

การพัฒนาระบบสารสนเทศระบบฝึกงานและระบบทะเบียน
โดยใช้ CASE TOOLS ORACLE DESIGNER/2000

THE DEVELOPMENT OF A STUDENT'S ACTIVITIES AND
REGISTRATION INFORMATION SYSTEM USING
THE ORACLE DESIGNER/2000 CASE TOOL



T T 0 3 5 9 3 8 T



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย

ฉบับที่.....สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เลขทะเบียน..... 35938
..... น, เดือน, ปี..... 3 ก.ค. 2543

พ.ศ. 2543

ISBN 974-622-815-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THE DEVELOPMENT OF A STUDENT'S ACTIVITIES AND
REGISTRATION INFORMATION SYSTEM USING
THE ORACLE DESIGNER/2000 CASE TOOL**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2000**

ISBN 974-622-815-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2000

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การพัฒนาระบบสารสนเทศระบบฝึกงานและระบบทะเบียน โดยใช้ CASE TOOLS ORACLE DESIGNER/2000.
นักศึกษา	นายสมเกียรติ บริบาลประสิทธิ์
รหัสประจำตัว	36064020
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
พ.ศ.	2543
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.ศุภมิตร จิตตะยโสธร

บทคัดย่อ

การจัดการและการพัฒนาระบบฐานข้อมูล ในปัจจุบันพร้อมที่จะได้รับการปรับปรุงและเปลี่ยนแปลง โดยการนำเทคโนโลยีพร้อมทั้งทฤษฎีใหม่ๆ เข้ามาช่วยในการจัดการและการพัฒนาระบบให้ดีขึ้นโดยใช้เวลาในการพัฒนาน้อยลง สามารถปรับปรุงหรือแก้ไขระบบได้ง่ายและมีประสิทธิภาพขึ้น เพื่อช่วยแก้ปัญหาเกี่ยวกับ บุคลากรและผลผลิตที่ต้องการโดยเทคโนโลยีใหม่ที่จะนำเข้ามาช่วยในการพัฒนาและจัดการระบบฐานข้อมูลก็คือ Computer Aided Software Engineering (CASE) โดยเครื่องมือที่จะใช้และสนับสนุนทฤษฎีของ CASE ก็คือ Oracle Designer/2000

CASE ในที่นี้หมายถึง การพัฒนาระบบซอฟต์แวร์โดยอัตโนมัติ เราสามารถนำทฤษฎีของ CASE มาใช้ในการพัฒนาระบบขึ้นมาใหม่ ไม่ว่าระบบนั้นจะใหญ่หรือเล็กก็ตาม และในวิทยานิพนธ์ขอเสนอการใช้ทฤษฎีของ CASE TOOL METHOD มาทำการวิเคราะห์และออกแบบระบบโดยมีเครื่องมือที่ใช้คือ Oracle Designer/2000 ซึ่งจะช่วยให้ดำเนินการได้ง่ายและสะดวกต่อการปรับปรุงแก้ไขเนื่องจาก Oracle Designer/2000 นี้สนับสนุนหลักการทำงานแบบ Waterfall และ Spiral ทำให้ลดงานที่ซ้ำซ้อนลงได้ ซึ่งตัว Oracle Designer/2000 นี้จะช่วยในการจัดการฐานข้อมูล มีการกำหนดตาราง, คอลัมน์ (แอททริบิวต์) และข้อกำหนดต่างๆ มีตัวเจนเนอเรเตอร์ (Generator) ที่ช่วยในการเปลี่ยนแปลงคำจำกัดความของโพรซีเจอร์ (Procedure) ทริกเกอร์ (Trigger) ฟังก์ชัน (Function) และแพ็คเกจ (Package) โดยในตัวของ Oracle Designer/2000 ได้มีการแบ่งการทำงานออกเป็น 4 ส่วนใหญ่ คือ ส่วนของ Process Modeling, System Modeling, System Design และ System Generation

จากหลักการและเหตุผลที่กล่าวมาจึงได้ยกกรณีศึกษาขึ้นมาโดยทำการพัฒนาระบบฝึกงานและระบบทะเบียนที่มีส่วนเกี่ยวข้องกันโดยใช้ทฤษฎีของ CASE TOOLS METHOD พัฒนาโดยใช้ Oracle Designer/2000 และขั้นตอนที่กล่าวมาทั้งหมด

Thesis Title	The Development of A Student's Activities and Registration Information System Using The Oracle Designer/2000 Case Tool
Student	Mr. Somkiat Borribranprasit
Student ID.	36064020
Degree	Master of Science
Programme	Computer Science and Information Technology
Year	2000
Thesis Advisor	Assoc. Prof. Ph.D. Suphamit Chittayasothorn

ABSTRACT

At present the development of database systems is using new technologies and theories such as Computer Aided System Engineering (CASE) tools. They reduce time and improve performance of system design and development. CASE is a set of tools for automatic software development.

This paper presents system design and development with Oracle Designer/2000 which is one of the most popular CASE tools. This tool supports the Waterfall model and spiral model. It reduce the repeating processes in analyzing steps and helps system maintenance. In addition, Oracle Designer/2000 provides tool for creating tables and generating procedures, triggers, functions, and packages automatically. The Oracle Designer/2000 itself is composed of four major parts which are process modeling, system modeling, system design, and system generation.

The case study is the design and development of a training and registration information system for the Faculty of Engineering at KMITL by using the Oracle Designer/2000 CASE tool which is a new application development environment.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงด้วยดีเนื่องมาจากความร่วมมือ และช่วยเหลือจากบุคคลหลายฝ่ายด้วยกัน โดยเฉพาะ รศ.ดร.ศุภมิตร จิตตะยโสธร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่คอยแนะนำและให้คำปรึกษา ทั้งทางด้านวิชาการและด้านอื่น ๆ ตลอดการทำวิทยานิพนธ์นี้ ซึ่งต้องขอขอบพระคุณท่านอาจารย์เป็นอย่างมาก และต้องขอขอบพระคุณบุคคลสำคัญอีกสองท่านที่คอยถามถึงและให้กำลังใจข้าพเจ้าตลอดเวลา คือ บิดา มารดา อันเป็นที่รักเคารพยิ่งของข้าพเจ้า ซึ่งในชีวิตช่วงวัยศึกษาของข้าพเจ้า ท่านทั้งสองได้ทุ่มเททั้งกำลังกาย กำลังใจ อย่างมากจนหาที่เปรียบมิได้ ทำให้ข้าพเจ้าผ่านพ้นอุปสรรคต่างๆ จนมาถึง ณ จุดนี้ได้ แต่ก็เป็นที่น่าเสียใจที่บิดาของข้าพเจ้าไม่ได้อยู่ดูความสำเร็จของข้าพเจ้า พระคุณอันยิ่งใหญ่นี้ข้าพเจ้าจะขอร่ำลึกถึงและจดจำไปชั่วชีวิต

ขอขอบคุณ พี่ เพื่อน และน้อง ชาวลาดกระบังทุกท่าน ที่คอยแนะนำ ให้กำลังใจ และช่วยเหลือข้าพเจ้ามาตลอดในการวิทยานิพนธ์นี้ ความเป็นพี่ เพื่อน น้อง และความช่วยเหลือเหล่านี้จะคงอยู่ในใจของข้าพเจ้าตลอดไป

ขอขอบคุณเจ้าหน้าที่แผนกกิจกรรมนักศึกษาและแผนกทะเบียน คณะวิศวกรรมศาสตร์ ทุกท่าน ที่ให้ข้อมูลและรายละเอียดต่างๆ เกี่ยวกับระบบงาน

ขอขอบคุณแผนกสารสนเทศ คณะวิศวกรรมศาสตร์ที่อนุเคราะห์ให้ข้าพเจ้ามีเครื่องคอมพิวเตอร์และใช้เครื่องเพื่อทำวิทยานิพนธ์

Oracle Systems (Thailand) Co., Ltd. ที่ให้ความรู้พื้นฐานเกี่ยวกับ Oracle Desinger/2000 พร้อมทั้งเอกสารต่างๆ ที่เป็นประโยชน์ต่อการทำวิทยานิพนธ์นี้เป็นอย่างมาก

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่บิดา มารดา และผู้มีพระคุณทุกท่าน

สมเกียรติ บริบาลประสิทธิ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	2
1.2 ขอบเขตของงานวิจัย	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	3
1.4 วิธีการดำเนินการและส่วนประกอบของวิทยานิพนธ์	3
บทที่ 2 การออกแบบฐานข้อมูลแบบรีเลชันแนล	4
2.1 ฐานข้อมูลแบบรีเลชันแนล	4
2.2 โครงสร้างข้อมูลแบบรีเลชันแนล	4
2.3 การออกแบบฐานข้อมูลในรูปนอร์มอลฟอร์ม	7
2.4 กฎ 12 ข้อของโมเดมเชิงสัมพันธ์	17
2.5 ประโยชน์จากการใช้ระบบฐานข้อมูล	19
บทที่ 3 ทฤษฎี CASE*METHOD	21
3.1 ฟังก์ชันโมเดล(Function Model)	21
3.2 ฟังก์ชันไฮราเคียล(Function Hierarchy)	23
3.3 ฟังก์ชันดีเพนเดนซีไดอะแกรม(Function Dependency Diagram)	26
3.4 ฟังก์ชันลอจิก(Function Logic)	28
3.5 โพรเซสโมเดลลิง(Process Modeling)	28
3.6 ดาต้าโฟลว์ไดอะแกรม(Dataflow Diagram)	29
3.7 การเขียนเอนติตี้รีเลชันชิพโมเดล(Entity Relationship Model)	30
บทที่ 4 สถาปัตยกรรม ORACLE	35
4.1 หลักการเบื้องต้นของอราเคิล	35
4.2 สถาปัตยกรรมของอราเคิล (Oracle Architecture)	36

เอกสารนี้เป็นเอกสารลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.3 Semantic Data Control	45
4.4 Transaction Management.....	46
4.5 Backup และ Recovery.....	52
4.6 Query Processing	52
4.7 Additional Tool	56
4.8 Distributed Database Management.....	60
บทที่ 5 หลักการทำงานของ Oracle Designer/2000	62
5.1 Oracle Designer/2000.....	62
5.2 องค์ประกอบของ Designer/2000	62
5.3 Process Modeler	64
5.4 Systems Modeler	65
5.5 Systems Designer	67
5.6 Generator.....	69
5.7 Repository Administrator.....	70
5.7 สรุป.....	70
บทที่ 6 วิเคราะห์และออกแบบระบบงานฝึกงาน	71
6.1 การศึกษาระบบเก่า.....	71
6.2 ปัญหาต่าง ๆ ในระบบเก่า	73
6.3 ขั้นตอนการทำงาน.....	73
6.4 การศึกษาระบบใหม่	76
6.5 การแก้ไขปัญหาดัง ๆ ในระบบเดิม.....	79
บทที่ 7 สรุปผลการศึกษาและวิจัย	82
บรรณานุกรม	86
ภาคผนวก	87
ภาคผนวก ก ผลงานที่ได้รับการตีพิมพ์.....	88
ภาคผนวก ข ตัวอย่างสคริปในการสร้างฐานข้อมูล.....	98
ภาคผนวก ค รายงานรายละเอียดของ Function.....	105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
ภาคผนวก ง รายละเอียดของ Process Modeler.....	113
ภาคผนวก จ รายละเอียดของ Entity Relationship Model.....	115
ภาคผนวก ฉ รายละเอียดของ Dataflow Diagrammer	117
ประวัติผู้เขียน.....	124



สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปนอร์มอลฟอร์ม	8
2.2 แสดงตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้นแรก.....	9
2.3 แสดงตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้นสอง.....	10
2.4 แสดงตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้นสาม.....	11
2.5 แสดงตารางที่เป็น 3NF แต่ไม่เป็นอยู่ในรูปนอร์มอลฟอร์มขั้น Boyce/Codd.....	12
2.6 แสดงตารางที่แบ่งแล้วอยู่ในรูปแบบนอร์มอลฟอร์ม Boyce/Codd.....	13
2.7 แสดงตารางที่ไม่อยู่ในรูปแบบนอร์มอลฟอร์มขั้นสี่	13
2.8 แสดงตารางที่อยู่ในนอร์มอลฟอร์มขั้นสี่.....	14
2.9 แสดงตารางที่ไม่อยู่ในนอร์มอลฟอร์มขั้นห้า.....	15



สารบัญภาพ

ภาพ	หน้า
1.1 แสดงขั้นตอนการพัฒนาระบบสารสนเทศ.....	1
2.1 แสดงตารางความสัมพันธ์ S (C.J. Date 1986 : 233).....	4
2.2 นอร์มอลฟอร์ม	7
2.3 แสดง Function Dependency ของตาราง S.....	10
2.4 แสดงตารางข้อมูลที่อยู่ในนอร์มอลฟอร์มขั้นห้า	16
3.1 Business Model	22
3.2 ภาพแทน Business Function.....	23
3.3 แสดง Function Hierarchy.....	24
3.4 Top-Down plus Bottom-Up	25
3.5 A Function Dependency	27
3.6 A Detail Function Dependency Diagram	27
3.7 ดาต้าโฟลว์.....	30
3.8 แสดงสัญลักษณ์ของ Entity.....	31
3.9 แสดงสัญลักษณ์ของ Entity, Attribute และสัญลักษณ์ที่ใช้.....	32
3.10 แสดงสัญลักษณ์ของความสัมพันธ์.....	32
3.11 แสดง Entity Relationship.....	34
4.1 แสดงความสัมพันธ์ระหว่าง ดาต้าบล็อก, เอ็กเทนต์ และเชกเมนต์.....	35
4.2 ออราเคิลดาต้าเบส อินสแตนซ์ (Oracle Database Instance).....	37
4.3 ออราเคิลอินสแตนซ์แบบหลายโพรเซส	38
4.4 แสดงภาพ USER และ SERVER PROCESS	39
4.5 แสดงภาพโครงสร้างโดยรวมของออราเคิล.....	41
5.1 แสดงองค์ประกอบของ Oracle Designer/2000	63
5.2 แสดงหน้าจอของ Designer/2000.....	64
5.3 แสดงตัวอย่างของ Process Modeler.....	65
5.4 Entity Relationship Diagrammer	66
5.5 Dataflow Diagrammer	66
5.6 Data Diagrammer	67
6.1 แสดง Function Hierarchy ของแผนกกิจกรรมนักศึกษา	72
6.2 แสดงภาพดาต้าโฟลว์ระบบเก่า	74
6.3 แสดง Process การทำงานของระบบ	77
6.4 แสดงภาพ ER-Diagram ของระบบ	78
6.5 แสดงภาพ Dataflow Level 0	79

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ การใช้หรือการแก้ไขใดๆโดยไม่ได้รับอนุญาตเป็นการฝ่าฝืน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพ	หน้า
6.6 แสดงหน้าจอที่สร้างได้จาก Designer/2000.....	80
6.7 แสดงหน้าจอการใส่ข้อมูลของบริษัท.....	80
6.8 แสดงภาพหน้าจอการใส่เงื่อนไขการรับนักศึกษา.....	81

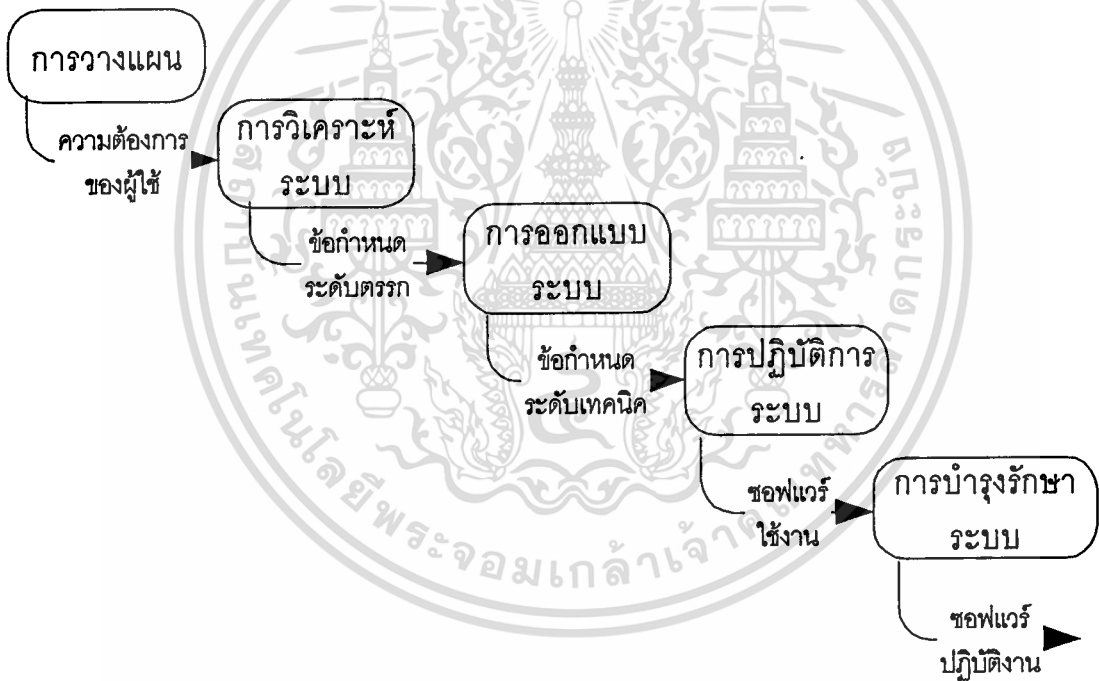


บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

จะเห็นว่าในปัจจุบัน คอมพิวเตอร์ได้เข้ามามีบทบาทอย่างมากต่อการปฏิบัติงานในหน่วยงานต่างๆ เพิ่มมากขึ้น โดยมีการนำเอาความสามารถของคอมพิวเตอร์ที่ได้รับการพัฒนาขึ้นมาอย่างรวดเร็วทั้งทางด้านฮาร์ดแวร์ (Hardware) ซอฟต์แวร์ (Software) เข้ามาประยุกต์ใช้กับงานในส่วนต่างๆ และได้มีแนวคิดและเสนอแนะทฤษฎีใหม่ๆ ขึ้นมารองรับกับการพัฒนา ซึ่งทฤษฎีเหล่านี้ก็สามารถที่จะช่วยทำให้การพัฒนาระบบซอฟต์แวร์ใหม่นั้นสามารถที่จะพัฒนาได้ง่าย สามารถลดข้อผิดพลาดต่างๆ ลงได้มาก และมีหลักการที่สามารถที่จะใช้ตรวจสอบเพื่อหาข้อผิดพลาดได้ง่ายขึ้นมากและเราสามารถที่ปรับปรุงแก้ไขเพื่อพัฒนาต่อไปในอนาคตได้ง่ายและสะดวก



ภาพที่ 1.1 แสดงขั้นตอนการพัฒนาระบบสารสนเทศ

ในแนวทางการพัฒนาซอฟต์แวร์ ในยุคเก่าๆ ตามหลักการและทฤษฎีแล้วจะมีขั้นตอนต่างๆ มากพอสมควรแต่ในทางปฏิบัติแล้วผู้พัฒนามักจะละเว้นการทำงานในส่วนของการเอกสารไป เช่น ไม่เขียนขั้นตอนการทำงาน ไม่เขียนอีอาร์โมเดล (ER-Model) ไม่เขียนดาต้าโฟลว์ (Dataflow Diagram) ซึ่งขั้นตอนต่างๆ ที่กล่าวมาอาจจะดูเหมือนไม่ค่อยมีความเป็นที่จะต้องทำ แต่จะเห็นได้ว่าเมื่อเกิดปัญหาระบบงานที่พัฒนาขึ้นมาเกิดทำงานผิดพลาดหรือมีความต้องการที่จะขยายขีดความสามารถของระบบงานเดิมแล้วละก็ ผู้พัฒนาถึงจะเห็นถึงความสำคัญของงานเอกสารที่ถูกละเลยไปในช่วงแรก เพราะเอกสารต่างๆ หรือขั้นตอนที่ถูกละเลยไปนั้นมีประโยชน์มากกว่าที่คิดอีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการศึกษาระบบงานว่าถ้าเกิดข้อผิดพลาดขึ้นมาน่าจะเกิดจากจุดใดและควรที่จะแก้ไขที่จุด ดัง นั้นจึงได้เริ่มมีแนวคิดที่จะสร้างทฤษฎีและเครื่องมือที่จะมารองรับขั้นตอนต่างๆ ในการพัฒนาระบบงานขึ้นมา

CASE*METHOD เป็นทฤษฎีที่ช่วยในการพัฒนาและจัดการระบบฐานข้อมูลให้ดีและมีประสิทธิภาพ โดยใช้ระยะเวลาในการพัฒนาระบบน้อยกว่าทฤษฎีที่ใช้อยู่ในปัจจุบัน สามารถที่จะปรับปรุงหรือแก้ไขระบบได้ง่ายขึ้น เพื่อช่วยในการแก้ปัญหา ไม่ว่าจะเป็นปัญหาที่เกี่ยวกับบุคลากรและผลผลิตที่ต้องการ โดยเทคโนโลยีที่จะนำมาช่วยในการพัฒนาและจัดการระบบฐานข้อมูล คือ Computer Aided Software Engineering (CASE) โดยจะใช้เครื่องมือที่สนับสนุนทฤษฎี CASE คือ Oracle Designer/2000 ซึ่งเป็นเครื่องมือตัวใหม่ที่น่าจะได้นำมาพัฒนาระบบเพราะโดยหลักการและทฤษฎีของ CASE เองแล้วการที่พัฒนาระบบจะต้องเป็นลักษณะของการพัฒนาแบบอัตโนมัติโดยเราจะต้องทำการใส่ข้อมูลที่ระบบหรือเครื่องนั้นๆ ใช้ในการสร้างระบบ และการที่ในวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ เครื่องมือ Oracle Designer/2000 เพราะในตัวของ Oracle Designer/2000 นั้นสนับสนุนหลักการการทำงานแบบ Spiral

1.2 ขอบเขตของงานวิจัย

ขอบเขตของงานวิจัยนี้ได้แบ่งออกเป็น 3 ส่วน คือ

- ทำการศึกษาทฤษฎี CASE*METHOD ที่จะนำมาใช้ในการออกแบบและพัฒนาระบบงาน
- ทำการศึกษาระบบฐานข้อมูล Oracle 7.x และ ศึกษาหลักการพร้อมทั้งการใช้งานของเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบงานซึ่งเครื่องมือนี้ได้สนับสนุนทฤษฎีของ CASE*METHOD โดยจะใช้เครื่องมือ คือ Oracle Designer/2000 release 1.3.2 พร้อมทั้ง Developer/2000 release 1.3.2 ซึ่งใช้ในการต้นแบบของฟอร์มและต้นแบบรายงานที่ Oracle Designer/2000 ต้องนำไปใช้งาน
- ขอบเขตในการพัฒนาระบบฝึกงานและระบบทะเบียนนักศึกษา ซึ่งระบบงานทั้งสองนี้จะใช้เป็นตัวอย่างเพื่อที่จะสนับสนุนหลักการของ CASE*METHOD และเครื่องมือที่ใช้ในการพัฒนา โดยระบบงานทั้งสองจะถูกพัฒนาขึ้นจากหลักการของ CASE*METHOD

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- ช่วยทำให้การทำงานและมีแนวคิดที่เป็นระบบมากขึ้น
- สามารถที่จะไปช่วยในการปรับปรุงคุณภาพของซอฟต์แวร์ให้ดีขึ้น
- ช่วยให้การพัฒนาและออกแบบระบบได้เร็วขึ้น
- สามารถนำความรู้และแนวคิดของ CASE มาปรับปรุงองค์กรได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นแนวทางใหม่ที่จะพัฒนาระบบขึ้นมาใช้
- สามารถที่จะนำทฤษฎีของ CASE มาทำการ Reengineer ภายในองค์กร
- เห็นความสำคัญและให้ความสำคัญกับการออกแบบมากกว่าการ Coding
- สามารถที่จะนำความรู้ทางด้านการบริหารองค์กรมาผสมผสานกับความรู้ทางคอมพิวเตอร์ได้
- สามารถที่จะเอาเครื่องมือ Oracle Designer/2000 มาทำการแก้ปัญหา Y2K
- หวังว่าจะสามารถที่จะนำความรู้ส่วนนี้มาช่วยในการพัฒนาประเทศได้

1.4 วิธีการดำเนินการและส่วนประกอบของวิทยานิพนธ์

ในการทำวิทยานิพนธ์นี้ได้แบ่งการดำเนินงานออกเป็น 4 ส่วนใหญ่ คือ

- ศึกษาทฤษฎี CASE*METHOD โดยมีขั้นตอนในการศึกษาดังนี้คือ
 - Function Modeling
 - Function Hierachy
 - Function Dependency Diagram
 - Function Logic
 - Process Modeling
 - Dataflow Diagram
 - Entity Relational Model
- ศึกษาและติดตั้งฐานข้อมูล Oracle
- ศึกษาหลักการทำงานและวิธีการใช้งาน Oracle Designer/2000
- พัฒนาระบบงานฝึกงานนักศึกษาและระบบงานทะเบียนนักศึกษา

ในวิทยานิพนธ์นี้ประกอบด้วยเนื้อหาโดยย่อๆ ดังนี้คือ

บทที่ 2 การออกแบบฐานข้อมูลแบบรีเลชันนอล

บทที่ 3 ทฤษฎี CASE*METHOD

บทที่ 4 สถาปัตยกรรม Oracle

บทที่ 5 หลักการทำงานของ Oracle Designer/2000

บทที่ 6 วิเคราะห์และออกแบบระบบงานฝึกงานและระบบงานทะเบียนนักศึกษา

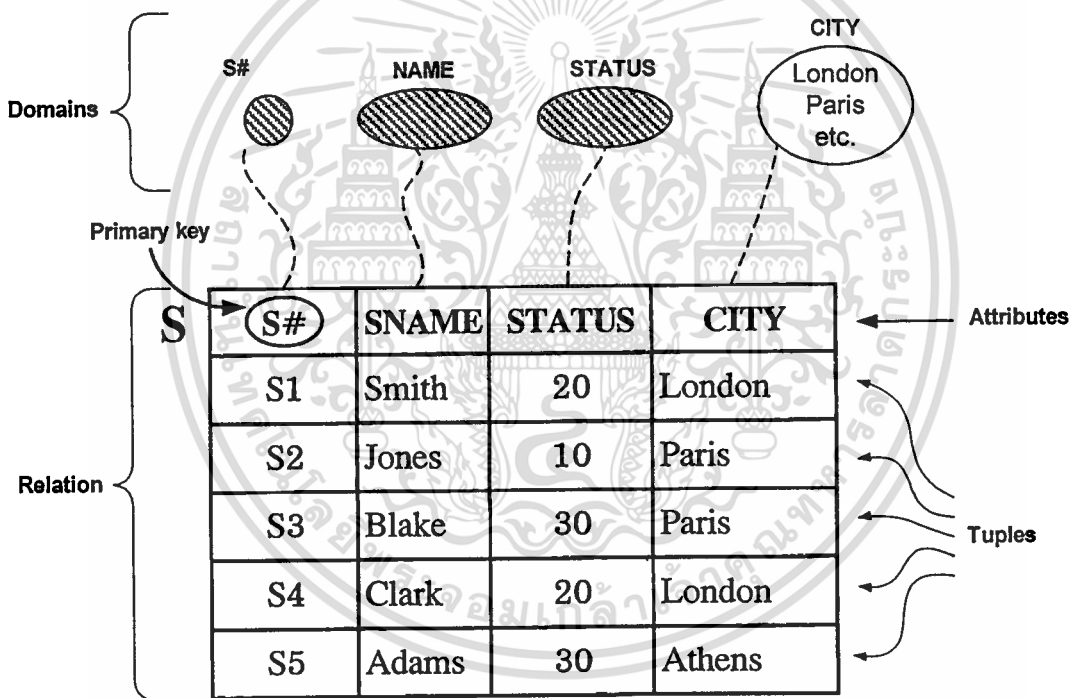
บทที่ 7 สรุปผลการศึกษาและวิจัย

บทที่ 2

การออกแบบฐานข้อมูลแบบรีเลชันแนล

2.1 ฐานข้อมูลแบบรีเลชันแนล

ฐานข้อมูล คือ ที่เก็บข้อมูลและความสัมพันธ์ระหว่างข้อมูล ระบบฐานข้อมูลที่ใช้กันอยู่ในปัจจุบันนี้ สามารถที่จะแบ่งออกเป็นโมเดลข้อมูลใหญ่ๆ ได้ 3 โมเดล คือ ไฮราจิคัลดาต้าเบสโมเดล เน็ตเวิร์กดาต้าเบสโมเดล และรีเลชันแนลดาต้าเบสโมเดล ซึ่งเป็นโมเดลที่ได้รับความนิยมแพร่หลายมากที่สุดและเป็นระบบฐานข้อมูลที่ใช้มากที่สุดในปัจจุบันนี้ เพราะมีความยืดหยุ่นสามารถที่จะออกแบบและทำการแก้ไขได้ง่าย เนื่องจากฐานข้อมูลแบบรีเลชันแนลเป็นฐานข้อมูลที่ถูกพัฒนามาจากทฤษฎีทางคณิตศาสตร์ซึ่งสามารถที่จะพิสูจน์หาเหตุผลมาประกอบได้



ภาพที่ 2.1 แสดงตารางความสัมพันธ์ S (C.J. Date 1986 : 233)

2.2 โครงสร้างข้อมูลแบบรีเลชันแนล

โมเดลเชิงสัมพันธ์เป็นโมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกเก็บด้วยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ฐานข้อมูลแบบรีเลชันแนลจะแสดงการเก็บข้อมูลในรูปของตารางดังที่แสดงไว้ในภาพที่ 2.1 ซึ่งเป็นรูปแบบที่ง่ายต่อการเข้าใจ และสามารถที่จะตอบสนองต่อคำถามที่ซับซ้อนได้ดีกว่าโมเดลฐานข้อมูล

ชนิดอื่น องค์ประกอบที่สำคัญของระบบจัดการฐานข้อมูลแบบรีเลชันแนลแบ่งออกเป็น 3 ส่วนใหญ่คือ

2.2.1 โครงสร้างข้อมูล

2.2.1.1 โครงสร้างฐานข้อมูลมืองค์ประกอบดังนี้

- แอททริบิวต์ (Attributes) หมายถึง ชื่อคอลัมน์ของตารางความสัมพันธ์ ที่ได้บ่งชี้ถึงคุณสมบัติของสิ่งต่างๆ ที่มีความสัมพันธ์กันประกอบขึ้นมาเป็นตารางฐานข้อมูล เช่น แอททริบิวต์ S# หมายถึงรหัสของพนักงาน, แอททริบิวต์ SNAME หมายถึงชื่อของพนักงาน และแอททริบิวต์ CITY หมายถึงชื่อเมือง โดยที่แอททริบิวต์ดังกล่าวมีความสัมพันธ์กันดังนี้คือ พนักงานที่รหัส (S#) QC1203 มีชื่อ (SNAME) ว่า “John” และอาศัยอยู่ในเมือง (CITY) “New York”
- โดเมน (Domains) หมายถึง ขอบเขตของข้อมูลในแต่ละแอททริบิวต์ เช่น ข้อมูลที่เก็บลงแอททริบิวต์ CITY จะต้องเป็นชื่อของเมืองในทวีปยุโรปเท่านั้น
- ทัปเปิล (Tuples) หมายถึง แถวของข้อมูลภายในตารางความสัมพันธ์
- ไพรมารีคีย์ (Primary key) หมายถึง แอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่ทำหน้าที่เป็นตัวบ่งบอกถึงเอกลักษณ์ของข้อมูลในแต่ละทัปเปิลของตารางความสัมพันธ์ โดยที่แอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่เป็นไพรมารี (Primary Key) จะต้องไม่มีค่าที่ซ้ำกันในทัปเปิลอื่นๆ เลย
- รีเลชัน (Relation) หมายถึง ผลคูณคาร์ทีเซียนของโดเมนที่สนใจในแอปพลิเคชัน ประกอบขึ้นด้วยองค์ประกอบ 2 ส่วน คือ เฮดดิ้ง (Heading) และ บอดี้ (Body)

2.2.1.2 คุณสมบัติที่สำคัญของฐานข้อมูลแบบรีเลชันแนล

- ต้องไม่มีทัปเปิลที่ซ้ำกันภายในรีเลชัน เนื่องจากบอดี้ (Body) ของรีเลชันซึ่งประกอบขึ้นจากทัปเปิลหลายทัปเปิล มีลักษณะเป็นเซตทางคณิตศาสตร์ในรูปแบบของผลคูณคาร์ทีเซียน การซ้ำกันของทัปเปิลจึงเปรียบเสมือนการซ้ำของสมาชิกภายในเซต ซึ่งผิดนิยามทางคณิตศาสตร์ ดังนั้นจึงกำหนดคุณสมบัติข้อนี้ไว้ เพื่อให้โครงสร้างข้อมูลแบบสัมพันธ์เป็นไปตามกฎทางคณิตศาสตร์ และเนื่องจากคุณสมบัติความเป็นเอกลักษณ์ของแต่ละทัปเปิลนี้ ทำให้เรากล่าวได้ว่า “ทุกๆ รีเลชันจะต้องมีไพรมารีคีย์ (Primary Key) เสมอ” เพราะอย่างน้อยที่สุดกลุ่มของแอททริบิวต์ที่กำหนดความเป็นเอก

ลักษณะของแต่ละทาบเปิล ก็คือเซตของแอททริบิวต์ทั้งหมดภายในรีเลชันนั่นเอง

- ลำดับของทาบเปิลไม่มีความสำคัญในการจัดเก็บ จากนิยามทางคณิตศาสตร์ที่ถือว่าลำดับของสมาชิกภายในเซตไม่มีความสำคัญ บอด้ของรีเลชันซึ่งถือว่าเป็นเซตที่มีสมาชิกเป็นทาบเปิลก็เช่นกัน ที่ลำดับในการจัดเก็บของทาบเปิลต้องไม่มีผลในการใช้งานรีเลชัน
- ลำดับของแอททริบิวต์ไม่มีความสำคัญในการจัดเก็บ จากเหตุผลเดียวกันกับคุณสมบัติในข้อสอง ดังนั้นลำดับของแอททริบิวต์ภายในทาบเปิลภายในรีเลชันจึงไม่มีผลต่อการจัดเก็บและการใช้งานแต่อย่างใด
- ข้อมูลของแต่ละทาบเปิลต้องเป็นค่าเดียวเสมอ หมายถึงการกำหนดแอททริบิวต์นั้น ค่าของแอททริบิวต์ภายในทาบเปิลใดๆ จะต้องดึงค่าข้อมูลจากโดเมนที่มันอ้างอิงอยู่มาเพียงค่าเดียวเท่านั้น

2.2.2 ความถูกต้องของข้อมูล (Integrity)

การบังคับความถูกต้องของข้อมูลบนระบบฐานข้อมูลแบบรีเลชันอย่างน้อยต้องสามารถที่จะบังคับความถูกต้องของข้อมูลต่อไปนี้ได้

2.2.2.1 กฎข้อบังคับความถูกต้องของข้อมูลชนิดเอนติตี้ (Entity type Constraint)

กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับว่าค่าของข้อมูลภายใน แอททริบิวต์ซึ่งเป็นแอททริบิวต์ของไพรมารีคีย์จะต้องไม่เป็นค่า null (ค่า null หมายถึงข้อมูลที่ไม่มีค่าอะไรเก็บอยู่เลย)

2.2.2.2 กฎข้อบังคับความถูกต้องของข้อมูลในการอ้างอิง (Referential Integrity Constraint)

ถ้ามีความสัมพันธ์ R2 ซึ่งมีฟอร์เรนคีย์ FK ที่อ้างอิงกับไพรมารีคีย์ PK ของความสัมพันธ์ R1 แล้ว ทุกๆ ค่าของข้อมูลที่ปรากฏอยู่ใน FK ของความสัมพันธ์ R2 จะต้องปรากฏอยู่ภายในค่า PK ของความสัมพันธ์ R1 ด้วยเสมอ ยกเว้นกรณีที่ค่าของ FK เป็นค่า NULL

2.2.3 การจัดการข้อมูล

ข้อมูลภายในของข้อมูลแบบรีเลชันแนล จะต้องถูกจัดการโดยภาษาที่ใช้จัดการข้อมูลระบบจัดการฐานข้อมูลเท่านั้น โดยภาษาดังกล่าวจะต้องมีคุณสมบัติตามนิยามของ “ความสัมพันธ์ที่สมบูรณ์ (Relational Complete)” กล่าวคือจะต้องสนับสนุนกลุ่มปฏิบัติการพื้นฐานที่นิยามไว้ในภาษาพีชคณิตสัมพันธ์ (Relational Algebra) หรือ Relational Calculus ซึ่งประกอบด้วยการปฏิบัติการ 8 อย่าง โดยแบ่งออกเป็น 2 กลุ่มได้ดังนี้

เอกพจน์ภาษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันเป็นตาราง เรียกว่า นอร์มอลฟอร์มขั้นแรก (1NF) อาจมีหลายตาราง และขั้นสุดท้ายคือการนำ ตารางที่ได้จาก 1NF มาเทียบกับมาตรฐานระดับสูงขึ้นคือ 2NF ไปจนถึง 5NF ถ้าผ่านไปถึง 5NF โดยไม่มีคุณสมบัติใดขัดแย้งเลยก็แสดงว่าตารางนั้นๆ ถูกนอร์มอลฟอร์มเรียบร้อยแล้วแต่ ถ้ามีคุณสมบัติที่ขัดแย้งจะต้องทำการแบ่งตารางออกเป็นตารางย่อยที่ได้ตามมาตรฐาน เรียกว่า Decomposition Method

ตารางที่ 2.1 แสดงตารางฐานข้อมูลที่ไม่อยู่ในรูปนอร์มอลฟอร์ม

FIRST

Repeating Group

S #	STATUS	CITY	P #	QTY
S 1	20	LONDON	P 1	300
	20	LONDON	P 2	200
	20	LONDON	P 3	400
	20	LONDON	P 4	200
	20	LONDON	P 5	100
	20	LONDON	P 6	100
S 2	10	PARIS	P 1	300
	10	PARIS	P 2	400
S 3	10	PARIS	P 2	200
S 4	20	LONDON	P 2	200
	20	LONDON	P 4	300
	20	LONDON	P 4	400

2.3.1 นอร์มอลฟอร์มขั้นแรก(FIRST NORMAL FORM : 1NF)

ตารางฐานข้อมูลที่สามารถใช้ได้กับระบบจัดการฐานข้อมูลแบบรีเลชันแนล จะ ต้องมี คุณสมบัติอย่างน้อยที่สุดตามคุณสมบัติของรีเลชันที่ได้กล่าวไว้ในหัวข้อ 2.2.1.2 และตารางที่จะเป็น 1NF ได้ก็ต่อเมื่อ ค่าทุกค่าของทุกแอททริบิวต์ในรีเลชันเป็นอะตอมมิก-แวลลู หรือจะกล่าวง่ายๆ คือ จะต้องไม่มี Repeating group

ข้อเสียของนอร์มอลฟอร์มขั้นแรก

- การทำ Multiple Update เนื่องจากความซ้ำซ้อน แต่ปัจจุบันสามารถใช้ภาษา SQL ช่วยในการทำ Multiple Update ได้
- ข้อมูลอาจจะหายไปโดยไม่ตั้งใจ และอาจเกิดปัญหาจากการเพิ่มข้อมูลเข้าไปในตาราง ได้โดยอาจทำให้ข้อมูลขัดแย้งกันได้

ตารางที่ 2.2 แสดงตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้นแรก

S	S#	STATUS	CITY	P#	QTY
	S1	20	LONDON	P1	300
	S1	20	LONDON	P2	200
	S1	20	LONDON	P3	400
	S1	20	LONDON	P4	200
	S1	20	LONDON	P5	100
	S1	20	LONDON	P6	100
	S2	10	PARIS	P1	300
	S2	10	PARIS	P2	400
	S3	10	PARIS	P2	200
	S4	20	LONDON	P2	200
	S4	20	LONDON	P4	300
	S4	20	LONDON	P4	400

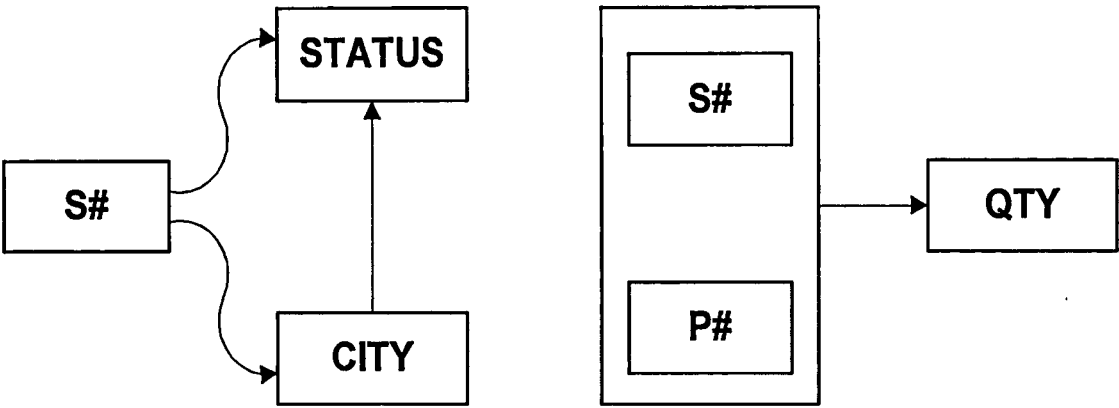
- เปลื้องเนื้อที่ ๆ ใช้เก็บข้อมูล และแรงงานในการป้อนข้อมูลเข้ามา
สรุปคือ 1NF คือการนำเอาแอททริบิวต์ที่เกี่ยวข้องมาอยู่ด้วยกัน ยังไม่มีการดูแลความซ้ำซ้อนเลย ความซ้ำซ้อนที่เกิดขึ้นทำให้เกิดปัญหา Update, Delete และ Insert

2.3.2 นอร์มอลฟอร์มขั้นสอง(SECOND NORMAL FORM : 2NF)

ตารางใดก็ตามจะเป็นนอร์มอลฟอร์มขั้นที่สองได้ก็ต่อเมื่อ

- ต้องเป็นฐานข้อมูลที่ต้องอยู่ในรูปแบบนอร์มอลฟอร์มขั้นแรกและ
 - ทุกแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์ (non-key) ของตาราง ต้องขึ้นอยู่กับไพรมารีคีย์เท่านั้นจะขึ้นอยู่กับสับเซตของไพรมารีคีย์ไม่ได้ หรืออาจกล่าวได้ว่าทุกแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์ของตารางต้องขึ้นอยู่กับทั้งหมด(Fully Depend on) กับไพรมารีคีย์
- การที่จะตรวจสอบว่าแอททริบิวต์ใดขึ้นอยู่กับแอททริบิวต์ใดนั้น จะต้องทำการเขียนความสัมพันธ์ของแอททริบิวต์แต่ละตัวให้อยู่ในรูปแบบของการขึ้นอยู่กับทั้งหมด(Function Dependency : FD) เป็น FD Diagram หรือรูปแบบของภาษาเขียนซึ่งมีรูปแบบดังนี้

[Determinant Attribute] ----> [Dependent Attribute]



ภาพที่ 2.3 แสดง Function Dependency ของตาราง S

โดยความสัมพันธ์นี้จะต้องอยู่ในรูปแบบของ one to one หรือ many to one หรืออาจกล่าวได้ว่า Determinant Attribute คือ แอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่สามารถกำหนดค่าของ Dependent Attribute หนึ่งค่าไม่ซ้ำกันได้หรือมีคุณสมบัติเป็นหนึ่ง (unique) นั้นเอง ดังตัวอย่างต่อไปนี้

- S#,P# ----> Qty
- S# ----> Status
- S# ----> City
- City ----> Status

ตารางที่ 2.3 แสดงตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มชั้นสอง

S	S			SP	SP		
	S#	STATUS	CITY		S#	P#	QTY
	S1	20	LONDON		S1	P1	300
	S2	10	PARIS		S1	P2	200
	S3	10	PARIS		S1	P3	400
	S4	20	LONDON		S1	P4	200
	S5	30	ATHENS		S1	P5	100
					S1	P6	100
					S2	P1	300
					S2	P2	400
					S3	P2	200
					S4	P2	200
					S4	P4	300
					S4	P5	400

จากตัวอย่างทั้ง FD Diagram และจากการเขียน FD จะเห็นว่าแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์ (non key) ขึ้นอยู่กับสับเซตของไพรมารีคีย์ ดังนั้นถือว่ายังไม่ขึ้นอยู่ทั้งหมดกับไพรมารีคีย์ เรียกว่ายังไม่เป็น Fully FD ดังนั้นจะต้องทำการแบ่งตารางออกและเมื่อแบ่งตารางออกแล้วก็ต้องทำการตรวจสอบว่าแต่ละตารางจะต้องเป็น Fully FD เท่านั้นจึงจะถือว่าตารางนั้นอยู่ในนอร์มอลฟอร์มขั้นที่สอง (2NF) แล้ว

2.3.3 นอร์มอลฟอร์มขั้นสาม (THIRD NORMAL FORM : 3NF)

ตารางใดก็ตามจะเป็นนอร์มอลฟอร์มขั้นสามได้ก็ต่อเมื่อ

- ต้องเป็นฐานข้อมูลที่ต้องอยู่ในรูปแบบนอร์มอลฟอร์มขั้นสองและ
- ทุกแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์ ต้องไม่ขึ้นอยู่กับแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์ของตารางด้วยตนเอง

จากคุณสมบัติที่ใช้พิจารณาว่าตารางใดเป็นนอร์มอลฟอร์มขั้นสามหรือไม่นั้นอาจกล่าวได้อีกนัยหนึ่งได้ว่า ตารางที่จะเป็นนอร์มอลฟอร์มขั้นสามจะต้องเป็นนอร์มอลฟอร์มขั้นสองก่อนและทุกแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์จะต้องไม่ขึ้นต่อกันเอง เรียกว่า Transitivity dependent on primary key

จากตัวอย่างตารางที่อยู่ในนอร์มอลฟอร์มขั้นสองจะเห็นว่าในตาราง S จะมีแอททริบิวต์ที่ไม่ใช่ไพรมารีคีย์ขึ้นอยู่กันเอง ดังนั้นตาราง S จึงไม่เป็นนอร์มอลฟอร์มขั้นสามส่วนตารางอื่นๆ จากตัวอย่างไม่ว่าจะเป็นตาราง P หรือตาราง SP, เป็นนอร์มอลฟอร์มขั้นสามแล้ว ดังนั้นการที่จะทำให้ตาราง S เป็นนอร์มอลฟอร์มขั้นสามจะต้องการแบ่งตาราง S ออกมา ขั้นตอนการแบ่งก็คือ จะต้องเขียนหรือหาแอททริบิวต์ที่ขึ้นกันเองออกมาให้หมดจากตัวอย่างจะได้

S# ----> City
City ----> Status
S# ----> Status เรียกว่า Transitive FD

ตารางที่ 2.4 แสดงตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้นสาม

SC	S#	CITY	CS	CITY	STATUS
	S1	LONDON		ATHENS	30
	S2	PARIS		LONDON	20
	S3	PARIS		PARIS	10
	S4	LONDON		ROME	50
	S5	ATHENS			

เมื่อเขียนแอททริบิวต์ที่ขึ้นกันเองออกมาแล้วง่ายขึ้น คือ จะต้องพยายามกำจัดไม่ให้ตารางนั้นมี Transitive FD เมื่อตารางไม่มี Transitive FD แล้ว ตารางที่ถูกแบ่งออกมาก็จะเป็นนอร์มอลฟอร์มขั้นสาม

2.3.4 ฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้น Boyce/Codd(BCNF)

ตารางใดก็ตามจะเป็นนอร์มอลฟอร์มขั้น Boyce/Codd ได้ก็ต่อเมื่อ

- ต้องเป็นตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้นสามและ
- ทุกแอททริบิวต์ที่เป็นไพรมารีคีย์ของตารางจะต้องไม่ขึ้นอยู่กับสับเซตของ ไพรมารีคีย์อื่น หรือกล่าวอีกนัยหนึ่งว่าทุก determinant ต้องเป็น candidate key

