**แนวทางการออกแบบและพัฒนาแอปพลิเคชันจองตั๋วเครื่องบินด้วยเอสคิวแอลวิว**

**An Approach for Developing and Designing a Book Airline Tickets Application Using the SQL View**

**คำสำคัญ** แอปพลิเคชันจองตั๋วเครื่องบิน, ฐานข้อมูลเชิงสัมพันธ์, เอสคิวแอลวิว

**วัตถุประสงค์การดำเนินงาน**

งานวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ

1) ออกแบบและพัฒนาแอปพลิเคชันการจองตั๋วเครื่องบินและสร้างโครงสร้างฐานข้อมูลเชิงสัมพันธ์

2) สรุปข้อมูลจากตารางในรูปแบบของรายงานที่ผู้ใช้ต้องการ

2) ประยุกต์การใช้งานวิว โดยไม่ให้ผู้ใช้เรียกดูข้อมูลคิวรีที่ต้องการหรือทำการจองตั๋วเครื่องบิน โดยไม่กระทบกับระบบฐานข้อมูลหลักได้

4) ลดระยะเวลาการประมวลผลข้อมูลในการเขียนคำสั่งสำหรับการพัฒนาแอปพลิเคชันระบบจองตั๋วเครื่องบิน

**1. บทนำ**

เนื่องจากในปัจจุบันการทำธุรกรรมออนไลน์เข้ามามีบทบาทมากขึ้นทำให้ประชากรจำนวนมากมักจะไม่ต้องการที่จะออกจากที่พักอาศัยและต้องการทำกิจกรรมต่าง ๆ ผ่านระบบออนไลน์ทาง ดังนั้น ผู้พัฒนาจึงได้เล็งเห็นความสำคัญระบบออนไลน์จะเข้ามามีบทบาทเป็นอย่างมากจึงได้ทำการวิจัยและพัฒนาระบบจองตั๋วเครื่องบินออนไลน์ด้วยการใช้วิวมาประยุกต์ใช้เพื่ออำนวยความสะดวกแก่ผู้ใช้งานที่ต้องการจองตั๋วโดยที่ไม่จำเป็นต้องออกจากที่พักอาศัย

สาเหตุที่นำวิวมาประยุกต์ใช้นั้นก็เพื่อควบคุมความปลอดภัยให้กับข้อมูลสามารถสร้างโครงสร้างข้อมูลใหม่ ที่ทำให้ผู้ใช้เห็นแค่บางส่วนของฐานข้อมูลเท่านั้น ที่เป็นการป้องกันไม่ให้ผู้ใช้ได้เห็นข้อมูลทั้งหมด เราเรียกตารางข้อมูลประเภทนีว่า “ตารางเสมือน” หรือ “วิว” บางครังการออกแบบฐานข้อมูลในระดับกายภาพก็ต้องกําหนดตารางเป็ นลักษณะวิว เพื่อเป็นการป้องกันรักษาความปลอดภัยของข้อมูลและการควบคุมการใช้งานในระบบฐานข้อมูล เพราะหากสร้างเป็ นตารางข้อมูลจริงอาจจะเกิดความซ้ำซ้อนของข้อมูลมากซึ่งยากต่อการควบคุมการใช้งาน

อีกทั้งวิวยังช่วยลดระยะเวลาประมวลผลได้อีกทำให้ผู้พัฒนาไม่จำเป็นต้องเขียนคำสั่ง Join Query เพื่อให้แสดงผลในหน้าเว็บเนื่องจากมีสร้างวิวไว้อยู่แล้วจึงทำการ Query จากวิวได้เลยทำให้ลดระยะเวลาและลดหน่วยความจำของข้อมูลอีกด้วย

**2. ทฤษฎีที่เกี่ยวข้อง** (\*\*\*ควรเรียบเรียงให้กระชับไม่เกิน 15 บรรทัดต่อหัวข้อ\*\*\*)

2.1 แอปพลิเคชัน (Application) ~~แอพพลิเคชั่น คืออะไร~~[1]

