

การพัฒนาระบบงานประกันคุณภาพด้านมาตรฐานโรงพยาบาลและบริการ
สุขภาพ

THE DEVELOPMENT OF HOSPITAL ACCREDITATION SYSTEM



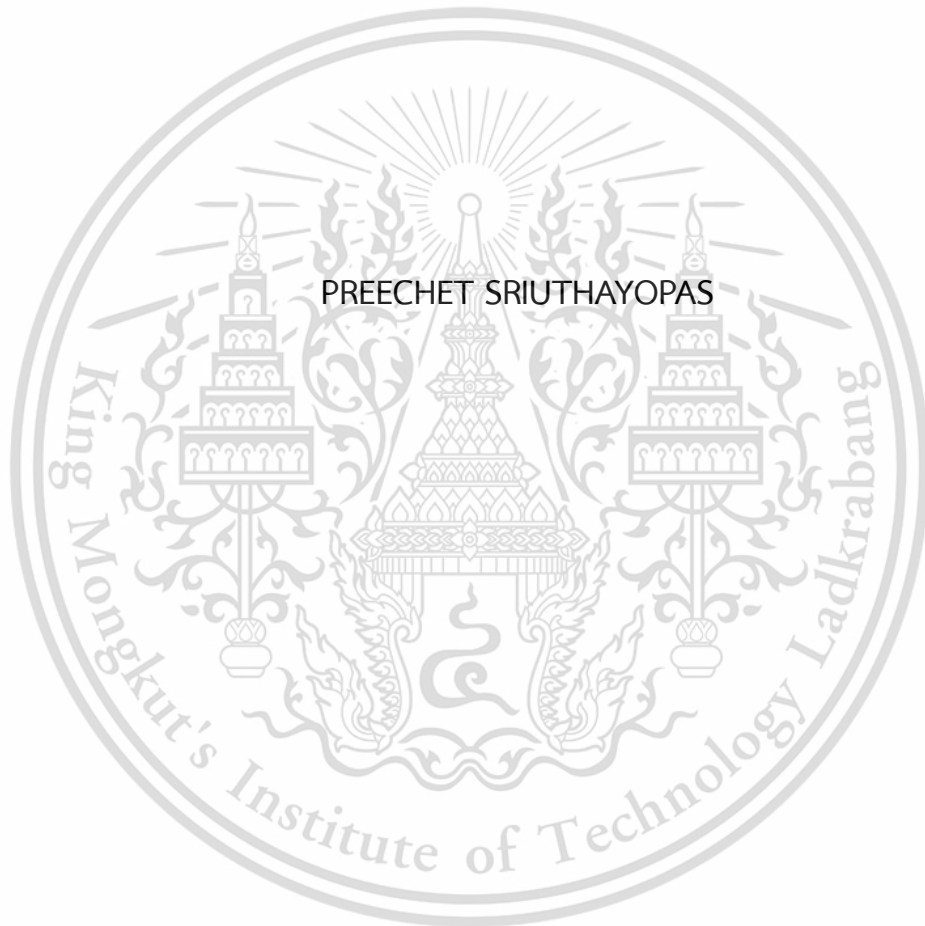
การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์
ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2566

KMITL-2023-SC-M-017-076

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DEVELOPMENT OF HOSPITAL ACCREDITATION SYSTEM

PREECHET SRIUTHAYOPAS



AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE
IN DATA SCIENCE AND ANALYTICS
KMITL DIGITAL ANALYTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2023

KMITL-2023-SC-M-017-076

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2023

SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Independent Study Title	The Development of hospital Accreditation System
Student Name	Preechet Sriuthayopas
Student ID	64605068
Degree	Master of Science (Data Science and Analytics) KMITL-Digital Analytics and Intelligence Center
Year	2023
Independent Study Advisor	Asst. Prof. Dr. Pattama Charoenporn

Abstract

The creation and development of quality assurance system programs in the medical field at Phramongkutklao Hospital are crucial for improving and enhancing medical services efficiently. In this research, a key program will focus on measuring the time taken by each physician to examine a patient in each examination room. This program aims to assist in measuring and evaluating the medical care provided, impacting the satisfaction of service recipients and hospital management. The program will check the duration of patient services in each examination room against the hospital's standards and evaluate the medical care processes. The developed program for this research is IPD Center, emphasizing outpatient service duration. It will summarize the assessment results and help set goals for continuous improvement, contributing to the ongoing and increasingly effective development of Phramongkutklao Hospital's medical services. This includes future system developments.

Keywords : Learning information in system, Programming, Understanding coding, SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงได้ด้วยความรู้และความช่วยเหลือเป็นอย่างดีจาก อาจารย์ที่ปรึกษาการค้นคว้าอิสระ อาจารย์ ผู้ช่วยศาสตราจารย์ ดร.ปัทมา เจริญพร ที่กรุณาเสียสละเวลาให้คำปรึกษา ชี้แนะวิธีการดำเนินงานที่เกี่ยวข้องกับการจัดทำ การค้นคว้าอิสระนี้ ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาของอาจารย์ และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณท่านคณาจารย์และวิทยากรทุกท่านที่มาถ่ายทอดความรู้ตลอดระยะเวลา 2 ปีทำให้สามารถนำความรู้ต่างๆ ที่ได้เรียนมาใช้ประโยชน์ในงานวิจัยนี้ได้อย่างเต็มที่ และความร่วมมือต่างๆ ของหลายท่าน ที่ให้การสนับสนุนผู้วิจัยตั้งแต่เริ่มต้นจนเสร็จสมบูรณ์ พร้อมทั้งขอบคุณเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือทั้งในเรื่องการเรียน งานกลุ่ม การสอบ และคำแนะนำด้านเทคนิคการเขียนโปรแกรม และช่วยหาข้อมูลที่จะนำมาใช้ในการค้นคว้าอิสระให้สำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณครอบครัว บิดา และมารดา ที่ให้การเลี้ยงดูสนับสนุนทุกอย่าง และอบรมส่งเสริมด้านการศึกษาเป็นอย่างดีตลอดมา

นายปรีดิ์เชษฐ์ ศรีอุทโยภาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ระบบสารสนเทศของโรงพยาบาล	4
2.2 ประโยชน์ของระบบสารสนเทศโรงพยาบาล	5
2.3 เครื่องมือที่ใช้ในการพัฒนาระบบ	6
2.3.1 Oracle	6
2.3.2 ความแตกต่างระหว่าง Oracle DB และ MySQL	6
2.3.3 Oracle Application Express	7
2.3.4 คำจำกัดความของ SQL Server	8
2.3.5 สรุปความแตกต่างที่สำคัญระหว่าง Oracle และ SQL Server	8
2.3.6 ความแตกต่างระหว่าง Oracle DB และ MySQL	9
2.3.7 Oracle Application Express	9
2.3.8 GOLDEN32 SQL	11
2.4 Structured Query Language	12
2.4.1 ความแตกต่างระหว่าง SQL และ Oracle SQL	13
2.5 ทฤษฎีการเขียน SQL และหลักการต่างๆในการใช้แต่ละ syntax	14
2.5.1 การเลือกข้อมูล	14
2.5.2 การรวมข้อมูล	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.5.3 การร่วมตาราง	15
2.5.4 การใช้คำสั่ง SUBQUERY	16
2.6 งานวิจัยที่เกี่ยวข้อง	17
2.6.1 การลดความแออัด ณ แผนกอุบัติเหตุและฉุกเฉิน	17
2.6.2 การจำลองสถานการณ์เพื่อลดเวลารอคอยของผู้ป่วย	18
บทที่ 3 วิธีการดำเนินงานวิจัย	19
3.1 สิ่งพื้นฐานข้อมูลมี และ Data Warehouse ใน HIS	19
3.2 การเตรียมข้อมูล และการกำหนด คำนิยามให้กับข้อมูล	20
3.2.1 Query ชุดที่ 1	21
3.2.2 Query ชุดที่ 2	24
3.2.3 Query ชุดที่ 3	26
3.3 เครื่องมือที่ใช้สำหรับการวิจัย	32
บทที่ 4 การดำเนินการสร้างและการอภิปรายผล	33
4.1 Cleaning Data & Summary Result	34
4.1.1 Cleaning Data & Summary Result Part 2	35
4.2 การสร้าง Table ที่ได้หลักจากการทำ Summary query	36
4.3 การสร้าง Web Application	38
4.3.1 ชั้นที่ 1 หลังจาก Login	38
4.3.2 หลังจากที่ได้ Application	40
4.3.3 การสร้างหน้า 16. OPD_TIME	41
4.4 การทดสอบโปรแกรม	44
บทที่ 5 สรุปผล	48
5.1 สรุปผลโปรแกรม	49
5.2 ข้อเสนอแนะ	50
เอกสารอ้างอิง	51
ประวัติผู้เขียน	52

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางเปรียบเทียบ PL / SQL	7
2.2 สรุป Oracle DB กับ MySQL	9
3.1 นิยามตัวชี้วัด	21
3.2 รายละเอียด field ข้อมูลใน DISEASE_WAREHOUSE@pmk	23
3.3 สรุปรูปแบบข้อมูลตัวอย่างที่ได้จากการเขียน Query ชุดที่ 1	24
3.4 สรุปรูปแบบข้อมูลที่จำเป็นจากตาราง DATE_DBFS	26
4.1 สรุปคอลัมน์ทั้งหมดที่จำเป็นต้องใช้	35
4.2 รายการหน้าต่างเมนูใน IPD Center	45



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างหน้าเว็บ APEX ORACLE	10
2.2 ตัวอย่างหน้าในการสร้างโปรแกรมต่างใน APEX ORACLE	11
2.3 ตัวอย่างโปรแกรม Golden 32	12
2.4 SQL JOINS	16
3.1 หน้าแรกของเว็บ APEX ORACLE	20
3.2 รูปแบบการเขียน Query ชุดที่ 1	22
3.3 ตัวอย่างวิธีการ Left join	23
3.4 ตัวอย่างบรรทัดตัวกรอง Placecode	23
3.5 รูปแบบข้อมูลที่ได้จาก Query ชุดที่ 1	24
3.6 รูปแบบการเขียน Query ชุดที่ 2	25
3.7 รูปแบบข้อมูลที่ได้จากตาราง Date_DBFS	25
3.8 รูปแบบข้อมูลที่ได้จาก Query ชุดที่สอง	26
3.9 รูปแบบการเขียน Query ชุดที่ 3	29
3.10 แสดงข้อมูล max_Diag_time	30
3.11 ตัวอย่างการแปลงค่าเวลาให้เป็นนาที่	30
3.12 ตัวอย่างชุดคำสั่งต่างๆในการแปลงค่า	31
4.1 รูปแบบข้อมูลที่ได้จาก Query ชุดที่ 3	34
4.2 รูปแบบข้อมูลสรุปคอลัมน์ทั้งหมดที่จำเป็น	35
4.3 หน้าเข้าใช้งาน APEX ORACLE	36
4.4 หลังจาก LOGIN เข้ามาจะพบหน้าต่างเมนูต่อไปนี้	36
4.5 ตัวอย่างหน้าสร้าง Authentication	37
4.6 เมนูต่างๆในการสร้าง Application	39
4.7 SQL Workshop	39
4.8 หน้า Create Application	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.9 การสร้าง Page และ Pages ต่างๆใน Application	41
4.10 ตัวอย่างหน้าระยะเวลาบริการผู้ป่วยนอก	41
4.11 โครงสร้างในหน้าโปรแกรมส่วน Content Body	42
4.12 Items ต่างๆ ในหน้าการสร้างหน้าระยะเวลาบริการผู้ป่วยนอก	43
4.13 ตัวอย่าง Query ในการสร้างกราฟแสดงผลหน้า opd_time	43
4.14 ตัวอย่างหน้า LOGIN	44
4.15 หน้าแรกของโปรแกรม IPD Center	45
4.16 ตัวอย่างหน้าระยะเวลาบริการผู้ป่วยนอกเมื่อลองใส่ค่าตัวค้นหา	46
4.17 ตัวอย่างข้อมูลเมื่อกดปุ่มเฉพาะข้อมูลที่นับตามเกณฑ์	47
4.18 ตัวอย่างเมื่อใส่ตัวกรอง ปี 2023 ของห้องตรวจ 0111	47
5.1 หน้าตรวจสอบการเข้าชม	48
5.2 ตัวอย่างรูปแบบข้อมูลสรุปผลตามห้องและเวลาที่ค้นหา	49
5.3 ช่องกรองข้อมูลช่วงเวลาตรวจรักษาคนไข้	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

การดูแลและตรวจผู้ป่วยในโรงพยาบาลเป็นเรื่องที่สำคัญอย่างยิ่งในการให้บริการทางการแพทย์ที่มีคุณภาพและเป็นประสิทธิภาพ ตัวชี้วัดหรือการประเมิน KPI (Key Performance Indicators) ระยะเวลาบริการผู้ป่วยนอกต่อแพทย์ 1 คนในแต่ละห้องตรวจเป็นเครื่องมือที่สำคัญในการวัดและปรับปรุงกระบวนการดังกล่าว เพื่อเพิ่มประสิทธิภาพในการดูแลและบริการต่อผู้รับบริการทางการแพทย์อย่างต่อเนื่องและมีคุณภาพที่ดีในโรงพยาบาลพระมงกุฎเกล้า ซึ่งเป็นโรงพยาบาลที่เป็นที่ยอมรับและเชื่อถือมาอย่างยาวนานในการให้บริการด้านสุขภาพแก่ประชาชนและประชาชนในพื้นที่ใกล้เคียง

ความสำคัญของการประเมินระยะเวลาบริการผู้ป่วยนอกต่อแพทย์ 1 คน ในแต่ละห้องตรวจในโรงพยาบาลพระมงกุฎเกล้านี้ อย่างแน่นอนจะมีผลต่อคุณภาพของการบริการที่โรงพยาบาลนี้มอบให้กับประชาชนโดยตรง การให้บริการที่ดีต้องเป็นเรื่องที่สำคัญเสมอในโรงพยาบาล เพื่อให้ผู้รับบริการรู้สึกว่าจะได้รับการดูแลอย่างใกล้ชิดและมีความพึงพอใจในผลการรักษาของตน โดยโปรแกรมที่จะได้สร้างต่อไปนี้จะเป็เครื่องมือที่ช่วยวัดความสำเร็จในการให้บริการเหล่านี้ได้อย่างชัดเจน ซึ่งมีหลายด้านที่สำคัญที่ควรให้ความสำคัญในการดูแลและตรวจผู้ป่วยต่อแพทย์ในแต่ละห้องตรวจ เช่น ด้านเวลาในการรอรับบริการ การตรวจผู้ป่วยต่อแพทย์ในระยะเวลาที่เหมาะสมเป็นสิ่งสำคัญในการสร้างความพึงพอใจให้กับผู้รับบริการ การให้บริการที่ไม่ตรงต่อเวลาอาจทำให้ผู้รับบริการรู้สึกไม่พอใจและส่งผลกระทบต่อภาพลักษณ์ของโรงพยาบาลในด้านนี้ โดยโปรแกรมที่จะพัฒนามีชื่อว่า IPD Center และหน้าหลักที่อยู่ในเนื้อหางานวิจัยนี้คือหน้า ระยะเวลาบริการผู้ป่วยนอก

นอกจากนี้การพัฒนาโปรแกรม IPD Center หน้าระยะเวลาบริการผู้ป่วยนอก ยังเป็นเครื่องมือที่ช่วยให้โรงพยาบาลสามารถวิเคราะห์ปัญหาและพบแนวทางการปรับปรุงกระบวนการในการดูแลและตรวจผู้ป่วยต่อแพทย์ ซึ่งส่งผลให้การบริการด้านสุขภาพเป็นไปในทิศทางที่ดีขึ้นเสมอ การใช้โปรแกรม IPD Center เป็นเครื่องมือในการสร้างเป้าหมายและควบคุมความสำเร็จของเป้าหมายเหล่านี้จึงเป็นสิ่งสำคัญในการนำทางและบริหารงานในโรงพยาบาลอย่างมีประสิทธิภาพ

ดังนั้นความสำเร็จในการดูแลและตรวจคนไข้ต่อแพทย์ในแต่ละห้องตรวจเป็นตัวบ่งชี้ที่ชัดเจนให้โรงพยาบาลพระมงกุฎเกล้ามีคุณภาพในการให้บริการทางการแพทย์อย่างมีความเป็นมาตรฐาน การประเมินตัวชี้วัดระยะเวลาในการตรวจคนไข้ต่อแพทย์ 1 คนในแต่ละห้องตรวจเป็นเรื่องที่สำคัญและไม่ควรถูกละเลยในกระบวนการดังกล่าว เพื่อสร้างความพึงพอใจและความเชื่อมั่นให้กับประชาชนที่มีความ

ต้องการรับการรักษาและดูแลจากโรงพยาบาลนี้อย่างต่อเนื่องและสำคัญอย่างยิ่ง หวังว่าผลงานนี้จะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำไปสู่การพัฒนาและปรับปรุงด้านการให้บริการสุขภาพที่มีคุณภาพและความเป็นมาตรฐานให้กับบุคลากรทุกคนในโรงพยาบาลในอนาคตอันดีขึ้นไปอีก

1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 เพื่อสร้างโปรแกรมในการวัดและประเมินความสำเร็จและประสิทธิภาพในการตรวจคนไข้
- 1.2.2 เพื่อศึกษาและวิเคราะห์ปัจจัยที่ส่งผลกระทบต่อระยะเวลาการตรวจผู้ป่วย

1.3 ขอบเขตของงานวิจัย

การใช้ข้อมูลที่มีอยู่ด้วย Oracle SQL ซึ่งเป็นระบบฐานข้อมูลหลักของโรงพยาบาลพระมงกุฎเกล้าในการวิเคราะห์และตรวจสอบระยะเวลาบริการผู้ป่วยนอกนั้นจะช่วยให้โรงพยาบาลพระมงกุฎเกล้าสามารถพัฒนาและปรับปรุงกระบวนการให้บริการทางการแพทย์ให้มีประสิทธิภาพและมีคุณภาพมากขึ้น ซึ่งจากวิเคราะห์ตัวข้อเสร็จจะนำข้อมูลที่ได้มาสร้าง Web Application ด้วยโปรแกรมที่สำเร็จรูปของ Oracle ได้แก่ Apex

1.3.1 การวัดเวลาตรวจผู้ป่วย งานวิจัยจะทำการวัดและบันทึกเวลาที่ใช้ในการตรวจผู้ป่วยในแต่ละห้องตรวจของโรงพยาบาลพระมงกุฎเกล้า ตั้งแต่เวลาที่ผู้รับบริการเข้าห้องตรวจจนถึงเวลาที่เสร็จสิ้นการตรวจ การวัดเวลานี้จำเป็นเพื่อประเมินประสิทธิภาพในการให้บริการและเพื่อให้ทราบเวลาเฉลี่ยในการตรวจผู้ป่วยที่แตกต่างกันในแต่ละห้องตรวจ

1.3.2 ใช้ฐานข้อมูลให้เกิดประโยชน์ งานวิจัยนี้จัดทำขึ้นเพื่อนำข้อมูลที่มีอยู่ในระบบฐานข้อมูลโรงพยาบาลมาใช้ให้เกิดประโยชน์ และสามารถนำมาพัฒนาเป็นเว็บแอปพลิเคชันเพื่อเพิ่มความสะดวกให้กับบุคลากรทางการแพทย์ของโรงพยาบาลพระมงกุฎเกล้า

1.3.3 ความพึงพอใจของผู้รับบริการ งานวิจัยนี้อาจทำการสำรวจความพึงพอใจของผู้รับบริการที่ได้รับการตรวจรักษา โดยใช้เครื่องมือสำหรับสำรวจความพึงพอใจหรือการนำเสนอแบบสอบถามเพื่อประเมินความพึงพอใจที่แตกต่างกันระหว่างห้องตรวจ

1.3.4 การปรับปรุงกระบวนการดูแลและตรวจคนไข้ งานวิจัยนี้จะเสนอข้อเสนอแนะหรือแนวทางในการปรับปรุงกระบวนการดูแลและตรวจผู้ป่วยในโรงพยาบาลพระมงกุฎเกล้า เพื่อเพิ่มประสิทธิภาพในการให้บริการและลดระยะเวลาในการตรวจผู้ป่วยที่มีความแตกต่างกันในแต่ละห้องตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่สำคัญของงานวิจัยนี้ คือ การสร้างหน้าโปรแกรมเพื่อแสดงข้อมูลต่างๆและเน้นรายละเอียดหน้าระยะบริการผู้ป่วยนอก โดยจะสร้างในรูปแบบ Web Application ด้วย Apex Oracle เพื่อที่จะให้บุคลากรทางการแพทย์เข้าถึงข้อมูลในที่ได้นำเสนอและนำข้อมูลที่ได้มาปรับปรุงกระบวนการดูแลผู้ป่วยในโรงพยาบาลพระมงกุฎเกล้าเพื่อให้ผู้ป่วยได้รับการตรวจที่มีคุณภาพและรวดเร็ว หลังจากได้พัฒนาโปรแกรมจนได้มีการนำไปใช้จริง ตั้งแต่ช่วงเดือนมิถุนายนของปี 2566 จนถึงปัจจุบัน ทำให้เห็นถึงความพึงพอใจ รวมถึงปัญหาต่างๆในโปรแกรมซึ่งได้มีการแก้ไขปรับปรุงโปรแกรมและได้พัฒนาโปรแกรมอย่างต่อเนื่องจนถึงปัจจุบัน

1.4.1 ช่วยให้ทีมงานทางการแพทย์และผู้บริหารสามารถตัดสินใจเรื่องที่สำคัญในการปรับปรุงกระบวนการดูแลและตรวจผู้ป่วย ตัวอย่างเช่น การกำหนดเป้าหมายเวลาในการตรวจผู้ป่วยที่สอดคล้องกับมาตรฐานในสายงานทางการแพทย์

1.4.2 สร้างเว็บ Application: Web Application เพื่อให้บุคลากรในโรงพยาบาลสามารถใช้งานได้อย่างสะดวก ปลอดภัย และช่วยให้สามารถตรวจสอบระยะเวลาบริการผู้ป่วยนอก และวัดผลการดำเนินงานของโรงพยาบาลในด้านต่างๆ ได้อย่างรวดเร็วและแม่นยำ ทำให้ทีมงานสามารถระบุและแก้ไขปัญหากระบวนการได้ทัน่วงที

1.4.3 ช่วยให้ทีมงานสามารถพัฒนากระบวนการดูแลและตรวจผู้ป่วยให้มีประสิทธิภาพมากขึ้น ประสิทธิภาพที่เพิ่มขึ้นนี้อาจส่งผลให้เกิดประโยชน์ทางการเศรษฐกิจ รวมถึงเพิ่มรายได้ให้กับโรงพยาบาลเนื่องจากการบริการที่ดีและรวดเร็ว

1.4.4 การสนับสนุนการตัดสินใจทางกฎหมายและนโยบาย ข้อมูลจากหน้าระยะเวลาบริการผู้ป่วยนอก ที่คัดลอกและตีความจาก Oracle SQL อาจเป็นแหล่งข้อมูลที่สำคัญในการตัดสินใจเกี่ยวกับนโยบายและกฎหมายที่เกี่ยวข้องกับระบบบริการสุขภาพ ทำให้สามารถมีการวางแผนการพัฒนาโดยให้เป็นไปในทิศทางที่ดีสำหรับโรงพยาบาลพระมงกุฎเกล้า

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและองค์ประกอบต่างๆในการนำมาพัฒนาระบบงานประกันคุณภาพ ด้านมาตรฐานโรงพยาบาลพระมงกุฎเกล้าซึ่งประกอบไปด้วย ระบบสารสนเทศของ โรงพยาบาล (Hospital Information System) เครื่องมือที่ใช้ในการพัฒนาระบบ ระบบสารสนเทศของ โรงพยาบาล

2.1 ระบบสารสนเทศของโรงพยาบาล (Hospital Information System)