ตารางที่เป็นอยู่ในนอร์มอลฟอร์มขั้น 3 (3NF) ส่วนใหญ่จะเป็นนอร์มอลฟอร์มขั้น Boyce/Codd แล้ว ยกเว้นตารางที่มีลักษณะดังนี้คือ

- มีแคนดิเดตคีย์ (candidate key คือ คีย์ที่สามารถที่จะเป็นไพรมารีคีย์ได้) มากกว่าหนึ่งตัว
- แคนดิเดตคีย์เหล่านั้นเป็นคอมโพสิทคีย์ (composite key คือ คีย์ที่ประกอบด้วยแอททริบิวต์มากกว่าหนึ่งตัว)

ตารางที่ 2.5 แสดงตารางที่เป็น 3NF แต่ไม่เป็นอยู่ในรูปนอร์มอลฟอร์มขั้น Boyce/Codd

SSP	S#	SNAME	P#	QTY
	S1	SMITH	P1	300
	S1	SMITH	P2	200
	S1	SMITH	P3	400
	S1	SMITH	P4	200

- คอมโพสิทคีย์เหล่านั้นมีส่วนแอททริบิวต์บางตัวซ้ำกันอยู่ (overlap)

จากตารางที่ 2.5 ตาราง SSP เป็นตารางที่อยู่ในนอร์มอลฟอร์มขั้นที่สามเนื่องจากแอททริบิวต์ที่ไม่ใช่แคนดิเดตคีย์ของตารางจะเหลือเพียงแอททริบิวต์เดียวเท่านั้น คือ แอททริบิวต์ QTY โดยมีแคนดิเดตคีย์ของตารางจำนวนสองตัว ได้แก่ กลุ่มของแอททริบิวต์ (S#,P#) และ กลุ่มของแอททริบิวต์ (Sname,P#) ตาราง SSP นี้ไม่เป็นนอร์มอลฟอร์มขั้น Boyce/Codd เนื่องจากแอททริบิวต์ที่เป็นส่วนหนึ่งของแคนดิเดตคีย์ของตารางตัวหนึ่งคือ P# ซ้ำกันอยู่ (overlap)

ตารางที่ 2.6 แสดงตารางที่แบ่งแล้วอยู่ในรูปแบบนอร์มอลฟอร์ม Boyce/Codd

SS	S#	SNAME	SP	S#	P#	QTY
	S1	SMITH		S1	P1	300
	S2	JONES		S1	P2	200
	S3	TOMMY		S1	P3	400
	S4	TOM		S1	P4	200

หรือ

SS	S#	SNAME	SP	SNAME	P#	QTY
	S1	SMITH		SMITH	P1	300
	S2	JONES		SMITH	P2	200
	S3	TOMMY		SMITH	P3	400
	S4	TOM		SMITH	P4	200

ดังนั้นเมื่อทำการพิจารณาแล้วตาราง SSP ไม่เป็นนอร์มอลฟอร์มชั้น Boyce/Codd แต่เป็นนอร์มอลฟอร์มชั้นที่สามดังนั้นจะต้องทำการแบ่งตาราง SSP ออกเป็นตารางย่อยซึ่งจะมีทางเลือกที่จะแบ่งได้อยู่สองทางคือทางแรกเมื่อแบ่งแล้วจะได้ตารางที่มีแอททริบิวต์คือ (S#,SNAME) กับ (S#,P#,QTY) หรือทางที่สองจะได้ตารางที่มีแอททริบิวต์คือ (S#,SNAME) และ (SSAME,P#,QTY) เมื่อทำการแบ่งตารางออกเป็นตารางย่อยแล้วจะต้องทำการตรวจสอบดูว่าตารางที่ถูกแบ่งออกมานั้นอยู่ในรูปแบบนอร์มอลฟอร์ม Boyce/Codd หรือไม่

ตารางที่ 2.7 แสดงตารางที่ไม่อยู่ในรูปแบบนอร์มอลฟอร์มชั้นสี่

CTX	COURSE	TEACHER	TEXT
	PHYSICS	Prof. Green	Basic Mechanics
	PHYSICS	Prof. Green	Principles of Optics
	PHYSICS	Prof. Brown	Basic Mechanics
	PHYSICS	Prof. Brown	Principles of Optics
	MATH	Prof. Green	Basic Mechanics
	MATH	Prof. Green	Victor Analysis
	MATH	Prof. Green	Trigonometry

2.3.5 นอร์มอลฟอร์มขั้นสี่(FOUTH NORMAL FORM : 4NF)

ตารางใดก็ตามจะเป็นนอร์มอลฟอร์มขั้นสี่ได้ก็ต่อเมื่อ

- ต้องเป็นตารางฐานข้อมูลที่อยู่ในรูปนอร์มอลฟอร์มขั้น Boyce/Codd และ
- ต้องไม่มี Multivalued dependence (MVD) มากเกินกว่าหนึ่งตัว

จากตารางที่ 2.7 จะเห็นได้ว่าตาราง CTX มีข้อมูลที่เรียกกันว่าเป็นลักษณะของ Multivalued dependence (MVD) นั่นก็คือในแต่ละวิชา (Course) มีอาจารย์สอน (Teacher) ได้หลายท่าน และในแต่ละวิชา (Course) ก็ใช้หนังสือ (Text) ในการสอนหลายเล่ม ในลักษณะของ Multivalued dependence สามารถที่จะเขียนโดยใช้สัญลักษณ์ คือ

R.A ---->----> R.B
R.A ---->----> R.C

จากตาราง CTX ซึ่งมีข้อมูลในลักษณะที่เป็น Multivalued dependence สามารถที่จะนำมาเขียนเป็นสัญลักษณ์ได้ดังนี้ คือ

Course ---->----> Teacher
Course ---->----> Text

ตาราง CTX ที่แสดงในตารางที่ 2.7 นั้นเป็นตารางที่อยู่ในรูปนอร์มอลฟอร์มขั้น Boyce/Codd เพราะมี Candidate key เพียงตัวเดียว แม้ว่าจะเป็น BCNF แต่ถ้าดูตารางจะเห็นว่ามีความสัมพันธ์ซ้ำกันมาก เช่น Physics กับ Prof. Green, Physics กับ Prof. Brown ถ้ามีการเพิ่มข้อมูล Prof. White สอน Physics จะต้องทำการเพิ่มข้อมูลถึงสองท่ปเปิลคือ (Physics, Prof. White, Basic Mechanics) และ (Physics, Prof. White, Principle of Optics) แล้วถ้าต้องการลบข้อมูล Prof. Green ที่สอน Math ออกจะต้องทำการลบถึงสามท่ปเปิล นั้นแสดงว่า การปรับปรุงข้อมูลขึ้นอยู่กับ Text แบบนี้เรียกว่ามี M:N หรือ 1:N ซ่อนอยู่ได้คีย เช่น อาจารย์หนึ่งคนใช้หนังสือหลายเล่ม หรือ หนังสือหนึ่งเล่มอาจใช้โดยอาจารย์หลายคนก็ได้ นั่นคือ M:N

ตารางที่ 2.8 แสดงตารางที่อยู่ในนอร์มอลฟอร์มขั้นสี่

CT	COURSE	TEACHER	CX	COURSE	TEXT
	PHYSICS	Prof. Green		PHYSICS	Basic Mechanics
	PHYSICS	Prof. Brown		PHYSICS	Principles of Optics
	MATH	Prof. Green		MATH	Basic Mechanics
				MATH	Victor Analysis
				MATH	Trigonometry

ดังนั้นถ้ามี MVD เกิดขึ้นมากกว่าหนึ่งในตารางเดียวกัน จะต้องทำการกำจัด MVD หรือ ความซ้ำซ้อนของตารางนั้นออกไป โดยวิธีการแบ่งตารางออกเป็นตารางย่อย ดังจะแสดงในตารางที่ 2.8 ซึ่งจะได้ออกมาสองตารางคือ ตาราง CT กับตาราง CX ซึ่งทั้งสองตารางจะอยู่ในนอร์มอลฟอร์มขั้นสี่

2.3.6 นอร์มอลฟอร์มขั้นห้า(FIFTH NORMAL FORM : 5NF)

ตารางใดก็ตามจะเป็นนอร์มอลฟอร์มขั้นห้าได้ก็ต่อเมื่อ

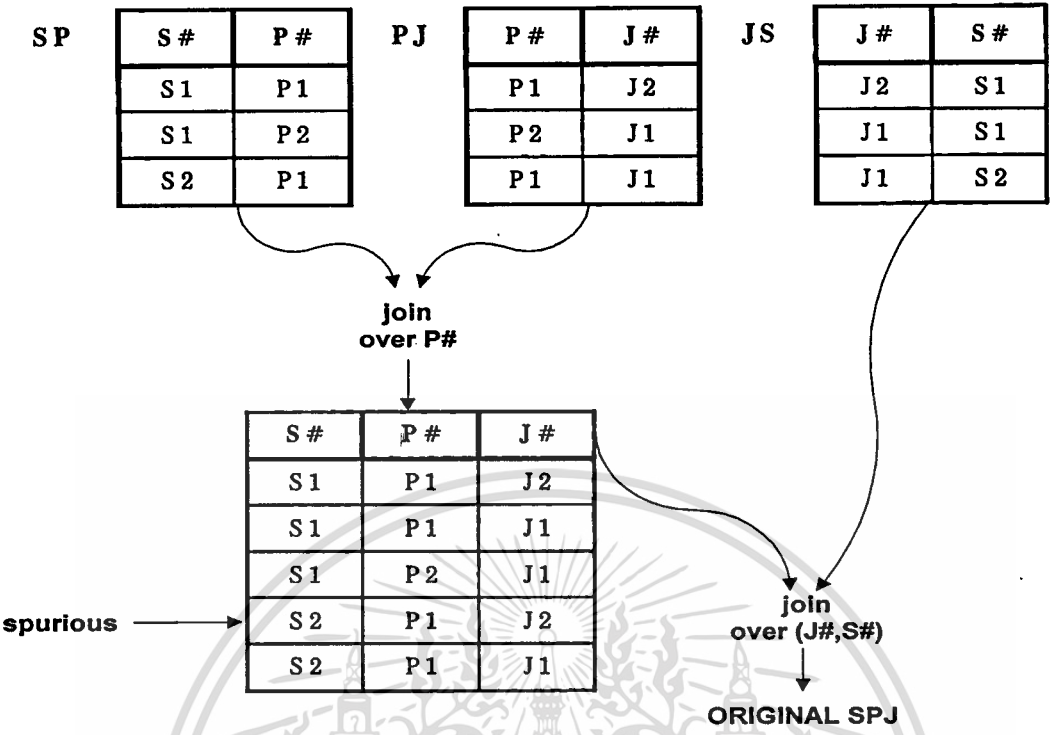
- ต้องเป็นตารางฐานข้อมูลที่อยู่ในนอร์มอลฟอร์มขั้นสี่และ
- ผลลัพธ์ของการ Join ของทุกๆ Join dependency ต้องเหมือนกับตารางฐานข้อมูลเดิมก่อนที่จะทำการแบ่งตาราง หรือข้อมูลจะต้องไม่การสูญหาย(Lossless Join)

จากตารางที่ 2.9 จะเห็นได้ว่าตารางข้อมูล SPJ อยู่ในนอร์มอลฟอร์มขั้นสี่แล้ว เพราะมีคุณสมบัติก็คือไม่มี MVD มากกว่าหนึ่งค่า แต่ตารางฐานข้อมูลที่อยู่ในนอร์มอลฟอร์มขั้นสี่ไม่จำเป็นที่จะเป็นหรืออยู่ในนอร์มอลฟอร์มขั้นห้าเสมอไป จะต้องทำการตรวจสอบดูว่าสามารถที่จะแบ่งเป็นตารางย่อยได้หรือเปล่าและถ้าแบ่งได้เมื่อทำการ Join กับแล้วได้เป็นตารางเดิมหรือไม่ถ้าได้เป็นตารางเดิมก็ทำการแบ่งตารางนั้นออกมา แต่มีข้อแม้ว่าถ้าทำการแบ่งแล้วมีค่าตารางที่แบ่งติดค่า Candidate key มาตลอด ก็ไม่สมควรที่จะแบ่งให้เก็บตารางใหญ่ไว้และถือว่าตารางนั้น

ตารางที่ 2.9 แสดงตารางที่ไม่อยู่ในนอร์มอลฟอร์มขั้นห้า

SPJ	S#	P#	J#
	S1	P1	J2
	S1	P2	J1
	S2	P1	J1
	S1	P1	J1

อยู่ในรูปแบบนอร์มอลฟอร์มขั้นที่ห้าแล้ว สาเหตุที่ไม่ต้องแบ่งก็คือถ้าทำการแบ่งแล้ว จะทำให้มีจำนวนของตารางมากขึ้นและถ้าต้องการข้อมูลจะต้องมีการ Join กันมากไม่ตรงกับความต้องการ แต่จากตารางข้อมูล SPJ ที่แสดงในตารางที่ 2.9 ไม่สามารถที่จะแบ่งตารางออกเป็นสองตารางย่อยได้ก็เนื่องมาจากเมื่อทำการแบ่งแล้วทำการ Join กับมีข้อมูลเกินมา แต่สามารถที่จะแบ่งเป็นตารางย่อยได้ 3 ตารางดังภาพที่ 2.4 ซึ่งแสดงการแบ่งตารางข้อมูล SPJ ออกเป็นสามตารางและแสดงผลของการ Join กับสามารถที่จะได้เป็นตารางข้อมูลเดิม ก็ให้ทำการแบ่ง แสดงให้เห็นว่าการแบ่งตารางหนึ่งตารางไม่จำเป็นที่จะต้องแบ่งได้สองตารางเสมอไป แต่สามารถที่จะแบ่งได้เป็น N ตาราง โดยที่ N มีค่ามากกว่าสอง แต่ในทางความเป็นจริงแล้วจะหาตารางในลักษณะนี้ยากมาก ลองมาดูลักษณะการ Join ของตารางข้อมูล SPJ ที่แบ่งออกมาเป็น ตารางข้อมูล SP ตารางข้อมูล PJ และตารางข้อมูล JS สามารถที่อธิบายได้ดังนี้



ภาพที่ 2.4 แสดงตารางข้อมูลที่อยู่ในนอร์มอลฟอร์มขั้นห้า

ถ้ามีข้อมูล (S1,P1) ปรากฏในตาราง SP และมีข้อมูล (P1,J1) ปรากฏในตาราง PJ และมีข้อมูล (J1,S1) ปรากฏในตาราง JS แล้วจะต้องมีข้อมูล (S1,P1,J1) ปรากฏในตาราง SPJ หรือกล่าวอีกนัยหนึ่งว่า ถ้ามีข้อมูล (S1,P1,J2), (S2,P1,J1), (S1,P2,J1) ปรากฏในตาราง SPJ แล้วจะต้องมี (S1,P1,J1) ปรากฏอยู่ในตาราง SPJ

จากตาราง SPJ ถ้าต้องการที่จะแทรกข้อมูล (S2,P1,J1) จะต้องทำการแทรกข้อมูล (S1,P1,J1) ด้วยเช่นกัน และถ้าต้องการที่จะลบข้อมูล (S2,P1,J1) จะไม่ผลกระทบใดๆ กับตาราง SPJ แต่ถ้าทำการลบข้อมูล (S1,P1,J1) ข้อมูลอื่นจะต้องถูกลบทิ้งด้วย จะเห็นได้ว่าการแทรกข้อมูลและการแก้ไขข้อมูลของตาราง SPJ จะมีความยุ่งยากมากและเกิดความผิดพลาดได้ง่าย ดังนั้นเมื่อแยกตาราง SPJ ออกเป็นสามตารางแล้ว ปัญหาต่างๆ ที่เกิดขึ้นจะหมดไป ซึ่งตารางทั้งสามตารางที่เกิดขึ้นเป็นตารางที่อยู่ในนอร์มอลฟอร์มขั้นห้า เพราะการ Join กันของตารางทั้งสามไม่ทำให้มีข้อมูลส่วนเกินและสูญหายจากตารางเดิมเลย

2.4 กฎ 12 ข้อของโมเดลเชิงสัมพันธ์

นับตั้งแต่ Codd ได้เสนอผลงานทางวิชาการเกี่ยวกับระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งพัฒนามาจากพื้นฐานทางคณิตศาสตร์ของเซตและตรรกศาสตร์ หลังจากนั้นนักวิชาการคอมพิวเตอร์หรือผู้ค้าคอมพิวเตอร์ต่างๆ ก็เริ่มหันเหความสนใจมาพัฒนาเกี่ยวกับโมเดลนี้ จนระบบฐานข้อมูลโมเดลเชิงสัมพันธ์มีประสิทธิภาพมากยิ่งขึ้นและมีผู้นิยมใช้กันอย่างแพร่หลาย โดย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มจากการใช้งานกับเครื่องระดับเมนเฟรม จนถึงปัจจุบันได้แพร่หลายลงมาใช้กับเครื่องระดับไมโครด้วย เนื่องจากบริษัทผู้ค้ามากมายต่างแข่งขันกันผลิตระบบจัดการฐานข้อมูลชนิดนี้ออกมาสู่ตลาด โดยที่ต่างก็แสดงความสามารถของผลิตภัณฑ์ของตนว่ามีประสิทธิภาพใช้งานง่ายเป็นมาตรฐาน และมีจุดเด่นอื่นๆ เพิ่มมากขึ้น เพื่ออำนวยความสะดวกในการใช้ ซึ่งถ้าพิจารณาตามหลักเกณฑ์แต่ละตัว อาจขาดฟังก์ชันบางอย่างและกลบเกลื่อนด้วยฟังก์ชันใหม่ที่เพิ่มเติมเข้ามา นั่นคือ ผลิตภัณฑ์ต่างๆ นั้น จะมีความเป็นไปได้ตามทฤษฎีโมเดลเชิงสัมพันธ์มากน้อยแตกต่างกัน ถึงอย่างไรก็ตามผลิตภัณฑ์ต่างๆ เหล่านั้น ก็นับได้ว่ายังมีส่วนประกอบของความเป็นมาตรฐานอยู่

เพื่อพิจารณาระบบจัดการฐานข้อมูลเชิงสัมพันธ์นั้นว่า มีความสมบูรณ์ตามทฤษฎีโมเดลเชิงสัมพันธ์มากน้อยเพียงใด Codd จึงได้สรุปกฎเกณฑ์ในการพิจารณาไว้ 12 ข้อ โดยกล่าวว่าระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่สมบูรณ์จะต้องมีคุณสมบัติครบ 12 ข้อ ดังนี้

2.4.1 กฎข่าวสาร (Information Rule)

ข้อมูลทั้งหมดในฐานข้อมูลเชิงสัมพันธ์ จะถูกแสดงในระดับตรรกะ (Logical Level) ด้วยวิธีการเดียวกันเท่านั้น โดยอยู่ในรูปของตารางซึ่งประกอบด้วย ชื่อตาราง ชื่อคอลัมน์ ชนิดของข้อมูลในแต่ละคอลัมน์ อาจจะมีบางคอลัมน์เป็นคีย์ (Key) หรือเป็นอินเด็กซ์ (Index) และมีขอบเขตในแต่ละคอลัมน์ ที่เรียกว่า โดเมน (Domain) เป็นต้น

2.4.2 กฎการประกันการเข้าถึงข้อมูล (Guaranteed Access Rule)

ในแต่ละและทุก ๆ ค่าของข้อมูลในฐานข้อมูลเชิงสัมพันธ์ จะต้องรับประกันได้ว่าผู้ใช้สามารถที่จะเข้าถึงข้อมูลทุกตัวในทุกตารางได้ โดยระบุชื่อตาราง, ค่าคีย์หลัก (Primary Key), ชื่อคอลัมน์ และจะได้ผลลัพธ์เพียงค่าเดียวเท่านั้น

2.4.3 วิธีแสดงข้อมูลบางรายการที่ยังไม่พร้อมหรือยังไม่มีค่า (Systematic Treatment of Null Value)

นั่นคือ จะมีค่านัล (Null) ซึ่งเป็นตัวบอกความแตกต่างระหว่างข้อมูลที่ยังไม่พร้อม (Empty Character String) หรือข้อมูลที่ยังไม่มีค่า (String of Blank Character) หรือ ค่าศูนย์ ซึ่งความแตกต่างนี้จะไม่ทำให้เกิดความเข้าใจผิดในการใช้

2.4.4 กฎโครงสร้างแคตตาล็อกระบบจะต้องมีสภาพโมเดลเชิงสัมพันธ์ (Dynamic On-line Catalog must be based on The Relation Model)

โครงสร้างแคตตาล็อกจะต้องมีสภาพของโมเดลเชิงสัมพันธ์ คือ อยู่ในรูปตารางเช่นเดียวกับข้อมูลทั้งหมด ก็จะทำให้ผู้ใช้สามารถเรียกดูหรือแก้ไขโครงสร้างต่างๆ ได้ เช่นเดียวกับการเรียกใช้ข้อมูลทั่วไป

2.4.5 กฎการใช้ภาษา (Comprehensive Data Sublanguage Rule)

ระบบฐานข้อมูลเชิงสัมพันธ์ อาจจะมีภาษาที่ใช้หลายภาษาหรือมีลักษณะการติดต่อการใช้ อยู่หลายวิธีการได้น้อยที่สุด จะต้อง มี 1 ภาษา ที่มีความสามารถ ดังนี้

- นิยามโครงสร้างของข้อมูล (Data Definitions)
- นิยามโครงสร้างของวิว (View Definitions)
- เปลี่ยนแปลงหรือแก้ไขข้อมูลทั้งลักษณะโต้ตอบทันที (Interactive) หรือการเขียนโปรแกรม
- กำหนดกฎเกณฑ์ควบคุม Integrity ของข้อมูล (Integrity Constraints)
- กำหนดสิทธิการใช้ข้อมูล (Authorization) เป็นต้น

2.4.6 กฎการแก้ไขข้อมูลผ่านทางวิว (View Updating)

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ จะต้องสามารถตัดสินใจได้ว่าผู้ใช้สามารถเพิ่ม, แก้ไขข้อมูลโดยกระทำผ่านทางวิวได้หรือไม่

2.4.7 ความสามารถในการเพิ่ม ลด และแก้ไขข้อมูล (High-level Insert, Update and Delete)

ระบบจัดการฐานข้อมูล ควรจะมีการรักษาที่ความสามารถให้ผู้ใช้เพิ่ม ลด หรือแก้ไขข้อมูลในหลายๆ แถวได้ด้วยคำสั่งเพียงคำสั่งเดียว

2.4.8 ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical Data Independence)

ผู้ใช้ไม่จำเป็นต้องรับรู้ข้อมูลจริงจะมีการจัดเก็บอย่างไร แม้ว่าจะมีการเปลี่ยนแปลงการจัดเก็บข้อมูลอย่างไร ก็จะไม่มีผลกระทบต่องานของผู้ใช้แต่อย่างใด

2.4.9 ความเป็นอิสระของข้อมูลในระดับตรรก (Logical Data Independence)

แม้ว่าจะมีการเปลี่ยนแปลงโครงสร้างข้อมูล เช่น การเพิ่มคอลัมน์ในตาราง หรือการสร้างตารางใหม่ เป็นต้น การเปลี่ยนแปลงเหล่านั้น จะไม่มีผลกระทบต่อระบบงานของผู้ใช้หรือโปรแกรมต่างๆ ที่เขียนไว้ก่อน

2.4.10 ความเป็นอิสระของความคงสภาพ (Integrity Independence)

การกำหนดกฎเกณฑ์การควบคุมความคงสภาพของข้อมูล (Integrity) ควรจะเป็นส่วนหนึ่งของภาษาที่ใช้ในระบบจัดการข้อมูลเชิงสัมพันธ์และเก็บรวบรวมไว้ในระบบแคตตาล็อก ซึ่งเป็นอิสระจากโปรแกรม คือ โปรแกรมไม่ต้องเป็นตัวจัดการตรวจสอบ ถ้ายอมให้โปรแกรมตรวจสอบแทน แล้วหากแต่ละโปรแกรมทำงานไม่ครบถ้วนตามกฎเกณฑ์ก็จะเกิดความไม่สอดคล้องหรือความไม่ถูกต้องของข้อมูล

2.4.11 ความเป็นอิสระของการกระจาย (Distribution Independence)

ผู้ใช้ไม่ต้องสนใจว่าข้อมูลจะเก็บอยู่ส่วนใด หรือโยกย้ายไปยังคอมพิวเตอร์เครื่องอื่นที่เชื่อมโยงกันเป็นระบบ

2.4.12 การไม่ยอมให้ภาษาอื่นทำลายกฎ (Nonsubversion Rule)

ระบบจัดการฐานข้อมูล จะไม่ยอมให้ภาษาอื่นใด ไม่ว่าจะเป็นภาษาระดับล่าง ก็ตามไม่สามารถทำลายกฎต่างๆ ที่กำหนดไว้ในระบบฐานข้อมูล โดยเฉพาะกฎเกณฑ์ความคงสภาพของข้อมูล

2.5 ประโยชน์จากการใช้ระบบฐานข้อมูล

- ลดความซ้ำซ้อนของข้อมูล (Redundancy can be reduce) การประมวลผลโดยใช้ไฟล์ธรรมดานั้น จำเป็นที่ผู้ใช้แต่ละกลุ่มต้องมีไฟล์ส่วนตัวเอาไว้ดังนั้นจึงเกิดเหตุการณ์ที่ข้อมูลชนิดเดียวกันถูกเก็บไว้หลาย ๆ แห่ง การนำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกันในฐานข้อมูลเป็นการลดความซ้ำซ้อนลงไปได้
- หลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง (Inconsistency can be avoided) ประโยชน์ในข้อนี้สืบเนื่องมาจากข้อที่แล้วคือ การเก็บข้อมูลไว้ในหลาย ๆ แห่งอาจก่อให้เกิดปัญหา ข้อมูลชุดเดียวกันอาจมีค่าในแต่ละแห่งไม่เหมือนกันดังนั้นการใช้ระบบฐานข้อมูลทำให้สามารถที่จะลดความขัดแย้งของข้อมูลลงได้ โดยมี DBMS เป็นตัวคอยควบคุมดูแลว่าเมื่อเกิดการแก้ไขข้อมูลขึ้นมาเมื่อใด จะต้องแก้ไขให้เหมือนกันให้ครบ
- ใช้ข้อมูลร่วมกันได้ (Data can be shared) การใช้ข้อมูลร่วมกันได้นี้ไม่จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้นแต่จะครอบคลุมถึงโปรแกรมที่จะถูกพัฒนาขึ้นมาใหม่อีกด้วย ที่จะสามารถใช้ข้อมูลที่มีอยู่ได้เลย โดยไม่เป็นที่จะต้องเสียเวลาในการเตรียมข้อมูลชุดใหม่ถ้าข้อมูลนั้นมีอยู่แล้ว
- ควบคุมความมาตรฐานได้ (Standards can be enforced) จากการที่เรานำข้อมูลมาเก็บไว้รวมกันในระบบฐานข้อมูลเช่นนี้ ทำให้ผู้ที่มีหน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูลสามารถที่กำหนดมาตรฐานของข้อมูลขึ้นมาได้ ทำให้ข้อมูลเป็นมาตรฐานเดียวกันทั้งหมด ทำให้การแลกเปลี่ยนข้อมูลระหว่างระบบงานเป็นไปอย่างสะดวกและถูกต้อง
- ระบบความปลอดภัย (Security restrictions can be applied) ระบบรักษาความปลอดภัยในที่นี้หมายถึงระบบป้องกันไม่ให้ผู้ที่ไม่มีสิทธิใช้ข้อมูลเข้ามาใช้หรือกระทำการเปลี่ยนแปลงข้อมูล โดยระบบรักษาความปลอดภัยนี้ผู้ที่ควบคุมสามารถที่จะกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือให้สิทธิแก่ผู้ใช้แต่ละคนไม่เท่ากัน ขึ้นอยู่ความจำเป็นในการการใช้ข้อมูลของผู้ใช้แต่ละคน

- ควบคุมความคงสภาพของข้อมูลได้(Integrity can be maintained) ความคงสภาพของข้อมูลกับการใช้ข้อมูลร่วมกันมีความสำคัญมาก เมื่อเทียบการใช้ไฟล์ข้อมูลส่วนตัวคนเดียวเพราะการที่ผู้ใช้หลายคนทำให้เกิดโอกาสที่ผู้ใช้คนใดคนหนึ่งจะเกิดการผิดพลาดจากการแก้ไขข้อมูลผิดหรือทำเพิ่มข้อมูลที่ไม่เป็นจริงเข้าไปในระบบฐานข้อมูล ข้อมูลเหล่านั้นจะกระทบต่อระบบงานทั้งระบบได้ ดังนั้นในระบบฐานข้อมูลที่ดี DBMS จะต้องสามารถที่จะให้ผู้ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์เพื่อควบคุมการคงสภาพได้
- สร้างสมดุลของความต้องการได้(Requirements can be balanced) การที่ผู้ใช้ทั้งหมดขององค์กรใช้ระบบฐานข้อมูลเดียวกันเช่นนี้ ทำให้ DBA ทราบถึงความต้องการและความสำคัญของของผู้ใช้งานทั้งหมด จึงสามารถที่จะกำหนดโครงสร้างของฐานข้อมูลเพื่อให้บริการแก่ผู้ใช้ให้ดีที่สุดได้ เพราะการออกแบบนั้นกระทำไปในแนวทางที่จะมุ่งให้ประโยชน์แก่ส่วนรวมดีที่สุด



บทที่ 3

ทฤษฎี CASE*METHOD

เป็นที่ทราบกันดีว่าในการพัฒนาระบบ ไม่ว่าจะเป็นการจัดทำระบบขึ้นใหม่หรือปรับปรุงระบบงานที่มีอยู่เดิมให้ประสบผลสำเร็จได้นั้น ความยากอยู่ที่การทำความเข้าใจกันระหว่างฝ่ายผู้พัฒนาระบบกับฝ่ายผู้ใช้งานหรือองค์กร ทั้งความเข้าใจในกิจกรรมต่างๆ ที่ดำเนินอยู่ ในขณะที่ต้องตระหนักถึงเป้าหมายหรือหัวใจทางธุรกิจขององค์กร อันจะทำให้ทุกฝ่ายสามารถมองเห็นถึงระบบที่ต้องการจริง ๆ ในที่สุด เนื้อหาในส่วนนี้จะกล่าวถึงหลาย ๆ เทคนิควิธีที่จะช่วยในการดังกล่าวโดยเป็นการจัดทำแผนภาพอธิบายว่าภายในองค์กรมีการทำงานใด ๆ อย่างเป็นระบบ อันจะช่วยลดโอกาสที่จะเกิดความคลุมเครือในสิ่งที่นำเสนอ ทำให้มั่นใจในการสื่อสารว่ามีความหมายในสิ่งที่ตรงกัน ทั้งยังมีกระบวนการที่สามารถตรวจสอบความถูกต้องในการใช้เทคนิคเหล่านี้

วิธีการนำเสนอนี้มีมุมมองอยู่ 3 ระดับด้วยกันคือ ระดับ Business, System และ Program โดยในแต่ละระดับมีจุดแตกต่างกันคือ ระดับ Business นั้นมุ่งที่จะนำเสนอภาพกิจกรรมต่างๆ ที่องค์กรนั้นดำเนินอยู่หรือจำเป็นต้องดำเนินการเพื่อให้บรรลุเป้าหมายทางธุรกิจและเป็นไปตามนโยบายที่ฝ่ายบริหารได้วางไว้ ผู้บริหารองค์กรจะมีบทบาทต่องานชิ้นนี้อย่างยิ่ง ส่วนที่กิจกรรมต่างๆ เหล่านี้จะนำมาจัดทำเป็นระบบอะไรด้วยวิธีการเช่นไรบ้างนั้นจะเป็นในส่วนของระดับ System ซึ่งจะเป็นการมองในเชิง Logical และในระดับนี้เองที่เราจะมองเห็นได้ชัดในภาพต่างๆ ขององค์กรต่างองค์กร สำหรับในส่วนในระดับ Program หรือ Procedure นั้นจะเป็นอย่าง System โดยจะมองในเชิง Physical เช่นระบบชี้ชัดว่างานใดใช้พนักงานลงมือทำหรือจะใช้คอมพิวเตอร์หรือเครื่องมืออื่นช่วย ปัจจัยที่จะเลือกใช้เครื่องมือใดนั้นได้แก่ ราคา หามาใช้ได้ไม่ยาก ประยุกต์เข้ากับปัญหาได้ เป็นเทคโนโลยีที่ดี ใช้งานง่าย สามารถที่จะรองรับเทคโนโลยีในอนาคตได้ เป็นต้น หลายๆ เทคนิคที่จะได้กล่าวต่อไปสามารถใช้ได้กับทั้ง 3 ระดับที่กล่าวมาโดยแสดงด้วยสัญลักษณ์ที่ต่างกัน

โดยในส่วนของ CASE*METHOD นั้นได้กล่าวถึงส่วนประกอบหลัก ๆ ซึ่งจะขออธิบายไว้พอสังเขป คือ

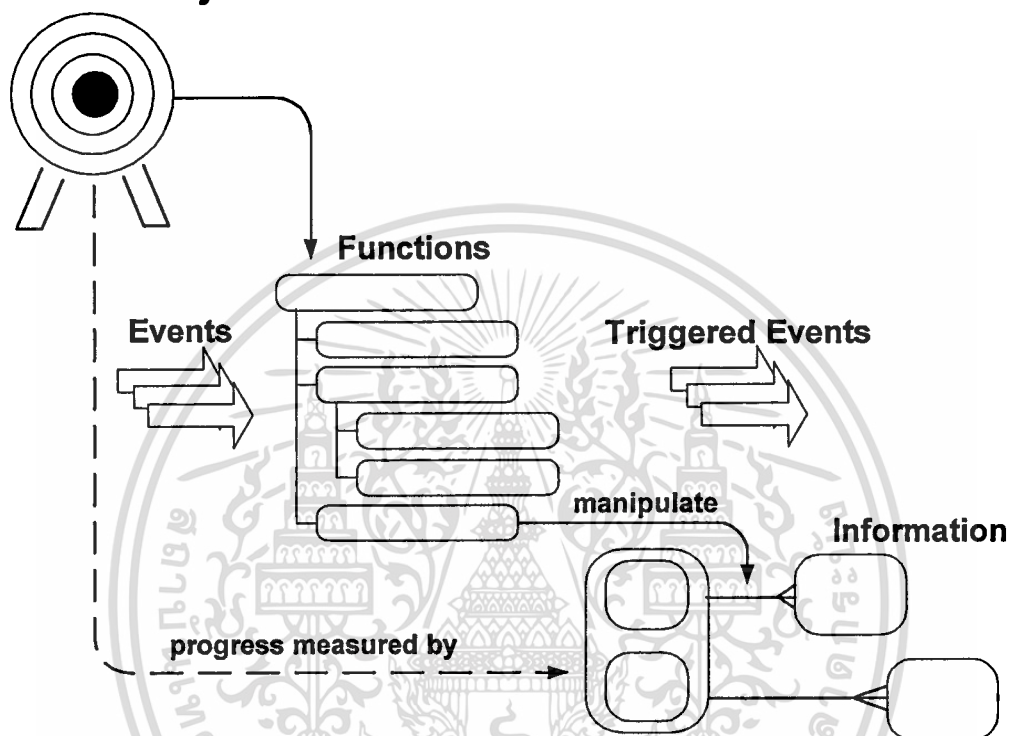
3.1 ฟังก์ชันโมเดล(Function Model)

ตามหลักการของการออกแบบและพัฒนาระบบให้กับองค์กรหรือหน่วยงานหนึ่ง ๆ นั้น ขั้นตอนแรกคือการรวบรวมข้อมูล ความสัมพันธ์ และรายละเอียดต่างๆ ที่เกี่ยวข้องกับระบบที่จะทำการออกแบบหรือพัฒนา เพื่อที่จะนำข้อมูลต่างๆ ที่รวบรวมมาได้นำมาวิเคราะห์และทำความเข้าใจว่าในองค์กรหรือหน่วยงานนั้นมีกิจกรรมอะไรบ้างที่จะต้องดำเนินการเพื่อที่จะให้งานในองค์กรหรือหน่วยงานสำเร็จตามวัตถุประสงค์และเป้าหมายที่วางไว้ เมื่อทำความเข้าใจได้ทั้งหมดแล้วก็ต้องทำการรายการของกิจกรรมทั้งหมดที่จะต้องทำออกมาว่ามีกิจกรรมอะไรบ้าง เราจะเรียก

กิจกรรมเหล่านั้นว่า Business Function ซึ่งกิจกรรมต่างๆ นี้ถือได้ว่าเป็นเป้าหมายหรือวัตถุประสงค์ของการดำเนินงานนั่นเอง การดำเนินการพัฒนาระบบให้สอดคล้องกับความต้องการขององค์กรนั้น ไม่ใช่ว่าจะดำเนินการตามขั้นตอนที่กำหนดไว้ตายตัวเสมอไป เพราะในบางครั้งอาจมีเหตุจำเป็นที่ต้องปรับเปลี่ยนหรือเพิ่ม/ลดขั้นตอนการทำงานได้ ซึ่งการดำเนินการเช่นนี้จะต้องมีการพิจารณาถึงผลกระทบต่อระบบโดยรวม และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสงค์ขององค์กรที่สามารถวัดได้ว่าองค์กรหรือหน่วยงานนั้นๆ สามารถทำงานได้บรรลุวัตถุประสงค์หรือไม่ หรือสามารถที่จะให้คำจำกัดง่ายๆ ของ Business Function ได้ว่า “Business Function คือ สิ่งที่จะคิดว่าจะทำหรือสิ่งที่ต้องการทำในอนาคต เพื่อที่จะช่วยให้องค์กรหรือหน่วยงานสามารถดำเนินงานได้ลุล่วงตามวัตถุประสงค์ที่ตั้งไว้”

Aim and Objectives



ภาพที่ 3.1 Business Model

ในขณะจัดทำ Business Function จะต้องตระหนักว่าแต่ละฟังก์ชันเราต้องการที่จะระบุเพียงแค่ชื่อของฟังก์ชันหรือกิจกรรมเท่านั้นว่าจะต้องมีการทำอะไรบ้าง โดยไม่ต้องมีการอธิบายว่าแต่ละฟังก์ชันนั้นมีวิธีการทำงานอย่างไร ดังนั้นในชั้นของ Business Level จะอธิบายแต่ละฟังก์ชันไว้โดยอธิบายแต่สิ่งที่ต้องการจะทำเท่านั้น ให้ละเว้นที่จะระบุถึงวิธีการ (Independence from Mechanism) การที่ไม่ต้องระบุวิธีการทำงานของฟังก์ชันหรือกิจกรรมแต่ละอย่างว่ามีการทำงานอย่างไรมีประโยชน์คือ ทำให้ไม่มีการจำกัดวิธีการทำงานที่ตายตัวไว้ เพื่อการมีอิสระของผู้ออกแบบและพัฒนาระบบที่จะได้ไม่ต้องยึดติดกับวิธีการทำงานนี้เสมอไป และเป็นการให้อิสระทางด้านการแสดงแนวคิดต่างๆ ที่สามารถจะมาทำงานให้กิจกรรมนั้นบรรลุวัตถุประสงค์ได้

การเขียนและสัญลักษณ์ที่ใช้ในการเขียน Function Model มีดังนี้ คือ Business Function จะแสดงในอยู่ในกรอบสี่เหลี่ยมมุมมน (softbox) และ short name หรือ label จะแสดงอยู่ที่นอกกรอบมุมมนด้านซ้าย และด้านนอกกรอบมุมมนด้านขวามีสัญลักษณ์ (...) แสดงว่าภายใน Business Function นั้นๆ มีฟังก์ชันที่ซ่อนอยู่อีกหรือกล่าวอีกนัยหนึ่งได้ว่ามีฟังก์ชันย่อยอยู่ข้างในนั่นเอง และที่สำคัญอีกอย่างคือในการ Function Model จะต้องไม่คำนึงถึงลำดับการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R2

Establish scheduled flights best meet passenger requirements

ภาพที่ 3.2 ภาพแทน Business Function

3.2 ฟังก์ชันไฮราคี (Function Hierarchy)

Function Hierarchy เป็นเทคนิคพื้นฐานที่มีผู้นิยมใช้มากในการเขียน Business Function โดยการนำฟังก์ชันมาเขียนไล่เป็นลำดับชั้น ซึ่งแต่ละฟังก์ชันจะมีหลักการตั้งชื่อคือ ชื่อของฟังก์ชันจะต้องบ่งบอกถึงคำจำกัดความว่าฟังก์ชันนั้นมีหน้าที่ทำอะไรให้ชัดเจน โดยจะต้องไม่บ่งบอกหรือระบุถึงวิธีการปฏิบัติและคำขึ้นต้นของชื่อฟังก์ชันจะต้องขึ้นต้นด้วยคำกริยา โดยแต่ละฟังก์ชันจะเป็นส่วนหนึ่งของโครงสร้างขององค์กรหรือหน่วยงาน ฟังก์ชันที่อยู่ชั้นบนจะถูกแตกหรือถูกอธิบายโดยฟังก์ชันย่อยในลำดับชั้นถัดลงมา จนกระทั่ง Hierarchy สมบูรณ์และมีความครอบคลุมทำให้การทำงานเข้าใจได้ง่าย การแตกย่อยฟังก์ชันก็จะสิ้นสุด ในการแตกฟังก์ชันย่อยนี้ลำดับที่ปรากฏของฟังก์ชันไม่ใช่สิ่งที่สำคัญดังแสดงในภาพที่ 3.3 ด้วยเทคนิคนี้มุ่งที่จะให้เห็นภาพความครอบคลุมของฟังก์ชันใด ๆ มากกว่า ส่วนเรื่องของลำดับต่อเนื่องของการเกิดฟังก์ชันนั้นจะอยู่ในเทคนิคถัดไป เทคนิคนี้ใช้ได้กับทั้งระดับ Business, System และ Program

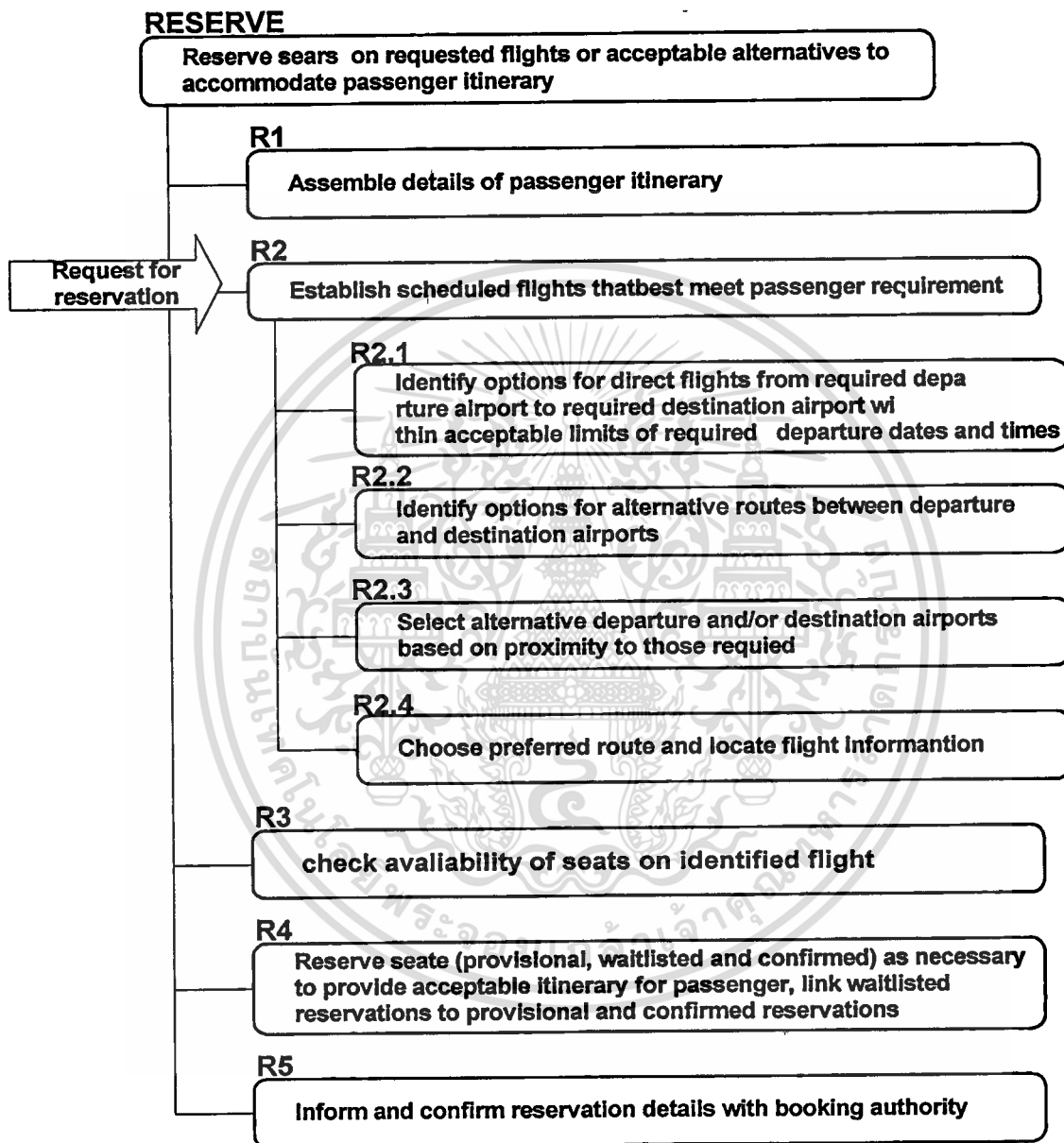
สัญลักษณ์ที่ใช้แสดง Business Function เป็นรูปกรอบสี่เหลี่ยมที่มีมุมขอบมน ภายในกรอบคือรายละเอียดของฟังก์ชัน ที่มุมซ้ายบนแสดงชื่อสั้น ๆ ไว้เป็นเลเบล (Label) และอาจมีเครื่องหมายจุดประ (...) ที่มุมขวาบนเพื่อแสดงว่ายังสามารถแตกรายละเอียดฟังก์ชันนั้นออกเป็นฟังก์ชันย่อย ๆ ลงไปได้อีก แต่รายละเอียดไม่ได้ปรากฏให้เห็นบนแผนภาพในขณะนั้น การเลือกใช้ชื่อฟังก์ชันควรเลือกถ้อยคำให้กระชับแต่ชัดเจนครอบคลุมใจความและเงื่อนไขที่ต้องเป็นไป ผู้จัดทำควรเลี่ยงการใช้คำศัพท์เฉพาะทาง อักษรย่อและคำย่อที่อาจสื่อความกำกวม

