

การพัฒนาระบบสารสนเทศโดยใช้ Oracle Designer6i CASE Tools ส่วนที่ 3
(กรณีศึกษา: ระบบทะเบียนนักศึกษา)

Development of an Information System using Oracle Designer6i CASE Tools Part III
(Case Study: Student Registration System)



วัน เดือน ปี.....	25	ต.ค.	2550
เลขทะเบียน.....	02058		
เลขเรียกหนังสือ.....	วท.	ก631ก	2546
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."			

รายงานนี้เป็นส่วนหนึ่งของวิชา โครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบสารสนเทศโดยใช้ Oracle Designer6i CASE Tools ส่วนที่ 3 (กรณีศึกษา: ระบบทะเบียนนักศึกษา)
นักศึกษา	นางสาวกิงกาญจน์ วิไลคารา
อาจารย์ที่ปรึกษา	ผศ.ดร.ประจวบ วานิชชัชวาล
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ในปัจจุบันการพัฒนาระบบสารสนเทศได้มีการนำซอฟต์แวร์ที่ช่วยในการออกแบบ (CASE Tools) มาใช้มากขึ้น เนื่องจากจะช่วยลดระยะเวลา และความซับซ้อนในการออกแบบ ดังนั้น การศึกษาเกี่ยวกับรายละเอียดและการทำงานของ CASE Tools จึงเป็นสิ่งที่น่าสนใจ โดยได้มีการนำ Oracle Designer6i ซึ่งเป็นหนึ่งใน CASE Tools ที่ได้รับความนิยมและสมบูรณ์มากที่สุดอันหนึ่งมา เป็นตัวศึกษาการทำงานของ CASE Tools และนำระบบทะเบียนนักศึกษามาเป็นกรณีศึกษา เพื่อใช้ในการออกแบบ ซึ่งจะช่วยให้ง่ายต่อการศึกษาการทำงานและรายละเอียดยิ่งขึ้น และทำให้ง่ายต่อการศึกษาการทำงานของ CASE Tools อันอื่นยิ่งขึ้น

Title Development of an Information System using Oracle Designer6i
CASE Tools Part III (Case Study: Student Registration System)

Student Miss Kingkan Vilaidaraga

Advisor Asst. Prof. Prachuab Vanitchatchavan, Ph.D.

Level of Study Master of Science in Information Technology

Major Information Science

Academic Year 2003

ABSTRACT

Nowdays, technology has been advancing which introduction new software called CASE Tools to be use more commonly. This software helps in reducing the time and complication in designing. Therefore, details and functions of the CASE Tools has been an interesting topic in the study field. Oracle Designer6i, one of the CASE Tools, is so far the most complete function and becoming very popular in education of the CASE Tools. The student registration system is being used to do the case study in order to make leaning about details and functions of the CASE Tools more easier and understandable for the other CASE Tools.

กิตติกรรมประกาศ

โครงการพัฒนาระบบสารสนเทศโดยใช้ OracleDesigner6i CASE Tools ส่วนที่ 3 (กรณีศึกษา: ระบบทะเบียนนักศึกษา) สำเร็จลงด้วยดีอันเนื่องมาจากความร่วมมือ และช่วยเหลือจากบุคคลหลายฝ่ายด้วยกัน โดยเฉพาะอย่างยิ่ง ผศ.ดร.ประจวบ วานิชชัชวาล อาจารย์ที่ปรึกษา ที่คอยให้คำแนะนำและคำปรึกษา ตลอดจนการทำโครงการพัฒนาระบบงานนี้ ซึ่งต้องกราบขอบพระคุณอาจารย์เป็นอย่างมาก

รวมไปถึงนางสาวราพร ถิมวัฒนาเลิศ และนายสมบูรณ์ พวงแก้ว ผู้ร่วมงานในโครงการนี้ ในส่วนที่ 1 และส่วนที่ 2 จนทำให้โครงการนี้สำเร็จไปได้ด้วยดี

ขอขอบคุณ บิดา มารดา ของข้าพเจ้า ที่ให้การอบรมสั่งสอน และแนะนำในสิ่งที่ดีให้ข้าพเจ้าตลอดมา ตลอดจนเพื่อน ๆ ที่คอยให้คำแนะนำและให้กำลังใจ

นางสาวกัญญาญจน์ วิไลคารกา

ผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1. บทนำ.....	1
1.1 ความสำคัญและที่มา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 เครื่องมือที่ใช้ในการพัฒนา.....	2
1.5 ขั้นตอนและวิธีการดำเนินงาน.....	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	4
2. ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ระบบจัดการฐานข้อมูล.....	5
2.2 ระบบจัดการฐานข้อมูลเชิงสัมพันธ์.....	7
2.3 ระบบจัดการฐานข้อมูลเชิงวัตถุ.....	8
2.4 Oracle8i.....	9
2.5 CASE Tools.....	11
3. Oracle Designer6i CASE Tools.....	14
3.1 ส่วนประกอบของ Oracle Designer6i.....	14
3.2 การออกแบบ Process Model.....	19
3.3 การออกแบบ Dataflow Diagram (DFD).....	21
3.4 การออกแบบ Function Hierarchy Diagram (FHD).....	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
3.5 การออกแบบ Entity Relationship Diagram (ERD).....	22
3.6 การสร้าง Database.....	23
3.7 การสร้าง Application.....	25
4. การวิเคราะห์และออกแบบระบบงาน	28
4.1 ระบบทะเบียนนักศึกษา.....	28
4.2 Dataflow Diagram ของระบบทะเบียนนักศึกษา.....	32
4.3 แบบจำลองแสดงความสัมพันธ์ของข้อมูล.....	36
4.4 รายละเอียดของข้อมูลที่จัดเก็บในระบบ.....	38
5. การพัฒนาระบบงาน	48
5.1 การพัฒนาระบบงานส่วนของฐานข้อมูล.....	48
5.2 การพัฒนาระบบงานส่วนของโปรแกรมการใช้งาน.....	54
6. สรุปและข้อเสนอแนะ.....	68
6.1 ข้อดีของการนำ Oracle Designer6i มาพัฒนาระบบ.....	68
6.2 ข้อจำกัดของการนำ Oracle Designer6i มาพัฒนาระบบ.....	69
6.3 ข้อเสนอแนะ.....	69
บรรณานุกรม.....	70

สารบัญตาราง

ตารางที่	หน้า
3.1 Response ที่มีผลต่อ Application Module.....	26
4.1 ตาราง FACULTY	38
4.2 ตาราง DEPARTMENT	38
4.3 ตาราง STUDENT	38
4.4 ตาราง STUDENT_TYPE	39
4.5 ตาราง STUDENT_INFO	40
4.6 ตาราง SUB_DISTRICT	40
4.7 ตาราง DISTRICT	41
4.8 ตาราง PROVINCE	41
4.9 ตาราง ST_DOCUMENT	41
4.10 ตาราง DOCUMENT_TYPE	42
4.11 ตาราง INSTRUCTOR	42
4.12 ตาราง CURRICULUM	42
4.13 ตาราง COURSE_OFFERED	43
4.14 ตาราง COURSE_SCHEDULE	43
4.15 ตาราง SEMESTER	44
4.16 ตาราง REGISTRATION_RECORD	45
4.17 ตาราง FEE.....	45
4.18 ตาราง FEE_CATEGORY	46
4.19 ตาราง RECEIPT_HEADER.....	46
4.20 ตาราง RECEIPT_DETAIL1.....	46
4.21 ตาราง RECEIPT_DETAIL2.....	47
4.22 ตาราง RECEIPT_DETAIL3	47
4.23 ตาราง OTHER_FEE	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1	โครงสร้างระบบจัดการฐานข้อมูล..... 5
2.2	การจัดเก็บข้อมูลแบบ File Approach..... 6
2.3	การจัดเก็บข้อมูลแบบ Database Approach..... 6
2.4	ขั้นตอนการพัฒนาระบบแบบ SDLC และ CASE Tools..... 13
3.1	หน้าจอแรกของ Oracle Designer6i..... 15
3.2	สัญลักษณ์ที่ใช้ใน Process Modeler..... 20
3.3	สัญลักษณ์ที่ใช้ใน Dataflow Diagram..... 21
3.4	สัญลักษณ์ที่ใช้ใน Function Hierarchy Diagram..... 22
3.5	สัญลักษณ์ที่ใช้ใน Entity Relationship Diagram..... 23
3.6	หน้าจอ Database Design Transformer (DDT)..... 23
3.7	การกำหนดค่า Display Type..... 26
3.8	หน้าจอ Application Design Transformer (ADT)..... 27
4.1	Context Diagram ของระบบทะเบียนนักศึกษา..... 32
4.2	Dataflow Diagram ระดับที่ 1 ของระบบทะเบียนนักศึกษาเฉพาะส่วนของข้อมูล นักศึกษา การเงิน และรายงานต่าง ๆ..... 33
4.3	Dataflow Diagram ระดับที่ 2 ของระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา 34
4.4	Dataflow Diagram ระดับที่ 2 ของระบบทะเบียนนักศึกษาในส่วนของการเงิน..... 35
4.5	Dataflow Diagram ระดับที่ 2 ของระบบทะเบียนนักศึกษาในส่วนของการออก รายงานต่าง ๆ..... 36
4.6	ER-Diagram ของระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา การเงิน และ การออกรายงานต่าง ๆ..... 37
5.1	Entity Relationship Diagram ของระบบทะเบียนนักศึกษา..... 49
5.2	Database Design Transformer..... 50
5.3	Design Editor..... 51
5.4	Server Model Diagram 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.5	Function Database from Server Model..... 53
5.6	Script File ที่ได้จาก Function Generate Database from Server Model..... 53
5.7	Dataflow Diagram ระดับที่ 1 ของระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา การเงิน และการออกรายงานต่าง ๆ..... 55
5.8	Dataflow Diagram ระดับที่ 2 ของข้อมูลนักศึกษา..... 56
5.9	Dataflow Diagram ระดับที่ 2 ของการเงิน..... 57
5.10	Dataflow Diagram ระดับที่ 2 ของการออกรายงานต่าง ๆ..... 58
5.11	Function Hierarchy Diagram ของระบบทะเบียนนักศึกษา..... 59
5.12	การกำหนด Response Type..... 60
5.13	การกำหนดการใช้ Entity ต่าง ๆ ให้กับ Function..... 61
5.14	การกำหนดการใช้ Attribute ต่าง ๆ ให้กับ Function 61
5.15	การกำหนด Display Type..... 62
5.16	Application Design Transformer..... 63
5.17	การเปลี่ยนคุณสมบัติ Candidate..... 63
5.18	โครงสร้าง Module..... 65
5.19	โครงสร้างของ LOV..... 66
5.20	Generator Preference..... 66
5.21	สร้าง Form โดยใช้ Generate Form..... 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

กระบวนการในการพัฒนาระบบสารสนเทศนั้นมีอยู่หลายขั้นตอน ซึ่งในแต่ละขั้นตอนล้วนแต่มีความสำคัญ ถ้าเกิดปัญหาหรือข้อผิดพลาดขึ้นมาจะเกิดการเสียหายเป็นอย่างมากทั้งในเรื่องของเวลา และค่าใช้จ่าย ดังนั้นจึงมีการทำซอฟต์แวร์ที่ใช้ในการลดข้อผิดพลาด และระยะเวลาในการพัฒนาระบบขึ้น ซึ่งเรียกว่า CASE Tools และในปัจจุบันนี้ CASE Tools ก็ได้รับความนิยมในการนำไปใช้พัฒนาระบบมากขึ้นเรื่อย ๆ ทำให้การศึกษางานเกี่ยวกับ CASE Tools จึงมีความน่าสนใจเป็นอย่างยิ่ง

1.1 ความเป็นมาสำคัญและที่มา

ในปัจจุบันเทคโนโลยีทางด้านซอฟต์แวร์และฮาร์ดแวร์นั้นมีการเปลี่ยนแปลงตลอดเวลาและมีการเปลี่ยนแปลงที่รวดเร็วเป็นอย่างมากจากในอดีตที่มีการเขียนโปรแกรมด้วยตัวเองเนื่องจากเป็นโปรแกรมที่ไม่ซับซ้อนมากนัก จนกระทั่งคอมพิวเตอร์และเทคโนโลยีต่าง ๆ เข้ามามีบทบาทภายในธุรกิจและชีวิตประจำวัน ทำให้ซอฟต์แวร์ที่ใช้งานมีความซับซ้อนมากขึ้น ทำให้มีการพัฒนาเครื่องมือที่ช่วยในการจัดการ และออกแบบระบบขึ้นมา ซึ่งเรียกว่า “Computer-Aided Software Engineering (CASE Tools)” ซึ่งได้เริ่มเข้ามามีบทบาทสำคัญเพิ่มขึ้นเรื่อย ๆ จนเป็นที่ยอมรับว่าเป็นเทคนิคที่ใช้ในการพัฒนาระบบงานคอมพิวเตอร์ที่ดีที่สุดในปัจจุบัน และน่าจะเป็นเครื่องมือที่ใช้กันอย่างแพร่หลายมากขึ้นเรื่อย ๆ ในอนาคต เนื่องจาก CASE Tools เป็นเครื่องมือที่ช่วยให้สามารถพัฒนาระบบงานที่ซับซ้อนได้ง่ายและรวดเร็วขึ้น สะดวกต่อการปรับปรุงและแก้ไข อีกทั้งยังช่วยลดข้อผิดพลาดที่จะเกิดขึ้นในระหว่างขั้นตอนการพัฒนาอีกด้วย

โครงการนี้ได้นำเอา CASE Tools ที่พัฒนาโดยบริษัท ออราเคิล คอร์ปอเรชั่น ที่ชื่อว่า Oracle Designer6i มาใช้ในการออกแบบและพัฒนาระบบงานทะเบียนนักศึกษา ซึ่งเป็นระบบงานที่มีความซับซ้อนจากที่ Oracle Designer6i สนับสนุนการทำงานแบบ Spiral ช่วยลดงานที่ซับซ้อนลงได้ อีกทั้งขอบเขตของระบบงานพัฒนาที่ขนาดใหญ่มาก มีหน่วยงานที่เกี่ยวข้องและมีข้อมูลภายในจำนวนมาก ทำให้การนำเอา CASE Tools มาช่วยในการพัฒนาระบบทะเบียนนักศึกษา และใช้ในการจัดการข้อมูลและจัดทำเอกสารจึงเป็นสิ่งที่ยังจำเป็นอยู่ ทั้ง Oracle Designer6i นี้ยังมีส่วนที่เกี่ยวกับการติดต่อกับผู้ใช้ (User Interface) ที่ง่ายต่อการใช้งาน โดย Oracle Designer6i นี้จะช่วยในการจัดการและออกแบบโครงสร้างโดยรวมของระบบ ช่วยจัดการฐานข้อมูลของระบบและช่วยใน

การวิเคราะห์กิจกรรมต่าง ๆ ของระบบ แล้วยังช่วยในการจัดทำเอกสารคู่มือและรายงานต่าง ๆ ในทุกขั้นตอนของการพัฒนาระบบ และช่วยในการเขียน Diagram ต่าง ๆ ที่ใช้ในการออกแบบระบบอีกด้วย

1.2 วัตถุประสงค์ของโครงการ

ในการศึกษาและพัฒนาโครงการนี้มีวัตถุประสงค์ของโครงการดังนี้

- 1.2.1 ศึกษากระบวนการในการทำงานของ CASE Tools เพื่อใช้เป็นเครื่องมือในการออกแบบและพัฒนาระบบงาน
- 1.2.2 นำระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา และในส่วนของการเงิน มาเป็นกรณีศึกษาในการออกแบบและพัฒนาระบบงานโดยใช้ Oracle designer6i CASE Tools
- 1.2.3 เปรียบเทียบข้อดีและข้อเสียระหว่างการพัฒนาแบบที่ไม่ใช้ CASE Tools กับการพัฒนาแบบที่ใช้ CASE Tools

1.3 ขอบเขตของโครงการ

โครงการพัฒนาระบบงานนี้จะทำการศึกษากระบวนการทำงาน ขั้นตอนและวิธีการใช้งานด้านต่าง ๆ โดยใช้ CASE Tools ในการพัฒนาระบบงาน ซึ่งจะมีการนำ Oracle Designer6i CASE Tools มาใช้ในการพัฒนาและวิเคราะห์ระบบโดยมีการนำเอาระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษาและส่วนของการเงินมาเป็นกรณีศึกษา เพื่อให้ง่ายต่อการศึกษาการทำงานของ CASE Tools ได้ดียิ่งขึ้น

1.4 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาโครงการนี้ได้มีการนำซอฟต์แวร์ต่าง ๆ มาใช้เพื่อช่วยในการพัฒนาระบบโดยมีซอฟต์แวร์ต่าง ๆ ที่ใช้ดังนี้

- 1.4.1 Oracle Designer6i เป็น CASE Tools สำหรับใช้ในการศึกษาการทำงานในโครงการเกี่ยวกับการวิเคราะห์ ออกแบบและพัฒนาระบบ
- 1.4.2 Oracle8i Enterprise Edition Release 8.1.6.0.0 ซึ่งเป็นระบบจัดการฐานข้อมูลของข้อมูลที่สร้างขึ้นมาจาก Oracle Designer
- 1.4.3 Oracle Forms6i เป็นซอฟต์แวร์ที่ใช้ Compile และทำการ Run โปรแกรมที่สร้างขึ้นมาจาก Oracle Designer

1.5 ขั้นตอนและวิธีการดำเนินงาน

ในการพัฒนาระบบทะเบียนนักศึกษาที่นำมาเป็นกรณีศึกษาในโครงการนี้ได้มีการจัดขั้นตอนและวิธีดำเนินงานดังนี้

- 1.5.1 ทำการศึกษาข้อมูลเกี่ยวกับหลักการ วิธีใช้งาน และทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาระบบ เช่น CASE Tools โดยเฉพาะอย่างยิ่ง Oracle Designer6i CASE Tools, ภาษา PL/SQL, รูปแบบและลักษณะของฐานข้อมูล
- 1.5.2 ศึกษารายละเอียดเกี่ยวกับระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา และส่วนของการเงิน เพื่อใช้ในการออกแบบระบบ
- 1.5.3 วิเคราะห์และออกแบบระบบในส่วนของโครงสร้างข้อมูล ในรูปแบบของ Entity Relationship Diagram (ERD) โดยใช้เครื่องมือที่ชื่อว่า Entity Relationship Diagrammer ที่อยู่ใน Oracle Designer6i
- 1.5.4 วิเคราะห์และออกแบบโครงสร้างข้อมูลแบบ Dataflow Diagram (DFD) โดยใช้ Dataflow Diagrammer ที่อยู่ใน Oracle Designer6i
- 1.5.5 ทำการแปลงโครงสร้างแบบ Entity Relationship Diagram ให้อยู่ในรูปของ Server Model Diagram โดยใช้เครื่องมือ Database Design Transformer
- 1.5.6 ทำการสร้างฐานข้อมูลที่ต้องการ ซึ่งได้มาจากใน Server Model Diagram โดยใช้ฟังก์ชันที่ชื่อ Generate Database from Server Model Diagram ใน Design Editor
- 1.5.7 ทำการแปลงโครงสร้าง Dataflow Diagram ที่ได้ออกแบบไว้แล้วให้อยู่ในรูปของโครงสร้างแบบ Function Hierarchy Diagram (FHD) โดยใช้เครื่องมือที่ชื่อ Function Hierarchy Diagrammer
- 1.5.8 กำหนด Entity และ Attribute ที่จะต้องใช้ในแต่ละ Function ในระบบ
- 1.5.9 ทำการแปลงโครงสร้างแบบ Function Hierarchy Diagram ให้อยู่ในรูปของ Module โดยใช้เครื่องมือ Application Design Transformer
- 1.5.10 ทำการสร้างโปรแกรมให้อยู่ในรูปของ Form ที่ต้องการโดยใช้ Oracle Form6i โดยใช้ฟังก์ชัน Generate ของ Oracle Designer6i

1.6 ประโยชน์ที่คาดว่าจะได้รับ

การพัฒนาการพัฒนาระบบโดยใช้ CASE Tools ในการพัฒนาภายในโครงการนี้ คาดว่าจะได้รับประโยชน์ต่าง ๆ ดังนี้

- 1.6.1 ทำให้ทราบถึงรายละเอียดของคุณสมบัติ ความสามารถ และวิธีใช้งานของ Oracle Designer6i CASE Tools
- 1.6.2 ทำให้สามารถทำการพัฒนาระบบ ได้อย่างมีขั้นตอนการทำงานที่เป็นไปในลักษณะที่มีมาตรฐานเดียวกัน
- 1.6.3 เพื่อเป็นพื้นฐานในการทำงานโดยใช้ CASE Tools ในการพัฒนาระบบอื่นๆ ต่อไป



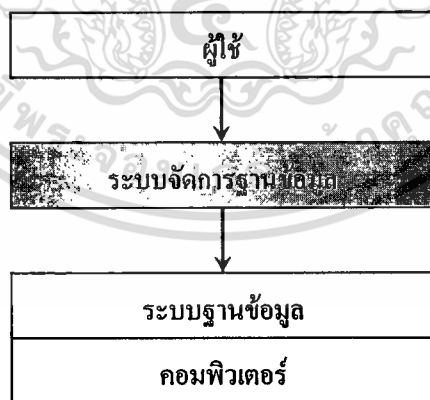
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ภายในโครงการนี้ได้มีการนำเครื่องมือต่าง ๆ มาใช้ดังนั้นเพื่อให้ง่ายต่อการศึกษาและทำความเข้าใจแล้วจึงควรมีการทำความเข้าใจเกี่ยวกับทฤษฎี และลักษณะของข้อมูลพื้นฐานที่เกี่ยวข้องกับการพัฒนาโครงการนี้อย่างคร่าว ๆ ก่อนดังนี้

2.1 ระบบจัดการฐานข้อมูล

ระบบจัดการฐานข้อมูล (Database Management System) หรือ DBMS เป็นเครื่องมือที่ใช้ในการควบคุมการสร้างและการเรียกใช้ข้อมูลในระบบฐานข้อมูล โดยทำหน้าที่เสมือนเป็นตัวกลางในการติดต่อสื่อสารระหว่างผู้ใช้ระบบกับระบบฐานข้อมูลดังที่แสดงในรูปที่ 2.1 ซึ่งช่วยในการสร้าง เรียกใช้ข้อมูล และปรับปรุงฐานข้อมูล ทำให้ผู้ใช้สามารถเพิ่ม ลบ แก้ไขและทำการค้นหาข้อมูลได้อย่างสะดวก รวดเร็ว และถูกต้องแม่นยำ และยังช่วยการเขียน โปรแกรมประยุกต์มีความสะดวกยิ่งขึ้น เนื่องจาก ผู้เขียนโปรแกรมไม่ต้องยุ่งเกี่ยวหรือสนใจว่าในทางกายภาพข้อมูลถูกเก็บอยู่ในลักษณะใด

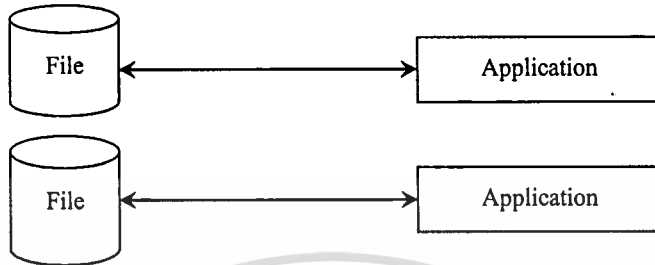


รูปที่ 2.1 แสดงโครงสร้างระบบจัดการฐานข้อมูล

วิธีที่ใช้ในการจัดการฐานข้อมูลนั้นมีวิวัฒนาการมาจากการจัดเก็บเอกสารเป็นตู้ต่อมาได้มีการเปลี่ยนแปลงวิธีการจัดเก็บมาเป็นแบบไฟล์ (File) โดยใช้คอมพิวเตอร์มาเป็นเครื่องมือในการจัดเก็บ

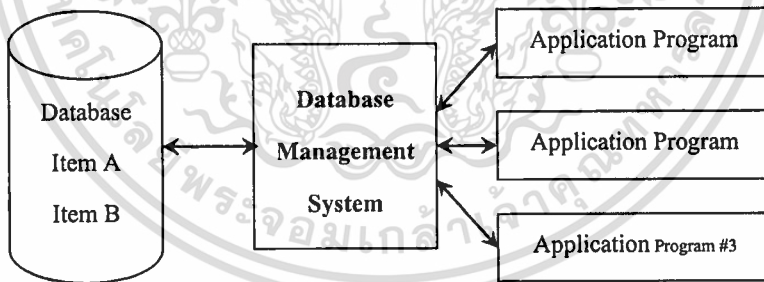
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจัดเรียงลำดับตามเรคอร์ด (Record) การจัดเก็บแบบนี้เรียกว่า การจัดเก็บแบบไฟล์ (File Approach) ดังรูปที่ 2.2