ระบบสารสนเทศของโรงพยาบาล เป็นเครื่องมือในการจัดการฐานข้อมูลขนาดใหญ่ เพื่อให้ แพทย์ พยาบาลหรือบุคลากรที่เกี่ยวข้อง สามารถเข้าสู่ระบบและดึงข้อมูลเกี่ยวกับสุขภาพของผู้ป่วยได้ ทั้งจากฐานข้อมูลของโรงพยาบาลหรือจากสถานบริการสุขภาพอื่นๆ เพื่อวางแผนกระบวนการรักษา ผู้ป่วยและประเมินรูปแบบการดูแลผู้ป่วยได้อย่างมีประสิทธิภาพรวมถึงรักษาผู้ป่วยได้อย่างรวดเร็ว นอกจากนี้ยังสามารถนำข้อมูลทั้งหมดให้กับทางผู้บริหารไปประกอบการตัดสินใจในวางแผนกลยุทธ์และ จัดสรรทรัพยากรต่างๆ ภายในโรงพยาบาลได้อย่างคุ้มค่าและเหมาะสม

การนำเทคโนโลยีสารสนเทศสำหรับการบริหารจัดการมาประยุกต์เข้ากับกฎ ระเบียบ ข้อบังคับและมาตรฐานของระบบรับรองคุณภาพโรงพยาบาลแบบต่างๆ ตามที่โรงพยาบาลนั้นต้องการ เรียกว่า ระบบสารสนเทศโรงพยาบาล ระบบสารสนเทศของโรงพยาบาลเป็นนาระบบสารสนเทศ ย่อยหลายระบบมาใช้ร่วมกัน จึงต้องออกแบบให้อยู่ในระบบเครือข่าย เพื่อเชื่อมโยงสื่อสารข้อมูลของ ฝ่ายงานต่างๆ จึงทำให้เกิดระบบการทำงานมีประสิทธิภาพมากยิ่งขึ้นและประสานงานกันได้อย่างเป็น ระบบ เช่น

- ระบบการรักษาพยาบาล
- ระบบตรวจทางห้องปฏิบัติการ

การทำงานของระบบสารสนเทศที่มีประสิทธิภาพ ประกอบไปด้วยการใช้เทคโนโลยีเข้ามาทำงาน ร่วมกัน

โดยมี 4 องค์ประกอบสำคัญ ได้แก่

1. ฮาร์ดแวร์ (Hardware) ทั้งหมดที่เกี่ยวข้องกับการดำเนินงาน ไม่ว่าจะเป็นคอมพิวเตอร์ , โน้ตบุ๊ก , สมาร์ทโฟน, แท็บเล็ต, เครื่องแม่ข่าย (Server), เครื่องลูกข่าย (Clients) รวมไปถึงระบบเครือข่าย (Network) ที่ทำงานร่วมกันอยู่
2. ซอฟต์แวร์ (Software) แบ่งออกเป็น 3 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบบปฏิบัติการที่ทำงานอยู่ภายในฮาร์ดแวร์เช่น Windows, MacOS, Linux ที่ใช้งานบนคอมพิวเตอร์และโน้ตบุ๊ก รวมถึง Android และ IOS ที่ใช้งานผ่านสมาร์ทโฟนและแท็บเล็ต

- โปรแกรมอรรถประโยชน์ (Utility Program) ที่ช่วยเพิ่มสมรรถนะในการใช้เครื่องคอมพิวเตอร์ ซึ่งจะไม่ใช้โปรแกรมใช้งาน หรือโปรแกรมที่ผลิตงานออกมาด้วยตัวเองได้ แต่ช่วยให้การใช้โปรแกรมอื่นสะดวกขึ้น เช่น โปรแกรมจัดการไฟล์ (File Manager), โปรแกรมจัดเรียงพื้นที่เก็บข้อมูล (Disk Defragmenter), โปรแกรมป้องกันไวรัส (Anti-Virus Program), โปรแกรมบีบอัดไฟล์ (File Compression Utility) เป็นต้น

- โปรแกรมประยุกต์ (Application) ที่เกี่ยวข้องกับการจัดการข้อมูลสำหรับโปรแกรมสำหรับระบบโรงพยาบาล เช่น Hospital Information System (HIS), Electronic Medical Records (EMR), Enterprise Resource Planning (ERP) เป็นต้น

3. บุคลากรที่มีส่วนเกี่ยวข้องกับการใช้งานระบบ (People ware)

ผู้จัดการระบบ, นักวิเคราะห์ระบบ, โปรแกรมเมอร์ รวมไปถึงผู้ใช้งานระบบตามตำแหน่งหน้าที่สำหรับระบบโรงพยาบาล ผู้ใช้งานของโรงพยาบาลคือบุคลากรในโรงพยาบาล เช่น แพทย์พยาบาล และ รวมถึงบุคลากรที่ดำเนินงานภายในโรงพยาบาลในทุกแผนก

4. ข้อมูล (Data) ทั้งหมดภายในโรงพยาบาลที่ถูกจัดเก็บเอาไว้ในระบบ เช่น ข้อมูลประวัติผู้ป่วย ข้อมูลทางการแพทย์ ข้อมูลจากห้องปฏิบัติการ ข้อมูลเกี่ยวกับยา เป็นต้น

2.2 ประโยชน์ของระบบสารสนเทศโรงพยาบาล

ระบบสารสนเทศโรงพยาบาล มีประโยชน์อย่างยิ่งในทางเทคโนโลยีสำหรับอุตสาหกรรมสุขภาพ ซึ่งช่วยให้การจัดการข้อมูลด้านการดูแลสุขภาพมีประสิทธิภาพสูงสุด การนำระบบนี้ไปใช้จะช่วยให้การปรับปรุงคุณภาพการดูแลผู้ป่วย ลดต้นทุนการดำเนินงาน ทำให้ข้อมูลการบริหารไม่มีข้อผิดพลาด และทำให้กระบวนการจัดการภายในทั้งหมดมีระเบียบมากขึ้น ตัวอย่างประโยชน์ของระบบสารสนเทศโรงพยาบาลมีประโยชน์มากมายหลายด้าน ได้แก่

1. การจัดเก็บข้อมูลที่ต้องการและปลอดภัย ระบบสารสนเทศช่วยในการบันทึกข้อมูลผู้ป่วย, ประวัติการรักษา, และข้อมูลทางการแพทย์อื่นๆ ทำให้ข้อมูลมีความถูกต้องและสามารถเข้าถึงได้ตลอดเวลา

2. การเพิ่มประสิทธิภาพในการดูแลรักษา ระบบสารสนเทศช่วยลดเวลาในการค้นหาข้อมูล, ทำให้ทีมแพทย์และพยาบาลสามารถให้การรักษาที่เร็วและมีประสิทธิภาพมากขึ้น

3. การวิเคราะห์ข้อมูลสำหรับการตัดสินใจระบบสารสนเทศช่วยในการวิเคราะห์ข้อมูลทางการแพทย์, การจัดการทรัพยากร, และการวางแผนที่ดีขึ้น

4. การลดความผิดพลาด ระบบสารสนเทศช่วยลดความผิดพลาดที่เกิดขึ้นจากระบบการบันทึกข้อมูลที่ไม่เป็นระบบ, ทำให้การดูแลรักษามีความปลอดภัยมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 เครื่องมือที่ใช้ในการพัฒนาระบบ (System Development Tools)

ในบทย่อที่จะกล่าวถึงเครื่องมือที่ใช้ในการพัฒนาโครงการรวมถึงองค์ความรู้ที่เกี่ยวข้อง เช่น Sqldeveloper, Golden32SQL, ภาษา SQL, ความสามารถของ SQL ที่นำมาใช้ใน (ตารางเสมือน ชุดคำสั่ง และ การกำหนดเวลาประมวลผลชุดคำสั่ง) และ Oracle Apex เป็นต้น

2.3.1 Oracle

Oracleเป็นระบบจัดการฐานข้อมูลที่มีบทบาทสำคัญในการเป็นตัวกลางในการสื่อสารระหว่างผู้ใช้และฐานข้อมูล ทำให้ผู้ใช้สามารถดำเนินการจัดการฐานข้อมูลเชิงสัมพันธ์หรือที่เรียกว่า DBMS (Relational Database Management System) ได้อย่างมีประสิทธิภาพ และยังมี ความสามารถการทำงานในรูปแบบ Rational บางคุณสมบัติของ Object Oriented นอกจากการทำหน้าที่เป็นตัวกลางในการสื่อสารระหว่างผู้ใช้และฐานข้อมูล ระบบ Oracle แล้วยังช่วยให้ผู้ใช้งาน สามารถใช้ฐานข้อมูลได้อย่างสะดวกสบายมากยิ่งขึ้น โดยการเพิ่มความสะดวกเวลาค้นหาข้อมูลต่างๆ ภายในฐานข้อมูล โดยผู้ใช้ไม่จำเป็นต้องทราบถึงโครงสร้างภายในของฐานข้อมูลอีกต่อไปทำให้การใช้งานฐานข้อมูลเป็นเรื่องที่ง่ายและมีประสิทธิภาพมากยิ่งขึ้น

Oracle มีสองประเภท ได้แก่ Personal Oracle และ Oracle Serverโดยทั้งสองรูปแบบนี้มีลักษณะการใช้งานและคำสั่งที่คล้ายกัน แต่มีความแตกต่างกันอย่างสำคัญที่อธิบายไว้ดังนี้

1. Personal Oracle: นี้เป็นฐานข้อมูลที่ติดตั้งบนเครื่องคอมพิวเตอร์ของผู้ใช้เอง ซึ่งผู้ใช้งานจะต้องทำงานกับ Oracle ได้เฉพาะบนเครื่องนั้นเท่านั้น กล่าวคือ ไม่สามารถเข้าถึงข้อมูลจากเครื่องอื่นได้ โดยผู้ใช้งานจะต้องนั่งทำงานกับ Oracle ที่หน้าเครื่องที่ติดตั้งไว้

2. Oracle Server: ซึ่งเป็นฐานข้อมูลส่วนเซิร์ฟเวอร์ที่ติดตั้งอยู่บนเครื่องคอมพิวเตอร์ และยืนยันการขออนุญาตจากผู้ใช้งานหรือจัดการข้อมูลจากเครื่องอื่นๆ ต่อมาในส่วนของเครื่องอื่นๆ ได้มีการเรียกใช้ข้อมูลจากฐานข้อมูล Oracle Server ถูกเรียกว่า "โคลเอนต์" ซึ่งนำมาแสดงถึงความสามารถในการเชื่อมต่อและให้บริการฐานข้อมูลให้กับผู้ใช้หลาย ๆ คนพร้อมกัน

ดังนั้น หากต้องการให้มีผู้ใช้งานหลายคนสามารถเข้าถึงและจัดการข้อมูลในฐานข้อมูลได้พร้อมกัน จะต้องเลือกรูปแบบ Oracle Server เนื่องจากมีความสามารถในการให้บริการฐานข้อมูลแบบพร้อมเพียงสำหรับผู้ใช้งานหลายคนได้อย่างมีประสิทธิภาพสูงสุด

2.3.2 ความแตกต่างระหว่าง Oracle และ SQL server

ในบรรดาระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ที่นิยมใช้กันมากคือ Oracle และ SQL Server มีความแตกต่างพื้นฐานมากมายระหว่าง Oracle และ SQL Server หนึ่งในความแตกต่างที่สำคัญที่สุดระหว่างพวกเขา คือ ภาษาที่พวกเขาใช้ Oracle ใช้ PL / SQL (กระบวนการภาษา SQL) และ SQL Server ใช้ T-SQL เช่น Transact-SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางเปรียบเทียบ PL / SQL

ขั้นพื้นฐาน	ภาษาที่ใช้โดย Oracle คือ PL / SQL (ภาษาขั้นตอน / SQL)	ภาษาที่ใช้โดย SQL Server คือ T-SQL (Transact-SQL)
การซื้อขาย	ใน Oracle จะไม่มีการทำธุรกรรมจนกว่า DBA จะออกคำสั่ง COMMIT อย่างชัดเจน	หากไม่ได้ระบุคำสั่ง BEGIN TRANSACTION และ COMMIT คำสั่งจะเรียกใช้งาน แต่ละคำสั่ง
องค์กร	มีการแชร์ฐานข้อมูลระหว่าง schema และผู้ใช้ทั้งหมด	ฐานข้อมูลไม่แบ่งปันระหว่างผู้ใช้
รองรับระบบปฏิบัติการ	Windows, Linux, Solaris, HP-UX, OS X, z / OS, AIX	Windows และ Linux
ความซับซ้อน	ซับซ้อน แต่ทรงพลัง	เรียบง่ายและใช้งานง่าย

2.3.3 คำจำกัดความของ Oracle

Oracle เป็นระบบฐานข้อมูลเชิงสัมพันธ์ที่รองรับคุณสมบัติหลักทั้งหมดของ SQL นอกจากนี้ Oracle ยังรองรับภาษาอื่น ๆ อีกมากมาย ระบบปฏิบัติการหลายระบบรองรับ Oracle เช่น Windows, Linux, Solaris, HP-UX, OS X, z / OS, AIX ภาษาต้นฉบับที่ใช้โดย Oracle คือ PL / SQL เช่น กระบวนการ Language SQL ซึ่งเป็นส่วนเสริมของ SQL Oracle จัดทำ Packet ที่สามารถสรุปขั้นตอนการทำงานและตัวแปรของฐานข้อมูลเพื่อสร้างหน่วยเดียว

ใน Oracle การดำเนินการตามคำสั่งหรือคำสั่งการเปลี่ยนแปลงจะเกิดขึ้นกับหน่วยความจำเท่านั้น ไม่ทำการเปลี่ยนแปลงจนกว่า DBA (ผู้ดูแลระบบฐานข้อมูล) จะออกคำสั่ง COMMIT อย่างชัดเจน ทันทีที่คำสั่ง COMMIT ถูกดำเนินการการเปลี่ยนแปลงจะเกิดขึ้นกับ ดิสก์ และคำสั่งหลังจากที่ COMMIT เริ่มทำธุรกรรมใหม่

ที่มาฐานข้อมูลของ Oracle จัดกลุ่มวัตถุฐานข้อมูลทั้งหมดวัตถุฐานข้อมูลเหล่านี้ ใช้อย่างร่วมกัน ระหว่างสกีมาและผู้ใช้ฐานข้อมูล Oracle แม้ว่าวัตถุฐานข้อมูลจะถูกแชร์ในหมู่ผู้ใช้ทั้งหมด แต่ผู้ใช้สามารถถูก จำกัด ไม่ให้เข้าถึงฐานข้อมูลผ่านบทบาทหรือการอนุญาต Oracle นั้น ซับซ้อน แต่เป็น RDBMS ที่ทรงพลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 คำจำกัดความของ SQL Server

เช่นเดียวกับ Oracle SQL Server ยังเป็นระบบฐานข้อมูลเชิงสัมพันธ์ ภาษาที่ใช้โดย SQL Server คือ T-SQL เช่น Transact-SQL SQL Server รองรับระบบปฏิบัติการ Windows เท่านั้น เช่นเดียวกับ Oracle SQL Server ไม่มี แพ็คเกจที่ จะรวม Procedure and Function ทั้งหมดของ ฐานข้อมูลใน SQL Server หากไม่ได้ระบุคำสั่ง BEGIN TRANSACTION และ COMMIT คำสั่งแต่ละ คำสั่งจะถูกดำเนินการและยืนยันเป็นรายบุคคล ในกรณีที่หากพบข้อผิดพลาดใด ๆ ในระหว่างการ ดำเนินการตามคำสั่งทั้งหมดการย้อนกลับจะกลายเป็นเรื่องยากเนื่องจากคำสั่งที่กำหนดไว้จะไม่ ย้อนกลับ หากใช้คำสั่ง ROLLBACK อย่างระมัดระวังสามารถลดความเสียหายของข้อมูล SQL Server ไม่แชร์ ฐานข้อมูลกับผู้ใช้ ฐานข้อมูลถูกเก็บไว้ในดิสก์ที่ไม่ได้แชร์ในเซิร์ฟเวอร์ หากผู้ใช้รายใด ต้องการเข้าถึงฐานข้อมูลผู้ใช้นั้นจะได้รับรหัสล็อกอิน

2.3.5 สรุปความแตกต่างที่สำคัญระหว่าง Oracle และ SQL Server

- ภาษาที่ใช้โดย Oracle คือ PL / SQL เช่น Procedure ภาษา SQL ในขณะที่ SQL Server ใช้ T-SQL เช่น Transact-SQL นอกจากนี้ DBA จะออกคำสั่ง COMMIT จะไม่มีการทำ ธุรกรรมใด ๆ ในทางกลับกันใน SQL ถ้าไม่ได้ระบุการเริ่มต้นการทำธุรกรรมและความมุ่งมั่นแต่ละคำสั่ง จะถูกดำเนินการและยืนยันเป็นรายบุคคล
- ใน Oracle นั้นฐานข้อมูลจะถูกแชร์ใน Schema SQL และผู้ใช้ทั้งหมดแม้ว่าผู้ใช้ สามารถถูก จำกัด ให้เข้าถึง Schema SQL หรือตารางบางอย่างผ่านบทบาทและการอนุญาต ในทาง ตรงกันข้ามในเซิร์ฟเวอร์ SQL ฐานข้อมูลจะถูกเก็บไว้เป็นส่วนบนเซิร์ฟเวอร์ผู้ใช้ได้รับการเข้าสู่ระบบ เพื่อให้การเข้าถึงฐานข้อมูล
- ใน Oracle กระบวนการฟังก์ชันและตัวแปรจะถูกห่อหุ้มในแพ็คเกจ อย่างไรก็ตาม SQL ไม่มีแพ็คเกจ Oracle รองรับระบบปฏิบัติการหลายระบบเช่น Windows, Linux, Solaris, HP-UX, OS X, z / OS, AIX ในทางกลับกัน SQL รองรับ Windows และ Linux
- Oracle มีความซับซ้อน แต่มีประสิทธิภาพในขณะที่ SQL นั้นง่ายและใช้งานง่าย ไวยากรณ์ของคำสั่งที่ใช้โดย Oracle และ SQL แตกต่างกัน

2.3.6 ความแตกต่างระหว่าง Oracle DB และ MySQL

MySQL เป็นระบบจัดการฐานข้อมูลที่รู้จักกันดีซึ่งใช้เพื่อจัดการฐานข้อมูลเชิงสัมพันธ์ Oracle Corporation ให้การสนับสนุนสำหรับฐานข้อมูล Open source นี้ เป็นกระบวนการการ จัดการฐานข้อมูลที่รวดเร็ว ปรับขนาดได้ และใช้งานง่ายเมื่อเปรียบเทียบกับ Microsoft SQL Server และ Oracle Database มักใช้ร่วมกับสคริปต์ PHP เพื่อสร้างแอปพลิเคชันระดับองค์กรหรือเซิร์ฟเวอร์ บนเว็บที่มีประสิทธิภาพและแบบไดนามิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MySQL AB ซึ่งเป็นบริษัทสัญชาติสวีเดน ได้พัฒนาและสนับสนุน และเขียนด้วยภาษาคอมพิวเตอร์ C และ C++ ธุรกิจขนาดเล็กและขนาดใหญ่จำนวนมากใช้ MySQL รองรับระบบปฏิบัติการที่หลากหลาย รวมถึง Windows, Linux, macOS และอื่นๆ โดยใช้ภาษาการเขียนโปรแกรม C, C++ และ Java

ตารางที่ 2.2 สรุป Oracle DB กับ MySQL

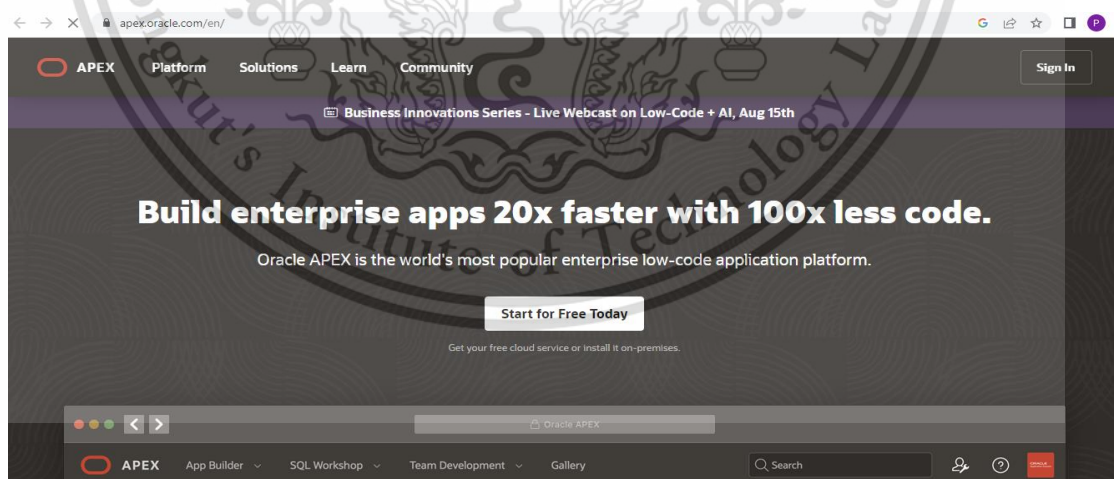
Oracle DB	MySQL
Oracle เป็นฐานข้อมูลทางเลือกสำหรับการปรับใช้ขนาดใหญ่	MySQL เป็นฐานข้อมูลที่ใช้โดยทั้งบริษัทขนาดเล็กและขนาดใหญ่
CHAR, VARCHAR2, NCHAR และ NVARCHAR2 เป็นอักขระที่ Oracle รองรับ	MySQL รองรับอักขระสองตัวเท่านั้น: CHAR และ VARCHAR
สำรองข้อมูล สำรองข้อมูลด่วน นำเข้า ส่งออก และขั้นตอนการสำรองข้อมูลอื่น ๆ ใน oracle	เฉพาะ mysqlhotcopy และ mysqldump เท่านั้นที่มีให้เป็นตัวเลือกรสำรอง
Oracle ไม่รองรับค่า Null	ค่า Null รองรับโดย MySQL
มันสนับสนุน XML ได้รับการปล่อยตัวในปี 1980	ไม่รองรับ XML ได้รับการปล่อยตัวในปี 1995
Oracle มีทั้งความสามารถในการล๊อคตารางและแถว	MySQL เสนอคุณสมบัติการล๊อคตารางเท่านั้น
Oracle รองรับทั้ง SQL และ PL/SQL	MySQL รองรับเฉพาะภาษา SQL

2.3.7 Oracle Application Express

การพัฒนาเว็บแอปพลิเคชันในปัจจุบันมีเครื่องมือสำหรับพัฒนาอยู่มากมายแต่ที่นิยมใช้กันมาก คือ ภาษา PHP ซึ่งมีข้อดีเนื่องจากเป็น Free License มีความยืดหยุ่นและง่ายต่อการพัฒนา ถึงแม้ว่าจะมีความง่ายในการพัฒนาแต่โปรแกรมเมอร์ยังต้องมีความรู้ความเข้าใจในโครงสร้างของภาษา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

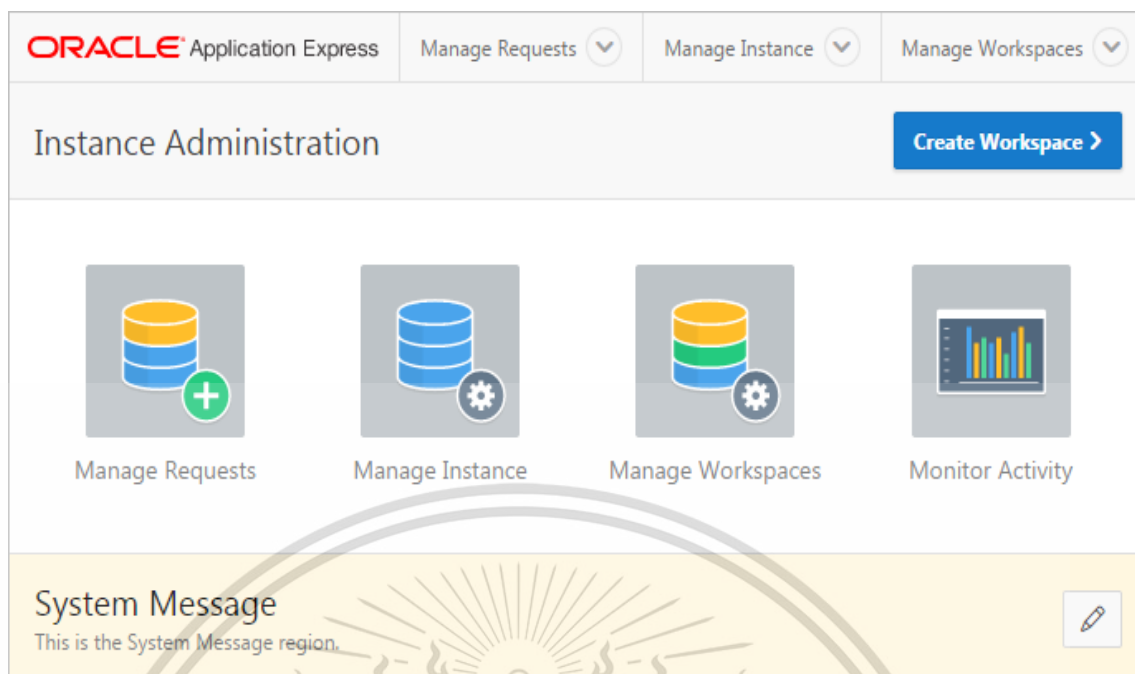
และถ้าต้องเชื่อมต่อกับฐานข้อมูลก็ควรที่จะต้องเขียนในส่วนของการเชื่อมต่อกับฐานข้อมูล รวมถึงภาษา SQL ที่ต้องมีความชำนาญในระดับหนึ่งและยังไม่จำเป็นรวมถึงความรู้ในภาษา HTML ที่จำเป็นต่อการพัฒนาแอปพลิเคชันบนเว็บอีก