อีกองค์ประกอบหนึ่งที่แสดงอยู่ใน Model คือ Event จะแสดงด้วยสัญลักษณ์ลูกศรที่มีข้อความภายในลูกศร Event คือเหตุการณ์ที่อาจเป็นเหตุการณ์ผลลัพธ์ (Key Result) เมื่อเกิดฟังก์ชันใด ๆ หรือเป็นเหตุการณ์ใด ๆ ที่กระตุ้น (Trigger Event) ให้แต่ละฟังก์ชันเกิดขึ้น เราสามารถแบ่ง Event ออกได้เป็น 4 ประเภท ได้แก่ External Event คือเหตุการณ์ที่ภายนอกที่มีผลกระทบ เช่น เกิดภัยธรรมชาติรุนแรงจนทำให้มหาวิทยาลัยและแผนกหรือหน่วยงานที่เกี่ยวข้องต้องปิดชั่วคราว การปิดบริการชั่วคราวนี้จะเลื่อนกำหนดการลงทะเบียนฝึกงานออกไป Change Event คือการเปลี่ยนแปลงบางอย่างภายในองค์กรที่กระตุ้นให้เกิดฟังก์ชันอื่น ๆ เช่น เมื่อมีบางบริษัทบอกยกเลิกที่จะรับนักศึกษาฝึกงานอย่างกะทันหันทางแผนกกิจกรรมจะต้องทำเรื่องแจ้งให้นักศึกษาทราบและปรับปรุงฐานข้อมูลที่เกี่ยวข้องเพื่อให้ข้อมูลบริษัทฝึกงานเป็นไปตามความเป็นจริง Time หรือ Realtime Event คือเหตุการณ์ที่เกิดขึ้นเมื่อถึงจุดหนึ่งของเวลา เช่น เมื่อถึงช่วงเวลาหนึ่งทางแผนกกิจกรรมจะต้องทำการสรรหาบริษัทที่ต้องการรับนักศึกษาฝึกงาน System พบเหตุการณ์ที่เข้าเงื่อนไขบางอย่าง เช่น นักศึกษาที่จะลงชื่อฝึกงานจะต้องลงทะเบียนฝึกงานก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Event คือ เหตุการณ์สำคัญบางอย่างที่มีผลต่อการควบคุมกระบวนการงานของธุรกิจเกิดขึ้น อย่างการถึงจะมีสิทธิในการลงชื่อฝึกงาน เป็นต้น วิธีการตรวจสอบความถูกต้องของเทคนิคนี้กระทำได้หลายทาง ดังนี้

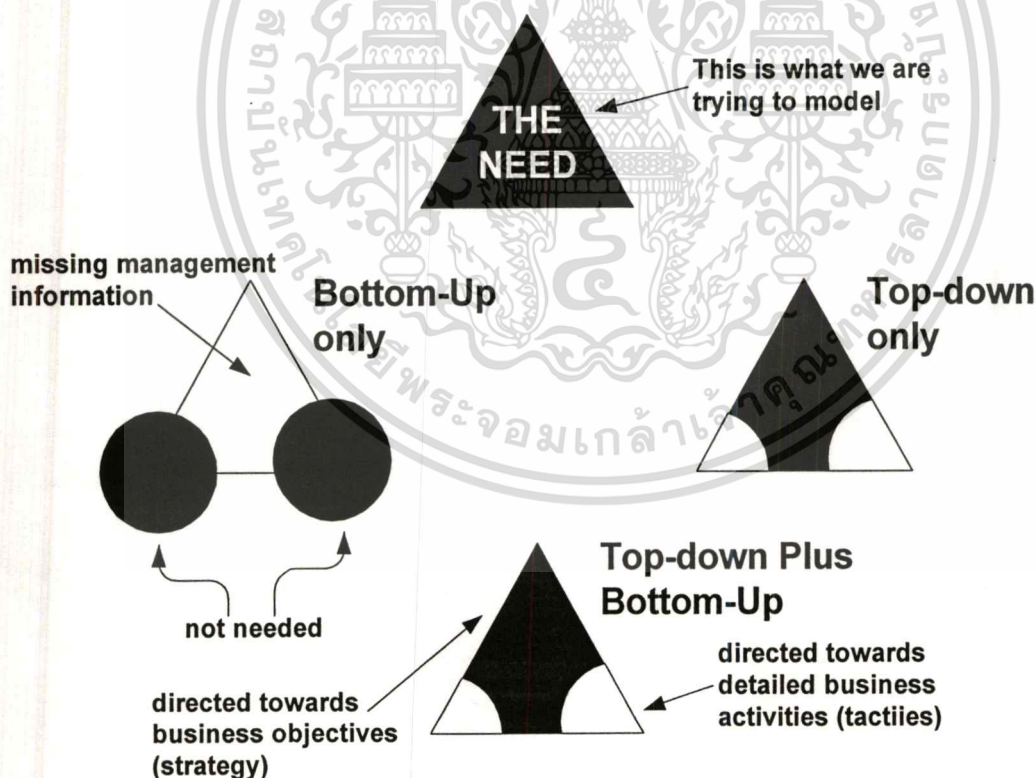


ภาพที่ 3.3 แสดง Function Hierarchy

- Generic Hierarchies คือใช้ความเข้าใจในลักษณะของธุรกิจทั่วไปว่ามีกิจกรรมโดยอยู่บ้าง แล้วมองกลับมาที่ Hierarchy ที่ถูกตรวจสอบว่าน่าจะเกี่ยวข้องกับกิจกรรมในลักษณะเช่นนั้นด้วยหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Entity Life-Cycles เป็นการตรวจสอบโดยทดสอบกับข้อมูลภายในองค์กรแต่ละตัว (Entity) ว่าข้อมูลนั้นถูกใช้อย่างไรบ้างในองค์กร การใช้ข้อมูลนั้นก็คือฟังก์ชันที่ควรมีอยู่ใน Function Hierarchy
- Analogies ทดสอบโดยเปรียบเทียบ Hierarchy ของเรากับหลายๆโมเดลของธุรกิจอื่นที่ผู้ทดสอบต้องมีความเข้าใจในธุรกิจนั้นๆเป็นอย่างดี เช่น งานของที่ทำงานเดิม เป็นต้น
- Event List ที่ได้จากบันทึกการสัมภาษณ์ การสังเกตการณ์กับเอกสารต่างๆ นำมาพิจารณาโดยตั้งคำถามกับแต่ละ event ว่าถ้าเหตุการณ์นี้จะเกิดขึ้นได้ จะต้องมีการฟังก์ชันใดดำเนินมาก่อนบ้าง และถ้าเกิดเหตุการณ์นี้เกิดขึ้นจะมีฟังก์ชันใดเกิดขึ้นบ้าง คำตอบที่ได้เป็นรายการฟังก์ชันที่จะนำมาใช้ตรวจสอบต่อไปว่ามีอยู่ใน Hierarchy ครบถ้วนอย่างเหมาะสมหรือไม่
- Role Playing คือให้ผู้ทดสอบลองสมมติบทบาทต่างๆ ของบุคคลที่เกี่ยวข้องกับธุรกิจ เช่น เป็นสมาชิก บรรณารักษ์ เจ้าหน้าที่ร้านหนังสือ แล้วจินตนาการว่าบทบาทนั้นๆ จะได้ประสบกับเหตุการณ์ใดบ้าง



ภาพที่ 3.4 Top-Down plus Bottom-up

เมื่อทราบว่าในการ Function Hierarchy มีสัญลักษณ์อะไรบ้างเป็นส่วนประกอบ ต่อไปจะขอกล่าวถึงวิธีการหรือเทคนิคที่ใช้เขียน Function Hierarchy ในการเขียน Function Hierarchy เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นมียุทธวิธีในการลักษณะที่เป็นในแบบ Top-Down Modeling และเมื่อเขียนเสร็จจะมีการตรวจสอบตามหลักเกณฑ์ที่กล่าวไปแล้วข้างต้นแต่จะใช้วิธีของ Bottom-Up Modeling มาทำการตรวจสอบ

3.2.1 บนลงล่าง (Top-Down Modeling)

การเขียน Top-Down Modeling มีหลักการที่สำคัญ คือ จะต้องเริ่มเขียนหรือกำหนดส่วนบนสุดของโครงสร้างก่อนนั่นคือ จุดหลักหรือเป้าหมายหลักที่จะทำการเขียน Function Hierarchy และจะมีการอธิบายออกมาในรูปของประโยคและผู้ที่เขียนได้นั้นจะต้องรู้จักและเข้าใจในระบบขององค์กรหรือหน่วยงานนั้นๆ อย่างถ่องแท้ เมื่อได้กำหนดส่วนบนสุดแล้ว อาจจะยังไม่ได้ใจความถ้าต้องการที่จะอธิบายหรือขยายความของฟังก์ชันนั้นก็สามารถที่จะทำได้โดยการเขียนฟังก์ชันในระดับล่างลงมา และควรที่จะเขียน Function Hierarchy เมื่อเขียนเสร็จจะต้องมีลำดับชั้นไม่ควรเกิน 7 - 8 ระดับ ในทางที่ดีไม่ควรเกิน 7 ระดับ ดังภาพที่ 3 - 3

3.2.2 ล่างขึ้นบน (Bottom-Up Modeling)

จากข้อที่คงบอกแล้วว่าจะมีการทำงานจากส่วนล่างขึ้นมาข้างบนนั่นคือหลักการทำงานของ Bottom-Up Modeling จะอาศัยวิธีการทำงานของ Bottom-Up Modeling มาใช้ในการตรวจสอบความถูกต้องว่า Function Hierarchy ที่เขียนขึ้นมานั้นถูกต้องหรือไม่ โดยข้อมูลที่ใช้ในการตรวจสอบนั้นได้มาจาก การสัมภาษณ์, รายละเอียดของธุรกิจ, แหล่งวัตถุดิบ และคู่มือปฏิบัติงาน พร้อมทั้งวิธีการตรวจสอบที่กล่าวมาข้างต้น ซึ่งข้อมูลทุกอย่างที่ได้มาจะต้องเป็นจริง ถ้าเกิดว่ามีเหตุการณ์ที่ว่าเมื่อตรวจสอบแล้วพบว่ามีฟังก์ชันบางฟังก์ชันที่ขาดหายไปก็ให้ทำการเพิ่มเติมเข้าไปได้เลยแล้วก็จะต้องทำการตรวจสอบใหม่อีกครั้งจนกระทั่งคิดว่าสมบูรณ์ที่สุดก็แสดงว่า Function Hierarchy ที่เขียนขึ้นมานั้นตรงตามวัตถุประสงค์หรือเป้าหมายขององค์กรหรือหน่วยงานนั้นๆ แล้วก็ยืนยันว่าสมบูรณ์

มีข้อให้สังเกต 2 ประการว่า Function Hierarchy ที่สร้างได้ดีคือ โมเดลนั้นช่วยให้บุคลากรต่างๆที่เกี่ยวข้องสามารถที่จะมององค์กรในแง่มุมใหม่ๆ ต่างไปจากสิ่งที่เป็นอยู่หรือไม่ โมเดลนั้นถูกจัดโครงสร้างไว้อย่างดีอำนวยความสะดวกต่อการแก้ไขเพิ่มเติมสิ่งใหม่ๆ ต่อไปหรือไม่

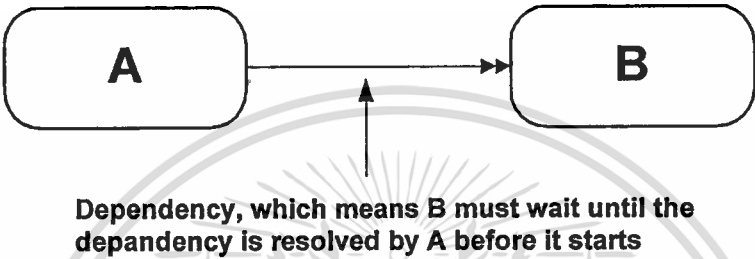
3.3 ฟังก์ชันดีเพนเดนซีไดอะแกรม (Function Dependency Diagram)

เทคนิคนี้ใช้แสดงให้เห็นลำดับการเกิดขึ้นฟังก์ชันใด ๆ จะต้องขึ้นกับฟังก์ชันใด และจะกระตุ้นให้เกิดฟังก์ชันใดต่อไปบ้าง การที่ฟังก์ชันหนึ่ง ๆ ต้องขึ้นกับฟังก์ชันอื่น ๆ มีสาเหตุ 3 อย่าง คือ Data, Legislation, Corporate Policy เทคนิคนี้ใช้ได้กับทั้งโมเดลระดับ Business, System และ Program โดยให้เปลี่ยนสัญลักษณ์แสดงฟังก์ชันไปตามชนิดของแต่ละระดับ และสัญลักษณ์ที่ใช้แทน Dependency คือเส้นที่มีสองหัวลูกศรซ้อนกัน นอกจากนี้ยังมีอีกสองส่วนประกอบคือ Event ทั้งที่เป็น Trigger Event และ Key Result

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

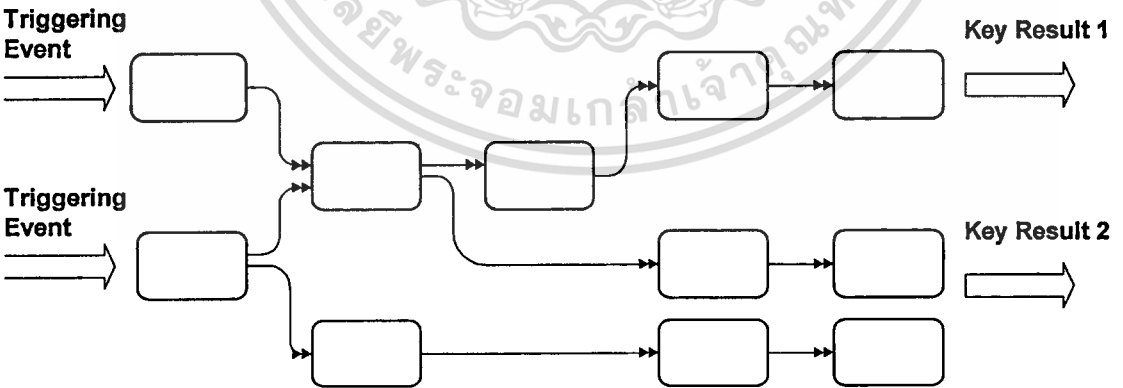
การใช้ประโยชน์จากไดอะแกรมมีในสองลักษณะ คือ

- ภาพที่ระดับบนสุด - ใช้นำเสนอให้ผู้มีอำนาจตัดสินใจใช้ประกอบการตัดสินใจว่ามีส่วนใดบ้างที่ควรพัฒนาระบบขึ้นมารองรับกับช่วยกำหนดลำดับงานว่าควรพัฒนาระบบใดก่อนหรือหลังระบบใด ที่ระดับนี้จะไม่แสดง trigger event และ key result
- ภาพที่ระดับล่างสุด - การสร้างไดอะแกรมนี้เป็นงานที่ต้องใช้เวลามาก ควรสร้างแต่เฉพาะจุดที่สำคัญๆหรือมีความซับซ้อน ประโยชน์ก็เพื่อให้ผู้เกี่ยวข้องทุกฝ่ายสามารถเข้าใจในจุดซับซ้อนเหล่านั้น



ภาพที่ 3.5 A Function Dependency

เมื่อสร้างไดอะแกรมได้แล้ว จะต้องมีการตรวจสอบความถูกต้องโดยพิจารณาแต่ละส่วนในไดอะแกรมพร้อมกับตั้งคำถาม ข้อควรตระหนักในการใช้ Data Dependency Diagram คือ แม้ว่าไดอะแกรมนี้จะมีประโยชน์ต่อการวางแผนพัฒนาระบบงานมาก แต่เราไม่สามารถใช้ไดอะแกรมนี้โดดๆ เพียงอย่างเดียวได้ เพราะจะเห็นได้ว่าไม่มีข้อมูลเพียงพอที่จะทำให้เราสามารถประเมินค่าใช้จ่ายและเวลาในการพัฒนาระบบงาน



ภาพที่ 3.6 A Detail Function Dependency Diagram

3.4 ฟังก์ชันลอจิก (Function Logic)

Function Logic เป็นเทคนิคที่ใช้กันในระดับ Business และ System เพื่อแสดง specification ในรายละเอียดอย่างครบถ้วนของบางฟังก์ชันที่มีความสำคัญหรือซับซ้อน มากกว่าที่จะใช้กับทุก ๆ ฟังก์ชัน เนื่องจากจะต้องให้รายละเอียดครบถ้วน แต่ละฟังก์ชันที่อ้างถึงใน Function Logic นี้จะต้องเป็น Atomic Function เท่านั้นและฟังก์ชันนั้นอาจเป็น Elementary Function ด้วยก็ได้ ประโยชน์ 3 อย่างที่จะได้จากเทคนิคนี้คือ

- เป็นการแสดงให้เห็นการ navigating the information model ซึ่งจะไม่เน้นถึงลำดับของการ navigate แต่อย่างใด
- แสดงถึงการสร้างข้อมูลใหม่ ๆ ซึ่งอาจเกี่ยวข้องกับอัลกอริธึมการคำนวณที่ซับซ้อน
- แสดงถึงขอบเขตของฟังก์ชันที่มีผลกระทบต่อระบบ

การนำเสนออีกลักษณะหนึ่งที่น่าสนใจคือ Annotating Information Models เป็นการนำ Information Model ที่จัดทำไว้มาใช้ประโยชน์อีกต่อหนึ่งโดยดึงมาเฉพาะส่วนที่ฟังก์ชันนั้น ๆ ใช้ด้วยการใช้สีเน้นภาพส่วนนั้น แล้วใส่รายละเอียดที่เป็นการสร้างคำนวณค่ากับพวกเงื่อนไขต่าง ๆ เพิ่มเติมเข้าไป

3.5 โพรเซสโมเดลลิง (Process Modeling)

การจัดทำโมเดลของระบบเป็นการทำงานในช่วงของการออกแบบระบบ ซึ่งก่อนจะกล่าวถึงเทคนิคการเขียนโมเดล จะมาทำความเข้าใจคำว่าโพรเซสและระบบกันก่อน โพรเซสเป็นแนวคิดที่คล้าย ๆ กับฟังก์ชัน คือ เป็นกิจกรรมหรือการทำงานอย่างใดอย่างหนึ่ง โดยโพรเซสจะเป็นการทำงานภายในระบบ ซึ่งบางโพรเซสเป็นการทำงานเพื่อรองรับฟังก์ชันต่าง ๆ ที่ได้กำหนดมาในขั้นตอน Function Modeling บางโพรเซสจะเป็นงานที่จำเป็นต่อการทำงานของระบบให้เป็นไปอย่างราบรื่น เช่น งานติดตั้งระบบ งานสำรองข้อมูลกับกระบบ งานเพื่อความปลอดภัยของระบบ และอื่น ๆ อีกมาก ข้อที่ทำให้โพรเซสต่างจากฟังก์ชันอีกประการหนึ่งก็คือ ในขั้นนี้จะระบุวิธีการหรือระบุเชิงเครื่องมือ เทคโนโลยีที่จะนำมาใช้งานด้วย สัญลักษณ์ของโพรเซสคือรูปสี่เหลี่ยมตัดมุม ดังนั้นระบบก็เป็นกลุ่มของโพรเซสที่ทำงานเพื่อรองรับกลุ่มฟังก์ชันใด ๆ จะมีการระบุวิธีการทำงานที่อาจนำคอมพิวเตอร์ เครื่องจักรกล เครื่องมือต่าง ๆ มาใช้หรือเป็นการทำงานของพนักงานที่ไม่ใช้เครื่องมือใด ๆ ช่วยเลยก็ได้

หัวข้อย่อยในส่วนนี้จะเป็นเรื่องของเทคนิค 2 วิธีที่ใช้กันมากในระดับ System ได้แก่ Dataflow Diagram และ State-Transition Diagram แต่ก่อนที่จะกล่าวถึงสองเทคนิคนั้น เราจะย้อนไปกล่าวถึงเทคนิคอื่น ๆ ที่แสดงไว้ใน Function Modeling เพื่อนำมาเขียนโมเดลของโพรเซส

จากที่เราได้ใช้ Function Dependency Diagram และ Function Hierarchy ช่วยในการกำหนดขอบเขตของฟังก์ชันว่ามีกลุ่มใดที่ควรจัดทำเป็นระบบงาน เราสามารถนำเทคนิค Hierarchy มาสร้าง Process Hierarchy ให้เป็นภาพแสดงขอบเขตการทำงานของระบบ ซึ่งอาจมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับขอบเขตการทำงานนั้นลดเพิ่มตามสมควร โดยพิจารณาจากหลายๆปัจจัย ได้แก่ เวลา ความพร้อมด้านเงินทุนและเทคโนโลยี นอกจากนั้น Function Dependency Diagram ยังใช้ช่วยประกอบการพิจารณาในขั้นเริ่มต้นว่าภายในระบบควรมีโพรเซสโดยอยู่บ้าง และโพรเซสเหล่านั้นจะรองรับการทำงานของฟังก์ชันใดบ้าง โดยให้จับกลุ่มฟังก์ชันที่เกิดขึ้นประจำกับเป็นงานของผู้ใช้ประเภทเดียวกันไว้ในโพรเซสเดียวกัน

เทคนิค Dependency Diagram ยังถูกใช้ในการสร้าง Process Dependency Diagram เพราะทั้งวิธีการทำงาน (Mechanism) กับเทคโนโลยีที่นำมาประยุกต์ใช้อาจส่งผลทำให้เกิด Data Dependency ระหว่างโพรเซส เช่น การขายผักผลไม้ในซูเปอร์มาร์เก็ตจะมีเครื่องชั่งที่จะคำนวณราคาสินค้าจากน้ำหนักที่ชั่งและพิมพ์ป้ายราคาใช้ติดไว้สินค้านั้น ๆ เพื่อนำไปชำระเงินที่แคชเชียร์ต่อไป แต่ก่อนที่จะทำขั้นตอนงานทั้งหมดนี้ได้ ต้องมีการป้อนราคาสินค้าต่อหน่วยให้กับเครื่องชั่งนั้นเสียก่อน

เทคนิค Function Logic ถูกนำมาใช้สร้าง Process Logic และเช่นเดียวกับการเขียน Function Logic ที่ว่าเป็นงานที่ต้องใช้เวลามาก ดังนั้นผู้ออกแบบระบบควรเลือกเขียนเฉพาะโพรเซสสำคัญๆ หรือซับซ้อนเท่านั้น ผู้ออกแบบอาจไม่จำเป็นต้องเขียน Process Logic สำหรับโพรเซสที่รองรับการทำงานของฟังก์ชันก็ได้เพราะมีรายละเอียดอยู่ใน Function Logic แล้ว แต่โพรเซสสำหรับงานระบบที่สำคัญๆ อย่างงานสำรวจข้อมูลหรือกู้ระบบ นั้นจำเป็นต้องสร้าง Process Logic เพื่อใช้แสดงรายละเอียดต่างๆ ของโพรเซสนั้น

3.6 ดาต้าโฟลว์ไดอะแกรม (Dataflow Diagram)

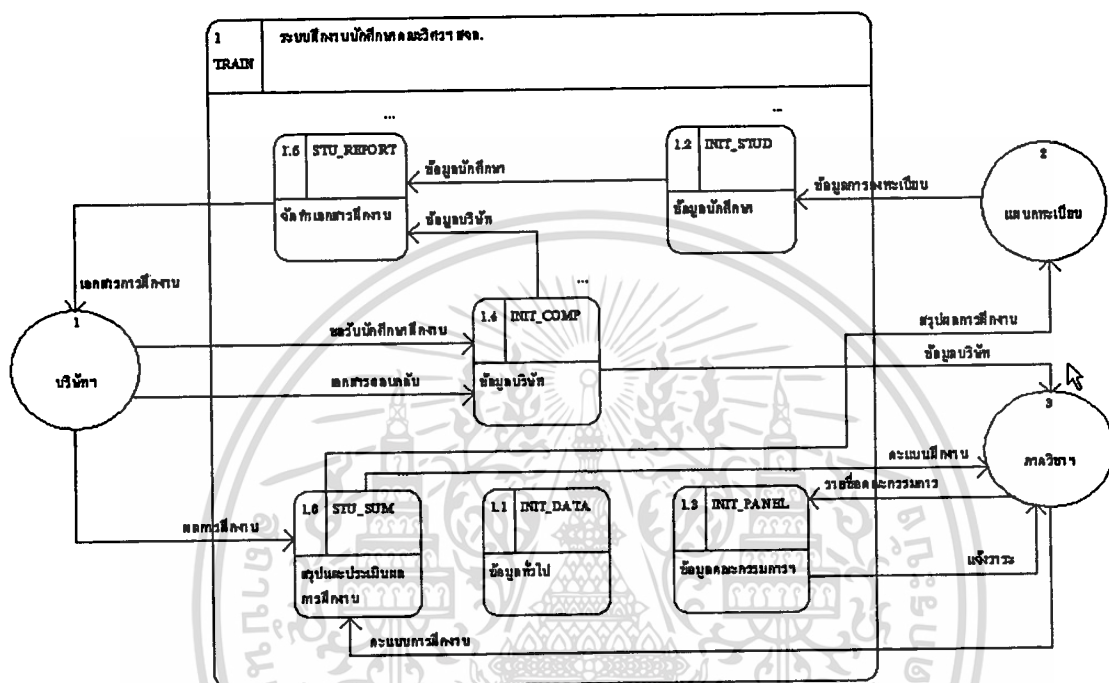
ไดอะแกรมนี้จัดเป็น Dependency Diagram เช่นกัน แต่จะพิเศษกว่าตรงที่จะแสดง Data Dependency ในรายละเอียดมากขึ้น ทำให้เห็นภาพว่าข้อมูลใดจะถูกจัดเก็บ และแสดงว่าข้อมูลต่างๆจะถูกดำเนินการผ่านในโพรเซสใดบ้าง

ประโยชน์ของไดอะแกรมชนิดนี้ คือ ใช้ช่วยในการตัดสินใจว่าควรจะทำการแตกโพรเซสลงไปเป็นโพรเซสย่อยอีกหรือไม่ จนกระทั่งได้ไดอะแกรมในระดับที่เป็นรายละเอียดที่สุดแล้ว ไดอะแกรมนั้นก็เหมาะที่จะใช้ในการกำหนด specification สำหรับงานออกแบบ/เขียนโปรแกรม, เขียนเอกสารคู่มือหรือใช้ออกแบบอุปกรณ์ต่างๆเพื่อใช้ในระบบต่อไป มีข้อสังเกตว่าไดอะแกรมนี้ไม่สามารถอธิบายการทำงานภายในโพรเซสหรือฟังก์ชันได้อย่าง Process Logic หรือ Function Logic และไม่สามารถบอกได้ว่าเมื่อใดจึงจะเกิดการดำเนินงานของโพรเซสใด ๆ เพราะไม่มีการแสดง Event ไว้ในไดอะแกรมนี้เลย เทคนิคนี้สามารถนำไปใช้ในระดับของ Business และ Program ได้ ในไดอะแกรมนี้จะมีส่วนประกอบเพิ่มขึ้นอีกสามส่วนคือ

- Dataflow ใช้แสดงข้อมูลใดๆที่ถูกแลกเปลี่ยนระหว่างสองโพรเซส สัญลักษณ์ของ Dataflow คือเส้นตรงที่มีหัวลูกศรชี้ไปทิศทางที่เกิดการไหลของข้อมูล ซึ่งอาจเกิดได้ทั้งสองทางก็จะเป็นเส้นตรงที่มีหัวลูกศรทั้งสองปลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Dastore ใช้แทนหน่วยข้อมูลที่ถูกจัดเก็บ โดยจะมีการกำหนดชื่อไว้ใช้อย่างถึงจากโปรเซสต่าง ๆ สัญลักษณ์ของ Dastore ที่ระดับ System จะเป็นรูปกรอบสี่เหลี่ยมตัดมุมที่เปิดด้านหนึ่ง ภายในกรอบแสดงชื่อย่อและคำอธิบาย การที่ Dastore ในระดับ System จะจัดเก็บอย่างไรนั้นไม่ได้มีผลต่อการเก็บข้อมูลจริงๆว่าต้องเก็บในลักษณะนั้น



ภาพที่ 3.7 ดาต้าโฟลว์

- External Entity เป็นสัญลักษณ์ที่ใช้แสดงสิ่งใด ๆ ที่อยู่นอกขอบเขตระบบงาน แต่สัมพันธ์กับระบบด้วยการส่งผ่านของข้อมูลถึงกันอยู่ คำว่า Entity ในที่นี้เป็นคนละแนวคิดกับ Entity ใน Information Model

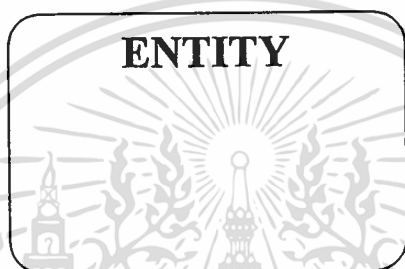
ถ้อยคำที่ใช้อธิบายทั้งใน Dataflow และ Datastore จะเป็นสิ่งต่างๆที่อยู่ใน Information Model นั่นคือจะระบุด้วยชื่อของ Entity, Attribute และ Relationship ต่างๆ

3.7 การเขียนเอนติตี้รีเลชันชิพโมเดล (Entity Relationship Model)

ถือเป็น High Level Conceptual Shema หรือ High Level Conceptual Model ซึ่งหมายถึงไม่ได้เก็บเฉพาะความสัมพันธ์ระหว่างข้อมูล แต่ยังเก็บความสัมพันธ์ระหว่าง Rules และ Integrity Constraint ด้วยกฎความถูกต้องของข้อมูล

- เอนติตี้ (Entity) คือ สิ่งของหรือสิ่งที่เราสนใจหรือหัวข้อที่เป็นหลักสำคัญอาจจะเป็นสิ่งจริงๆ ที่มีอยู่หรือถูกสมมุติขึ้นมาก็ได้ จะเป็นสิ่งที่บรรจุหรือเกี่ยวข้องกับข้อมูลด้านต่างๆ ที่จำเป็นต้องรู้ เช่น คน, สถานที่, สิ่งของ, สินค้า, การสั่งซื้อ เป็นต้น โดยจะ

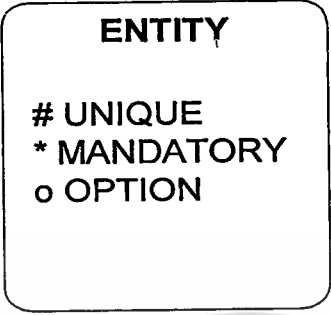
แสดงอยู่ในกรอบสี่เหลี่ยมขอบมน (Soft Box) บางเอนติตี้ในฐานะข้อมูลจะไม่มี ความหมายมากหากไม่มีเอนติตี้อื่นในฐานะข้อมูล เอนติตี้ประเภทนี้ เรียกว่า เอนติตี้ชนิดอ่อน(Weak Entity) ตัวอย่างเช่น เอนติตี้ประวัติครอบครัวของพนักงานเป็นเอนติตี้ ชนิดอ่อน เพราะปราศจากเอนติตี้พนักงานแล้วเอนติตี้นี้ก็ไม่มี ความหมาย นอกจากนี้ ยังมีการแบ่งเอนติตี้ออกเป็น Supertype และ Subtype โดยที่ Subtype ประกอบด้วย ข้อมูลเฉพาะนอกเหนือจากที่มีอยู่ใน Supertype ดังนั้น เมื่อมีข้อมูลของ Subtype ก็ จะต้องมียข้อมูลของ Supertype อยู่เช่นกัน เช่น พนักงานเป็น Supertype ที่มีการจัด กลุ่มพนักงานโดยแบ่งออกเป็นพนักงานที่มีเงินเดือนประจำและพนักงานที่คิดค่าแรง เป็นชั่วโมง



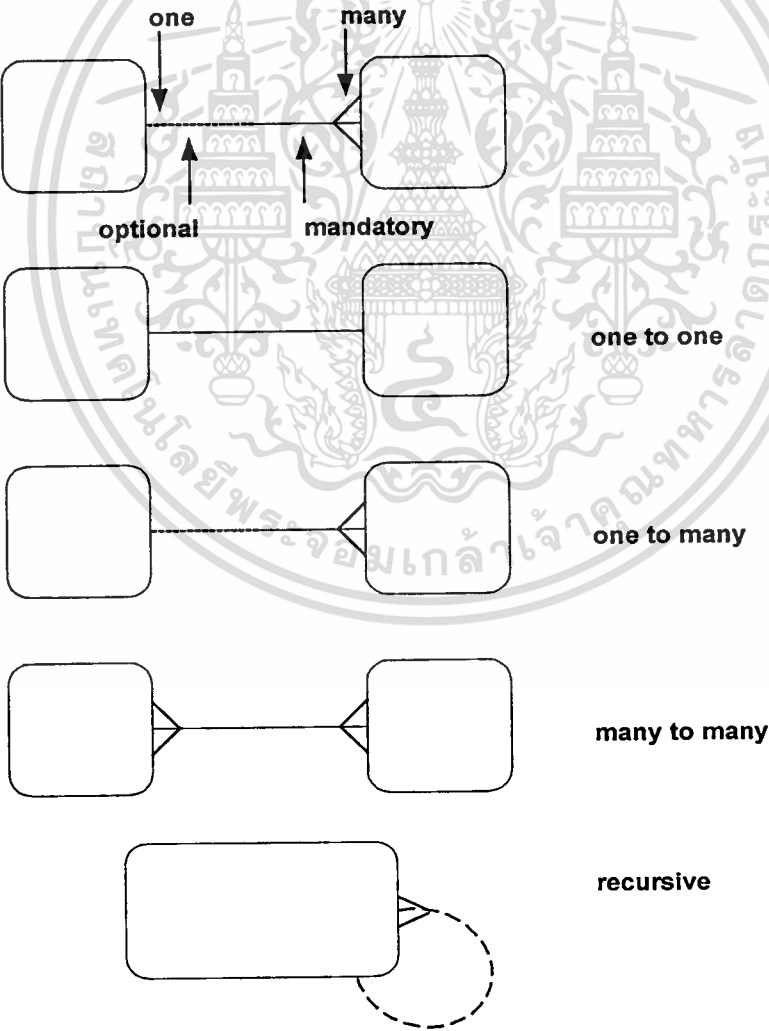
ภาพที่ 3.8 แสดงสัญลักษณ์ของ Entity

- แอททริบิวต์ (Attribute) คือ รายละเอียดต่างๆ ที่ครอบคลุมถึงคุณภาพ ลักษณะ ประเภท จำนวนหรือ ข้อกำหนดคุณลักษณะที่เกี่ยวข้องกับสถานะของเอนติตี้ เช่น เอนติตี้พนักงานประกอบด้วยแอททริบิวต์ รหัสพนักงาน ชื่อ เงินเดือน เป็นต้น แอททริบิวต์ของแอททริบิวต์ประกอบด้วยข้อมูลหลายส่วนมารวมกันซึ่งอาจแยกชื่อ แอททริบิวต์ย่อยได้อีก แอททริบิวต์ที่มีคุณสมบัติอย่างนี้ เรียกว่า แอททริบิวต์ผสม (Composite Attribute) เช่น แอททริบิวต์ที่อยู่ เป็นแอททริบิวต์ผสมที่ประกอบด้วย ข้อมูล บ้านเลขที่ ถนน ซอย อำเภอ จังหวัด และรหัสไปรษณีย์ ซึ่งสามารถแยกออกเป็นแอททริบิวต์ได้อีก (ถ้าต้องการ) เช่น แยกเป็นแอททริบิวต์ที่อยู่1 ซึ่งอาจ ประกอบด้วย บ้านเลขที่ ถนน ซอย อำเภอ และแอททริบิวต์ที่อยู่2 ซึ่งประกอบด้วย จังหวัดและรหัสไปรษณีย์ เป็นต้น โดยแอททริบิวต์สามารถแบ่งออกเป็น 3 ลักษณะ คือ
 - แอททริบิวต์นั้นไม่ซ้ำกัน และไม่สามารถที่จะเป็นนัล (null) ได้ ใช้สัญลักษณ์ นำหน้าแอททริบิวต์ คือ เครื่องหมาย “#”
 - แมนดาทอรี แอททริบิวต์ (Mandatory Attribute) เป็นแอททริบิวต์ที่จะต้อง มีค่าอยู่เสมอ และไม่สามารถที่จะเป็นนัล (null) ได้ ใช้สัญลักษณ์นำหน้า แอททริบิวต์ คือเครื่องหมาย “*”

- ออฟชั่น แอททริบิวต์ (Option Attribute) เป็นแอททริบิวต์ที่อาจจะมีค่าหรือไม่มีก็ได้ สามารถเป็นค่านัล (null) ได้ ใช้สัญลักษณ์นำหน้าแอททริบิวต์ คือ เครื่องหมาย “o”



ภาพที่ 3.9 แสดงสัญลักษณ์ของ Entity, Attribute และสัญลักษณ์ที่ใช้

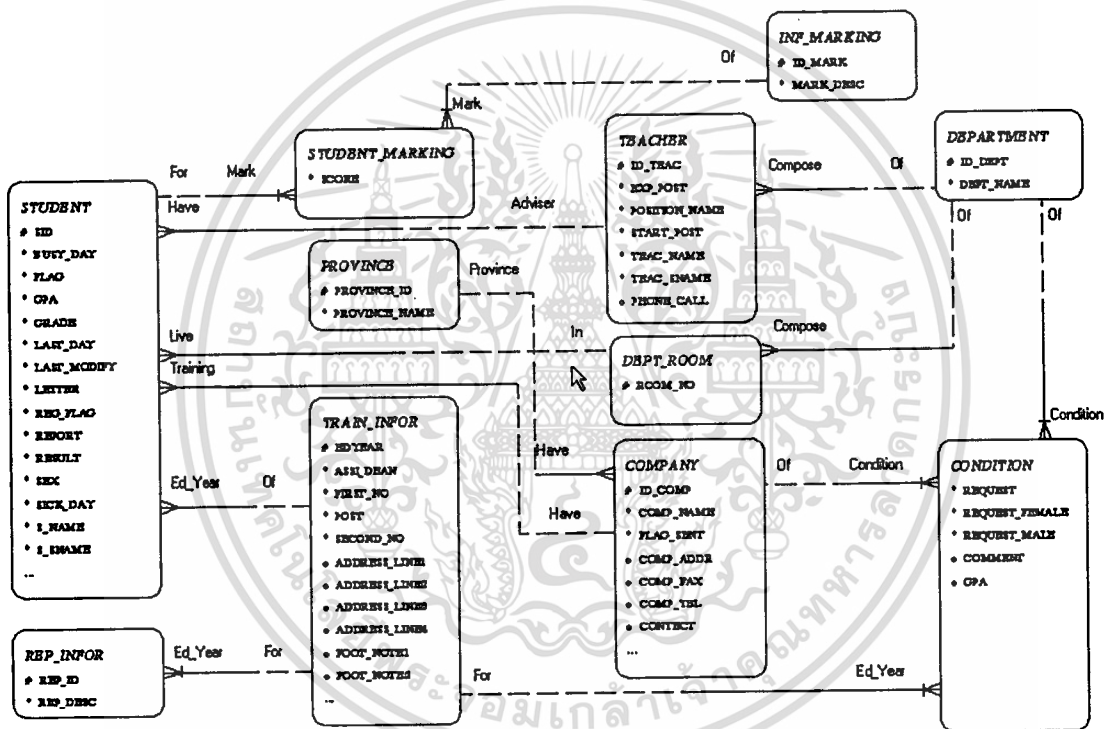


ภาพที่ 3.10 แสดงสัญลักษณ์ของความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสัมพันธ์ (Relationship) เป็นสิ่งที่สำคัญในการเชื่อมโยงระหว่าง 2 เอนทิตีซึ่งสามารถแสดงความสัมพันธ์ได้ทั้งที่เป็นออฟชั่นนอล (เป็นเส้นประ) โดยมีความหมายว่าถ้าเป็นความสัมพันธ์ แบบออฟชั่นนอลแอททริบิวต์ที่มีความสัมพันธ์จะมีค่าข้อมูลหรือไม่มีค่าของข้อมูลก็ได้ (null) หรือแมนดาทอรี (เส้นทึบ) โดยมีความหมายว่าถ้าเป็นความสัมพันธ์แบบแมนดาทอรีแอททริบิวต์นั้น ๆ จะต้องมีความสัมพันธ์เสมอจะไม่สามารถเป็นค่านัล (null) ได้ และยังแสดงถึงความสัมพันธ์ในลักษณะที่เป็น ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many) และความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many)
 - ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationship) เป็นการแสดงความสัมพันธ์ของแอททริบิวต์ในเอนทิตีหนึ่งว่ามีความสัมพันธ์กับอีกแอททริบิวต์ในอีกเอนทิตีหนึ่งอย่างมากเพียงค่าข้อมูลเดียว เช่น ในแต่ละประเทศจะมีผู้นำประเทศได้เพียงคนเดียวเท่านั้น ความสัมพันธ์ระหว่างเอนทิตีรายชื่อผู้นำกับเอนทิตีรายชื่อประเทศมีความสัมพันธ์กันในลักษณะที่เป็นแบบหนึ่งต่อหนึ่ง (1:1)
 - ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many Relationship) เป็นการแสดงความสัมพันธ์ของแอททริบิวต์ในเอนทิตีหนึ่งว่ามีความสัมพันธ์กับอีกแอททริบิวต์ในอีกเอนทิตีหนึ่งหลายค่า เช่น ความสัมพันธ์ของลูกค้ากับใบสั่งซื้อสินค้า เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many) นั่นคือลูกค้าหนึ่งคนสามารถที่จะมีใบสั่งซื้อได้มากกว่าหนึ่งใบ (1:N) ในทางกลับกัน ความสัมพันธ์ของใบสั่งซื้อไปสู่ลูกค้าจะเป็นลักษณะหนึ่งต่อหนึ่ง (1:1) เพราะว่าหนึ่งใบสั่งซื้อเกิดจากลูกค้าเพียงรายเดียว ดังนั้นความสัมพันธ์ระหว่างเอนทิตีลูกค้ากับเอนทิตีการสั่งซื้อจึงเป็นความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many)
 - แสดงความสัมพันธ์ของสองเอนทิตีในลักษณะแบบกลุ่มไปกลุ่ม เช่น เอนทิตีของของนักศึกษา กับเอนทิตีการลงทะเบียน โดยนักศึกษาหนึ่งคนสามารถที่จะลงทะเบียนเรียนได้หลายวิชา (1:N) และในหนึ่งวิชาสามารถที่จะมีนักศึกษาลงทะเบียนเรียนได้หลายคน (1:N) ดังนั้นความสัมพันธ์ของสองเอนทิตีจึงเป็นแบบกลุ่มไปกลุ่ม (M:N) ความสัมพันธ์ระหว่างสองเอนทิตีที่เป็นแบบกลุ่มไปกลุ่ม (M:N) เป็นเรื่องที่ยากจะยุ่งยากในการออกแบบฐานข้อมูล เช่น อาจมีปัญหาในด้านของการปรับปรุงแก้ไขข้อมูล โดยทั่ว ๆ จะมีการสร้างเอนทิตีใหม่ขึ้นมา (Associative Entity) เพื่อเป็นเอนทิตีเชื่อมความสัมพันธ์กับสองเอนทิตีเดิม โดยมีวัตถุประสงค์เพื่อปรับความสัมพันธ์ให้อยู่ในรูปของหนึ่งต่อกลุ่ม (1:N)

- ไพรมารีคีย์ (Primary Key) เป็นแอททริบิวต์ที่มีคุณสมบัติของข้อมูลที่เป็นเอกลักษณ์หรือมีค่าที่ซ้ำซ้อนกัน คุณสมบัติดังกล่าวจะสามารถที่จะระบุข้อมูลเป็นข้อมูลของ Record ใดในรีเลชัน แอททริบิวต์ที่มีคุณสมบัติเป็นคีย์หลักสามารถที่จะประกอบด้วยแอททริบิวต์มากกว่าหนึ่งแอททริบิวต์ได้ เพื่อที่จะให้ค่าเป็นค่าเอกลักษณ์ได้ คีย์หลักที่ประกอบด้วยหลายแอททริบิวต์นี้เรียกว่า คีย์ผสม (Composite Key) ในรีเลชันหนึ่งๆ สามารถที่จะมีแอททริบิวต์ที่มีคุณสมบัติที่สามารถเป็นคีย์หลักได้มากกว่าหนึ่งแอททริบิวต์ แอททริบิวต์เหล่านี้ เรียกว่า คีย์คู่แข่ง (Candidate Key) ถ้าแอททริบิวต์หนึ่งถูกกำหนดคีย์หลัก อีกแอททริบิวต์หนึ่งที่มีคุณสมบัติเป็นคีย์หลักแต่ไม่ได้ถูกเลือกเป็นคีย์หลัก จะเรียกแอททริบิวต์เหล่านั้นว่า คีย์สำรอง (Alternate Key)



ภาพที่ 3.11 แสดง Entity Relationship

- ฟอเรนคีย์ (Foreign Key) เป็นแอททริบิวต์ในรีเลชันหนึ่งที่ถูกใช้อ้างอิงถึงแอททริบิวต์ในอีกรีเลชันหนึ่ง โดยที่แอททริบิวต์นี้มีคุณสมบัติเป็นคีย์หลักในรีเลชันที่ถูกอ้างอิง การที่มีแอททริบิวต์นี้ปรากฏอยู่ในรีเลชันทั้งสองก็เพื่อประโยชน์ในการเชื่อมโยงข้อมูลซึ่งกันและกัน

สถาปัตยกรรม ORACLE

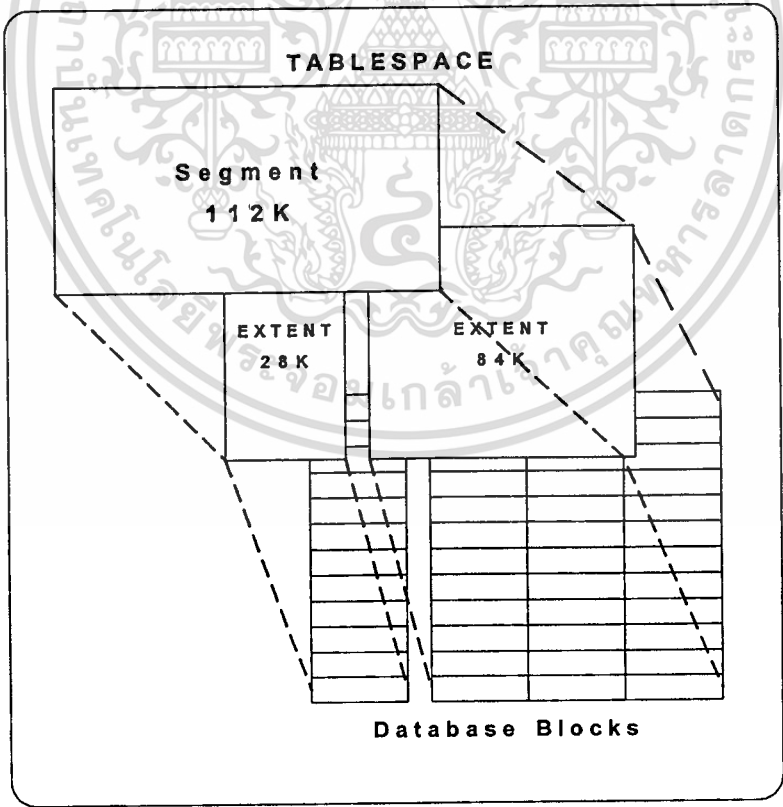
4.1 หลักการเบื้องต้นของออรากิล

4.1.1 โครงสร้างของฐานข้อมูล (Database Structure)

หน่วยของการจัดจรงข้อมูลของออรากิล (Oracle) จะมีรูปแบบที่เป็นดาต้าบล็อก (Data Block) เอ็กเทินต์ (Extents) และ เซกเมนต์ (Segments)

4.1.1.1 ดาต้าบล็อก (DATA BLOCK)

ออรากิลจะเก็บข้อมูลในดาต้าบล็อก หรือบางครั้งอาจจะเรียกว่า โลจิกคอล บล็อก (Logical Block) ออรากิลบล็อกหรือเพจ (Oracle Block or Pages) 1 ดาต้าบล็อกจะมีขนาดเท่ากับจำนวนไบต์ของพื้นที่ฐานข้อมูลอย่างแท้จริง (Physical Database Space) บนดิสก์ ซึ่งขนาดของพื้นที่จะถูกกำหนดค่าขณะที่มีการสร้างฐานข้อมูล



ภาพที่ 4.1 แสดงความสัมพันธ์ระหว่าง ดาต้าบล็อก เอ็กเทินต์ และเซกเมนต์

4.1.1.2 เอ็กเทนต์ (EXTENTS)

เป็นจำนวนของดาต้าบล็อกซึ่งอยู่ติด ๆ กันที่ทำการระบุไว้ ซึ่งถูกจับจองเพื่อการจัดเก็บชนิดของข้อมูลโดยเฉพาะ

4.1.1.3 เซ็กเมนต์ (SEGMENT)