รูปที่ 2.2 การจัดเก็บข้อมูลแบบ File Approach

แต่การจัดเก็บแบบนี้ไม่มีมาตรฐานพอ โปรแกรมที่ใช้ในการค้นหา และปรับปรุงข้อมูลนั้นขึ้นอยู่กับโปรแกรมเมอร์ ดังนั้นในปัจจุบันได้มีบริษัทผลิตซอฟต์แวร์ที่ทำหน้าที่ในการจัดการข้อมูล โดยเฉพาะขึ้นมาเรียกว่า “Database Management System (DBMS)” ทำให้หน้าที่เกี่ยวกับการจัดการข้อมูลเป็นของซอฟต์แวร์เหล่านี้ แนวคิดในการใช้ DBMS นี้เรียกว่า การจัดการระบบฐานข้อมูล (Database Approach) ดังรูปที่ 2.3



รูปที่ 2.3 การจัดการข้อมูลแบบ Database Approach

ในระบบจัดการฐานข้อมูลนั้นควรมีคุณสมบัติเบื้องต้นเพื่อใช้ในการจัดการกับข้อมูลดังนี้

1. สามารถเพิ่มเติมข้อมูลลงในฐานข้อมูลได้
2. สามารถแก้ไขข้อมูลลงในฐานข้อมูลได้
3. สามารถลบข้อมูลจากฐานข้อมูลได้
4. สามารถค้นหา และเรียกข้อมูลจากฐานข้อมูลมาดูและใช้งานได้
5. สามารถแบ่งระดับ และกำหนดสิทธิในการเข้าใช้ข้อมูลในฐานข้อมูลได้

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการจัดเก็บในระบบฐานข้อมูลแทนที่จะจัดเก็บแบบไฟล์ ก็คือ

1. ข้อมูลมีการเก็บรวมกันอยู่และสามารถใช้ข้อมูลร่วมกันได้
2. ลดความซ้ำซ้อนของข้อมูล
3. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลที่อาจเกิดขึ้นได้
4. การควบคุมความคงสภาพ (Integrity) ของข้อมูล
5. การจัดการข้อมูลในฐานข้อมูลจะทำได้ง่าย
6. โปรแกรมประยุกต์ที่สร้างขึ้นมาจะไม่ขึ้นกับโครงสร้างของตารางที่เปลี่ยนแปลง
7. มี Database Administrator (DBA) เป็นผู้ควบคุม บริหาร และจัดการระบบฐานข้อมูลทั้งหมดเพียงคนเดียว

ถึงแม้ว่าการจัดเก็บในระบบฐานข้อมูลนั้นมีข้อดีอยู่มาก แต่ก็ยังมีข้อเสียด้วย ซึ่งข้อเสียของการจัดเก็บในระบบฐานข้อมูลคือ

1. ในการใช้งานฐานข้อมูลนั้นจะต้องเสียค่าใช้จ่ายในส่วนจากราคาของ DBMS ที่ค่อนข้างสูง
2. ถ้าหากคิสก์ที่ใช้ในการเก็บฐานข้อมูลนั้นอยู่เกิดมีปัญหา ทำให้อาจจะเสียข้อมูลทั้งหมดในฐานข้อมูลได้

2.2 ระบบจัดการฐานข้อมูลเชิงสัมพันธ์

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) หรือ RDBMS ได้มีการคิดค้นและพัฒนาโดย E. F. Codd เมื่อประมาณปี ค.ศ.1969

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์เป็นการจัดเก็บรวบรวมข้อมูลเป็นคอลัมน์ (Column) และแถว (Row) ในลักษณะของตารางสองมิติ (Table)

โมเดลที่ได้รับความนิยมในการใช้สำหรับงานออกแบบฐานข้อมูลโมเดลหนึ่งคือ โมเดล ER (Entity Relationship Model) ที่นำเสนอโดย Peter Chen ประมาณปีค.ศ. 1976 โดยจะมีการนำเสนอโครงสร้างของฐานข้อมูลในระดับความคิด (Conceptual Level) ให้ออกมาในลักษณะของแผนภาพ (diagram) ที่มีโครงสร้างที่ง่ายต่อการทำความเข้าใจ ทำให้สามารถมองเห็นภาพรวมของ Entity Type ทั้งหมดของระบบที่มีในระบบฐานข้อมูล รวมทั้งความสัมพันธ์ระหว่าง Entity Types เหล่านั้น และนอกจากนี้ยังเป็นแผนภูมิที่อิงกับระบบจัดการฐานข้อมูลที่มีโมเดลฐานข้อมูลแบบใด

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) หรือ RDBMS ได้มีการคิดค้นและพัฒนาโดย E. F. Codd เมื่อประมาณปี ค.ศ.1969

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์เป็นการจัดเก็บรวบรวมข้อมูลเป็นคอลัมน์ (Column) และแถว (Row) ในลักษณะของตารางสองมิติ (Table) เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเดลที่ได้รับความนิยมในการใช้สำหรับงานออกแบบฐานข้อมูลโมเดลหนึ่งคือ โมเดล E-R (Entity Relationship Model) ที่นำเสนอโดย Peter Chen ประมาณปีค.ศ.1976 โดยจะมีการนำเสนอโครงสร้างของฐานข้อมูลในระดับความคิด (Conceptual Level) ให้ออกมาในลักษณะของแผนภาพ (Diagram) ที่มีโครงสร้างที่ง่ายต่อการทำความเข้าใจ ทำให้สามารถมองเห็นภาพรวมของ Entity Type ทั้งหมดของระบบที่มีในระบบฐานข้อมูล รวมทั้งความสัมพันธ์ระหว่าง Entity Types เหล่านั้น และนอกจากนี้ยังเป็นแผนภูมิที่อิงกับระบบจัดการฐานข้อมูลที่มีโมเดลฐานข้อมูลแบบใด

2.3 ระบบจัดการฐานข้อมูลเชิงวัตถุ

ระบบจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Management System) หรือ OODBMS คือระบบการจัดการที่เกี่ยวกับฐานข้อมูลในรูปแบบของออบเจกต์ (Object) โดยที่ออบเจกต์ในที่นี้อาจหมายถึงรูปภาพ หรือกลุ่มข้อมูล เช่น ออบเจกต์ของที่อยู่ จะประกอบด้วยข้อมูลบ้านเลขที่ ถนน อำเภอ จังหวัด รหัสไปรษณีย์ ประเทศ เป็นต้น

OODBMS แตกต่างจาก RDBMS ที่เป็นระบบการจัดการเกี่ยวกับฐานข้อมูลในรูปแบบของตาราง ที่มีความสัมพันธ์กัน และระบบ ฐานข้อมูล RDBMS นั้นไม่ได้ถูกออกแบบมาเพื่อจัดเก็บข้อมูลในลักษณะของออบเจกต์โดยตรง ในขณะที่ OODBMS นั้นถูกออกแบบมาเพื่อการเก็บข้อมูลในรูปแบบของออบเจกต์โดยเฉพาะ ทำให้มีประสิทธิภาพในการจัดเก็บข้อมูลที่มีลักษณะเป็นออบเจกต์ได้ดีกว่า RDBMS

OODBMS ที่ดีควรให้การสนับสนุนคุณสมบัติพื้นฐานสำหรับออบเจกต์ดังต่อไปนี้

1. Encapsulation คือคุณสมบัติที่ออบเจกต์สามารถป้องกันข้อมูลที่อยู่ในออบเจกต์นั้นไม่ให้ถูกเปลี่ยนแปลงหรือแก้ไขได้โดยตรงจากออบเจกต์อื่นที่อยู่ภายนอก ยกเว้นมีการร้องขอผ่านทางวิธีที่ออบเจกต์นั้นเตรียมไว้ให้โดยเฉพาะ
2. Inheritance คือการที่คลาส (Class) หนึ่งมีการสืบทอดคุณสมบัติทุกอย่างมาจากอีกคลาสหนึ่ง ซึ่งคลาสที่สืบทอดคุณสมบัติมานั้นสามารถที่จะเพิ่มคุณสมบัติอื่นเพิ่มเติม นอกเหนือจากคลาสที่ถูกสืบทอดคุณสมบัติได้
3. Polymorphism คือการที่ออบเจกต์ที่ต่างกันสามารถแสดงพฤติกรรมตอบสนองที่ต่างกันเมื่อได้รับคำสั่ง (Message) ชนิดเดียวกัน
4. Object Identity คือออบเจกต์แต่ละตัวใน OODBMS จะมี Logical Object Identifier (LOID) เป็นเลขที่บอก ID ของออบเจกต์นั้น โดยที่ LOID จะไม่ซ้ำกันเลข และจะไม่มี การนำกลับมาใช้ใหม่ ถึงแม้ว่าออบเจกต์จะถูกลบออกไปจากฐานข้อมูลแล้วก็ตาม ในทางกลับกันเมื่อมีการสร้างออบเจกต์ใหม่ขึ้นมา ก็จะมีการกำหนด LOID ให้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจกต์นั้นด้วย โดยที่ LOID นี้จะไม่มีเปลี่ยนแปลง ถึงแม้ว่าจะมีการนำออบเจกต์นั้นไปเก็บยังฐานข้อมูลตัวอื่นก็ตาม

5. References Among Objects คือ OODBMS มีการอ้างอิงถึงออบเจกต์ต่าง ๆ โดยใช้ Links ซึ่ง Link ของออบเจกต์ในฐานข้อมูล แทนด้วย LOID ของออบเจกต์
6. Basic Database Operations คือคำสั่งพื้นฐานในการทำงานของ OODBMS ซึ่งมีดังต่อไปนี้

- Storing Objects คือการจัดเก็บข้อมูลออบเจกต์ลงในฐานข้อมูล ซึ่ง OODBMS จะทำการจัดเก็บข้อมูลลงในฐานข้อมูลได้โดยตรงทำให้การจัดเก็บมีความง่าย และไม่ยุ่งยาก
- Changing Objects คือการเปลี่ยนแปลงแก้ไขข้อมูลของออบเจกต์นั้นทำได้ โดยการเรียกข้อมูลนั้นขึ้นมาแล้วทำการแก้ไข จากนั้นทำการบันทึกได้ทันที
- Deleting Objects คือการลบออบเจกต์ โดยการค้นหาออบเจกต์ที่ต้องการ จากนั้นทำการลบออบเจกต์นั้นทิ้งตามปกติ
- Query คือการค้นหาข้อมูลจากฐานข้อมูล ทั้ง RDBMS และ OODBMS

จากคุณสมบัติของออบเจกต์ที่กล่าวมานั้นจะเห็นว่า ระบบจัดการฐานข้อมูลเชิงวัตถุ นั้นระบบจัดการฐานข้อมูลแบบหนึ่งที่มีคุณสมบัติเหมาะสมที่จะใช้ในการจัดการระบบฐานข้อมูล

2.4 Oracle8i

ในระบบฐานข้อมูลของออราเคิลได้มีการรองรับการทำงานของทั้ง RDBMS OODBMS และ ORDBMS ซึ่งในส่วนของออบเจกต์นั้นมีมากับผลิตภัณฑ์ของออราเคิลตั้งแต่เวอร์ชันที่ 8 เป็นต้นไป ในส่วนของออบเจกต์ได้รองรับการทำงานประเภทต่าง ๆ ดังนี้

1. Abstract Data types

คือ Data types ที่ประกอบด้วย Subtype หลาย ๆ ตัว โดยมีค่าเทียบเท่ากับ Data types มาตรฐานของออราเคิลตัวอื่น ๆ เช่น NUMBER, DATE, VARCHAR2 เป็นต้น

2. Nested Tables

คือตารางซ้อนตาราง กล่าวคือเป็นวิธีการจัดเก็บตารางความสัมพันธ์แบบ One-to-Many ลงในตารางเพียงตารางเดียว ถ้าเป็น RDBMS จะต้องใช้ Foreign Key เป็นตัวเชื่อมตาราง แต่ถ้าเป็นออบเจกต์แล้ว สามารถที่จะจัดเก็บได้โดยไม่ต้องมีการ Join Table

3. Varying Arrays

Varying Arrays หรือ VARRAYS วิธีการนี้จะมีแนวคิดคล้ายกับแบบ Nested Tables คือจัดเก็บข้อมูลหลาย ๆ ชุดโดยไม่ต้องสร้างตารางความสัมพันธ์ แต่ต้องเป็นข้อมูลที่มี Data types แบบเดียวกันเท่านั้น และขนาดของ array มีจำกัดตามที่กำหนดไว้ตอนสร้างครั้งแรก

Data Type ที่มีอยู่ใน Oracle มีอยู่หลายประเภท เช่น VARCHAR2, LONG, CHAR, Large Objects (LOB) เป็นต้น ซึ่ง LOB เป็น Data Types ประเภทหนึ่งที่มีความน่าสนใจซึ่งมีในออร์เคิลเวอร์ชัน 8 เป็นต้นไป สร้างขึ้นมาเพื่อลดจุดอ่อนของ Data Types แบบ Long และ Long Raw ที่สามารถประกาศใช้ได้ตารางละ 1 ตัวและขนาดของข้อมูลได้ 2 GB เท่านั้น นอกจากนั้นยังสร้างขึ้นมาเพื่อการจัดเก็บข้อมูลขนาดใหญ่ โดยแบ่งประเภทของข้อมูล (LOB Data Types) ออกเป็น 4 แบบคือ

- BLOB คือจัดเก็บข้อมูลเป็น Binary Data ที่สามารถขยายขนาดได้สูงสุด 4 GB. และการแก้ปัญหาคาบั้นท์กได้เพียง 1 คอลัมน์ต่อตารางของ LONG RAW ทำให้ไม่มีการจำกัดจำนวนที่ใช้จัดเก็บข้อมูล
- CLOB คือจัดเก็บข้อมูลเป็น Character Data ที่สามารถขยายขนาดได้สูงสุด 4 GB.
- NCLOB คือการเก็บ CLOB สำหรับ Multibyte Characters Sets
- BFILE คือตัวชี้ (Pointer) ซึ่งไปที่ไฟล์ข้างนอก Database (จัดเก็บอยู่ในฮาร์ดดิสก์) โดยจะจำกัดขนาดของไฟล์ตามข้อจำกัดของระบบปฏิบัติการที่ใช้ ว่ารองรับไฟล์ได้ใหญ่ที่สุดเท่าไร และเมื่อข้อมูลถูกเก็บอยู่ภายนอกระบบจัดการฐานข้อมูลออร์เคิลก็จะไม่รับเรื่องเรื่องความซ้ำซ้อนของข้อมูล และการใช้งานพร้อมกันของผู้ใช้ ซึ่งเป็นคุณสมบัติพื้นฐานของระบบจัดการฐานข้อมูล แต่ไฟล์ดังกล่าวจะทำงานอยู่ภายใต้ระบบปฏิบัติการ

วิธีการที่ใช้ในการจัดเก็บ Nested Tables และ Varying Arrays นั้นเป็นการทำแบบฝังออบเจกต์ลงไปในฐานะข้อมูล (Embedded Objects) คือจะทำการผูกโครงสร้างกับออบเจกต์อื่นที่เรียกว่า Referenced Objects ซึ่งเป็นออบเจกต์ที่จัดเก็บแยกออกมาจากออบเจกต์ที่ทำการอ้างอิงถึงมัน ดังนั้น References หรือ REFs ก็คือตัวชี้ไปยังออบเจกต์ดังกล่าว

แนวคิดที่ใช้ในการผสานระหว่าง Relational และ Object-Oriented เข้าด้วยกัน โดยจะยอมให้เพิ่ม Object-Oriented เข้าไปใน Relational Table ที่มีอยู่ เรียกว่า Object View

สำหรับการตั้งชื่อของออบเจกต์นั้น เพื่อให้สื่อถึงสิ่งเดียวกัน เราก็ควรที่จะสร้างมาตรฐานในการตั้งชื่อ โดยมาตรฐานที่นิยมใช้กันนั้นจะยึดหลักการดังนี้

1. Abstract Data Types จะเป็นชื่อที่ลงท้ายด้วย TY
2. Object Views จะเป็นชื่อที่ลงท้ายด้วย OV
3. Nested Tables จะเป็นชื่อที่ลงท้ายด้วย NT
4. Varying Arrays จะเป็นชื่อที่ลงท้ายด้วย VA

2.5 CASE Tools

CASE Tools (Computer-Aided Software Engineer) เป็นเครื่องมือที่พัฒนาขึ้นมาเพื่อใช้ในการออกแบบและสร้างระบบสารสนเทศ ประกอบไปด้วยการวิเคราะห์ระบบ การออกแบบระบบ และการสร้างระบบดังกล่าวให้เป็นระบบสารสนเทศที่ต้องการขึ้นมา ซึ่งจะเป็นเครื่องมือที่ช่วยในการทำงานของนักโปรแกรม นักวิเคราะห์และออกแบบระบบ รวมทั้ง Database Administrator ด้วย โดย CASE Tools นั้นจะใช้วิธีในการออกแบบที่เรียกว่า Top-Down Design มาช่วยในการระบุขั้นตอนสำคัญ ๆ เพื่อช่วยในการออกแบบระบบ

ปัจจุบันนี้การนำเอา CASE Tools มาช่วยในการออกแบบ วิเคราะห์ ควบคุมและทำเอกสารในระบบงานใหญ่ ๆ นั้นเป็นที่แพร่หลายมากขึ้น เนื่องจากบริษัท และองค์กรต่าง ๆ เห็นว่าการนำ CASE Tools นั้นช่วยในการสร้าง Diagram และช่วยในการแก้ไข ปรับปรุงงานในระบบที่มีขนาดใหญ่มากนั้นสามารถทำได้เป็นอย่างดี

อีกทั้งโปรแกรม CASE Tools ต่าง ๆ ที่มีการพัฒนาขึ้นมาในปัจจุบันมีองค์ประกอบที่ทำให้สามารถใช้งานได้ง่ายและมีประสิทธิภาพมากขึ้น ตัวอย่างเช่น

1. มีการใช้ Graphics ช่วยในการออกแบบและสร้าง Diagrams เช่น Data Flow Diagrams, E-R Diagrams เป็นต้น
2. มีตัว Generators สำหรับสร้างหน้าจอเพื่อใช้ในการแสดงผล input/output ของระบบสารสนเทศ เช่นหน้าจอสำหรับ User
3. มี Repository ที่เก็บข้อมูลเกี่ยวกับการออกแบบทั้งหมดไว้ รวมทั้งยังมี Data Dictionary ที่เข้าใจได้ง่าย
4. มีการแบ่งส่วนการทำงานในการตรวจสอบความถูกต้อง โครงสร้าง และความสมบูรณ์ของระบบ

2.5.1 ประโยชน์ของ CASE Tools

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการนำ CASE Tools มาช่วยในการออกแบบและวิเคราะห์ระบบนั้น มีส่วนช่วยในขั้นตอนต่าง ๆ มากมายโดยมีประโยชน์ ดังต่อไปนี้

- ช่วยลดระยะเวลาที่ใช้ในการพัฒนา
- ช่วยลดค่าใช้จ่ายในการพัฒนา
- ช่วยในการทำงานของ SDLC (System Development Life Cycle)
- มีระเบียบวิธีในการพัฒนาระบบที่เป็นมาตรฐาน
- ในการดูแลรักษา Application ของระบบที่พัฒนาโดยใช้ CASE Tools นั้นสามารถทำได้ง่าย

2.5.2 การทำงานของโปรแกรม CASE Tools

ในการทำงานของ CASE Tools ส่วนใหญ่ที่มีการพัฒนาขึ้นมาในปัจจุบัน มีการแบ่งชนิดออกเป็น 3 แบบโดยแบ่งจากลักษณะของคุณสมบัติในการจัดการระบบ คือ

2.5.2.1 Upper CASE Tools

Upper CASE Tools หรือ Front-End CASE Tools ซึ่งเป็นโปรแกรมที่มึการทำงานในส่วนที่เกี่ยวกับการวิเคราะห์เกี่ยวกับการ Create และ Modify การออกแบบระบบ โดยสารสนเทศภายในระบบทั้งหมดจะถูกเก็บรวบรวมไว้ที่ CASE Repository ซึ่งข้อมูลที่ถูกเก็บไว้ประกอบไปด้วย Records, Elements, Diagrams, Screens, Reports และ Information อื่น ๆ

Upper CASE Tools สามารถช่วยสนับสนุนการแปรความต้องการเกี่ยวกับหน้าที่ขององค์กร ช่วยในการวิเคราะห์ และช่วยในการ Drawing ขอบเขตของโครงการนั้น ซึ่ง Upper CASE Tools บางตัวยังช่วยสนับสนุนส่วนของ Prototype ของการออกแบบหน้าจอ และรายงานต่าง ๆ ด้วย

2.5.2.2 Lower CASE Tools

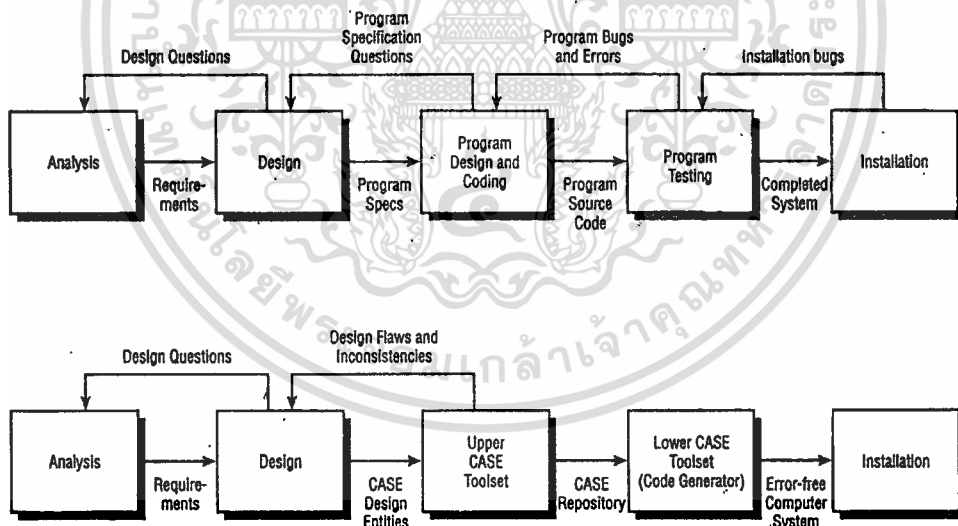
Lower CASE Tools หรือ Back-End CASE Tools เป็นโปรแกรมที่มีการ Generate Source Code ช่วยลดขั้นตอนการเขียน โปรแกรมของระบบ ซึ่งมีข้อดีคือ

1. ในการ Generate Code โดย CASE Tools นั้นจะเร็วกว่าการที่ Programmer จะเป็นผู้เขียนเอง
2. ช่วยลดระยะเวลาที่เสียไปในการ Maintenance และพัฒนาระบบ เนื่องจากการสามารถ Modify การออกแบบ CASE และ Regenerate Code ได้ทันที

2.5.2.3 Integrated CASE Tools

Integrated CASE Tools เป็นโปรแกรมที่รวมการทำงานของทั้ง Upper CASE Tools และ Lower CASE Tools เข้าไว้ในโปรแกรม โดยสามารถที่จะช่วยในการออกแบบระบบ และทำการ Generate Code ได้ภายในโปรแกรมเดียวกัน

สำหรับลักษณะและขั้นตอนในการทำงานของ CASE Tools นั้นได้มีการออกแบบให้สอดคล้องกับขั้นตอนและการทำงานในการออกแบบของ System Development Life Cycle ดังจะเห็นได้จากรูปที่ 2.4 ทางด้านล่าง ซึ่งจะแสดงให้เห็นขั้นตอนและการทำงานของทั้ง SDLC และ CASE Tools ซึ่งมีลักษณะที่คล้ายคลึงกัน



รูปที่ 2.4 แสดงขั้นตอนการพัฒนาแบบ SDLC และ CASE Tools (ที่มา: Kenneth, E.K. and Julie, E. 1999. System Analysis and Design. New Jersey: Prentice-Hall.)

