเอพีไอ ซึ่งสามารถจัดหามาสำหรับการติดต่อฐานข้อมูลที่มีประสิทธิภาพสูง ไดร์ฟเวอร์เหล่านี้สามารถสร้างจากตัวแทนจำหน่ายระบบจัดการฐานข้อมูลเท่านั้น เนื่องจากความจริงที่ว่าความรู้ของโปรโตคอลอยู่ที่ตัวแทนจำหน่าย การนำการติดต่อประเภทนี้มาใช้งานมีอยู่เล็กน้อยมากในปัจจุบัน แต่ในอนาคตคาดว่าจะเพิ่มจำนวนขึ้นเรื่อย ๆ



รูปที่ 6-8 การติดต่อเจดีบีซีโดยรูปแบบที่ 4 Native-Protocol (All-Java) Driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของไดร์ฟเวอร์ชนิดนี้คือ ไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละองค์กร และไม่มีความจำเป็นที่ต้องติดตั้งไดร์ฟเวอร์ตัวกลาง เช่น โอดีบีซี บนไคล์เอนท์แต่ละเครื่อง จึงเหมาะสำหรับการพัฒนาการเชื่อมต่อฐานข้อมูลแบบทูเทียร์ สำหรับเครือข่ายอินทราเน็ตภายในองค์กร ไดร์ฟเวอร์ชนิดนี้มีข้อเสียคือ ไม่เหมาะที่จะนำไปใช้บนเครือข่ายอินเตอร์เน็ต เพราะปัญหาเรื่องความปลอดภัย การเชื่อมต่อฐานข้อมูลไม่สามารถกระทำข้ามระบบที่แตกต่างกันได้ ต้องใช้ไดร์ฟเวอร์เฉพาะสำหรับฐานข้อมูลตัวนั้น ๆ

เจดีบีซีไดร์ฟเวอร์ชนิดที่ 3 และ 4 คือไดร์ฟเวอร์ที่คาดว่าจะเป็นสิ่งที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคตเพราะถูกเขียนขึ้นมาจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและความคล่องมากกว่า เช่นมีการติดตั้งไดร์ฟเวอร์โดยอัตโนมัติเมื่อมีการดาวน์โหลดจาวาแอปเพล็ต สำหรับไดร์ฟเวอร์ชนิดที่ 1 และ 2 คือไดร์ฟเวอร์ที่พัฒนาขึ้นมาเพื่อใช้งานร่วมกับไดร์ฟเวอร์ดั้งเดิม (ซึ่งก็คือ โอดีบีซี) หรือเน็ตเวิร์คโปรโตคอลเฉพาะฐานข้อมูลที่มีอยู่แล้วไปพลาง ๆ ก่อน ซึ่งแน่นอนว่าไดร์ฟเวอร์ชนิดที่ 1 และ 2 มีข้อจำกัดในการใช้งานมากกว่าชนิดที่ 3 และ 4 โดยเฉพาะเรื่องความปลอดภัยและบริดจ์ร่วมกับโอดีบีซีไดร์ฟเวอร์บนวินโดวส์ 95/เอ็นที ผลปรากฏว่าไม่ยุ่งยากและให้ประสิทธิภาพที่ดีพอสมควร เหมาะสมในการเป็นเครื่องมือในการทดลองและศึกษาขั้นตอนในการเชื่อมต่อกับฐานข้อมูลตั้งแต่ขั้นพื้นฐานจนถึงขั้นตอนใช้ปฏิบัติงานจริง

6.5 ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล

1. เปิดการเชื่อมต่อกับฐานข้อมูล
2. ส่งคำสั่ง SQL ไปให้แก่ฐานข้อมูล
3. จัดการกับผลลัพธ์ที่ได้รับ
4. ปิดการเชื่อมต่อ

และเพื่อให้เข้าใจการทำงานของเจดีบีซียิ่งขึ้น ตัวอย่างโปรแกรมบางส่วนต่อไปนี้จะแสดงให้เห็นถึงขั้นตอนทั้งสี่ในการติดต่อกับฐานข้อมูลผ่านโอดีบีซีไดร์ฟเวอร์ โดยใช้ JDBC-ODBC Bridge (ฟังก์ชันเอพีไอต่าง ๆ ของเจดีบีซีที่มาพร้อมกับเจดีเคเวอร์ชัน 1.1 ขึ้นไปถูกเก็บรวบรวมไว้ในแพ็คเกจ java.sql.*)

```
// (1) เปิดการเชื่อมต่อกับฐานข้อมูล
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection("jdbc:odbcmyDB",login,password);
// (2) ส่งคำสั่ง SQL ไปให้แก่ฐานข้อมูล
Statement stmt = con.createStatement();
ResultSet rs = stmt.execcuteQuery("SELECT a,b,c FROM Table1");
// (3) จัดการกับผลลัพธ์ที่ได้
while (rs.next()){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i = rs.getInt("a");
String s = rs.getString("b");
float f = rs.getFloat("c");
}
// (4) ปิดการเชื่อมต่อ
stmt.close();
con.close();
```

จากตัวอย่างสิ่งแรกที่ทำคือเลือกชนิดของเจดีบีซีไดร์ฟเวอร์ (ในที่นี้เลือก JDBC-ODBC Bridge) ระบุชื่อฐานข้อมูล (DSN : Data Source Name) รวมถึงชื่อผู้ใช้ (Login) และรหัสผ่าน (Password) จากนั้นเป็นการสร้างออบเจกต์ stmt (คลาส Statement) สำหรับการเรียกค้นข้อมูล และจัดส่งคำสั่ง SQL ไปในรูปแบบสตริงผ่านเมทธอด executeQuery() ผลลัพธ์ที่ได้จากเมทธอดนี้คือผลลัพธ์จากการประมวลผลของฐานข้อมูลซึ่งจะเก็บอยู่ในออบเจกต์ rs (คลาส ResultSet) จากนั้นข้อมูลที่ได้รับนี้จะถูกถ่ายทอดให้อยู่ในชนิดข้อมูลที่จาวารู้จัก เช่น int , String, float ฯลฯ เพื่อนำไปใช้ประโยชน์อื่นต่อไป สุดท้ายคือการปิดการเชื่อมต่อกับฐานข้อมูล

ภาษาจาวาและเจดีบีซีคือเครื่องมือที่จะทำให้การเชื่อมต่อสื่อสารกับระบบฐานข้อมูลเป็นไปอย่างสะดวกรวดเร็วและเป็นหนึ่งเดียว (Database-neutal Communication) อย่างไรก็ตามพื้นฐานที่จะขาดเสียไม่ได้สำหรับการเชื่อมต่อฐานข้อมูลนั้นก็คือความรู้ความเข้าใจเกี่ยวกับหลักการทำงาน หลักการออกแบบฐานข้อมูลเชิงสัมพันธ์ ระบบจัดการฐานข้อมูล และภาษา SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การพัฒนาต้นแบบระบบสารสนเทศอาคารและสถานที่ของสจล.

7.1 ภาพรวมของโครงงาน

- เป็นต้นแบบระบบสารสนเทศอาคารและสถานที่ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (KMITL) โดยสามารถใช้งานได้ 2 ทางคือ เป็นแอปพลิเคชั่น และใช้งานผ่านอินเตอร์เน็ต
- เป็นระบบที่พัฒนาขึ้นมาโดยใช้ภาษาจาวา (ใช้เครื่องมือ JBuilder)
- พัฒนาขึ้นมาโดยใช้หลักการของออบเจกต์โอเรียนเต็ด
- ใช้ฐานข้อมูลโอราเคิล8.0.5 ในการจัดเก็บข้อมูล ทั้งข้อมูลเชิงพื้นที่และข้อมูลเชิงบรรยาย
- ใช้ Spatial Cartridge ของโอราเคิล ซึ่งมีฟังก์ชัน และ โพรซิเดอร์ที่ช่วยในการจัดการและจัดเก็บข้อมูลเชิงพื้นที่ ช่วยให้สามารถจัดการกับข้อมูลได้ง่ายขึ้น
- ใช้เมาส์และคีย์บอร์ดในการควบคุมการทำงาน
- รองรับผู้ใช้หลายคน (Multi User) ผ่านเครือข่ายอินเตอร์เน็ต
- สำหรับการใช้งานผ่านอินเตอร์เน็ตผู้ใช้จากทั่วทุกมุมโลกสามารถเข้าใช้บริการได้ ด้วยบราวเซอร์ที่สนับสนุน JDK เวอร์ชัน 1.1 ขึ้นไป เช่น เน็ตสเคปเวอร์ชัน 4 ขึ้นไป
- ใช้ผู้พัฒนาจำนวน 2 คน

7.2 ฟังก์ชันการทำงานของระบบ

- แสดงภาพแผนที่อาคาร หมุดหลักฐาน ขอบเขตสถาบัน ตลอดจนห้องต่างๆ ภายใน สจล. โดยสามารถเลือกแสดงเฉพาะเลเยอร์ที่ต้องการได้
- เลือกดูรายละเอียดข้อมูลในส่วนที่ต้องการได้
- ย่อ และ ขยายภาพแผนที่ได้
- สืบค้นข้อมูลโดยการกำหนดเงื่อนไข
- เปลี่ยนสีการแสดงผลของแต่ละเลเยอร์ได้
- เปลี่ยนหน้าต่างการแสดงผลได้ 3 แบบ ได้แก่ แบบ Java , แบบ Motif และแบบ Window

7.3 กลุ่มผู้ใช้เป้าหมาย

- อาจารย์ ข้าราชการ สจล.
- นิสิต นักศึกษา สจล.
- บุคคลทั่วไป

7.4 เงื่อนไขทั่วไป

- การย่อ และการขยาย จะมีขีดจำกัดที่ระดับหนึ่ง ซึ่งไม่สามารถย่อหรือขยายภาพต่อไปได้
- ใช้เมาส์และคีย์บอร์ดในการควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 ข้อกำหนดความต้องการของระบบ

7.5.1 ความต้องการของระบบในส่วนต่างๆ (Requirement)

7.5.1.1 บทนำ

ระบบนี้เป็นต้นแบบระบบสารสนเทศอาคารและสถานที่ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อเข้าสู่ระบบแล้ว ระบบจะแสดงภาพแผนที่ของทั้งสถาบันให้ผู้ใช้ได้ดู ผู้ใช้สามารถเลือกดูเฉพาะเลเยอร์ที่สนใจได้ เช่น เลือกดูเฉพาะเลเยอร์อาคาร และขอบเขตสถาบันเท่านั้น สามารถย่อและขยายภาพได้ และสามารถเข้าไปดูภายในตัวอาคารต่างๆ ว่าประกอบไปด้วยชั้นต่างๆกี่ชั้น แต่ละชั้นประกอบไปด้วยห้องอะไรบ้าง โดยสามารถเข้าไปดูรายละเอียดในแต่ละห้องได้ นอกจากนี้ผู้ใช้ยังสามารถทำการค้นหาข้อมูลโดยการกำหนดเงื่อนไขต่างๆ ตามที่ต้องการได้อีกด้วย

7.5.1.2 อินพุท (Input)

รับอินพุทจากเมาส์และคีย์บอร์ดเพื่อใช้งานระบบ

7.5.1.3 กระบวนการ (Processing)

- การตั้งค่าเริ่มต้นต่างๆ ของระบบ
- การติดต่อกับฐานข้อมูลโอราเคิล
- การตรวจสอบชื่อผู้ใช้และรหัสผ่าน
- การสร้างและการแสดงผลข้อมูลภาพแผนที่
- การย่อและขยายภาพแผนที่
- การแสดงข้อมูลรายละเอียด
- การสืบค้นข้อมูลโดยกำหนดเงื่อนไข

7.5.1.4 ผลที่ได้รับ (Output)

หากมีการรับอินพุทเข้ามา จะผ่านขั้นตอนตามกระบวนการดังกล่าวแล้วแสดงภาพแผนที่ และข้อมูลรายละเอียดออกทางจอภาพ

7.5.2 ความต้องการในการติดต่อกับภายนอก (External Interface Requirement)

7.5.2.1 ส่วนติดต่อกับผู้ใช้ (User Interface)

ส่วนติดต่อกับผู้ใช้จะแบ่งออกเป็นหน้าต่างหลักซึ่งแสดงภาพแผนที่โดยรวมทั้งสถาบันและหน้าต่างซึ่งแสดงแผนที่ภายในตัวอาคาร ประกอบไปด้วยส่วนแสดงภาพแผนที่ Panel แสดงแต่ละเลเยอร์ เมนูคำสั่งต่างๆ และปุ่มต่างๆ ส่วนติดต่อกับผู้ใช้จะต้องใช้งานได้ง่าย ไม่ซับซ้อน ผู้ใช้สามารถศึกษาและใช้งานได้อย่างรวดเร็ว

7.5.2.2 ส่วนติดต่อกับอุปกรณ์ภายนอก (Hardware Interface)

- ใช้เมาส์ในการออกคำสั่งแก่ระบบ เช่นการย่อ/ขยาย แผนที่ การเลือกแสดงเฉพาะเลเยอร์ที่ต้องการ การเปิด/ปิดหน้าต่าง การเปลี่ยนสีเลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้คีย์บอร์ดในการป้อนข้อมูลต่างๆ เช่นข้อมูลรายละเอียดของห้อง ข้อมูลที่จะทำการสืบค้น เป็นต้น

7.6 ความต้องการด้านประสิทธิภาพ (Performance Requirement)

- สำหรับการใช้งานผ่านอินเตอร์เน็ต เครื่องคอมพิวเตอร์ต้องสามารถเชื่อมต่ออินเตอร์เน็ตได้
- จอแสดงผลที่สามารถแสดงผลได้ด้วยความละเอียดอย่างต่ำ 640x800
- โปรแกรมบราวเซอร์ที่รองรับ JDK1.1 (เช่น Netscape เวอร์ชัน 4 ขึ้นไป)

7.7 คุณสมบัติ (Attribute)

7.7.1 ความปลอดภัยของระบบ

มีการแบ่งระดับการเข้าถึงเข้ามูลของผู้ใช้แต่ละประเภท โดยที่ User ปกติจะไม่สามารถแก้ไขข้อมูลได้ สามารถดูข้อมูลได้เพียงอย่างเดียว จะมีเพียง Admin เท่านั้นที่สามารถแก้ไขข้อมูลได้

7.7.2 การดูแลรักษาระบบ

เนื่องจากระบบนี้พัฒนาขึ้นมาโดยใช้หลักการของออบเจกต์โอเรียนเต็ด ดังนั้นการดูแลรักษาระบบ สามารถดูแลรักษาได้ง่าย เพราะโมดูล (Module) ต่างๆ ในระบบมีความเป็นอิสระต่อกัน การจะทำการแก้ไขโมดูลใดโมดูลหนึ่งจึงไม่มีผลกระทบต่อโมดูลอื่นๆ

7.7.3 ความง่ายในการใช้งาน

สามารถใช้งานได้ง่าย ใช้เป็นภายในเวลา 1 ชั่วโมง

7.8 การสร้างระบบ

การสร้างระบบมีส่วนประกอบดังนี้

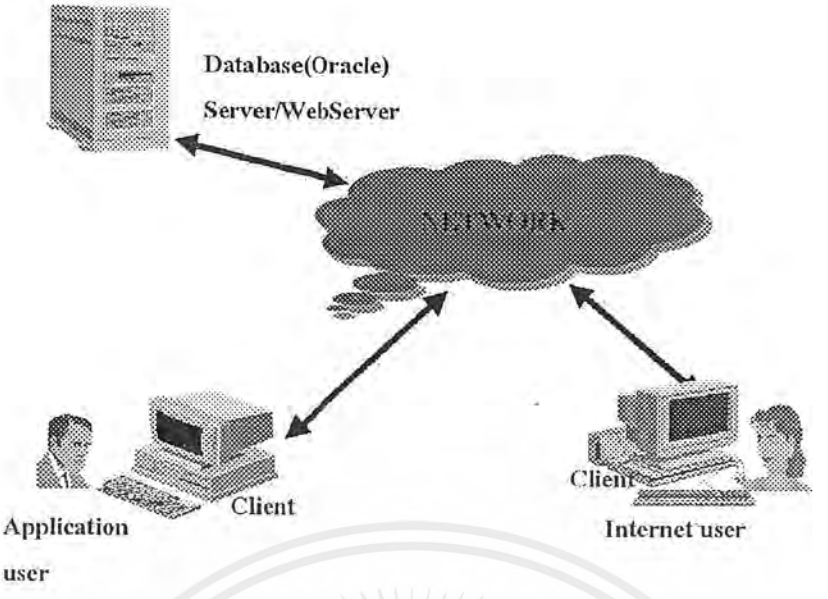
1. โครงสร้างของระบบ
2. การออกแบบฐานข้อมูล
3. การนำข้อมูลเข้า
4. การเขียนโปรแกรมติดต่อกับฐานข้อมูล