Application (แอพพลิเคชั่น) หรือที่ทุกคนเรียกกันสั้นๆ ว่า App (แอพ) มันคือ โปรแกรมที่อำนวยความสะดวกในด้านต่างๆ ที่ออกแบบมาสำหรับ Mobile (โมบาย) Tablet (แท็บเล็ต) หรืออุปกรณ์เคลื่อนที่ ที่เรารู้จักกัน ซึ่งในแต่ละระบบปฏิบัติการจะมีผู้พัฒนาแอพพลิเคชั่นขึ้นมามากมายเพื่อให้ตรงกับความต้องการของผู้ใช้งาน ซึ่งจะมีให้ดาวน์โหลดทั้งฟรีและจ่ายเงิน ทั้งในด้านการศึกษา ด้านกรสื่อสารหรือแม้แต่ด้านความบันเทิงต่างๆ เป็นต้น

โมบายแอพฯ จะแบ่งออกเป็น 3 ประเภท คือ Native Application, Hybrid Applicationและ Web Application

Native App (เนทีฟ แอพ) คือ Application ที่ถูกพัฒนามาด้วย Library (ไลบรารี่) หรือ SDK (เอส ดี เค) เครื่องมือที่เอาไว้สำหรับพัฒนาโปรแกรมหรือแอพพิเคชั่น ของ OS Mobile (โอ เอส โมบาย) นั้นๆโดยเฉพาะ อาทิ Android (แอนดรอยด์) ใช้ Android SDK (แอนดรอยด์ เอส ดี เค), IOS (ไอ โอ เอส) ใช้ Objective c (ออปเจคทีฟ ซี), Windows Phone (วินโดว์ โฟน) ใช้ C# (ซีฉาบ) เป็นต้น

Hybrid Application (ไฮบริด แอพพลิเคชั่น) คือ Application ที่ถูกพัฒนาขึ้นมาด้วยจุดประสงค์ ที่ต้องการให้สามารถ รันบนระบบปฏิบัติการได้ทุก OS โดยใช้ Framework (เฟรมเวิร์ก) เข้าช่วย เพื่อให้สามารถทำงานได้ทุกระบบปฏิบัติการ

Web Application (เว็บ แอพพลิเคชั่น) คือ Application ที่ถูกเขียนขึ้นมาเพื่อเป็น Browser (บราวเซอร์) สำหรับการใช้งานเว็บเพจต่างๆ ซึ่งถูกปรับแต่งให้แสดงผลแต่ส่วนที่จำเป็น เพื่อเป็นการลดทรัพยากรในการประมวลผล ของตัวเครื่องสมาร์ทโฟน หรือ แท็บเล็ต ทำให้โหลดหน้าเว็บไซต์ได้เร็วขึ้น อีกทั้งผู้ใช้งานยังสามารถใช้งานผ่าน อินเตอร์เน็ตและอินทราเน็ต ในความเร็วตํ่าได้

2.2 ฐานข้อมูลเชิงสัมพันธ์ [2]

ฐานข้อมูลเชิงสัมพันธ์ นั้นหมายความว่า จะมีการจัดเก็บข้อมูลในลักษณะที่เป็นกลุ่มของข้อมูลที่มีความสัมพันธ์กัน ในฐานข้อมูลหนึ่งๆ สามารถที่จะมีตารางตั้งแต่ 1 ตารางเป็นต้นไป และในแต่ละตารางนั้นก็สามารถมีได้หลายคอลัมน์ (Column) หลายแถว (Row) ตัวอย่างเช่น เราต้องการเก็บข้อมูลพนักงาน ในตารางของข้อมูลพนักงานก็จะประกอบด้วยคอลัมน์ ที่อธิบายชื่อ นามสกุล ที่อยู่ เงินเดือน แผนกที่สังกัด เป็นต้น และในตารางนั้น ก็สามารถที่จะมีข้อมูลพนักงานได้มากกว่า 1 คน (Row) และตารางข้อมูลพนักงานนั้นอาจจะมีความสัมพันธ์กับตารางอื่น เช่น ตารางที่เก็บชื่อและจำนวนบุตรของพนักงาน

ฐานข้อมูลเชิงสัมพันธ์ถูกออกแบบมาเพื่อลดความซ้ำซ้อนของการเก็บข้อมูล และสามารถเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ โดยมีหลักดังนี้

1. ตารางจะต้องมีชื่อไม่ซ้ำกัน
2. แต่ละฟิลด์จะบรรจุประเภทข้อมูลเพียงชนิดเดียวเท่านั้นแน่นอน
3. ข้อมูลในแต่ละเรคอร์ดจะต้องไม่ซ้ำกัน

นอกจากนี้แต่ละตารางยังสามารถเริยกได้อีกอย่างว่ารีเลชัน (Relation) แถวแต่ละแถวภายในตารางเรียกว่าทูเปิล (Tuple) และคอลัมน์เรียกว่าแอททริบิวต์ (Attribute)