บทที่ 3

Oracle Designer6i CASE Tools

บริษัท Oracle เป็นบริษัทหนึ่งที่ได้มีการพัฒนา CASE Tools ขึ้นมา ซึ่ง Oracle Designer6i ก็เป็น CASE Tool ตัวหนึ่งของบริษัท Oracle ที่ได้ทำการพัฒนาให้สามารถใช้พัฒนาระบบงานได้ง่าย รวดเร็ว และสร้าง Application ได้อย่างรวดเร็ว มีประสิทธิภาพ แล้วยังรองรับการออกแบบระบบฐานข้อมูลแบบ Object และฐานข้อมูลแบบ Relational Database ได้โดยไม่มีปัญหาใด

การเก็บรวบรวมข้อมูลต่าง ๆ ของ Oracle Designer6i นั้นจะเก็บไว้ที่ส่วนกลางที่เรียกว่า Repository ซึ่งจะเก็บ Object, Meta Data ที่เกี่ยวกับการวิเคราะห์ ออกแบบและการสร้างระบบทั้งหมด ของระบบที่ทำการพัฒนา

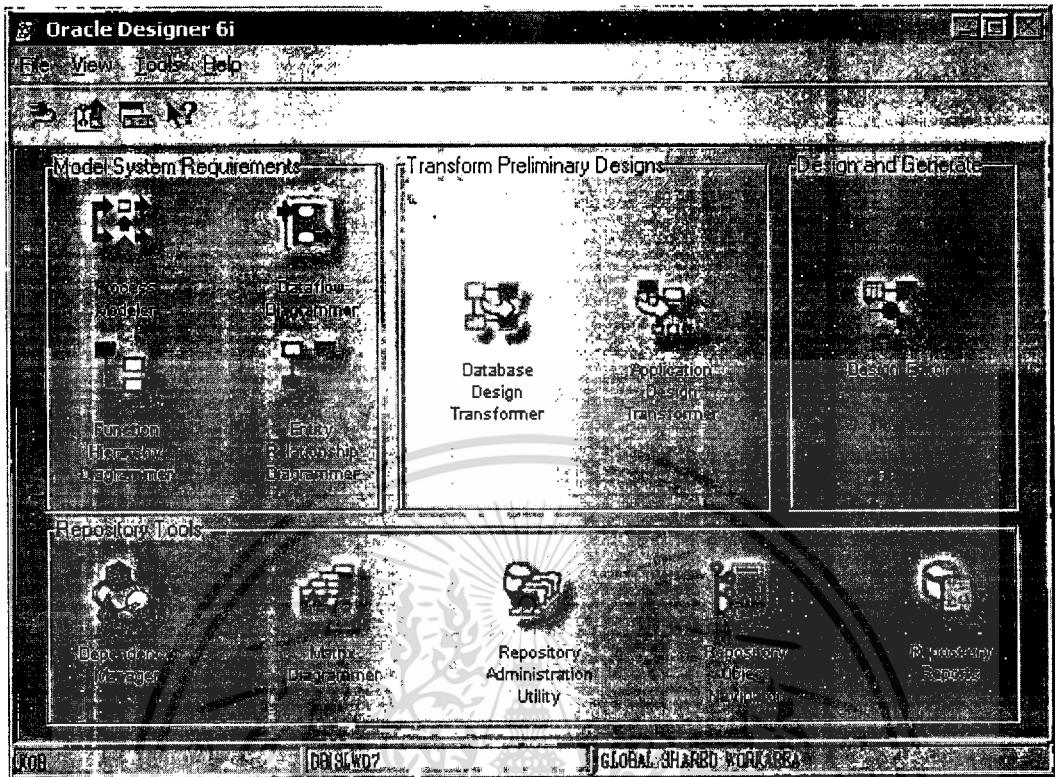
ความสามารถในการสร้างระบบสารสนเทศของ Oracle Designer6i นั้นมีความสามารถที่ครอบคลุมการออกแบบระบบกล่าวคือ สามารถที่จะสร้าง Forms ต่าง ๆ Web Applications ซึ่งมีการสนับสนุนภาษา Java Reports ของระบบสามารถสร้าง Menu รวมไปถึงสามารถที่จะช่วยในการสร้าง Document ของระบบ ได้อีกด้วย

3.1 ส่วนประกอบของ Oracle Designer6i

ในการทำงานของ Oracle Designer6i จะมีส่วนประกอบในการทำงานจะแบ่งออกเป็น 2 ส่วนใหญ่ ๆ ซึ่งประกอบไปด้วยในส่วนของฝั่งของไคลเอนต์ซึ่งจะเกี่ยวข้องกับเครื่องมือ และ Utility ที่ใช้ในการออกแบบ การพัฒนา และการจัดการรายละเอียดต่าง ๆ ภายในระบบ และอีกส่วนจะเป็นของฝั่งเซิร์ฟเวอร์ซึ่งจะมีการทำงานที่เกี่ยวข้องกับการจัดการคลังข้อมูลส่วนกลางที่เป็นศูนย์รวมของข้อมูลและอ็อบเจกต์ทั้งหมดที่มีภายในระบบ

3.1.1 ฝั่งของไคลเอนต์ (Client-side)

ในฝั่งของไคลเอนต์เครื่องมือส่วนใหญ่ที่ใช้พัฒนาระบบจะปรากฏที่ในหน้าจอแรกเมื่อเข้าสู่ระบบของโปรแกรม Oracle Designer6i โดยจะมีรายละเอียดตามในรูปที่ 3.1



รูปที่ 3.1 แสดงหน้าจอแรกของ Oracle Designer 6i

ซึ่งจะมีการแบ่งเครื่องมือออกเป็นกลุ่มตามลักษณะการทำงานแบบต่าง ๆ ที่อยู่ในส่วนประกอบหลัก ๆ จะมี 3 ส่วนคือ

Client-Side Toolkit เป็นชุดเครื่องมือที่ใช้ในการออกแบบระบบ มีดังนี้

1. Model System Requirements

เครื่องมือในส่วนนี้จะมีการทำงานดังต่อไปนี้จำลองการดำเนินงานขององค์กร การเปลี่ยนแปลงระเบียบวิธีเพื่อให้องค์กรได้บรรลุเป้าหมาย การสร้างแผนผังการดำเนินงานขององค์กร การเก็บรายละเอียดเกี่ยวกับองค์กร รายละเอียดความต้องการขององค์กร การสร้างรูปแบบแผนผังของเอนทิตี (Entities) และหน้าที่ต่าง ๆ รวมทั้งการไหลของข้อมูลภายในระบบ เครื่องมือที่ใช้ทำงานภายในกลุ่มเครื่องมือนี้คือ

- **Process Modeler**

เป็นเครื่องมือที่ใช้ในการนำเสนอเป้าหมายของการทำงานขององค์กรทั้งในปัจจุบันและอนาคต โดยมีการเพิ่มการวัดค่าต่าง ๆ เช่น หน่วยของเวลา ค่าใช้จ่าย ระยะเวลา เป็นต้น ให้ออกมาในรูปแบบของ โครงสร้าง (Model)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยส่วนนี้ ได้มีการจัดการที่ยืดหยุ่นสำหรับรองรับในกรณีที่จะทำการปรับเปลี่ยนโครงสร้างการทำงานขององค์กรใหม่

- **Function Hierarchy Diagrammer**
เป็นเครื่องมือที่ใช้ในการแสดงเกี่ยวกับลักษณะและการทำงานขององค์กรเบื้องต้น โดยแสดงให้เห็นว่าใช้ Entities อย่างไรและ Attribute ที่ใช้ในการสร้างแบบแอปพลิเคชันที่สนับสนุนการวิเคราะห์หน้าที่เพื่อที่จะใช้ในการจัดการ โปรเจ็คและการเข้าถึงตัว Application Design Transformer เพื่อใช้สำหรับสร้างแอปพลิเคชัน
- **Dataflow Diagrammer**
เครื่องมือที่ใช้ในการทำแผนผังในการแสดงการไหลของข้อมูลภายในธุรกิจที่มีการระบุในระดับต่าง ๆ ว่ามีการใช้ข้อมูลอะไรภายในระบบและข้อมูลนั้นมีความสัมพันธ์และข้องเกี่ยวกับข้อมูลอื่น ๆ
- **Entity Relationship Diagrammer**
เป็นเครื่องมือที่ใช้แสดงความต้องการภายในระบบขององค์กรและการเข้าถึงตัว Database Design Transformer สำหรับสร้างแบบฐานข้อมูล โดยมีการสร้างแผนผัง ER Diagram ซึ่งเป็นสารสนเทศที่มีเครื่องหมายที่เป็นมาตรฐานในการแทนสิ่งสำคัญต่าง ๆ ได้แก่ entity, attribute และ relationship

2 Transform Preliminary Designs

เป็นเครื่องมือที่ใช้ในการเปลี่ยนแปลงการรูปแบบโครงสร้างอย่างง่าย ให้เป็นการออกแบบเบื้องต้น เครื่องมือที่จัดอยู่ในการทำงานของกลุ่มเครื่องมือนี้คือ

- **Database Design Transformer**
เป็นตัวที่ใช้ในการจัดการแผนผังการออกแบบฐานข้อมูลที่ได้จากการออกแบบ ER Diagram ตามความต้องการของระบบที่ได้จากในขั้นตอนของ Entity Relationship Diagrammer ให้อยู่ในรูปของข้อมูลที่สามารถที่จะเปลี่ยนให้เป็นฐานข้อมูลได้
- **Application Design Transformer**
เป็นตัวที่ใช้ในการจัดการข้อมูลที่ได้จากขั้นตอนของ Function Hierarchy Diagrammer และ Dataflow Diagrammer เพื่อให้อยู่ในรูปแบบที่สามารถจัดทำโครงสร้างเป็นแอปพลิเคชันที่ตั้งอยู่บนพื้นฐานของหน้าที่และหน่วย

ทางธุรกิจ ที่ได้จากความต้องการของระบบได้ โดยในการทำงานมีการสร้างกฎเกณฑ์ที่ง่ายต่อการใช้งาน เช่น Oracle Forms Oracle Reports เป็นต้น

3. Design and Generate

ในกลุ่มเครื่องมือที่ใช้ในการออกแบบแนวทางของระบบให้เป็นฟอร์ม (Form) ที่สอดคล้องและตรงตามความต้องการขององค์กรสำหรับผู้ที่ทำหน้าที่จัดการระบบและผู้ที่ออกแบบระบบ โดยมีการสร้างองค์ประกอบต่าง ๆ ในฝั่งของเซิร์ฟเวอร์และแอปพลิเคชัน ในฝั่งของไคลแอนต์จากรายละเอียดเรคอร์ดที่อยู่ภายในคลังข้อมูล ส่วนกลาง เครื่องมือที่ทำหน้าที่ในส่วนนี้มีเพียงตัวเดียวคือ

- Design Editor

เป็นศูนย์กลางในการควบคุมและจัดการข้อมูลที่ได้จากข้อ 4.2.2 ให้การออกมาเป็นฐานข้อมูลและแอปพลิเคชันที่ต้องการในการพัฒนาระบบ การสร้างระบบและการจับภาพการในส่วนของระบบที่ออกแบบ

ในการเข้าถึง โครงสร้าง และกฎเกณฑ์ต่าง ๆ ทางแผนผังธุรกิจของ Design Editor จะช่วยสนับสนุนการออกแบบ และความสัมพันธ์ต่าง ๆ ในการสร้าง และโครงสร้างของฐานข้อมูลแบบอ็อบเจกต์ รีเลชันนอล ดังนี้

- สร้างแอปพลิเคชันสำหรับเป้าหมายจาก Meta data ที่เก็บอยู่ในคลังข้อมูล ส่วนกลาง
- สร้าง ออราเคิลฟอร์ม (Oracle Forms) ออราเคิลรีพอร์ต (Oracle Reports) และเว็บที่เป็น PL/SQL ได้จากหนึ่งไดอะแกรม
- กำหนดและสร้างแอปพลิเคชัน ตามเหตุผลจากข้อมูลที่เป็นภาษา PL/SQL หรือ จาวาสคริปต์ (JavaScript)
- ทำการสร้างและจับภาพที่ออกแบบทั้งที่เป็นฐานข้อมูลของออราเคิล และฐานข้อมูลอื่นที่ไม่ใช่ออราเคิล
- จับภาพที่ออกแบบแอปพลิเคชันเพื่อใช้ในการวิเคราะห์ ปรับปรุงการออกแบบ และปรับปรุงระบบที่สร้างไว้

The Generator ใช้ในการสร้างส่วนต่าง ๆ ของระบบ โดยแบ่งเป็น 2 ประเภท ดังนี้

1. Server Generator

ซึ่งสามารถที่จะสร้างในส่วนของฐานข้อมูลของ Oracle, ODBC และ Database ประเภทอื่นได้

2. Client Generator

สามารถที่จะสร้าง application ได้ในหลายแพลตฟอร์มตามที่ผู้ใช้ต้องการ เช่น

- Oracle Forms และ Oracle Reports
- Oracle Web PL/SQL application
- Visual Basic Project
- MS Help application

Utilities and Service เป็นส่วนจัดการอื่น ๆ ซึ่งประกอบด้วย

1. Repository Tools

กลุ่มเครื่องมือในกลุ่มนี้ทำเกี่ยวกับการนำสารสนเทศเข้าไปในคลังข้อมูล ส่วนกลาง และแก้ไขสารสนเทศที่อยู่ในคลังข้อมูลส่วนกลาง มีการแสดงความสัมพันธ์ระหว่าง element ในคลังข้อมูลส่วนกลาง รวมถึงการส่งเสริมต่าง ๆ ที่คลังข้อมูลส่วนกลาง เครื่องมือที่ใช้ในการจัดการเกี่ยวกับส่วนนี้คือ

- **Dependency Manager**
การวิเคราะห์ความสัมพันธ์ระหว่างข้อมูลที่มีโครงสร้างและที่ไม่มีโครงสร้างที่เก็บอยู่ในคลังข้อมูลส่วนกลาง การวิเคราะห์ผลกระทบเมื่อมีการเปลี่ยนแปลงอ็อบเจกต์ และการแก้ไขการจัดการต่าง ๆ
- **Matrix Diagrammer**
จะมีการสนับสนุนในส่วนของการประเมินเกี่ยวกับมุมมองของโปรเจกต์ การวิเคราะห์เกี่ยวกับผลกระทบ การวางแผนเครือข่าย และการควบคุมคุณภาพ
- **Repository Administration Utility**
เป็นเสมือนเส้นประสาทส่วนกลางในการควบคุมคลังข้อมูลส่วนกลางของออราเคิล ใช้ในการดูแลคลังข้อมูลส่วนกลางแบบวันต่อวัน เช่นทำการตรวจสอบการทำงานของอ็อบเจกต์ในคลังข้อมูลส่วนกลาง การสำรองอ็อบเจกต์ในคลังข้อมูลส่วนกลาง และการดูแลการเข้าถึงของผู้ใช้งาน เป็นต้น
- **Repository Object Navigator**
เป็นโครงสร้างพื้นฐานสำหรับการจัดเก็บและจัดการอ็อบเจกต์ทั้งหมดที่มีในการพัฒนาระบบ และมีการจัดเก็บไฟล์ลงในคลังข้อมูลส่วนกลาง โดยจะมีการใช้งานร่วมกับ **Repository Administration Utility**

เพื่อใช้ในการดูแลความสัมพันธ์ทั้งหมดที่อยู่ภายในและการเข้าถึง
อ็อบเจกต์ของคลังข้อมูลส่วนกลาง

- Repository Reports

ได้มีการจัดหารายงานเกือบ 100 รายงานเพื่อใช้สำหรับตรวจสอบสิ่งที่อยู่ในคลังข้อมูลกลาง มีการแสดงการเขียนตัวนำทางรายงานต่าง ๆ โดยการ
ใช้ชื่อหรือการแบ่งแยกที่เป็นกลุ่มขึ้นอยู่กับการใช้งานหรือสารสนเทศที่
เราต้องการที่จะแสดง

2. Online Documentation

เป็นเอกสารที่คอยแนะนำและให้ความช่วยเหลือ เมื่อมีข้อสงสัยเกี่ยวกับการใช้
งาน Oracle Designer 6i

3.1.2 ฝั่งของเซิร์ฟเวอร์ (Server-side)

มีส่วนประกอบหลักในการจัดการอยู่ 2 ส่วน คือ

1. Repository

ซึ่งเป็นส่วนหลักของ Oracle Designer6i โดยเป็นศูนย์กลางในการเก็บข้อมูลต่าง ๆ
ของ Definition ของ Object เท่านั้น โดยใน Repository นี้สามารถที่จะกำหนดสิทธิ์ได้
ว่าใครสามารถที่จะเข้ามาใช้งานในส่วนใดของ Repository ได้บ้าง โดย Repository
Administration ผู้ที่กำหนดสิทธิ์ในการเข้ามาใช้งาน Repository

2. API

เป็นส่วนที่ทำหน้าที่ในการ Update Repository ซึ่งจะ Update ทันที เมื่อมีการ
เปลี่ยนแปลงและแก้ไขข้อมูลผ่านทาง Tools ต่าง ๆ ของ Oracle Designer6i

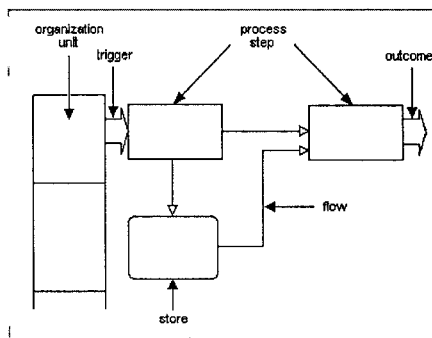
3.2 การออกแบบ Process Model

การออกแบบ Process Model เป็นส่วนที่ใช้อธิบายว่าระบบสารสนเทศที่จะทำการพัฒนามี
ขั้นตอนการดำเนินงานอย่างไร และแสดงลำดับการทำงานว่าขั้นตอนใดเกิดก่อนเกิดหลัง อีกทั้งยัง
แสดงด้วยมีหน่วยงานใดเกี่ยวข้องกับขั้นตอนการทำงานนั้นบ้าง และยังอธิบายถึงเหตุการณ์ที่ทำให้
เกิดหรือเป็นผลจากขั้นตอนการทำงานนั้น และช่วยให้เห็นและเข้าใจลักษณะการดำเนินงานของ
ระบบที่จะพัฒนามากยิ่งขึ้น

โดยในการออกแบบ Process Modeler มีสัญลักษณ์ต่าง ๆ เพื่อใช้ในการออกแบบดังในรูปที่

3.2 คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงสัญลักษณ์ที่ใช้ใน Process Modeler

รายละเอียดต่าง ๆ เกี่ยวกับสัญลักษณ์ที่อยู่ภายในการออกแบบของ Process Modeler มีดังนี้

1. Organization Unit เป็นสัญลักษณ์ที่ใช้แทนหน่วยงานต่าง ๆ ที่มีหน้าที่เกี่ยวข้องในต่างของการรับผิดชอบกิจกรรมนั้น ๆ
2. Process Step เป็นสัญลักษณ์ที่แสดงถึงกระบวนการดำเนินงานภายในธุรกิจ ซึ่งสามารถแบ่ง Process Step ออกได้ 4 ประเภท ดังนี้
 - 2.1 Generic เป็นกระบวนการทำงานที่มีการประมวลผลเพื่อให้ได้ผลลัพธ์ออกมา โดยที่ไม่มีการแก้ไขเปลี่ยนแปลงข้อมูลใด ๆ
 - 2.2 Data Entry เป็นกระบวนการทำงานที่เกี่ยวข้องกับการเพิ่ม แก้ไข ลบ หรือเปลี่ยนแปลงข้อมูล
 - 2.3 Report เป็นกระบวนการที่แสดงข้อมูลให้อยู่ในรูปแบบของรายงาน
 - 2.4 Decision Point เป็นกระบวนการทำงานที่แสดงเงื่อนไขซึ่งเป็นทางเลือกในการตัดสินใจการทำงานของกิจกรรมนั้น ๆ
3. Flow เป็นสัญลักษณ์ที่แสดงลำดับการทำงานก่อนและหลังของ Process Step ว่าจากขั้นตอนนี้แล้วจะทำขั้นตอนใดต่อไป
4. Event ในทำงานของ Flow นั้น Process Model จะมีการระบุเหตุการณ์อยู่ 2 ประเภท ดังนี้
 - 4.1 Trigger

เป็นเหตุการณ์ที่เป็นจุดเริ่มต้นของ Process Flow โดย Trigger จะแสดงทางซ้ายของ Process Step
 - 4.2 Outcome

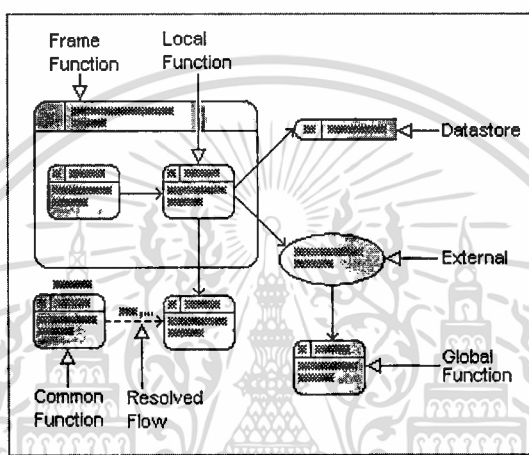
เป็นเหตุการณ์ที่เป็นจุดสิ้นสุดหรือเป็นผลลัพธ์ของ Process Flow โดย Outcome จะแสดงทางขวาของ Process Step

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Store เป็นสัญลักษณ์ที่ใช้แสดงแหล่งที่เก็บข้อมูลของระบบ

3.3 การออกแบบ Dataflow Diagram (DFD)

การออกแบบ DFD เป็นการออกแบบที่ทำให้ผู้พัฒนาระบบสามารถมองเห็นกระบวนการทำงานและการไหลของข้อมูล ซึ่ง Dataflow Diagrammer จะมีสัญลักษณ์ซึ่งใช้แทนส่วนประกอบต่างๆ ของ DFD ดังที่ได้แสดงในรูปที่ 3.3



รูปที่ 3.3 แสดงสัญลักษณ์ที่ใช้ใน Dataflow Diagrammer

จากรูปที่ 3.3 จะเห็นว่าสัญลักษณ์และรายละเอียดต่าง ๆ โดยแบ่งออกเป็น 2 ส่วนใหญ่ ๆ ดังนี้

1. ส่วนที่อยู่ภายใน Frame Function

ซึ่งเป็นกระบวนการทำงานหลักที่เราสนใจ โดยภายใน Frame Function นั้นก็จะ เป็นกระบวนการทำงานย่อย ๆ ที่เรียกว่า Local Function อยู่โดย Local function จะไม่สามารถอยู่นอก Frame Function ได้ ซึ่ง Function ต่างๆ เหล่านี้ก็จะมีการรับและส่ง ข้อมูลเข้ามาประมวลผลจาก Datastore ซึ่งเป็นส่วนที่ทำหน้าที่เก็บข้อมูล และอาจจะมีการรับและส่งข้อมูลไปยังภายนอกก็ได้

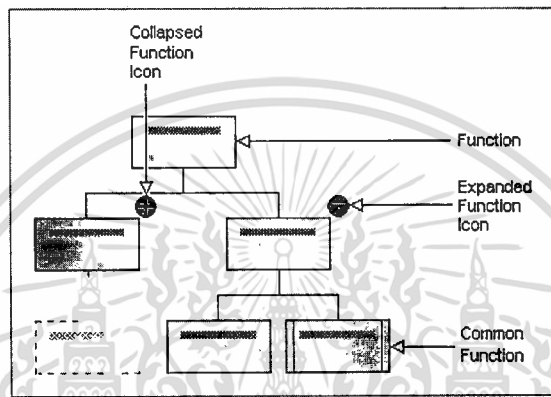
2. ส่วนที่อยู่ภายนอก Frame function ซึ่งมีการทำงานเป็น Function มีอยู่ 2 ชนิด คือ

2.1 Common Function ซึ่งเป็นกระบวนการทำงานที่เกิดขึ้นมาจาก Frame Function อื่น ๆ

2.2 Global Function ซึ่งจะเป็นกระบวนการทำงานที่ได้มาจาก Function อื่นๆ ที่ นอกเหนือจาก Frame Function นั้นเอง

3.4 การออกแบบ Function Hierarchy Diagram (FHD)