เป็นกลุ่มของเอ็กเทนต์ซึ่งถูกจับจองไว้สำหรับชนิดที่มีลักษณะเฉพาะของโครงสร้างข้อมูลและทั้งหมดเหล่านี้จะถูกเก็บในเมเบิลสเปซ (Tablespaces) เดียวกัน

4.1.1.4 เทเบิลสเปซ (TABLESPACES)

ฐานข้อมูลจะแบ่งออกเป็นหนึ่ง หรือหลายเทเบิลสเปซ ผู้จัดการฐานข้อมูล (DBA) สามารถที่จะทำการใช้เทเบิลสเปซในการทำสิ่งต่าง ๆ ดังนี้

- ควบคุมการจับจองที่ว่างสำหรับข้อมูลของข้อมูลในดิสก์
- กำหนดส่วนแบ่งดิสก์ (Disk Quota) สำหรับผู้ใช้งานระบบฐานข้อมูลของออราเคิล (Oracle)
- ควบคุมการใช้งานของข้อมูลในเทเบิลสเปซโดยทำการตั้งให้เป็นออนไลน์ (Online) หรือ ออฟไลน์ (Offline)
- กระทำการสำรองข้อมูล (Backup) หรือการคืนสู่สภาพเดิม (Recovery) ของฐานข้อมูลเฉพาะบางส่วนได้
- ทำการจับจองหน่วยสำรองข้อมูล (Data Storage) ข้ามอุปกรณ์สำหรับการปรับปรุงประสิทธิภาพ

4.1.1.5 ซิสเต็มเทเบิลสเปซ (System Tablespace)

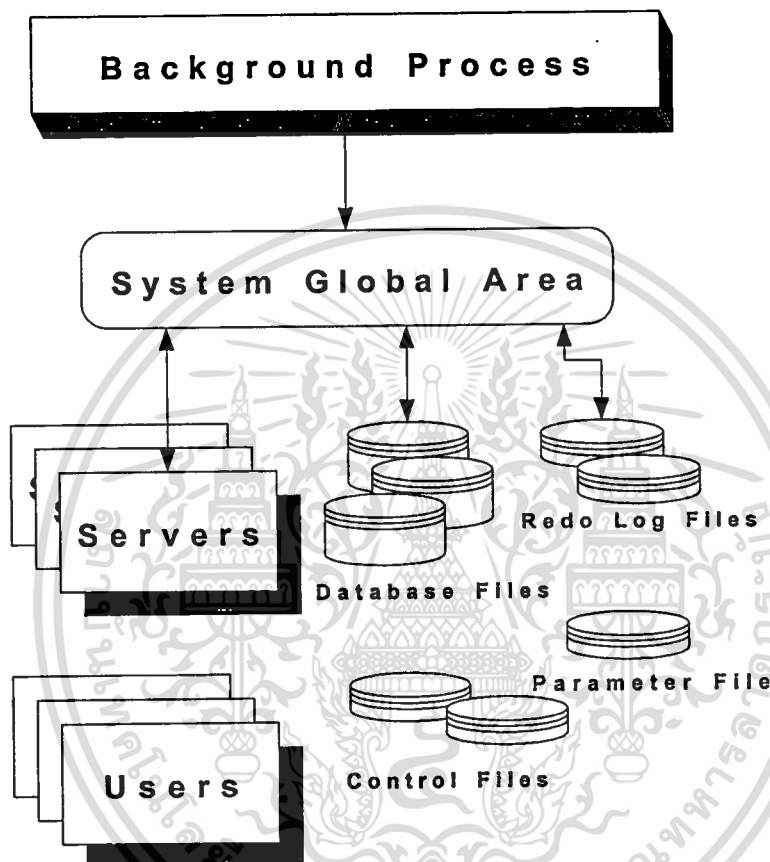
จัดเป็นเทเบิลสเปซชนิดหนึ่งซึ่งทุก ๆ ฐานข้อมูลของออราเคิลจะมีเทเบิลสเปซที่ชื่อว่าซิสเต็ม (System) ซึ่งออราเคิลจะทำการสร้างเทเบิลสเปซชนิดนี้ขึ้น เมื่อมีการสร้างฐานข้อมูลขึ้นโดยอัตโนมัติ ซิสเต็มเทเบิลสเปซ จะทำการบรรจุตารางของดาต้าดิกชันนารี (Data Dictionary) สำหรับฐานข้อมูลทั้งหมด ในกรณีที่ฐานข้อมูลมีขนาดเล็ก อาจไม่จำเป็นที่จะต้องมีการสร้างเทเบิลสเปซใหม่ขึ้นมา แต่ในกรณีที่มิขนาดของฐานข้อมูลที่มีขนาดใหญ่ อาจจำเป็นที่จะมีการสร้างเทเบิลสเปซขึ้นใหม่สำหรับผู้ใช้งานระบบ ซึ่งจะเป็นการปรับปรุงประสิทธิภาพของการเข้าถึงข้อมูลบนฐานข้อมูลได้เป็นอย่างดี (ซิสเต็มเทเบิลสเปซควรที่จะมีการออนไลน์ตลอดเวลา)

4.2 สถาปัตยกรรมของออราเคิล (ORACLE ARCHITECTURE)

4.2.1 ออราเคิลอินสแตนซ์ (ORACLE INSTANCE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกๆ ครั้งที่เริ่มต้นการใช้งานฐานข้อมูล บนตัวให้บริการฐานข้อมูล(Database Server) ฐานข้อมูลของออราเคิล จะทำการจับจองเนื้อที่ในหน่วยความจำส่วนหนึ่งที่เรียกว่า เอสจีเอ (System Global Area: SGA) แล้วหลังจากนั้นจึงจะทำการเริ่มต้นโปรเซสของออราเคิล(Oracle Process) ออราเคิลดาต้าเบส อินสแตนซ์(Oracle Database Instance) เป็นการรวมกันของเอสจีเอ และโปรเซสของออราเคิล



ภาพที่ 4.2 ออราเคิลดาต้าเบส อินสแตนซ์ (Oracle Database Instance)

4.2.2 โครงสร้างของโปรเซส (PROCESS STRUCTURE)

โปรเซส (Process) หมายถึงกระบวนการของระบบปฏิบัติการซึ่งสามารถกระทำลำดับของขั้นตอน ในการสร้างของโปรเซส (Process) จะมีเป้าหมายในการกระทำอยู่ 2 ลักษณะ คือ

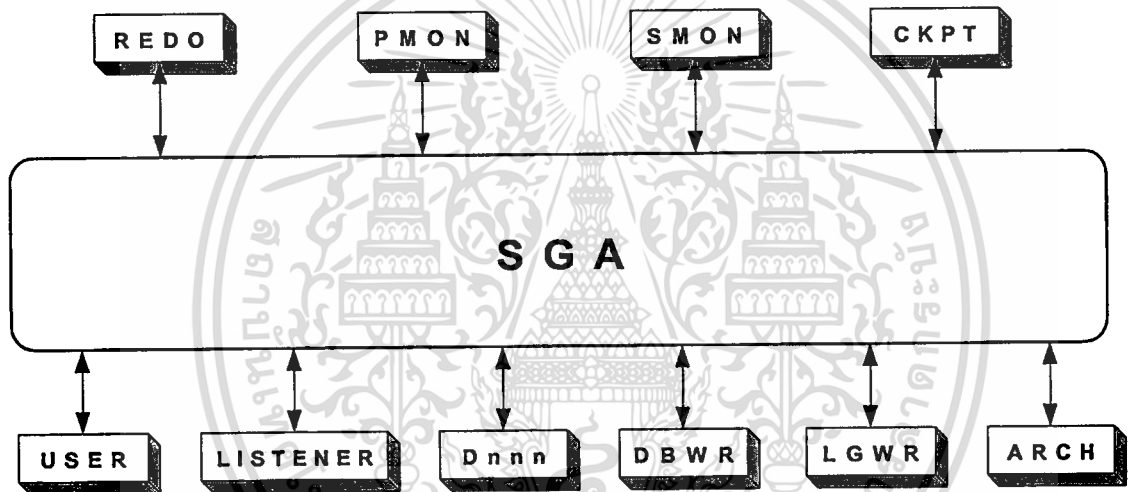
- จำลองสภาพแวดล้อมส่วนตัว (Private Environment) สำหรับโปรเซสหลายอย่างที่จะทำงานไปพร้อมๆ กัน ซึ่งแต่ละโปรเซสจะมีสิ่งแวดล้อมเป็นของตัวเอง
- อนุญาตให้โปรเซสหลายๆ โปรเซส ได้มีการใช้ทรัพยากรของคอมพิวเตอร์ในลักษณะร่วมกันแต่จะใช้ได้ในเวลาสั้นๆ เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ออราเคิลอินสแตนซ์แบบหลายโพรเซส (MULTIPLE PROCESS ORACLE INSTANCE)

โพรเซสหลายๆ โพรเซสสามารถที่จะเอ็กซีคิวต์ส่วนต่างๆ กันของออราเคิลได้และโพรเซสต่างๆ เหล่านั้นจะถูกแยกออกไปเป็นส่วนของแต่ละผู้ใช้งานอีกด้วย

- งานต่างๆ ของออราเคิลและโปรแกรมประยุกต์ ฐานข้อมูลสามารถที่จะรองรับการใช้งานได้หลายๆ โพรเซส
- ผู้ใช้งานหลายๆ คนและ หลายๆ โปรแกรมประยุกต์สามารถที่จะติดต่อถึงดาต้าเบสอินสแตนซ์เดียวกันได้ในเวลาที่พร้อมกัน โดยยังมีประสิทธิภาพที่ดีอยู่



ภาพที่ 4.3 ออราเคิลอินสแตนซ์แบบหลายโพรเซส

ในระบบที่มีผู้ใช้งานหลายๆ คน ออราเคิลได้แบ่งโพรเซสของการทำงานออกเป็น 2 ชนิด คือ

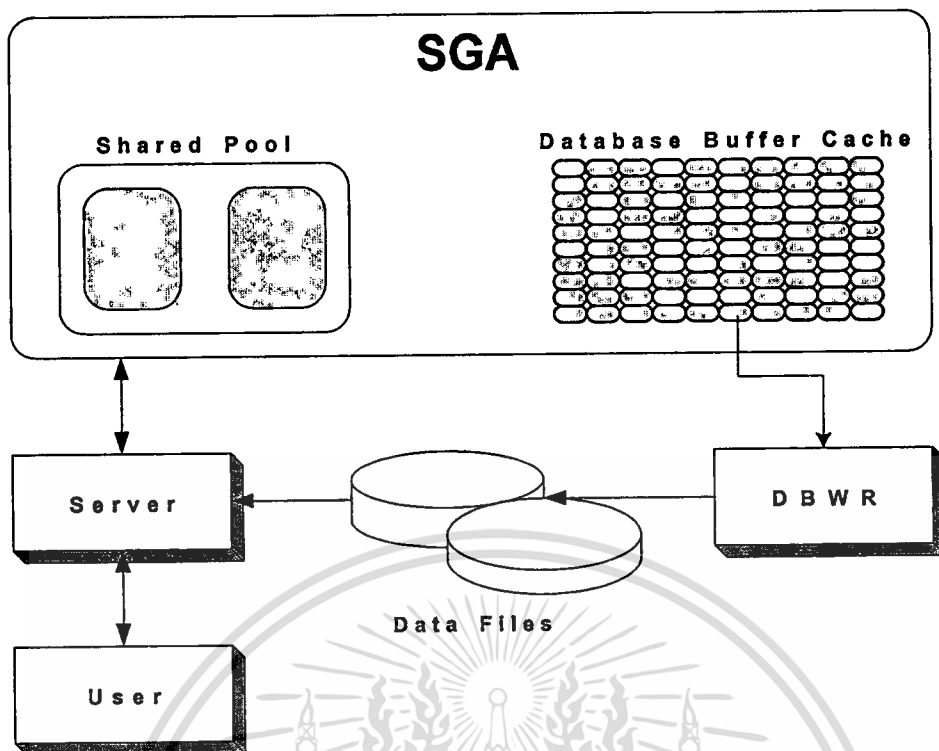
4.2.3.1 โพรเซสของผู้ใช้งานระบบ (User Process)

จะถูกสร้างเมื่อผู้ใช้งานระบบทำการเอ็กซีคิวต์โปรแกรมประยุกต์ที่ติดต่อกับฐานข้อมูลออราเคิล

4.2.3.2 โพรเซสของออราเคิล (Oracle Process)

Oracle process แบ่งออกเป็น 2 ชนิด คือ

- โพรเซสของตัวให้บริการฐานข้อมูล (Server Process)
- โพรเซสเบื้องหลัง (Background Process)



ภาพที่ 4.4 แสดงภาพ USER และ SERVER PROCESS

4.2.3.2.1 โพรเซสของตัวให้บริการฐานข้อมูล (Server Process)

เป็นตัวที่ทำหน้าที่ในการควบคุมการร้องขอของโพรเซสของผู้ใช้งานระบบ (User Process) ที่ทำการเชื่อมต่อถึงอินสแตนซ์ หน้าที่ๆ สำคัญมีดังนี้

- ทำการพาร์ส (Parse) และเอ็กซีคิวต์คำสั่งเอสคิวแอล ที่ส่งผ่านมาจากโปรแกรมประยุกต์
- ทำการอ่านดาต้าบล็อกในดาต้าไฟล์ขึ้นไปบนเอสจีเอ
- ส่งผลลัพธ์ที่ได้ให้กับโปรแกรมประยุกต์
- ทำการเคลื่อนย้ายบล็อกข้อมูลจากเดอร์ตี้บัฟเฟอร์ (Dirty Buffer) ไปยังเดอร์ตี้ลิสต์ (Dirty List)

4.2.3.2.2 โพรเซสเบื้องหลัง (Background Process)

โพรเซสเบื้องหลังสามารถที่จะแบ่งแยกความสำคัญของโพรเซสออกเป็น 2 อย่าง คือ โพรเซสที่จะต้องทำการโพรเซสตลอดเวลา (Mandatory System Process) และโพรเซสที่จะทำการโพรเซสก็ต่อเมื่ออินสแตนซ์เริ่มทำงานและจะหยุดทำงานเมื่ออินสแตนซ์หยุดทำงาน

- ดาต้าเบส ไรเตอร์ (Database Writer: DBWR) จะทำการเขียนบัฟเฟอร์ของฐานข้อมูล (Database Buffer) ไปที่ดาต้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ไฟล์ซึ่งบัฟเฟอร์ในบัฟเฟอร์แคช (Buffer Cache) ได้ถูกทำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

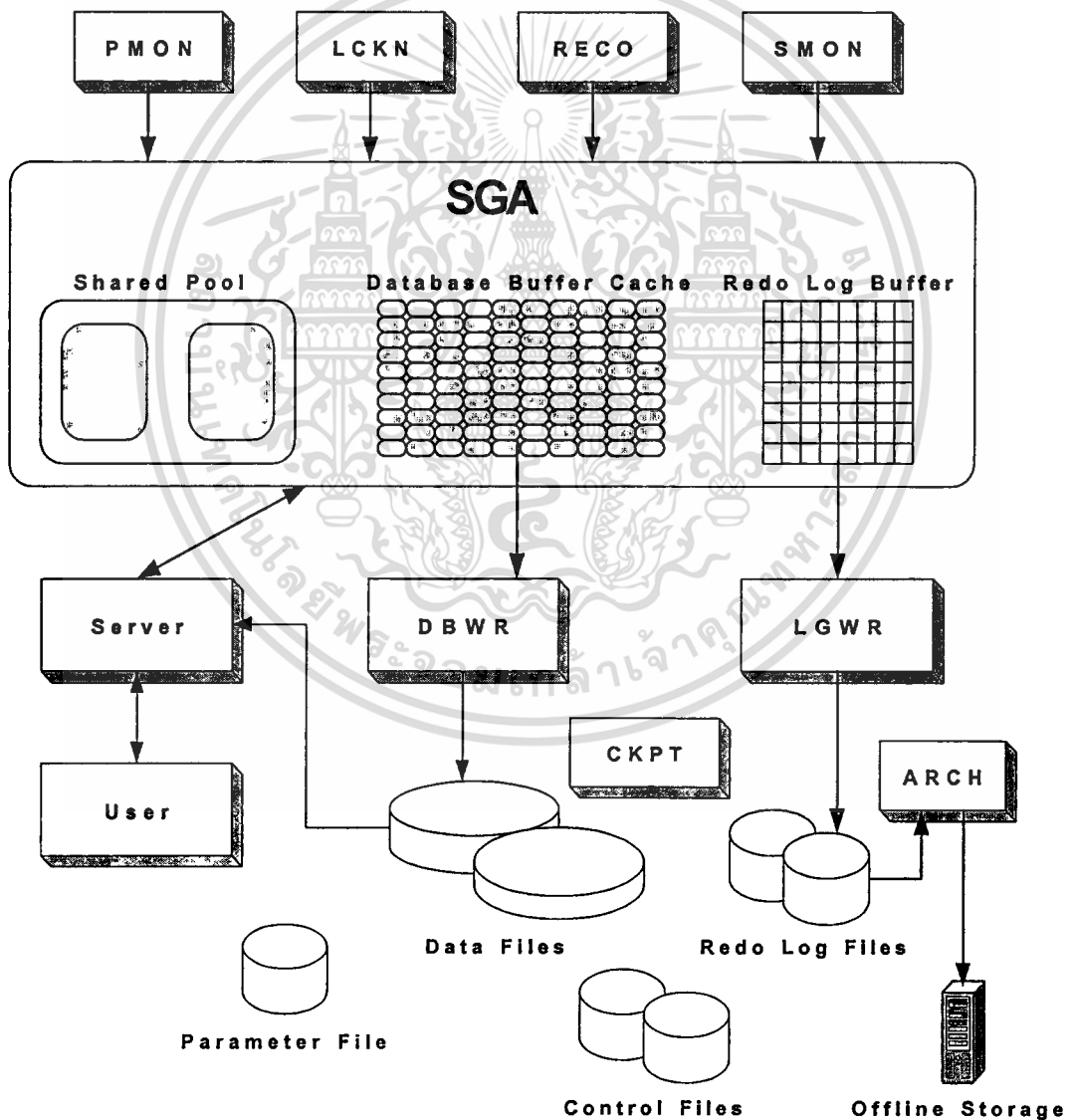
การเปลี่ยนแปลงเรียบร้อยแล้ว มันจะถูกระบุเป็นขยะ (Dirty) เนื่องจากดาต้าเบสไรต์เตอร์จะทำหน้าที่หลักคือ ทำให้บัฟเฟอร์แคชนั้นสะอาดหรือว่างอยู่เสมอ ซึ่งบัฟเฟอร์ต่างๆ ซึ่งบัฟเฟอร์ต่างๆ จะถูกใช้งานโดยโปรแกรมเมอร์ใช้งานระบบทำให้จำนวนบัฟเฟอร์ว่างน้อยลง(Free Buffer) เรื่อยๆ และเมื่อโปรแกรมเมอร์ใช้งานระบบ ต้องการเนื้อที่ในบัฟเฟอร์แต่ในบัฟเฟอร์มีเนื้อที่ไม่เพียงพอ ดังนั้นดาต้าเบสไรต์เตอร์จึงต้องทำการหาเนื้อที่ว่างหรือทำให้บัฟเฟอร์มีที่ว่างมากขึ้น ซึ่งดาต้าเบส ไรต์เตอร์จะทำการเปลี่ยนแปลงดาต้าไฟล์เมื่อ

- เดอร์ตี้ลิส (Dirty List) ถูกทำให้เต็มแล้ว
- หมดเวลา (Time Out) ทำให้ดาต้าเบส ไรต์เตอร์ทำการกำหนดตัวเองให้ทำงาน (Active) ขึ้นโดยเวลาดังกล่าว
- เกิดการเช็คพอยท์ (Checkpoint) ขึ้น
- ไม่มีบัฟเฟอร์อิสระเพียงพอสำหรับการจับจองสำหรับโปรแกรมเมอร์ใช้งานระบบ
- ล็อกไรเตอร์ (Log Writer: LGWR) เป็นกระบวนการที่ทำการเขียนรีดูล็อกบัฟเฟอร์ (Redo Log Buffer) ลงไปที่รีดูล็อกไฟล์ (Redo Log File) บนดิสก์ โดยล็อกไรเตอร์ (LGWR) จะเขียนทุก ๆ รีดูล็อกที่อยู่ในบัฟเฟอร์ตั้งแต่ครั้งสุดท้ายที่ได้ทำการเขียน โดยสิ่งที่ล็อกไรเตอร์(LGWR) เขียนมีดังนี้
 - Commit record เมื่อผู้ใช้ทำการ process commit transaction
 - Redo buffer ทุก ๆ 3 วินาที
 - Redo buffer เมื่อ redo log buffer เต็มไป 1 ใน 3
 - Redo buffer เมื่อ DBWR ทำการ process write modified buffer ไปยังดิสก์
- เช็คพอยท์ (Checkpoint: CKPT) เมื่อเช็คพอยท์เกิดขึ้น Oracleจะต้องทำการเปลี่ยนแปลงส่วนหัวของดาต้าไฟล์ทั้งหมดที่ระบุการเช็คพอยท์ ซึ่งปกติทำการแยกหน้าที่เช็คพอยท์โดยโปรแกรมเมอร์เช็คพอยท์(CKPT) ซึ่งแยกออกจากล็อกไรเตอร์จะทำให้แน่ใจว่า data sequent block ในหน่วยความจำ ที่มีการเปลี่ยนแปลงบ่อยๆ จะถูกเขียนลงไฟล์ข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในองค์กรเท่านั้น ไม่ควรเผยแพร่สู่สาธารณะโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลเป็นระยะๆ เนื่องจากการใช้ LRU algorithm ในการทำ DBWR จะทำให้ data segment block ที่มีการเปลี่ยนแปลงบ่อยๆ จะไม่ถูกเขียนลงดิสก์เลย จึงทำให้จำเป็นต้องมี checkpoint

- ซิสเต็มมอนิเตอร์ (System Monitor: SMON) จะทำให้อินสแตนซ์กลับสู่สภาพเดิม (Instance Recovery) ที่เวลาเริ่มต้นอินสแตนซ์และจะลบข้อมูลที่มีการเช็กเมนต์ชั่วคราว (Temporary Segment) ซึ่งไม่จำเป็นต้องใช้แล้ว จับกลุ่มของเอ็กเทนต์อิสระ (Free Extents) เพื่อให้มีขนาดใหญ่ขึ้นด้วย ซิสเต็ม มอนิเตอร์ (SMON) จะปลุกเมื่อต้องการการตรวจสอบเป็นบางครั้งบางคราว



ภาพที่ 4.5 แสดงภาพโครงสร้างโดยรวมของออรากเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โพรเซส มอนิเตอร์ (Process Monitor: PMON) จะทำให้โพรเซสกลับสู่สภาพเดิม (Process Recovery) เมื่อโพรเซสของผู้ใช้งานระบบล้มเหลว (fail) ลบข้อมูลในแคช(Cache) คืนทรัพยากรซึ่งโพรเซสกำลังใช้งานอยู่ เช่น ปลดล็อกตาราง (Lock Table) ลบรหัสโพรเซส(Process ID) กำหนดสถานะ (Reset Status) ของตารางทรานแซกชันที่กำลังใช้งานอยู่ใหม่ เป็นต้น โพรเซส มอนิเตอร์จะทำการเช็คสถานะของดิสแพชเชอร์ (Dispatcher) และตัวให้บริการฐานข้อมูลเป็นครั้งคราว
- รีคอบเวอร์เรอร์ (Recovered: RECO) เป็นการแก้ปัญหาที่เกิดการผิดพลาดเกี่ยวกับดิสทริบิลเติ้ล ทรานแซกชัน (Distributed Transaction) อย่างอัตโนมัติ โดยมันจะทำการเชื่อมต่อถึงฐานข้อมูลดังกล่าวอย่างอัตโนมัติเพื่อทำการติดต่อกับฐานข้อมูล (Re-Establish) อีกครั้ง เพื่อทำการแก้ปัญหา การรีเอสทาบิช คอนเนค (Re-Establish Connect) ทันทีเมื่ออยู่ในช่วงไทม์อินเทอร์วอล (Time Interval) แต่ยังไม่สามารถสร้างได้ มันจะคอยเป็นการเพิ่มเวลา, รีคอบเวอร์เรอร์จะถูกสร้างตอนเริ่มต้น ถ้าพารามิเตอร์เตอร์ ดิสทริบิวท์ทรานแซกชัน (Distributed Transaction) มีค่ามากกว่า 0
- อาไคฟ์เวอร์ (Archiver: ARCH) จะทำการสำเนาเรดูล็อกไฟล์ที่สามารถเข้าถึงได้ (Online Redo Log File) ไปยังหน่วยเก็บข้อมูลสำรอง (Storage) ที่ถูกออกแบบไว้ แต่ต้องอยู่ในโหมดอาไคฟ์ล็อก (Archivelog Mode) เท่านั้น
- ดิสแพชเชอร์ (Dispatcher Process: DNNN) จะอนุญาตให้โพรเซสของตัวให้บริการฐานข้อมูลที่มีจำนวนจำกัดสามารถใช้งานร่วมกันได้ สามารถที่จะทำการสร้างมัลติเพล็กซ์ดิสแพชเชอร์ สำหรับดาต้าเบส อินสแตนซ์แบบโพรเซสเดี่ยว (Single Database Instance) ซึ่งอย่างน้อยที่สุด 1 ดิสแพชเชอร์ จะมีไว้สำหรับโปรโตคอลของเครือข่าย (Network Protocol) ซึ่งถูกใช้สำหรับออราเคิล ดังนั้นผู้จัดการฐานข้อมูลจะต้องมีการกำหนดค่าของจำนวนโพรเซสของดิสแพชเชอร์ (Dispatcher Process) ให้เหมาะสมกับขอบเขตของระบบปฏิบัติการบนจำนวนของการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรเซสและสามารถเพิ่มและลบโพรเซสของดิสแพชเชอร์ได้
ในขณะที่อินสแตนซ์กำลังทำงานอยู่

4.2.4 โครงสร้างหน่วยความจำของออรากิล (ORACLE MEMORY STRUCTURE)

ออรากิลใช้หน่วยความจำในการเก็บ

- รหัสของโปรแกรมที่กำลังถูกเอ็กคิวต์
- ข้อมูลเกี่ยวกับการติดต่อเซสชัน (Session) ถึงแม้ว่าปัจจุบันมันจะไม่ทำงาน
- ข้อมูลที่จำเป็นระหว่างที่จะทำการเอ็กคิวต์ของโปรแกรม
- ข้อมูลซึ่งถูกใช้งานร่วมกันและใช้ติดต่อสื่อสารกันระหว่างโพรเซสของออรากิล
- แคชข้อมูล โดยรวมแล้วมีความสำคัญเกี่ยวกับออรากิลดังนี้
 - ซอฟต์แวร์โค้ดแอเรีย (Software Code Area)
 - เอสจีเอ (SGA)
 - ดาต้าเบสบัฟเฟอร์แคช (Database Buffer Cache)
 - รีโดล็อกบัฟเฟอร์ (Redo Log Buffer)
 - แชร์พูล (Share Pool)
 - พีจีเอ (PGA)
 - สแต็กแอเรีย (Stack Areas)
 - ดาต้าแอเรีย (Data Areas)
 - ซอร์ตแอเรีย (Sort Areas)

4.2.4.1 ซอฟต์แวร์โค้ดแอเรีย (Software Code Area)

ส่วนที่ทำการเก็บรหัสซึ่งกำลังถูกเอ็กคิวต์ซึ่งรหัสของออรากิลก็จะถูกเก็บอยู่ในส่วนของซอฟต์แวร์ โค้ดแอเรียเช่นเดียวกัน โดยทั่วไปแล้วนั้นตำแหน่งที่ตั้งมักจะอยู่แยกต่างจากโปรแกรมของผู้ใช้งาน (User Program) หมายถึงมันเป็นตำแหน่งที่ถูกป้องกัน(Protect) เอาไว้ ขนาดของพื้นที่นี้มักจะไม่เปลี่ยน (Static) ยกเว้นเมื่อมีการเปลี่ยนแปลงหรือติดตั้งใหม่ (Reinstall) ทั้งนี้ทั้งนั้นก็ขึ้นอยู่กับระบบปฏิบัติการด้วยว่ามีการจัดการหน่วยความจำอย่างไรและโดยทั่วไปแล้ว ซอฟต์แวร์ต่าง ๆ นี้สามารถกำหนดให้เป็นการอ่านอย่างเดียว (Read Only) ใช้งานร่วมกันได้ และใช้งานร่วมกันไม่ได้ (Non Shared) ก็ได้

4.2.4.2 เอสจีเอ (SGA)

คือกลุ่มของโครงสร้างหน่วยความจำที่ถูกใช้งานร่วมกันได้หลาย ๆ ผู้ใช้งานระบบซึ่งส่วนดังกล่าวจะทำการบรรจุข้อมูลและข่าวสารการควบคุมสำหรับ 1 ออราเคิลดาต้าเบส อินสแตนซ์ (Oracle Database Instance) ถ้าผู้ใช้งานหลาย ๆ คนทำการเปลี่ยนแปลงอินสแตนซ์เดียวกัน ข้อมูลในเอสจีเอก็จะถูกใช้งานร่วมกันระหว่างผู้ใช้งานเหล่านั้นเอสจีเอจะบรรจุส่วนต่าง ๆ เหล่านี้คือ

- ดาต้าเบสบัฟเฟอร์แคช (Database Buffer Cache)
- รีดูล็อกบัฟเฟอร์ (Redo Log Buffer)
- แชร์พูล (Share Pool)
- การร้องขอและลำดับของการตอบสนอง (Request and Response queue) สำหรับมัลติเทรเด็ตเชิร์ฟเวอร์ (Multi-Threaded Server)
- ดาต้าดิกชันนารีแคช (Data Dictionary Cache)
- ข้อมูลอื่น ๆ

4.2.4.3 ดาต้าเบสบัฟเฟอร์แคช (Database Buffer Cache)

เป็นส่วนของเอสจีเอซึ่งเก็บส่วนที่เป็นสำเนาของฐานข้อมูลที่ถูกอ่านจากดาต้าไฟล์ ซึ่งทุกคนสามารถทำการเข้าถึงข้อมูลนี้ได้ในเวลาพร้อม ๆ กัน

4.2.4.4 รีดูล็อกบัฟเฟอร์ (Redo Log Buffer)

มีโครงสร้างที่เป็นบัฟเฟอร์วงกลม (Circular Buffer) ซึ่งจะเก็บข้อมูลเกี่ยวกับการเปลี่ยนแปลงที่ถูกกระทำขึ้นกับฐานข้อมูล ซึ่งข้อมูลเหล่านี้มีลักษณะที่เป็นรีดูล็อกเอนทิตี (Redo Entity) โดยจะบรรจุข้อมูลที่จำเป็นสำหรับการสร้างใหม่อีกครั้งหรือการกระทำใหม่ของการเปลี่ยนแปลงที่ทำทั้งฐานข้อมูล โดยการเพิ่มข้อมูล (Insert) เปลี่ยนแปลง (Update) ลบ (Delete) แก้ไข (Alter) การสร้าง (Create) รีดูล็อกเอนทิตีจะถูกใช้สำหรับการกู้ข้อมูลให้กลับคืนมา (Database Recovery) รีดูล็อกเอนทิตีจะถูกทำสำเนาโดยโปรแกรมของตัวเองให้บริการฐานข้อมูลจากเพื่อที่หน่วยความจำของผู้ใช้ระบบไปที่รีดูล็อกบัฟเฟอร์ในเอสจีเอ โดยรีดูล็อกเอนทิตีจะถูกทำสำเนาในลักษณะที่เรียงกัน (Sequence) ในบัฟเฟอร์ หลังจากนั้นโปรแกรมเบื้องหลังที่ชื่อล็อกไรท์เตอร์จะเขียนรีดูล็อกบัฟเฟอร์ไปยังรีดูล็อกไฟล์ที่สามารถเข้าถึงได้ (Active online Redo Log File Group) บนดิสก์ ขนาดของรีดูล็อกบัฟเฟอร์จะขึ้นอยู่กับพารามิเตอร์ล็อกบัฟเฟอร์ (Log Buffer Parameter) ซึ่งมีขนาดเป็นไบต์ (Byte) ของรีดูล็อกบัฟเฟอร์ ดังนั้นถ้าหากทราบนแซกชันนั้นมีขนาดใหญ่ การมีบัฟเฟอร์เนื้อที่มาก ๆ จะช่วยลดกิจกรรมของหน่วยรับและแสดงผลข้อมูล (Input/Output)

4.2.4.5 แชร์พูลบัฟเฟอร์ (Share Pool Buffer)

เป็นส่วนของหน่วยความจำที่เรียกว่า เอสจีเอ (SGA) ซึ่งประกอบไปด้วย 3 ส่วนหลัก คือ โลบารีแคช (Library Cache) ดิกชันนารีแคช (Dictionary Cache) และโครงสร้างโดยรวม (Central Structure) ขนาดของแชร์พูลบัฟเฟอร์จะถูกกำหนดโดยพารามิเตอร์ที่เรียกว่า SHARE_POOL_SIZE

4.2.4.6 พีจีเอ (PGA)

เมื่อไคลเอนต์ (Client) และเซิร์ฟเวอร์อยู่ต่างเครื่องกัน พีจีเอ จะถูกจับจองบนตัวให้บริการฐานข้อมูลที่จุดเวลาของการต่อเชื่อม (Connect Time) ถ้าหน่วยความจำไม่เพียงพอ จะทำให้เกิดการผิดพลาด เกิดขึ้น เมื่อทำการต่อเชื่อมได้แล้ว ผู้ใช้งานระบบสามารถใช้เนื้อที่ของพีจีเอ ได้ในขอบเขตจำกัดเท่านั้นพารามิเตอร์ที่มีผลต่อพีจีเอ ได้แก่ OPEN_LINK, DB_FILES, LOG_FILES

4.2.4.7 ซอร์ตแเอเรีย (Sort Areas)

ส่วนของหน่วยความจำซึ่งออราเคิลทำการจับจองไว้สำหรับเรียงข้อมูลเรียกว่า ซอร์ตแเอเรีย (Sort Area) ซอร์ตแเอเรียสามารถใช้ได้อย่างต่อเนื่อง ตามขนาดของข้อมูลที่เพิ่มขึ้นแต่ก็จะไม่เก็บค่าที่ถูกกำหนดจากพารามิเตอร์ SORT_AREA_SIZE ระหว่างที่มีการเรียงข้อมูล ออราเคิลอาจจะกระทำการบางอย่าง ซึ่งไม่ได้ย้ายข้อมูลไปในพื้นที่สำหรับการเรียงข้อมูล ในกรณีนี้ออราเคิลจะทำการลดขนาดของพื้นที่เรียงข้อมูล โดยการการเขียนข้อมูลลงไปที่เซกเมนต์ชั่วคราว (Temporary Segment) บนดิสก์และทำการคืนเนื้อที่ (Deallocate) ด้านของพื้นที่เรียงข้อมูลซึ่งบรรจุข้อมูลดังกล่าว ถ้าข้อมูลที่จะถูกจัดเรียงไม่สามารถเข้าไปในพื้นที่เรียงข้อมูล หลังจากนั้นข้อมูลจะถูกแบ่งออกเป็นส่วนเล็กๆ แล้วเรียงข้อมูลแยกของส่วนเล็กๆ ทั้งหมดหลังจากนั้นก็นำมารวมเข้าด้วยกัน

4.3 SEMANTIC DATA CONTROL

4.3.1 View Management

ORACLE จะยอมให้ผู้ใช้สามารถที่จะทำการแทรกแก้ไข หรือลบผ่าน view ได้โดยมีกฎข้อบังคับ คือ

- ถ้า view ที่ถูกสร้างขึ้นจาก join operation, distinct operation หรือ group by clause จะไม่สามารถแทรก แก้ไข หรือ ลบ ผ่าน view ได้
- ถ้า view ถูกระบุโดยมี with check option row จะไม่สามารถ แก้ไข หรือ ลบ ผ่าน view ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าการตีพิมพ์สิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้า view ถูกระบุให้เป็น not null column (คอลัมน์ที่มีค่าเสมอ เป็น Null ไม่ได้) ที่ไม่มี default clause ใน base table จะไม่สามารถแทรก row ผ่านทาง view ได้
- ถ้า view ถูกสร้างโดย expression จะไม่สามารถแทรก หรือแก้ไขผ่านทาง view ได้โดยรวมแล้ว ORACLE สามารถควบคุมการแก้ไขผ่านทาง view ให้ถูกต้องได้ โดยจะไม่อนุญาตให้มีการแก้ไขตามกรณีค่ากล่าวมาแล้วข้างต้น

4.3.2 Data Integrity

Integrity constraint สามารถแบ่งได้เป็น 2 แบบ คือ

- Entity integrity ซึ่งแบ่งเป็น
 - DEFAULT integrity constraint : สำหรับคอลัมน์ที่ไม่ได้กำหนดระหว่างที่ insert
 - NOT NULL integrity constraint : เป็น integrity constraint ซึ่งใช้กำหนดให้คอลัมน์ที่ต้องการในตาราง จะต้องไม่เป็นค่า NULL
 - UNIQUE integrity constraint : เป็น integrity constraint ซึ่งใช้กำหนด เมื่อไม่ต้องการให้ rows 2 rows ใด ๆ ในตารางมีค่าเดียวกันในคอลัมน์
 - CHECK integrity constraint : เป็น integrity constraint ซึ่งกำหนดคอลัมน์ หรือชุดของคอลัมน์ที่ต้องการให้เป็นไปตามเงื่อนไขที่กำหนด โดยตรวจสอบทุก ๆ rows ในตารางแล้วถ้าพบค่าเงื่อนไขนั้นเป็นเท็จ จะรับ statement นั้นและจะหยุดการเปลี่ยนแปลงข้อมูล
- Referential Integrity ซึ่งแบ่งเป็น
 - PRIMARY KEY integrity constraint
 - แต่ละตารางในฐานข้อมูลจะมี constraint นี้ได้เพียง constraint เดียว
 - rows ในตารางจะถูกกำหนดระบุและอ้างอิงโดย primary key ซึ่งต้องมีความถูกต้องตาม constraint นี้
 - Foreign Key Integrity Constraint
 - เป็น integrity constraint ที่ทำหน้าที่รักษาความสัมพันธ์ระหว่างค่าของ primary key ใด ๆ จะต้องมีความ match กับค่าของ foreign key ในอีกตารางหนึ่งซึ่งการ match ก็สามารถแบ่งได้เป็น
 - match full
 - match partial
 - match null
 - Actions ซึ่งกำหนดโดย referential integrity constraint แบ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Update และ delete restrict : ค่าของคีย์ที่เป็นตัวอ้างอิงให้กับคีย์อื่น ๆ จะถูกเปลี่ยนแปลงหรือลบไม่ได้ เช่น primary key จะถูกเปลี่ยนแปลงหรือลบไม่ได้เพราะยังมีค่านั้นปรากฏอยู่เป็น foreign key ในตารางอื่น ๆ
2. Delete cascade : ถ้า row ที่มีคีย์ที่เป็นตัวอ้างอิงให้กับคีย์อื่น ๆ ถูกลบแล้วทุก rows ที่มี foreign key ที่อ้างอิงจากคีย์นั้นจะต้องถูกลบไปด้วย

4.3.3 Database Trigger

ORACLE ยอมให้ผู้บริหารฐานข้อมูล และผู้ใช้สามารถควบคุม integrity ได้โดยไม่จำเป็นต้องประกาศแบบที่เรียกว่า “non-declarative” โดยใช้ database trigger

Database Trigger หมายถึง database procedure ที่ใช้ควบคุม integrity ซึ่งเก็บไว้และสามารถเรียกใช้ในการ insert, delete หรือ update operations

Database Trigger นั้นถูกสร้างโดยผู้บริหารฐานข้อมูลและผู้ใช้ หากผู้ใช้เป็นผู้สร้างก็ควรอยู่ภายใต้ความควบคุมของผู้บริหารฐานข้อมูล เหตุการณ์ที่ใช้ Database Trigger

- เมื่อใช้ referential integrity โดยไม่ใช้ integrity constraint (database trigger ต่างจาก integrity constraint ตรงที่ database trigger จะใช้ integrity rules โดยไม่ต้องตรวจสอบข้อมูลในตาราง ในขณะที่ integrity constraint จะมีการตรวจสอบข้อมูลในตารางก่อน ดังนั้นควรจะใช้ database trigger เมื่อไม่สามารถใช้ integrity constraint ได้)
- ใช้ referential integrity เมื่อตาราง child และ parent อยู่ต่าง node กันในระบบ distributed database
- ใช้ business rules ที่ยุ่งยากซับซ้อนและไม่สามารถใช้ integrity constraint ได้

4.4 TRANSACTION MANAGEMENT

4.4.1 Concurrency Control

ORACLE จัดการ Concurrency control โดยใช้

4.4.1.1 Lock Mechanism

- Data Lock (DML Lock) เพื่อป้องกันตารางซึ่งมั่นใจได้ว่าข้อมูลในตารางจะถูกต้องตรงกันแม้ว่าจะมีผู้ใช้หลายคนเข้าถึงตารางเดียวกัน แบ่งระดับของ data lock ได้ 2 level คือ
 - Row lock เป็น data lock ที่จะทำการ lock เฉพาะ rows ที่ต้องการ โดยจะจัดเป็น “exclusive data lock” ก็ต่อเมื่อ lock เพื่อต้องการที่จะแก้ไขฐานข้อมูลโดยใช้ statement ต่อไปนี้ : INSERT, UPDATE, DELETE, SELECT พร้อมกับ FOR UPDATE CLAUSE row จะถูก lock เพื่อว่าผู้ใช้อื่น ๆ จะได้ไม่สามารถมาแก้ไข Row ที่เข้าถึงอยู่และ hold lock จนกว่า transaction ที่ hold lock นั้น commit หรือ roll back
 - Table lock Transaction ที่ต้องการแก้ไขตารางด้วย statement ต่อไปนี้ : INSERT, UPDATE, DELETE, SELECT พร้อมกับ FOR UPDATE CLAUSE และ LOCK TABLE table lock มีไว้เพื่อ
 - รักษาระดับของการเข้าถึงตารางแทน transaction
 - ป้องกันการขัดแย้งกับ DDL operations
 - Mode ของ table lock แบ่งออกเป็น 5 mode คือ
 - Row Share Table Lock (RS)
 - เมื่อ transaction hold table lock ในขณะที่มีการ lock rows ในตารางและมีเจตนาที่จะแก้ไข โดยทำให้เกิดระดับของ concurrency มากที่สุดด้วย statement “LOCK TABLE table IN ROW SHARE MODE”
 Operations ที่อนุญาตให้เกิด คือ
 1. จะยอมให้แทรก, เพิ่ม, แก้ไข, ลบ หรือ lock rows ในตารางเดียวกัน
 2. จะยอมให้ transaction อื่น สามารถทำงานใน mode row share, row exclusive share, share, share row exclusive mode ได้ในตารางเดียวกัน
 Operations ที่ห้ามไม่ให้เกิด คือ
 1. Transaction อื่น ๆ จะอยู่ exclusive mode ไม่ได้
 - Row Exclusive Table Lock (RX) เมื่อ transaction hold lock จะต้องเกิดการแก้ไข rows ในตารางแน่นอนด้วย statement “LOCK TABLE table IN ROW EXCLUSIVE MODE”
 Operations ที่อนุญาตให้เกิด คือ

1. จะยอมให้แทรก, เพิ่ม, แก้ไข, ลบ หรือ lock rows ในตารางเดียวกัน
2. จะยอมให้ transaction อื่น lock แบบ row exclusive และ row share mode ในตารางเดียวกันกับที่อีก transaction lock อยู่ได้

Operations ที่ห้ามไม่ให้เกิด คือ

1. Transaction อื่น ไม่สามารถจะ lock แบบ share mode, share row exclusive, exclusive mode
- Share Table Lock (S) เมื่อ transaction hold lock จะยอมให้ transaction อื่น ๆ ทำได้เพียง query และ lock rows ที่ระบุเท่านั้น ซึ่งจะไม่ยอมให้ transaction อื่น ๆ มาแก้ไขด้วย statement "LOCK TABLE table IN ROW SHARE MODE"

Operations ที่อนุญาตให้เกิด คือ

1. จะยอมให้ transaction อื่น ๆ lock แบบ share mode และ row share table mode ได้

Operations ที่ห้ามไม่ให้เกิด คือ

1. Transaction อื่นไม่สามารถจะ lock แบบ share mode, share row exclusive, exclusive mode
- Share Row Exclusive Table Lock (SRX) จะมีเพียง transaction เดียวเท่านั้น ที่จะ hold lock แบบ share row exclusive table lock ด้วย statement "LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE"

Operations ที่อนุญาตให้เกิด คือ

1. จะยอมให้ transaction อื่น ทำ query และ lock rows ที่ระบุโดยไม่มีแก้ไข
2. จะยอมให้ transaction อื่น lock แบบ row share mode เท่านั้น

Operations ที่ห้ามไม่ให้เกิด คือ

1. Transaction อื่นไม่สามารถจะ lock แบบ row exclusive, share, share row exclusive, exclusive mode
- Exclusive Table Lock (X) จะไม่มีการทำ transaction hold lock ไม่ว่าจะเป็น mode ใดด้วย statement "LOCK TABLE IN EXCLUSIVE MODE"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operations ที่อนุญาตให้เกิด คือ

1. จะยอมให้ transaction อื่น ทำ query ที่ไม่มีการ lock เท่านั้น

Operations ที่ห้ามไม่ให้เกิด คือ

1. Transaction อื่นไม่สามารถจะ lock แบบใดได้เลย

Data lock Conversion และ Escalation ORACLE มีความสามารถ convert table lock ซึ่งมีความเข้มงวดน้อยกว่าไปเป็น lock แบบที่มีความเข้มงวดมากกว่าได้ เช่น ถ้ามี transaction ที่ hold แบบ row share table lock หากต้องการแก้ไขตารางจริง ๆ (อาจด้วย statement “SELECT...FOR UPDATE”) ORACLE สามารถเปลี่ยนให้เป็น row exclusive table lock ได้เองโดยอัตโนมัติ

- Dictionary Lock (DDL lock) จะ protect ส่วนที่เป็นคำนิยามของ statement object ในขณะที่ object นั้นอยู่ภายใต้ DDL operations ใน transaction เช่น เมื่อผู้ใช้สร้าง product แล้วผู้ใช้จำเป็นต้องจัดการเกี่ยวกับการป้องกัน เพราะ ORACLE จะจัดการ lock ให้โดยจะ lock object ที่อ้างอิงทั้งหมดไม่ให้ถูกเปลี่ยนแปลงหรือ drop ทั้งจนกว่า procedure นั้นจะ complied ใหม่เสร็จสมบูรณ์ หากมีการ lock เกิดขึ้นไม่ว่าแบบใด

DDL lock แบ่งเป็น 3 กรณี คือ

- Exclusive DDL lock สำหรับ resource ซึ่งต้องการการป้องกันการแทรกแซง ในระหว่างการทำ DDL operation เช่น ในระหว่าง ALTER TABLE จะไม่ทำ DROP TABLE operation หาก object ที่ถูก hold จะต้องรอจนกว่า exclusive DDL lock นั้นถูก released ก่อน
- Share DDL lock สำหรับ resource ซึ่งต้องการป้องกันการแทรกแซง ในระหว่าง DDL operation ซึ่ง อาจทำให้เกิดการ conflict DDL operation เช่นมี operation CREATE PROCEDURE จะทำให้เกิด share DDL lock และ transaction อื่น ๆ จะสามารถ operation ซึ่ง อาจทำให้เกิดการ conflict DDL operation เช่นมี operation CREATE PROCEDURE ที่ reference table เดียวกันได้ หมายเหตุ : หากมีการ lock เกิดขึ้นไม่ว่าเป็นแบบใด transaction อื่นจะไม่สามารถ lock แบบ exclusive DDL lock ได้ ทำให้มั่นใจได้ว่า definition ของ object ที่ใช้ reference จะเป็น definition เดียวกัน ตลอดช่วงของ transaction นั้น
- Breakable DDL lock SQL statement (หรือ PL/SQL program unit) ที่ใช้ร่วมกันจะ hold parse lock เฉพาะ object ที่ถูกอ้างอิงโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SQL statement ซึ่งเป็นผลทำให้จะไม่มี DDL operation ซึ่งขัดแย้งกันเลย