7.8.1 โครงสร้างของระบบ

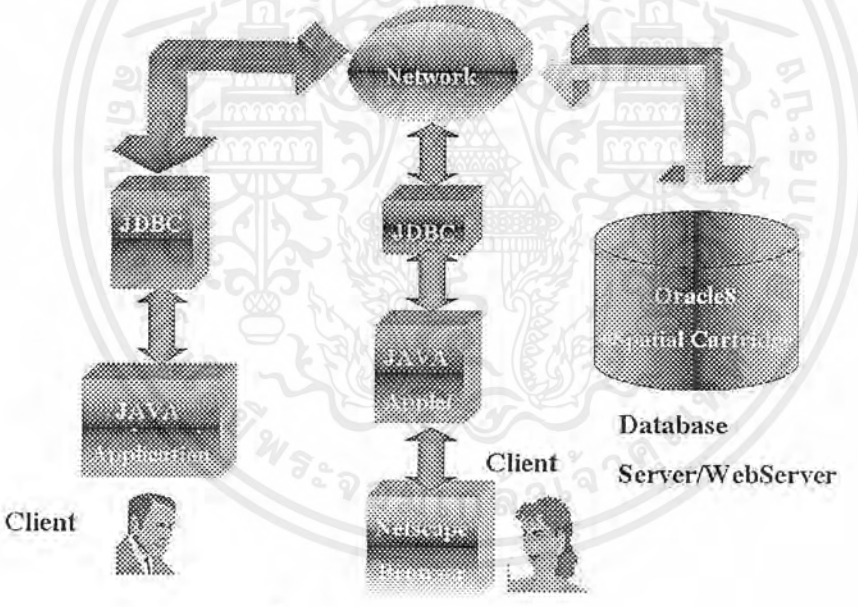
ระบบจะประกอบไปด้วย

1. เครื่องเซิร์ฟเวอร์ ซึ่งเป็นทั้งเดต้าเบสเซิร์ฟเวอร์ (Database Server) และเว็บเซิร์ฟเวอร์ (Web Server) เป็นส่วนที่เก็บข้อมูลทั้งข้อมูลเชิงบรรยายและข้อมูลเชิงพื้นที่ รวมถึงเก็บคลาสต่างๆที่โปรแกรมต้องใช้
2. เครื่องไคล์เอนต์ เป็นส่วนที่ทำหน้าที่ประมวลผลและแสดงผลหน้าจอแผนที่ให้แก่ผู้ใช้ โดยแบ่งเป็น 2 ชนิด ได้แก่
   - ไคล์เอนต์ที่ใช้แอปพลิเคชั่น ใช้งานผ่านแอปพลิเคชั่น
   - ไคล์เอนต์ที่ใช้อินเตอร์เน็ต ใช้งานผ่านอินเตอร์เน็ตโดยใช้เน็ตสเคปเวอร์ชัน 4 ขึ้นไป
3. เครือข่ายคอมพิวเตอร์ เป็นส่วนที่ใช้เชื่อมการติดต่อระหว่างไคล์เอนต์กับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-1 โครงสร้างของระบบ



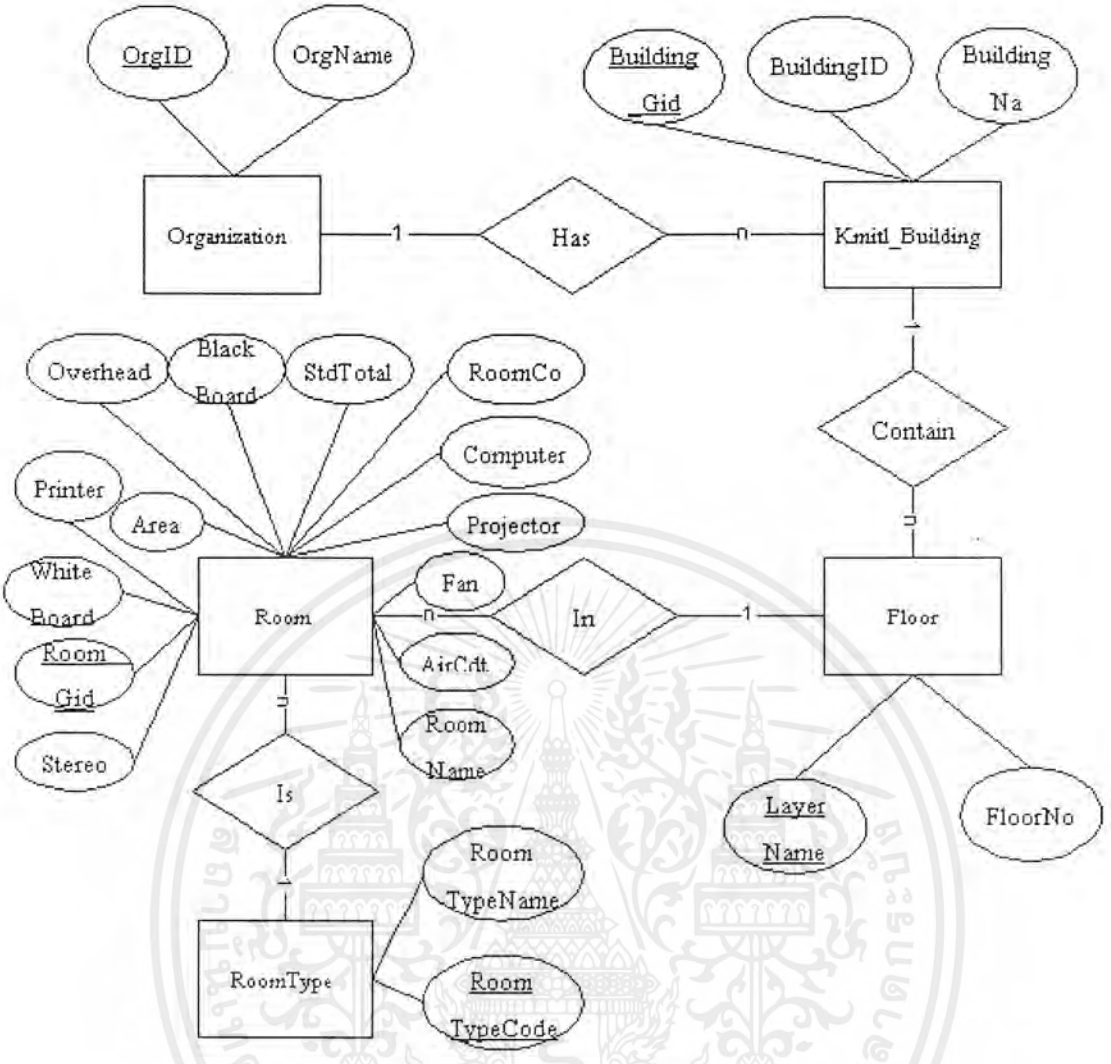
รูปที่ 7-2 โครงสร้างการเชื่อมต่อของระบบ

7.8.2 การออกแบบฐานข้อมูล

7.8.2.1 อีอาร์ไดอะแกรม (ER Diagram)

อีอาร์ไดอะแกรมแสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-3 อีอาร์ไดอะแกรม

7.8.2.2 เดต้าดิคชันนารี (Data Dictionary)

ชื่อ	คำอธิบาย	ชนิด
AirCdt (Room)	จำนวนเครื่องปรับอากาศภายในห้อง	Attribute
Area (Room)	พื้นที่ของห้อง	Attribute
BlackBoard (Room)	จำนวนกระดานดำภายในห้อง	Attribute
Building_GID (Kmitl_Building)	หมายเลขประจำอาคารสถานที่	Attribute
BuildingID (Kmitl_Building)	รหัสอาคารสถานที่	Attribute
BuildingNa(Kmitl_Building)	ชื่ออาคารสถานที่	Attribute
Computer (Room)	จำนวนเครื่องคอมพิวเตอร์ภายในห้อง	Attribute
Contain	ความสัมพันธ์แบบ 1 ต่อ หลาย ระหว่าง Kmitl_Building กับ Floor อาคารหนึ่งมีได้หลายชั้น	Relation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	แต่ชั้นชั้นหนึ่งจะอยู่ภายในอาคารเดียว	
Fan (Room)	จำนวนพัดลมภายในห้อง	Attribute
Floor	ชั้นภายในอาคาร	Entity
FloorNo (Floor)	หมายเลขชั้นภายในอาคาร	Attribute
Has	ความสัมพันธ์แบบ 1 ต่อ หลาย ระหว่าง Organization กับ Kmitl_Building ในหน่วยงานหนึ่ง ๆ มีอาคารสถานที่ได้หลายแห่ง แต่อาคารสถานที่หนึ่งๆ จะสังกัดในหน่วยงานเดียว	Relation
In	ความสัมพันธ์แบบหลายต่อ 1 ระหว่าง Room กับ Floor ห้อง 1 ห้องอยู่ได้ในชั้นเดียว แต่ชั้น 1 ชั้นมีได้หลายห้อง	Relation
Is	ความสัมพันธ์แบบหลายต่อ 1 ระหว่าง Room กับ RoomType ห้อง 1 ห้องมีชนิดของห้องได้เพียงชนิดเดียว แต่ ชนิดของห้องชนิดหนึ่งมีได้หลายห้อง	Relation
Kmitl_Building	อาคารสถานที่	Entity
LayerName (Floor)	ชื่อเลเยอร์ของชั้น (ใช้ประโยชน์ในการเขียนโปรแกรม)	Attribute
Organization	หน่วยงาน	Entity
OrgID (Organization)	หมายเลขประจำหน่วยงาน	Attribute
OrgName (Organization)	ชื่อหน่วยงาน	Attribute
Overhead (Room)	จำนวนเครื่องฉายข้ามศีรษะภายในห้อง	Attribute
Printer (Room)	จำนวนเครื่องพิมพ์ภายในห้อง	Attribute
Projector (Room)	จำนวนเครื่องโปรเจคเตอร์ภายในห้อง	Attribute
Room	ห้องภายในชั้น	Entity
RoomCo (Room)	รหัสอ้างอิงสถานที่	Attribute
Room_GID (Room)	หมายเลขประจำห้อง	Attribute
RoomName (Room)	ชื่อห้อง	Attribute
RoomType	ชนิดของห้อง	Entity
RoomTypeCode (RoomType)	รหัสอ้างอิงชนิดห้อง	Attribute
RoomTypeName (RoomType)	ชื่อชนิดห้อง	Attribute
StdTotal (Room)	จำนวนคนที่ห้องสามารถรองรับได้	Attribute
Stereo (Room)	จำนวนเครื่องเสียงภายในห้อง	Attribute
WhiteBoard (Room)	จำนวนไวท์บอร์ดภายในห้อง	Attribute

ตารางที่ 7-1 เดต้าดิคชันนารี

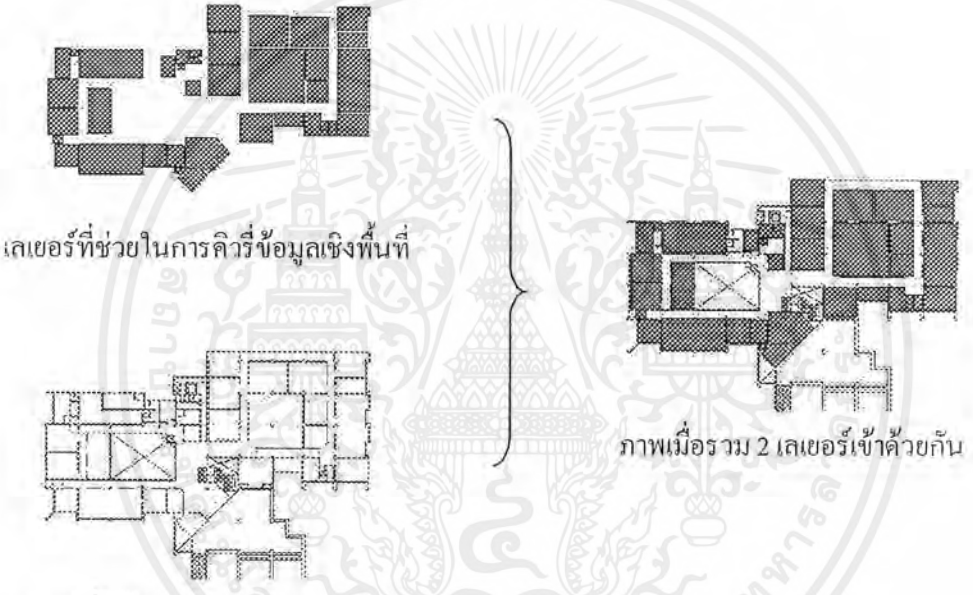
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.3.3 โครงสร้างการจัดเก็บข้อมูล

เราจะสร้างตาราง ที่มีโครงสร้างตามที่ Spatial cartridge กำหนดเพื่อให้สามารถใช้ฟังก์ชันและโพรซิเดอร์ของ Spatial cartridge ได้

สำหรับข้อมูลที่ใช้ จะใช้ข้อมูลแผนที่จากผังแม่บท สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งประกอบไปด้วยเลเยอร์ต่าง ๆ ดังนี้

- เลเยอร์อาคาร
- เลเยอร์หมุดหลักฐาน
- เลเยอร์ขอบเขตสถาบัน
- เลเยอร์ชั้นต่างๆ ภายในอาคาร



รูปที่ 7-4 เลเยอร์ที่ใช้แสดงภาพกราฟฟิกและเลเยอร์ที่ช่วยในการคิวรี่

แต่ละเลเยอร์ในทางลอจิกจะประกอบไปด้วย เลเยอร์ที่ทำการจัดเก็บจริง 2 เลเยอร์ ได้แก่ เลเยอร์ที่ใช้ในการแสดงภาพกราฟฟิก และ เลเยอร์ที่ใช้ในการกำหนดส่วนที่สนใจเพื่อช่วยในการคิวรี่ข้อมูลเชิงพื้นที่

สำหรับแต่ละเลเยอร์ที่จัดเก็บจริงจะประกอบด้วย 4 ตาราง ตามโครงสร้างที่ Spatial cartridge กำหนดไว้ได้แก่

- <layername>_sdolayer
- <layername>_sdodim
- <layername>_sdogeom
- <layername>_sdoindex

สำหรับรายละเอียดแต่ละคอลัมน์ในตารางดังกล่าว ได้กล่าวอธิบายไว้แล้วในบทที่ 5 (หัวข้อ 5.1.3 โครงสร้างของฐานข้อมูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเลเยอร์ที่ใช้ช่วยในการคิวรี่ข้อมูลเชิงพื้นที่ จะมีตารางเพิ่มขึ้นมาอีก 1 ตาราง สำหรับแต่ละเลเยอร์เพื่อใช้ในการจัดเก็บข้อมูลเชิงบรรยายของเลเยอร์นั้น

นอกเหนือจาก ตารางที่มีโครงสร้างตามที่ Spatial cartridge แล้ว ยังมีตารางอื่นๆ อีก ได้แก่ ที่ใช้ในการแม็พระหว่างอาคารกับชั้นต่างๆ ภายในอาคาร , ตารางที่ใช้สำหรับช่วยในการสืบค้นข้อมูล

เลเยอร์อาคาร (Building Layer) ประกอบด้วยตารางต่าง ๆ ดังนี้

- ตาราง Building_sdolayer
- ตาราง Building_sdodim
- ตาราง Building_sdogeom
- ตาราง Building_sdoindex
- ตาราง Kmitl_Building เก็บข้อมูลที่บอกรายละเอียดของเลเยอร์อาคาร

เลเยอร์หมุดหลักฉาวร

ประกอบไปด้วยตาราง ที่ขึ้นต้นด้วย Bmp 4 ตาราง ตามที่ Spatial cartridge กำหนดไว้

เลเยอร์ขอบเขตสถาบัน

ประกอบไปด้วยตาราง ที่ขึ้นต้นด้วย Boundary 4 ตาราง ตามที่ Spatial cartridge กำหนดไว้

เลเยอร์ชั้นภายในอาคาร

แต่ละชั้นภายในอาคารประกอบไปด้วยตารางตามที่ Spatial cartridge กำหนด 4 ตาราง

ตารางที่ใช้ในการแม็พระหว่างอาคารกับชั้นต่างๆ ภายในอาคาร

ตาราง Floor เป็นตารางที่ช่วยในการเชื่อมโยงระหว่างแผนที่หน้าหลักของสถาบัน กับชั้นต่าง ๆ ภายในอาคาร ซึ่งมีฟิลด์ต่าง ๆ ดังนี้

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	Key
1	LayerName	Varchar(200)	ชื่อเลเยอร์ของชั้น	P.F
2	FloorNo	Number(3)	หมายเลขชั้นภายในอาคาร	
3	Building_GID	Number (10)	หมายเลขประจำอาคารสถานที่	

ตารางที่ 7-2 ตารางที่ช่วยในการแม็พระหว่างอาคารกับชั้นต่าง ๆ ภายในอาคาร

การสร้างตารางทำโดยใช้คำสั่ง CREATE TABLE ในภาษา SQL

นอกจากนี้ยังมีตารางที่ช่วยในการคอนฟิก (Configuration) ระบบ ชื่อว่าตาราง GISDESKTOP_CONFIG ซึ่งมีคอลัมน์ต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคอลัมน์	ความหมาย
TABLE_KEY	ชื่อของ table key ที่ใช้ในโปรแกรม
TABLE_NAME	ชื่อของตารางที่เก็บจริงในฐานข้อมูล