เมื่อได้ข้อมูลการออกแบบและสร้าง DFD เรียบร้อยแล้ว ก็จะนำเอา DFD ที่ได้นั้นมาทำการสร้าง FHD ขึ้นมา ซึ่ง FHD นั้นจะช่วยทำให้เราเห็นภาพรวมของกระบวนการทำงานทั้งหมดว่า มีกระบวนการทำงานอะไรบ้าง และแต่ละกระบวนการมีกระบวนการทำงานย่อย ๆ เป็นอย่างไร ซึ่งสัญลักษณ์ที่ใช้ในการออกแบบ FHD นั้นมีดังในรูปที่ 3.4



รูปที่ 3.4 แสดงสัญลักษณ์ที่ใช้ใน Function Hierarchy Diagrammer

โดย Function ที่อยู่ภายใน FHD นั้นจะแบ่งตามระดับของ node สามารถที่จะแบ่งได้ 3 ประเภท ได้แก่

1. Root Function ก็คือ Function ที่แสดงอยู่ในรูปแบบของ node ที่อยู่บนสุดของ FHD
2. Full Function คือ Function ที่มี Function ย่อย ๆ อีกอย่างน้อย 1 Function ซึ่งก็คือ Function ที่ทำหน้าที่เป็น Parent node นั้นเอง
3. Atomic Function คือ Function ที่ไม่มี Child node ย่อยลงไปอีก ซึ่งสามารถเรียกได้อีกอย่างว่า เป็น Leaf Function นั้นเอง

สำหรับวิธีการในการสร้าง FHD ขึ้นมานั้นมีอยู่ 2 วิธี ดังนี้

1. ทำการออกแบบและสร้างด้วยตัวเอง
2. ให้ Function Hierarchy Diagrammer ทำการสร้าง FHD ให้อัตโนมัติ ซึ่งวิธีนี้จะทำได้ก็ต่อเมื่อได้ทำการออกแบบและสร้าง DFD ไว้เรียบร้อยแล้ว

3.5 การออกแบบ Entity Relationship Diagram

การสร้าง ER Diagram ถือได้ว่าเป็นส่วนที่สำคัญที่สุดในการออกแบบฐานข้อมูล ซึ่งถ้าออกแบบ ER Diagram ไม่ดีก็จะส่งผลกระทบต่อฐานข้อมูลที่จะนำไปใช้งาน และจะส่งผลกระทบต่อเอกสารทุกชิ้นที่เกี่ยวข้องกับการนำข้อมูลไปใช้ ไม่ว่าจะเป็นกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.6 จะแสดงถึง DDT ซึ่งจะมีตัวเลือกให้ผู้ใช้ตัดสินใจว่าจะทำการแปลงทุกเอนทิตีหรือเฉพาะบางเอนทิตี นอกจากนี้ผู้ใช้สามารถที่จะตั้งค่าต่าง ๆ เช่น ขนาดของ Table Space, Constraint หรือ Index ต่าง ๆ ได้ ซึ่ง DDT จะทำการแปลง ER Diagram ให้อยู่ในรูปของ First-cut Data Design โดยจะแปลง ER Diagram เป็นลักษณะของฐานข้อมูล ดังนี้

1. Entity จะถูกแปลงเป็น Table
2. UID จะถูกแปลงเป็น Primary Key
3. Attribute จะถูกแปลงเป็น Column
4. Relationship จะถูกแปลงเป็น Foreign Key และ Constraint ต่าง ๆ

เมื่อทำการแปลงข้อมูลให้อยู่ในรูปของ First-cut Data Design แล้ว ก็จะใช้ Design Editor ในการสร้างฐานข้อมูลให้ โดย Design Editor จะนำงานต่าง ๆ ที่ได้จากขั้นตอนของ DDT มาทำการสร้าง Server Model Diagram ขึ้น เพื่อให้ผู้ใช้ได้เห็นถึงรายละเอียดของตารางและ Constraint ต่าง ๆ ที่ได้จากการทำงานของ DDT

ซึ่งในโปรแกรม Design Editor จะมีหน้าต่างหลักที่ใช้แสดงข้อมูลอยู่ 2 หน้าต่าง คือ

1. Navigator ที่ใช้แสดงอ็อบเจกต์ที่ได้จากการสร้างมาจาก DDT ทั้งหมด
2. Property Palette ที่จะแสดงให้เห็นคุณสมบัติของอ็อบเจกต์

การทำงานของ Server Model Diagram สามารถที่จะทำการเพิ่มหรือแก้ไขคอลัมน์ในแต่ละตารางได้ ซึ่งทำการปรับปรุงและเปลี่ยนแปลงข้อมูลต่าง ๆ ใน Server Model Diagram เรียบร้อยแล้ว ก็สามารถที่จะทำการสร้างฐานข้อมูลได้ ซึ่งการสร้างฐานข้อมูลขึ้นมานั้น อาจจะสร้างเป็น Table ในฐานข้อมูล หรืออาจจะสร้างให้อยู่ในรูปของ Script DDL File ก็ได้ ซึ่งจะปรากฏ Dialog Box ให้ผู้ใช้เลือกว่า ต้องการจะสร้างฐานข้อมูลในรูปแบบใด ซึ่งสามารถที่จะกำหนดรูปแบบการสร้างฐานข้อมูลได้ 3 แบบ คือ

1. DDL File

Design Editor จะทำการสร้าง Script File ซึ่งผู้ใช้สามารถที่จะนำ Script File ที่ได้ไปสร้างฐานข้อมูลเองได้ โดยวิธีนี้สามารถที่จะสร้างฐานข้อมูลได้หลายชนิด ไม่จำกัดเฉพาะว่าต้องเป็นฐานข้อมูล Oracle เพียงอย่างเดียวเท่านั้น

2. Database

Design Editor จะทำการสร้าง Table ลงไปในตัวฐานข้อมูล Oracle โดยจะให้ผู้ใช้ใส่ Username, Password และชื่อของเครื่องคอมพิวเตอร์ที่ต้องการจะสร้างฐานข้อมูลลงไป

3. ODBC

Design Editor จะสร้างฐานข้อมูลผ่านทาง ODBC ให้โดยผู้ใช้สามารถเลือกชนิดและรูปแบบของฐานข้อมูลได้ตามที่ต้องการ

เมื่อเลือกรูปแบบและชนิดของฐานข้อมูลที่ต้องการจะสร้างขึ้นเรียบร้อยแล้ว Design Editor จะทำการสร้างฐานข้อมูลตามค่าที่ผู้ใช้กำหนด

3.7 การสร้าง Application

การที่จะสร้าง Application Module ขึ้นมานั้น จะต้องมีการกำหนดก่อนว่า ในแต่ละ Function นั้นใช้ข้อมูลอะไรในการทำงานบ้าง ซึ่งจะใช้ข้อมูลที่มาจากใน ER Diagram ที่ได้สร้างไว้ก่อนแล้ว ซึ่งการกำหนดข้อมูลให้แก่ Function นั้นจะทำใน Edit Function Dialog Box ของแต่ละ Function ซึ่งจะมีการให้เลือกลูก Entity และมีการกำหนดด้วยว่า จะให้มีการใช้ข้อมูลได้ยังไงบ้าง เช่น การกำหนดให้ฟังก์ชันใช้ข้อมูลจาก Entity ใด และสามารถที่จะเพิ่ม ลบ หรือแก้ไขข้อมูลใดได้บ้าง เป็นต้น

เมื่อได้ทำการกำหนดรายละเอียดของ Entity ทุกตัวแล้ว จะมีการใช้ Function/Attribute Matrix Utility ที่อยู่ใน Function Hierarchy Diagram เป็นตัวจัดการว่าในแต่ละ Entity ที่เข้าไปใช้ข้อมูลนั้น ต้องใช้ Attribute ใดบ้าง ซึ่งเมื่อทำเรียบร้อยแล้ว ใน Edit Function Dialog Box ก็จะมีการเซตค่าเกี่ยวกับส่วน Attribute ให้แล้วโดยอัตโนมัติ

หลังจากทำการกำหนดข้อมูลให้แก่แต่ละ Function แล้ว ก็ต้องกำหนดคุณสมบัติของ Function โดยจะเป็นการกำหนดคุณสมบัติในเรื่องของการให้ผลลัพธ์ (Response) ออกมา ซึ่งมีอยู่ 2 ประเภทคือ

1. Immediate ซึ่งจะให้ผลลัพธ์ในทันทีทันใด
2. Overnight ซึ่งจะให้ผลลัพธ์ตามกำหนดเวลาที่เรานำเลือก

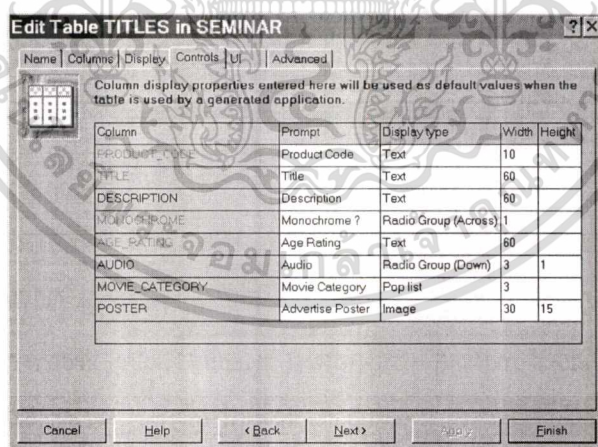
ซึ่งการเลือกคุณสมบัติของ Function นั้นจะส่งผลต่อ Application Module ด้วย เช่นถ้าเราต้องการให้ Application Module ของเราแสดงผลทางหน้าจอ ก็ต้องเลือกคุณสมบัติเป็น Immediately มิฉะนั้น ก็อาจจะกลายเป็น Utility หรือไม่ก็อาจจะกลายเป็น Report ซึ่งอาจจะผิดพลาดประสงคได้ ดังนั้นจึงต้องควรระวังในส่วนของการเลือกด้วย โดยคุณสมบัติของ Function นั้นมีดังในตารางที่ 3.1

ตารางที่ 3.1 Response ที่มีผลต่อ Application Module

Entity Usage		
Response	Any of C/U/D	R only
Immediate	Screen	Screen
Overnight	Utility	Report

ในการตรวจสอบอีกครั้งหนึ่งว่าข้อมูลกับ Function ที่ใช้นั้นสัมพันธ์กันหรือไม่นั้น จะทำได้โดยการ Cross-check ระหว่าง Function กับข้อมูลนั้น โดยการใช้ Utility ที่มีชื่อว่า Create Retrived Update Delete matrix (CRUD matrix) โดยกฎของการ Cross-checking ก็คือแต่ละ Entity จะต้องมีการสร้าง ลบ และเข้าไปอ่านข้อมูลโดย Function โดยที่การ Update นั้นอาจจะมีหรือไม่ก็ได้

ก่อนที่จะทำการสร้าง Module ขึ้นมานั้นจะต้องทำการกำหนดค่าของ Display Type ของแต่ละ column ในแต่ละ Table ก่อน โดยกำหนดว่า Column นี้จะปรากฏทางหน้าจอในรูปแบบใดเช่น Text, Checkbox , Pop List เป็นต้น ดังที่แสดงในรูปที่ 3.7

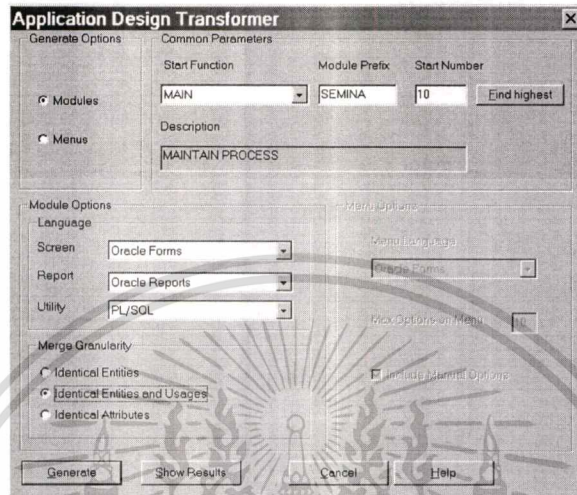


รูปที่ 3.7 แสดงการกำหนดค่า Display Type

ขั้นตอนต่อไป ก็คือจะทำการใช้ Application Design Transformer (ADT) ซึ่งจะทำการแปลง Function และข้อมูลที่ใช้ให้กลายเป็น Application First-cut Design โดยจะมีหน้าจอดังในรูปที่ 3.8 ซึ่งจะให้ผู้ผู้ตั้งค่าว่าต้องการสร้างให้อยู่ในรูปแบบของ Module หรือ Menu และต้องการให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ใน Response แบบ Report, Utility หรือ Screen ที่สร้างขึ้นในรูปแบบของภาษาใด ซึ่งสามารถที่จะสร้างในรูปแบบของภาษาใดหลายภาษา ไม่เฉพาะแต่ภาษาที่ใช้ใน Oracle อย่างเดียวนั่น



รูปที่ 3.8 แสดงหน้าจอ Application Design Transformer (ADT)

เมื่อทำการตั้งค่าและสร้างเรียบร้อยแล้วก็จะได้ Candidate Module แล้วจะทำการเลือก Module ว่า Module ไหนยอมรับได้หรือไม่ได้ ซึ่งในส่วนนี้มียูทิลิตี้ที่ช่วยในการตัดสินใจ โดยการเปลี่ยนใน Property Candidate จาก Yes ให้เป็น No ใน Designer Editor ที่อยู่ในส่วนของ Module Applications

ต่อจากนั้นก็สร้าง User Interface ของแต่ละ Module ออกมา ซึ่งในการสร้าง Form จะมี Dialog Box ที่ใช้ในการเลือกรูปแบบในการสร้าง Form เพื่อให้ผู้ใช้ได้กำหนดค่า โดยส่วนของ Option จะเป็นการกำหนดข้อมูลต่าง ๆ เช่น เมื่อ compile แล้ว Form ที่ได้จะนำไปไว้ที่ Folder ไດ เมื่อทำการกำหนดค่าเรียบร้อยแล้ว และทำการสร้าง Form ขึ้นมาเรียบร้อยแล้ว ก็จะได้ User Interface ที่ต้องการ

บทที่ 4

การวิเคราะห์และออกแบบระบบงาน

ในการวิเคราะห์และออกแบบระบบงานภายในโครงการนี้ ได้มีนำระบบทะเบียนนักศึกษา มาเป็นตัวอย่างในการศึกษาและพัฒนาระบบ โดยมีการวิเคราะห์และออกแบบระบบงานดังนี้

4.1 ระบบทะเบียนนักศึกษา

ระบบทะเบียนนักศึกษาที่นำมาใช้ในการวิเคราะห์และออกแบบระบบงานนี้เป็นส่วนที่เกี่ยวข้องกับสำนักทะเบียนและประมวลผลเป็นส่วนใหญ่ โดยภายในระบบมีการแบ่งการทำงาน ออกเป็น 7 ส่วนดังนี้

4.1.1 ข้อมูลคณะและภาควิชา

ส่วนของข้อมูลคณะและภาควิชาจะมีการจัดการข้อมูลต่าง ๆ ที่เกี่ยวข้องกับคณะ และภาควิชาภายในสถานศึกษา ซึ่งในส่วนคณะและภาควิชา

4.1.2 ข้อมูลหลักสูตร หมวดวิชา และรายวิชา

ส่วนของข้อมูลหลักสูตร หมวดวิชา และรายวิชา นี้จะเป็นส่วนของการจัดการ ข้อมูลหลักสูตรต่าง ๆ ทั้งหมดที่มีอยู่ในสถานศึกษา ซึ่งจะเป็นข้อมูลเกี่ยวกับรายชื่อวิชาต่าง ๆ ที่มีอยู่ในหลักสูตร หมวดวิชาที่อยู่ในแต่ละหลักสูตร จำนวนหน่วยกิตที่มีอยู่ในแต่ละ หลักสูตร แต่ละรายวิชานั้น ๆ มีวิชาบังคับก่อนคือวิชาอะไรบ้าง รวมไปถึงข้อมูลของ รายวิชาที่เปิดสอนในแต่ละภาคการศึกษา วันและเวลาเรียน-สอบ ห้องเรียน ซึ่งข้อมูลส่วน นี้ก็มาจากคณะและภาควิชาเป็นผู้กำหนดนั่นเอง

4.1.3 ข้อมูลนักศึกษา

ในส่วนนี้จะเป็นส่วนของการจัดการข้อมูลนักศึกษา โดยจะจัดการเกี่ยวกับข้อมูล ต่าง ๆ ดังนี้

- ประวัตินักศึกษาต่าง ๆ อย่างเช่น ชื่อ นามสกุล เพศ ภูมิลำเนา วันเดือนปีเกิด อีเมล เป็นต้น
- ข้อมูลที่เกี่ยวข้องกับการศึกษา อย่างเช่น ผู้ที่จบการศึกษามา อาจารย์ที่ปรึกษา คณะและภาควิชาที่สังกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถานภาพต่างๆ เช่น สถานภาพทางการศึกษา ค้างชำระหนี้กับทางสถานศึกษา หรือค้ำหนังสือทางห้องสมุดหรือไม่
- บันทึกข้อมูลเอกสารที่เกี่ยวข้องกับนักศึกษาของแต่ละคน อย่างเช่น เอกสารแต่งตั้งอาจารย์ที่ปรึกษาวิทยานิพนธ์ หรือเอกสารอนุมัติให้ลาพักการศึกษา เป็นต้น

4.1.4 การลงทะเบียน

ในส่วนนี้จะเป็นส่วนของการลงทะเบียน การเพิ่ม-ถอนรายวิชา การรักษาสภาพนักศึกษา การลาพักการศึกษา ซึ่งจะมีขั้นตอนการทำงาน ดังนี้

1. ตรวจสอบสิทธิ์ของนักศึกษาว่า สามารถที่จะทำการลงทะเบียนได้หรือไม่ โดยดูจากสถานภาพการศึกษานักศึกษาจะต้องอยู่ในสถานภาพที่สามารถจะทำการลงทะเบียนได้ โดยที่
 - นักศึกษาที่มีสถานภาพปกติสามารถที่จะทำการลงทะเบียนและลาพักการศึกษาได้ โดยการลาพักการศึกษานั้น สามารถลาได้ไม่เกิน 2 ภาคการศึกษา
 - นักศึกษาที่มีสถานภาพปกติสามารถที่จะทำการลงทะเบียนรักษาสภาพได้ ถ้าลงทะเบียนครบรายวิชาตามกำหนดหลักสูตร แต่ยังไม่สามารถจบได้ อย่างเช่น รอสอบวิทยานิพนธ์ เป็นต้น
 - นักศึกษาที่มีสภาพพินิจ ไม่สามารถที่จะทำการลงทะเบียนรักษาสภาพและลาพักการศึกษาได้
2. ตรวจสอบการค้างชำระเงินและการค้ำหนังสือห้องสมุดว่า นักศึกษายังค้างชำระเงินหรือค้ำหนังสือห้องสมุดหรือไม่ ถ้าไม่มีก็สามารถทำการลงทะเบียนได้ แต่ถ้าตรวจสอบว่า มีการค้ำหนังสือหรือค้างชำระเงิน ก็ไม่สามารถที่จะทำการลงทะเบียนได้
3. ตรวจสอบรายวิชาที่ลงทะเบียนว่า มีเวลาเรียนที่ซ้ำกันหรือไม่ ถ้ามี ไม่สามารถจะทำการลงทะเบียนได้ รวมไปถึงรายวิชาที่ลงทะเบียนนั้น มีรายวิชาต่อเนื่องหรือไม่ ถ้ามี ก็จะตรวจสอบว่า นักศึกษาได้ทำการลงทะเบียนเรียนรายวิชาต่อเนื่องนั้นแล้วหรือยัง ถ้ามีก็สามารถลงทะเบียนเรียนได้ แต่ถ้าไม่มีก็ไม่สามารถลงทะเบียนรายวิชานั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในการลงทะเบียนต้องลงทะเบียนตามจำนวนหน่วยกิตที่กำหนดเอาไว้ในหลักสูตร เช่น ระดับปริญญาตรีลงได้ 12-22 หน่วยกิต หรือระดับบัณฑิตวิทยาลัยลงได้ตั้งแต่ 9-15 หน่วยกิต ยกเว้นมีคำสั่งจากทางสถานศึกษา หรือจำนวนหน่วยกิตที่ลงทะเบียนเหลือน้อยกว่าที่ได้กำหนดเอาไว้ เป็นต้น

สำหรับการเพิ่มรายวิชานั้น สามารถที่จะทำการเพิ่มได้ โดยที่รายวิชาที่เพิ่มนั้นไม่มีเวลาเรียนซ้ำกับรายวิชาที่ได้ลงทะเบียนเรียนไปเรียบร้อยแล้ว รวมไปถึงจำนวนหน่วยกิตที่ได้ลงทะเบียนเรียบร้อยแล้วกับจำนวนหน่วยกิตที่เพิ่มเข้ามา จะต้องไม่เกินกว่าจำนวนหน่วยกิตที่ได้กำหนดเอาไว้

สำหรับการถอนรายวิชานั้น ถ้าถอนรายวิชาภายในระยะที่ได้กำหนดเอาไว้ว่าจะไม่บันทึกสัญลักษณ์ W ก็จะไม่ปรากฏข้อมูลในส่วนของ การลงทะเบียน แต่ถ้าถอนรายวิชาภายในระยะที่กำหนดโดยบันทึกสัญลักษณ์ W ผลการเรียนของนักศึกษาคนนั้นจะบันทึกสัญลักษณ์ W แทนที่ โดยจะไม่นำรายวิชาที่บันทึกสัญลักษณ์ W มาทำการคำนวณผลการเรียนเฉลี่ย ในกรณีที่นักศึกษามาถอนรายวิชาหลังจากกำหนดระยะเวลาไว้ ก็ไม่สามารถที่จะทำการถอนได้อีกแล้ว

เมื่อได้ทำการลงทะเบียนเรียบร้อยแล้ว ข้อมูลการลงทะเบียนส่วนนี้จะถูกส่งไปให้ส่วนของการเงิน เพื่อทำการคิดเงินที่นักศึกษาจะต้องไปชำระเงินต่อไป

4.1.5 การเงิน

ในส่วนนี้ จะเป็นส่วนของการคิดเงิน เพื่อที่จะออกไปชำระเงินให้แก่นักศึกษาเพื่อไปชำระเงินต่อไป โดยจะนำข้อมูลจากส่วนของการลงทะเบียนเรียนวิชาต่าง ๆ รวมทั้งค่าธรรมเนียมต่าง ๆ ที่นักศึกษาจะต้องใช้ตามสังกัดคณะหรือภาควิชา ซึ่งค่าธรรมเนียมต่าง ๆ ในแต่ละคณะหรือภาควิชา นั้นจะไม่เหมือนกัน รวมไปถึงการชำระเงินเพื่อออกเอกสารต่าง ๆ ตามใบคำร้อง อย่างเช่น ทรานสคริปต์หรือใบรับรองต่าง ๆ เป็นต้น

4.1.6 การประมวลผล

ในส่วนของการประมวลผล จะแบ่งการทำงานออกเป็น 2 ขั้นตอน ก็คือ การบันทึกผลการเรียนและการปรับสถานภาพนักศึกษา

- การบันทึกผลการเรียน

ข้อมูลผลการเรียนนี้จะได้มาจากคณะและภาควิชา โดยจะทำการบันทึกลงไปในแต่ละรายวิชาที่มีการแจ้งผลการเรียนเข้ามา โดยการบันทึกผลการเรียนนี้

เป็นส่วนที่สำคัญมาก จะผิดพลาดไม่ได้ต้องมีการตรวจสอบความถูกต้องก่อน ถึง

จะทำการบันทึกลงไปได้ สำหรับผลการเรียนที่ได้สัญลักษณ์ I จะสามารถเปลี่ยนแปลงแก้ไขผลการเรียนที่หลังได้

● **การปรับสถานภาพของนักศึกษา**

เมื่อมีการบันทึกผลการเรียนเรียบร้อยแล้ว ก็จะทำการคำนวณหาผลการเรียนเฉลี่ยประจำภาคและผลการเรียนเฉลี่ยสะสมของนักศึกษาแต่ละคนที่ได้ลงทะเบียนไว้ในภาคการศึกษาปัจจุบัน เมื่อคำนวณผลการเรียนเฉลี่ยเรียบร้อยแล้ว ก็จะทำการปรับสถานภาพทางการศึกษา ซึ่งจะพิจารณาจากระดับหลักสูตรที่นักศึกษา โดยแบ่งได้เป็น 2 ประเภท คือ ระดับปริญญาตรีและระดับบัณฑิตศึกษา