จุดเด่นของข้อมูลเชิงสัมพันธ์

1. ง่ายต่อการเรียนรู้ และการนำไปใช้งาน ทำให้เห็นภาพข้อมูลชัดเจน
2. ภาษาที่ใช้จัดการข้อมูลเป็นแบบซีเควล ซึ่งมีประสิทธิภาพสูงเข้าใจง่าย
3. การออกแบบระบบมีทฤษฎีรองรับ สามารถลดความซ้ำซ้อนของข้อมูลได้

กฎที่เกี่ยวข้องกับคีย์ในฐานข้อมูลเชิงสัมพันธ์

1. กฎความบูรณภาพของเอนทิตี้ (The Entity Intergrity Rule)

กฎนี้ระบุไว้ว่าแอททริบิวต์ใดที่เป็นคีย์หลัก ข้อมูลในแอททริบิวต์นั้นจะเป็นค่าว่าง(Null) ไม่ได้ ความหมายของการเป็นค่าว่างไม่ได้(Not Null) หมายความถึง ข้อมูลของแอททริบิวต์ที่เป็นคีย์หลักจะไม่ทราบค่าที่แน่นอนหรือไม่มีค่าไม่ได้

2. กฎความบูรณภาพของการอ้างอิง (The Referential IntegrityRule)

การอ้างอิงข้อมูลระหว่างรีเลชั่นในฐานข้อมูลเชิงสัมพันธ์จะใช้คีย์นอกของรีเลชั่นหนึ่งไปตรวจสอบกับค่าของแอททริบิวต์ที่เป็นคีย์หลักของอีกรีเลชั่นหนึ่งเพื่อเรียกดูข้อมูลอื่นๆที่เกี่ยวข้องหรือค่าของคีย์นอกจะต้องอ้างอิงให้ตรงกับค่าของคีย์หลักได้จึงจะสามารถเชื่อมโยงข้อมูลระหว่างสองรีเลชั่นได้สำหรับคีย์นอกจะมีค่าว่างได้หรือไม่ขึ้นอยู่กับกฎเกณฑ์การออกแบบฐานข้อมูล เช่น ในกรณีที่รีเลชั่นพนักงานมี Depnoเป็นคีย์นอกอาจจะถูกระบุว่าต้องทราบค่าแต่ในกรณีพนักงานทดลองงานอาจยังไม่มีค่า Depno เพราะยังไม่ได้ถูกบรรจุในกรณีที่มีการลบหรือแก้ใขข้อมูลของแอททริบิวต์ที่เป็นคีย์หลักซึ่งมีคีย์นอก จากอีกรีเลชั่นหนึ่งอ้างอิงถึง จะทำการลบหรือแก้ใขข้อมูลได้หรือไม่ ขึ้นอยู่กับการออกแบบฐานข้อมูล ว่าได้ระบุให้แอททริบิวต์มีคุณสมบัติอย่างไร ซึ่งมีโอกาสเป็นไปได้ 4 ทางเลือก

การลบหรือแก้ไขข้อมูลแบบมีข้อจำกัด (Restrict) การลบหรือแก้ไขข้อมูลจะกระทำได้ เมื่อข้อมูลของคีย์หลักในรีเลชั่นหนึ่งไม่มีข้อมูลที่ถูกอ้างอิง โดยคีย์นอกของอีกรีเลชั่นหนึ่งเช่น รหัสแผนก Depno ในรีเลชั่นDepจะถูกแก้ใขหรือลบทิ้งต่อเมื่อไม่มีพนักงานคนใดสังกัดอยู่ในแผนกนั้น

การลบหรือแก้ไขข้อมูลแบบต่อเรียง (Cascade) การลบหรือการแก้ใขข้อมูล จะทำแบบเป็นลูกโซ่ คือ หากมีการแก้ไขหรือลบข้อมูลของคีย์หลักในรีเลชั่นหนึ่งระบบจะทำการลบหรือแก้ใขข้อมูลของคีย์นอกในอีกรีเลชั่นหนึ่งที่อ้างอิงถึงข้อมูลของคีย์หลักที่ถูกลบให้ด้วย เช่น ในกรณีที่ยกเลิกแผนก 9 ในEntityแผนก ข้อมูลของพนักงานที่อยู่แผนก 10 ในEntityพนักงานจะถูกลบออกไปด้วย

