สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์กับการประยุกต์เชิงเวลา

Object-Relational Database in Temporal Application

นายกฤษฎา ศรีแผ้ว
นายปัญจ แซ่อุย

เลขหมู่.....
เลขทะเบียน 42795
วัน, เดือน, ปี 10 มิ.ย. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์กับการประยุกต์เชิงเวลา

Object-Relational Database in Temporal Application

โดย

นายกฤษฎา ศรีแผ้ว

นายปัญจ แซ่อุย

อาจารย์ที่ปรึกษา

รศ.ดร. ศุภมิตร จิตตะยโศธร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์กับการประยุกต์เชิงเวลา

Object-Relational Database in Temporal Application

ผู้จัดทำ

1. นายกฤษฎา ศรีแผ้ว รหัสประจำตัว 40010017
2. นายปัญจ แซ่อุย รหัสประจำตัว 40010454

อาจารย์ที่ปรึกษา

(รศ.ดร. ศุภมิตร จิตตะยโศธร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์กับการประยุกต์เชิงเวลา

นายกฤษฎา ศรีแผ้ว 40010017
นายปัญจ แซ่อุย 40010454
รศ.ดร. ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ฐานข้อมูลเชิงเวลา เป็นฐานข้อมูลที่เก็บข้อมูลในช่วงเวลาต่างๆ ทั้งในอดีต ปัจจุบัน และอนาคต โดยข้อมูลใหม่ที่แก้ไขลงไปในฐานข้อมูล จะไม่ทับข้อมูลเดิม หมายความว่า ข้อมูลยังคงอยู่ในฐานข้อมูลเสมอ ประโยชน์ที่ได้รับจากการจัดเก็บแบบนี้คือ สามารถติดตามการเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูล และสามารถตอบคำถามที่ซับซ้อนในงานบางประเภทได้ ซึ่งประโยชน์ที่จะได้รับนี้ต้องแลกด้วยความยุ่งยากในการจัดการกับข้อมูลเหล่านี้ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ทั้งการใส่ข้อมูล (Insert), การแก้ไข (Update), การลบ (Delete) และการป้องกันความซ้ำซ้อน จึงได้นำเสนอวิธีการจัดการกับความยุ่งยากเหล่านี้ ด้วยการประยุกต์ใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์

ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ ที่ได้นำมาประยุกต์ใช้ คือ อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ เวอร์ชัน 9.14 (Informix Universal Server version 9.14) โดยในการวิเคราะห์และออกแบบพัฒนาระบบงานที่ใช้ฐานข้อมูลเชิงเวลา ได้เลือกระบบงานโรงพยาบาลเป็นกรณีศึกษา

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอแนวทางหนึ่ง ในการประยุกต์ใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์เพื่อจัดการกับข้อมูลเชิงเวลา ซึ่งเป็นวิธีการที่สามารถทำความเข้าใจได้ง่าย เนื่องจากเป็นระบบจัดการฐานข้อมูลที่นำแนวคิดเชิงสัมพันธ์ในแบบเก่า รวมเข้ากับแนวคิดเชิงวัตถุ ซึ่งเป็นส่วนเชื่อมต่อที่สำคัญในการพัฒนาเพื่อรองรับการเปลี่ยนแปลงในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Object-Relational Database in Temporal Application

Kritsada Sriphaew

Panja Sae-Ui

Assoc. Prof. Dr. Suphamit Chitayasothorn

Advisor

ABSTRACT

Temporal database is a database storing data in various times. Update a new data to database does not replace the old parts that means these old parts are still store in the database. Advantages of this sort of database are ability to account for data changing and to answer complex questions in some applications. To achieve these advantages we have to manipulate complex management with temporal data using Relation Database Management System (RDBMS) such as insert, update, deletion and avoid duplicate data. So we would like to present the new techniques to eliminate these problems using Object-Relational Database Management System (ORDBMS).

ORDBMS used in this project is Informix Universal Server version 9.14. We chose a hospital application to be the case study to analyze and develop a temporal application.

This thesis presented the one of choices to apply object-relational database to manage temporal data. This is an easily understanding solution because there is the database management system, which uses old-fashioned and object-oriented concepts. Therefore, object-relational database is an important connection for future evolution.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความสนับสนุนจากบุคคลและหน่วยงานต่างๆ ที่ได้ให้ความอนุเคราะห์ในข้อมูลและการดำเนินการต่างๆ ไปได้ด้วยดี ขอขอบคุณอาจารย์บัณฑิต พัสยา ที่ได้ช่วยเหลือและอนุเคราะห์ซอฟท์แวร์และแหล่งข้อมูลในการค้นคว้า รวมทั้งโรงพยาบาลที่ได้ให้ข้อมูลทางการแพทย์ที่เป็นข้อมูลลับของทางโรงพยาบาลที่มีประโยชน์มากในการศึกษา ควบคุมพนักงานบรรณารักษ์ห้องสมุดคณะสารสนเทศที่ได้ความสะดวกในการค้นหาวิทยานิพนธ์ฉบับเก่าๆ ขอบคุณกระทรวงกลาโหม สหรัฐอเมริกาที่ได้สร้างระบบอินเตอร์เน็ตขึ้นมา ในการใช้ค้นหาข้อมูลที่ทรงคุณค่า และขอขอบคุณท่านอาจารย์ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ คอยช่วยเหลือ และให้คำแนะนำที่ก่อให้เกิดความเข้าใจ ความคิดริเริ่มสร้างสรรค์ รวมถึงเป็นการสร้างวิสัยทัศน์ในมุมมองของนักวิชาการ

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้

กฤษฎา ศรีแผ้ว
ปัญจ แซ่อุย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ของเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีเชิงเวลา	4
2.1 ฐานข้อมูลเชิงเวลา	4
2.2 ประเภทของฐานข้อมูลเชิงเวลา	5
2.3 ความซ้ำซ้อน	7
2.4 การป้องกันความซ้ำซ้อน	9
2.5 การค้นหาข้อมูลของตารางวาลิดไทม์	10
2.6 การแก้ไขข้อมูลเชิงเวลา	17
2.7 ปัญหาของฐานข้อมูลเชิงสัมพันธ์กับฐานข้อมูลเชิงเวลา	24
2.8 ภาษา SQL กับข้อมูลเชิงเวลา	25
บทที่ 3 ฐานข้อมูลเชิงวัตถุสัมพันธ์	28
3.1 ฐานข้อมูลเชิงวัตถุ	28
3.2 ระบบฐานข้อมูลเชิงสัมพันธ์	29
3.3 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์	31
3.4 โมเดลที่ใช้ในการออกแบบ	36
บทที่ 4 อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์	38
4.1 อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์	38
4.2 สถาปัตยกรรมของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์	38
4.3 ชนิดข้อมูล	40
4.4 การควบคุมความถูกต้องของข้อมูล	43
4.5 อินฟอร์มิกซ์กับคุณสมบัติเชิงวัตถุ	44
4.6 เครื่องมือช่วยในการทำงาน	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

บทที่ 5 การประยุกต์ฐานข้อมูล 49
5.1 การสร้างฐานข้อมูลเชิงเวลาบนอินฟอร์มิกซ์ 49
5.2 การแก้ไขปัญหาเชิงเวลาโดยใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์ 49
5.3 รูปแบบการประยุกต์การจัดเก็บเชิงเวลาใน IUS 50
5.4 การจัดการกับคอลเลคชั่นไทป์ 51
5.5 การทดลองใช้ข้อมูลชนิด BLOB เพื่อเก็บไฟล์รูปภาพ 56
5.6 สรุปการประยุกต์ฐานข้อมูล 57

บทที่ 6 การวิเคราะห์และออกแบบระบบฐานข้อมูล 58
6.1 ระบบงานที่เลือกศึกษา 58
6.2 ขอบเขตและบุคคลที่ใช้ระบบงานที่จะพัฒนาขึ้น 58
6.3 ประโยชน์ที่คาดว่าจะได้รับ 58
6.4 การวิเคราะห์ระบบงานโรงพยาบาล 59
6.5 Data Dictionary ของระบบ 63
6.6 การออกแบบระบบงานโรงพยาบาลในส่วนฐานข้อมูล 71

บทที่ 7 การสร้างฐานข้อมูลและแอพพลิเคชัน 78
7.1 การสร้างฐานข้อมูล 78
7.2 การพัฒนาโปรแกรมประยุกต์ 86

บทที่ 8 บทสรุปและวิจารณ์ 94
8.1 สรุปผลและวิจารณ์ผลโครงงาน 94
8.2 แนวทางการพัฒนาต่อ 96

บรรณานุกรม 97

ภาคผนวก

ภาคผนวก ก การติดตั้งระบบงานโรงพยาบาล 98
ภาคผนวก ข การใช้งานโปรแกรมระบบงานโรงพยาบาล 107
ภาคผนวก ค ส่วนประกอบของโปรแกรมระบบงานโรงพยาบาล 137

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้าที่
ตารางที่ 2-1	ตารางวาลิดไทม์ของข้อมูลเด็กในโรงพยาบาลแห่งหนึ่ง	7
ตารางที่ 2-2	ตารางแสดงความสัมพันธ์ของความซ้ำซ้อนแบบต่างๆ	8
ตารางที่ 2-3	ตาราง LOT_LOC	10
ตารางที่ 2-4	คำตอบของตัวอย่างการตอบคำถามแบบปัจจุบัน	11
ตารางที่ 2-5	คำตอบของตัวอย่างการตอบคำถามแบบซีเควนต์	12
ตารางที่ 2-6	คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนต์	12
ตารางที่ 2-7	คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนต์โดยใช้การจอย	13
ตารางที่ 2-8	คำตอบของตัวอย่างการตอบคำถามแบบซีเควนต์โดยใช้การจอย	16
ตารางที่ 2-9	ตารางเพศวัว	17
ตารางที่ 2-10	ผลของตัวอย่างการลบแบบปัจจุบัน	18
ตารางที่ 2-11	การจัดเก็บฐานข้อมูลเชิงสัมพันธ์ทั่วไป	24
ตารางที่ 2-12	การอับเดทข้อมูลฐานข้อมูลเชิงสัมพันธ์ทั่วไป	24
ตารางที่ 2-13	การจัดเก็บฐานข้อมูลเชิงเวลาโดยใช้ฐานข้อมูลเชิงสัมพันธ์	24
ตารางที่ 3-1	แสดงความสัมพันธ์ของแอตทริบิวต์ที่ไม่อะตอมมิค	32
ตารางที่ 4-1	ชนิดของอินเตอร์เฟซกับโปรโตคอลที่ยูนิเวอร์แซลเซิร์ฟเวอร์สนับสนุน	39
ตารางที่ 4-2	แสดง Built-in Data Type ในอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์	41
ตารางที่ 6-1	อธิบายความสัมพันธ์ของแต่ละเอนทิตี้	74
ตารางที่ 7-1	แสดงคอลัมน์ในตาราง MEDICINE	78
ตารางที่ 7-2	แสดงคอลัมน์ในตาราง DISEASE	78
ตารางที่ 7-3	แสดงคอลัมน์ในตาราง DIAGNOSIS	79
ตารางที่ 7-4	แสดงคอลัมน์ในตาราง DOCTOR	79
ตารางที่ 7-5	แสดงคอลัมน์ในตาราง PATIENCE	80
ตารางที่ 7-6	แสดงคอลัมน์ในตาราง LRNP	80
ตารางที่ 7-7	แสดงคอลัมน์ในตาราง LRCP	81
ตารางที่ 7-8	แสดงคอลัมน์ในตาราง LRPP	81
ตารางที่ 7-9	แสดงคอลัมน์ในตาราง LRND	81
ตารางที่ 7-10	แสดงคอลัมน์ในตาราง LRCD	81
ตารางที่ 7-11	แสดงคอลัมน์ในตาราง LRPD	82
ตารางที่ 7-12	แสดงคอลัมน์ในตาราง LRNA	82
ตารางที่ 7-13	แสดงคอลัมน์ในตาราง LRCA	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

ตารางที่ 7-14 แสดงคอลัมน์ในตาราง LRPA 82
ตารางที่ 7-15 แสดงคอลัมน์ในตาราง ROOM 83
ตารางที่ 7-16 แสดงคอลัมน์ในตาราง MOP 83
ตารางที่ 7-17 แสดงคอลัมน์ในตาราง MOD 83
ตารางที่ 7-18 แสดงคอลัมน์ในตาราง MOA 83
ตารางที่ 7-19 แสดงคอลัมน์ในตาราง LAB_NUMBER 84
ตารางที่ 7-20 แสดงคอลัมน์ในตาราง LAB_CHAR 84
ตารางที่ 7-21 แสดงคอลัมน์ในตาราง LAB_PHOTO 84
ตารางที่ 7-22 แสดงคอลัมน์ในตาราง ADMISSION 85
ตารางที่ 7-23 แสดงคอลัมน์ในตาราง AD_DIAGNOSIS 85
ตารางที่ 7-24 แสดงคอลัมน์ในตาราง ADMISSION_PACKAGE 86
ตารางที่ 7-25 แสดงคอลัมน์ในตาราง PHOTO 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้าที่
รูปที่ 2-1 ฐานข้อมูลประวัติ	5
รูปที่ 2-2 ฐานข้อมูลย้อนกลับ	5
รูปที่ 2-3 ฐานข้อมูลสแน็ปช็อท	6
รูปที่ 2-4 ฐานข้อมูลไบเทมโพรอล	7
รูปที่ 2-5 กรณีที่ 1 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย	14
รูปที่ 2-6 กรณีที่ 2 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย	14
รูปที่ 2-7 กรณีที่ 3 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย	15
รูปที่ 2-8 กรณีที่ 4 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย	15
รูปที่ 2-9 การเปลี่ยนแปลงเพศของวัว	17
รูปที่ 2-10 กรณีของการแก้ไขแบบปัจจุบัน	19
รูปที่ 2-11 กรณีของการลบแบบซีเควนต์	20
รูปที่ 2-12 กรณีของการแก้ไขแบบซีเควนต์	22
รูปที่ 3-1 แสดงตัวอย่างตารางความสัมพันธ์	29
รูปที่ 3-2 แสดงความสัมพันธ์ระหว่างเอนทิตี้	30
รูปที่ 3-3 แสดงความสัมพันธ์แบบ 1 : 1	30
รูปที่ 3-4 แสดงความสัมพันธ์แบบ 1 : กลุ่ม	30
รูปที่ 3-5 แสดงความสัมพันธ์แบบ กลุ่ม : กลุ่ม	31
รูปที่ 3-6 แสดงลักษณะของฐานข้อมูลแบบ RDBMS, ORDBMS และ ODBMS	31
รูปที่ 3-7 แสดงการทำงานของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์	32
รูปที่ 3-8 แสดงโครงสร้างเอนแคปซูเลชัน	34
รูปที่ 3-9 แสดงการถ่ายทอดคุณสมบัติ	35
รูปที่ 3-10 แสดงคลาสแบบลำดับชั้นด้วยอินเฮอร์ริแทนซ์แบบเดี่ยวและรวม	36
รูปที่ 3-11(ก) แสดงโมเดล ER (CASE TOOL)	37
รูปที่ 3-11(ข) แสดงการแปลงจาก ER (CASE TOOL) มาเป็น EER	37
รูปที่ 3-11(ค) แสดงตัวอย่างการแปลง ER โมเดลเป็นรีเลชั่น	37
รูปที่ 4-1 รูปแสดงสถาปัตยกรรมของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์	39
รูปที่ 4-2 แสดงชนิดข้อมูลของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์	40
รูปที่ 5-1 แสดงตารางที่เก็บข้อมูลเชิงเวลาโดยเอาวาลิดไทม์เกาะทั้งโรว์	49
รูปที่ 5-2 แสดงการเก็บข้อมูลเชิงเวลาโดยเอาวาลิดไทม์เกาะแต่ละแอตทริบิวต์ โดยเก็บข้อมูลในเซ็ตของแต่ละคอลัมน์	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 5-3 แสดงตารางข้อมูลเชิงเวลาเป็น 3 มิติ 50
รูปที่ 5-4 แสดงข้อมูลที่ Insert ลงในตาราง 52
รูปที่ 5-5 แสดงผลลัพธ์การ execute ฟังก์ชัน FIND_SALARY 53
รูปที่ 5-6 อธิบายวิธีการเข้าถึงเซตของ SPL 54
รูปที่ 5-7 แสดงผลการอับเดทเงินเดือน 55
รูปที่ 5-8 อธิบายวิธีการเพิ่มโรว์ลงในเซตของ SPL 56
รูปที่ 5-9 แสดงผลการใส่ไฟล์รูปลงตาราง 57
รูปที่ 5-10 แสดงเทคนิคการอ้างถึง BLOB โดยเก็บ ref ไว้ในเซต 57
รูปที่ 6-1 แสดงแผนภาพของระบบงานโรงพยาบาลโดยรวม 59
รูปที่ 6-2 แสดงแผนภาพดาต้าโฟลว์ของระบบงานโรงพยาบาล ระดับที่ 1 60
รูปที่ 6-3 แสดงแผนภาพดาต้าโฟลว์ของการค้นหาข้อมูลผู้ป่วย ระดับที่ 2 60
รูปที่ 6-4 แสดงแผนภาพดาต้าโฟลว์ของการวินิจฉัย ระดับที่ 2 61
รูปที่ 6-5 แสดงแผนภาพดาต้าโฟลว์ของการตรวจแลป ระดับที่ 2 61
รูปที่ 6-6 แสดงแผนภาพดาต้าโฟลว์ของการเข้าพัก ระดับที่ 2 62
รูปที่ 6-7 แสดงแผนภาพดาต้าโฟลว์ของการคิดค่าบริการ ระดับที่ 2 62
รูปที่ 6-8 แสดงความสัมพันธ์เอนทิตี้แบบต่างๆ โดยใช้ ER แบบ CASE TOOL 71
รูปที่ 6-9 แสดงการแปลงความสัมพันธ์จาก many-to-many ไปเป็นความสัมพันธ์เป็นของแบบ Multivalue 72
รูปที่ 6-10 แสดงการแปลงความสัมพันธ์จาก one-to-many ไปเป็นความสัมพันธ์เป็นของแบบเชิงเวลา 72
รูปที่ 6-11 แสดงแผนภาพฐานข้อมูลของระบบงานโรงพยาบาล 73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในระบบการจัดการฐานข้อมูลปัจจุบัน โมเดลข้อมูล (Data Model) ที่นิยมใช้กันอย่างแพร่หลายก็คือ ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งเป็นฐานข้อมูลที่ง่าย และสามารถรองรับการใช้งานได้อย่างกว้างขวาง แต่ฐานข้อมูลชนิดนี้ก็มีข้อจำกัดอยู่หลายประการ ได้แก่ ชนิดของข้อมูล ซึ่งมีแต่ข้อมูลชนิดง่ายๆ ไม่ค่อยมีความซับซ้อน ทำให้ไม่สามารถรองรับงานบางประเภทที่ใช้ชนิดข้อมูลที่มีความซับซ้อนได้ รวมถึงการจัดเก็บลงฐานข้อมูลต้องเป็นไปตามทฤษฎีนอร์มอลฟอร์ม (Normal Form Theorem) จึงจะช่วยกำจัดความซ้ำซ้อนลงได้ ซึ่งข้อจำกัดนี้ทำให้ฐานข้อมูลเชิงสัมพันธ์ไม่มีความสามารถในการจัดการกับข้อมูลเชิงเวลา เพราะจะทำให้ไม่เป็นไปตามทฤษฎีนอร์มอลฟอร์ม

จากข้อจำกัดต่างๆ ของฐานข้อมูลเชิงสัมพันธ์ ทำให้เกิดปัญหาในการจัดเก็บข้อมูลลงฐานข้อมูลในระบบฐานข้อมูลเชิงธุรกิจทั่วๆ ไป ซึ่งมีผลต่อความก้าวหน้าของการพัฒนาการจัดเก็บข้อมูล ดังนั้นจึงได้มีโมเดลข้อมูลชนิดใหม่เกิดขึ้น เพื่อแก้ไขข้อจำกัดต่างๆ ของฐานข้อมูลเชิงสัมพันธ์นี้ โมเดลตัวหนึ่งที่น่าสนใจในการนำมาศึกษาก็คือ ฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object-Relational Database)

ฐานข้อมูลเชิงวัตถุสัมพันธ์ เป็นฐานข้อมูลที่นำแนวคิดเชิงวัตถุ มาประยุกต์ใช้เข้ากับฐานข้อมูลเชิงสัมพันธ์ เพื่อให้ฐานข้อมูลเชิงสัมพันธ์มีความสามารถทางด้านแนวคิดเชิงวัตถุต่างๆ นอกจากนี้ยังทำให้ฐานข้อมูลมีชนิดข้อมูลที่มีความซับซ้อนเพิ่มขึ้นด้วย อาทิเช่น ข้อมูลชนิด เซต (set) หรือ ลิสต์ (list) ซึ่งเป็นชนิดข้อมูลที่สามารถแบ่งแยกย่อยได้อีก (non-atomic) ข้อมูลชนิด บล็อบ (BLOB) หรือ คล็อบ (CLOB) ซึ่งเป็นข้อมูลชนิดพิเศษที่ใช้จัดเก็บไบนารี่ไฟล์และไฟล์ข้อความขนาดใหญ่ ตามลำดับ ซึ่งสิ่งต่างๆ เหล่านี้ ทำให้ความสามารถในฐานข้อมูลเชิงวัตถุสัมพันธ์ เพิ่มขึ้นจากฐานข้อมูลเชิงสัมพันธ์ แต่ยังมองเห็นฐานข้อมูลเป็นฐานข้อมูลเชิงสัมพันธ์อยู่ ซึ่งเป็นสิ่งที่ผู้ใช้คุ้นเคยมานาน

จากความสามารถของฐานข้อมูลเชิงวัตถุสัมพันธ์ดังกล่าว ทำให้สามารถพัฒนาให้ฐานข้อมูลเก็บข้อมูลเชิงเวลาได้โดยมีการจัดการกับความซ้ำซ้อนได้อย่างมีประสิทธิภาพ ซึ่งจะมีประโยชน์อย่างมากในการใช้งานกับการจัดเก็บข้อมูลในยุคปัจจุบัน

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาและออกแบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ให้มีความสามารถในการรองรับการประยุกต์เชิงเวลาได้

1.2.2 เพื่อพัฒนาระบบงานเชิงเวลาให้มีความสมบูรณ์ในด้านการจัดการกับข้อมูลเชิงเวลาได้

1.2.3 เพื่อเปรียบเทียบให้เห็นถึงความสามารถ ของการใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์กับฐานข้อมูลเชิงสัมพันธ์ ในการนำมาใช้กับการประยุกต์เชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะทำการศึกษาข้อมูลเชิงเวลา ที่มีผลกระทบต่อการจัดการกับฐานข้อมูลเชิงสัมพันธ์ และการแก้ปัญหาในการประยุกต์ใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์กับข้อมูลเชิงเวลา โดยจะได้ทำการออกแบบและพัฒนาการจัดการบนฐานข้อมูลเชิงวัตถุสัมพันธ์ ซึ่งในการศึกษาครั้งนี้ ได้เลือกใช้ ระบบการจัดการฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ (Informix Universal Server DBMS) ซึ่งเป็นระบบการจัดการฐานข้อมูลที่ใช้โมเดลข้อมูลแบบฐานข้อมูลเชิงวัตถุสัมพันธ์มาเป็นกรณีศึกษา

ในการประยุกต์ให้ระบบการจัดการฐานข้อมูลสามารถรองรับแอพพลิเคชันเชิงเวลาได้นั้น ต้องมีการเพิ่มโมดูลพิเศษบางส่วนในระบบการจัดการฐานข้อมูลของอินฟอร์มิกซ์เข้าไปบางส่วน รวมถึงฐานข้อมูลที่จะออกแบบต้องมีลักษณะการออกแบบพิเศษเพิ่มเติม คือ มองว่าแต่ละแถวเป็นหนึ่งออบเจ็กต์ ที่สามารถมีค่าแอตทริบิวต์หลายค่าได้ (Multi-value Attribute) โดยแต่ละแอตทริบิวต์นั้นจะประกอบไปด้วยค่าแอตทริบิวต์นั้น และช่วงเวลาที่แอตทริบิวต์นั้นมีค่า ซึ่งจะจัดเก็บในรูปของเซต ซึ่งทำให้รองรับข้อมูลเชิงเวลาได้ โดยมีการจัดการกับความซ้ำซ้อนได้อย่างมีประสิทธิภาพ และต้องมีการพัฒนาเพิ่มเติมในส่วนของระบบการจัดการฐานข้อมูล เพื่อให้สามารถค้นหา แก้ไข เพิ่มเติมข้อมูลเชิงเวลาได้อย่างถูกต้อง

กรณีศึกษาในการสร้างฐานข้อมูล เป็นฐานข้อมูลที่จัดเก็บเกี่ยวกับประวัติคนไข้ในโรงพยาบาล อาทิเช่น ประวัติทั่วไปของคนไข้, ประวัติการตรวจรักษาและวินิจฉัยโรค และประวัติผลทางห้องปฏิบัติการของคนไข้ ซึ่งจะมีหลายแอตทริบิวต์ของคนไข้ที่มีการเปลี่ยนแปลงตามเวลาได้ ขอบเขตของการพัฒนาต้องสามารถจัดการในการ แทรก, ลบ, แก้ไข และค้นหารายการต่างๆ ที่เกี่ยวกับคนไข้ได้ โดยสามารถรายงานแสดงผลและติดต่อกับผู้ใช้ ผ่านทางโปรแกรมวิชวลเบสิค (Visual Basic) ที่พัฒนาขึ้น ใช้ติดต่อกับฐานข้อมูล โดยมีคาต้าไดเรกเตอร์ (Data Director) ซึ่งเป็นเครื่องมือที่ใช้ติดต่อกับระบบการจัดการฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ได้ โดยระบบที่พัฒนาขึ้นต้องสามารถรองรับข้อมูลเชิงเวลา และคำถามเชิงเวลาที่นิยมนำมาใช้งานได้อย่างถูกต้อง

1.4 วิธีการดำเนินงาน

การดำเนินงานจะแบ่งออกเป็น 4 ขั้นตอน ดังนี้ คือ

ขั้นที่ 1 เป็นการศึกษาทฤษฎีพื้นฐานต่างๆ ที่จำเป็น ได้แก่ ทฤษฎีเชิงเวลา แนวคิดเชิงวัตถุ รวมถึงศึกษาเกี่ยวกับโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ที่จะนำมาใช้งาน พร้อมทั้งวิเคราะห์ถึงปัญหาและความเป็นไปได้ในการนำทฤษฎีเชิงเวลามาใช้ในโมเดลข้อมูลชนิดต่างๆ

ขั้นที่ 2 เป็นการศึกษาเครื่องมือที่จะนำมาเป็นกรณีศึกษา นั่นคือระบบการจัดการฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ ซึ่งเป็นระบบการจัดการฐานข้อมูลที่ใช้โมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยศึกษาถึงการติดตั้ง การทำงานต่างๆ การใช้งานระบบ รูปแบบของชนิดข้อมูลที่ใช้งานได้ ภาษาที่ใช้งาน และเครื่องมือที่ช่วยในการทำงานอื่นๆ ซึ่งจะช่วยให้การพัฒนาฐานข้อมูลให้ใช้กับข้อมูลเชิงเวลาเป็นไปได้อย่างสะดวกและรวดเร็วยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 เป็นการพัฒนาเครื่องมือศึกษา ให้สามารถรองรับและจัดการกับข้อมูลเชิงเวลาได้อย่างถูกต้อง ตรงตามวิธีการที่ประยุกต์ขึ้นและทฤษฎีต่างๆ โดยการประยุกต์ใช้ความสามารถของโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ให้มีประโยชน์ต่อการพัฒนา รวมทั้ง ทดสอบโดยการจำลองฐานข้อมูลง่ายๆ และจัดการกับข้อมูลเชิงเวลานั้น ๆ ให้ครอบคลุมปัญหาเชิงเวลาได้ทุกปัญหา เพื่อประโยชน์ต่อไปในการพัฒนาในระดับของผู้เขียนโปรแกรม

ขั้นที่ 4 พัฒนาโปรแกรมประยุกต์จำลองเชิงเวลา ในการจัดเก็บข้อมูลประวัติของคนไข้ในโรงพยาบาล ซึ่งมีข้อมูลต่างๆ ที่เปลี่ยนแปลงตามเวลามากมาย โดยชี้ให้เห็นถึงความสามารถของระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ที่ประยุกต์ให้ใช้ได้กับข้อมูลเชิงเวลาแล้ว ในการจัดการกับข้อมูลเชิงเวลานั้นๆ รวมถึงสามารถออกแบบและพัฒนาโปรแกรมให้มีความครอบคลุมในการจัดการข้อมูลเชิงเวลา และตอบคำถามเชิงเวลาต่างๆ ที่มักจะเกิดขึ้นในการปฏิบัติงานจริงได้อย่างถูกต้อง เพื่อประโยชน์ต่อไปในการพัฒนาในระดับของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเชิงเวลา

2.1 ฐานข้อมูลเชิงเวลา (Temporal Databases)

ฐานข้อมูลเชิงเวลา เป็นฐานข้อมูลที่บันทึกข้อมูลที่เปลี่ยนแปลงตามเวลา (time-varying) ฐานข้อมูลที่มีการเปลี่ยนแปลงจะใช้เมื่อมีการเก็บข้อมูลสถานะปัจจุบันในขณะนั้นในฐานข้อมูล (เช่น ปริมาณสินค้าในขณะนั้น, เงินเดือนและงานของพนักงานในขณะนั้น เป็นต้น) ฐานข้อมูลชนิดนี้จะเปลี่ยนแปลงตามการเปลี่ยนแปลงจริง และสามารถพิจารณาให้เป็นฐานข้อมูลที่เปลี่ยนแปลงตามเวลาได้ ส่วนฐานข้อมูลในอดีต (historical database) จะหมายถึง ฐานข้อมูลที่มีการเก็บข้อมูลในอดีตเท่านั้น แต่ฐานข้อมูลเชิงเวลาจะสามารถเก็บข้อมูลในอนาคตได้ เช่น การวางตารางเวลา หรือแผนงาน

นิยามของฐานข้อมูลเชิงเวลาก็คือ ฐานข้อมูลที่รองรับเวลาบางรูปแบบ ซึ่งไม่รวมเวลาที่ผู้ใช้กำหนด โดยเวลาที่ผู้ใช้กำหนด ได้แก่ ข้อมูลจำพวกที่มีแอตทริบิวต์เป็นชนิดวันที่ (date) หรือ เวลา (time) เช่น วันกิด หรือ วันที่จ้างงาน เป็นต้น ซึ่งจะเป็นเพียงการเพิ่มคอลัมน์นั้นเข้าไปในตาราง ไม่ได้ดูข้อมูลตัวนั้นเป็นเชิงเวลา ซึ่งข้อมูลนั้นจะไม่มีการเปลี่ยนแปลงตลอดไป ซึ่งฐานข้อมูลเชิงเวลานั้น จะต้องบันทึกข้อมูลที่มีการเปลี่ยนแปลงตามเวลาฐานข้อมูลเชิงเวลา ได้รับการศึกษาค้นคว้ากันมากมาย เพราะเนื่องจากระบบจัดการฐานข้อมูลเชิงพาณิชย์ เช่น ออราเคิล, ไซเบส, ไมโครซอฟท์ ซีเควลซิร์ฟเวอร์ ยังคงใช้การจัดเก็บเฉพาะข้อมูลที่เป็นจริงในปัจจุบันอยู่ โดยไม่คำนึงถึงข้อมูลในอดีตและในอนาคต จึงจำเป็นที่จะต้องมีการพัฒนาระบบจัดการฐานข้อมูลเหล่านี้ให้สามารถรองรับข้อมูลเชิงเวลาให้ได้

ข้อมูลเชิงเวลาที่จัดเก็บในฐานข้อมูลเชิงเวลานั้น จะแตกต่างจากข้อมูลที่ถูกเก็บในฐานข้อมูลทั่วๆ ไป ตรงที่จะเก็บช่วงเวลา (Timestamp) ไว้กับข้อมูล เพื่อกำกับว่าข้อมูลนั้นมีค่าเป็นจริงเมื่อใด หรือเก็บข้อมูลเมื่อใด ซึ่งจะทำให้ฐานข้อมูลสามารถเก็บข้อมูลในอดีต ปัจจุบัน และข้อมูลในอนาคตได้

ช่วงเวลาที่ใช้ในการฐานข้อมูลเชิงเวลา แบ่งออกเป็น 3 ชนิด ดังนี้คือ

1) *วาลิดไทม์ (Valid Time)* คือช่วงเวลาที่ข้อมูลมีค่าเป็นจริง
2) *ทรานส์แอกชันไทม์ (Transaction Time)* คือช่วงเวลาที่ข้อมูลถูกเก็บลงฐานข้อมูล
3) *เวลาที่กำหนดโดยผู้ใช้ (User-Defined Time)* เป็นข้อมูลชนิดวันที่และเวลา เช่น วันเกิด เวลาปฏิบัติงาน เป็นต้น

จะเห็นได้ว่า ช่วงเวลาวาลิดไทม์และช่วงเวลาทรานส์แอกชันไทม์ ไม่จำเป็นต้องเป็นช่วงเวลาเดียวกันสำหรับข้อมูลตัวเดียวกัน ตัวอย่างเช่น ฐานข้อมูลเก็บชื่อของพนักงานคนหนึ่งไว้ตั้งแต่ปี 2540หากพนักงานคนนี้มีการเปลี่ยนแปลงชื่อในปี 2543 แต่มาตรวจพบว่าชื่อมีการเปลี่ยนแปลงในปี 2544 จะทำให้วาลิดไทม์ กับ ทรานส์แอกชันไทม์ เป็น 2543 และ 2544 ตามลำดับ ซึ่งมีค่าไม่เท่ากัน

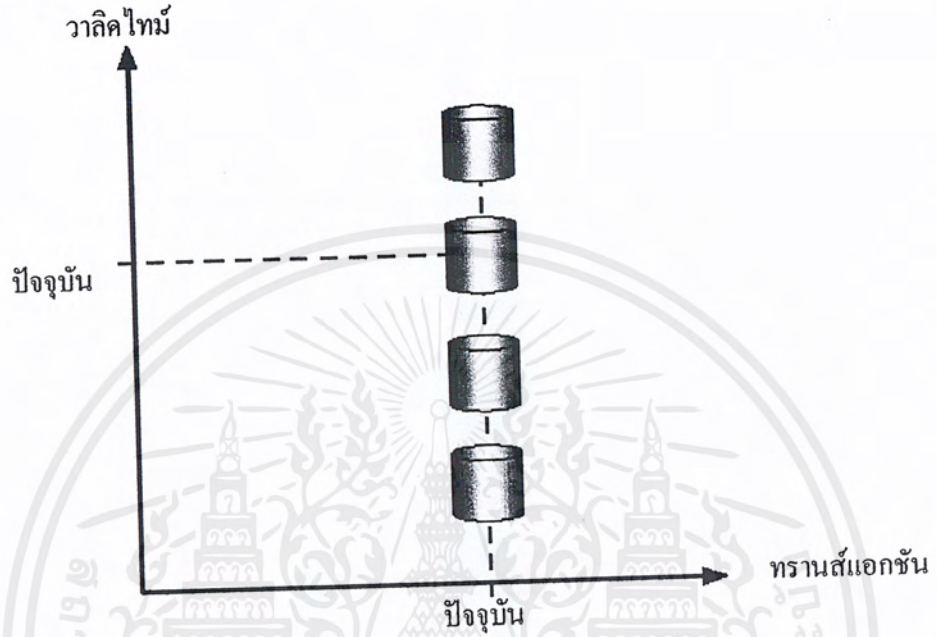
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ประเภทของฐานข้อมูลเชิงเวลา

ช่วงเวลาวาลิดไทม์ และทรานส์แอกชันไทม์ จะทำให้เกิดฐานข้อมูลในหลายประเภท ดังนี้

2.2.1 ฐานข้อมูลวาลิดไทม์ (Valid-Time Database)

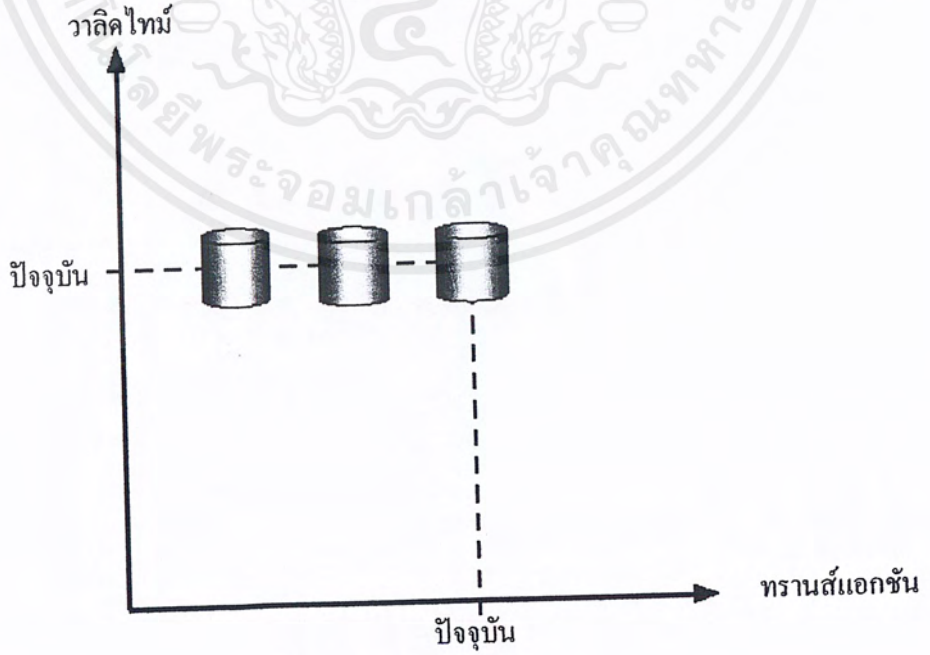
หรือฐานข้อมูลประวัติ (Historical Database) จะเก็บเฉพาะวาลิดไทม์ ซึ่งจะเก็บข้อมูลในอดีตกับปัจจุบัน แสดงดังรูปที่ 2-1



รูปที่ 2-1 ฐานข้อมูลประวัติ

2.2.2 ฐานข้อมูลทรานส์แอกชันไทม์ (Transaction-Time Database)

หรือฐานข้อมูลย้อนกลับ (Rollback Database) ซึ่งจะเก็บเฉพาะทรานส์แอกชันไทม์ แสดงดังรูปที่ 2-2



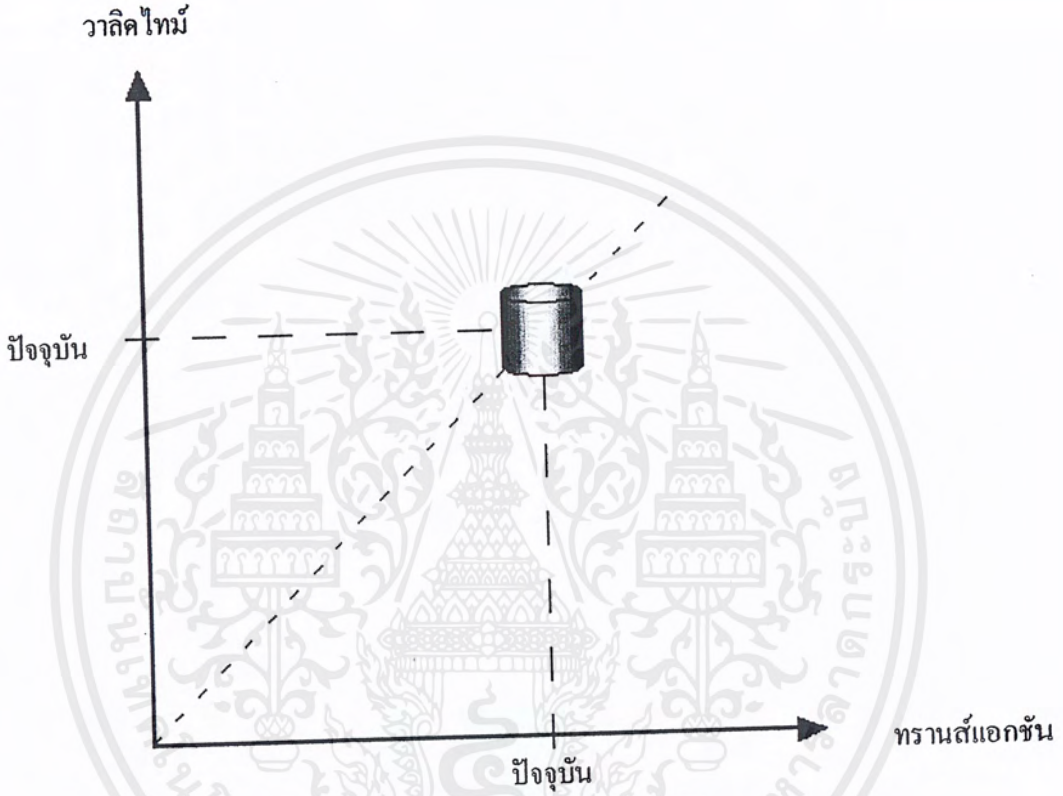
รูปที่ 2-2 ฐานข้อมูลย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ฐานข้อมูลไบเทมโพรอล (Bitemporal Database)

เก็บทั้งวาลิดไทม์และทรานส์แอกชันไทม์ จะทำให้เก็บข้อมูลในอดีต อนาคต และปัจจุบันได้

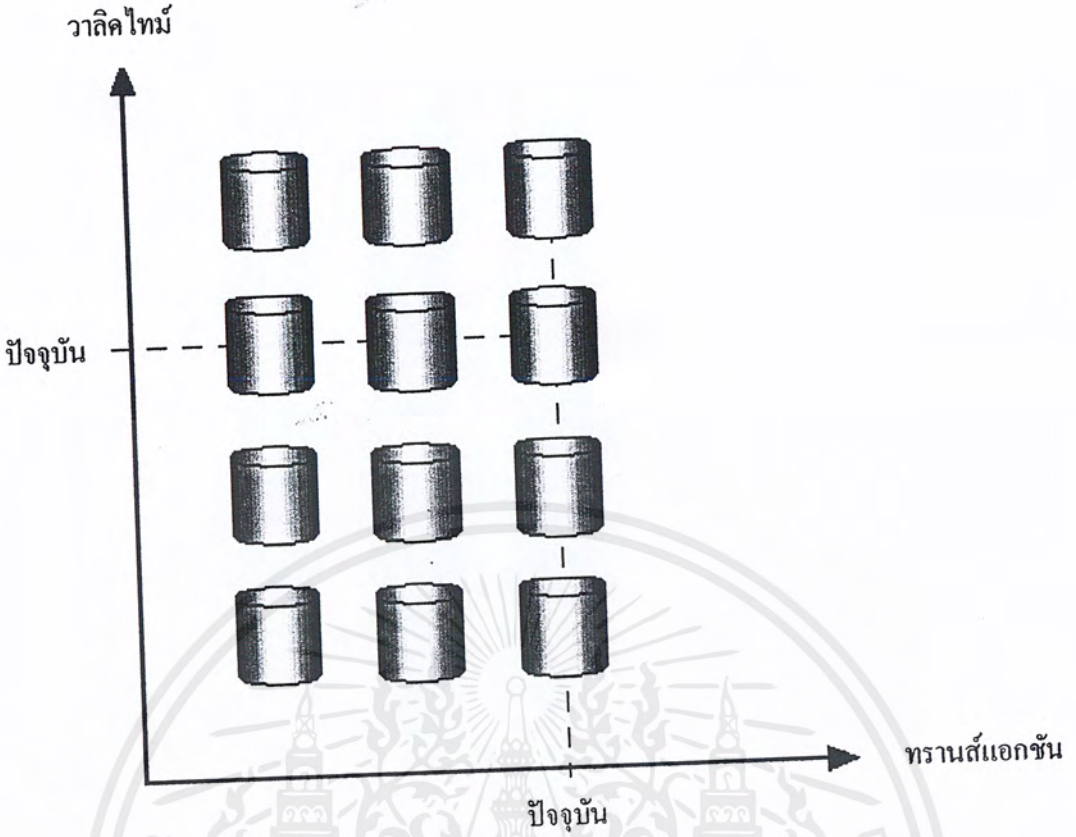
ในระบบจัดการฐานข้อมูลเชิงพาณิชย์จะเก็บเฉพาะข้อมูลที่เป็นจริงในปัจจุบันเท่านั้น ฐานข้อมูลชนิดนี้จะเรียกว่าฐานข้อมูลสแนปช็อท (Snapshot Database) โดยฐานข้อมูลสแน็ปช็อทในแบบของวาลิดไทม์และทรานส์แอกชันไทม์ แสดงดังรูปที่ 2-3



รูปที่ 2-3 ฐานข้อมูลสแน็ปช็อท

และฐานข้อมูลแบบไบเทมโพรอล อาจจะเป็นได้ทั้งตารางสแน็ปช็อท คือเก็บตารางวาลิดไทม์ คือเก็บว่าข้อมูลเป็นจริงเมื่อใด โดยมีลักษณะฐานข้อมูล หรือเก็บตารางทรานส์แอกชันไทม์ คือ เก็บว่าข้อมูลถูกบันทึกลงในฐานข้อมูลเมื่อใด หรืออาจจะเป็นตารางไบเทมโพรอล ที่เก็บทั้งวาลิดไทม์และทรานแซคชันไทม์ก็ได้ โดยมีลักษณะของฐานข้อมูลดังรูปที่ 2-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 ฐานข้อมูลไบเทมโพรอล

2.3 ความซ้ำซ้อน (Duplicates)

Name	Status	FROM_DATE	TO_DATE
Kenneth Robert	Serious	1997-11-19	1997-11-21
Alexis May	Serious	1997-11-19	1997-11-27
Natalie Sue	Serious	1997-11-19	1997-11-25
Kelsey Ann	Serious	1997-11-19	1997-11-26
Joel Steven	Critical	1997-11-19	1997-11-20
Joel Steven	Serious	1997-11-20	1997-11-26
Kenneth Robert	Fair	1997-11-21	1998-01-03
Alexis May	Fair	1997-11-27	1998-01-11
Alexis May	Fair	1997-12-02	9999-12-31
Alexis May	Fair	1997-12-02	9999-12-31

ตารางที่ 2-1 ตารางวาลิดไทม์ของข้อมูลเด็กในโรงพยาบาลแห่งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2-1 เป็นข้อมูลของเด็กในโรงพยาบาลแห่งหนึ่งที่มีอาการต่างๆ กัน ในช่วงเวลาต่างๆ โดยมีคอลัมน์ From_date และ To_date เป็นช่วงเวลา (Timestamp) ที่ข้อมูลในแถวนั้นๆ มีค่าวาลิด ซึ่งถ้าค่า To_date เป็น "9999-12-31" จะหมายถึงข้อมูลในแถวนั้นมีค่าวาลิดมาจนถึงปัจจุบัน

จากตารางเชิงเวลานี้ จะพบว่าข้อมูลมีความซ้ำซ้อนอยู่ ซึ่งเราสามารถจำแนกความซ้ำซ้อนออกได้เป็น 4 ประเภท ดังนี้

2.3.1 ความซ้ำซ้อนแบบค่าเท่ากัน (Value Equivalent Duplicate)

เป็นความซ้ำซ้อนที่แถวนั้นมีค่าของคอลัมน์ทุกคอลัมน์เท่ากันหมด ยกเว้นคอลัมน์ช่วงเวลา (Timestamp Column) ดังเช่น 3 แถวสุดท้ายของตาราง มีความซ้ำซ้อนแบบค่าเท่ากัน เพราะ 2 คอลัมน์แรกมีค่าเท่ากัน

2.3.2 ความซ้ำซ้อนแบบนอนซีเควนต์ (Nonsequenced Duplicate)

เป็นความซ้ำซ้อนที่ทุกคอลัมน์รวมทั้งคอลัมน์ช่วงเวลา (Timestamp Column) มีค่าเท่ากันหมด ดังเช่น 2 แถวสุดท้ายของตารางมีความซ้ำซ้อนแบบนอนซีเควนต์นี้

2.3.3 ความซ้ำซ้อนแบบซีเควนต์ (Sequenced Duplicate)

เป็นความซ้ำซ้อนที่เกิดขึ้นในบางช่วงของเวลา คือ มีค่าซ้ำซ้อนกันในช่วงเวลาใดเวลาหนึ่ง ดังเช่น 3 แถวสุดท้ายของตาราง เป็นความซ้ำซ้อนแบบซีเควนต์ ในช่วง "1997-11-27" ถึง "1998-01-11" เป็นต้น

2.3.4 ความซ้ำซ้อนแบบปัจจุบัน (Current Duplicate)

เป็นความซ้ำซ้อนที่เกิดขึ้น เมื่อเทียบกับ ณ เวลาปัจจุบัน เช่น สมมติว่าวันนี้มีค่าวันที่เป็น "1998-01-06" จะทำให้ 3 แถวสุดท้ายของตารางมีความซ้ำซ้อนแบบปัจจุบันเกิดขึ้น แต่ถ้าวันนี้มีค่าวันที่เป็น "1998-01-12" จะทำให้ 2 แถวสุดท้ายมีความซ้ำซ้อนกันเท่านั้น

ตารางที่ 2-2 จะแสดงให้เห็นถึงความสัมพันธ์ของความซ้ำซ้อนแบบต่างๆ อาทิเช่น หากเป็นความซ้ำซ้อนแบบนอนซีเควนต์แล้ว จะเป็นความซ้ำซ้อนแบบซีเควนต์ และความซ้ำซ้อนแบบปัจจุบันด้วย เป็นต้น

ความซ้ำซ้อน	ซีเควนต์	ปัจจุบัน	ค่าเท่ากัน	นอนซีเควนต์
ซีเควนต์	✓		✓	
ปัจจุบัน	✓	✓	✓	
ค่าเท่ากัน			✓	
นอนซีเควนต์	✓		✓	✓

ตารางที่ 2-2 ตารางแสดงความสัมพันธ์ของความซ้ำซ้อนแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การป้องกันความซ้ำซ้อน

การป้องกันความซ้ำซ้อนสามารถทำได้ตอนสร้างตารางโดยจำแนกตามประเภทของความซ้ำซ้อนได้ ดังนี้

2.4.1 การป้องกันความซ้ำซ้อนแบบค่าเท่ากัน

จากตารางที่แล้ว เราสามารถเขียนภาษา SQL ในการสร้างตารางที่ป้องกันความซ้ำซ้อนแบบค่าเท่ากัน ได้ดังนี้

```
CREATE TABLE NICUStatue (
        Name        CHAR (15),
        Status      CHAR (8),
        From_date   DATE,
        To_date     DATE,
        UNIQUE ( Name, Status ) );
```

ซึ่งเป็นการบังคับว่าคอลัมน์ทุกคอลัมน์ ยกเว้นคอลัมน์ที่เป็น Timestamp คอลัมน์ต้องเป็นเอกลักษณ์ (Unique)

2.4.2 การป้องกันความซ้ำซ้อนแบบนอนซีเควนซ์

```
CREATE TABLE NICUStatue (
    ...
    UNIQUE ( Name, Status, From_date, To_date ) );
```

ซึ่งเป็นการบังคับความถูกต้องให้ทุกคอลัมน์เป็นเอกลักษณ์

2.4.3 การป้องกันความซ้ำซ้อนแบบปัจจุบัน

```
CREATE TABLE NICUStatus (
    ...
CHECK (NOT EXISTS (SELECT N1.SSN
                   FROM NICUStatus AS N1
                   WHERE 1 < (SELECT COUNT(Name)
                    FROM NICUStatus AS N2
                    WHERE N1.Name = N2.Name AND N1.Status = N2.Status
                    AND N1.from_date <= CURRENT_DATE
                    AND CURRENT_DATE < N1.to_date
                    AND N2.from_date <= CURRENT_DATE
                    AND CURRENT_DATE < N2.to_date )))
```

ซึ่งลักษณะของการสร้างตารางจะบังคับความถูกต้องว่าค่าข้อมูลในปัจจุบันนั้นมีค่าเดียว โดยจะไม่ซ้ำซ้อนกับข้อมูล ณ เวลาปัจจุบัน แต่ข้อมูลอาจมีการซ้ำซ้อนกันได้ในอนาคตแม้ตารางจะไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนแปลง แต่หากทราบว่าแอพพลิเคชันไม่มีการจัดเก็บข้อมูลที่เป็นอนาคต เราสามารถบังคับเอกลักษณ์ของตาราง ได้โดย

```
CREATE TABLE NICUStatue (
    ...
    UNIQUE ( Name, Status, To_date ) );
```

เนื่องจากข้อมูลในปัจจุบันจะมีค่า to_dateเดียวกัน คือ "9999-12-31"

2.4.4 การป้องกันความซ้ำซ้อนแบบซีเควนต์

```
CREATE TABLE NICUStatus (
    ...
    CHECK (NOT EXISTS (SELECT N1.Name
        FROM NICUStatus AS N1
        WHERE 1 < (SELECT COUNT (NAME)
          FROM NICUStatus AS N2
          WHERE N1.Name = N2.Name AND N1.Status = N2.Status
        AND N1.from_date < N2.to_date AND N2.from_date < N1.to_date ))))
```

การบังคับความซ้ำซ้อนของข้อมูลแบบซีเควนซ์ จะคล้ายกับการป้องกันความซ้ำซ้อนแบบปัจจุบัน แต่แตกต่างกันตรงเงื่อนไขในการตรวจสอบเวลาเริ่มต้นกับเวลาสิ้นสุด โดยจะตรวจสอบค่าของข้อมูลที่ช่วงเวลามีการซ้อนทับกัน

2.5 การค้นหาข้อมูลของตารางวาลิดไทม์

ในที่นี้จะยกตัวอย่างตาราง LOT_LOC ซึ่งเป็นตารางที่เก็บการเปลี่ยนแปลงของจำนวนวัวในแต่ละล็อทที่อยู่ในแต่ละคอกของแต่ละฟาร์ม ดังแสดงในตารางที่ 2-3

FDYD_ID	LOT_ID_NUM	PEN_ID	HD_CNT	FROM_DATE	TO_DATE
1	137	1	17	1998-02-07	1998-02-18
1	219	1	43	1998-02-25	1998-03-01
1	219	1	20	1998-03-01	1998-03-14
1	219	2	23	1998-03-01	1998-03-14
1	219	2	43	1998-03-14	9999-12-31
1	374	1	14	1998-02-20	9999-12-31

ตารางที่ 2-3 ตาราง LOT_LOC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางนี้เป็นตารางวาลิดไทม์ ซึ่งบันทึกข้อมูลที่มีค่าเป็นจริงและสถานะตามแต่ละช่วงเวลา ตัวอย่างเช่น มีวัวจำนวน 17 ตัวอยู่ในคอกที่ 1 11 วัน แล้วย้ายออกเมื่อวันที่ 1998-02-18 , มีวัวจำนวน 23 ตัวจากล็อท 219 ย้ายจากคอกที่ 1 ไปคอกที่ 2 วันที่ 1998-03-01 แล้วย้ายวัว 20 ตัวที่เหลือไปคอกที่ 2 วันที่ 1998-03-14 และยังคงอยู่ในคอกนั้นจนถึงปัจจุบัน เป็นต้น

2.5.1 การตอบคำถามเชิงเวลาโดยใช้การ Projection และ Selection

เราสามารถแบ่งประเภทของการค้นหาออกได้เป็นหลายประเภท ดังนี้

2.5.1.1 การค้นหาข้อมูลโดยปกติทั่วๆ ไป (Coventional Query ; Non-temporal Query)

คือการค้นหาคำถามที่ไม่คำนึงถึงเวลา เช่น "มีวัวจำนวนกี่ตัวในแต่ละคอกที่มาจากล็อทที่ 219 ฟาร์มที่ 1" จะต้องใช้ภาษา SQL ดังนี้

```
SELECT PEN_ID, HD_CNT
FROM LOT_LOC
WHERE FDYD_ID = 1 AND LOT_ID_NUM = 219
```

2.5.1.2 การค้นหาข้อมูลแบบปัจจุบัน (Current Query)

คือ การค้นหาคำถามที่เกี่ยวข้องกับ ณ เวลาขณะหนึ่ง เช่น "มีวัวจำนวนกี่ตัว ในแต่ละคอกในปัจจุบันนี้ที่มาจากล็อทที่ 219 ฟาร์มที่ 1" จะต้องใช้ภาษา SQL ดังนี้

```
SELECT PEN_ID, HD_CNT
FROM LOT_LOC
WHERE FDYD_ID = 1 AND LOT_ID_NUM = 219
      AND TO_DATE = DATE '9999-12-31"
```

ซึ่งจะได้ผลลัพธ์ดังตารางที่ 2-4

PEN_ID	HD_CNT
2	43

ตารางที่ 2-4 คำตอบของตัวอย่างการตอบคำถามแบบปัจจุบัน

2.5.1.3 การค้นหาข้อมูลแบบซีเควนซ์ (Sequenced Query)

คือ การค้นหาคำถามที่เกี่ยวข้องกับช่วงของเวลาช่วงหนึ่ง เช่น "จงแสดงประวัติตั้งแต่อดีตจนถึงปัจจุบันของจำนวนวัวในแต่ละคอกที่มาจากล็อทที่ 219 ฟาร์มที่ 1" จะต้องใช้ภาษา SQL ดังนี้

```
SELECT PEN_ID, HD_CNT, FROM_DATE, TO_DATE
FROM LOT_LOC
WHERE FDYD  = 1 AND LOT_ID_NUM = 219
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ผลลัพธ์ดังตารางที่ 2-5

PEN_ID	HD_CNT	FROM_DATE	TO_DATE
1	43	1998-02-25	1998-03-01
1	20	1998-03-01	1998-03-14
2	23	1998-03-01	1998-03-14
2	43	1998-03-14	9999-12-31

ตารางที่ 2-5 คำตอบของตัวอย่างการตอบคำถามแบบซีเควนซ์

2.5.1.4 การค้นหาข้อมูลแบบนอนซีเควนต์ (Nonsequenced Query)

คือ การค้นหาคำถามที่ไม่คำนึงถึงช่วงเวลา ซึ่งจะมีลักษณะคล้ายกับการค้นหาข้อมูลโดยปกติทั่วๆ ไป (Nontemporal Query) เช่น "ณ เวลาใดๆ มีวัวกี่ตัวในแต่ละคอกที่มาจากล็อทที่ 219 ในฟาร์มที่ 1" จะต้องใช้ภาษา SQL ดังนี้

```
SELECT PEN_ID, HD_CNT
FROM   LOT_LOC
WHERE FDYD_ID = 1 AND LOT_ID_NUM = 219
```

จะได้ผลลัพธ์ดังตารางที่ 2-6

PEN_ID	HD_CNT
1	43
1	20
2	23
2	43

ตารางที่ 2-6 คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนซ์

2.5.2 การค้นหาข้อมูลโดยใช้การจอย (Join) กันของตาราง

2.5.2.1 ค้นหาข้อมูลแบบปกติทั่ว ๆ ไปโดยใช้การจอย (Conventional Join Query)

พิจารณาคำถามที่ว่า "ในแต่ละคอกมีวัวล็อทใดบ้างที่อยู่ด้วยกัน" จะต้องใช้ SQL ดังต่อไปนี้

```
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AND L1.FDYD_ID = L2.FDYD_ID
AND L1.PEN_ID = L2.PEN_ID
```

2.5.2.2 ค้นหาข้อมูลแบบปัจจุบัน โดยใช้การจอย (Current Join Query)

พิจารณาคำถามที่ว่า "ในแต่ละคอกในปัจจุบันมีวัวล็อทใดบ้างที่อยู่ด้วยกัน" จะต้องใช้ SQL ดังต่อไปนี้

```
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
    AND L1.FDYD_ID = L2.FDYD_ID
    AND L1.PEN_ID = L2.PEN_ID
    AND L1.TO_DATE = '9999-12-31'
    AND L2.TO_DATE = '9999-12-31'
```

2.5.2.3 ค้นหาข้อมูลแบบนอนซีเควนต์โดยใช้การจอย (Nonsequenced Join Query)

พิจารณาคำถามที่ว่า "ณ เวลาใดๆ ในแต่ละคอกมีวัวล็อทใดบ้างที่อยู่ด้วยกัน" จะต้องใช้ SQL ดังต่อไปนี้

```
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
    AND L1.FDYD_ID = L2.FDYD_ID
    AND L1.PEN_ID = L2.PEN_ID
```

จะได้ผลลัพธ์ดังตารางที่ 2-7

L1	L2	PEN_ID
137	219	1
137	219	1
137	374	1
219	374	1
219	374	1

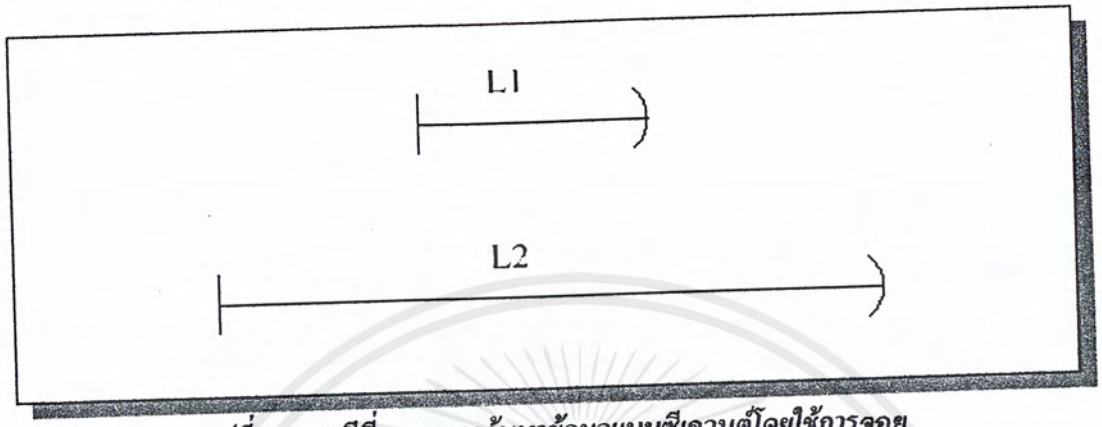
ตารางที่ 2-7 คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนซ์โดยใช้การจอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.4 การค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย (Sequenced Join Query)

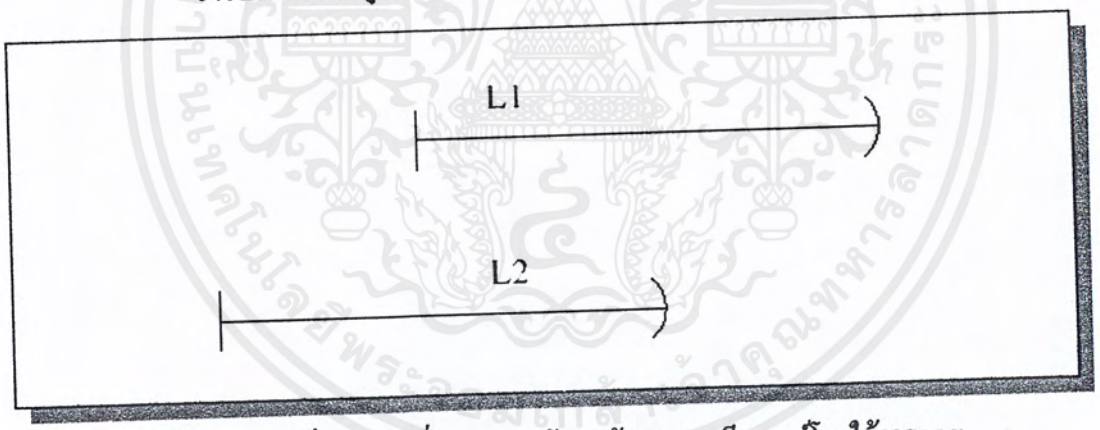
พิจารณาคำถามที่ว่า "จงแสดงประวัติตั้งแต่ในอดีตจนถึงปัจจุบันว่ามีวัวล็อทใดบ้างที่อยู่ด้วยกันในแต่ละคอก" คำถามแบบนี้จะต้องแบ่งช่วงเวลาการซ้อนทับกันของตาราง ออกเป็น 4 กรณีดังนี้

กรณีที่ 1 ช่วงเวลาของตาราง L1 อยู่ในช่วงเวลาของตาราง L2 แสดงดังรูปที่ 2-3



รูปที่ 2-5 กรณีที่ 1 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย

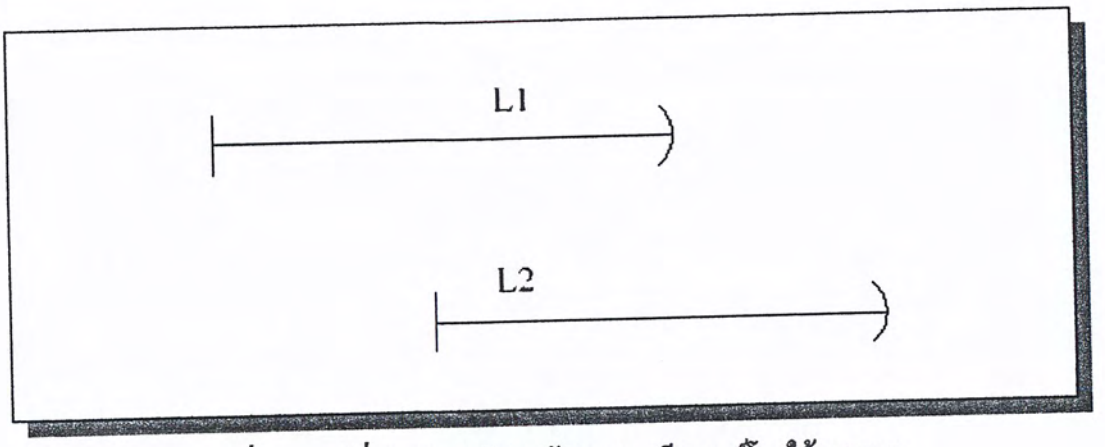
กรณีที่ 2 FROM_DATE ของตาราง L1 มากกว่า FROM_DATEของตาราง L2 แต่น้อยกว่า TO_DATE ของตาราง L2 และTO_DATE ของตาราง L1 มากกว่า TO_DATE ของตาราง L2 แสดงดังรูปที่ 2-4



รูปที่ 2-6 กรณีที่ 2-4 กรณีที่ 2 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย

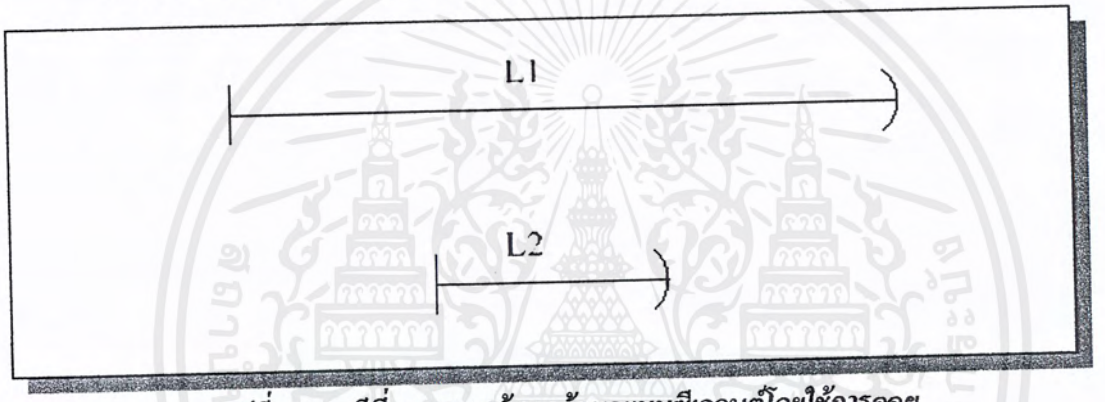
กรณีที่ 3 FROM_DATE ของตาราง L2 มากกว่า FROM_DATE ของตาราง L1 แต่น้อยกว่า TO_DATE ของตาราง L1 และ TO_DATE ของตาราง L2 มากกว่า TO_DATE ของตาราง L1 แสดงดังรูปที่ 2-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-7 กรณีที่ 3 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย

กรณีที่ 4 ช่วงเวลาของตาราง L2 อยู่ในช่วงเวลาของตาราง L1 แสดงดังรูปที่ 2-6



รูปที่ 2-8 กรณีที่ 4 ของการค้นหาข้อมูลแบบซีเควนต์โดยใช้การจอย

ซึ่งสามารถใช้ SQL ในการตอบคำถามดังต่อไปนี้