ตารางที่ 7-3 โครงสร้างตารางที่ช่วยในการคอนฟิกระบบ

สำหรับข้อมูลในตารางนี้ ในคอลัมน์ TABLE_KEY ห้ามทำการปรับเปลี่ยนแก้ไข แต่สามารถแก้ไขข้อมูลในคอลัมน์ TABLE_NAME ได้ โดยข้อมูลที่เก็บในตารางนี้มีดังนี้

TABLE_KEY	TABLE_NAME
ORGANIZATION_TABLE	ORGANIZATION
BUILDING_TABLE	BUILDING
FLOOR_TABLE	FLOOR
ROOMTYPE_TABLE	ROOMTYPE
ROOM_TABLE	ROOM
MAINMAP_TABLE	MAINMAP
ROOMRESERVATION_TABLE	ROOMRESERV

ตารางที่ 7-4 ข้อมูลในตาราง GISDESKTOP_CONFIG

ความหมายของแต่ละ TABLE_KEY เป็นดังตาราง

TABLE_KEY	ความหมาย
ORGANIZATION_TABLE	เก็บข้อมูลเกี่ยวกับหน่วยงาน
BUILDING_TABLE	เก็บข้อมูลเกี่ยวกับอาคาร
FLOOR_TABLE	เก็บข้อมูลเกี่ยวกับชั้นภายในอาคาร
ROOMTYPE_TABLE	เก็บข้อมูลเกี่ยวกับประเภทห้อง
ROOM_TABLE	เก็บข้อมูลเกี่ยวกับห้อง
MAINMAP_TABLE	เก็บข้อมูลของเลเยอร์ที่จะนำมาแสดงในส่วนของแผนที่หลักของสจล.
ROOMRESERVATION_TABLE	เก็บข้อมูลเวลาการจองห้อง

ตารางที่ 7-5 ความหมายของแต่ละ TABLE_KEY

หากต้องการแก้ไขเปลี่ยนแปลงจำนวนแอตทริบิวต์ หรือเปลี่ยนแปลงตารางที่ใช้เก็บข้อมูลเชิงบรรยาย ก็ทำได้โดยการใช้คำสั่ง UPDATE เพื่อเปลี่ยนค่าชื่อตารางในคอลัมน์ TABLE_NAME โดยไม่ต้องแก้ไขโค้ดของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น หากต้องการปรับเปลี่ยนแอตทริบิวต์ของตาราง ROOM ก็ทำได้การสร้างตารางใหม่ที่มีแอตทริบิวต์ตามต้องการ (แต่ต้องมีคอลัมน์ Layer_Name และ Room_Gid เป็นไพรมารี่คีย์) อาจตั้งชื่อเป็น ROOM2 แล้วทำการเปลี่ยนแปลงข้อมูลในตาราง GISDESKTOP_CONFIG โดยการเปลี่ยนค่าของ TABLE_NAME ที่อยู่ในแถวของ TABLE_KEY ROOM_TABLE ให้เป็น ROOM2 โดยใช้คำสั่ง

UPDATE GISDESKTOP_CONFIG SET TABLE_KEY='ROOM2' WHERE TABLE_NAME = ROOM_TABLE;

หากต้องการเปลี่ยนแปลงจำนวนและชนิดของเลเยอร์ที่จะนำมาแสดงในส่วนของแผนที่หลัก ทำได้โดยการเปลี่ยนแปลงข้อมูลในตาราง MAINMAP ซึ่งมีโครงสร้างตารางดังนี้

คอลัมน์	ความหมาย
LayerNo	หมายเลขประจำเลเยอร์ตั้งแต่ 1 ขึ้นไป
LayerName	ชื่อของเลเยอร์นั้น ซึ่งจะเป็นชื่อตารางที่เก็บข้อมูลของเลเยอร์นั้น
LayerInfoName	ชื่อของตารางที่เก็บข้อมูลเชิงบรรยายของเลเยอร์นั้น
Title	ชื่อเลเยอร์ที่จะถูกแสดงออกที่หน้าจอ (เป็นภาษาไทยได้)

ตารางที่ 7-6 โครงสร้างตาราง MAINMAP

ตัวอย่างเช่น หากต้องการเพิ่มเลเยอร์ที่ 4 ที่จอดรถ จะต้องทำการนำเข้าข้อมูลที่จอดรถก่อน ดังขั้นตอนการนำเข้าข้อมูลดังหัวข้อ ก.2 ก่อน จะได้ตารางซึ่งเก็บข้อมูลภาพแผนที่ของที่จอดรถชื่อเลเยอร์เป็น Car และตารางที่เก็บข้อมูลเชิงบรรยายของเลเยอร์นี้คือตาราง CarFence ถ้าต้องการให้หน้าจอแสดงชื่อเลเยอร์นี้เป็น "ที่จอดรถ" ก็ทำได้โดยการใช้คำสั่ง INSERT

INSERT INTO MAINMAP VALUES(4,'Car','CarFence','ที่จอดรถ');

สำหรับการลบเลเยอร์ที่ไม่ต้องการแสดงผลออกจากแผนที่ ก็สามารถทำได้โดยใช้คำสั่ง Delete เช่น หากไม่ต้องการให้แผนที่แสดงเลเยอร์หมุดหลักถาวร ทำได้โดยใช้คำสั่ง

DELETE MAINMAP WHERE LayerName = 'Bmp';

สำหรับการแสดงผลชื่อคอลัมน์เป็นภาษาไทย ทำโดยการใช้คำสั่ง COMMENT บนชื่อคอลัมน์นั้น เช่น ให้คอลัมน์ Computer ของตาราง ROOM แสดงผลเป็น 'จำนวนคอมพิวเตอร์' ใช้คำสั่งดังนี้

comment on column room.computer is 'จำนวนคอมพิวเตอร์';

7.8.3 การนำข้อมูลเข้า[1]

การนำข้อมูลเข้าสู่ฐานข้อมูลมีขั้นตอนดังนี้

1. นำแผนที่ซึ่งเป็นฟอร์แมตไฟล์ของโปรแกรม Autocad (DWG) มาแก้ไขเพื่อคัดเลือกข้อมูลที่ต้องการ
2. ใช้โปรแกรม FME (Feature Manipulation Engine) แปลงไฟล์ DWG ของAutoCAD ไปเป็น Shape File ของ ArcView

[1] สำหรับรายละเอียดขั้นตอนการนำเข้าข้อมูล ดูได้ที่ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ใช้โปรแกรม ArcView เปิด Shape File แก้ไขข้อมูล และเพิ่มเลเยอร์ใหม่เพื่อใช้ประโยชน์ในการสืบค้นข้อมูล
4. ใช้โปรแกรม FME แปลง Shape File ไปเก็บในฐานข้อมูลโอราเคิล

7.8.4 การเขียนโปรแกรมติดต่อกับฐานข้อมูล

ที่ฝั่งเซิร์ฟเวอร์

ใช้ภาษา PL/SQL ช่วยในการเขียนฟังก์ชันและโพรซิเดอร์ในการเรียกใช้ Spatial Cartridge

ที่ฝั่งไคล์เอนท์

ใช้ภาษาจาวาในการเขียนโปรแกรมติดต่อกับฐานข้อมูลของโอราเคิล

7.9 ซอฟต์แวร์ที่ใช้

7.9.1 ใช้ โอราเคิล 8.0.5 เป็น DBMS ของระบบ และ ใช้ Spatial cartridge ในการจัดการกับข้อมูลเชิงพื้นที่

7.9.2 AutoCAD Release 14 ใช้ดูและจัดการกับข้อมูลแผนที่สถาบันที่เป็นฟอร์แมต DWG

7.9.3 FME (Feature Manipulation Engine) ใช้แปลงข้อมูลรูปแบบมาตรฐานต่างๆ เป็นรูปแบบของ Oracle Spatial Cartridge

7.9.4 ArcView ใช้ดูข้อมูลเชิงพื้นที่และดัดแปลงแก้ไขข้อมูลของ Shape File

7.10 แนวทางการพัฒนาระบบต่อโดยมีการจัดเก็บข้อมูลในโมเดลเชิงวัตถุสัมพันธ์

เนื่องจากระบบที่ได้พัฒนาขึ้นมายังมีการจัดเก็บข้อมูลโดยใช้โมเดลเชิงสัมพันธ์อยู่ ดังนั้นสำหรับแนวทางในการพัฒนาต่อ เพื่อให้มีการจัดเก็บข้อมูลในโมเดลเชิงสัมพันธ์อาจทำได้โดย

1. ใช้โอราเคิล 8i เป็น DBMS ซึ่งมีการจัดเตรียมโครงสร้างการจัดเก็บข้อมูลเชิงวัตถุสัมพันธ์ไว้ให้แล้ว รวมทั้งมีฟังก์ชันและโพรซิเดอร์ของ Spatial Cartridge เตรียมไว้ให้สำหรับจัดการและจัดเก็บข้อมูลเชิงพื้นที่
2. รวมตารางข้อมูลเชิงบรรยายเข้ากับตารางข้อมูลเชิงพื้นที่ โดยให้คอลัมน์หนึ่งเก็บข้อมูลเชิงพื้นที่ (มีไทป์เป็นออบเจกต์ไทป์ SDO_GEOMETRY) เช่น ตาราง Kmitl_Building ให้มีโครงสร้างตารางดังนี้

คอลัมน์	ชนิดของข้อมูล	ความหมาย
BuildingID	Varchar(2)	รหัสประจำอาคารสถานที่
BuildingName	Varchar(200)	ชื่ออาคารสถานที่
OrgID	Varchar(2)	รหัสหน่วยงาน
BuildingGeometry	SDO_GEOMETRY	ข้อมูลเชิงพื้นที่ของอาคารสถานที่

ตารางที่ 7-7 โครงสร้างของตาราง Kmitl_Building โดยจัดเก็บข้อมูลในโมเดลเชิงวัตถุสัมพันธ์

สำหรับเลเยอร์อื่นๆ ก็สามารถรวมข้อมูลเชิงบรรยายกับข้อมูลเชิงพื้นที่ไว้ในตารางเดียวกันได้เช่นเดียวกับเลเยอร์ของอาคารและสถานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทสรุปและวิจารณ์

8.1 สรุปผลการทำงานโครงการ

1. การจัดเก็บข้อมูลภายในโอราเคิล ยังคงมีการจัดเก็บข้อมูลเป็นแบบตารางอยู่ แต่จะมีการขยายความสามารถที่เพิ่มขึ้น ดังนี้

- คอลัมน์หนึ่งๆ ในตาราง สามารถมีชนิดของข้อมูลตามชนิดของข้อมูลที่เรากำหนดขึ้นมาเอง ซึ่งแบ่งเป็นออบเจกต์ไทป์และ Collection Type
- ออบเจกต์ไทป์ประกอบไปด้วยแอตทริบิวต์และเมทธอด ซึ่งออบเจกต์ไทป์มีคุณสมบัติโพลิมอร์ฟิซึม และคุณสมบัติการเอ็นแคปซูเลชัน

2. โอราเคิล 8.0.5 ยังเป็น ORDBMS ที่ยังไม่สมบูรณ์ เนื่องจาก

- ออบเจกต์ไทป์ ยังขาดคุณสมบัติการสืบทอด
- การเอ็นแคปซูเลชันของออบเจกต์ไทป์ ในส่วน Specification มีเฉพาะแบบ Public เท่านั้น ยังขาดการเอ็นแคปซูเลชันแบบ Private และ Protected

3. Spatial Cartridge ที่โอราเคิล 8.0.5 มีให้ ช่วยให้การจัดการและการจัดเก็บข้อมูลเชิงพื้นที่ ทำได้ง่ายขึ้น แต่ยังคงเป็นการจัดเก็บข้อมูลในโมเดลเชิงสัมพันธ์ อยู่

4. Spatial Cartridge ของโอราเคิล 8i มีส่วนแตกต่างจากโอราเคิล 8.0.5 คือ มีการจัดเก็บข้อมูลเชิงพื้นที่ในโมเดลเชิงวัตถุสัมพันธ์ โดยมีการกำหนดโครงสร้างของการจัดเก็บรูปทรงเรขาคณิตเป็นออบเจกต์ไทป์ ซึ่งมีเฉพาะโครงสร้างเท่านั้น ไม่มีเมทธอดสำหรับการจัดการกับข้อมูลออบเจคไทป์เหล่านั้น การจัดการกับข้อมูลยังคงกระทำผ่านฟังก์ชันและโพรซิเดอร์ของ Spatial Cartridge

5. ต้นแบบระบบสารสนเทศอาคารและสถานที่ของ สจล. ที่พัฒนาขึ้นมา

- ใช้ Spatial Cartridge ของ โอราเคิล 8.0.5 ดังนั้นระบบที่เราได้พัฒนาขึ้นมาจึงยังเป็นลักษณะของการจัดเก็บในโมเดลเชิงสัมพันธ์
- สามารถใช้งานได้ 2 ทาง คือ ใช้งานเป็นแอปพลิเคชั่น และใช้งานผ่านอินเตอร์เน็ต
- สามารถดูข้อมูล และสืบค้นข้อมูลของอาคารและสถานที่ต่าง ๆ ภายในสจล.

8.2 ปัญหาที่เกิดขึ้น

ปัญหาที่เกิดขึ้นในการทำโครงงานนี้คือ

1. การพัฒนาระบบมีเวลาจำกัด เนื่องจากเริ่มต้นพัฒนาในภาคเรียนที่ 2
2. ความต้องการของผู้ใช้ระบบไม่ชัดเจน และมีการเปลี่ยนแปลง ทำให้การพัฒนาเป็นไปด้วยความยากลำบาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การพัฒนาระบบใช้เครื่องมือ Jbuilder (เครื่องมือที่ช่วยในการพัฒนาโปรแกรมด้วยภาษาจาวา) ทำให้การพัฒนาโปรแกรมใช้เวลานาน เนื่องจาก Jbuilder มีความต้องการใช้หน่วยความจำมาก ประกอบกับเครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนา มีหน่วยความจำน้อย
4. การใช้งานระบบผ่านอินเตอร์เน็ตต้องใช้กับเน็ตสเคปเวอร์ชัน 4 ขึ้นไปเท่านั้น
5. ระบบที่พัฒนาให้ใช้งานผ่านอินเตอร์เน็ตนั้น เว็บเซิร์ฟเวอร์และเคต้าเบสเซิร์ฟเวอร์ต้องอยู่ที่เครื่องเซิร์ฟเวอร์เดียวกัน เพราะติดข้อจำกัดทางด้านความปลอดภัยของแอปเพล็ต
6. มีปัญหาในการนำข้อมูลเข้า เนื่องจากข้อมูลแผนที่ที่ได้จากกองแผนยังไม่สามารถนำไปเก็บเข้าฐานข้อมูลได้โดยตรง เช่น ในเลเยอร์อาคาร เส้นที่ประกอบกันเป็นอาคาร ไม่ได้เป็นชนิดเส้นแบบพื้นที่รูปปิด แต่เกิดจากหลายๆ เส้นประกอบกัน ทำให้ต้องทำการแก้ไขข้อมูล เพื่อช่วยให้การคิวรี่ข้อมูลสะดวกขึ้น เป็นต้น ซึ่งวิธีการแปลงข้อมูลได้เสนอไว้ในภาคผนวก ก.