ดังนั้นการที่จะพัฒนาแอปพลิเคชันบนเว็บในปัจจุบันนั้นเพื่อใช้ในทางธุรกิจก็คงจะไม่ยุ่งยาก เนื่องจากความซับซ้อนของธุรกิจทำให้ต้องออกแบบตาราง (Table) สำหรับเก็บข้อมูลมากขึ้น เงื่อนไขและการใช้งานของโปรแกรมก็มากขึ้นทำให้ต้องมา Coding ถ้าจะมีเครื่องมือที่ดี ที่ได้รวมเอาความยุ่งยากของภาษาโปรแกรม การเชื่อมต่อกับฐานข้อมูล ความเข้าใจในภาษา SQL และภาษา HTML ไว้เข้าด้วยกัน แล้วสร้างเป็น Framework ให้สามารถใช้งานได้ง่ายโดย Coding น้อยที่สุดหรือแทบไม่ต้อง Coding เลยคงจะดีไม่น้อย ปัจจุบันเครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชันที่ดีนั้นสามารถที่จะใช้เลย Oracle ได้ออกเครื่องมือสำหรับพัฒนาที่ชื่อว่า Oracle Application Express (APEX) ขณะที่กำลังเขียนอยู่นี้เป็น Version 3.2 พัฒนามาจากเดิมที่ใช้ชื่อว่า HTML DB ซึ่งเป็น Software ที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันบนฐานข้อมูล Oracle โดยติดตั้งมากับ Oracle 11g ด้วยเครื่องมือนี้ผู้พัฒนาแทบจะไม่ต้อง Coding เลย ทำให้ไม่ต้องไปเสียเวลาศึกษาโครงสร้างของภาษาที่ใช้สำหรับเขียนโปรแกรมใด ๆ นอกจากภาษา SQL ซึ่งเป็นสิ่งจำเป็นพื้นฐานสำหรับนักพัฒนา ผู้พัฒนาสามารถสร้างแอปพลิเคชันได้ง่าย ด้วยการใช้ Wizard แบบ GUI แล้วคลิกเลือกไปตามขั้นตอน ก็สามารถสร้างแอปพลิเคชันได้หลายรูปแบบ ไม่ว่าจะเป็นหน้าจอ Data Entry หรือรายงานต่างๆ ได้อย่างสวยงามโดยไม่ต้องรู้ภาษา HTML แม้แต่น้อย และสามารถเริ่มใช้งานได้ฟรี โดยไม่มีค่า License นับเป็นเครื่องมือที่ช่วยให้พัฒนาเว็บแอปพลิเคชันได้อย่างรวดเร็ว



รูปที่ 2.1 ตัวอย่างหน้าเว็บ APEX ORACLE

รูปที่ 2.1 หน้าเว็บ APEX ORACLE ที่เปิดให้ผู้ที่อยากลองใช้งานสามารถไปสมัครใช้งาน ซึ่งสามารถใช้ได้ฟรีและอัพเนื้อที่เก็บข้อมูลเพิ่มได้แต่ต้องมีการจ่ายรายปี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ตัวอย่างหน้าในการสร้างโปรแกรมต่างๆใน APEX ORACLE

2.3.8 GOLDEN32 SQL

Golden32 เป็นโปรแกรมที่ใช้ในการเขียนและจัดการฐานข้อมูลแบบ SQL (Structured Query Language) ซึ่งเป็นภาษาที่ใช้ในการจัดการข้อมูลในฐานข้อมูล โดยส่วนใหญ่จะใช้ในการสร้าง แก้ไข ลบ และสอบถามข้อมูลในฐานข้อมูลของระบบสารสนเทศต่างๆ รวมถึงการจัดการข้อมูลในฐานข้อมูลสำหรับแอปพลิเคชันต่างๆ อาทิเช่น ระบบบริหารจัดการลูกค้า (CRM) ระบบบัญชี ระบบจัดการคลังสินค้า เป็นต้น

Golden32 เป็นเครื่องมือที่ช่วยในการเขียนและทำงานกับ SQL โดยมีพีเจอรและอินเตอร์เฟซที่ช่วยให้ผู้ใช้สามารถสร้างคำสั่ง SQL ได้อย่างสะดวกและรวดเร็ว โปรแกรมนี้ช่วยให้ผู้ใช้งานที่ไม่มีความเชี่ยวชาญในการเขียน SQL ก็สามารถดำเนินการต่างๆ กับฐานข้อมูลได้ง่ายขึ้น อีกทั้งยังมีการจัดการข้อมูลอย่างสมบูรณ์ เช่น การเปิด-ปิดการเชื่อมต่อกับฐานข้อมูล การดูแลโครงสร้างของฐานข้อมูล การดำเนินการ CRUD (Create, Read, Update, Delete) ข้อมูล และการสร้างคำสั่ง SQL อื่นๆ ได้อย่างสะดวกมากขึ้นด้วย

โปรแกรมประเภทนี้มักถูกนำมาใช้ในองค์กรหรือบริษัทที่มีความต้องการจัดการข้อมูลในระบบสารสนเทศอย่างมาก เช่น การเก็บข้อมูลลูกค้า การจัดการการสั่งซื้อ การวิเคราะห์ข้อมูลการขาย เป็นต้น โดยสรุป Golden32 เป็นโปรแกรมที่ช่วยในการเขียนและจัดการคำสั่ง SQL และการจัดการข้อมูลในฐานข้อมูลอย่างมีประสิทธิภาพและสะดวกสำหรับผู้ใช้งานที่ไม่มีความเชี่ยวชาญในการเขียน SQL ในการทำงานกับระบบสารสนเทศและฐานข้อมูลต่างๆ ในองค์กรหรือบริษัทที่มีความ

ต้องการในด้านนี้ตัวอย่างโปรแกรม Golden32 รูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot displays the Golden Software interface. The main window shows an SQL query in the SQL Builder pane. The query is as follows:

```

select A.hn,A.min_main_date,A.max_main_date,A.opd_date,A.TEST_DATE,
      B.Other_date,(A.opd_date - B.Other_date) as def_other_date,All_drug
from (select hn,min_main_date,max_main_date,opd_date,TEST_DATE
      from (select A.hn,
                A.min_main_date,A.max_main_date,
                A.START_AND_LAST,
                B.opd_date,
                (B.opd_date - A.max_main_date) as TEST_DATE,
                case when B.opd_date - A.max_main_date >= '365' then 'CASE' else 'G' end as IN_CASE_OR_G
          from (select hn,min_main_date,max_main_date,(max_main_date - min_main_date) as START_AND_LAST
            from (select hn,min(main_date) as min_main_date,max(main_date) as max_main_date

```

The results pane shows a table with the following data:

#	HN	MIN_MAIN_DATE	MAX_MAIN_DATE	OPD_DATE	TEST_DATE	OTHER_DATE	DEF_OTHER_DATE
1	11101016	11/10/2016 00:00:00	07/04/2020 00:00:00	03/07/2023 00:00:00	1182	28/06/2016 00:00:00	28/06/2016 00:00:00
2	19042019	19/04/2019 00:00:00	19/04/2019 00:00:00	21/07/2023 00:00:00	1554	09/05/2022 00:00:00	
3	13062018	13/06/2018 00:00:00	10/01/2019 00:00:00	10/08/2023 00:00:00	1673	08/06/2023 00:00:00	
4	07062017	07/06/2017 00:00:00	27/10/2021 00:00:00	04/07/2023 00:00:00	615	08/02/2017 00:00:00	

The right pane shows the SQL Builder with a list of functions and a table structure for 'DOCTOR'.

รูปที่ 2.3 ตัวอย่างโปรแกรม Golden 32

2.4 Structured Query Language (SQL)

Structured Query Language (SQL) คือภาษาที่ใช้ในการจัดการฐานข้อมูล (Database) ซึ่งได้รับการยอมรับมากที่สุดในโลกภาษาหนึ่งและได้รับการยอมรับในมาตรฐาน American National Standards Institute (ANSI) สามารถใช้งานร่วมกับเว็บไซต์ ระบบฐานข้อมูล SQL Server ไปจนถึงการสร้างระบบวิเคราะห์ข้อมูลด้วยตนเอง ผู้ใช้งานสามารถทำการดึงข้อมูล คำนวณข้อมูล ไปจนถึงหาความเชื่อมโยงข้อมูลที่เหมือนกันในฐานข้อมูล (Relational Database) การทำงานของ Structured Query Language (SQL)

SQL ถือเป็นภาษาที่ไม่ซับซ้อน เข้าใจง่าย สามารถใช้งานได้หลายรูปแบบ โดยจะมีการทำงานหลักๆ ด้วยกัน 4 ประเภท ดังนี้

- Select query ใช้สำหรับเลือกข้อมูล
- Update query ใช้สำหรับเปลี่ยนแปลงข้อมูล
- Insert query ใช้สำหรับการเพิ่มข้อมูล
- Delete query ใช้สำหรับการลบข้อมูล

นอกเหนือจากคำสั่งหลัก 4 ประเภท ยังมีคำสั่ง SQL เพิ่มเติมอีก 3 ประเภท

- Data Definition Language (DDL) คำสั่งสำหรับการสร้างฐานข้อมูล เช่น CREATE, DROP
- Data Manipulation Language (DML) คำสั่งสำหรับการจัดการฐานข้อมูล เช่น SELECT, INSERT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Control Language (DCL) คำสั่งสำหรับการอนุมัติและกำหนดสิทธิ์ต่างๆ ในฐานข้อมูล เช่น GRANT, REVOKE

ประโยชน์ของ SQL

- ใช้เพื่อสร้างฐานข้อมูลตารางแสดงผลข้อมูล
- ใช้ในการจัดการฐานข้อมูล เช่น การเพิ่ม เปลี่ยนแปลง ปรับรูปแบบ จนถึงการลบข้อมูล
- ใช้เพื่อเรียกใช้ข้อมูลค้นหาข้อมูลที่ต้องการไปจนถึงการอนุมัติการเข้าถึงฐานข้อมูลของบุคคลต่างๆ
- SQL รองรับปริมาณข้อมูลมหาศาล โดยฐานข้อมูลจะมีความเสถียร มีความไวในการประมวลผล
- ง่ายต่อการดึงข้อมูลมานำเสนอสามารถนำโปรแกรมที่ใช้ร่วมกับภาษา SQL ที่หลากหลาย

2.4.1 ความแตกต่างระหว่าง SQL และ Oracle SQL

เมื่อพูดถึง SQL และ Oracle SQL นั้นเป็นสองระบบที่มีความแตกต่างที่สำคัญ ไม่ว่าจะเป็นในเชิงประสิทธิภาพของการประมวลผล หรือความสามารถในการจัดการข้อมูลที่ซับซ้อน ดังนี้คือบทความที่เปรียบเทียบความแตกต่างระหว่าง SQL และ Oracle SQL

SQL (Structured Query Language) หรือ ANSI SQL เป็นภาษาคิวรีที่มีความสามารถในการจัดการฐานข้อมูลแบบมาตรฐาน ซึ่งใช้ได้กับส่วนใหญ่ของระบบการจัดการฐานข้อมูลที่มีอยู่ในตลาด สิ่งที่แตกต่างกันกับ Oracle SQL คือ SQL มักจะไม่มีความสามารถที่มากพอในการจัดการกับการประมวลผลที่ซับซ้อน และบางครั้งอาจมีความจำกัดในการดำเนินการทางธุรกิจที่ซับซ้อน

Oracle SQL คือภาษาคิวรีที่เฉพาะเจาะจงไปที่ระบบการจัดการฐานข้อมูล Oracle ซึ่งมีความสามารถที่มากขึ้นในการจัดการกับข้อมูลที่มีความซับซ้อนและการประมวลผลที่จำเป็น โดย Oracle SQL มีการเพิ่มฟีเจอร์และความสามารถเพิ่มเติมเช่น การจัดการตัวแปรแบบคุณสมบัติ การจัดการตารางแบบพิเศษ การจัดการกับคิวรีที่มีประสิทธิภาพสูง และการใช้งานที่เกี่ยวข้องกับคลัสเตอร์และแอปพลิเคชันอื่น ๆ ที่เกี่ยวข้องกับ Oracle database SQL และ Oracle SQL มีความแตกต่างกันอย่างมีนัยสำคัญ โดย Oracle SQL มีความสามารถที่เฉพาะเจาะจงและมีประสิทธิภาพสูงกว่า SQL ในการจัดการข้อมูลที่ซับซ้อนและการประมวลผลทางธุรกิจ หากต้องการใช้งานที่มีความยืดหยุ่นสูงและการจัดการฐานข้อมูลที่ซับซ้อนอย่างมีประสิทธิภาพ Oracle SQL อาจเป็นตัวเลือกที่ดีกว่า SQL อย่างไรก็ตาม การเลือกใช้ระบบที่เหมาะสมจะขึ้นอยู่กับความต้องการและความซับซ้อนของโครงการหรือธุรกิจที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ทฤษฎีการเขียน SQL และหลักการต่างๆในการใช้แต่ละ syntax

ส่วนนี้จะในรูปแบบ Syntax SQL พื้นฐานต่างๆที่ใช้ในการเขียนโปรแกรมรวมถึงเป็นรูปแบบการเขียนที่จำเป็นต้องใช้ในการสร้างโปรแกรม IPD Center ทั้งนี้จึงจะขอยกรูปแบบที่จำเป็นมาขยายความ

2.5.1 การเลือกข้อมูล

การใช้คำสั่ง 'SELECT' เป็นการเลือกข้อมูลจากตารางในฐานข้อมูล ตัวอย่าง

SELECT * FROM employees; -- เลือกข้อมูลทั้งหมดจากตาราง employees

SELECT first_name, last_name FROM employees; -- เลือกชื่อและนามสกุลเฉพาะ

การเรียงลำดับข้อมูล (ORDER BY)การใช้คำสั่ง 'ORDER BY' เพื่อเรียงลำดับผลลัพธ์ตามคอลัมน์ที่กำหนด ตัวอย่าง

SELECT * FROM products ORDER BY price DESC; -- เรียงลำดับราคาจากสูงไปต่ำ

SELECT * FROM customers ORDER BY last_name, first_name; -- เรียงลำดับตามนามสกุลและชื่อ

การกรองข้อมูล (WHERE)การใช้คำสั่ง 'WHERE' เพื่อกรองข้อมูลเฉพาะแถวที่ตรงกับเงื่อนไขที่กำหนด ตัวอย่าง

SELECT * FROM orders WHERE order_date > '2023-01-01'; -- เลือกคำสั่งที่สร้างหลังจาก 2023-01-01

SELECT * FROM employees WHERE department = 'HR' AND salary > 50000; -- เลือกพนักงานในแผนก HR ที่เงินเดือนมากกว่า 50000

2.5.2 การรวมข้อมูล

การใช้ฟังก์ชันรวม (SUM, AVG, COUNT)การใช้ฟังก์ชันรวมใน SQL เพื่อคำนวณค่ารวม (SUM), ค่าเฉลี่ย (AVG), หรือจำนวนแถว (COUNT) ตัวอย่าง

SELECT SUM(total_price) FROM orders; -- คำนวณราคารวมของคำสั่งทั้งหมด

SELECT AVG(age) FROM employees; -- คำนวณอายุเฉลี่ยของพนักงาน

SELECT COUNT(*) FROM products; -- นับจำนวนรายการสินค้าทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดกลุ่มข้อมูล (GROUP BY) การใช้คำสั่ง 'GROUP BY' เพื่อกลุ่มข้อมูลตามคอลัมน์ที่กำหนดและใช้ฟังก์ชันรวมในการคำนวณค่ารวมหรือค่าเฉลี่ยในแต่ละกลุ่ม ตัวอย่าง

```
SELECT department, AVG(salary) FROM employees GROUP BY department;
```

-- คำนวณเงินเดือนเฉลี่ยในแต่ละแผนก

```
SELECT category, COUNT(*) FROM products GROUP BY category;
```

-- นับจำนวนสินค้าในแต่ละหมวดหมู่

การกรองกลุ่มข้อมูล (HAVING) การใช้คำสั่ง 'HAVING' เพื่อกรองกลุ่มข้อมูลที่คำนวณจากฟังก์ชันรวม ตัวอย่าง

```
SELECT department, AVG(salary) FROM employees GROUP BY department HAVING AVG(salary) > 50000;
```

-- เลือกแผนกที่มีเงินเดือนเฉลี่ยมากกว่า 50000

```
SELECT category, COUNT(*) FROM products GROUP BY category HAVING COUNT(*) > 10;
```

-- เลือกหมวดหมู่ที่มีสินค้ามากกว่า 10 รายการ

2.5.3 การร่วมตาราง

INNER JOIN, LEFT JOIN, RIGHT JOIN การร่วมตารางเป็นเทคนิคสำคัญใน SQL เพื่อรวมข้อมูลจากตารางหลายๆ ตารางเข้าด้วยกัน มีหลายวิธีในการร่วมตาราง INNER JOIN ใช้เพื่อรวมแถวที่ตรงกันในทั้งสองตาราง แถวที่ไม่ตรงกันจะไม่ถูกรวมเข้ามาด้วย ตัวอย่าง

```
SELECT orders.order_id, customers.customer_name
```

```
FROM orders
```

```
INNER JOIN customers ON orders.customer_id = customers.customer_id;
```

LEFT JOIN (หรือ LEFT OUTER JOIN) ใช้เพื่อรวมแถวจากตารางซ้าย (ตารางแรก) และเอาแถวที่ตรงกันจากตารางขวา (ตารางที่ร่วม) และแถวจากตารางซ้ายที่ไม่มีคู่ตรงจากตารางขวาจะถูกรวมเข้ามาและเติมค่า NULL

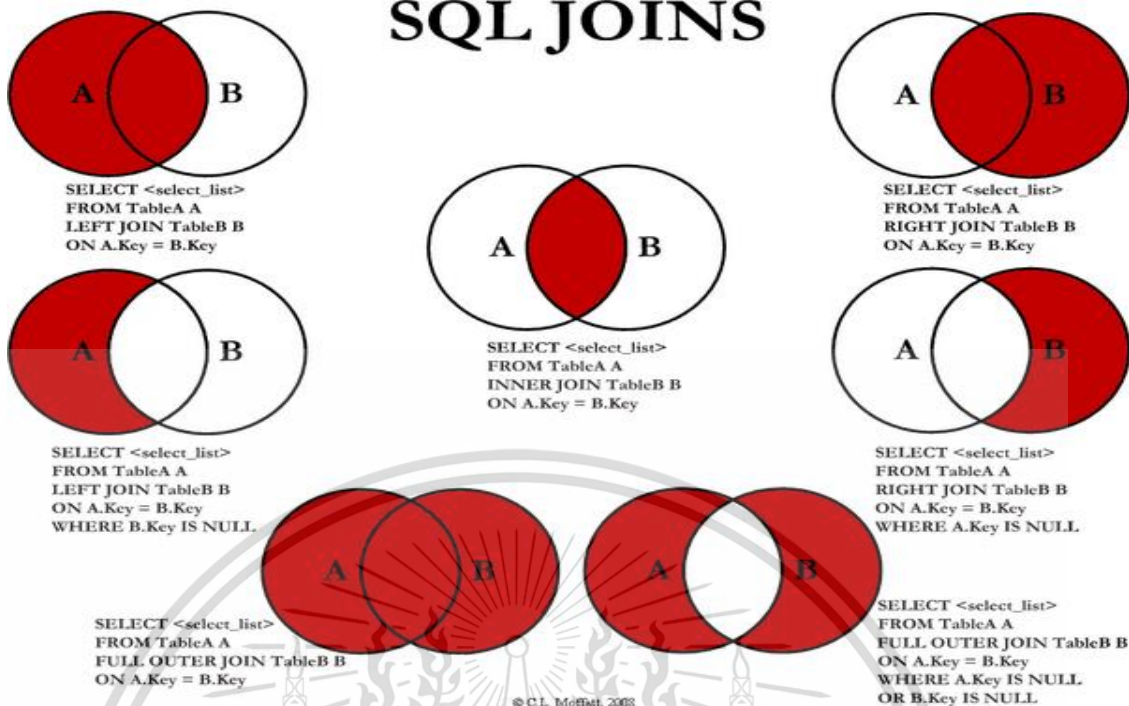
```
SELECT customers.customer_name, orders.order_id
```

```
FROM customers
```

```
LEFT JOIN orders ON customers.customer_id = orders.customer_id;
```

RIGHT JOIN (หรือ RIGHT OUTER JOIN) ใช้เพื่อรวมแถวจากตารางขวา (ตารางที่ร่วม) และเอาแถวที่ตรงกันจากตารางซ้าย (ตารางแรก) และแถวจากตารางขวาที่ไม่มีคู่ตรงจากตารางซ้ายจะถูกรวมเข้ามาและเติมค่า NULL เพิ่มเติมโดยรูปที่ 2.4

SQL JOINS



รูปที่ 2.4 SQL JOINS

(<https://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins>)

2.5.4 การใช้คำสั่ง SUBQUERY

คำสั่ง SUBQUERY (หรือ INNER QUERY) คือคำสั่ง SELECT ซึ่งอยู่ภายในคำสั่ง SELECT, INSERT, UPDATE, หรือ DELETE อื่น ๆ ใช้ในการดึงข้อมูลหรือคำนวณผลลัพธ์ย่อย สามารถใช้ SUBQUERY เพื่อดึงข้อมูลจากรายย่อย หรือทำการคำนวณย่อยเพื่อใช้ในคำสั่งหลัก ตัวอย่าง

```
SELECT product_name, unit_price
FROM products WHERE unit_price > (SELECT AVG(unit_price) FROM products);
```

ตัวอย่างนี้ SUBQUERY ถูกใช้ใน WHERE clause เพื่อเปรียบเทียบราคาของสินค้ากับค่าเฉลี่ยของราคาสินค้าในตาราง Products เนื้อหาส่วนนี้เป็นเพียงเรื่องเริ่มต้นของ SQL ยังมีหลายเรื่องที่ยังสามารถเรียนรู้ ทั้งนี้การเขียน SQL ได้อย่างชำนาญและถูกต้องนั้นจำเป็นต้องใช้ประสบการณ์ในการเขียนหลายๆครั้ง และสิ่งที่สำคัญที่สุดของการใช้ SQL ได้อย่างถูกต้องคือการที่รู้รายละเอียดของฐานข้อมูลที่มีอยู่ เพราะในความเป็นจริงมนุษย์ไม่สามารถจดจำได้ทุกวิธีการเขียนได้ ดังนั้นสิ่งที่สำคัญคือการปรับตัวให้เข้ากับระบบฐานข้อมูลแต่ละองค์กร ซึ่งวิธีการเก็บข้อมูลจะต่างกันออกไป เมื่อสามารถทำความเข้าใจกับฐานข้อมูลที่ต้องใช้ ก็สามารถนำข้อมูลเหล่านั้นมาวิเคราะห์ สร้างโจทย์ต่างๆ และพัฒนาเพื่อองค์กรและระบบสืบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 งานวิจัยที่เกี่ยวข้อง