```
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID, L1.FROM_DATE, L1.TO_DATE
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND L1.FDYD_ID = L2.FDYD_ID
        AND L1.PEN_ID = L2.PEN_ID
        AND L2.FROM_DATE <= L1.FROM_DATE
        AND L1.TO_DATE <= L2.TO_DATE
UNION
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID, L1.FROM_DATE, L2.TO_DATE
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND L1.FDYD_ID = L2.FDYD_ID
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AND L1.PEN_ID = L2.PEN_ID
        AND L1.FROM_DATE > L2.FROM_DATE
        AND L2.TO_DATE < L1.TO_DATE
        AND L1.FROM_DATE < L2.TO_DATE
UNION
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID, L2.FROM_DATE, L1.TO_DATE
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND L1.FDYD_ID = L2.FDYD_ID
        AND L1.PEN_ID = L2.PEN_ID
        AND L2.FROM_DATE > L1.FROM_DATE
        AND L1.TO_DATE < L2.TO_DATE
        AND L2.FROM_DATE < L1.TO_DATE
UNION
SELECT L1.LOT_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID, L2.FROM_DATE, L2.TO_DATE
FROM LOT_LOC AS L1, LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND L1.FDYD_ID = L2.FDYD_ID
        AND L1.PEN_ID = L2.PEN_ID
        AND L2.FROM_DATE >= L1.FROM_DATE
        AND L2.TO_DATE <= L1.TO_DATE
```

จะได้ผลลัพธ์ดังตารางที่ 2-8

LOT_ID_NUM	LOT_ID_NUM	PEN_ID	FROM_DATE	TO_DATE
219	374	1	1998-02-25	1998-02-28
219	374	1	1998-03-01	1998-03-13

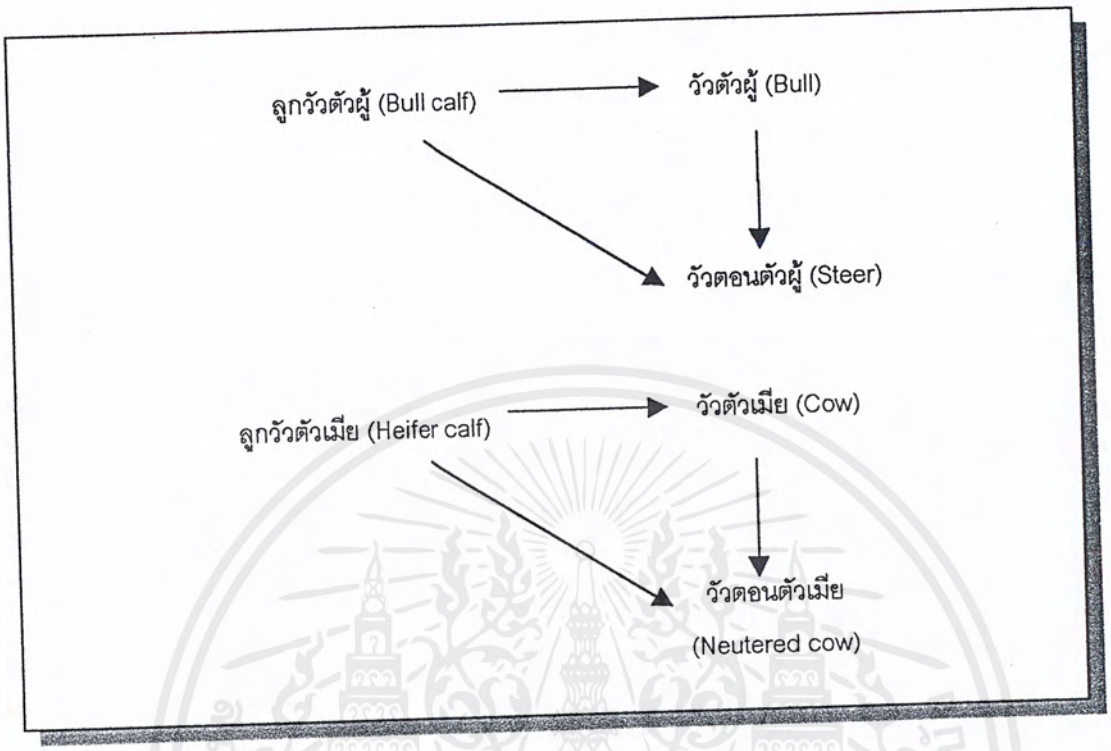
ตารางที่ 2-8 คำตอบของตัวอย่างการตอบคำถามแบบซีเควนซ์โดยใช้การจอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.6 การแก้ไขข้อมูลเชิงเวลา

รูปที่ 2-7 จะแสดงถึงการเปลี่ยนแปลงเพศของวัว



รูปที่ 2-9 การเปลี่ยนแปลงเพศของวัว

เพื่อแสดงให้เห็นถึงการเปลี่ยนแปลงตารางวาลิดไทม์โดยใช้ภาษา SQL จึงขอยกตัวอย่างตาราง เพศวัวดังตารางที่ 2-9

LOT_ID_NUM	GNDR_CODE	FROM_DATE	TO_DATE
101	C	1998-01-01	1998-03-23
101	S	1998-03-23	9999-12-31
234	C	1998-02-17	9999-12-31
799	S	1998-03-12	9999-12-31

ตารางที่ 2-9 ตารางเพศวัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
42795

2.6.1 การเปลี่ยนแปลงแบบปัจจุบัน (Current Modification)

2.6.1.1 การลบแบบปัจจุบัน (Current Deletion)

พิจารณาการเปลี่ยนแปลงที่ว่า "วัวล็อท 234 ย้ายออกจากฟาร์ม" จะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

```
UPDATE LOT
SET TO_DATE = CURRENT_DATE
WHERE LOT_ID_NUM = 234
        AND TO_DATE >= CURRENT_DATE
        AND FROM_DATE < CURRENT_DATE
DELETE FROM LOT
WHERE LOT_ID_NUM = 234
        AND FROM_DATE > CURRENT_DATEถ้า CURRENT DATE = '1998-
```

10-17' จะได้ผลดังตารางที่ 2-10

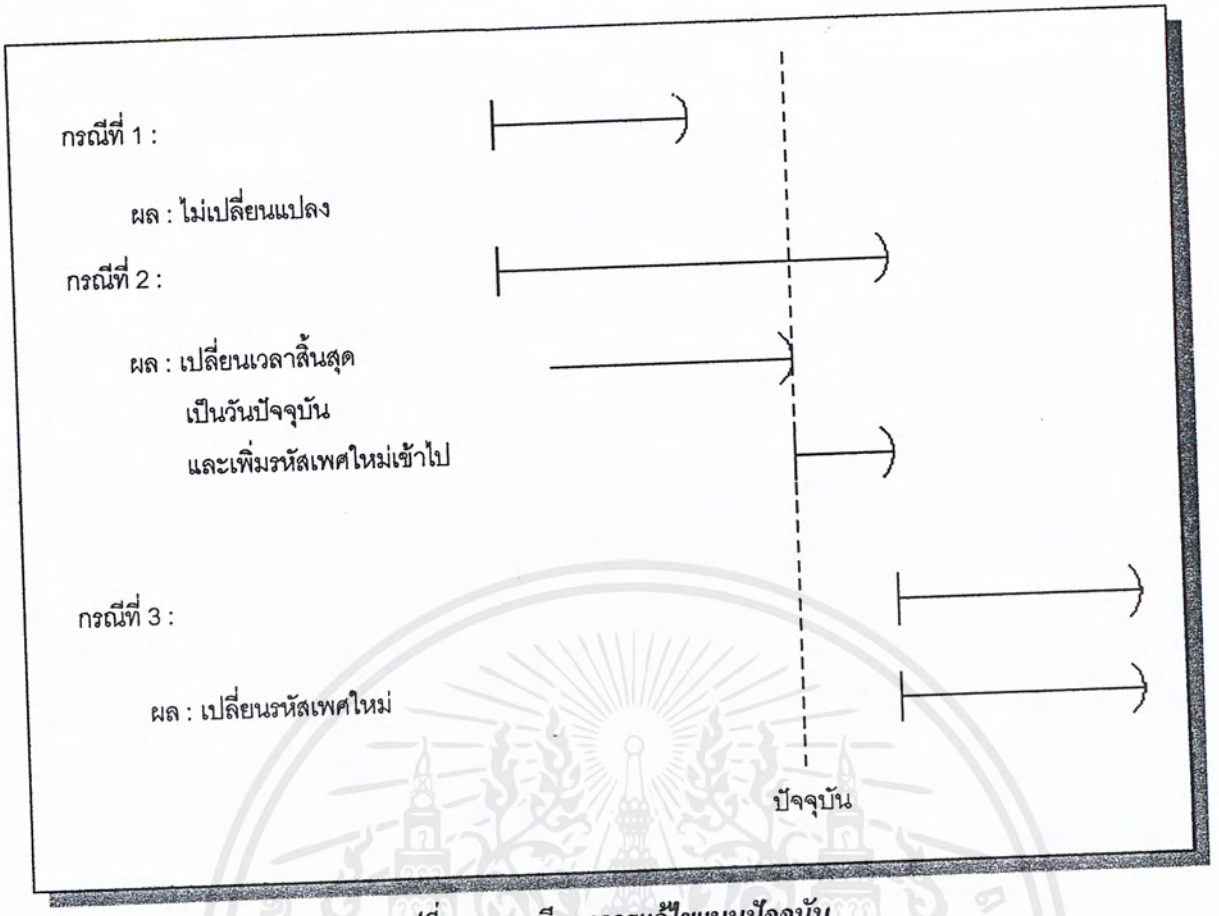
LOT_ID_NUM	GNDR_CODE	FROM_DATE	TO_DATE
101	C	1998-01-01	1998-03-23
101	S	1998-03-23	9999-12-31
234	C	1998-02-17	1998-10-17
799	S	1998-03-12	9999-12-31

ตารางที่ 2-10 ผลของตัวอย่างการลบแบบปัจจุบัน

2.6.1.2 การแก้ไขแบบปัจจุบัน (Current Update)

พิจารณาการเปลี่ยนแปลงที่ว่า "เปลี่ยนวัวล็อท 799 เป็นวัวตอนตัวผู้" การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 3 กรณี แสดงดังรูปที่ 2-8 ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-10 กรณีของการแก้ไขแบบปัจจุบัน

ดังนั้นจะสามารถแก้ไขแบบปัจจุบันได้โดยใช้ SQL ดังต่อไปนี้

```
INSERT INTO LOT
SELECT LOT_ID_NUM, 'S', CURRENT_DATE, TO_DATE
FROM LOT
WHERE LOT_ID_NUM = 799
        AND FROM_DATE <= CURRENT_DATE
        AND TO_DATE > CURRENT_DATE
UPDATE LOT
SET TO_DATE = CURRENT_DATE
WHERE LOT_ID_NUM = 799
        AND GNDR_CODE <> 'S'
        AND FROM_DATE < CURRENT_DATE
        AND TO_DATE > CURRENT_DATEUPDATE LOT
SET GNDR_CODE = 'S'
WHERE LOT_ID_NUM = 799 AND FROM_DATE > = CURRENT_DATE
```

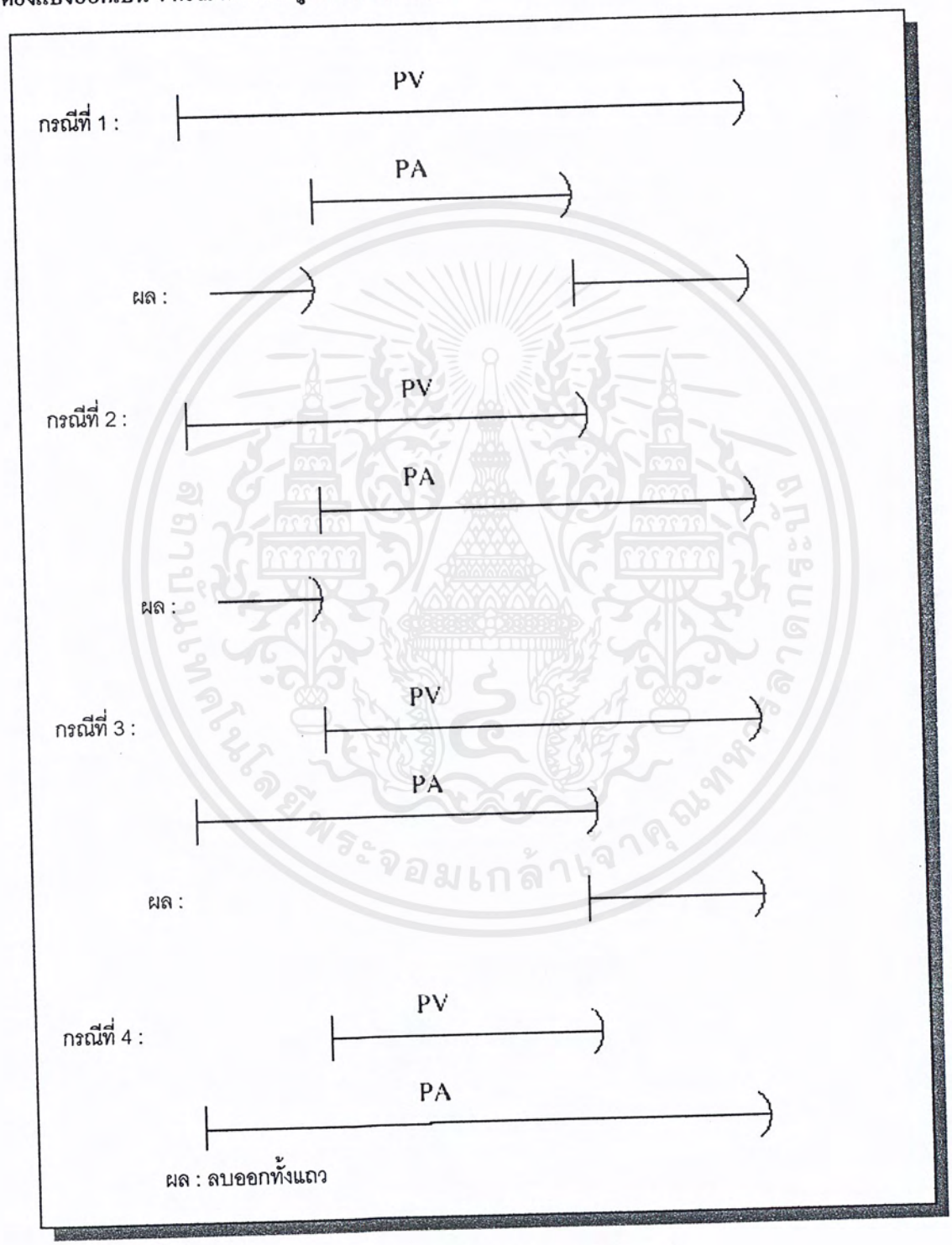
2.6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 การเปลี่ยนแปลงแบบซีเควนซ์ (Sequenced Modification)

2.6.2.1 การลบแบบซีเควนซ์ (Sequenced Deletion)

พิจารณาการเปลี่ยนแปลงที่ว่า "วัวล็อท 234 ย้ายออกจากฟาร์มใน 3 สัปดาห์แรกของเดือนตุลาคม ซึ่งในช่วงเวลานั้นได้มีการวางแผนที่จะเปลี่ยนวัวล็อท 234 นี้เป็นวัวตอนตัวผู้" การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 4 กรณี แสดงดังรูปที่ 2-9 ดังต่อไปนี้



รูปที่ 2-11 กรณีของการลบแบบซีเควนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

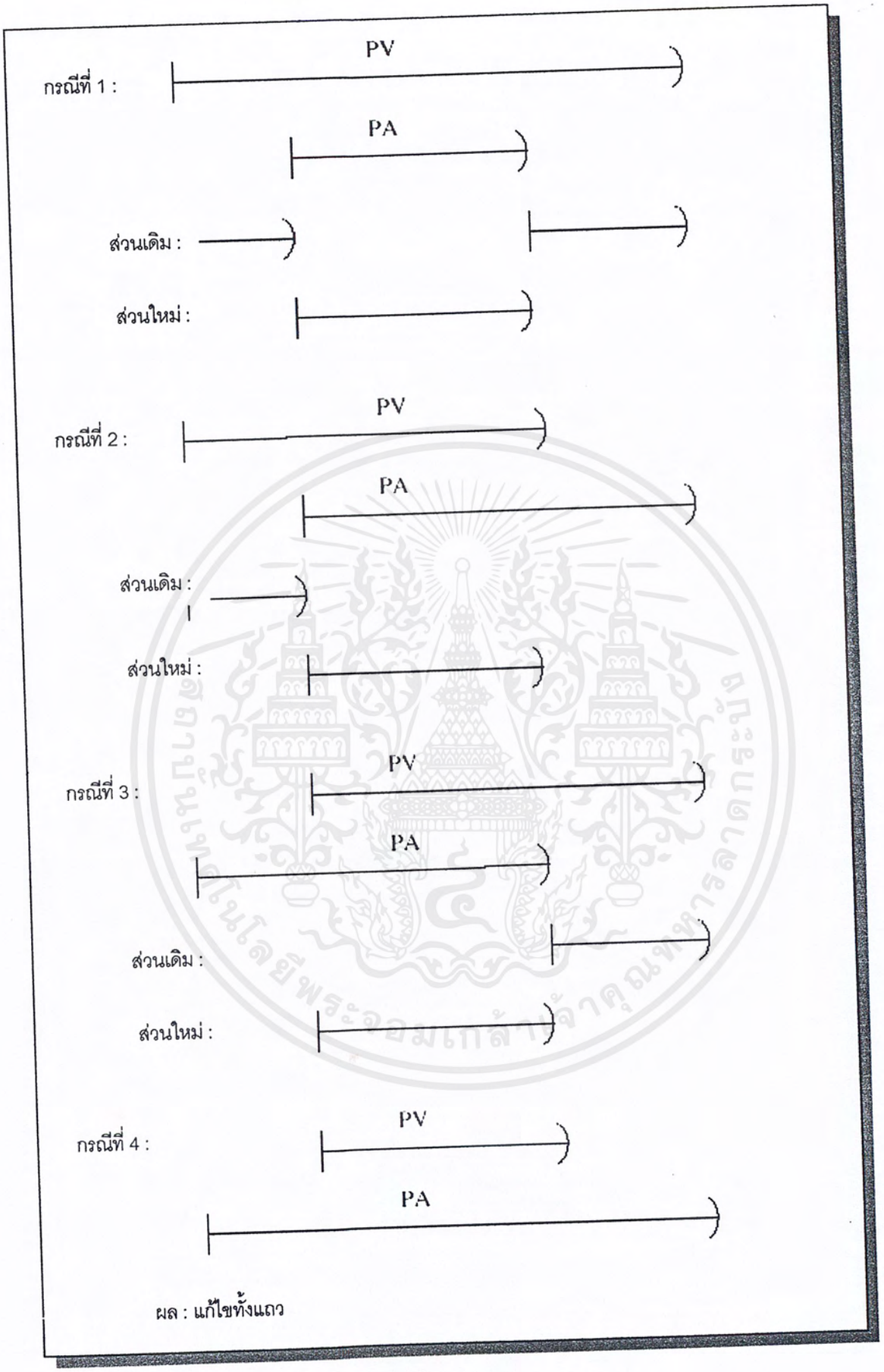
โดย PV คือ ช่วงเวลาของข้อมูลเดิม (Period Of Validity) และ PA คือ ช่วงเวลาที่ต้องการเปลี่ยนแปลง (Period Of Applicability) ดังนั้นจะสามารถทำการลบแบบซีเควนต์ได้โดยใช้ SQL ดังต่อไปนี้

```
INSERT INTO LOT
SELECT LOT_ID_NUM, GNDR_CODE, '1998-10-22', TO_DATE
FROM LOT
WHERE LOT_ID_NUM = 234
        AND FROM_DATE <= '1998-10-01'
        AND TO_DATE > '1998-10-22'
UPDATE LOT
SET TO_DATE = '1998-10-01'
WHERE LOT_ID_NUM = 234
        AND FROM_DATE < '1998-10-01'
        AND TO_DATE >= '1998-10-01'UPDATE LOT
SET FROM_DATE = '1998-10-22'
WHERE LOT_ID_NUM = 234
        AND FROM_DATE < '1998-10-22'
        AND TO_DATE >= '1998-10-22'
DELETE FROM LOT
WHERE LOT_ID_NUM = 234
        AND FROM_DATE >= '1998-10-01'
        AND TO_DATE <= '1998-10-22'
```

2.6.2.2 การแก้ไขแบบซีเควนซ์ (Sequenced Update)

พิจารณาการเปลี่ยนแปลงที่ว่า "เปลี่ยนวัวล็อท 799 เป็นวัวตอนตัวผู้เฉพาะในเดือนมีนาคม" การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 4 กรณี แสดงดังรูปที่ 2-10 ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-12 กรณีของการแก้ไขแบบซีเควนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะสามารถทำการแก้ไขแบบซีเควนต์ได้โดยใช้ SQL ดังต่อไปนี้

```
INSERT INTO LOT
SELECT LOT_ID_NUM, GNDR_CODE, FROM_DATE, '1998-03-01'
FROM LOT
WHERE LOT_ID_NUM = 799
        AND FROM_DATE < '1998-03-01'
        AND TO_DATE > '1998-03-01'INSERT INTO LOT
SELECT LOT_ID_NUM, GNDR_CODE, '1998-04-01', TO_DATE
FROM LOT
WHERE LOT_ID_NUM = 799
        AND FROM_DATE < '1998-04-01'
        AND TO_DATE > '1998-04-01'
UPDATE LOT
SET GNDR_CODE = 'S'
WHERE LOT_ID_NUM = 799
        AND FROM_DATE < '1998-04-01'
        AND TO_DATE > '1998-03-01'UPDATE LOT
SET FROM_DATE = '1998-03-01'
WHERE LOT_ID_NUM = 799
        AND FROM_DATE < '1998-03-01'
        AND TO_DATE > '1998-03-01'
UPDATE LOT
SET TO_DATE = '1998-04-01'
WHERE LOT_ID_NUM = 799
        AND FROM_DATE < '1998-04-01'
        AND TO_DATE > '1998-04-01'
```

2.6.3 การเปลี่ยนแปลงแบบนอนซีเควนซ์ (Nonsequenced Modification)

พิจารณาการเปลี่ยนแปลงที่ว่า "ลบข้อมูลของวัวล็อท 234 ที่มีช่วงเวลามากกว่า 3 เดือน" จะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

```
DELETE FROM LOT
WHERE LOT_ID_NUM = 234
AND (TO_DATE – FROM_DATE MONTH) > INTERVAL '3' MONTH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ปัญหาของฐานข้อมูลเชิงสัมพันธ์กับฐานข้อมูลเชิงเวลา

ฐานข้อมูลเชิงสัมพันธ์ โดยปกติจะเก็บเฉพาะข้อมูลที่เป็นจริงในปัจจุบันเท่านั้น และจะมีคีย์หลัก (Primary Key) ของตาราง ดังเช่นตารางที่ 2-11

ID	NAME	SURNAME	SALARY
1	JAME	SMITH	20000
2	ANNA	CUNICOVA	25000

ตารางที่ 2-11 การจัดเก็บฐานข้อมูลเชิงสัมพันธ์ทั่วไป

หากมีการแก้ไขค่าของแอตทริบิวต์ใดแอตทริบิวต์หนึ่ง จำเป็นต้องมีการอับเดท (Insert) ทับข้อมูลเดิม ถ้ายังคงต้องการใช้ ID ตัวเดิมในการอ้างถึงอยู่

ID	NAME	SURNAME	SALARY
~~1~~	~~JAME~~	~~SMITH~~	~~20000~~
2	ANNA	CUNICOVA	25000
1	JOHN	SMITH	20000

ตารางที่ 2-12 การอับเดทข้อมูลฐานข้อมูลเชิงสัมพันธ์ทั่วไป

จากตารางที่ 2-12 จะเป็นการอับเดทแอตทริบิวต์ Name ซึ่ง JAME เปลี่ยนชื่อเป็น JOHN ซึ่งจะทำให้ต้องทำการอับเดททับโรว์เดิม ทำให้ไม่สามารถทราบได้ว่า JOHN เคยมีชื่อเป็น JAME มาก่อน และไม่สามารถทราบได้ว่าแอตทริบิวต์ใดเปลี่ยน

หากต้องการให้ตารางฐานข้อมูลเชิงสัมพันธ์สามารถจัดเก็บข้อมูลเชิงเวลาได้ จำเป็นต้องเพิ่ม 2 คอลัมน์คือ FROM_DATE และ TO_DATE ซึ่งจะเป็นตัวบอกว่าข้อมูลมีการเปลี่ยนแปลงเมื่อใด และต้องคอมไบน์คีย์หลัก ให้เป็น ID, FROM_DATE และ TO_DATE ดังตารางที่ 2-13

ID	NAME	SURNAME	SALARY	FROM_DATE	TO_DATE
1	JAME	SMITH	20000	1/1/1980	1/1/1990
2	ANNA	CUNICOVA	25000	1/1/1980	31/12/9999
1	JOHN	SMITH	20000	1/1/1990	31/12/1999

ตารางที่ 2-13 การจัดเก็บฐานข้อมูลเชิงเวลาโดยใช้ฐานข้อมูลเชิงสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดเก็บข้อมูลดังตารางที่ 2-13 จะทำให้ฐานข้อมูลเชิงสัมพันธ์สามารถจัดเก็บข้อมูลเชิงเวลาได้ ซึ่งจากตาราง JAME จะมีการเปลี่ยนชื่อ JOHN เมื่อวันที่ 1/1/1990 ซึ่งจะทำให้ทราบว่าชื่อ JAME และ JOHN มีค่าช่วงใดโดยดูจากคอลัมน์ FROM_DATE และ TO_DATE และสามารถที่จะอ้างได้ว่า JAME และ JOHN เป็นคนคนเดียวกัน โดยดูจากหมายเลข ID

แต่การจัดเก็บข้อมูลเชิงเวลาโดยใช้ฐานข้อมูลเชิงสัมพันธ์ก็มีปัญหา ดังนี้

1. การเอาช่วงเวลาเกาะกับทั้งทัพเปิล ทำให้ไม่ทราบว่าแอตทริบิวต์ใดที่มีการเปลี่ยนค่าตามเวลา
2. การอ้างอิงคีย์หลัก (Primary Key) คีย์อ้างอิง (Foreign Key) ทำได้ยุ่งยาก จากตารางจะเห็นได้ว่าคีย์หลักจะเพิ่มเป็น 3 ตัว ซึ่งถ้านำไปเป็นคีย์อ้างอิงกับตารางอื่น ก็จะเกิดความยุ่งยากขึ้น

2.8 ภาษา SQL กับข้อมูลเชิงเวลา

ปัญหาของการใช้ SQL ธรรมดาในการจัดการกับข้อมูลเชิงเวลา

1. ถ้าต้องการหลีกเลี่ยงความซ้ำซ้อนในการจัดเก็บลงตารางที่เก็บข้อมูลที่มีการเปลี่ยนแปลงตามเวลา จำเป็นต้องการใช้ aggregate และ trigger ที่ซับซ้อน
2. การจอยตารางปกติ ซึ่งจะใช้แค่ 3 บรรทัด คือ

```
SELECT X1.Id, X1.Name, X2.City
FROM  X1 AS TABLE1,  X2 AS TABLE1
WHERE X1.Id = X2.Id
```

แต่ถ้านำมาประยุกต์ใช้ในตารางที่เก็บข้อมูลที่มีการเปลี่ยนแปลงตามเวลา จะต้องใช้ถึง 37 บรรทัด ซึ่งประกอบด้วย 4 SELECT ดังนี้

```
SELECT X1.Id, X1.Name, X2.City
FROM  X1 AS TABLE1,  X2 AS TABLE1
WHERE X1.Id < X2.Id
   AND X2.FROM_DATE <= X1.FROM_DATE
   AND X1.TO_DATE <= X2.TO_DATE
UNION
SELECT X1.Id, X1.Name, X2.City
FROM  X1 AS TABLE1,  X2 AS TABLE1
WHERE X1.Id < X2.Id
   AND X1.FROM_DATE > X2.FROM_DATE
   AND X2.TO_DATE < X1.TO_DATE
   AND X1.FROM_DATE < X2.TO_DATE
UNION
SELECT X1.Id, X1.Name, X2.City
FROM  X1 AS TABLE1,  X2 AS TABLE1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WHERE X1.Id < X2.Id
   AND X2.FROM_DATE > X1.FROM_DATE
   AND X1.TO_DATE < X2.TO_DATE
   AND X2.FROM_DATE < X1.TO_DATE
UNION
SELECT X1.Id, X1.Name, X2.City
FROM  X1 AS TABLE1,  X2 AS TABLE1
WHERE X1.Id < X2.Id
   AND X2.FROM_DATE > = X1.FROM_DATE
   AND X2.TO_DATE <= X1.TO_DATE
```

3. การอับเดทตารางปกติ ซึ่งจะใช้แค่ 3 บรรทัด คือ

```
UPDATE INTO TABLE1
SET NAME = 'SOMCHAI'
WHERE ID = 'SOMC'
```

แต่ถ้านำมาประยุกต์ใช้ในตารางที่เก็บข้อมูลที่มีการเปลี่ยนแปลงตามเวลา จะต้องใช้โค้ดมากขึ้น

4. การปรับปรุงเปลี่ยนแปลง ต้องการการทริกเกอร์หลายครั้ง ซึ่งจะทำให้ต้องเขียนโค้ดหลายบรรทัดมากกว่าปกติ

SQL-92 รองรับการจัดการข้อมูลที่เปลี่ยนแปลงตามเวลาผ่านชนิดข้อมูล DATE, TIME และ TIMESTAMP แต่ภาษาจะไม่เข้าใจข้อมูลเชิงเวลา ไม่สามารถเข้าถึงการค้นหา และการเปลี่ยนแปลงข้อมูลแบบปัจจุบัน และแบบซีเควนต์ ทั้งตารางแบบวาลิดไทม์ ทรานแซคชั่นไทม์ และไบเท็มโพรอล แต่จะรองรับกระบวนการแบบนอนซีเควนต์ทั้งหมด

SQL 3 เป็นภาษาเรียกค้นข้อมูลที่สนับสนุนข้อมูลเชิงเวลา และจะรองรับการจัดการข้อมูลเชิงเวลาในตารางได้ ซึ่งตารางจะมีวาลิดไทม์, ทรานแซคชั่นไทม์ หรือเป็นไบเทมโพรอลก็ได้ และจะทำให้ SQL3 สามารถทำกระบวนการ ปัจจุบัน ซีเควนต์ และนอนซีเควนต์ได้

ตัวอย่างเช่น พิจารณาตาราง NICUStatus และการควบคุมความถูกต้องของการเป็นเอกลักษณ์ โดยสมมติว่าตารางยังไม่สามารถจัดเก็บในเชิงเวลาได้ ซึ่งจะมีโค้ดในการสร้างตาราง ดังนี้

```
CREATE TABLE NICUStatus (
        Name  CHAR(15),
        Status  CHAR(8),
        UNIQUE (Name, Status) )
```

ซึ่งถ้าเราต้องการเก็บประวัติการเปลี่ยนแปลงด้วย คือทำเป็นตารางเชิงเวลา เราจำเป็นต้องเพิ่มคอลัมน์ วาลิดไทม์ โดยใช้ภาษา SQL3 ได้ดังนี้

```
ALTER TABLE ADD VALIDTIME PERIOD(DATE)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SQL3 เป็นภาษาในการเรียกค้นฐานข้อมูลที่เพิ่มความสามารถในการจัดการฐานข้อมูลเชิงเวลาเข้าไป โดยไม่มีผลกระทบใดๆ ต่อฐานข้อมูลที่ไม่เป็นเชิงเวลาเลย ซึ่งทำให้สามารถเปลี่ยนแปลงฐานข้อมูลปกติที่ไม่เป็นเชิงเวลา มาเป็นฐานข้อมูลเชิงเวลาได้ โดยไม่ต้องแก้โค้ดการสร้างตารางใหม่ เพียงแต่เพิ่มเข้าไป และการค้นหาและการเปลี่ยนแปลงข้อมูลด้วยภาษา SQL92 ก็จะเหมือนกับการค้นหาและเปลี่ยนแปลงแบบปัจจุบันตามทฤษฎีเชิงเวลา

นอกจากนี้การใช้ภาษา SQL3 จะเพิ่มชนิดข้อมูลใหม่ๆ และความสามารถในการจัดการข้อมูลเชิงเวลา จากภาษา SQL92 ทำให้สามารถลดโค้ดในการโปรแกรมด้วยภาษา SQL3 แทนการใช้ภาษา SQL92 ลงได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ฐานข้อมูลเชิงวัตถุสัมพันธ์

ฐานข้อมูลเชิงวัตถุสัมพันธ์ เป็นโมเดลที่นำโมเดลของฐานข้อมูลเชิงวัตถุและโมเดลของฐานข้อมูลเชิงสัมพันธ์ มาประยุกต์รวมกัน ทำให้ฐานข้อมูลเชิงวัตถุสัมพันธ์มีความสามารถของโมเดลทั้งสอง ซึ่งจะได้อธิบายลักษณะและความสามารถของโมเดลต่างๆ ดังนี้

3.1 ฐานข้อมูลเชิงวัตถุ (Object Oriented Database : OODB)

ฐานข้อมูลเชิงวัตถุ คือ ฐานข้อมูลชนิดหนึ่งที่ถูกแสดงด้วยชนิดของข้อมูลที่กำหนดขึ้น และโอเปอร์เรชัน (Operation) ที่ถูกออกแบบมาเพื่อใช้กับชนิดของข้อมูลเหล่านั้น

ฐานข้อมูลเชิงวัตถุ เป็นฐานข้อมูลที่สนับสนุนแนวคิดเชิงวัตถุ โดยใช้ออบเจ็กต์เป็นพื้นฐาน ฐานข้อมูลเชิงวัตถุ เกิดขึ้นมาจากการรวมกันของภาษาการโปรแกรมเชิงวัตถุ (Object-Oriented Programming) และเทคโนโลยีทางด้านฐานข้อมูล ซึ่งก่อให้เกิดประโยชน์อย่างมากมาย ดังนี้

1. การที่แนวคิดเชิงวัตถุมีความยืดหยุ่น (Flexibility) จึงอำนวยความสะดวกในการออกแบบงานทางด้านฐานข้อมูลที่มีโครงสร้างสลับซับซ้อน เช่น CAD
2. งานทางด้านฐานข้อมูลที่มีโครงสร้างสลับซับซ้อน อาจเขียนขึ้นภายในการโปรแกรมภาษาซิงเกิลออบเจ็กต์โอเรียนเต็ดดาต้าเบส (Single Object Oriented Database Programming Language) ได้
3. การใช้ภาษาที่เป็นการโปรแกรมแบบเชิงวัตถุจะได้รับประโยชน์จากคุณลักษณะของฐานข้อมูล เช่น การคงอยู่ของข้อมูล (Persistance) , การจัดการทรานแซคชั่น (Transaction Management) และเซตโอเรียนเต็ดโพรเซสซิ่ง (Set-Oriented Processing)

ความแตกต่างของฐานข้อมูลเชิงวัตถุและการโปรแกรมเชิงวัตถุ (Programming Language Object)

1. งานทางด้านฐานข้อมูลหลายๆ งานต้องการความสามารถในการสร้าง (Create) และ การเข้าถึง (Access) กับออบเจ็กต์ในแบบของมัลติเพิลเวอร์ชัน (Multiple Version)
2. ข้อมูลของฐานข้อมูลแบบออบเจ็กต์โอเรียนเต็ดจะต้องคงอยู่ (Persist) ตลอดภายใต้การทำงานของโปรแกรมที่สร้างมันขึ้นมา
3. ฐานข้อมูลต้องการประสิทธิภาพของการคิวรีในเพรดิเคทเบส (Predicate -Based) บนออบเจ็กต์
4. ฐานข้อมูลที่มีความเร็วสูงๆ เช่น ฐานข้อมูลที่ใช้ในการควบคุมการจราจรทางอากาศและฐานข้อมูลที่ใช้จัดการกับการกระจาย (Distribution) ให้มีประสิทธิภาพต้องการความสามารถที่เกี่ยวข้องกับเงื่อนไข (Condition) และการกระทำ (Action) ซึ่งการดำเนินการจะถูกทำให้สมบูรณ์ เมื่อมีเงื่อนไขที่เหมาะสมกับออบเจ็กต์นั้น

ซึ่งภาษาโปรแกรมมิ่ง (Programming Language) ทั่วไปส่วนใหญ่จะไม่สนับสนุนการคงอยู่ของออบเจ็กต์หรือมัลติเพิลเวอร์ชัน (Multiple Version) และไม่อำนวยความสะดวกเกี่ยวกับเรื่องคอนสเตรนท์ (Constraint)

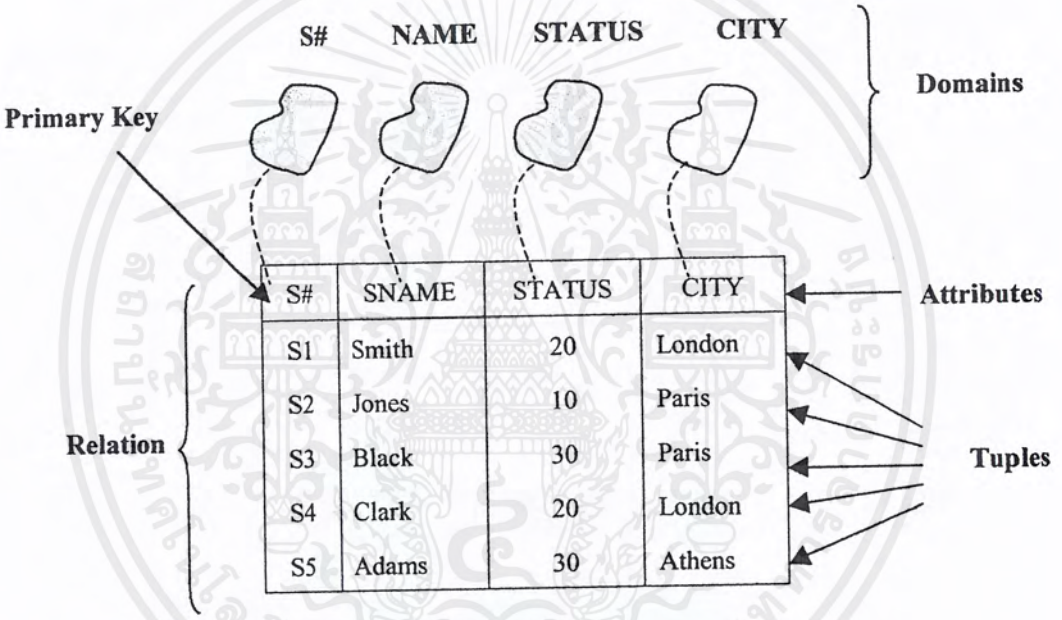
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database System)

ระบบฐานข้อมูลเชิงสัมพันธ์มีพื้นฐานมาจากเซตทางคณิตศาสตร์ และมีภาษาเอสคิวแอล (Structure Query Language) เป็นภาษาที่ใช้ในการกำหนดโครงสร้างและจัดการกับข้อมูลของฐานข้อมูลแบบเชิงสัมพันธ์นี้

การที่จะกล่าวว่าระบบฐานข้อมูลใดเป็นโมเดลเชิงสัมพันธ์ (Relation Model) นั้น ต้องพิจารณาว่าฐานข้อมูลดังกล่าวมีองค์ประกอบครบทั้ง 3 ส่วน ดังนี้

1. โครงสร้างข้อมูลเป็นไปตามนิยามคุณสมบัติของรีเลชัน (Relations)
2. ความถูกต้องของข้อมูล (Data Integrity) เป็นไปตามความถูกต้องของข้อมูลทั้งสอง
3. การจัดการข้อมูล (Data Manipulation) มีภาษาที่เป็นรีเลชันนัล คอมพลีท (Relational Complete) ในการจัดการฐานข้อมูล



รูปที่ 3-1 แสดงตัวอย่างตารางความสัมพันธ์

3.2.1 โครงสร้างข้อมูลแบบสัมพันธ์ (Relational Data Structure)

1. โดเมน (Domain) หมายถึง หน่วยข้อมูลที่เก็บอยู่ในตารางความสัมพันธ์
2. ทัพเพิล (Tuple) หมายถึง แถวของข้อมูลหนึ่งแถวของตารางความสัมพันธ์
3. เอนทิตี้ (Entity) หมายถึง ชื่อของสิ่งใดสิ่งหนึ่งอาจเกี่ยวข้องกับคน สถานที่ สิ่งของ การกระทำซึ่งต้องการจัดเก็บข้อมูลไว้ เช่น เอ็นทิตี้ของสินค้า
4. แอตทริบิวต์ (Attribute) หมายถึง รายละเอียดของข้อมูลในเอนทิตี้หนึ่งๆ เช่น เอนทิตี้ของสินค้าประกอบด้วยแอตทริบิวต์รหัสสินค้า ชื่อ สินค้า และราคาสินค้า หรือถ้าในตารางความสัมพันธ์จะหมายถึง ชื่อคอลัมน์ของตารางความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คีย์หลัก (Primary Key) หมายถึง แอตทริบิวต์หรือฟิล์ดที่มีลักษณะเฉพาะตัว (Unique) หรือหมายถึงแคนดิเดทคีย์ตัวหนึ่งของรีเลชันนั้นที่ได้จากการกำหนดของผู้ออกแบบฐานข้อมูล
6. คีย์อ้างอิง (Foreign Key) หมายถึง แอตทริบิวต์หรือกลุ่มของแอตทริบิวต์บนรีเลชัน ซึ่งเป็นไพรมารีคีย์ของรีเลชันอื่น ซึ่งความสัมพันธ์ระหว่างฟอเรจ์นคีย์ไปยังไพรมารีคีย์อื่นนี้เรียกว่า การอ้างถึง หรือ เรฟเฟอเรจ์น (Reference)
7. ความสัมพันธ์ (Relationship) หมายถึง คำที่แสดงความสัมพันธ์ระหว่าง 2 เอนทิตี้
8. ความสัมพันธ์ระหว่างเอนทิตี้ ซึ่งจะมีการระบุชื่อความสัมพันธ์ระหว่างเอนทิตี้ โดยกำหนดทิศทางความสัมพันธ์จากเอ็นทิตี้หนึ่งไปยังอีกเอ็นทิตี้หนึ่งว่ามีความสัมพันธ์กันอย่างไร เช่น เอนทิตี้สินค้าจะอยู่ในเอนทิตี้หมวดหมู่ เอนทิตี้หมวดหมู่จะประกอบด้วยเอนทิตี้สินค้า ดังรูป

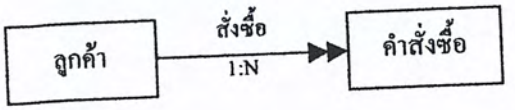
รูปที่ 3-2 แสดงความสัมพันธ์ระหว่างเอนทิตี้

3.2.2 ชนิดของความสัมพันธ์ แบ่งออกเป็น

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-To-One Relationship) คือ การแสดงความสัมพันธ์ของข้อมูลของเอนทิตี้หนึ่งว่ามีความสัมพันธ์กับข้อมูลของอีกเอนทิตี้หนึ่งอย่างมากเพียง 1 ข้อมูล

รูปที่ 3-3 แสดงความสัมพันธ์แบบ 1 : 1

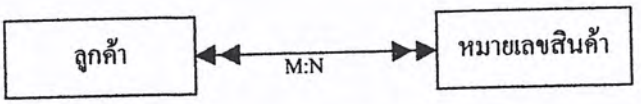
2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-To-Many Relationship) คือ การแสดงความสัมพันธ์ของข้อมูลของเอนทิตี้หนึ่งว่ามีความสัมพันธ์กับข้อมูลของอีกเอนทิตี้หนึ่งอยู่หลายข้อมูล



รูปที่ 3-4 แสดงความสัมพันธ์แบบ 1 : กลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-To-Many Relationship) คือ การแสดงความสัมพันธ์ของข้อมูลของทั้งสองเอนทิตี้ในลักษณะแบบกลุ่มต่อกลุ่ม



รูปที่ 3-5 แสดงความสัมพันธ์แบบ กลุ่ม : กลุ่ม

3.3 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object Relational Database : ORDB)

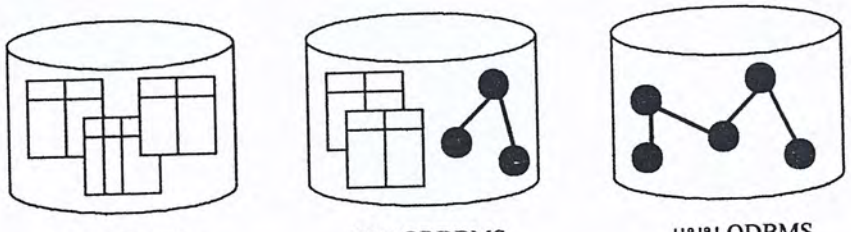
เนื่องจากการจัดเก็บข้อมูลในปัจจุบันโดยทั่วไปมักใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ แต่เมื่อมีปัญหากับการใช้งานชนิดข้อมูลที่มีแต่ข้อมูลชนิดพื้นฐานไม่สามารถรองรับการใช้งานในปัจจุบัน ซึ่งนอกจากฐานข้อมูลเชิงสัมพันธ์แล้วยังมีระบบฐานข้อมูลเชิงวัตถุ (Object-Oriented Database) เป็นระบบฐานข้อมูลที่มีลักษณะเป็นแบบแนวคิดเชิงวัตถุโดยใช้ออบเจ็กต์เป็นพื้นฐานในการสนับสนุนการใช้งานข้อมูลที่มีความซับซ้อน แต่ก็ยังมีข้อเสียบางประการ และยังไม่สามารถสนับสนุนงานในลักษณะของฐานข้อมูลได้เต็มที่ รวมทั้งภาษาที่ใช้ในการจัดการฐานข้อมูลยังไม่เป็นมาตรฐานจึงเกิดแนวคิดในการนำข้อดีของฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลเชิงวัตถุมารวมกัน

ฐานข้อมูลเชิงวัตถุสัมพันธ์ คือ ฐานข้อมูลที่นำแนวคิดเชิงวัตถุมารวมเข้ากับฐานข้อมูลเชิงสัมพันธ์เพื่อทำให้ฐานข้อมูลเชิงสัมพันธ์มีความสามารถทางด้านแนวคิดเชิงวัตถุ เช่นการปกป้องข้อมูล การสืบทอดคุณสมบัติ และ โพลิมอร์ฟิซึ่มเป็นต้น

ฐานข้อมูลเชิงวัตถุสัมพันธ์ เป็นฐานข้อมูลเชิงสัมพันธ์ที่สามารถมีแอททริบิวเป็นแอททริบิวที่สามารถแบ่งแยกย่อยได้อีก โดยภาพรวมแล้วเรายังมองเห็นฐานข้อมูลเป็นฐานข้อมูลเชิงสัมพันธ์ ในคอลัมน์สามารถเป็นได้ทั้ง ออบเจ็กต์หรือเป็นแอตทริบิวต์ที่มีหลายค่าเช่นลิสท์หรือเซต โดยทั่วไปโมเดลที่เป็นฐานข้อมูลเชิงวัตถุสัมพันธ์สามารถใช้งานเสมือนเป็นฐานข้อมูลเชิงสัมพันธ์ ได้

3.3.1 เทคโนโลยีที่ใช้ในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

1. การเพิ่มชั้นออบเจ็กต์โอเรียนเต็ดครอบบนชั้นระบบฐานข้อมูลเชิงสัมพันธ์ที่มีอยู่
2. การออกแบบให้ระบบฐานข้อมูลเชิงสัมพันธ์รองรับงานประเภทมัลติมีเดีย และการกำหนดชนิดของข้อมูลเป็นออบเจ็กต์



รูปที่ 3-6 แสดงลักษณะของฐานข้อมูลแบบ RDBMS , ORDBMS และ ODBMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 คุณสมบัติของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

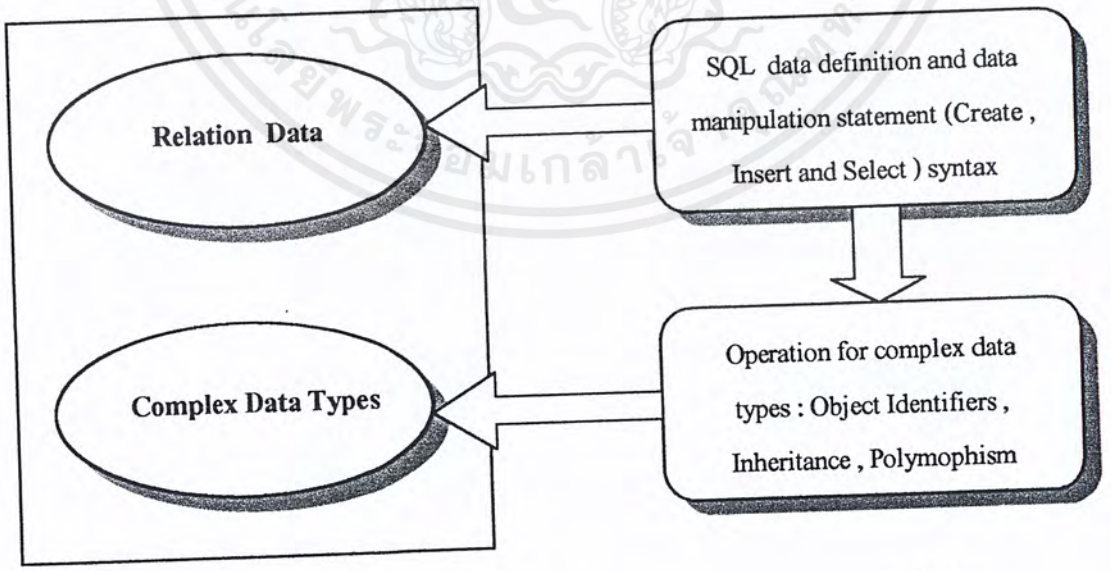
3.3.2.1 สนับสนุนตารางที่มีความซับซ้อนมากขึ้น ในโมเดลฐานข้อมูลเชิงสัมพันธ์ (Relational Data Model)นิยามไว้ว่าโดเมนของทุกๆ แอตทริบิวต์ต้องอะตอมมิก หมายถึงในแอตทริบิวต์หนึ่งๆ สามารถเก็บข้อมูลได้เพียงค่าเดียวและข้อมูลนั้นไม่สามารถแยกย่อยต่อไปอีกได้ แต่ในระบบฐานข้อมูลเชิงสัมพันธ์อนุญาตให้ค่าของข้อมูลในแอตทริบิวต์มีหลายค่าได้ ดังตารางตัวอย่างเช่น

Title	Author-list	Date (day,month,year)	Keyword-list
Salesplan	{Smith,Jones}	(1,April,89)	{profit,strategy}
Status report	{Jones,Frick}	(17,June,94)	{profit,personnel}

ตารางที่ 3-1 แสดงความสัมพันธ์ของแอตทริบิวต์ที่ไม่อะตอมมิก

จากตารางจะเห็นได้ว่า ภายในคอลัมน์เก็บข้อมูลหลายค่า เนื่องจากในเอกสารหนึ่งอาจมีผู้แต่งหลายคน กำหนดคำเป็นคีย์ของเอกสารได้หลายคำ หรือการเก็บวันที่ที่แต่ง ซึ่งถึงแม้จะไม่ได้เก็บหลายๆ วันในหนึ่งคอลัมน์ของแถว แต่เมื่อพิจารณาแล้วจะพบว่า วันที่สามารถแยกเป็นฟิลด์ย่อยได้ คือ วัน เดือน ปี นั่นก็หมายความว่า วันที่ไม่อะตอมมิก

3.3.2.2 คำสั่งภาษาฐานข้อมูล (SQL) ที่มีความสมบูรณ์และมีประสิทธิภาพ มีการปรับปรุงให้คำสั่งภาษาฐานข้อมูลสามารถจัดการกับชนิดข้อมูลที่ซับซ้อนและนิยามคุณสมบัติของออบเจ็กต์ได้ เช่น การสืบทอดคุณสมบัติ(Inheritance) , โพลิมอร์ฟิซึม (Polymorphism) , การเอ็นแคปซูเลชัน (Encapsulation) ให้กับข้อมูลได้ ซึ่งภาษาฐานข้อมูลที่ถูกปรับปรุงขึ้นมาใหม่นี้ มีชื่อเรียกว่า SQL3



รูปที่ 3-7 แสดงการทำงานของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.3 มีคุณสมบัติทางด้านแนวคิดเชิงวัตถุ 4 ประการ ดังนี้

- แอปสแตรกชัน (Abstraction)

แอปสแตรกชัน คือ การแทนความคิดที่ยุ่งยากซับซ้อนให้สั้น กระทัดรัดและรัดกุมขึ้นโดยใช้ออบเจ็กต์ ซึ่งในออบเจ็กต์จะรวมทั้งข้อมูล (Data) และการทำงาน (Process) ที่เกี่ยวข้องเขาไว้ด้วยกันเพื่อเป็นตัวแทนของออบเจ็กต์นั้น หรืออาจกล่าวได้ว่า ออบเจ็กต์ เป็นเอ็นแคปซูเลชันของแอปสแตรกชัน (Encapsulation Of Abstractions) นั่นเอง

การกำหนดแอปสแตรกชันขึ้นมาใหม่ในวิธีการแก้ปัญหาแบบออบเจ็กต์โอเรียนเต็ดจะต้องการออบเจ็กต์ใหม่ และสำหรับแต่ละออบเจ็กต์ใหม่ต้องกำหนดคุณสมบัติและเมสเซจที่จะติดต่อจัดการเกี่ยวกับอบบเจ็กต์นั้นได้

วิธีการแก้ปัญหาหนึ่งอาจประกอบด้วยหลายออบเจ็กต์ บางออบเจ็กต์ก็มีคุณสมบัติคล้ายกัน ซึ่งเรียกกลุ่มออบเจ็กต์ที่มีคุณสมบัติคล้ายกันนี้ว่า คลาสคุณสมบัติพื้นฐานของออบเจ็กต์ แต่ละออบเจ็กต์เป็นอินสแตนท์ของคลาส และการกระทำของออบเจ็กต์ถูกกำหนดโดยกลุ่มของคุณสมบัติ

- เอนแคปซูเลชัน (Encapsulation)

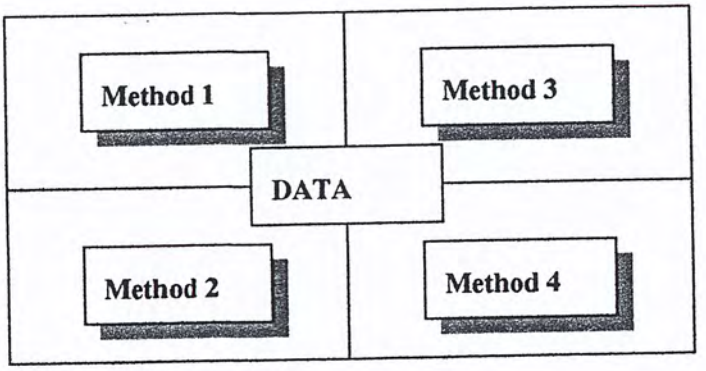
เอนแคปซูเลชัน คือ การป้องกันโครงสร้างข้อมูล (Data Structure) กับวิธีการทำงานที่เกี่ยวข้องหรือความสามารถในการซ่อนข้อมูล (Information Hiding) จากระบบภายนอกได้

เอนแคปซูเลชัน มีลักษณะดังนี้

- แชร์ดาต้า (Share Data) องค์ประกอบส่วนนี้จะมีค่าเหมือนกันสำหรับออบเจ็กต์ทั้งหมดที่เป็นอินสแตนท์ในคลาส อินสแตนท์ทั้งหมดในคลาสสามารถเข้าถึงและใช้ข้อมูลร่วมกันนี้ได้
- ไพรเวทดาต้า (Private Data) แต่ละออบเจ็กต์จะกำหนดส่วนประกอบของแต่ละอินสแตนท์ในคลาส ซึ่งออบเจ็กต์ทั้งหมดที่เป็นอินสแตนท์ในคลาสจะมีองค์ประกอบส่วนนี้ แต่จะมีค่าต่างกัน โดยออบเจ็กต์หนึ่งๆจะสามารถเข้าถึงไพรเวทดาต้า (Private Data) ของตนเองได้เท่านั้น
- เมสเซส (Message) กลุ่มของเมสเซจ คือ สิ่งที่ออบเจ็กต์ตอบสนองต่อออบเจ็กต์ทั้งหมด ออบเจ็กต์ที่อยู่ในคลาสเดียวกันจะตอบสนองต่อกลุ่มของเมสเซจเดียวกัน
- โกลบอล-แชร์ดาต้า (Global-Shared Data) เป็นข้อมูลที่สามารถถูกใช้ร่วมกันโดยอินสแตนท์ต่างคลาสได้ โกลบอล-แชร์ดาต้าจะมีค่าเหมือนกันสำหรับทุกออบเจ็กต์ที่ใช้งานร่วมกันอยู่

ข้อดีของการทำเอนแคปซูเลชัน คือ ทำให้เราสามารถแบ่งระดับของการเข้าถึงข้อมูลได้ด้วยเมทธอดของตัวออบเจ็กต์เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 แสดงโครงสร้างเอนแคปซูเลชัน

- *โพลิมอร์ฟีซึม (Polymorphism)*

โพลิมอร์ฟีซึม คือ การที่เราส่งเมสเซจที่เหมือนกันไปในออบเจ็กต์ที่ต่างกัน แต่ละออบเจ็กจะตอบสนองออกมาไม่เหมือนกันตามแต่ชนิดและหน้าที่ของออบเจ็กต์นั้น ซึ่งก็คือการสร้างฟังก์ชันที่มีหน้าที่แตกต่างกัน แต่เรียกใช้ได้ด้วยชื่อเดียวกัน เรียกว่า 'One interface , Multiple methods' เพื่อให้ใช้งานได้ง่าย หรือเรียกอีกอย่างว่า การทำโอเวอร์โหลดดิ้ง (Overloading) โดยจะแยกความแตกต่างของการใช้ชื่อที่เหมือนกันได้ด้วยจำนวนและชนิดของพารามิเตอร์

ตัวอย่างของโพลิมอร์ฟีซึม

- AB + CD ได้ผลลัพธ์ ABCD (Concat)
- 5 + 3 ได้ผลลัพธ์ 8 (Add)

ทั้งสองแบบนี้เป็นการส่งเมสเซจ '+' เข้าไปยังออบเจ็กต์ภายในคลาส String และ Integer ตามลำดับ

โพลิมอร์ฟีซึม แบ่งออกได้เป็น 2 ลักษณะ คือ

1. ช่วงการคอมไพล์โปรแกรม (Compile Time) การตรวจสอบชนิดออบเจ็กต์ของฟังก์ชันที่ถูกเรียกมาทำงาน รวมทั้งการเตรียมข้อมูลบางอย่างที่จำเป็น จะทำขณะคอมไพล์โปรแกรม การตัดสินว่าจะใช้ฟังก์ชันใดขึ้นอยู่กับคุณสมบัติของพารามิเตอร์ของฟังก์ชันนั้น
2. ช่วงการรันโปรแกรม (Run Time) การตรวจสอบจะทำขณะที่โปรแกรมกำลังทำงาน จะทำการตัดสินใจเลือกฟังก์ชันใดฟังก์ชันหนึ่งที่มีชื่อซ้ำกันในแต่ละคลาส มาใช้งาน

- *อินเฮอริแทนซ์ (Inheritance)*

อินเฮอริแทนซ์ คือ การสร้างคลาสขึ้นมาใหม่โดยมีการสืบทอดคุณสมบัติจากคลาสที่มีอยู่เดิม แต่จะมีข้อมูลหรือเมทธอดที่พิเศษเป็นของตัวเองเพิ่มขึ้นมาจากคลาสเดิม ซึ่งการสืบทอดคลาสนี้จะช่วยให้การพัฒนาโปรแกรมใหม่ทำได้เร็วยิ่งขึ้น

ถ้าซับคลาส B เป็นซับคลาสของคลาส A ออบเจ็กต์ที่เป็นอินสแตนท์ของซับคลาส B จะมีคุณสมบัติพิเศษกว่าอินสแตนท์ของคลาส A แต่อย่างน้อยที่สุด จะต้องมีคุณสมบัติเหมือนอินสแตนท์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส A เรียกว่า อินสแตนท์ของซับคลาส B จะถ่ายทอดคุณสมบัติมาจากคลาส A การถ่ายทอดคุณสมบัตินี้จะรวมทั้ง Private Data , Shared Data และ Message และคลาส A จะถูกเรียกว่าเป็นซุปเปอร์คลาสของคลาส B

อินเฮอริแทนซ์เป็นคุณสมบัติของออบเจ็กต์ที่เป็นอินสแตนท์ของคลาส ซึ่งจะถ่ายทอดคุณสมบัติจากซุปเปอร์คลาสของมัน ซับคลาสมีหลายระดับ ซึ่งซับคลาสที่อยู่ต่ำสุดก็จะมีคุณสมบัติของซุปเปอร์คลาสของมันทุกๆตัว

การสืบทอดคุณสมบัติต่างๆ จะเป็นดังนี้

- คลาสและอินสแตนท์ การขยายความของคลาสจะนำเสนอโดยกลุ่มของอินสแตนท์ที่มีคุณสมบัติเดียวกัน คือความสัมพันธ์ของซุปเปอร์คลาสและซับคลาสจะถูกนำเสนอออกมาโดยผ่านทางอินสแตนท์
- แอตทริบิวต์ ซับคลาสจะสืบทอดแอตทริบิวต์ต่างๆ จากซุปเปอร์คลาสโดยตรง ทั้งชื่อ ข้อจำกัด และความสัมพันธ์ต่างๆ ภายในแต่ละแอตทริบิวต์
- ความสัมพันธ์ การสืบทอดคุณสมบัติความสัมพันธ์จะรวมในส่วนของชื่อ ชนิดของความสัมพันธ์ ข้อจำกัดต่างๆ และความสัมพันธ์ที่เกี่ยวข้องกับคลาสอื่นๆ
- โอเปอเรชัน จะถูกสืบทอดโดยซับคลาส ซึ่งซับคลาสสามารถเพิ่มโอเปอเรชันเข้าไปได้โดยอัตโนมัติ และ โอเปอเรชันของซับคลาสสามารถเป็นส่วนขยายของซุปเปอร์คลาสได้

ตัวอย่างของอินเฮอริแทนซ์ เช่น คลาสของนักเรียน แต่ละประเภทของนักเรียนจะถูกแยกออกเป็นชั้นๆ ที่ยอดบนสุดเป็นคลาสพื้นฐาน คือนักเรียน และชั้นถัดมาเป็นนักเรียนประถม นักเรียนมัธยมต้นและนักเรียนมัธยมปลาย ซึ่งได้รับการถ่ายทอดคุณสมบัติของซุปเปอร์คลาสข้างบนลงมา และมีคุณสมบัติต่างๆ เพิ่มเติมสำหรับนักเรียนแต่ละประเภท ดังรูปที่ 3-9

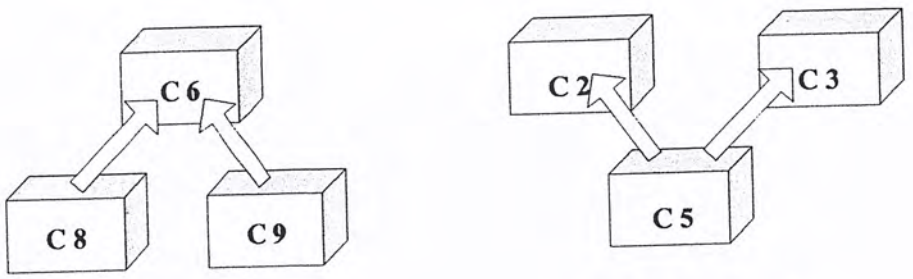
รูปที่ 3-9 แสดงการถ่ายทอดคุณสมบัติ

อินเฮอริแทนซ์แบ่งได้เป็น 2 แบบ คือ

1. อินเฮอริแทนซ์แบบเดี่ยว หรือ Single Inheritance
2. อินเฮอริแทนซ์แบบรวม หรือ Multiple Inheritance

ซึ่งสามารถแสดงลักษณะของอินเฮอริแทนซ์ทั้งสองแบบได้ดังรูปที่ 3-10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Single Inheritance Multiple Inheritance

รูปที่ 3-10 แสดงคลาสแบบลำดับชั้นด้วยอินเฮอริแทนซ์แบบเดี่ยวและรวม

3.3.3 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์จะสนับสนุนฟังก์ชันพื้นฐาน คือ

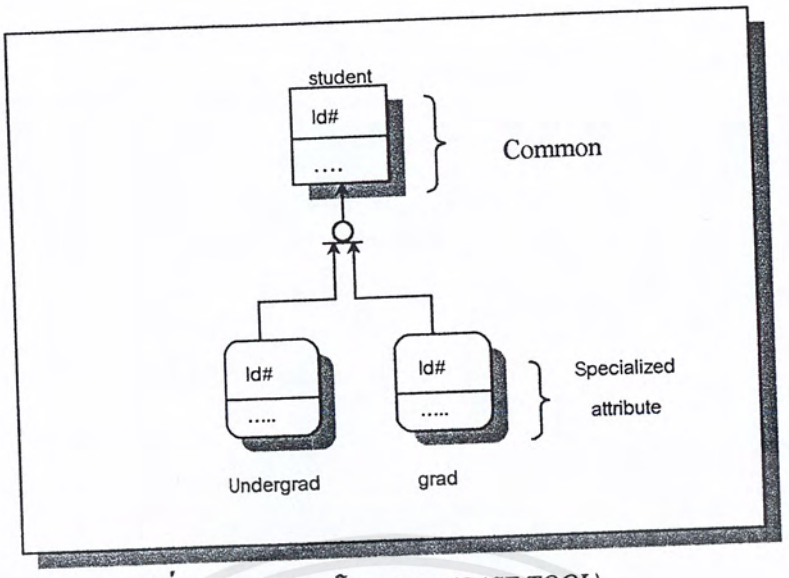
1. ความคงอยู่ของออบเจ็กต์ (Persistance of Object)
2. ความถูกต้องของข้อมูลและทรานแซกชัน (Data and Transaction Integrity)
3. ความปลอดภัย
4. การกู้ข้อมูลคืนกลับ (Recovery)
5. การใช้ข้อมูลร่วมกันจากผู้ใช้หลายๆคน (Sharing and Concurrent multi-user Access)
6. การเข้าถึงข้อมูลและการคิวรีข้อมูล (Access or Query Language)
7. การจัดการกับข้อมูลจำนวนมาก

3.4 โมเดลที่ใช้ในการออกแบบ

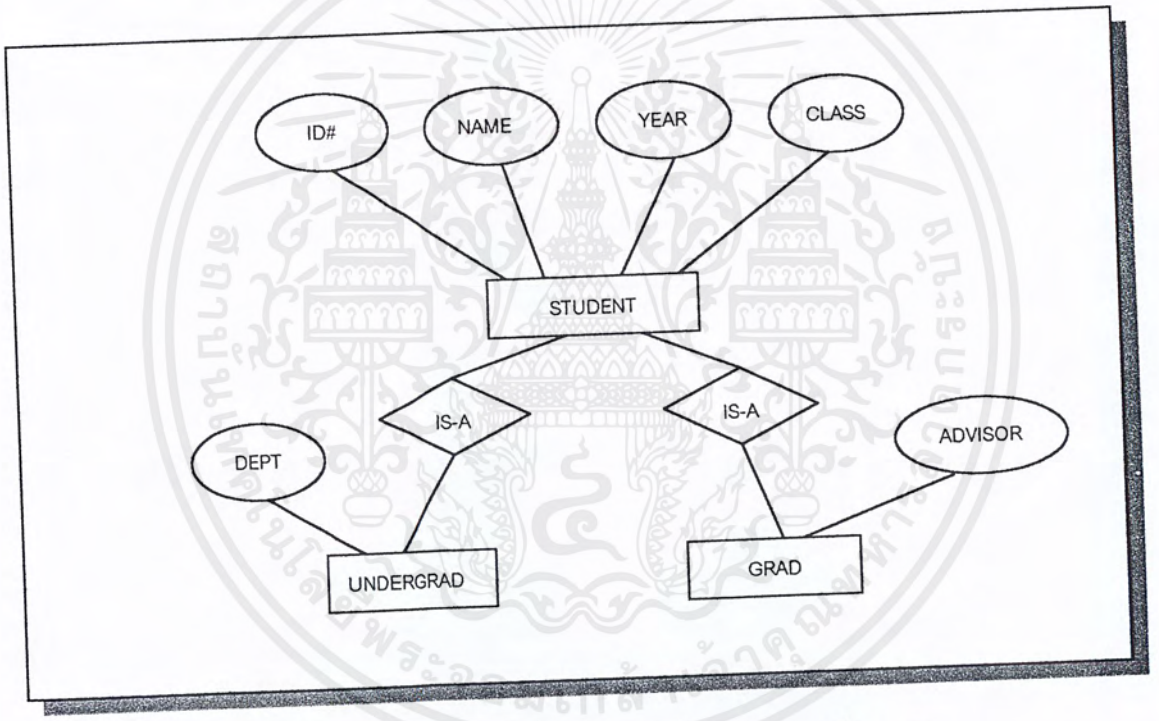
สำหรับในการออกแบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ยังคงมองฐานข้อมูลเหมือนฐานข้อมูลเชิงสัมพันธ์ได้ โดยสำหรับโมเดลที่ใช้ในการออกแบบก็ยังสามารถที่จะใช้ ER หรือ ไนแอม (NIAM) ได้ เพราะทั้งสองโมเดลนี้มีหลักการของการทำลำดับขั้นของชนิดข้อมูล (Subtype hierarchy) อยู่แล้วแต่ในการแปลงจากโมเดลเป็นตารางนั้นยังไม่มีหลักการที่แน่นอนอันเนื่องมาจากระบบฐานข้อมูลเดิมเป็นแบบสัมพันธ์ธรรมดาที่ยังไม่สนับสนุนหลักการดังกล่าว ทำให้ในการแปลงมาเป็นตารางของแต่ก่อนนั้นขึ้นอยู่กับงานซึ่งอาจจะมีการรวมเอาทั้งหมดอยู่ในตารางเดียว (ซุปเปอร์คลาสและซับคลาส) หรืออาจจะแบ่งให้มีตารางที่ประกอบด้วยแอตทริบิวต์ร่วมอยู่ตารางหนึ่งแล้วแยกตารางของแต่ละซับคลาสเป็นอีกตารางโดยจะประกอบด้วยแอตทริบิวต์ของแต่ละซับคลาสที่มีต่างหากออกมาก็ได้

โมเดล EER (Enhanced-Entity Relationship) เป็นโมเดลที่ได้เพิ่มแนวคิดเกี่ยวกับซับคลาส (Sub-class) และซุปเปอร์คลาส (Super-class) กับโมเดล ER ที่มีการกำหนดวิธีการในการแปลงจากโมเดลที่ชัดเจนกล่าวคือปกติในการแปลง ER จะมีขั้นตอนอยู่ 7 ขั้นตอน แต่สำหรับโมเดล EER จะเพิ่มขั้นตอนที่ 8 ซึ่งเป็นขั้นตอนที่จะนำแอตทริบิวต์ทั้งหมดจากซุปเปอร์คลาสมารวมกับแอตทริบิวต์ของซับคลาสดังตัวอย่างในรูป 3-11(ก), 3-11(ข) และ 3-11(ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-11 (ก) แสดงโมเดล ER (CASE TOOL)



รูปที่ 3-11 (ข) แสดงการแปลงจาก ER (CASE TOOL) มาเป็น EER

STUDENT

ID#	NAME	YEAR	CLASS

UNDERGRAD

ID#	NAME	YEAR	CLASS	DEPT

GRAD

ID#	NAME	YEAR	CLASS	ADVISOR

รูปที่ 3-11 (ค) แสดงตัวอย่างการแปลง EER โมเดลเป็นรีเลชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์
(INFORMIX- UNIVERSAL SERVER)

4.1 อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์

อินฟอร์มิกซ์เป็นระบบการจัดการฐานข้อมูล (DBMS) เชิงพาณิชย์ที่รองรับโมเดลของฐานข้อมูลได้หลายรูปแบบ ซึ่งมีผลิตภัณฑ์หลายตัวที่รองรับระบบงานในสภาวะแวดล้อมที่แตกต่างกันออกไป ผลิตภัณฑ์ตัวหนึ่งของอินฟอร์มิกซ์ที่นำมาใช้ศึกษา ก็คือ อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ (Informix Universal Server ; IUS) เวอร์ชั่น 9.14 ซึ่งเป็นเซิร์ฟเวอร์ฐานข้อมูลที่รองรับโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ ทำงานบนระบบปฏิบัติการวินโดว์เอ็นที (WindowsNT) และมีความสามารถในการจัดการกับฐานข้อมูลที่มีความซับซ้อนได้ เช่น ภาพ 2 มิติ, ภาพ 3 มิติ, ข้อมูลเสียง, ข้อมูลวีดิโอ, ข้อมูลจากสัญญาณดาวเทียม เป็นต้น ซึ่งใช้ภาษาในการจัดการข้อมูลโดยอ้างอิงตามมาตรฐานข้อง SQL3

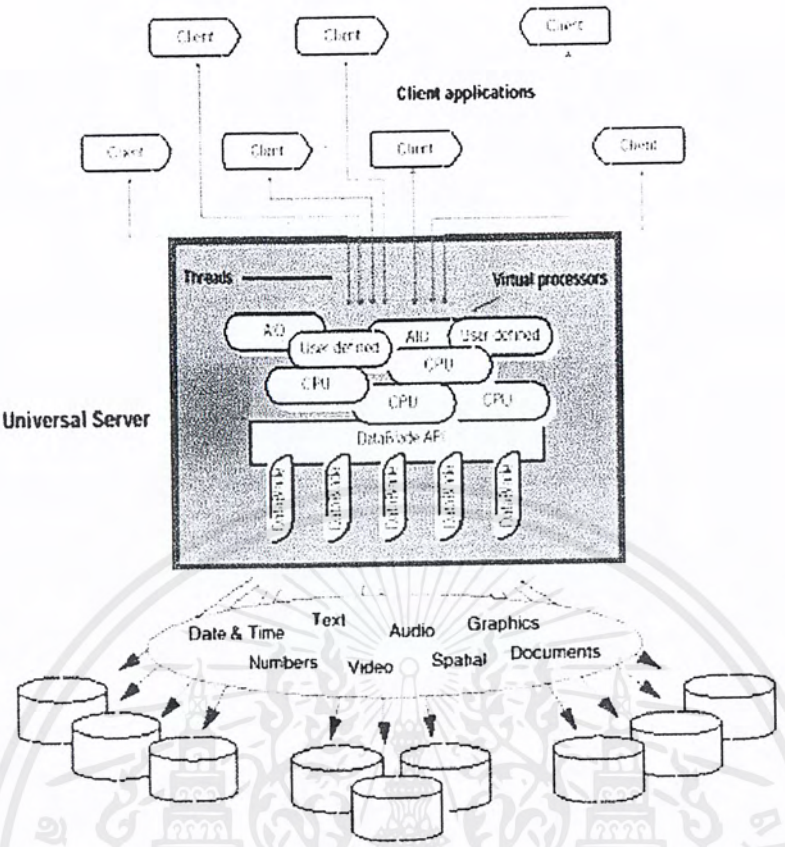
ซึ่งอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ จะมีลักษณะพิเศษ ดังต่อไปนี้

- รองรับชนิดข้อมูลได้หลายชนิด อาทิเช่น
  - ชนิดข้อมูลที่มีอยู่แล้ว (Built-In Data Type)
  - ชนิดข้อมูลที่ผู้ใช้กำหนด (User-Defined Data Type)
  - ชนิดข้อมูลที่มีความซับซ้อน (Complex Data Type)
- ผู้ใช้สามารถกำหนดรูทีนขึ้นใช้เองได้
- มีดาต้าเบลดโมดูล
- รองรับข้อมูลชนิดลาจออบเจ็กต์ (Large Object) คือ เป็นข้อมูลที่มีขนาดใหญ่ เช่น ไฟล์ภาพ ไฟล์ข้อความ ไฟล์เสียง เป็นต้น
- สามารถถ่ายทอดคุณสมบัติ (Inheritance) ของโรว์ไทป์ (Row Type) และเทเบิลไทป์ (Table Type)

4.2 สถาปัตยกรรมของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์

อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์สนับสนุนการทำงานแบบไคลเอ็นต์/เซิร์ฟเวอร์ (Client/Server) โดยจะเป็นในลักษณะของการติดต่อระหว่างโปรแกรมประยุกต์ไคลเอ็นต์กับดาต้าเบสเซิร์ฟเวอร์ ซึ่งสามารถจำลองสถาปัตยกรรมของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ ได้ดังรูปที่ 4-1 โดยจะแบ่งออกเป็นระดับต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-1 รูปแสดงสถาปัตยกรรมของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์

4.2.1 ระดับไคลเอ็นต์แอพพลิคชั่น (Client Application)

จะเป็นส่วนที่มีแอพพลิเคชั่นโปรแกรมที่ติดต่อกับผู้ใช้ โดยฝั่งไคลเอ็นต์จะสร้างลอจิกคอลคอนเนคชั่น (Logical Connection) กับเซิร์ฟเวอร์โดยมีชนิดของการเชื่อมต่อ ดังนี้

- **Local Connection** ได้แก่ Shared Memory และ Stream Pipe
- **Network Connection** ได้แก่ Computer-to-Computer และ Local Loopback

Interface	Network Protocol
Socket	TCP/IP
TLI	TCP/IP
TLI	SPX/IPX

ตารางที่ 4-1 ชนิดของอินเตอร์เฟซกับโปรโตคอลที่ยูนิเวอร์แซลเซิร์ฟเวอร์สนับสนุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ระดับของยูนิเวอร์แซลเซิร์ฟเวอร์ (Universal Server)

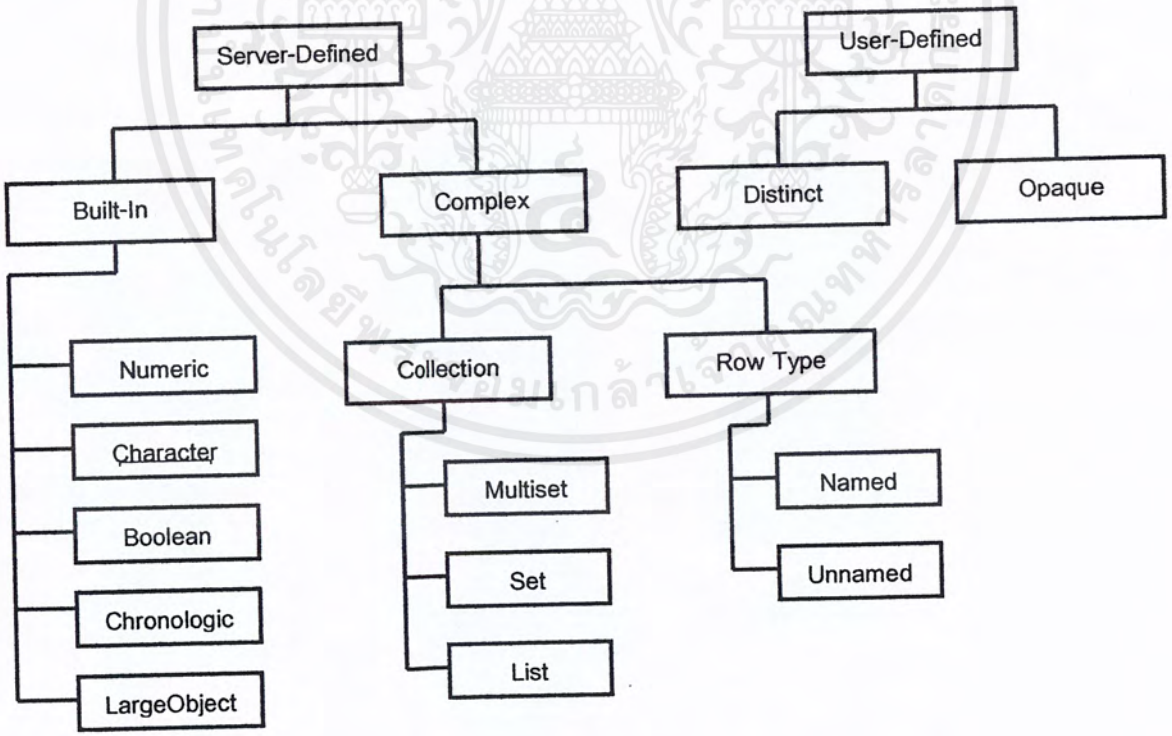
ซึ่งเป็นดาต้าเบสเซิร์ฟเวอร์ ที่ใช้ในการจัดการเอ็กซ์เทอร์นอลวิว (External View) และ คอนเซพชวลวิว (Conceptual View) โดยมีส่วนประกอบ ดังนี้

- **Parser** ทำหน้าที่อ่านและตรวจสอบไวยากรณ์ โดยฟังก์ชั่นที่ผู้ใช้สร้างขึ้นมา (User-defined Function) และชนิดข้อมูลจะถูกบันทึกลงในแค้ตตาลอกเทเบิล (Catalog Table) และจะถูกตรวจสอบโดยพาร์เซอร์
- **Optimizer** ทำหน้าที่เลือกทางที่ดีที่สุดเพื่อการเข้าถึง (Access) ข้อมูลที่ถูกคิวรี (Query)
- **Function Manager** ทำหน้าที่หาและจัดการ User-defined Function เพราะว่าฟังก์ชั่นที่ผู้ใช้สร้างขึ้นมาจะถูกเก็บไว้ในแค้ตตาลอกเทเบิล
- **Access Method** ที่เหมาะสมจะถูกเลือกมาสำหรับข้อมูลที่กำลังถูกแก้ไข
- **Data Manager** ทำหน้าที่เคลื่อนย้ายข้อมูลเข้าออกจากดิสก์

4.2.3 ระดับฐานข้อมูลเชิงกายภาพ (Physical Database)

เป็นส่วนที่ใช้จัดเก็บข้อมูลจริงๆ ของฐานข้อมูล

4.3 ชนิดข้อมูล (Data Type)



รูปที่ 4-2 แสดงชนิดข้อมูลของอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดข้อมูลในอินฟอร์มิกซ์ สามารถแบ่งอธิบายออกได้เป็น 2 หัวข้อใหญ่ๆ ดังนี้

4.3.1 ชนิดข้อมูลที่เซิร์ฟเวอร์กำหนด (Server-Defined) จำแนกได้เป็น

4.3.1.1 บิวท์อินดาต้าไทป์ (Built-in data Types) มีลักษณะดังนี้

- เป็นชนิดข้อมูลพื้นฐานซึ่งดาต้าเบสเซิร์ฟเวอร์มีให้ใช้
- ไม่สามารถแบ่งเป็นส่วนย่อยๆ ได้อีก
- บิวท์อินดาต้าไทพ์ในอินฟอร์มิกซ์ มีดังตารางที่

DATA TYPE	EXPLANATION
BLOB	เก็บข้อมูลไบนารีในชังค์ที่เข้าถึงแบบสุ่ม (Random-access Chunks)
BOOLEAN	เก็บค่าบูลีน คือ True และ False.
BYTE	เก็บข้อมูลไบนารีในชังค์ที่ไม่เข้าถึงแบบสุ่ม
CHAR(n)	เก็บซิงเกิลไบท์ (Single-byte) หรือมัลติเพิล (Multibyte) ตามลำดับของตัวอักษร รวมทั้งตัวหนังสือ,ตัวเลขและสัญลักษณ์ซึ่งมีความยาวจำกัดตามค่า n
CHARACTER(n)	เหมือนกับข้อมูลชนิด CHAR
CHARACTER	เก็บซิงเกิลไบท์ (Single-byte) หรือมัลติเพิล (Multibyte) ตามลำดับของตัวอักษร
VARYING(m,r)	รวมทั้งตัวหนังสือ,ตัวเลขและสัญลักษณ์
CLOB	เก็บข้อมูลที่เป็น TEXT ในชังค์ที่เข้าถึงแบบสุ่ม
DATETIME	เก็บข้อมูลวันที่และเวลา
DATE	เก็บข้อมูลวันที่ตามปฏิทิน
DECIMAL	เก็บค่าตัวเลขที่เป็นทศนิยมที่แน่นอน
DEC	เหมือนกับ DECIMAL
FLOAT(n)	เก็บค่า floating-point numbทำให้เกิด double data type ในภาษาซี
DOUBLE PRECISION	เหมือนกับ FLOAT.
INT8	เก็บค่าจำนวนเต็มขนาด 8 ไบท์ มีค่าระหว่าง –9,223,372,036,854,775,87 ถึง 223,372,036,854,775,807 หรือ $-(2^{63}-1)$ ถึง $(2^{63}-1)$
INTEGER	เก็บค่าจำนวนเต็ม ที่มีค่าระหว่าง –2,147,483,647 ถึง +2,147,483,647 หรือ $-(2^{31}-1)$ ถึง $(2^{31}-1)$
INT	เมือนกับ INTEGER
INTERVAL	เก็บค่าช่วงของเวลา
LVARCHAR	เก็บค่าซิงเกิลไบท์หรือมัลติเพิลไบท์ของตัวอักษร , ตัวเลข และสัญลักษณ์ซึ่งมีความยาวเปลี่ยนแปลงได้มากถึง 32 กิโลไบท์ ส่วนใหญ่จะเป็นรูปแบบเอ็กซ์เทอนอลสตอร์เรจสำหรับโอเปคไทพ์
MONEY(p,S)	เก็บอัตราแลกเปลี่ยนของเงิน
NCHAR(n)	เก็บค่าซิงเกิลไบท์และมัลติเพิลไบท์ของตัวอักษร , ตัวเลข และสัญลักษณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA TYPE	EXPLANATION
NUMERIC(p,s)	เก็บค่าซิงเกิลไบท์และมัลติเพิลไบท์ของตัวอักษร , ตัวเลข และสัญลักษณ์ต่างๆ เหมือนกับ DECIMAL
NVARCHAR(m,r)	เก็บค่าซิงเกิลไบท์และมัลติเพิลไบท์ของตัวอักษร , ตัวเลข และสัญลักษณ์ต่างๆ ซึ่งมีความยาวได้มากที่สุด 255 ไบท์
REAL	เหมือนกับ SMALLFLOAT.
SERIAL	เก็บจำนวนเต็มที่เป็นลำดับ ซึ่งมีขอบเขตของค่าเหมือน INTEGER
SERIAL8	เก็บจำนวนเต็มที่เป็นลำดับ ซึ่งมีขอบเขตของค่าเหมือน INT8
SMALLFLOAT	เก็บค่า single-precision floating-point numbers ซึ่งเป็น float data type ในภาษาซี
SMALLINT	เก็บค่าตัวเลขตั้งแต่ค่า –32,767 ถึง +32,767 หรือ $-(2^{15}-1)$ ถึง $(2^{15}-1)$
TEXT	เก็บข้อมูลชนิดเท็กซ์ในชังค์ที่ไม่เข้าถึงแบบสุ่ม
VARCHAR(m,r)	เก็บค่าซิงเกิลไบท์และมัลติเพิลไบท์ของตัวอักษร , ตัวเลข และสัญลักษณ์ต่างๆ ซึ่งมีความยาวได้มากที่สุด 255 ไบท์

ตารางที่ 4-2 แสดง Built-in Data Type ในอินฟอร์มิกซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

ซึ่งจะขอกล่าวถึงชนิดข้อมูลแบบลาจออบเจ็กต์ (Large Object) ซึ่งเป็นข้อมูลชนิดพื้นฐานชนิดหนึ่งที่มีเพิ่มเข้ามาในอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ ซึ่งเพิ่มความสามารถด้านมัลติมีเดีย ใช้เก็บข้อมูลที่มีขนาดใหญ่ เช่น ไฟล์ภาพ ไฟล์เสียง ไฟล์ข้อความ ต่างๆ ได้ มีอยู่ด้วยกัน 2 แบบ คือ

- ซิมเปิลลาจออบเจ็กต์ (Simple Large Object) มีขนาดไม่เกิด 4 กิกะไบท์ มี 2 ชนิดคือ BYTE (ใช้เก็บข้อมูลแบบไบนารี่ดาต้า) และ TEXT (ใช้เก็บข้อมูลแบบเท็กซ์ดาต้า) ซึ่งทั้งคู่ไม่สนับสนุนการเข้าถึงแบบสุ่ม หมายความว่า ถ้าจะเข้าถึงข้อมูลต้องเข้าถึงข้อมูลทั้งชุด
- สมาร์ทลาจออบเจ็กต์ (Smart Large Object) เป็นชนิดข้อมูลแบบลาจออบเจ็กต์ที่สามารถทำการค้นหา อ่าน และเขียนข้อมูลภายในออบเจ็กต์ได้ นอกจากนี้ยังทำการกู้คืนและการโรลแบคทรานแซคชั่นได้อีกด้วย โดยในตารางฐานข้อมูลจะมิได้เก็บข้อมูลชนิดนี้ไว้ในตารางโดยตรง แต่จะเป็นเพียงการเก็บพอยเตอร์ที่ชี้ไปยังตำแหน่งที่เก็บข้อมูลนั้นใน sbspace

4.3.1.2 คอมเพล็กซ์ดาต้าไทป์ (Complex Data Type) เป็นชนิดข้อมูลที่ประกอบไปด้วยข้อมูลชนิดอื่นๆ ที่เป็นบิวท์อินไทป์ ซึ่งแบ่งออกได้เป็น 2 ชนิด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *คอลเลคชั่นไทป์ (Collection Types)* เกิดจากการรวมกันของชนิดข้อมูลชนิดเดียวกัน ซึ่งอาจเป็นชนิดข้อมูลพื้นฐาน หรืออาจเป็นคอมเพล็กซ์ดาต้าไทป์ก็ได้ โดยคอลเล็คชั่นไทป์มี 3 ชนิด ดังนี้
  - เซต (SET) : สมาชิกภายในไม่มีลำดับและซ้ำกันไม่ได้ (ซึ่งจะเป็นชนิดข้อมูลหลักที่ใช้ในการแก้ปัญหาเชิงเวลา และจะกล่าวถึงวิธีการในการเข้าถึงเซต ต่อไป)
  - ลิส (LIST) : สมาชิกภายในมีลำดับและซ้ำกันได้
  - MULTISET : สมาชิกภายในไม่มีลำดับและซ้ำกันได้
- *โรว์ไทป์ (Row Types)* เป็นชนิดข้อมูลที่มีลักษณะคล้ายกับข้อมูลแบบโครงสร้างในภาษาซี โดยภายในจะประกอบด้วยฟิลด์ (Field) ซึ่งแต่ละฟิลด์จะมีชื่อและชนิดเป็นของตนเอง ในอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์จะมีโรว์ไทป์อยู่ 2 แบบ คือ Named Row Type และ Unnamed Row Type (ซึ่งเราจะใช้ข้อมูลชนิดโรวท์ในการสร้างโรว์ไทป์ของแอททริบิวท์ของข้อมูลที่เกาะกับเวลา)

4.3.2 ชนิดข้อมูลที่ผู้ใช้กำหนด (User-Defined Type)

เป็นชนิดข้อมูลที่ผู้ใช้สร้างขึ้นใหม่ นอกเหนือจากข้อมูลพื้นฐานที่มีให้ในอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ เพื่อเพิ่มความยืดหยุ่นในการเก็บข้อมูลและการจัดการ โดยแบ่งออกเป็น 2 ชนิด ดังนี้

4.3.2.1 โอเปคไทป์ (Opaque Type)

เป็นชนิดข้อมูลที่มีลักษณะซ่อนเร้นข้อมูลไว้ภายใน อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์จะไม่รู้จักโครงสร้างภายในของมัน ดังนั้นในการกำหนดโอเปคไทป์จึงต้องมีการกำหนดฟังก์ชันและโอเปอเรชั่นที่ใช้ในการติดต่อกับโครงสร้างภายในของโอเปคไทป์นั้นด้วย รวมถึงต้องกำหนดแคสติ้งฟังก์ชันที่ทำหน้าที่ในการแปลงชนิดข้อมูลอื่นไปเป็นโอเปคไทป์หรือแปลงชนิดข้อมูลโอเปคไทป์ไปเป็นชนิดข้อมูลอื่นด้วย

4.3.2.2 คิสติงค์ไทป์ (Distinct Type)

เป็นชนิดข้อมูลที่โครงสร้างภายในเหมือนกับชนิดข้อมูลที่มีอยู่แล้ว โดยที่จะมีชื่อและฟังก์ชันที่ถูกสร้างมาแตกต่างไปจากข้อมูลที่เป็นต้นแบบ

4.4 การควบคุมความถูกต้องของข้อมูล (Integrity Constrant)

อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์สามารถควบคุมความถูกต้องของข้อมูลได้ 3 ทาง คือ

1) การควบคุมความถูกต้องของเอนติตี้ (Entity Integrity) เป็นการควบคุมความถูกต้องของแต่ละเอนติตี้ คือ ทุกๆ แถวจะต้องมีตัวระบุความแตกต่าง (Unique Identifier)
2) การควบคุมความถูกต้องของความหมาย (Semantic Integrity) เป็นการควบคุมความถูกต้องของความหมายของข้อมูล คือ ทุกๆ คอลัมน์จะต้องมีค่าถูกต้องตามที่กำหนดไว้
3) การควบคุมความถูกต้องของการอ้างถึง (Referential Integrity) เป็นการควบคุมความถูกต้องที่เกี่ยวกับความสัมพันธ์ระหว่างตาราง ในส่วนของคีย์หลัก (Primary key) และคีย์ภายนอก (Foreign key)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 อินฟอร์มิกซ์กับคุณสมบัติเชิงวัตถุ

เนื่องจากในอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ เป็นฐานข้อมูลเชิงวัตถุสัมพันธ์จึงสามารถแสดงคุณสมบัติเชิงวัตถุได้ ดังนี้

4.5.1 การสืบทอดคุณสมบัติ (Inheritance)

ไทป์ตาราง (Table Type) หรือโรว์ไทป์มีชื่อ (Named Row Type) สามารถที่จะทำการสืบทอดคุณสมบัติในลักษณะการจัดลำดับชั้นได้ และให้มีการถ่ายทอดคุณสมบัติจากไทป์แม่ไทป์เดียวเท่านั้น (Single Inheritance) โดยสามารถเพิ่มคุณสมบัติของตัวเองเข้าไปได้ ซึ่งแบ่งการถ่ายทอดคุณสมบัติเป็น 2 ชนิด คือ

1) การถ่ายทอดคุณสมบัติของชนิด (Type Inheritance) จะสามารถถ่ายทอดคุณสมบัติได้เฉพาะโรว์ไทป์มีชื่อ (Named Row Type) เท่านั้น โดยใช้คีย์เวิร์ด UNDER ดังตัวอย่างโค้ดภาษา SQL3 ดังนี้