8.3 แนวทางการพัฒนาโครงงานวิจัยต่อ

แนวทางการพัฒนาต่อมีดังนี้คือ

1. พัฒนาระบบการจองห้องของสถาบันให้ใช้ฐานข้อมูลร่วมกันกับระบบนี้
2. ในส่วนของการใช้งานระบบผ่านอินเตอร์เน็ต เพื่อให้สามารถแยกเคต้าเบสเซิร์ฟเวอร์และเว็บเซิร์ฟเวอร์ออกจากเครื่องเดียวกัน สามารถทำได้โดยการ Sign Applet
3. พัฒนาระบบต่อโดยปรับโครงสร้างให้เป็นแบบ 3 เทียร์
4. พัฒนาระบบต่อโดยมีการจัดเก็บข้อมูลในโมเดลเชิงวัตถุสัมพันธ์

8.4 วิจารณ์

1. ทางภาควิชาวิศวกรรมคอมพิวเตอร์ ควรมีการสนับสนุนเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงให้กับโครงงานที่มีความจำเป็นต้องใช้งานให้เพียงพอ เช่น โครงงานทางด้านฐานข้อมูลที่มีความต้องการเครื่องเคต้าเบสเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง โดยเฉพาะหน่วยความจำ ดังจะเห็นได้จากโอราเคิล 8i ซึ่งต้องการหน่วยความจำขั้นต่ำสูงถึง 96 MB เป็นผลให้บางโครงงานไม่สามารถทดลองใช้ซอฟต์แวร์ตัวนี้ได้ที่ภาควิชา
2. ข้อมูลปริญญานิพนธ์ ควรมีการจัดเก็บรวบรวมและเผยแพร่ทางอินเตอร์เน็ตให้ดียิ่งขึ้น เช่น เพิ่มส่วนที่ช่วยในการค้นหาปริญญานิพนธ์ , เพิ่มข้อมูลที่จัดเก็บ ได้แก่ ไฟล์นำเสนอโครงงาน (Power Point) รูปภาพผู้ทำโครงงาน รูปภาพชิ้นงานของโครงงานหรือตัวอย่างโปรแกรม ไฟล์โปรแกรมของโครงงาน เป็นต้น
3. การทำโครงงานของนักศึกษา ทางภาควิชาควรส่งเสริมให้มีความร่วมมือการพัฒนาโครงงานกับภาครัฐและเอกชนเพื่อให้สามารถนำไปใช้งานได้ และเกิดประโยชน์อย่างแท้จริงต่อประเทศชาติและสังคมสืบไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

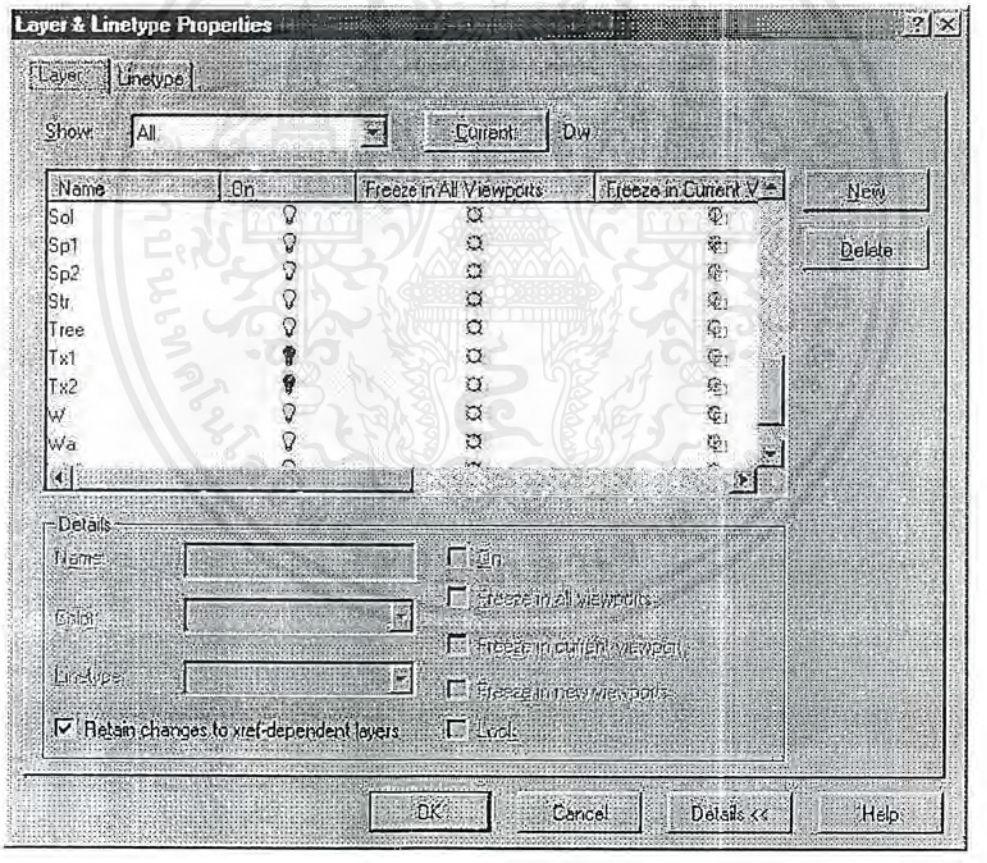
ภาคผนวก ก
การแก้ไขและการนำข้อมูลแผนที่เข้า

การแก้ไขและการนำเข้าข้อมูล แบ่งเป็น 2 ส่วน ได้แก่ การแก้ไขและนำเข้าข้อมูลแผนที่ชั้นต่างๆ ภายในอาคาร และ การแก้ไขและนำเข้าข้อมูลแผนที่สถาบัน

ก.1 ขั้นตอนการแก้ไขและนำเข้าข้อมูลแผนที่ชั้นต่างๆ ภายในอาคาร

ก.1.1 ขั้นตอนการดัดแปลงแก้ไขไฟล์ .dwg ของ AutoCAD

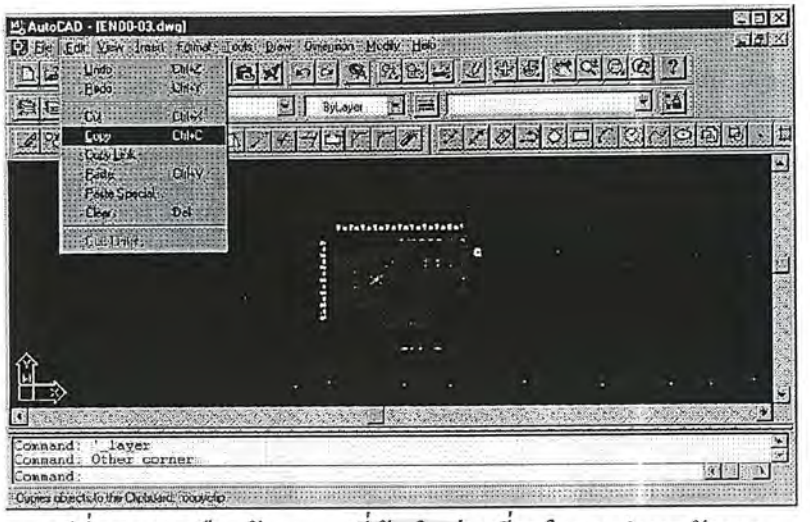
1. ใช้ AutoCAD Release14 เปิดไฟล์นามสกุล .dwg ของชั้นภายในอาคารที่ต้องการ
2. คลิ๊กที่ปุ่ม จะปรากฏหน้าต่าง Layer & Linetype Properties ขึ้นมา ดังรูป จากนั้น กำหนดให้ทุกเลเยอร์ มีสถานะ On (รูปหลอดไฟเป็นสีเหลือง) ยกเว้นเลเยอร์ชื่อ Cen, Tx1 และ Tx2 ให้มีสถานะเป็น Off (รูปหลอดไฟสีน้ำเงิน) แล้วคลิ๊กปุ่ม OK



รูปที่ ก-1 หน้าต่าง Layer & Linetype Properties

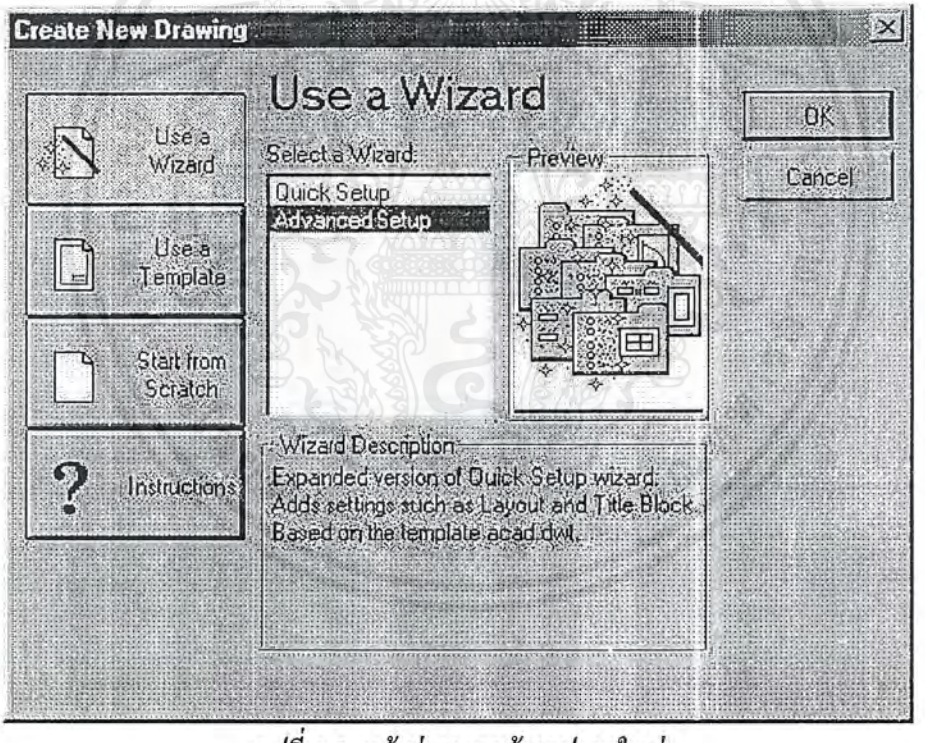
3. ล้อมกรอบรอบรูปเฉพาะส่วนที่ต้องการ แล้ว Copy (Edit >> Copy หรือ Ctrl-C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-2 การเลือกข้อมูลแผนที่ห้องในส่วนที่สนใจและทำการคัดลอก

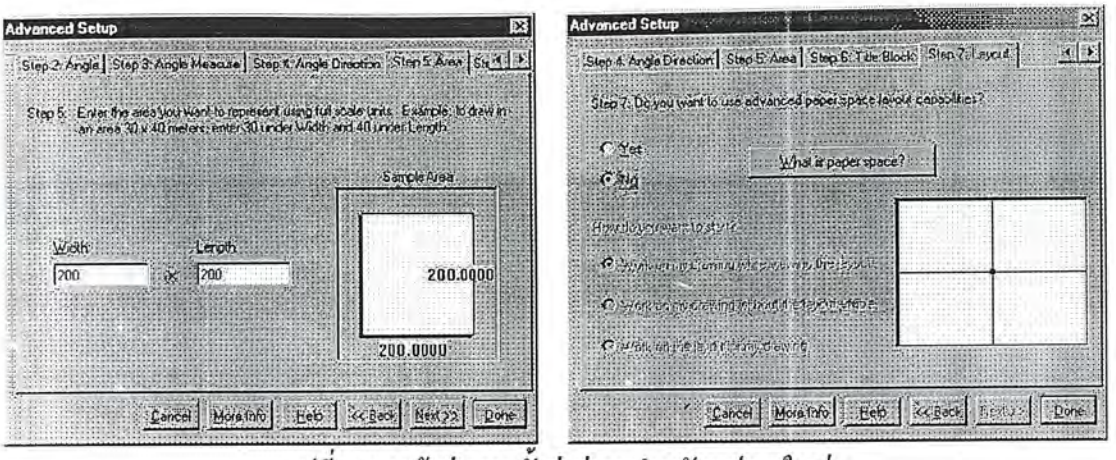
4. สร้างไฟล์ใหม่ขึ้นมา (File >> New หรือ Ctrl-N) จะปรากฏหน้าต่าง Create New Drawing ดังรูป



รูปที่ ก-3 หน้าต่างการสร้างรูปวาดใหม่

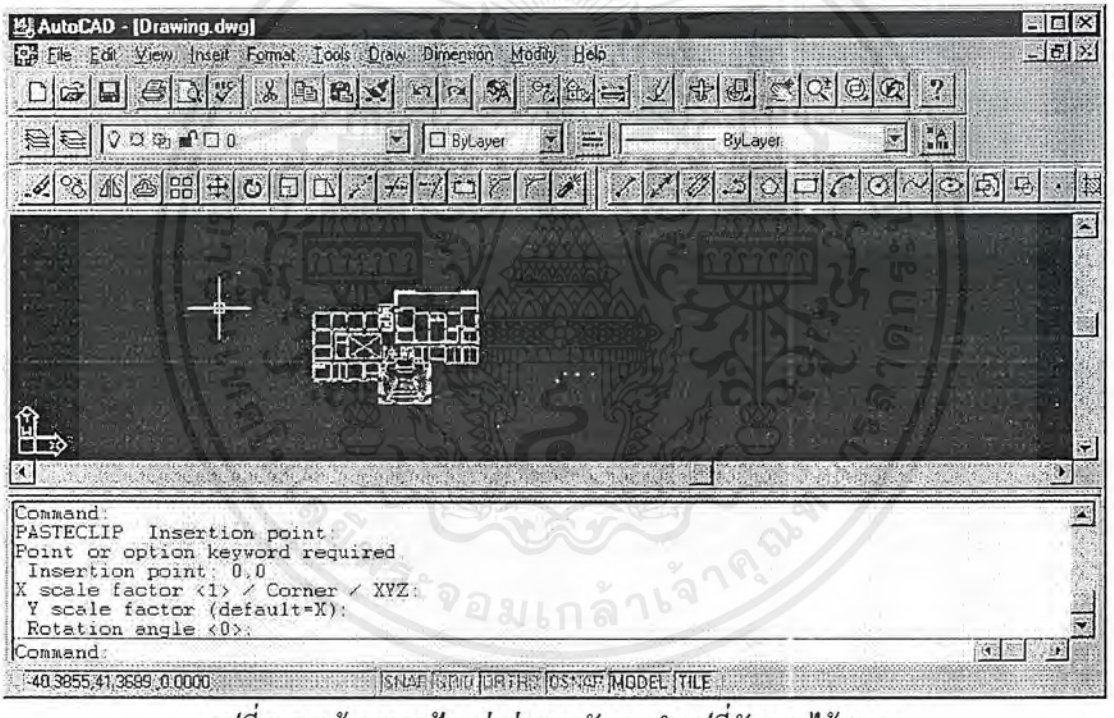
5. ทางด้านซ้ายให้เลือก Use a Wizard และในส่วนของ Select a Wizard: ให้เลือก Advanced Setup แล้วคลิ๊กปุ่ม OK จะปรากฏหน้าต่าง Advance Setup ขึ้นมาดังรูป แล้วให้เลือกไปที่ Tab Step5: Area กำหนดให้ Width เป็น 200 และ Length เป็น 200 และที่ Tab Step7: Layout ให้เลือกตอบ No แล้วคลิ๊กที่ปุ่ม Done

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-4 หน้าต่างการตั้งค่าต่างๆ สำหรับรูปวาดใหม่

6. แปะรูปที่ Copy มา (Edit >> Paste หรือ Ctrl-V) ที่ช่องด้านล่างจะขึ้นข้อความ "Insertion Point :" ให้ป้อน 0,0 แล้วกด Enter จากนั้นจะขึ้นคำถามต่อไป ให้กด Enter ผ่านไป 3 ครั้ง



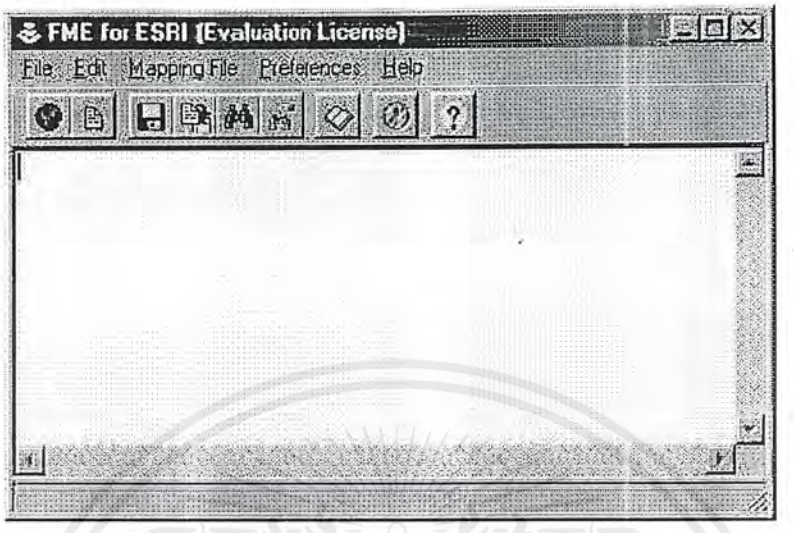
รูปที่ ก-5 หน้าจอการป้อนค่าต่างๆ หลังจากนำรูปที่คัดลอกไว้มาวาง

7. คลิ๊กเลือกที่รูปที่เพิ่ง Paste เสร็จ แล้วทำการ Explode (Modify >> Explode)
8. Save File โดยตั้งชื่อไฟล์ตามชื่อไฟล์เดิม (เปลี่ยนเครื่องหมาย – เป็น _) แต่ให้เก็บไว้ในอีก Directory หนึ่ง ซึ่งมีชื่อ Directory ที่เหมือนกับที่เก็บไฟล์ต้นฉบับก่อนการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