- Internal Lock and Latches จะปกป้องในส่วนภายในของฐานข้อมูลรวมทั้งโครงสร้างของหน่วยความจำลักษณะเป็นไฟล์
- Distributed Lock เพื่อให้มั่นใจว่าข้อมูลและ resource ที่กระจายอยู่ตาม server ต่างๆ ในระบบ ORACLE Parallel Server จะยังคงถูกต้องตรงตามกันอยู่
- Parallel Cache Management Lock เป็น distributed lock ซึ่งครอบคลุม data block (ตาราง และ index block) หนึ่ง block หรือมากกว่า ภายใน buffer ของ cache

4.4.1.2 Multi – Version Consistency Model

ORACLE จะสร้างชุดของข้อมูล ในขณะที่กำลังทำ queries (read) พร้อมกับการแก้ไข (write) ดังนั้นเมื่อเกิดการแก้ไข จะมีการเก็บค่าเดิมของข้อมูลไว้ในฐานข้อมูล และหาก transaction นั้น ยังไม่ commit และมี user query, ORACLE จะ read จาก version ที่จะทำให้ consistency ที่สุด แต่เมื่อ transaction นั้น commit แล้วค่าของข้อมูลจะถูกเปลี่ยนแปลงอย่างถาวร ซึ่งจะเป็น 2 level คือ

- Statement-level read consistency : เพื่อรับรองว่า ข้อมูลหลังจากการทำ single query จะถูกต้องตรงกันก่อนทำ query ดังนั้น statement INSERT, DELETE, UPDATE, จะต้องทำให้ข้อมูลก่อนและหลัง จากการ execute statement จะต้องถูกต้องตรงกัน
- Transaction- level read consistency : เพื่อ guarantee ว่าในสภาวะ multiple-queries ใน transaction หนึ่ง นั้นการ queries ทั้งหมดจะมองเห็น ข้อมูลที่เป็นค่าตรงกันทุกการ queries ที่เวลาใดเวลาหนึ่ง

ORACLE จะจับว่ามี deadlock และเมื่อจับได้แล้วก็จะแก้ปัญหาโดยอัตโนมัติด้วยการ roll back transaction ที่เป็นเจ้าของ statement ที่ทำให้เกิด dead lock ทันทีเพื่อ release lock ซึ่งขัดแย้งกันอยู่

ORACLE จะจับ deadlock โดยมีวิธีต่างกันแบ่งตามประเภทของ deadlock

- ใช้วิธี “waits for graph” เพื่อจับ deadlock แบบ local deadlock
- ใช้วิธี “time-out” เพื่อจับ deadlock แบบ global deadlock

4.5 Backup และ Recovery

วิธีการ Backup และ Recovery ใน ORACLE นั้นจะใช้ log-based recovery โดยเมื่อผู้ใช้ส่ง commit statement มาแล้ว commit record ก็จะถูกนำไปเก็บ redo log buffer ทันที แต่จะคอยการเขียนข้อมูลลงดิสก์ จนกว่าจะมีการเปลี่ยนแปลงใน data buffer

- Group commit ในเวลาที่มี activity มาก ๆ log writer (LGWR) อาจทำการเขียน redo log file โดยใช้ group commit ซึ่งเป็นการประหยัด disk I/O โดยรวมเอา redo entry จากหลาย ๆ ผู้ใช้ แล้วจึงเขียนลงดิสก์ทีเดียว
- Two-phase commit คือในระบบ multi-user ถ้า site หนึ่งทำการ commit หรือ rollback อีก site หนึ่งก็ต้อง commit หรือ rollback ตามเช่นกัน

ขั้นตอนในการทำ recovery

ORACLE จะมีขบวนการในการ recovery 2 ขั้นตอนคือ

- Redo Log and Rolling Forward

การ roll forward เป็นการย้อนกลับการเปลี่ยนแปลงทั้งหมดที่ได้เก็บไว้ใน redo log ไปยังไฟล์ข้อมูล และเนื่องจากข้อมูลในการ rollback จะถูกเก็บเอาไว้ใน redo log ด้วย ดังนั้น roll forward ต้องไปเรียกใช้ rollback segment ด้วย ซึ่งเป็นขั้นตอนของการ Rollback

Rolling forward จะทำการจัดการกับ redo log file ที่จำเป็นสำหรับการนำฐานข้อมูลกลับคืนมา โดยจะใช้ทั้ง online redo log file และ archive redo file หลังจาก roll forward แล้วภายในฐานข้อมูลจะมีทั้งการเปลี่ยนแปลงที่ commit แล้วทั้งหมดและการเปลี่ยนแปลงที่ยังไม่ commit ที่ถูกเก็บอยู่ใน redo log file

- Rollback Segment and Rolling Back

Rollback segment จะถูกใช้เพื่อที่จะยกเลิก (undo) uncommitted transaction ที่ถูกทำการเปลี่ยนแปลงมาก่อนหน้านี้ โดย rolling forward phase

การทำ roll forward นั้นการเปลี่ยนแปลงใด ๆ ที่ยังไม่ commit จะต้องถูก undo หลังจาก redo log file ได้ทำการย้อนกลับการเปลี่ยนแปลงที่ทำต่อฐานข้อมูลทั้งหมดแล้ว จะต้องมีการใช้ rollback segment ด้วยเพื่อที่จะ undo transaction ที่ยังไม่ได้ commit เนื่องจาก transaction ที่ยังมิได้ commit นั้นการเปลี่ยนแปลงทั้งหมดยังอยู่ใน redo log buffer โดยยังไม่ได้เขียนลงใน redo log file โดย process นี้เรียกว่า Rolling back ซึ่งสามารถใช้ integrity constraint นี้กับโปรแกรมประยุกต์ที่เกี่ยวกับฐานข้อมูล โดยไม่ต้องเขียนส่วนนี้ขึ้นเอง

Immediate user feedback : ก่อนที่จะเรียกใช้คำสั่ง SQL จะมีการตรวจสอบความถูกต้องของข้อมูลก่อนโดยใช้ integrity constraint ที่เก็บไว้ในพจนานุกรมข้อมูล

4.6 QUERY PROCESSING

4.6.1 Storage Structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORACLE มีวิธีในการจัดการการเก็บข้อมูลในตารางหลายวิธีดังนี้

4.6.1.1 Cluster

Cluster หมายถึงกลุ่มของตารางซึ่งใช้ data block ร่วมกัน ในลักษณะใช้คอลัมน์ร่วมกันโดยตาราง 2 ตารางซึ่งใช้คอลัมน์ร่วมกันจะถูกเก็บไว้ใน data block เดียวกัน

Clutter Key คือคอลัมน์หรือชุดของคอลัมน์ซึ่งใช้ตารางแบบ cluster ร่วมกันโดยคอลัมน์ดังกล่าวนี้ จะต้องระบุตอนสร้างและคอลัมน์เดียวกันนี้จะอยู่ในตารางซึ่งจะเพิ่มเข้า Cluster แบบอัตโนมัติ โดยที่

- Clutter Key มีได้ไม่เกิน 16 คอลัมน์ และค่าของ Clutter Key จะไม่เกิน 1/3 ของเนื้อที่ของข้อมูลใน block ของข้อมูล
- หากค่าของข้อมูลใน cluster คอลัมน์มีการแก้ไข จะต้องมีการเปลี่ยนแปลงตามตำแหน่งที่เก็บซึ่งอยู่ในระดับกายภาพ

4.6.1.2 Hash

การใช้ hash จะต้องสร้าง hash cluster และเรียกตารางเข้าไปใน cluster โดย row ของตารางใน hash cluster จะเก็บและดึงข้อมูลตามผลที่ได้จาก hash function

- “hash function” ใช้ในการสร้างค่าตัวเลข ที่มีความกระจายซึ่งค่านี้จะเรียกว่า “hash value”
- คีย์ของ hash cluster อาจเป็นคอลัมน์ หรือชุดของคอลัมน์
- ORACLE จะจัดการเก็บ row ใน hash cluster โดยใช้ cluster key value ที่ได้จาก hash function
- ในการหาหรือจัดเก็บ row จะต้องใช้อย่างน้อย I/O 2 ครั้ง (มักจะมากกว่านี้) โดยใช้ I/O 1 ครั้งหรือมากกว่าในการหาและเก็บคีย์ value ใน index และใช้ I/O มากกว่า 1 ครั้ง สำหรับอ่าน หรือเขียน row ในตารางหรือ cluster

4.6.2 Data Independence ของ ORACLE

4.6.2.1 Physical Data Independence

ORACLE มีคุณสมบัตินี้คือไม่ว่าจะเปลี่ยนลักษณะการเข้าถึงไฟล์ในระดับล่างเป็นแบบใด ก็จะไม่กระทบเฉพาะในระดับล่างเท่านั้นระดับ โปรแกรมประยุกต์ที่เหนือกว่าไม่จำเป็นต้องทำ การเปลี่ยนแปลง

4.6.2.2 Logical Data Independence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORACLE มีคุณสมบัติข้อนี้คือ ผู้บริหารฐานข้อมูล สามารถที่จะเปลี่ยนแปลงตาราง คอลัมน์ ตัดคอลัมน์ เพิ่มตาราง สิ่งเหล่านี้จะไม่มีผลต่อการเข้าถึงตารางของผู้ใช้ โดยไม่ต้องมีการแก้ไขโปรแกรมใหม่

4.6.3 Query Optimization

Optimization ของ ORACLE มีจุดมุ่งหมายเพื่อหาเส้นทางที่มีประสิทธิภาพที่สุดในการประมวลผล SQL statement โดยวิธีในการทำ Optimization สำหรับ SQL statement นั้นมีด้วยกัน 2 วิธีคือ

- Rule-based approach

optimizer จะเลือกเส้นทางที่ใช้ในการประมวลผลโดยดูจาก access path และพิจารณาการทำงานนั้นมีค่าของ rank ต่ำซึ่งจะแสดงถึงการเข้าถึงข้อมูลที่เร็วกว่า โดย ORACLE ได้ กำหนด rank ไว้ดังนี้

Rank	Access Path
1	Single row by ROWID
2	Single row by cluster join
3	Single row by hash cluster key with unique or primary key
4	Single row by unique or primary key
5	Cluster join
6	Hash cluster key
7	Index cluster key
8	Composite index
9	Singer-column index
10	Bounded rang search on indexed columns
11	Unbounded rang search on indexed columns
12	Sort-merge join
13	MAX of MIN of indexed column
14	ORDER BY on indexed column
15	Full scan table

- Cost-based approach

ใช้วิธีการเก็บสถิติไว้เป็นจำนวนมาก ซึ่งจะรวมรายละเอียดที่เกี่ยวกับตัวข้อมูล, index, จำนวนแถว, จำนวนหลัก และสถิติที่เกี่ยวกับตาราง โดย optimizer จะเก็บค่าที่เป็นไปได้ของข้อมูล รวมทั้งค่าสูงสุดของข้อมูลที่มีอยู่ซึ่งไม่ได้เก็บข้อมูลในรูป histogram

4.6.4 Stored Procedure and Function

เป็น object ที่รวมเอากลุ่มของ SQL และ PL/SQL programming Language statement เข้าด้วยกันในทาง logical เพื่อแสดงการทำงานเฉพาะขึ้นมา และเก็บ code ที่ compile แล้วนี้ไว้ใน share pool ของ SGA และ compiled version นี้จะอยู่ใน shared pool ตาม LRU algorithm

Procedure และ Function นี้จะถูกสร้างขึ้นในระดับของผู้ใช้และเก็บไว้ในฐานข้อมูลสำหรับงานต่อไป โดยจะถูกประมวลผลอย่าง interactive โดยใช้ ORACLE tool เช่น SQL*DBA หรืออาจถูกเรียกตรงจากโปรแกรมประยุกต์ของฐานข้อมูล เช่น SQL form, procedure อื่น ๆ หรือ trigger

นอกจากนี้ยังสามารถนำ procedures, function, cursors และ variable ที่สัมพันธ์กันมา รวมเข้าด้วยกันได้รวมเป็นหนึ่งหน่วยเรียกว่า package และถูกนำไปเก็บไว้ในฐานข้อมูลเพื่อการใช้งานต่อไป

ประโยชน์ของ stored procedure

- ทางด้าน security เนื่องจาก stored procedure สามารถช่วยควบคุมความปลอดภัยของข้อมูล ดังนั้นจึงสามารถจำกัดการทำงานระหว่างผู้ใช้และฐานข้อมูลของแต่ละผู้ใช้ได้ เพราะในการเรียกใช้ procedure ORACLE จะมีการตรวจสอบสิทธิในการใช้ข้อมูลของผู้ใช้
- Stored procedure สามารถเพิ่มประสิทธิภาพของฐานข้อมูลเนื่องจากสามารถลดข้อมูลที่ต้องการส่งข้ามเครือข่ายเมื่อเทียบกับการส่ง SQL statement เข้าไปและยังไม่ต้องมีการ compile ซ้ำอีกในการ execute code นอกจากนี้ ถ้าเกิดมี procedure อยู่แล้วใน shared pool ก็ไม่จำเป็นต้องเรียก Procedure ขึ้นมาจากดิสก์ สามารถประมวลผลได้ทันที
- ประหยัดหน่วยความจำเนื่องจาก stored procedure ใช้ประโยชน์ของ shared memory ดังนั้นจึงไม่เกิดความซ้ำซ้อนของคำสั่งในหน่วยความจำ เมื่อมีผู้ใช้หลายๆ คนเรียกใช้งานร่วมกัน
- Stored procedure จะช่วยเพิ่มประสิทธิภาพ เนื่องจากในการออกแบบโปรแกรมประยุกต์โดยใช้ชุดของ procedure ทำให้สามารถที่จะเลี่ยงความซ้ำซ้อน เช่น procedure จะถูกเขียนขึ้นมาเพื่อแก้ไข แทรก และลบ row ออกจากตาราง ซึ่งโปรแกรมประยุกต์จะสามารถเรียกใช้ procedure เหล่านี้โดยไม่ต้องเขียน statement ใหม่ และเกิดวิธีการจัดการข้อมูลเปลี่ยนไป เพียงทำการแก้ไข procedure ใหม่ เท่านั้นไม่ต้องทำการแก้ไขโปรแกรมประยุกต์ที่ใช้ procedure นั้น
- Stored procedure จะช่วยเพิ่ม integrity และ consistency ของโปรแกรมประยุกต์ให้ถูกแน่นอนขึ้น คือแต่ละ procedure หรือ function ต้องการการทดสอบเพียงครั้งเดียวเพื่อที่จะรับประกันว่าได้ผลลัพธ์ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ADDITIONAL TOOLS

Cooperative Development Environment (CDE) เป็นชุดเครื่องมือที่ใช้สำหรับพัฒนาระบบงานคอมพิวเตอร์ ซึ่งจะครอบคลุมทุกขั้นตอนในการพัฒนาโปรแกรมประยุกต์ โดยเริ่มตั้งแต่การ reengineer กระบวนการทำงาน (business process) และการให้คำจำกัดความตามข้อกำหนดของโปรแกรมที่จะใช้งาน (application requirement definition) รวมถึงขั้นตอนการวิเคราะห์ออกแบบพัฒนางานและการแก้ไขเปลี่ยนแปลง โดยใช้ได้ตั้งแต่ single-user จนกระทั่งถึง enterprise-wide system

CDE ประกอบด้วย เครื่องมือ 3 ประเภทได้แก่

- CASE tools
- Application development
- Office automation and end user

4.7.1 CASE Tools

ORACLE มี case family ซึ่งจะรวบรวมเอาเครื่องมือที่ใช้สำหรับการพัฒนา คือ ตั้งแต่การกำหนดวิธีไปจนถึงการสร้างออกมาเป็นโปรแกรม โดย ORACLE/CASE ประกอบด้วย

- CASE Method เป็นเครื่องมือที่ช่วยในด้านวิธีการพัฒนาระบบ โดยใช้วิธีของ top-down design ซึ่งใช้การระบุขั้นตอนสำคัญ ๆ เพื่อเป็นช่วยในการออกแบบ
- CASE Dictionary (Shared Portable Repository) เป็นที่เก็บข้อมูลสำหรับทุก ๆ ขั้นตอน และข้อมูลเหล่านั้นจะสามารถถูกควบคุมได้โดยง่ายผ่านทาง form interface หรือผ่านเครื่องมือทางกราฟฟิค CASE Dictionary มีความสามารถในการสร้างเอกสารประกอบโดยอัตโนมัติ โดยสามารถเลือกได้ถึง 3 ชนิดคือ predefined reports, impact analysis reports และ custom reports โดยใช้ SQL reportwriter สำหรับการสร้างเอกสารของระบบทำให้ผู้พัฒนาสามารถสร้างรายงานเองได้อย่างรวดเร็ว และ CASE Dictionary นี้สามารถที่จะรันได้หลาย platform ด้วย
- CASE Designer เป็นเครื่องมือที่ใช้กำหนดความต้องการของผู้ใช้และระบบ โดยใช้ graphic model ในขณะที่ผู้ใช้งานดูรูปลงใน diagram ตัวใดตัวหนึ่ง CASE Designer จะทำการแก้ไข CASE Dictionary ในทันทีโดยไม่มีการสูญเสีย concurrency CASE Designer จะสนับสนุนการทำงานในระบบ multi-user คือผู้ใช้หลาย ๆ คนสามารถที่จะเข้าถึง CASE Designer ได้ในเวลาเดียวกัน แม้จะเป็นการทำงานบน models ที่เหลื่อมซ้อนกันก็ตาม และผู้ใช้แต่ละคนสามารถที่จะเปิด windows ได้โดยไม่จำกัดหลาย ๆ พื้นที่ของ model ที่ทำโดยแต่ละ window จะมีการประมวลผลที่แยกออกจากกัน นอกจากนี้ CASE Designer ยังจะสามารถเคลื่อนย้ายไปรันบนสภาพแวดล้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล้อมทางกราฟฟิคต่าง ๆ ได้เช่น X-window, DEC window และ presentation manager เป็นต้น

- CASE Generator Series เป็นเครื่องมือที่ใช้เพื่อการสร้าง source code โดยใช้ข้อมูลที่เกิดขึ้นใน CASE dictionary มีทั้งสิ้น 4 ตัว คือ

1. CASE Generator for SQL Forms

ใช้ในการสร้างโปรแกรมประยุกต์จาก diagram สำหรับ SQL Forms โปรแกรมประยุกต์ที่ถูกสร้างขึ้นนี้จะสามารถรันได้ทุก ๆ แพลตฟอร์มที่ ORACLE สนับสนุน output ของ generator ตัวนี้เป็น functional interactive application ที่จะรวมเอา SQL statement ต่าง ๆ เช่น enter, edit, retrieve ข้อมูลจากฐานข้อมูลไว้ด้วยและยังรวม logic ต่าง ๆ ที่ใช้ในการบังคับกฎทางธุรกิจที่ถูกกำหนดไว้ใน CASE dictionary ทำให้เราไม่ต้องเขียน code ซ้ำในส่วนของการควบคุมหน้าจอ, การตรวจสอบข้อมูลและ integrity constraint ซึ่งจะถูกบังคับใช้โดยอัตโนมัติและโดยการให้ SQL form designer ประกอบ ทำให้เราสามารถที่จะ override default เช่น เปลี่ยนรูปหน้าจอ เพิ่มความสามารถอื่น ๆ เข้าไปในโปรแกรมประยุกต์โดยเขียน routines เพิ่มเข้าไป โปรแกรมประยุกต์ที่ถูกสร้างขึ้นจะ compatible กับ block-mode, character-mode และ bitmap devices

2. CASE Generator for SQL Forms/SQL Menu

เป็น CASE generator ที่สามารถจะสร้างทั้ง form layout และ menu layout และการเข้าถึงฐานข้อมูลได้โดยอัตโนมัติ และโปรแกรมประยุกต์ที่สร้าง สามารถจะสร้างขึ้นใหม่ได้ทุก ๆ เวลาที่ความต้องการของผู้ใช้เปลี่ยนไป ในการสร้างโปรแกรมประยุกต์ CASE generator จะมีการสร้างระบบเมนูเต็มรูปแบบให้ด้วย คือมีทั้ง menu item help, menu navigation, menu access to form, other menus และมีลักษณะของเมนูให้เลือก เช่น full screen, pull down เป็นต้น

3. CASE Generator Database Tables

เป็น CASE generator ที่สามารถออกแบบรูปแบบของตารางให้กับผู้ใช้ โดยจะ map มาจาก diagram ที่ผู้ใช้สร้างขึ้นมา

4. CASE Generator for SQL Report Writer/SQL Plus

เป็น reporting tool ที่ใช้ร่วมกับ CASE dictionary เพื่อสร้าง layout และ report ตามที่ผู้ใช้งานต้องการ โดย CASE generator จะสร้างโปรแกรมประยุกต์สำหรับออกรายงานจากรายละเอียดต่าง ๆ ที่ถูกเก็บใน CASE dictionary

4.7.2 Application Development Tools

4.7.2.1 PL/SQL

เป็น procedural language ที่เพิ่มเติมจาก SQL มาตรฐาน ทำให้เราสามารถที่จะประมวลผล multi-statement หรือ multi-transaction PL/SQL procedure โดยใช้คำสั่งเดียว ไปยัง ORACLE จึงทำให้ลดการติดต่อสื่อสารบนเครือข่ายแบบ client/server นอกจากนี้ PL/SQL เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังสามารถที่จะประมวลผลการคำนวณของข้อมูล และ control logic ได้โดยไม่ต้องส่งคำสั่งไปที่ฐานข้อมูลซึ่งเป็นการเพิ่มประสิทธิภาพของโปรแกรมประยุกต์เราสามารถเขียน PL/SQL เขียนโปรแกรมประยุกต์ร่วมกับ ORACLE tool อื่น ๆ ได้

ใน PL/SQL procedure สามารถที่จะมี SQL statement มาตรฐานและ conditional control statement ได้ โดยที่ PL/SQL procedure ที่กำหนดขึ้นมา จะสามารถถูกเรียกใช้ได้จาก form อื่น ๆ ด้วย PL/SQL จะสามารถรันได้บนทุก ๆ แพลตฟอร์ม ที่รัน ORACLE ด้วย

4.7.2.2 Precompilers

จะอนุญาตให้ฝัง SQL ลงในโปรแกรมหลักที่เขียนด้วย Ada, C, Cobol, Fortran, Pascal, PL/1 ได้ และสามารถจะควบคุมการติดต่อกับระบบจัดการมาตรฐานข้อมูลได้ด้วย

ORACLE precompilers เป็น full ANSI-compatible ซึ่งโปรแกรมประยุกต์ที่พัฒนาโดยใช้ precompilers สามารถที่จะรันได้ทั้งบน ORACLE ANSI-compatible SQL based DBMS และ DB2 จึงสามารถที่จะสร้าง customized application ได้ เช่น เราสามารถจะออกแบบโปรแกรมประยุกต์ให้ดึงข้อมูลจาก ORACLE บนเมนเฟรมแล้วให้ประมวลผลข้อมูลบนไมโครคอมพิวเตอร์และให้แสดงข้อมูลบนเวิร์คสเตชันได้

4.7.2.3 SQL Plus

ใช้ในการทำ query, define, และ control data ที่เก็บใน ORACLE DBMS ซึ่งมีความสามารถเช่นเดียวกับ ORACLE/SQL และ PL/SQL

ด้วย syntax ง่าย ๆ ของ SQL plus ผู้ใช้สามารถที่จะเข้าถึงข้อมูลได้อย่างรวดเร็ว และยังสามารถที่จะดึงข้อมูล, ทำการคำนวณและสร้าง adhoc report ได้

SQL plus เป็น dynamic SQL-based queries คือผู้ใช้จะสามารถเปลี่ยน section ของ query ได้ขณะรันเราสามารถจะเพิ่ม PL-SQL block ลงใน SQL statement ได้ (เป็น procedure logic) โดยการใช้ SQL PLUS

4.7.3 Office Automation and End-User Tools

4.7.3.1 ORACLE Data Query

ถูกออกแบบมาเพื่อให้ผู้ใช้ทำการระบุและรัน adhoc queries โดยไม่จำเป็นต้องรู้ SQL ในการสร้าง queries ผู้ใช้สามารถเลือกตาราง และคอลัมน์จาก lists การใช้งานเป็นแบบ step by step โดยใช้วิธี fill in the blank ข้อมูลที่ผู้ใช้ป้อนเข้ามาจะถูกตรวจสอบทุกขั้นตอน หากพบข้อผิดพลาดจะแก้ไขในทันที

ORACLE data query จะอนุญาตให้มีการสร้าง report หลาย ๆ แบบได้จาก 1 query ซึ่งสามารถคำนวณข้อมูลทางสถิติ และสามารถจะถูกเรียงลำดับข้อมูลที่ได้หลายวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะที่ query ถูก execute อยู่สามารถจะแสดงผลขึ้นจอ เขียนลงไฟล์ หรือส่งตรงไปยัง printer ได้โดยตรง

ORACLE data query สามารถจะรันได้บนหลาย ๆ แพลตฟอร์ม (รวมทั้ง client/server configuration) และสนับสนุนทั้ง character-mode และ block-mode terminal

ORACLE data query เป็นเครื่องมือแบบ read only ซึ่งผู้ใช้สามารถแทรก, แก้ไข, หรือลบข้อมูลได้และ ผู้บริหารฐานข้อมูลสามารถจำกัดการทำ query ได้

4.7.3.2 SQL QMX

เป็น query manager และ report writer ซึ่งสามารถเอา browser, query facilities และ report writer ไว้ใน package เดียวกัน

ในการใช้ SQL QMX ผู้ใช้สามารถใช้ function key-driven menus หรือจะใส่คำสั่งที่ยังไม่สมบูรณ์ โดย SQL QMX จะเติมส่วนที่เหลือให้โดยผู้ใช้ไม่จำเป็นต้องรู้ command syntax สำหรับ report writer จะเป็นแบบ fill-in-the-blank formatting panel SQL QMX browser จะช่วยให้ผู้ใช้สามารถหาตารางหรือ view ได้โดยไม่ต้องสร้าง query และเมื่อผู้ใช้สร้าง report browser ก็จะทำให้ผู้ใช้สามารถที่จะ preview report บนหน้าจอก่อนจะ print ได้

ผู้ใช้สามารถที่จะเลือกเขียน SQL statement แทนการใช้ prompt panel ได้ และ SQL QMX ยังสนับสนุน macro procedure และ substitution variables ด้วย

4.7.3.3 SQL Form และ SQL Menu

โปรแกรมประยุกต์ที่เขียนด้วย SQL Form และ SQL Menu จะสามารถรันได้บนทุก ๆ แพลตฟอร์ม ที่ ORACLE สนับสนุน โดยสามารถสร้างโปรแกรมประยุกต์โดยไม่ต้องเขียน code แต่ระบุคุณสมบัติของโปรแกรมประยุกต์แทน โดยใช้ menu-driven user interface specification ดังกล่าวร่วมกับข้อมูลที่เก็บใน ORACLE data dictionary ซึ่งจะถูกใช้ในการสร้างโปรแกรมประยุกต์และสามารถใช้ร่วมกับ PL/SQL routine รวมทั้งสามารถเพิ่ม 3GL routine ลงในโปรแกรมประยุกต์ได้ด้วย

4.7.3.4 SQL Text Retrieval

เป็นความสามารถที่สามารถที่จะเพิ่มเข้าไปในโปรแกรมประยุกต์ของ ORACLE โดยเป็นการขยายส่วนของ SQL ซึ่งผลที่จะได้รับเป็น common language สำหรับที่เก็บของข้อมูล และตัวอักษร SQL text retrieval จะเก็บข้อมูลทั้งหมดรวมทั้งตัวอักษรต้นแบบและ index โดยผู้ใช้จะเข้าถึงข้อมูลทุกตัวได้ภายในการควบคุมของ ORACLE ผู้ใช้จะสามารถเข้าถึง SQL text retrieval software tools ได้โดยผ่านทาง SQL form หรือ 3GL programs

SQL text retrieval ทำให้การค้นหาข้อมูลมีประสิทธิภาพขึ้นโดย query สามารถที่จะรวมเอา words, concepts และโครงสร้างของฟิลด์ข้อมูลไว้ได้ และสามารถที่จะใช้ตัวอักษร

จากที่ต่าง ๆ เช่น external file, screen input, export ตัวอักษรระหว่างโปรแกรมประยุกต์ SQL text retrieval สะดวกในการเคลื่อนย้ายบนหลาย ๆ แพลตฟอร์ม

4.7.3.5 SQL Reportwriter

เป็นเครื่องมือที่ทำให้ผู้ใช้สามารถที่จะสร้างรายงานได้หลายรูปแบบ โดยใช้ข้อมูลจากหลายฐานข้อมูลทั้ง ORACLE และ non-ORACLE เช่น DB2 โดยไม่ต้องเขียนโปรแกรมได้เพียงแต่ระบุ report specification ลง forms (ซึ่งเป็นรูปแบบ spread sheet) SQL Reportwriter จะมี default ให้ตำแหน่งคอลัมน์, column heading และ display format เราสามารถที่จะ override default ได้ถ้าต้องการ SQL Reportwriter สามารถใช้เชื่อมกับ ORACLE tools อื่น ๆ ได้

4.8 DISTRIBUTED DATABASE MANAGEMENT

การเชื่อมต่อระหว่างโปรแกรมประยุกต์กับฐานข้อมูลยังคงถูกจัดอยู่ภายในข้อจำกัดของการเชื่อมต่อของคอมพิวเตอร์ 2 เครื่อง SQL*Net ของ ORACLE ได้ทำลายเส้นกันของโปรโตคอลสำหรับโปรแกรมประยุกต์ client/server Client สามารถทำงานได้โดย server ไม่ต้องพึ่งพาโปรโตคอลของระบบเครือข่าย SQL*Net เป็นระบบเครือข่ายที่เหมือน ORACLE โดยอนุญาตให้มีการเชื่อมต่อระหว่างโปรแกรมประยุกต์ client และ server ของฐานข้อมูลซึ่งแยกออกจากการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ที่กำลังทำงานอยู่

4.8.1 Client/Server Computing

SQL*Net เป็นโปรแกรมควบคุมระยะไกลที่ใช้โดยโปรแกรมประยุกต์ของผู้ใช้และฐานข้อมูล หรือระหว่างข้อมูลหลาย ๆ ตัว และด้วยโปรแกรมประยุกต์ของ SQL*Net ฐานข้อมูลสามารถที่จะกระจายทางด้านกายภาพออกไป โดยแยกเครื่องที่แตกต่างกันและเริ่มติดต่อกันเป็นแบบ local

4.8.2 Network Transparency

SQL*Net โปรแกรมประยุกต์ client ทั้งหมด จะทำหน้าที่เหมือนกับว่าฐานข้อมูลระยะไกลเป็นของตัวเอง ซึ่งทำให้ระบบเครือข่ายสามารถรวมกัน และดูการเชื่อมต่อกันผ่านทางโปรแกรมประยุกต์ client ได้

4.8.3 Multinetwork Data Access

Client และ server สามารถกระจายออกไปได้ไม่เพียงแต่บนเครื่องที่แตกต่างกันแต่รวมถึงเครือข่ายที่แตกต่างกัน เครื่อง client/server ต้องการใช้เพียงโปรโตคอลพื้นฐานที่มีมาให้ในตัวแทนนั้น และยังสามารถติดต่อกับโปรแกรมประยุกต์ด้วยกัน Client และ server ทั้งหมดในระบบเครือข่าย ซึ่งสามารถแบ่งข้อมูลผ่านทางความสัมพันธ์แบบ client/server หรือ server/server ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8.4 The Interchange for Cross Protocol Support

SQL*Net มี multi-protocol interchange เพื่อที่จะใช้ในการเชื่อมต่อแบบที่ไม่เหมือนระบบเครือข่าย interchange แบบ single ระหว่างระบบเครือข่ายที่มากกว่า 2 ระบบจะชี้ความต้องการในระบบเครือข่ายรวมจากโปรแกรมประยุกต์แบบ transparent network substrate และส่งผ่านไปยังจุดหมายปลายทาง การรวมกันของ SQL*Net กับ interchange จะยอมให้โปรแกรมประยุกต์ของ client สามารถติดต่อกับ server ของฐานข้อมูลใดก็ได้ โดยไม่ต้องใช้โปรโตคอลของระบบเครือข่ายบนเครื่องแต่ละเครื่อง

Protocol Support

- TCT/IP
- DECnet
- APPC (LU6.2)
- ISO/OSI
- SPX/IPX
- Named Pipes
- NetBIOS
- Banyan Vines

นอกจากนี้ยังสนับสนุน SQL*Net Stream และโปรโตคอลแบบ Packet-based ดังนี้

- VTAM (LUO)
- 3270 (LU2)
- Asynchronous (LU1, RS232)
- X.25

หลักการทำงานของ Oracle Designer/2000

5.1 Oracle Designer/2000

เทคโนโลยีทางด้านซอฟต์แวร์คอมพิวเตอร์ได้เปลี่ยนแปลงไปรวดเร็วมาก มีซอฟต์แวร์ที่เก่งฉลาดเกิดขึ้นมาอยู่เรื่อยๆ จากที่เคยเขียนโปรแกรมด้วย Third-Generation Language ต่อมาเปลี่ยนเป็น Fourth-Generation Language จนเมื่อหลายปีผ่านมานี้ CASE TOOLS ได้เริ่มมีความสำคัญเพิ่มขึ้นมาเรื่อย ๆ จนเป็นที่ยอมรับกันว่าเป็นเทคนิคการพัฒนาระบบงานคอมพิวเตอร์ที่ฉลาดที่สุดในยุคปัจจุบันและน่าจะเป็นเครื่องมือในการพัฒนาระบบงานที่ใช้กันอย่างแพร่หลายในอนาคต

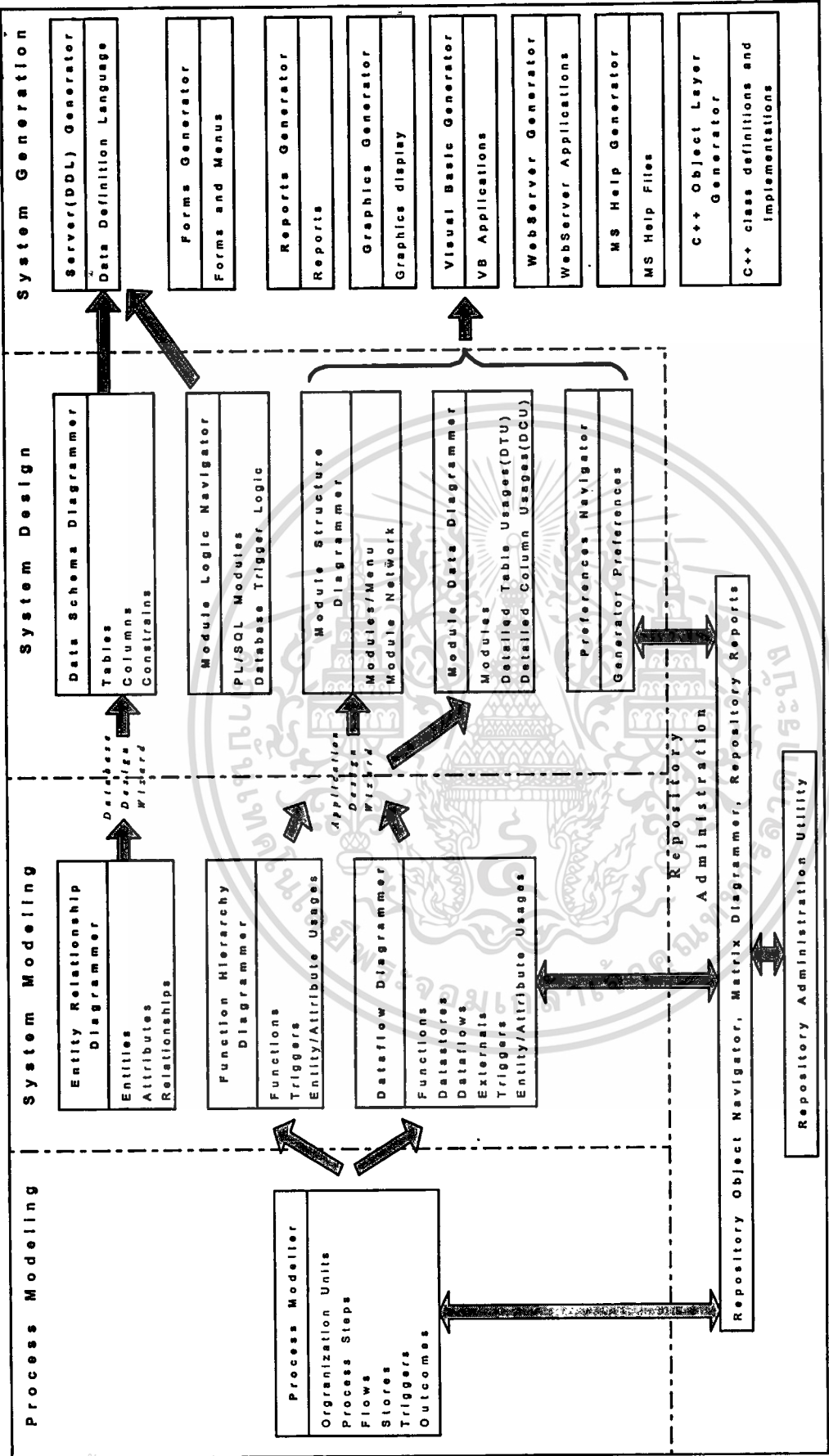
สำหรับบริษัท Oracle ล่าสุดได้ออก CASE TOOLS รุ่นใหม่ล่าสุดเรียกว่า Designer/2000 ปัจจุบันเป็น Version 6 แล้วและในรุ่นนี้ทาง Oracle ได้ทำการพัฒนาตัว Designer/2000 ให้สามารถรองรับการออกแบบระบบฐานข้อมูลแบบอ็อบเจกต์ ivo ด้วย แต่ถ้าผู้ใช้อย่างยังไม่มีความรู้ทางด้านอ็อบเจกต์ ก็ยังที่จะสามารถใช้การออกแบบระบบฐานข้อมูลแบบรีเลชันนอลคาค้าเบสมาใช้ได้โดยไม่เกิดความผิดพลาดแต่อย่างใด

Oracle Designer/2000 เป็น CASE TOOLS รุ่นใหม่ล่าสุดของบริษัท Oracle เป็นเครื่องมือที่ช่วยให้ผู้พัฒนาระบบงานสามารถพัฒนาระบบงานได้ง่ายและรวดเร็วขึ้นสามารถที่จะพัฒนาระบบที่ซับซ้อน ประกอบด้วยเครื่องมือในการเขียน Model ในรูปแบบต่าง ๆ เช่น Business Process Reengineering (BPR) ช่วยในการวิเคราะห์ระบบงานและเขียน Diagram, Dataflow Diagram, Entity Relationship Diagram เรียกได้ว่าในขั้นตอนการพัฒนาระบบงานทุกขั้นตอนสามารถนำ Designer/2000 นี้ไปช่วยได้ทั้งในการเขียน Diagram ต่าง ๆ การออกรายงาน จนกระทั่งสร้างตัวโปรแกรมในลักษณะที่เป็นโปรแกรมต้นแบบได้โดยง่ายให้

5.2 องค์ประกอบของ Designer/2000

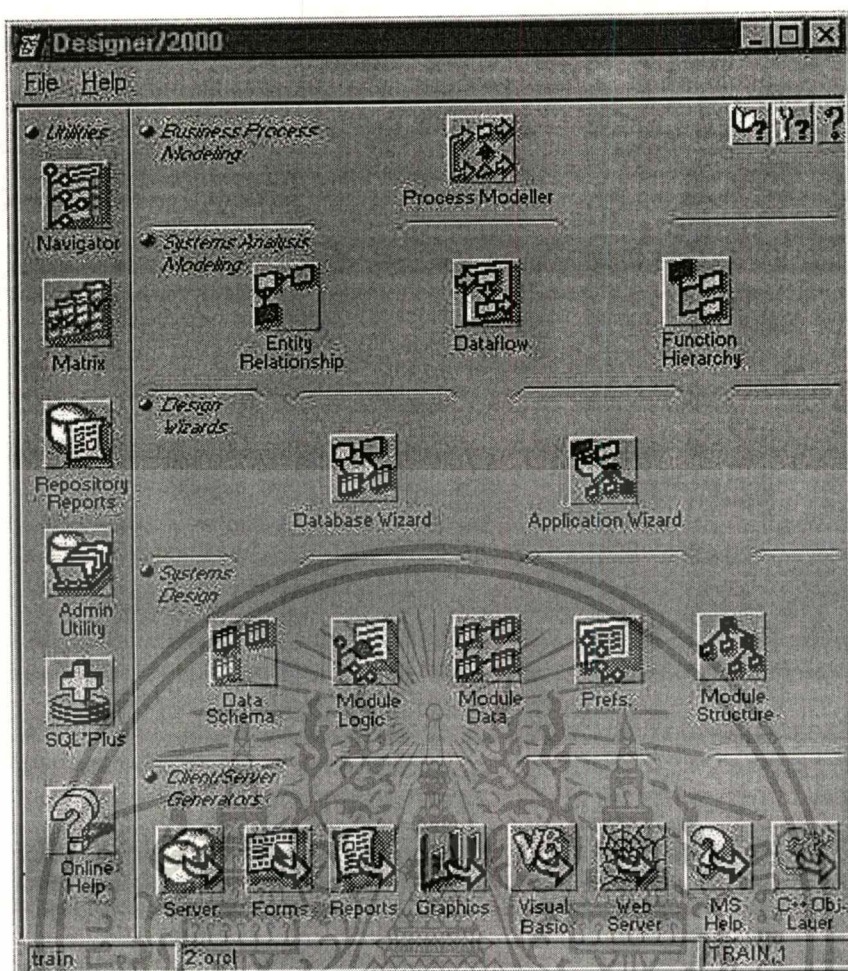
Designer/2000 ประกอบไปด้วย Module ต่าง ๆ หลายตัว แต่ละ Module จะทำหน้าที่แตกต่างกันไปแต่การทำงานจะเชื่อมต่อและประสานกันได้อย่างดี ตั้งแต่เริ่มพัฒนาระบบงานจนได้ตัวโปรแกรมออกมา ซึ่งจะแบ่งออกเป็น Module ตามลำดับ ดังนี้

- Process Modeler
- Systems Modeler
- Systems Designer
- Generator



ภาพที่ 5.1 แสดงองค์ประกอบของ Oracle Designer/2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

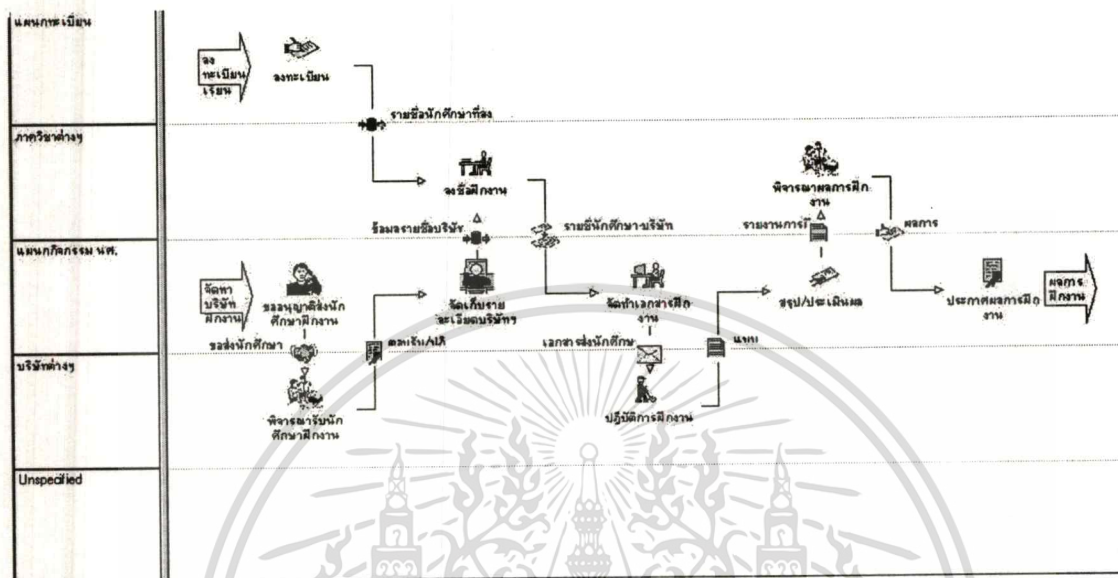


ภาพที่ 5.2 แสดงหน้าจอของ Designer/2000

5.3 Process Modeler

เป็นโปรแกรมที่จะช่วยให้ผู้ออกแบบได้บรรยายถึง Business Process ออกมาเป็นลักษณะของรูปภาพแสดงให้เห็นถึง Business Process นั้นมีความสัมพันธ์และต่อเนื่องกันอย่างไร ซึ่งผู้ออกแบบระบบสามารถนำรูปภาพ วิดีโอ เช่น ภาพวิดีโอเกี่ยวกับพนักงานกำลังตรวจดูของภายใน Stock เข้ามาอยู่ใน Process เป็นต้น เพื่อให้ผู้พัฒนาระบบและผู้ใช้ระบบได้เห็นลักษณะการทำงานจริงๆ รวมทั้งจะช่วยให้ผู้ใช้เกิดความเพลิดเพลินและให้ความร่วมมือ ในการสร้างและแก้ไขเปลี่ยนแปลงโมเดล โมเดลที่สร้างขึ้นนี้จะเป็พื้นฐานจึงต้องถูกต้อง ทั้งผู้พัฒนาและผู้ใช้จะต้องเข้าใจตรงกันเมื่อเราได้สร้าง Objects ในโมเดลนี้แล้วจะถูกนำไปใช้ในขั้นตอนต่อไปของการพัฒนาระบบงานที่สำคัญคือเราสามารถแก้ไขเปลี่ยนแปลง Objects เหล่านั้นที่ขั้นตอนใดของการพัฒนาระบบงานก็ได้ทำให้เกิดความยืดหยุ่นในการพัฒนาระบบงาน ซึ่งประโยชน์โดยตรงของการทำ Process Modeler นี้ก็คือ การแสดงความเข้าใจของผู้พัฒนาระบบงานหลังจากที่ได้เก็บรวบรวมข้อมูลและรายละเอียดต่างมาแล้วว่าเข้าถึงระบบการทำงานที่ถูกต้องหรือไม่ เพื่อที่จะไปแสดงให้กับทางผู้ใช้ออกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าเข้าใจถูกต้องหรือไม่ และยังเป็นแนวทางที่ทางผู้พัฒนาระบบงานจะใช้ Process Modeler ที่เขียนขึ้นมาใช้ในการ Reengineer องค์กรด้วย ประโยชน์ทางอ้อมก็คือสามารถที่จะใช้เป็นเอกสารอย่างหนึ่งในการเสนอขอ ISO ได้ด้วย เพราะถือว่าเป็น Process การทำงานขององค์กรได้



ภาพที่ 5.3 แสดงตัวอย่างของ Process Modeler

5.4 System Modeler

System Modeler เป็นกลุ่มของ Tools ที่ช่วยในการเขียน Diagram ต่าง ๆ ที่ช่วยในงานพัฒนาระบบ เช่น Entity Relationship Diagram, Function Hierarchy Diagram Dataflow และ Matrix Diagram เพื่อแสดงถึงความสัมพันธ์ของ System Objects ที่มีอยู่ในระบบงาน นอกจากนั้นระหว่างที่เราใช้ System Modeler นั้น เราสามารถดึง Objects ต่าง ๆ ที่สร้างไว้ด้วย Process Modeler เข้ามาใช้งานต่อใน Systems Modeler เช่น Process, Business Unit เป็นต้นในการทำงานเดียวกัน Objects ที่สร้างใน Systems Modeler นี้ ยังสามารถเชื่อมโยงนำไปใช้ต่อใน Systems Designer ทั้งในลักษณะที่เป็นการดึงมาใช้โดยตรง และดึงมาเพื่อทำการเพิ่มรายละเอียดลงไป Diagram ต่าง ๆ ที่อยู่ใน Systems Designer ประกอบไปด้วย