2.6.1 การลดความแออัด ณ แผนกอุบัติเหตุและฉุกเฉิน ในโรงพยาบาลชุมชนแห่งหนึ่ง จังหวัดศรีสะเกษ

โรงพยาบาลราชสีลา จังหวัดศรีสะเกษ วิทยาลัยพยาบาลบรมราชชนนี สุรินทร์รายงานวิจัยเกี่ยวกับการลดความแออัดในแผนกอุบัติเหตุและฉุกเฉิน ในโรงพยาบาลชุมชน ได้สรุปผลการวิจัยเป็นที่เรียบร้อยโดยมุ่งเน้นการพัฒนาและประเมินผลของแนวทางการลดความแออัดที่มีประสิทธิภาพในการดูแลผู้ป่วยในสถานการณ์ฉุกเฉินการวิจัยนี้ได้คัดเลือกกลุ่มตัวอย่างเป็นผู้ปฏิบัติงานในแผนกอุบัติเหตุและฉุกเฉินจำนวน 25 คน และผู้ใช้บริการจำนวน 5,909 คน ที่แบ่งเป็นกลุ่มที่ใช้แนวทางการดูแลแบบปกติจำนวน 3,111 คน และกลุ่มที่ใช้แนวทางที่พัฒนาขึ้นจำนวน 2,798 คน ในการดำเนินการวิจัยระหว่างเดือน เมษายน - กันยายน 2561 ในช่วงเวลา 16.00-24.00 น. ในโรงพยาบาลชุมชนแห่งหนึ่ง

ผลการวิจัยแสดงให้เห็นถึงแนวทางการลดความแออัดในแผนกอุบัติเหตุและฉุกเฉินซึ่งประกอบด้วยหลายมาตรการที่มีผลดีต่อการดูแลผู้ป่วย อาทิเช่นการจัดตั้งคณะกรรมการพัฒนาแผนกอุบัติเหตุและฉุกเฉินคุณภาพ การคัดกรองผู้ป่วยการกำหนดระยะเวลาสำหรับผู้เจ็บป่วยวิกฤตฉุกเฉินในแผนกไม่เกิน 4 ชั่วโมง รวมถึงการจัดระบบการเงินและการชำระเงินในการรักษาและการให้ข้อมูลความรู้ในการดูแลตนเองหลังการจำหน่าย นอกจากนี้ยังมีการประยุกต์ใช้ระบบ Fast track และระบบการตรวจวิเคราะห์ทางห้องปฏิบัติการ ณ จุดดูแลผู้ป่วย รวมทั้งการนำเสนอการใช้รูปแบบ patient journey model เพื่อลดระยะเวลารอคอยให้น้อยลง

จากผลการวิจัยพบว่าระยะเวลาเฉลี่ยที่ใช้ในการรอคอยลดลงอย่างมีนัยสำคัญทางสถิติ และมีการลดลงของค่าคะแนน NEDOCS Score ให้อยู่ในช่วง 101-140 คะแนน การปฏิบัติการรักษาโดยไม่รอตตรวจยอมลดลงอย่างมีนัยสำคัญทางสถิติ จากร้อยละ 3.05 เหลือร้อยละ 0.89 ผลการวิจัยชัดเจนว่าแนวทางการลดความแออัดที่นำเสนอมีประสิทธิภาพ ทำให้ระยะเวลารอคอยสำคัญลดลงอย่างมีนัยสำคัญทางสถิติต่ำกว่า 4 ชั่วโมง ดังนั้น คำสำคัญที่ได้รับการเน้นในการวิจัยประกอบด้วย การลดความแออัด, ระยะเวลารอคอย, แผนกอุบัติเหตุและฉุกเฉิน และการปรับปรุงกระบวนการดูแลผู้ป่วยในทางที่ดีและมีประสิทธิภาพเพื่อตอบสนองความต้องการของผู้รับบริการอย่างเหมาะสมและรวดเร็ว.

2.6.2 การจำลองสถานการณ์เพื่อลดเวลารอคอยของผู้ป่วยแผนกผู้ป่วยนอกโรงพยาบาลชุมชน จังหวัดสุพรรณบุรี

โรงพยาบาลชุมชนแห่งหนึ่งในจังหวัดสุพรรณบุรี มีผู้ป่วยเข้ารับบริการแผนกผู้ป่วยนอกเป็นจำนวนมากในแต่ละวัน ทำให้เวลารอคอยเฉลี่ยคิดเป็น 77.92% ของเวลาทั้งหมด การศึกษานี้มีวัตถุประสงค์ในการประยุกต์ใช้โปรแกรมคอมพิวเตอร์ในการจำลองสถานการณ์กระบวนการทำงานของแผนกผู้ป่วยนอกเพื่อลดเวลารอคอยเฉลี่ยของผู้ป่วย วิธีดำเนินงานวิจัย: เก็บรวบรวมข้อมูลของผู้ป่วย 2 ประเภท คือ ผู้ป่วยที่มีนัด และผู้ป่วยที่ไม่มีนัดของแผนกผู้ป่วยนอก ประกอบด้วย ระยะเวลา รอคอยและระยะเวลาในการให้บริการของแต่ละกระบวนการ ตลอดจนจำนวนทรัพยากรของแผนกผู้ป่วยนอก และจำลองสถานการณ์ของระบบการให้บริการผู้ป่วยนอกด้วยโปรแกรม Arena เวอร์ชัน 14.0 จากนั้นกำหนดแนวทางในการปรับปรุงทั้งหมด 3 แนวทาง ได้แก่

- 1) การจัดการสรรกรเข้ามาของผู้ป่วยที่มีนัด
- 2) การปรับตารางการทำงานของผู้ให้บริการ
- 3) ปรับทั้งการจัดการสรรกรเข้ามาของผู้ป่วยและตารางการทำงาน

ผลการศึกษา: ผลจากการจำลองสถานการณ์ปัจจุบัน พบว่า ระยะเวลาารอคอยเฉลี่ยของผู้ป่วยอายุรกรรมที่มีตรวจ Lab เฉลี่ยที่ 187.20 ± 7.31 นาที และผลจากการจำลองสถานการณ์ของแนวทางการปรับปรุงทั้ง 3 แนวทาง พบว่า ระยะเวลาารอคอยเฉลี่ยของผู้ป่วยอายุรกรรมที่มีตรวจ Lab เฉลี่ยที่ 141.70 ± 4.33 นาที 132.79 ± 5.52 นาที และ 98.85 ± 3.11 นาที ตามลำดับ โดยแนวทางการปรับปรุงที่ 3 สามารถลดระยะเวลาารอคอยเฉลี่ยของผู้ป่วยอายุรกรรมที่มีตรวจ Lab คิดเป็น 47.20% ของระยะเวลาทั้งหมด สรุปผล: แนวทางการปรับปรุงที่ 3 เป็นแนวทางที่ดีจากการจัดการสรรกรเข้ามาของผู้ป่วยที่มีนัด แบบนัดให้มาครึ่งวันเช้า ช่วง 7.00 น. – 12.00 น. โดยเฉลี่ยให้เท่าๆ กันในแต่ละช่วงชั่วโมง และการปรับตารางการทำงานของเจ้าหน้าที่ ได้แก่ จุดวัดสัญญาณชีพ แพทย์ การเงิน และการจ่ายยา ทำให้ลดระยะเวลาารอคอยเฉลี่ยของผู้ป่วยนอก (OPD) อายุรกรรมทั้งที่มีนัดและไม่มีนัดได้อย่างมีนัยสำคัญ

บทที่ 3

วิธีการดำเนินงานวิจัย

การวิเคราะห์ความสำคัญของตัวชี้วัดในการตรวจรักษาของแพทย์เป็นสิ่งที่สำคัญเนื่องจากมีผลต่อคุณภาพการดูแลสุขภาพของผู้ป่วยแต่ละคนในอดีต ลูกค้ำหรือผู้ป่วยในโรงพยาบาลเคยเป็นกลุ่มราชการที่เฉพาะเจาะจงไปในเหล่าทหาร ซึ่งการความล่าช้าในการตรวจรักษาผู้ป่วยอาจส่งผลทำให้จำนวนผู้ป่วยที่มาใช้บริการมีน้อยกว่าที่ควรและปัจจุบันที่มีผู้ประกอบการอาชีพอื่นๆ มาให้บริการนอกจากเหล่าทหารด้วย นอกจากนี้ การเติบโตของเทคโนโลยีทางการแพทย์ที่มีการพัฒนาอย่างรวดเร็วส่งผลให้กระบวนการตรวจรักษาของแพทย์ต้องเป็นไปอย่างรวดเร็วและมีประสิทธิภาพมากขึ้น

ในอดีตมีการตั้งกฎและหลักการเพื่อประเมินระยะเวลาในการตรวจรักษาผู้ป่วยนอกหรือผู้ป่วยที่มีการตรวจรักษาที่โอพีดี แต่กฎและหลักการเหล่านั้นไม่ได้รับการพัฒนาและปรับปรุงอย่างต่อเนื่อง ทำให้ความแม่นยำของกระบวนการนี้ลดลงและมีคุณภาพที่ไม่แน่นอน ซึ่งส่งผลให้ไม่มีบุคลากรในโรงพยาบาลนำไปใช้ต่อ ดังนั้นการวิเคราะห์ความสำคัญของตัวชี้วัดระยะเวลาบริการผู้ป่วยนอกนั้นมีความสำคัญอย่างยิ่งในการพัฒนาระบบการดูแลสุขภาพและการรักษาในโรงพยาบาลพระมงกุฎเกล้าในปัจจุบัน

สำหรับสมมติฐานของตัวชี้วัดที่ได้กล่าวถึงนั้น เป็นการกำหนดตัวชี้วัดสำหรับการประเมินประสิทธิภาพในการให้บริการทางการแพทย์ซึ่งนำมาใช้ในการวัดระยะเวลาที่ใช้ในการนัดหมายของแพทย์ รวมถึงเวลาที่ใช้ในกระบวนการวินิจฉัยโรคของแต่ละผู้ป่วย โดยมีการเพิ่มเวลา 30 นาทีเพื่อการบันทึกข้อมูลเพิ่มเติมที่เกี่ยวข้อง

การใช้สมมติฐานนี้เพื่อกำหนดตัวชี้วัด เกิดขึ้นเนื่องจากความจำเป็นในการประเมินคุณภาพของโรงพยาบาลรัฐในปี 2023 นั้น ส่งผลให้ต้องมีตัวชี้วัดที่เป็นมาตรฐานที่แพทย์ทุกคนและบุคลากรทางการแพทย์ทุกคนในสถานพยาบาลยอมรับและใช้งานอย่างสม่ำเสมอ โดยสมมติฐานที่กล่าวถึงนี้จึงส่งผลให้เกิดการนำเสนอสูตรตัวชี้วัดระยะเวลาตรวจที่เหมาะสมและมีความน่าเชื่อถือสำหรับการประเมินคุณภาพของการให้บริการทางการแพทย์ในโรงพยาบาลพระมงกุฎเกล้าในปี 2023

3.1 สิ่งพื้นฐานข้อมูลมี และ Data Warehouse ใน HIS

ผู้พัฒนาระบบสารสนเทศของทางโรงพยาบาลพระมงกุฎเกล้านี้ซึ่งถือว่าเป็นโรงพยาบาลแรกๆ ในประเทศที่มีระบบสารสนเทศโรงพยาบาลที่เป็น Data Warehouse และมีความเสถียร,เร็วในการประมวลผล และความปลอดภัยของข้อมูล ซึ่งระบบฐานข้อมูลที่โรงพยาบาลพระมงกุฎเกล้าใช้จะเป็นตัว Oracle server ซึ่งต้องใช้ SQL, และ PLSQL ในการสื่อสารกับตัว Warehouse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ดังนั้นสิ่งที่จำเป็นต้องมีในงานวิจัยนี้หลักๆ คือ ฐานข้อมูลและข้อมูลต่างๆที่เก็บย้อนหลังไว้ตั้งแต่ปี 2000
- Tool หลักสำหรับการเขียน SQL หลักๆที่ใช้โรงพยาบาลพระมงกุฎคือตัว Golden32 อาจจะเก่าหน่อยแต่เร็วและเสถียร
- Apex Oracle เป็นเครื่องมือของ Oracle ที่พัฒนาแบบรวมบนเว็บที่เป็นกรรมสิทธิ์ (IDE) ที่เป็นกรรมสิทธิ์ซึ่งพัฒนาโดย Oracle Corporation สำหรับการสร้างและจัดการฐานข้อมูล Oracle และฟังก์ชันอื่นๆ ของฐานข้อมูลที่เกี่ยวข้อง APEX ตั้งใจที่จะลดความซับซ้อนของกระบวนการสร้างเว็บแอปพลิเคชันที่ทำหน้าที่เป็นส่วนหน้าของเซิร์ฟเวอร์ฐานข้อมูลตัวอย่างหน้าเว็บรูปที่ 3.1



รูปที่ 3.1 หน้าแรกของเว็บ APEX ORACLE

3.2 การเตรียมข้อมูล และการกำหนด คำนียามให้กับข้อมูล

สำหรับโปรแกรมการพัฒนาระบบงานประกันคุณภาพด้านมาตรฐานโรงพยาบาลและบริหารสุขภาพ จะมีหน้าเมนูทั้งหมด 5 เมนู แต่สำหรับงานวิจัยนี้จะขอยกมานำเสนอ 1 หน้าต่างเมนูซึ่งคือหน้าระยะเวลาบริการผู้ป่วยนอก

หลังจากได้ทราบโจทย์จึงได้ทำ SQL Query เพื่อดึงข้อมูลออกมาตามที่ต้องการ โดยจะมีตาราง(Table)ที่จำเป็นในการสร้างหลักๆทั้งหมด 3 ตาราง และความซับซ้อนของ Query ขึ้นอยู่กับเงื่อนไขต่างๆของโจทย์ ซึ่งเนื้อหาหลักในบทที่ 3 จะเน้นในเรื่องทฤษฎีการเขียน SQL การใส่พารามิเตอร์ต่างๆในการสร้างข้อมูลรูปแบบใหม่ที่ไม่ได้มีเก็บไว้แต่แรกเดิม และทำไมจำเป็นต้อง SQL แต่ไม่ใช่ NoSQL ในการสร้างโปรแกรม IPD Center เพราะระบบฐานข้อมูลในโรงพยาบาลพระมงกุฎใช้ Oracle และเป็นฐานข้อมูลที่ใช้เพียงแค่ผ่านในองค์กร ในงานวิจัยนี้หัวใจสำคัญที่ได้มาของข้อมูลคือ ความรู้และเงื่อนไขวิธีการกรองข้อมูล เพื่อที่จะ Query ข้อมูลออกมาด้วย SQL

ตารางที่ 3.1 นียามตัวชี้วัด

<p>นิยามตัวชี้วัด: ร้อยละของผู้รับบริการที่ได้รับการตรวจตามเวลาที่กำหนด</p> <p>ถ้า OPD นัด 09:00-10:00 น.แพทย์ลง ICD-10 ภายใน 10:30 น.ถือว่าได้รับการตรวจตามเวลาที่กำหนด</p> <p>ถ้า OPD นัด 10:00-11:00 น.แพทย์ลง ICD-10 ภายใน 11:30 น.ถือว่าได้รับการตรวจตามเวลาที่กำหนด</p> <p>ถ้า OPD นัด 11:00-12:00 น.แพทย์ลง ICD-10 ภายใน 12:30 น.ถือว่าได้รับการตรวจตามเวลาที่กำหนด</p> <p>เฉพาะกองจักษุ ใช้เวลาคัดกรองแทนเวลาการลง ICD-10</p>

จากนิยามตัวชี้วัดที่ทางแพทย์และผู้บริหารต้องการให้เพิ่มเวลาอีก 30 นาที เนื่องจากการนัดคนไข้และการตรวจคนไข้ของแพทย์แต่ละท่านในโรงพยาบาลพระมงกุฎเกล้าไม่มีเวลาที่แน่นอน รวมถึงจำนวนคนไข้ที่นัดมา ในบางช่วงเวลา เช่น ช่วง 09:00-10:00 อาจมีมากกว่า 1 คน ดังนั้น คณะแพทย์และผู้บริหารได้ตกลงให้เพิ่มเวลา 30 นาที เพื่อไม่ให้ค่า KPI ที่ได้แสดงผลลัพธ์ที่น้อยลง ความสำคัญของการปรับเวลานี้จะถูกสำรวจอีกครั้งในส่วนของสรุปผลเพื่อให้เห็นถึงความสำคัญของการวัดตัวชี้วัดประสิทธิภาพและคุณภาพในการให้บริการทางการแพทย์ในอนาคตของโรงพยาบาลพระมงกุฎเกล้า ก่อนเริ่มการเขียน Query ต้องเข้าใจโจทย์และเป้าหมายของข้อมูลที่ต้องการนำข้อมูลออกมาใช้ก่อน

จุดดีและข้อดีของการงานวิจัยนี้คือการที่สามารถสร้างฐานข้อมูลและสามารถนำข้อมูลออกมาใช้ได้โดยอิสระซึ่งการนำข้อมูลออกมาใช้ในที่นี้หมายถึงการนำข้อมูลมาพัฒนาภายในองค์กร ซึ่งฐานข้อมูลของทางโรงพยาบาลมีข้อมูลเก็บย้อนหลังได้ 20 ปี ซึ่งเป็นข้อมูลขนาดใหญ่แต่มีข้อเสียในบางส่วน คือ การมีเครื่องมือที่จำกัดและการนำข้อมูลออกมาใช้ข้างนอกจำเป็นต้องทำเรื่องขออนุญาตกับทางฝ่ายปกครองและผู้บริหารโรงพยาบาลซึ่งอาจใช้เวลาหลายเดือนเนื่องจากเป็นระบบราชการ ดังนั้นรูปแบบการวิเคราะห์ในงานวิจัยนี้อาจไม่ซับซ้อนมาก แต่มีประโยชน์ที่เหมาะสมสำหรับทางโรงพยาบาลพระมงกุฎเกล้า อย่างไรก็ตามการใช้ข้อมูลในระบบครั้งนี้เพื่อพัฒนาโปรแกรม IPD Center หน้าระยะเวลาบริการผู้ป่วยนอกได้รับอนุญาตจากทางผู้บริหารและทางหัวหน้าศูนย์คอมพิวเตอร์โรงพยาบาลพระมงกุฎเกล้าเป็นที่เรียบร้อยแล้ว และสามารถนำเนื้อหาบางส่วนมาเขียนในบทวิทยานิพนธ์ฉบับนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 Query ชุดที่ 1 เริ่มต้น

```

select d.hn,
d.opd_opd_no,
d.pla_placecode,to_char(o.date_created,'dd-mm-yyyy') as date_DAIG,
to_char(o.date_created,'hh24:mi:ss') as
Diag_time,d.doctor,trunc(o.date_created,'MM') mon
from DISEASE_WAREHOUSE@pmkd,opddiags@pmk o
where d.opdipd = 'O'
and d.opd_opd_no = o.opd_opd_no
and d.icd_code = o.icd_code
and d.exam_Date between to_date('01042022','ddmmyyyy') and
to_date('30042023','ddmmyyyy')
and d.pla_placecode = '0603' ----OPD for test
and o.type = '1'
group by d.hn,
d.opd_opd_no,trunc(o.date_created,'MM'),
d.pla_placecode,to_char(o.date_created,'dd-mm-
yyyy'),to_char(o.date_created,'hh24:mi:ss'),d.doctor,trunc(o.date_created,'MM')
order by to_date(date_DAIG)

```

รูปที่ 3.2 รูปแบบการเขียน Query ชุดที่ 1

รูปที่ 3.2 สามารถอธิบายถึงรูปแบบการเขียน Query ชุดแรกซึ่งจะเป็นข้อมูลที่อยู่ใน Table DISEASE_WAREHOUSE@pmk และ opddiags@pmk โดยข้อมูลของทั้งสองตารางจะเป็น ข้อมูลประวัติการลงโรคของคนไข้แต่ละคน จะต่างกันว่า DISEASE_WAREHOUSE@pmk จะมี คอลัมน์รายละเอียดคนไข้มากกว่ามากกว่า opddiags@pmk เช่น DISEASE_WAREHOUSE จะมีพวก อายุเพศของคนไข้ เป็นต้น ในโจทย์ข้อนี้จำเป็นต้องใช้ 2 ตารางข้างต้น เพราะ 2 ตารางนี้ มีช่วงเวลา ที่แพทย์ลงโรคของผู้ป่วยหลังจากตรวจ(Date_created) ซึ่งจากโจทย์หลังจากตีความแล้ว จะนำช่วงเวลา ที่แพทย์ลงโรคมาระยะห่างกับเวลาที่ผู้ป่วยนัดมาตรวจ

ตารางที่ 3.2 รายละเอียด field ข้อมูลใน DISEASE_WAREHOUSE@pmk

● Hn	- เลขประจำตัวผู้ป่วย
● opdipd	- ระบุงว่าเป็นการตรวจที่ OPD(คนไข้นอก O) หรือ IPD(คนไข้ใน I)
● opd_opd_no	- เลขการมาตรวจในวันนั้นของคนไข้ (เลข Visit)
● icd_code	- รหัสโรคที่คนไข้เป็นในแต่ละ เลข Visit
● exam_Date	- วันที่มาตรวจ (ใช้กับคนไข้นอกเท่านั้น OPD)
● pla_placecode	- ห้องที่คนไข้ไปตรวจ รหัสสถานที่

อธิบายการเขียน Query ในส่วนของรูปแบบลักษณะการเขียนจะไม่เหมือนการเขียน SQL ทั่วไป แต่จะเป็นรูปแบบเฉพาะที่ใช้ได้กับ Oracle SQL เท่านั้น เป็นความพิเศษของ Oracle เช่น

```
from DISEASE_WAREHOUSE@pmk d,opddiags@pmk o
and d.opd_opd_no = o.opd_opd_no
```

รูปที่ 3.3 ตัวอย่างวิธีการ Left join

รูปที่ 3.3 จะเป็นการอธิบายถึงวิธีการเขียน Left Join ในรูปแบบ Oracle SQL ระหว่าง 2 tables เพื่อที่จะนำ Date_Created ของตาราง opddiags@pmk แสดงในบรรทัด Select

```
and d.exam_Date between to_date('01042022','ddmmyyyy') and
to_date('30042023','ddmmyyyy')
and d.pla_placecode = '0603' ----OPD for test
```

รูปที่ 3.4 ตัวอย่างบรรทัดตัวกรอง Placecode

รูปที่ 3.4 จะเป็นการอธิบายการใส่ค่า Parameters เพื่อกำหนดเงื่อนไขและกรองข้อมูล And D.exam_date between to_date('01042022','ddmmyyyy') and to_date('30042023','ddmmyyyy') เป็นการกำหนดช่วงเวลาของข้อมูลที่ต้องการดึงในที่นี้ จำเป็นต้องใช้คำสั่ง Between ส่วนในบรรทัด and d.pla_placecode = '0603' จะเป็นการระบุสถานที่หรือห้องตรวจที่ต้องการนำมาแสดง ในตัวอย่างที่นำมาจะเป็นห้อง 0603 หรือ ห้อง OPD EYE

ตารางที่ 3.3 สรุปรูปแบบข้อมูลตัวอย่างที่ได้จากการเขียน Query ชุดที่ 1

Selectd.hn	- เลขประจำตัวผู้ป่วย
d.opd_opd_no	- เลขประจำการรักษารั้งนั้น
d.pla_placecode	- ห้องที่ตรวจและลงโรค
to_char(o.date_created,'dd-mm-yyyy') as date_DAIG	- วันที่ตรวจและลงโรค
to_char(o.date_created,'hh24:mi:ss') as Diag_time	- ช่วงเวลาที่ลงรหัสโรค
d.doctor	- รหัสแพทย์ผู้ตรวจ
trunc(o.date_created,'MM') mon	- เดือนที่ตรวจ