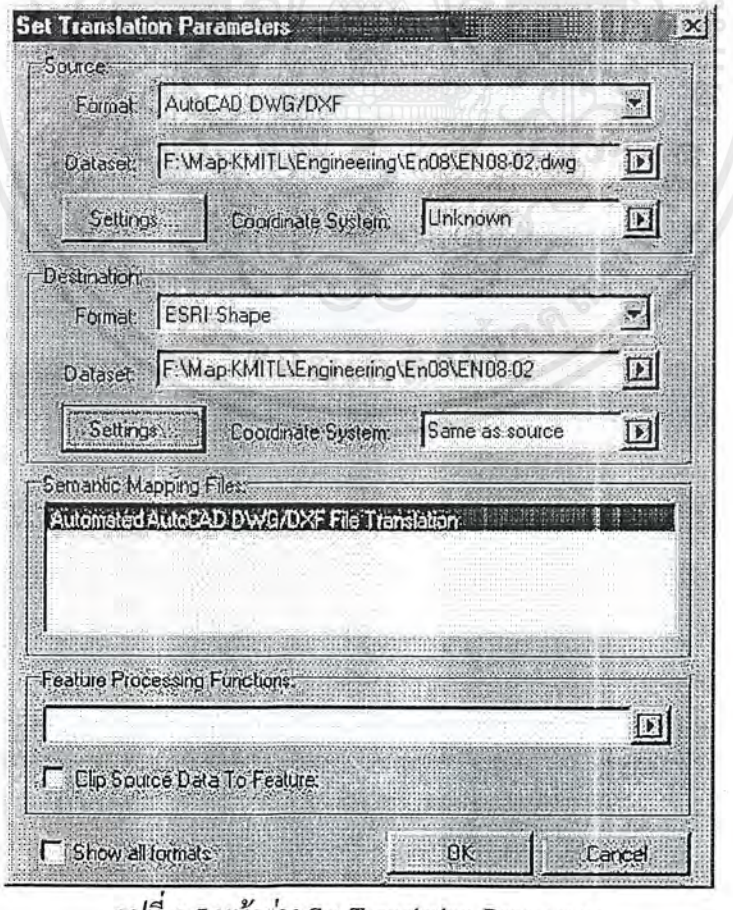
ก.1.2 ขั้นตอนการแปลงจาก .dwg ของ AutoCAD เป็น ESRI Shape File

1. เปิดโปรแกรม FME ขึ้นมา



รูปที่ ก-6 โปรแกรม FME

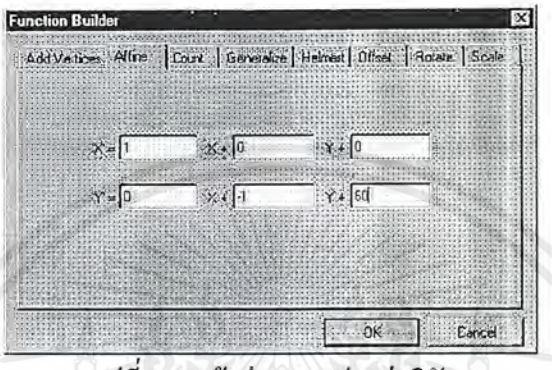
2. เลือกที่ File >> Translate... หรือกด Ctrl-T จะปรากฏหน้าต่าง Set Translation Parameters ดังรูป



รูปที่ ก-7 หน้าต่าง Set Translation Parameters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ในส่วนของ Source ที่ช่อง Format ให้เลือก AutoCAD DWG/DXF และที่ช่อง dataset ให้คลิ๊กที่ปุ่ม แล้วเลือกไฟล์ .dwg ที่ต้องการแปลง
4. ในส่วนของ Destination ที่ช่อง Format ให้เลือก ESRI Shape และที่ช่อง dataset ให้เลือก Directory ที่จะใช้ในการเก็บ Shape File โดยให้เก็บใส่ใน Directory ใหม่ โดยใช้ ชื่อ Directory เดียวกันกับชื่อของ Shape File
5. ที่ช่อง Feature Processing Functions ให้คลิ๊กที่ปุ่ม จะปรากฏหน้าต่างดังรูป

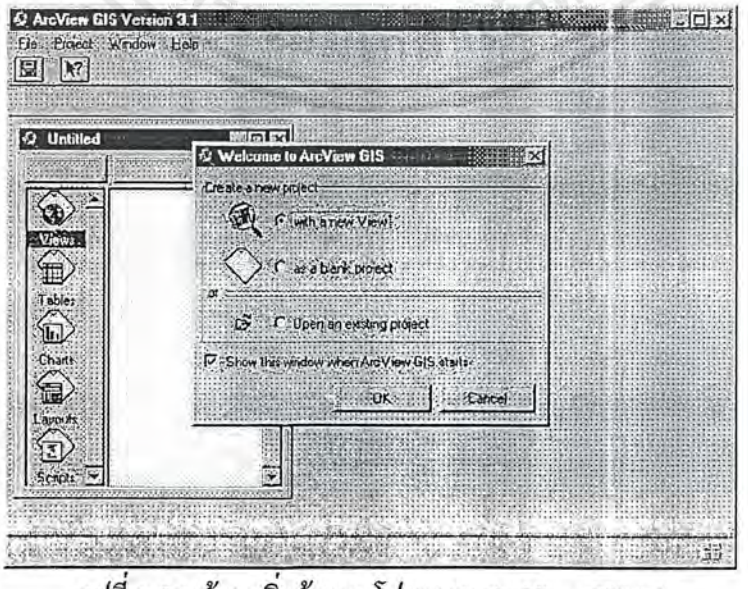


รูปที่ ก-8 หน้าต่างการแปลงค่าพิกัด

6. ป้อนตัวเลข โดยเปิดดูไฟล์ที่ต้องการจะแปลงข้อมูล (จาก dwg ของ AutoCAD) ดูว่ารูปนั้นสูงเท่าไรในแนวแกน Y แล้วนำค่านั้นมาใส่ที่ช่องสุดท้าย ของแถ็บ Affine และที่ช่องกลางให้ใส่ –1 ดังรูป
7. คลิ๊กปุ่ม OK แล้วรอดูผลลัพธ์ ถ้าสำเร็จ จะมีข้อความว่า SUCCESFUL

ก.1.3 ขั้นตอนการดัดแปลงแก้ไข Shape File

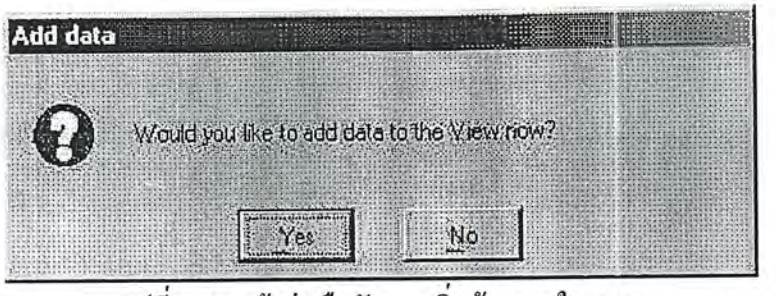
1. เปิดโปรแกรม ArcView GIS 3.1 ขึ้นมาดังรูป ให้เลือก with a new View แล้วคลิ๊กปุ่ม OK



รูปที่ ก-9 หน้าจอเริ่มต้นของโปรแกรม ArcView GIS 3.1

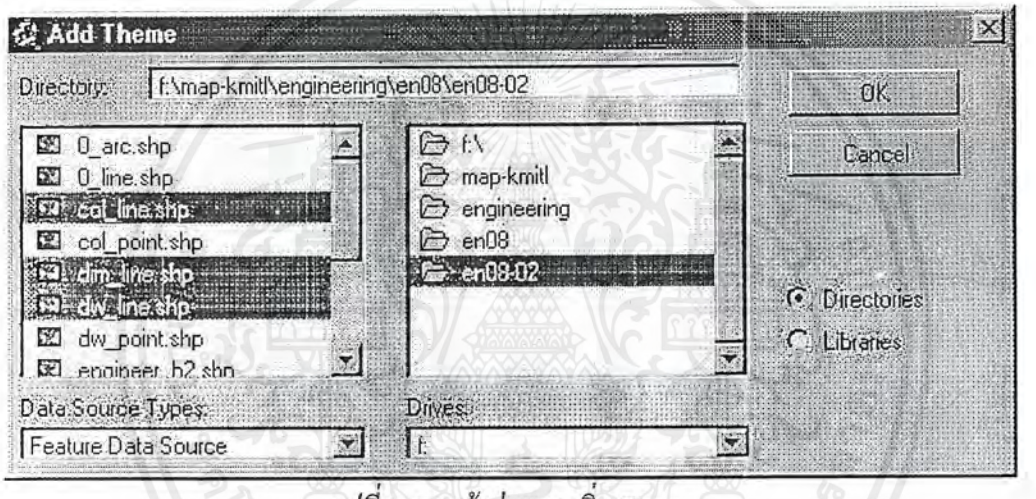
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หลังจากคลิ๊กปุ่ม OK จะปรากฏหน้าต่าง Add Data ขึ้นมาถาม ดังรูป ให้ตอบ Yes



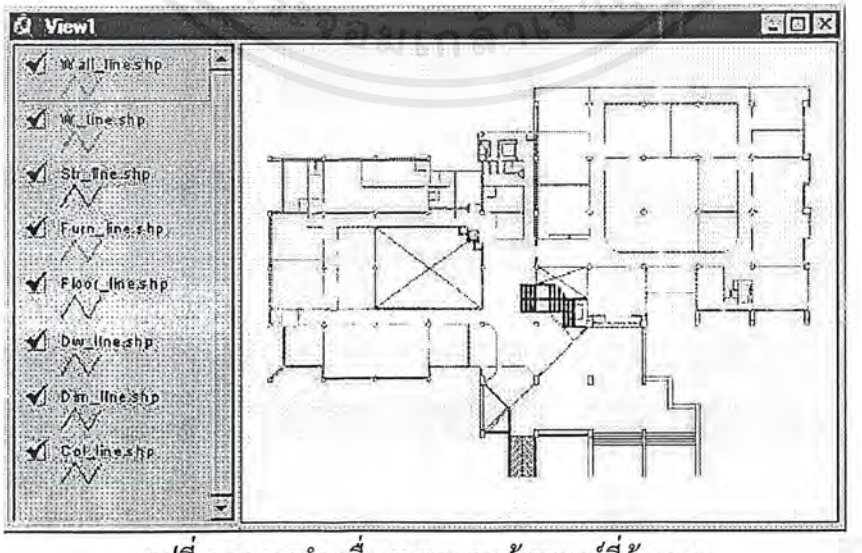
รูปที่ ก-10 หน้าต่างยืนยันการเพิ่มข้อมูลลงใน View

3. จากนั้นจะปรากฏหน้าต่าง Add Theme ขึ้นมาดังรูป ให้เลือกไฟล์ .shp ที่มีคำว่า line ทั้งหมด (กด Shift ค้างไว้ พร้อมทั้งใช้เมาส์เลือก) ยกเว้น ไฟล์ 0_line.shp แล้วคลิ๊กปุ่ม OK



รูปที่ ก-11 หน้าต่างการเพิ่ม Theme

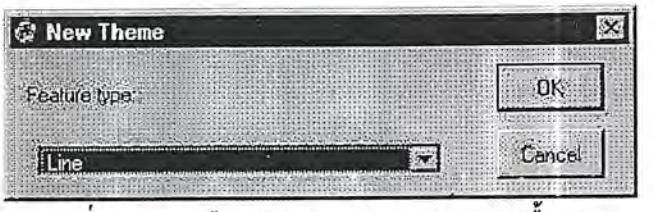
4. ทำเครื่องหมายถูก หน้าเลเยอร์ที่ต้องการทั้งหมด (ที่สามารถประกอบกันเป็นรูปร่างได้)



รูปที่ ก-12 การทำเครื่องหมายถูกหน้าเลเยอร์ที่ต้องการ

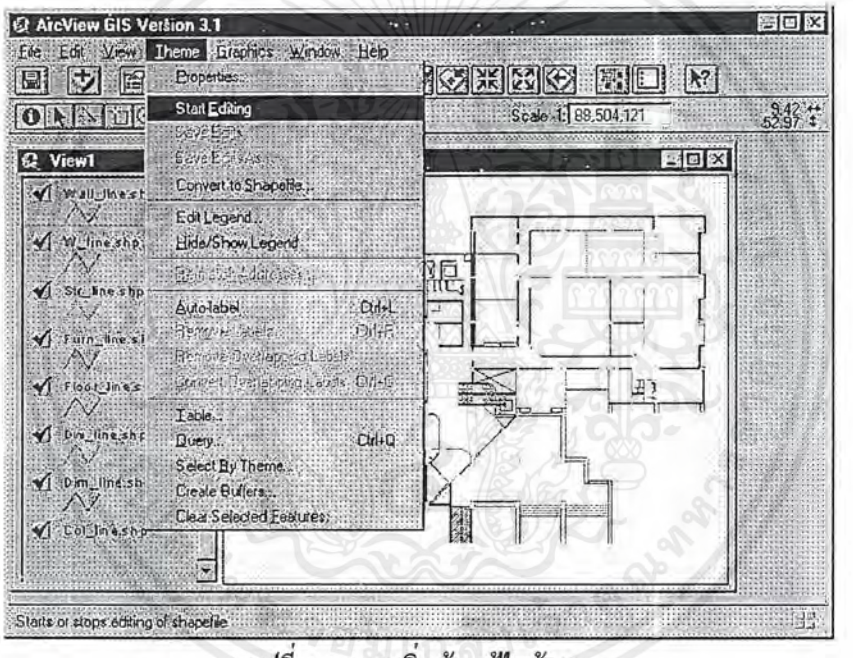
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สร้าง View ใหม่ (Double Click ที่ View หรือคลิ๊กที่ View แล้วกดปุ่ม New จะได้ View2)
6. สร้าง Theme ใหม่ขึ้นมา โดยไปที่ View >> New Theme... แล้วเลือก Feature Type เป็น line จากนั้นตั้งชื่อไฟล์ .shp โดยใช้ชื่อเดียวกับชื่อไฟล์ .dwg เก็บไว้ใน Directory เดียวกันกับที่เก็บ .shp



รูปที่ ก-13 การเลือกชนิดของรูปทรงเรขาคณิตพื้นฐาน

7. ที่ View1 คลิ๊กที่ Panel (ทางด้านซ้ายมือ) ของเลเยอร์แรกที่ถูกทำเครื่องหมายถูกแล้ว ให้นูนขึ้น
8. จากนั้นไปที่ Theme >> Start Editting...



รูปที่ ก-14 การเริ่มต้นแก้ไขข้อมูล

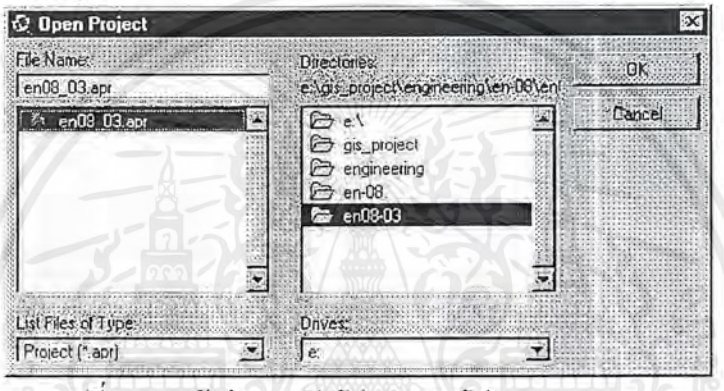
9. คลิ๊กที่ หรือ จากนั้นทำกรอบล้อมรอบรูปทั้งหมด แล้ว Copy (Edit >> Copy Features หรือ Ctrl-C)
10. ไปที่ View2 จากนั้นทำการ Paste (Edit >> Paste หรือ Ctrl-V) (ในครั้งแรกอาจมองไม่เห็นการเปลี่ยนแปลง ให้คลิ๊กที่)
11. ทำการรวมเส้นต่างๆเข้าด้วยกัน (Edit >> Union Features)
12. กลับไปที่ View1 คลิ๊กบริเวณพื้นที่ว่างเพื่อให้ส่วนที่เลือกหายไป
13. คลิ๊กที่ Panel ของเลเยอร์ถัดไปที่มีเครื่องหมายถูกให้นูนขึ้น แล้วทำซ้ำข้อ8-13 ไปเรื่อยๆ จนกว่าจะครบทุกเลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. สร้าง Theme ใหม่ (View >> New Theme...) เลือก Feature Type เป็น Polygon จากนั้นตั้งชื่อไฟล์ .shp เหมือนกับชื่อไฟล์ .dwg แต่ให้ตามด้วย _fence เช่น จากเดิมไฟล์ En08-02.dwg ก็ให้ตั้งเป็น En08_02_fence.shp
15. สร้างรูป Polygon ทับเป็นรูปห้องแต่ละห้อง โดยให้เปิดไฟล์ .dwg ของ AutoCAD ดู แล้วให้สร้างเรียงตามลำดับหมายเลข วิธีสร้างทำโดยคลิ๊กที่ แล้วจึงทำการวาด แต่ในกรณีที่รูปห้องนั้นไม่เป็นรูปสี่เหลี่ยม ให้ใช้ปุ่ม ช่วย