- Entity Relationship Diagrammer
- Function Hierarchy Diagrammer
- Dataflow Diagrammer
- Matrix Diagrammer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

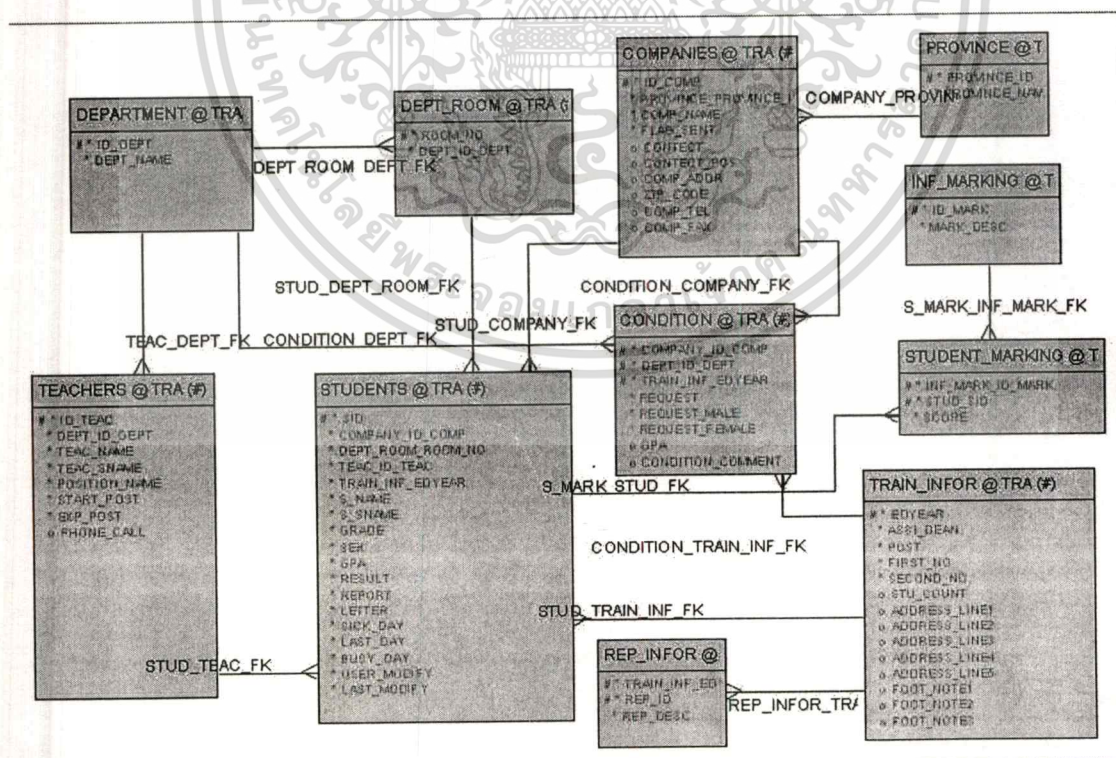
5.5 System Designer

เป็นกลุ่มของ Tools Designer/2000 ที่ช่วยให้ผู้พัฒนาระบบงานได้มองเห็นรูปภาพของ Physical Database Design หรือ Table และความสัมพันธ์ของแต่ละ Table (Relationship) รวมทั้งความสัมพันธ์ของ Programs ต่าง ๆ ที่จะสร้างด้วยตัว Designer/2000 ซึ่งแบ่งแยกย่อยออกเป็น

- Data Diagrammer
- Module Structure Diagrammer
- Module Data Diagrammer
- Module Logic Navigator
- Preference Navigator
- Design Wizards

5.5.1 Data Diagrammer

Data Diagrammer จะช่วยให้ผู้พัฒนาระบบงานมองเห็น Table และความสัมพันธของแต่ละ Table ออกมาเป็นรูปภาพ ทำให้ง่ายต่อการมองภาพรวมของ Table และ Column, Primary key, Foreign key นอกจากนี้ยังง่ายต่อการสร้างและแก้ไขเปลี่ยนแปลง



ภาพที่ 5.6 Data Diagrammer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.2 Module Structure Diagrammer

Module Structure Diagrammer จะช่วยให้ผู้พัฒนาระบบงานมองเห็น Workflow ของระบบงานเป็นรูปภาพ จากโปรแกรมหนึ่งเชื่อมต่อไปอีกโปรแกรม นั่นคือภาพรวมของระบบงานจะถูกสร้างเป็นรูปภาพทำให้ง่ายต่อการเข้าใจและสามารถแก้ไขได้ง่าย

5.5.3 Module Data Diagrammer

Module Data Diagrammer จะเป็นตัวเพิ่มเติมรายละเอียดลงไปในแต่ละ Module ว่าหน้าตาจอภาพจะประกอบด้วยข้อมูลอะไรบ้างที่จะเรียกใช้จากฐานข้อมูลมี List ที่เรียกว่า Pop List ตรง Field ไหน ทำให้ผู้พัฒนาระบบงานมองเห็นสิ่งดังกล่าวเป็นรูปภาพ ง่ายต่อการเข้าใจและแก้ไขเปลี่ยนแปลง จะเห็นว่า Module Structure Diagrammer จะเป็นเรื่องของ Option ใน Menu ของระบบงานแต่ไม่ได้กล่าวถึงว่าแต่ละ Module นั้นมีรายละเอียดเป็นอย่างไร Module Data Diagrammer จะบอกถึงรายละเอียดของแต่ละ Module จุดเด่นก็คือโปรแกรมจะสร้างภาพให้ผู้พัฒนาได้เข้าใจถึงภาพรวมได้ง่ายและเข้าไปแก้ไขได้ง่าย

5.5.4 Module Logic Navigator

Module Logic Navigator บางครั้งการสร้างโปรแกรมจำเป็นต้องมีการคำนวณค่าต่าง ๆ หรือที่เรียกว่า Business Rule ที่ต้องสร้างโปรแกรมให้มีลักษณะเป็น Procedure ขึ้นมาเพื่อรองรับ Business Rule ดังกล่าว Module Logic Navigator สามารถช่วยให้ผู้พัฒนาระบบงานได้เขียน PL/SQL Packages, Procedure Function และ Triggers ได้ง่ายขึ้นโดยสามารถ Drag and Drop คำสั่งที่มีให้เข้ามาไว้ในตัว Editor รวมทั้งยังตรวจสอบความถูกต้องของคำสั่งได้ด้วย

5.5.5 Preferences Navigator

Preferences Navigator การพัฒนาระบบงานจำเป็นต้องมีกฎเกณฑ์ที่จะใช้ร่วมกัน ทั้งระบบงานหรือบางส่วนของระบบงาน เช่น รูปร่างหน้าตาของจอภาพหรือที่เรียกว่า Layout จะใช้ลักษณะไหน จะมีการบอกไว้ใน Template สีและขนาดของตัวหนังสือหรือ Font ที่จะใช้บนหน้าจอสิ่งเหล่านี้กำหนดไว้ใน Preference ซึ่งเราสามารถกำหนดไว้ใช้ในแต่ละระบบหรือบางส่วนของระบบ

Preference Navigator จะช่วยให้การกำหนด Preference ได้ง่ายและสะดวกเพราะเป็นลักษณะ Point and Click และมี Preference ให้เลือกเป็นจำนวนมาก

5.5.6 Design Wizards

Design Wizards เมื่อเราพัฒนาระบบงานไปได้ระดับหนึ่งแล้วบางที่เราอาจอยากเห็น Table, Column Relationship ต่าง ๆ โดยอาศัยข้อมูลที่เราได้ใส่ลงไป ใน Designer/2000 Database Wizard จะสร้างสิ่งเหล่านี้ให้โดยดูข้อมูลที่ทำไว้ใน Entity Relationship Diagram เพื่อนำมาสร้างเป็น Table, Column และ Index ในส่วนของ Application จะมี Application Design Wizard จะสร้าง Menu Structure เพื่อเชื่อมโยง Module ต่าง ๆ เข้าด้วยกันเป็นระบบงานและแต่ละ Module แต่ละโปรแกรมจะไปใช้ข้อมูลส่วนไหนในฐานข้อมูล

5.6 Generator

เมื่อสิ้นสุดขั้นตอนของการออกแบบระบบฐานข้อมูลที่อยู่ทางด้าน Server และ Application ที่จะอยู่ในส่วนของ Client แล้ว ตัว Generator ใน Designer/2000 จะดึงข้อมูลที่เราได้ใส่ไว้ในระหว่างที่เรา กำลังออกแบบระบบมาสร้างเป็นโปรแกรมเพื่อใช้งาน ทั้งในฝั่ง Client และ Server โดยฝั่ง Client จะสร้าง Source Code ที่เป็น 4GL Tools ของ Oracle Forms และ Oracle Reports ส่วนทางด้าน Server จะสร้าง SQL Scripts เพื่อนำไปสร้างเป็น Table, Column, Trigger, Store Procedure เป็นต้น ซึ่งพอจะแบ่งได้ดังนี้

5.6.1 Forms Generator

ตัว Generator ตัวนี้จะดึงข้อมูลจาก Repository ออกมาสร้างเป็น Source Code ที่เป็น ภาษาของ Oracle Form Version 4.5 โดยก่อนที่จะสร้างออกมาเป็น Form จะเข้าไปดูว่า Form ที่กำลังจะสร้างใช้ Table ไหน Column อะไร มีการ Insert, Update, Delete ไหม มี Preferences อะไรที่กำหนดไว้บ้าง เป็นต้น ข้อมูลดังกล่าวจะถูกนำมาสร้างเป็นตัวโปรแกรม

5.6.2 Report Generator

เช่นเดียวกับ Forms Generator ตัวสร้าง Report นี้จะอ่านเอาข้อมูลจาก Repository มาสร้างเป็น Report โดยสามารถที่จะหาผลรวม Breaks รวมทั้งทำ Drill-Downs Report, Matrix Report

5.6.3 Server Generator

สิ่งที่ Server Generator ทำก็คือนำ Source Program ที่เป็น Procedure, Trigger, Function และ Package ที่สร้างไว้โดยใช้ Module Logic Navigator นำมา Execute สร้างอย่างสอดคล้องกับ

Oracle Version 7 ที่สามารถสร้าง Procedure, Trigger, Function และ Package ดังกล่าวข้างต้นไว้ในตัว Database ได้

5.7 Repository Administrator

Modules ต่าง ๆ ที่กล่าวมาข้างต้นประกอบกันเป็น Designer/2000 ตั้งแต่การทำ Process Model, ER-Diagram, Function Hierarchy Diagram, Dataflow Diagram เรื่อยมาจนสร้างตัวโปรแกรมให้ โดยใช้ Generator จะเห็นว่าครอบคลุมทั้ง Upper Case และ Lower Case

อย่างไรก็ตามแต่ละ Module ทั้งหมดที่กล่าวมายังไม่สามารถทำให้เรามองเห็น Objects ทั้งหมดที่มีอยู่ในระบบได้ ดังนั้น Repository Administrator จะนำ Objects ที่มีอยู่ในระบบทั้งหมดมาแสดง โดยใช้ Repository Object Navigator (RON)

5.8 สรุป

Designer/2000 เป็น CASE Tools ที่มีประสิทธิภาพตัวหนึ่ง โดยเฉพาะถ้าใช้กับ Oracle Database ช่วยให้การพัฒนาระบบงานง่ายขึ้นสะดวกสำหรับผู้พัฒนาระบบงานและผู้ใช้ (End User) องค์กรได้รับประโยชน์ทั้งในแง่ความรวดเร็วในการพัฒนาระบบและเอกสาร (Document) ที่ครบถ้วนรวมทั้งความเป็นมาตรฐานของระบบงาน และความเป็นมาตรฐานของเอกสารที่ได้ Designer/2000 เองสามารถที่จะทำการ Generate เป็น Form, Report, Utility ได้แน่นอน และสามารถที่จะนำไปใช้งานจริง แต่ถ้าจะให้ระบบงานที่ได้มีประสิทธิภาพมากขึ้น ก็สามารถที่จะนำส่วนที่ Generate แล้วไปทำการพัฒนาต่อโดยใช้ Developer ซึ่งเป็นเครื่องมือตัวหนึ่งของ Oracle ที่มีประสิทธิภาพ แต่ผู้ใช้จะต้องสามารถเขียนและเข้าใจในภาษา PL/SQL เป็นอย่างดี

จากการที่ได้ใช้ CASE Tools ของทาง Oracle มีความเห็นว่ามีประโยชน์มากยิ่งในเรื่องของการออกแบบระบบฐานข้อมูลที่มีจำนวน TABLE มากๆ การเขียน ER-Diagram จะช่วยเราได้มาก ไม่ต้องมานั่งจดจำความสัมพันธ์ทั้งหมด แถมยังที่จะสามารถแก้ไขและสร้างเป็น TABLE ได้อย่างรวดเร็วแถมในตัวของ Designer เองยังที่จะสามารถทำการสร้างสคริป เพื่อที่จะนำเอาสคริปที่สร้างไปสร้าง TABLE ในดาต้าเบสตัวอื่นๆ ได้อีกด้วย โดยไม่มีข้อผิดพลาดอะไรเลย

บทที่ 6

วิเคราะห์และออกแบบระบบงานฝึกงาน

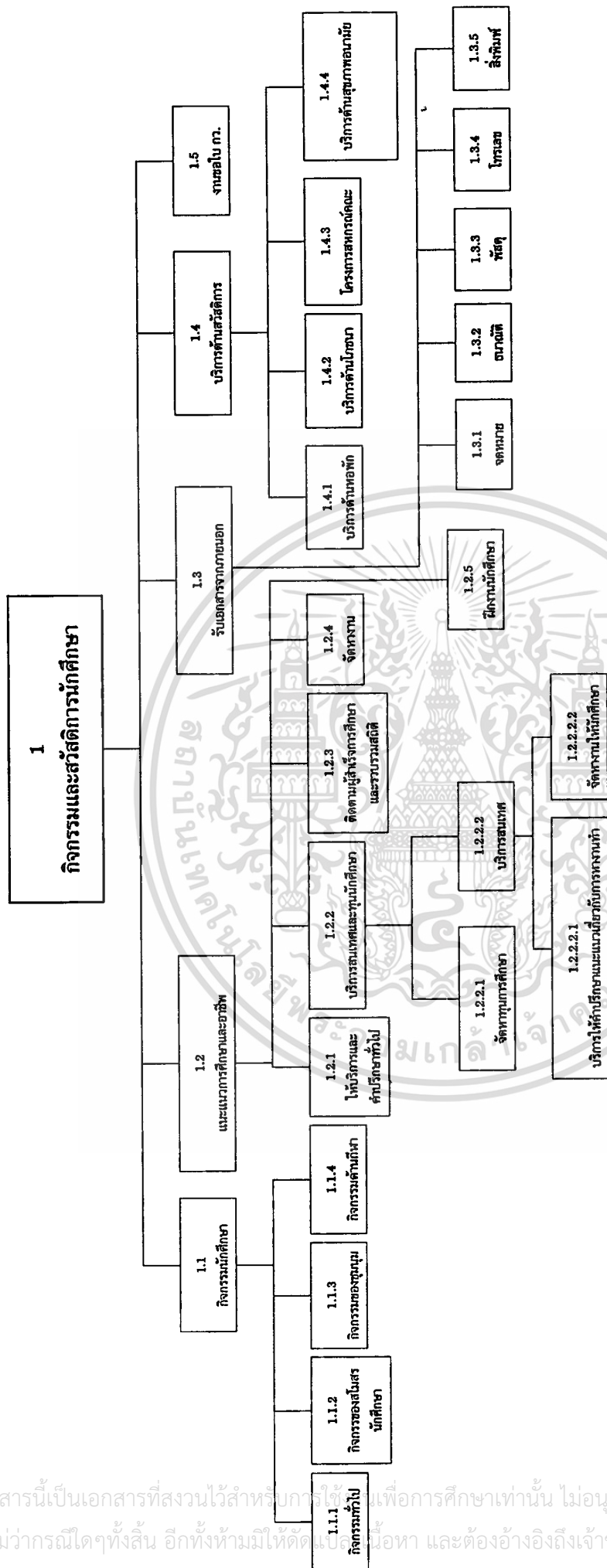
6.1 การศึกษาระบบเก่า

ลักษณะงานของระบบการฝึกงานตั้งแต่เดิมนั้นจะเป็นการที่ผู้ใช้ต้องทำงานด้วยตนเอง (Manual) โดยมีได้นำระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการกับข้อมูลต่างๆ ที่มีอยู่เป็นจำนวนมาก และมีได้นำระบบคอมพิวเตอร์มาอำนวยความสะดวกต่างๆ เว้นแต่ใช้ระบบคอมพิวเตอร์ในการทำเอกสารที่มีรูปแบบและรูปลักษณะที่สวยงามเท่านั้น แต่การเตรียมข้อมูลสำหรับเอกสารต่างๆ ก็ยังคงเป็นระบบที่ผู้ใช้ต้องเป็นคนจัดการอยู่ทั้งสิ้น เราสามารถสรุปออกเป็นข้อๆ ได้ดังนี้

- เป็นระบบที่ไม่มีการนำคอมพิวเตอร์มาช่วยในการประมวลผลข้อมูลซึ่งการทำงานส่วนมากจะมีลักษณะที่ผู้ใช้ ทำงานด้วยตนเองซึ่งอาจจะมีการใช้คอมพิวเตอร์อยู่บ้างก็เพียงเพื่อการทำเอกสารหรือรายงานโดยที่ข้อมูลสำหรับเอกสารหรือรายงานได้มีการจัดเตรียมโดยผู้ใช้ ก่อนแล้วจึงใช้โปรแกรมหรือแอปพลิเคชันต่างๆ เช่น Microsoft Word หรือ Microsoft Excel เป็นต้น ในการสร้างเอกสารหรือรายงาน ไม่ว่าจะเป็นการทำจดหมาย, ตารางและกราฟ
- การจัดเก็บข้อมูลของระบบงานเดิมนั้นอยู่ในรูปของเอกสารต่างๆ ซึ่งจัดเก็บอยู่ในตู้เอกสารดังนั้นการค้นหาหรือการนำเอาข้อมูลเหล่านั้นมาใช้งานในแต่ละครั้งจำเป็นที่จะต้องมีการค้นหาจากเอกสารที่ถูกเก็บอยู่ในตู้เอกสาร ดังนั้นในการเก็บเอกสารจึงจำเป็นที่จะต้องมีการจัดเก็บอย่างมีระเบียบและเป็นหมวดหมู่ เพื่อสะดวกต่อการนำข้อมูลเหล่านั้นมาใช้งาน และในบางกรณีที่เป็นจะต้องทำการปรับปรุงข้อมูลหรือแก้ไขข้อมูลเดิมที่มีอยู่ให้ถูกต้องตามความเป็นจริงในปัจจุบัน ซึ่งจะทำให้ยากลำบากมากในการดูแลข้อมูลต่างๆ นั้นให้ครบถ้วนและถูกต้องเสมอสำหรับการใช้งาน
- งานต่างๆ นั้นจะมีผู้รับผิดชอบอยู่เพียงผู้เดียว เป็นผู้รู้ว่ามีข้อมูลต่างๆ มีอะไรอยู่บ้าง และถูกเก็บอยู่ที่ใดและทราบถึงขั้นตอนการทำงานทั้งหมดในระบบเป็นอย่างดี นั้นหมายความว่าการทำงานทั้งหมดของระบบฝึกงานเดิมนั้นจะตกอยู่ที่บุคคลเพียงคนเดียว ถ้าบุคคลที่รับผิดชอบไม่สามารถปฏิบัติหน้าที่ได้ การทำงานของระบบก็จะต้องหยุดชะงักเพราะไม่มีผู้ใดที่จะสามารถทำงานแทนได้เพราะไม่ทราบถึงข้อมูลที่ถูกจัดเก็บและขั้นตอนการทำงานที่แท้จริง และนอกจากนี้เมื่อจำเป็นที่จะต้องเปลี่ยนบุคคลที่รับผิดชอบอยู่ระบบของการทำงานอาจจะต้องถูกเปลี่ยนแปลงด้วย
- มีความซ้ำซ้อนของในการทำงานบางอย่าง เช่น การออกจดหมายที่มีรูปแบบเหมือนกันแต่อาจจะแตกต่างกันเพียงข้อมูลบางอย่าง ก็จะทำให้ผู้ใช้นั้นต้องทำงานส่วนใหญ่ ซึ่งเป็นการทำงานที่ซ้ำซ้อนกันเป็นการเสียเวลาไปโดยเปล่าประโยชน์

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.1 แสดง Function Hierarchy ของแผนกกิจกรรมนักศึกษา

6.2 ปัญหาต่างๆ ในระบบเก่า

การทำงานในระบบเดิมที่ทำโดยที่ผู้ใช้จะต้องทำทุกอย่างเองหรือระบบที่ยังมิได้นำระบบคอมพิวเตอร์เข้ามาช่วยงาน มักจะเกิดปัญหาต่างๆ และพอที่จะสรุปปัญหาเหล่านั้นได้ออกมาเป็นข้อๆ ได้ดังนี้

- ปัญหาในการจัดเก็บข้อมูล เนื่องจากระบบงานฝึกงานของนักศึกษานั้นต้องอาศัยข้อมูลเป็นจำนวนมากในการทำงาน ดังนั้นจึงจำเป็นต้องมีระบบการจัดเก็บข้อมูลที่เป็นระบบระเบียบเพื่อความสะดวกในการนำข้อมูลเหล่านั้นมาใช้งาน
- ปัญหาในการพิมพ์เอกสาร ในการพิมพ์เอกสารหรือจดหมายที่ใช้ในงานด้านนี้นั้นจะมีการพิมพ์เอกสารหรือจดหมายที่มีลักษณะเหมือนกันเป็นจำนวนมาก แต่อาจจะแตกต่างกันเพียงชื่อหรือที่อยู่ของผู้รับจดหมาย หรือเอกสารเท่านั้น ดังนั้นจึงเกิดปัญหาขึ้น คือ ผู้พิมพ์จดหมายจะต้องทำงานที่มีลักษณะงานที่ซ้ำซากและเสียเวลาในการพิมพ์มากแม้ว่าผู้พิมพ์จะมีความชำนาญในการพิมพ์มากแค่ไหนก็ตาม
- ปัญหาในการค้นหาข้อมูล เนื่องจากระบบงานการฝึกงานของนักศึกษานั้นต้องอาศัยข้อมูลเป็นจำนวนมากในการทำงาน ดังนั้นในการค้นหาข้อมูลสักอย่างหนึ่งเกี่ยวกับนักศึกษา หน่วยงานหรือบริษัทต่างๆ ไม่ใช่เรื่องง่ายเลย ที่จะทำการหาด้วยมือจากเอกสารที่มีอยู่ ซึ่งข้อมูลอาจจะไม่อยู่ในเอกสารชุดเดียวกันกันยิ่งทำให้การค้นหาข้อมูลเหล่านั้นล่าช้ายิ่งขึ้น
- ปัญหาในการสรุปผลข้อมูล เนื่องจากการสรุปข้อมูลต่างๆ นั้นจำเป็นอย่างยิ่งที่จะต้องอาศัยข้อมูลจากหลายที่(เอกสารหลายชุด) และยังอาจจะต้องมีการคำนวณทางสถิติอีก ซึ่งมีโอกาสผิดพลาดได้มากเมื่อทำการคำนวณด้วยมือ และในบางครั้งอาจจะต้องมีการสรุปข้อมูลต่างๆ เหล่านั้นออกมาเป็นกราฟ เพื่อความสะดวกในการพิจารณาผลสรุปในแต่ละปี นอกจากนี้ยังมีการนำข้อมูลจากปีก่อนๆ มาใช้ในการสรุปผลอีกเพื่อที่จะดูแนวโน้มความต้องการรับนักศึกษาฝึกงานในสาขาวิชาต่างๆ ของตลาดแรงงาน

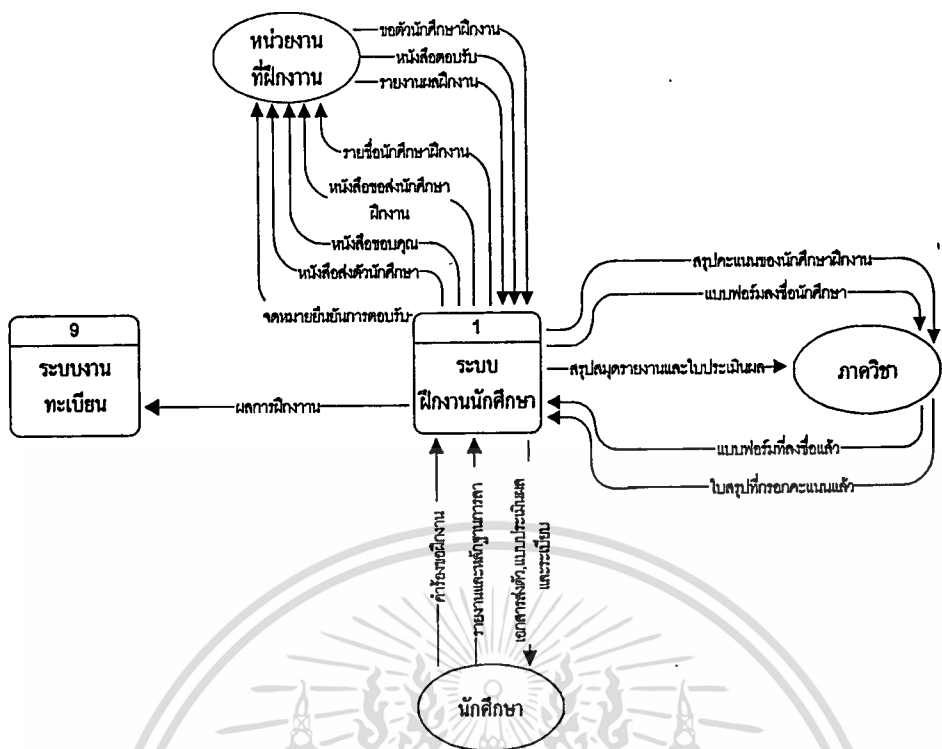
6.3 ขั้นตอนการทำงาน

ขั้นตอนในการทำงานของระบบงานการฝึกงานของนักศึกษา พอที่จะแยกออกเป็นส่วนๆ ได้ดังนี้

6.3.1 ขั้นตอนในการเตรียมข้อมูล

สำหรับระบบงานฝึกงานของนักศึกษาระบบเดิมนั้นข้อมูลต่างๆ จะถูกเตรียมขึ้นสำหรับการฝึกงานของนักศึกษาแต่ละปีซึ่งจะจำแนกเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.2 แสดงภาพดาต้าโฟลว์ระบบเก่า

6.3.1.1 การเตรียมข้อมูลนักศึกษา

การเตรียมข้อมูลนักศึกษาที่มีสิทธิ์ฝึกงานประจำปีการศึกษา ในระบบเดิมจะนำข้อมูลรายชื่อนักศึกษามาจากใบรายชื่อนักศึกษาที่ทางแผนกทะเบียนนักศึกษาของคณะวิศวกรรมศาสตร์เป็นผู้จัดทำขึ้นมา ส่วนรายละเอียดต่างๆ นั้นจะถูกจัดหามาเมื่อจำเป็นที่จะต้องใช้ ตัวอย่างเช่น เกรดเฉลี่ยของนักศึกษา ถ้ามีหน่วยงานระบุเกรดเฉลี่ยต่ำสุดที่จะรับนักศึกษาเข้าฝึกงานจำเป็นที่จะต้องตรวจสอบเกรดเฉลี่ยของนักศึกษาด้วยโดยอาจจะสอบถามจากนักศึกษาโดยตรงหรือไปค้นที่แผนกทะเบียนนักศึกษา ซึ่งวิธีการแรกอาจจะดูไม่ค่อยน่าเชื่อถือและไม่มีหลักเกณฑ์มากนัก แต่นิยมที่จะปฏิบัติเพราะความสะดวก

6.3.1.2 การเตรียมข้อมูลหน่วยงาน

การเตรียมข้อมูลของหน่วยงานที่จะสามารถส่งนักศึกษาเข้าฝึกงานทั้งหมดสำหรับการจัดส่งเอกสารขออนุญาตในการจัดส่งนักศึกษาเข้าฝึกงาน ในทางปฏิบัติแต่เดิมนั้นรายละเอียดของหน่วยงานต่างๆ จะถูกจัดเก็บไว้เป็นเอกสารดังนั้นในการค้นหาข้อมูลต่างๆ จะต้องทำการค้นหาจากตู้เอกสารซึ่งไม่ถายนักที่จะค้นหาข้อมูลดังกล่าว และข้อมูลอีกส่วนหนึ่งอาจจะมาจากการที่นักศึกษาแจ้งความนงที่จะหาหน่วยงานฝึกงานด้วยตนเอง ดังนั้นทางแผนกกิจกรรมนักศึกษาที่จะเก็บข้อมูลรายละเอียดของหน่วยงานต่างๆ นี้ไว้ด้วย นอกจากนี้ข้อมูลที่มีอยู่จำเป็นที่จะต้องถูกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไขให้ถูกต้องตามสภาพที่เป็นอยู่ปัจจุบันอีกด้วย ตัวอย่างเช่น การแก้ไขที่อยู่ของหน่วยงานเมื่อหน่วยงานนั้นมีการย้ายสำนักงานหรือย้ายที่ทำการไปที่แห่งใหม่ เป็นต้น

6.3.1.3 การเตรียมข้อมูลคณะกรรมการอุตสาหกรรมสัมพันธ์

การเตรียมข้อมูลของคณะกรรมการอุตสาหกรรมสัมพันธ์ เนื่องจากคณะกรรมการอุตสาหกรรมสัมพันธ์มีการเปลี่ยนแปลงทุกปีการศึกษา เนื่องจากการหมดวาระของการเป็นคณะกรรมการอุตสาหกรรมสัมพันธ์และการแต่งตั้งคณะกรรมการชุดใหม่ขึ้นมาทุก ๆ ปีการศึกษาจึงจำเป็นที่จะต้องทำการแก้ไขข้อมูลของคณะกรรมการอุตสาหกรรมสัมพันธ์ให้ถูกต้องเพื่อใช้ในการออกเอกสารต่าง ๆ

6.3.2 ขั้นตอนในการออกเอกสาร

ขั้นตอนในการออกเอกสารขออนุญาตส่งนักศึกษาฝึกงานประจำปีการศึกษา ในทางปฏิบัติจะต้องร่างรูปแบบเอกสารโดยการนำข้อมูลต่าง ๆ ที่มีอยู่มาประกอบกัน และในการ จัดพิมพ์นั้น เมื่อมีการออกเอกสารสำหรับหน่วยงานหลาย ๆ หน่วยงาน ผู้ปฏิบัติจะต้องทำการ แก้ไขรายละเอียดต่าง ๆ ของหน่วยงานด้วยตนเอง

6.3.3 รับและจัดเก็บเงื่อนไขการฝึกงานของหน่วยงาน

การรับและจัดเก็บเงื่อนไขการฝึกงานของหน่วยงาน เมื่อได้จัดทำเอกสารขออนุญาตส่งนักศึกษาฝึกงานไปแล้วทางหน่วยงานเหล่านั้นจะจัดส่งหนังสือตอบรับหรือปฏิเสธการรับนักศึกษาเข้าฝึกงานกลับมาทางแผนกกิจกรรมนักศึกษาคณะฯ ซึ่งรายละเอียดของหนังสือ ตอบรับนั้นจะเป็นข้อมูลของเงื่อนไขการรับนักศึกษาฝึกงานที่ทางหน่วยงานต้องการ

6.3.4 จัดทำแบบฟอร์มการลงชื่อฝึกงานให้กับภาควิชา

การจัดทำแบบฟอร์มการลงชื่อฝึกงานให้กับภาควิชาต่าง ๆ จะถูกจัดทำขึ้นหลังจากที่ได้รับเอกสารตอบรับมาจากหน่วยงานต่าง ๆ แล้ว โดยในการจัดทำแบบฟอร์มจะต้องมีการแยกแยะความต้องการรับนักศึกษาภาคต่าง ๆ ออกมาเพื่อที่จะทำแบบฟอร์มการลงชื่อฝึกงานเพื่อส่งให้กับภาควิชา และทางภาควิชาจะจัดให้นักศึกษามีการลงชื่อฝึกงานของนักศึกษาฝึกงานอีกต่อหนึ่ง

6.3.5 จัดทำเอกสารยืนยันการส่งนักศึกษาฝึกงานกับหน่วยงาน

การจัดทำเอกสารยืนยันการส่งนักศึกษาฝึกงานกับหน่วยงาน จะถูกจัดทำขึ้นเมื่อทราบว่านักศึกษาคนใดที่ลงชื่อฝึกงานกับหน่วยงานใดแล้ว ก็จะทราบว่าจำนวนนักศึกษาที่ลงชื่อ ฝึกงาน ครบตามจำนวนนักศึกษาที่ลงชื่อฝึกงานครบตามจำนวนนักศึกษาที่หน่วยงานต้องการนั้นหรือไม่ ถ้าครบก็จะจัดทำเอกสารยืนยันการจัดส่งนักศึกษาฝึกงานเพื่อยืนยันการจัดส่งนักศึกษาฝึกงานกับหน่วยงาน แต่ถ้าไม่ครบก็จะต้องทำการแจ้งถึงจำนวนที่หน่วยงานต้องการและจำนวนนักศึกษาที่จะจัดส่งไปฝึกงานจริง และต้องอธิบายถึงสาเหตุของการส่งนักศึกษาฝึกงานไม่ครบตามจำนวนด้วย เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.6 จัดทำเอกสารที่ใช้ในการรายงานตัวเข้าฝึกงานของนักศึกษา

การจัดทำเอกสารที่ใช้ในการรายงานตัวเข้าฝึกงานของนักศึกษา เป็นเอกสารที่ใช้ในการส่งมอบนักศึกษาฝึกงานให้กับหน่วยงานซึ่งประกอบไปด้วยรายละเอียดต่าง ๆ เกี่ยวกับตัวนักศึกษา 1 ฉบับ ได้แก่เอกสารส่งตัวนักศึกษา และอีกหนึ่งฉบับเป็นรายละเอียดการฝึกงานของนักศึกษา ประจำปีการศึกษา

6.3.7 รับและจัดทำใบสรุปคะแนนการฝึกงาน

การจัดทำใบสรุปคะแนนการฝึกงาน จะถูกจัดทำเมื่อการฝึกงานของนักศึกษาสิ้นสุดลง ทางหน่วยงานที่รับนักศึกษาฝึกงานจะทำการจัดส่งใบประเมินผลการฝึกงานของนักศึกษามายังแผนกกิจกรรมนักศึกษา เมื่อได้รับใบประเมินจากทางหน่วยฝึกงาน ก็จะนำมาจำแนกออกตามภาควิชาของนักศึกษาฝึกงานและทำการสรุปผลการปฏิบัติหน้าที่ต่าง ๆ ลงในตารางสรุปคะแนนการฝึกงาน จากนั้นจะส่งใบสรุปคะแนนนี้ให้อาจารย์ที่ปรึกษาฝึกงานประจำภาควิชาพิจารณาเกรดการฝึกงานของนักศึกษา

6.3.8 สรุปและประเมินผลการฝึกงานของนักศึกษา

เมื่อได้ข้อมูลการฝึกงานของนักศึกษาประจำปีการศึกษานั้น ๆ มาแล้วจะนำข้อมูลเหล่านั้นมาประเมินการฝึกงานประจำปีการศึกษา เพื่อที่จะพิจารณาแนวโน้มความเชื่อถือของหน่วยงานต่าง ๆ ที่มีต่อคณะและเพื่อทำการปรับปรุงแนวทางการเรียนการสอนให้เหมาะสมมากยิ่งขึ้น

6.3.9 จัดทำใบประกาศผลการฝึกงาน

ในระบบเดิมนั้นจะประกาศผลการฝึกงานของนักศึกษาโดยการนำเอาใบรายชื่อนักศึกษาที่จัดทำโดยแผนกทะเบียนมากรอกเกรดของการฝึกงานของนักศึกษาเพื่อติดประกาศ

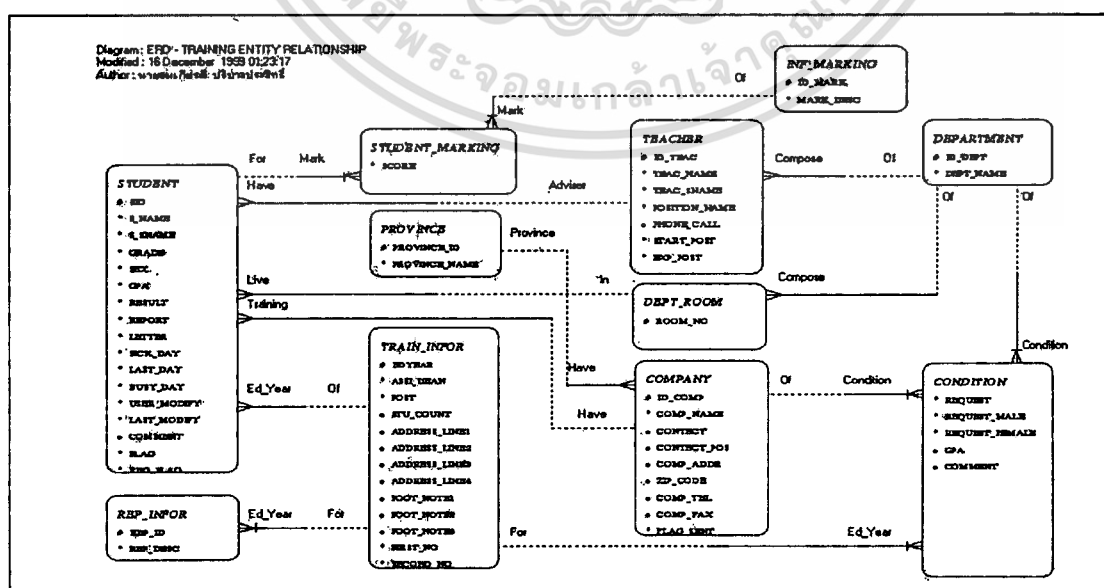
6.4 การศึกษาระบบใหม่

ลักษณะงานของระบบการฝึกงานที่พัฒนาขึ้นมาใหม่จะเป็นระบบที่ผู้ใช้ต้องทำงานกับระบบฐานข้อมูลซึ่งมีรูปแบบของการทำงานเป็นแบบไคลส์เอ็นเซอร์ฟเวอร์ โดยได้นำระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการกับระบบฐานข้อมูลที่มีอยู่ และได้นำระบบคอมพิวเตอร์มาอำนวยความสะดวกต่าง ๆ ตั้งแต่ใช้ระบบคอมพิวเตอร์ในการจัดเก็บข้อมูลและสร้างเอกสารที่มีรูปลักษณะที่สวยงามและมีระเบียบแบบแผนมากยิ่งขึ้น โดยการเตรียมข้อมูลให้กับเอกสารนั้นเป็นหน้าที่ของโปรแกรมและระบบฐานข้อมูลที่ได้พัฒนาขึ้นมา สามารถที่จะสรุปออกมาเป็นข้อ ๆ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

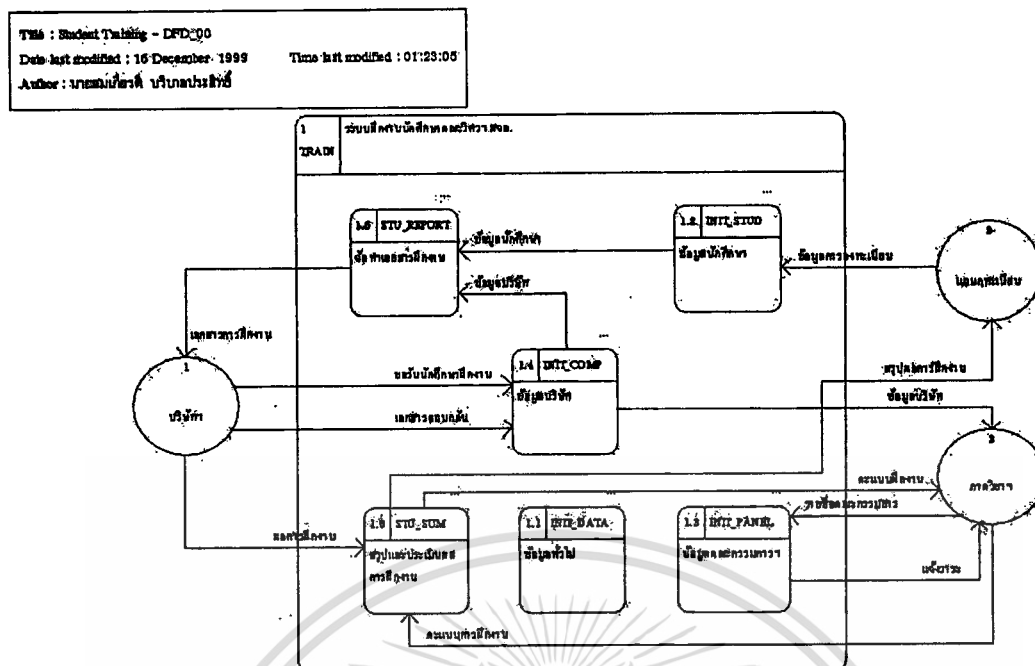
แอปพลิเคชันที่มีอยู่แค่นั้นก็เพียงพอแล้วนั้นหมายถึงการทำงานทั้งหมดของระบบฝึกงานใหม่นั้นจะสามารถกระจายงานให้กับบุคลากรได้หลายๆ คน ถ้าบุคคลคนนั้นไม่สามารถที่จะทำงานได้ การทำงานของระบบก็สามารถที่จะดำเนินงานต่อไปได้เพราะมีผู้ที่สามารถทำงานแทนได้ถ้าสามารถใช้แอปพลิเคชันที่มีอยู่ได้โดยไม่ต้องทราบถึงข้อมูลที่ถูกจัดเก็บและขั้นตอนการทำงานที่แท้จริง และนอกจากนี้เมื่อจำเป็นที่จะต้องเปลี่ยนบุคลากรที่รับผิดชอบระบบงานอยู่ ไม่จำเป็นที่จะต้องเปลี่ยนแปลงการทำงานของระบบงานที่มีอยู่เดิม

- ความซ้ำซ้อนในการทำงานบางอย่างไม่ขึ้นอยู่กับการทำงานของผู้ใช้ แต่จะขึ้นอยู่กับการทำงานของระบบคอมพิวเตอร์แทนซึ่งทำให้ประสิทธิภาพของการทำงานสูงขึ้น เช่น การออกจดหมายที่มีรูปแบบเหมือนกันให้กับหลายๆ หน่วยงานการทำงานจะเป็นเพียงแค่ผู้ใช้เข้าไปยังเมนูการทำงานในส่วนของการพิมพ์จดหมายและเลือกเรื่องของการจดหมายที่ต้องการจะพิมพ์ เลือกหน่วยงานต่างๆ ที่เป็นผู้รับผิดชอบและทำการยืนยันการพิมพ์ โปรแกรมจะเป็นผู้จัดการเตรียมข้อมูลสำหรับจดหมายดังกล่าวโดยการค้นหาข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลของเราจนครบถ้วนและนำมาผนวกกับรูปแบบของจดหมายที่มีอยู่จากนั้นก็พิมพ์เอกสารออกทางเครื่องพิมพ์ ทำเช่นนั้นจนครบตามจำนวนหน่วยงานที่ผู้ใช้เลือกขึ้นมาตั้งแต่แรก เป็นต้น
- ระบบที่พัฒนาขึ้นมานั้นเป็นการทำงานบนวินโดวส์ซึ่งยึดหลักการทำงานแบบ มูฟ แอนด์คลิก (Move and Click) ทำให้สามารถใช้งานได้ง่ายมากดังนั้นการเรียนรู้ในการใช้งานจึงไม่จำเป็นต้องอาศัยเวลามากนัก และการทำงานส่วนใหญ่จะอยู่ในรูปของหน้าจอกรอกข้อมูลซึ่งมีบริเวณที่ให้กรอกข้อมูลและมีการตรวจสอบก่อนนำไปเก็บลงฐานข้อมูลเท่าที่สามารถตรวจสอบได้เพื่อความถูกต้องของข้อมูลมีมากขึ้น



ภาพที่ 6.4 แสดงภาพ ER-Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.5 แสดงภาพ Dataflow Level 0

6.5. การแก้ไขปัญหาดังกล่าว ในระบบเดิม

- ปัญหาในการเก็บข้อมูล เนื่องจากกระบวนการฝึกงานของนักศึกษานั้นต้องอาศัยข้อมูลเป็นจำนวนมากในการทำงาน ดังนั้นจึงจำเป็นต้องมีระบบการเก็บข้อมูลที่เป็นระบบอย่างมีระเบียบเพื่อความสะดวกในการนำข้อมูลมาใช้งานต่อไป การนำระบบคอมพิวเตอร์ และระบบฐานข้อมูลมาใช้งานจะช่วยให้การบริหารฐานข้อมูลของระบบเป็นอย่างถูกต้องและง่ายดายยิ่งขึ้น เริ่มตั้งแต่การจัดเก็บข้อมูล การค้นหาและแก้ไขข้อมูล ตลอดจนการนำข้อมูลในฐานข้อมูลมาใช้งานโดยภาระงานส่วนใหญ่ในการทำงานจะตกอยู่หรือถูกโยนไปให้ระบบคอมพิวเตอร์และระบบฐานข้อมูลแทนที่จะเป็นการทำงานโดยบุคคลผู้ทำงานในระบบฝึกงาน
- ปัญหาในการออกเอกสารหรือพิมพ์เอกสาร ในการพิมพ์เอกสารหรือจดหมายที่ใช้ในงานด้านนี้นั้นจะมีการพิมพ์เอกสาร หรือจดหมายที่มีลักษณะที่เหมือนกันเป็นจำนวนมาก แต่อาจจะต่างกันเพียงชื่อและที่อยู่ของผู้รับจดหมายหรือเอกสารเท่านั้น ดังนั้นจึงเกิดปัญหาขึ้น คือ ผู้พิมพ์จดหมายจะต้องทำงานที่ซ้ำซากและเสียเวลาในการพิมพ์มากแม้ผู้พิมพ์จะมีความชำนาญในการพิมพ์มากแค่ไหนก็ตาม ระบบที่พัฒนาขึ้นมาใหม่จะช่วยอำนวยความสะดวกในเรื่องของการพิมพ์เอกสารโดยเพียงแค่ผู้ใช้เข้าไปยังรายการการทำงานในส่วนของการพิมพ์เอกสารและเลือกประเภทของเอกสารที่ต้องการจะพิมพ์ เลือกส่วนประกอบต่างๆ ที่เกี่ยวข้องกับเอกสารและทำการยืนยันการพิมพ์ โปรแกรมก็จะจัดการเตรียมข้อมูลสำหรับเอกสารดังกล่าวโดยการค้นหา

และดึงข้อมูลที่ถูกจัดเก็บไว้ในระบบฐานข้อมูลมาจนครบถ้วนและนำมาผนวกกับรูปแบบของเอกสารที่มีอยู่จากนั้นก็พิมพ์เอกสารออกทางเครื่องพิมพ์

ภาพที่ 6.6 แสดงหน้าจอที่สร้างได้จาก Designer/2000

- ปัญหาในการค้นหาข้อมูล เนื่องจากระบบงานการฝึกงานของนักศึกษานั้นต้องอาศัยข้อมูลเป็นจำนวนมากในการทำงาน ดังนั้นในการค้นหาข้อมูลสักอย่างหนึ่งเกี่ยวกับนักศึกษานั้นไม่ใช่เรื่องง่ายเลยที่จะทำการค้นหาด้วยมือจากเอกสารที่มีอยู่ ซึ่งข้อมูลอาจจะไม่อยู่ในเอกสารชุดเดียวกันก็เป็นได้ ทำให้การค้นหายิ่งล่าช้ายิ่งขึ้น ระบบงานที่

ภาพที่ 6.7 แสดงหน้าจอการใส่ข้อมูลของบริษัท

เอกสารเป็นเอกสารที่ส่งงานวิชาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาขึ้นมาใหม่ จะช่วยในการค้นหาและแก้ไขข้อมูลได้อย่างมีประสิทธิภาพและรวดเร็ว เพียงผู้ใช้เลือกรายการในการค้นหาข้อมูลและกำหนดว่าจะหาข้อมูลของอะไร ตัวอย่างเช่น ต้องการค้นหานักศึกษาคนนี้ฝึกงานอยู่ที่หน่วยงานใด เพียงแต่ผู้ใช้ใส่รหัสนักศึกษาหรือชื่อและนามสกุล เท่านั้น โปรแกรมก็จะทำการค้นหาข้อมูลทั้งหมดที่เกี่ยวข้องกับนักศึกษาคนนี้และนำออกมาแสดงที่หน้าจอ และถ้าผู้ใช้ต้องการที่จะทำการแก้ไขข้อมูลก็สามารถที่จะทำได้โดยการเลือกโหมดการทำงานให้อยู่ในการทำงานแบบการแก้ไขข้อมูลเมื่อทำการแก้ไขข้อมูลเสร็จแล้วก็ต้องยืนยันการจัดเก็บข้อมูลถ้าตกลงโปรแกรมก็จะทำการแก้ไขที่มีอยู่ให้ถูกต้องตามข้อมูลที่แก้ไขไปใหม่

- ปัญหาในการทำรายงานสรุปข้อมูล เนื่องจากการสรุปข้อมูลต่าง ๆ นั้นจำเป็นที่จะต้องอาศัยข้อมูลจากหลายที่ (เอกสารหลายชุด) และยังอาจจะมีการคำนวณทางสถิติซึ่งโอกาสผิดพลาดได้มากเมื่อทำการคำนวณด้วยมือ และในบางครั้งอาจสรุปข้อมูลต่าง ๆ เหล่านั้นออกมาเป็นกราฟ เพื่อความสะดวกในการพิจารณาผลสรุปในแต่ละปี นอกจากนี้ ยังมีการนำข้อมูลจากปีก่อน ๆ มาใช้ในการสรุปผลอีกเพื่อที่จะดูแนวโน้มความต้องการรับนักศึกษาฝึกงานในสาขาวิชาต่าง ๆ ของตลาดแรงงาน ระบบที่พัฒนาขึ้นมานั้นสามารถที่จะสรุปผลการฝึกงานออกมาเป็นตารางและกราฟต่าง ๆ ได้ โดยที่ผู้ใช้ไม่ต้องทำการค้นหาข้อมูลต่าง ๆ และนำข้อมูลเหล่านั้นมาคำนวณและรวมไปถึงการวาดกราฟด้วยตนเอง ระบบจะเป็นผู้ทำให้ทั้งหมดเพียงแต่ผู้ใช้เลือกรายการและเลือกรูปแบบของการสรุปผลที่ต้องการตามที่ระบบกำหนดไว้เท่านั้น ระบบก็จะทำให้ทุกอย่าง

ระบบงานฝึกงานนักศึกษา คณะวิศวกรรมศาสตร์ สงข.