การลบหรือแก้ไขข้อมูลโดยเปลี่ยนเป็นค่าว่าง (Nullify) การลบหรือแก้ใขข้อมูลจะทำได้เมื่อมีการเปลี่ยนค่าของคีย์นอกในข้อมูลที่ถูกอ้างอิงให้เป็นค่าว่างเสียก่อน เช่น พนักงานที่อยู่ในแผนกที่ 9 จะถูกเปลี่ยนค่าเป็นค่าว่างก่อนหลังจากนั้น การลบข้อมูลของแผนกที่มีรหัส 9 จะถูกลบทิ้งหรือแก้ไขทันที ภายใน Entity แผนก

การลบหรือแก้ไขข้อมูลแบบใช้ค่าโดยปริยาย ( Default) การลบหรือแก้ไขข้อมูลของคีย์หลัก สามารถทำได้โดยถ้าหากมีคีย์นอกที่อ้างอิงถึงคีย์หลักที่ถูกลบหรือแก้ไข ก็จะทำการปรับค่าของคีย์นอกนั้นโดยปริยาย (Default Value) ที่ถูกกำหนดขึ้นเช่น ในกรณีที่ยกเลิกแผนก 9 ในEntity แผนก ข้อมูลของพนักงานที่อยู่แผนก 9 ใน Entity พนักงานจะถูกเปลี่ยนค่าเป็น 00 ซึ่งเป็นค่าโดยปริยาย ที่หมายความว่า ไม่ได้สังกัดแผนกใด ไม่เข้าใจกลับไปอ่านใหม่

2.3 Database SQL – VIEW [3]

จาการใช้งาน SQL Select statement จะเห็นว่า ในบางครั้งเรามักจะมี Select statement ที่ต้องใช้งานประจำเพื่อดึงข้อมูลออกมาในรูปแบบที่ต้องการ เราสามารถใช้ SQL View นี้ช่วยได้ โดย SQL View จะทำการเก็บรูปแบบการ query หรือ SQL Select statement ที่เราตั้งไว้ใน SQL View ซึ่งสามารถทำได้ตั้งแต่ query ข้อมูลทั้งหมดของ table รวมถึงการทำ SQL Join, SQL Union, SQL Intersect และ SQL Except ที่ผลลัพธุ์ออกมาในรูปแบบตาราง หรือ table

View Table

คือ การแสดงข้อมูลที่ผู้ใช้งานสร้างขึ้น ข้อมูลอาจมาจากการ Select 1 Table หรือ มาจากการ join กันของหลาย Tables ก็ได้ โดยเราจะใช้งาน View Table ก็ต่อเมื่อ มีการเรียกใช้ข้อมูลนั้นซ้ำๆเพื่อประหยัดเวลาในการเขียนคำสั่ง SQL

คุณสมบัติของ VIEW

* สร้างโครงสร้างของข้อมูลในรูปแบบที่ user ใช้งานปกติจาก table
* ปรับเปลี่ยนข้อมูลภายใน table ให้แสดงออกมาในรูปแบบที่ต้องการ โดยไม่กระทบกับข้อมูลหลัก
* สรุปข้อมูลจาก table ในรูปแบบของ report ที่ user ต้องการ

(\*\*\*การยกตัวอย่าง สามารถยกตัวอย่างมาแค่ 1 ตัวอย่างได้\*\*\*)

รูปแบบการสร้าง VIEW (Syntax)

การใช้งาน SQL View สามรถสร้างได้จากรูปแบบของ SQL Select statement ที่ต้องการ โดยสามารถสร้างจาก table เดียว หรือ หลาย table ก็ได้ แต่ต้องมีสิทธิ์ในการเข้าถึงข้อมูลภายใต้ table นั้น (GRANT VIEW)

CREATE VIEW view\_name AS

SELECT column1, column2.....