#	HN	OPD_OPD_NO	PLA_PLACECODE	DATE_DAIG	DIAG_TIME	DOCTOR	MON
1	[REDACTED]	32989403	0603	01-04-2022	10:27:04	[REDACTED]	01/04/2022 00:00:00
2	[REDACTED]	33274212	0603	20-05-2022	09:42:48	[REDACTED]	01/05/2022 00:00:00
3	[REDACTED]	33388280	0603	10-06-2022	11:15:15	[REDACTED]	01/06/2022 00:00:00
4	[REDACTED]	33629179	0603	22-07-2022	09:50:46	[REDACTED]	01/07/2022 00:00:00
5	[REDACTED]	33692281	0603	03-08-2022	09:23:24	[REDACTED]	01/08/2022 00:00:00
6	[REDACTED]	33845727	0603	26-08-2022	10:15:15	[REDACTED]	01/08/2022 00:00:00
7	[REDACTED]	34187788	0603	21-10-2022	09:16:04	[REDACTED]	01/10/2022 00:00:00
8	[REDACTED]	34282149	0603	04-11-2022	10:57:09	[REDACTED]	01/11/2022 00:00:00
9	[REDACTED]	34539980	0603	09-12-2022	10:33:06	[REDACTED]	01/12/2022 00:00:00
10	[REDACTED]	34810702	0603	19-01-2023	11:26:14	[REDACTED]	01/01/2023 00:00:00
11	[REDACTED]	35068376	0603	28-02-2023	10:55:48	[REDACTED]	01/02/2023 00:00:00
12	[REDACTED]	35119024	0603	09-03-2023	11:39:21	[REDACTED]	01/03/2023 00:00:00
13	[REDACTED]	35337803	0603	20-04-2023	09:54:44	[REDACTED]	01/04/2023 00:00:00

รูปที่ 3.5 รูปแบบข้อมูลที่ได้จาก Query ชุดที่ 1

รูปที่ 3.5 สรุปข้อมูลของ Query ชุดที่ 1 คือการที่นำ foreign key ทั้งสองตารางได้แก่ opd_opd_no มาทำการ Join กันเพื่อที่จะนำข้อมูลบางส่วนในแต่ละตารางมาสร้างเป็นตารางข้อมูลใหม่ที่ไม่มีสร้างไว้แต่แรกเริ่มในระบบ ซึ่งเนื้อหาหลังจากในส่วนของ Query ชุดที่ 2 และ Query ชุดที่ 3 จะมีลักษณะการสร้างข้อมูลรูปแบบเดียวกับ Query ชุดที่ 1

3.2.2 Query ชุดที่ 2

```

select pat_run_hn||'/'||pat_year_hn as hn,
to_char(app_date,'dd-mm-yyyy') as app_date,
to_char(date_created,'dd-mm-yyyy') as date_created,
appoint_name,doc_appoint_name,dd_doc_code

from DATE_DBFS@pmk

where 1=1

and opd_no is not null

and date_created between to_date('01042022','ddmmyyyy') and
to_date('30042023','ddmmyyyy')

and del_flag is null

and pla_placecode = '0603' ---OPD for test

order by to_date(date_created)

```

รูปที่ 3.6 รูปแบบการเขียน Query ชุดที่ 2

จากรูปที่ 3.6 จะบอกในส่วนของ query ชุดที่ 2 เป็นการใช้เพียง 1 ตารางคือ DATE_DBFS@PMK โดยตารางนี้จะเก็บข้อมูลจำพวกช่วงเวลานัดของผู้ป่วย วันที่มาตรวจ วันที่สร้างนัด รวมถึงแพทย์นัด ในแต่ละห้องตรวจ โดยจะมีรูปแบบข้อมูลตั้งต้นจะมีประมาณรูปที่ 3.7 แต่จำนวนคอลัมน์มีมากกว่านี้เยอะจึงนำมาแสดงเป็นตัวอย่างเพียงบางส่วน

#	PAT_RUN_HN	PAT_YEAR_HN	PLA_PLACECODE	DATE_SQ	APP_DATE	APP_TIME	USER_CREATED	DD_DOC_CODE	NOTE	DATE_CREATED
1	██████████	66 0603	17437019	05/04/2023 00:00:00	01/04/2023 00:00:00	opd655	██████████		01/04/2023 10:09:16	
2	██████████	42 0603	17437075	04/04/2023 00:00:00	01/04/2023 00:00:00	opd655	██████████		01/04/2023 10:24:15	
3	██████████	52 0603	17437219	03/04/2023 00:00:00	01/04/2023 00:00:00	opd655	██████████		01/04/2023 11:03:53	
4	██████████	62 0603	17437334	18/04/2023 00:00:00	01/04/2023 00:00:00	opd655	██████████		01/04/2023 11:36:28	
5	██████████	51 0603	17437379	04/04/2023 00:00:00	01/04/2023 00:00:00	opd655	██████████		01/04/2023 11:57:13	
6	██████████	47 0603	17437425	04/04/2023 00:00:00	01/04/2023 00:00:00	opd655	██████████		01/04/2023 12:25:31	
7	██████████	53 0603	17437579	04/04/2023 00:00:00	01/04/2023 00:00:00	58032	██████████		02/04/2023 10:52:04	
8	██████████	45 0603	17437622	03/04/2023 00:00:00	01/04/2023 00:00:00	58032	██████████		02/04/2023 22:06:16	
9	██████████	45 0603	17437623	05/04/2023 00:00:00	01/04/2023 00:00:00	58032	██████████		02/04/2023 22:21:51	
10	██████████	60 0603	17437755	24/04/2023 00:00:00	01/12/2022 00:00:00	w3549	██████████		03/04/2023 08:36:56	

รูปที่ 3.7 รูปแบบข้อมูลที่ได้จากตาราง Date_DBFS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 จะมี 3 คอลัมน์ ที่เป็นเกี่ยวกับช่วงเวลาโดยจะใช้ App_Date ซึ่งคือ คอลัมน์ที่คนไข้ได้นัดมาตรวจครั้งต่อไปโดยจะมี Date_Created เป็นตัวบอกวันที่สร้าง App_Date ในอีกความหมายคอลัมน์ Date_created คือเหตุที่เกิดก่อน App_date และ Date_created คือวันที่คนไข้มาตรวจในครั้งนั้นๆ เมื่อมีการตรวจรักษาแล้วเสร็จแพทย์จึงได้สร้าง App_date เพื่อนัดหมายการมาตรวจในครั้งต่อไป ในส่วนตัว Parameter ที่ได้ใช้ใน Query ชุดนี้จะมีตัวคำสั่ง Is not null จะเป็นการกรองซึ่งคอลัมน์ต่างๆที่อยู่ภายใต้คำสั่ง Is not null จะไม่เป็น คอลัมน์ว่าง แล้วใช้ Pla_placecode เป็นกำหนดรหัสสถานที่ซึ่งข้อมูลสุดท้ายมีดังรูปที่ 3.8

ตารางที่ 3.4 สรุปรูปแบบข้อมูลที่จำเป็นจากตาราง Date_DBFS

pat_run_hn '/' pat_year_hn as hn	- เลขประจำตัวผู้ป่วย
to_char(app_date,'dd-mm-yyyy') as app_date	- วันที่นัดมาตรวจครั้งต่อไป
to_char(date_created,'dd-mm-yyyy')	- วันที่สร้างรายการนัด(App_date)
appoint_name	- ช่วงเวลานัด
doc_appoint_name	- ช่วงเวลานัดตามรหัสที่แพทย์ตั้ง
dd_doc_code	- รหัสแพทย์ผู้นัดตรวจ

#	HN	APP_DATE	DATE_CREATED	APPOINT_NAME	DOC_APPOINT_NAME	DD_DOC_CODE
1	██████████	19-08-2022	01-04-2022	9.00-10.00	9.00-10.00	██████████
2	██████████	22-07-2022	01-04-2022	8.00-9.00	8.00-9.00	██████████
3	██████████	03-04-2023	01-04-2022	8.00-9.00	8.00-9.00	██████████
4	██████████	06-05-2022	01-04-2022	10.00-12.00	10.00-12.00	██████████
5	██████████	18-07-2022	01-04-2022	10.00-12.00	10.00-12.00	██████████
6	██████████	19-04-2022	01-04-2022	10.00-12.00	10.00-12.00	██████████

รูปที่ 3.8 รูปแบบข้อมูลที่ได้จาก Query ชุดที่ 2

3.2.3 Query ชุดที่ 3

หลังจากที่ได้ Query 1 และ Query 2 มาแล้วจะนำ 2 ชุดคำสั่ง Query มาประกอบเข้าด้วยกัน ซึ่งจะมีการ Select Column ที่ไม่มีอยู่ในทั้ง 2 ชุด แต่จะเป็นการสร้างตัวข้อมูลขึ้นมาใหม่ด้วยการเขียนSQLหรือเรียกอีกอย่างว่า New feature ทั้งหมดนี้เพื่อที่จะได้ตัวโมเดลตามสมมติฐานที่ต้องการมาวิเคราะห์ในส่วนต่อไปก่อนจะนำมาสรุปและนำมาสร้างตารางข้อมูลสำหรับการสร้างโปรแกรม IPD Center หนัาระยะบริการผู้ป่วยนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Select A.hn,A.opd_opd_no,A.pla_placecode,A.date_DAIG,A.Diag_time,
      REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') AS max_Diag_time,
      B.doc_appoint_name,
      SUBSTR(B.doc_appoint_name, 1, INSTR(B.doc_appoint_name, '-') - 1) AS
appoint_start_time,
      SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1) AS
appoint_end_time,
      (SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) +
.30AS appoint_end_pus30, -- เวลานั้นถึงบวก 30 นาที
      REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') -
((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30)
ASDiag_di_end30,--max_Diag_timeลบด้วย appoint_end_pus30
      (TRUNC(REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') -
((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30)) * 60
      + ROUND((REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') -
((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30) -
TRUNC(REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') -
((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30))) *
100)) as dde30_to_minutes,--แปลง Diag_di_end30 ให้เป็นนาที
      (case when REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') -
((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30) <=
'0' then 'Good'else 'BAD' end) AS Goodorbad,
A.mon
from (select d.hn,
d.opd_opd_no,
d.pla_placecode,to_char(o.date_created,'dd-mm-yyyy') as date_DAIG,
to_char(o.date_created,'hh24:mi:ss') as Diag_time,trunc(o.date_created,'MM')
mon
from DISEASE_WAREHOUSE@pmkd,opddiags@pmk o
where d.opdipd = 'O'
and d.opd_opd_no = o.opd_opd_no
and d.icd_code = o.icd_code

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

and d.exam_Date between to_date('01032023','ddmmyyy') and
to_date('30042023','ddmmyyy')
and d.pla_placecode<> '8888' ----OPD for test
and o.type = '1'
group by d.hn,
d.opd_opd_no,trunc(o.date_created,'MM'),
d.pla_placecode,to_char(o.date_created,'dd-mm-
yyyy'),to_char(o.date_created,'hh24:mi:ss'),trunc(o.date_created,'MM')) A,
(select pat_run_hn||'/'||pat_year_hn as hn,
to_char(app_date,'dd-mm-yyyy') as app_date,
to_char(date_created,'dd-mm-yyyy') as date_created,
appoint_name,doc_appoint_name,dd_doc_code
from DATE_DBFS@pmk
where 1=1
and opd_no is not null
and date_created between to_date('01032023','ddmmyyy') and
to_date('30042023','ddmmyyy')
and del_flag is null
and doc_appoint_name is not null
and pla_placecode<> '8888' ----OPD for test) B
where 1=1
and A.hn = B.hn(+)
and A.date_daig = B.App_date
group by A.hn,A.opd_opd_no,A.pla_placecode,A.date_DAIG,A.Diag_time,
B.appoint_name,B.doc_appoint_name,B.dd_doc_code,A.mon

```

รูปที่ 3.9 รูปแบบการเขียน Query ชุดที่ 3

ก่อนที่จะอธิบายคำสั่ง Query จากรูปที่ 3.9 การแบ่งแยกข้อมูลเป็นส่วนย่อยเป็นการที่จำเป็นอย่างยิ่งเพื่อเพิ่มความเข้าใจในข้อมูลที่ถูกค้นหา ในที่นี้ สีม่วงถูกใช้เพื่อแทน Query ชุดที่ A ซึ่งเป็นตารางย่อย (Sub table) ที่มีชื่อว่า Disease_warehouse ในระบบฐานข้อมูล ในขณะที่สีเขียวถูกใช้เพื่อแทน Query ชุดที่ B ซึ่งเป็นตารางย่อย (Sub table) อีกชุดที่มีชื่อว่า Date_DBFS ภายในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดียวกันการแบ่งแยกข้อมูลออกเป็นส่วนย่อยทำให้สามารถจัดการและทำความเข้าใจข้อมูลที่ซับซ้อนได้ดีขึ้นและสามารถนำข้อมูลมาใช้งานอย่างมีประสิทธิภาพในการดำเนินงานต่อไป

ต่อมาจะทำการรวม (Join) 2 Sub tables เข้าด้วยกัน โดยจำเป็นต้องรู้ว่าข้อมูลทั้งสอง Query มีค่าที่ตรงกัน คือคอลัมน์ HN ดังนั้นจึงใช้การเชื่อมโยงแบบซ้าย (Left join) โดยหากไม่มีค่า HN บางค่าที่ไม่มีใน A.hn จะนำ B.hn มาแสดงแทน และจะใช้คอลัมน์วันที่ ที่ได้สร้างขึ้นใหม่มาเชื่อมโยง (Join) ด้วย เพื่อให้ช่วงเวลาที่กำหนดตรงกัน ได้แก่ A.date_daig = B.App_date

การใช้เงื่อนไข 1=1 ใน SQL เป็นเทคนิคที่สามารถใช้ในการสร้าง dynamic SQL โดยไม่ต้องกังวลเกี่ยวกับการเพิ่ม WHERE หรือ AND ต่อจาก SELECT xxx FROM yyyy ซึ่งช่วยให้สามารถสร้างเงื่อนไขเพิ่มเติมได้อย่างสะดวกสบาย ดังนั้นเมื่อต้องการเขียน Query ที่มีเงื่อนไขมากกว่า 1 พารามิเตอร์ขึ้นไป จะใช้คำว่า and เพื่อเชื่อมต่อเงื่อนไขเหล่านั้นไปเรื่อยๆ และไม่จำเป็นต้องใช้เงื่อนไข 1=1 นี้ เว้นแต่จะใช้ในการสร้าง Dynamic SQL หรือ หากว่าต้องการให้เงื่อนไขเป็นจริงตลอดเวลาโดยไม่ได้ระบุเงื่อนไขที่แน่นอนไว้ล่วงหน้าใน Query ที่กำหนด Query ชุดที่ 3 ได้สรุปข้อมูลโดยการเลือกคอลัมน์ออกมาเพื่อให้ข้อมูลถูกต้องตามโมเดลที่ได้รับโจทย์ พร้อมกับการสร้างคอลัมน์ใหม่ขึ้นจากที่มีอยู่ในฐานข้อมูลโดยจะใช้คำสั่ง SQL ที่ไม่ได้ใช้บ่อย ในส่วนต่อไปจะอธิบาย ชุดคำสั่งแรกเริ่มต้นจาก REPLACE(max(SUBSTR(A.Diag_time,0,5)),':','.')ASmax_Diag_time โดย REPLACE จะทำหน้าที่ในการแทนที่ข้อความใหม่จากข้อความเดิมตามที่ต้องการที่จะเปลี่ยน สำหรับในชุดคำสั่งดังกล่าวคือ แกะไขจากคอลัมน์ A.Diag_time ที่มีค่าเป็น Varchar เช่น 09:39:07 เป็น 09.39 วิธีการนี้จะใช้คำสั่ง SUBSTR มาช่วยในการปรับแต่ง - SUBSTR ทำหน้าที่ในการตัดข้อความตามตำแหน่งที่ต้องการได้ โดยมีรูปแบบใน Oracle คือ SUBSTR(Column XXX,0 (ลำดับตัวอักษรที่ต้องการเริ่มให้อ่าน),5 (ลำดับตัวอักษรที่ต้องการให้หยุดที่ตำแหน่งนั้น)) ในที่นี้จะเลือก 0 ถึง 5 จาก 09:39:07 เพื่อเปลี่ยนเป็น 09:39 จากนั้นร่วมกับคำสั่ง REPLACE เพื่อทำการแทนที่ ':' ด้วย '.'

เทคนิคคืออย่างที่ใช้ใน Query นี้คือ SUBSTR ที่ใช้ร่วมกับ INSTR ตัวอย่างเช่น การปรับรูปแบบข้อมูลของ Column doc_appoint_name จากรูปแบบเดิม ' 9.00-10.00 ' เป็น 9.00 และ 10.00 โดยใช้คำสั่ง SUBSTR(B.doc_appoint_name, 1, INSTR(B.doc_appoint_name, '-') - 1) AS appoint_start_time โดยในกรณีนี้คอลัมน์ doc_appoint_name หมายถึงช่วงเวลาแพทย์นัดผู้ป่วยมาตรวจ คำสั่ง INSTR ใช้ในการค้นหาตำแหน่งของข้อความตั้งแต่ตำแหน่งแรกที่พบ

นอกจากนี้SQLยังสามารถนำมาใช้ในการคำนวณที่ซับซ้อนได้อีกด้วยเช่นคอลัมน์ dde30_to_minuteซึ่งทำการลบช่วงเวลาที่แพทย์ใช้ในการวินิจฉัยโรค(max_Diag_time) ด้วยช่วงเวลาที่แพทย์นัดมาบวก 30 นาที (appoint_end_pus30) ตัวอย่างเช่น dde30_to_minute

HN	OPD_OPD_NO	PLA_PLACECODE	DATE_DAIG	DIAG_TIME	MAX_DIAG_TIME	DOC_APPOINT_NAME	APPOINT_END_TIME	APPOINT_END_PUS30	DIAG_DI_END30
		0119	04-04-2023	12:08:53	12.08	10.01-11.00	11.00	11.3	.78

รูปที่ 3.10 แสดงข้อมูล max_Diag_time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 อธิบายลักษณะความเข้าใจถ้าเป็นมนุษย์จะสามารถเข้าใจว่าให้หน้า 12.08 – 11.30 จะได้ผลเท่ากับ .78 หรือ 78 นาที หรือ 1 ชั่วโมง กับ 18 นาที แต่สำหรับตัว SQL โปรแกรมจะไม่สามารถเข้าใจได้เองว่า 12.08 – 11.30 ให้ผลลัพธ์ออกมาเป็นค่า 78 หรือ 78 นาที หรือกรณีตัวอย่างเพิ่มเติมจากรูปที่ 3.11

HN	OPD_OPD_NO	PLA_PLACECODE	DATE_DAIG	DIAG_TIME	MAX_DIAG_TIME	DOC_APPOINT_NAME	APPOINT_END_TIME	APPOINT_END_PUS30	DIAG_DI_END30
██████	██████	0119	04-04-2023	12:08:53	12.08	10.01-11.00	11.00		11.3
██████	██████	0501	11-04-2023	13:05:49	13.05	10.00-11.00	11.00		11.3 → 1.75
██████	██████	0101	31-03-2023	10:27:15	10.27	9.00-10.00	10.00		10.3
██████	██████	0108	31-03-2023	09:29:14	09.29	9.00-10.00	10.00		10.3 → -1.01

รูปที่ 3.11 ตัวอย่างการแปลงค่าเวลาให้เป็นนาที

รูปที่ 3.11 การที่โปรแกรมมองค่า 13.05 – 11.3 = 1.75 และ 09.29 – 10.3 = ติดลบ -1.01 ถือว่าถูกต้องตามหลักคณิตศาสตร์ แต่ในที่นี้ จะต้องการให้โปรแกรมดูค่า 1.75 เป็นรูปแบบนาฬิกาของช่วงเวลา ซึ่งค่า 1.75 ก็คือ 1.00 = 60 นาที บวกกับ 75 นาทีเท่ากับ 135 นาที หรือในตัวอย่างจากรูปกรณี 09.29 – 10.3 = -1.01 โดยต้องการแปลงให้เป็น - 61 นาที กระบวนการในการเขียนจะมีความซับซ้อนและคำสั่งคำนวณต่างๆดังรูปที่ 3.12

```
( TRUNC(REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.')) - ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30)) * 60 + ROUND((REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.')) - ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30) - TRUNC(REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.')) - ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30)) * 100))
```

รูปที่ 3.12 ตัวอย่างชุดคำสั่งต่างๆในการแปลงค่า

จากชุดคำสั่งรูปที่ 3.12 จะมี 3 คำสั่งที่ใช้หลักๆได้แก่ TRUNC, REPLACE และ SUBSTR ส่วนที่ ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30)) * 60 ใน SQL นั้นคือการเปลี่ยนหน่วยเวลาจากชั่วโมงและนาทีเป็นหน่วยนาฬิกาเพื่อทำให้การคำนวณเวลาเป็นไปอย่างถูกต้องตามที่ต้องการ หากไม่มีการคูณด้วย 60 หน่วยนาฬิกาจะไม่ถูกแปลงอย่างถูกต้อง โดยเฉพาะเมื่อต้องการนำผลลัพธ์มาบวกหรือลบกับเวลาอื่น ๆ ที่ถูกแสดงในรูปแบบที่มนุษย์เข้าใจ ดังนั้นการคูณด้วย 60 ในที่นี้จึงเป็นขั้นตอนสำคัญเพื่อแปลงหน่วยเวลาให้เป็นรูปแบบที่ต้องการ โดยในส่วนถัดไปจะเป็นชุดคำสั่งเพื่อแปลงหน่วยเวลาและใช้งานได้ในกรณีประมวลผลข้อมูลใน SQL โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งต่อไปนี้ได้แก่ TRUNC(REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') - ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30))) * 100) คือการแปลงหน่วยเวลาที่เป็นทศนิยมหรือทศนิยมแบบมีชั่วโมงและนาทีไปเป็นหน่วยที่ถูกแสดงในรูปแบบของจำนวนเต็ม เพื่อให้การประมวลผลเวลาเป็นไปอย่างถูกต้องตามที่ต้องการ การคูณด้วย 100 ในที่นี้จึงเป็นขั้นตอนสำคัญที่ช่วยให้ได้ผลลัพธ์ที่ถูกต้องและเข้าใจได้ง่ายยิ่งขึ้นในการประมวลผลข้อมูล

ต่อไปจะเป็น คำสั่ง 'CASE WHEN' ใช้ใน SQL การใช้ 'CASE WHEN' สามารถช่วยในการแปลงข้อมูลหรือสร้างเงื่อนไขการกรองข้อมูลที่มีความซับซ้อนในการคำนวณหรือใช้ในการจัดรูปแบบข้อมูลให้เหมาะสมกับความต้องการที่กำหนดไว้ ทำให้สามารถดำเนินการประมวลผลข้อมูลได้อย่างมีประสิทธิภาพและเหมาะสมกับความต้องการทางธุรกิจหรือการวิเคราะห์ข้อมูลในแต่ละกรณีได้ ตัวอย่างต่อไปนี้จะเป็นการใช้ชุดคำสั่ง CASE WHEN เพื่อแปลงค่าตามที่ต้องการ