17-FEB-00 โดย สมเกียรติ บริบาลประสิทธิ์

ข้อมูลบริษัท

รหัสบริษัท: 22

ชื่อบริษัท: บริษัท ออราเคิล (ไทยแลนด์) จำกัด

ชื่อบุคคลที่ติดต่อ: คุณมานะภักดิ์

หมายเลขโทรศัพท์: 3289068

เงื่อนไข

ปีการศึกษา: 2543

ชื่อรองคณบดีฝ่ายกิจกรรม: ศาสตราจารย์

รหัสภาควิชา: 01

ชื่อภาควิชา: วิศวกรรมศาสตร์

จำนวนที่ต้องการ: 10

จำนวนนักศึกษาราย: 5

จำนวนนักศึกษาหญิง: 5

เกรดที่ต้องการ: 2.50

หมายเหตุ:

ภาพที่ 6.8 แสดงหน้าจอการใส่เงื่อนไขการรับนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการศึกษาและวิจัย

ในทฤษฎีของ CASE*METHOD นั้น แสดงถึงหลักการทางธุรกิจที่นำมาประยุกต์เพื่อใช้ในการพัฒนาระบบคอมพิวเตอร์ เมื่อพัฒนาแล้วผู้ที่ไม่รู้เรื่องเกี่ยวกับระบบงานหรือขอบวงการทางคอมพิวเตอร์สามารถที่จะทำความเข้าใจกับระบบงานนั้นๆ ได้โดยการอ่านทำความเข้าใจกับเอกสารต่างๆ ที่จัดทำขึ้นมาจากทฤษฎีของ CASE*METHOD ซึ่งพอที่จะสรุปสาระสำคัญของเอกสารที่จะทำได้แก่

- การเขียน Function Model

การเขียน Function Model นั้นก็คือการ นำข้อมูลขององค์กรและรายละเอียดต่างๆ ที่ทำการเก็บรวบรวมมาทำการวิเคราะห์และทำความเข้าใจถึงเป้าหมายขององค์กรนั้นๆ ว่าองค์กรที่เรา กำลังศึกษาอยู่นั้นมีเป้าหมายหรือวัตถุประสงค์อะไรบ้าง แล้วก็มีกิจกรรมอะไรบางอย่างที่จะต้องทำหรือ ดำเนินการ เพื่อที่จะทำให้บรรลุวัตถุประสงค์ จะเรียกกิจกรรมต่างๆ เหล่านั้นว่า Business Function อาจจะให้ความหมายหรือคำจำกัดความสั้นๆ ของ Business Function ได้ว่า “Business Function คือ สิ่งที่จะคิดว่าจะทำหรือสิ่งที่ต้องทำในอนาคต เพื่อที่จะช่วยให้องค์กรสามารถที่จะ ดำเนินงานได้บรรลุวัตถุประสงค์ที่ตั้งไว้”

- การเขียน Function Hierarchy

หลังจากทำการวิเคราะห์และทำความเข้าใจข้อมูลต่างๆ ที่เก็บรวบรวมมาแล้วนั้น ก็จะต้อง ทำการเขียน Business Function และเทคนิคพื้นฐานที่มีผู้นิยมใช้ในการเขียน Business Function ก็คือ การเขียน “Function Hierarchy” ซึ่งหลักในการเขียน Function Hierarchy ก็คือ

- ชื่อของฟังก์ชันจะต้องบ่งบอกถึงคำจำกัดความว่าฟังก์ชันนั้นมีหน้าที่ทำอะไร อย่างชัดเจน
- ชื่อของฟังก์ชันจะต้องไม่ระบุถึงวิธีการปฏิบัติ
- ชื่อของฟังก์ชันจะต้องขึ้นต้นด้วยคำกริยา
- ในการเขียนฟังก์ชันลำดับในแต่ชั้น จะไม่มีความสำคัญ
- ฟังก์ชันที่อยู่ในชั้นที่สูงกว่าจะถูกอธิบายโดยฟังก์ชันที่อยู่ด้านล่างที่แตกออกมาเรื่อยๆ
- การเขียน Function Hierarchy ไม่ควรที่จะเขียน เกินกว่า 7 ระดับ
- สัญลักษณ์ที่ใช้แทน Business Function ใน Function Hierarchy ก็คือ รูป กรอบสี่เหลี่ยมที่มีมุมมนทั้งสี่มุม

- การเขียนหรือการหา Function Dependency Diagram

การเขียน Function Dependency Diagram ก็ต้องการที่จะรู้ถึงลำดับการเกิดหรือลำดับที่ จะต้องทำฟังก์ชันอะไรก่อนหลัง และต้องการที่จะรู้ว่าฟังก์ชันใดจะต้องเกิดก่อนถึงจะทำฟังก์ชันถัดไปได้ การที่ฟังก์ชันหนึ่งต้องขึ้นอยู่กับฟังก์ชันอื่นๆ มี 3 สาเหตุ คือ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data
- Legislation
- Corporate Policy

ดังนั้นการเขียน Function Dependence จะเป็นการเรียงลำดับการทำงานในแง่ของเซตของฟังก์ชันที่สามารถที่จะทำเสร็จและถูกต้องตามความต้องการของธุรกิจและสามารถที่จะใช้ในการตรวจสอบว่าขบวนการทำงานภายในองค์กร นั้นจะติดขัดหรือจะทำให้การทำงานช้าลงอยู่ที่ฟังก์ชันใดบ้าง เพื่อเป็นข้อมูลเบื้องต้นในการทำการจัดการองค์กรใหม่ หรือเพื่อใช้เป็นข้อมูลในการขอการรับรอง ISO ก็ย่อมได้

- การเขียน Process Modeling

การเขียน Process Modeling นั้นเป็นการทำงานในช่วงของการออกแบบระบบงาน การเขียน Process Modeling อาจจะใช้เทคนิคในการเขียน Function Dependence มาเขียนก็ได้

- ดาต้าโฟลว์ไดอะแกรม Dataflow Diagram

ไดอะแกรมนี้จัดเป็น Dependence Diagram เช่น แต่จะพิเศษตรงที่จะแสดง Data Dependence ในรายละเอียดมากขึ้น

ประโยชน์ของไดอะแกรมชนิดนี้ คือ ใช้ช่วยในการตัดสินใจว่าควรจะทำการแตกโปรเซสลงไปเป็นโปรเซสย่อยอีกหรือไม่ จนกระทั่งได้ไดอะแกรมในระดับที่เป็นรายละเอียดที่สุดแล้ว ไดอะแกรมนั้นก็เหมาะที่จะใช้เป็น specification สำหรับงานออกแบบ/เขียนโปรแกรม, เขียนเอกสารคู่มือหรือใช้ออกแบบอุปกรณ์ต่าง ๆ เพื่อใช้ในระบบต่อไป

- การเขียนเอนติตี้รีเลชันชิพโมเดล Entity Relationship Model

ถือเป็น high level conceptual schema หรือ high level conceptual model ซึ่งหมายถึงไม่ได้เก็บเฉพาะความสัมพันธ์ระหว่างข้อมูล แต่ยังเก็บความสัมพันธ์ระหว่าง Rules และ Integrity constraint ด้วยกฎความถูกต้องของข้อมูล

- เอนติตี้ (Entity) คือ สิ่งของหรือสิ่งที่เราสนใจหรือหัวข้อที่เป็นหลักสำคัญ อาจจะเป็นสิ่งจริง ๆ ที่มีอยู่หรือถูกสมมุติขึ้นมาก็ได้ จะเป็นสิ่งที่บรรจุหรือเกี่ยวข้องกับข้อมูลด้านต่าง ๆ ที่จำเป็นต้องรู้
- แอททริบิวต์ (Attribute) คือ รายละเอียดต่าง ๆ ที่ครอบคลุมถึงคุณภาพ, ลักษณะ, ประเภท, จำนวนหรือ ข้อกำหนดคุณลักษณะที่เกี่ยวข้องกับสถานะของเอนติตี้
- ความสัมพันธ์ (Relationship) เป็นสิ่งที่สำคัญในการเชื่อมโยงระหว่าง 2 เอนติตี้ซึ่งสามารถแสดงความสัมพันธ์ได้ทั้งที่เป็น ออฟชั่นนอล(เป็นเส้นประ) โดยมีความหมายว่าถ้าเป็นความสัมพันธ์ แบบออฟชั่นนอลแอททริบิวต์ที่มีความสัมพันธ์จะมีค่าข้อมูลหรือไม่มีค่าของข้อมูลก็ได้ (null) หรือแมนดาทอรี (เส้นทึบ)

จากหลักการของ CASE*METHOD ที่มีการขั้นตอนการทำงานเป็นขั้นตอนตั้งแต่การ
การสร้าง Function Model, Function Hierarchy, Function Dependence, Process Modeling, การค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dataflow Diagram และ Entity Relationship Diagram ในวิทยานิพนธ์ฉบับนี้ได้นำเอาเครื่องมือในการพัฒนาโปรแกรมของบริษัท ออราเคิล มาเป็น เครื่องมือที่ใช้ทดสอบว่าจากหลักการและทฤษฎีดังกล่าว สามารถที่จะนำมาใช้ในการพัฒนาได้จริงหรือไม่

จากการทดลองสรุปได้ว่าทฤษฎีของ CASE*METHOD สามารถที่จะนำมาใช้ในการพัฒนาระบบงานได้จริงและทำให้ระบบงานที่พัฒนาขึ้นมานั้นมีประสิทธิภาพ ในแง่ของการปฏิบัติจริงโดยจะขอเปรียบเทียบในส่วนของ CASE*METHOD กับ เครื่องมือของบริษัท ออราเคิลเป็นลำดับๆ โดยเครื่องมือของออราเคิลที่ใช้มีชื่อว่า Oracle Designer/2000 release 1.3.2 ซึ่งภายในเครื่องมือนี้เราสามารถที่จะแบ่งขั้นตอนการทำงานออกได้เป็น 4 ขั้นตอนใหญ่ๆ คือ

- โพรเซสโมเดลลิง (Process Modeling)

โพรเซสโมเดลลิง เป็นการนำขั้นตอนการทำงานของสิ่งที่สนใจมาเขียนเป็นโมเดล เมื่อโมเดลแล้ว โมเดลนี้จะเป็นพื้นฐานของและหลักการของระบบงาน โดย Designer/2000 จะทำการเก็บรายละเอียดทุกอย่างที่ทำการเขียนลงที่เซิร์ฟเวอร์ ในส่วนที่เรียกว่า “รีพอสิตอรี (Repository)” ซึ่งสามารถที่จะมาแก้ไขเปลี่ยนแปลงได้ตลอดเวลา และยังเป็นข้อมูลเบื้องต้นของการทำงานในขั้นต่อไปอีกด้วย ดังนั้นการที่ทำการเขียน โพรเซสโมเดลลิงจะทำให้รู้ว่าผู้พัฒนาระบบงานมีความเข้าใจกับระบบงานที่จะทำมากน้อยแค่ไหนและรู้ขอบเขตของระบบงานทั้งหมด

- ซิสเต็มโมเดลลิง (Systems Modeling)

ซิสเต็มโมเดลลิง จะเป็นการระบุว่าจะระบบนั้นๆ ต้องการข้อมูลอะไรบ้างหรืออาจจะพูดได้ว่าเป็นส่วนของการออกแบบฐานข้อมูลนั่นเองซึ่งการออกแบบตรงจุดนี้ผู้ออกแบบ ควรที่จะคำนึงความต้องการใช้ข้อมูลในอนาคตไว้ด้วย สำหรับเครื่องในส่วนนี้ที่ Designer/2000 มีให้ใช้ได้แก่

- ฟังก์ชันไฮราซี ไดอะแกรมเมอร์ (Function Hierarchy Diagrammer)
- ดาต้าโฟลว์ ไดอะแกรมเมอร์ (Dataflow Diagrammer)
- เอ็นติตี้ รีเลชันชิฟ ไดอะแกรมเมอร์ (Entity Relationship Diagrammer)

- ซิสเต็มดีไซน์ (Systems Design)

ซิสเต็มดีไซน์ เป็นการออกแบบลักษณะของระบบงาน ว่าจะมีหน้าจอะไรบ้างและมีเมนูเป็นอย่างไร สำหรับเครื่องในส่วนนี้ที่ Designer/2000 มีให้ใช้ได้แก่

- ดาต้า ไดอะแกรมเมอร์ (Data Diagrammer)
- โมดูล สตรัคเจอร์ ไดอะแกรมเมอร์ (Module Structure Diagrammer)
- โมดูล ดาต้า ไดอะแกรมเมอร์ (Module Data Diagrammer)
- โมดูล ลอจิก เนวิกเกตเตอร์ (Module Logic Navigator)
- พรีเฟอเรน เนวิกเกตเตอร์ (Preferences Navigator)

- ซิสเต็ม เจเนอเรชัน (Systems Generation)

ซิสเต็ม เจเนอเรชัน จะเป็นการนำข้อมูลที่มีการกำหนดไว้ในส่วนต่างของระบบงานที่ทำมาตั้งแต่ต้น มาทำการสร้างเป็นต้นแบบของแอปพลิเคชันและจะทำการสร้างส่วนฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และความสัมพันธ์ระหว่างตารางต่าง ๆ ที่ออกแบบเอาไว้จริง ในส่วนของเซิร์ฟเวอร์ สำหรับเครื่องในส่วนนี้ที่ Designer/2000 มีให้ใช้ได้แก่

- ฟอรัม เจนเนอเรเตอร์(Form Generator)
- รีพอร์ท เจนเนอเรเตอร์(Report Generator)
- เซิร์ฟเวอร์ เจนเนอเรเตอร์(Server Generator)

เมื่อทำตามขั้นตอนครบทั้ง 4 ขั้นตอนตามลำดับแล้วก็จะได้ระบบงานที่เป็นลักษณะต้นแบบของระบบงานจริงจะต้องมีการนำไปพัฒนาต่ออีกระดับหนึ่งแต่ก็จะช่วยประหยัดเวลาในการที่จะมานั่งสร้างต้นแบบด้วยวิธีการอื่นๆ จากที่ได้ลองใช้เครื่องมือตัวนี้ดู ภายในมีลักษณะที่สนับสนุนทฤษฎีแนวคิดแบบในการพัฒนาระบบงานแบบ Spiral ซึ่งเป็นแบบที่เหมาะสมที่จะใช้ในการพัฒนาระบบงานที่มีการเปลี่ยนแปลงเรื่อย สิ่งหนึ่งที่เครื่องมือนี้ทำได้ดีมาก ๆ คือ การออกแบบระบบฐานข้อมูล โดยการสร้าง Entity Relationship Diagram ขึ้นมาก่อน ซึ่งเป็นการออกแบบระบบฐานข้อมูลที่เป็นลักษณะ Logical ก่อน ทำให้รู้ถึงความสัมพันธ์ระหว่าง Entity ต่างว่ามีความสัมพันธ์กันอย่างไร และถ้าต้องการที่เปลี่ยนระบบฐานข้อมูลหรือมีการแก้ไขก็กลับมาแก้ไขที่ Entity Relationship Diagram ได้ทันทีแล้วก็ทำการ Generate ตารางใหม่อีกครั้งโดยไม่ต้องเสียเวลามากเท่าไร แล้วก็ยังสามารถประกอบระบบงานโดยไม่ต้องเสียเวลามาทำเอกสารใหม่อีกครั้ง ซึ่งเอกสารที่ทางเครื่องมือชุดนี้มีให้ค่อนข้างที่จะสมบูรณ์ จากการที่ทำตามขั้นตอนของเครื่องนี้และในวิทยานิพนธ์ฉบับนี้ได้นำเอกสารมาเป็นตัวอย่างซึ่งสามารถดูได้ในส่วนของภาคผนวก

ในเรื่องของปัญหาต่างที่พบค่อนข้างที่จะมีมากแต่ก็สามารถที่จะแก้ไขผ่านไปได้ด้วยความลำบากเพราะตัว Designer/2000 รุ่น 1.3.2 เองยังมีข้อจำกัดมากและไม่ค่อยมีหนังสือออกมาสักเท่าไร แล้วอีกประการหนึ่งในส่วนของซิสเต็ม เจนเนอเรเตอร์ นั้นถ้าทำการกำหนดคุณสมบัติผิดไปเพียงนิดเดียว ก็จะทำให้การเจนเนอเรตผิดไปได้ ดังนั้นจะต้องระวังเป็นอย่างมากในส่วนนี้ แต่ก็ยังถือว่าเป็น CASE Tools ที่น่าใช้มากและน่าที่จะเป็นเครื่องมือที่เป็นที่นิยมกันในอนาคตอย่างแน่นอน ในปัจจุบันนี้ทางบริษัท ออราเคิลได้ทำการออก Designer รุ่นใหม่ออกมาได้แก่ Designer6 ซึ่งมีการทำงานโดยทั่วไปไม่แตกต่างจากรุ่นเดิมสักเท่าแต่จะเป็นการตัดขั้นตอนบางขั้นตอนที่ไม่จำเป็นออกไปและยังเพิ่มในส่วนของการเจนเนอเรเตอร์ในส่วนของการสร้างแอปพลิเคชันให้สามารถที่จะเล่นผ่านอินเทอร์เน็ตได้ด้วย และยังสามารถที่จะสร้างสคริปต์ของคำสั่ง SQL ที่ใช้ในการสร้างตารางฐานข้อมูลของ ดาต้าเบสตัวอื่นๆ ได้ด้วย

บรรณานุกรม

- ขวัญตา อาชีวะ ให้สัมภาษณ์, 7 พฤศจิกายน 2540. สัมภาษณ์ บริบาลประสิทธิ์ ผู้สัมภาษณ์.
ขั้นตอนการทำงานในการจัดการฝึกงานนักศึกษา. คณะวิศวกรรมศาสตร์ แผนกสวัสดิการ
 และกิจการนักศึกษา สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- Alison Durnford and Jonathan Russell. 1995. A Guide to Process Modeling. Californis,
 USA. : Oracle Corporation.
- Alison Durnford and Jonathan Russell. 1995. A Guide to System Modeling. Californis,
 USA. : Oracle Corporation.
- Alison Durnford and Jonathan Russell. 1995. A Guide to System Design. Californis,
 USA. : Oracle Corporation.
- Alison Durnford and Jonathan Russell. 1995. A Guide to Generator. Californis, USA
 : Oracle Corporation.
- Amanda Marconi et. al. 1993. Administer the Oracle Database I (Oracle7). Volumn 1
 Californis, USA : Oracle Corporation.
- Amy Coppola and Andy Page. 1995. A Guide to Developer/2000 Generation.
 Volumn 1. Californis, USA : Oracle Corporation.
- Amy Coppola and Andy Page. 1995. A Guide to Developer/2000 Generation.
 Volumn 2. Californis, USA : Oracle Corporation.
- Carl Gohoringer. et. al. 1995. Generate Applications Using Designer/2000.
 Volumn 1. Californis, USA : Oracle Corporation.
- Carl Gohoringer. et. al. 1995. Generate Applications Using Designer/2000. Volumn 2.
 Californis, USA : Oracle Corporation.
- C.J. Date. 1990. An Introlduction to Database System. 5th. England : Addison-
 Wesley.
- Paul Dorsey and Perter Koletzke. 1997. Oracle Designer/2000 Handbook. Californis,
 U.S.A : Mcgraw-Hill.
- Richad Barker and Cliff Longman. 1990. CASE*METHOD Function and Process
 Modelling. England : Addison-Wesley.
- Richad Barker. 1990. CASE*METHOD Tasks and Deliverables. England : Addison-
 Wesley.
- Richad Barker. 1990. CASE*METHOD Entity Ralationship Modelling.
 Wokingham, England : Addison-Wesley.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สารสนเทศลาดกระบัง

Ladkrabang Information Journal

ISSN 0859 - 5208

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Faculty of Information Technology, King Mongkut's Institute of Technology, Ladkrabang, Bangkok 10520

การพัฒนา

และการจัดการ

ระบบสารสนเทศ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การพัฒนาระบบฝึกงานนักศึกษา
คณะวิศวกรรมศาสตร์ สจล. บนระบบฐานข้อมูลไคลน์/เซิร์ฟเวอร์
Client/Server Database Development
of a Training Information System Engineering Faculty of KMITL**

สมเกียรติ บริบาลประสิทธิ์* รศ.ดร.ศุภมิตร จิตตะขุโสธร**

บทคัดย่อ

บทความนี้ขอเสนอการพัฒนาระบบฐานข้อมูลเชิงสัมพันธ์แบบไคลน์/เซิร์ฟเวอร์ โดยในส่วนของการออกแบบฐานข้อมูลได้ใช้โนแอมเป็นเครื่องมือในการออกแบบ และได้ทำการพัฒนาระบบฝึกงานนักศึกษา ของแผนกกิจกรรมนักศึกษา คณะวิศวกรรมศาสตร์ สจล. จากหลักการของไคลน์/เซิร์ฟเวอร์จะมีการแยกส่วนออกเป็นสองส่วน คือ ส่วนของผู้ใช้ (ไคลน์) และ ส่วนของผู้ให้บริการ (เซิร์ฟเวอร์) ซึ่งการที่ทั้งสองส่วนจะทำงานร่วมกันได้จะต้องอาศัย SQL Link ทำหน้าที่ติดต่อและเข้าถึงจากแอปพลิเคชันสู่ระบบบริหารข้อมูลจึงทำให้ระบบสามารถทำงานได้ ในส่วนของผู้ให้บริการ (ไคลน์) พัฒนาโดยใช้เดลไฟล์ (Delphi) และในส่วนของผู้ให้บริการ (เซิร์ฟเวอร์) ได้ใช้ SQL Server 6.0 เป็นตัวที่คอยจัดการฐานข้อมูลบนระบบปฏิบัติการ Windows NT4.0

ABSTRACT

This paper presents client/server relational database system development. The designing of database used NIAM. The application is a student training information system for engineering faculty of KMITL. The separation of front-end interface and back-end data services referred to a client/server database technology. The system using SQL Link (native of Delphi 3.0) driver connect to between the front-end and the database server. The database system is implemented on SQL SERVER 6.0 on Windows NT 4.0. The front-end system is implemented by using DELPHI 3.0 on Windows'95.

* นักศึกษาปริญญาโท หลักสูตรวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพมหานคร 10520

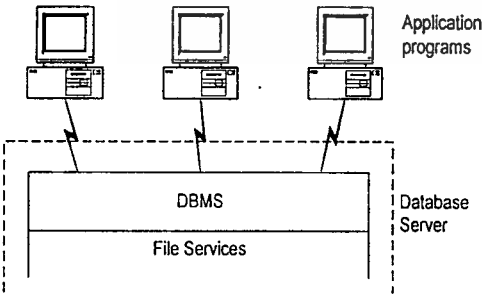
** รองศาสตราจารย์ ประจำคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพมหานคร 10520

บทนำ

ในปัจจุบันนี้การที่คณะวิศวกรรมศาสตร์ จะทำการจัดส่งนักศึกษาออกไปฝึกงานตามหน่วยงานต่างๆ นั้นขั้นตอนการทำงานมีความซับซ้อนทำให้เสียเวลาในการทำงานไปโดยเปล่าประโยชน์ จึงได้ทำการสำรวจและศึกษาถึงขั้นตอนการทำงานองเจ้าหน้าที่ผู้ปฏิบัติงาน แล้วเห็นว่าน่าที่จะนำคอมพิวเตอร์มาช่วยในการทำงาน พร้อมกันนั้นในส่วนของคุณสมบัติยังต้องการระบบจัดการฐานข้อมูลที่ดี เพื่อแก้ปัญหาต่างๆ ที่เกิดขึ้นในปัจจุบัน เพื่อให้เกิดประสิทธิภาพในการทำงานที่ดีและเกิดความรวดเร็ว ดังนั้นจึงได้ทำการวิจัยและพัฒนา ระบบฝึกงานนักศึกษา บนระบบจัดการฐานข้อมูลแบบไคลน์/เซิร์ฟเวอร์ ขึ้นมาเพื่อให้เจ้าหน้าที่วางแผนกิจกรรมนักศึกษา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้ใช้ในการปฏิบัติงานในส่วนนี้ ซึ่งระบบที่ทำการออกแบบและพัฒนาขึ้นมานั้น ครอบคลุมการทำงานทุกส่วนไม่ว่าจะเป็นการเก็บข้อมูล การออกจดหมาย การทำรายงานสรุป และการสืบค้นข้อมูล ตามความต้องการของผู้ปฏิบัติงาน

ระบบไคลน์/เซิร์ฟเวอร์

ระบบไคลน์/เซิร์ฟเวอร์ได้นำเอาข้อเด่นของระบบรวมศูนย์ประมวลผล (Centralized Host System) และระบบการใช้ทรัพยากรร่วมกันในเครือข่ายคอมพิวเตอร์ โดยแทนที่จะมีเพียงไฟล์เซิร์ฟเวอร์ (file server) บนเครื่อง เซิร์ฟเวอร์ ยังมี DBMS ที่ทำหน้าที่ Data Services อีกด้วย โดยลักษณะของไคลน์/เซิร์ฟเวอร์ เป็นการแบ่งสรรทรัพยากรต่างๆ ให้ใช้ร่วมกันโดยที่ผู้ใช้คิดว่าทรัพยากรเหล่านั้นอยู่บนเครื่องของตนเอง เช่น การแบ่งการใช้ข้อมูล, การแบ่งการใช้หน่วยประมวลผล, การแบ่งโปรแกรมการใช้งาน เป็นต้น



รูปที่ 1 ระบบแยกส่วนผู้ใช้บริการ/ผู้ให้บริการ

ระบบการทำงานแบบไคลน์/เซิร์ฟเวอร์ โดยทั่วไปจะเป็นการทำงานที่มีการแบ่งการประมวลผลและฐานข้อมูลออกจากกันหรืออาจมองง่าย ๆ คือการแบ่งส่วนของ การติดต่อผู้ใช้ ออกจากเซิร์ฟเวอร์นั่นเอง ซึ่งเรียกว่า ฟรอนเอนด์ (front-end) และแบ็กเอนด์ (back end) ตามลำดับ โดยในส่วนของการติดต่อผู้ใช้งานนั้นจะเป็นโปรแกรมประยุกต์ต่างๆ ที่เรียกใช้ข้อมูลจากตัวเซิร์ฟเวอร์ โดยการส่งรีเควส (request) ที่โปรแกรมประยุกต์ต้องการจากฐานข้อมูลที่เก็บไว้ที่เซิร์ฟเวอร์ ซึ่งอาจจะ เป็นข้อมูลหรือโปรแกรม หรือการบริการอื่นๆ จากเซิร์ฟเวอร์ก็ได้ เมื่อส่วนของการติดต่อผู้ใช้ (front-end) ได้รับรีเควส (request) ตามต้องการแล้วอาจจะนำรีเควสที่ได้มาทำการประมวลผลที่ตัวฟรอนเอนด์เอง ซึ่งถือว่าเป็นข้อดีของระบบไคลน์/เซิร์ฟเวอร์ เพราะจะช่วยลดการทำงานของเซิร์ฟเวอร์ ทำให้การทำงานทำได้รวดเร็วขึ้นและมีประสิทธิภาพมากขึ้น

ขั้นตอนการพัฒนา ระบบสารสนเทศ

การที่จะสร้างหรือพัฒนาระบบขึ้นมาใช้ในองค์กรหรือหน่วยงานนั้น จะต้อง มีลำดับขั้นตอนในการทำงานซึ่งพอที่จะสรุปออกมาเป็นขั้นตอนใหญ่ๆ ได้ 3 ขั้นตอน คือ

ขั้นตอนที่ 1 การวิเคราะห์ระบบ

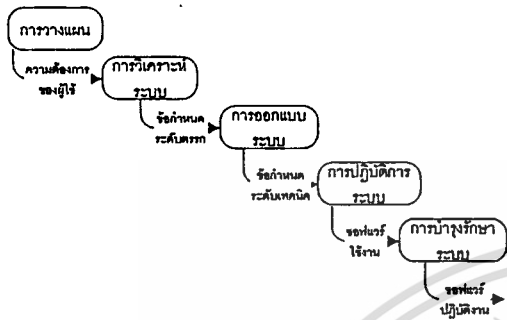
การวิเคราะห์ระบบเป็นการศึกษาถึงปัญหาของการปฏิบัติงานในระบบปัจจุบันที่ดำเนินอยู่ว่ามีปัญหาอะไรบ้าง วัตถุประสงค์ของการวิเคราะห์ระบบก็คือ จะต้องทำการค้นหาสาเหตุที่ทำให้เกิดปัญหาขึ้นมาว่ามาจากสาเหตุอะไร และเสนอระบบที่สามารถจะช่วยแก้ปัญหาที่เกิดขึ้น เมื่อสามารถที่จะสรุปปัญหาต่างๆ และหาวิธีการแก้ปัญหาเหล่านั้นก็ถือว่าเราได้ความต้องการของผู้ใช้ ได้รายละเอียดขั้นตอนการทำงาน ของระบบงานปัจจุบัน และเค้าโครงของระบบงานใหม่

ขั้นตอนที่ 2 การออกแบบระบบ

การออกแบบระบบเป็นการที่เรานำข้อมูลที่เรารวบรวมและวิเคราะห์จากขั้นตอนที่ 1 มาทำการเปลี่ยนแปลงให้เป็น Technical Design Specification ซึ่งผลที่ได้จะเป็นการกำหนดว่าโปรแกรมคอมพิวเตอร์ที่จำเป็นมีอะไรบ้าง จะต้องเขียนโปรแกรมอย่างไร มีการออกแบบส่วนข้อมูลนำเข้า (Inputs) ข้อมูลนำออก (Output) แฟ้มข้อมูล (Files) ฐานข้อมูล (Databases) ตลอดจนการควบคุมการประมวลผล (Control Design)

ขั้นตอนที่ 3 การปฏิบัติการระบบ

การปฏิบัติการออกแบบระบบคือการที่นำส่วนของข้อมูลที่ได้จากขั้นตอนที่ 2 มาทำการเขียนโปรแกรมคอมพิวเตอร์ ที่จะนำไปใช้งานจริง



รูปที่ 2 แสดงขั้นตอนการพัฒนากระบวนงานสารสนเทศ

การศึกษาลักษณะระบบเดิม

ลักษณะงานทั่วไปของระบบฝึกงานนักศึกษาเดิมนั้นจะเป็นระบบที่ผู้ใช้ต้องทำงานด้วยตนเองโดยมิได้นำระบบคอมพิวเตอร์เข้ามาช่วยเหลือนแต่อย่างใดไม่ว่าจะเป็นการจัดเก็บเกี่ยวกับฐานข้อมูลที่มีมากและต้องการปรับปรุงทุก ๆ ปี โดยมิได้นำระบบคอมพิวเตอร์มาช่วยอำนวยความสะดวกต่าง ๆ ยกเว้นแต่ในเรื่องของการจัดทำเอกสารที่ต้องการรูปแบบที่สวยงามและรายงานการสรุปผลต่าง ๆ ซึ่งก็จะใช้โปรแกรมสำเร็จรูปมาช่วยในส่วนนี้เท่านั้น แต่ในส่วนของการจัดการเกี่ยวกับฐานข้อมูลก็ยังไม่ได้มีการนำคอมพิวเตอร์เข้ามาช่วยแต่อย่างใด สามารถที่จะสรุปออกมาเป็นข้อได้ดังนี้

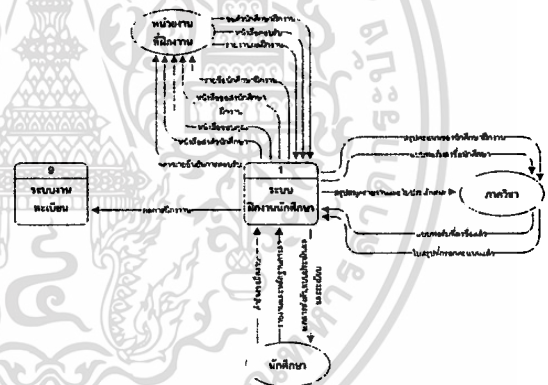
1. เป็นระบบที่ยังมิได้นำคอมพิวเตอร์เข้ามาช่วยในการประมวล การทำงานส่วนใหญ่จะอยู่ในลักษณะที่ผู้ปฏิบัติงานทำงานด้วยตนเองซึ่งอาจมีการนำเอาคอมพิวเตอร์เข้ามาช่วยเหลือนบ้างเพียงเล็กน้อยกับการทำงานบางอย่าง เช่น การออกเอกสาร และ รายงานต่าง ซึ่งข้อมูลที่จะนำมาใช้ในการออกเอกสารและรายงานนั้นผู้ปฏิบัติงานจะต้องทำการประมวลผลให้เสร็จสิ้นก่อนจึงจะนำข้อมูลที่ได้มาประมวลผลแล้วมาใช้ในการออกเอกสารและรายงานซึ่งคอมพิวเตอร์ที่นำมาช่วยนั้นได้ช่วยเหลือนในเรื่องของรูปแบบและความเป็นระเบียบเรียบร้อยเท่านั้น

2. การจัดเก็บข้อมูลของระบบจะเห็นว่าข้อมูลต่าง ๆ จะเป็นเอกสารเป็นส่วนใหญ่ ซึ่งการจัดเก็บจะต้องทำการจัดเก็บเอกสารเหล่านั้นเข้าแฟ้ม ซึ่งในแต่ละปีการศึกษา

เอกสารที่จะต้องจัดเก็บเป็นจำนวนมากจึงมีปัญหาเกี่ยวกับอุปกรณ์และสถานที่ที่ใช้จัดเก็บ เท่านั้นยังไม่พอยังเป็นอุปสรรคในเรื่องการค้นหาข้อมูลและการปรับปรุงข้อมูลถึงแม้ว่าจะจัดเก็บเอกสารเหล่านั้นอย่างเป็นระเบียบแล้วก็ตาม

3. เกิดความซ้ำซ้อนในการทำงานมาก เช่น การออกจดหมายที่มีรูปแบบและข้อความเหมือนกันเพียงอาจจะเปลี่ยนหรือแตกต่างกันที่ข้อมูลบางอย่าง ก็จะทำให้ผู้ปฏิบัติงานทำงานในส่วนนี้ซ้ำซ้อนกันทำให้เสียเวลาไปโดยเปล่าประโยชน์

4. ในแนวทางการปฏิบัติในปัจจุบันนี้จะมีผู้รับผิดชอบในส่วนนี้และเข้าใช้เกี่ยวกับการทำงานเพียงคนเดียวทำให้งานเกิดความล่าช้าหรือหยุดการทำงานได้หากเกิดเหตุการณ์บางอย่างที่เราไม่คาดคิด เช่น เจ้าหน้าที่คนนั้นเกิดลาออกหรือป่วย เป็นต้น จะไม่มีเจ้าหน้าที่คนอื่นที่จะสามารถมาปฏิบัติหน้าที่แทนได้ในทันที จะต้องเกิดความล่าช้าซึ่งอาจจะส่งผลกระทบต่อการศึกษาของนักศึกษาได้



รูปที่ 3 ผังแสดงการทำงานโดยรวม

ขั้นตอนการทำงาน

1. ขั้นตอนในการเตรียมข้อมูล

สำหรับการทำงานของระบบเดิมนั้น การเตรียมข้อมูลนักศึกษาฝึกงาน จะเตรียมในลักษณะที่เป็นข้อมูลของปีต่อปี ซึ่งสามารถแยกได้เป็น

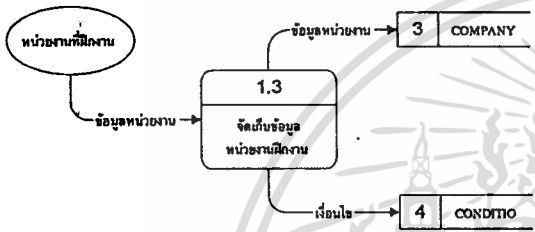
- 1.1 การเตรียมข้อมูลนักศึกษาที่มีสิทธิ์ฝึกงานประจำปีการศึกษานั้น ๆ โดยจะเตรียมข้อมูลจากใบรายชื่อนักศึกษาที่ทางแผนกทะเบียนเป็นผู้จัดทำ ส่วนรายละเอียดของข้อมูลนอกจากนั้น จะถูกจัดเตรียมเมื่อมีความจำเป็นเท่านั้น เช่น ข้อมูลเกรดเฉลี่ยของนักศึกษา ถ้ามีหน่วยงานใดต้องการทางเจ้าหน้าที่ผู้ปฏิบัติงานจึงจะทำหนังสือขอข้อมูลดังกล่าวจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนกทะเบียน หรือข้อมูลอื่นๆ ที่ทางหน่วยงานต้องการนั้นๆ ต้องการจึงจะทำหนังสือขอข้อมูลจากหน่วยงานที่เกี่ยวข้อง

1.2 การเตรียมข้อมูลของหน่วยฝึกงานทั้งหมด สำหรับการจัดส่งเอกสารขออนุญาตส่งนักศึกษาฝึกงานซึ่งข้อมูลของบริษัทต่างๆ นั้นได้เก็บรวบรวมเอาไว้แล้ว แต่ก็จะต้องมีการปรับปรุงข้อมูลให้ทันสมัยอยู่เสมอ เช่น อาจต้องตามแก้ไขที่อยู่ของบางบริษัทที่อาจจะย้ายสำนักงาน, ตามแก้ไขชื่อผู้ที่จะต้องติดต่อด้วย แม้ว่าทางแผนกจะมีข้อมูลของบริษัทอยู่แล้วก็ตาม ก็เป็นการยากที่จะทำการค้นหาและปรับปรุงข้อมูลเหล่านั้นให้ทันสมัยอยู่เสมอ



รูปที่ 4 ผังการไหลข้อมูลในการจัดเก็บข้อมูลหน่วยงาน

1.3 การเตรียมข้อมูลของคณะกรรมการอุตสาหกรรมสัมพันธ์ เนื่องจากคณะกรรมการอุตสาหกรรมสัมพันธ์ มีการเปลี่ยนแปลงกันทุกปีการศึกษาเพราะคณะกรรมการจะอยู่ในวาระๆ ละหนึ่งปีการศึกษาเท่านั้น ดังนั้นเมื่อเริ่มปีการศึกษาใหม่ก็จะต้องมีการแต่งตั้งและคัดเลือกใหม่เมื่อได้กรรมการชุดใหม่แล้วทางเจ้าหน้าที่ก็จะต้องบันทึกรายชื่อคณะกรรมการชุดใหม่แทนคณะกรรมการชุดเก่าทั้งหมด

2. ขั้นตอนในการออกเอกสารขออนุญาตส่งนักศึกษาฝึกงานประจำปีการศึกษา

ในทางปฏิบัติทางเจ้าหน้าที่ผู้รับผิดชอบจะต้องทำการพิมพ์จดหมายส่งตัวนักศึกษาเองซึ่งจดหมายส่งตัวนักศึกษาอาจมีการใช้เครื่องคอมพิวเตอร์เข้ามาช่วยแต่เราจะเห็นว่าโดยข้อความในจดหมายส่งตัวนั้นข้อความส่วนใหญ่จะเหมือนกันเพียงแตกต่างกันที่ชื่อบริษัท, ชื่อผู้ติดต่อ เท่านั้น จึงทำให้การทำงานในส่วนนี้ของเจ้าหน้าที่เกิดความซ้ำซ้อนและเกิดความล่าช้าในการทำงานมาก

3. รับและจัดเก็บเงื่อนไขการฝึกงาน

เมื่อทางเจ้าหน้าที่ได้ทำการจัดส่งเอกสารขอส่งนักศึกษาฝึกงานแล้วทางบริษัทก็จะจัดส่งเอกสารตอบรับกลับมาพร้อมทั้งจัดส่งเอกสารรายละเอียดกฎเกณฑ์เงื่อนไขที่จะรับนักศึกษาเข้าฝึกงาน ทางเจ้าหน้าที่จะต้องทำการจัดเก็บเงื่อนไข

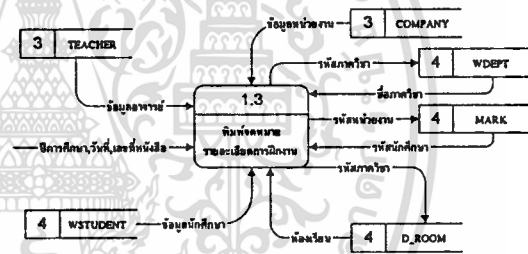
และรายละเอียดเหล่านั้นไว้เพื่อเป็นกฎเกณฑ์ในการคัดเลือกนักศึกษาฝึกงาน

4. จัดทำแบบฟอร์มการลงชื่อฝึกงานให้กับภาควิชา

เมื่อได้รับจดหมายตอบรับกลับมาจากทางหน่วยงานแล้วจะต้องทำการลงชื่อฝึกงานเพื่อส่งให้กลับภาควิชา และทางภาควิชาจะจัดให้มีการลงชื่อฝึกงานของนักศึกษาฝึกงาน

5. จัดทำเอกสารยืนยันการส่งนักศึกษาฝึกงานกับหน่วยงาน

เมื่อทราบว่านักศึกษาคนใดลงชื่อจะฝึกงานกับหน่วยงานใด ก็จะทราบว่าจำนวนนักศึกษาที่ลงชื่อฝึกงานครบตามจำนวนนักศึกษาที่หน่วยงานต้องการนั้นหรือไม่ ถ้าครบก็จะจัดทำเอกสารยืนยันการจัดส่งนักศึกษาฝึกงานเพื่อยืนยันการส่งนักศึกษาฝึกงานกับหน่วยงาน แต่ถ้าไม่ครบจะต้องทำการแจ้งถึงจำนวนที่หน่วยงานต้องการและจำนวนนักศึกษาที่จะจัดส่งไปฝึกงานจริง



รูปที่ 5 ผังการไหลข้อมูลในการพิมพ์รายละเอียดการฝึกงาน

6.จัดทำเอกสารที่ใช้ในการรายงานตัวเข้าฝึกงานของนักศึกษาฝึกงาน

เอกสารดังกล่าวจะเป็นเอกสารที่จะใช้ในการส่งมอบนักศึกษาฝึกงานให้กับหน่วยงานซึ่งจะประกอบไปด้วยรายละเอียดต่างๆ เกี่ยวกับตัวของนักศึกษา 1 ฉบับ ได้แก่เอกสารส่งตัวนักศึกษา และอีกหนึ่งฉบับเป็นรายละเอียดการฝึกงานของนักศึกษาประจำปีการศึกษา

7. รับและจัดทำใบสรุปคะแนนการฝึกงาน

เมื่อการฝึกงานของนักศึกษาสิ้นสุดลงทางหน่วยงานที่รับนักศึกษาฝึกงานจะทำการจัดส่งใบประเมินผลการฝึกงานของนักศึกษากลับมายังแผนกกิจกรรมนักศึกษา เมื่อได้รับใบประเมินจากทางหน่วยงานฝึกงานแล้วก็จะนำมาจำแนกออกตามภาควิชาของนักศึกษาฝึกงานและทำการสรุปผลการต่างๆ ลงในตารางสรุปคะแนนการฝึกงาน จากนั้นจะส่งใบสรุปคะแนนนี้ไปให้อาจารย์ที่ปรึกษาฝึกงานประจำภาควิชาพิจารณาเกรดการฝึกงานของนักศึกษาต่อไป

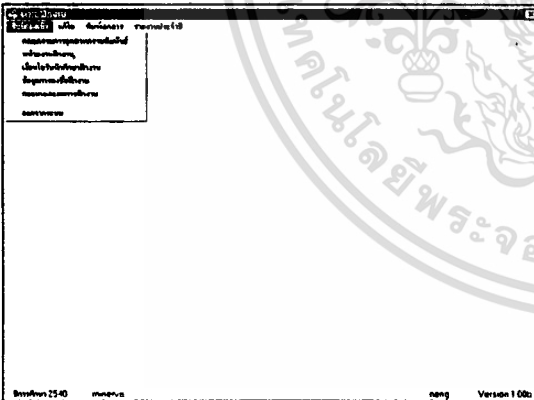
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. สรุปและประเมินผลการฝึกงานของนักศึกษา เมื่อได้ข้อมูลผลการฝึกงานของนักศึกษาประจำปี การศึกษานั้น ๆ มาแล้วจะนำข้อมูลเหล่านั้นมาประเมินการฝึกงานประจำปีการศึกษา เพื่อที่จะวิเคราะห์แนวโน้มความเชื่อถือของหน่วยงานต่าง ๆ ที่มีต่อคณะและเพื่อทำการปรับปรุงแนวทางการเรียนการสอนให้เหมาะสมมากยิ่งขึ้น

9. จัดทำใบประกาศผลการฝึกงาน
ในระบบเดิมนั้นจะประกาศผลการฝึกงานของนักศึกษาโดยการนำเอาใบรายชื่อนักศึกษาที่จัดทำโดยแผนกทะเบียนมากรอกเกรดฝึกงานของนักศึกษาเพื่อติดประกาศ

การศึกษาระบบใหม่

ลักษณะทั่วไปของระบบที่พัฒนาขึ้นมาใหม่นั้นจะเป็นระบบที่ผู้ใช้ต้องทำงานกับระบบฐานข้อมูลซึ่งรูปแบบของการทำงานเป็นแบบไคลน์/เซิร์ฟเวอร์ โดยได้นำระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการกับฐานข้อมูลที่มีอยู่ และอำนวยความสะดวกต่าง ๆ ตั้งแต่การใช้ระบบคอมพิวเตอร์ในการจัดเก็บข้อมูลและสร้างเอกสารที่มีรูปแบบสวยงามและมีระเบียบแบบแผนมากยิ่งขึ้น โดยการเตรียมข้อมูลให้กับเอกสารนั้นเป็นหน้าที่ของโปรแกรมและระบบฐานข้อมูลที่ทำการพัฒนาขึ้นมา สามารถที่จะสรุปออกเป็นข้อ ๆ ได้ดังนี้



รูปที่ 6 หน้าจอเมนูหลักในส่วนของการเพิ่มข้อมูล

1. เป็นระบบที่มีการนำเอาคอมพิวเตอร์เข้ามาช่วยในการประมวลผลข้อมูล ซึ่งการทำงานส่วนมากจะมีลักษณะที่ใช้ เพียงแต่กรอกรายละเอียดข้อมูลที่เป็นให้กับระบบโดยจะมีหน้าจอสำหรับทำการกรอกข้อมูล จากนั้นเมื่อผู้ใช้ยืนยันความถูกต้องของข้อมูลที่กรอกข้อมูลเหล่านั้นก็จะถูกจัดเก็บไว้ที่ตัวเซิร์ฟเวอร์ และเมื่อผู้ใช้ต้องการที่จะนำข้อมูลมาใช้งานไม่ว่าจะเป็นการออกเอกสารการทำงานส่วนใหญ่อีกจะ

ตกอยู่กับคอมพิวเตอร์เพียงแต่ผู้ใช้กำหนดสิ่งต่าง ๆ ที่ต้องการ และข้อมูลที่เป็นบางอย่างสำหรับการออกเอกสาร โดยที่ข้อมูลสำหรับการออกเอกสารและรายงานจะถูกจัดเตรียมโดยระบบคอมพิวเตอร์ที่ถูกพัฒนาขึ้นมาใหม่ ซึ่งผู้ใช้ไม่ต้องอาศัยแอปพลิเคชันอื่น ๆ มาช่วยในการออกเอกสารรายงาน ไม่ว่าจะเป็นการออกจดหมาย, ตาราง, และกราฟ เป็นต้น

2. การจัดเก็บข้อมูลของระบบใหม่นั้นอยู่ในรูปของระบบฐานข้อมูล ซึ่งจัดเก็บอยู่ที่เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นดาต้าเบสเซอร์เวอร์ ซึ่งผู้ใช้สามารถมองเห็นข้อมูล, ทำการแก้ไขข้อมูลและการนำข้อมูลต่าง ๆ ที่มีอยู่ไปใช้ได้โดยผ่านฟังก์ชันการทำงานของแอปพลิเคชันของระบบที่พัฒนาขึ้นมา ดังนั้นในการค้นหาข้อมูลหรือ การนำข้อมูลมาใช้งานในแต่ละครั้งจึงไม่ใช่เรื่องยุ่งยากและซับซ้อนอีกต่อไป นอกจากนี้ยังช่วยแก้ปัญหาในเรื่องของการแก้ไขข้อมูลสามารถทำได้สะดวกรวดเร็ว และอีกประการหนึ่งก็คือช่วยลดความซ้ำซ้อนของการเก็บข้อมูลได้อีกด้วย

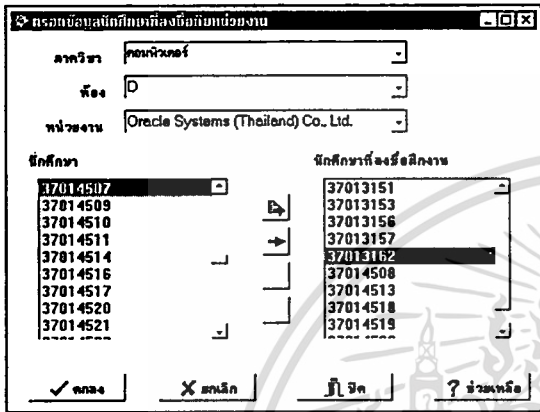
A screenshot of a data entry form titled 'เก็บข้อมูลหน่วยงาน' (Collect Data of Organization). The form contains several fields with pre-filled information: 'ชื่อหน่วยงาน' (Organization Name) is 'Oracle Systems (Thailand) Co., Ltd.', 'ที่อยู่' (Address) is 'ชั้น 16-17 อาคารราชนาถ 952 ถนนพระราม 4 กรุงเทพฯ' (16-17th floor, Rajanath Building 952, Rama 4 Road, Bangkok), 'เบอร์โทรศัพท์' (Phone Number) is '(662)632-9400-1', 'เบอร์โทรสาร' (Fax Number) is '(662)632-9349', 'ผู้รับผิดชอบ' (Responsible Person) is 'จอห์น แอล อดส์' (John Al Oads), and 'ตำแหน่ง' (Position) is 'กรรมการผู้จัดการ' (Managing Director). At the bottom, there are four buttons: 'ตกลง' (OK), 'ยกเลิก' (Cancel), 'ไป' (Next), and '? ช่วยเหลือ' (Help).