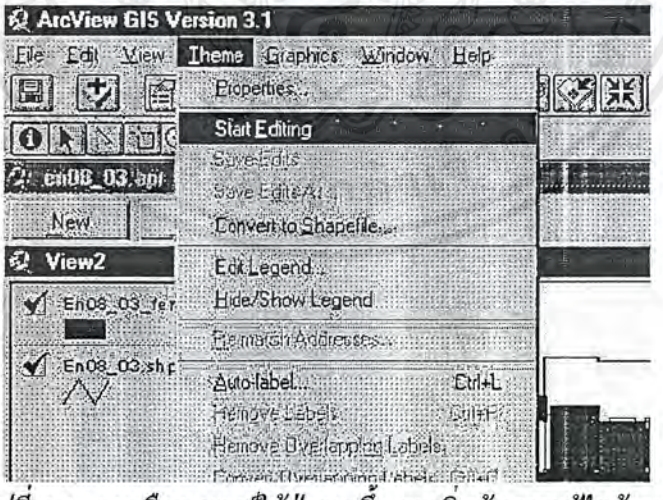
ก.1.4 ขั้นตอนการสร้างตารางเก็บข้อมูลแอตทริบิวต์

1. ใช้โปรแกรม ArcView เปิดไฟล์โปรเจ็คขึ้นมา (ไปที่ File>>Open Project แล้วเลือกไฟล์นามสกุล .apr แล้วคลิ๊กปุ่ม OK) ดังรูป



รูปที่ ก-15 หน้าต่างการเปิดโปรเจ็คของโปรแกรม ArcView

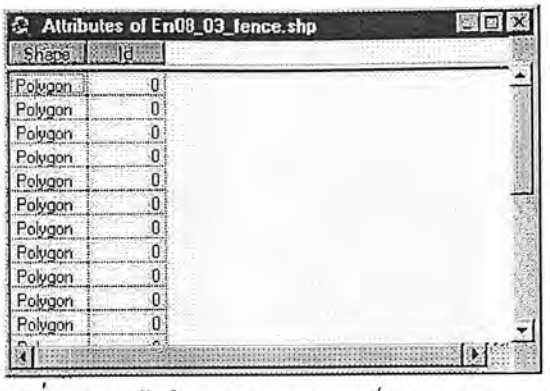
2. ที่ View2 ให้เลือก Panel ที่มีชื่อลงท้ายด้วย fence ให้นูนขึ้น แล้วไปที่ Theme>> Start Editing



รูปที่ ก-16 การเลือกเลเยอร์ให้ปุ่มนูนขึ้นและเริ่มต้นการแก้ไขข้อมูล

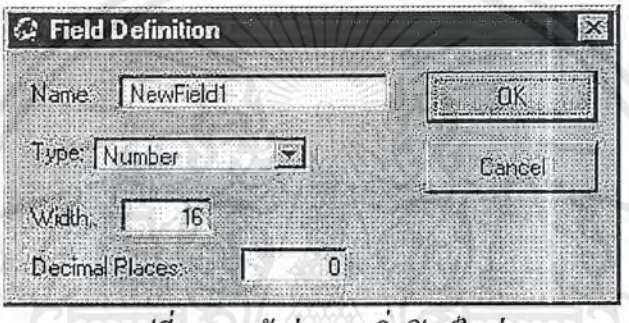
3. เรียก Table ขึ้นมาโดยไปที่ Theme>> Table... จะปรากฏหน้าต่าง Attribute ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-17 หน้าต่างตารางของเลเยอร์ En08_03_fence

4. ทำการเพิ่มฟิลด์ใหม่เข้าไป โดยไปที่ Edit>> Add Fields... จะปรากฏหน้าต่าง Field Definition ดังรูป



รูปที่ ก-18 หน้าต่างการเพิ่มฟิลด์ใหม่

5. สร้างฟิลด์ดังรายละเอียดดังต่อไปนี้

Name	Type	Width	Decimal Places
LayerName	String	200	-
Room_GID	Number	10	0
RoomTypeCo	Number	2	0
RoomName	String	200	-
RoomCo	String	6	-
StdTotal	Number	6	0
Area	Number	10	2
Computer	Number	3	0
Printer	Number	3	0
Overhead	Number	3	0
Projector	Number	3	0
WhiteBoard	Number	3	0
BlackBoard	Number	3	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

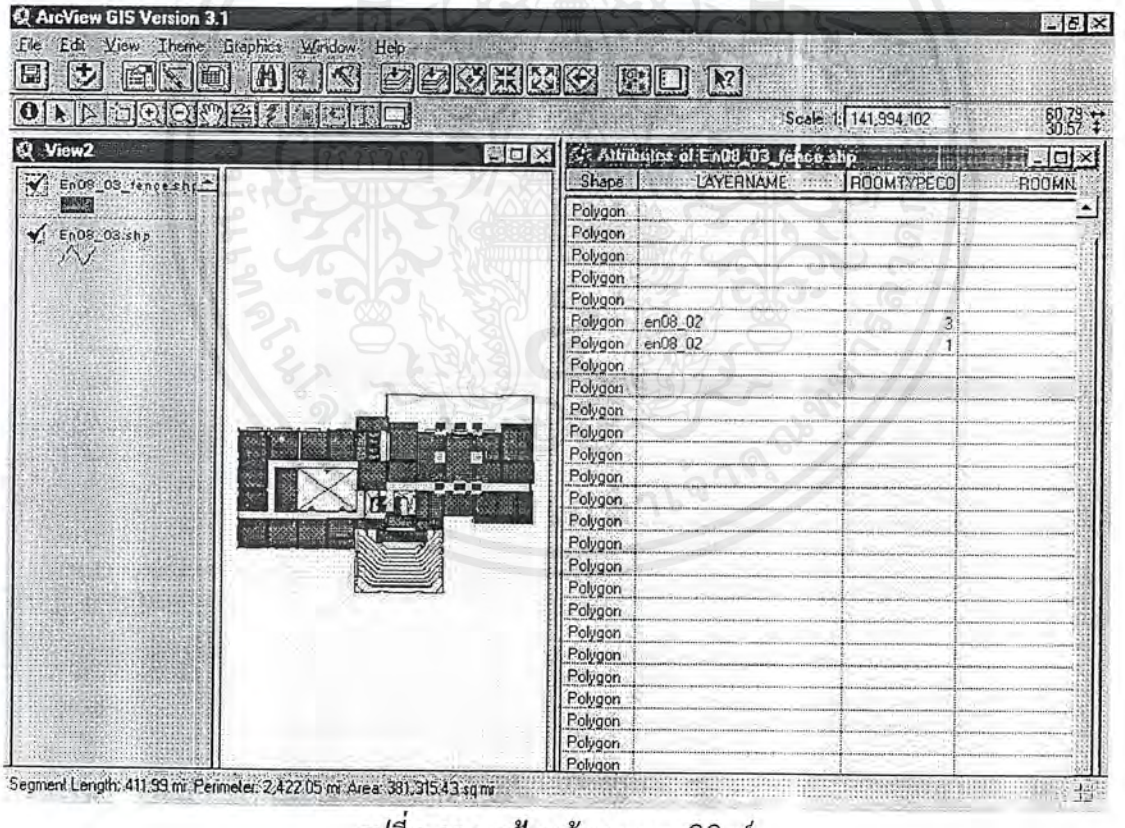
Stereo	Number	3	0
AirCdt	Number	3	0
Fan	Number	3	0

ตารางที่ ก-1 ตารางข้อมูลแอตทริบิวต์ที่ต้องป้อนสำหรับห้อง

6. ลบฟิลด์ ID ทิ้ง โดยคลิ๊กที่หัวคอลัมน์ ID ก่อน แล้วไปที่ Edit>> Delete Field จะปรากฏหน้าต่างขึ้นมาถามยืนยัน ให้ตอบ Yes (ในกรณีที่ต้องการลบฟิลด์อื่นที่สร้างผิด สามารถทำได้โดยวิธีเดียวกันนี้)
7. Save โดยคลิ๊กที่ปุ่ม แล้วตอบ Yes

ก.1.5 ขั้นตอนการป้อนข้อมูลแอตทริบิวต์

1. ทำตามขั้นตอนที่ 1-3 ของ ก.1.4 (ขั้นตอนการสร้างตารางเก็บข้อมูลแอตทริบิวต์)
2. ที่ View2 ให้คลิ๊กที่ปุ่มลูกศร แล้วคลิ๊กเลือกห้องที่ต้องการป้อนข้อมูลแอตทริบิวต์ จะปรากฏแถบสีเหลืองที่หน้าต่างสำหรับแก้ไขแอตทริบิวต์ ดังรูป



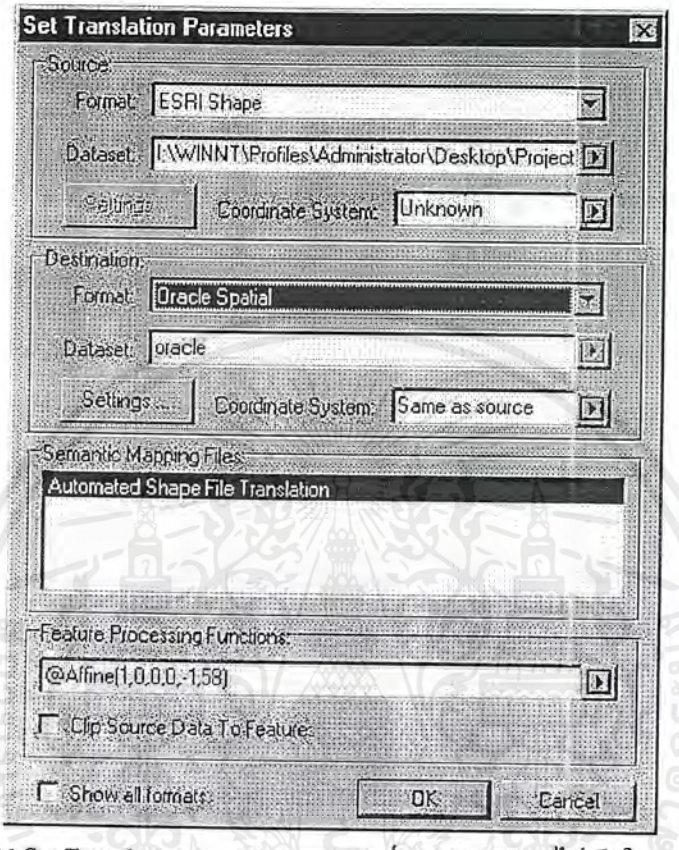
รูปที่ ก-19 การป้อนข้อมูลแอตทริบิวต์

3. ที่หน้าต่างการป้อนข้อมูลแอตทริบิวต์ คลิ๊กที่ปุ่ม แล้วคลิ๊กที่แถวที่เป็นแถบสีเหลืองเพื่อป้อนข้อมูล
4. ปฏิบัติตามข้อ 2-3 ไปเรื่อยๆ จนกระทั่งป้อนข้อมูลครบทุกห้อง แล้ว Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

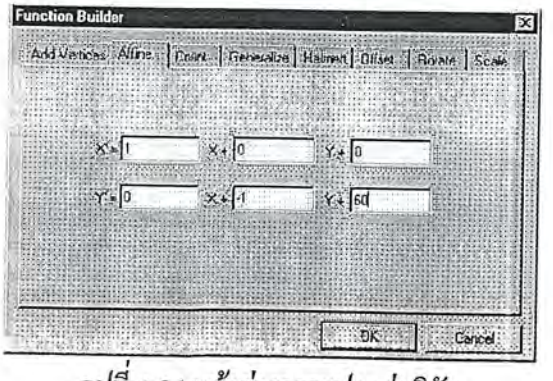
ก.1.6 ขั้นตอนการแปลง Shape File ไปเก็บในฐานข้อมูลโอราเคิล

1. เปิดโปรแกรม FME ขึ้นมาและไปที่ File>>Translate... หรือกด CTRL-T จะปรากฏหน้าต่าง Set Translation Parameters ขึ้นมา โดยให้ป้อนข้อมูลต่าง ๆ ดังขั้นตอนต่อ ๆ ไป



รูปที่ ก-20 หน้าต่าง Set Translation Parameters การแปลง Shape File ไปเก็บในฐานข้อมูลโอราเคิล

2. ในส่วนของ Source ที่ช่อง Format ให้เลือก ESRI Shape และที่ช่อง dataset ให้คลิ๊กที่ปุ่ม แล้วเลือกไฟล์ .shp ที่ต้องการแปลง เช่น en08_02.shp, en08_02_fence.shp เป็นต้น
3. ในส่วนของ Destination ที่ช่อง Format ให้เลือก Oracle Spatial (Relational)
4. ที่ช่อง Feature Processing Functions ให้คลิ๊กที่ปุ่ม จะปรากฏหน้าต่างดังรูป



รูปที่ ก-21 หน้าต่างการแปลงค่าพิกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ป้อนตัวเลข โดยเปิดดูไฟล์ที่ต้องการจะแปลงข้อมูล (จาก dwg ของ AutoCAD หรือจาก Shape File ของ ArcView) ดูว่ารูปนั้นสูงเท่าไรในแนวแกน Y แล้วนำค่านั้นมาใส่ที่ช่องสุดท้าย ของแถบ Affine และที่ช่องกลางให้ใส่ –1 ดังรูป
6. คลิ๊กปุ่ม OK แล้วรอดูผลลัพธ์ ถ้าสำเร็จ จะมีข้อความว่า SUCCESFUL
7. นำข้อมูลแอตทริบิวต์จากตาราง ที่มีชื่อต่อท้ายเป็น _fence ที่ได้ ไป INSERT ลงในตาราง ROOM โดยใช้ SQL Plus 8.0 ดังคำสั่ง SQL ต่อไปนี้

insert into room select layerName,sdo_gid,RoomTypeCo,RoomName,RoomCo,StdTotal, Area,Computer,Printer,Overhead,Projector,WhiteBoard,Blackboard,Stereo,AirCdt,Fan from <layername>_fence;

ก.2 ขั้นตอนการแก้ไขและนำเข้าข้อมูลแผนที่สถาบัน

ก.2.1 ขั้นตอนการดัดแปลงแก้ไขไฟล์ .dwg ของ AutoCAD

1. ใช้ AutoCAD Release14 เปิดไฟล์นามสกุล .dwg ของแผนที่หลักของสจล.
2. คลิ๊กปุ่ม แล้วกำหนดให้เลเยอร์ building, 16-car, railway, road, bmp, boundary มีสถานะเป็น On นอกนั้นเป็น Off
3. ทำเหมือนขั้นตอนที่ 3-5 ของการแก้ไขข้อมูลห้อง ในหัวข้อ ก.1.1 ขั้นตอนการดัดแปลงแก้ไขไฟล์ .dwg ของ AutoCAD เพียงแต่ในขั้นตอนที่ 5 ให้กำหนด Width และ Length เป็น 1500
4. ทำตามขั้นตอนที่ 6-8 ในหัวข้อ ก.1.1 แต่ในขั้นตอนที่ 8 ให้เซฟไฟล์เป็นอีกชื่อหนึ่งเก็บไว้

ก.2.2 ขั้นตอนการแปลงจาก .dwg ของ AutoCAD เป็น ESRI Shape File

ขั้นตอนนี้มีขั้นตอนการทำเหมือนกับหัวข้อ ก.1.2

ก.2.3 ขั้นตอนการดัดแปลงแก้ไข Shape File

ขั้นตอนนี้คล้ายกับขั้นตอนในหัวข้อ ก.1.3 แต่ในขั้นตอนที่ 3 ให้เลือกShape File ที่เหมาะสมจากของแต่ละเลเยอร์ แล้วทำการแก้ไขแต่ละเลเยอร์ดังนี้

ก.2.3.1 เลเยอร์อาคาร

ให้สร้างเลเยอร์ใหม่ขึ้นมาโดยไปที่ View>>New Theme... เลือก Feature Type เป็น Polygon ตั้งชื่อไฟล์ แล้วทำการเริ่มต้นการแก้ไขข้อมูลของเลเยอร์ใหม่นี้ โดยวาดรูปพื้นที่รูปปิดทับรูปของอาคารสถานที่ทั้งหมดที่สนใจ

ก.2.3.2 เลเยอร์หมุดหลักฉาวร

สำหรับเลเยอร์นี้ ข้อมูลของ Shape File จะเป็นจุดและตัวหนังสือ ให้สร้างเลเยอร์ใหม่ที่มีรูปทรงพื้นฐานเป็นพื้นที่รูปปิด แล้ววาดทับจุดเป็นวงกลม

ก.2.3.2 เลเยอร์ขอบเขตสถาบัน

ลักษณะเดียวกันกับเลเยอร์อาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 การเพิ่มหรือลดเลเยอร์ที่แสดงในแผนที่

หากต้องการเปลี่ยนแปลงจำนวนและชนิดของเลเยอร์ที่จะนำมาแสดงในส่วนของแผนที่หลัก ทำได้โดยการเปลี่ยนแปลงข้อมูลในตาราง MAINMAP ซึ่งมีโครงสร้างตารางดังนี้