```
(case when REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') - ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30) <= '0' then 'Good' else 'BAD' end) AS Goodorbad
```

รูปที่ 3.13 ตัวอย่างชุดคำสั่งการแปลงค่าด้วย case when

จากรูปที่ 3.13 ได้อธิบายความหมายของชุดคำสั่งโดยจะเริ่มจากชุดคำสั่งแรกได้แก่ REPLACE(max(SUBSTR(A.Diag_time,0,5)), ':', '.') หรือในกรณีนี้คือ MAX_DIAG_TIME จากรูปที่ 3.10 ลบกับ ((SUBSTR(B.doc_appoint_name, INSTR(B.doc_appoint_name, '-') + 1)) + .30) หรือในกรณีนี้คือ APPOINT_END_PUS30 แล้วได้ค่าน้อยกว่าหรือเท่ากับ 0 ให้แสดงผลออกมาเป็น Good ถ้าผลที่ได้้นอกเหนือจากนี้หรือถ้าได้ผลมากกว่า 0 จะให้ผลแสดงออกเป็น BAD

สุดท้ายผลที่ได้จาก Query ชุดที่ 3 จะสรุปและ Select Column โดยอธิบายรายละเอียดส่วนนี้อีกครั้งในบทที่ 4 ซึ่งจะเป็นบทที่อธิบายถึงสรุปผลจากการทำ Query ชุดที่ 3 จากนั้นนำผลที่ได้ไปทำการเลือกคอลัมน์ที่ต้องการเพื่อที่จะนำไปสร้างเป็นตารางข้อมูลใหม่ แล้วนำข้อมูลนำมาพัฒนาโปรแกรม IPD Center ในส่วนของหน้าระยะเวลาบริการผู้ป่วยนอก

3.3 เครื่องมือที่ใช้สำหรับการวิจัย

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมประกอบไปด้วย

1) คอมพิวเตอร์สำหรับพัฒนา โดยมีสเปคขั้นต่ำ คือ:

- Window 8 – 11
- CPU Inter I5 ขึ้นได้
- หน่วยความจำหลัก ใช้ได้ตั้งแต่ 8 GB ขึ้นไป

2) ระบบโปรแกรมประยุกต์

- Apex Oracle server : 191.123.95.34:8080/ords เป็นตัวเซิร์ฟเวอร์ที่ทางโรงพยาบาลสร้าง
- Apex Oracle server : 202.123.95.34:8080/ords เป็นตัวเซิร์ฟเวอร์ที่ทางโรงพยาบาลสร้าง
- โปรแกรมในการใช้ SQL : Golder32, Golder64, SQLdeveloper



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การดำเนินการสร้างและการอภิปรายผล

ในส่วนนี้จะเป็นส่วนขั้นตอนหลังจากได้ตัวข้อมูลที่ต้องการมาสรุปผลและนำไปพัฒนาตัว Web Application ซึ่งก็คือเป้าหมายหลักของงานวิจัยนี้ เพราะปัจจุบันทางโรงพยาบาลยังไม่มีแอปพลิเคชันที่สามารถตรวจสอบสถานะต่างๆในรูปแบบ Online Application ดังนั้นการที่ได้มีการพัฒนาหน้าระยะเวลาบริการผู้ป่วยนอกขึ้นมาจึงเป็นเรื่องให้ความสนใจสะดวกแก่ผู้ใช้งานโรงพยาบาล เช่น แพทย์พยาบาล และผู้บริหาร โดยจากแผนการสร้างแอปพลิเคชัน IPD Center หน้าระยะเวลาบริการผู้ป่วยนอกจะแบ่งขั้นตอนหลักๆในบทนี้เป็น 3 ส่วน คือ

1. Cleaning Data & Summary Result - จะเป็นการใช้ SQL เพื่อตรวจสอบความถูกต้องของข้อมูล การใช้ Parameters ต่างๆในการเขียนเช็คตรวจสอบ
2. การสร้าง Table ที่ได้หลักจากการทำ Summary query - โดยขั้นตอนนี้เป็นเทคนิคการสร้าง Table ใน Oracle โดยข้อมูลนั้นมาจากฐานข้อมูลเดิม แต่จะมีการสร้างรูปแบบข้อมูลใหม่ที่ได้หลังจากขั้นตอนที่ 1 แล้วนำข้อมูลมาใส่ใน Table ที่สร้างเพื่อรับข้อมูล
3. การสร้าง Web Application - นำผลที่ได้มาทุกขั้นตอนก่อนหน้า มาพัฒนาเป็น Web Application

ในส่วนนี้สามารถสรุปขั้นตอนต่างๆได้สั้นและเข้าใจง่าย แต่การจะพัฒนาหน้าระยะเวลาบริการผู้ป่วยนอกจนพร้อมใช้งานนั้นได้พบปัญหาต่างๆที่ทำให้เกิดความล่าช้า ซึ่งอุปสรรคแรกที่ยากนำเสนอรวมถึงเป็นอุปสรรคหลักที่น่าจะพบทุกทีในแต่ละองค์กร คือ ปัญหาของบุคลากร ซึ่งปัญหานี้ทำให้เกิดปัญหาอื่นๆตามมาและทำก่อให้เกิดความล่าช้า เช่น เมื่อผู้บริหารต้องการเปลี่ยนวิธีการคำนวณค่า หรือการปรับรูปแบบโปรแกรม สิ่งเหล่านี้จะมาผลกับการใส่ค่า Parameters ต่างๆ อีกอุปสรรคที่และเป็นความท้าทายในการเรียนรู้ คือการเรียนรู้พร้อมทั้งการทำงานร่วมด้วย เรื่องนี้เป็นประสบการณ์ที่หาได้ยากเพราะเรื่องนี้ทำให้เกิดการเรียนรู้ถึงวิธีการตารางเวลาและพัฒนากิจการจัดการตารางงาน

จากนี้จะเป็นการอธิบายรายละเอียดการสร้างโปรแกรม IPD Center และวิธีการสร้างหน้าระยะเวลาบริการผู้ป่วยนอกหลังจากนั้นมีรายละเอียดการใช้งาน IPD Center ในหน้าต่างเมนูระยะเวลาตรวจผู้ป่วยนอก ที่เป็นหัวใจหลักของงานวิจัยนี้

4.1 Cleaning Data & Summary Result

จากข้อมูลที่ได้สร้างขึ้นมาจากใหม่จาก Query ชุดที่ 3 ในบทที่ 3 จะได้รูปแบบข้อมูลตั้งต้นตามรูปที่ 4.1 ซึ่งยังไม่ได้ผ่านการเลือกคอลัมน์(Column)เพื่อสร้างหน้าระยะเวลาผู้ป่วยน้อย

#	HN	OPD_OPD_NO	PLA_PLACECODE	DATE_DAIG	DIAG_TIME	MAX_DIAG_TIME	DOC_APPOINT_NAME	APPOINT_START_TIME	APPOINT_END_TIME
1			0101	22-03-2023	09:39:07	09:39	9:00-10:00	9:00	10:00
2			0101	07-03-2023	12:23:16	12:23	10:00-11:00	10:00	11:00
3			0101	27-03-2023	09:39:20	09:39	9:00-10:00	9:00	10:00
4			0101	18-04-2023	11:00:02	11:00	9:00-10:00	9:00	10:00
5			0101	21-03-2023	11:18:01	11:18	9:00-10:00	9:00	10:00
6			0101	24-04-2023	09:17:28	09:17	9:00-10:00	9:00	10:00
7			0101	25-04-2023	09:22:51	09:22	9:00-10:00	9:00	10:00
8			0101	26-04-2023	10:29:11	10:29	9:00-10:00	9:00	10:00

APPOINT_END_PUS30	DIAG_DI_END30	DDE30_TO_MINUTES	GOODORBAD	MON
10.3	-91	-91	Good	01/03/2023 00:00:00
11.3	.93	93	BAD	01/03/2023 00:00:00
10.3	-91	-91	Good	01/03/2023 00:00:00
10.3	.7	70	BAD	01/04/2023 00:00:00
10.3	.88	88	BAD	01/03/2023 00:00:00
10.3	-1.13	-73	Good	01/04/2023 00:00:00
10.3	-1.08	-68	Good	01/04/2023 00:00:00
10.3	-01	-1	Good	01/04/2023 00:00:00
10.3	-87	-87	Good	01/04/2023 00:00:00

รูปที่ 4.1 รูปแบบข้อมูลที่จาก Query ชุดที่ 3

จาก Query ชุดท้ายในบทที่ 3 หลังจากที่เราลองกด Run query คอลัมน์ที่สำคัญจะมี DDE30_TO_MINUTES ใช้ในการบอกระยะเวลา ระหว่าง เวลาแพทย์ลงโรค (MAX_DIAG_TIM) ลบกับ เวลานั้นถึงบวก 30 นาที (APPOINT_END_PUS30)

$$\text{สรุป } DDE30_TO_MINUTES = \text{MAX_DIAG_TIM} - \text{APPOINT_END_PUS30}$$

เมื่อได้ค่าใน คอลัมน์ DDE30_TO_MINUTES จะมี คอลัมน์GOODORBAD ที่เป็นตัวแปลงค่าของDDE30_TO_MINUTES โดยถ้าค่าในช่อง ≤ 0 (น้อยกว่าหรือเท่ากับ 0) ให้ผลออกมาเป็น Good เนื่องจากถ้าค่ามากกว่า 0 ก็คือจำนวนนาทีที่เกินมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อมา จะเป็นการใช้ชุด Query ในการ Cleaning Data และ Summary Result เพื่อจะให้ได้ก่อนของรูปแบบข้อมูลที่ดีนำไปใส่ Table ที่ได้จัดดำเนินการเตรียมไว้

4.1.1 Cleaning Data & Summary Result Part 2

หลังจากได้ดูรูปแบบของข้อมูลตั้งต้น จึงได้สรุปและเลือกคอลัมน์ที่จะต้องการใช้จริง เพื่อนำมาสร้างตาราง(opd_wait_time) สำหรับการสร้างหน้าระยะเวลาบริการผู้ป่วยนอกในรูปแบบที่ 4.2

EXAM_DATE	PLA_PLACECODE	OPD_OPD_NO	HN	DEPT_CODE	APPTIME	DX_TIME	DOC_CODE	SPEND_TIME	EVAL
01/11/2018	1606	24114001	1606001	0001	18:00	18:08	0001	9	G
01/11/2018	1606	24114028	1606001	0001	19:00	18:41	0001	-18	G
01/11/2018	1606	24113972	1606001	0001	17:00	16:50	0001	-10	G
01/11/2018	1606	24114016	1606001	0001	17:00	18:01	0001	61	B
01/11/2018	1606	24113885	1606001	0001	18:00	18:21	0001	22	G
01/11/2018	1606	24113999	1606001	0001	18:00	18:24	0001	25	G
01/11/2018	1606	24114013	1606001	0001	19:00	18:19	0001	-41	G
01/11/2018	1606	24113976	1606001	0001	17:00	18:20	0001	80	B
01/11/2018	1606	24114101	1606001	0001	17:00	16:28	0001	-31	G
01/11/2018	1606	24114012	1606001	0001	19:00	16:53	0001	-127	G

รูปที่ 4.2 รูปแบบข้อมูลสรุปคอลัมน์ทั้งหมดที่จำเป็น

ตารางที่ 4.1 สรุปคอลัมน์ทั้งหมดที่จำเป็นต้องใช้ โดยจะมีทั้งหมด 10 คอลัมน์ได้แก่

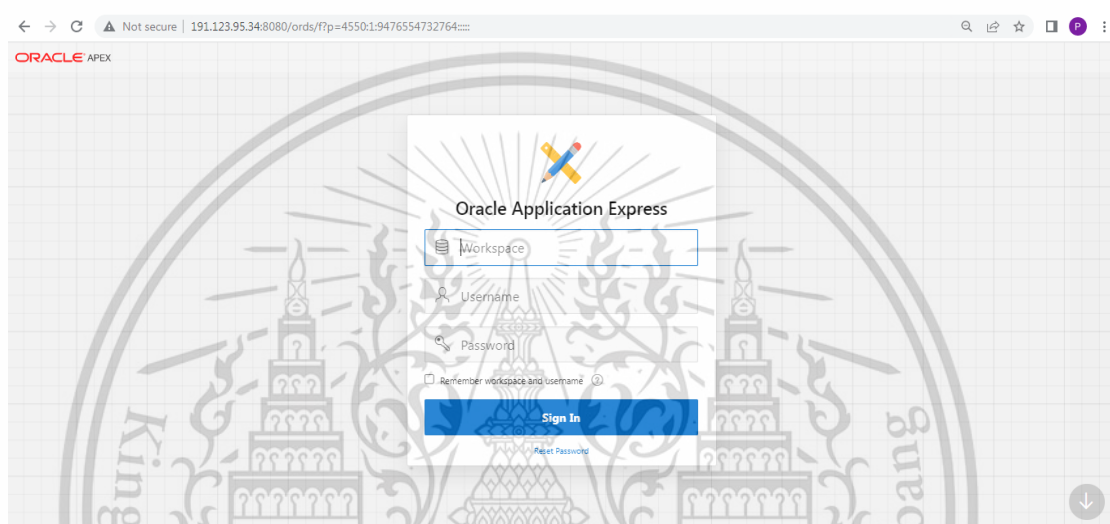
1. EXAM_DATE :	ซึ่งก็คือตัวบอกช่วงเวลา ใช้ในการแบ่งช่วงเดือน ช่วงปี
2. PLA_PLACECODE :	รหัสห้องตรวจ เพื่อแยกตามห้องตรวจ
3. OPD_OPD_NO :	เลข VN หรือเลขประจำการตรวจครั้งนั้นๆ
4. HN :	เลขประจำตัวผู้ป่วย
5. DEPT_CODE :	เลขแยกรหัสสถานที่เป็นกองอีกที เช่น 0630(OPD EYE) จะอยู่ในกอง 0001 (กองตรวจโรค) เป็นต้น
6. APPTIME :	เวลาที่คนไข้นัดครั้งนั้นๆ
7. DX_TIME :	เวลาที่แพทย์ลงโรคในแต่ VN นั้นๆ
8. DOC_CODE :	รหัสแพทย์ที่ตรวจโรค VN นั้นๆ
9. SPEND_TIME :	ระยะเวลาที่ต่างกันระหว่าง APPTIME กับ DX_TIME
10. EVAL :	ผลว่าทำเวลาดีหรือแย่ G คือ ดี, B คือ แย่

ผลของข้อมูลที่ได้มาผ่านวิธีการคัดกรองและ Select Columns รวมถึงมีการตัดข้อมูลที่มีค่าว่างออกเรียบร้อยก่อนนำมาใช้งาน โดยสาเหตุหลักที่ต้องวิเคราะห์ข้อมูลด้วย SQL เพราะว่ามีข้อจำกัดของอุปกรณ์รวมถึงความยากในนำมาข้อมูลออกมาใช้โดยไม่ได้รับอนุญาตจากผู้บริหาร รวมถึงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้งานในโรงพยาบาลต้องการได้ข้อมูลที่เรียบง่ายไม่ซับซ้อนจนเกินไป งานวิจัยนี้ในส่วนโมเดลข้อมูลจึงมีรูปแบบที่ไม่ซับซ้อนมากซึ่งเหมาะกับการนำไปใช้งานกับบุคลากรในระบบโรงพยาบาลพระมงกุฎเกล้า

4.2 การสร้าง Table ที่ได้หลักจากการทำ Summary query

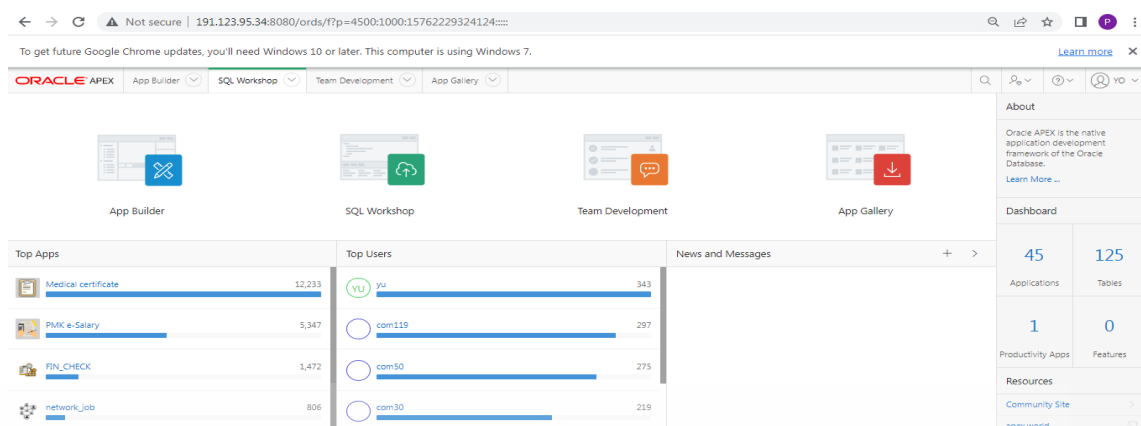
สำหรับขั้นตอนนี้จะเป็นการสร้างฐานข้อมูลขึ้นมาจาก Query สรุปผลจากขั้นตอนก่อนหน้า โดยจะได้เริ่มใช้ Apex Oracle เป็นเครื่องมือในการสร้างทั้งฐานข้อมูลและโปรแกรม



รูปที่ 4.3 หน้าเข้าใช้งาน APEX ORACLE