```
CREATE ROW TYPE HUMAN(NAME CHAR(15),AGE INT);

CREATE ROW TYPE MALE UNDER HUMAN;
CREATE ROW TYPE FEMALE (PREGNANT CHAR) UNDER HUMAN;
```

จากตัวอย่างจะเป็นการสร้างชนิดแถว HUMAN ชนิดแถว MALE และ FEMALE ตามลำดับ ซึ่งมีการถ่ายทอดคุณสมบัติมาจากชนิดแถว HUMAN โดยชนิดแถว MALE จะมีทั้งหมด 2 แอตทริบิวต์ คือ NAME และ AGE ส่วนชนิดแถว FEMALE จะมีทั้งหมด 3 แอตทริบิวต์ คือ NAME,AGE และ PREGNANT

2) การถ่ายทอดคุณสมบัติของตาราง (Table Inheritance) เป็นการถ่ายทอดคุณสมบัติจากตารางที่กำหนดจากโรว์ไทป์มีชื่อ โดยจะถ่ายทอดคุณสมบัติ ดังต่อไปนี้ทั้งหมด

- คอลัมน์ทั้งหมดของซูเปอร์เทเบิล
- การควบคุมความถูกต้องของการอ้างถึง
- การควบคุมความถูกต้องของตารางต่างๆ
- ดัชนี (Index)
- รายละเอียดในการจัดเก็บ
- ทริกเกอร์
- วิธีการเข้าถึงข้อมูล (Access Method)

ตัวอย่างคำสั่งภาษา SQL3 ในการถ่ายทอดคุณสมบัติของตาราง

```
CREATE TABLE Person OF TYPE Person_t;
CREATE TABLE Employee OF TYPE Employee_t
UNDER Person;
```

จากตัวอย่างจะเป็นการสร้างตาราง Person ที่มีชนิดเป็น Person_t และตาราง Employee ที่มีชนิดเป็น Employee_t โดยสืบทอดคุณสมบัติมาจากตาราง Person

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 โพลิมอร์ฟิซึม (Polymorphism)

เป็นการสร้างรูทีนที่ทำงานต่างกัน หรือมีอาร์กิวเมนต์ต่างกัน แต่มีชื่อเดียวกัน ซึ่งมีประโยชน์เพื่อ

- สร้างรูทีนที่มีชื่อเหมือนกับบิวท์อินฟังก์ชัน เพื่อใช้งานกับชนิดข้อมูลที่ผู้ใช้สร้างขึ้นใหม่
- สร้างลำดับชั้นของไทป์ โดยที่สับไทป์มีการถ่ายทอดคุณสมบัติต่างๆ และรูทีนจากซูเปอร์ไทป์ด้วย
- สร้างดิสติงค์ไทป์ ที่มีโครงสร้างภายในเหมือนกับชนิดข้อมูลเดิมที่มีอยู่แล้ว เพียงแต่มีชื่อต่างกัน ซึ่งจะมีการถ่ายทอดรูทีนที่ใช้ของไทป์เริ่มต้นมาด้วย ซึ่งเราสามารถทำโอเวอร์โหลดกับรูทีนเหล่านี้ได้

ตัวอย่างคำสั่งภาษา SQL3 ในการสร้างดิสติงค์ไทป์

```
CREATE DISTINCT TYPE Fahrenheit AS INTEGER';
```

จากตัวอย่างเป็นการสร้างชนิดข้อมูล Fahrenheit ขึ้นมาจากชนิดข้อมูล INTEGER ซึ่งเป็นบิวท์อินไทป์ ทำให้สามารถใช้ชนิดข้อมูลใหม่นี้แทนชนิดข้อมูลเดิมได้ซึ่งจะช่วยให้สื่อความหมายได้ดีกว่า

ตัวอย่างการทดลองคุณสมบัติโพลิมอร์ฟิซึมของ Informix โดยใช้โค้ด SQL3 เป็นดังนี้

```
CREATE ROW TYPE HUMAN(NAME CHAR(15),AGE INT);
CREATE ROW TYPE MALE UNDER HUMAN;
CREATE ROW TYPE FEMALE (PREGNANT CHAR) UNDER HUMAN;

CREATE ROW TYPE PLANT(NAME CHAR(15),AGE INT);
```

จากโค้ด จะได้ชนิดแถว HUMAN เป็นซูเปอร์ไทป์และชนิดแถว MALE และ FEMALE เป็นซับไทป์ของชนิดแถว HUMAN ส่วนชนิดแถว PLANT เป็นอีกชนิดแถวหนึ่งที่มีชนิดข้อมูลเหมือนกับชนิดแถว HUMAN จากนั้นจึงได้สร้างฟังก์ชันเพื่อทดสอบคุณสมบัติโพลิมอร์ฟิซึมแบบโอเวอร์โหลด (Overload) ดังนี้

```
CREATE FUNCTION TEST1(X HUMAN)
        RETURNING CHAR(8);
        RETURN 'IS HUMAN';
END FUNCTION;

CREATE FUNCTION TEST1(X PLANT)
        RETURNING CHAR(8);
        RETURN 'IS PLANT';
END FUNCTION;

EXECUTE FUNCTION TEST1(ROW('ADAM',20)::HUMAN);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EXECUTE FUNCTION TEST1(ROW('BEAN',100)::PLANT);
```

จากโค้ด ฟังก์ชันชื่อ TEST1 ซึ่งรับพารามิเตอร์ที่มีชนิดข้อมูลต่างกัน คือ HUMAN และ PLANT ตามลำดับ เมื่อทำการทดสอบโดยการเรียกฟังก์ชัน TEST1 โดยส่งพารามิเตอร์เป็นชนิดข้อมูล HUMAN จะได้ผลลัพธ์เป็น 'IS HUMAN' แต่ถ้าส่งพารามิเตอร์เป็นชนิดข้อมูล PLANT จะได้ผลลัพธ์เป็น 'IS PLANT' จึงสรุปได้ว่า INFORMIX ยอมให้ใช้ชื่อรูทีนซ้ำกันได้ โดยรับพารามิเตอร์ต่างกัน ซึ่งเป็นคุณสมบัติสมบัติโพลิมอร์ฟิซึมแบบโอเวอร์โหลดนั่นเอง

```
CREATE FUNCTION TEST2(X HUMAN)
        RETURNING CHAR(15);
        RETURN 'IS HUMAN';
END FUNCTION;

CREATE FUNCTION TEST2(X MALE)
        RETURNING CHAR(15);
        RETURN 'IS MALE';
END FUNCTION;

EXECUTE FUNCTION TEST2(ROW('ADAM',22)::HUMAN);
EXECUTE FUNCTION TEST2(ROW('ADAM',22)::MALE);
EXECUTE FUNCTION TEST2(ROW('EVE',16,'Y')::FEMALE);
```

จากโค้ดข้างต้น เป็นการสร้างฟังก์ชัน TEST2 เพื่อทดสอบคุณสมบัติโพลิมอร์ฟิซึมแบบโอเวอร์ไรด์ (OVERRIDE) โดยรับพารามิเตอร์เป็นชนิดข้อมูล HUMAN และ MALE ตามลำดับ ซึ่ง MALE และ FEMALE เป็นซับไทป์ของ HUMAN เมื่อเรียกใช้ฟังก์ชัน TEST2 โดยส่งพารามิเตอร์เป็นชนิดข้อมูล HUMAN จะได้ผลลัพธ์เป็น 'IS HUMAN' ถ้าส่งพารามิเตอร์เป็นชนิดข้อมูล MALE จะได้ผลลัพธ์เป็น 'IS MALE' แต่ถ้าส่งพารามิเตอร์เป็นชนิดข้อมูล FEMALE ซึ่งไม่ได้สร้างฟังก์ชันขึ้นมารองรับสำหรับซับไทป์นี้ ก็จะทำการเรียกฟังก์ชัน TEST2 ที่รับชนิดข้อมูล HUMAN ซึ่งเป็นซูเปอร์ไทป์ จึงได้ผลลัพธ์เป็น 'IS HUMAN' สรุปได้ว่า INFORMIX ยอมให้ใช้ชื่อรูทีนซ้ำกันได้ สำหรับซูเปอร์ไทป์และซับไทป์ ซึ่งเป็นคุณสมบัติสมบัติโพลิมอร์ฟิซึมแบบโอเวอร์ไรด์นั่นเอง

4.5.3 การปกป้องข้อมูล (Encapsulation)

การปกป้องข้อมูลนั้น สามารถพิจารณาได้จากการกำหนดรูทีนขึ้นใช้เอง ซึ่งเราสามารถกำหนดรูทีนขึ้นใช้เองได้จากภาษา SQL (Structure Procedure Language) หรือสเตตเมนต์ภายนอกอื่นๆ (External Statement) เช่น อาจเขียนด้วยภาษาซี หรือภาษาเชิงโครงสร้าง SPL (Structure Procedure Language) ซึ่งสามารถประกาศเป็นฟังก์ชันหรือโพรซีเยอร์ก็ได้ และสามารถกำหนดสิทธิการเข้าใช้ข้อมูลได้ รวมทั้งยังสามารถรวบรวมคำสั่งภาษาเอสคิวแอลหลายๆ คำสั่งไว้ด้วยการเรียกใช้รูทีนเพียงตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 เครื่องมือช่วยในการทำงาน

ในอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ จะมีเครื่องมือที่ช่วยในการทำงานที่มาพร้อมกับระบบจัดการฐานข้อมูล อยู่หลายตัวด้วยกัน ที่สำคัญ ๆ ได้แก่

4.6.1 เบลดสมิธ (BladSmith)

จะเป็นเครื่องมือที่ใช้สร้างโปรเจ็กต์และกำหนดวัตถุต่างๆ เช่น ชนิดข้อมูลและรูทีน เป็นต้น และจะทำการรวมชนิดข้อมูลและรูทีนต่างๆ ที่สร้างขึ้นเป็นดาต้าเบลดโมดูลหนึ่งให้เป็นแพ็กเกจจิ่งไฟล์ เพื่อใช้ติดตั้งต่อไป

4.6.2 เบลดแพก (BladePack)

เป็นเครื่องมือที่นำแพ็กเกจจิ่งไฟล์จากเบลดสมิธที่พร้อมจะนำไปติดตั้ง ไปเก็บไว้ในยูนิเวอร์แซลเซิร์ฟเวอร์ เพื่อรอการลงทะเบียน (register)

4.6.3 เบลดเมเนเจอร์ (BladeManager)

เป็นเครื่องมือยูทิลิตี้ ที่ทำการลงทะเบียนและยกเลิกการลงทะเบียดาต้าเบลดโมดูลที่ต้องการหรือไม่ต้องการในฐานข้อมูลหนึ่งๆ

4.6.4 อินฟอร์มิกซ์เซตเนต 32 (Informix Setnet32)

เป็นเครื่องมือที่ใช้ในการกำหนดหรือแก้ไขตัวแปรสภาวะแวดล้อม (Environment Variable) และพารามิเตอร์ของเนตเวิร์ค (Network Parameter) ขณะรันไทม์ โปรแกรมนี้จะติดตั้งบนฝั่งไคลเอ็นต์

4.6.5 สคีมาโนเลดจ์ (Schema Knowledge)

เป็นเครื่องมือที่ใช้แสดงข้อมูลของฐานข้อมูลแบบกราฟฟิก ใช้แสดงเมต้าดาต้าของฐานข้อมูล แต่ไม่สามารถใช้แก้ไขข้อมูลใดๆ บนฐานข้อมูลได้ ซึ่งสามารถแสดงมุมมองของออบเจ็กต์ หรือมุมมองของการถ่ายทอดคุณสมบัติของตารางก็ได้ โดยเมต้าดาต้าที่แสดงนั้น ได้แก่

- ชื่อฐานข้อมูล
- วิธีการเข้าถึงข้อมูลที่มี
- บิวท์อินฟังก์ชันที่มี
- การแคสข้อมูลที่มี
- โอเปอเรเตอร์ที่มี
- รูทีนที่มี
- ตารางที่มี
- ชนิดข้อมูลและชนิดของแถวที่มี

4.6.6 เอสคิวแอลอีดิเตอร์ (SQL Editor)

เป็นเครื่องมือที่ใช้ในการติดต่อกับยูนิเวอร์แซลเซิร์ฟเวอร์ เพื่อทำการจัดการฐานข้อมูล ซึ่งจะแสดงผลเป็นกราฟฟิก เช่น การค้นหาข้อมูล จะแสดงเอาท์พุตเป็นตารางข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.7 ดาต้าไดเร็กเตอร์ (Data Director)

เป็นเครื่องมือช่วยให้ติดต่อโครงของฐานข้อมูล กับฟอร์มที่เราสร้างขึ้นในสภาพแวดล้อมของการพัฒนาแอพพลิเคชัน ซึ่งมีประโยชน์โดยหลักๆ ดังนี้

- สามารถเข้าถึงฟังก์ชันในการจัดการข้อมูลได้โดยอัตโนมัติ ได้แก่ การประสานกันของ master-detail ซึ่งจะช่วยให้ประหยัดเวลาในการโปรแกรม
- สามารถเข้าถึงความสามารถที่ขยายเพิ่มขึ้นของ Informix Dynamic Server with Universal Data Option
- Data Director จะเข้าถึงข้อมูลให้โดยอัตโนมัติโดยการค้นหาข้อมูลตามเงื่อนไขการค้นหาที่กำหนด, การจัดการการเชื่อมต่อฐานข้อมูลและตำแหน่งที่อยู่ รวมทั้งสามารถทำการจัดการและมีความสามารถในด้านความปลอดภัย และการจัดเรียงข้อมูลด้วย

ดาต้าไดเรกเตอร์จะอำนวยความสะดวกในการจัดการข้อมูล โดยกระบวนการของการจัดการข้อมูลพื้นฐานจะกระทำโดยอัตโนมัติ ซึ่งได้แก่ การบันทึก, การแก้ไข และการเลื่อนไปยังตำแหน่งข้อมูลที่ต้องการ ซึ่งการแสดงผลและการสร้างการเชื่อมต่อกับฐานข้อมูล จะแสดงผลในรูปแบบกราฟฟิกซึ่งง่ายต่อการติดต่อกับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การประยุกต์ฐานข้อมูล

5.1 การสร้างฐานข้อมูลเชิงเวลาบนอินฟอร์มิกซ์

ในบทนี้เป็นการทดลองคุณสมบัติต่างๆของระบบฐานข้อมูล Informix ในเรื่องของชนิดข้อมูลที่จะใช้ในการเก็บข้อมูลที่มีการเปลี่ยนแปลงเชิงเวลา ซึ่งจากเดิมการสร้างตารางที่เป็นเชิงเวลานั้นจะเอาวาลิดไทม์เกาะกับทั้งทัพเปิล (Tuple) และใช้การอ้างถึงกันแบบ primary key-foreign key ซึ่งทำให้เกิดปัญหาใน 2 ประเด็น คือ

1. ปัญหาเรื่อง Identifier : พบว่าถ้าใช้ข้อมูลเชิงเวลาในฐานข้อมูลเชิงสัมพันธ์นั้น จะมีปัญหาเรื่อง primary key และการใช้อ้างถึงแบบเป็น foreign key จะทำได้ยุ่งยาก เพราะต้องรวมเอาวาลิดไทม์เข้ากับ primary key ด้วย เนื่องจากฐานข้อมูลเชิงเวลาเก็บทุกค่าความจริงที่เปลี่ยนแปลงไป เมื่อเวลาเปลี่ยนไป
2. การเอาวาลิดไทม์เกาะกับทั้งทัพเปิล ทำให้ไม่ทราบว่าในการเปลี่ยนแปลงแต่ละครั้งนั้น มีแอตทริบิวต์ใดบ้างที่เปลี่ยนแปลง ซึ่งควรจะต้องบอกได้ว่าแอตทริบิวต์ใดที่เปลี่ยนไป ยกตัวอย่างดังรูปที่ 5-1

EMP

NAME	SURNAME	SALARY	VALID_FROM	VALID_TO
Som	Sak	10000	12/12/1997	12/12/1998
Som	Sak	20000	12/12/1998	12/12/1999
Ra	Dom	20000	12/12/1999	12/12/2000
Chay	Dom	30000	12/12/2000	31/12/9999

รูปที่ 5-1 แสดงตารางที่เก็บข้อมูลเชิงเวลา โดยเอาวาลิดไทม์เกาะทั้งทัพเปิล

5.2 การแก้ไขปัญหาเชิงเวลาโดยใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์

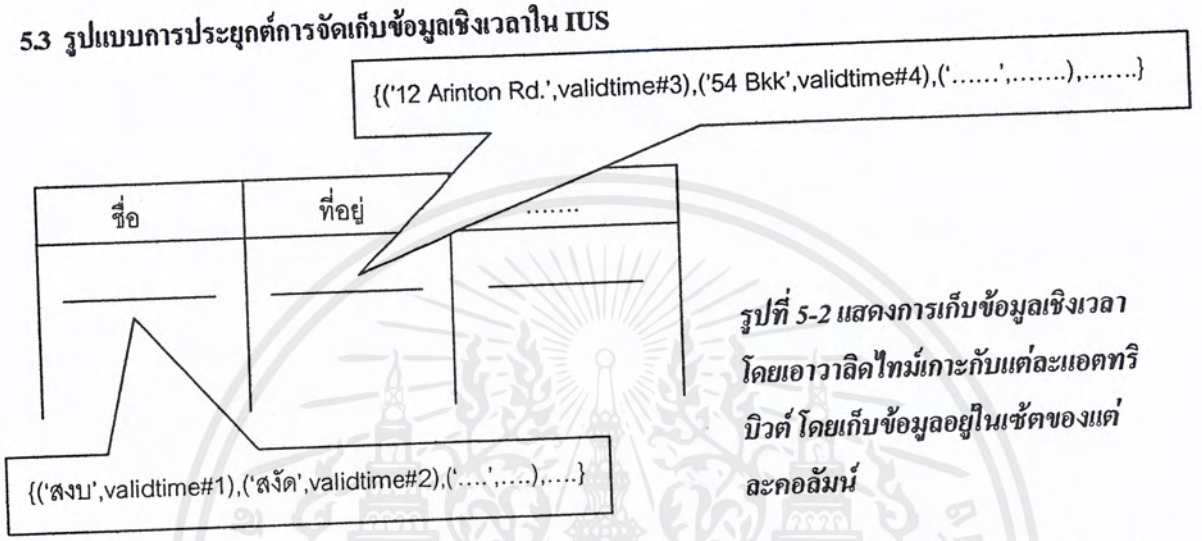
จากปัญหาทั้ง 2 ข้อ สามารถแก้ได้โดยใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object-Relational Database) ได้ดังนี้

1. เรื่อง Identifier : เนื่องจากใน IUS (Informix Universal Server) มี rowid ซึ่งเป็นข้อมูลที่เป็นเอกลักษณ์ (Unique) ภายในตาราง ซึ่งสามารถนำมาใช้เป็น Identifier ได้ จึงแก้ปัญหาเรื่อง Identifier ได้ โดยผู้ใช้ไม่จำเป็นต้องทราบว่ามี rowid นี้อยู่เลย เพราะ rowid เป็นคอลัมน์ที่ถูกซ่อนอยู่ในแต่ละตาราง
2. แทนการนำวาลิดไทม์เกาะกับทั้งทัพเปิล แก้ไขได้โดยการเอาวาลิดไทม์เกาะกับแต่ละแอตทริบิวต์ที่สามารถเปลี่ยนแปลงได้ตามเวลา ทำให้ทราบได้แน่ว่าแอตทริบิวต์ใดเปลี่ยนแปลง โดยจะกำหนดให้แต่ละคอลัมน์เก็บกลุ่มของข้อมูลที่มีวาลิดไทม์กำกับอยู่ ดังรูปที่ 5-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

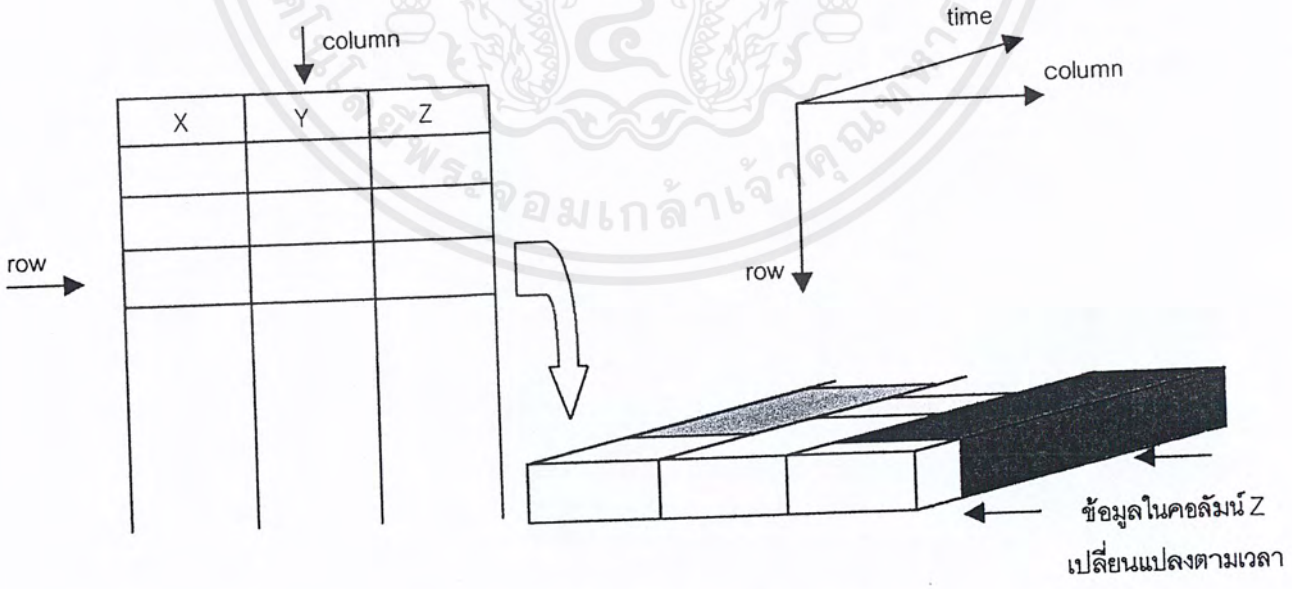
ซึ่งแนวทางการแก้ปัญหาทั้ง 2 ข้อนี้ เป็นข้อได้เปรียบของฐานข้อมูลเชิงวัตถุสัมพันธ์ที่เหนือกว่าฐานข้อมูลเชิงสัมพันธ์ โดยเฉพาะข้อ2 เพราะเนื่องจากฐานข้อมูลเชิงสัมพันธ์นั้น ถูกจำกัดที่ชนิดข้อมูลที่จะต้องเป็นข้อมูลที่ไม่สามารถแยกย่อยได้อีก (atomic) ทำให้ไม่สามารถเก็บข้อมูลเป็นกลุ่มตามรูปแบบดังรูปที่ 5-2 ได้ จึงนับได้ว่าเป็นประโยชน์ของฐานข้อมูลเชิงวัตถุสัมพันธ์นั้น เกิดมาจากการที่มีชนิดข้อมูลให้ใช้ได้หลากหลายกว่านั่นเอง

5.3 รูปแบบการประยุกต์การจัดเก็บข้อมูลเชิงเวลาใน IUS



รูปที่ 5-2 แสดงการเก็บข้อมูลเชิงเวลาโดยเอาวาลิดไทม์เกาะกับแต่ละแอตทริบิวต์ โดยเก็บข้อมูลอยู่ในเซ้ตของแต่ละคอลัมน์

ในรูปแบบการจัดเก็บแบบนี้ ทำให้สามารถมองตารางออกเป็น 3 มิติได้ดังนี้ คือ คอลัมน์ โรว์ และมิติของเวลา ดังรูปที่ 5-3



รูปที่ 5-3 แสดงตารางที่เก็บข้อมูลเชิงเวลาเป็น 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากจะมองตารางเป็น 2 มิติ คือ ทัพเปิลกับคอลัมน์ ตามปกติแล้ว ในการค้นหาข้อมูลยังต้องพิจารณาถึงเวลาอีกด้วย จากรูปที่ 5-3 จะเห็นว่าในแต่ละคอลัมน์มีการเปลี่ยนแปลง ณ เวลาที่ต่างกันไป ดังนั้นการคิวรี่โดยกำหนดเวลาต่างกัน ก็ย่อมจะได้ข้อมูลที่ต่างกันด้วย

จากรูปแบบที่จะใช้ในการเก็บข้อมูลที่มีการเปลี่ยนแปลงตามเวลานั้น จะเห็นได้ว่ามีการใช้ชนิดข้อมูลที่เป็นโรว์ไทป์และคอลเลคชั่นไทป์ ซึ่งพอจะอธิบายได้ดังนี้

{.....,.....,.....}	Collection Type	: เก็บข้อมูลชนิดเดียวกัน
(.....,.....,.....)	Row Type	: เก็บข้อมูลต่างชนิดกัน
{(...,...,...),(...,...,...)}	Collection of Row type	: เก็บข้อมูลที่เป็นโรว์ไทป์ชนิดเดียวกัน

ในโรว์ไทป์นั้นจะนำมาใช้เก็บตัวข้อมูลและวาลิดไทม์ (Valid from,Valid to) ซึ่งถ้าใช้เก็บข้อมูลที่เป็นชนิดจำนวนเต็มจะมีรูปแบบการกำหนด ดังนี้

```
CREATE ROW TYPE ROW_INT (        → ชื่อโรว์ไทป์
        DATA INT,                → ตัวข้อมูล
        VALID_FROM DATE,         → วาลิดไทม์
        VALID_TO DATE
);
```

5.4 การจัดการกับคอลเลคชั่นไทป์

ส่วนคอลเลคชั่นไทป์นั้น ในIUSมี 3 ตัวให้เลือกใช้ คือ เซ็ต,มัลติเซ็ต และ ลิสท์ (Set,Multiset and List) ซึ่งในที่นี้ได้เลือกใช้เซ้ต เนื่องจากคุณสมบัติของเซ็ต ไม่สามารถมีสมาชิกซ้ำกันได้ คือ จะไม่มีโรว์ไทป์ตัวไหนในเซ็ตที่เหมือนกันเลย (ซ้ำซ้อนแบบนอนซีเควนท์) ทำให้เป็นการเช็คเรื่องของ integrity constraint ตั้งแต่ในระดับชนิดข้อมูล แต่ก็มีข้อเสียในเรื่องของการโปรแกรมและประยุกต์ใช้งาน เนื่องจากเซ็ตไม่มีลำดับและข้อมูลที่ใส่เข้าไปใหม่ไม่ได้รับประกันว่าจะอยู่ท้ายสุดของเซ็ต ทำให้ในการโปรแกรมต้องวนลูปเช็คตั้งแต่ต้นจนถึงท้ายเซ็ต ซึ่งเซ็ตมีรูปแบบการกำหนดดังนี้

```
CREATE TABLE EMP(
        NAME CHAR(20),
        SURNAME CHAR(20),
        SALARY SET(ROW_INT NOT NULL)
);
```

→ เงินเดือนเป็นข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ จึงกำหนดชนิดข้อมูลเป็นเซ็ตของ row_int (ซึ่งได้กำหนดไปข้างต้นแล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดชนิดข้อมูลเป็นเซ็ตนั้น ภายในเซ็ตจำเป็นต้องมีสมาชิกอยู่อย่างน้อย 1 ตัว (not null) การพยายามใส่ Null ในโรว์ไทป์ทั้งก้อน (row(null,null,null)) นั้นไม่สามารถทำได้ สำหรับการใส่ข้อมูล (Insert) ลงเซ็ตสามารถทำได้ดังนี้

```
INSERT INTO EMP VALUES(
        'Som',
        'Suk',
        "SET{ROW(10000,'9/9/1997','9/12/2000'),ROW(20000,'9/12/2000','12/31/9999')}"
);
```

จากโค้ด Insert ข้างต้น มีความหมายดังนี้ พนักงานที่ชื่อ 'Som' นามสกุล 'Suk' มีเงินเดือน 10,000 บาท ตั้งแต่วันที่ 9 กันยายน 1997 ถึงวันที่ 12 กันยายน 2000 จึงได้เงินเดือนเพิ่มเป็น 20,000 บาท จนถึงปัจจุบันนี้

```
SELECT ROWID,* FROM EMP
```

	rowid	name	surname	salary
1	257	Som	Suk	set(2) of row_int

	data	valid_from	valid_to
1	10000	09/09/1997	09/12/2000
2	20000	09/12/2000	12/31/9999

รูปที่ 5-4 แสดงข้อมูลที่ Insert ลงในตาราง

จากรูปที่ 5-4 จะเห็นว่ามีคอลัมน์ 'rowid' ซึ่งถูกซ่อนไว้อยู่ และตัวเลข rowid นั้น IUS จะสร้างขึ้นให้เอง โดยจะไม่ซ้ำกันในตารางเดียวกัน (unique)

สำหรับการเข้าถึงข้อมูลภายในเซ็ตนั้นต้องเขียนรูทีนขึ้นมาเอง (User-Defined Routine: UDR) โดยใช้ภาษา Stored Procedure Language (SPL) ซึ่งทำให้ซ่อนคำสั่ง SQL หลายๆคำสั่งในรูทีนเดียว สามารถสรุปข้อดีของ UDR ได้ดังนี้

- รวมคำสั่ง SQL หลายๆคำสั่งที่มีความซับซ้อนไว้ในรูทีนเดียวได้ ทำให้มีประโยชน์อย่างมากในการทำงานแบบ client/server เช่น ในเรื่องการบำรุงรักษา เพราะ สามารถแก้ไขรูทีนภายหลังได้ง่าย เพราะตัวรูทีนอยู่ที่ฝั่งระบบฐานข้อมูล ไม่ได้อยู่ที่ไคลเอ็นท์แต่ละตัว
- เป็นการขยายความสามารถในการทำงานกับชนิดข้อมูลพื้นฐาน นอกเหนือจากรูทีนที่มีอยู่แล้ว
- เป็นการสนับสนุนชนิดข้อมูลที่สร้างขึ้นมาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การกำหนดสิทธิในการเข้าถึงข้อมูล สามารถทำได้โดยผู้บริหารระบบ (Administrator) โดยกำหนดสิทธิในการใช้รูทีนที่สามารถใช้กับข้อมูลที่สร้างขึ้น หรือที่มีอยู่แล้วได้นั่นเอง
- ระบบฐานข้อมูลจะทำการ Opitimize โค้ด SPL ตั้งแต่ตอนคอมไพล์ ซึ่งเป็นการกำหนดเส้นทางที่ดีที่สุดในการเข้าถึงให้กับโค้ดตัวนั้น ทำให้สามารถค้นหาข้อมูลได้อย่างรวดเร็วตอนนำรูทีนไปใช้งาน (run time)

5.4.1 การเข้าถึงเซ็ต

การเข้าถึงเซ็ตโดยใช้ภาษา SPL นั้น จะขออธิบายโดยการยกตัวอย่าง การค้นหาเงินเดือนของนาย 'Som Suk' ณ วันที่ 9 กันยายน 1998 สามารถเขียนฟังก์ชันที่รับพารามิเตอร์เป็น ชื่อ-นามสกุลพนักงานและวันที่ แล้วรีเทิร์นค่าเป็นเงินเดือนได้ดังนี้

```
CREATE FUNCTION FIND_SALARY(THENAME CHAR(20),THESURNAME CHAR
                            (20),THEDATE DATE)
RETURNING INT;
DEFINE THESET COLLECTION;
DEFINE THEROW ROW_INT;

SELECT SALARY INTO THESET FROM EMP
        WHERE NAME=THENAME AND SURNAME=THESURNAME;
FOREACH CURSOR1 FOR
        SELECT * INTO THEROW FROM TABLE(THESET)
        IF (THEROW.VALID_FROM<THEDATE
        AND THEROW.VALID_TO>=THEDATE) THEN
                RETURN THEROW.DATA;
        END IF;
END FOREACH;
END FUNCTION;
```

ชนิดข้อมูลที่จะ return

Select เซ็ตของเงินเดือนลงตัวแปรชนิด Collection

สามารถเข้าถึงข้อมูลภายในเซ็ตได้แล้ว

Select ข้อมูลจากเซ็ตลงตัวแปรชนิดโรว์ไทป์ ซึ่งต้องเป็นชนิดเดียวกับโรว์ไทป์ในเซ็ตนั้น

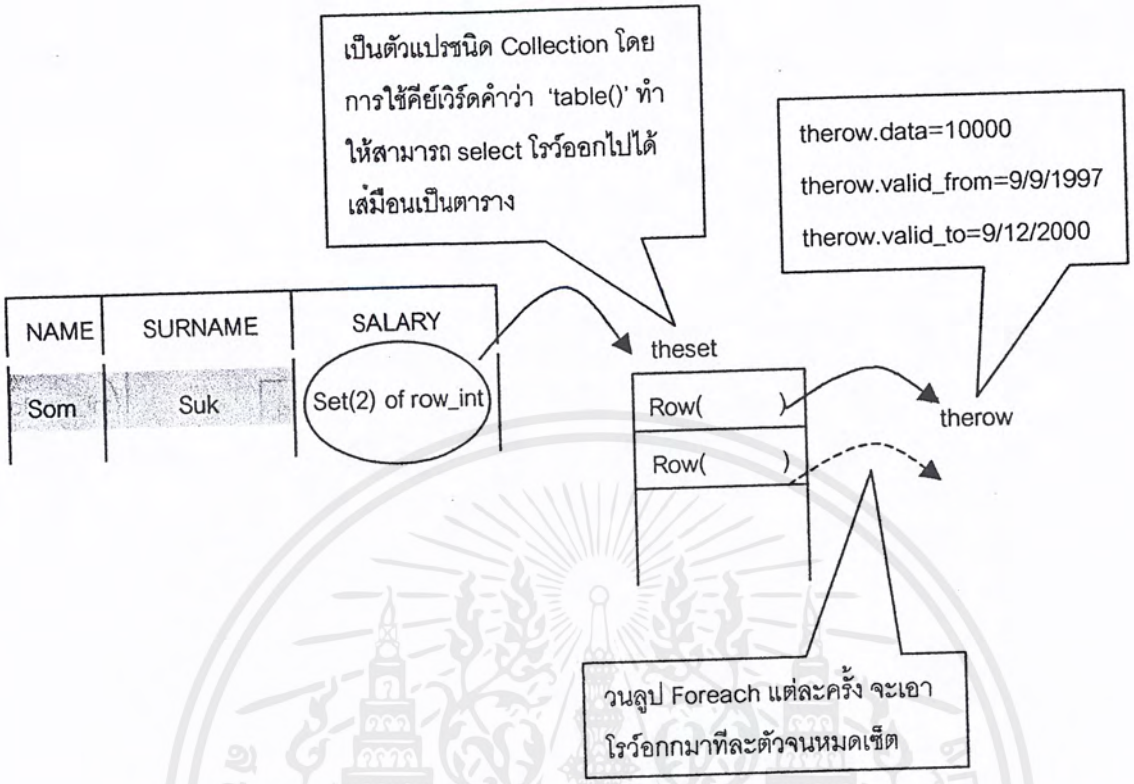
```
EXECUTE FUNCTION FIND_SALARY('Som','Suk','9/9/1998');
```

	(expression)
1	10000

รูปที่ 5-5 แสดงผลลัพธ์การ execute ฟังก์ชัน FIND SALARY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสรุปเป็นรูปได้ ดังนี้



รูปที่ 5-6 อธิบายวิธีการเข้าถึงเซ็ตของ SPL

5.4.2 การเพิ่มข้อมูลลงในเซ็ต

ส่วนการอัพเดทเงินเดือนนั้น คือการแก้ไขวันที่สิ้นสุดของเงินเดือน 20,000 บาท เป็นวันปัจจุบัน และเพิ่มโรว์ใหม่ในเซ็ต เช่น จะเพิ่มเงินเดือนเป็น 30,000 บาท สามารถเขียน SPL โดยรับพารามิเตอร์เข้าเป็น rowid ของพนักงานและเงินเดือนใหม่ ได้ดังนี้

```
CREATE PROCEDURE UPDATE_SALARY(THEROWID INT,THEMONEY INT)
        DEFINE THESET COLLECTION;
        DEFINE THEROW ROW_INT;

        SELECT SALARY INTO THESET FROM EMP WHERE ROWID=THEROWID;
        FOREACH CURSOR1 FOR
                SELECT * INTO THEROW FROM TABLE(THESET)
                IF THEROW.VALID_TO='12/31/9999' THEN
                        LET THEROW.VALID_TO=TODAY;
                        UPDATE TABLE(THESET)(X) SET X=THEROW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    WHERE CURRENT OF CURSOR1;
                LET THEROW=CAST(ROW(THEMONEY,TODAY,'12/31/9999')
                    AS ROW_INT);
                EXIT FOREACH;
            END IF;
        END FOREACH;
        INSERT INTO TABLE(THESET) VALUES(THEROW);
        UPDATE EMP(SALARY) SET SALARY=THESET
            WHERE ROWID=THEROWID;
END PROCEDURE;
```

อธิบายได้ด้วยรูปดังนี้

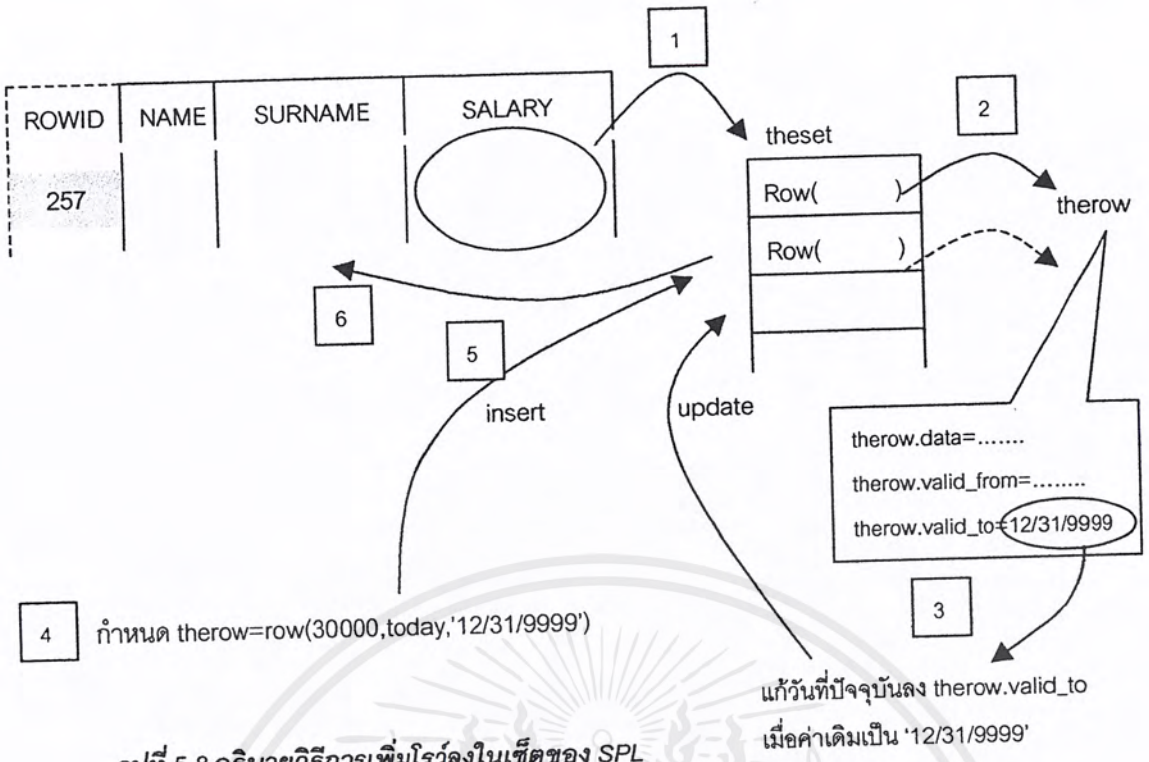
```
EXECUTE PROCEDURE UPDATE_SALARY(257,30000);
```

	rowid	name	surname	salary
1	257	Som	Suk	set(3) of row_int

	data	valid_from	valid_to
1	10000	09/09/1997	09/12/2000
2	20000	09/12/2000	03/18/2001
3	30000	03/18/2001	12/31/9999

รูปที่ 5-7 แสดงผลการอัพเดทเงินเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-8 อธิบายวิธีการเพิ่มโรว์ลงในเซ็ตของ SPL

แบ่งเป็นขั้นตอนได้ดังนี้

1. Select เงินเดือน(ซึ่งเป็นเซ็ต) เก็บในตัวแปร 'theset' ซึ่งมีชนิดเป็น Collection
2. Select โรว์ใน theset ลงในตัวแปร 'therow' ทีละตัวเพื่อหา valid_to ที่เป็น '12/31/9999' (current)
3. เมื่อพบตัว current ให้อัพเดทโรว์ใน theset ที่เคอร์เซอร์กำลังชี้อยู่ ให้ valid_to เป็นวันที่ปัจจุบัน
4. ให้ตัวแปร therow เท่ากับโรว์ใหม่ที่จะเพิ่มลงในเซ็ต
5. insert โรว์ใหม่ลงตาราง table(theset) จะทำให้โรว์ใหม่เข้าไปอยู่ในเซ็ต
6. นำ theset ไปอัพเดทในตารางจริง(EMP) อีกครั้งหนึ่ง

5.5 การทดลองใช้ข้อมูลชนิด blob เพื่อเก็บไฟล์รูปภาพ

การใช้ชนิดข้อมูล blob ในการเก็บไฟล์นั้น ต้องใช้บิวท์อินฟังก์ชันที่สำคัญ 2 ฟังก์ชัน คือ filetoblob และ lotofile มีวิธีการใช้งานดังนี้

FILETOBLOB('filepath','destination') เพื่อเอาไฟล์เก็บลงใน blob

- 'filepath' — ตำแหน่งของไฟล์
- 'destination' — ไฟล์อยู่ที่ client หรือ server

LOTOFILE(column_name,'filepath','destination') เพื่อเอาข้อมูลจาก blob เก็บลงในไฟล์

- column_name — ชื่อคอลัมน์ที่เป็นชนิด blob
- 'filepath' — ตำแหน่งของไฟล์ที่จะใช้เก็บข้อมูล
- 'destination' — ไฟล์อยู่ที่ client หรือ server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ ถ้าต้องการให้ชื่อไฟล์ที่จะเอา blob มาลงนั้นตรงกับที่ตั้งเลย ให้ใส่ '!' ตามชื่อไฟล์ เช่น 'C:\a.jpg!' จะได้ไฟล์ตรงตามชื่อที่ตั้งเลย ไม่เช่นนั้นIUS จะสร้างนามสกุลไฟล์ที่เป็นตัวเลขที่ unique ให้

ยกตัวอย่างโค้ดดังนี้

```
CREATE TABLE PICTURE(PIC BLOB);
INSERT INTO PICTURE VALUES(FILETOBLOB('C:\TEST1.JPG','CLIENT'));
SELECT ROWID,* FROM PICTURE;
```

ได้ผลดังนี้

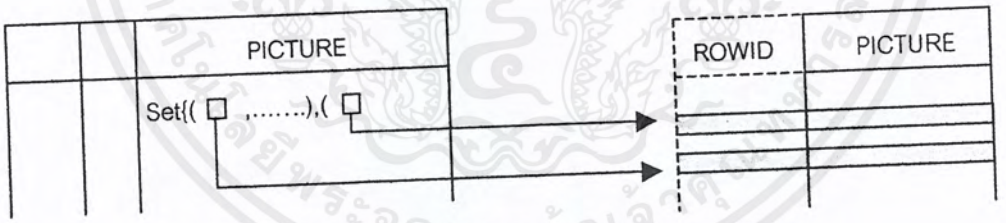
	rowid	pic
1	257	<blob; length = 62601>

รูปที่ 5-9 แสดงผลการใส่ไฟล์รูปลงตาราง

การเอารูปใน blob ออกมาที่ไฟล์ทำได้ดังนี้