1. **ระดับปริญญาตรี** จะมีการปรับสถานภาพทางการศึกษาและนักศึกษาดังนี้

- ถ้าได้ผลการเรียนเฉลี่ยสะสมมากกว่า 2.00 ให้ปรับสถานภาพทางการศึกษาเป็น “ปกติ” (Normal)
- ถ้าได้ผลการเรียนเฉลี่ยสะสมน้อยกว่า 1.00 ให้ปรับสถานภาพทางการศึกษาเป็น “พ้นสภาพ” (Retired)
- ถ้าได้ผลการเรียนเฉลี่ยสะสมน้อยกว่า 2.00 แต่มากกว่า 1.00 ให้ปรับสถานภาพทางการศึกษาเป็น “รอพินิจ” (Probation)
- กรณีที่นักศึกษามีสถานภาพ “รอพินิจ” และได้ผลการเรียนได้ผลการเรียนเฉลี่ยประจำภาคน้อยกว่า 2.00 ให้ปรับสถานภาพทางการศึกษาเป็น “พ้นสภาพ” (Retired)
- กรณีที่นักศึกษามีสถานภาพ “รอพินิจ” แต่ได้ผลการเรียนได้ผลการเรียนเฉลี่ยประจำภาคมากกว่า 2.00 ให้คงสถานภาพทางการศึกษาเป็น “รอพินิจ” (Probation) เอาไว้

2. **ระดับบัณฑิตศึกษา** จะมีการปรับสถานภาพทางการศึกษาและนักศึกษาดังนี้

- ถ้าได้ผลการเรียนเฉลี่ยสะสมมากกว่า 3.00 ให้ปรับสถานภาพทางการศึกษาเป็น “ปกติ” (Normal)
- ถ้าได้ผลการเรียนเฉลี่ยสะสมน้อยกว่า 2.50 ให้ปรับสถานภาพทางการศึกษาเป็น “พ้นสภาพ” (Retired)
- ถ้าได้ผลการเรียนเฉลี่ยสะสมน้อยกว่า 3.00 แต่มากกว่า 2.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานให้ปรับสถานภาพทางการศึกษาเป็น “รอพินิจ” (Probation) การค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีที่นักศึกษามีสถานภาพ “รอพินิจ” และได้ผลการเรียน ได้ผลการเรียนเฉลี่ยประจำภาคน้อยกว่า 3.00 ให้ปรับสถานภาพทางการศึกษาเป็น “พ้นสภาพ”
- กรณีที่นักศึกษามีสถานภาพ “รอพินิจ” แต่ได้ผลการเรียน ได้ผลการเรียนเฉลี่ยประจำภาคมากกว่า 3.00 ให้คงสถานภาพทางการศึกษาเป็น “รอพินิจ” เอาไว้

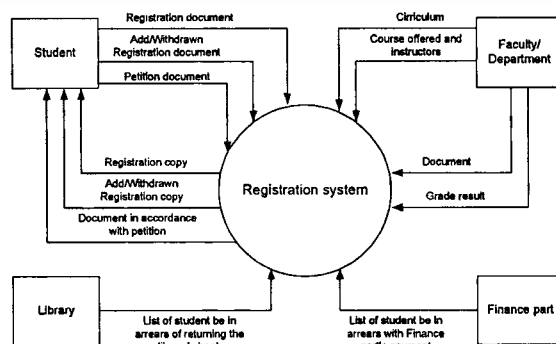
4.1.7 รายงาน

ในส่วนนี้จะเป็นการสร้างรายงานขึ้นมาจากส่วนของข้อมูลต่าง ๆ ทั้ง 6 ส่วนที่ได้กล่าวมาแล้วข้างต้น โดยจะแบ่งเป็นรายงานต่าง ๆ ได้อย่างคร่าว ๆ ดังนี้

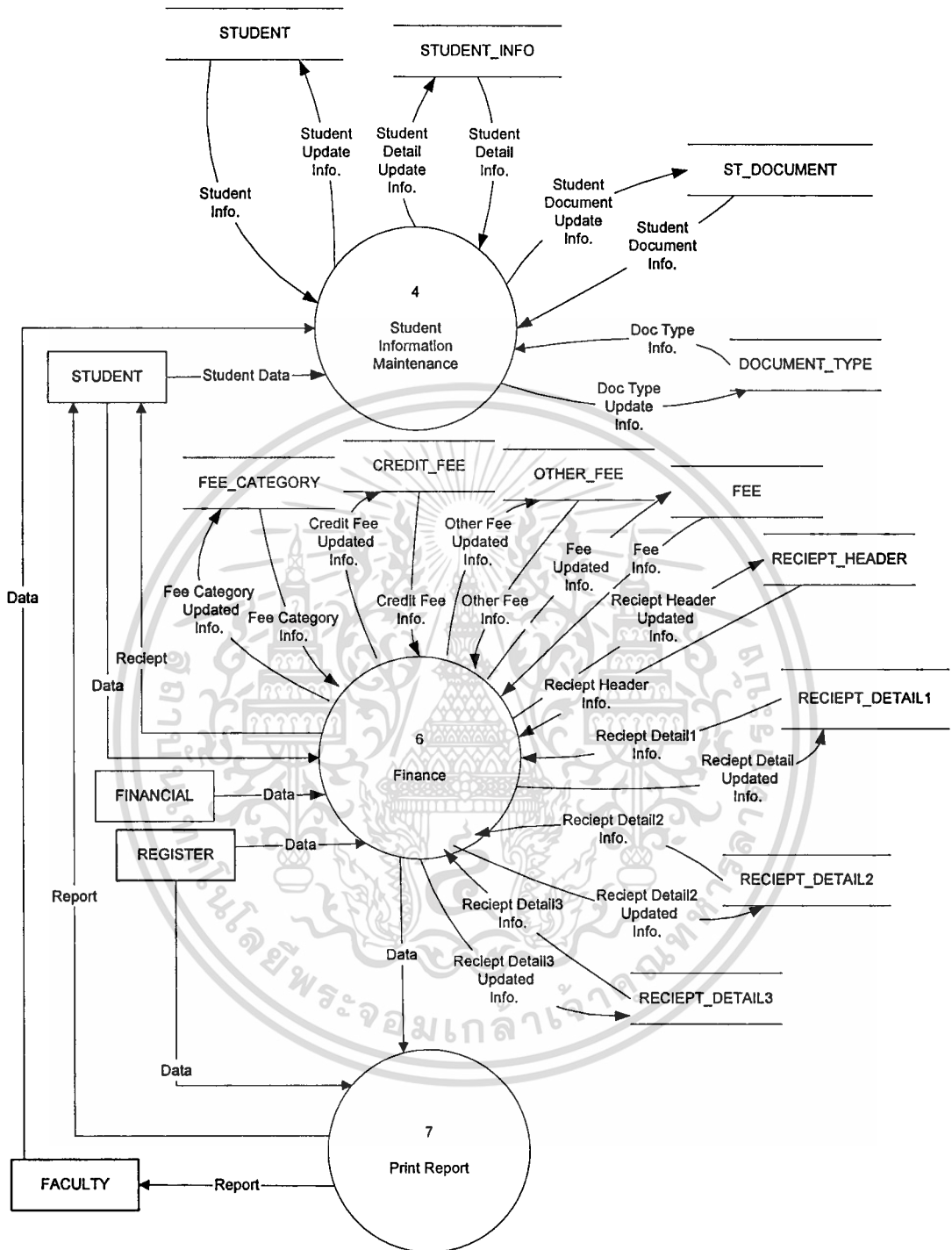
- รายงานที่เกี่ยวข้องกับการเรียนการสอน อย่างเช่น รายชื่อนักศึกษาที่มีสิทธิเข้าเรียน รายงานแสดงรายชื่อนักศึกษาที่มีสิทธิเข้าสอบ เป็นต้น
- รายงานที่ข้องกับผลการเรียน อย่างเช่น ใบแจ้งผลการเรียน ทรานสคริปต์ เป็นต้น
- รายงานอื่น ๆ อย่างเช่น ใบรับรองการเป็นนักศึกษา เป็นต้น

4.2 Dataflow Diagram ของระบบทะเบียนนักศึกษา

ในการศึกษาขั้นตอนการทำงานของระบบทะเบียนสามารถทำการสรุปการทำงานออกมาโดยใช้แผนภาพ Dataflow Diagram เพื่อที่จะทำให้เข้าใจถึงภาพรวมของการทำงานของระบบทะเบียนนักศึกษาได้มากขึ้น โดยจะแสดง Context Diagram ของระบบทะเบียนไว้ในรูปที่ 4.1 Dataflow Diagram ระดับที่ 1 จะแสดงเฉพาะในส่วนของข้อมูลนักศึกษา การเงิน และการออกรายงานต่าง ๆ ดังในรูปที่ 4.2 และ Dataflow ระดับที่ 2 ของแต่ละส่วนจะแสดงไว้ในรูปที่ 4.3-4.5 ตามลำดับ

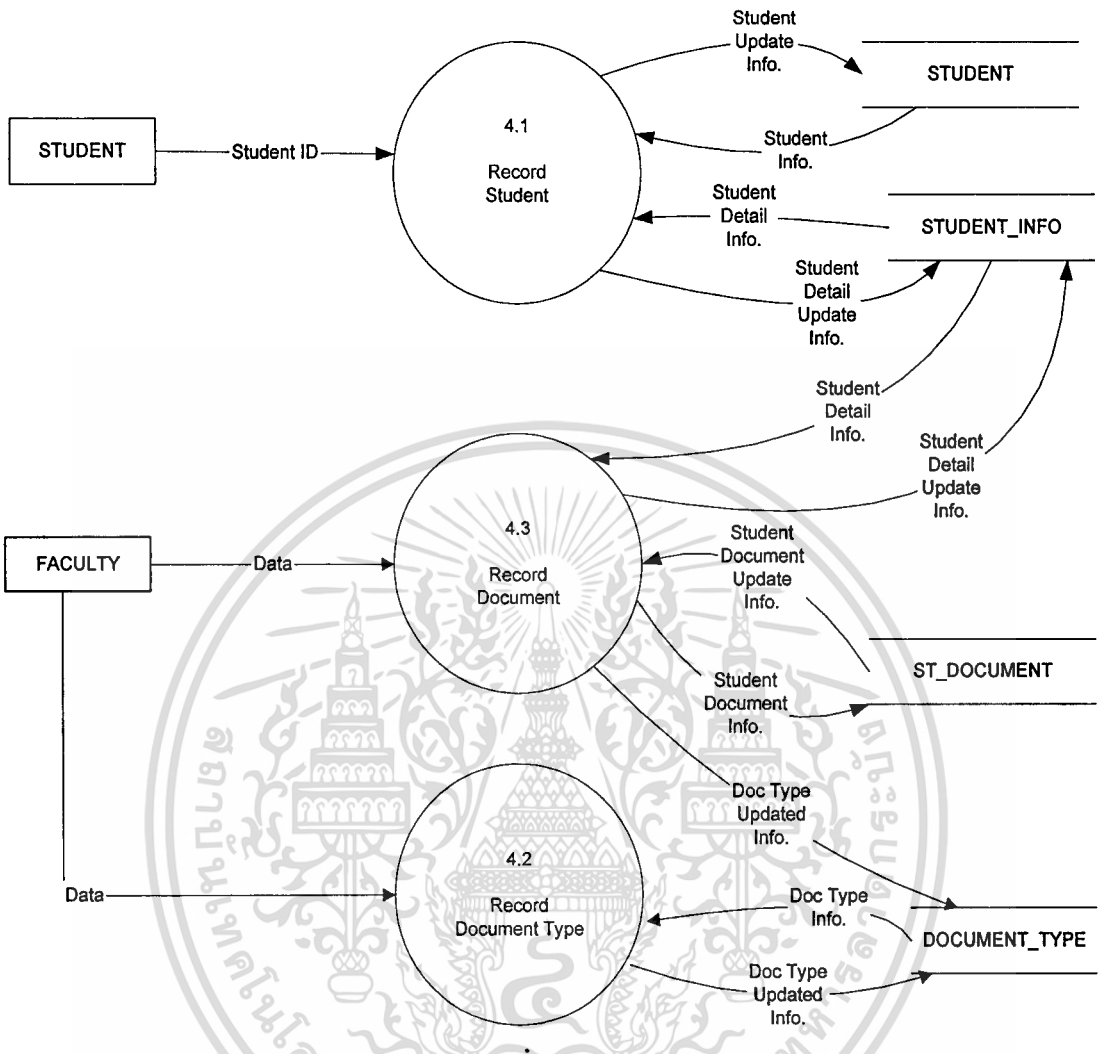


เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.1 Context Diagram ของระบบทะเบียนนักศึกษาใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



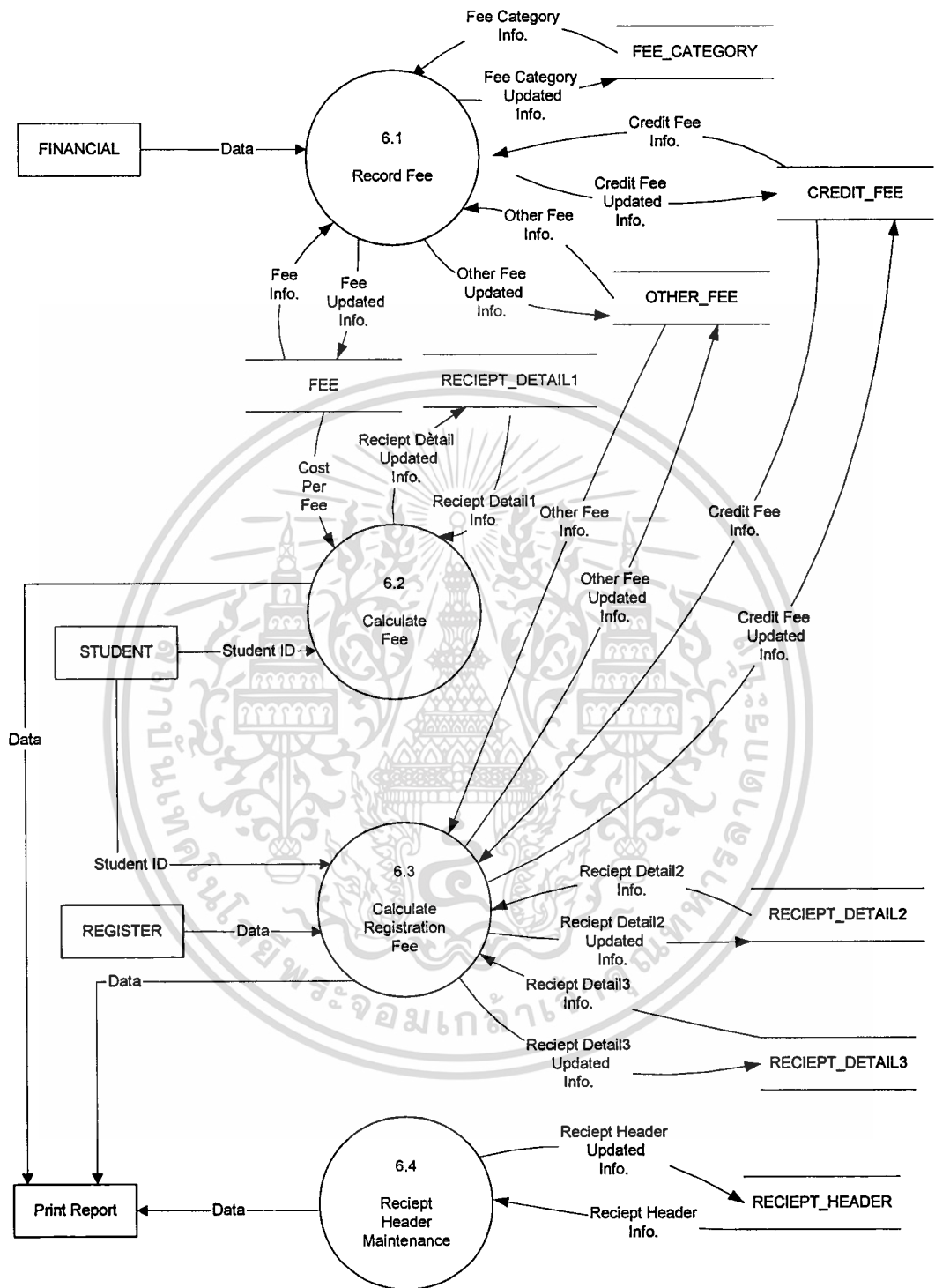
รูปที่ 4.2 Dataflow Diagram ระดับที่ 1 ของระบบทะเบียนนักศึกษาส่วนของข้อมูลนักศึกษา
การเงิน และการออกรายงานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



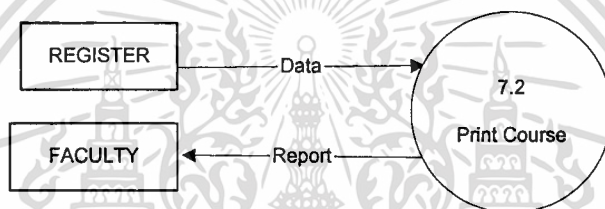
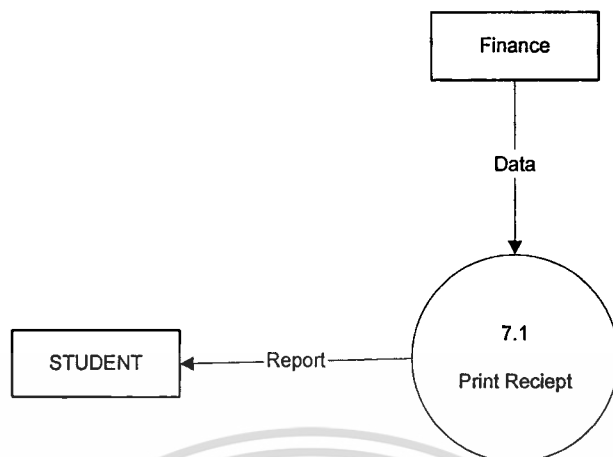
รูปที่ 4.3 Dataflow Diagram ระดับที่ 2 ของระบบทะเบียนนักศึกษาในส่วนของ ข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Dataflow Diagram ระดับที่ 2 ของระบบทะเบียนนักศึกษาในส่วนของ
การเงิน

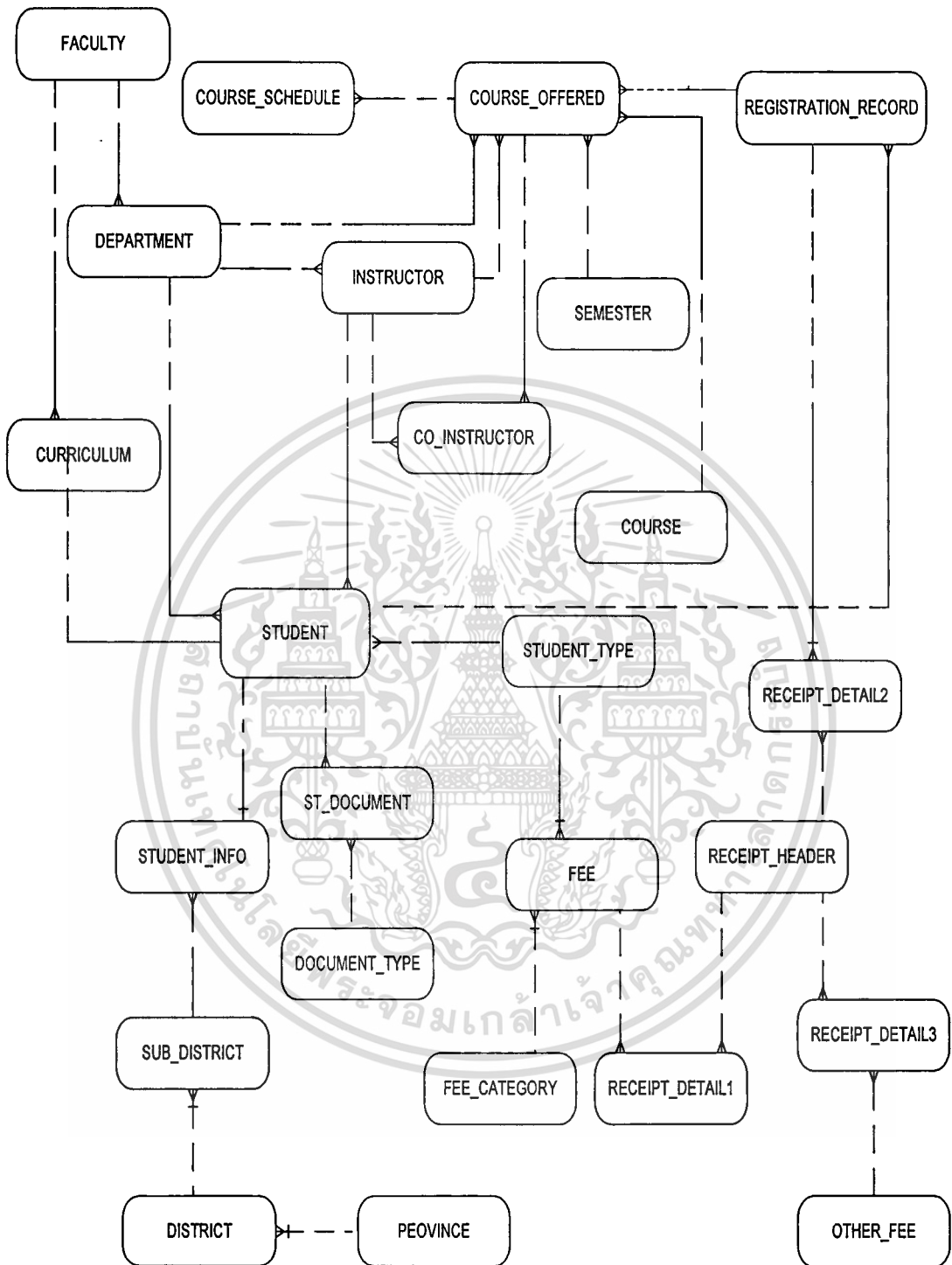
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Dataflow Diagram ระดับที่ 2 ของระบบทะเบียนนักศึกษาในส่วนของ การออกรายงานต่าง ๆ

4.3 แบบจำลองแสดงความสัมพันธ์ของข้อมูล (Entity Relationship Diagram)

ข้อมูลที่เกี่ยวข้องในระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา การเงิน และการออกรายงานต่าง ๆ ที่สัมพันธ์กันนั้นจะสามารถแสดงความสัมพันธ์ออกมาในรูปแบบของแผนภาพ ER-Diagram ดังที่แสดงในรูปด้านล่าง



รูปที่ 4.6 ER-Diagram ของระบบทะเบียนนักศึกษาในส่วนจากระบบทะเบียนนักศึกษาส่วนของ ข้อมูลนักศึกษา การเงิน และการออกรายงานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.	FK	ST_DEPT_ID	Varchar2	6	Department Id
4.	FK	ST_CURR_ID	Varchar2	6	Curriculum Id
5.	FK	ST_TYPE_ID	Varchar2	2	Student Type Id
6.	FK	ST_INST_ID	Varchar2	10	Instructor Id
7.		ST_TTITLE	Varchar2	10	Student Title
8.		ST_F_NAME_T	Varchar2	30	Student First name in Thai
9.		ST_L_NAME_T	Varchar2	30	Student Last name in Thai
10.		ST_GENDER	Varchar2	2	Gender
11.		ST_DOB	DATE		Date of Birth
12.		ST_E_MAIL	Varchar2	30	E-mail Address
13.		ST_ADM_DATE	DATE		Date of Admission
14.		ST_GRAD_DATE	DATE		Date of Graduated
15.		ST_ACC_CR_A	NUMBER	3,0	Academic Credit Acquired
16.		ST_ACC_CR_E	NUMBER	3,0	Academic Credit Entry
17.		ST_GPA	NUMBER	4,3	GPA
18.		ST_ACC_STATUS	CHAR	2	Academic Status
19.		ST_FIN_STATUS	CHAR	1	Financial Status
20.		ST_LIB_STATUS	CHAR	1	Library Status

ตารางที่ 4.4 ตาราง STUDENT_TYPE

STUDENT_TYPE					
ประเภทของนักศึกษา					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	ST_TYPE_ID	Varchar2	15	Id Of Student Type
2.		ST_TYPE_NAME	Varchar2	30	Detail Of Student Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 ตาราง STUDENT_INFO