คอลัมน์	ความหมาย
LayerNo	หมายเลขประจำเลเยอร์ตั้งแต่ 1 ขึ้นไป
LayerName	ชื่อของเลเยอร์นั้น ซึ่งจะเป็นชื่อตารางที่เก็บข้อมูลของเลเยอร์นั้น
LayerInfoName	ชื่อของตารางที่เก็บข้อมูลเชิงบรรยายของเลเยอร์นั้น
Title	ชื่อเลเยอร์ที่จะถูกแสดงออกที่หน้าจอ (เป็นภาษาไทยได้)

ตารางที่ ก-2 โครงสร้างตาราง MAINMAP

ตัวอย่างเช่น หากต้องการเพิ่มเลเยอร์ที่ 4 ที่จอดรถ จะต้องทำการนำเข้าข้อมูลที่จอดรถก่อน ดังขั้นตอนการนำเข้าข้อมูลดังหัวข้อ ก.2 ก่อน จะได้ตารางซึ่งเก็บข้อมูลภาพแผนที่ของที่จอดรถชื่อเลเยอร์เป็น Car และตารางที่เก็บข้อมูลเชิงบรรยายของเลเยอร์นี้คือตาราง CarFence ถ้าต้องการให้หน้าจอแสดงชื่อเลเยอร์นี้เป็น "ที่จอดรถ" ก็ทำได้โดยการใช้คำสั่ง INSERT

```
INSERT INTO MAINMAP VALUES(4,'Car','CarFence','ที่จอดรถ');
```

สำหรับการลบเลเยอร์ที่ไม่ต้องการแสดงผลออกจากแผนที่ ก็สามารถทำได้โดยใช้คำสั่ง Delete เช่น หากไม่ต้องการให้แผนที่แสดงเลเยอร์หมุดหลักถาวร ทำได้โดยใช้คำสั่ง

```
DELETE MAINMAP WHERE LayerName = 'Bmp';
```