```
SELECT LOTOFILE(PIC,'C:A.JPG!','CLIENT') FROM PICTURE WHERE ROWID=257;
```

สรุป หากแอตทริบิวต์ใดสามารถเปลี่ยนแปลงตามเวลาได้ ให้กำหนดชนิดเป็นเซ็ตของโรว์ไทป์ โโยกำหนดโรว์ไทป์ตามชนิดข้อมูลที่ต้องการเก็บ พร้อมทั้งวาลิดไทม์ด้วย แต่เนื่องจากเซ็ตไม่สามารถเก็บ blob ได้ ดังนั้นจึงต้องแยกเก็บ blob ไว้อีกตาราง แล้วใช้เซ็ตเก็บเฉพาะ ref (rowid) เพื่ออ้างถึงภาพเท่านั้น



รูปที่ 5-10 แสดงเทคนิคการอ้างถึง blob โดยเก็บ ref ไว้ในเซ็ต

5.6 สรุปการประยุกต์ฐานข้อมูล

สรุปขั้นตอนการสร้างฐานข้อมูลเชิงเวลา

1. สร้างโรว์ไทป์ที่เก็บข้อมูลชนิดที่ต้องการ และมีวาลิดไทม์อยู่ด้วย ดังนั้นจึงต้องรู้ก่อนว่ามีแอตทริบิวต์ใดบ้างที่เปลี่ยนตามเวลา เพื่อที่จะได้กำหนดชนิดข้อมูลในโรว์ไทป์ได้ถูกต้อง
2. สร้างตารางโดยกำหนดแอตทริบิวต์ที่เป็นเชิงเวลา ให้มีชนิดเป็นเซ็ตของโรว์ไทป์ตามชนิดข้อมูลนั้นๆ
3. Insert ข้อมูลลงตาราง
4. เขียน UDR โดยใช้ภาษา SPL เพื่อเข้าถึงข้อมูลในเซ็ตตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การวิเคราะห์และออกแบบระบบฐานข้อมูล

6.1 ระบบงานที่เลือกศึกษา

ระบบงานที่ได้เลือกนำมาเป็นกรณีศึกษา คือ ระบบงานโรงพยาบาล ซึ่งจะศึกษาในส่วนของการนำทฤษฎีเชิงเวลามาประยุกต์ใช้กับระบบงานจริง โดยเหตุที่เลือกนำระบบงานโรงพยาบาลมาเป็นกรณีศึกษานั้น เนื่องมาจากงานโรงพยาบาลมีข้อมูลที่เปลี่ยนแปลงเชิงเวลามากมาย นับตั้งแต่ประวัติผู้ป่วย ผู้ป่วยสามารถเปลี่ยนแปลงชื่อ นามสกุล ที่อยู่ได้ โดยยังคงเก็บข้อมูลเดิมของผู้ป่วยไว้ในฐานข้อมูล รวมทั้งผลการตรวจทางห้องปฏิบัติการ การวินิจฉัยโรค และประวัติการแพ้ยา ซึ่งเป็นข้อมูลที่สามารถเปลี่ยนแปลงในเชิงเวลาทั้งสิ้น และเหตุผลอีกประการหนึ่งก็คือ ในระบบงานโรงพยาบาลจะมีข้อมูลประเภทรูปภาพ ซึ่งถ้าเก็บลงฐานข้อมูล จะทำให้สามารถใช้ประโยชน์จากข้อมูลรูปภาพได้มีประสิทธิภาพกว่าแบบเดิม เช่น แพทย์หลายๆ ท่านสามารถช่วยกันในการวิเคราะห์ภาพเอ็กซ์เรย์ได้จากเครื่องไคลเอ็นท์ของตนเอง เป็นต้น ซึ่งถ้าเป็นการเก็บฟิล์มเอ็กซ์เรย์แบบเดิม จะทำให้การร่วมกันวิเคราะห์ทำได้ยากลำบาก

6.2 ขอบเขตและบุคคลที่ใช้ระบบงานที่จะพัฒนาขึ้น มีดังนี้

1) พนักงานโรงพยาบาลในแผนกทะเบียนผู้ป่วย จะใช้ระบบในส่วนของการป้อนข้อมูลผู้ป่วยใหม่ ค้นหาประวัติผู้ป่วย เพื่อส่งให้พยาบาลหน้าห้องตรวจ สำหรับการจัดคิวในการเข้ารับการรักษาของผู้ป่วย และสามารถแก้ไขข้อมูลของผู้ป่วยได้
2) แพทย์ จะสามารถค้นหาข้อมูลการวินิจฉัยครั้งก่อนๆ ของผู้ป่วย รวมทั้งผลการตรวจทางห้องปฏิบัติการ เพื่อประโยชน์ในการวินิจฉัย แพทย์สามารถบันทึกการวินิจฉัยครั้งใหม่ของผู้ป่วยลงในฐานข้อมูลได้ และสามารถสั่งตรงห้องปฏิบัติการ และสั่งยาได้อีกด้วย
3) เจ้าหน้าที่ห้องปฏิบัติการ จะสามารถค้นหารายการสั่งตรวจทางห้องปฏิบัติการที่ต้องตรวจสำหรับการวินิจฉัย (หรือการเข้าพักรักษาเป็นผู้ป่วยใน) ในแต่ละครั้งได้ และสามารถบันทึกผลการตรวจลงในฐานข้อมูลได้
4) เจ้าหน้าที่ฝ่ายผู้ป่วยใน จะสามารถบันทึกการเข้าพักเพื่อรับการรักษา และผลการตรวจประจำวันได้ รวมทั้งสามารถสั่งตรงทางห้องปฏิบัติการและสั่งยาสำหรับคนไข้ในได้
5) เจ้าหน้าที่ฝ่ายการเงิน จะสามารถออกใบเสร็จได้ โดยการใส่หมายเลขการวินิจฉัย (หรือหมายเลขการเข้าพักเพื่อรับการรักษา)

6.3 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาระบบนี้ แล้วนำไปใช้ คือ

1) การจัดเก็บข้อมูลผู้ป่วยเป็นระบบขึ้น และสาสมารถใช้ข้อมูลร่วมกันผ่านเครือข่ายได้

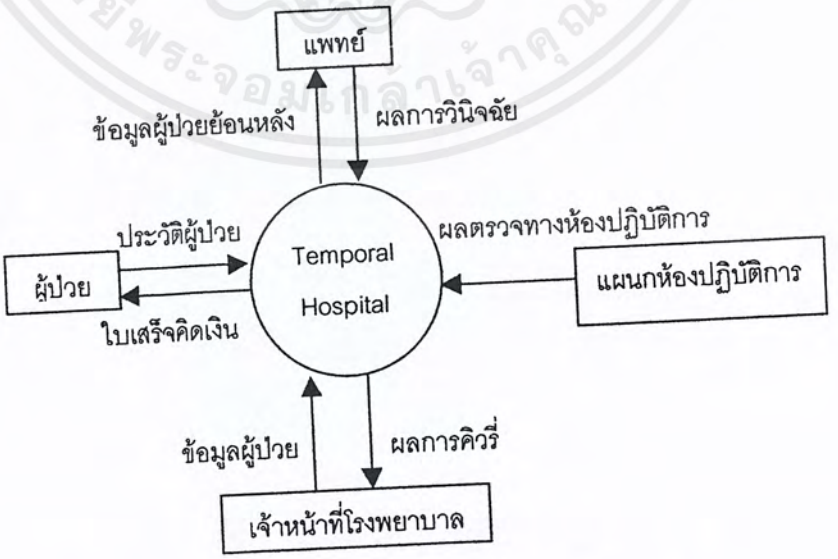
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เนื่องจากเป็นระบบฐานข้อมูลเชิงเวลา ทำให้สามารถตอบคำถามได้หลากหลายทั้งปัญหาเชิงเวลา และปัญหาปกติทั่วๆ ไป และเป็นประโยชน์ขึ้นโดยเฉพาะอย่างยิ่ง ในเรื่องของการวินิจฉัยโรคของแพทย์ แพทย์สามารถป้อนดูข้อมูลในอดีตของคนไข้ได้สะดวกขึ้น โดยไม่ต้องให้เจ้าหน้าที่เตรียมเอกสารให้ เพราะแพทย์สามารถค้นหาจากฐานข้อมูลได้ทันที
3) ความปลอดภัยในการเข้าถึงข้อมูล เนื่องจากการแยกหน้าจอการปฏิบัติของไคลเอ็นท์ในแต่ละเครื่อง ซึ่งแอพพลิเคชั่นจะขึ้นอยู่กับงานในฝ่ายของตน ทำให้แต่ละฝ่ายไม่สามารถเข้าถึงข้อมูลในส่วนที่ตนเองไม่เกี่ยวข้องได้
4) จัดการกับข้อมูลที่เปลี่ยนแปลงตามเวลาได้ โดยไม่ทำให้เกิดความยุ่งยากในการใช้งานและทำความเข้าใจ คือผู้ใช้สามารถใช้งานได้ง่ายและเข้าใจได้อย่างรวดเร็ว
5) เป็นคลังข้อมูลประวัติผู้ป่วย ทำให้แพทย์หลายๆท่านร่วมกันวินิจฉัยโรค โดยใช้ข้อมูลร่วมกันได้
6) ประโยชน์อื่นๆ เช่น ประหยัดกระดาษในการบันทึกผลการตรวจ ลดค่าใช้จ่ายของโรงพยาบาลในระยะยาว และประโยชน์อื่นๆทั้งหลายทั้งปวงของการนำระบบสารสนเทศมาใช้เป็นต้น

นอกจากประโยชน์ในแง่ของการนำไปใช้งานแล้ว ระบบนี้ยังมีคุณค่าในแง่ของการนำไปใช้เป็นระบบตัวอย่าง เพื่อที่จะออกแบบและพัฒนาระบบงานที่ใช้ฐานข้อมูลเชิงเวลา ต่อไป

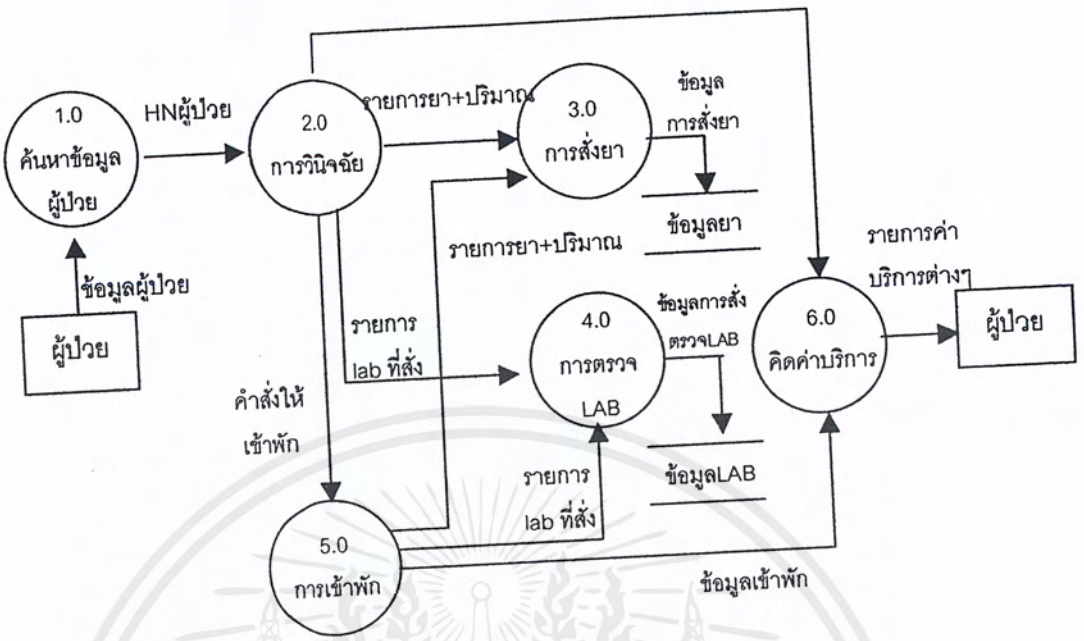
6.4 การวิเคราะห์ระบบงานโรงพยาบาล

จากที่ได้กล่าวถึงระบบอย่างคร่าวๆไปแล้วข้างต้น สามารถอธิบายระบบงานอย่างละเอียด ด้วยแผนภาพคาต้าโฟลว์ ได้ดังต่อไปนี้

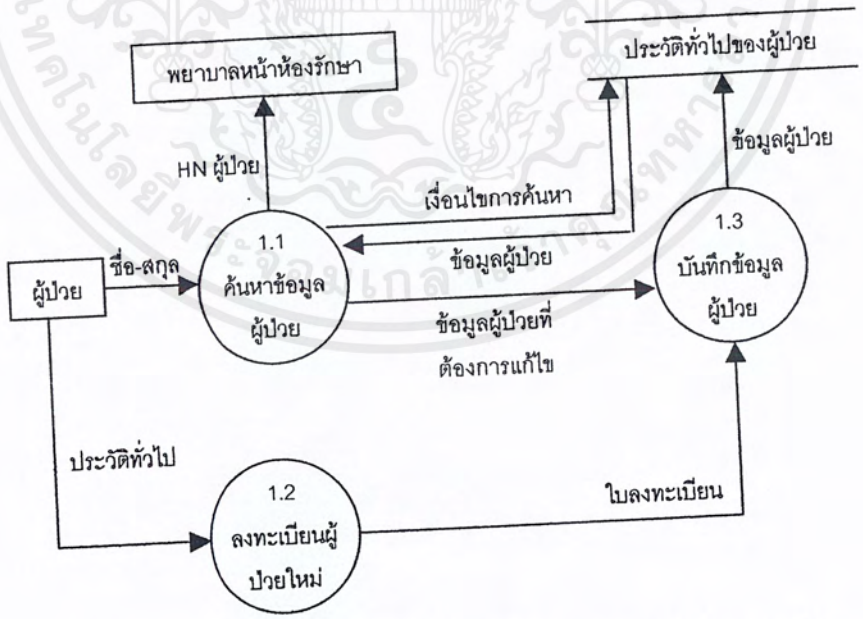


รูปที่ 6-1 แสดงแผนภาพของระบบงานโรงพยาบาลโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

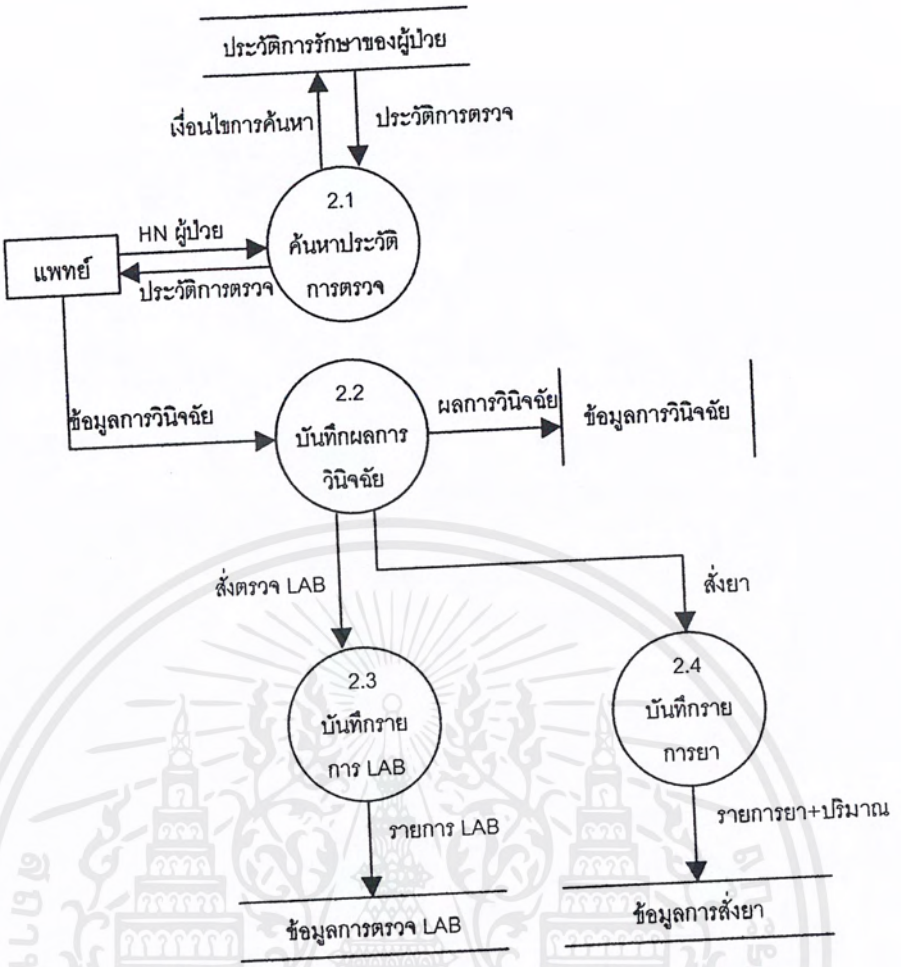


รูปที่ 6-2 แสดงแผนภาพดาต้าโฟลว์ของระบบงานโรงพยาบาล ระดับที่1

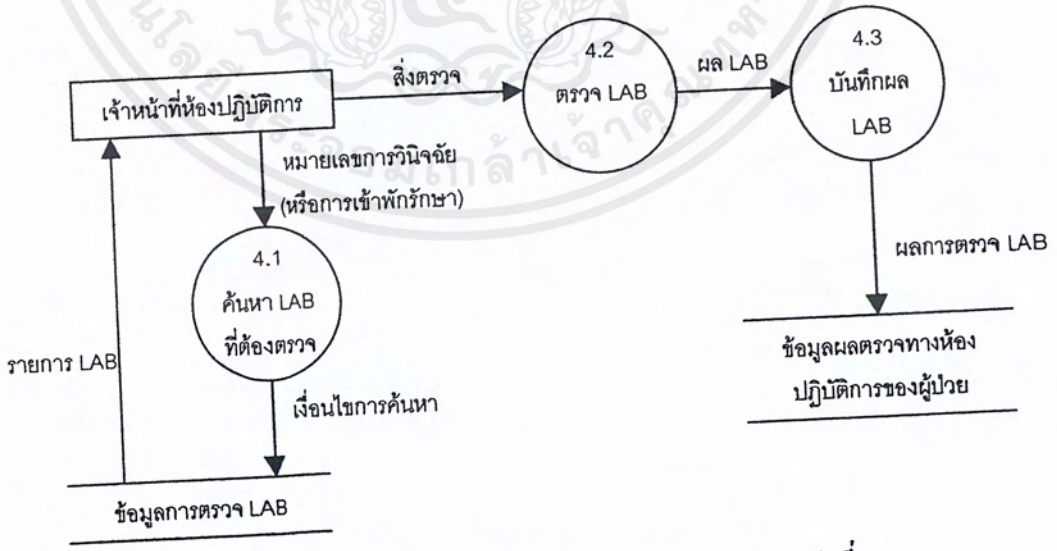


รูปที่ 6-3 แสดงแผนภาพดาต้าโฟลว์ของการค้นหาข้อมูลผู้ป่วย ระดับที่2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

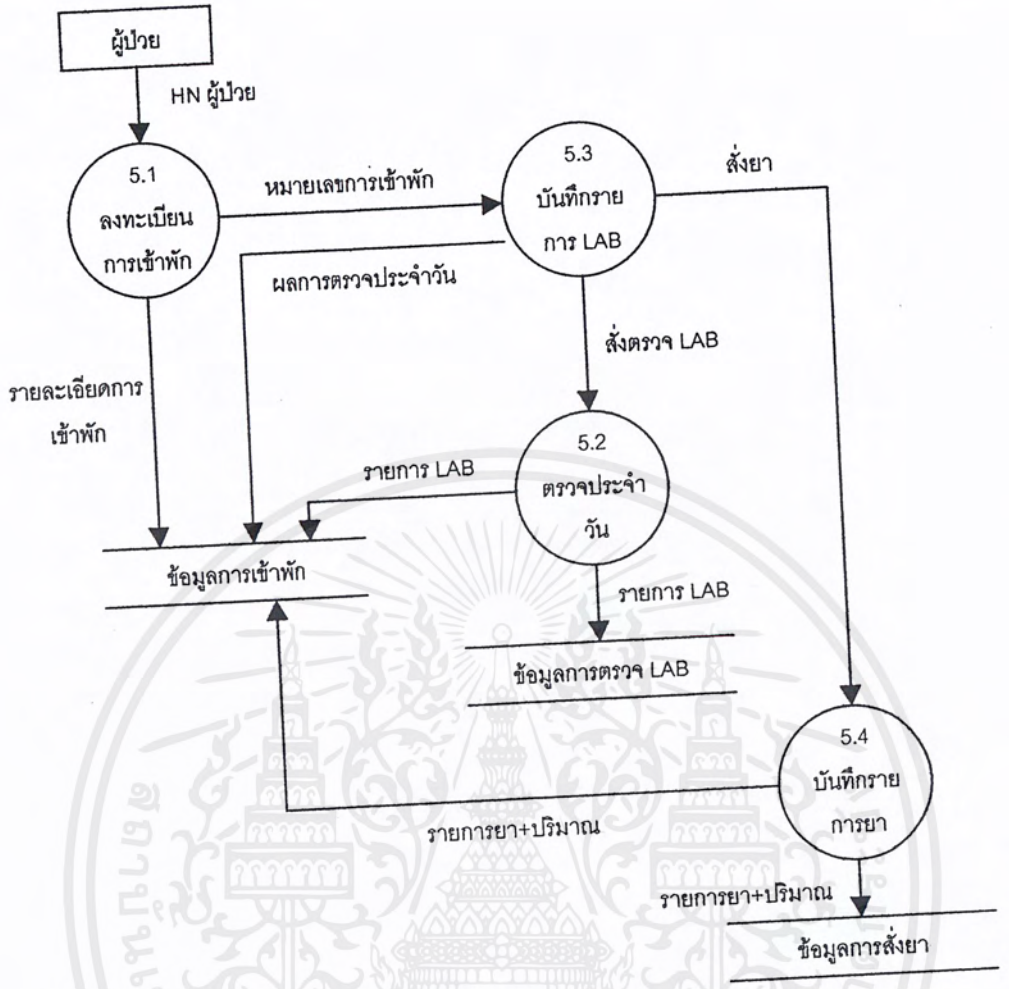


รูปที่ 6-4 แสดงแผนภาพดาต้าโฟลว์ของการวินิจฉัย ระดับที่2

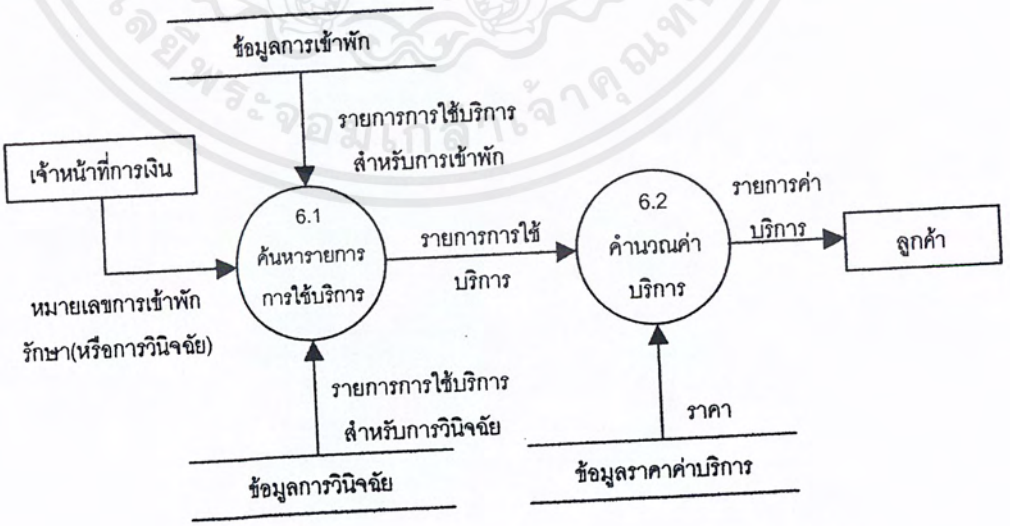


รูปที่ 6-5 แสดงแผนภาพดาต้าโฟลว์ของการตรวจLAB ระดับที่2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-6 แสดงแผนภาพดาต้าโฟลว์ของการเข้าพัก ระดับที่2



รูปที่ 6-7 แสดงแผนภาพดาต้าโฟลว์ของการคิดค่าบริการ ระดับที่2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 Data Dictionary ของระบบ มีดังนี้

Elementary Process Description
Process ID: 1.0
Process Name: **ค้นหาข้อมูลผู้ป่วย**
Description: เป็นการค้นหา HN ของผู้ป่วยจากชื่อ-นามสกุลของผู้ป่วย

Elementary Process Description
Process ID: 2.0
Process Name: **การวินิจฉัย**
Description: เป็นขั้นตอนการตรวจวินิจฉัยของแพทย์ โดยจะมีการค้นหาประวัติการตรวจรักษาของคนไข้ เพื่อช่วยในการวินิจฉัย มีการบันทึกผลการวินิจฉัยลงฐานข้อมูล สั่งตรวจทางห้องปฏิบัติการ และสั่งยา

Elementary Process Description
Process ID: 3.0
Process Name: **การสั่งยา**
Description: รับคำสั่งยา และบันทึกผลลงฐานข้อมูล เพื่อเก็บปริมาณยาที่ผู้ป่วยได้รับ และเพื่อการคิดเงินค่ายา

Elementary Process Description
Process ID: 4.0
Process Name: **การตรวจLAB**
Description: รับคำสั่งการตรวจLAB และบันทึกผลลงฐานข้อมูล เพื่อเก็บผลทางห้องปฏิบัติการของผู้ป่วยและเพื่อการคิดเงินค่าตรวจทางห้องปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Elementary Process Description
Process ID: 5.0
Process Name: **การเข้าพัก**
Description: เป็นขั้นตอนการลงทะเบียนเข้าพักรักษาและการวินิจฉัยในระหว่างที่พักอยู่ในโรงพยาบาล โดยจะมีการค้นหาประวัติการตรวจรักษาของคนไข้ เพื่อช่วยในการวินิจฉัย มีการบันทึกผลอาการของผู้ป่วย แผนการรักษาลงฐานข้อมูล สั่งตรวจทางห้องปฏิบัติการ และสั่งยา

Elementary Process Description
Process ID: 6.0
Process Name: **คิดค่าบริการ**
Description: เป็นการค้นหารายการการใช้บริการต่างๆในการรักษา ทั้งการวินิจฉัย(ผู้ป่วยนอก) และการเข้าพักรักษา(ผู้ป่วยใน) และราคาค่าบริการจากฐานข้อมูล โดยจะนำมาคำนวณหาค่าใช้จ่ายทั้งหมด

Elementary Process Description
Process ID: 1.1
Process Name: **ค้นหาข้อมูลผู้ป่วย**
Description: ค้นหาประวัติทั่วไปของผู้ป่วยตามเงื่อนไขที่ใส่เข้าไป

Elementary Process Description
Process ID: 1.2
Process Name: **ลงทะเบียนผู้ป่วยใหม่**
Description: กรณีที่ผู้ป่วยยังไม่เคยมารักษาเลย ต้องให้ผู้ป่วยกรอกประวัติเพื่อทำการลงทะเบียนเป็นผู้ป่วยของโรงพยาบาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Elementary Process Description
Process ID: 1.3
Process Name: **บันทึกข้อมูลผู้ป่วย**
Description: เจ้าหน้าที่โรงพยาบาลบันทึกข้อมูลผู้ป่วยลงฐานข้อมูล ทั้งบันทึกลงใหม่และการแก้ไขของเดิมด้วย

Elementary Process Description
Process ID: 2.1
Process Name: **ค้นหาประวัติการตรวจย้อนหลัง**
Description: แพทย์ค้นหาประวัติการตรวจย้อนหลัง เพื่อใช้ในการวินิจฉัย

Elementary Process Description
Process ID: 2.2
Process Name: **บันทึกผลการวินิจฉัย**
Description: แพทย์บันทึกผลการวินิจฉัย โดยเก็บเป็นรูปสแกนของเอกสารที่แพทย์จดบันทึกการวินิจฉัย

Elementary Process Description
Process ID: 2.3
Process Name: **บันทึกรายการLAB**
Description: รับรายการLAB ที่แพทย์สั่งตรวจ เพื่อเก็บลงฐานข้อมูล

Elementary Process Description
Process ID: 2.4
Process Name: **บันทึกรายการยา**
Description: รับรายการยาและปริมาณ ที่แพทย์สั่งให้กับผู้ป่วย เพื่อเก็บลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Elementary Process Description
Process ID: 4.1
Process Name: **ค้นหาLABที่ต้องตรวจ**
Description: เจ้าหน้าที่ห้องปฏิบัติการค้นหารายการLAB ที่ต้องตรวจ โดยใส่หมายเลขการวินิจฉัยหรือการเข้าพักรักษา

Elementary Process Description
Process ID: 4.2
Process Name: **การตรวจLAB**
Description: เจ้าหน้าที่ห้องปฏิบัติการตรวจLAB ตามรายการที่สั่งมา

Elementary Process Description
Process ID: 4.3
Process Name: **บันทึกผลLAB**
Description: บันทึกผลการตรวจลงฐานข้อมูลผู้ป่วย

Elementary Process Description
Process ID: 5.1
Process Name: **ลงทะเบียนการเข้าพัก**
Description: บันทึกการเข้าพักรักษาของผู้ป่วยลงฐานข้อมูลเพื่อเป็นประโยชน์ทางการแพทย์และการคิดค่ารักษา

Elementary Process Description
Process ID: 5.2
Process Name: **ตรวจประจำวัน**
Description: เป็นการตรวจประจำวันขณะรักษาอยู่ในโรงพยาบาล เพื่อติดตามผลการรักษาและวางแนวทางการรักษาที่ถูกต้องต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Elementary Process Description
Process ID: 5.3
Process Name: **บันทึกรายการLAB**
Description: เป็นการบันทึกรายการLAB ที่เกิดจากการเข้าพักรักษา เพื่อเป็นประโยชน์ทางการแพทย์และการคิดค่าตรวจ

Elementary Process Description
Process ID: 5.4
Process Name: **บันทึกรายการยา**
Description: เป็นการบันทึกรายการยาและปริมาณ ที่ได้รับในการเข้าพักรักษา เพื่อเป็นประโยชน์ทางการแพทย์และการคิดค่ายา

Elementary Process Description
Process ID: 6.1
Process Name: **ค้นหารายการการใช้บริการ**
Description: ค้นหารายการการใช้บริการทั้งหมดที่เกิดจากการวินิจฉัยผู้ป่วยนอก หรือการเข้าเป็นผู้ป่วยใน ทั้งแพทย์ การตรวจทางห้องปฏิบัติการ ยาที่ได้รับ ห้องพัก และ การบริการอื่นๆ

Elementary Process Description
Process ID: 6.2
Process Name: **คำนวณค่าบริการ**
Description: นำรายการการใช้บริการต่างๆ มาคิดตามราคาในฐานข้อมูล และรวมเป็นค่าใช้จ่ายทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O Description			
From	To	Data Flow Name	Description
ลูกค้า	1.0	ข้อมูลผู้ป่วย	ชื่อ-นามสกุล หรือข้อมูลอื่นๆที่ทำให้หาผู้ป่วยคนนั้นพบในฐานข้อมูล
1.0 1.1 แพทย์ ผู้ป่วย	2.0 พยาบาลหน้าห้องรักษา 2.1 5.1	HN ผู้ป่วย	หมายเลขประจำตัวของผู้ป่วย
2.0 ข้อมูลการวินิจฉัย	6.0 6.1	ข้อมูลการวินิจฉัย รายการการใช้บริการสำหรับการวินิจฉัย	ข้อมูลที่บอกรายการการใช้บริการต่างๆ ของการวินิจฉัย เพื่อคิดค่าบริการ
2.0 5.0 3.0 2.2 2.4 5.4 5.4 5.2	3.0 3.0 ข้อมูลยา 2.4 ข้อมูลการสั่งยา ข้อมูลการสั่งยา ข้อมูลการเข้าพัก 5.4	รายการยา+ปริมาณ ข้อมูลการสั่งยา สั่งยา	รายการยาและปริมาณที่แพทย์สั่งให้ผู้ป่วย
2.0 5.0 4.0 2.2 2.3 ข้อมูลการตรวจ LAB 5.2 5.3 5.3	4.0 4.0 ข้อมูลLAB 2.3 ข้อมูลการตรวจ LAB เจ้าหน้าที่ห้องปฏิบัติการ 5.3 ข้อมูลการตรวจ LAB ข้อมูลการเข้าพัก	รายการLABที่สั่งตรวจ ข้อมูลการสั่งตรวจLAB สั่งตรวจLAB รายการLAB	รายการLAB ที่แพทย์สั่งตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

From	To	Data Flow Name	Description
2.0	5.0	คำสั่งให้เข้าพัก	คำสั่งแพทย์ให้เข้ารักษาในโรงพยาบาล
5.0 ข้อมูลการเข้าพัก	6.0 6.1	ข้อมูลการเข้าพัก รายการการใช้บริการสำหรับการเข้าพัก	ข้อมูลที่บอกรายการการใช้บริการต่างๆ ของการเข้าพักรักษา เพื่อคิดค่าบริการ
6.0	ลูกค้า	รายการค่าบริการต่างๆ	รายการการใช้บริการ และค่าใช้จ่ายในการรักษา
ผู้ป่วย	1.1	ชื่อ-สกุล	ชื่อ-นามสกุลผู้ป่วย
ผู้ป่วย	1.2	ประวัติทั่วไป	ประวัติของผู้ป่วยเพื่อทำการลงทะเบียนเป็นคนไข้ใหม่
1.1 2.1	ประวัติทั่วไปของผู้ป่วย ประวัติการรักษาผู้ป่วย	เงื่อนไขการค้นหา	ข้อมูลผู้ป่วยเพื่อใช้ค้นการประวัติของผู้ป่วยคนนั้น
1.1	1.3	ข้อมูลผู้ป่วยที่ต้องการแก้ไข	ข้อมูลผู้ป่วยที่ต้องการแก้ไขประวัติ
ประวัติทั่วไปของผู้ป่วย 1.3	1.1 ประวัติทั่วไปของผู้ป่วย	ข้อมูลผู้ป่วย	ประวัติทั่วไปของผู้ป่วย
1.2	1.3	ใบลงทะเบียน	ประวัติผู้ป่วย
ประวัติการรักษาผู้ป่วย 2.1	2.1 แพทย์	ประวัติการตรวจ	ผลการวินิจฉัย ผลLAB ยาที่เคยได้รับของผู้ป่วย
แพทย์	2.2	ข้อมูลการวินิจฉัย	การวินิจฉัยของแพทย์
2.2	ข้อมูลการวินิจฉัย	ผลการวินิจฉัย	รูปแบบการวินิจฉัยของแพทย์
4.1	ข้อมูลการตรวจ LAB	เงื่อนไขการค้นหา	เงื่อนไขในการค้นหารายการLAB จากฐานข้อมูล
เจ้าหน้าที่ห้องปฏิบัติการ เจ้าหน้าที่การเงิน	4.1 6.1	หมายเลขของการวินิจฉัย(หรือการเข้าพัก)	หมายเลขของการวินิจฉัย(หรือการเข้าพัก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

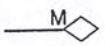
From	To	Data Flow Name	Description
เจ้าหน้าที่ห้องปฏิบัติการ	4.2	สิ่งตรวจ	สิ่งที่จะใช้ในการตรวจทางห้องปฏิบัติการ เช่น เลือด เป็นต้น
4.2 4.3	4.3 ข้อมูลผลตรวจทางห้องปฏิบัติการของผู้ป่วย	ผลLAB ผลตรวจLAB	ผลตรวจทางห้องปฏิบัติการ
5.1	ข้อมูลการเข้าพัก	รายละเอียดการเข้าพัก	รายละเอียดการเข้าพักต่างๆ เช่น ห้องพัก ระดับการให้การดูแล (Package) และอื่นๆ
5.2	ข้อมูลการเข้าพัก	ผลการตรวจประจำวัน	อุณหภูมิ ความดัน อัตราการเต้นของหัวใจ ของผู้ป่วย รวมทั้งแผนการรักษาและการวินิจฉัยด้วย
5.1	5.2	หมายเลขการเข้าพัก	หมายเลขการเข้าพัก
6.1	6.2	รายการการใช้บริการ	รายการต่างๆในการรักษา เช่น รายการการตรวจทางห้องปฏิบัติการ รายการยา แพทย์ที่ทำการรักษาปเน ต้น
ข้อมูลราคาค่าบริการ	6.2	ราคา	ราคาค่าบริการและเวชภัณฑ์ต่างๆ
6.2	ลูกค้า	รายการค่าบริการ	รายการการใช้บริการต่างๆในการรักษาและค่าใช้จ่ายต่างๆในการรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

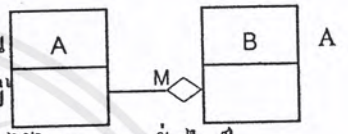
6.6 การออกแบบระบบงานโรงพยาบาลในส่วนฐานข้อมูล

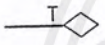
สัญญลักษณ์และหลักการในการออกแบบ : ในการออกแบบฐานข้อมูลเชิงเวลาสำหรับงานโรงพยาบาลนี้ ได้นำเสนอสัญญลักษณ์ใหม่ที่ใช้ในการออกแบบฐานข้อมูล เพื่อประโยชน์ในเรื่องของความเข้าใจในขั้นตอนการออกแบบ แต่สัญญลักษณ์ที่ใช้ส่วนใหญ่นั้นยังคงเป็น ER-Diagram แบบ case-tool แต่มีการปรับปรุงให้เหมาะสมกับการออกแบบอีกเล็กน้อย ดังนี้

: เอนทิตี้ สามารถแปลง 1 เอนทิตี้เป็นตารางได้ 1 ตาราง โดยแต่ละเอนทิตี้ในแผนภาพไม่จำเป็นต้องมี primary key เพราะจะมี ref (rowid) เพื่อที่จะใช้อ้างถึงในแต่ละโรว์ในตารางอยู่แล้ว ซึ่งในการอ้างอิงทั้งหมดจะใช้ ref ในการอ้างอิง

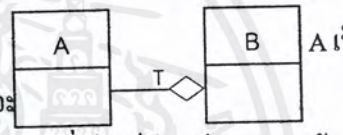


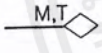
: ความสัมพันธ์'เป็นของ'แบบMultivalue เช่น A เป็นของ B แบบ Multivalue โดย A สามารถมีค่า ในการใช้งาน ให้ B เก็บ ref ที่อ้างไปถึง A ไว้ในคอลเลคชั่นไทป์



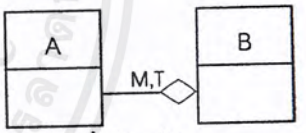


: ความสัมพันธ์เป็นของแบบเชิงเวลา เช่น A เป็นของ B แบบเชิงเวลา โดย ณ เวลาหนึ่ง B จะไปถึง A 1 ค่า โดยเก็บ ref ของ A ลงในคอลเลคชั่นไทป์ โดยมีเวลาเกาะกับ ref แต่ละตัว ซึ่งเวลาต้องไม่ซ้อนทับกัน



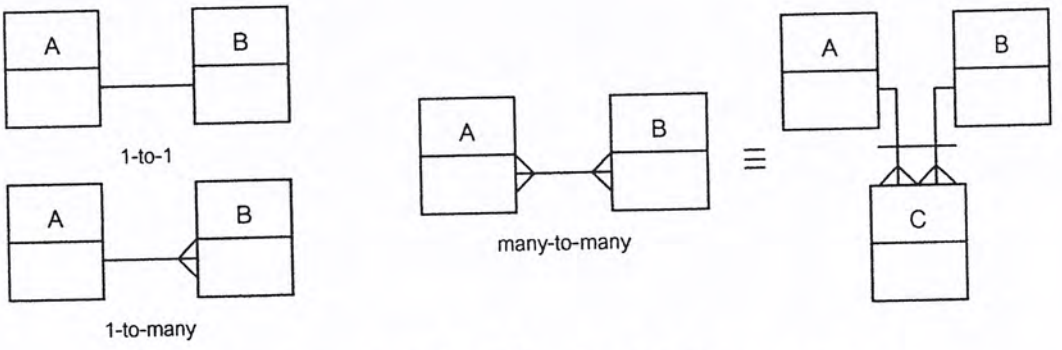


: ความสัมพันธ์เป็นของแบบ Multivalue เชิงเวลา เช่น A เป็นของ B แบบ Multivalue เชิงเวลา โดยณ เวลาหนึ่ง B จะมี ref ไปถึง A ได้หลายค่า โดยเก็บ ref ของ A ลงในคอลเลคชั่นไทป์ โดยมีเวลาเกาะกับ ref แต่ละตัว ซึ่งเวลาอาจซ้อนทับกันได้



หลักการออกแบบมีขั้นตอนดังนี้

1. ให้พิจารณาความสัมพันธ์ของแต่ละเอนทิตี้ โดยใช้หลักการออกแบบฐานข้อมูลรีเลชันแนลแบบเดิม โดยใช้ ER-Diagram แบบ case-tool ดังนี้

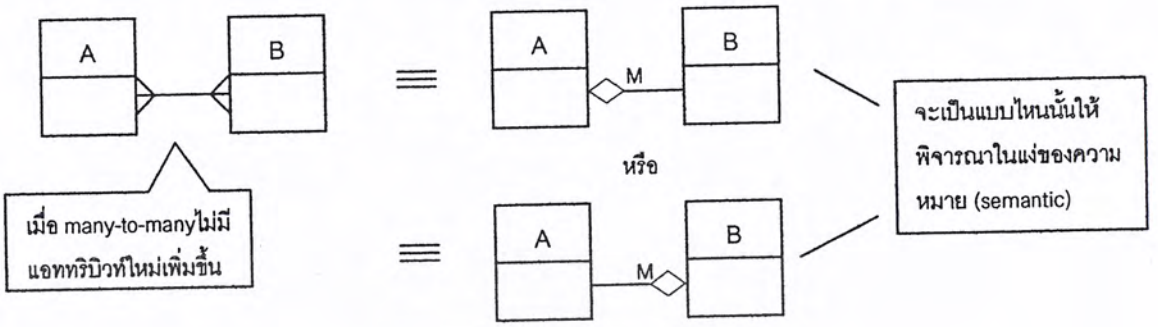


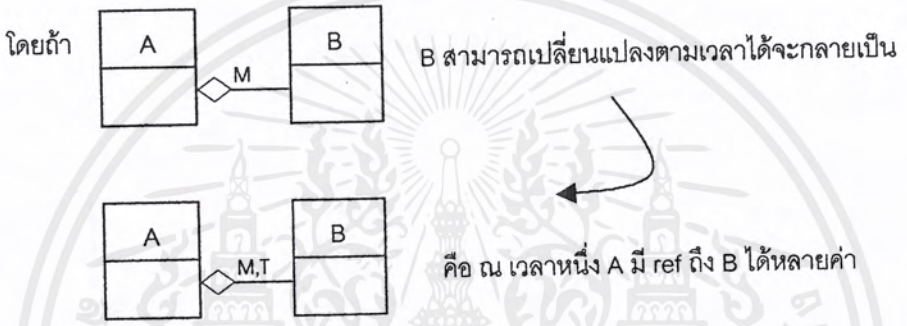
รูปที่ 6-8 แสดงความสัมพันธ์เอนทิตี้แบบต่างๆ โดยใช้ ER แบบ case-tool

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ต่างจากแบบเดิมที่ใช้ ref แทน primary key และการอ้างถึงแบบ foreign key ทั้งหมด

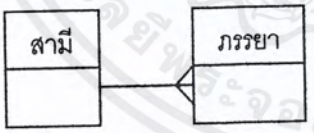
2. พิจารณาความสัมพันธ์แบบ many-to-many ที่ไม่มีแอททริบิวท์ใหม่เพิ่มขึ้นจากความสัมพันธ์ สามารถแปลงเป็นความสัมพันธ์เป็นของ แบบ Multivalue ได้ ดังรูปที่ 6-9



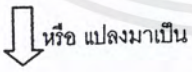


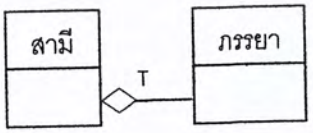
รูปที่ 6-9 แสดงการแปลงความสัมพันธ์จาก many-to-many ไปเป็นความสัมพันธ์เป็นของ แบบ Multivalue

3. พิจารณาความสัมพันธ์แบบ 1-to-many ที่เป็น many เพราะเวลา เช่น กำหนดให้สามีมีภรรยาได้ทีละ 1 คน แสดงว่าจริงๆแล้วสามีมีภรรยาได้หลายคน(รวมๆกันณ เวลาต่างๆ) สามารถ implement ได้ 2 รูปแบบ ดังนี้



โดยเป็น 1-to-many เพราะเวลา (ถ้า ณ เวลาหนึ่งจะเป็น 1-to-1) ให้เอา ref ของสามีไปเกาะฝั่งภรรยาแล้วเกาะเวลากับทั้งโรว์





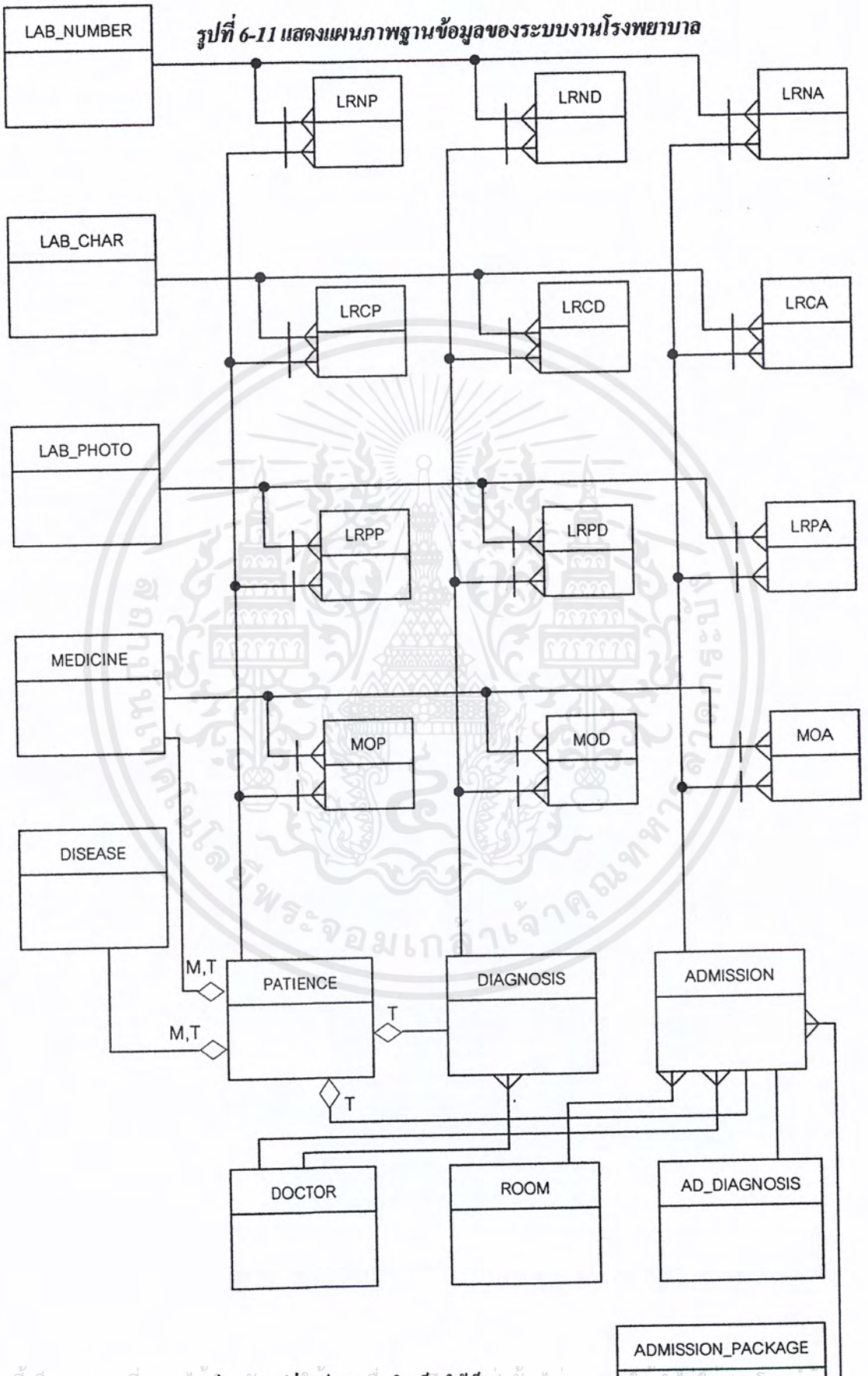
Implement โดยเอา ref ภรรยามาเกาะกับเวลาแล้วเก็บลงคอลลเลคชั่นไทป์ที่ฝั่งสามี

รูปที่ 6-10 แสดงการแปลงความสัมพันธ์จาก 1-to-many ไปเป็นความสัมพันธ์เป็นของแบบเชิงเวลา

4. พิจารณาแอททริบิวท์อื่นๆว่าเปลี่ยนแปลงตามเวลาได้หรือไม่ ถ้าเปลี่ยนแปลงได้ให้เก็บเป็นแบบเชิงเวลา ดังที่ได้กล่าวไปในบทที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหลักการข้างต้น สามารถนำมาใช้ออกแบบฐานข้อมูลของระบบงานโรงพยาบาลได้ดังรูปที่ 6-11



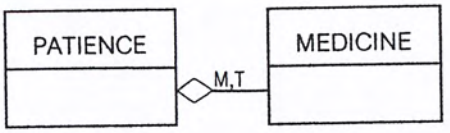
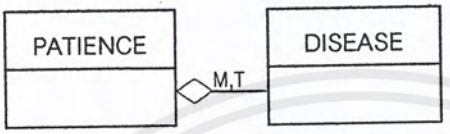
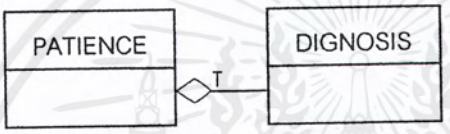
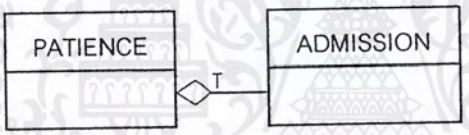
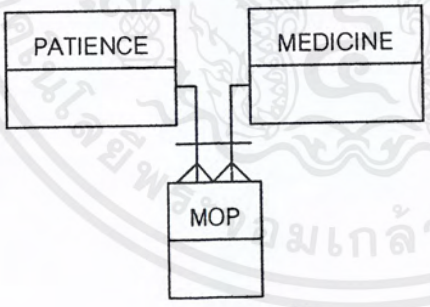
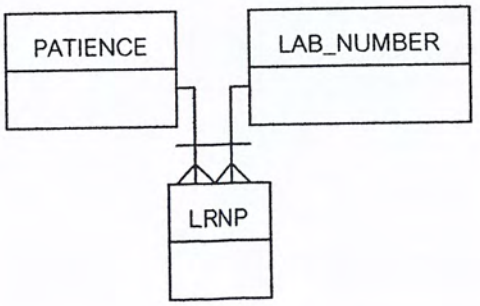
รูปที่ 6-11 แสดงแผนภาพฐานข้อมูลของระบบงานโรงพยาบาล

หมายเหตุ ทุกตารางที่มีคอลัมน์ที่จะเก็บblobลงในเซ็ต ให้เก็บเป็น ref ไปยังตาราง PHOTO แทน

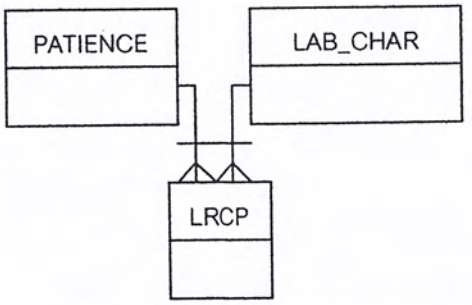
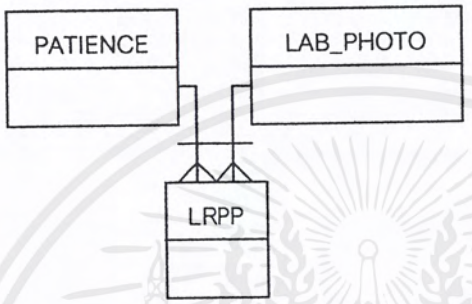
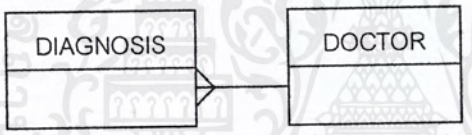
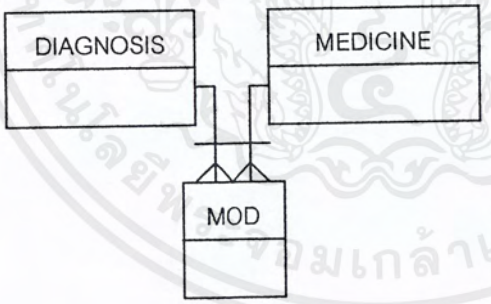
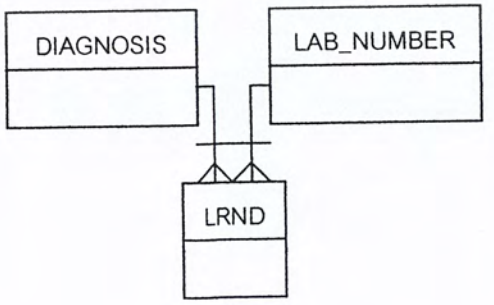
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายความหมายของความสัมพันธ์ในแต่ละเอนทิตี้ ได้ดังตารางที่ 6-1

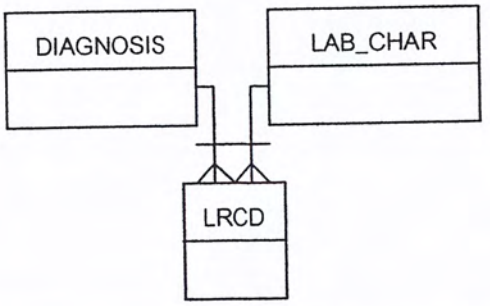
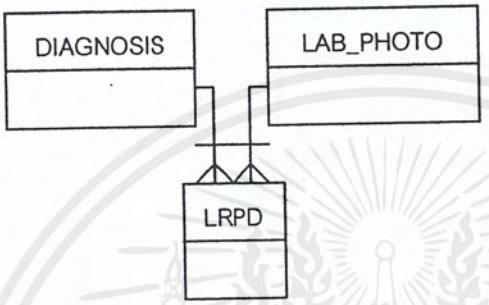
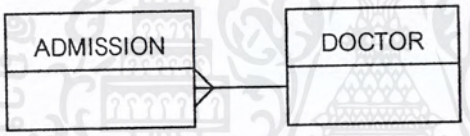
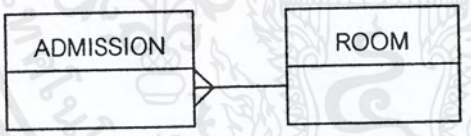
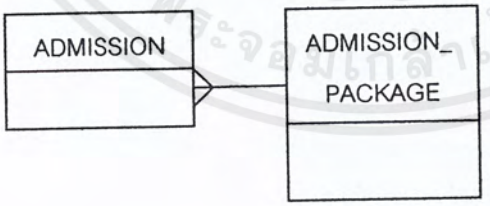
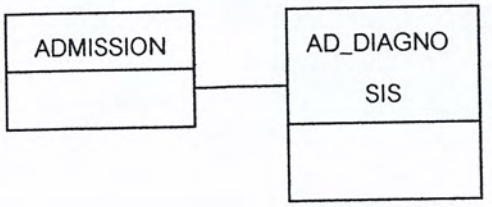
ตารางที่ 6-1 อธิบายความสัมพันธ์ของแต่ละเอนทิตี้

ความสัมพันธ์	อธิบาย
	ผู้ป่วยสามารถแพ้ยาได้หลายชนิดณ เวลาหนึ่ง และยาที่แพ้สามารถเปลี่ยนแปลงได้ตามเวลา เช่น ยาบางชนิดอาจไม่ได้แพ้อีกต่อไปแล้ว เป็นต้น
	ผู้ป่วยสามารถมีโรคประจำตัวได้หลายโรคพร้อมกัน และเปลี่ยนแปลงได้ตามเวลา
	ผู้ป่วยมีการวินิจฉัยหลายครั้งตามเวลา
	ผู้ป่วยมีการเข้าพักรักษาหลายครั้งตามเวลา
	ผู้ป่วยได้รับยาหลายชนิด ยาชนิดหนึ่งถูกสั่งให้ผู้ป่วยหลายคน
	ผู้ป่วยมีผลการตรวจLAB ชนิดตัวเลขได้หลายประเภท และมีผู้เคยตรวจLAB แต่ละตัว หลายคน

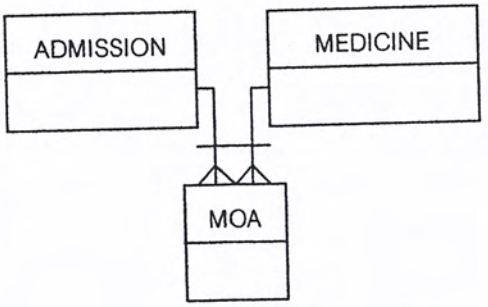
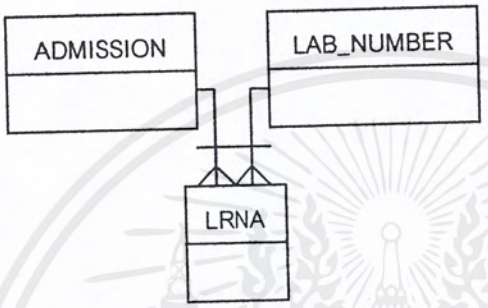
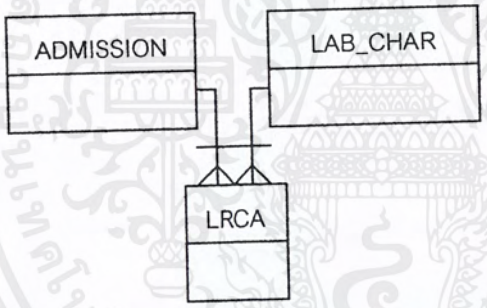
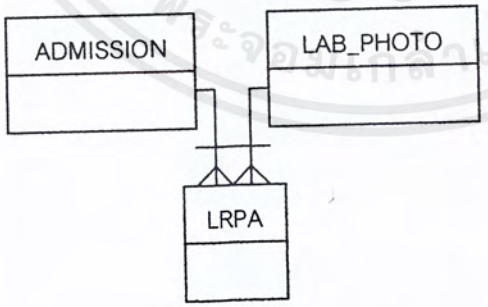
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์	อธิบาย
	ผู้ป่วยมีผลการตรวจLAB ชนิดตัวอักษร ได้หลายประเภท และมีผู้เคยตรวจLAB แต่ละตัว หลายคน
	ผู้ป่วยมีผลการตรวจLAB ชนิดตัวรูปภาพ ได้หลายประเภท และมีผู้เคยตรวจLAB แต่ละตัว หลายคน
	แพทย์สามารถวินิจฉัยโรคได้หลายครั้ง แต่ละครั้งของการวินิจฉัยมีแพทย์ได้เพียงท่านเดียว
	ในการวินิจฉัยแต่ละครั้ง สามารถสั่งยาได้หลายชนิด โดยแต่ละชนิดถูกสั่งได้จากหลายการวินิจฉัย
	การวินิจฉัยแต่ละครั้ง มีการตรวจLAB ชนิดตัวเลขได้หลายตัว โดยแต่ละLAB ถูกสั่งตรวจได้จากหลายการวินิจฉัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์	อธิบาย
	การวินิจฉัยแต่ละครั้ง มีการตรวจLAB ชนิดตัวอักษรได้หลายตัว โดยแต่ละLAB ถูกสั่งตรวจได้จากหลายการวินิจฉัย
	การวินิจฉัยแต่ละครั้ง มีการตรวจLAB ชนิดรูปภาพได้หลายตัว โดยแต่ละLAB ถูกสั่งตรวจได้จากหลายการวินิจฉัย
	แพทย์สามารถเป็นเจ้าของไข้ในการเข้าพักรักษาได้หลายครั้ง การเข้าพักรักษาครั้งหนึ่ง มีแพทย์ดูแลได้เพียงท่านเดียว
	การเข้าพักรักษาแต่ละครั้งจะอยู่ที่ห้องพักห้องเดียว เปลี่ยนห้องไม่ได้ และห้องพักหนึ่งสามารถอยู่ได้หลายคน
	การเข้าพักรักษาแต่ละครั้ง มีแพคเกจในการดูแลรักษาได้ 1 แพคเกจ
	การตรวจประจำวันเป็นของการเข้าพักรักษาแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์	อธิบาย
	การเข้าพักรักษาแต่ละครั้งสามารถสั่งยาได้หลายชนิด โดยยาแต่ละชนิดถูกสั่งได้จากหลายการเข้าพักรักษา
	การเข้าพักรักษาแต่ละครั้ง มีการตรวจ LAB ชนิดตัวเลขได้หลายตัว โดยแต่ละ LAB ถูกสั่งตรวจได้จากหลายการเข้าพักรักษา
	การเข้าพักรักษาแต่ละครั้ง มีการตรวจ LAB ชนิดตัวอักษรได้หลายตัว โดยแต่ละ LAB ถูกสั่งตรวจได้จากหลายการเข้าพักรักษา
	การเข้าพักรักษาแต่ละครั้ง มีการตรวจ LAB ชนิดรูปภาพได้หลายตัว โดยแต่ละ LAB ถูกสั่งตรวจได้จากหลายการเข้าพักรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การสร้างฐานข้อมูลและแอพพลิเคชั่น

7.1 การสร้างฐานข้อมูล

ตารางของระบบงานโรงพยาบาล มีรายละเอียด ดังนี้

1. ตาราง MEDICINE (ยา): เก็บข้อมูลเกี่ยวกับยา ชื่อยารวมทั้งราคายาด้วย

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
MED_CODE(P.K.)	SERIAL	หมายเลขประจำยา
COMM_NAME	SET OF ROW_CHAR	ชื่อทางการค้า
SCI_NAME	CHAR(50)	ชื่อทางวิทยาศาสตร์
USE	CHAR(20)	การใช้ กิน ทา หรือดม เป็นต้น
MFT_NAME	CHAR(50)	ชื่อโรงงานผลิต
PRICE	SET OF ROW_INT	ราคายาต่อหน่วย

ตารางที่ 7-1 แสดงคอลัมน์ในตาราง MEDICINE

2. ตาราง DISEASE (โรค): เก็บข้อมูลเกี่ยวกับโรค ชื่อโรค ชนิดเชื้อโรค

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DIS_CODE(P.K.)	SERIAL	หมายเลขประจำโรค
GENERAL_NAME	SET OF ROW_CHAR	ชื่อทั่วไป
SCI_NAME	CHAR(50)	ชื่อโรคทางวิทยาศาสตร์
DIS_TYPE	CHAR(30)	ชนิดของเชื้อโรค
DIS_NAME	CHAR(50)	ชื่อของตัวเชื้อโรค

ตารางที่ 7-2 แสดงคอลัมน์ในตาราง DISEASE

3. ตาราง DIAGNOSIS (การวินิจฉัย): เก็บรายละเอียดของการวินิจฉัยแต่ละครั้ง สำหรับผลการวินิจฉัยที่เป็นลายมือแพทย์นั้นถูกเก็บเป็นไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DIA_CODE(P.K.)	SERIAL	หมายเลขการวินิจฉัย
WEIGHT	INT	น้ำหนักผู้ป่วย
HEIGHT	INT	ความสูงผู้ป่วย
DOCTOR	INT	Ref ไปที่ตาราง DOCTOR
DIAGNOSIS_RECORD	BLOB	รูปผลการวินิจฉัยของแพทย์

ตารางที่ 7-3 แสดงคอลัมน์ในตาราง DIAGNOSIS

4. ตาราง DOCTOR (แพทย์): เก็บข้อมูลเกี่ยวกับแพทย์ ชื่อแพทย์ความชำนาญระดับขั้น รวมทั้งค่าวิชาชีพด้วย

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DOC_CODE(P.K.)	SERIAL	หมายเลขแพทย์
NAME	SET OF ROW_CHAR	ชื่อแพทย์ สามารถเปลี่ยนแปลงได้
SURNAME	SET OF ROW_CHAR	นามสกุลแพทย์ สามารถเปลี่ยนแปลงได้
SEX	CHAR	เพศ
LEVEL	SET OF ROW_CHAR	ระดับ หรือตำแหน่ง
SKILL	CHAR(40)	ความชำนาญเฉพาะทาง
CHARGE	SET OF ROW_INT	ค่าวิชาชีพ

ตารางที่ 7-4 แสดงคอลัมน์ในตาราง DOCTOR

5. ตาราง PATIENCE (ผู้ป่วย): เก็บประวัติทั่วไปของผู้ป่วย โรคประจำตัว ยาที่แพ้ และมี ref อ้างถึงการวินิจฉัยและการเข้าพักรักษาของผู้ป่วยอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
HN(P.K.)	SERIAL	หมายเลขผู้ป่วย
PICTURE	SET OF ROW_INT	เซ็ตของ ref ที่อ้างไปที่ตาราง PHOTO
NAME	SET OF ROW_CHAR	ชื่อผู้ป่วย สามารถเปลี่ยนแปลงได้
SURNAME	SET OF ROW_CHAR	นามสกุลผู้ป่วย สามารถเปลี่ยนแปลงได้
BIRTHDAY	DATE	วันเกิด
SEX	CHAR	เพศ
ID_NUM	CHAR(13)	หมายเลขบัตรประชาชน
BLOOD_GROUP	CHAR(2)	กรุ๊ปเลือด
ADDRESS	SET OF ADDRESS	ที่อยู่
TEL_NUM	SET OF ROW_CHAR_S	เบอร์โทรศัพท์ มีได้ทีละหลายเบอร์ (Multivalues)
CHRONIC_DIS	SET OF ROW_INT	โรคประจำตัว เก็บของเซ็ต ref ที่อ้างไปตารางโรค
ALLERGY_MED	SET OF ROW_INT	ยาที่แพ้เก็บเซ็ตของ ref ที่อ้างไปตารางยา
DIAGNOSIS	SET OF ROW_INT	เซ็ตของ ref ไปที่การวินิจัยของผู้ป่วย
ADMISSION	SET OF ROW_INT	เซ็ตของ ref ไปที่การเข้าพักรักษาของผู้ป่วย

ตารางที่ 7-5 แสดงคอลัมน์ในตาราง PATIENCE

6. ตาราง LRNP (ผลLABชนิดตัวเลขที่สัมพันธ์กับผู้ป่วย): เก็บผลLAB ชนิดตัวเลขของผู้ป่วยแต่ละคน เพื่อประโยชน์ในการค้นหา

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
PATIENCE(P.K.)	INT	Ref ไปที่ผู้ป่วย
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_NUMBER
LAB_RESULT	SET OF ROW_INT	เซ็ตของผลLAB ชนิดตัวเลข

ตารางที่ 7-6 แสดงคอลัมน์ในตาราง LRNP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ตาราง LRCP (ผลLABชนิดตัวอักษรที่สัมพันธ์กับผู้ป่วย): เก็บผลLAB ชนิดตัวอักษรของผู้ป่วยแต่ละคน เพื่อประโยชน์ในการค้นหา

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
PATIENCE(P.K.)	INT	Ref ไปที่ผู้ป่วย
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_CHAR
LAB_RESULT	SET OF ROW_CHAR	เซ็ตของผลLAB ชนิดตัวอักษร

ตารางที่ 7-7 แสดงคอลัมน์ในตาราง LRCP

8. ตาราง LRPP (ผลLABชนิดรูปภาพที่สัมพันธ์กับผู้ป่วย): เก็บผลLAB ชนิดรูปภาพของผู้ป่วยแต่ละคน เพื่อประโยชน์ในการค้นหา

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
PATIENCE(P.K.)	INT	Ref ไปที่ผู้ป่วย
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_PHOTO
LAB_RESULT	SET OF ROW_INT	เซ็ตของ ref ไปที่ตาราง PHOTO

ตารางที่ 7-8 แสดงคอลัมน์ในตาราง LRPP

9. ตาราง LRND (ผลLABชนิดตัวเลขที่สัมพันธ์กับการวินิจฉัย): เก็บผลLAB ชนิดตัวเลขที่สั่งมาจากการวินิจฉัยในแต่ละครั้ง

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DIAGNOSIS(P.K.)	INT	Ref ไปที่การวินิจฉัย
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_NUMBER
LAB_RESULT	INT	ผลLAB ชนิดตัวเลข

ตารางที่ 7-9 แสดงคอลัมน์ในตาราง LRND

10. ตาราง LRCD (ผลLABชนิดตัวอักษรที่สัมพันธ์กับการวินิจฉัย): เก็บผลLAB ชนิดตัวอักษรที่สั่งมาจากการวินิจฉัยในแต่ละครั้ง

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DIAGNOSIS(P.K.)	INT	Ref ไปที่การวินิจฉัย
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_CHAR
LAB_RESULT	CHAR(50)	ผลLAB ชนิดตัวอักษร

ตารางที่ 7-10 แสดงคอลัมน์ในตาราง LRCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ตาราง LRPD (ผลLABชนิดรูปภาพที่สัมพันธ์กับการวินิจฉัย): เก็บผลLAB ชนิดรูปภาพที่สั่งมาจากการวินิจฉัยในแต่ละครั้ง

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DIAGNOSIS(P.K.)	INT	Ref ไปที่การวินิจฉัย
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_PHOTO
LAB_RESULT	INT	Ref ไปที่ตาราง PHOTO

ตารางที่ 7-11 แสดงคอลัมน์ในตาราง LRPD

12. ตาราง LRNA (ผลLABชนิดตัวเลขที่สัมพันธ์กับการเข้าพักรักษา): เก็บผลLAB ชนิดตัวเลขที่สั่งตรวจมาจากการเข้าพักรักษาในครั้งหนึ่งๆ

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
ADMISSION(P.K.)	INT	Ref ไปที่การเข้าพักรักษา
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_NUMBER
LAB_RESULT	SET OF ROW_INT	เซ็ตของผลLAB ชนิดตัวเลข

ตารางที่ 7-12 แสดงคอลัมน์ในตาราง LRNA

13. ตาราง LRCA (ผลLABชนิดตัวอักษรที่สัมพันธ์กับการเข้าพักรักษา): เก็บผลLAB ชนิดตัวอักษรที่สั่งตรวจมาจากการเข้าพักรักษาในครั้งหนึ่งๆ

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
ADMISSION(P.K.)	INT	Ref ไปที่การเข้าพักรักษา
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_CHAR
LAB_RESULT	SET OF ROW_CHAR	เซ็ตของผลLAB ชนิดตัวอักษร

ตารางที่ 7-13 แสดงคอลัมน์ในตาราง LRCA

14. ตาราง LRPA (ผลLABชนิดรูปภาพที่สัมพันธ์กับการเข้าพักรักษา): เก็บผลLAB ชนิดรูปภาพที่สั่งตรวจมาจากการเข้าพักรักษาในครั้งหนึ่งๆ

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
ADMISSION(P.K.)	INT	Ref ไปที่การเข้าพักรักษา
LAB(P.K.)	INT	Ref ไปที่ตาราง LAB_PHOTO
LAB_RESULT	SET OF ROW_INT	เซ็ตของ ref ไปที่ตาราง PHOTO

ตารางที่ 7-14 แสดงคอลัมน์ในตาราง LRPA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. ตาราง ROOM (ห้อง): เก็บราคาห้องพักตามหมายเลขห้อง

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
ROOM_ID(P.K.)	SERIAL	หมายเลขห้อง
PRICE	SET OF ROW_INT	ราคาห้อง เปลี่ยนแปลงตามเวลาได้

ตารางที่ 7-15 แสดงคอลัมน์ในตาราง ROOM

16. ตาราง MOP (ปริมาณยาที่สั่งที่สัมพันธ์กับผู้ป่วย): เก็บปริมาณยาที่ผู้ป่วยได้รับในช่วงเวลาต่างๆ เพื่อใช้ประโยชน์ในการคิวรี่

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
PATIENCE(P.K.)	INT	Ref ไปที่ผู้ป่วย
MEDICINE(P.K.)	INT	Ref ไปที่ตารางยา
QUANTITY	SET OF ROW_INT	ปริมาณยา เปลี่ยนได้ตามเวลา

ตารางที่ 7-16 แสดงคอลัมน์ในตาราง MOP

17. ตาราง MOD (ปริมาณยาที่สั่งที่สัมพันธ์กับการวินิจฉัย): เก็บปริมาณยาที่แพทย์สั่งให้ผู้ป่วยจากการวินิจฉัย

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
DIAGNOSIS(P.K.)	INT	Ref ไปที่การวินิจฉัย
MEDICINE(P.K.)	INT	Ref ไปที่ตารางยา
QUANTITY	SET OF ROW_INT	ปริมาณยา เปลี่ยนได้ตามเวลา

ตารางที่ 7-17 แสดงคอลัมน์ในตาราง MOD

18. ตาราง MOA (ปริมาณยาที่สั่งที่สัมพันธ์กับการเข้าพักรักษา): เก็บปริมาณยาที่แพทย์สั่งให้ผู้ป่วยในช่วงที่เข้าพักรักษา

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
ADMISSION(P.K.)	INT	Ref ไปที่การเข้าพักรักษา
MEDICINE(P.K.)	INT	Ref ไปที่ตารางยา
QUANTITY	SET OF ROW_INT	ปริมาณยา เปลี่ยนได้ตามเวลา

ตารางที่ 7-18 แสดงคอลัมน์ในตาราง MOA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19. ตาราง LAB_NUMBER (LAB ชนิดตัวเลข): เก็บรายละเอียดของLAB ชนิดตัวเลขแต่ละประเภท ตามชื่อLAB โดยเก็บค่ามาตรฐาน หน่วยการวัด และราคาค่าตรวจ

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
LAB_CODE(P.K.)	SERIAL	หมายเลขLAB
NAME	CHAR(20)	ชื่อLAB
STANDARD_FROM	INT	ค่ามาตรฐานเริ่มต้น
STANDARD_TO	INT	ค่ามาตรฐานสุดท้าย
UNIT	CHAR(10)	หน่วยการวัด
PRICE	SET OF ROW_INT	ราคา เปลี่ยนแปลงได้ตามเวลา

ตารางที่ 7-19 แสดงคอลัมน์ในตาราง LAB_NUMBER

20. ตาราง LAB_CHAR (LAB ชนิดตัวอักษร): เก็บรายละเอียดของLAB ชนิดตัวอักษร และราคาค่าตรวจ

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
LAB_CODE(P.K.)	SERIAL	หมายเลขLAB
NAME	CHAR(20)	ชื่อLAB
PRICE	SET OF ROW_INT	ราคา เปลี่ยนแปลงได้ตามเวลา

ตารางที่ 7-20 แสดงคอลัมน์ในตาราง LAB_CHAR

21. ตาราง LAB_PHOTO (LAB ชนิดตัวรูปภาพ): เก็บชื่อLAB ชนิดรูปภาพ และราคาค่าตรวจ

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
LAB_CODE(P.K.)	SERIAL	หมายเลขLAB
NAME	CHAR(30)	ชื่อLAB
PRICE	SET OF ROW_INT	ราคา เปลี่ยนแปลงได้ตามเวลา

ตารางที่ 7-21 แสดงคอลัมน์ในตาราง LAB_PHOTO

22. ตาราง ADMISSION (การเข้าพักรักษา): เก็บรายละเอียดในการเข้าพักรักษา เช่น แพทย์ที่รักษา ห้อง วันเข้า วันออก เป็นต้น โดยมีการอ้างถึงตารางการตรวจประจำวันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
AD_CODE(P.K.)	SERIAL	หมายเลขการเข้าพัก
DOCTOR	INT	Ref ไปที่แพทย์ที่รักษา
PACKAGE	INT	Ref ไปแพคเกจ
ROOM	INT	Ref ไปที่ห้องของการเข้าพักครั้งนั้น
AD_DIAGNOSIS	INT	Ref ไปที่การตรวจประจำวัน
IN_TIME	DATETIME YEAR TO MINUTE	วันที่ที่เข้ารักษา
OUT_TIME	DATETIME YEAR TO MINUTE	วันที่ที่ออกจากโรงพยาบาล

ตารางที่ 7-22 แสดงคอลัมน์ในตาราง ADMISSION

23. ตาราง AD_DIAGNOSIS (การตรวจประจำวัน): เก็บผลการตรวจประจำวัน เช่น อุณหภูมิ ความดัน อัตราการเต้นของหัวใจ แผนการรักษา เป็นต้น ซึ่งค่าเหล่านี้สามารถเปลี่ยนแปลงได้ตามเวลา และมีการเปลี่ยนแปลงบ่อย

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
AD_DIAGNOSIS_CODE (P.K.)	SERIAL	หมายเลขการตรวจประจำวันของการเข้าพัก
TEMPERATURE	SET OF ROW_INT	อุณหภูมิ เปลี่ยนแปลงตามเวลาได้
PULSE	SET OF ROW_INT	
REPISTORY	SET OF ROW_INT	
BLOOD_PRESSURE_L	SET OF ROW_INT	ความดันโลหิต(ต่ำ) เปลี่ยนแปลงตามเวลาได้
BLOOD_PRESSURE_H	SET OF ROW_INT	ความดันโลหิต(สูง) เปลี่ยนแปลงตามเวลาได้
NURSING_PLAN	SET OF ROW_INT	แผนการรักษา เป็นเซ็ตของ ref ที่ชี้ไปที่ตาราง PHOTO
DIAGNOSIS_RECORD	SET OF ROW_INT	บันทึกการวินิจฉัย เป็นเซ็ตของ ref ที่ชี้ไปที่ตาราง PHOTO

ตารางที่ 7-23 แสดงคอลัมน์ในตาราง AD DIAGNOSIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24. ตาราง ADMISSION_PACKAGE (แพคเกจการเข้าพักรักษา): เก็บราคาของแพคเกจแต่ละตัว ซึ่งเป็นราคารวมค่าบริการต่างๆ ทั้งค่าพยาบาล ค่าอาหาร ค่าดูแลรักษา แต่ไม่รวมค่าห้องพัก

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
PACKAGE_CODE (P.K.)	SERIAL	หมายเลขแพคเกจ
PACKAGE_NAME	CHAR(20)	ชื่อแพคเกจ
PRICE	SET OF ROW_INT	ราคาค่าบริการ เปลี่ยนได้ตามเวลา

ตารางที่ 7-24 แสดงคอลัมน์ในตาราง ADMISSION PACKAGE

25. ตาราง PHOTO (ภาพ): เก็บรูปภาพ เพื่อให้ตารางอื่นมาอ้างไปใช้

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
PHOTO_ID	SERIAL	หมายเลขรูป
PIC	BLOB	เก็บไฟล์ประเภทรูปภาพ

ตารางที่ 7-25 แสดงคอลัมน์ในตาราง PHOTO

7.2 การพัฒนาโปรแกรมประยุกต์

จากการวิเคราะห์และออกแบบระบบดังที่ได้กล่าวมา จะทำให้สามารถพัฒนาโปรแกรมประยุกต์ขึ้นใช้งานได้อย่างถูกต้องและตรงตามความต้องการของระบบ โดยระบบที่เลือกเป็นกรณีศึกษาคือระบบงานโรงพยาบาล ซึ่งเป็นระบบงานที่ตัวระบบและข้อมูลของระบบค่อนข้างมีความซับซ้อนสูง จึงจำเป็นที่จะต้องมีการวิเคราะห์ ออกแบบ และพัฒนาระบบอย่างถี่ถ้วน และครอบคลุมงานต่างๆ ให้มากที่สุด เพื่อแสดงให้เห็นถึงการจัดการข้อมูลเชิงเวลาของระบบงานที่ได้พัฒนาขึ้น

7.2.1 การสร้างฐานข้อมูล

ก่อนที่จะมีการสร้างโปรแกรมประยุกต์ขึ้นใช้งาน จำเป็นที่จะต้องมีฐานข้อมูลของระบบงาน เพื่อใช้ในการทดลองออกแบบ และจะได้ทราบถึงโครงสร้างของตารางต่างๆ และรูปแบบการติดต่อกับตารางในฐานข้อมูล ซึ่งจะทำให้โปรแกรมประยุกต์สามารถทำงานได้อย่างถูกต้อง

การออกแบบฐานข้อมูลจะกระทำจาก DBA โดยจะเป็นผู้สร้างโครงสร้างของตาราง (Schema) ทุกตารางที่ได้ทำการออกแบบไว้ ให้เป็นไปตาม Specification ที่ได้กำหนดไว้ รวมถึงการกำหนด Key ต่างๆ ของตารางในการเป็น Primary Key หลักของตาราง และเป็น Referential Key ชี้ไปยังตารางอื่น รวมถึงการกำหนดชนิดของแอททริบิวท์ต่างๆ ในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างฐานข้อมูลจะใช้ภาษา SQL ซึ่งเป็นภาษามาตรฐานที่ใช้ในการ manipulate ฐานข้อมูลโดยทั่วไป โดยอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ 9.14 จะใช้ตามมาตรฐาน SQL3 ซึ่งจะรองรับแนวคิดเชิงวัตถุด้วย การสร้างตารางฐานข้อมูลจะกระทำโดยผ่านเครื่องมือใช้งานของ Informix คือ SQL Editor ที่จะเป็นตัวจัดการรับภาษา SQL เข้าไปแล้วสามารถดำเนินการกับฐานข้อมูลได้

นอกจากนี้ อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ 9.14 ยังสามารถรองรับภาษาที่ใช้ในการจัดการฐานข้อมูลอีกชนิดหนึ่ง คือ SPL (Structure Procedural Language) ซึ่งเป็นภาษาเชิงโครงสร้าง ใช้ในการจัดการกับข้อมูล โดยมีลักษณะภาษาคล้ายภาษาในการโปรแกรมมิ่ง เช่น ภาษาปาสคาล ซึ่งได้นำภาษา SPL นี้มาใช้ในการสร้าง Routine ต่างๆ ทั้งที่เป็น Procedure และ Function ขึ้นมาใช้งาน ทำให้การติดต่อกับฐานข้อมูลทำได้สะดวกยิ่งขึ้น โดยส่วนใหญ่แล้วในโครงงานนี้ได้นำภาษา SPL มาใช้ในการเข้าถึงแต่ละ Element ของ Set เพราะข้อมูลชนิด Set จะไม่สามารถเข้าถึงได้จากการใช้ภาษา SQL เพียงอย่างเดียว จึงจำเป็นที่จะต้องเขียน Routine ที่ใช้ในการ Access Element ต่างๆ ของข้อมูลที่มีการจัดเก็บแบบ Set ขึ้น ตัวอย่างการสร้างตารางในฐานข้อมูล

7.2.2 การติดต่อกับระบบจัดการฐานข้อมูล

ระบบที่พัฒนาขึ้นจะต้องสามารถติดต่อกับระบบจัดการฐานข้อมูลได้ ซึ่งในที่นี้ใช้ระบบจัดการฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ 9.14 โดยระบบจัดการฐานข้อมูลตัวนี้สามารถรองรับสถาปัตยกรรมของโปรแกรมประยุกต์ได้หลายแบบ อาทิเช่น สถาปัตยกรรมแบบ two-tier คือมี Database Server ต่อกับ Client หรือสถาปัตยกรรมแบบ two-tier ที่มี Database Server, Application Server หรือ Web Server และ Client หรือสถาปัตยกรรมแบบ Multitier ซึ่งในระบบงานที่ได้พัฒนาขึ้นนั้น มิได้เน้นไปในส่วนของโปรแกรมประยุกต์มากเท่าใดนัก โดยโปรแกรมประยุกต์เป็นมุ่งเน้นไปในการแสดงให้เห็นถึงการจัดการกับข้อมูลเชิงเวลาในฐานข้อมูลเชิงวัตถุสัมพันธ์ จึงได้ใช้สถาปัตยกรรมของโปรแกรมประยุกต์แบบ two-tier คือมี Database Server และ Client เชื่อมต่อกัน โดยเครื่อง Client/Server เป็นเครื่องเดียวกันเพื่อง่ายต่อการประยุกต์ใช้งาน แต่ถึงอย่างไร การติดต่อกับ Database Server ก็ยังต้องติดต่อผ่าน ODBC (Open Database Connectivity) ซึ่งเป็นตัวที่จะช่วยในการสื่อสารข้อมูลกับระบบจัดการฐานข้อมูล

ODBC จะช่วยในการเชื่อมต่อกับผลิตภัณฑ์ฐานข้อมูลเชิงพาณิชย์ตัวอื่นๆ, ใช้ในการเข้าถึงดาต้าซอร์สของไฟล์ต่างๆ ในฐานข้อมูล รวมทั้งมีฟังก์ชันให้ใช้กับฐานข้อมูลด้วย เช่น การทำงานกับคีย์ที่ระบบสร้างขึ้น (system-generated key), ดาต้าไทป์ชนิดบลือบ และสตอร์โพรซีเยอร์ ในการพัฒนาแอพพลิเคชั่นจะใช้ Data Director ODBC DataDriver ในการเชื่อมการติดต่อข้อมูลของแอพพลิเคชั่นกับฐานข้อมูลอินฟอร์มิกซ์ ซึ่งจะต้องมีการสร้างดาต้าซอร์สขึ้นมาใหม่ โดยใช้ไดรเวอร์ INFORMIX 2.80 32 BIT ซึ่งเป็นไดรเวอร์ ในการสร้างดาต้าซอร์สขึ้นมาติดต่อ กับระบบจัดการข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ 9.14

หลังจากได้สร้าง DSN ที่ใช้ในการติดต่อกับ Database Server ในระบบจัดการฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ 9.14 ผ่าน ODBC แล้ว เครื่องมืออีกตัวหนึ่งที่สามารถช่วยในการติดต่อกับฐานข้อมูลได้ คือ Data Director ซึ่งเป็นเครื่องมือตัวหนึ่งในการสร้าง Model ของ Database Server โดยตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

model นี้จะเป็นเก็บ Schema ของตารางทั้งหมดในฐานข้อมูลนั้น ซึ่งเวลาพัฒนาโปรแกรมประยุกต์จะทำให้ไม่ต้องติดต่อกับฐานข้อมูลตลอดเวลา โดยสามารถใช้ Model เป็นตัวจำลองข้อมูลของฐานข้อมูลได้ แต่หากฐานข้อมูลมีการแก้ไข เช่น การแก้ไข Schema ของตาราง การสร้าง Routines ขึ้นใหม่ การสร้างตารางขึ้นใหม่ Model จะมองไม่เห็นถึงการเปลี่ยนแปลงนี้ จำเป็นที่จะต้อง Refresh Model ใหม่ เพื่อ update ข้อมูลต่างๆ Model

เมื่อได้ Model ขึ้นมาแล้ว เราสามารถที่จะสร้างตารางเสมือน (Virtual table) ขึ้นมาใช้เองได้ โดยที่จะมองไม่เห็นตารางนี้จริงๆ ในฐานข้อมูล นอกจากนี้เรายังสามารถกำหนดความสัมพันธ์ระหว่างแอททริบิวท์และตารางต่างๆ ให้เป็นไปตามฐานข้อมูล หรือกำหนดขึ้นใหม่ภายในโมเดลเองก็ได้

เมื่อได้ Model ขึ้นมาแล้วจำเป็นที่จะต้องสร้าง Datagroup สำหรับแต่ละงานของโปรแกรมประยุกต์ เพื่อประโยชน์ในการกำหนดขอบเขตของการเข้าถึงข้อมูลของงานนั้นๆ และเมื่อนำ Datagroup เชื่อมต่อไปในโปรแกรมประยุกต์แล้ว จะทำให้โปรแกรมประยุกต์สามารถทราบถึงตารางหลักที่จะใช้ในการค้นหาข้อมูลได้ ทราบถึงความสัมพันธ์ของตารางต่างๆ เพื่อการเข้าถึงข้อมูลของตารางอื่นจากตารางหลัก

หลังจากสร้าง Datagroup ต่างๆ ขึ้นมาแล้ว ถ้าหากต้องการเชื่อมต่อส่วนต่างๆ ของโปรแกรมประยุกต์กับแอททริบิวท์ต่างๆ ของตารางในฐานข้อมูล จะต้องสร้าง Datalink เพื่อเชื่อมต่อส่วนต่างๆ ของโปรแกรมประยุกต์กับฐานข้อมูล ในระบบของแอททริบิวท์

จะเห็นได้ว่าเครื่องมือ Data Director จะมีประโยชน์อย่างยิ่งในการอำนวยความสะดวกในการพัฒนาโปรแกรมประยุกต์ขึ้นมาก แต่ถึงอย่างไร ผู้ใช้ก็จำเป็นที่จะต้องกำหนดการเชื่อมต่อกับฐานข้อมูลขึ้นเอง และหากฟังก์ชันของระบบงานมีความหลากหลายมากขึ้น และไม่สามารถใช้เครื่องมือช่วยในการติดต่อได้อย่างเพียงพอ ผู้ใช้ก็จำเป็นต้องเขียนโปรแกรมเพื่อการใช้งานขึ้นมาเอง

7.2.3 โปรแกรมวิชวลเบสิค

ในการพัฒนาโปรแกรมประยุกต์ได้ใช้โปรแกรมวิชวลเบสิค เป็นตัวทำการพัฒนา เนื่องจากโปรแกรมวิชวลเบสิคเอง มีความสามารถในการโปรแกรมเชิงวัตถุ มีความสามารถในการโปรแกรมให้เป็นมิตรกับการติดต่อกับผู้ใช้ (user interface friendly) พร้อมทั้งยังสามารถแสดงผลข้อมูลได้หลายรูปแบบ อาทิเช่น เท็กซ์บ็อกซ์ คอมโบ้บ็อกซ์ ลิสต์ กริด และอิมเมจบ็อกซ์ เป็นต้น ซึ่งความสามารถเหล่านี้ทำให้โปรแกรมที่พัฒนามาจากโปรแกรมวิชวลเบสิค มีความน่าใช้งานและมีความสามารถมากมาย นอกจากนี้ตัวโปรแกรมวิชวลเบสิคเองยังสามารถรองรับแอพพลิเคชันที่เป็นลักษณะในการติดต่อกับฐานข้อมูลอีกด้วย พร้อมทั้งมี ActiveX Control ที่เกี่ยวกับการควบคุม การติดต่อกับฐานข้อมูลให้ใช้มากมาย เช่น DBGrid DBList DBCombo เป็นต้น

การโปรแกรมในวิชวลเบสิค จะมีลักษณะเป็นการโปรแกรมตามอีเวนต์ต่างๆ ของ ActiveX Control แต่ละตัว โดยสามารถกำหนดโค้ดการทำงานของแต่ละส่วน หรือหลายๆส่วนของตัวโปรแกรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการติดต่อกับฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิรฟเวอร์ 9.14 นั้น การโปรแกรมโดยใช้วิชวลเบสิคจะสามารถติดต่อได้ โดยใช้เครื่องมือช่วย คือ Data Director for Visual Basic มาช่วยในการติดต่อ โดยสามารถใช้การ Drag and Drop โดยการลากแอททริบิวท์ที่ต้องการทำ DataLink จาก Datagroup มาใส่ไว้ใน ActiveX Control ตัวที่ต้องการทำการเชื่อมต่อ ซึ่งจะช่วยอำนวยความสะดวกในการทำงานได้มาก แต่เนื่องจากการจัดเก็บในฐานข้อมูลเชิงวัตถุสัมพันธ์ที่ได้ออกแบบนั้น จะเน้นไปที่การจัดเก็บแอททริบิวท์ในลักษณะของวัตถุโดยแต่ละแอททริบิวท์ที่เปลี่ยนแปลงตามเวลา จะเก็บข้อมูลลงคอลเลคชั่นไทพ์ชนิดเซต ทำให้ไม่สามารถสร้าง Datalink แบบ Drag and drop จาก Data Director ไปยังโปรแกรมวิชวลเบสิคได้ เพราะลักษณะข้อมูลแบบเซตมีการเข้าถึงข้อมูลที่ต่างออกไป ต้องเรียกใช้รูทีนที่สร้างขึ้นมา ดังนั้นจึงสามารถใช้วิธีการ Drag and Drop กับ ActiveX Control ตัวที่มีค่าไม่เปลี่ยนตามเวลาเท่านั้น ส่วนแอททริบิวท์ส่วนอื่นที่เหลือจะต้องใช้วิธีการโปรแกรมในการติดต่อกับฐานข้อมูล และการเข้าถึงข้อมูล ซึ่งจะได้กล่าวต่อไป

7.2.4 การติดต่อกับฐานข้อมูลโดยการโปรแกรม

ในการติดต่อกับฐานข้อมูลนั้น เราสามารถที่จะติดต่อกับฐานข้อมูลได้โดยการเขียนโปรแกรม โดยเราจะทำการเขียนโปรแกรมผ่าน ActiveX Control ของอินฟอร์มิกซ์ที่ช่วยในการติดต่อคือ Navigation Control ซึ่งจะติดตั้งให้เมื่อมีการลง Data Director for Visual Basic แล้ว หากใช้การเชื่อมต่อแบบ Drag and Drop ก็จะมี Navigation Control ของ Datagroup นั้นปรากฏในฟอร์มเช่นกัน ซึ่งจะสามารถกำหนดตารางหลัก หรือตารางรองในการจัดการข้อมูลของ Navigation Control ก็ได้ โดย Navigation Control 1 ตัว จะสามารถจัดการข้อมูลได้ทีละตาราง (หากมีการกำหนดความสัมพันธ์ของตารางหลักกับตารางอื่นไว้แล้ว ก็สามารถเชื่อมต่อไปยังข้อมูลของตารางอื่นได้ด้วย) และ Navigation Control ยังมีปุ่มช่วยในการทำงาน เช่น ปุ่มในการค้นหาข้อมูล ป้อนข้อมูล แก้ไขข้อมูล สร้างเรคคอร์ดใหม่ เลื่อนไปยังเรคคอร์ดที่ต้องการได้ ซึ่งอำนวยความสะดวกอย่างมากสำหรับฐานข้อมูลเชิงสัมพันธ์ แต่ในการโปรแกรมนั้นเราจะใช้ Navigation Control เป็นเพียงตัวที่ใช้ในการเชื่อมต่อกับฐานข้อมูลเชิงวัตถุสัมพันธ์เท่านั้น โดยไม่ต้องคำนึงถึงตารางหลัก ตารางรองของ Navigation Control ตัวนั้น

ตัวอย่างเช่น สมมติเรามี datagroup ตัวหนึ่ง ชื่อ patience ซึ่งมีตารางหลักคือตาราง patience แล้วเราก็ทำการสร้าง Navigation Control ชื่อ patience_nav ขึ้นในฟอร์ม โดยอ้างอิงกับ datagroup ตัวนี้ เราจะได้ patience_nav เป็น datasource ตัวนึงที่สามารถเชื่อมต่อได้

การโปรแกรมผ่านตัว datasource นี้ ส่วนใหญ่จะใช้ในการส่งคำสั่งภาษา SQL ไปยังฐานข้อมูลเพื่อจัดการกับข้อมูล ค้นหาข้อมูล หรือ execute โพรซีเดอร์หรือฟังก์ชันในฐานข้อมูลนั้น โดยมีรูปแบบการใช้งาน ดังนี้