STUDENT_INFO					
ข้อมูลนักศึกษา					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	ST_ID	Varchar2	15	Student Id
2.		ST_F_NAME_E	Varchar2	30	Student First name in English
3.		ST_L_NAME_E	Varchar2	30	Student Last name in English
4.		ST_ADDR_NO	Varchar2	2	Address No.
5.		ST_ADDR_LANE	Varchar2	30	Address street.
6.		ST_ADDR_STR	Varchar2	30	Address Soi.
7.		ST_PHONE	Varchar2	9	Phone No Of Studernt
8.		ST_PHOTO	Varchar2	100	Photo Of Student

ตารางที่ 4.6 ตาราง SUB_DISTRICT

SUB_DISTRICT					
ตำบล					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	S_DSTR_ID	Varchar2	15	Id Of Sub District
2.		S_DSTR_T	Varchar2	30	Sub District name in Thai
3.		S_DSTR_E	Varchar2	30	Sub District name in English

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ตาราง DISTRICT

DISTRICT					
อำเภอ					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	DSTR_ID	Varchar2	15	Id Of District
2.		DSTR_T	Varchar2	30	District name in Thai
3.		DSTR_E	Varchar2	30	District name in English

ตารางที่ 4.8 ตาราง PROVINCE

PROVINCE					
จังหวัด					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	PROV_ID	Varchar2	15	Id Of Province
2.		PROV_T	Varchar2	30	Province name in Thai
3.		PROV_E	Varchar2	30	Province name in English

ตารางที่ 4.9 ตาราง ST_DOCUMENT

ST_DOCUMENT					
เอกสารเกี่ยวกับนักศึกษา					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	DOC_ID	Varchar2	15	Id Of Document
2.		DOC_DATE	Varchar2	30	Date
3.		DOC_DATA	Varchar2	30	Detail Of Document

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 ตาราง DOCUMENT_TYPE

DOCUMENT_TYPE					
ประเภทของเอกสาร					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	DOC_TYPE_ID	Varchar2	15	Id Of Document Type
2.		DOC_TYPE_DESC	Varchar2	30	Description Of Document Type

ตารางที่ 4.11 ตาราง INSTRUCTOR

INSTRUCTOR					
ข้อมูลอาจารย์					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	INST_ID	Varchar2	10	Instructor Id
2.		INST_NAME_T	Varchar2	40	Instructor Name in Thai
3.		INST_NAME_E	Varchar2	40	Instructor Name in English
4.	FK	INST_FAC_ID	Varchar2	6	Faculty Id
5.	FK	INST_DEPT_ID	Varchar2	6	Department Id

ตารางที่ 4.12 ตาราง CURRICULUM

CURRICULUM					
ข้อมูลหลักสูตร					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	CURR_ID	Varchar2	6	Curriculum Id
2.		CURR_FAC_ID	Varchar2	6	Faculty Id
3.		CURR_NAME_T	Varchar2	60	Curriculum Name in Thai
4.		CURR_NAME_E	Varchar2	40	Curriculum Name in English
5.		CURR_DEG_NAME_T	Varchar2	60	Degree Name in Thai

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.		CURR_DEG_NAME_E	Varchar2	40	Degree Name in English
7.		CURR_LEVEL	CHAR	1	Curriculum Level

ตารางที่ 4.13 ตาราง COURSE_OFFERED

COURSE_OFFERED					
ข้อมูลวิชาที่เปิดสอน					
Seq.	Key	Name	Data Type	Len.	Comment
1	PK	CO_YEAR	NUMBER	40	Year

38

4.4 รายละเอียดของข้อมูลที่จัดเก็บในระบบ

รายละเอียดต่าง ๆ ของข้อมูลภายใน ER-Diagram ที่แสดงในข้างต้นจะมีรายละเอียดต่าง ๆ ในแต่ละตาราง ดังนี้

ตารางที่ 4.1 ตาราง FACULTY

FACULTY					
ข้อมูลคณะ					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	FAC_ID	Varchar2	6	Faculty Id
2.		FAC_NAME_E	Varchar2	40	Faculty Name in English
3.		FAC_NAME_T	Varchar2	60	Faculty Name in Thai

ตารางที่ 4.2 ตาราง DEPARTMENT

DEPARTMENT					
ข้อมูลภาควิชา					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	DEPT_FAC_ID	Varchar2	6	Faculty Id
2.	PK	DEPT_ID	Varchar2	6	Department Id
3.		DEPT_NAME_E	Varchar2	40	Department Name in English
4.		DEPT_NAME_T	Varchar2	60	Department Name in Thai

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการอ้างอิงเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ตาราง STUDENT

STUDENT					
ข้อมูลนักศึกษา					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	ST_ID	Varchar2	15	Student Id
2.	FK	ST_FAC_ID	Varchar2	6	Faculty Id

7.		CS_TIME_END	TIME		Finished Time of Course Offered
8.		CS_MID_DATE	DATE		Midterm Exam Date
9.		CS_MID_TIME_START	TIME		Midterm Exam Start Time
10.		CS_MID_TIME_END	TIME		Midterm Exam Finished Time
11.		CS_FINAL_DATE	DATE		Final Exam Date
12.		CS_FINAL_TIME_START	TIME		Final Exam Start Time
13.		CS_FINAL_TIME_END	TIME		Finished Time of Final Exam

ตารางที่ 4.15 ตาราง SEMESTER

SEMESTER ข้อมูลวิชาที่เปิดสอน					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	CO_YEAR	NUMBER	4,0	Year
2.	PK	CO_SEMESTER	NUMBER	2,0	Semester
3.	PK	SE_START	DATE		Start Date of Semester
4.	PK	SE_END	DATE		End Date of Semester
5.		SE_ADD_LIMIT	DATE		Due Date of Add Course
6.	FK	SE_W_LIMIT	DATE		Due Date of Withdrawn
7.	FK	SE_DROP_LIMIT	DATE		Due Date of Drop Course

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.16 ตาราง REGISTRATION_RECORD

REGISTRATION_RECORD					
ข้อมูลการลงทะเบียนเรียน					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	CO_YEAR	NUMBER	4,0	Year
2.	PK	CO_SEMESTER	NUMBER	2,0	Semester
3.	PK	CO_C_ID	Varchar2	12	Course Id
4.	PK	CO_SECTION	NUMBER	2,0	Section
5.	FK	RR_ST_ID	Varchar2	15	Student Id
6.		RR_CR	NUMBER	2,0	Credit of Course
7.		RR_GRADE	Varchar2	2	Grade
8.		RR_DATE	DATE		Registration Date
9.		RR_TX	Varchar2	1	Transfer Course Mark
10.		RR_USER	Varchar2	15	Operator Name
11.		RR_RD2_ID	NUMBER	10,0	Receipt No.

ตารางที่ 4.17 ตาราง FEE

FEE					
ค่าธรรมเนียม					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	FEE_COST	Varchar2	15	Cost Of Fee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.18 ตาราง FEE_CATEGORY

FEE_CATEGORY ประเภทของค่าธรรมเนียม					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	FC_ID	Varchar2	15	Id Of Fee
2.		FC_NAME	Varchar2	30	Description Of Fee

ตารางที่ 4.19 ตาราง RECEIPT_HEADER

RECEIPT_HEADER ประเภทของใบเสร็จ					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	RH_ID	Varchar2	15	Id Of Receipt Header
2.		RH_DATE	Varchar2	30	Date Of Receipt Header
3.		RH_QTY	Varchar2	30	Quaintly Of Receipt Header

ตารางที่ 4.20 ตาราง RECEIPT_DETAIL1

RECEIPT_DETAIL1 ใบเสร็จค่าธรรมเนียม					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	RD1_ID	Varchar2	15	Id Of Receipt Detail1
2.		RD1_QTY	Varchar2	30	Quaintly Of Receipt Detail1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.21 ตาราง RECEIPT_DETAIL2

RECEIPT_DETAIL2					
ใบเสร็จค่าลงทะเบียน					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	RD2_ID	Varchar2	15	Id Of Receipt Detail2
2.		RD2_QTY	Varchar2	30	Quaintly Of Receipt Detail2

ตารางที่ 4.22 ตาราง RECEIPT_DETAIL3

RECEIPT_DETAIL3					
ใบเสร็จค่าอื่น ๆ					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	RD3_ID	Varchar2	15	Id Of Receipt Detail3
2.		RD3_QTY	Varchar2	30	Quaintly Of Receipt Detail3

ตารางที่ 4.23 ตาราง OTHER_FEE

OTHER_FEE					
ประเภทของค่าอื่น ๆ					
Seq.	Key	Name	Data Type	Len.	Comment
1.	PK	OF_ID	Varchar2	15	Id Of Other Fee
2.		OF_NAME	Varchar2	30	Name Of Other Fee
3.		OF_COST	Varchar2	30	Cost Of Other Fee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การพัฒนาระบบงาน

ในการพัฒนาระบบงานโดยใช้ Oracle Designer6i นั้นจะมีการแบ่งการพัฒนาออกเป็น 2 ส่วนใหญ่ๆ คือ การพัฒนาในส่วนของฐานข้อมูล และการพัฒนาในส่วนของโปรแกรมการใช้งาน ดังต่อไปนี้

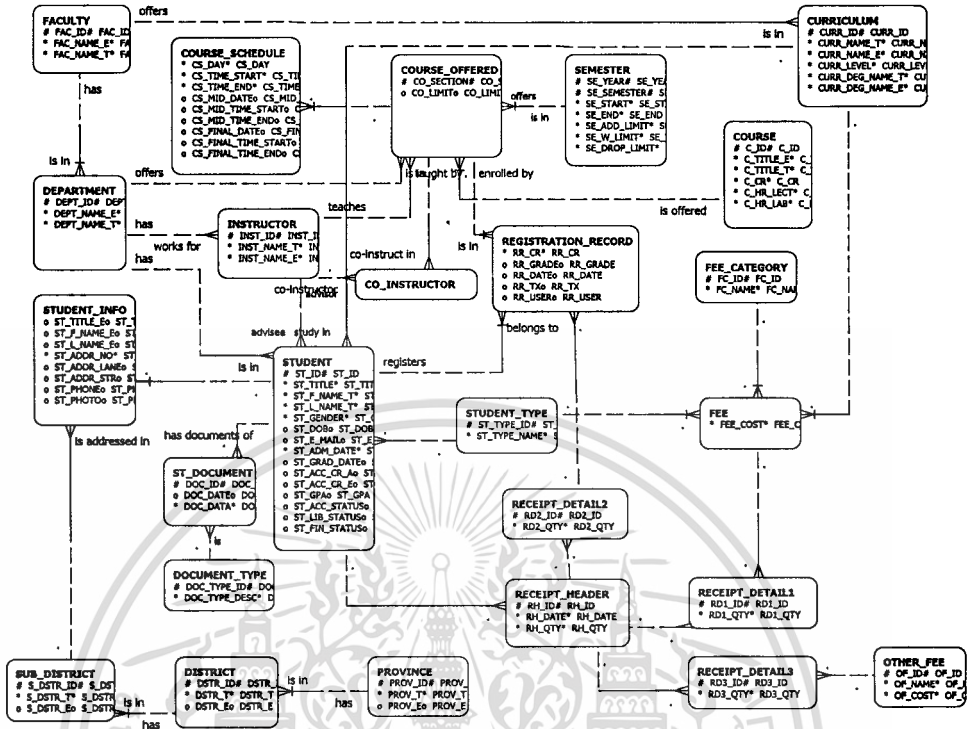
5.1 การพัฒนาระบบงานส่วนของฐานข้อมูล

ในการพัฒนาระบบงานส่วนของฐานข้อมูลนี้ Oracle Designer6i ได้มีการใช้ Tools ที่ชื่อว่า Entity Relationship Diagrammer (ERD) เพื่อใช้ในการสร้าง Entity Relationship Diagram เพื่อใช้ในการสร้างฐานข้อมูลของระบบ เมื่อจากนั้นจะใช้ Database Design Transformer (DDT) เพื่อทำให้ Entity Relationship Diagram อยู่ในรูปแบบ first-cut design หลังจากนั้น ก็ใช้ Design Editor ทำการสร้าง Server Model Diagram ขึ้นมา เพื่อให้ผู้ใช้สามารถเข้าใจและปรับแต่งตารางที่จะนำไปทำการสร้างเป็นฐานข้อมูลจริง ๆ ขึ้นมา เมื่อมีการปรับแต่ง Server Model Diagram เรียบร้อยแล้ว ก็พร้อมที่จะนำไปสร้างเป็นฐานข้อมูลจริงที่ใช้ในการพัฒนาระบบ โดยใช้ Function Generate Database from Server Model ในการสร้างฐานข้อมูลในรูปแบบที่ผู้ใช้ต้องการ จากที่ได้กล่าวมานั้น การพัฒนาระบบงานส่วนของฐานข้อมูล สามารถกำหนดเป็นขั้นตอนการทำงานอย่างละเอียดได้ ดังนี้

5.1.1 การสร้าง Entity Relationship Diagram

สำหรับการสร้าง Entity Relationship Diagram ใน Oracle Designer6i จะใช้ Entity Relationship Diagrammer ในการสร้าง โดยใน Entity Relationship Diagram จะทำการสร้าง Entity และ Relationship ขึ้นมา ภายใน Entity ก็จะมีการสร้าง Attribute ขึ้นมา

โดย Entity Relationship Diagram ของระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา การเงิน และการออกรายงานต่าง ๆ นั้นมีดังรูปด้านล่างนี้



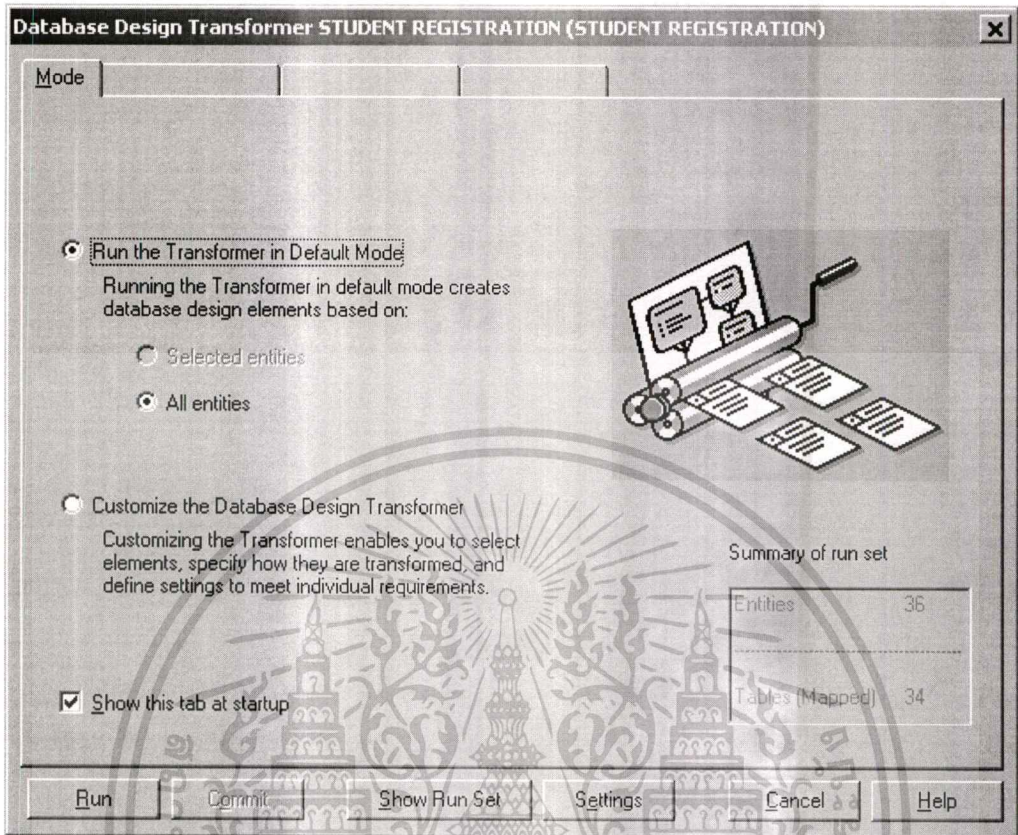
รูปที่ 5.1 Entity Relationship Diagram ของระบบทะเบียนนักศึกษา

5.1.2 การใช้ Database Design Transformer

การใช้ Database Design Transformer (DDT) นี้ ทำเพื่อที่จะทำการแปลง Entity Relationship Diagram ให้อยู่ในรูปแบบของ First-cut Data Design หรือรูปแบบที่สามารถนำไปสร้างเป็นฐานข้อมูลได้นั่นเอง ซึ่ง DDT จะทำการแปลง Entity Relationship Diagram ให้อยู่ในรูปแบบที่สามารถสร้างฐานข้อมูลได้ โดยใช้หลักเกณฑ์ในการแปลงข้อมูลดังนี้

- Entity จะถูกแปลงเป็น Table
- UID จะถูกแปลงเป็น primary key
- Attribute จะถูกแปลงเป็น column
- Relationship จะถูกแปลงเป็น Foreign key และ constraint ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

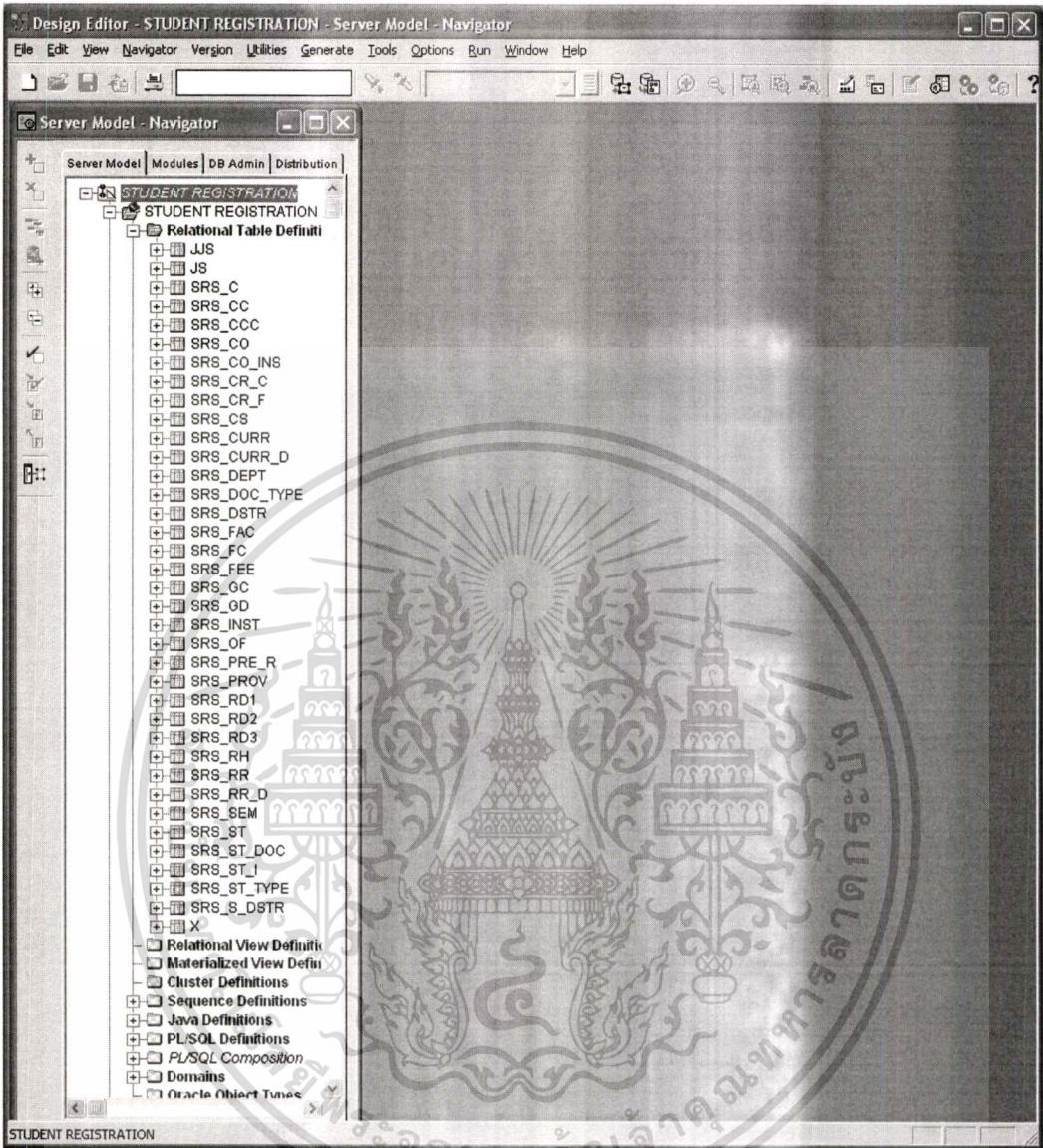


รูปที่ 5.2 Database Design Transformer

ผลลัพธ์ที่ได้จากการใช้ DDT นั้นจะได้ออกมาในรูปแบบของตารางและคอลัมน์ต่าง ๆ ซึ่งอยู่ในรูปของ first-cut data design แล้ว

5.1.3 สร้าง Server Model Diagram

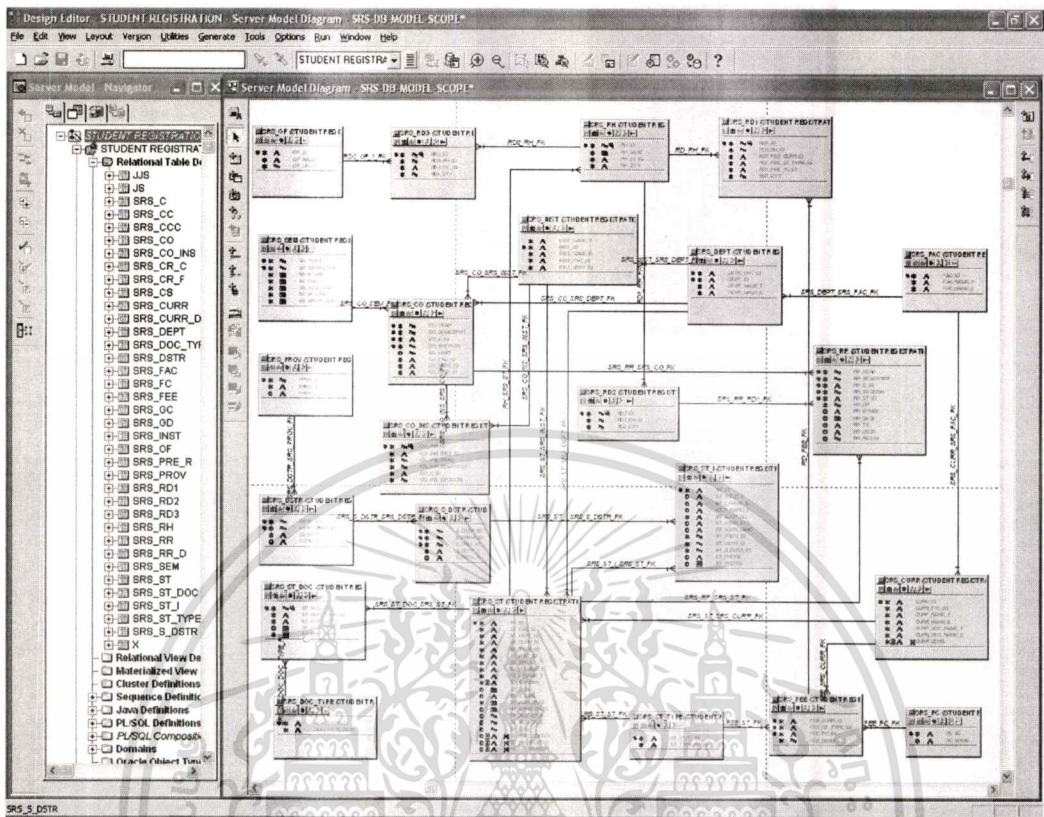
เมื่อได้ข้อมูลที่อยู่ในรูปแบบ first-cut data design แล้ว จะนำผลลัพธ์ที่ได้มาทำการสร้าง Server Model Diagram ขึ้นมาเพื่อให้เข้าใจได้ง่ายและปรับแต่งตารางต่าง ๆ ตามที่เรารต้องการ โดยใช้ Design Editor ดังรูปด้านล่าง



รูปที่ 5.3 Design Editor

จากรูปที่ 5.3 นั้นทางด้านซ้ายมือ จะเห็นชื่อ Table ต่าง ๆ ที่ได้สร้างไว้ซึ่งได้จากการใช้ DDT ในการสร้างนั่นเอง หลังจากนั้นจะทำ Server Model Diagram ได้โดยการเลือกตารางที่ต้องการจะสร้างฐานข้อมูล แล้วมาลากไว้ในพื้นที่ว่าง ๆ ทางด้านขวามือ ก็จะเป็น Server Model Diagram ซึ่งสามารถที่จะปรับแต่งเพิ่มเติม แก้ไขได้ตามต้องการ ดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