FROM table\_name

WHERE [condition];

ตัวอย่างคำสั่ง VIEW TABLE

Table – CUSTOMERS

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

ทำการสร้าง VIEW จากการ Select ข้อมูลบน table CUSTOMERS

SQL > CREATE VIEW CUSTOMERS\_VIEW AS

SELECT name, age

FROM CUSTOMERS;

ผลลัพธ์ที่ได้ จากการลอง select ข้อมูลใน VIEW

SQL > SELECT \* FROM CUSTOMERS\_VIEW;

+----------+-----+

| name | age |

+----------+-----+

| Ramesh | 32 |

| Khilan | 25 |

| kaushik | 23 |

| Chaitali | 25 |

| Hardik | 27 |

| Komal | 22 |

| Muffy | 24 |

+----------+-----+

เงื่อนไข WITH CHECK OPTION

WITH CHECK OPTION กำหนดมาเพื่อใช้งานการ CREATE VIEW statement เป้าหมายของ WITH CHECK OPTION มีเพื่อให้มั่นใจได้ว่าการ INSERT หรือ UPDATE อยู่ในเงื่อนไขของ VIEW ซึ่งถ้าไม่อยู่ในเงื่อนไข การ INSERT หรือ UPDATE นั้นจะได้ผล error

CREATE VIEW CUSTOMERS\_VIEW AS

SELECT name, age

FROM CUSTOMERS

WHERE age IS NOT NULL

WITH CHECK OPTION;

ด้วยเงื่อนไข WITH CHECK OPTION จะเห็นว่าข้อมูล column age ห้ามมีค่าเป็น NULL

การ UPDATE VIEW

view จะสามารถ update ภายใต้เงื่อนไข statement ดังนี้

* SELECT clause ห้ามใช้ DISTINCT
* SELECT clause ห้ามใช้ summary function
* SELECT clause ห้ามใช้ set function
* SELECT clause ห้ามใช้ set operator
* SELECT clause ห้ามใช้ ORDER BY
* FROM clause ห้าม JOIN table
* WHERE clause ห้ามทำ subquery
* ห้ามใช้งาน GROUP BY และ HAVING
* NOT NULL column ทั้งหมดจะต้องอยู่ใน VIEW ด้วยเพื่อให้ INSERT function ทำงานถูกต้อง

ถ้าเงื่อนไขทั้งหมดถูกต้อง เราจึงสามารถใช้งาน SQL VIEW update ได้

SQL > UPDATE CUSTOMERS\_VIEW

SET AGE = 35

WHERE name = 'Ramesh';

ผลลัพธ์ที่ได้

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 35 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

การ INSERT VIEW

ใช้เงื่อนไขเดียวกับ UPDATE VIEW ซึ่งจากตัวอย่าง เราไม่สามารถทำการ INSERT VIEW บน CUSTOMERS\_VIEW นี้ได้เพราะ column ทั้งหมดของ VIEW ไม่ได้มีเงื่อนไข NOT NULL ทั้งหมด

การ DELETE VIEW

ใช้เงื่อนไขเดียวกับ UPDATE VIEW และ INSERT VIEW

SQL > DELETE FROM CUSTOMERS\_VIEW

WHERE age = 22;

ผลลัพธ์ที่ได้

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 35 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

รูปแบบการลบ VIEW (Syntax)

DROP VIEW view\_name;

ตัวอย่างการ DROP VIEW

DROP VIEW CUSTOMERS\_VIEW;

**3. งานวิจัยที่เกี่ยวข้อง**

รัตนา สุวรรณวิชนีย์, ปราลี มณีรัตน์ และ บุญศิริ มะสัน (2560) [4] ได้ออกแบบมุมมองข้อมูลเพื่อพัฒนาระบบรายงานหลายมิติของสำนักงานคณะกรรมการวิจัยแห่งชาตินำมาใช้งานในการบริหารจัดการข้อมูลงานวิจัยและงบประมาณวิจัยของหน่วยงานภาครัฐและแหล่งทุนของหน่วยงานต่าง ๆ แต่ทาง วช. ยังขาดเครื่องมือในการวิเคราะห์ข้อมูลจำนวนมากในระบบ NRMS ให้ได้ทันเวลาและนำเสนอข้อมูลในหลากหลายมิติที่ตรงตามความต้องการใช้ประโยชน์ของผู้ใช้ เพื่อนำเสนอรัฐบาลหรือหน่วยงานนโยบายสำหรับใช้ในการตัดสินใจเชิงนโยบาย โดยการใช้ Microsoft SQL Server 2012 และการกำหนดมิติของข้อมูลโดยการใช้โปรแกรมทาโบล (Tableau) ซึ่งเป็นโปรแกรมทางด้านระบบธุรกิจอัจฉริยะ (Business Intelligence : BI) ด้วย (view) ในการออกแบบและสร้างมุมมองข้อมูล (view) จากระบบ NRMS จากนั้นเชื่อมต่อมุมมองข้อมูลกับโปรแกรมทาโบล เพื่อกำหนดมิติของข้อมูล และค่าที่ใช้ในการคำนวณ เพื่อใช้ในการพัฒนาระบบรูปแบบรายงานหลายมิติเพื่อสนับสนุนการตัดสินใจเชิงนโยบาย ของ วช.