```
sql = "select hn,birth_day from patience"
set oVtable = patience_nav.ddodatagroup.ExecuteSQLCommand(sql)
```

จากตัวอย่างจะเป็นการกำหนดสตริงภาษา sql ให้กับตัวแปร sql โดยเป็นภาษา sql ในการเลือกค่าในคอลัมน์ hn และ birth_day ทั้งหมดจากตาราง patience โดยในการเรียกใช้จะผ่านทาง datagroup ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Navigation Control ที่ชื่อ patience_nav และผลลัพธ์ที่ได้จะมาเก็บในตัวแปรชื่อ oVtable ซึ่งเป็นตัวแปรชนิด ddoTable โดยมีลักษณะคล้ายกับตารางข้อมูลเสมือนที่สร้างขึ้นชั่วคราว และถ้าหากต้องการนำค่าที่ Select มาได้ออกมา จะสามารถทำได้ดังนี้

```
While Not oVtable.EOT
combo1.Additem(oVtable.Columns(1).value)
combo2.Additem(oVtable.Columns(2).value)
oVtable.Nextrecord
Wend
```

จากตัวอย่างจะเป็นการนำค่าของคอลัมน์ที่ 1 ของตัวแปร oVtable ซึ่งก็คือ ค่าในคอลัมน์ hn ของตาราง patience ที่ได้จากการ Select มาใส่เพิ่มใน ActiveX Control ที่ชื่อ combo1 และจะนำค่าของคอลัมน์ที่ 2 ของตัวแปร oVtable ซึ่งก็คือค่า birth_day ของตาราง patience ที่ได้จากการ Select มาใส่เพิ่มใน ActiveX Control ที่ชื่อ combo2 จากนั้นก็จะทำการไปยังเรคคอร์ดถัดไปของตารางด้วยคำสั่ง oVtable.Nextrecord แล้ววนลูปใส่ค่าไปจนกว่าจะถึงท้ายตาราง (EOT ; End of Table)
หากมีการใช้งานตารางเสมือนที่สร้างขึ้นเสร็จเรียบร้อยแล้ว จำเป็นที่จะต้องมีการลบตารางเสมือนนั้นทิ้งเพื่อเป็นการคืนการล็อกตารางในฐานข้อมูลเพื่อกระทำการกับข้อมูลในตารางนั้น โดยใช้คำสั่ง

```
patience_nav.ddoDatagroup.DeleteVirtualTable (oVtable.Name)
```

จากตัวอย่างจะเป็นการลบตารางเสมือน oVtable ซึ่งในที่นี้ตาราง oVtable จะล็อกตาราง patience ไว้ ทำให้กระบวนการอื่นๆ ที่จะทำกับตาราง patience ไม่สามารถทำได้ เนื่องจากมีการล็อกตารางไว้ ซึ่งเป็นการป้องกัน concurrency control ของระบบจัดการฐานข้อมูลอินฟอร์มิกซ์เอง หากเราไม่มีการลบตารางเสมือนทิ้งเพื่อปลดล็อกตาราง patience จะทำให้เกิดความผิดพลาดกับกระบวนการอื่นที่จำเป็นต้องใช้ตาราง patience ในการปฏิบัติงาน

นอกจากนี้เรายังสามารถใช้ Navigation Control ในการเรียกโพรซีเยอร์หรือฟังก์ชันที่เป็นรูทีนที่สร้างขึ้นเองในฐานข้อมูลได้อีกด้วย ตัวอย่างเช่น สมมติมีรูทีนหนึ่งเป็นโพรซีเยอร์ชื่อว่า UPDATE_SALARY

```
CREATE PROCEDURE update_salary(THEROWID INT,THEMONEY INT)
        DEFINE THESET COLLECTION;
        DEFINE THEROW ROW_INT;
        . . .
END PROCEDURE;
```

จะสามารถเรียกใช้งานโพรซีเยอร์ได้ดังนี้

```
sql = "execute procedure update_salary(" & INT1 "," & INT2 & ")"
patience_nav.ddoDatagroup.ExecuteSQLCommand(sql)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งก็จะทำให้โพรซีเยอร์นั้นสามารถทำงานได้ โดยสามารถส่งค่า parammeter เป็นตัวแปรที่ได้กำหนดค่าไว้แล้ว ในที่นี้คือ INT1 และ INT2 เข้าไป execute ก็ได้ แต่หากเป็นฟังก์ชันซึ่งจะต้องมีการคืนค่ากลับมาจะต้องนำตารางมารับค่าที่คืนกลับมาด้วย ตัวอย่างเช่น มีการสร้างฟังก์ชันไว้ในฐานข้อมูล ด้วยภาษ SPL ดังนี้

```
create function newrowid_ad_dia()
        returning int, char;
        define therowid int;
        . . .
        return therowid, thechar;
end function;
```

ซึ่งในการเรียนใช้โดยการ โปรแกรมในวิชวลเบสิค จะต้องกำหนดโค้ดโปรแกรม ดังนี้

```
sql = "execute function newrowid ad_dia()"
set oVtable = patience_nav.ddodatagroup.ExecuteSQLCommand(sql)
```

ค่าที่ได้จะคืนกลับมาอยู่ในตาราง oVtable โดยค่า parameter ตัวแรกที่คืนกลับมาจะอยู่ใน oVtable.Columns(1).value และค่า parameter ตัวที่สองจะอยู่ใน oVtable.Columns(2).value เช่นนี้ไปเรื่อยๆ ขึ้นอยู่กับจำนวนพารามิเตอร์ที่คืนค่ากลับมา

7.2.5 เป้าหมายการใช้งานแอพพลิเคชัน

การนำแอพพลิเคชันที่พัฒนาขึ้นมาใช้งานนั้น จะมีขอบเขตของการใช้งานจำกัดตามระบบที่ได้ทำการออกแบบไว้ โดยสามารถจัดกลุ่มประเภทของผู้ใช้งานแอพพลิเคชัน และขอบเขตของระบบงานในส่วนต่างๆ ของผู้ใช้ได้ ดังต่อไปนี้

1) ผู้ใช้ที่เป็นพนักงานฝ่ายทะเบียน ผู้ใช้งานในระดับนี้จะสามารถจัดการเกี่ยวกับทะเบียนคนไข้ ที่เป็นคนไข้นอกทั้งหมด ซึ่งจะมีงานต่างๆ ดังต่อไปนี้

1.1) การสร้างเรคคอร์ดใหม่ เป็นการสร้างเรคคอร์ดใหม่ของคนไข้ เมื่อมีคนไข้ใหม่เข้ามาในโรงพยาบาล โดยจะต้องทำการใส่ค่าแอทริบิวท์ต่างๆ ที่จำเป็นของคนไข้ลงไป เพื่อ Insert Row ใหม่เข้าไปในตาราง patience

1.2) การแก้ไขข้อมูล เป็นการแก้ไขข้อมูลของคนไข้โดยจะสามารถแก้ไขข้อมูลในตาราง patience ได้ 2 ลักษณะ คือ

- การแก้ไขข้อมูลเนื่องจากการป้อนค่าผิด ซึ่งเป็นการแก้ไขข้อมูลแบบ Non-sequenced โดยเสมือนเป็นการแก้ไขข้อมูลที่ผิดนั้นทันที โดยไม่คำนึงถึงช่วงเวลาในการแก้ไข ซึ่งจะทำให้ข้อมูลเก่าหายไป เก็บเป็นข้อมูลใหม่ในตารางแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแก้ไขข้อมูลเมื่อข้อมูลมีการเปลี่ยนค่า ซึ่งจะเป็นการแก้ไขข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ อาทิเช่น ข้อมูลชื่อ ซึ่งเป็นข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ เช่นเมื่อผู้ป่วยมาติดต่อกับโรงพยาบาลอีกครั้ง แล้วมีการเปลี่ยนชื่อ จะต้องทำการแก้ชื่อของผู้ป่วยนั้นให้ตรงตามความเป็นจริง แต่ก็ยังคงเก็บชื่อเก่าของผู้ป่วยไว้ โดยไม่ให้เกิดการซ้อนทับของเวลากัน เพราะค่าของแต่ละชื่อจะมีค่าอยู่คนละช่วงเวลากัน

1.3) การค้นหาข้อมูล เป็นการค้นหาข้อมูลของคนไข้ตามเงื่อนไขที่ใช้ในการค้นหา โดยสามาถกำหนดเงื่อนไขในการค้นหาจากหมายเลข HN, ค้นหาจากชื่อ, ค้นหาจากนามสกุล, หรือค้นหาจากชื่อและนามสกุล ก็ได้ และมีช่วงเวลาที่เป็นเงื่อนไขในการค้นหาเป็น 3 ประเภท ดังนี้

การค้นหาแบบ Current เป็นการค้นหาข้อมูลที่มีค่าวาลิด ณ เวลาปัจจุบันนั้น โดยการนำวันเวลา ณ ปัจจุบันนั้นเป็นเงื่อนไขในการค้นหา

- การค้นหาแบบ Sequenced เป็นการกำหนดช่วงเวลาในอดีตที่ต้องการค้นหา เช่น การหาจากข้อมูลของคนไข้ จากชื่อในอดีตของคนไข้ จะทำให้สามารถทราบรายละเอียดต่างๆ ของคนไข้คนนั้นได้
- การค้นหาแบบ Non-Sequenced จะมีลักษณะคล้ายๆ กับการค้นหาทั้งตารางฐานข้อมูล โดยไม่คำนึงถึงเวลาที่เป็นเงื่อนไขในการค้นหา เช่น การค้นหาข้อมูลของคนไข้ทั้งหมด จากชื่อชื่อหนึ่งที่เป็นเงื่อนไข

2) ผู้ใช้ที่เป็นผู้วินิจฉัย หรือแพทย์ผู้วินิจฉัย จะเป็นผู้ใช้ที่จะทำการวินิจฉัยคนไข้รายใดรายหนึ่งที่เป็นคนไข้นอก โดยสามารถที่จะกระทำการค้นหาข้อมูลเก่าๆ ของการวินิจฉัยของคนไข้รายนั้นได้, สามารถทำการวินิจฉัยคนไข้รายนั้นๆ ได้ ซึ่งจะรวมถึงการสั่งตรวจทางห้องปฏิบัติการ และการสั่งยาให้แก่คนไข้ด้วย โดยจะสามารถแบ่งงานต่างๆ ได้ดังนี้

2.1) การทำการวินิจฉัยใหม่ ผู้ใช้จะต้องทราบก่อนว่าต้องการทำการวินิจฉัยคนไข้รายใด โดยเลือกจาก HN ของคนไข้ ซึ่งจะทราบประวัติบางส่วนของคนไข้รายนั้นด้วย เพื่อยืนยันว่าเป็นคนไข้รายนั้นจริงก่อนทำการวินิจฉัย หลังจากนั้นก็จะทำการผลการวินิจฉัย โดยจะจัดเก็บในลักษณะของรูปภาพเพื่อความสะดวก เพราะในงานจริงนั้น แพทย์ผู้วินิจฉัยจะไม่ถนัดในการป้อนข้อมูลเข้าระบบด้วยการพิมพ์ จึงใช้วิธีการเก็บเป็นไฟล์รูปภาพแทน ซึ่งขั้นตอนนี้มิได้กล่าวถึงในรายละเอียดของการนำข้อมูลจากผู้ใช้เข้าสู่ระบบ โดยถือว่าระบบมีไฟล์รูปภาพการวินิจฉัยอยู่แล้ว นอกจากนี้ผู้ใช้ยังสามารถทำการสั่งตรวจทางห้องปฏิบัติการให้แก่คนไข้ เมื่อข้อมูลในการวินิจฉัยไม่เพียงพอ พร้อมทั้งจะสามารถสั่งยาในการรักษา คนไข้รายนั้นๆ ได้ด้วย

2.2) การค้นหาข้อมูลการวินิจฉัย ผู้ใช้จะต้องทราบหมายเลข HN ของคนไข้ที่จะค้นหาการวินิจฉัยก่อน แล้วจึงสามารถค้นหาข้อมูลการวินิจฉัยในอดีตได้ โดยเมื่อทราบหมายเลข HN ของคนไข้รายนั้นแล้ว ระบบก็จะทำการแสดงผลการวินิจฉัยครั้งล่าสุดของคนไข้ แล้วผู้ใช้สามารถที่จะย้อนกลับไปดูการวินิจฉัยครั้งก่อนๆ ของคนไข้ โดยเลือกจากเวลาหรือปุ่มควบคุมในหน้าจอได้ รวมทั้งผู้ใช้ยังสามารถดูผลการตรวจทางห้องปฏิบัติการของคนไข้รายนั้นได้ ซึ่งจะแสดงเฉพาะผลตรวจที่ได้รับการป้อนค่าแล้วจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องปฏิบัติการเท่านั้น และสามารถแสดงผลตรวจย้อนหลังได้ รวมทั้งผู้ใช้ยังสามารถดูผลการสั่งยาของคนไข้ โดยค้นหาจากเงื่อนไขของเวลา หรือชนิดของยาที่ต้องการก็ได้

3) ผู้ใช้ที่เป็นเจ้าหน้าที่ทางห้องปฏิบัติการ จะเป็นผู้ใช้ที่ทำการจัดการใส่ผลทางห้องปฏิบัติการของคนไข้รายนั้นๆ เข้าไป โดยแบ่งคนไข้ออกเป็น 2 กรณีคือ

3.1) บันทึกผลการตรวจทางห้องปฏิบัติการ ที่สั่งจากการวินิจฉัยคนไข้นอก โดยจะต้องป้อนหมายเลขการวินิจฉัย (DN) เพื่อระบุคนไข้ที่ต้องการใส่ค่าผลการตรวจ
บันทึกผลการตรวจทางห้องปฏิบัติการ ที่สั่งจากการวินิจฉัยคนไข้ใน ซึ่งจะต้องป้อนหมายเลขคนไข้ใน (AN) เพื่อระบุคนไข้ที่ต้องการใส่ค่าผลการตรวจ

3.2) การใส่ผลทางห้องปฏิบัติการทั้ง 2 กรณี จะมีลักษณะเหมือนกัน คือ สามารถเลือกผลที่ต้องการใส่ค่าได้ แล้วใส่ค่า โดยค่าที่ใส่ในผลทางห้องปฏิบัติการจะมี 3 ชนิด คือ ผลการตรวจที่เป็นตัวเลข, ตัวอักษร และรูปภาพ (หรือผลเอ็กซ์เรย์) แต่กระบวนการในการทำงานของทั้ง 2 กรณีจะไม่เหมือนกัน เพราะจะกระทำการกับข้อมูลคนละตารางกัน ซึ่งจะมองไม่เห็นในระดับของผู้ใช้

4) ผู้ใช้ที่เป็นเจ้าหน้าที่ที่เกี่ยวข้องกับคนไข้ใน เช่น เจ้าหน้าที่จัดการคนไข้ใน, พยาบาลผู้ดูแลคนไข้ และแพทย์ผู้ทำการวินิจฉัยคนไข้ใน ซึ่งจะมีงานต่างๆ ดังต่อไปนี้

4.1) การ Admit คนไข้ในรายใหม่ เมื่อต้องการทำการ Admit คนไข้ในรายใหม่ จำเป็นที่จะต้องรู้ HN ของคนไข้ก่อน เพื่อระบุรายละเอียดของคนไข้ พร้อมทั้งเลือกห้อง เลือกแพ็กเกจการรักษา และกำหนดเวลาการ Admit (หากไม่ใช่วันปัจจุบัน) ซึ่งจะทำให้สร้างเรคคอร์ดใหม่ของคนไข้ใน ในตาราง admission งานนี้จะกระทำโดยเจ้าหน้าที่ที่จัดการคนไข้ใน

4.2) การป้อนข้อมูลต่างๆ ของคนไข้ใน เมื่อคนไข้ได้ทำการ Admit แล้ว จะมีการป้อนข้อมูลของคนไข้ในอยู่เป็นประจำ เช่น ข้อมูลการตรวจประจำวัน อันได้แก่ อุณหภูมิ ความดันเลือด อัตราการหายใจ และอัตราการเต้นของหัวใจ การใส่ข้อมูลแผนการรักษาพยาบาล (Nursing Plan) การวินิจฉัย การสั่งการตรวจทางห้องปฏิบัติการ และการสั่งยาให้แก่คนไข้ใน รวมถึงการ Check Out คนไข้ใน โดยงานส่วนนี้จะกระทำโดยพยาบาลผู้ดูแลคนไข้ และแพทย์ผู้ทำการวินิจฉัยคนไข้ใน

4.3) การค้นหาข้อมูลของคนไข้ใน เมื่อมีการจัดเก็บข้อมูลต่างๆ ของคนไข้ในไว้ ก็จะต้องสามารถทำการค้นหาข้อมูลของคนไข้ใน เพื่อใช้ในการวางแผนการรักษา หรือการวินิจฉัยในครั้งต่อๆ ไปของคนไข้ โดยใช้เงื่อนไขในการค้นหาคือ AN กับช่วงเวลา ทำให้สามารถค้นหาข้อมูลการตรวจประจำวันของคนไข้คนนั้นได้ รวมถึงข้อมูลแผนการรักษา การวินิจฉัย การสั่งยา และผลทางห้องปฏิบัติการได้ โดยงานส่วนนี้จะกระทำโดยพยาบาลผู้ดูแลคนไข้ และแพทย์ผู้ทำการวินิจฉัยคนไข้ใน

5) ผู้ใช้ที่เป็นเจ้าหน้าที่คำนวณค่าใช้จ่ายของคนไข้ จะเป็นผู้ที่ออกรายงานค่าใช้จ่าย ของคนไข้ พร้อมทั้งคำนวณค่าใช้จ่ายต่างๆ ที่ใช้ในการรักษาพยาบาลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทสรุปและวิจารณ์

8.1 สรุปผลและวิจารณ์ผลโครงงาน

จุดประสงค์โดยรวมของการทำโครงงานนี้ คือ การประยุกต์ใช้ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์จัดการกับฐานข้อมูลเชิงเวลา โดยในโครงงานนี้ได้ใช้ระบบจัดการฐานข้อมูลของอินฟอร์มิกซ์ ในชุดของผลิตภัณฑ์ อินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ เวอร์ชั่น 9.14 จากจุดประสงค์ดังกล่าว จึงได้ทำการศึกษาและพัฒนาตามลำดับขั้นตอน ดังต่อไปนี้

- ศึกษาทฤษฎีเชิงเวลา ว่าฐานข้อมูลเชิงเวลาคืออะไร สามารถใช้ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์จัดการได้อย่างไร และความยุ่งยากที่เกิดขึ้น ทั้งในการออกแบบฐานข้อมูล การค้นหาข้อมูล และการแก้ไขข้อมูลในฐานข้อมูลเชิงเวลา
- ศึกษาระบบจัดการฐานข้อมูลอินฟอร์มิกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ เวอร์ชัน 9.14 โดยศึกษาทั้งในเรื่องวิธีการติดตั้ง (Install) การตั้งค่า (Setup) และการปรับแต่งระบบจัดการฐานข้อมูลให้เหมาะสมกับการใช้งาน รวมถึงการศึกษาถึงความสามารถของอินฟอร์มิกซ์ในเรื่องของชนิดข้อมูล, คุณสมบัติเชิงวัตถุ, วิธีการติดต่อกับโปรแกรมภายนอกและไวยกรณ์ภาษาแอสคิวแอลที่ใช้ในการจัดการข้อมูลในอินฟอร์มิกซ์ ซึ่งในขั้นตอนนี้ต้องใช้ระยะเวลาในการศึกษา และทำความเข้าใจ พอสมควร
- วิเคราะห์และหาวิธีการใช้อินฟอร์มิกซ์เพื่อจัดการกับความยุ่งยากที่เกิดขึ้นในการจัดการกับฐานข้อมูลเชิงเวลา รวมทั้งการแก้ปัญหาที่เกิดขึ้น เมื่อใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์จัดการ ซึ่งได้ข้อสรุป 2 ประเด็นคือ ใช้ โรว์ไอดี (rowid) เป็นตัวอ้างอิงถึงข้อมูล แทนคีย์หลัก (Primary Key) ทั้งหมด และเก็บเวลาที่ข้อมูลมีการเปลี่ยนแปลงไว้กับแต่ละแอตทริบิวต์ แทนการเก็บไว้ติดกับทั้งโรว์ โดยใช้ชนิดข้อมูลเป็นเซ็ต เนื่องจากคุณสมบัติของเซ็ต ไม่สามารถมีสมาชิกซ้ำกันได้ คือ จะไม่มีโรว์ไทป์ตัวไหนในเซ็ตที่เหมือนกันเลย (ซ้ำซ้อนแบบนอนซีเควนท์) ทำให้เป็นการเช็คเรื่องของ integrity constraint ตั้งแต่ในระดับชนิดข้อมูล แต่ก็มีข้อเสียในเรื่องของการโปรแกรมและประยุกต์ใช้งาน เนื่องจากเซ็ตไม่มีลำดับและข้อมูลที่ใส่เข้าไปใหม่ไม่ได้รับประกันว่าจะอยู่ท้ายสุดของเซ็ต ทำให้ในการโปรแกรมต้องวนลูปเช็คตั้งแต่ต้นจนถึงท้ายเซ็ต
- ทดลองเขียนโปรแกรมและออกแบบฐานข้อมูลระบบงานเล็กๆ ตามทฤษฎีและแนวทางที่ได้กำหนดไว้ เพื่อเป็นการทดลองและพิสูจน์ทฤษฎีและแนวทางที่ได้วางไว้ ปรากฏว่าเกิดปัญหาขึ้นมากมาย ส่วนใหญ่มักเป็นปัญหาเกี่ยวกับระบบจัดการฐานข้อมูลอินฟอร์มิกซ์เอง จึงต้องย้อนกลับไปศึกษาในบางเรื่อง ซึ่งเป็นปัญหาในเรื่องของเทคนิคและสภาพแวดล้อมที่ใช้งาน ส่วนในเรื่องของทฤษฎีและแนวทางในการประยุกต์ใช้ ยังคงใช้ได้อยู่
- ศึกษาแอพพลิเคชันที่นำมาเป็นกรณีศึกษา ได้หาข้อมูลและศึกษาทำความเข้าใจกับระบบที่จะเลือกศึกษามากมาย เช่น ด้านเวชระเบียน, คลังยา, ฐานข้อมูลบุคลากร, ฐานข้อมูลเกี่ยวกับการจัดการภายในวัด, ฐานข้อมูลผู้ป่วยใน, ฐานข้อมูลผู้ป่วยนอก, ฐานข้อมูลคลีนิคขนาดเล็ก เป็นต้น ซึ่งในที่สุดแล้วได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกกรณีศึกษาเป็น ระบบงานในโรงพยาบาล เพราะเล็งเห็นถึงข้อมูลที่มีการเปลี่ยนแปลงเชิงเวลาอย่างชัดเจน รวมทั้งมีข้อมูลหลากหลายประเภท และข้อมูลมีจำนวนมาก ซึ่งทำให้การจัดการยังไม่เป็นระบบเท่าที่ควร จึงนำมาเป็นระบบที่เลือกใช้ในกรณีศึกษา

- วิเคราะห์และออกแบบระบบงานโรงพยาบาล โดยรวบรวมข้อมูลจากโรงพยาบาลต่างๆ จากนั้นจึงออกแบบฐานข้อมูลเชิงเวลา ซึ่งในโครงงานนี้ได้มีการเสนอหลักการที่ใช้ในการออกแบบเอาไว้ด้วย เพื่อประโยชน์ในการนำไปใช้พัฒนาระบบอื่นๆ ที่เป็นเชิงเวลาต่อไป
- เขียนโปรแกรมและสร้างข้อมูล รวมทั้งรูทีนต่างๆ ในอินฟอร์มิกซ์ จากนั้นจึงเป็นการทดสอบความถูกต้องและความสมบูรณ์ของโปรแกรมแอพพลิเคชัน และเพิ่มฟังก์ชันการใช้งาน ที่แสดงให้เห็นประโยชน์ของฐานข้อมูลเชิงเวลาขึ้นอีกเล็กน้อย

จากการศึกษาทฤษฎีเชิงเวลา พบว่าฐานข้อมูลเชิงเวลานั้นไม่ใช่เป็นเพียงฐานข้อมูลที่เหมาะกับงานบางประเภทเท่านั้น แต่เป็นฐานข้อมูลที่เหมาะสม และควรนำมาประยุกต์ใช้กับระบบงานในทุกระบบงาน เพราะฐานข้อมูลควรเป็นสิ่งที่สามารถให้ข้อมูลได้ ไม่ว่าจะเป็นข้อมูลในเวลาใดๆ ก็ตาม และคาดว่าในอนาคตอันใกล้นี้ แอพพลิเคชันจะเก็บข้อมูลลงฐานข้อมูลเชิงเวลาแทบทั้งสิ้น แต่ความยุ่งยากในการจัดการฐานข้อมูลเชิงเวลา เป็นปัจจัยสำคัญที่ทำให้ผู้คนไม่สนใจที่จะใช้ฐานข้อมูลเชิงเวลา ซึ่งเป็นการเสียโอกาสในการใช้ประโยชน์อันทรงคุณค่าจากฐานข้อมูลเชิงเวลา

ดังนั้นจึงควรมีการกำหนดวิธีการ หรือแนวทางปฏิบัติในการออกแบบและสร้างฐานข้อมูลเชิงเวลา เพื่อกำจัดความยุ่งยากที่จะเกิดขึ้น ซึ่งในวิทยานิพนธ์ฉบับนี้ได้พยายามนำเสนอวิธีการเหล่านี้ และคาดว่าแนวทางปฏิบัติที่ได้นำเสนอ น่าจะมีประโยชน์ต่อการนำไปใช้ เพื่อพัฒนาวิธีการที่ดีกว่านี้ต่อไป

การนำคุณสมบัติของระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์มาประยุกต์ใช้กับฐานข้อมูลเชิงเวลา เป็นเพียงทางเลือกหนึ่งในการจัดการกับข้อมูลเชิงเวลาเท่านั้น ยังมีอีกหลายทางเลือกในการประยุกต์ใช้ เปรียบเสมือนการทำอาหาร ไม่ว่าจะใช้เตาถ่าน เตาแก๊ส หรือเตาอบ ก็สามารถทำอาหารได้ทั้งนั้น โดยแต่ละวิธีก็มีความแตกต่างกันในเรื่องของความเหมาะสม ความยากง่ายในการใช้งานก็ต่างกันไป ซึ่งข้อดีประการหนึ่งของการประยุกต์ใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ คือ ผู้ที่เคยใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์สามารถทำความเข้าใจระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ได้ง่าย เพราะโดยส่วนใหญ่ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ยังคงเหมือนระบบจัดการฐานข้อมูลเชิงสัมพันธ์อยู่ จะต่างกันเพียงบางส่วนเท่านั้น ที่เห็นได้ชัดเจนคือ ความหลากหลายของชนิดข้อมูล และคุณสมบัติทางวัตถุ ซึ่งทำให้ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ข้ามพ้นข้อจำกัดต่างๆ ที่มีในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ และทำให้สามารถประยุกต์ใช้ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์กับฐานข้อมูลเชิงเวลาได้สะดวกขึ้น

สำหรับระบบงานโรงพยาบาลที่ได้พัฒนาขึ้น เป็นตัวอย่างทั้งทางด้านแนวทางในการพัฒนาและประโยชน์ที่จะได้รับจากการใช้งานแอพพลิเคชันที่เป็นฐานข้อมูลเชิงเวลานั้น เป็นแอพพลิเคชันที่ค่อนข้างสมบูรณ์ แต่การจะนำไปใช้ในระบบงานจริงนั้นต้องมีการเพิ่มเติมในส่วนของเครือข่าย, ความปลอดภัยในการใช้ข้อมูล การควบคุมการเข้าใช้งานพร้อมกัน และการจัดลำดับคิวในระบบงานให้เป็นระบบอัตโนมัติทั้งหมด (เพราะอยู่นอกขอบเขตของโครงงาน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 แนวทางการพัฒนาต่อ

จากที่ได้กล่าวมาแล้วว่า วิธีการที่ได้นำเสนอเป็นเพียงทางเลือกหนึ่งในการจัดการกับความยุ่งยากและปัญหาที่เกิดกับฐานข้อมูลเชิงเวลาเท่านั้น ดังนั้นแนวทางการพัฒนาต่อ คือ ศึกษาและกำหนดวิธีการใหม่ๆ ในการออกแบบฐานข้อมูลเชิงเวลาที่มีประสิทธิภาพ และง่ายในการทำความเข้าใจ รวมทั้งค้นหาเครื่องมือที่เหมาะสมในการนำทฤษฎีที่ได้กำหนดขึ้นมาประยุกต์ใช้ให้มีประสิทธิภาพมากที่สุด ซึ่งแนวทางที่น่าจะดีที่สุดคือพัฒนาเครื่องมือขึ้นมาเอง โดยเป็นเครื่องมือที่ทำให้การสร้างฐานข้อมูลเชิงเวลาทำได้ง่าย ผู้ใช้เครื่องมือสามารถใช้ในการพัฒนาระบบที่ใช้ฐานข้อมูลเชิงเวลา โดยไม่จำเป็นต้องรู้ทฤษฎีเชิงเวลาเลยก็ได้ เพื่อประโยชน์สูงสุดในการนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] C.J. Date, "An Introduction to Database Systems Volume I", ADDISON-WESLEY PUBLISHING COMPANY, 1986

[2] Richard T. Snodgrass, "Managing Temporal Data A Five-Part Series", 1998

[3] พิมพ์ผกา อังสวานนท์ และภุชงค์ ตั้งเจตน์, "การจัดเก็บวงจรดิจิตอลโดยใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์" ; ปริญญานิพนธ์, 2541

[4] ปริญญา ดิษฐจร และ สหศิลป์ ดรหมั่น, "ระบบฐานข้อมูลลายนิ้วมือโดยใช้ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์" ; ปริญญานิพนธ์, 2541

[5] ชวลิต ศิวบวรวัฒนา และ สุรชัย เลิศบุญช่วยกุล, "ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์เพื่อการรู้จำอักษรภาษาไทย" ; ปริญญานิพนธ์, 2541

[6] ชญานี ตั้งสุขเกษมสันต์ และ ชุติมาภรณ์ แก้วประทุม, "การเก็บดาต้าโฟลว์ไดอะแกรมโดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์" ; ปริญญานิพนธ์, 2541

[7] ปวีณา โสภณธนเศรษฐ์, "การออกแบบฐานข้อมูลห้องปฏิบัติการ" ; รายงาน, 2541

[8] ศิริพร วรางคณาวดี, "ระบบฐานข้อมูลและการให้การพยาบาลผู้ป่วยใน" ; รายงาน, 2541

[9] กิตติ ภักดีวัฒนะกุล และ จำลอง ครูอุตสาหะ, "วิชวลเบสิค 6 ฉบับโปรแกรมเมอร์", ไทยเจริญการพิมพ์, 2543

[10] Informix Inc., "Informix Manual"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การติดตั้งโปรแกรมระบบงานโรงพยาบาล

ความต้องการของระบบ

1. ระบบปฏิบัติการ WindowsNT โดยมีระบบไฟล์เป็น NTFS
2. หน่วยความจำหลัก อย่างน้อย 16 MB
3. เนื้อที่ว่างในฮาร์ดดิสอย่างน้อย 40 MB สำหรับตัวโปรแกรม และอย่างน้อย 20 MBสำหรับ DBSpace
4. โปรโตคอลในการสื่อสาร TCP/IP
5. โปรแกรม Visual Basic Version 6.0 ขึ้นไป

ขั้นตอนการติดตั้ง

จะแบ่งขั้นตอนการติดตั้ง ออกเป็น 4 ขั้นตอน ดังนี้

1. การติดตั้ง Informix Universal Server

Informix Universal Server แบ่งเป็น 2 ส่วน ได้ดังนี้

- ส่วนเซิร์ฟเวอร์ ประกอบไปด้วยยูนิเวอร์แซลเซิร์ฟเวอร์ ที่สนับสนุนความสามารถฐานข้อมูลเชิงวัตถุ สัมพันธ์และยูทิลิตี้ต่างๆ เช่น Dbaccess,Onmode,Oninit และ Onspaces เป็นต้น
- ส่วนไคลเอ็นท์ ประกอบไปด้วยทูลต่างๆ ดังนี้
  - SetNet32 Utility ใช้กำหนด Environment Variable และ Network Parameter ขณะรันไทม์ ซึ่งเป็นยูทิลิตี้ที่ติดตั้งบนไคลเอ็นท์พีซี
  - Schema Knowledge ใช้แสดงรายละเอียดภายในฐานข้อมูล รวมทั้งเมต้าดาต้าของฐานข้อมูล แบบกราฟิก โดยไม่สามารถแก้ไขข้อมูลใดๆบนฐานข้อมูลทาง Schema Knowledge ได้ รายละเอียดของฐานข้อมูลที่สามารถดูได้จาก Schema Knowledge มีดังนี้ Database,Routine,Table,View,Row Type,Inheritance View เป็นต้น
  - SQL Editor เป็นเอดิเตอร์ที่ได้เขียนภาษา SQL เพื่อติดต่อกับเซิร์ฟเวอร์ และแสดงผลการคิวรี่ในแบบกราฟิก
  - DBDK (DataBlade Developer Kit) เป็นทูลที่ใช้พัฒนาดาต้าเบลดโมดูล (Datablade Module) ซึ่งเป็นการขยายความสามารถของ IUS

การเตรียมการติดตั้ง

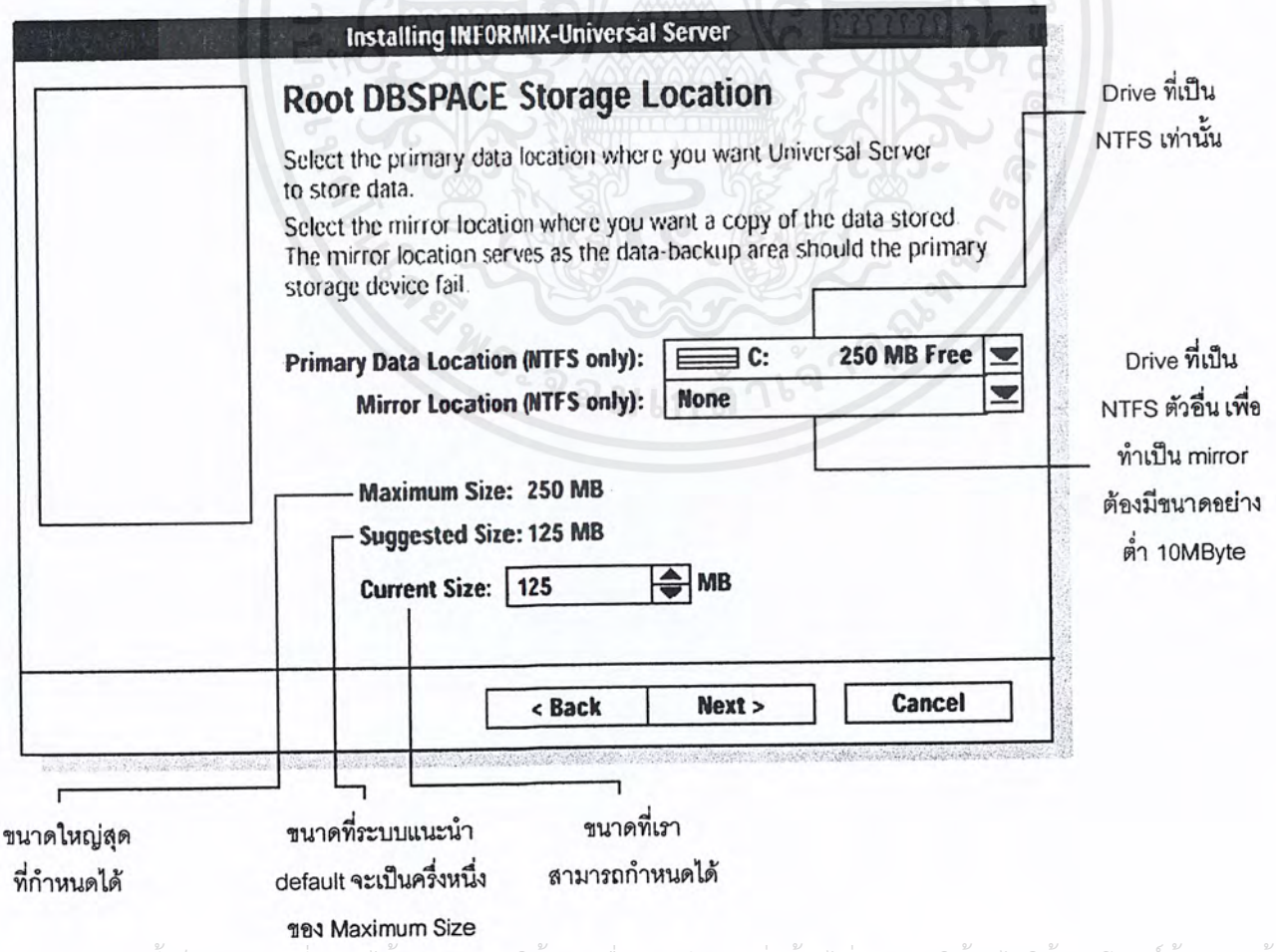
สามารถติดตั้งได้ทั้งบนยูนิกซ์และวินโดว์เอ็นที ในการทำโครงงานนี้ได้ใช้ระบบปฏิบัติการวินโดว์เอ็นที ซึ่งมีความต้องการของระบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. WindowNT 4.0 โดยต้องมีอย่างน้อยหนึ่งพาร์ทิชั่นที่เป็น NTFS เพียงใช้ลงIUS
2. โปรโตคอล TCP/IP
3. หน่วยความจำอย่างต่ำ 16 เมกะไบต์
4. เนื้อที่บนฮาร์ดดิสก์อย่างน้อย 40 เมกกะไบต์

ขั้นตอนการติดตั้ง

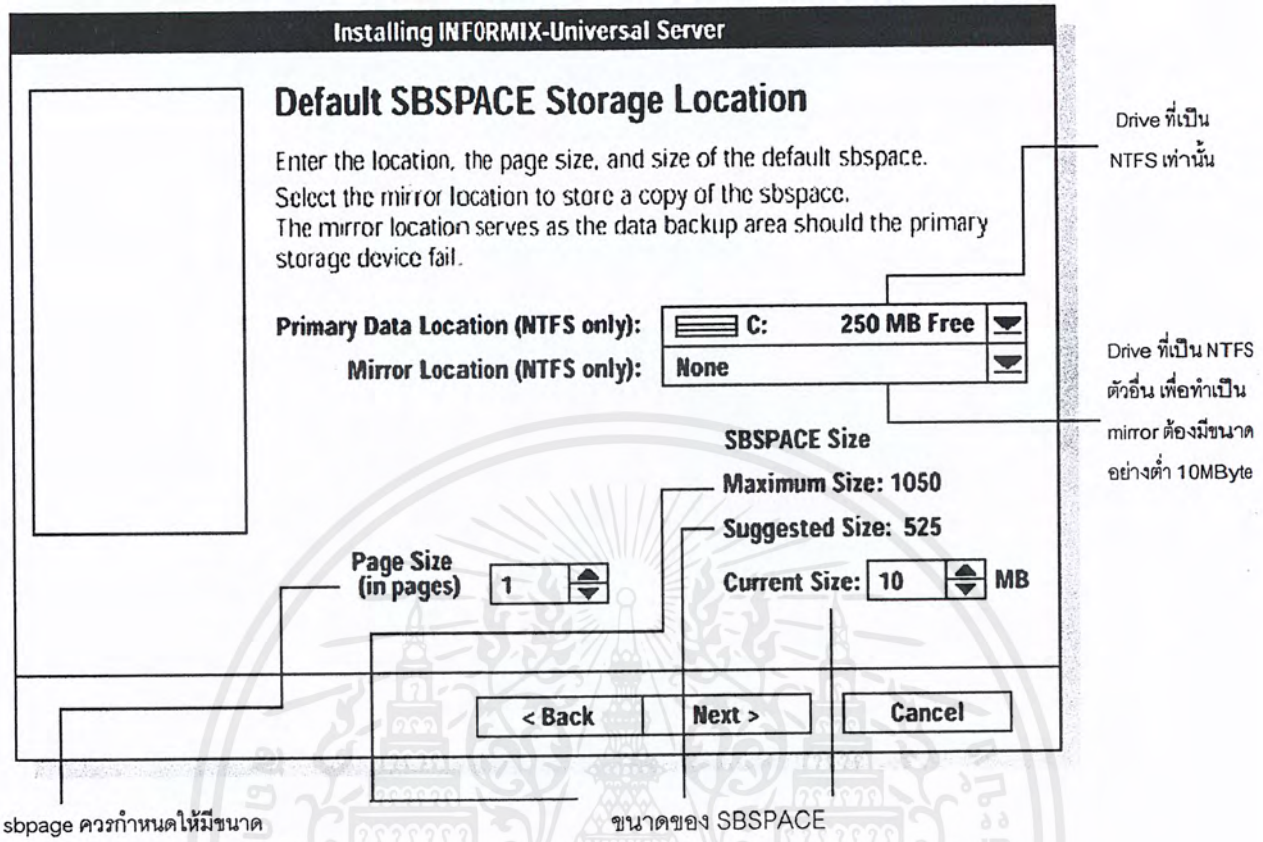
1. ผู้ติดตั้งต้องเป็นผู้ใช้ในกลุ่มของ Administration Group
2. การติดตั้งนั้น ให้ติดตั้งตัวเซิร์ฟเวอร์และ START เซิร์ฟเวอร์ใน services ของวินโดว์เอ็นทีก่อน แล้วจึงลงตัวไคลเอ็นท์ ไม่เช่นนั้นจะทำให้ไคลเอ็นท์ติดต่อกับเซิร์ฟเวอร์ไม่ได้ โดยการลงเซิร์ฟเวอร์ให้รัน Setup.exe เพื่อเริ่มติดตั้ง ส่วนรายละเอียดการใส่ค่าต่างๆนั้น ขอให้ดูในเอกสารของอินฟอร์มิกซ์ ในที่นี้จะกล่าวเฉพาะในส่วนที่สำคัญเท่านั้น
3. กำหนดไดเร็คทอรี่ที่จะทำการติดตั้ง โดยต้องเป็น NTFS เท่านั้น
4. จะมีช่องให้เลือกกำหนดบทบาทของ DB-Admin ว่าจะแยกหน้าที่กันในแต่ละงานหรือรวมกันใช้เพียง Account เดียว
5. กำหนดชื่อของดาต้าเบสเซิร์ฟเวอร์ และเริ่มถ่ายโอนข้อมูลไฟล์ลงสู่ระบบ
6. กำหนดไดร์ฟที่จะใช้เก็บฐานข้อมูล (DBSPACE Storage Location) รวมทั้งกำหนดขนาดของ DBSPACE ด้วย ดังรูป