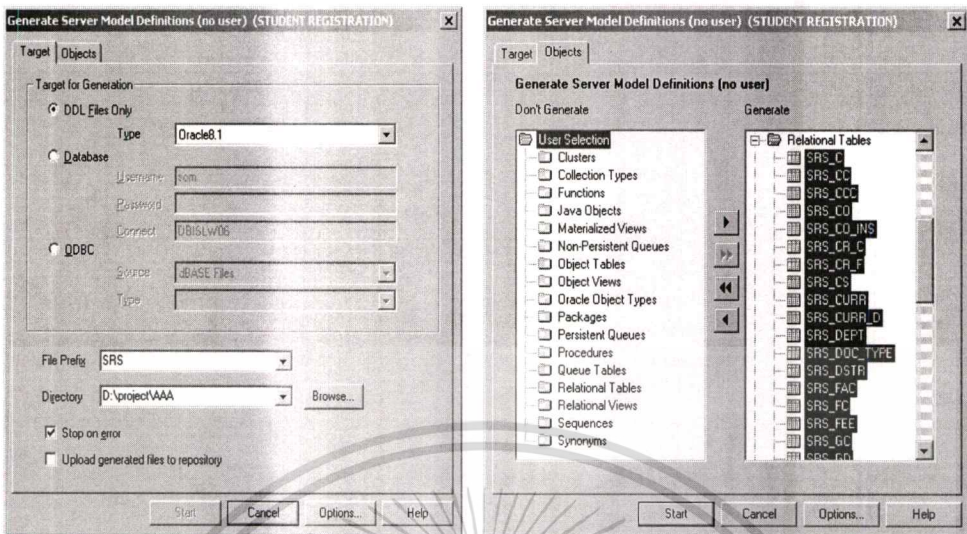


รูปที่ 5.4 Server Model Diagram

5.1.4 การสร้างฐานข้อมูล

หลังจากทำการปรับแต่ง Server Model Diagram แล้วก็จะทำการสร้างฐานข้อมูลจริงที่ใช้งานขึ้น ซึ่งเป็นขั้นตอนสุดท้ายในการพัฒนาระบบงานในส่วนของฐานข้อมูล โดยในการสร้างฐานข้อมูลจะใช้ Function Generate Database from Server Model ใน Design Editor ดังรูปที่ 5.5 ทางด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 Function Database from Server Model

จากรูปที่ 5.5 ทางซ้ายมือในส่วนของ Target Generation จะเป็นส่วนที่ให้เลือกว่าต้องการที่จะสร้างฐานข้อมูลในแบบใด โดยอาจจะสร้างเป็น Script File ก็ได้ ซึ่งจะได้ออกมาในรูปแบบดังรูปที่ 5.6 ทางด้านล่าง หรือจะสร้างลงไปในฐานข้อมูลจริง ๆ ก็ได้ ซึ่งการสร้างฐานข้อมูลนี้ ไม่ได้จำเพาะจะต้องเป็น Oracle เพียงอย่างเดียวเท่านั้น สามารถที่จะใช้กับผลิตภัณฑ์ของบริษัทอื่นๆ ได้เช่นเดียวกัน ทำให้เห็นว่า เราออกแบบเพียงครั้งเดียวสามารถใช้งาน ได้หลากหลาย และเพิ่มความสะดวกให้แก่ผู้พัฒนาระบบได้เป็นอย่างดี

```

SRS.tab - Notepad
File Edit Format Help
-- D:\project\AAA\SRS.tab
-- Generated for Oracle 8.1 on Thu Dec 26 02:19:20 2002 by Server Gener

PROMPT Creating Table 'SRS_ST_DOC'
CREATE TABLE SRS_ST_DOC
(ST_DOC_ID NUMBER(10,0) NOT NULL
,ST_DOC_TYPE_ID VARCHAR2(6) NOT NULL
,ST_DOC_ST_ID VARCHAR2(15) NOT NULL
,ST_DOC_DATE DATE
,ST_DOC_DATA BLOB NOT NULL
)

COMMENT ON COLUMN SRS_ST_DOC.ST_DOC_ID IS 'Document Number ID'

COMMENT ON COLUMN SRS_ST_DOC.ST_DOC_TYPE_ID IS 'Document Type Identifier'

COMMENT ON COLUMN SRS_ST_DOC.ST_DOC_ST_ID IS 'Student Identifier'

COMMENT ON COLUMN SRS_ST_DOC.ST_DOC_DATE IS 'Document Date'

COMMENT ON COLUMN SRS_ST_DOC.ST_DOC_DATA IS 'Document Data'

PROMPT Creating Table 'SRS_DEPT'
CREATE TABLE SRS_DEPT

```

รูปที่ 5.6 Script File ที่ได้จาก Function Generate Database from Server Model

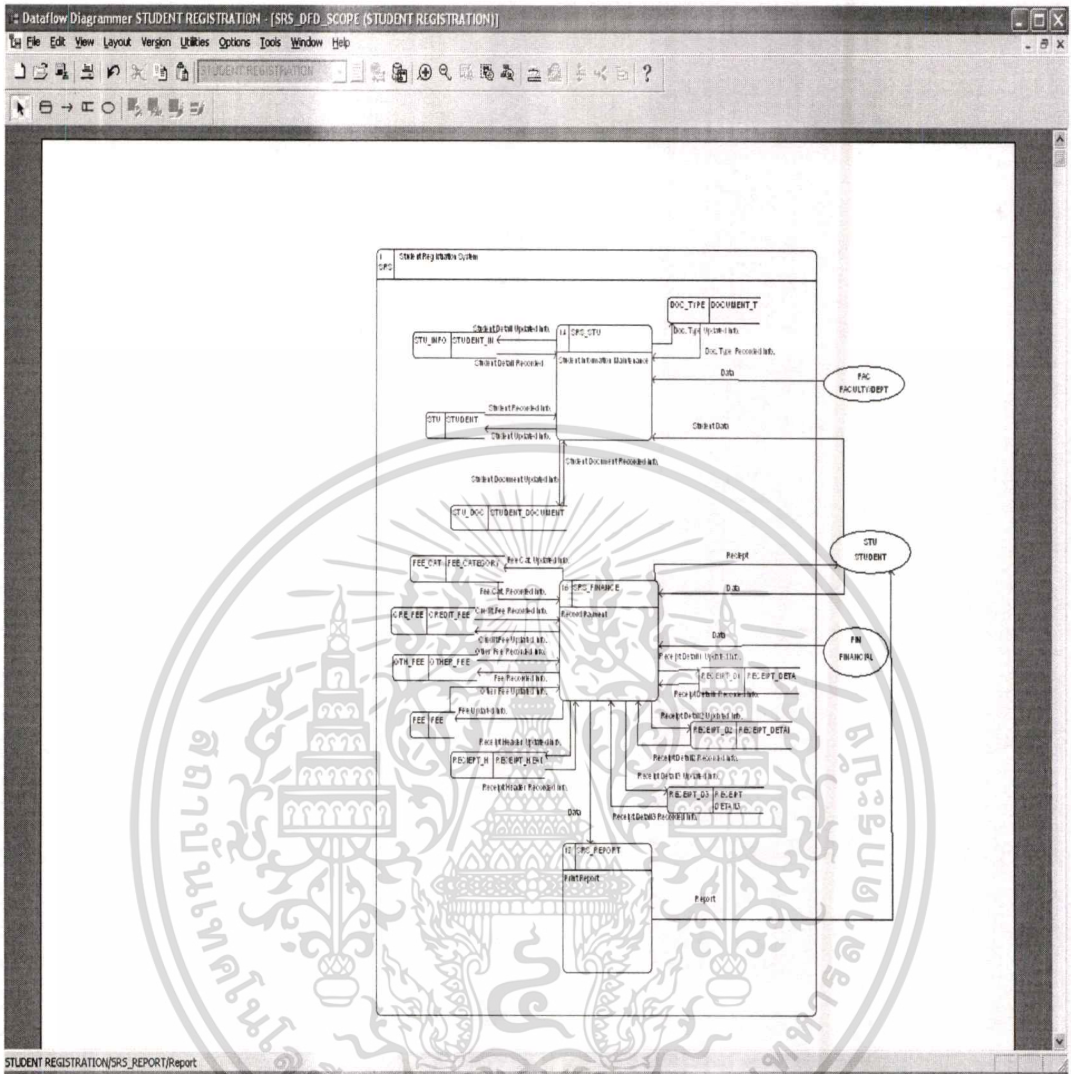
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ ห้ามเผยแพร่หรือดัดแปลงโดยไม่ขออนุญาตจากเจ้าของลิขสิทธิ์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การพัฒนาระบบงานส่วนของโปรแกรมการใช้งาน

ในการพัฒนาระบบงานส่วนของโปรแกรมใช้งาน โดยการพัฒนาระบบงานส่วนนี้จะใช้ Dataflow Diagram เพื่อสร้างกระบวนการทำงานทั้งหมดของระบบที่ต้องการจะพัฒนา หลังจากนั้นก็จะทำการสร้าง Function Hierarchy Diagram เพื่อให้มองเห็น Function ต่าง ๆ ภายในระบบได้อย่างชัดเจนทั้งหมด หลังจากนั้นก็จะทำการเลือกข้อมูลต่าง ๆ ที่ใช้ในแต่ละ Function ว่ามีการใช้ข้อมูลอะไรบ้าง เมื่อเลือกข้อมูลที่จะใช้เรียบร้อยแล้ว ก็จะทำการแปลง Dataflow Diagram ที่สร้างไว้ให้เป็น Module โดยใช้ Application Design Transformer (ADT) เมื่อได้ Module เรียบร้อยแล้วก็จะนำไปสร้างเป็นโปรแกรมการใช้งาน โดยใช้ Function Generate Module ก็จะได้ Module ที่สามารถนำไปใช้งานได้ ซึ่งที่กล่าวมา สามารถกำหนดเป็นขั้นตอนต่างๆได้ ดังนี้

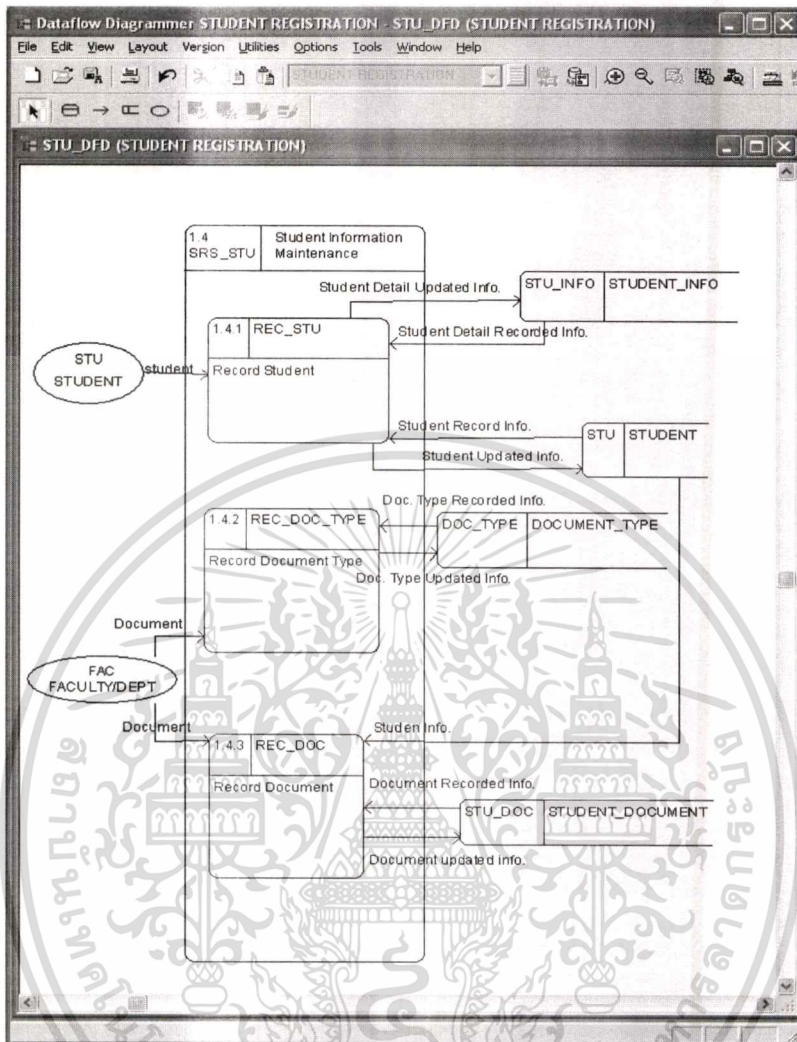
5.2.1 การสร้าง Dataflow Diagram

การสร้าง Dataflow Diagram ใน Oracle Designer จะใช้ Dataflow Diagrammer ในการสร้าง ซึ่งใน Dataflow Diagram จะประกอบด้วย Frame Function ซึ่งเป็นกระบวนการทำงานหลักที่เราสนใจ โดยสิ่งที่อยู่ภายใน Frame Function นั้นก็จะเป็นกระบวนการทำงานย่อย ๆ ลงไป ที่เรียกว่า Local Function โดย Local Function นั้นก็สามารถที่จะมี Local Function ย่อยลงไปได้อีก Function ต่างๆ เหล่านี้ก็จะมีการส่งหรือรับข้อมูลเข้ามาประมวลผลจากส่วนที่ทำหน้าที่เก็บข้อมูลที่เรียกว่า Data Store และอาจจะมีการส่งหรือรับข้อมูลไปยังภายนอกที่เรียกว่า External โดยในรูปแบบที่ 5.7 – รูปที่ 5.10 ซึ่งจะสร้างตาม Dataflow Diagram ที่สร้างขึ้นในขั้นตอนของการวิเคราะห์และออกแบบระบบงานนั่นเอง



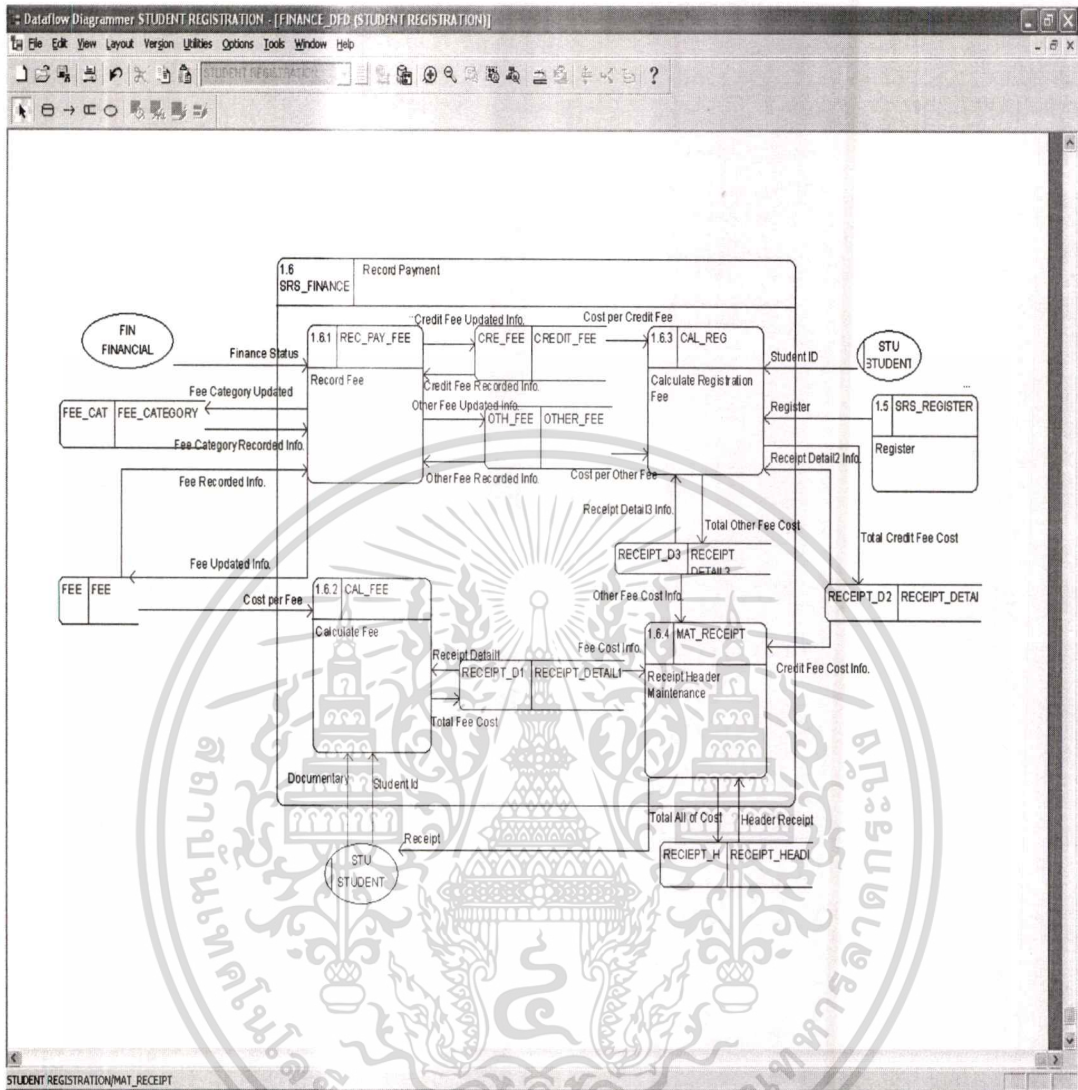
รูปที่ 5.7 Dataflow Diagram ระดับที่ 1 ของระบบทะเบียนนักศึกษาในส่วนของข้อมูลนักศึกษา การเงิน และการออกรายงานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



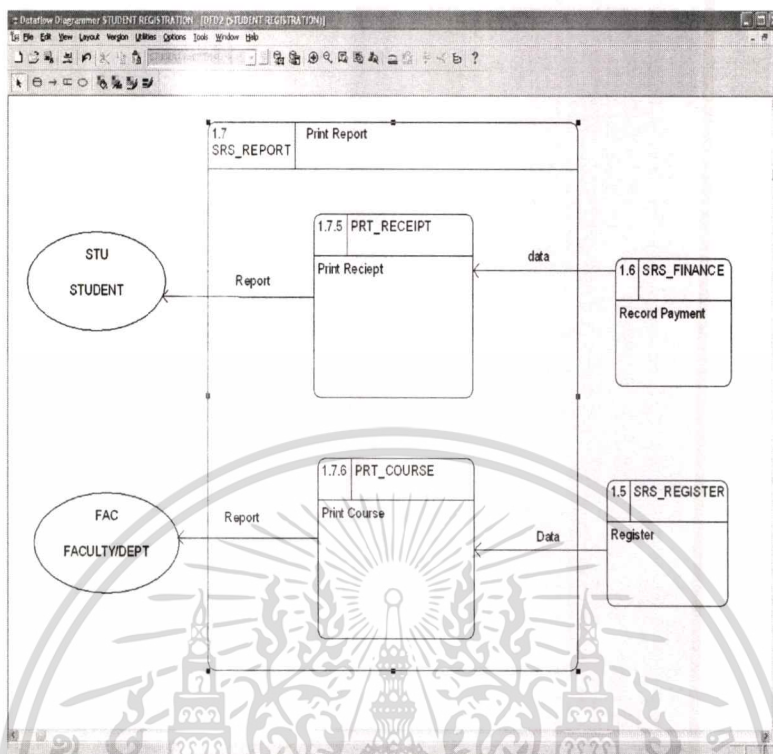
รูปที่ 5.8 Dataflow Diagram ระดับที่ 2 ของข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 Dataflow Diagram ระดับที่ 2 ของการเงิน

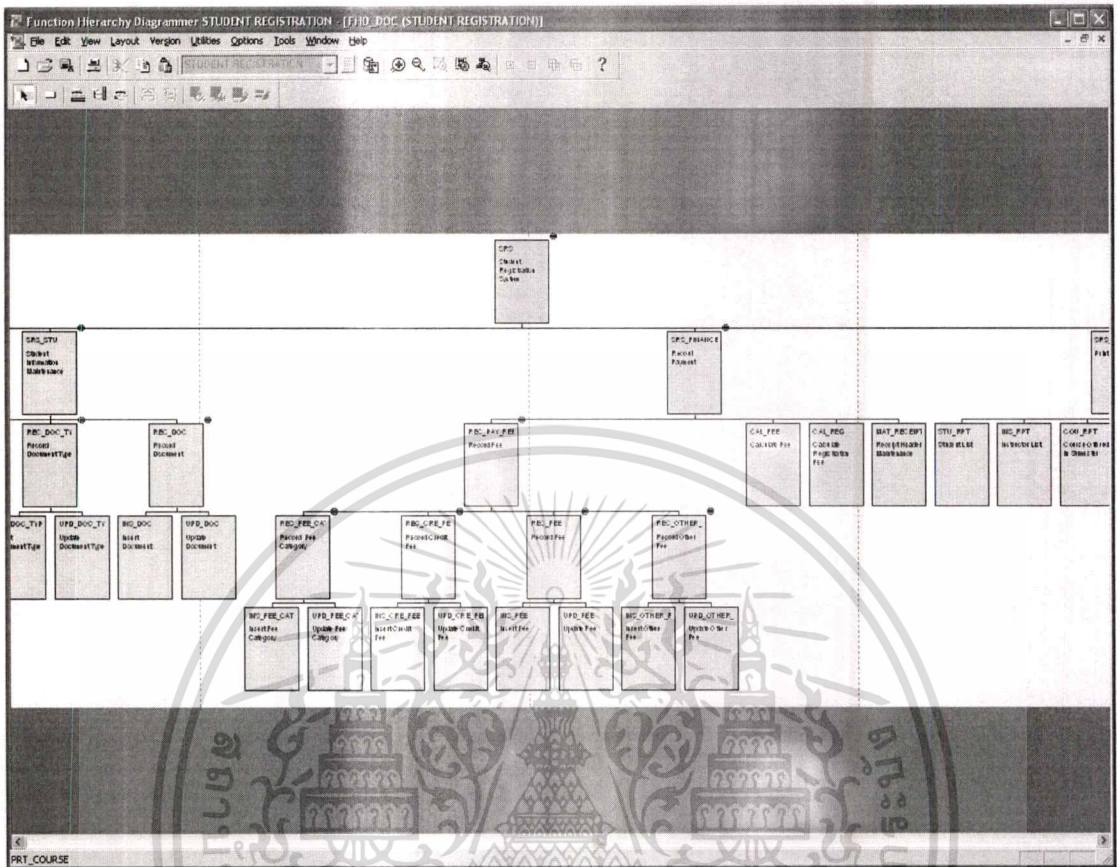
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 Dataflow Diagram ระดับที่ 2 ของการออกรายงานต่าง ๆ

5.2.2 การสร้าง Function Hierarchy Diagram

เมื่อสร้าง Dataflow Diagram เรียบร้อยแล้ว ก็จะทำการสร้าง Function Hierarchy Diagram โดยใช้ Function Hierarchy Diagrammer เพื่อที่จะทำให้เห็น Function ต่าง ๆ ที่มีในระบบทั้งหมดว่ามี Function อะไรบ้าง เพื่อสะดวกแก่การกำหนดคุณสมบัติต่าง ๆ ให้แต่ละ Function โดย Function ที่อยู่ภายใน Function Hierarchy Diagram มีอยู่ 3 ประเภท ได้แก่ Root Function, Full Function และ Atomic Function โดยที่ Root Function ก็คือ Function ที่อยู่บนสุด Full Function คือ Function ที่มี Child อย่างน้อย 1 Function ซึ่งก็คือ Function ที่ทำหน้าที่เป็น Parent และ Atomic Function ซึ่งเป็น Function ไม่มี Child หรืออีกอย่างได้ว่าเป็น Leaf Function นั่นเอง ซึ่งการสร้าง Function Hierarchy Diagram จะทำการสร้างให้อัตโนมัติ ดังรูปด้านล่าง ซึ่งจำนวนของ Leaf Function จะขึ้นอยู่กับการออกแบบ Dataflow Diagram นั่นเอง