สำหรับการแสดงผลชื่อคอลัมน์เป็นภาษาไทย ทำโดยการใช้คำสั่ง COMMENT บนชื่อคอลัมน์นั้น เช่น ให้คอลัมน์ Computer ของตาราง ROOM แสดงผลเป็น 'จำนวนคอมพิวเตอร์' ใช้คำสั่งดังนี้

```
comment on column room.computer is 'จำนวนคอมพิวเตอร์';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
คู่มือการติดตั้งและใช้งานโปรแกรม

ข.1 การติดตั้งโปรแกรม

การติดตั้งโปรแกรม จะแบ่งออกเป็น การติดตั้งสำหรับใช้งานเป็นแอปพลิเคชั่น และการติดตั้งสำหรับใช้งานผ่านอินเตอร์เน็ต ซึ่งทั้ง 2 ประเภทนี้จำเป็นจะต้องมีการติดตั้งโอราเคิล 8.0.5 พร้อมทั้ง Spatial Cartridge ที่เครื่องเซิร์ฟเวอร์ก่อน แล้วสร้างฐานข้อมูลเพื่อเก็บข้อมูลแผนที่ สจล. โดยการใช้ SQL Plus 8.0 รันสคริปต์ Setup.txt

ข.1.1 ความต้องการของระบบด้านซอฟต์แวร์

- สำหรับการใช้งานผ่านเว็บบราวเซอร์ ต้องใช้เน็ตสเคปเวอร์ชัน 4.0 ขึ้นไป

ข.1.2 ความต้องการของระบบด้านฮาร์ดแวร์

- จอแสดงผลที่สามารถแสดงผลได้ด้วยความละเอียดอย่างต่ำ 640x800 พิกเซล
- ซีพียู 200 เมกะเฮิร์ต หรือสูงกว่า (แนะนำ 350 เมกะเฮิร์ตขึ้นไป)
- แรม 64 เมกะไบต์ หรือสูงกว่า
- มีเมาส์และคีย์บอร์ดในการติดต่อกับระบบ

ข.1.3 การติดตั้งสำหรับใช้งานเป็นแอปพลิเคชั่น

- Unzip ไฟล์ GISDESKTOP1.1.8.zip ออกมาไว้ที่ไดเร็คทอรี่หนึ่ง
- ไปที่ไดเร็คทอรี่ที่ Unzip ไฟล์ออกมา แล้วรันไฟล์ Setup.exe จะปรากฏหน้าต่างขึ้นมา แล้วให้คลิ๊กที่ปุ่ม Setup

ข.1.4 การติดตั้งสำหรับใช้งานผ่านเว็บบราวเซอร์

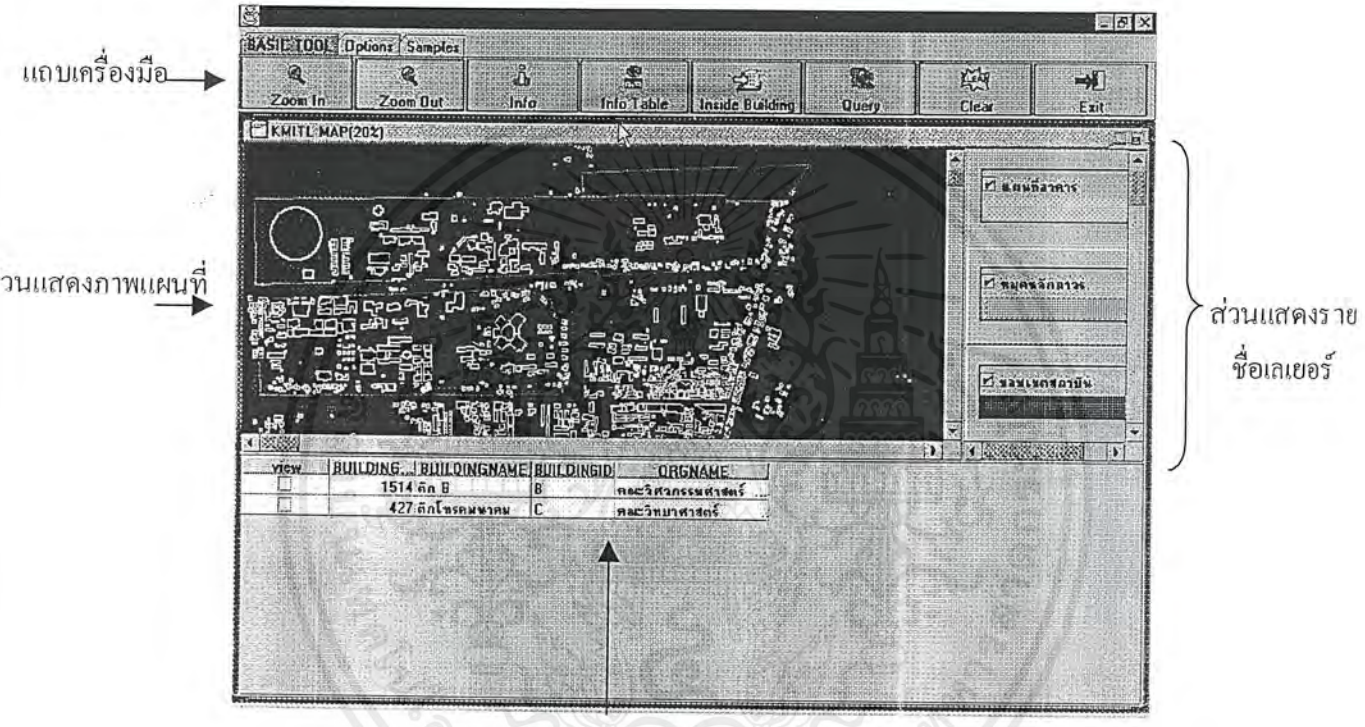
- **เครื่องเซิร์ฟเวอร์**
  - ติดตั้งเว็บเซิร์ฟเวอร์
  - Unzip ไฟล์ GISSERVER ไว้ที่ไดเรคทอรี่ ซึ่งถูกกำหนดให้เป็นที่เก็บไฟล์ของเว็บเซิร์ฟเวอร์
- **เครื่องไคล์เอนท์**
  - ติดตั้งโปรแกรม เน็ตสเคปเวอร์ชัน 4 ขึ้นไป โดยรันไฟล์ที่ใช้ในการติดตั้ง แล้วปฏิบัติตามขั้นตอนที่กำหนดไว้
  - หากมีปัญหาการแสดงผลภาษาไทย ให้ทำการสำรองไฟล์ font.properties ใน Program Files\Netscape\Communicator\Program\Java\Classes\ เอาไว้ จากนั้นคัดลอกไฟล์ font.properties ที่ได้จัดเตรียมไว้ให้มาแทนของเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 ส่วนประกอบของโปรแกรม

โปรแกรมประกอบไปด้วยส่วนต่าง ๆ ดังนี้

1. แถบเครื่องมือ
2. ส่วนแสดงรายชื่อเลเยอร์
3. ส่วนแสดงภาพแผนที่
4. ส่วนแก้ไขข้อมูล
5. ส่วนแสดงข้อมูลแอตทริบิวต์

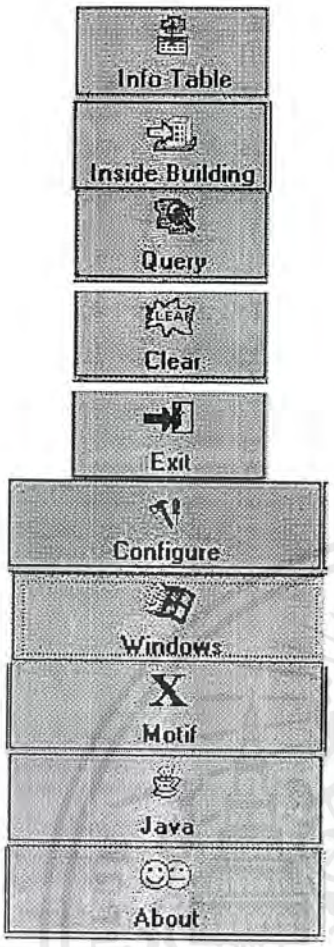


รูปที่ ข-1 ส่วนประกอบของโปรแกรม

ข.2.1 แถบเครื่องมือ ประกอบไปด้วยแถบเครื่องมือ 2 แถบ ได้แก่ Basic Tool และ Option

ปุ่มเครื่องมือ	การใช้งาน
Zoom In	ขยายแผนที่
Zoom Out	ย่อแผนที่
Info	แสดงข้อมูลเชิงบรรยายของวัตถุที่ถูกเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ดูตารางข้อมูลเชิงบรรยายของเลเยอร์ที่ต้องการ

เข้าไปภายในตัวอาคาร

สืบค้นหาห้องที่ต้องการโดยกำหนดเงื่อนไข

ลบเครื่องหมายถูกในคอลัมน์ View ของตารางข้อมูลเชิงบรรยาย และลบภาพที่ถูกระบายสี

ออกจากโปรแกรม

ตั้งค่าให้แก่ระบบ

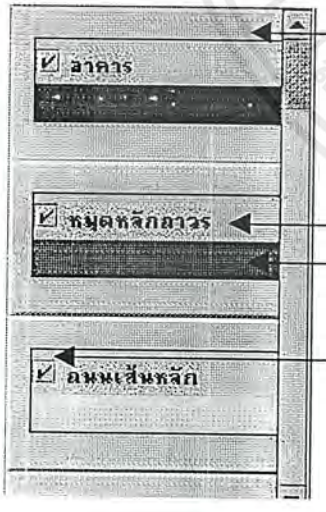
เปลี่ยนหน้าต่างเป็นแบบ Windows

เปลี่ยนหน้าต่างเป็นแบบ Motif

เปลี่ยนหน้าต่างเป็นแบบ Java

เกี่ยวกับโปรแกรมและผู้พัฒนาโปรแกรม

ข.2.2 ส่วนแสดงรายการเลเยอร์



เมื่อถูกคลิ๊กเป็นกรอบสีเหลืองจะแสดงตารางแอตทริบิวต์ของเลเยอร์นั้น

ชื่อเลเยอร์

สีที่แสดงวัตถุในเลเยอร์

กำหนดให้มีการแสดงภาพแผนที่ของเลเยอร์นั้นเมื่อมีเครื่องหมายถูก

รูปที่ ข-2 ส่วนแสดงรายการเลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3 การใช้งานโปรแกรม

ข.3.1 การเข้าโปรแกรม

- ในกรณีของแอปเพล็ต การเข้าโปรแกรม ให้เปิดโปรแกรมเน็ตสเคป แล้วพิมพ์ URL 161.246.10.36/GisDesktop.html
- ในกรณีของแอปพลิเคชั่น การเข้าโปรแกรมทำได้โดยการรันไฟล์ GISDESKTOP.bat

ข.3.2 การออกจากโปรแกรม

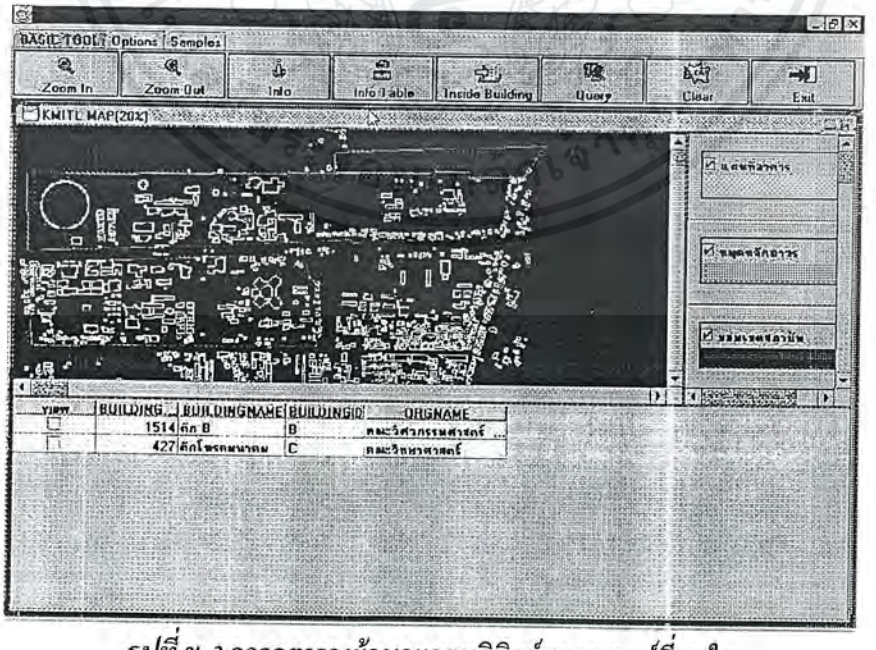
การออกจากโปรแกรม หากเป็นแอปพลิเคชั่นทำโดยการคลิ๊กที่ปุ่มกากบาทที่มุมขวาบนของโปรแกรม หรือ คลิ๊กที่ปุ่ม Exit แต่หากเป็นการใช้งานผ่านบราวเซอร์ ให้คลิ๊กที่ปุ่มกากบาทปิดที่มุมบนขวาของบราวเซอร์

ข.3.3 การย่อ/ขยายแผนที่

- การย่อแผนที่ทำโดยคลิ๊กที่ปุ่ม Zoom Out ขนาดของแผนที่จะลดลง ทำให้มองเห็นแผนที่ได้ครอบคลุมพื้นที่มากขึ้น
- การขยายแผนที่ทำโดยคลิ๊กที่ปุ่ม Zoom In ขนาดของแผนที่จะขยายใหญ่ขึ้น ทำให้ผู้ใช้เห็นรายละเอียดชัดเจนขึ้น

ข.3.4 การดูตารางข้อมูลแอตทริบิวต์ของเลเยอร์ที่สนใจ

ทำโดยการคลิ๊กที่บริเวณกรอบสี่เหลี่ยมของเลเยอร์ที่สนใจ (ซึ่งอยู่ทางด้านขวาของหน้าจอ) ให้มีกรอบสี่เหลี่ยมสีเหลืองล้อมรอบ จะปรากฏตารางข้อมูลแอตทริบิวต์ของเลเยอร์นั้นบริเวณส่วนล่างของหน้าจอ ดังรูป

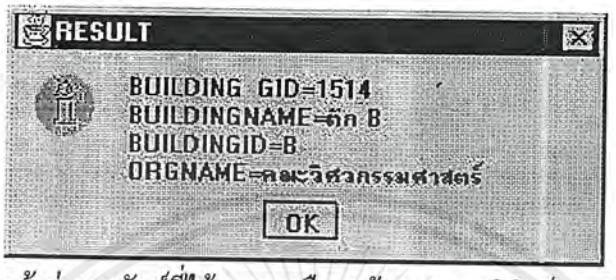


รูปที่ ข-3 การดูตารางข้อมูลแอตทริบิวต์ของเลเยอร์ที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3.5 การเลือกดูข้อมูลแอตทริบิวต์ของวัตถุที่สนใจ

การเลือกดูข้อมูลแอตทริบิวต์เฉพาะของวัตถุที่สนใจ ทำโดยคลิ๊กเลือกที่กรอบของเลเยอร์ที่ต้องการให้มีกรอบสีเหลืองล้อมรอบ จากนั้นคลิ๊กที่ปุ่ม Info แล้วไปคลิ๊กที่วัตถุในแผนที่ที่ต้องการทราบข้อมูล ซึ่งจะปรากฏหน้าต่างแสดงข้อมูลแอตทริบิวต์ของเลเยอร์นั้นออกมาดังรูป



รูปที่ ข-4 หน้าต่างผลลัพธ์ที่ได้จากการเลือกดูข้อมูลแอตทริบิวต์ของวัตถุที่สนใจ

ข.3.6 การเลือกแสดงเฉพาะเลเยอร์ที่สนใจ

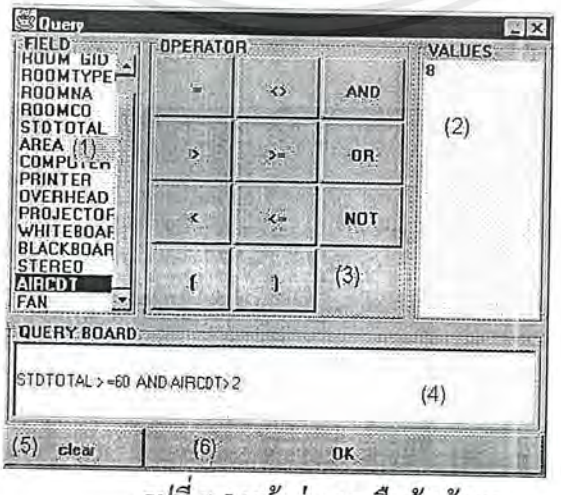
สามารถเลือกแสดงผลเฉพาะเลเยอร์ที่สนใจ และปิดเลเยอร์ที่ไม่สนใจได้โดยการคลิ๊กเครื่องหมายถูกหน้าเลเยอร์ที่จะให้แสดงผล และเอาเครื่องหมายถูกออกจากหน้าเลเยอร์ที่ไม่ต้องการให้แสดงผล

ข.3.7 การเข้าไปภายในอาคาร

การเข้าอาคารทำโดยการคลิ๊กที่ปุ่ม Inside Building จากนั้นให้คลิ๊กเลือกอาคารที่ผู้ใช้ต้องการที่จะเข้าไปดูรายละเอียดของชั้นต่างๆ ภายในตัวอาคาร หน้าต่างแผนที่ชั้นต่างๆ ภายในอาคารจะถูกแสดงออกมา

ข.3.8 การสืบค้นข้อมูลแบบกำหนดเงื่อนไข

ทำการคลิ๊กที่ปุ่ม Query จะปรากฏหน้าต่างให้ทำการกำหนดเงื่อนไขขึ้นมา



รูปที่ ข-5 หน้าต่างการสืบค้นข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบของหน้าต่างการสืบค้นข้อมูล

1.) เป็นส่วนแสดงรายการของฟิลด์ต่าง ๆ

2.) เป็นส่วนแสดงค่าต่างๆ ที่ถูกเก็บไว้ในฐานข้อมูล เมื่อชื่อฟิลด์ในส่วนที่ (1) ถูกเลือก

3.) เป็นโอเปอเรเตอร์ที่ใช้ในการสร้างเงื่อนไขเปรียบเทียบ

4.) เป็นช่องสำหรับใส่เงื่อนไขในการเปรียบเทียบ โดยผู้ใช้อาจพิมพ์เอง หรือสร้างขึ้นมาจากการคลิ๊กหรือดับเบิ้ลคลิ๊กจาก 3 ส่วนข้างต้น (ในส่วนที่ (1) และ (2)) ต้องกดดับเบิ้ลคลิ๊ก และในส่วนที่ (3) กดคลิ๊ก

5.) ลบเงื่อนไขที่กำหนดในส่วนที่ (4) ออกทั้งหมด เพื่อป้อนเงื่อนไขในการค้นหาใหม่

6.) เพื่อเริ่มต้นในการค้นหาข้อมูลตามเงื่อนไขที่มีอยู่ในส่วนที่ (4)

ตัวอย่าง การค้นหาห้องที่ต้องการ สมมติว่าขณะนี้อยู่ที่หน้าต่างแผนที่หลักของสถาบัน ให้คลิ๊กที่ปุ่ม Query จะปรากฏหน้าต่างการสืบค้นข้อมูลออกมาดังรูปที่ ข-3

หากต้องการค้นหาว่ามีห้องไหนบ้างที่จุคนได้อย่างน้อย 60 คน และมีเครื่องปรับอากาศมากกว่า 2 เครื่อง สามารถค้นหาได้โดยการกำหนดเงื่อนไขในช่องส่วนที่ (4) เป็น

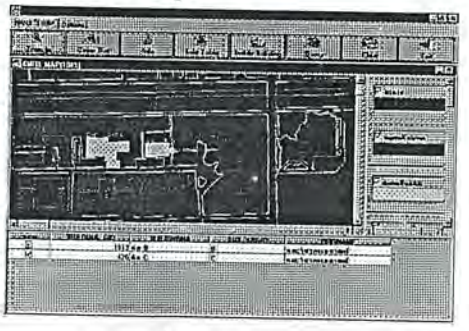
STDTOTAL >= 60 AND AIRCDT > 2

การกำหนดเงื่อนไขดังกล่าวนี้สามารถทำได้ 2 วิธี คือ

1. ป้อนเงื่อนไขดังกล่าวลงไปโดยตรง โดยใช้เมาส์คลิ๊กในส่วนที่ (4) แล้วพิมพ์ลงไป
2. สร้างจากการคลิ๊กและดับเบิ้ลคลิ๊ก

- ดับเบิ้ลคลิ๊กที่ฟิลด์ STDTOTAL ในส่วนที่ (1) จะปรากฏข้อมูลจำนวนคนที่จุได้ที่เก็บอยู่ในฐานข้อมูลในส่วนที่ (2) แล้วคลิ๊กที่โอเปอเรเตอร์ >= จากนั้นดับเบิ้ลคลิ๊ก 60 ที่ส่วนที่ (2) หากไม่มีเลขนี้ก็ป้อนเองโดยตรง
- คลิ๊กที่โอเปอร์เรเตอร์ AND แล้วดับเบิ้ลคลิ๊กที่ AIRCDT จากนั้นคลิ๊กที่ > แล้วดับเบิ้ลคลิ๊กที่เลข 2 ในส่วนที่ (2) แต่หากไม่มีเลขนี้อยู่ก็ให้ป้อนเองโดยตรง

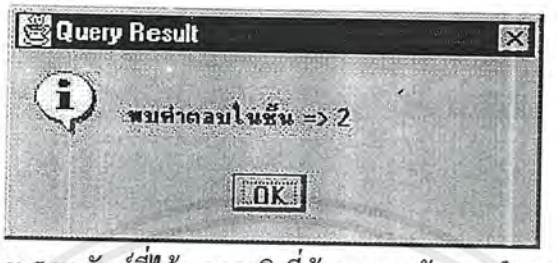
เมื่อได้เงื่อนไขที่ต้องการในส่วนที่ (4) แล้ว คลิ๊กที่ปุ่ม OK เพื่อทำการค้นหา หากค้นหาไม่พบหรือเงื่อนไขในการค้นหาผิดพลาดก็จะมีหน้าต่างปรากฏขึ้นมาบอก แต่หากค้นหาพบ จะปรากฏสีระบายอาคารทุกอาคารที่มีห้องตามเงื่อนไขดังกล่าว และที่ช่องตารางด้านล่างในคอลัมน์ View จะเป็นปรากฏเครื่องหมายถูกหน้าแถวของอาคารที่มีห้องที่ค้นหาดังรูป



รูปที่ ข-6 ผลลัพธ์จากการคิวรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

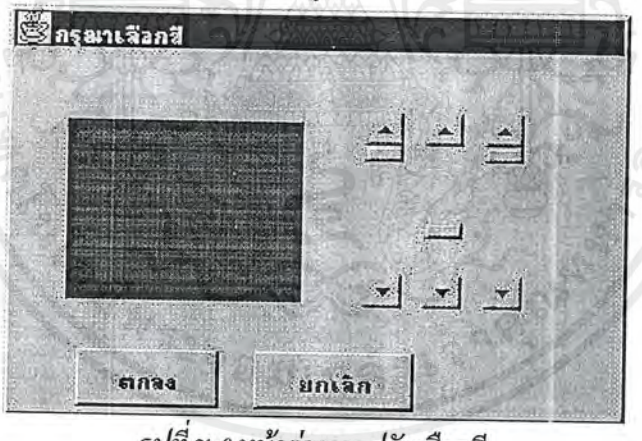
หากต้องการค้นหาห้องภายในอาคารใด ก็ให้เข้าอาคารที่สนใจโดยการคลิ๊กที่ปุ่ม Inside Building แล้วไปคลิ๊กตัวอาคารที่ต้องการ จะปรากฏแผนที่ชั้นต่างๆ ภายในอาคารนั้นขึ้นมา จากนั้นคลิ๊กที่ปุ่ม Query อีกแล้วทำการป้อนเงื่อนไขในส่วนที่ (4) เช่นเดียวกับในส่วนของแผนที่หลักของสถาบัน แต่ผลลัพธ์ที่ได้หากค้นหาพบ จะปรากฏหน้าต่างแจ้งว่าพบในชั้นใดบ้าง ดังรูป เมื่อผู้ใช้ต้องการจะดูว่าห้องนั้นอยู่ที่ใดบ้าง ก็ให้คลิ๊กทำเครื่องหมายถูก หน้าเลเยอร์ของชั้นที่ต้องการ ห้องที่ค้นหาพบจะถูกระบายสีไว้



รูปที่ ข-7 ผลลัพธ์ที่ได้จากการคิวรี่ข้อมูลของห้องภายในอาคาร

ข.3.9 การเปลี่ยนสีการแสดงผลของเลเยอร์

สามารถเปลี่ยนสีการแสดงผลของเลเยอร์ที่ต้องการได้โดยการคลิ๊ก ที่แถบสีของเลเยอร์ที่ต้องการเปลี่ยนสี จะปรากฏหน้าต่างให้เลือกปรับสีตามต้องการ โดยการเลื่อน Scroll Bar ทั้ง 3 อันแล้วกดปุ่ม ตกลง ดังรูป



รูปที่ ข-8 หน้าต่างการปรับเลือกสี

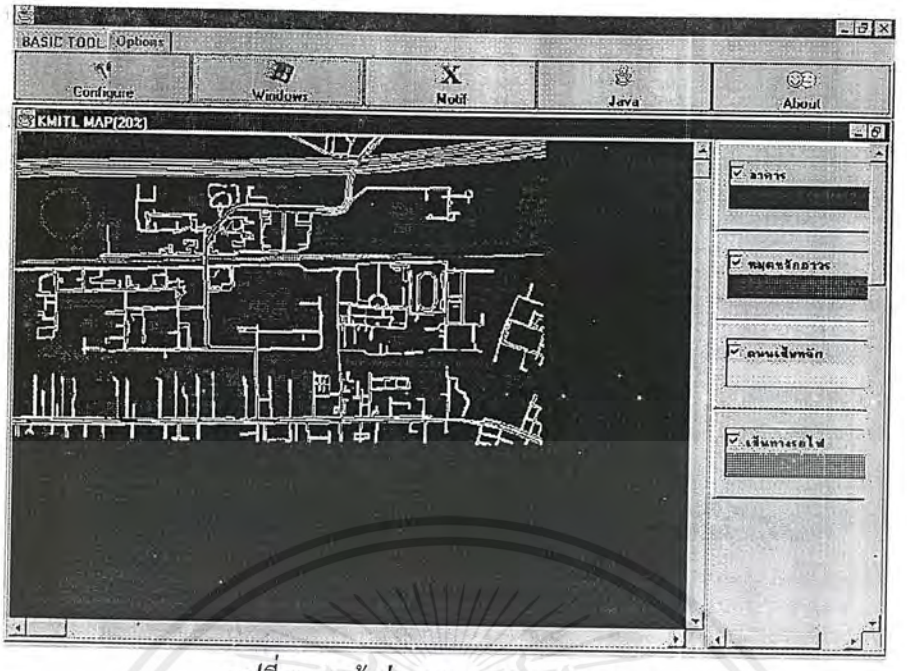
ข.3.10 การเปลี่ยนรูปแบบของหน้าต่างแสดงผล

สามารถเปลี่ยนรูปแบบของหน้าต่างการแสดงผลได้ 3 แบบ ได้แก่

ข.3.10.1 หน้าต่างแสดงผลแบบ Windows

เป็นรูปแบบหน้าต่างของระบบ Window สามารถเปลี่ยนเป็นรูปแบบนี้ได้โดยการคลิ๊กที่ปุ่ม Windows ในแถ็บ Option

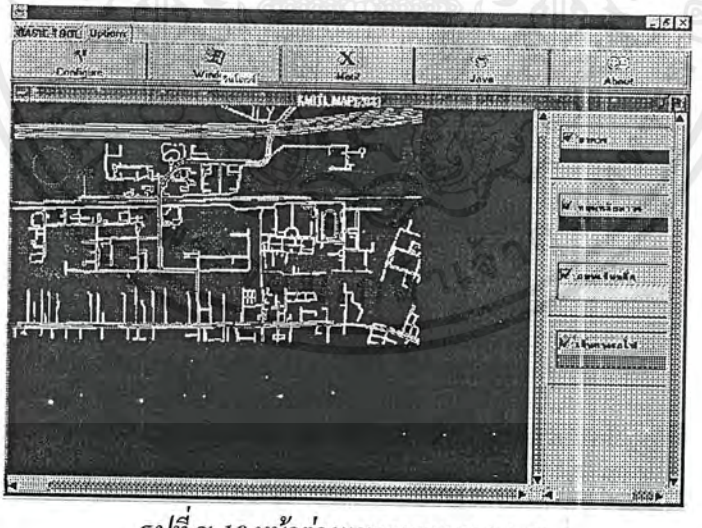
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-9 หน้าต่างแสดงผลแบบ Windows

ข.3.10.2 หน้าต่างแสดงผลแบบ Motif

เป็นหน้าต่างการแสดงผลของระบบ X-Windows เปลี่ยนเป็นหน้าต่างแสดงผลแบบ Motif ได้โดยการคลิ๊กที่ปุ่ม Motif ในแถ็บ Option

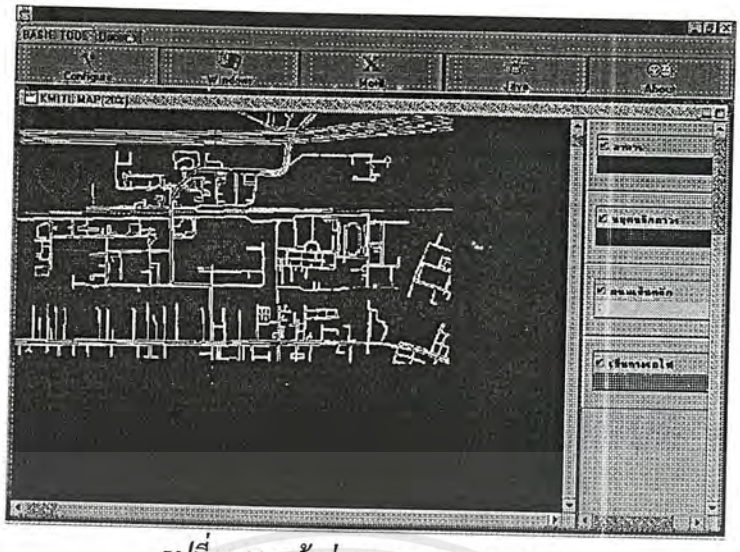


รูปที่ ข-10 หน้าต่างแสดงผลแบบ Motif

ข.3.10.3 หน้าต่างแสดงผลแบบ Java

เปลี่ยนเป็นหน้าต่างแสดงผลแบบ Motif ได้โดยการคลิ๊กที่ปุ่ม Motif ในแถ็บ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-11 หน้าต่างแสดงผลแบบ Java]

ข.3.11 การดูข้อมูลเกี่ยวกับโปรแกรมและผู้พัฒนา

สามารถดูรายละเอียดเกี่ยวกับโปรแกรมและผู้พัฒนาโปรแกรมได้โดยคลิ๊กที่ปุ่ม About ในแถ็บ Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Feuerstein, Steven. : " *Advanced Oracle PL/SQL : programming with packages* ", O'Reilly,1996

[2] "*Oracle8 Spatial Cartridge User's Guide and Reference Release 8.0.4*",Oracle Corporation.,1998

[3] *"Oracle 8i Spatial Cartridge User's Guide and Reference Release 8.1.5"*, Oracle Oracle Corporation.,1999

[4] "*Oracle8 Concepts Release 8.0"*,Oracle Corporation.,1997

[5] สุลักษณ์ สุ่มมาตย์ : "*Java กับการเชื่อมต่อฐานข้อมูลด้วย JDBC*", อินเทอร์เน็ต-อินทราเน็ต ปีที่2 ฉบับที่10, หน้าที่ 79-84, บริษัท อินทราเน็ตคอม จำกัด, กรุงเทพ, 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้