รูปแสดงหน้าการกำหนดขนาดของ DBSPACE

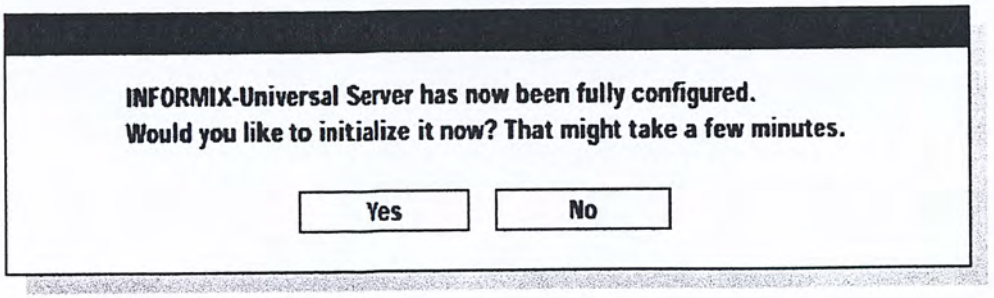
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. กำหนดที่และขนาดที่ใช้เก็บ Smart Large Object (SBSPACE) ดังรูปที่ ป2



รูปแสดงหน้าการกำหนดขนาดของ SBSPACE

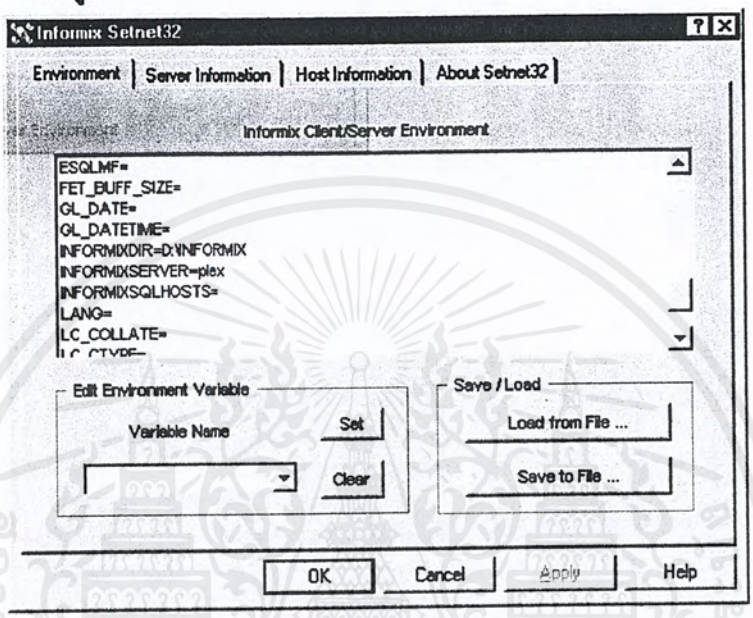
8. กำหนด path ของ Tape Device ถ้าไม่มีให้กำหนดใน path ให้เป็น 'NUL'
9. กำหนดรหัสผ่านของ Accout Informix เนื่องจาก Informix จะสร้าง Account Informix มีหน้าที่ดูแลระบบ (DBadmin)
10. กำหนดที่เก็บ Sqlhost File
11. ระบบจะบอกว่าได้ปรับแต่ง IUS เสร็จแล้ว ต้องการจะ initialize เลยหรือไม่ ขอแนะนำให้ตอบ 'Yes' เพื่อเป็นการเริ่มการทำงานของ IUS ซึ่งในขั้นตอนนี้จะใช้เวลาค่อนข้างมาก ประมาณ4-5นาที แต่ถ้าตอบ 'No' ในขั้นนี้ ต้องไปทำการ initialize เองโดยการไป start IUS ที่ services ของวินโดว์เอ็นทีโดยใส่พารามิเตอร์เป็น '-iy'



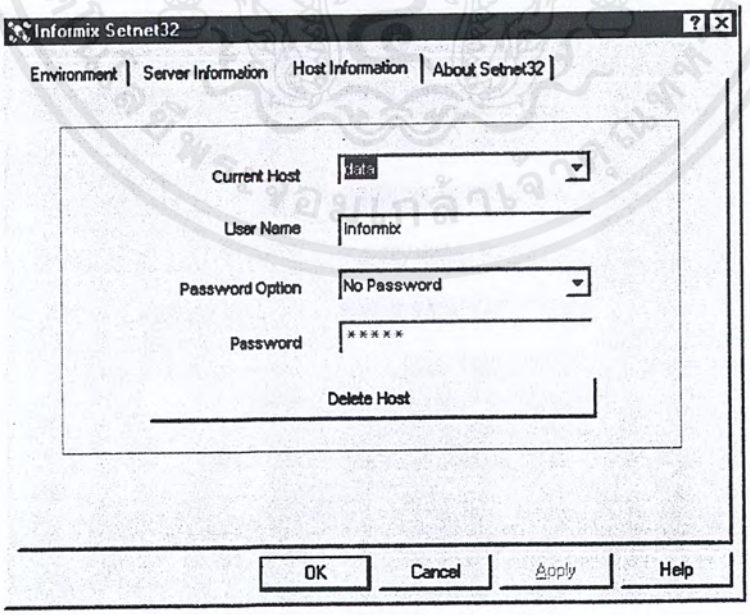
รูปแสดงคำถามว่าจะทำการ Initialize หรือไม่ หลังจากที่ได้ปรับแต่ง IUS แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ขั้นตอนการติดตั้งเซิร์ฟเวอร์ก็ได้เสร็จสิ้นแล้ว ต่อไปเป็นขั้นตอนการติดตั้งไคลเอ็นท์ ซึ่งทำได้ง่ายมาก เหมือนการลงโปรแกรมทั่วไป แต่ก่อนอื่นต้องแน่ใจว่าได้ initialize และ start IUS แล้ว โดยเริ่มจาก DBA และ DBDK ตามลำดับ โดยหลังจากที่ติดตั้ง DBA แล้ว ให้ทำการปรับค่าตัวแปรใน Setnet32 ซึ่งตัวแปรที่สำคัญคือ INFORMIXDIR และ INFORMIXSERVER โดยต้องกำหนด INFORMIXDIR ให้เป็นไดเร็คทอรี่ที่เก็บ IUS และ INFORMIXSERVER ให้เป็นชื่อเซิร์ฟเวอร์ที่ได้กำหนดในขั้นตอนการติดตั้งเซิร์ฟเวอร์ ดังรูป



รูปแสดงการกำหนดค่าตัวแปร INFORMIXDIR และ INFORMIXSERVER ในตัวอย่างนี้เซิร์ฟเวอร์มีชื่อว่า 'plex'

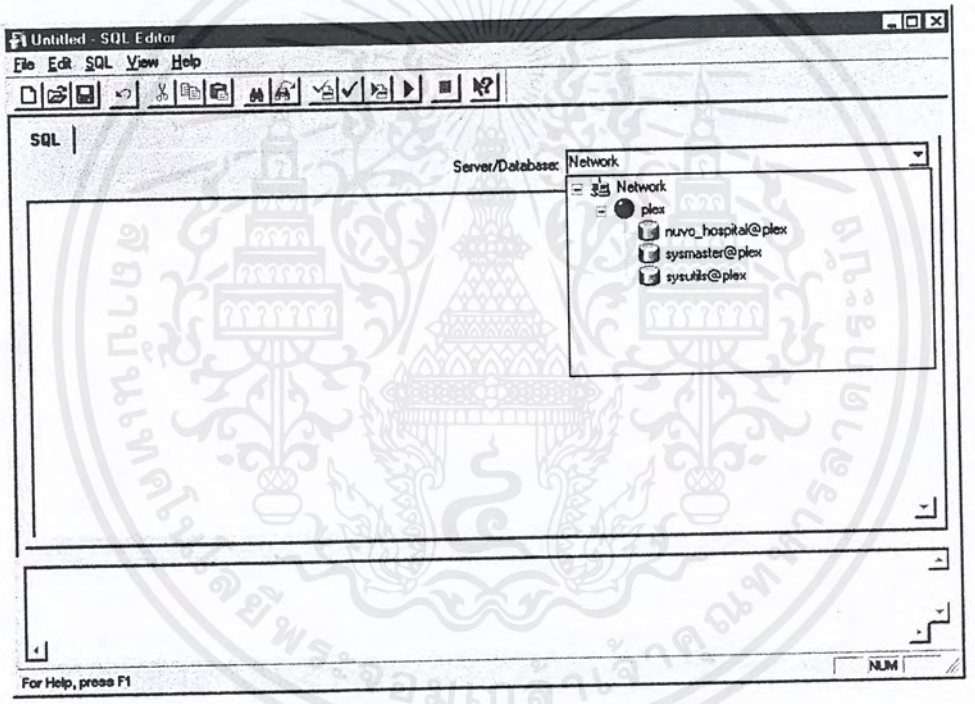


รูปแสดงการกำหนดรหัสผ่านในการเข้าถึงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังสามารถกำหนดรหัสผ่านในการเข้าถึงฐานข้อมูลอีกด้วย ดังรูปที่ ป5 จะเห็นได้ว่ากำหนดเป็น No Password หมายความว่าในการเข้าถึงฐานข้อมูลแต่ละครั้งนั้นไม่ต้องใส่รหัสผ่าน

13. ถึงขั้นนี้ก็เป็นอันเสร็จขั้นตอนการติดตั้งทั้งหมดแล้ว ในการทดลองว่าการติดตั้งนั้นสำเร็จหรือไม่ สามารถทำได้โดยการทดลองส่งคำสั่งSQL ไปยังเซิร์ฟเวอร์ โดยใช้ SQL editor ติดต่อไปยังเซิร์ฟเวอร์ ถ้าหากติดต่อ(connect) ได้ ให้ลองสร้างฐานข้อมูลขึ้นมาใหม่ โดยใช้คำสั่ง 'CREATE DATABASE <<ชื่อฐานข้อมูล>>;' แล้วลองสร้างตาราง,insert,query ดูตามลำดับ ถ้าไม่มีปัญหาใดๆเกิดขึ้น แสดงว่าการติดตั้งได้สำเร็จแล้ว ถ้าเกิดปัญหาเช่น ติดต่อเซิร์ฟเวอร์ไม่ได้ ให้ลองตรวจดูว่า ได้ start IUS หรือยัง,initialize IUS หรือยัง,กำหนดค่าตัวแปรใน Setnet32 ถูกต้องหรือไม่ หรืออาจเกิดจากตอนที่ลงไคลเอ็นท์ไม่ได้ start IUS ไว้เป็นต้น ซึ่งสามารถหาแนวทางแก้ไขปัญหาเหล่านี้ได้ใน Informix Find Error



รูปแสดงตัวอย่างหน้าจอของ SQL editor ถ้าสามารถติดต่อกับเซิร์ฟเวอร์ได้ จะแสดงชื่อเซิร์ฟเวอร์และฐานข้อมูลที่อยู่ในเซิร์ฟเวอร์นั้น

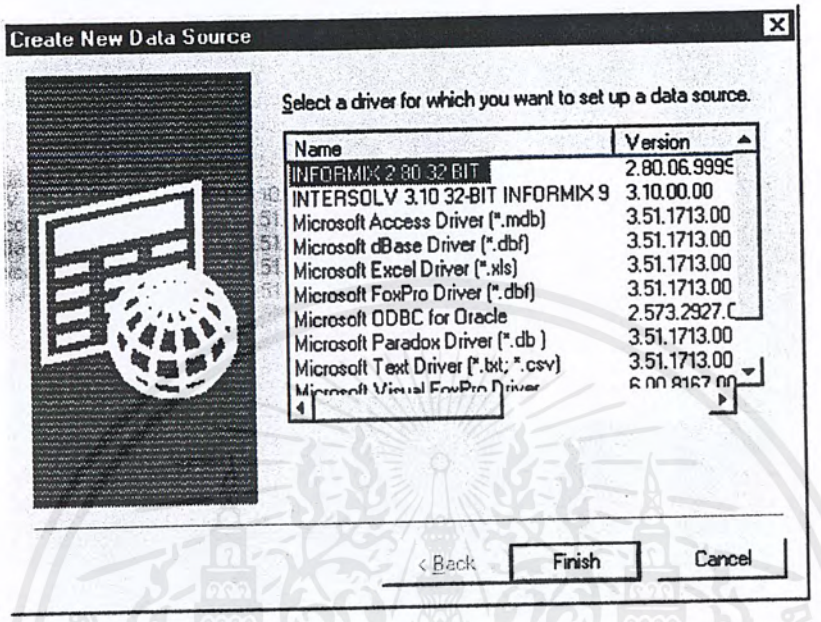
2. การติดตั้ง Data Director

Data Director เป็นเครื่องมือที่ช่วยในการติดต่อกับฐานข้อมูล ระหว่างฝั่ง Server กับฝั่ง Client โดยจะติดต่อผ่าน ODBC ซึ่งจะมี ODBC Driver ของ Informix หลายตัวที่ช่วยในการติดต่อ ได้แก่ INFORMIX 2.80 32 BIT, INTERSOLV 3.10 32 BIT INFORMIX 9 เป็นต้น ซึ่งสามารถรองรับการติดต่อกับ Informix DBMS ได้ทุกตัวแต่มีความสามารถแตกต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

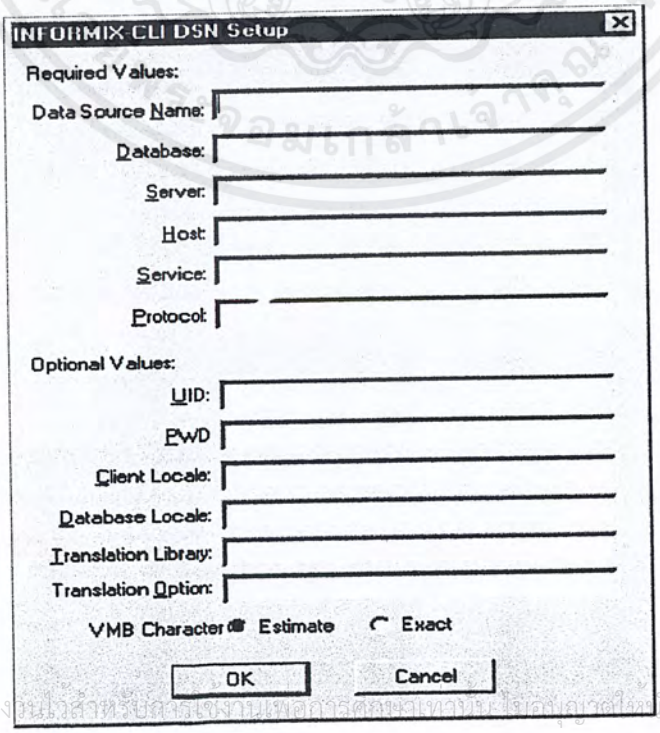
การเตรียมการติดตั้ง

ต้องทำการติดตั้ง ODBC Driver ก่อนโดยเรียกไฟล์ในการติดตั้ง ODBC Driver ของ Informix ที่ให้มา เมื่อติดตั้ง Driver เรียบร้อยแล้ว ก็จะปรากฏชื่อ Driver ในรายการ เพื่อให้ทำการสร้าง Data Source ใหม่ต่อไป



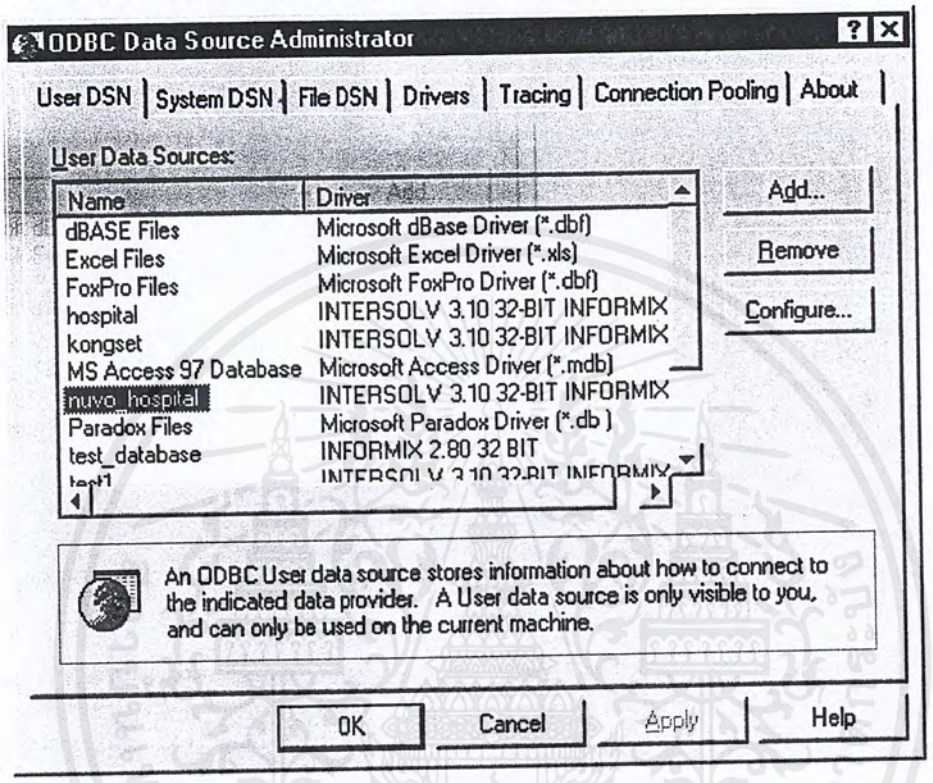
ต้องทำการสร้าง Data Source ใหม่โดยทำตามลำดับ ดังนี้

1. Control Panel => ODBC Data Sources
2. ไปที่ User DSN คลิกปุ่ม ADD เพื่อสร้าง Data Source ใหม่
3. เลือก Driver => แล้วคลิก Finish
4. จะปรากฏหน้าจอ INFORMIX-CLI DSN Setup เพื่อระบุรายละเอียดของ Data Source



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. กำหนดชื่อ Data Source Name ให้เป็น nuvo_hospital
6. กำหนด Database ให้เป็น nuvo_hospital
7. กำหนด Server, Host, Service, Protocol ให้เป็นไปตามที่กำหนดตอนติดตั้ง IUS (สามารถดูได้จาก Informix Setnet 32) แล้วคลิก OK
8. จะได้ Data Source ใหม่ที่สร้างขึ้นปรากฏใน User DSN ให้ใช้งาน



ขั้นตอนการติดตั้ง

1. ต้องทำการติดตั้งโปรแกรม Visual Basic Version 6.0 ให้เรียบร้อยก่อน
2. เรียกไฟล์เพื่อทำการติดตั้ง Data Director for Visual Basic โดยทำการติดตั้งตามขั้นตอนต่างๆ ของโปรแกรมติดตั้งจนเสร็จสมบูรณ์
3. เมื่อเรียกโปรแกรม Visual Basic จะเรียกใช้งาน Data Director ให้เองโดยอัตโนมัติ

3. การเริ่มต้น (Initialize) ฐานข้อมูล

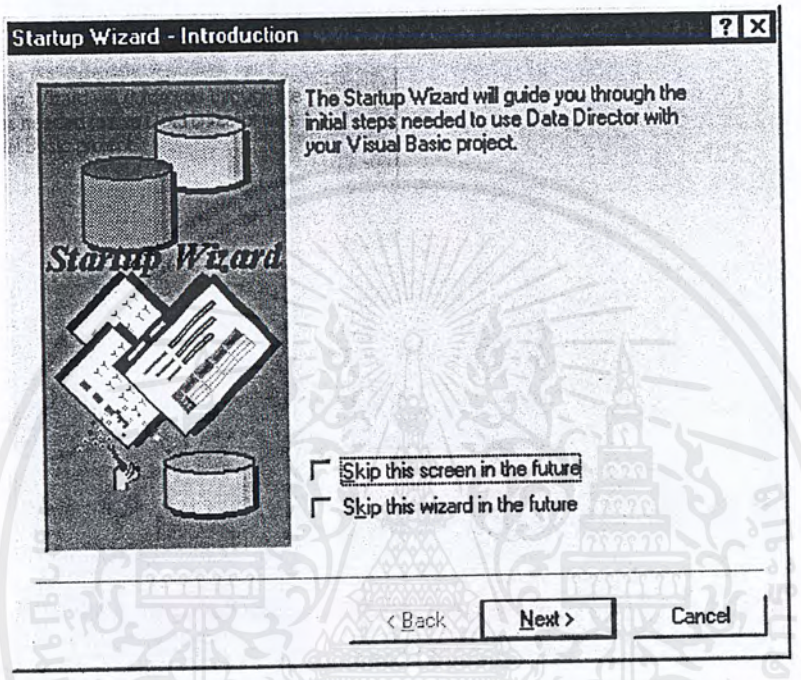
1. ทำการ Start Informix Universal Server ใน Control Panel => Service
2. รันไฟล์ SQL ต่างๆ ที่มีเพื่อทำการเริ่มต้น (Initialize) ฐานข้อมูล (ดูรายละเอียดของไฟล์ SQL ต่างๆ ได้ในคู่มืออธิบายโปรแกรมระบบงานโรงพยาบาล)
3. สร้างโมเดลที่ใช้ในการเชื่อมต่อฐานข้อมูลขึ้นใหม่ (อยู่ในหัวข้อต่อไป)
4. รันแอพพลิเคชั่นโดยเปิดไฟล์ Project1.vbp โดยใช้โปรแกรม Visual Basic 6.0 หรือเปิดไฟล์ HOSPITAL.EXE ขึ้นมาได้เลย ก็จะสามารถใช้งานแอพพลิเคชั่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

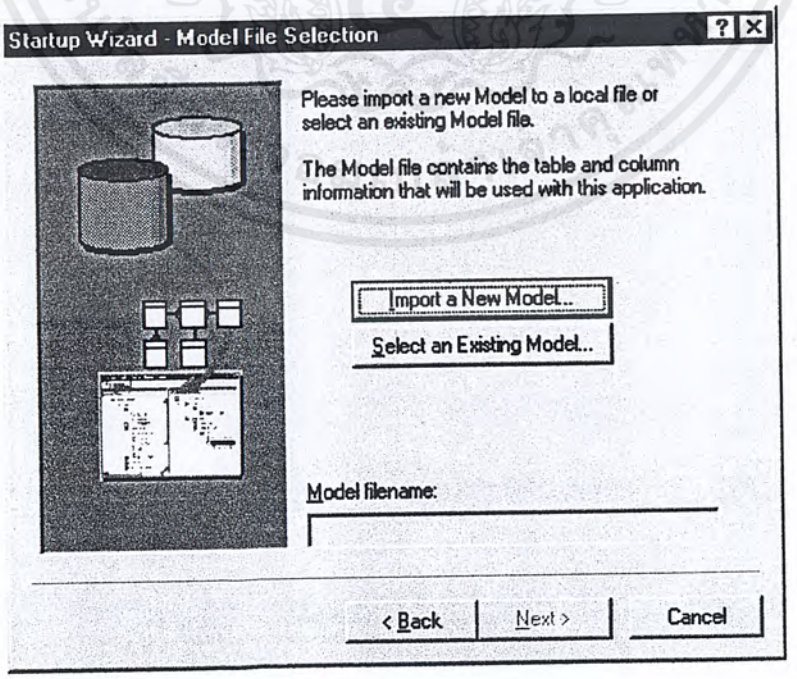
3. การสร้างโมเดลไฟล์เพื่อติดต่อกับแอพพลิเคชั่น

แอพพลิเคชั่นจำเป็นที่จะต้องมีไฟล์ที่ใช้ในการอ่านโครงตาราง รายละเอียดของฐานข้อมูล หรือไฟล์ต่างๆ ที่จำเป็นของฐานข้อมูลที่สร้างขึ้น เพื่อให้สามารถติดต่อกับฐานข้อมูลที่มีอยู่ได้ จึงต้องมีการสร้างโมเดลที่ใช้เชื่อมต่อฐานข้อมูลกับแอพพลิเคชั่น โดยมีขั้นตอนการสร้าง ดังนี้

1. เรียกโปรแกรม Visual Basic หรือ Data Director for Visual Basic ซึ่งจะเปิด Data Director ให้โดยอัตโนมัติ จะปรากฏ Startup Wizard



2. คลิกปุ่ม Next เพื่อทำการสร้างโมเดล จาก Startup Wizard – Introduction



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกปุ่ม Import New Model เพื่อทำการสร้างโมเดลใหม่ จาก Startup Wizard – Model File Selection
4. คลิกปุ่ม Next จาก Model Import Wizard
5. จะปรากฏ Data Source ต่างๆ ที่มีอยู่ ให้เลือก Data Source ที่เราสร้างขึ้นคือ nuvo_hospital จาก Import Model แล้วคลิกปุ่ม Next
6. จะปรากฏตารางทั้งหมดของ Data Source นั้น ให้เลือกตารางทุกตารางแล้วคลิกปุ่ม Finish จาก Items to Import
7. จะปรากฏ Import Status เพื่อทำการ Import Model หลังจากนั้นจะปรากฏ Save Model As เพื่อให้บันทึกชื่อโมเดล ให้ไปยังไดเรกทอรี่ที่เก็บไฟล์แอพพลิเคชั่นไว้ แล้วบันทึกในชื่อ nuvo_hospital จะได้ไฟล์ nuvo_hospital.mlt
8. จะกลับมาหน้า Startup Wizard และมีชื่อไฟล์ Model ที่สร้างขึ้นแล้ว ให้ทำการคลิกปุ่ม Next
9. เลือก Mastertable เป็น Informix.patience จาก Startup Wizard-Master table Selection จะได้ Datagroup Name เป็น patience แล้วคลิกปุ่ม Next
10. แล้วคลิก Finish จะสามารถใช้งานแอพพลิเคชั่นได้เรียบร้อยแล้ว

<u>การเรียกใช้งานแอพพลิเคชั่น</u>

สามารถเรียกใช้งานได้จากโปรแกรม Visual Basic หรือจากไฟล์ EXE ที่สร้างไว้ ดังนี้

1. เรียกไฟล์ Project1.vbp จะเรียกไฟล์ประกอบอื่นๆ รวมทั้งฟอร์มและโมดูลของแอพพลิเคชั่นทั้งหมด ถ้าหาไม่เจอโปรแกรมจะถามหาตำแหน่งที่เก็บไฟล์ ให้เลือกหาตำแหน่งที่เก็บแอพพลิเคชั่น และไฟล์ที่ต้องการ
2. หรือจะสามารถเรียกใช้งานโปรแกรมได้โดยเปิดโปรแกรม HOSPITAL.EXE ขึ้นมาทำงานได้เลย

<u>หมายเหตุ</u>

การเริ่มต้นฐานข้อมูล การสร้างโมเดล และใช้งานแอพพลิเคชั่น จำเป็นต้อง Initialize หรือ START Informix Universal Server เสียก่อน โดยไป START ที่ Control Panel => Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

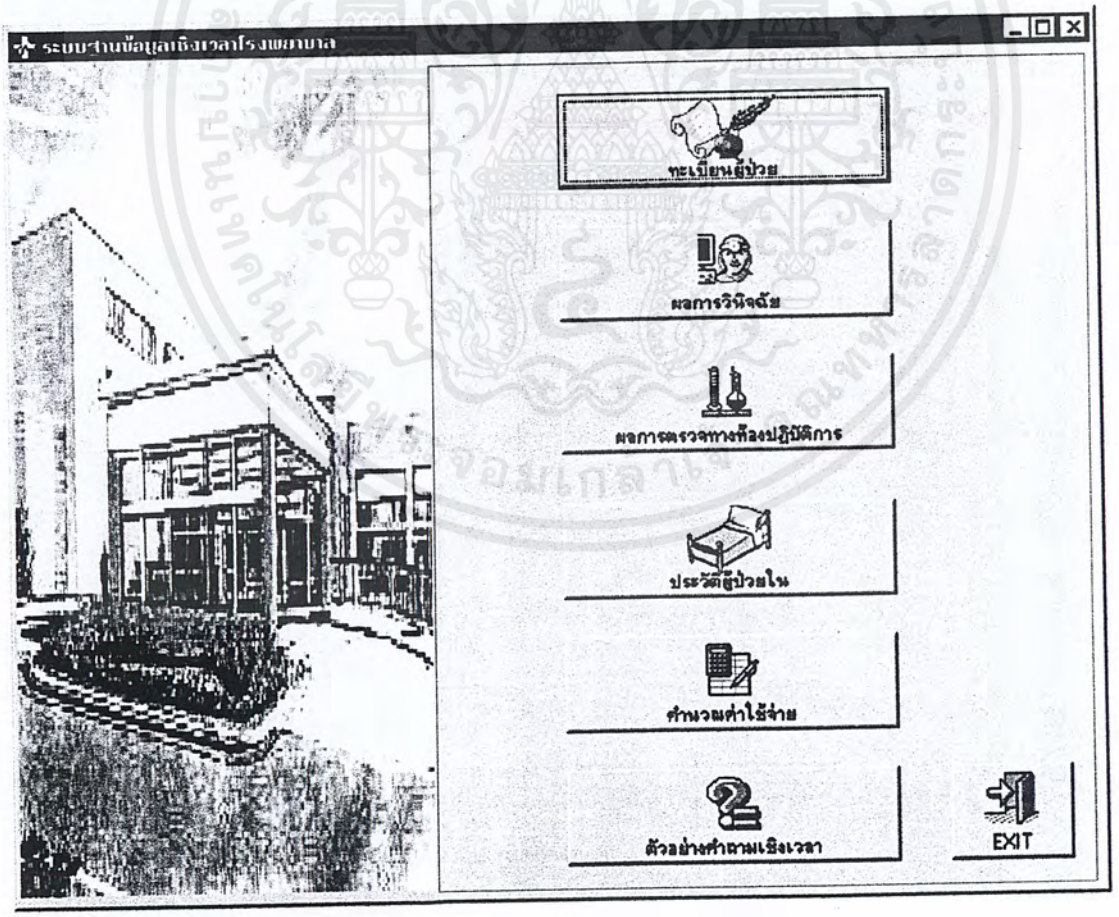
การใช้งานโปรแกรมระบบงานโรงพยาบาล

ฟังก์ชันงานของแอพพลิเคชั่นจะแบ่งออกเป็น 5 งาน ดังนี้ คือ

1. ทะเบียนประวัติผู้ป่วย
2. ผลการวินิจฉัย
3. ผลการตรวจทางห้องปฏิบัติการ
4. ทะเบียนประวัติผู้ป่วยใน
5. คำนวณค่าใช้จ่าย

นอกจากนี้ยังมีหน้าจอพิเศษที่ใช้ในการแสดงการตอบคำถามเชิงเวลาอีก 1 หน้าจอ

โดยหน้าจอหลักของแอพพลิเคชั่น เป็นดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

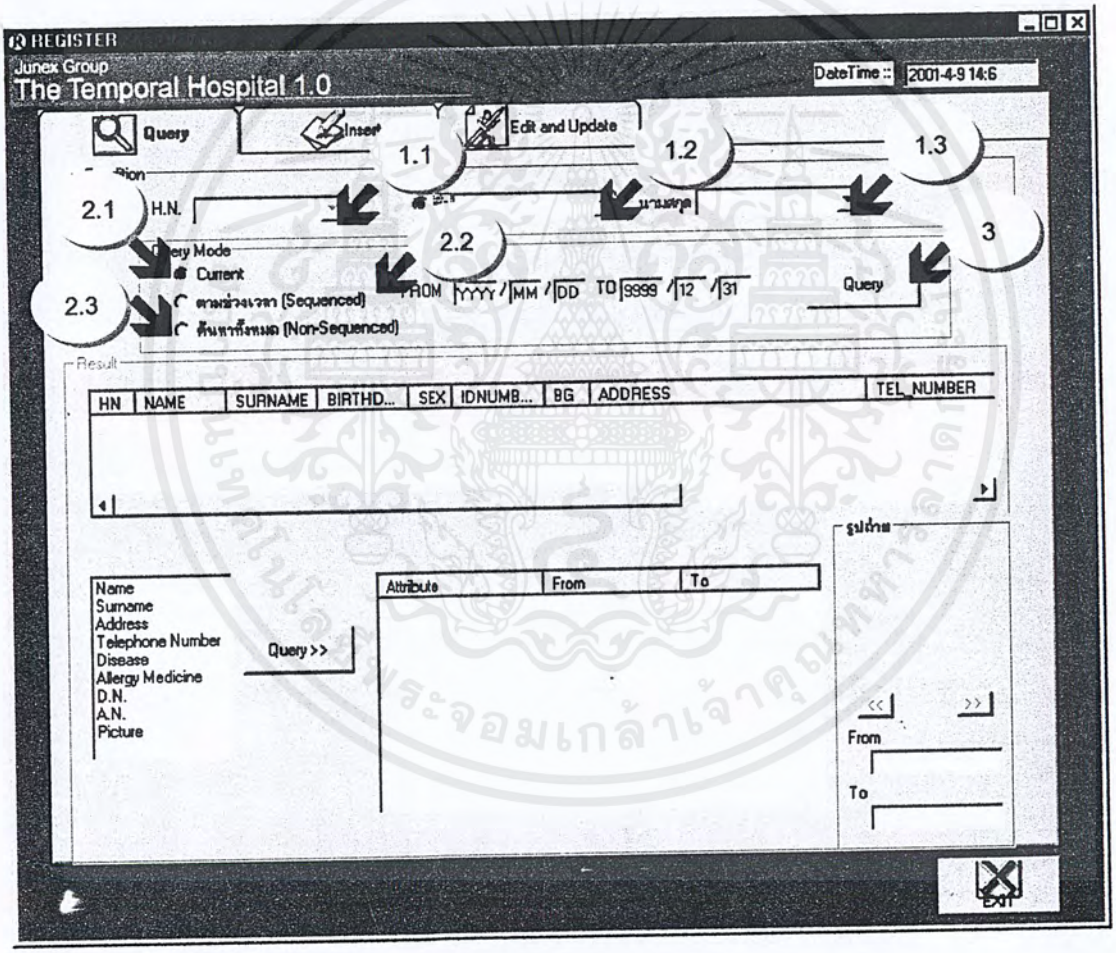
ประวัติผู้ป่วย

จะมีหน้าจอในการจัดการเกี่ยวกับประวัติผู้ป่วย 3 งานด้วยกันคือ

- หน้าจอการค้นหาผู้ป่วย (Query)
- หน้าจอการป้อนข้อมูลผู้ป่วยใหม่ (INSERT)
- หน้าจอการเปลี่ยนแปลงข้อมูล (Edit & Update)

หน้าจอการค้นหาผู้ป่วย (QUERY)

การใช้งาน



1. กำหนดเงื่อนไขเริ่มต้นในการค้นหา ซึ่งสามารถเลือกจากกรณีใดกรณีหนึ่งจาก 4 กรณี ดังนี้
   1.1 หมายเลขผู้ป่วย (HN ; Hospital Number)
   1.2 ชื่อผู้ป่วย (Name)
   1.3 นามสกุลผู้ป่วย (Surname)
   1.4 ชื่อและนามสกุลผู้ป่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กำหนดเงื่อนไขเวลาในการค้นหา ซึ่งสามารถเลือกจากกรณีใดกรณีหนึ่งจาก 3 กรณี ดังนี้

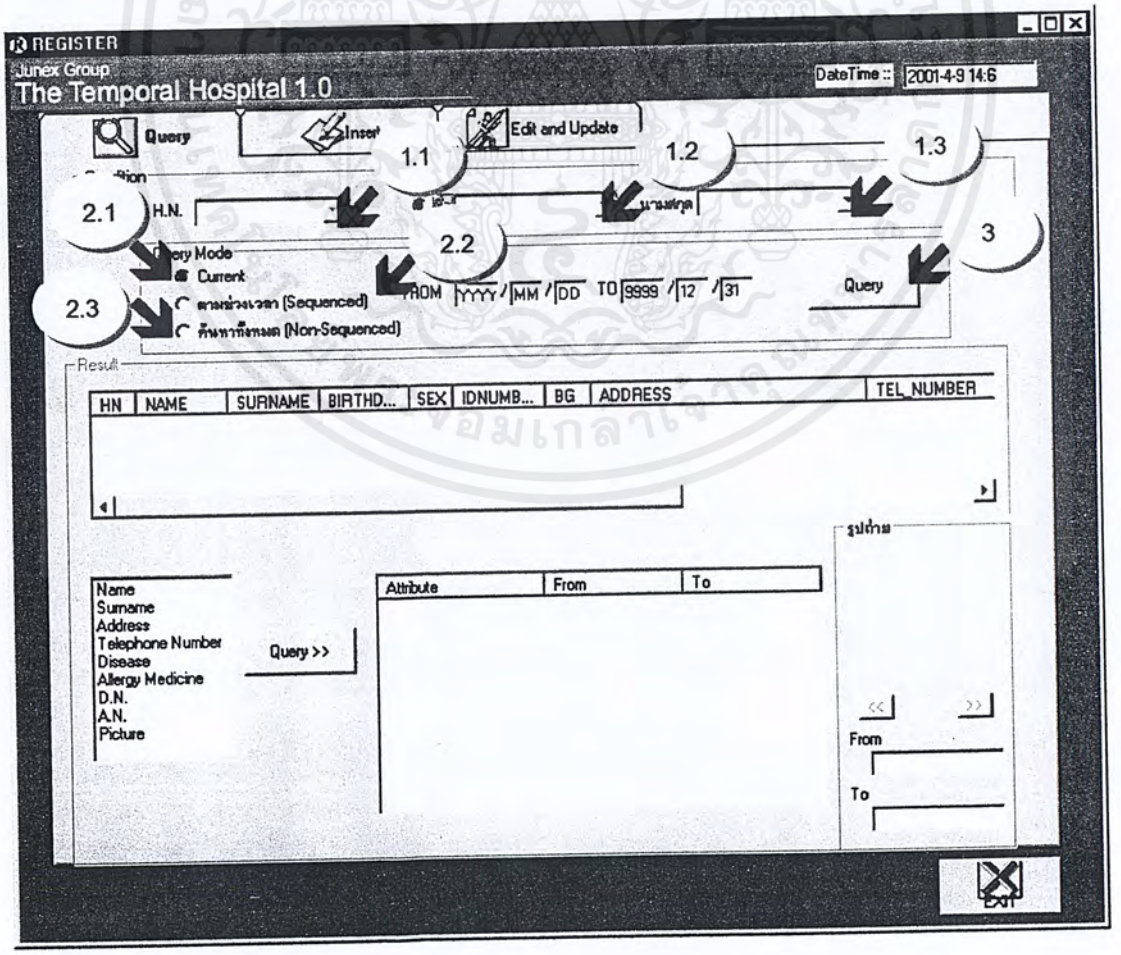
   2.1 Current เป็นการค้นหา ณ เวลาปัจจุบัน (เวลาปัจจุบันของระบบ)

   2.2 Sequenced เป็นการค้นหาตามช่วงเวลาที่เป็นเงื่อนไข ซึ่งต้องระบุช่วงเวลา ในรูปแบบของ ปี-เดือน-วัน (YYYY-MM-DD) ในช่อง FROM และ TO ซึ่งจะเป็นเงื่อนไขเวลาเริ่มต้นและเวลาสิ้นสุดที่ต้องการค้นหา ตามลำดับ

   2.3 Non-Sequenced เป็นการค้นหาโดยไม่ดูช่วงเวลาเป็นเงื่อนไขในการค้นหา เสมือนการค้นหาทั้งฐานข้อมูล ซึ่งจะทำให้หาประวัติของผู้ป่วยพบ แม้จะไม่ทราบช่วงเวลาที่ข้อมูลมีค่า เช่น การค้นหาผู้ป่วยชื่อสมชาย ซึ่งอาจจะเคยมีการเปลี่ยนชื่อไปแล้วหรือไม่ก็ไม่ทราบ ก็จะสามารถแสดงรายละเอียดของผู้ป่วยที่เคยหรือกำลังมีชื่อสมชายอยู่ทั้งหมดในโรงพยาบาลออกมา

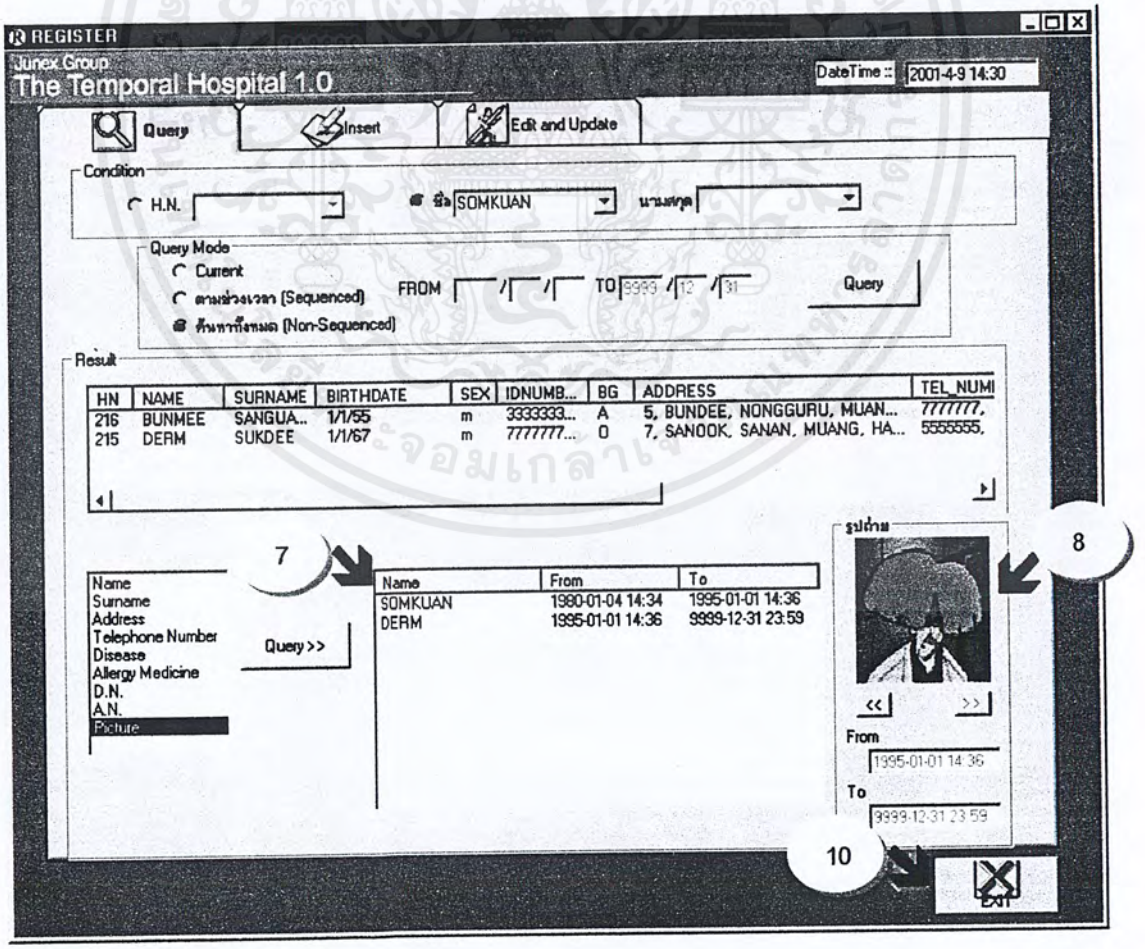
3. คลิกปุ่ม Query เพื่อทำการค้นหา

4. แอพพลิเคชั่นจะแสดงรายละเอียดของผู้ป่วย ณ ปัจจุบัน ตามเงื่อนไขเริ่มต้น และเงื่อนไขเวลาที่กำหนดออกมาทั้งหมด ซึ่งอาจจะมีผู้ป่วยที่มีข้อมูลตรงตามเงื่อนไขมากกว่า 1 ราย ก็จะแสดงข้อมูลปัจจุบันของผู้ป่วยแต่ละรายออกมาในตาราง โดยในแต่ละแถวจะหมายถึงผู้ป่วยคนละรายกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เลือกผู้ป่วยรายใดรายหนึ่งในตาราง เพื่อจะดูรายละเอียดที่เปลี่ยนแปลงตามเวลาของข้อมูลผู้ป่วยรายนั้น
6. เลือกแอตทริบิวต์ข้อมูลของผู้ป่วยที่ต้องการจะดูการเปลี่ยนแปลงตามเวลา ซึ่งมีแอตทริบิวต์ต่างๆ ดังนี้
   - 6.1 Name (ชื่อ)
   - 6.2 Surname (นามสกุล)
   - 6.3 Address (ที่อยู่)
   - 6.4 Telephone Number (หมายเลขโทรศัพท์)
   - 6.5 Disease (โรคที่เป็น)
   - 6.6 Allergy Medicine (ยาที่แพ้)
   - 6.7 D.N. (หมายเลขการวินิจฉัย)
   - 6.8 A.N. (หมายเลขการแอดมิด)
   - 6.9 Picture (รูปถ่าย)
7. เมื่อคลิกเลือกแอตทริบิวต์ใดแล้วจะแสดงข้อมูลของแอตทริบิวต์นั้น และเวลาที่ข้อมูลนั้นมีค่าในตาราง

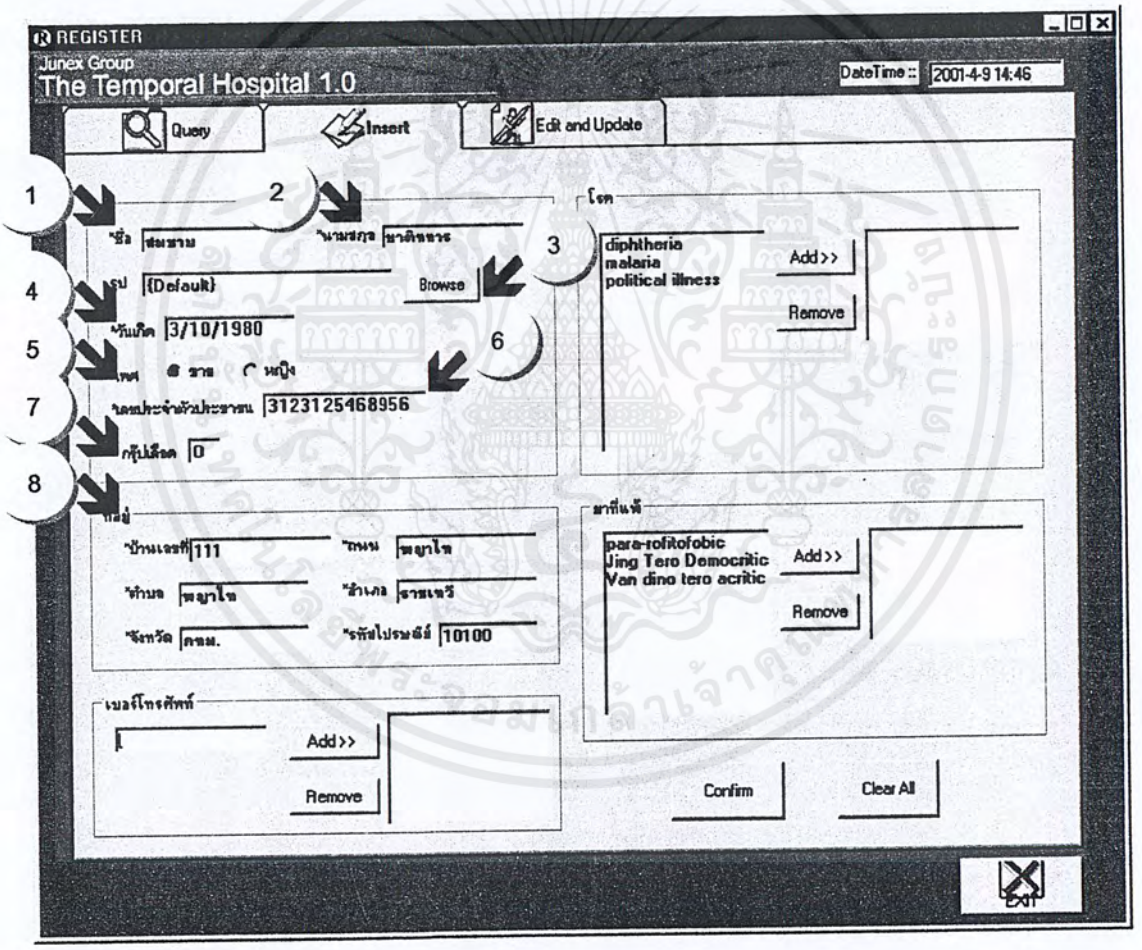


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ถ้าคลิกเลือกแอตทริบิวต์ Picture จะปรากฏรูปปัจจุบันของผู้ป่วยในกรอบ และแสดงช่วงเวลาที่รูปนั้นมีค่าในช่อง FROM และ TO
9. สามารถคลิกเลื่อนไปดูข้อมูลรูปของผู้ป่วยได้ โดยใช้ปุ่มดังนี้
   9.1 << เมื่อต้องการเลื่อนไปดูรูปก่อนรูปที่แสดงปัจจุบัน
   9.2 >> เมื่อต้องการเลื่อนไปดูรูปหลังรูปที่แสดงปัจจุบัน
10. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

หน้าจอการป้อนข้อมูลผู้ป่วยใหม่ (INSERT)

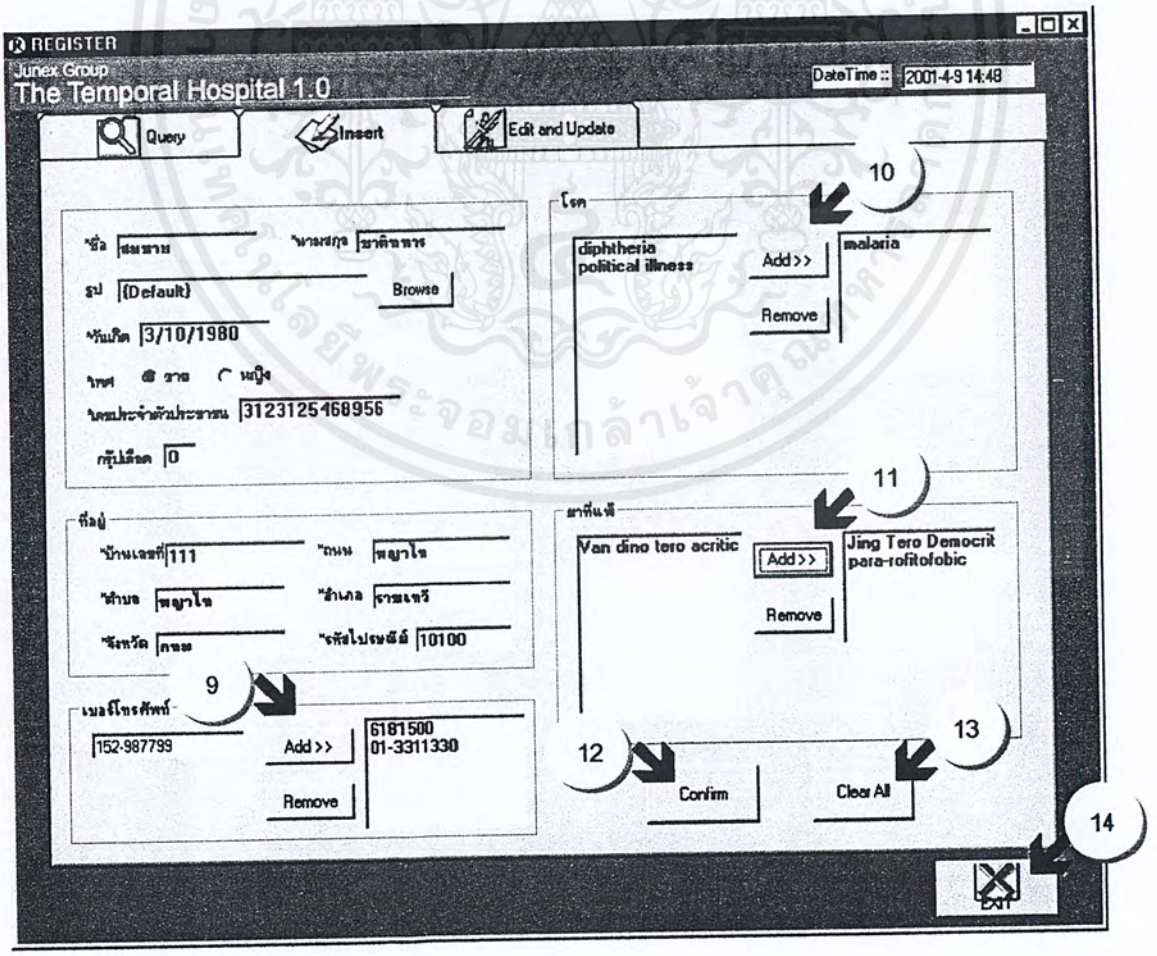
การใช้งาน



1. ป้อนชื่อผู้ป่วย
2. ป้อนนามสกุลผู้ป่วย
3. เลือกตำแหน่งที่เก็บรูปผู้ป่วยไว้ โดยคลิกที่ปุ่ม BROWSE แล้วเลือกหาไฟล์รูปภาพของผู้ป่วย หากไม่มีก็ไม่ต้องเลือกปุ่ม BROWSE เพราะแอพพลิเคชันจะตั้งค่ารูปเป็น Default อยู่แล้ว คือ หมายถึงไม่มีรูปถ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือกเพศของผู้ป่วย (จะตั้ง Default ไว้ที่ เพศชาย)
5. ป้อนวันเกิดของผู้ป่วย โดยต้องป้อนในรูปแบบ วัน/เดือน/ปี (DD/MM/YYYY)
6. ป้อนหมายเลขประชาชนของผู้ป่วย (ไม่เกิน 13 ตัว)
7. ป้อนกรุ๊ปเลือดของผู้ป่วย
8. ป้อนที่อยู่ของผู้ป่วย โดยมีแอตทริบิวต์ต่าง ๆ ดังนี้
   - 8.1 บ้านเลขที่ (เป็นตัวเลข)
   - 8.2 ถนน
   - 8.3 ตำบล
   - 8.4 อำเภอ
   - 8.5 จังหวัด
   - 8.6 รหัสไปรษณีย์ (เป็นตัวเลข)
9. ป้อนหมายเลขโทรศัพท์ของผู้ป่วย (ถ้ามี) เมื่อป้อนเสร็จให้คลิกปุ่ม ADD หมายเลขโทรศัพท์ที่ป้อนจะไปปรากฏทางขวา หากมีเบอร์โทรศัพท์อื่นๆ เพิ่มก็ป้อนใหม่เข้าไป แล้วคลิกปุ่ม ADD เช่นเดียวกัน หากต้องการเอาเบอร์โทรศัพท์ใดออก ให้คลิกเลือกเบอร์โทรศัพท์นั้น แล้วคลิกปุ่ม REMOVE

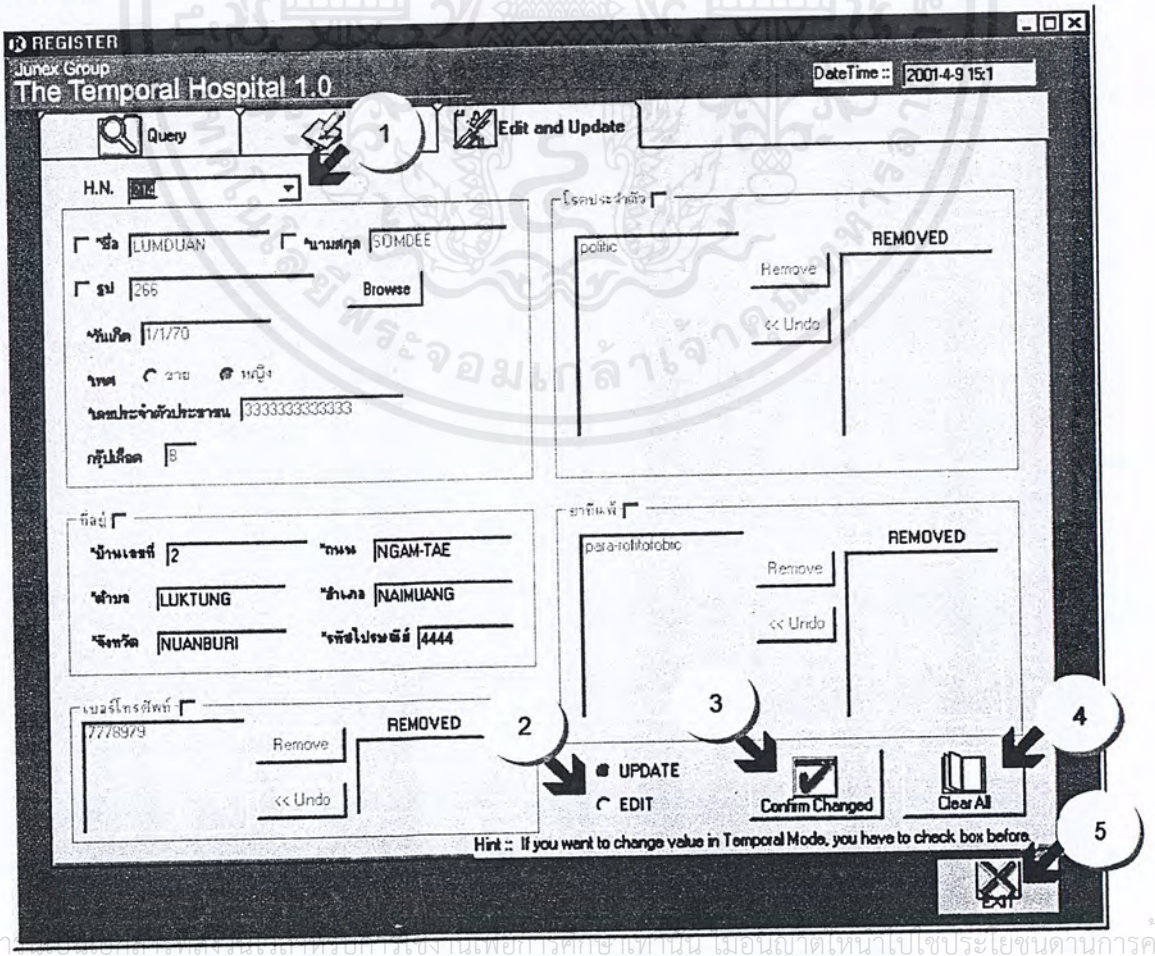


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เลือกโรคที่เป็นของผู้ป่วย (ถ้ามี) จากรายการโรคทางซ้าย ซึ่งจะแสดงชื่อโรคทั้งหมด เมื่อเลือกแล้วให้คลิกปุ่ม ADD ชื่อโรคจะไปปรากฏอยู่ทางขวา หากมีโรคอื่นๆ เพิ่มก็ป้อนใหม่เข้าไป แล้วคลิกปุ่ม ADD เช่นเดียวกัน หากต้องการเอาโรคใดออก ให้คลิกเลือกโรคนั้นแล้วคลิกปุ่ม REMOVE
11. เลือกยาที่แพ้ที่เป็นของผู้ป่วย (ถ้ามี) จากรายการยาที่แพ้ทางซ้าย ซึ่งจะแสดงชื่อยาที่แพ้ทั้งหมด เมื่อเลือกแล้วให้คลิกปุ่ม ADD ชื่อยาที่แพ้จะไปปรากฏอยู่ทางขวา หากมียาที่แพ้อื่นๆ เพิ่มก็ป้อนใหม่เข้าไป แล้วคลิกปุ่ม ADD เช่นเดียวกัน หากต้องการเอายาที่แพ้ใดออก ให้คลิกเลือกยาที่แพ้นั้น แล้วคลิกปุ่ม REMOVE
12. เมื่อใส่ข้อมูลเรียบร้อยแล้ว ให้คลิกปุ่ม Confirm
หากมีแอตทริบิวต์ใดที่จำเป็นต้องมีค่า ไม่ได้ป้อนข้อมูลไว้ แอพพลิเคชั่นจะขึ้นข้อความแจ้งว่าข้อมูลไม่ครบ แต่ถ้ามีการบันทึกข้อมูลเรียบร้อยแล้ว ปุ่ม Confirm จะ Disable ไป
13. ถ้าต้องการเคลียร์ค่าข้อมูลต่างๆ ที่ป้อนไว้ หรือต้องการเคลียร์ค่าข้อมูลต่างๆ ที่บันทึกลงไปแล้ว เพื่อทำการป้อนรายละเอียดของผู้ป่วยรายใหม่ ให้คลิกที่ Clear All
14. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

หน้าจอการเปลี่ยนแปลงข้อมูล (Edit & Update)

การใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการเปลี่ยนแปลงข้อมูล โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. เลือกเงื่อนไขในการเปลี่ยนแปลงข้อมูล ซึ่งมี 2 กรณี คือ
   2.1 Update เป็นการเปลี่ยนแปลงข้อมูลเชิงเวลา ซึ่งจะเป็นการบันทึกค่าข้อมูลใหม่และช่วงเวลาที่ข้อมูลนั้นมีค่าจากปัจจุบันไปจนถึงตลอดไป และยังคงเก็บค่าข้อมูลเก่าไว้ แต่จะเปลี่ยนวันสุดท้าย (TO) ที่ข้อมูลนั้นมีค่าจากตลอดไปเป็นวันปัจจุบัน ให้อัตโนมัติ
   2.2 Edit เป็นการแก้ไขข้อมูล ซึ่งจะเป็นการแก้ค่าข้อมูลทับข้อมูลเดิม โดยไม่มีการเปลี่ยนแปลงเชิงเวลาเลย และไม่มีการเก็บค่าข้อมูลเก่าไว้ ใช้ในกรณีการบันทึกข้อมูลผิดพลาด
3. เมื่อเลือกเงื่อนไขในการเปลี่ยนแปลงข้อมูลแล้ว ถ้าเป็นการ Update จะต้องทำการ Checkbox เลือกแอตทริบิวต์เชิงเวลาที่ต้องการเปลี่ยนค่าก่อน แล้วจึงทำการเปลี่ยนแปลงค่านั้น แต่ถ้าเป็นการ Edit จะสามารถแก้ไขได้ทุกแอตทริบิวต์ และจะทำการ Checkbox เลือกการเปลี่ยนแปลงแอตทริบิวต์ทั้งหมดให้โดยอัตโนมัติ
   เมื่อทำการเปลี่ยนแปลงเรียบร้อยแล้ว คลิกปุ่ม Confirm Changed เพื่อยืนยันการเปลี่ยนแปลง แต่ถ้ามีการบันทึกข้อมูลเรียบร้อยแล้ว ปุ่ม Confirm Changed จะ Disable ไป
4. ถ้าต้องการเคลียร์ค่าข้อมูลต่างๆ ที่ป้อนไว้ หรือต้องการเคลียร์ค่าข้อมูลต่างๆ ที่บันทึกการแก้ไขลงไปแล้ว เพื่อทำการแก้ไขรายละเอียดของผู้ป่วยรายใหม่ ให้คลิกที่ Clear All
5. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

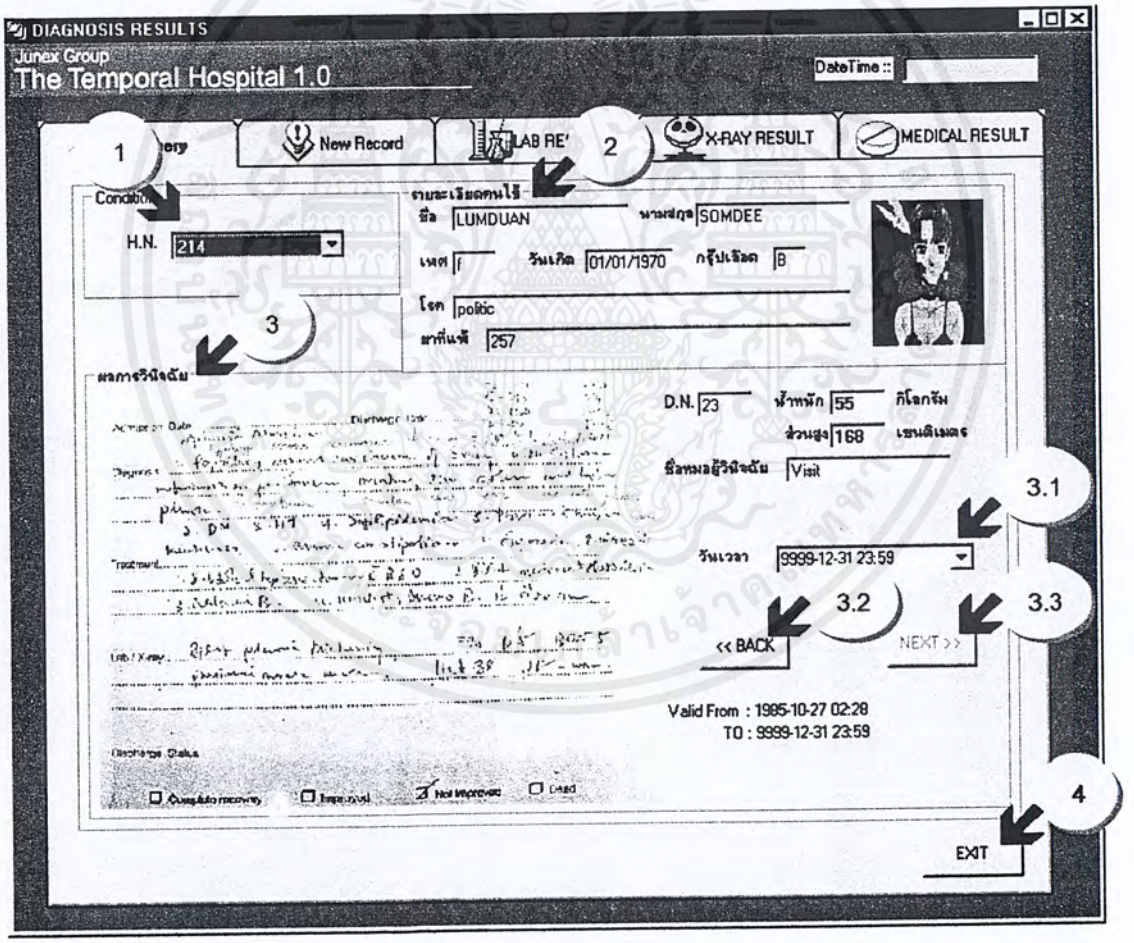
ผลการวินิจฉัย

จะมีหน้าจอในการจัดการเกี่ยวกับผลการวินิจฉัยผู้ป่วย 4 งานด้วยกันคือ

- หน้าจอการค้นหาผลการวินิจฉัย (Query)
- หน้าจอการป้อนข้อมูลผลการวินิจฉัยใหม่ (NEW RECORD)
- หน้าจอการค้นหาผลทางห้องปฏิบัติการ (LAB RESULT)
- หน้าจอการค้นหาผลเอ็กซเรย์ (X-RAY RESULT)
- หน้าจอการค้นหาผลการสั่งยา (MEDICAL RESULT)

หน้าจอการค้นหาผลการวินิจฉัย (QUERY)

การใช้งาน



1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดงรายละเอียดของผู้ป่วยแบบปัจจุบัน เพื่อยันยันความถูกต้องของผู้ป่วย
3. แอพพลิเคชั่นจะแสดงรายละเอียดผลการวินิจฉัยปัจจุบันของผู้ป่วย พร้อมทั้งหมายเลขการวินิจฉัย (DN) น้ำหนัก ส่วนสูง ชื่อหมอผู้วินิจฉัย และวันเวลาในการวินิจฉัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 เลือกคอมโบ้บ๊อกซ์วันเวลา เมื่อต้องการไปดูผลการวินิจฉัยวันที่ต้องการ

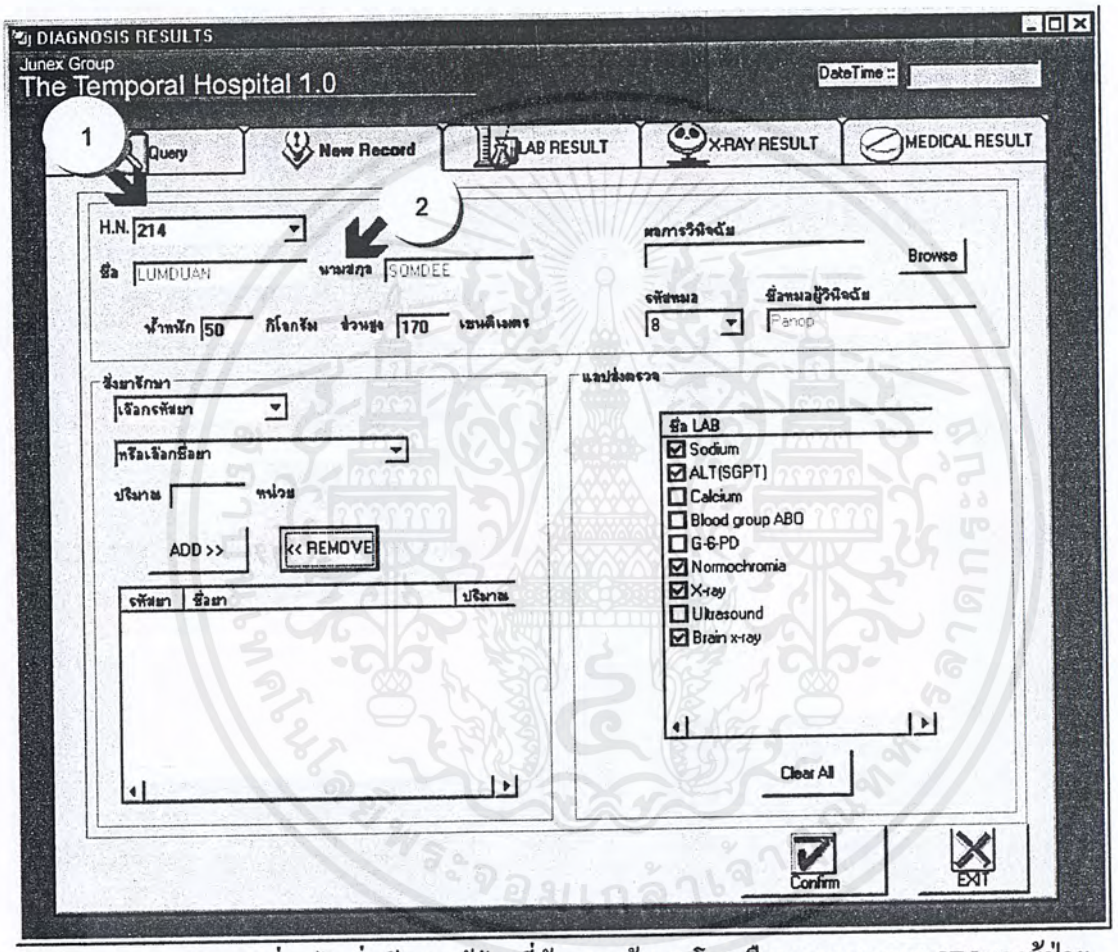
3.2 กดปุ่ม << เมื่อต้องการเลื่อนไปดูผลการวินิจฉัยก่อนผลการวินิจฉัยที่แสดงปัจจุบัน