รูปที่ 7 หน้าจอกรอกข้อมูลหน่วยงานฝึกงาน

3. งานต่าง ๆ นั้นจะไม่ขึ้นกับความรับผิดชอบของคนเพียงคนเดียว เนื่องจากการออกแบบในการพัฒนาโปรแกรมนั้นออกแบบมาตามแนวความคิดของระบบไคลน์/เซิร์ฟเวอร์ซึ่งเป็นระบบที่มีการทำงานเป็นแบบหลายคนได้ (Multiuser) ซึ่งผู้ใช้แต่ละคนไม่จำเป็นต้องรู้จำข้อมูลต่าง ๆ มิฉะนั้นอยู่บ้างและถูกเก็บอยู่ที่ใดเพียงขอให้ทราบถึงขั้นตอนการทำงานกับ แอปพลิเคชันที่มีอยู่เท่านั้นก็เพียงพอแล้ว นั้นหมายความว่าการทำงานทั้งหมดของระบบฝึกงานใหม่นั้นจะสามารถกระจายงานให้กับบุคลากรได้หลายๆ คน ถ้าบุคคลคนนั้นไม่สามารถที่จะมาทำงานได้ การทำงานของระบบก็สามารถที่จะดำเนินงานต่อไปได้เพราะมีผู้สามารถทำงานแทน

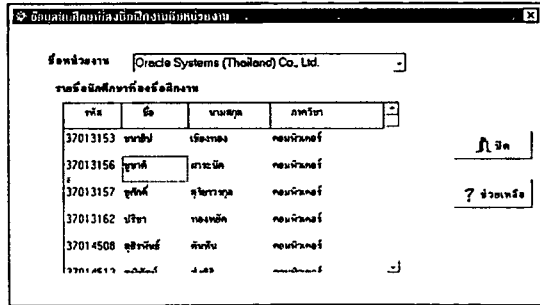
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันได้ถ้าสามารถที่จะใช้แอปพลิเคชันที่มีอยู่ได้โดยไม่ต้องทราบถึงข้อมูลที่ถูกจัดเก็บและขั้นตอนการทำงานที่แท้จริงและนอกจากนี้เมื่อจำเป็นที่จะต้องเปลี่ยนบุคลากรที่รับผิดชอบระบบงานอยู่ ไม่จำเป็นที่จะต้องเปลี่ยนแปลงการทำงานของระบบงานที่มีอยู่เดิม



รูปที่ 8 หน้าจอกรอกข้อมูลลงชื่อฝึกงาน

4. ความซ้ำซ้อนในการทำงานบางอย่างไม่ตกอยู่กับการทำงานของผู้ใช้แต่จะตกอยู่กับการทำงานของระบบคอมพิวเตอร์แทนซึ่งทำให้งานที่ออกมามีประสิทธิภาพของการทำงานสูงขึ้น เช่น การออกจดหมายที่มีรูปแบบที่เหมือนกันให้กับหลายๆ หน่วยงานการทำงานจะเป็นเพียงแค่ผู้ใช้เข้าไปยังเมนูการทำงานในส่วนของการพิมพ์จดหมายและเลือกเรื่องของการจดหมายที่ต้องการจะพิมพ์ เลือกหน่วยงานต่างๆ ที่เป็นผู้รับผิดชอบและยืนยันการพิมพ์ โปรแกรมจะเป็นผู้จัดการเตรียมข้อมูลสำหรับจดหมายดังกล่าวโดยการค้นหาข้อมูลที่มีอยู่ในฐานข้อมูลจนครบถ้วนและนำมาผนวกกับรูปแบบของจดหมายที่มีอยู่จากนั้นก็พิมพ์เอกสารออกทางเครื่องพิมพ์ ทำเช่นนั้นจนครบตามจำนวนหน่วยงานที่ผู้ใช้เลือกขึ้นมาตั้งแต่แรก เป็นต้น



รูปที่ 9 หน้าจอค้นหาข้อมูลนักศึกษาฝึกงานแยกตามหน่วยงาน

5. ระบบที่พัฒนาขึ้นมานั้นเป็นการทำงานบนวินโดวส์ ซึ่งยึดหลักการการทำงานแบบ มูฟแอนด์คลิก (move and click) ทำให้สามารถใช้งานได้ง่ายมาก ดังนั้นการเรียนรู้ในการทำงานจึงไม่จำเป็นต้องอาศัยเวลายาวนานนัก และการทำงานส่วนใหญ่จะอยู่ในรูปของหน้าจอกรอกข้อมูลก่อนนำไปเก็บลงในฐานข้อมูลเท่าที่สามารถตรวจสอบได้เพื่อความถูกต้องของข้อมูลมีมากขึ้น

6. ระบบฝึกงานนักศึกษายังมีเครื่องมือที่จะช่วยลดการกรอกข้อมูลนักศึกษาโดยการดึงข้อมูลนักศึกษาจากฐานข้อมูลของระบบทะเบียนทำให้สามารถลดงานของเจ้าหน้าที่ลงไปได้มากและทำให้ข้อมูลนักศึกษามีความถูกต้องเหมือนกับข้อมูลของแผนทะเบียนนักศึกษา

การแก้ไขปัญหาดังกล่าว ในระบบเดิม

1. ปัญหาในการจัดเก็บข้อมูลเนื่องจากระบบงานฝึกงานนักศึกษานั้นต้องอาศัยข้อมูลเป็นจำนวนมากในการทำงาน ดังนั้นจึงจำเป็นต้องมีระบบการเก็บข้อมูลที่เป็นระบบเพื่อความสะดวกในการนำข้อมูลมาใช้งาน การนำระบบคอมพิวเตอร์และระบบฐานข้อมูลมาใช้จะช่วยให้การบริหารฐานข้อมูลของระบบเป็นไปอย่างถูกต้อง เริ่มตั้งแต่การจัดเก็บข้อมูล การค้นหาและแก้ไขข้อมูล ตลอดจนการนำเอาข้อมูลมาใช้งานโดยการรายงานส่วนใหญ่ในการทำงานถูกโยนไปให้ระบบคอมพิวเตอร์และระบบจัดการฐานข้อมูลแทนที่จะเป็นการทำงานโดยบุคคลผู้ทำงานในระบบฝึกงานนักศึกษา

1. ปัญหาในการพิมพ์เอกสาร ในการพิมพ์เอกสารหรือจดหมายที่ใช้ในงานด้านนี้จะมีกรพิมพ์เอกสารหรือจดหมายที่มีลักษณะเหมือนกันเป็นจำนวนมาก แต่อาจจะต่างกันเพียงชื่อและที่อยู่ของผู้รับผิดชอบหรือเอกสารเท่านั้น ดังนั้นจึงเกิดปัญหาคือ ผู้พิมพ์จดหมายจะต้องทำงานที่ซ้ำซากและเสียเวลาในการพิมพ์มากแม้ว่าผู้พิมพ์จะมีความชำนาญในการพิมพ์มากก็ตาม ระบบที่พัฒนาขึ้นมาใหม่จะช่วยอำนวยความสะดวกในเรื่องของการพิมพ์เอกสารโดยเพียงแค่ผู้ใช้เข้าไปยังเมนูการทำงานในส่วนของการพิมพ์เอกสารและเลือกประเภทของเอกสารที่ต้องการจะพิมพ์ เลือกส่วนประกอบต่างๆ ที่เกี่ยวข้องกับเอกสารและยืนยันการพิมพ์ โปรแกรมจะเป็นผู้จัดการเตรียมข้อมูลสำหรับเอกสารดังกล่าวโดยการค้นหาข้อมูลที่อยู่ในฐานข้อมูลจนครบถ้วนและนำมาผนวกกับรูปแบบของเอกสารที่มีอยู่จากนั้นก็ทำการพิมพ์เอกสารออกทางเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ปัญหาในการค้นหาข้อมูล เนื่องจากระบบงานฝึกงานนักศึกษาจำเป็นต้องอาศัยข้อมูลเป็นจำนวนมากในการทำงาน ดังนั้นในการค้นหาข้อมูลสักอย่างเกี่ยวกับนักศึกษานั้นไม่ใช่เรื่องง่ายเลยที่จะทำการค้นหาด้วยมือจากเอกสารที่มีอยู่ซึ่งข้อมูลอาจจะไม่อยู่ในเอกสารชุดเดียวกันก็ยิ่งทำให้การค้นหาล่าช้ายิ่งขึ้น ระบบที่พัฒนาขึ้นมาใหม่จะช่วยในการค้นหาและแก้ไขข้อมูลได้อย่างรวดเร็ว

3. ปัญหาในการสรุปผลข้อมูล เนื่องจากการสรุปข้อมูลต่าง นั้นจำเป็นที่จะต้องอาศัยข้อมูลจากหลายที่ (เอกสารหลายชุด) และยังจะมีการคำนวณทางสถิติอีกซึ่งมีโอกาสผิดพลาดได้มากเมื่อทำการคำนวณด้วยมือ และในบางครั้งอาจจะต้องการสรุปข้อมูลต่างๆ เหล่านั้นออกมาเป็นกราฟเพื่อความสะดวกในการพิจารณาผลสรุปในแต่ละปี นอกจากนี้ยังมีการนำข้อมูลจากปีก่อนมาใช้ในการสรุปผลอีกเพื่อดูแนวโน้มความต้องการการรับนักศึกษาฝึกงานในสาขาวิชาต่างๆ ของตลาดแรงงาน แอปพลิเคชันที่พัฒนาขึ้นมานั้นสามารถจะสรุปผลการฝึกงานออกมาเป็นตารางและกราฟต่างได้ โดยที่ผู้ใช้ไม่ต้องทำการค้นหาข้อมูลต่างๆ และนำข้อมูลเหล่านั้นมาคำนวณและรวมไปถึงการวาดกราฟด้วยตนเอง โปรแกรมจะเป็นเป็นผู้ทำให้ทั้งหมดเพียงแต่ผู้ใช้เลือกรูปแบบของการสรุปที่ต้องตามที่โปรแกรมกำหนดไว้

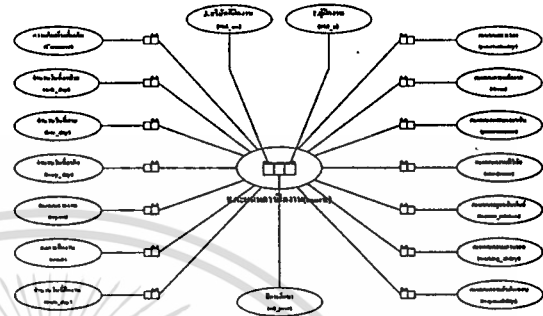
ขั้นตอนการทำงานของระบบที่พัฒนา

ขั้นตอนการทำงานทำงานในระบบใหม่จะยึดหลักการทำงานที่เคยปฏิบัติมาในระบบการทำงานเดิมและได้มีการลดขั้นตอนบางอย่างที่ซ้ำซ้อนออก แต่ขั้นตอนส่วนใหญ่จะเหมือนเดิมเพราะเป็นขั้นตอนการทำงานที่ผู้ใช้คุ้นเคยและมีขั้นตอนในการทำงานที่ดีแล้ว แต่จะแตกต่างจากระบบเดิมตรงที่มีการนำระบบจัดการฐานข้อมูลบนแนวคิดแบบโคลน/เซิร์ฟเวอร์เข้ามาประยุกต์ใช้เพื่อความสะดวกในการทำงานมากยิ่งขึ้น กล่าวคือการพัฒนาระบบขึ้นมาใหม่ในครั้งนี้เป็นการพัฒนาระบบที่มีลักษณะขั้นตอนการทำงานหลักๆ เหมือนกับการทำงานของระบบเดิม แนวความคิดของระบบฐานข้อมูลบนแนวคิดแบบโคลน/เซิร์ฟเวอร์เข้ามาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพในการทำงานสูงสุด

การออกแบบระบบฐานข้อมูล

เมื่อได้ทำการเก็บรวบรวมข้อมูลต่าง ๆ เกี่ยวกับระบบฝึกงานนักศึกษาและได้ทำการวิเคราะห์ข้อมูลต่างพร้อม

ทั้งมีการออกแบบระบบการทำงานไว้เรียบร้อยแล้วทำให้เราทราบถึงข้อมูลต่างๆ ที่จำเป็นที่จะต้องทำการจัดเก็บว่ามีข้อมูลอะไรและข้อมูลแต่ละส่วนมีความสัมพันธ์กันอย่างไร จากนั้นเราก็นำความสัมพันธ์ของข้อมูลที่ทราบมาทำการออกแบบระบบฐานข้อมูลโดยวิธีการของโนแอม (NIAM) แต่ในที่นี้จะไม่ขอลำดับถึงหลักการและวิธีการของโนแอม แต่ขอยกตัวอย่างการออกแบบระบบฐานข้อมูลด้วยโนแอมให้ดูดังรูปที่ 10



รูปที่ 10 การออกแบบฐานข้อมูลด้วย NIAM

id_co char(4)	id_a char(8)	id_year char(4)	punctuality char(1)	attendance char(1)	presence char(1)
obedience char(1)	humanity char(1)	working_ability char(1)	responsibility char(1)	salary char(1)	shift_day char(2)
late_day char(2)	busy_day char(2)	report char(1)	train_day char(2)	result char(1)	

รูปที่ 11 ตารางที่ได้จาก NIAM

สรุป

ระบบฝึกงานนักศึกษาที่พัฒนาขึ้นมาโดยอาศัยหลักการของโคลน/เซิร์ฟเวอร์ สามารถที่จะมาช่วยลดขั้นตอนและเวลาการทำงานของเจ้าหน้าที่ได้อย่างมาก ทำให้ระบบการทำงานของเจ้าหน้าที่มีประสิทธิภาพ แม้ว่าการทำงานจะทำงานกับขั้นตอนที่ซับซ้อนก็ตาม ระบบฝึกงานสามารถที่จะแก้ปัญหาต่างๆ ที่เจ้าเคยประสบมาได้เกือบทั้งหมดดังที่ได้กล่าวไปแล้วในส่วนของการแก้ปัญหาที่เกิดขึ้นในปัจจุบัน และระบบฝึกงานนักศึกษาที่พัฒนาขึ้นมาปัจจุบันทางเจ้าหน้าที่ของแผนกกิจกรรมนักศึกษา คณะวิศวกรรมศาสตร์ สจล. ได้ใช้ระบบงานนี้ในการจัดส่งนักศึกษาออกไปฝึกงานยังหน่วยงานต่างๆ แทนการปฏิบัติงานระบบเดิมที่เคยปฏิบัติมา

เอกสารอ้างอิง

- 1 รศ.ดร.ศุภมิตร จิตตะยโสธร. "ระบบฐานข้อมูลแบบแยกส่วนผู้ใช้บริการ/ผู้ให้บริการ" บทความวิชาการ
- 2 ขวัญตา อาชีวะ. หัวหน้าแผนกสวัสดิการและกิจกรรมนักศึกษา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอม

เกล้าเจ้าคุณทหารลาดกระบัง. สัมภาษณ์ 8 พฤศจิกายน
2540.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
ตัวอย่างสคริปต์ในการสร้างฐานข้อมูล ORACLE 7.X
และการสร้าง REPOSITORY ของ
Oracle Designer/2000 release 1.3.2

หลังจากที่ได้ทำการติดตั้งระบบฐานข้อมูลของ Oracle แล้วจะต้องทำการสร้างดาต้าเบส เพื่อที่จะใช้รองรับการทำงานไม่ว่าจะเป็นข้อมูลต่างๆ ที่จะต้องทำการเก็บ ซึ่งในการสร้าง Database Instance ขึ้นมานั้น สามารถที่จะทำตามขั้นตอนต่างๆ ตามลำดับ

1. จะต้องทำการสร้างไฟล์เพื่อใช้ในการเก็บค่าตัวแปรต่างๆ ของ Instance ะก่อน ซึ่งไฟล์ดังกล่าว จะต้องตั้งชื่อนี้ “INITxxxx.ORA” (xxxx คือ ชื่อ Instance ที่จะทำการสร้าง และจะต้องกำหนดค่าตัวแปร ในไฟล์ Initxxxx.Ora ดาตัวอย่างไฟล์ด้านล่างนี้

ตัวอย่างไฟล์ INTxxxx.ORA

db_name = XXXXXXXXX

ชื่อ Database Nmae จะต้องใช้ในการสร้าง DATABASE

db_files = 50

control_files = (D:\PROJECT\CTL1xxxx.ORA, C:\PROJECT\CTL2xxxx.ORA)

compatible = 7.3.0.0.0

db_file_multiblock_read_count = 8 # INITIAL

db_file_multiblock_read_count = 8 # SMALL

db_file_multiblock_read_count = 16 # MEDIUM

db_file_multiblock_read_count = 32 # LARGE

db_block_buffers = 1000 # INITIAL

db_block_buffers = 200 # SMALL

db_block_buffers = 550 # MEDIUM

db_block_buffers = 3200 # LARGE

shared_pool_size = 18000000 # INITIAL

shared_pool_size = 3500000 # SMALL

shared_pool_size = 6000000 # MEDIUM

shared_pool_size = 9000000 # LARGE

log_checkpoint_interval = 10000

processes = 50 # INITIAL

processes = 50 # SMALL

processes = 100 # MEDIUM

```

# processes = 200                                # LARGE

dml_locks = 100                                  # INITIAL
# dml_locks = 100                                # SMALL
# dml_locks = 200                                # MEDIUM
# dml_locks = 500                                # LARGE

log_buffer = 8192                                 # INITIAL
# log_buffer = 8192                               # SMALL
# log_buffer = 32768                             # MEDIUM
# log_buffer = 163840                            # LARGE

sequence_cache_entries = 10                      # INITIAL
# sequence_cache_entries = 10                    # SMALL
# sequence_cache_entries = 30                    # MEDIUM
# sequence_cache_entries = 100                   # LARGE

sequence_cache_hash_buckets = 10                 # INITIAL
# sequence_cache_hash_buckets = 10               # SMALL
# sequence_cache_hash_buckets = 89               # LARGE
# sequence_cache_hash_buckets = 23               # MEDIUM

# audit_trail = true                             # if you want auditing
# timed_statistics = true                        # if you want timed statistics
max_dump_file_size = 10240                       # limit trace file size to 5 Meg each

# log_archive_start = true                       # if you want automatic archiving

# define directories to store trace and alert files
background_dump_dest=%RDBMS73%\trace
user_dump_dest=%RDBMS73%\trace

db_block_size = 2048

snapshot_refresh_processes = 1

```

```
remote_login_passwordfile = shared
text_enable = true

#####

global_names = false
open_cursors = 200

#####

nls_language = thai
nls_date_language = thai
nls_territory = thailand
nls_sort = thai_dictionary
nls_date_format = DD-MON-YYYY
```

2. ให้ทำการตรวจสอบตัวแปรต่างของระบบดังนี้
 - ให้ไปที่ Menu Run.... แล้วพิมพ์คำสั่ง “REGEDIT”
แล้วทำการเลือก HKEY_LOCAL_MACHINE → Software → Oracle
ตรวจสอบดูว่ามีตัวแปรชื่อ “ORACLE_SID” หรือไม่ ถ้ามีให้ทำการแก้ไขค่า
ของตัวแปรโดยใส่ชื่อ INSTANCE ที่จะสร้างลงไปแทน แต่ถ้าไม่มีให้ทำการ
สร้างขึ้นมาก็ใส่ชื่อ INSTANCE ลงไป
3. ให้ทำการตรวจสอบ ว่าตัว OracleTNSlistener ได้ทำการ Starup หรือยัง โดยเข้าไปที่
Control Panel → Services ถ้ายังไม่ Start ให้ทำการ Start
4. ออกมาที่ Dos แล้วพิมพ์คำสั่งดังนี้ ข้อควรระวังคือ ห้ามกด Enter ก่อนที่จะหมด
ประโยค เพราะถือว่าเป็นคำสั่งเดียวและห้ามใส่คั่นผิด
“Oradim73 -new -sid <sid_name> -intpwd <password> -startmode auto
-pfile <initxxxx.ora>”
แต่ถ้าใส่คั่นผิดให้ทำตามคำสั่งดังต่อไปนี้แล้วค่อยทำคำสั่งด้านบนอีกครั้ง
“Oradim73 -delete -sid <sid_name>”
5. ออกไปที่ Dos แล้วพิมพ์คำสั่ง SVRMGR23 เพื่อเข้าสู่หน้าจอของ Server Manager
หลังจากเรียกคำสั่งนี้แล้วหน้าจอจะขึ้น Prompt ของ Server Manager คือ
SVRMGR> แล้วให้ทำตามคำสั่งด้านล่างลำดับ
 - connect internal
 - startup nomount pfile=d:\project\initxxxx.ora
 - create database XXXXXXXXX

controlfile reuse

logfile 'd:\project\log1xxxx.ora' size 200K reuse,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ```

'd:\project\log2xxxx.ora' size 200K reuse
datafile 'd:\project\sys1xxxx.ora' size 80M reuse autoextend on
next 20M maxsize 200M
character set TH8TISASCII;
- create rollback segment rb_temp storage (initial 50K next 100K
optimal 150K);
- create tablespace user_data
datafile 'd:\project\usr1xxxx.ora' size 10M reuse autoextend on
next 5M maxsize 150M;
- create tablespace rollback_data
datafile 'd:\project\rbs1xxxx.ora' size 15M reuse autoextend on
next 5M maxsize 150M;
- create tablespace temporary_data
datafile 'd:\project\tmp1xxxx.ora' size 15M reuse autoextend on
next 5M maxsize 150M;
- alter rollback segment rb_temp online;
- alter user system temporary tablespace temporary_data;
- alter user system default tablespace user_data;
- create public rollback segment rb1 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb2 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb3 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb4 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb5 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb6 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb7 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb8 storage(initial 50K next 50K)
tablespace rollback_data;
- create public rollback segment rb9 storage(initial 50K next 50K)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ```

tablespace rollback_data;
- create public rollback segment rb10 storage(initial 50K next 50K)
  tablespace rollback_data;
- create public rollback segment rb11 storage(initial 50K next 50K)
  tablespace rollback_data;
- create public rollback segment rb12 storage(initial 50K next 50K)
  tablespace rollback_data;
- create public rollback segment rb13 storage(initial 50K next 50K)
  tablespace rollback_data;
- create public rollback segment rb14 storage(initial 50K next 50K)
  tablespace rollback_data;
- create public rollback segment rb15 storage(initial 50K next 50K)
  tablespace rollback_data;
- create public rollback segment rb16 storage(initial 50K next 50K)
  tablespace rollback_data;
- @<ORACLE_HOME>\rdbms73\admin\catalog.sql
- @<ORACLE_HOME>\rdbms73\admin\catalog6.sql
- @<ORACLE_HOME>\rdbms73\admin\catproc.sql
- grant execute on dbms_lock to public;
- grant execute on dbms_pipe to public;
- grant execute on dbms_alert to public;
- connect system/manager
- @<ORACLE_HOME>\rdbms73\admin\catdbsyn.sql
- @<ORACLE_HOME>\dbs\publd.sql
- connect internal
- @<ORACLE_HOME>\rdbms73\admin\catrep.sql
- grant execute on dbms_defer to public;
- alter package dbms_snapshot compile body;
- connect internal
- shutdown

```

6. ให้ทำการ Restart เครื่องใหม่

7. เป็นขั้นตอนในการสร้าง Tablespace เพื่อใช้ในการทำงานของ Oracle Designer/2000 เพื่อที่จะเอาไว้เก็บ Repository โดยให้เข้าไปที่ SQL*PLUS โดยใช้ชื่อ user คือ “sys/change_on_install” แล้วทำตามขั้นตอนต่อไปนี้

- create tablespace des2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
datafile 'd:\project\des2.ora' size 40M autoextend on
next 5M maxsize 100M;
```

- create tablespace des2_I

```
datafile 'd:\project\des2_I.ora' size 15M autoextend on
next 5M maxsize 100M;
```

- create user des2 identified by des2

```
default tablespace des2
```

```
temporary tablespace des2;
```

- @<ORACLE_HOME>\repadm10\utl\ckrrole.sql คำสั่งจะเป็นการ ROLE ชื่อ “DES2000_OWNER”

- grant connect, resource, des2000_owner to des2;

- grant execute on sys.dbms_pipe to des2;

- grant execute on sys.dbms_lock to des2;

8. ออกจาก SQL*PLUS

9. ให้ทำการสร้าง Repository โดยเข้าไปใน Designer/2000 โดยใช้ user คือ “des2/des2” แล้วเลือก Reository Administration Utility → Install

ภาคผนวก ค
รายงานรายละเอียดของ Function ต่าง ๆ ว่ามีการใช้
Entity อะไรบ้าง

Function : AFT_REP_0

Definition: รายชื่อนักศึกษาฝึกงานทั้งหมด

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
DEPT_ROOM		Y				
REP_INFOR		Y				
STUDENT		Y				
TRAIN_INFOR		Y				

Function : AFT_REP_1

Definition: จดหมายปฏิเสธ

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : AFT_REP_2

Definition: สรุปรายละเอียดบริษัท (เงื่อนไขและรายชื่อน.ศ.)

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
CONDITION		Y				
DEPARTMENT		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : AFT_REP_3

Definition: สรุปการส่งรายงาน/ส่งใบประเมิน/สถิติการสนและคะแนน

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
DEPT_ROOM		Y				
INF_MARKING		Y				
REP_INFOR		Y				
STUDENT		Y				
STUDENT_MARKING		Y				
TRAIN_INFOR		Y				

Function : AFT_REP_4

Definition: สรุปจำนวนนักศึกษาที่บริษัทรับ/คัดส่ง

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
CONDITION		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : AFT_REP_5

Definition: สรุปบริษัทปฏิเสธรับนักศึกษา

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
COMPANY		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : AFT_REP_6

Definition: หนังสือขออนุมัติ

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
COMPANY		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : APPOINT

Definition: จัดทำหนังสือแต่งตั้ง

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
DEPARTMENT		Y				
TEACHER		Y				

Function : ASSESS_ALL

Definition: ประมวลผลรวม

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
COMPANY		Y				
DEPARTMENT		Y				
DEPT_ROOM		Y				
INF_MARKING		Y				
REP_INFOR		Y				
STUDENT		Y				
STUDENT_MARKING		Y				
TRAIN_INFOR		Y				

Function : ASSESS_DEP

Definition: ประเมินผลแยกตามภาค

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
DEPARTMENT		Y				
DEPT_ROOM		Y				
INF_MARKING		Y				
REP_INFOR		Y				
STUDENT		Y				
STUDENT_MARKING		Y				
TRAIN_INFOR		Y				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function : BEF_REP_0

Definition: ขออนุญาตให้นักศึกษาเข้าฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : BEF_REP_1

Definition: รายละเอียดการฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : BEF_REP_2

Definition: ส่งนักศึกษารายงานตัวเข้าฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
DEPARTMENT		Y				
DEPT_ROOM		Y				
REP_INFOR		Y				
STUDENT		Y				
TRAIN_INFOR		Y				

Function : BEF_REP_3

Definition: รายชื่อบริษัทขอรับ

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
REP_INFOR		Y				
TRAIN_INFOR		Y				

Function : BEF_REP_5

Definition: ขอยืนยันการขอรับนักศึกษาเข้าฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
DEPARTMENT		Y				
DEPT_ROOM		Y				
REP_INFOR		Y				
STUDENT		Y				
TRAIN_INFOR		Y				

Function : BEF_REP_7

Definition: จัดส่งใบประเมินผลของนักศึกษา

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
DEPARTMENT		Y				
DEPT_ROOM		Y				
REP_INFOR		Y				
STUDENT		Y				
TRAIN_INFOR		Y				

Function : COM_DETAIL

Definition: รายละเอียดบริษัท

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY	Y			Y		
PROVINCE		Y				

Function : DATA_STUD

Definition: จัดเตรียมข้อมูลนักศึกษา

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
DEPT_ROOM		Y				
STUDENT	Y	Y	Y	Y		

Function : EXP_DATE

Definition: พิมพ์คหหมายแจ้งการหมดวาระ

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
TEACHER		Y				

Function : INIT_COND

Definition: รายละเอียดเงื่อนไข

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY	Y	Y	Y	Y	Y	
CONDITION	Y	Y	Y	Y	Y	
DEPARTMENT		Y				
TRAIN_INFOR		Y				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function : INIT_DEPT

Definition: ข้อมูลภาควิชา/ห้อง

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT	Y	Y	Y	Y		
DEPT_ROOM	Y	Y	Y	Y		

Function : INIT_MARK

Definition: บันทึกผลการให้คะแนน

Entity	Create	Retrieve	Update	Delete	Archive	Other
INF_MARKING	Y	Y	Y	Y		

Function : INIT_PROV

Definition: ข้อมูลจังหวัด

Entity	Create	Retrieve	Update	Delete	Archive	Other
PROVINCE	Y	Y	Y	Y		

Function : INIT_TRAIN

Definition: ข้อมูลทั่วไป/รายชื่อเอกสาร

Entity	Create	Retrieve	Update	Delete	Archive	Other
REP_INFOR	Y	Y	Y	Y		
TRAIN_INFOR	Y	Y	Y	Y		

Function : INPUT_MARK

Definition: บันทึกคะแนนการฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
DEPT_ROOM		Y				
INF_MARKING		Y				
STUDENT		Y	Y			
STUDENT_MARKING		Y	Y			

Function : NOTICE_TRA

Definition: จัดทำประกาศผลการฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
DEPT_ROOM		Y				
STUDENT		Y				
TRAIN_INFOR		Y				

เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ฝึกสอน อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function : PANEL

Definition: เครื่องมือคัดกรองผลการ

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
TEACHER	Y	Y	Y	Y		

Function : PRN_PANEL

Definition: หนังสือเวียนคณะกรรมการ

Entity	Create	Retrieve	Update	Delete	Archive	Other
DEPARTMENT		Y				
TEACHER		Y				

Function : RECEL_TRA

Definition: รวบรวมคะแนนการฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
INF_MARKING		Y				
REP_INFOR		Y				
STUDENT		Y				
STUDENT_MARKING		Y				
TRAIN_INFOR		Y				

Function : REGIST

Definition: ลงชื่อฝึกงาน

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
CONDITION		Y	Y			
DEPARTMENT		Y				
DEPT_ROOM		Y				
INF_MARKING		Y				
STUDENT		Y	Y			
STUDENT_MARKING	Y	Y	Y	Y		

Function : REPORT_1

Definition: พิมพ์รายชื่อนักศึกษา (แยกตามหน่วยงาน)

Entity	Create	Retrieve	Update	Delete	Archive	Other
COMPANY		Y				
DEPARTMENT		Y				
DEPT_ROOM		Y				
STUDENT		Y				

Function : REPORT_2

Definition: พิมพ์รายชื่อนักศึกษา (แยกตามภาควิชา)

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
COMPANY		Y				
DEPARTMENT		Y				
DEPT_ROOM		Y				
STUDENT		Y				

Function : SUCCES_TRA

Definition: จัดทำสรุปผลการฝึกงาน

<u>Entity</u>	<u>Create</u>	<u>Retrieve</u>	<u>Update</u>	<u>Delete</u>	<u>Archive</u>	<u>Other</u>
DEPARTMENT		Y				
DEPT_ROOM		Y				
REP_INFOR		Y				
STUDENT		Y				
TRAIN_INFOR		Y				

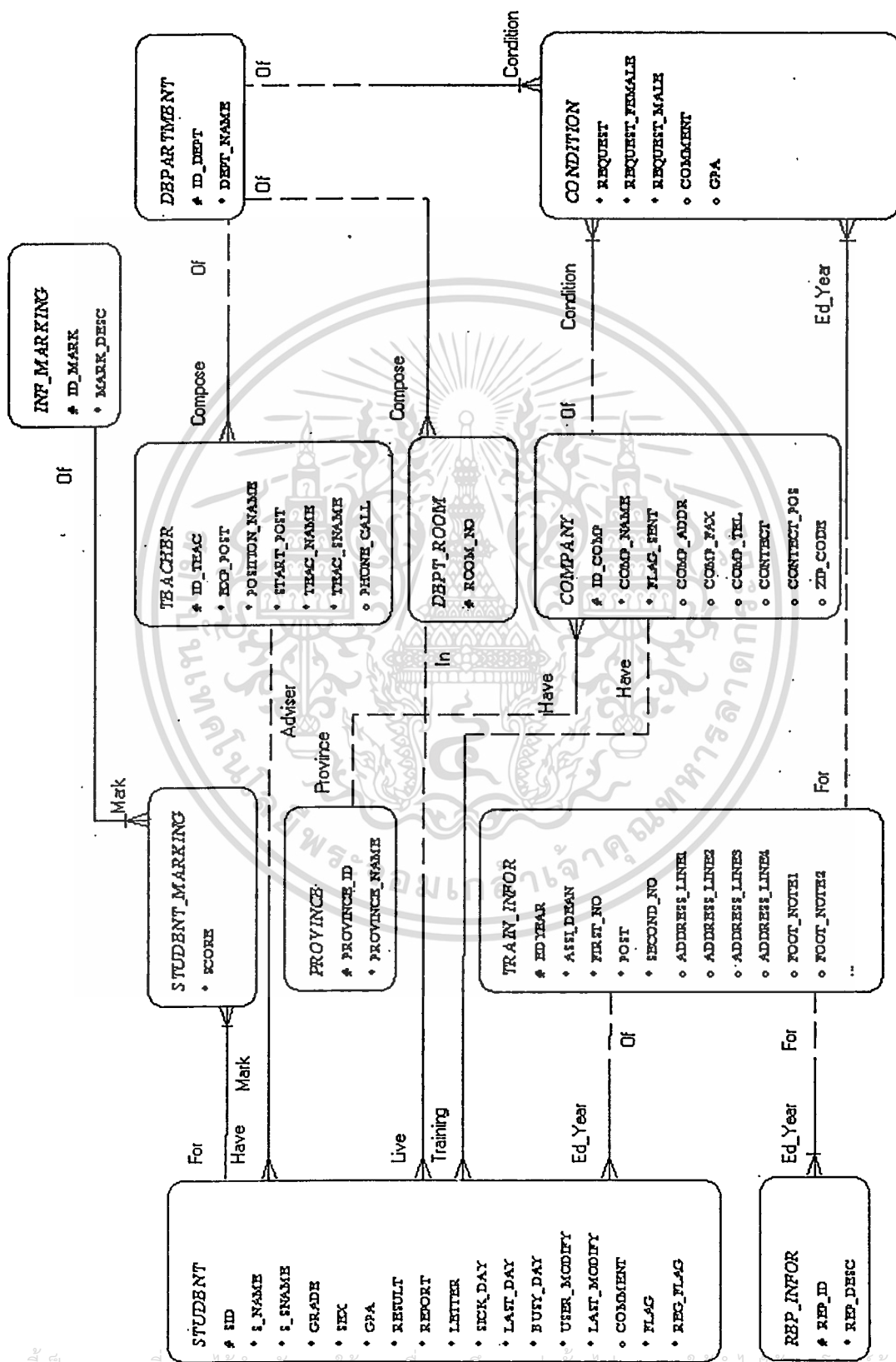
ภาคผนวก ง
รายละเอียดของ Process Modeller



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

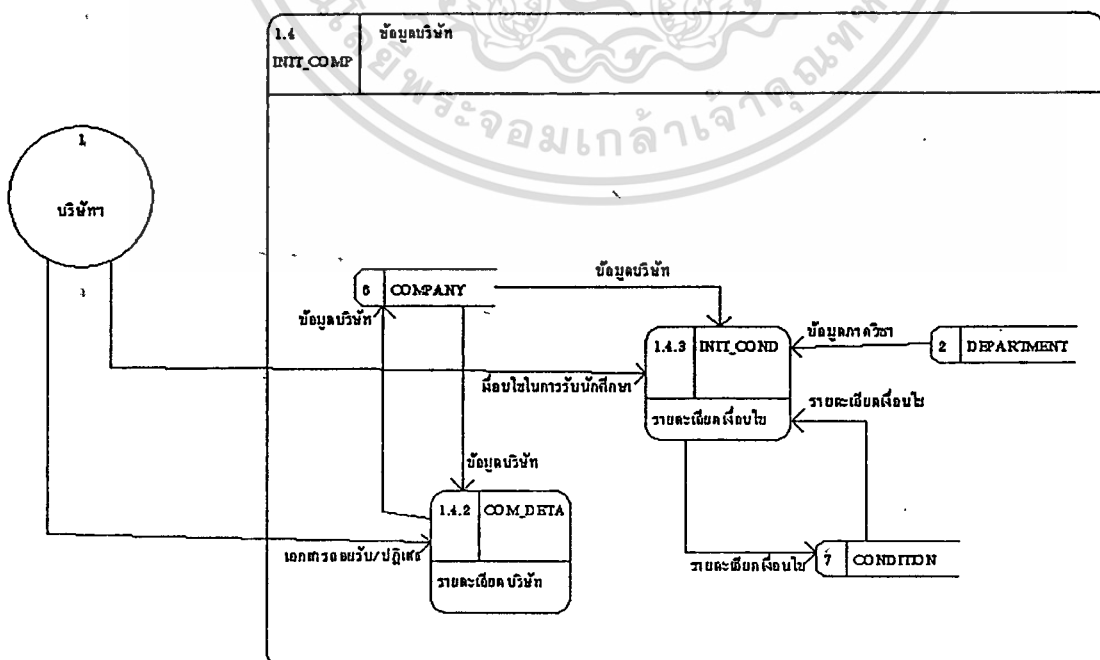
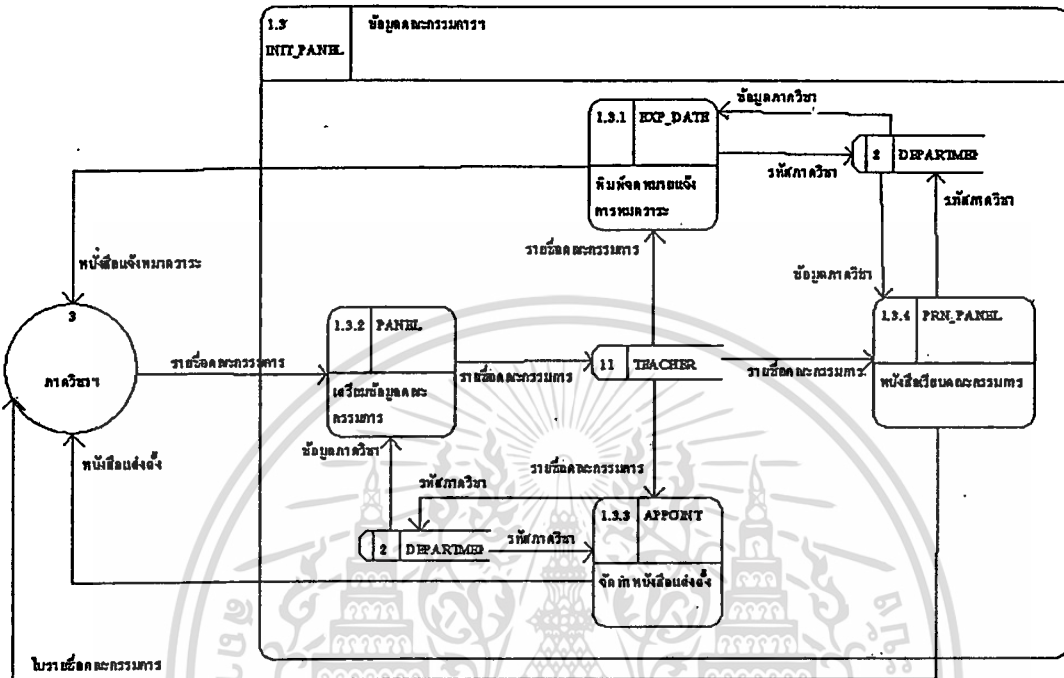
ภาคผนวก จ
รายละเอียดของ Entity Relationship Model

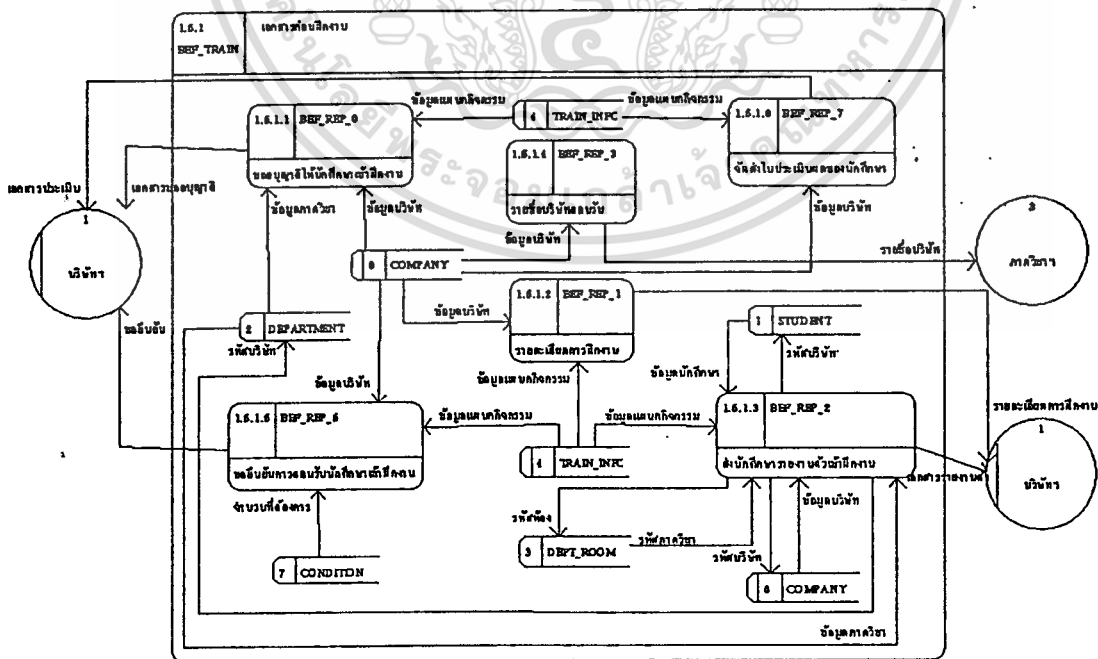
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ
รายละเอียดของ Dataflow Diagrammer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





ประวัติ

ชื่อ นายสมเกียรติ บริบาลประสิทธิ์
วัน-เดือน-ปีเกิด 10 พฤษภาคม พ.ศ. 2514
สัญชาติ ไทย
การศึกษา จบปริญญาตรี สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยกรุงเทพ
หน้าที่การงาน เป็นที่ปรึกษาและออกแบบระบบฐานข้อมูลให้กับ บริษัท I&J (Thailand) จำกัด เป็นอาจารย์พิเศษมหาวิทยาลัยกรุงเทพ, เป็นวิทยากรพิเศษให้กับบริษัท UCOM และเคยเป็นเจ้าหน้าที่แผนกสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