ใน Apex Oracle ที่เห็นในรูปที่ 4.3 เลข IP จะเป็น 191.123.95.34 ซึ่งอ้างอิงจากตัว Server ที่มีสร้างไว้แต่เดิมแล้วในสร้างโปรแกรมครั้งนี้จึงไม่มีขั้นตอนการสร้าง Server Oracle และเรื่องการสร้างฐานข้อมูลในบางส่วน หลังจากที่ได้ใส่ชื่อ Workspace และ Username จะเป็นการ login เข้าใช้งาน Workspace ที่ได้เลือกสำหรับงานนี้ให้ใส่ชื่อ Workspace ว่า Training ในการเข้าใช้งาน จากนั้นให้ใส่ user และ password ของแต่ละ user

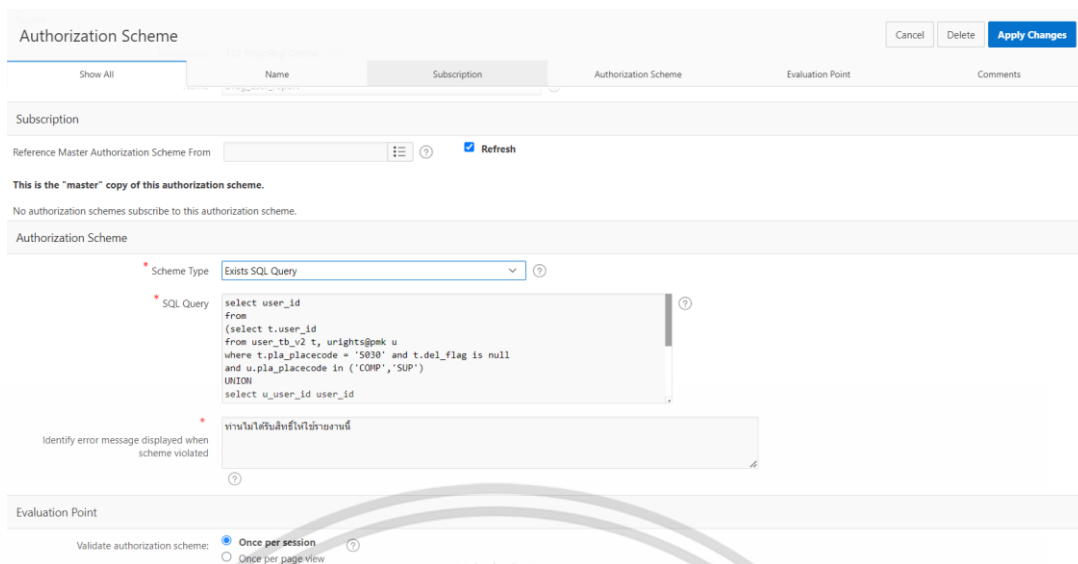
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 หลังจาก LOGIN เข้ามาจะพบหน้าต่างเมนูต่อไปนี้

หลังจาก login เข้ามาจะมีหน้าต่างเมนูต่างๆมีรายการแจ้งสถานะใช้งานโปรแกรมต่างๆใน Workspace นี้เบื้องต้นสามารถกดที่ SQL Workspace แล้วกด Create table เพื่อสร้างตารางข้อมูลที่ต้องการได้จะมี 3 รูปแบบในการสร้างตารางคือ 1.สร้างโดยการพิมพ์ข้อมูลเข้าโดยตรง 2.อัปโหลดด้วยไฟล์ และ 3.สร้างชุดคำสั่ง SQL แต่ในส่วนนี้จะขอข้ามในส่วนวิธีการสร้าง Table ด้วย Query เพราะอาจจะมีผลกระทบในส่วนของความปลอดภัยในระบบจึงไม่สามารถนำมาตีเขียนในวิทยานิพนธ์อย่างไรก็ตามเลข IP Address 191.123.95.34 เป็น IP ที่ทางโรงพยาบาลสร้างไว้เพื่อการใช้งานในระบบภายในเท่านั้น ดังนั้นกรณีถ้าต้องการให้ใช้ออนไลน์จึงต้องใช้ IP address อีกชุดคือ 202.28.80.34:8080/ords ซึ่ง IP 202 จะเป็น IP ที่ไว้สำหรับให้บุคลากรในโรงพยาบาลใช้งานโปรแกรมต่างๆโดยแต่ละโปรแกรมจะมีการสร้าง Authentication Schemes เพื่อกำหนดสิทธิ์ในการเข้าใช้งานแอปต่างๆในเซิร์ฟเวอร์ 202.28.80.34:8080/ords โดยการสร้าง Authentication โดยควรจะสร้างก่อนการสร้างโปรแกรมหรือหลังจากสร้างโปรแกรมแล้วค่อยกลับมาสร้างเพื่อกำหนดสิทธิ์เข้าถึงลักษณะการสร้าง Authentication มีทั้งแบบเขียน SQL ,PLSQL ตัวอย่างรูปแบบการสร้างจากรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

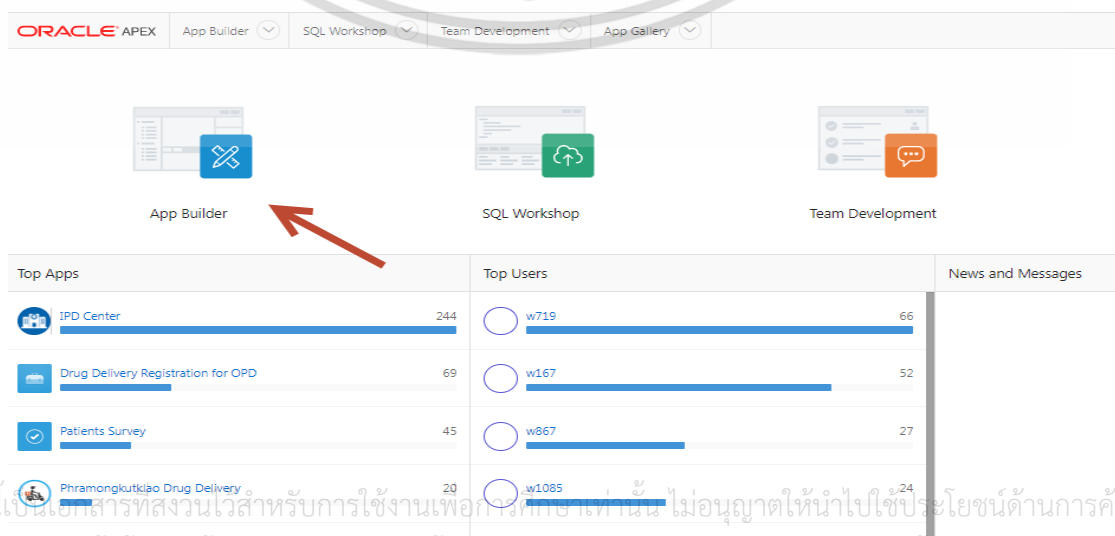


รูปที่ 4.5 ตัวอย่างหน้าสร้าง Authentication

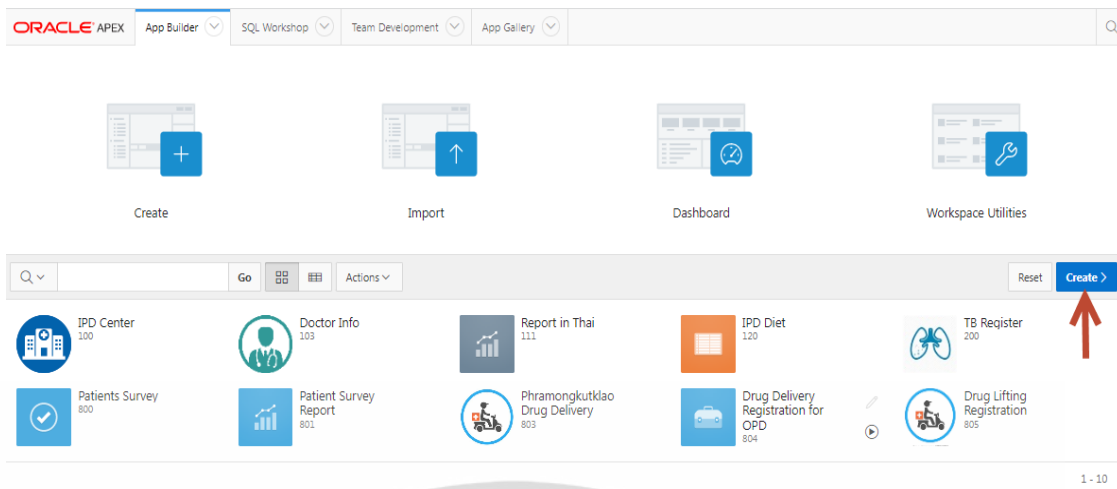
4.3 การสร้าง Web Application

ในสำหรับขั้นตอนนี้จะเป็นการนำตารางที่ได้สร้างในขั้นตอนก่อนหน้ามาสร้างหน้า Web Application โดยชื่อของ Oracle Apex คือเป็นแพลตฟอร์มที่ทำให้การเขียน Application มีความสะดวกในการเข้าใช้งานและสร้าง Application ได้เร็ว สามารถใช้กับ Application ต่างๆ ที่สำคัญคือการใช้งานไม่ซับซ้อนและไม่จำเป็นต้องเสียค่าบริการ ด้วยระบบฐานข้อมูลจาก Oracle ทำให้ลดข้อกังวลเรื่องเทคโนโลยี เพียงแค่ต้องรู้ว่าต้องการทำอะไรในธุรกิจ ก็สามารถสร้างสรรค์ให้เป็น Application สมัยใหม่ที่เข้าได้กับทุกแพลตฟอร์มได้ ในสำหรับขั้นตอนการสร้างจะขออธิบายเป็นลำดับดังนี้

4.3.1 ขั้นที่ 1 หลังจาก Login

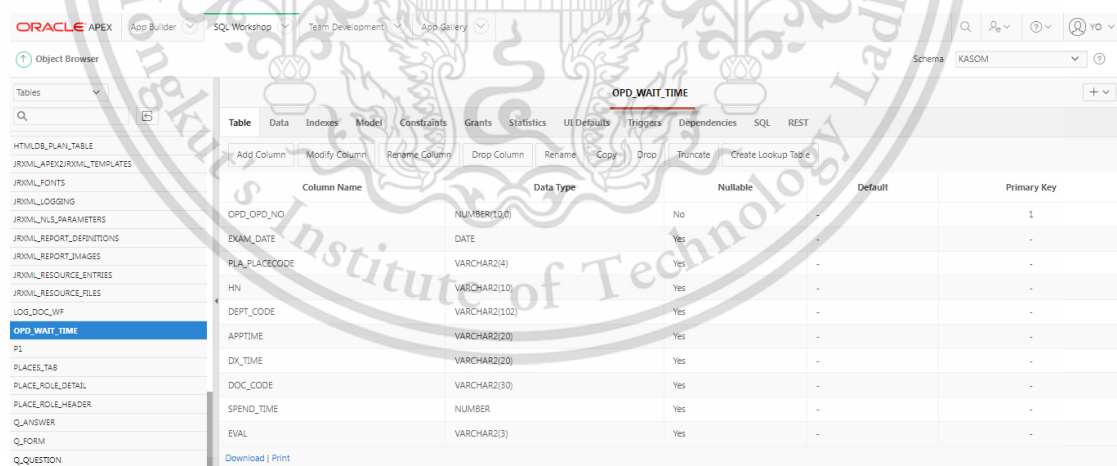


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 เมนูต่างๆในการสร้าง Application

จะเห็นว่าหลังจากกด App builder จะปรากฏหน้าต่างเมนูโปรแกรมและบริเวณด้านบนจะมีปุ่มเมนูให้กด ได้แก่ Create, Import, Dashboard และ โปรแกรมต่างๆที่ถูกสร้างไว้แล้วใน Server จากรูปที่ 4.6 ได้อธิบายเพิ่มเติมสำหรับหน้าต่างเมนู SQL Workshop จะเป็นเมนูเพื่อให้แต่ละ users ไปทดลองเขียน SQL หรือตรวจสอบว่าปัจจุบันใน Server ที่ใช้ในปัจจุบันมีตารางข้อมูลที่เคยได้สร้างไปแล้วก็ตาราง เช่น หลังจากที่ได้ลองกดเข้าเมนู SQL Workshop ตามรูปที่ 4.7 จะพบชื่อตารางที่ได้สร้างในขั้นตอน ‘4.2 การสร้าง Table’



รูปที่ 4.7 SQL Workshop

หลังจากที่กดปุ่ม Create จากรูปที่ 4.6 จะปรากฏหน้าต่างสร้าง Web Application ขึ้นมา ตามรูปที่ 4.8 ในขั้นตอนแรก Apex จะให้ user ตั้งชื่อ Application ที่กำลังสร้างรวมถึงสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

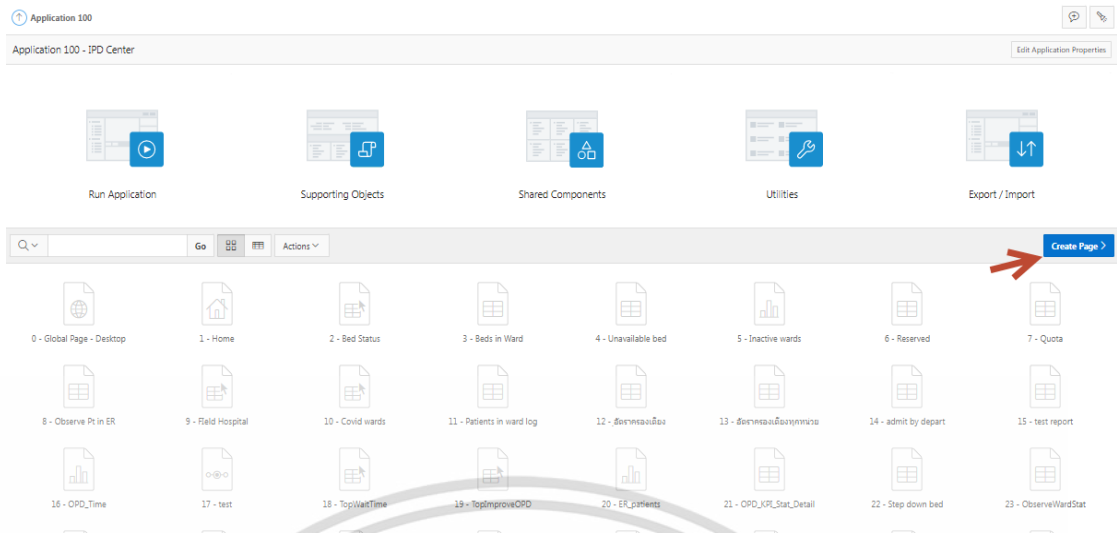
รูปที่ 4.8 กด Add page เพื่อที่จะเพิ่มหน้าใน Application ที่ต้องการสร้าง ซึ่งใน Application สามารถสร้างได้หลายหน้าเพื่อแยกส่วนในการทำงานโปรแกรม

รูปที่ 4.8 หน้า Create Application

4.3.2 หลังจากที่ได้ Application

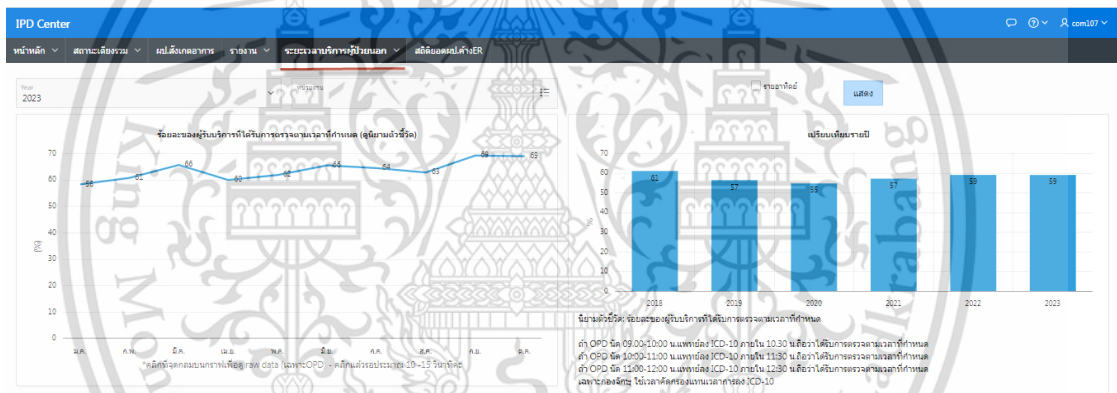
ในส่วนต่อไปเดิมทีหลังจากที่กดสร้าง Application เบื้องต้นจะมีเพียงหน้าเริ่มต้นที่ชื่อว่า home ปรากฏถ้า user ต้องการเพิ่มหน้าเมนูโปรแกรม ให้กดที่ปุ่ม Create page จะเป็นการสร้างหน้าต่างๆใน Application จากรูปที่ 4.9 จะแสดงหน้าต่างๆใน Application "IPU Center" หรือก็คือโปรแกรมที่ได้สร้างเพื่อนำเสนอในงานวิจัยนี้ ซึ่งจะประกอบไปด้วยหน้าเมนูต่าง ดังนั้นเพื่อสรุปในส่วนที่ได้สร้าง จึงขอยกตัวอย่างการสร้างบางหน้าเพื่ออธิบายขั้นตอนการสร้าง

เริ่มจากในกรณีที่ต้องการสร้างหน้าต่างเมนูในโปรแกรมให้กดที่ปุ่ม Create page ในรูปที่ 4.9 เพื่อเป็นการสร้างหน้าต่างเมนูขึ้นมาใหม่ หลังจากกดสร้างเสร็จจะปรากฏหน้าต่างปล่าวให้โดยใน ส่วนการสร้างหน้าระยะเวลาบริการผู้ป่วยนอกจะอธิบายในหัวข้อถัดไป



รูปที่ 4.9 การสร้าง Page และ Pages ต่างๆใน Application

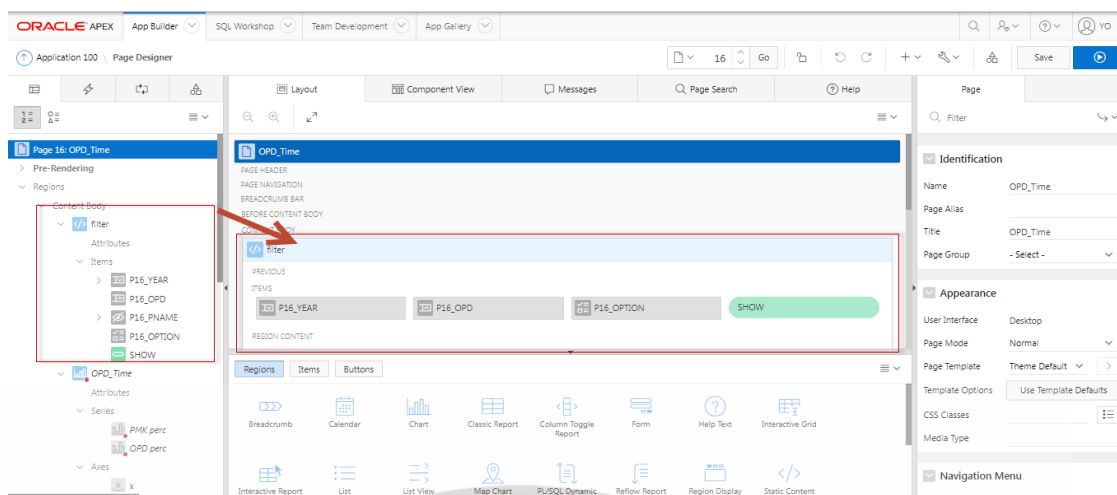
4.3.3 การสร้างหน้า 16. OPD_TIME หรือระยะเวลาบริการผู้ป่วยนอก



รูปที่ 4.10 ตัวอย่างหน้าระยะเวลาบริการผู้ป่วยนอกหลังสร้างเสร็จ

รูปที่ 4.10 คือผลตัวชี้วัดหน้าระยะเวลาบริการผู้ป่วยนอกที่แสดงเป็นกราฟเส้นและแท่งที่แสดงเป็นกึ่ง Dashboard เพื่อให้บุคลากรในโรงพยาบาลได้ดูผลข้อมูล ซึ่งขั้นตอนก่อนจะกลายเป็นโปรแกรมแบบในรูปที่ 4.10 จะมีขั้นตอนใดบ้าง อย่างแรกที่จำเป็นต้องเข้าใจคือใน 1 Application สามารถมีได้หลาย Pages ซึ่งแต่ละ Pages จะมีหน้าที่ต่างกันอย่างออกไปขึ้นอยู่กับความต้องการที่จะให้ Pages นั้นๆทำงานหรือแสดงผลอะไร เช่น จากรูปที่ 4.10 ใน Page นี้จะมีหน้าที่แสดง Dash board ข้อมูลต่างๆสำหรับเนื้อหาต่อจากนี้จะเป็นการอธิบายโครงสร้างของหน้า 16."opd_time"

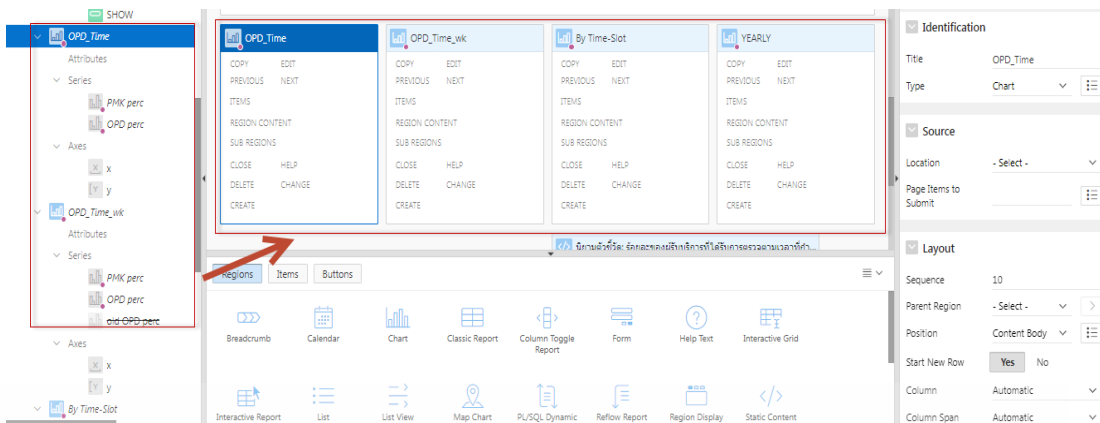
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 โครงสร้างในหน้าโปรแกรมส่วน Content Body

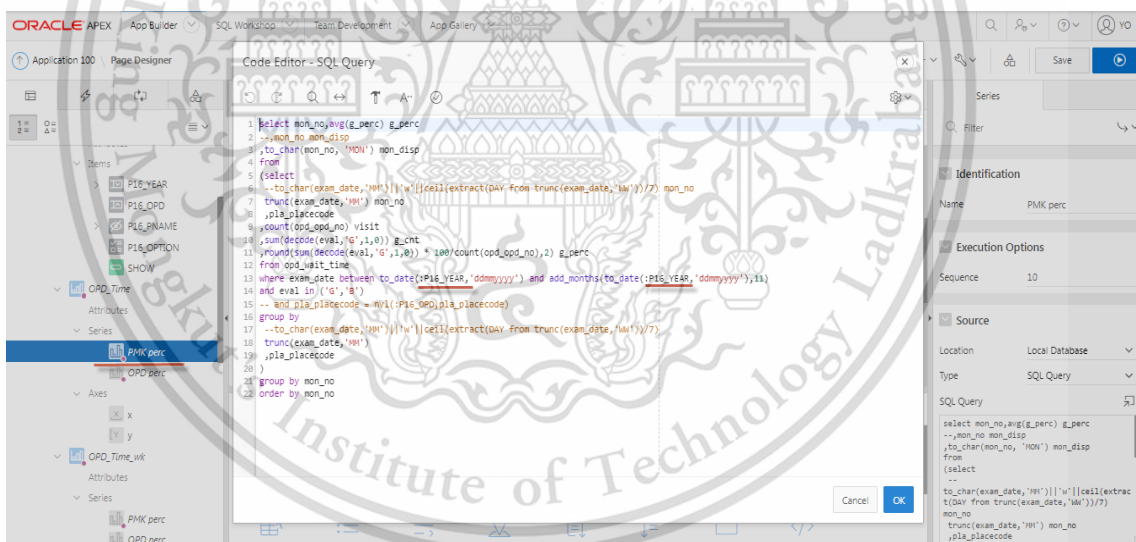
รูปที่ 4.11 ในส่วนแรกจะเริ่มด้วยบริเวณกรอบสีแดง ในส่วนนี้เรียกว่า Content Body ในการพัฒนา Application ด้วย Oracle Apex โดยทั่วไป Content Body หมายถึงส่วนหนึ่งของหน้าเว็บที่ใช้สำหรับแสดงเนื้อหาหลักของหน้านั้น ๆ ซึ่งสามารถประกอบด้วยข้อมูลหรือข้อความที่ต้องการแสดงให้กับผู้ใช้งานบนหน้านั้น ๆ ใน Oracle Apex เวลาที่คุณสร้างหน้าเว็บหรือหน้าจอแสดงผล คุณสามารถใช้ Content Body เพื่อเพิ่มเนื้อหาหลักที่ต้องการให้แสดงบนหน้านั้น ๆ ซึ่งสามารถเป็นข้อความปกติ รูปภาพ หรือสิ่งอื่น ๆ ที่คุณต้องการให้ผู้ใช้เห็นเมื่อเข้าถึงหน้านั้น การใช้ Content Body ทำให้คุณสามารถแสดงข้อมูลหรือเนื้อหาหลักให้กับผู้ใช้งานได้อย่างง่ายดาย และสามารถปรับแต่งการแสดงผลได้ตามที่คุณต้องการ โดยมีการเพิ่มรูปแบบหรือสไตล์ในการแสดงผลด้วย CSS หรือฟังก์ชัน JavaScript ต่าง ๆ ตามความเหมาะสม โดยใน Content Body ในกรอบแดงได้สร้าง ITEMS เพื่อส่งค่า Parameters ต่างๆเข้าไปในชุดคำสั่งต่างๆ(Queries) เพื่อให้โปรแกรมแสดงกราฟต่างๆให้เห็นรูปแบบกราฟต่างๆ ส่วนถัดไปจาก Parameters ได้สร้าง ITEMS ที่สามารถรับค่าจาก SQL ได้ โดยจะกำหนดสถานะ ITEM ให้ค่า Types เป็น Chart(ในที่นี้คือรูปแบบกราฟต่างๆ) โดยลักษณะโครงสร้างจะมีรูปแบบตามรูปที่ 4.12 วิธีการสร้าง ITEM ให้กดคลิกขวาที่ Content Body แล้วเลือก Created item

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 Items ต่างๆ ในหน้าการสร้างหน้าระยะเวลาบริการผู้ป่วยนอก

ให้สังเกตว่าช่อง Item ที่ชื่อ OPD_TIME จะแสดงตัวส่วนประกอบต่างๆที่จำเป็นต้องใส่ SQL เพื่อให้แสดงค่าที่ต้องการ เช่น PMK perc รูปที่ 4.13 เมื่อกดเข้าไปจะพบส่วนที่ให้ใส่ Source โดยสามารถนำ SQL ที่ได้จัดเตรียมไว้ลงมาใส่ได้เลยและจะเห็นจาก Query ว่ามีคำสั่งที่ให้ Query รับ Parameter จาก Item ก่อนหน้าที่ได้สร้างไว้



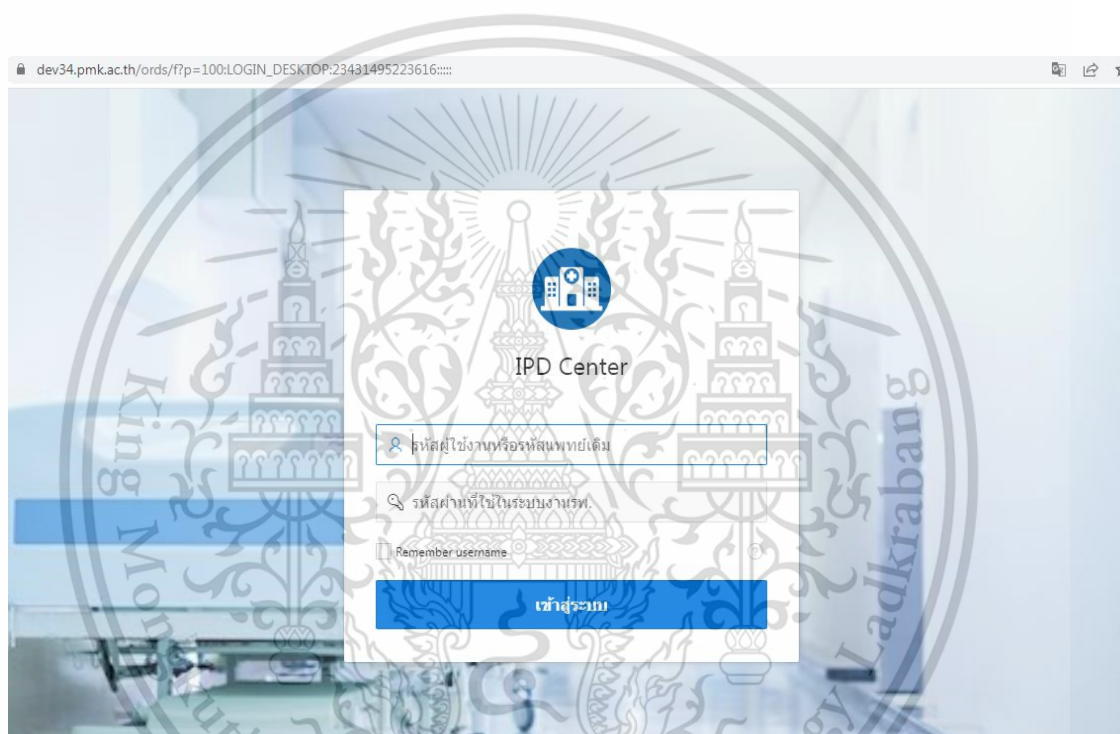
รูปที่ 4.13 ตัวอย่าง Query ในการสร้างกราฟแสดงผลหน้า opd_time

สำหรับ Query ตัวอย่างรูปที่ 4.13 ที่ได้นำเสนอจะเป็นตัวคำนวณค่าเฉลี่ยโดยรวมของทุก OPD ในโรงพยาบาลโดยมีค่า Parameter 'YEAR' เป็นตัวกำหนดระยะเวลาที่ต้องการดู เพื่อนำมาคำนวณหาค่าเฉลี่ยระยะเวลาบริการผู้ป่วยนอกตรงตามเวลาที่กำหนดที่ได้ตั้งขึ้นมาซึ่งจะมีผลสรุปให้ในบทถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดสอบโปรแกรม

หลังจากที่สร้างโปรแกรมเสร็จเรียบร้อยแล้วและดำเนินการเสนอผู้บริหารจึงเริ่มให้บุคลากรในโรงพยาบาลพระมงกุฎเกล้าได้ใช้งาน ก่อนอื่นนั้นจำเป็นต้องสร้าง URL ขึ้นมาใหม่เพื่อความสะดวกและทำให้ชื่อ URL สั้นลงดูเข้าใจง่าย โดยสาเหตุที่จำเป็นต้องเข้าผ่าน URL นั้นเพราะ Oracle Apex เป็นเครื่องมือที่สร้างได้แต่ Web application ซึ่งในปัจจุบันยังไม่มีเวอชันที่สามารถสร้าง Mobile Application ได้ สำหรับชื่อ URL ที่ได้สร้างขึ้นมาใหม่จะเป็น "dev34.pmk.ac.th/ords/f?p=100" และสร้าง QR CODE เพื่อประชาสัมพันธ์แจ้งใน Line กลุ่มของโรงพยาบาลเพื่อให้บุคลากรในโรงพยาบาลทราบและสามารถเข้าถึงได้

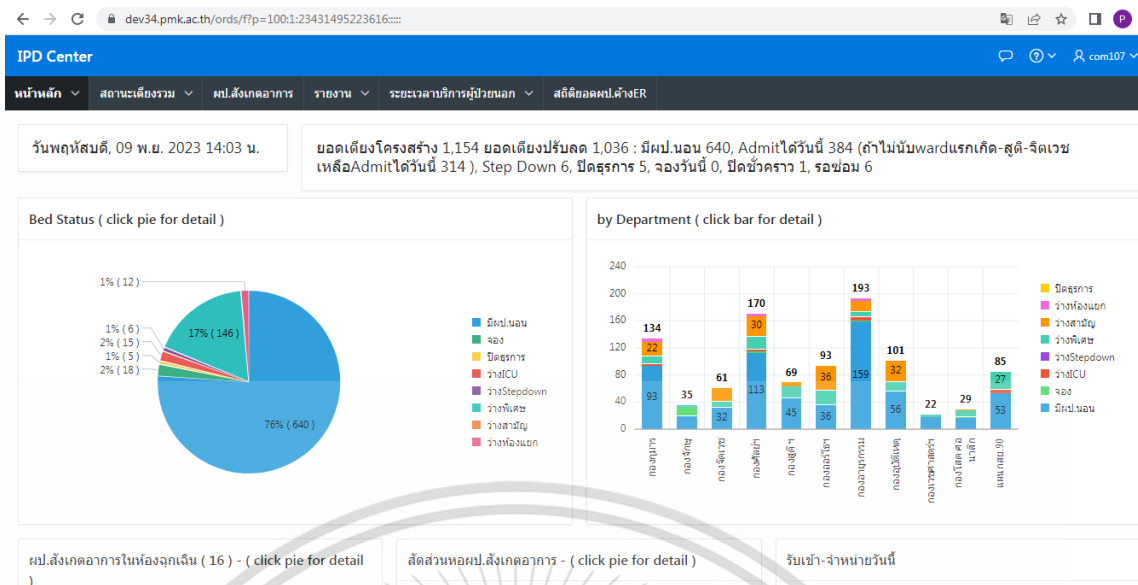


รูปที่ 4.14 ตัวอย่างหน้า LOGIN

Apex Oracle มีคุณสมบัติเป็นเว็บแอปพลิเคชันจึงจำเป็นต้องเข้าผ่านทาง Browser app. เช่น Google Chrome, Safari หรือ Browser อื่นๆตามอุปกรณ์ที่ผู้ใช้งานมี โดยเว็บแอป IPD Center สามารถเข้าเล่นทาง Smart phone ได้ด้วย