3.3 กดปุ่ม >> เมื่อต้องการเลื่อนไปดูผลการวินิจฉัยหลังผลการวินิจฉัยที่แสดงปัจจุบัน

4. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

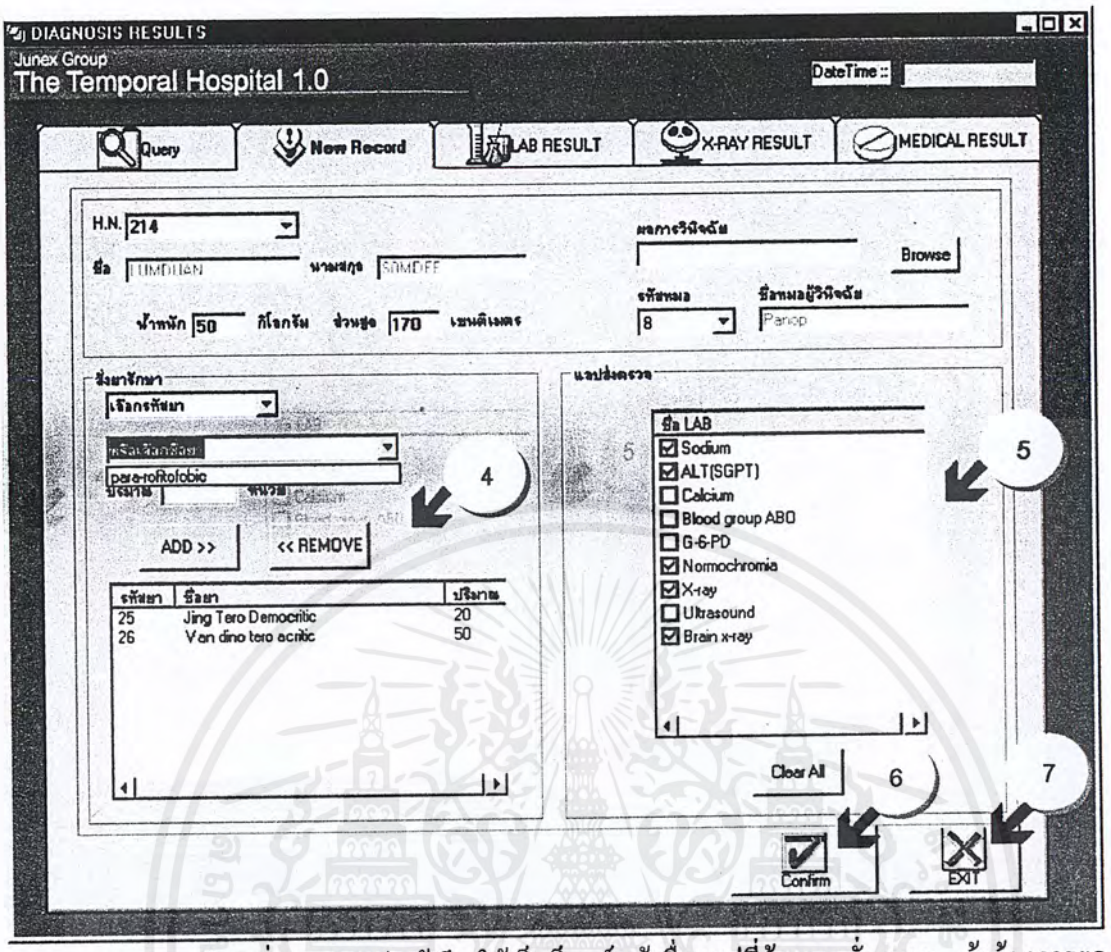
หน้าจอการป้อนข้อมูลผลการวินิจฉัยใหม่ (NEW RECORD)

การใช้งาน



1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดงชื่อและนามสกุลแบบปัจจุบันของผู้ป่วยเพื่อยืนยันความถูกต้อง
3. ป้อนน้ำหนัก ส่วนสูง ผลการวินิจฉัย และรหัสหมอ (ต้องมี) เพื่อบันทึกผลการวินิจฉัยของผู้ป่วยรายนั้น ผลการวินิจฉัยจะต้องเป็นไฟล์รูปภาพ ซึ่งต้องเลือกตำแหน่งที่เก็บไฟล์รูปภาพนั้น
4. หากมีการสั่งยารักษา (ถ้ามี) โดยเลือกจากคอมโบ้บ๊อกซ์รหัสยา หรือชื่อยาก็ได้ แล้วป้อนรหัสยา แล้วกดปุ่ม ADD เพื่อเพิ่มชื่อยาและปริมาณ ในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



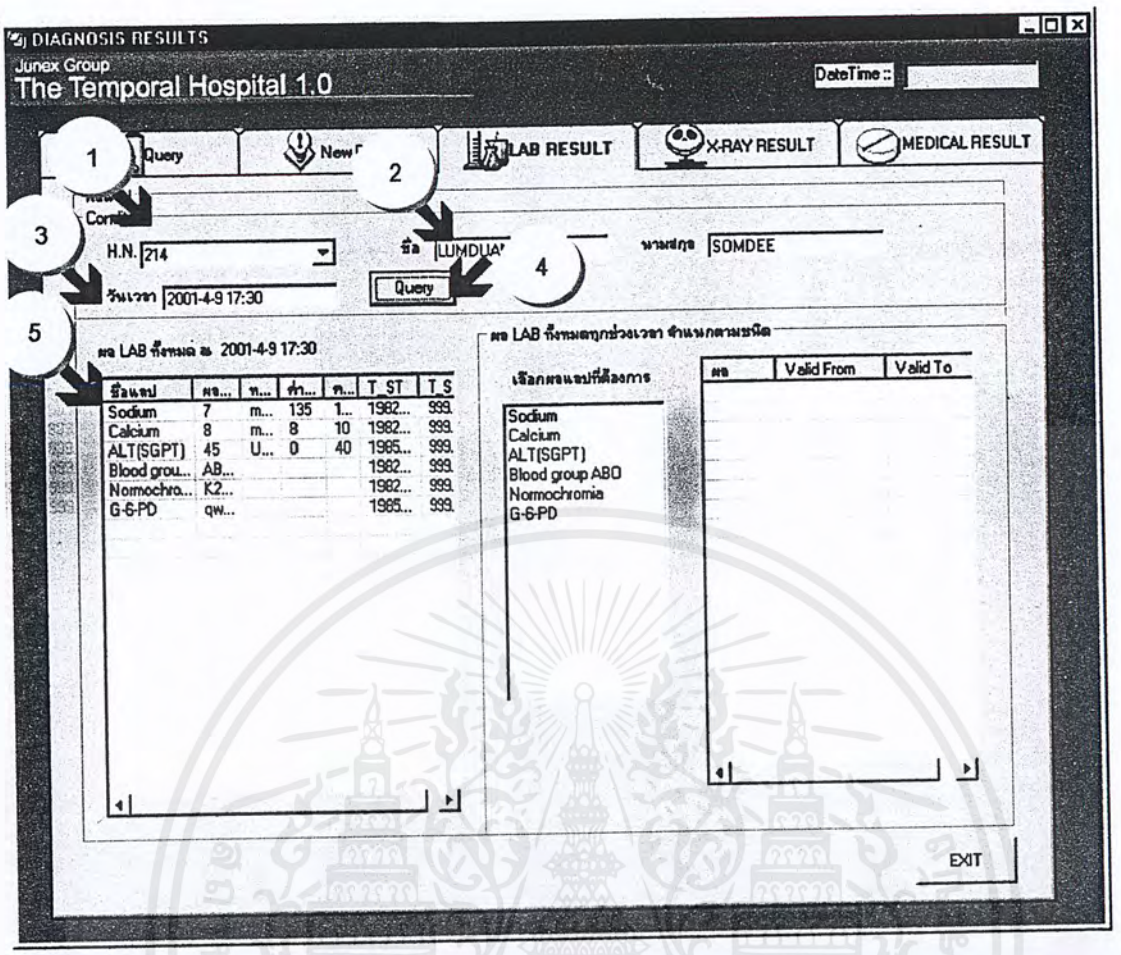
5. หากมีการสั่งตรวจแลป (ถ้ามี) ให้เช็คบ็อกซ์หน้าชื่อแลปที่ต้องการสั่งตรวจ ถ้าต้องการยกเลิกการสั่งตรวจแลปทุกชนิด ให้คลิกที่ปุ่ม Clear All
6. ต้องการยืนยันการบันทึกผลการวินิจฉัย การสั่งยารักษา (ถ้ามี) การสั่งตรวจแลป (ถ้ามี) ให้คลิกปุ่ม Confirm
7. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

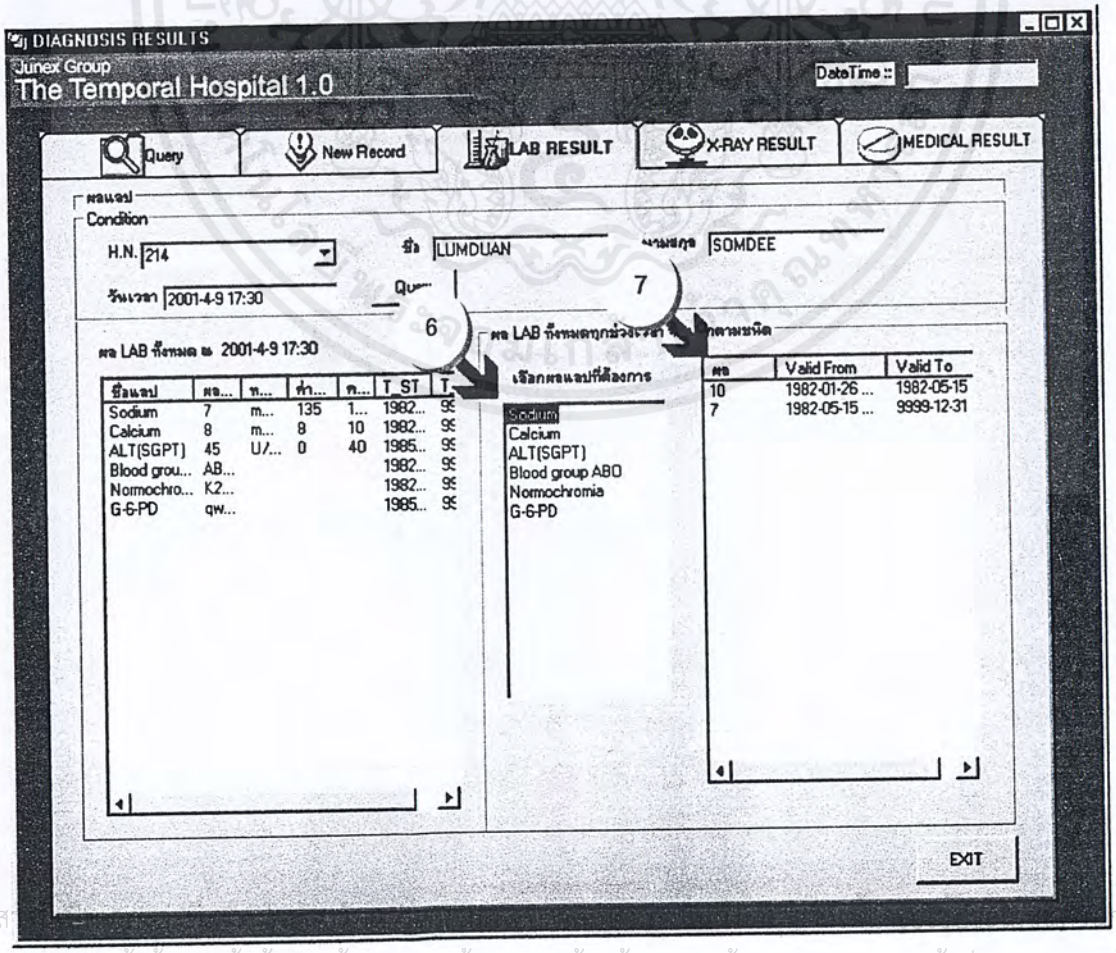
หน้าจอการค้นหาผลทางห้องปฏิบัติการ (LAB RESULT)

การใช้งาน

1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดงชื่อและนามสกุลแบบปัจจุบันของผู้ป่วยเพื่อยืนยันความถูกต้อง
3. กำหนดเงื่อนไขเชิงเวลาที่ช่องวันเวลา หมายถึง ต้องการดูผลทางห้องปฏิบัติการที่มีค่า ณ วันที่กำหนด ซึ่งวันที่จะต้องอยู่ในรูปแบบ [ปี –เดือน-วัน ชั่วโมง:นาที] ; [YYYY-MM-DD HH:MM] และแอพพลิเคชั่นจะตั้งให้เป็นวันที่ปัจจุบันอยู่แล้ว หมายถึง ต้องการดูผลทางห้องปฏิบัติการที่มีค่าอยู่ ณ วันปัจจุบัน
4. คลิกปุ่ม QUERY เพื่อทำการค้นหาผลทางห้องปฏิบัติการตามเงื่อนไขที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



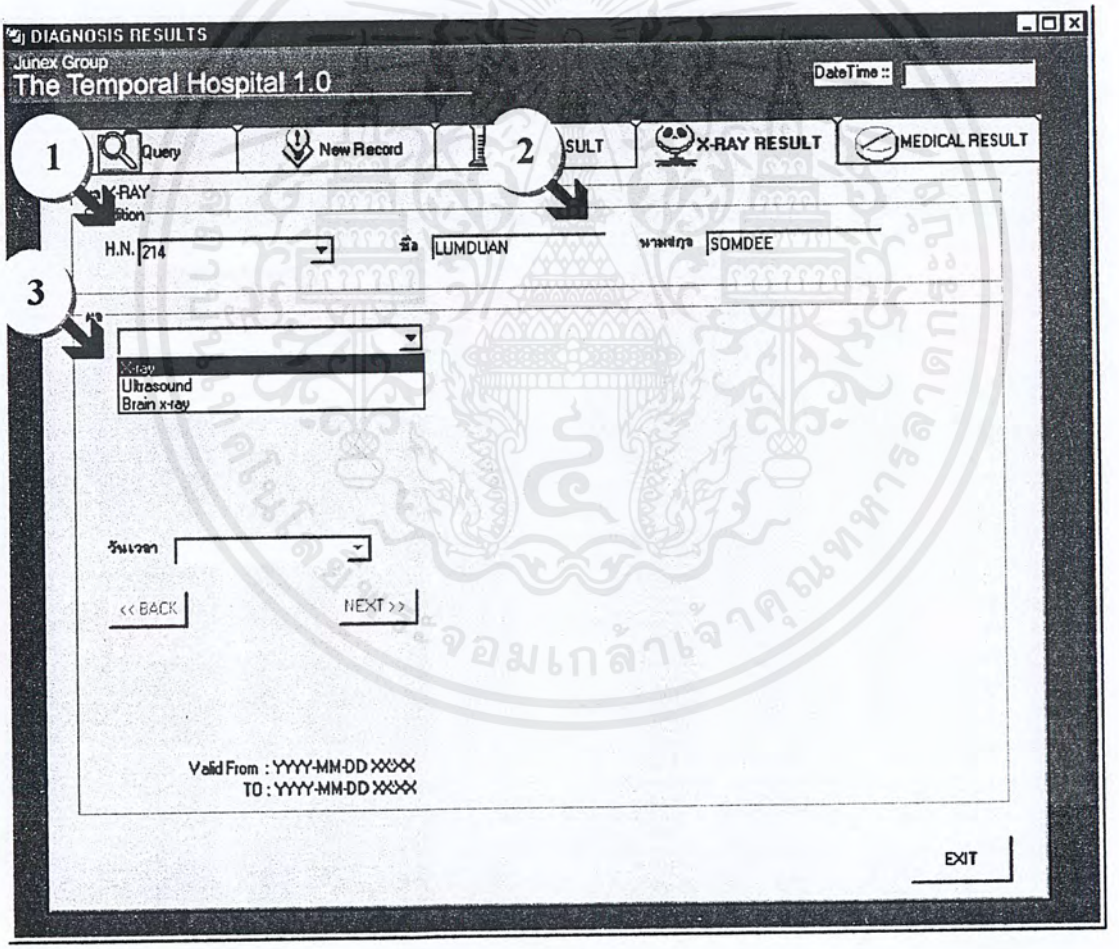


เอกสาร การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. แอพพลิเคชั่นจะแสดงชื่อแลป ผล หน่วย ค่าต่ำสุด ค่าสูงสุด ช่วงเวลาที่มีค่า ซึ่งจะแสดงค่าแลปของผู้ป่วยทุกชนิดที่มีค่าอยู่ตามเงื่อนไขเชิงเวลาที่กำหนด
6. ในลิสต์จะแสดงชื่อการสั่งตรวจทางห้องปฏิบัติการทั้งหมดที่เคยมีการสั่งตรวจและมีค่าแล้วของผู้ป่วย
7. แอพพลิเคชั่นจะแสดงผลทางห้องปฏิบัติการของชื่อการสั่งตรวจทางห้องปฏิบัติการนั้นทั้งหมด รวมถึงในอดีตในตาราง ซึ่งจะแสดงผลและช่วงเวลาที่ผลนั้นมีค่าด้วย
8. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

หน้าจอการค้นหาผลเอ็กซเรย์ (X-RAY RESULT)

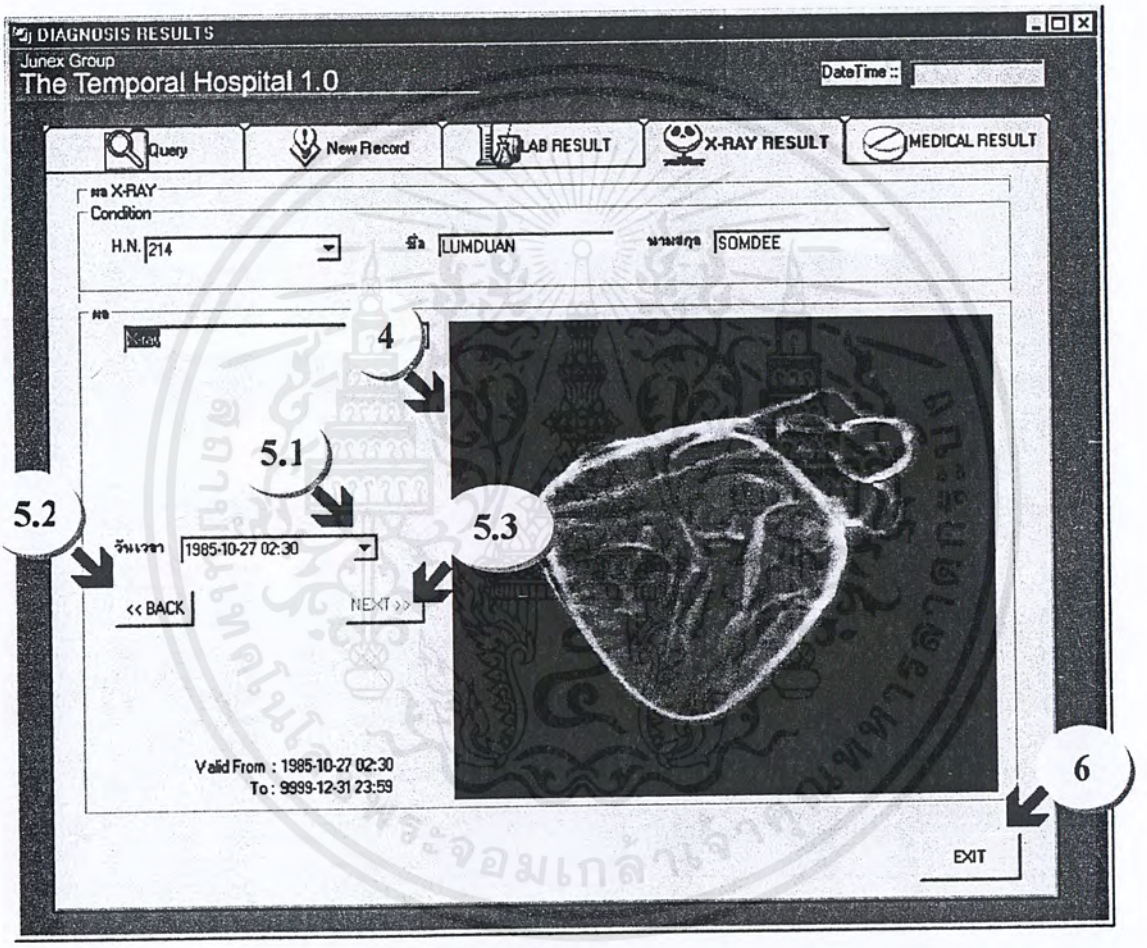
การใช้งาน



1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดงชื่อและนามสกุลแบบปัจจุบันของผู้ป่วยเพื่อยืนยันความถูกต้อง
3. เลือกชื่อผลเอ็กซเรย์ที่ต้องการค้นหา ซึ่งจะมีเฉพาะชื่อเอ็กซเรย์ที่ผู้ป่วยเคยตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ระบบจะแสดงผลเอ็กซเรย์ชนิดนั้นครั้งล่าสุดออกมา ซึ่งจะมี Valid From และ To เป็นตัวบ่งบอกช่วงเวลาที่ข้อมูลมีค่า
5. สามารถเลื่อนไปดูผลเอ็กซเรย์ย้อนหลังได้
   5.1 เลือกคอมโบ้บ๊อกซ์วันเวลา เมื่อต้องการไปดูผลเอ็กซเรย์วันที่ต้องการ
   5.2 กดปุ่ม << เมื่อต้องการเลื่อนไปดูผลเอ็กซเรย์ก่อนผลการวินิจฉัยที่แสดงปัจจุบัน
   5.3 กดปุ่ม >> เมื่อต้องการเลื่อนไปดูผลเอ็กซเรย์หลังผลการวินิจฉัยที่แสดงปัจจุบัน
6. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

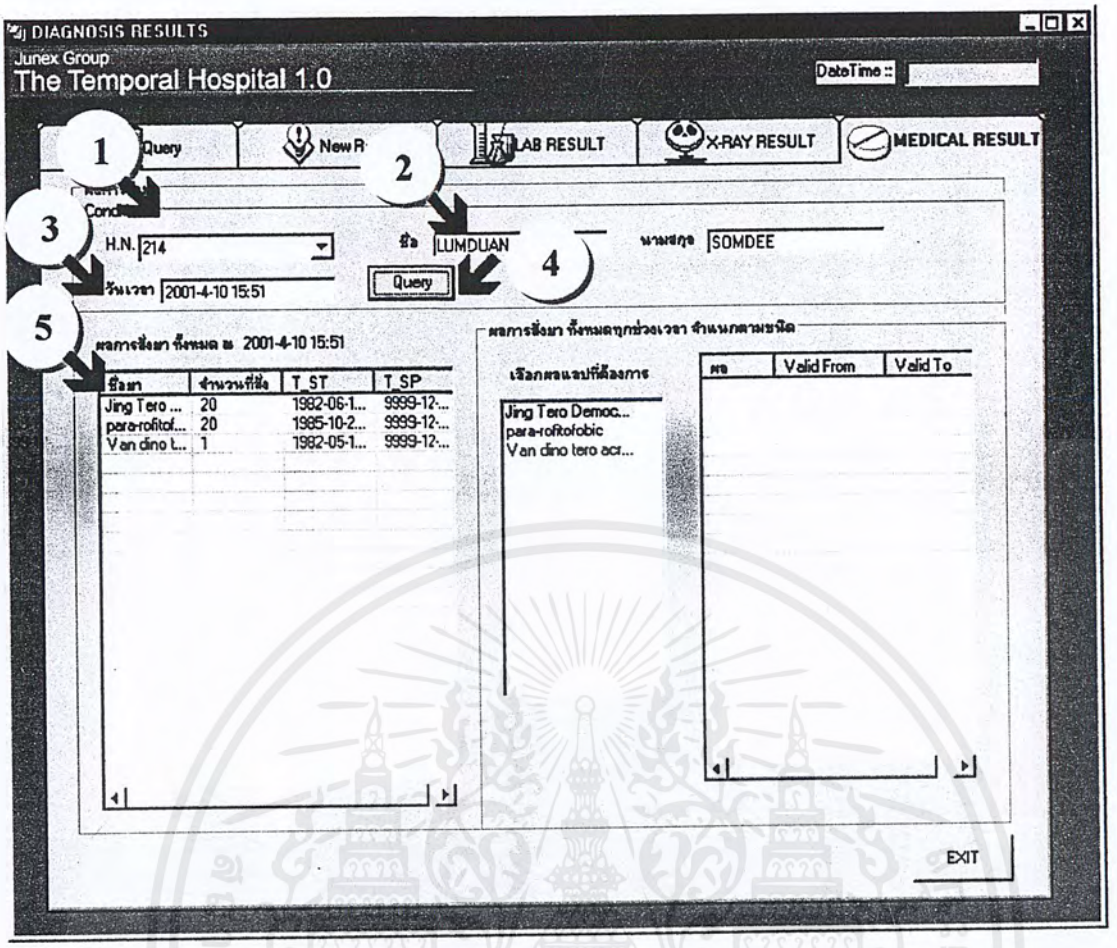


หน้าจอการค้นหาผลการสั่งยา (MEDICAL RESULT)

การใช้งาน

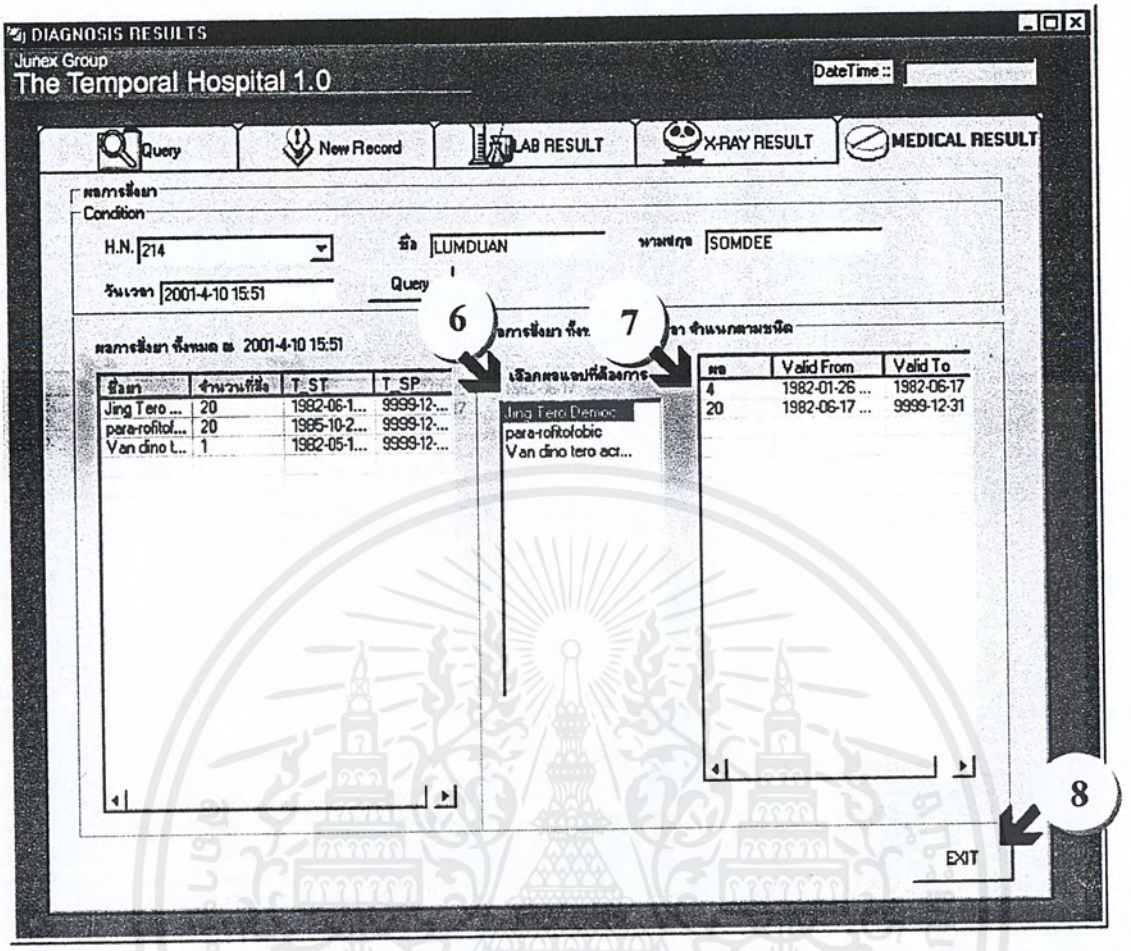
1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดงชื่อและนามสกุลแบบปัจจุบันของผู้ป่วยเพื่อยืนยันความถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3. กำหนดเงื่อนไขเชิงเวลาที่ช่องวันเวลา หมายถึง ต้องการดูผลการสั่งยาที่มีค่า ณ วันที่กำหนด ซึ่งวันที่จะต้องอยู่ในรูปแบบ [ปี –เดือน-วัน ชั่วโมง:นาที] ; [YYYY-MM-DD HH:MM] และแอพพลิเคชั่นจะตั้งให้เป็นวันที่ปัจจุบันอยู่แล้ว หมายถึง ต้องการดูผลการสั่งยาที่มีค่าอยู่ ณ วันปัจจุบัน
4. คลิกปุ่ม QUERY เพื่อทำการค้นหาผลการสั่งยาตามเงื่อนไขที่กำหนด
5. แอพพลิเคชั่นจะแสดงชื่อยา จำนวนที่สั่ง ช่วงเวลาที่มีค่า ซึ่งจะแสดงผลการสั่งยาของผู้ป่วยทุกชนิดที่มีค่าอยู่ตามเงื่อนไขเชิงเวลาที่กำหนด
6. ในลิสต์จะแสดงชื่อยาทั้งหมดที่เคยมีการสั่งของผู้ป่วย
7. แอพพลิเคชั่นจะแสดงผลการสั่งยาของชื่อยานั้นทั้งหมด รวมถึงในอดีตในตาราง ซึ่งจะแสดงผลการสั่งและช่วงเวลาที่ผลนั้นมีค่าด้วย
8. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

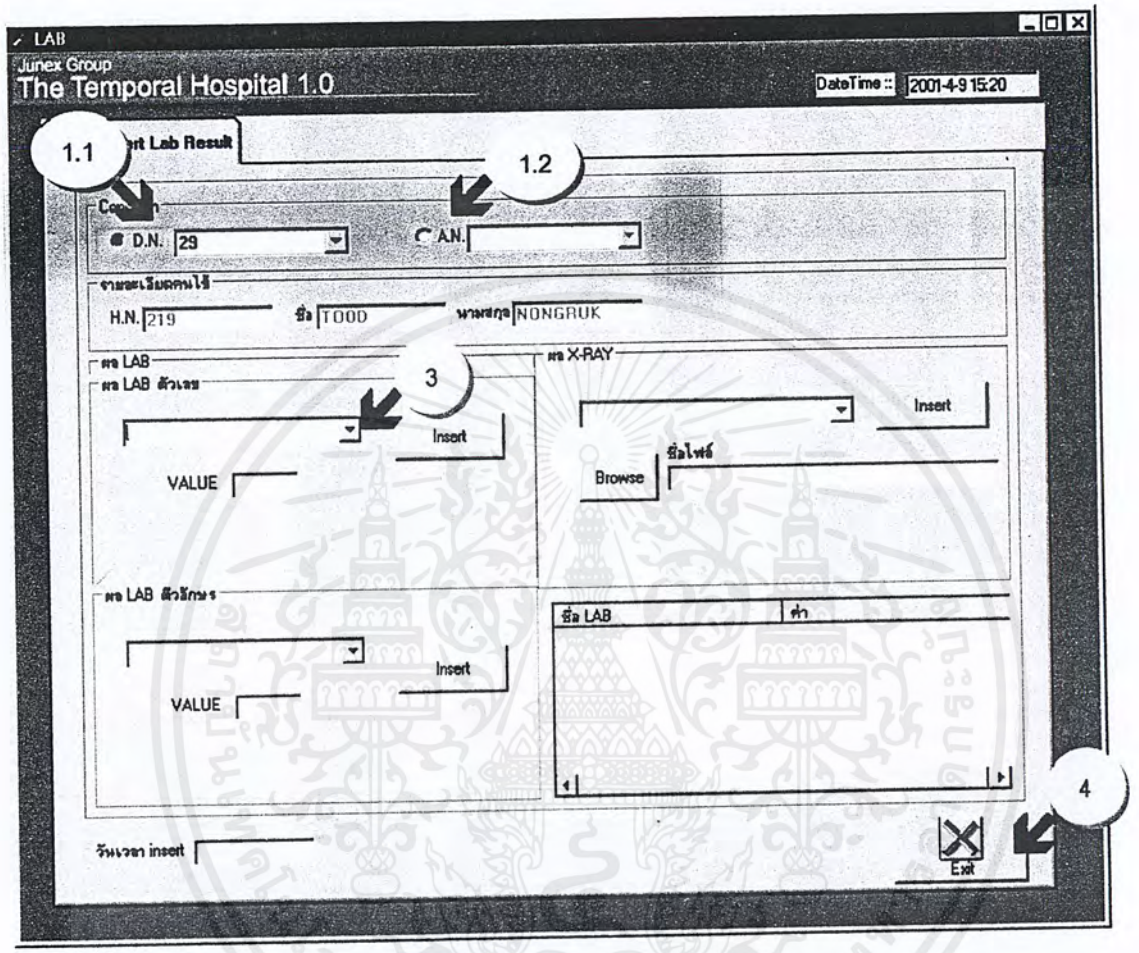
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลทางห้องปฏิบัติการ

การใช้งาน



1. เลือกบันทึกผลทางห้องปฏิบัติการของผู้ป่วย ได้ 2 กรณี ดังนี้
   1.1 เลือกจากหมายเลขการวินิจฉัย (DN)
   1.2 เลือกจากหมายเลขผู้ป่วยใน (AN)
2. แอพพลิเคชันจะแสดงรายละเอียดปัจจุบันของผู้ป่วยรายนั้น เพื่อยืนยันว่าผู้บันทึกต้องการบันทึกผลทางห้องปฏิบัติการให้แก่ผู้ป่วยรายนั้น แต่หากไม่มีการสั่งตรวจทางห้องปฏิบัติการ จะขึ้นข้อความแจ้งเตือน
3. แอพพลิเคชันจะแสดงชื่อการตรวจทางห้องปฏิบัติการ เฉพาะที่สั่งให้แก่ผู้ป่วยรายนั้นในคอมโบบ๊อกซ์ ซึ่งจะแยกประเภทตามชนิดของผลที่จะบันทึก ซึ่งจะเป็นตัวอักษร ตัวเลข และไฟล์รูปภาพ ซึ่งจะต้องเลือกชื่อการตรวจทางห้องปฏิบัติการก่อน แล้วใส่ผลตรวจ แล้วจึงกดปุ่ม INSERT ผลตรวจทางห้องปฏิบัติการของผู้ป่วยที่ได้รับการบันทึกแล้วจะไม่ปรากฏในคอมโบบ๊อกซ์นั้นอีกต่อไป
4. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

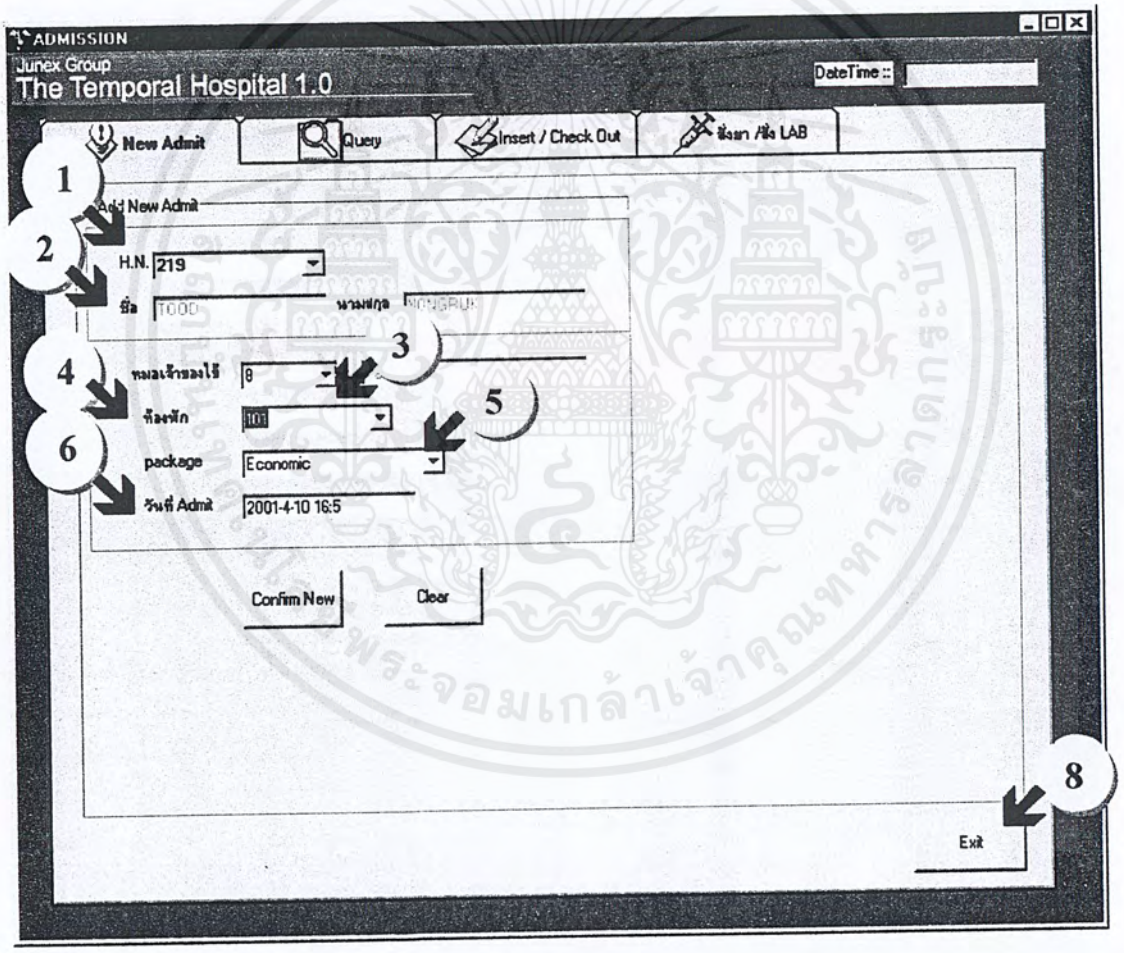
ประวัติผู้ป่วยใน

จะมีหน้าจอในการจัดการเกี่ยวกับประวัติผู้ป่วยใน 4 งานด้วยกันคือ

- หน้าจอการแอดมิดผู้ป่วยใหม่ (ADMIT)
- หน้าจอการค้นหาประวัติผู้ป่วยใน (QUERY)
- หน้าจอการบันทึกประวัติผู้ป่วยในและเช็คเอาท์ (INSERT & CHECK OUT)
- หน้าจอการสั่งยา/สั่งแลป

หน้าจอการแอดมิดผู้ป่วยใหม่ (ADMIT)

การใช้งาน



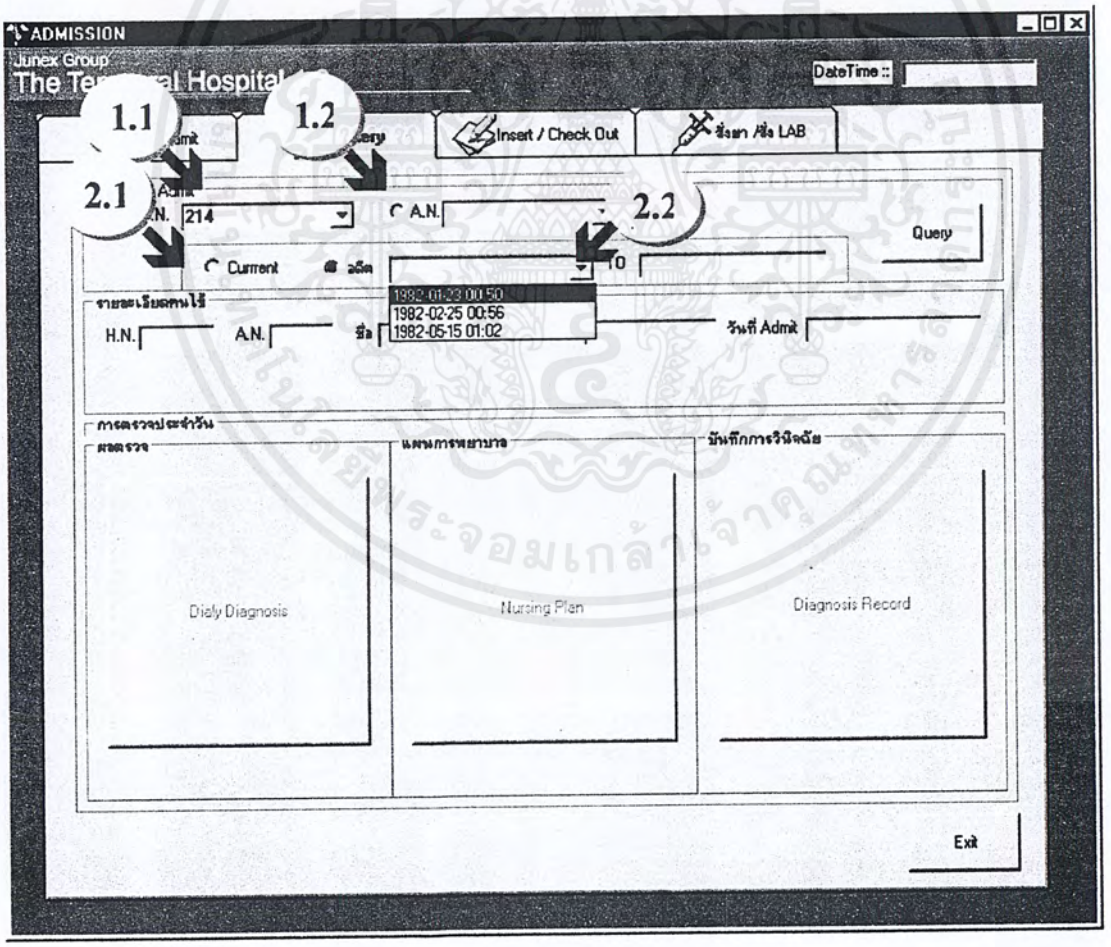
1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข HN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดงชื่อและนามสกุลแบบปัจจุบันของผู้ป่วยเพื่อยืนยันความถูกต้อง
3. เลือกรหัสหมอเจ้าของไข้ของผู้ป่วย จากคอมโบ้บ๊อกซ์ แอพพลิเคชั่นจะแสดงชื่อหมอให้โดยอัตโนมัติ
4. เลือกหมายเลขห้องพัก จากคอมโบ้บ๊อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เลือกแพกเกจการเข้าพักรักษา ซึ่งหมายถึงลักษณะการรักษา จากคอมโบ้บ๊อกซ์
6. ระบุวันที่แอดมิด ซึ่งจะสามารถกำหนดวันที่ทำการแอดมิดได้ ในรูปแบบ [ปี-เดือน-วัน ชั่วโมง:นาที] ; [YYYY-MM-DD HH:MM] ซึ่งแอพพลิเคชั่นจะตั้งเป็นวันที่ปัจจุบันให้โดยอัตโนมัติ
7. เมื่อป้อนข้อมูลเรียบร้อยแล้วให้คลิกปุ่ม Confirm New เพื่อยืนยันการแอดมิด แต่หากต้องการยกเลิกการแอดมิด หรือทำการแอดมิดผู้ป่วยรายใหม่ ให้คลิกปุ่ม Clear
8. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

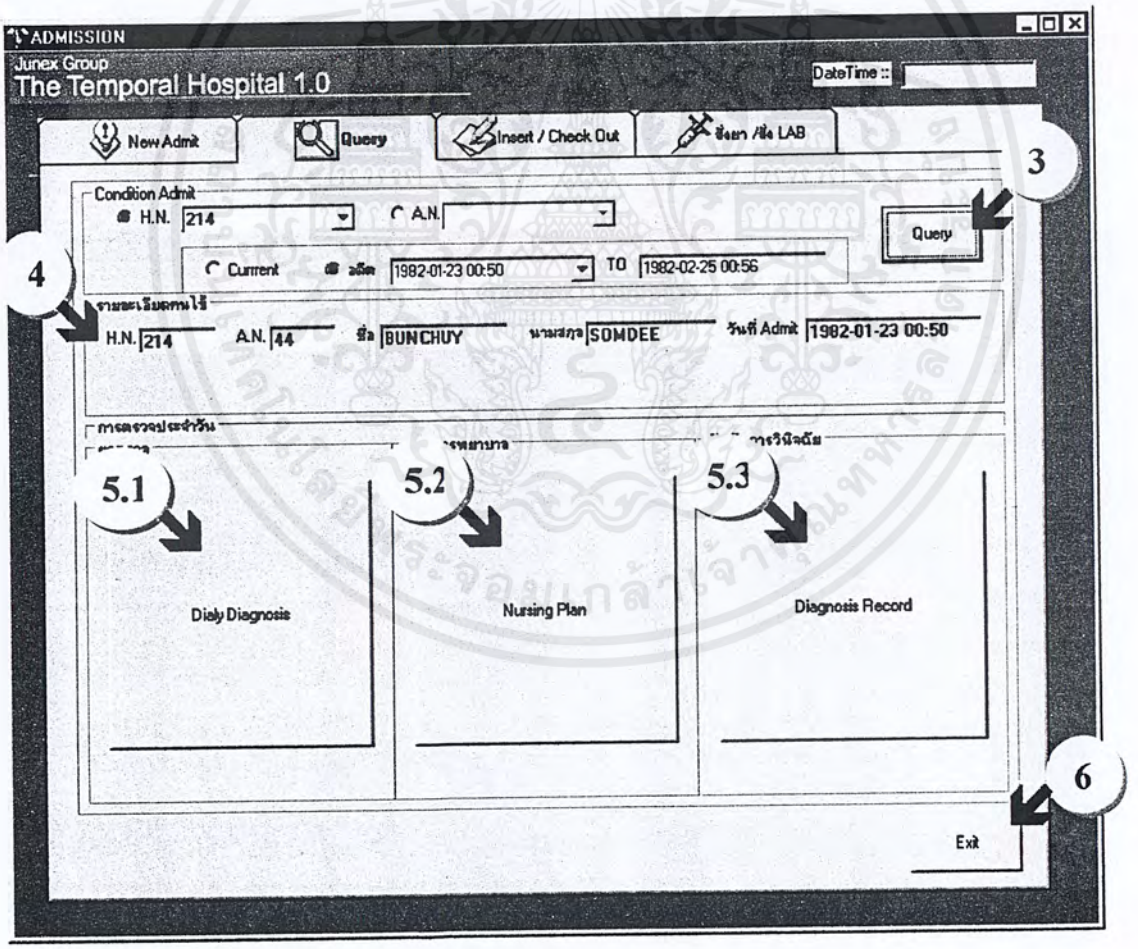
หน้าจอการค้นหาประวัติผู้ป่วยใน (QUERY)

การใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เลือกเงื่อนไขในการค้นหาประวัติของผู้ป่วยใน ซึ่งสามารถเลือกเงื่อนไขได้ 2 กรณี คือ
   1.1 เลือกจากหมายเลข HN ซึ่งจะเป็นการค้นหาประวัติผู้ป่วยใน ซึ่งผู้ป่วยอาจจะมีการแอดมิดอยู่ ณ ปัจจุบัน หรือเคยมีแต่การแอดมิดในอดีต ซึ่งจะทำให้สามารถเลือกเงื่อนไขของเวลาได้ 2 แบบ
   1.2 เลือกจากหมายเลข AN ซึ่งจะเป็นการค้นหาประวัติผู้ป่วยในจากหมายเลขการแอดมิด ซึ่งจะมีเพียงการค้นหาแบบปัจจุบันเท่านั้น
2. เลือกเงื่อนไขเชิงเวลาที่ต้องการค้นหา ซึ่งมี 2 กรณี คือ
   2.1 Current เป็นการค้นหาประวัติคนไข้ใน ที่มีค่า ณ เวลาปัจจุบัน
   2.2 อดีต (มีเฉพาะเมื่อเลือกเงื่อนไขการค้นหาจาก HN) เป็นการค้นหาประวัติคนไข้ในตามช่วงเวลาที่เคยมีการแอดมิดในอดีต ซึ่งจะมีเวลาที่เคยมีการแอดมิดทั้งหมดมาให้เลือกในคอมโบ้บ๊อกซ์ FROM และเมื่อเลือกเวลาการแอดมิดครั้งใดแล้ว ก็จะแสดง TO คือวันที่มีการเช็คเอาท์มาให้โดยอัตโนมัติ

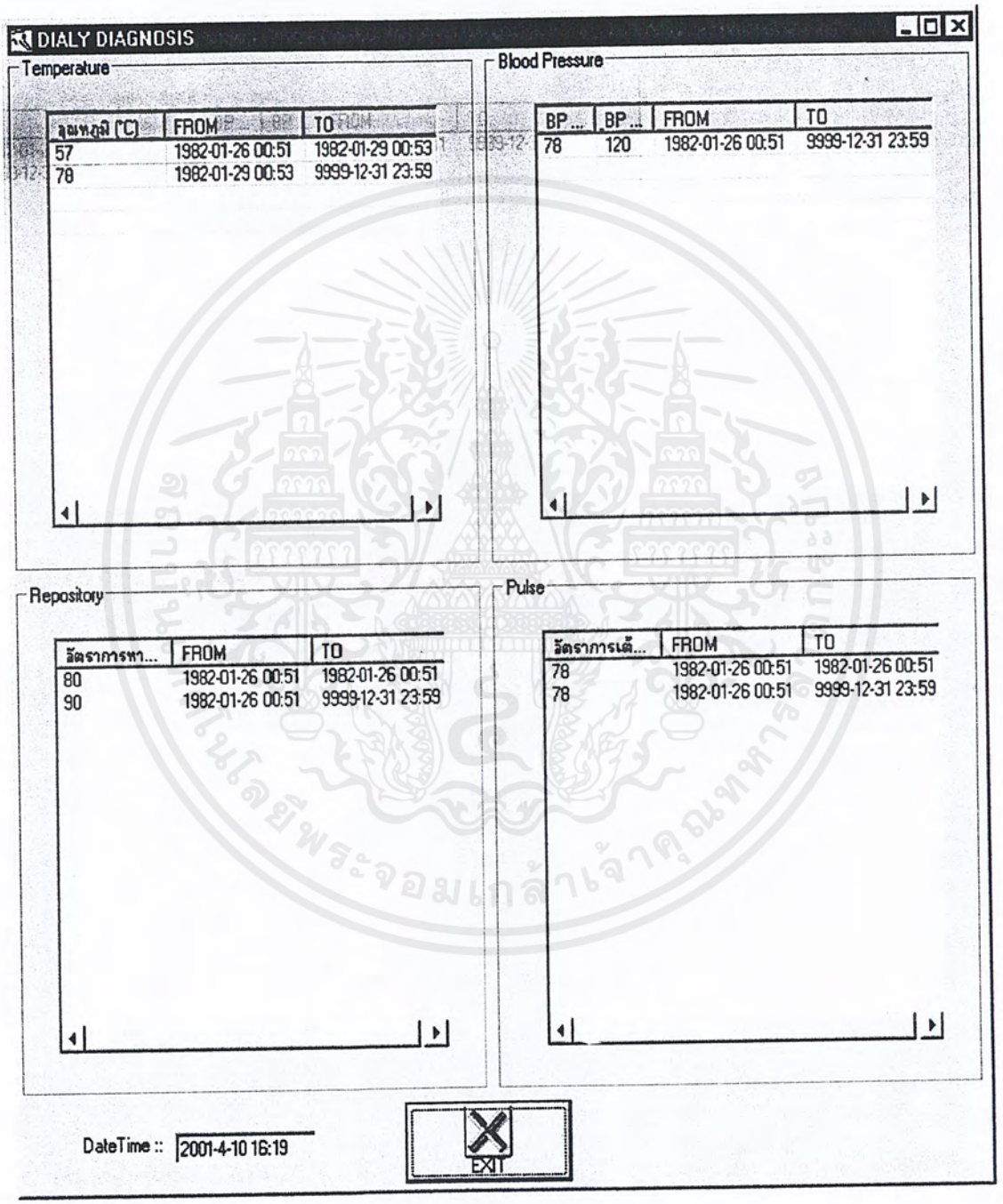


3. คลิกปุ่ม QUERY เพื่อทำการค้นหาประวัติคนไข้ในตามเงื่อนไขที่กำหนด
4. แอพพลิเคชั่นจะแสดงรายละเอียดแบบปัจจุบันของคนไข้ในเพื่อยืนยันความถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

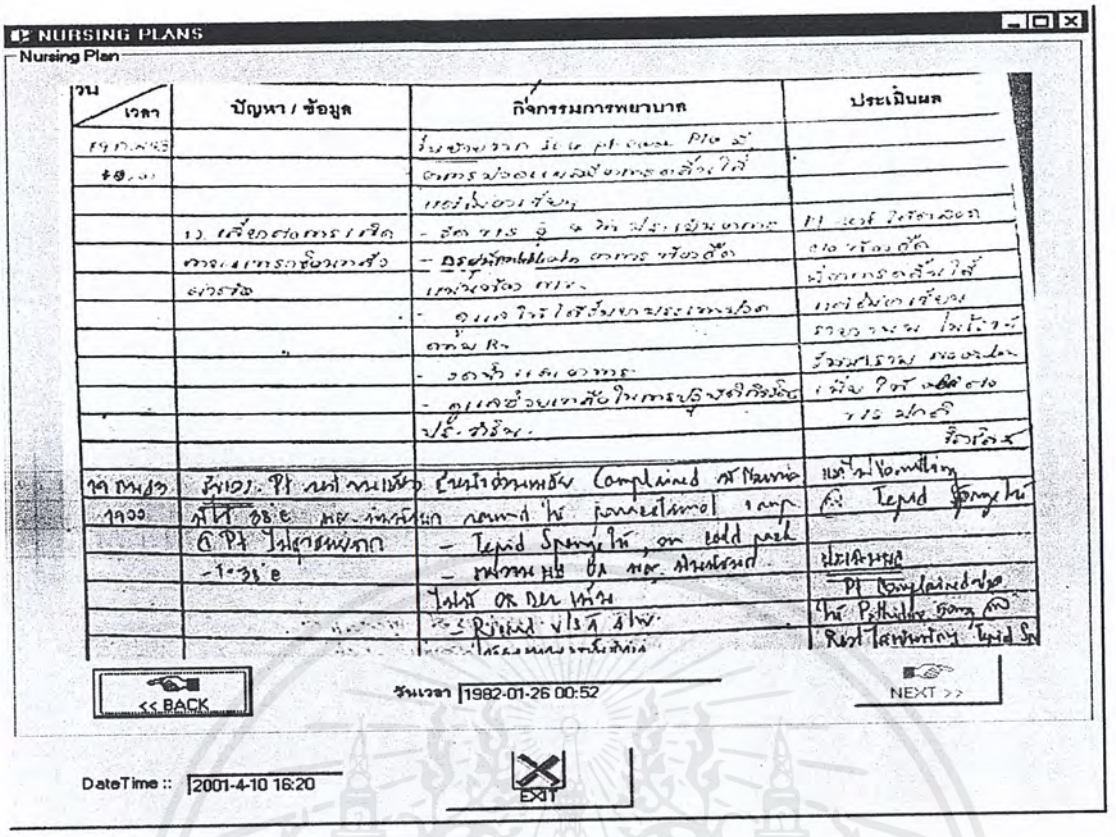
5. ถ้าต้องการดูรายละเอียดของประวัติคนไข้ในของการแอดมิดครั้งนั้น จะแยกออกเป็นหน้าจอต่างๆ 3 หน้าจอ ดังนี้

5.1 Dialy Diagnosis เป็นผลการวินิจฉัยประจำวันของผู้ป่วยใน ซึ่งจะมีผลการวัดอุณหภูมิ, ผลความดันโลหิต, ผลอัตราการเต้นหัวใจ และผลอัตราการหายใจ ของการแอดมิดครั้งนั้นทั้งหมด

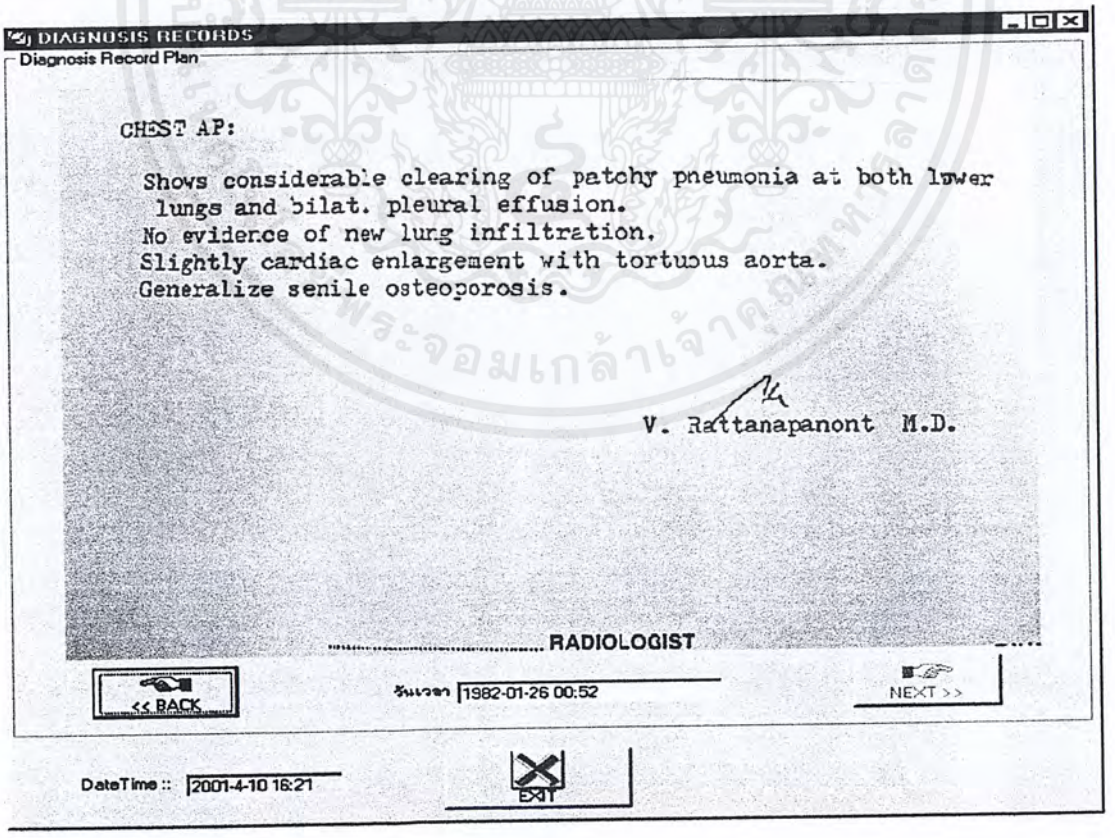


5.2 ฃNursing Plan เป็นแผนการพยาบาลของพยาบาลที่บันทึกแก่ผู้ป่วยใน ณ ช่วงเวลาต่างๆ ในการแอดมิดครั้งนั้น ซึ่งสามารถเลื่อนไปกลับไปมา เพื่อดูแผนการพยาบาลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5.3 Diagnosis Record เป็นผลการวินิจฉัยของแพทย์ที่บันทึกแก่ผู้ป่วยใน ณ ช่วงเวลาต่างๆ ในการแอดมิดครั้งนั้น ซึ่งสามารถเลื่อนไปกลับไปมาเพื่อดูผลการวินิจฉัยได้

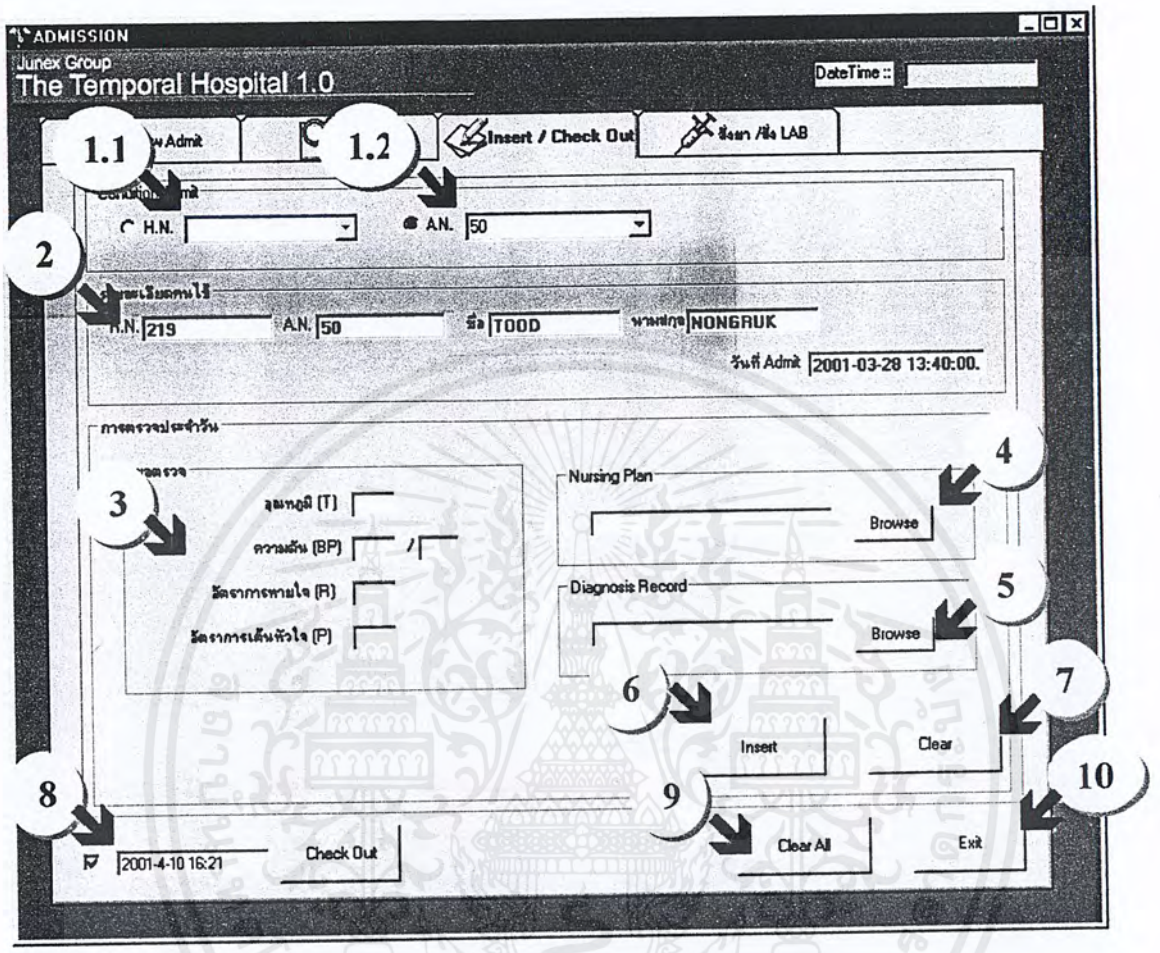


6. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

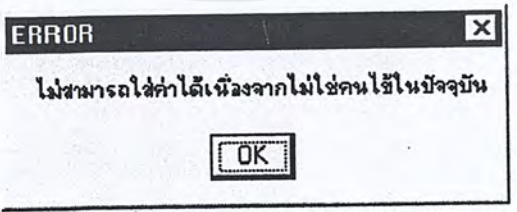
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอการบันทึกประวัติผู้ป่วยในและเช็คเอาท์ (INSERT & CHECK OUT)

การใช้งาน



1. เลือกเงื่อนไขในการบันทึกประวัติของผู้ป่วยใน ซึ่งสามารถเลือกเงื่อนไขได้ 2 กรณี คือ
   1.1 เลือกจากหมายเลข HN ซึ่งจะเป็นการบันทึกประวัติผู้ป่วยใน ได้เฉพาะผู้ป่วยที่มีการแอดมิดอยู่ ณ ปัจจุบันเท่านั้น ถ้าไม่มีการสั่งการแอดมิดอยู่ หรือเช็คเอาท์ออกไปแล้วจะปรากฏข้อความแจ้งเตือน



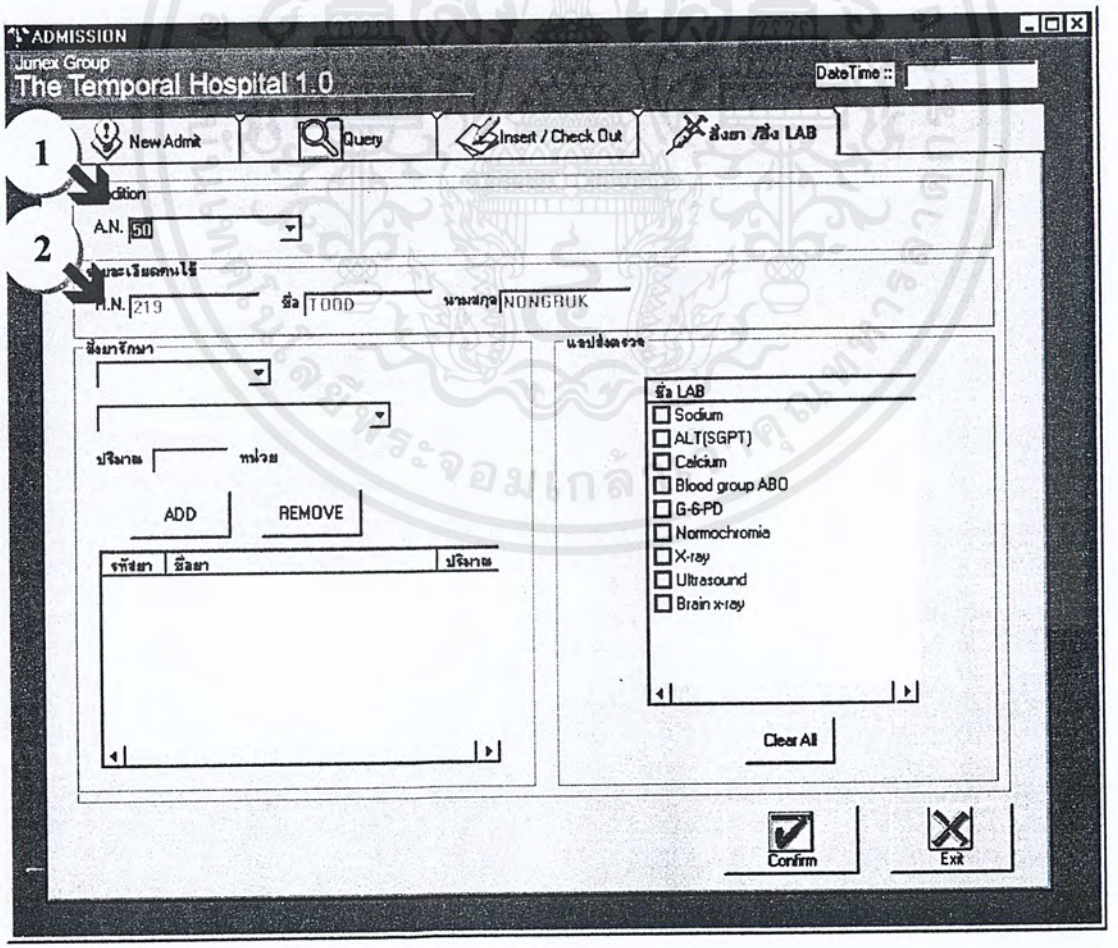
   1.2 เลือกจากหมายเลข AN ซึ่งจะเป็นการบันทึกประวัติผู้ป่วยในจากหมายเลขการแอดมิด ซึ่งจะมีเฉพาะหมายเลขการแอดมิดที่ยังอยู่ในโรงพยาบาลมาให้บันทึกเท่านั้น
2. แอพพลิเคชั่นจะแสดงรายละเอียดของผู้ป่วยแบบปัจจุบันเพื่อยืนยันความถูกต้อง
3. บันทึกผลการตรวจประจำวัน (ถ้ามี) ซึ่งได้แก่ อุณหภูมิ, ความดัน, อัตราการหายใจ และอัตราการเต้นของหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. บันทึกแผนการพยาบาล (ถ้ามี) โดยคลิกปุ่ม BROWSE เพื่อเลือกตำแหน่งที่เก็บไฟล์
5. บันทึกผลการวินิจฉัย (ถ้ามี) โดยคลิกปุ่ม BROWSE เพื่อเลือกตำแหน่งที่เก็บไฟล์
6. คลิกปุ่ม INSERT เมื่อต้องการบันทึกลงประวัติคนไข้ใน
7. คลิกปุ่ม Clear ถ้าต้องการยกเลิกค่าต่างๆ ที่ป้อนไว้
8. หากต้องการทำการเช็คเอาท์ เมื่อผู้ป่วยออกจากโรงพยาบาล ให้เช็คบ๊อกซ์ที่ Check out แล้วกำหนดเวลาในการเช็คเอาท์ ซึ่งจะอยู่ในรูปแบบ [ปี-เดือน-วัน ชั่วโมง:นาที] ; [YYYY-MM-DD HH:MM] ซึ่งเมื่อเช็คบ๊อกซ์แล้วแอพพลิเคชั่นจะตั้งให้เป็นวันที่ปัจจุบันโดยอัตโนมัติ แล้วทำการคลิกปุ่ม Check Out เพื่อทำการจำหน่ายผู้ป่วยใน
9. คลิกปุ่ม Clear All ถ้าต้องการยกเลิกทั้งหมด เพื่อเตรียมบันทึกประวัติผู้ป่วยในรายใหม่
10. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