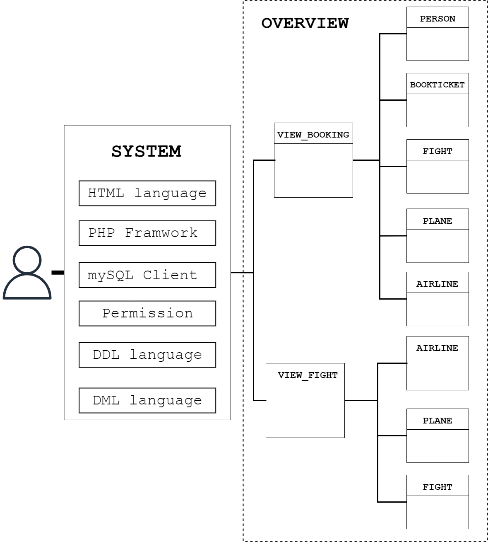
ปัทมา เที่ยงสมบุญและนิเวศ จิระวิชิตชัย (2561) [5] ได้พัฒนาระบบธุรกิจอัจฉริยะเพื่อสนับสนุนการพยากรณ์และการตัดสินใจของผู้บริหารกรณีศึกษากลุ่มโรงพยาบาล โดยใช้โปรแกรม Microsoft Power BI ร่วมกับ Oracle Database Server 11g และนำข้อมูลการรักษาโรคไม่ติดต่อเรื้อรัง (NCDs) ปี 2553-2560 จากฐานข้อมูลระบบสารสนเทศโรงพยาบาล ด้วยการสร้าง View Oracle ด้วยโปรแกรม PL/SQL Developer เพื่อเพิ่มประสิทธิภาพในการวิเคราะห์และการวางแผนกลยุทธ์ขององค์กรได้และใช้อัลกอริทึมการพยากรณ์เทคนิคเอ็กซ์โพเนนเชียลอย่างง่ายของโปรแกรม Microsoft Power BI ในการพยากรณ์แนวโน้มรายได้

การรักษาพยาบาลและจำนวนผู้ป่วยกลุ่มโรคเรื้อรังไม่ติดต่อ NCDs มีอัตราเพิ่มขึ้น และทดสอบประสิทธิภาพของระบบโดยการทำแบบสอบถามความพึงพอใจในการใช้งานผู้บริหาร

รองศาสตราจารย์ดร.นิตยา เกิดประสพ และ รองศาสตราจารย์ดร.กิตติศักดิ์ เกิดประสพ (2553) [6] ได้ทำรายงานวิจัยเรื่องการเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลและโมเดลจากการทำเหมืองข้อมูล โดยวัตถุประสงค์หลักของงานวิจัยนี้ต้องการพัฒนาแนวทางการใช้ประโยชน์วิวข้อมูลและโมเดลที่ได้จากการทำเหมืองข้อมูลเพื่อเพิ่มประสิทธิภาพการประมวลผลข้อคำถาม จากการพัฒนาระบบต้อนแบบและทดสอบประมวลด้วยคำถาม กับฐานข้อมูลจริง

**4. การดำเนินงานวิจัย**

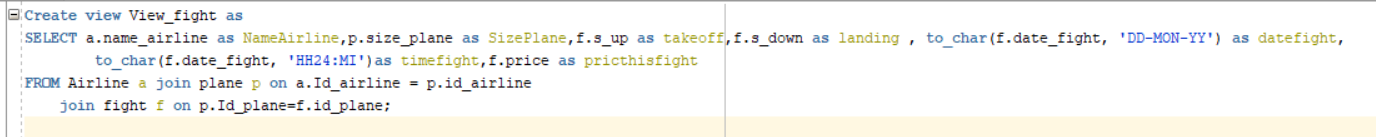
การดำเนินงานของการออกแบบและพัฒนาแอปพลิเคชันจองตั๋วเครื่องบินด้วยเอสคิวแอลวิว ซึ่งประกอบด้วยขั้นตอน ดังนี้ 1. ขั้นตอนการเตรียมข้อมูลนำเข้า (Input Data) 2. ขั้นตอนการประมวลผล (Processing) 3. ขั้นตอนการนำเสนอผลลัพธ์ของข้อมูล (Output Data) ดังภาพที่ 1