จากรูปที่ 4.14 เมื่อเข้าถึง URL ที่ส่งให้จะปรากฏหน้า Login โดยการ Login ใช้งานต้องเป็น User ID ที่ใช้แบบเดียวกับการใช้งานโรงพยาบาลซึ่งบุคลากรในโรงพยาบาลทุกคนจะทราบกันว่า ปัจจุบัน User ประจำตัวคืออะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 หน้าแรกของโปรแกรม IPD Center

ตารางที่ 4.2 รายการหน้าต่างเมนูใน IPD Center

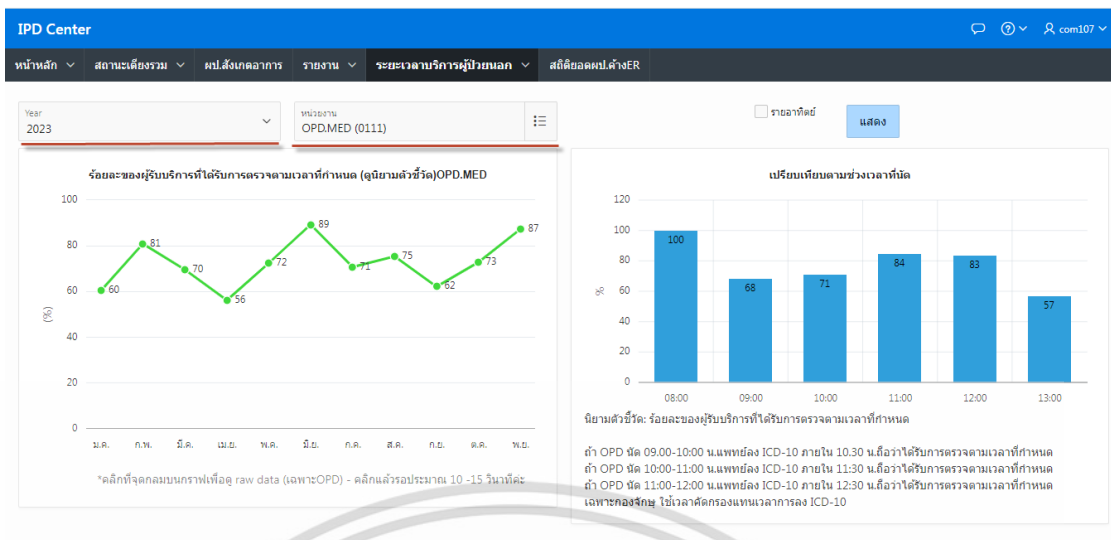
- | | |
|-----------------------|---------------------------------------------------------------|
| 1. หน้าหลัก | - จะเป็นหน้ารายงานเกี่ยวกับอัตราเตียงในปัจจุบัน |
| 2. สถานะเตียงรวม | - ไว้ดูสถานะเตียงต่างๆ เช่น ปิดเปิดเตียง, คนไข้ HN ไหนนอนอยู่ |
| 3. ผป. สังเกตการ | - ไว้สำหรับผู้ป่วยพิเศษ |
| 4. รายงาน | - เป็นหน้า Report ต่างๆที่แพทย์ใช้ |
| 5. ระยะเวลาผู้ป่วยนอก | - คืองานหลักของงานวิจัยนี้ ไว้ประเมินระยะเวลาคนคนไข้ในภาพรวม |
| 6. สถิติยอด ผป.ค้างER | - ยอดผู้ป่วยที่กำลังนอน ER (ห้องฉุกเฉิน) |

โดยเมนูที่ได้นำเสนอในงานวิจัยนี้คือเมนูที่ 5 ระยะเวลาผู้ป่วยนอก ในส่วนต่อไปจะเป็นการแนะนำวิธีการใช้งานของเมนูระยะเวลาผู้ป่วยนอก เมื่อกดดูสรุปสรณบริเวณแถบเมนูจะพบ 2 หน้าต่างเมนูย่อย ได้แก่

- อันดับ OPD ตามผลงาน
- อันดับ OPD ตามการพัฒนา

แต่ทั้งนี้ค่า KPI หลักที่ผู้บริหารและบุคลากรในโรงพยาบาลให้ความสนใจคือหน้าระยะเวลาบริการผู้ป่วยนอก โดยให้กดบริเวณแถบสีดำเพื่อเข้าเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ตัวอย่างหน้าระยะเวลาบริการผู้ป่วยนอกเมื่อลองใส่ค่าตัวค้นหา

เมื่อกดเข้ามาจะพบตัวกรอง 2 ตำแหน่งบริเวณที่มีเส้นสีแดง โดยในช่อง YEAR จะเป็นการเลือกช่วงเวลาตามปีที่ต้องการดูและตัวกรองอีกช่องจะเป็นรหัสห้องตรวจ จากตัวอย่างจะเป็นการดูข้อมูลในปี 2023 ของห้องตรวจ 0111 หรือ OPD อายุรกรรม เมื่อกดแสดงจะปรากฏกราฟ 2 ตัวโดยกราฟซ้ายมือจะเป็นร้อยละของผู้รับบริการที่ได้รับการตรวจตามเวลาที่กำหนด โดยจะแสดงเป็นรายเดือนในช่วงเวลาปีนั้นๆ ดังนั้นเมื่อได้เลือกช่วงเวลาปี 2023 ข้อมูลจึงแสดงถึงเพียงเดือนพฤศจิกายนจากรูปที่ 4.16

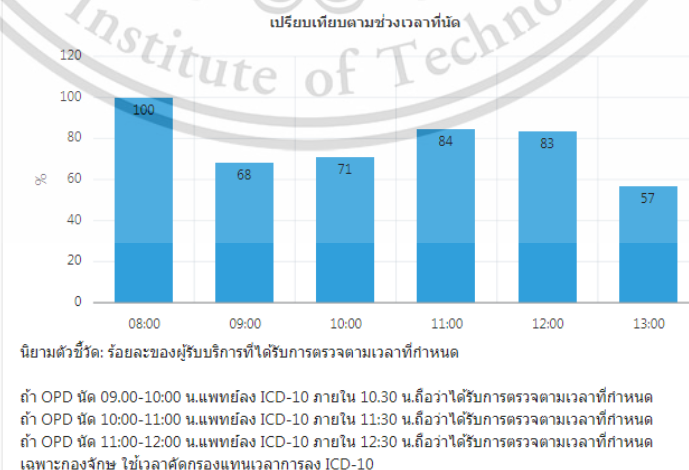
ผู้ใช้งานสามารถจุดจุดสีเขียวได้ เมื่อกดจะปรากฏตารางรายงานข้อมูลตั้งต้นที่จำเป็นใช้ก่อนจะนำไปหาค่าเฉลี่ยรายปีในกราฟฝั่งซ้ายเส้นสีเขียวในรูปที่ 4.16 เมื่อได้ลองกดปุ่มเฉพาะข้อมูลที่นับตามเกณฑ์จะแสดงข้อมูลที่สามารบอกผู้ใช้ได้ว่าการตรวจรักษาผู้ป่วยระยะเวลาไหนผ่านเกณฑ์หรือการตรวจครั้งไหนระยะเวลาไม่ผ่านเกณฑ์ตามรูปที่ 4.17 เมื่อมีข้อมูลพร้อมจึงสามารถนำข้อมูลมาประยุกต์สร้างเป็นกราฟแท่งที่แสดงตัวชี้วัดเปรียบเทียบตามช่วงเวลาที่น่าสนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hn ↑	วันตรวจ	เวลามา	รหัสพ.	เวลานัดพ.*	เวลาDx*	ตรวจแล้ว?	สัปดาห์ที่	ผลประเมิน
[REDACTED]	03 พ.ย. 2023	07:15	[REDACTED]	09.00-10.00	09:16	Y	5	ผ่านเกณฑ์
[REDACTED]	02 พ.ย. 2023	09:11	[REDACTED]	09.00-10.00	09:34	Y	5	ผ่านเกณฑ์
[REDACTED]	09 พ.ย. 2023	07:15	[REDACTED]	10.00-11.00	09:04	Y	1	ผ่านเกณฑ์
[REDACTED]	06 พ.ย. 2023	06:53	[REDACTED]	08.30-09.30	10:06	Y	1	ไม่ผ่าน
[REDACTED]	01 พ.ย. 2023	07:44	[REDACTED]	11.00-12.00	10:21	Y	5	ผ่านเกณฑ์
[REDACTED]	09 พ.ย. 2023	07:18	[REDACTED]	10.00-11.00	09:42	Y	1	ผ่านเกณฑ์
[REDACTED]	09 พ.ย. 2023	08:21	[REDACTED]	10.00-11.00	11:24	Y	1	ผ่านเกณฑ์
[REDACTED]	07 พ.ย. 2023	07:49	[REDACTED]	09.00-10.00	09:35	Y	1	ผ่านเกณฑ์
[REDACTED]	01 พ.ย. 2023	07:45	[REDACTED]	11.00-12.00	10:15	Y	5	ผ่านเกณฑ์
[REDACTED]	07 พ.ย. 2023	07:31	[REDACTED]	10.00-11.00	09:10	Y	1	ผ่านเกณฑ์
[REDACTED]	09 พ.ย. 2023	07:08	[REDACTED]	09.00-10.00	10:12	Y	1	ผ่านเกณฑ์
[REDACTED]	03 พ.ย. 2023	06:58	[REDACTED]	08.00-09.00	07:58	Y	5	ผ่านเกณฑ์
[REDACTED]	09 พ.ย. 2023	07:11	[REDACTED]	09.00-10.00	08:55	Y	1	ผ่านเกณฑ์
[REDACTED]	03 พ.ย. 2023	09:06	[REDACTED]	11.00-12.00	12:01	Y	5	ผ่านเกณฑ์

รูปที่ 4.17 ตัวอย่างข้อมูลเมื่อกดปุ่มเฉพาะข้อมูลที่นับตามเกณฑ์

รูปที่ 4.18 อธิบายในสกรีนกราฟฝั่งขวามือ ซึ่งจะบอกข้อมูลได้ว่าช่วงเวลาไหนที่หมอตรวจคนไข้ได้ตรงตามเกณฑ์ตัวชี้วัดซึ่งจะเป็นข้อมูลร้อยละของจำนวนทั้งหมดตามช่วงเวลาที่กำหนด(ปี) จากตัวอย่างจะเป็น ปี 2023 ห้องตรวจ 0111 โดยจะมีนิยามของตัวชี้วัด(KPI)อธิบายไว้ใต้กราฟ



รูปที่ 4.18 ตัวอย่างเมื่อใส่ตัวกรอง ปี 2023 ของห้องตรวจ 0111

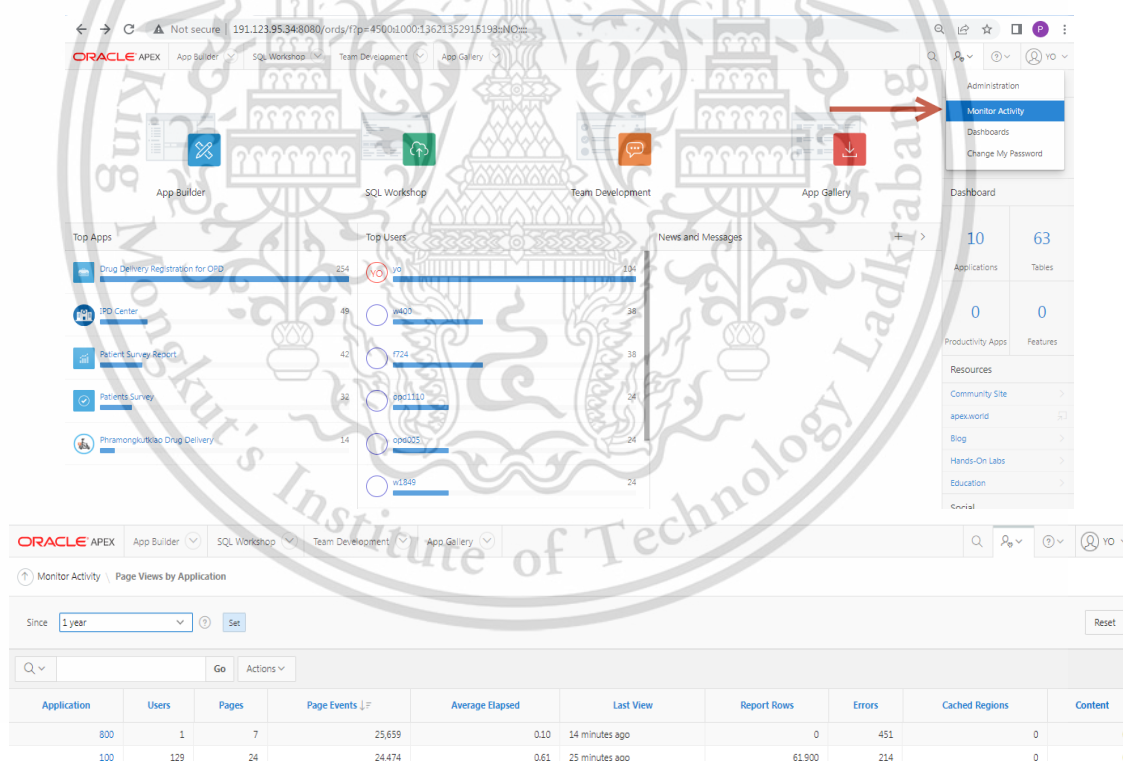
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผล

หลังการได้พัฒนาโปรแกรมตรวจและประเมินคุณภาพโรงพยาบาลได้มีการทดลองให้บุคลากรในโรงพยาบาลได้ใช้งานในช่วงเดือนพฤษภาคม ซึ่งผลตอบรับโดยรวมถือว่าดีโดยเฉพาะในส่วนของหน้าระยะเวลาบริการผู้ป่วยนอก ในส่วนของผลตอบรับผู้ใช้งานส่วนใหญ่ยอมรับผลของตัวชี้วัดระยะเวลาบริการผู้ป่วยนอก บางส่วนเกิดคำถามและนำไปสู่การแก้ไขปรับปรุง ซึ่งการแปรผลจะไม่ซับซ้อนมาก โดยหลักการของการใช้ SQL มาดึงข้อมูลแล้วมาคำนวณหาค่าเฉลี่ย

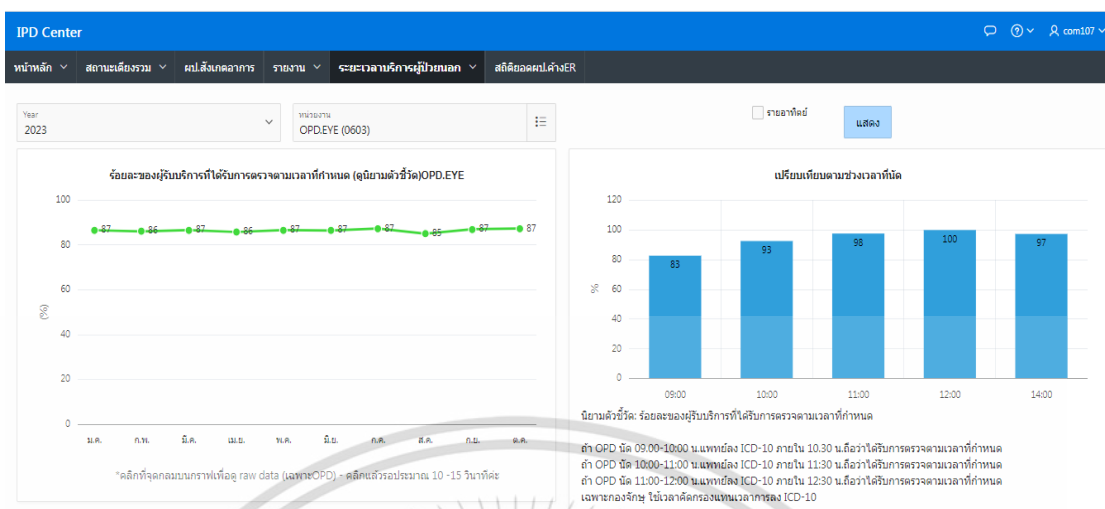
ในเบื้องต้นโปรแกรม APEX Oracle ยังไม่มีเมนูเก็บข้อมูลสรุปจำนวนผู้ใช้งานโดยรวมเท่าไร แต่สามารถเช็คได้จากเมนู Monitor Activity จะเป็นการแจ้งรายละเอียดว่าโปรแกรมมีสถานะอย่างไร Users ไหนเข้ามาใช้งานตัวอย่างจากรูปที่ 5.1



รูปที่ 5.1 หน้าตรวจสอบการเข้าชม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 สรุปผลโปรแกรม



รูปที่ 5.2 ตัวอย่างรูปแบบข้อมูลสรุปผลตามห้องและช่วงเวลาที่ยื่นหา

รูปที่ 5.2 อธิบายภาพสำหรับข้อสรุปที่ดีสามารถเข้าดูได้ในโปรแกรม เพราะสำหรับตัวหน้าระยะเวลาบริการผู้ป่วยนอกจะเป็นรูปแบบกราฟสรุปข้อมูล จะมี 2 ฝั่ง โดยกราฟด้านซ้ายจะเป็นข้อมูลสรุปรายเดือนของแต่ละปี ส่วนฝั่งขวาจะแบบตามช่วงเวลาซึ่งจะแตกต่างกันไปแต่ละห้องตรวจ ในภาพรวมพบว่าทุกห้องตรวจในโรงพยาบาลมีผลตัวชี้วัดอยู่ในระดับมาตรฐานตามระยะเวลาตรวจสำหรับปี 2018-2023 ค่าตัวชี้วัดอยู่ที่ 65 เปอร์เซ็นต์ ที่ให้บริการตรงตามเวลา

ซึ่งจากการได้ศึกษาข้อมูลย้อนหลังทำให้สามารถสรุปปัจจัยต่างๆที่ทำให้มีผลต่อระยะเวลาการตรวจคนไข้ผู้ป่วยนอก จากข้อมูลในระบบโรงพยาบาลตัวข้อมูลจะเป็นข้อมูลเชิงคณิตศาสตร์ที่สามารถนำไปหาค่าเฉลี่ยต่างๆได้แต่หาเหตุต่างๆไม่ได้ เช่น ข้อมูลสามารถบอกได้ว่าแพทย์ท่านนี้ตรวจคนไข้ซ้ำผิดปกติ ณ วันที่เท่าไร คนไข้ HN ประจำตัวอะไร ณ ห้องตรวจไหน แต่ข้อมูลไม่สามารถบอกได้ว่าทำไมแพทย์ท่านนี้ถึงตรวจลงโรคซ้ำ ดังนั้นการจะหาสาเหตุดังกล่าวจึงจำเป็นต้องไปสอบถามแพทย์ผู้นั้นเพื่อหาสาเหตุ ซึ่งตรงนี้เป็นจุดอ่อนจุดหนึ่งของระบบฐานข้อมูลโรงพยาบาลพระมงกุฎเกล้า ด้วยเหตุและสิ่งที่ได้จากการศึกษาข้อมูลย้อนหลังที่ได้จากการสร้างโปรแกรมหน้าระยะเวลาผู้ป่วยนอก ทำให้คณะผู้บริหารสามารถตัดสินใจในการปรับเปลี่ยนระบบบางอย่างเพื่อลดปัญหาและเปลี่ยนการเก็บข้อมูลโดยรายละเอียดจะอยู่ในส่วนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] อนุสร การเกษ พย.ม. ,สมชาย ภาณุมาศวิวัฒน์ ,วรรณชาติ ตาเลิศ พย.ม. (2564). การลดความแออัด ณ แผนกอุบัติเหตุและฉุกเฉินในโรงพยาบาลชุมชนแห่งหนึ่งจังหวัดศรีสะเกษ, สืบค้นเมื่อ 4 มิถุนายน 2566, จาก thaidj.org/index.php/JHS/article/view/10278/9043
- [2] ปรีตรา มั่นเหมาะ ,ธนัญญา วสุศรี. (2564). การจำลองสถานการณ์เพื่อลดเวลารอคอยของผู้ป่วยแผนกผู้ป่วยนอกโรงพยาบาลชุมชนจังหวัดสุพรรณบุรี, สืบค้นเมื่อ 4 มิถุนายน 2566, จาก ph01.tci-thaijo.org/index.php/kbej/article/download
- [3] Paramad. (2563). ระบบสารสนเทศโรงพยาบาลคืออะไร. สืบค้นเมื่อ 5 มิถุนายน 2566, จาก <https://bit2alone.wordpress.com/4-2/>
- [4] NAPGEO Oracle. What is Oracle Apex คืออะไร. สืบค้นเมื่อ 6 มิถุนายน 2566, จาก <https://apex-th.blogspot.com/2013/08/oracle-apex.html>
- [5] Oracle. (2565). Oracle คืออะไร สืบค้นเมื่อ 7 มิถุนายน 2566, จาก <http://mindphp.com/คู่มือ/73-คืออะไร/2283/>
- [6] Mldunbound Org. (2562). ความแตกต่างระหว่าง SQL และ Oracle SQL. สืบค้นเมื่อ 7 มิถุนายน 2566, จาก <https://th.mldunbound.org/contrast/difference-between-sql-server-and-oracle/>
- [7] วัชรเมธน์ ศรีเนธิโรทัย. (2552). SQL Syntax การใช้คำสั่ง ROUND. สืบค้นเมื่อ 8 มิถุนายน 2566, จาก <https://select2web.com/sql-round-function/>
- [8] Joshua Golub. (2564). SQL JOINS and SQL Syntax. สืบค้นเมื่อ 20 มิถุนายน 2566, จาก <https://www.finitewisdom.com/blogs/joshua-golub/2021/6/19/sql-joins>
- [9] แบริร์อีลเลน. (2565). ความแตกต่างระหว่าง Oracle DB และ MySQL. สืบค้นเมื่อ 20 มิถุนายน 2566, จาก <https://gocoding.org/th/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ	นาย ปรียเชษฐ์ ศรีอุทโยภาส
วัน เดือน ปีเกิด	06 พฤษภาคม พ.ศ.2536
ที่อยู่ปัจจุบัน	51 ถ.พหลโยธิน แขวงสามเสนใน เขตพญาไท กรุงเทพฯ 10400
ประวัติการศึกษา	(2562) Bachelor Technology management เกรตเฉลี่ย 2.5 (Assumption University)
ทุนการศึกษาที่ได้รับ	ไม่มี
ผลงานทางวิชาการ	ไม่มี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้