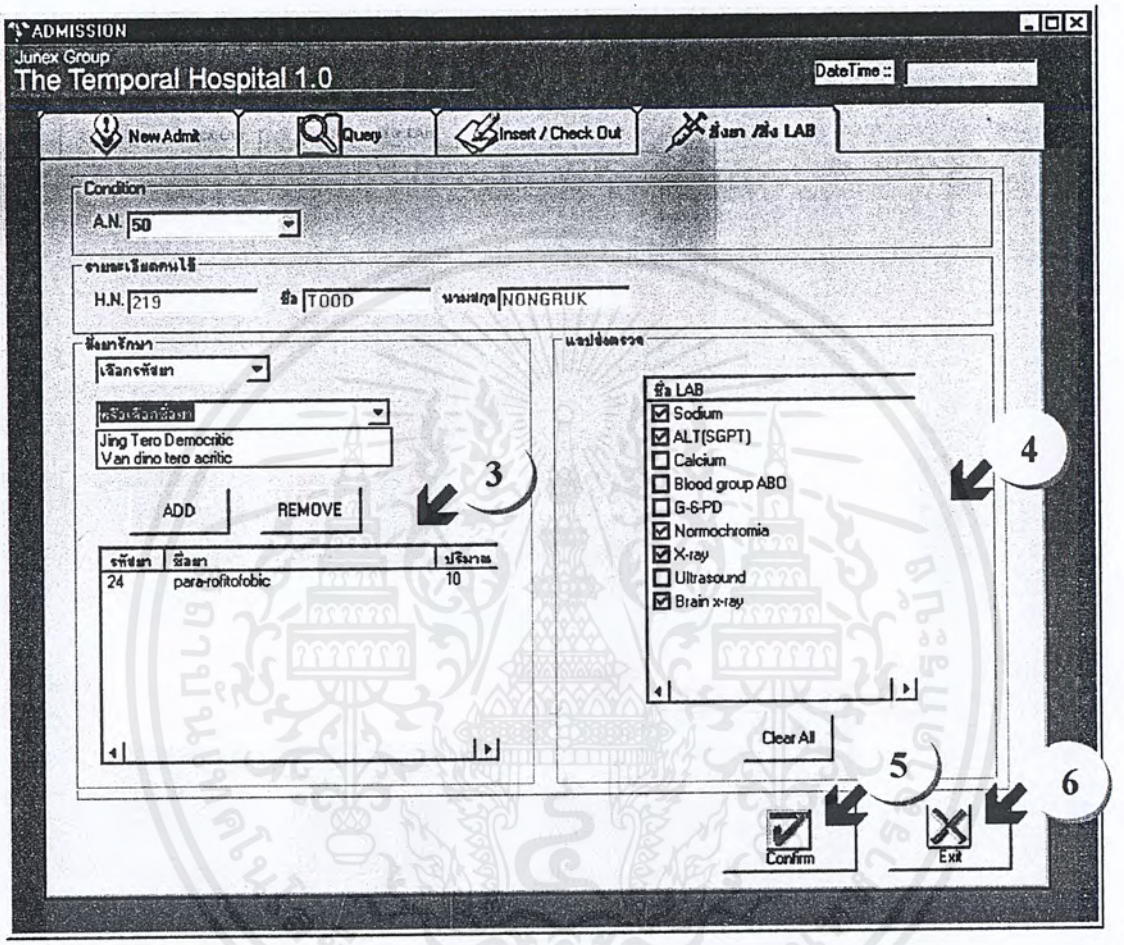
หน้าจอการสั่งยา/สั่งแลป

การใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กำหนดเงื่อนไขเริ่มต้นของผู้ป่วยที่ต้องการค้นหา โดยเลือกจากหมายเลข AN ของผู้ป่วย
2. แอพพลิเคชั่นจะแสดง HN ชื่อและนามสกุลแบบปัจจุบันของผู้ป่วยเพื่อยืนยันความถูกต้อง
3. หากมีการสั่งยารักษา (ถ้ามี) โดยเลือกจากคอมโบ้บ๊อกซ์รหัสยา หรือชื่อยาก็ได้ แล้วป้อนรหัสยา แล้วกดปุ่ม ADD เพื่อเพิ่มชื่อยาและปริมาณ ในตาราง

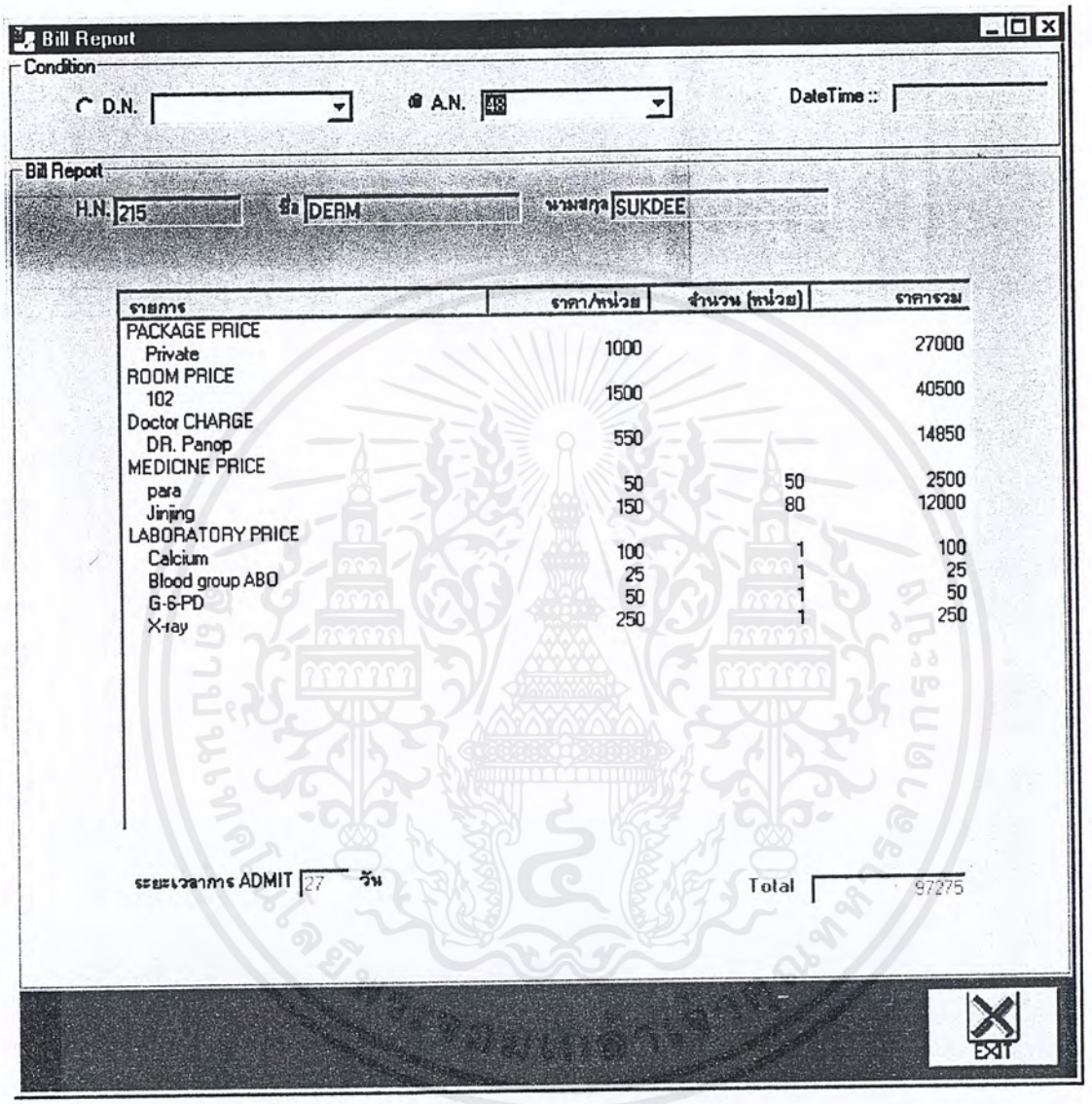


4. หากมีการสั่งตรวจแลป (ถ้ามี) ให้เช็คบ๊อกซ์หน้าชื่อแลปที่ต้องการสั่งตรวจ ถ้าต้องการยกเลิกการสั่งตรวจแลปทุกชนิด ให้คลิกที่ปุ่ม Clear All
5. ต้องการยืนยันการบันทึกผลการการสั่งยารักษา และ/หรือ การสั่งตรวจแลป ให้คลิกปุ่ม Confirm
6. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณค่าใช้จ่าย

การใช้งาน



1. เลือกเงื่อนไขในการคำนวณค่าใช้จ่ายของผู้ป่วย ซึ่งสามารถเลือกเงื่อนไขได้ 2 กรณี คือ
   1.1 เลือกจากหมายเลข DN ซึ่งจะเป็นการคำนวณค่าใช้จ่ายของผู้ป่วยนอก โดยไม่มีการเข้าพักรักษา ซึ่งจะมีค่าหมอ ค่าการตรวจทางห้องปฏิบัติการ ค่ายา ซึ่งจะแสดงรายละเอียดการตรวจและยาต่างๆ ที่แพทย์สั่ง รวมเป็นค่าใช้จ่ายทั้งหมด
   1.2 เลือกจากหมายเลข AN ซึ่งจะเป็นการคำนวณค่าใช้จ่ายของผู้ป่วยใน โดยจะคำนวณค่าใช้จ่ายตามจำนวนวันที่มีการเข้าพักรักษา ซึ่งจะรวมค่าหมอ ค่าตรวจทางห้องปฏิบัติการ ค่ายา ค่าห้องพัก ค่าแพกเกจที่เลือกพักรักษา รวมเป็นค่าใช้จ่ายทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

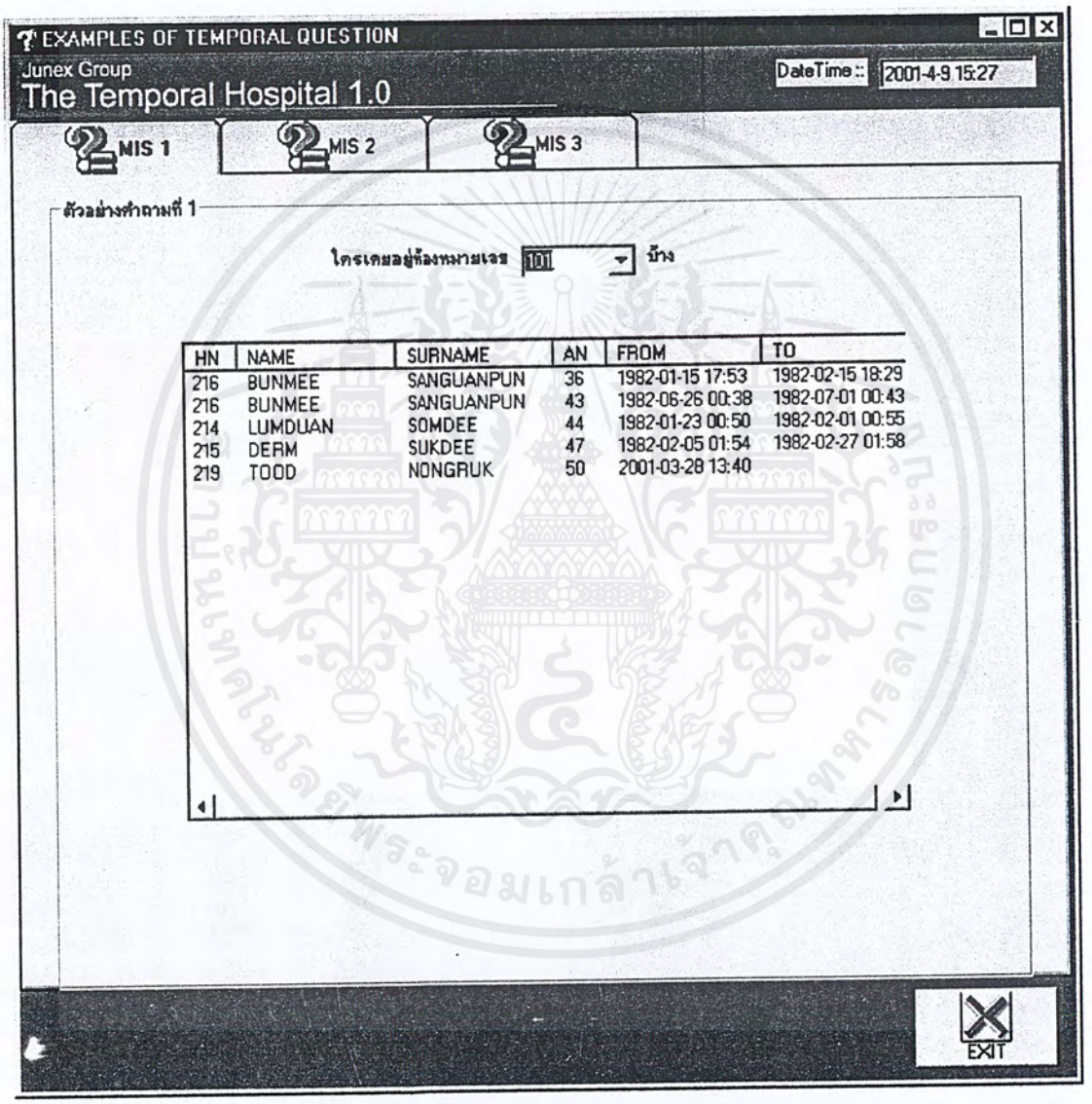
2. ค่าใช้จ่ายที่แสดงของผู้ป่วยรายนั้น จะเป็นค่าใช้จ่ายที่คำนวณถึง ณ เวลาปัจจุบัน หากผู้ป่วยในยังไม่มีการเช็คเอาท์ออกจากโรงพยาบาล ค่าใช้จ่ายก็จะคิดมาถึง ณ เวลาปัจจุบันของระบบ แต่ถ้าเช็คเอาท์ออกไปแล้ว หรือเป็นคนไข้นอก ค่าใช้จ่ายที่แสดง จะหมายถึงค่าใช้จ่ายครั้งล่าสุดของผู้ป่วย
3. ต้องการกลับไปยังหน้าจอหลัก ให้คลิก EXIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตอบคำถามเชิงเวลา

หน้าจอนี้จะเป็นหน้าจอในการตอบคำถามของผู้บริหาร ซึ่งในที่นี้ได้ทดลองจัดทำขึ้นเพื่อรองรับ 3 คำถาม ด้วยกัน คือ

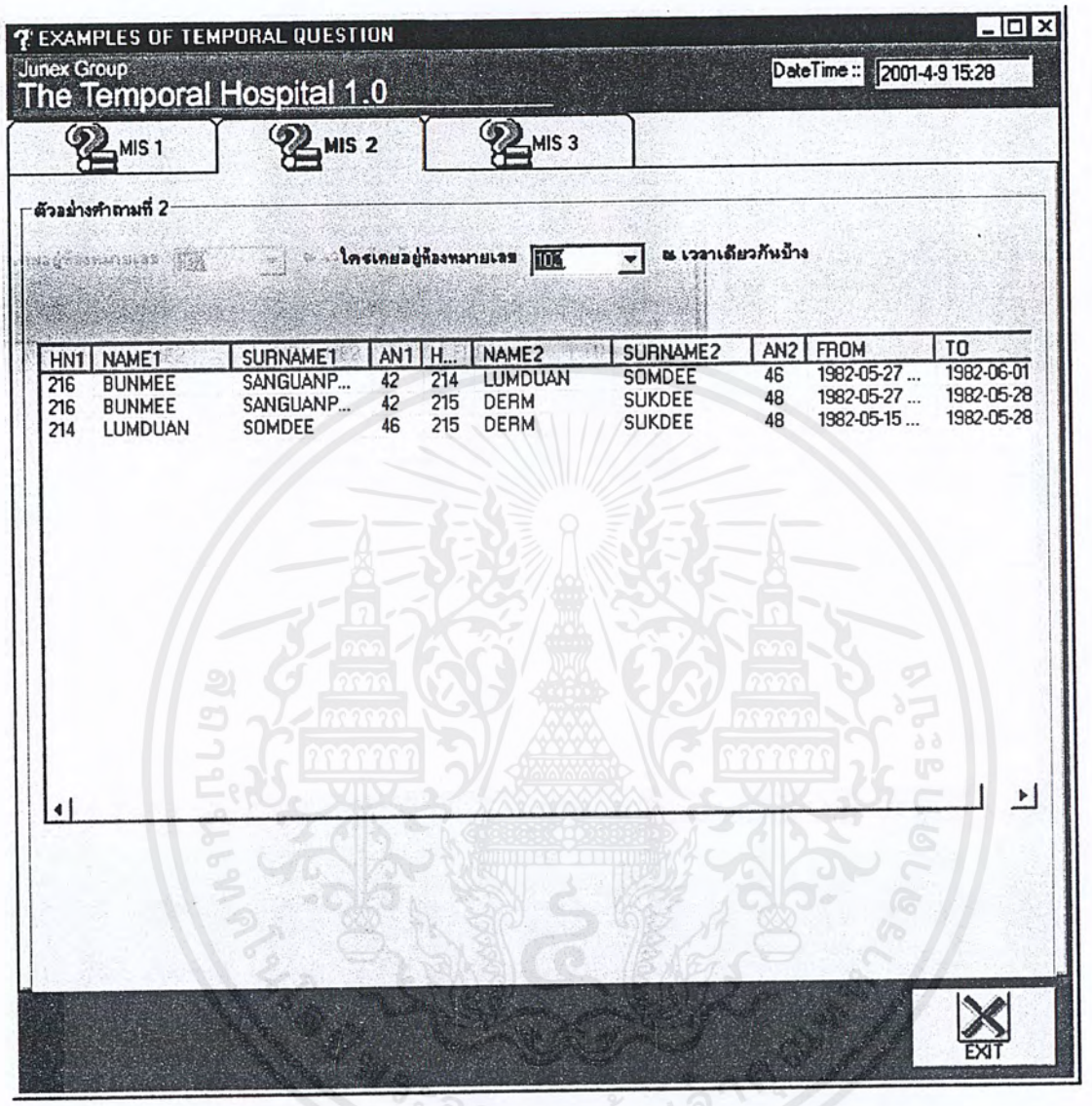
MIS1 : ใครเคยอยู่ห้องพักหมายเลขนั้นบ้าง



1. เลือกหมายเลขห้องจากคอมโบบ๊อกซ์ ซึ่งจะมีหมายเลขห้องทั้งหมดของโรงพยาบาล
2. จะแสดงผู้ป่วย และรายละเอียดของผู้ป่วยแบบปัจจุบันที่เคยอยู่ห้องนั้น ณ ช่วงเวลาต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIS2 : ใครเคยอยู่ห้องพักนั้นช่วงเวลาเดียวกันบ้าง

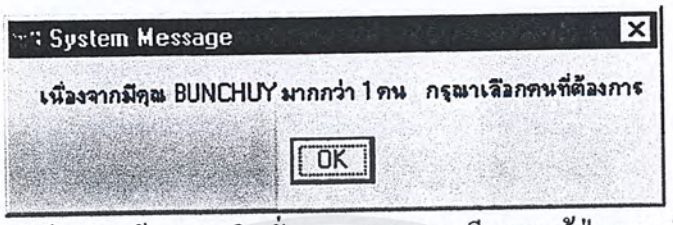


1. เลือกหมายเลขห้องจากคอมโบบ๊อกซ์ ซึ่งจะมีหมายเลขห้องทั้งหมดของโรงพยาบาล
2. จะแสดงผู้ป่วย และรายละเอียดของผู้ป่วยแบบปัจจุบัน 2 คนที่เคยอยู่ห้องเดียวกัน ณ ช่วงเวลาต่างๆ ทั้งหมด

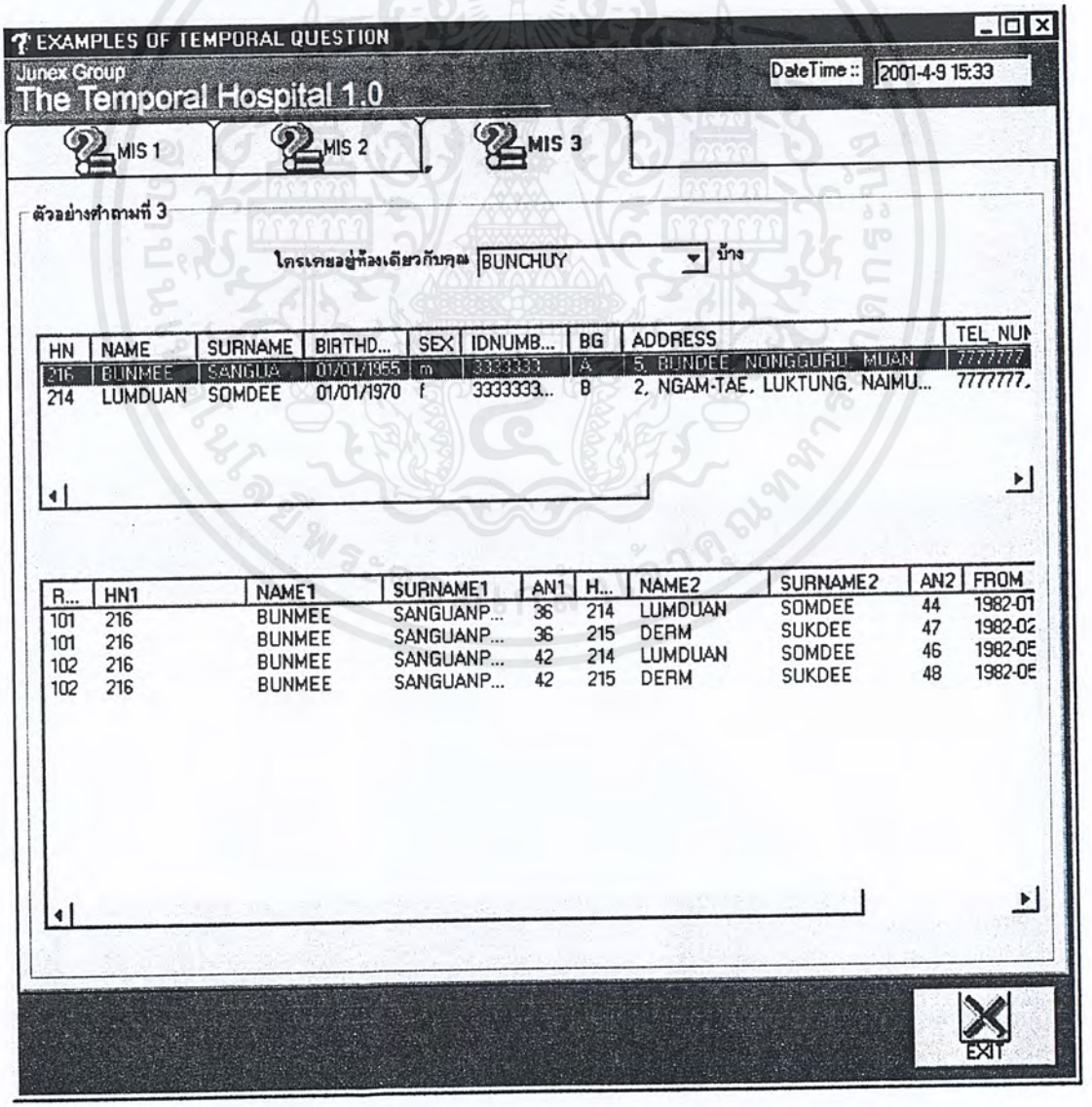
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIS3 : ใครเคยอยู่ห้องพักห้องเดียวกันกับคนไข้คนนั้นในช่วงเวลาเดียวกันบ้าง

1. เลือกชื่อผู้ป่วยจากคอมโบบ็อกซ์ ซึ่งจะมีชื่อทั้งหมดทั้งในอดีตและปัจจุบันของผู้ป่วยทั้งหมด
2. แอพพลิเคชั่นจะแสดงรายละเอียดแบบปัจจุบันของคนไข้ที่เคยมีชื่อหรือกำลังมีชื่อตามเงื่อนไขที่กำหนด ซึ่งถ้ามีคนไข้ที่ตรงตามเงื่อนไขมากกว่า 1 คน จะแสดงข้อความออกมา เพื่อให้เลือกเพื่อระบุว่าจะเป็นคนไข้รายใด



3. เมื่อระบุคนไข้คนใดแล้ว แอพพลิเคชั่นจะแสดงรายละเอียดของผู้ป่วยแบบปัจจุบันของผู้ป่วยคนนั้น กับผู้ป่วยรายอื่นที่เคยอยู่ห้องเดียวกัน ณ ช่วงเวลาต่างๆ ทั้งหมด แม้ว่าผู้ป่วยรายนั้นจะอยู่ห้องใดก็ตาม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ส่วนประกอบของโปรแกรมระบบงานโรงพยาบาล

ส่วนประกอบของซอร์สโค้ด

ซอร์สโค้ดของโปรแกรมประกอบด้วย ไฟล์ต่างๆ ดังนี้

1. ไฟล์ในไดเร็กทอรี่ nuvo_hospital เป็นไฟล์แอพพลิเคชั่นที่รันโดยโปรแกรมวิชวลเบสิค เวอร์ชั่น 6.0 ซึ่งจะมีไฟล์หลักๆ ของแอพพลิเคชั่น ดังนี้
   1.1 Project เป็นไฟล์โปรเจคหลักของแอพพลิเคชั่น ซึ่งเป็นไฟล์โปรเจคไฟล์เดียว ซึ่งเมื่อต้องการเรียนดูซอร์สโค้ดของโปรแกรมทั้งหมด ให้เปิดไฟล์นี้ขึ้นมา เพราะไฟล์โปรเจคนี้จะเรียกฟอร์มและโมดูลประกอบของแอพพลิเคชั่นขึ้นมาทั้งหมด
   1.2 Startform เป็นไฟล์ฟอร์มหน้าจอหลักของแอพพลิเคชั่น ซึ่งเป็นฟอร์มที่เชื่อมต่อไปยังฟอร์มอื่นๆ ตามฟังก์ชันงานของแอพพลิเคชั่น และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม
   1.3 Register เป็นไฟล์ฟอร์มหน้าจอประวัติคนไข้ ซึ่งเป็นฟอร์มที่จัดการเกี่ยวกับประวัติคนไข้ และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม มีฟังก์ชันงานดังนี้
      - การค้นหาผู้ป่วย (Query)
      - การป้อนข้อมูลผู้ป่วยใหม่ (INSERT)
      - การเปลี่ยนแปลงข้อมูล (Edit & Update)
   1.4 Diagnosis เป็นไฟล์ฟอร์มหน้าจอผลการวินิจฉัย ซึ่งเป็นฟอร์มที่จัดการเกี่ยวกับผลการวินิจฉัยของคนไข้ และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม มีฟังก์ชันงานดังนี้
      - การค้นหาผลการวินิจฉัย (Query)
      - การป้อนข้อมูลผลการวินิจฉัยใหม่ (NEW RECORD)
      - การค้นหาผลทางห้องปฏิบัติการ (LAB RESULT)
      - การค้นหาผลเอ็กซเรย์ (X-RAY RESULT)
      - การค้นหาผลการสั่งยา (MEDICAL RESULT)
   1.4 Lab เป็นไฟล์ฟอร์มหน้าจอผลทางห้องปฏิบัติการ ซึ่งเป็นฟอร์มที่จัดการบันทึกผลทางห้องปฏิบัติการของคนไข้ และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม มีฟังก์ชันงานดังนี้
      - การบันทึกผลทางห้องปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 Admission เป็นไฟล์ฟอร์มหน้าจอประวัติคนไข้ใน ซึ่งเป็นฟอร์มที่จัดการเกี่ยวกับประวัติของคนไข้ใน และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม มีฟังก์ชันงานดังนี้

- การแอดมิดผู้ป่วยใหม่ (ADMIT)
- การค้นหาประวัติผู้ป่วยใน (QUERY)
- การบันทึกประวัติผู้ป่วยในและเช็คเอาท์ (INSERT & CHECK OUT)
- การสั่งยา/สั่งแลป

1.6 Bill เป็นไฟล์ฟอร์มหน้าจอการคำนวณค่าใช้จ่าย ซึ่งเป็นฟอร์มที่จัดการเกี่ยวกับการคำนวณค่าใช้จ่ายในการรักษาพยาบาลของคนไข้ และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม มีฟังก์ชันงานดังนี้

- การคำนวณค่ารักษาพยาบาล

1.7 Temporal เป็นไฟล์ฟอร์มหน้าจอการตอบคำถามเชิงเวลา ซึ่งเป็นฟอร์มที่ใช้ในการแสดงผลการตอบคำถามเชิงเวลาของระบบงาน และเป็นไฟล์ที่เก็บลักษณะทางกราฟฟิคและซอร์สโค้ดของฟอร์ม ซึ่งจะตอบปัญหาเชิงเวลาต่างๆ ดังนี้

- MIS1: ใครเคยอยู่ห้องพักหมายเลขนั้นบ้าง
- MIS2: ใครเคยอยู่ห้องพักนั้นช่วงเวลาเดียวกันบ้าง
- MIS3: ใครเคยอยู่ห้องพักห้องเดียวกันกับคนไข้คนนั้นในช่วงเวลาเดียวกันบ้าง

1.8 Module1 จะเป็นไฟล์ที่เก็บตัวแปรต่างๆ ที่เป็นโกลบอล รวมทั้งโพรซีเยอร์ที่เป็นโกลบอลเพื่อให้ฟอร์มต่างๆ สามารถเรียกใช้งานได้

1.9 Null.jpg เป็นไฟล์รูปภาพพื้นฐานของแอพพลิเคชั่น ซึ่งจะเป็นไฟล์ภาพว่างเปล่า ใช้ในกรณีที่ผู้ป่วยไม่มีรูปภาพ

2. ไฟล์ในไดเร็กทอรี่ SQL จะเก็บ SQL Source Code ซึ่งจะเป็นคำสั่งภาษา SQL ที่ใช้ในการ Initaial ฐานข้อมูล โดยจะต้องเรียกไฟล์ SQL เหล่านี้ด้วยโปรแกรม SQL Editor แล้วทำการ Connect กับ nuvo_hospital Database แล้วทำการ Run File ต่างๆ ดังนี้

2.1 'create_table.sql' : เป็นไฟล์ที่รวบรวมคำสั่งในการสร้างฐานข้อมูลและตารางที่ใช้ในระบบ

2.2 'spl.sql' : เป็นไฟล์ที่รวบรวม รูทีนSPL ที่ใช้ในระบบ

2.3 'advanced_query.sql' : เป็นไฟล์ที่รวบรวมรูทีน SPL ที่ใช้การตอบคำถามเชิง MIS

2.4 'temp_table.sql' : เป็นไฟล์ที่รวบรวมคำสั่งในการสร้างตารางชั่วคราว ที่ใช้ในระบบ

2.5 'insert.sql' : เป็นไฟล์ที่รวมคำสั่งinsertข้อมูลพื้นฐานที่จำเป็น เช่น ข้อมูลยา ข้อมูลโรค และข้อมูลห้องพักเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายรูทีน SPL (spl.sql)

SPL Description
Procedure Name: UD2P
Input Type: int,int
Input Description: rowid ของการวินิจฉัย,rowid ของผู้ป่วย
Output Description: -
Description: เอา rowid ของการวินิจฉัยเก็บที่คนไข้

SPL Description
Function Name: D2P
Input Type: int
Input Description: rowid ของการวินิจฉัย
Output Description: rowid ของผู้ป่วย
Description: หา rowid ผู้ป่วยที่เป็นเจ้าของการวินิจฉัยครั้งนั้น

SPL Description
Function Name: newrowidPhoto
Input Type: -
Input Description: -
Output Description: rowid ของรูปที่ใส่ลงตาราง photo ล่าสุด
Description: หา rowid ของรูปที่ใส่ลงตาราง photo ล่าสุด

SPL Description
Procedure Name: set2table_disease
Input Type: datetime
Input Description: จุดของเวลาที่ต้องการรู้ชื่อโรค
Output Description: ตาราง result_rowchar แสดงชื่อโรค ณ ช่วงนั้น และช่วงเวลาที่วาลิด
Description: เอาชื่อโรค ณ ช่วงเวลาที่อินพุทเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: set2table_doctor
Input Type: datetime
Input Description: จุดของเวลาที่ต้องการรู้ชื่อหมอ
Output Description: ตาราง result_rowchar แสดงชื่อหมอ ณ ช่วงนั้น และช่วงเวลาที่วาลิด
Description: เอาชื่อหมอ ณ ช่วงเวลาที่อินพุทเข้าไป

SPL Description
Procedure Name: set2table_lab
Input Type: -
Input Description: -
Output Description: ตาราง lab_list แสดงชื่อ lab ทั้งหมด
Description: เอาชื่อ lab ทั้งหมดใส่ลงในตาราง lab_list

SPL Description
Function Name: newrowid_ad_dia
Input Type: -
Input Description: -
Output Description: rowid ของการวินิจฉัยขณะเข้าพักรักษา (ad_diagnosis)
Description: หา rowid ของการวินิจฉัยขณะเข้าพักรักษาตัวล่าสุด

SPL Description
Procedure Name: update_ad_dia
Input Type: int,int,int,int,int,int,blob,blob
Input Description: rowid ของการเข้าพักรักษา,อุณหภูมิ,อัตราการเต้นของหัวใจ,อัตราการหายใจ, ความดันโลหิต(ต่ำ),ความดันโลหิต(สูง),แผนการให้พยาบาล,ผลการวินิจฉัย
Output Description: -
Description: update ผลการตรวจประจำวันของการเข้าพักรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Function Name: updateinttoset
Input Type: collection,int
Input Description: -
Output Description: collection ที่ได้เพิ่ม int ที่รับเข้ามาแล้ว
Description: update ข้อมูลประเภท int ลงใน set

SPL Description
Procedure Name: UMOPD
Input Type: int,int,int
Input Description: rowid ของการวินิจฉัย,rowid ของยา,จำนวนยา
Output Description: -
Description: บันทึกการสั่งยาจากการวินิจฉัยลงตาราง MOP เพื่อเป็นประวัติการรับยาของคนไข้

SPL Description
Procedure Name: allocate_LRNA
Input Type: int,int
Input Description: rowid ของการเข้าพักรักษา,rowid ของ lab
Output Description: -
Description: เก็บผลการตรวจทางห้องปฏิบัติการชนิดตัวเลขที่ได้จากการเข้าพักรักษา ลงตาราง LRNA

SPL Description
Procedure Name: allocate_LRCA
Input Type: int,int
Input Description: rowid ของการเข้าพักรักษา,rowid ของ lab
Output Description: -
Description: เก็บผลการตรวจทางห้องปฏิบัติการชนิดตัวอักษรที่ได้จากการเข้าพักรักษา ลงตาราง LRNA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: allocate_LRPA
Input Type: int,int
Input Description: rowid ของการเข้าพักรักษา,rowid ของ lab
Output Description: -
Description: เก็บผลการตรวจทางห้องปฏิบัติการชนิดรูปภาพที่ได้จากการเข้าพักรักษา ลงตาราง LRNA

SPL Description
Procedure Name: null_LRA2table
Input Type: int
Input Description: rowid ของการเข้าพักรักษา
Output Description: ตาราง lab_list แสดงรายการ lab ที่ยังไม่มีผลการตรวจ
Description: เพื่อหารายการ lab ของการเข้าพักรักษาที่ยังไม่ได้รับการตรวจ

SPL Description
Procedure Name: UA2P
Input Type: int,int
Input Description: rowid ของการเข้าพักรักษา,rowid ของ ผู้ป่วย
Output Description: -
Description: เอา rowid ของการเข้าพักรักษาเก็บที่คนไข้

SPL Description
Function Name: A2P
Input Type: int
Input Description: rowid ของการเข้าพักรักษา
Output Description: rowid ของผู้ป่วย
Description: หา rowid ของผู้ป่วยที่เป็นเจ้าของการเข้าพักรักษาครั้งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Function Name: P2A
Input Type: int,datetime
Input Description: rowid ผู้ป่วย,จุดเวลาที่ต้องการหาการเข้าพักรักษา
Output Description: rowid ของการเข้าพักรักษา
Description: หา rowid ของการเข้าพักรักษาของผู้ป่วย ณ จุดเวลาที่กำหนด

SPL Description
Procedure Name: ULRP
Input Type: int,int,int,char
Input Description: rowid ของการวินิจฉัย,rowid ของ lab,ผล lab ชนิดตัวเลข,แหล่งที่มาของการสั่ง lab (d: การวินิจฉัย, a: การเข้าพักรักษา)
Output Description: -
Description: update ผลการตรวจทางห้องปฏิบัติการชนิดตัวเลขที่มาจากการเข้าพักรักษาหรือการวินิจฉัย ลงในตาราง LRNP

SPL Description
Procedure Name: ULRP
Input Type: int,int,char(50),char
Input Description: rowid ของการวินิจฉัย,rowid ของ lab,ผล lab ชนิดตัวอักษร,แหล่งที่มาของการสั่ง lab (d: การวินิจฉัย, a: การเข้าพักรักษา)
Output Description: -
Description: update ผลการตรวจทางห้องปฏิบัติการชนิดตัวอักษรที่มาจากการเข้าพักรักษาหรือการวินิจฉัย ลงในตาราง LRCP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: ULRP
Input Type: int,int,int,char,char
Input Description: rowid ของการวินิจฉัย,rowid ของ lab,rowid ของรูปในตาราง photo,แหล่งที่มาของการสั่ง lab (d: การวินิจฉัย, a: การเข้าพักรักษา),ตัวอักษรอะไรก็ได้ 1 ตัว
Output Description: -
Description: update ผลการตรวจทางห้องปฏิบัติการชนิดรูปภาพที่มาจากการเข้าพักรักษาหรือการวินิจฉัย ลงในตาราง LRPP

SPL Description
Procedure Name: ULRNA
Input Type: int,int
Input Description: rowid ของผลการตรวจทางห้องปฏิบัติการชนิดตัวเลขที่มาจากการเข้าพักรักษา (ตาราง LRNA),ผลการตรวจทางห้องปฏิบัติการชนิดตัวเลข
Output Description: -
Description: update ผลการตรวจทางห้องปฏิบัติการชนิดตัวเลข ลงในตาราง LRNA

SPL Description
Procedure Name: ULRPA
Input Type: int,int
Input Description: rowid ของผลการตรวจทางห้องปฏิบัติการชนิดรูปภาพที่มาจากการเข้าพักรักษา (ตาราง LRPA),ผลการตรวจทางห้องปฏิบัติการชนิดรูปภาพ
Output Description: -
Description: update ผลการตรวจทางห้องปฏิบัติการชนิดรูปภาพ ลงในตาราง LRPA

SPL Description
Procedure Name: ULRCA
Input Type: int,char(50)
Input Description: rowid ของผลการตรวจทางห้องปฏิบัติการชนิดตัวอักษรที่มาจากการเข้าพักรักษา (ตาราง LRCA),ผลการตรวจทางห้องปฏิบัติการชนิดตัวอักษร
Output Description: -
Description: update ผลการตรวจทางห้องปฏิบัติการชนิดตัวอักษร ลงในตาราง LRCA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: UMOPA
Input Type: int,int,int
Input Description: rowid ของการเข้าพักรักษา,rowid ของยา,จำนวนยา
Output Description: -
Description: update จำนวนยาที่สั่งจากการเข้าพักรักษาลงในตาราง MOP และ MOA

SPL Description
Procedure Name: hn2current_name
Input Type: int
Input Description: HN ของผู้ป่วย
Output Description: ตาราง result_rowchar แสดงชื่อปัจจุบัน ตาม HN ของผู้ป่วย
Description: หาชื่อปัจจุบันของผู้ป่วย จาก HN

SPL Description
Procedure Name: hn2current_surname
Input Type: int
Input Description: HN ของผู้ป่วย
Output Description: ตาราง result_rowchar แสดงนามสกุลปัจจุบัน ตาม HN ของผู้ป่วย
Description: หานามสกุลปัจจุบันของผู้ป่วย จาก HN

SPL Description
Procedure Name: current_pat_name
Input Type: datetime
Input Description: จุดเวลาที่ต้องการทราบชื่อผู้ป่วย
Output Description: ตาราง result_rowchar แสดงชื่อของผู้ป่วยทุกคน ณ เวลานั้น
Description: หาชื่อของผู้ป่วยทุกคน ณ เวลาใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: current_pat_sur
Input Type: datetime
Input Description: จุดเวลาที่ต้องการทราบนามสกุลผู้ป่วย
Output Description: ตาราง result_rowchar แสดงนามสกุลของผู้ป่วยทุกคน ณ เวลานั้น
Description: หานามสกุลของผู้ป่วยทุกคน ณ เวลาใดๆ

SPL Description
Procedure Name: room2table
Input Type: datetime
Input Description: จุดเวลาที่ต้องการทราบราคาห้อง
Output Description: ตาราง cost_list แสดงหมายเลขห้องและราคาห้องพักแต่ละห้อง ณ เวลาที่กำหนด
Description: หาราคาของห้องพักแต่ละห้อง ณ เวลาใดๆ

SPL Description
Procedure Name: current_doc_name
Input Type: datetime
Input Description: จุดเวลาที่ต้องการทราบชื่อหมอ
Output Description: ตาราง result_rowchar แสดงชื่อของหมอทุกคน ณ เวลานั้น
Description: หาชื่อของหมอทุกคน ณ เวลาใดๆ

SPL Description
Procedure Name: current_doc_sur
Input Type: datetime
Input Description: จุดเวลาที่ต้องการทราบนามสกุลหมอ
Output Description: ตาราง result_rowchar แสดงนามสกุลของหมอทุกคน ณ เวลานั้น
Description: หานามสกุลของหมอทุกคน ณ เวลาใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: updateNT
Input Type: int,int,char,char,date,char,char,char,int,char,char,char,char,char
Input Description: rowid ของผู้ป่วย,rowid ของรูปภาพผู้ป่วยที่เก็บในตาราง photo,ชื่อผู้ป่วย,นามสกุล,วันเกิด,เพศ,หมายเลขบัตรประจำตัวประชาชน,กรุ๊ปเลือด,เลขที่บ้าน,ชื่อถนน,ตำบล,อำเภอ,จังหวัด,รหัสไปรษณีย์
Output Description: -
Description: เป็นการ edit ข้อมูลลงในตาราง patience โดยเป็นการทับข้อมูลเดิม ไม่ได้เป็นการ update แบบ temporal

SPL Description
Function Name: editinttoset
Input Type: collection,int
Input Description: -
Output Description: set ที่ได้ลงตัวเลขที่รับเข้ามา ทับข้อมูลล่าสุดใน set
Description: เอาตัวเลขที่รับเข้ามาทับข้อมูลใน set ตัวล่าสุด (ตัวที่ยังคงวาลิดอยู่)

SPL Description
Function Name: editinttoset
Input Type: collection,char(50)
Input Description: -
Output Description: set ที่ได้ลงตัวอักษรที่รับเข้ามา ทับข้อมูลล่าสุดใน set
Description: เอาตัวอักษรที่รับเข้ามาทับข้อมูลใน set ตัวล่าสุด (ตัวที่ยังคงวาลิดอยู่)

SPL Description
Function Name: updatechartoset
Input Type: collection,char(50)
Input Description: -
Output Description: set ที่ได้ update ตัวอักษรที่รับเข้ามา
Description: update ข้อมูลประเภทตัวอักษรลงใน set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: updateT
Input Type: int,int,char,char,int,char,char,char,char,char
Input Description: rowid ผู้ป่วย,rowid รูปภาพ,ชื่อผู้ป่วย,นามสกุล,บ้านเลขที่,ถนน,ตำบล,อำเภอ, จังหวัด,รหัสไปรษณีย์
Output Description: -
Description: update ข้อมูลลงตาราง patience แบบ temporal คือ ข้อมูลเดิมยังคงอยู่ แต่ไม่วาลิดแล้ว

SPL Description
Procedure Name: updateNTtel
Input Type: int,char(15)
Input Description: rowid ผู้ป่วย,หมายเลขโทรศัพท์
Output Description: -
Description: การแก้ไข (edit) เบอร์โทรศัพท์แบบ nonsequence คือ ลบเบอร์นั้นออกจาก set ไปเลย

SPL Description
Procedure Name: updateNTdis
Input Type: int,int
Input Description: rowid ผู้ป่วย,rowid ของโรค
Output Description: -
Description: การแก้ไข (edit) โรคประจำตัวแบบ nonsequence คือ ลบโรคนั้นออกจาก set ไปเลย

SPL Description
Procedure Name: updateNTmed
Input Type: int,int
Input Description: rowid ผู้ป่วย,rowid ของยา
Output Description: -
Description: การแก้ไข (edit) ยาที่แพ้แบบ nonsequence คือ ลบยานั้นออกจาก set ไปเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: updateTtel
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: -
Description: update เบอร์โทรศัพท์แบบ temporal คือ มีเบอร์ใหม่เพิ่มเข้ามาหรืออาจยกเลิกการใช้เบอร์เดิม โดยก่อน execute ต้องสร้างตาราง set_table ก่อน โดย execute procedure createsettable และต้อง insert set เบอร์โทรที่ต้องการ update ลงไปในตารางนี้ก่อนด้วย จึง execute procedure updateTtel ได้

SPL Description
Procedure Name: updateTdis
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: -
Description: update โรคประจำตัวแบบ temporal คือ มีโรคใหม่เพิ่มเข้ามาหรืออาจหายจากโรคเดิม โดยก่อน execute ต้องสร้างตาราง set_table ก่อน โดย execute procedure createsettable และต้อง insert set โรคประจำตัวที่ต้องการ update ลงไปในตารางนี้ก่อนด้วย จึง execute procedure updateTdis ได้

SPL Description
Procedure Name: updateTmed
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: -
Description: update ยาที่แพ้แบบ temporal คือ มียาใหม่เพิ่มเข้ามาหรืออาจหายแพ้ยาเดิม โดยก่อน execute ต้องสร้างตาราง set_table ก่อน โดย execute procedure createsettable และต้อง insert set ยาที่แพ้ที่ต้องการ update ลงไปในตารางนี้ก่อนด้วย จึง execute procedure updateTmed ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: createsettable
Input Type: char
Input Description: เลือกชนิดคอลัมน์ในตารางว่าจะให้เป็นชนิดอะไร ระหว่าง c: set of char(15) กับ n: set of int
Output Description: ได้ตาราง set_table ตามชนิดที่ได้เลือก
Description: สร้างตารางตามชนิดที่เลือก

SPL Description
Procedure Name: HisRoom
Input Type: int
Input Description: rowid ของห้องพัก
Output Description: ตาราง hisroom เก็บคนที่เคยใช้ห้องพักนี้ โดยจะแสดงHN,ชื่อ,นามสกุล,ANและช่วงเวลาที่พัก
Description: หาผู้ป่วยที่เคยใช้ห้องนั้นทั้งหมด

SPL Description
Procedure Name: MatchRoom
Input Type: int
Input Description: rowid ของห้องพัก
Output Description: ตาราง matchroom เก็บผู้ป่วยที่เคยใช้ห้องนี้ร่วมกันเป็นคู่ๆ โดยจะแสดง HN,ชื่อ, นามสกุล,ANและช่วงเวลาที่พักอยู่ด้วยกัน
Description: หาผู้ป่วยที่เคยใช้ห้อง ณ เวลาเดียวกัน

SPL Description
Procedure Name: NFT2ROWID
Input Type: char,datetime,datetime
Input Description: ชื่อผู้ป่วย,เวลาที่เริ่ม,เวลาสิ้นสุด
Output Description: ตาราง result เก็บคนที่ชื่อตามที่รับเข้ามา ณ ช่วงเวลาที่กำหนดไว้ พร้อมทั้งแสดงช่วงเวลาที่ชื่อนั้นวาลิดด้วย
Description: หาผู้ป่วยจากชื่อที่วาลิด ณ ช่วงเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: SNFT2ROWID
Input Type: char,datetime,datetime
Input Description: นามสกุลผู้ป่วย,เวลาที่เริ่ม,เวลาสิ้นสุด
Output Description: ตาราง result เก็บคนที่มีนามสกุลตามที่รับเข้ามา ณ ช่วงเวลาที่กำหนดไว้ พร้อมทั้งแสดงช่วงเวลาที่นามสกุลนั้นวาลิดด้วย
Description: หาผู้ป่วยจากนามสกุลที่วาลิด ณ ช่วงเวลาที่กำหนด

SPL Description
Procedure Name: NSFT2ROWID
Input Type: char,char,datetime,datetime
Input Description: ชื่อผู้ป่วย,นามสกุลผู้ป่วย,เวลาที่เริ่ม,เวลาสิ้นสุด
Output Description: ตาราง result เก็บคนที่มีชื่อและนามสกุลตามที่รับเข้ามา ณ ช่วงเวลาที่กำหนดไว้ พร้อมทั้งแสดงช่วงเวลาที่ชื่อและนามสกุลนั้นวาลิดด้วย
Description: หาผู้ป่วยจากชื่อและนามสกุลที่วาลิด ณ ช่วงเวลาที่กำหนด

SPL Description
Procedure Name: UsedtoSleepwith
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง usedtosleep เก็บผู้ป่วยที่เคยอยู่ร่วมห้องกับผู้ป่วยคนที่รับเข้ามา พร้อมหมายเลขห้อง,หมายเลขการเข้าพักรักษาและช่วงเวลาที่อยู่ห้องเดียวกันด้วย
Description: หาผู้ป่วยที่เคยอยู่ร่วมห้องกับผู้ป่วยคนนี้บ้าง โดยแสดงหมายเลขห้องและช่วงเวลาที่อยู่ร่วมกันด้วย

SPL Description
Procedure Name: bill_diagnosis
Input Type: int
Input Description: rowid ของการวินิจฉัย
Output Description: ตาราง bill เก็บรายการการใช้บริการและราคาที่เกิดจากการรับการวินิจฉัยครั้งนั้น
Description: เพื่อแสดงรายการและราคาที่เกิดจากการวินิจฉัยในครั้งนั้นทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Function Name: TheIntCurrent
Input Type: collection
Input Description: -
Output Description: ข้อมูลตัวเลขที่เป็นค่าปัจจุบันใน set นั้น
Description: หาข้อมูลตัวเลขที่เป็นค่าปัจจุบันของ set ที่รับเข้ามา

SPL Description
Function Name: TheCharCurrent
Input Type: collection
Input Description: -
Output Description: ข้อมูลตัวอักษรที่เป็นค่าปัจจุบันใน set นั้น
Description: หาข้อมูลตัวอักษรที่เป็นค่าปัจจุบันของ set ที่รับเข้ามา

SPL Description
Procedure Name: bill_admission
Input Type: int
Input Description: rowid ของการเข้าพักรักษา
Output Description: ตาราง bill เก็บรายการการใช้บริการและราคาที่เกิดจากการเข้าพักรักษาครั้งนั้น
Description: เพื่อแสดงรายการและราคาที่เกิดจากการเข้าพักรักษาในครั้งนั้นทั้งหมด

SPL Description
Function Name: numofDay
Input Type: int,datetime
Input Description: rowid ของการเข้าพักรักษา,เวลาที่สิ้นสุดการเข้าพักรักษา (หรือเวลาปัจจุบัน เพื่อใช้คิดค่าใช้จ่าย)
Output Description: จำนวนวันที่เข้าพักรักษา
Description: เพื่อคำนวณหาจำนวนวันในการเข้าพักรักษาครั้งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: NT2ROWID
Input Type: char, datetime
Input Description: ชื่อผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง single_int แสดง rowid ของผู้ป่วย
Description: ค้นหา rowid ของผู้ที่มีชื่อ ณ จุดของเวลาตามที่กำหนด

SPL Description
Procedure Name: ST2ROWID
Input Type: char, datetime
Input Description: นามสกุลผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง single_int แสดง rowid ของผู้ป่วย
Description: ค้นหา rowid ของผู้ที่มีนามสกุล ณ จุดของเวลาตามที่กำหนด

SPL Description
Procedure Name: NST2ROWID
Input Type: char, char, datetime
Input Description: ชื่อผู้ป่วย, นามสกุลผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง single_int แสดง rowid ของผู้ป่วย
Description: ค้นหา rowid ของผู้ที่มีชื่อและนามสกุล ณ จุดของเวลาตามที่กำหนด

SPL Description
Procedure Name: R2APA
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง tmp_patience แสดงแอตทริบิวต์ทั้งหมดของผู้ป่วย ได้แก่ HN, rowid รูปของตาราง photo, ชื่อ, นามสกุล, วันเกิด, เพศ, หมายเลขบัตรประชาชน, กรุ๊ปเลือด, ที่อยู่, เซตของเบอร์โทรศัพท์, เซตของโรคที่เป็น, เซตของยาที่แพ้, หมายเลขการวินิจฉัย, หมายเลขการแอดมิด
Description: ค้นหาทุกแอตทริบิวต์ของผู้ป่วยในตาราง patience โดยรับ rowid และจุดของเวลาที่แอตทริบิวต์นั้นมีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: R2DI
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result_rowchar แสดง rowid ของผลการวินิจฉัยของผู้ป่วยรายนั้น และเวลาช่วงเวลาที่ผลการวินิจฉัยนั้นมีค่า
Description: ค้นหา rowid ผลการวินิจฉัยของผู้ป่วยทั้งหมด จาก rowid ผู้ป่วย

SPL Description
Procedure Name: RT2LRN
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง lab_number และผลการตรวจชนิดตัวเลขที่มีค่า ณ จุดของเวลานั้นทั้งหมด
Description: ค้นหาผลแลปชนิดตัวเลขทั้งหมดของคนไข้ ณ จุดของเวลานั้น

SPL Description
Procedure Name: RT2LRC
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง lab_char และผลการตรวจชนิดตัวอักษรที่มีค่า ณ จุดของเวลานั้นทั้งหมด
Description: ค้นหาผลแลปชนิดตัวอักษรทั้งหมดของคนไข้ ณ จุดของเวลานั้น

SPL Description
Procedure Name: RT2LRP
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง lab_photo และ rowid ชี้ไปยังผลการตรวจชนิด BLOB ในตาราง photo ที่มีค่า ณ จุดของเวลานั้นทั้งหมด
Description: ค้นหาผลแลปชนิด BLOB ทั้งหมดของคนไข้ ณ จุดของเวลานั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: RT2MOP
Input Type: int, datetime
Input Description: rowid ของผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง medicine และผลการสั่งยาที่มีค่า ณ จุดของเวลานั้น
Description: ค้นหาผลการสั่งยาทุกชนิดของคนไข้ ณ จุดของเวลานั้น

SPL Description
Procedure Name: RL2LRN
Input Type: int, int
Input Description: rowid ผู้ป่วย, rowid ของ lab ชนิดตัวเลข
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง lab_number และผลการตรวจชนิดตัวเลขทั้งหมด
Description: ค้นหาผลแลปชนิดตัวเลขตาม rowid ของ lab ชนิดตัวเลขที่ระบุ ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการตรวจมา

SPL Description
Procedure Name: RL2LRC
Input Type: int, int
Input Description: rowid ผู้ป่วย, rowid ของ lab ชนิดตัวอักษร
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง lab_char และผลการตรวจชนิดตัวเลขทั้งหมด
Description: ค้นหาผลแลปชนิดตัวอักษรตาม rowid ของ lab ชนิดตัวอักษรที่ระบุ ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการตรวจมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: RL2LRP
Input Type: int, int
Input Description: rowid ผู้ป่วย, rowid ของ lab ชนิด BLOB
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง lab_photo และ rowid ของผลการตรวจชนิด BLOB ที่ชี้ไปยังตาราง photo ทั้งหมด
Description: ค้นหาผลแลปชนิด BLOB ตาม rowid ของ lab ชนิด BLOB ที่ระบุ ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการตรวจมา

SPL Description
Procedure Name: RM2Q
Input Type: int, int
Input Description: rowid ผู้ป่วย, rowid ยา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง medicine และผลการสั่งยา ทั้งหมด
Description: ค้นหาผลการสั่งยาตาม rowid ของยาที่ระบุ ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการสั่งยา

SPL Description
Procedure Name: allpicture
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง photo ที่เป็นรูปของผู้ป่วยรายนั้น และช่วงเวลาที่รูปนั้นมีค่า
Description: ค้นหา rowid แอตทริบิวต์รูปภาพผู้ป่วยชนิด BLOB ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: allname
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดงชื่อทั้งหมดและช่วงเวลาที่ชื่อนั้นมีค่า ของผู้ป่วยรายนั้น
Description: ค้นหาแอตทริบิวต์ชื่อผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

SPL Description
Procedure Name: allsurname
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดงนามสกุลทั้งหมดและช่วงเวลาที่นามสกุลนั้นมีค่า ของผู้ป่วยรายนั้น
Description: ค้นหาแอตทริบิวต์นามสกุลผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

SPL Description
Procedure Name: alldiagnosis
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง diagnosis ที่เป็นผลการวินิจฉัยทั้งหมดของผู้ป่วยรายนั้น และช่วงเวลาที่ผลการวินิจฉัยนั้นมีค่า
Description: ค้นหา rowid ผลการวินิจฉัยของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: alladmission
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง admission ที่เป็นการแอดมิดทั้งหมดของผู้ป่วยรายนั้น และช่วงเวลาที่การแอดมิดนั้นมีค่า
Description: ค้นหา rowid การแอดมิดของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

SPL Description
Procedure Name: alladdress
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดงที่อยู่ทั้งหมด ประกอบด้วย บ้านเลขที่, ถนน, ตำบล, อำเภอ, จังหวัด, รหัสไปรษณีย์ และช่วงเวลาที่ที่อยู่นั้นมีค่า ของผู้ป่วยรายนั้น
Description: ค้นหาแอตทริบิวต์ที่อยู่ของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

SPL Description
Procedure Name: alltel
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดงเซ็ตของเบอร์โทรศัพท์ทั้งหมด และช่วงเวลาที่เซตของเบอร์โทรศัพท์นั้นมีค่า ของผู้ป่วยรายนั้น
Description: ค้นหาแอตทริบิวต์เบอร์โทรศัพท์ของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: alldis
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดงเซ็ตของ rowid ที่ชี้ไปยังตาราง disease ทั้งหมด และช่วงเวลาที่เซตโรคนั้นมีค่า ของผู้ป่วยรายนั้น
Description: ค้นหา rowid ของแอตทริบิวต์โรคของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

SPL Description
Procedure Name: allmed
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดงเซ็ตของ rowid ที่ชี้ไปยังตาราง medicine ทั้งหมด และช่วงเวลาที่เซตยาที่แพ้นั้นมีค่า ของผู้ป่วยรายนั้น
Description: ค้นหา rowid ของแอตทริบิวต์ยาที่แพ้ของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

SPL Description
Procedure Name: atpicture
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง photo ที่เป็นรูปของผู้ป่วยรายนั้น และช่วงเวลาที่รูปนั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา
Description: ค้นหา rowid แอตทริบิวต์รูปภาพผู้ป่วยชนิด BLOB ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: atname
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดงชื่อของผู้ป่วยรายนั้น และช่วงเวลาที่ชื่อนั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา
Description: ค้นหาแอตทริบิวต์ชื่อผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

SPL Description
Procedure Name: atsurname
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดงนามสกุลของผู้ป่วยรายนั้น และช่วงเวลาที่นามสกุลนั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา
Description: ค้นหาแอตทริบิวต์นามสกุลผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

SPL Description
Procedure Name: atdiagnosis
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง diagnosis ที่เป็นผลการวินิจฉัยของผู้ป่วยรายนั้น และช่วงเวลาที่การวินิจฉัยนั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา
Description: ค้นหา rowid ผลการวินิจฉัยของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: atadmission
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดง rowid ชี้ไปตาราง admission ที่เป็นการแอดมิดของผู้ป่วยรายนั้น และช่วงเวลาที่การแอดมิดนั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา
Description: ค้นหา rowid การแอดมิดของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

SPL Description
Procedure Name: ataddress
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดงที่อยู่ ประกอบด้วย บ้านเลขที่, ถนน, ตำบล, อำเภอ, จังหวัด, รหัสไปรษณีย์ และช่วงเวลาที่ที่อยู่นั้นมีค่าณ จุดของเวลาที่ต้องการค้นหา ของผู้ป่วยรายนั้น
Description: ค้นหาแอตทริบิวต์ที่อยู่ของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

SPL Description
Procedure Name: alltel
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดงเซ็ตของเบอร์โทรศัพท์ และช่วงเวลาที่เซ็ตของเบอร์โทรศัพท์นั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา ของผู้ป่วยรายนั้น
Description: ค้นหาแอตทริบิวต์เบอร์โทรศัพท์ของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้นทุกช่วงเวลาที่เคยมีการบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: atdis
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดงเซ็ตของ rowid ที่ชี้ไปยังตาราง disease ทั้งหมด และช่วงเวลาที่เซตโรคนั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา ของผู้ป่วยรายนั้น
Description: ค้นหา rowid ของแอตทริบิวต์โรคของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

SPL Description
Procedure Name: atmed
Input Type: int, datetime
Input Description: rowid ผู้ป่วย, จุดของเวลาที่ต้องการค้นหา
Output Description: ตาราง result แสดงเซ็ตของ rowid ที่ชี้ไปยังตาราง medicine ทั้งหมด และช่วงเวลาที่เซตยาที่แพ้นั้นมีค่า ณ จุดของเวลาที่ต้องการค้นหา ของผู้ป่วยรายนั้น
Description: ค้นหา rowid ของแอตทริบิวต์ยาที่แพ้ของผู้ป่วย ตาม rowid ของผู้ป่วยรายนั้น ณ จุดของเวลาที่ต้องการค้นหา

SPL Description
Function Name: CAPN
Input Type: --
Input Description: --
Output Description: int จำนวนชื่อผู้ป่วย
Description: นับจำนวนชื่อผู้ป่วยทั้งหมด จากตาราง patience

SPL Description
Function Name: CAPS
Input Type: --
Input Description: --
Output Description: int จำนวนนามสกุลผู้ป่วย
Description: นับจำนวนนามสกุลผู้ป่วยทั้งหมด จากตาราง patience

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: nameNSEQ
Input Type: char
Input Description: ชื่อผู้ป่วยที่ต้องการค้นหา
Output Description: ตาราง single_int แสดง rowid ของผู้ป่วย
Description: ค้นหา rowid ผู้ป่วยที่เคยมีชื่อตามเงื่อนไขที่ต้องการค้นหา โดยไม่คำนึงถึงช่วงเวลาที่ชื่อนั้นมีค่า

SPL Description
Procedure Name: surnameNSEQ
Input Type: char
Input Description: นามสกุลผู้ป่วยที่ต้องการค้นหา
Output Description: ตาราง single_int แสดง rowid ของผู้ป่วย
Description: ค้นหา rowid ผู้ป่วยที่เคยมีนามสกุลตามเงื่อนไขที่ต้องการค้นหา โดยไม่คำนึงถึงช่วงเวลาที่นามสกุลนั้นมีค่า

SPL Description
Procedure Name: nameNSEQ
Input Type: char, char
Input Description: ชื่อผู้ป่วยที่ต้องการค้นหา, นามสกุลผู้ป่วยที่ต้องการค้นหา
Output Description: ตาราง single_int แสดง rowid ของผู้ป่วย
Description: ค้นหา rowid ผู้ป่วยที่เคยมีชื่อและนามสกุลตามเงื่อนไขที่ต้องการค้นหา โดยไม่คำนึงถึงช่วงเวลาที่ชื่อและนามสกุลนั้นมีค่า

SPL Description
Procedure Name: admitOKorNot
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง result แสดง 't' หรือ 'f' บอกว่าผู้ป่วยรายนั้นการแอดมิดครั้งล่าสุดสิ้นสุดแล้วหรือไม่
Description: ตรวจสอบว่าผู้ป่วยเคยมีการแอดมิดหรือไม่ ถ้าเคยการแอดมิดครั้งล่าสุดสิ้นสุดแล้วหรือไม่ หมายถึงเช็คเอาท์ออกไปแล้วหรือยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: QLN
Input Type: int
Input Description: rowid ผู้ป่วย
Output Description: ตาราง lab_list แสดงชื่อแลป ทั้งหมดที่ผู้ป่วยรายนั้นเคยตรวจ
Description: ค้นหารายชื่อแลปที่เคยมีการตรวจของผู้ป่วยรายนั้นทั้งหมด

SPL Description
Procedure Name: alltemp
Input Type: int
Input Description: rowid การแอดมิด
Output Description: ตาราง result แสดงผลอุณหภูมิทั้งหมดของการแอดมิดครั้งนั้น
Description: ค้นหาผลอุณหภูมิทั้งหมดของการแอดมิดครั้งนั้น

SPL Description
Procedure Name: allpulse
Input Type: int
Input Description: rowid การแอดมิด
Output Description: ตาราง result แสดงผลอัตราการเต้นหัวใจทั้งหมดของการแอดมิดครั้งนั้น
Description: ค้นหาผลอัตราการเต้นหัวใจทั้งหมดของการแอดมิดครั้งนั้น

SPL Description
Procedure Name: allrep
Input Type: int
Input Description: rowid การแอดมิด
Output Description: ตาราง result แสดงผลอัตราการหายใจทั้งหมดของการแอดมิดครั้งนั้น
Description: ค้นหาผลอัตราการหายใจทั้งหมดของการแอดมิดครั้งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPL Description
Procedure Name: allbp
Input Type: int
Input Description: rowid การแอดมิด
Output Description: ตาราง result แสดงผลความดันเลือดต่ำสุด – สูงสุดทั้งหมดของการแอดมิดครั้งนั้น
Description: ค้นหาผลความดันเลือดต่ำสุด – สูงสุด ทั้งหมดของการแอดมิดครั้งนั้น

SPL Description
Procedure Name: allnurse
Input Type: int
Input Description: rowid การแอดมิด
Output Description: ตาราง result แสดงผล rowid ชี้ไปยังตาราง photo ซึ่งเก็บภาพแผนการพยาบาลทั้งหมด ของการแอดมิดครั้งนั้น
Description: ค้นหา rowid ข้อมูลชนิด BLOB ของแผนการพยาบาลทั้งหมดของการแอดมิดครั้งนั้น

SPL Description
Procedure Name: alldiag
Input Type: int
Input Description: rowid การแอดมิด
Output Description: ตาราง result แสดงผล rowid ชี้ไปยังตาราง photo ซึ่งเก็บภาพผลการวินิจฉัยทั้งหมด ของการแอดมิดครั้งนั้น
Description: ค้นหา rowid ข้อมูลชนิด BLOB ของผลการวินิจฉัยทั้งหมดของการแอดมิดครั้งนั้น

วิธีการคอมไพล์โปรแกรม

1. ทำการ Start Informix Universal Server ใน Control Panel => Service
2. รันไฟล์ SQL ต่างๆ ที่มีเพื่อทำการเริ่มต้น (Initialize) ฐานข้อมูล
3. สร้างโมเดลที่ใช้ในการเชื่อมต่อฐานข้อมูลขึ้นใหม่ (ดูการสร้างโมเดลใหม่ในการติดตั้งโปรแกรมระบบงานโรงพยาบาล)
4. รันแอพพลิเคชั่นโดยเปิดไฟล์ Project1.vbp โดยใช้โปรแกรม Visual Basic 6.0 หรือเปิดไฟล์ HOSPITAL.EXE ขึ้นมาได้เลย ก็จะสามารถใช้งานแอพพลิเคชั่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้