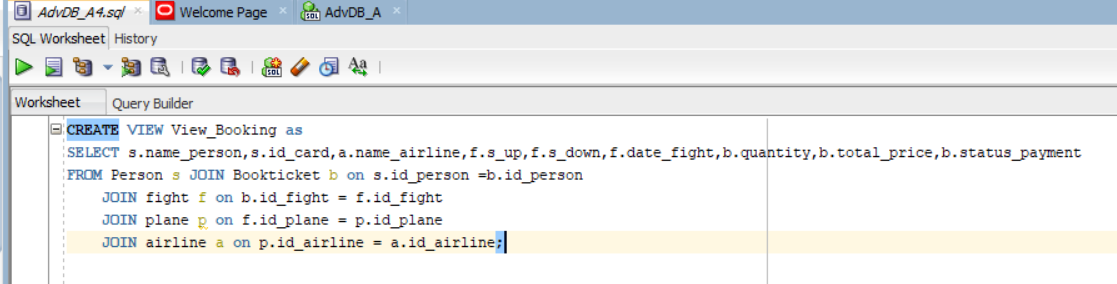
**ภาพที่ 1** แสดภาพรวมการดำเนินงาน

4.1 ขั้นตอนการเตรียมข้อมูลนำเข้า (Input Data)

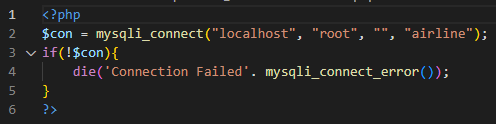
ในขั้นตอนการเตรียมข้อมูลนำเข้าจะเป็นการสร้างคำสั่งเพื่อดึงข้อมูลจากฐานข้อมูลระบบ โดยการสร้าง View SQL ในโปรแกรม Oracle เพื่อดึงเฉพาะข้อมูลที่ผู้ใช้ต้องการเรียกดูโดยต้องไม่กระทบกับฐานข้อมูลหลักดังภาพที่ 2 กับ 3 และทำการเชื่อมต่อฐานโดยการใช้โปรแกรม Visual Studio Code ดังภาพที่ 3



**ภาพที่ 2** การสร้างวิวข้อมูลไฟท์การบิน



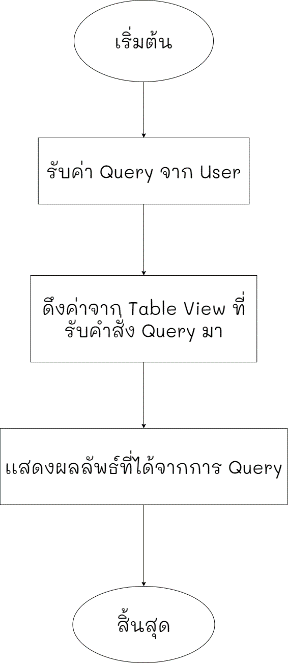
**ภาพที่ 3** การสร้างวิวข้อมูลการจอง



**ภาพที่ 4** คำสั่งการเชื่อมต่อฐานข้อมูลกับ Web Application

4.2 ขั้นตอนการประมวลผล (Processing)

ในขั้นตอนการประมวลผลข้อมูล จะเป็นส่วนต่อจากการนำเข้าข้อมูล โดยจะทำการสร้างคำสั่งเพื่อให้สามารถแสดงผลลัพท์ที่ผู้ใช้ต้องการเรียกดู โดยจะเริ่มทำการตรวจสอบว่าผู้ใช้ต้องการเรียกดูข้อมูลอะไร เช่น สายการบิน ไฟท์การบิน การจอง เป็นต้นโดยจะดึงข้อมูลจากวิวดังภาพที่ 5



**ภาพที่ 5** แสดงการประมวลผลข้อมูลเมื่อผู้ใช้เรียกดู

4.3 ขั้นตอนการนำเสนอผลลัพธ์ของข้อมูล (Output Data)