รูปที่ 5.11 Function Hierarchy Diagram ของระบบทะเบียนนักศึกษา

5.2.3 การกำหนด Response Type

การกำหนด Response Type นั้นเป็นการกำหนด Function ว่าเมื่อทำงานแล้ว Function นั้นจะส่งผลลัพธ์หรือตอบสนองได้เร็วเพียงใด Response Type มีอยู่ 2 รูปแบบ คือ Immediate และ Overnight ซึ่ง Response Type จะมีผลต่อการสร้าง Module ออกมาด้วย โดยที่ถ้า กำหนด Response Type เป็น Immediate ก็จะสร้างเป็น Screen แต่ถ้ากำหนดเป็น Overnight ก็จะทำการสร้างเป็น Utility หรือ Report เท่านั้น ดังรูปด้านล่างซึ่งจะเป็นการกำหนดค่า Response Type ของ Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit Function - STUDENT REGISTRATION/INS_STU

Definition | Common | Triggers | Entity Usages | Attribute Usages | Text

Label: Elementary Atomic

Short Definition:

To Be Automated

Parent: Elementary

Parent Label: Elementary

Parent Short Definition:

Frequency Times: Unit: Response:

OK Cancel Apply Help

รูปที่ 5.12 การกำหนด Response Type

5.2.4 การกำหนดการใช้ข้อมูลของแต่ละ Function

สำหรับขั้นตอนการใช้ข้อมูลของแต่ละ Function จะเป็นขั้นตอนที่จะทำการกำหนดว่าใน Function นี้จะใช้ข้อมูลอะไรบ้างในการทำงาน โดยเลือกที่ Entity Usage เพื่อเลือก Entity ที่ต้องใช้ และ Attribute Usage เพื่อเลือก Attribute ที่ต้องใช้ใน Function นี้ โดยรูปด้านล่างจะเป็นการกำหนดการใช้ข้อมูลของ Function

Edit Function - STUDENT REGISTRATION/INS_STU

Definition | Common | Triggers | Entity Usages | Attribute Usages | Text

Entity	Create	Retrieve	Update	Delete
FACULTY (STUDENT REGISTRAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CURRICULUM (STUDENT REGIST	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
STUDENT_TYPE (STUDENT REG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INSTRUCTOR (STUDENT REGIST	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DEPARTMENT (STUDENT REGIS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
STUDENT (STUDENT REGISTRA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
STUDENT_INFO (STUDENT REGI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Insert Delete Select All Reset Default

OK Cancel Apply Help

รูปที่ 5.13 การกำหนดการใช้ Entity ต่าง ๆ ให้กับ Function

Edit Function - STUDENT REGISTRATION/INS_STU

Definition | Common | Triggers | Entity Usages | Attribute Usages | Text

Entity
STUDENT (STUDENT REGISTRATION)

Attribute	Insert	Retrieve	Update	Nullity
ST_ACC_CR_A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_ACC_CR_E	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_ACC_STATUS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_ADM_DATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_DOB	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_E_MAIL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_FIN_STATUS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ST_F_NAME_T	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Insert Delete Select All Reset Default

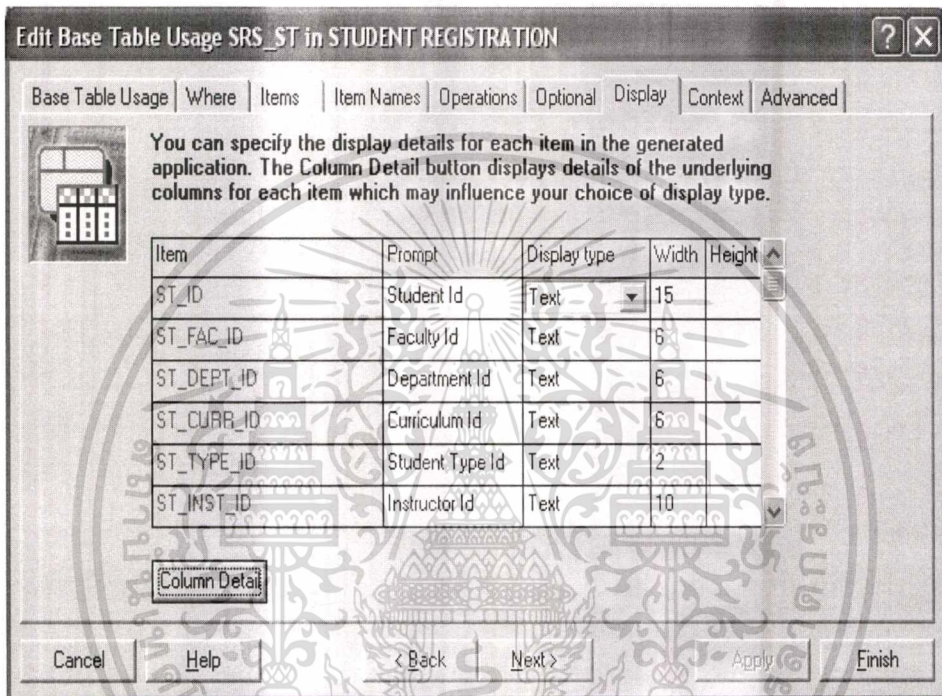
OK Cancel Apply Help

รูปที่ 5.14 การกำหนดการใช้ Attribute ต่าง ๆ ให้กับ Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5 การกำหนด Display Type ของข้อมูลที่ใช้

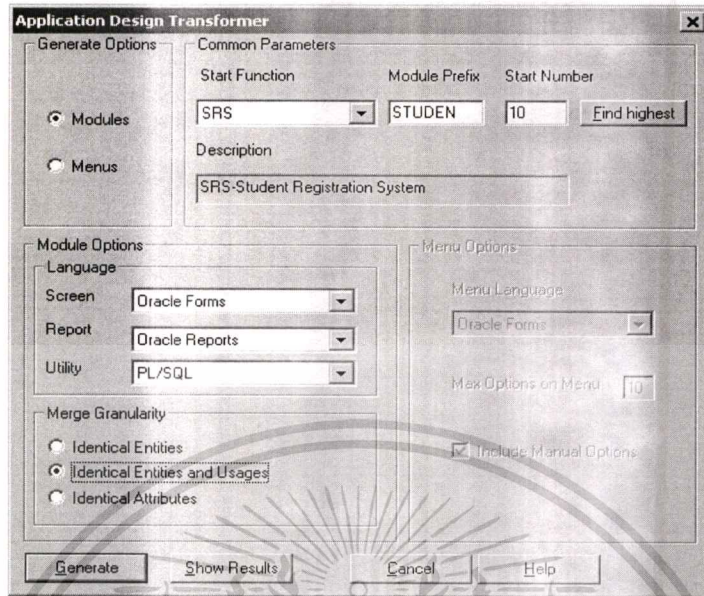
การกำหนด Display Type จะเป็นการกำหนดรูปแบบในการแสดงคอลัมน์ที่อยู่ใน Server Model Diagram ว่า ถ้ามีการใช้คอลัมน์นี้จะให้ปรากฏบนหน้าจอในรูปแบบใด อย่างเช่น Text Pop list Display Item เป็นต้น ดังรูปที่ 5.15



รูปที่ 5.15 การกำหนด Display Type

5.2.6 การใช้ Application Design Transformer

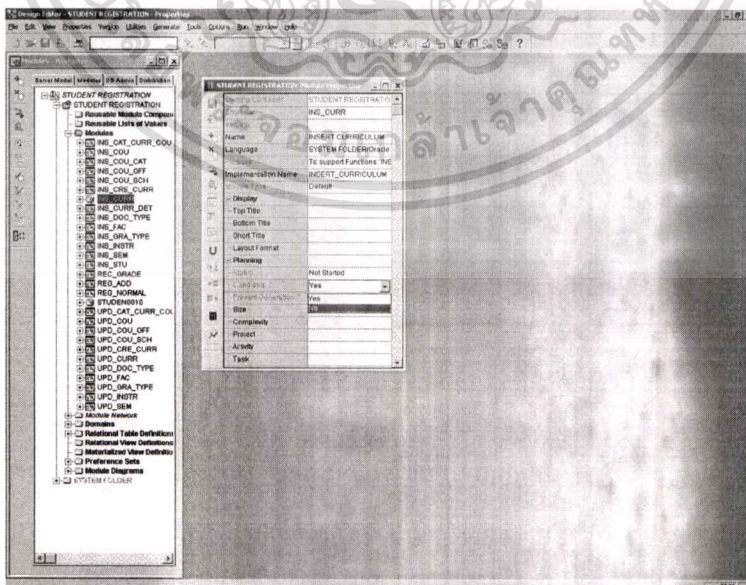
เมื่อทำการกำหนดค่าต่าง ๆ เรียบร้อยแล้ว ขั้นตอนต่อไป ก็คือจะทำการใช้ Application Design Transformer (ADT) ซึ่งจะทำการแปลง Function และข้อมูลที่ใช้ให้กลายเป็น Application Module ในรูปแบบ Application First-cut Design โดยจะมีฟอร์มขึ้นมาในรูปที่ 5.16 ที่ให้ผู้ใช้ทำการกำหนดค่าในการสร้างเป็นรูปแบบ Module หรือ Menu ต้องการให้ Application Module รูปแบบต่างๆ อย่างเช่น Report Screen Utility เป็นต้น โดยสามารถกำหนดภาษาที่ต้องการสร้างโดยไม่จำเป็นต้องเป็นภาษาของ Oracle เพียงอย่างเดียว เมื่อทำการ Generate แล้วก็จะได้ Module ต่าง ๆ ออกมา ตาม Function เริ่มต้นที่ได้เลือกเอาไว้



รูปที่ 5.16 Application Design Transformer

5.2.7 การเปลี่ยนคุณสมบัติ Candidate ของ Module

เมื่อทำการสร้าง Module จากการใช้ Application Design Transformer เรียบร้อยแล้ว Module ที่ได้จะเรียกว่าเป็น Candidate Module ซึ่งไม่สามารถที่จะทำการสร้าง Form ได้ ต้องทำการเปลี่ยนคุณสมบัติ Candidate จาก Yes ให้เป็น No เสียก่อน ดังในรูปที่ 5.17



รูปที่ 5.17 การเปลี่ยนคุณสมบัติ Candidate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.8 การปรับปรุงแก้ไข Application Module

เมื่อทำการเปลี่ยนคุณสมบัติ Candidate แล้ว ก็จะทำกรปรับปรุงแก้ไข Application Module เพื่อที่จะทำการสร้าง Form ที่ต้องการให้อยู่ในรูปแบบที่สลับซับซ้อน และมีประสิทธิภาพที่เพิ่มมากขึ้นได้ โดยมีขั้นตอนต่าง ๆ ดังนี้

- **การสร้าง Module Diagram**

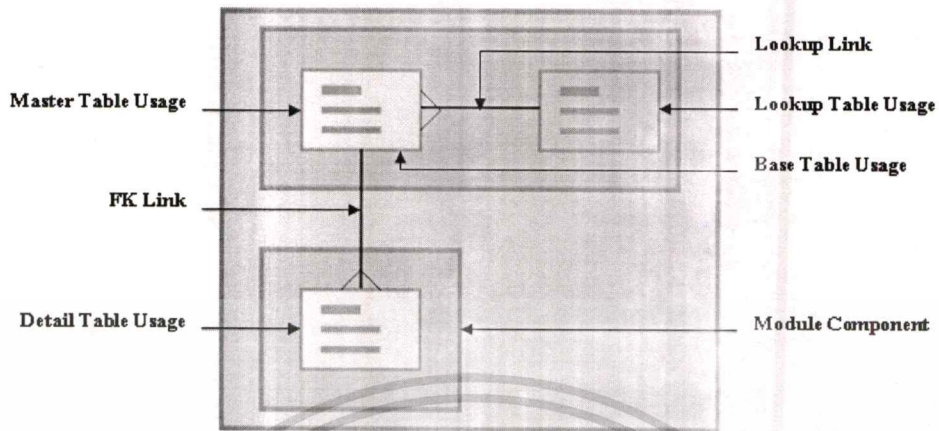
การสร้าง Module Diagram ใน Oracle Designer6i จะใช้ Design Editor ซึ่งสามารถที่จะแสดง Module Diagram ได้ 2 มุมมอง ก็คือ Data view และ Display View โดยที่ Data View จะแสดงโครงสร้างของ Module ว่า Module ประกอบด้วย Table Usage อะไรบ้าง และแสดงความสัมพันธ์ระหว่าง Table Usage ต่าง ๆ สำหรับ Display View จะแสดงให้เห็นว่า Module นี้เมื่อ generate ออกมาแล้ว จะมี Window และ Canvas อะไรบ้าง และมีการจัดเรียงลำดับ Item ที่จะแสดงออกมา

- **การจัดโครงสร้าง Module**

การจัดโครงสร้างของ Module นั้นจะเริ่มจากการจัดลำดับตำแหน่งของ Module Component ตามความสัมพันธ์ของ Table Usage ต่างๆ ดังนี้

- ถ้าเป็นความสัมพันธ์แบบ Master-Detail Master Table Usage ไว้ด้านบน Detail Table Usage เสมอ แล้วทำการสร้าง Link เชื่อม Table Usage
- ถ้าเป็นความสัมพันธ์แบบ Lookup Table Usage จะต้องอยู่ทางด้านขวาของ Base Table เสมอ แล้วทำการสร้าง Link เชื่อม Table Usage

โดยที่ความสัมพันธ์ของ Table Usage ทั้งแบบ Master-Detail Link และแบบ Lookup Link จะต้องมี Foreign Key จึงจะสามารถสร้างความสัมพันธ์ของ Table Usage ได้ ดังรูปที่ 5.18



รูปที่ 5.18 โครงสร้าง Module

- การสร้าง Bound Item

Bound Item เป็นการกำหนด Module Data Usage โดยกำหนดว่า ในแต่ละ Module นั้น จะใช้คอลัมน์อะไรบ้าง

- การกำหนดคุณสมบัติของ Module Component

การกำหนดคุณสมบัติของ Module Component ที่สำคัญ ๆ ดังนี้

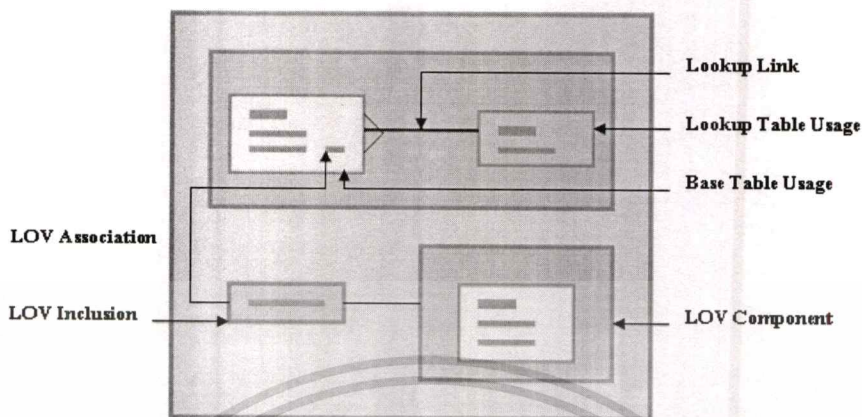
- ชื่อของ Module Component
- กำหนด Allowable Operation ต่าง ๆ ว่าใน Module นี้สามารถที่จะ Insert Update Delete ได้หรือไม่
- จำนวนแถวที่แสดงใน 1 บล็อกข้อมูล ซึ่งค่า Default จะกำหนดเป็น Maximum

- การสร้าง LOV Component

การสร้าง LOV Component นั้นเราจะใช้ Utilities ที่ชื่อว่า Default List of Value หลังจากที่ได้กำหนด Lookup Table แล้ว เพื่อทำการสร้าง LOV ทั้งหมดที่ Module ต้องการให้โดยอัตโนมัติ ซึ่ง LOV Component จะช่วยทำให้ผู้ใช้สามารถที่จะทำการป้อนค่าและสอบถามข้อมูลได้อย่างถูกต้องและรวดเร็วมากขึ้น โดยให้ผู้ใช้เลือกค่าจาก List of Value ที่แสดง โดยเราสามารถที่จะเพิ่มเติมได้ โดยการสร้าง LOV inclusion และใช้ LOV association เพื่อเรียกใช้ LOV ขึ้นมาได้ โดยในรูปที่ 5.19 เป็น โครงสร้างของ LOV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

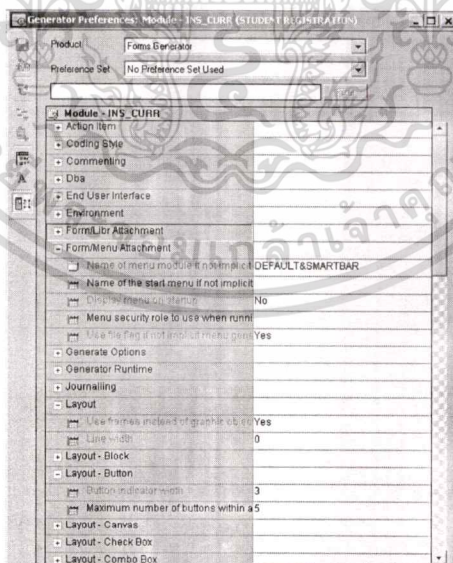
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 โครงสร้างของ LOV

5.2.9 การสร้าง Forms

ในการสร้าง Forms นั้นเราจะต้องทำการกำหนดค่าต่าง ๆ ของ Preference ที่อยู่ใน Generator Preference ก่อน เพื่อให้ Form ที่ทำการสร้างขึ้นมานั้น มีรูปแบบและการทำงานเป็นไปตามที่ผู้ใช้ต้องการ ดังรูปที่ 5.20



รูปที่ 5.20 Generator Preference

หลังจากทำการกำหนดค่าใน Preference เรียบร้อยแล้ว ก็จะทำการสร้าง Form

ขึ้นมาโดยใช้ Generate Form ดังรูปที่ 5.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.21 สร้าง Form โดยใช้ Generate Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและข้อเสนอแนะ

เพื่อศึกษาการทำงานในการพัฒนาระบบโดยใช้ CASE Tools นั้น จึงได้ทำการศึกษาและพัฒนาระบบทะเบียนนักศึกษา โดยนำ Oracle Designer6i CASE Tools มาใช้ในการพัฒนา ทำให้ทราบถึงข้อดี และข้อจำกัดต่าง ๆ ดังนี้

6.1 ข้อดีของการนำ Oracle Designer6i มาพัฒนาระบบ

ข้อดีของการพัฒนาระบบโดยใช้ Oracle Designer6i นั้นสรุปได้ ดังนี้

- ข้อมูลต่าง ๆ ที่เกี่ยวข้องในการพัฒนาระบบจะมีการกำหนดให้อยู่รูปแบบเดียวกันและเป็นมาตรฐานเดียวกันทั้งหมด ทำให้สามารถเข้าใจตรงกัน
- ในการพัฒนาระบบไม่จำกัดว่าต้องขึ้นอยู่กับภาษาใดภาษาหนึ่ง เพราะ Oracle Designer6i สามารถที่จะพัฒนาได้หลากหลายภาษา
- ลดค่าใช้จ่ายในการพัฒนาระบบในส่วนที่จะต้องจัดซื้อ Software อื่น ๆ ได้ เนื่องจากอาจจะสามารถงานใช้ Software ที่มีอยู่ร่วมกับ Oracle Designer6i ได้
- มีระบบ Security ในส่วนของการเข้าถึงข้อมูลที่อยู่ภายใน Repository ทำให้ข้อมูลมีความปลอดภัย
- มีการสร้างเอกสารประกอบการพัฒนาระบบทุกขั้นตอนการทำงาน ทำให้สามารถที่จะตรวจสอบความถูกต้องของระบบได้ง่าย
- มีการเก็บข้อมูลต่าง ๆ ที่เกี่ยวกับการพัฒนาระบบไว้ในที่เดียวคือ Repository ทำให้ง่ายต่อการค้นหาและปรับปรุงแก้ไขข้อมูล
- สามารถจะพัฒนาระบบได้อย่างรวดเร็วและมีประสิทธิภาพ ถ้ามีความรู้และความเข้าใจในการใช้งาน Oracle Designer6i
- สามารถที่จะนำระบบที่ได้พัฒนาเอาไว้ มาปรับปรุง เปลี่ยนแปลงแก้ไขได้อย่างสะดวกและง่ายดาย
- สามารถที่จะแบ่งการพัฒนาออกเป็นส่วน ๆ เพื่อทำพร้อมกันในแต่ละทีมได้ เนื่องจากถ้ามีการปรับปรุงข้อมูลแล้ว จะทำการปรับปรุงทุกในส่วนที่เกี่ยวข้องกัน

6.2 ข้อจำกัดของการนำ Oracle Designer6i มาพัฒนาระบบ

ข้อจำกัดของการพัฒนาระบบโดยใช้ Oracle Designer6i นั้นสรุปได้ดังนี้

- ในการพัฒนาระบบโดยแบ่งงานออกเป็นทีมนั้น จะต้องมีการกำหนดชื่อต่าง ๆ ที่ใช้ในการพัฒนาให้เข้าใจตรงกันทั้งหมด เนื่องจาก Oracle Designer6i ไม่ได้บังคับในส่วนนี้
- ค่อนข้างจะต้องใช้เวลาในการศึกษาเกี่ยวกับการใช้งาน Oracle Designer6i เพื่อที่จะทำให้ใช้งานได้อย่างมีประสิทธิภาพ
- User Interface ที่ได้จากการสร้างของ Oracle Designer6i นั้น มีการจัด Layout ที่ไม่สวยงามเท่าที่ควร ซึ่งอาจจะต้องมีการใช้โปรแกรมอย่างเช่น Oracle Forms, Microsoft Visual Basic หรือ Borland Delphi เป็นต้น เข้ามาช่วย จะทำให้โปรแกรมมี Layout ที่สวยงามขึ้นได้

6.3 ข้อเสนอแนะ

ในโครงการพัฒนาระบบนี้ ได้ทำการศึกษาและสร้างระบบทะเบียนนักศึกษาขึ้นมาโดยใช้ Oracle Designer6i นั้น ได้มีการพัฒนาระบบแต่ละส่วนในรูปแบบของ Module ซึ่งสามารถที่จะทำให้เป็น Network Module ซึ่งจะส่งค่ากันระหว่าง Module ได้ซึ่งสามารถที่จะใช้ Oracle Designer6i ในการพัฒนาเพิ่มเติมในส่วนนี้ได้

บรรณานุกรม

- ทรงพล บุรณะโอสถ และ สุรชาติ พงศ์สุทธนะ. 2544. **Oracle คู่มือการใช้ระบบฐานข้อมูล Oracle**. กรุงเทพฯ: พิมพ์ดี.
- สมจิตร อาจอินทร์ และ งามนิจ อาจอินทร์. 2541. **ระบบฐานข้อมูล Database System**. ขอนแก่น: ศูนย์หนังสือมหาวิทยาลัยขอนแก่น.
- พัชรี ประกอบกิจ. 2542. **การพัฒนาระบบซ่อมโทรศัพท์เคลื่อนที่โดยใช้โปรแกรม CASE**. กรุงเทพฯ : ปรินูญานิพนธ์ หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- สุกิจ กุชัยสิทธิ์. 2542. **การพัฒนาระบบสารสนเทศโดยใช้ CASE TOOL ORACLE DESIGNER/2000 : กรณีศึกษา แผนกพัสดุและจัดซื้อ**. กรุงเทพฯ : ปรินูญานิพนธ์ หลักสูตรวิทยาศาสตรมหาบัณฑิต คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- Kenneth, E.K. and Julie, E. 1999. **System Analysis and Design**. New Jersey: Prentice-Hall.
- Oracle Corporation. 1999. **Oracle Designer Tutorial**. [Online].
Available: http://www.uncg.edu/irc/oracle/DesignerDoc_Tut/cCTU71/toc.htm.
- Oracle Corporation. 2000. **Oracle Designer6i New Features**. [Online].
Available : <http://otn.oracle.com/products/designer/content.html>.
- Oracle Corporation. 2000. **Oracle Designer 6i Product Overview**. [Online].
Available: http://technet.oracle.com/products/designer/pdf/otn_des6i_pover_wp.pdf
- Oracle Corporation. 2001. **Oracle iDS Designer: First Class**.

ประวัติผู้เขียน

ชื่อ นางสาวกิ่งกาญจน์ วิไลदारกา
วันเดือนปีเกิด 4 ตุลาคม พ.ศ. 2522
สถานที่เกิด กรุงเทพฯ
การศึกษา จบชั้นมัธยมศึกษาจาก โรงเรียนสายน้ำผึ้ง
จบปริญญาตรี คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยหัวเฉียวเฉลิมพระเกียรติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้