ในขั้นตอนการนำเสนอผลลัพธ์ของข้อมูลนนั จะแสดงข้อมูลที่เป็นผลลัพธ์ที่ได้จากขั้นตอนการประมวผลโดยจะนำเสนอผ่านเว็บแอปพลิเคชัน เพื่อให้ผู้ใช้สามารถนำข้อมูลผลลัพธ์ไปใช้งานต่อยอดได้ เช่น การเรียกดูไฟท์การบิน การเรียกดูสายการบิน การจองตั๋วเครื่องบิน การเรียกดูข้อมูลการจอง โดยเลือกใช้ View เพื่อผู้ใช้นั้นมีส่วนกับข้อมูลโดยตรงเพื่อป้องกันอันตรายที่อาจจะเกิดขึ้นกับระบบฐานข้อมูลหลัก จากการดำเนินการทำให้ฐานข้อมูลทำงานร่วมกันกับเว็บแอปพลิเคชันได้เป็นอย่างดีโดยวิว View เป็นตัวกลาง

**5. ผลการวิจัย**

จากการดำเนินการวิจัยได้ต่อยอดออกมาเป็นรูปแบบส่วนต่อประสานระหว่างผู้ใช้งานภายในระบบซึ่งมีพีเจอร์การทำงานที่สามารถตอบสนองต่อผู้ใช้งาน ดังนี้

1. การแสดงข้อมูลไฟต์การบินดังภาพที่ 6



**ภาพที่ 6** แสดงส่วนต่อประสานไฟต์การบิน

2. การแสดงส่วนหน้าการจองเมื่อกดปุ่ม จอง จะทำการแสดงข้อมูลไฟท์บินที่ต้องการจอ ดังภาพที่ 7



**ภาพที่ 7** แสดงส่วนต่อประสานไฟต์การบิน

6**. รายการอ้างอิง**

[1] mango Grow your success. “Application แอพพลิเคชั่น คืออะไร.” Available Online at https://www.mangoconsultant.com/th/news-knowledge/knowledge/274-application-%E0%B9%81%E0%B8%AD%E0%B8%9E%E0%B8%9E%E0%B8%A5%E0%B8%B4%E0%B9%80%E0%B8%84%E0%B8%8A%E0%B8%B1%E0%B9%88%E0%B8%99-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3#:~:text=Application%20(%E0%B9%81%E0%B8%AD%E0%B8%9E%E0%B8%9E%E0%B8%A5%E0%B8%B4%E0%B9%80%E0%B8%84,%E0%B8%8B%E0%B8%B6%E0%B9%88%E0%B8%87%E0%B8%88%E0%B8%B0%E0%B8%A1%E0%B8%B5%E0%B9%83%E0%B8%AB%E0%B9%89%E0%B8%94%E0%B8%B2%E0%B8%A7%E0%B8%99%E0%B9%8C%E0%B9%82%E0%B8%AB%E0%B8%A5%E0%B8%94

[2] 2403:6200:8896:6084:507d:6203:f895:53f6. “ฐานข้อมูลเชิงสัมพันธ์.” Available Online at https://th.wikipedia.org/wiki/%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5%E0%B9%80%E0%B8%8A%E0%B8%B4%E0%B8%87%E0%B8%AA%E0%B8%B1%E0%B8%A1%E0%B8%9E%E0%B8%B1%E0%B8%99%E0%B8%98%E0%B9%8C

[3] Suphakit Annoppornchai. “Database SQL – VIEW” Available Online at https://saixiii.com/database-sql-view/

[4] รัตนา สุวรรณวิชนีย์. ปราลี มณีรัตน์. และ บุญศิริ มะสัน. “การออกแบบมุมมองข้อมูลเพื่อพัฒนาระบบรายงานหลายมิติของสำนักงานคณะกรรมการวิจัยแห่งชาติ.” การประชุมวิชาการระดับชาติครั้งที่ 2มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา ประจำปีการศึกษา 2560, หน้าที่

[5] ปัทมา เที่ยงสมบุญ. และนิเวศ จิระวิชิตชัย. “พัฒนาระบบธุรกิจอัจฉริยะเพื่อสนับสนุนการพยากรณ์และการตัดสินใจของผู้บริหารกรณีศึกษากลุ่มโรงพยาาล.”, ปีที่ 5, ฉบับที่ 4, หน้าที่ , เดือนกรกฎาคม – สิงหาคม 2561

[6] รองศาสตราจารย์ดร.นิตยา เกิดประสพ. และ รองศาสตราจารย์ดร.กิตติศักดิ์ เกิดประสพ. “การเพิ่มประสิทธิภาพการประมวลผลข้อคำถามด้วยวิวข้อมูลและโมเดลจากการทำเหมืองข้อมูล.” รายงานวิจัย : มหาวิทยาลัยเทคโนโลยีสุรนารี, 2553