ระบบคลังข้อมูล กรณีศึกษาสำนักทะเบียนและประมวลผล

DATA WAREHOUSE; CASE STUDY OF THE OFFICE OF THE REGISTRAR

ขวัญญู บุญศิริรัตนพันธ์

ธรถิน ค้ำแดงพร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ระบบคลังข้อมูล กรณีศึกษาสำนักทะเบียนและประมวลผล

DATA WAREHOUSE: CASE STUDY OF THE OFFICE OF THE REGISTRAR

ชนัญญ บุญศิริสัมพันธ์

สรสิช ด่านสถาพร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

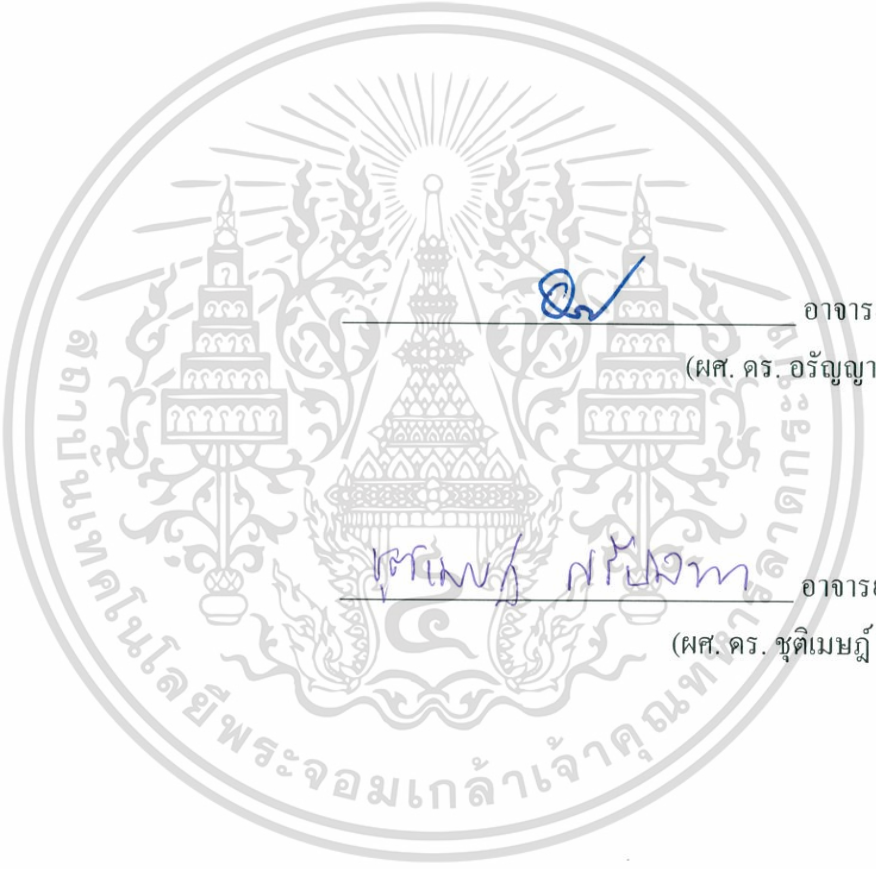
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบคลังข้อมูล กรณีศึกษาสำนักทะเบียนและประมวลผล

Data Warehouse: Case Study of the Office of the Registrar

ผู้จัดทำ

1. นายชนัญญ บุญสิริสัมพันธ์ รหัสนักศึกษา 54010266
2. นายสรสิช ด่านสถาพร รหัสนักศึกษา 54011336



_____ อาจารย์ที่ปรึกษา

(ผศ. ดร. อรัญญา วลัยรัชต์)

_____ อาจารย์ที่ปรึกษา

(ผศ. ดร. ชุติเมษฎ์ ศรีนิลทา)

Data Warehouse: Case Study of the Office of the Registrar

Mr. Chanunyoo	Boonsirisumpun	54010266
Mr. Sorasit	Dansataporn	54011336
Asst. Prof. Dr. Aranya	Walairacht	Advisor
Asst. Prof. Dr. Chutimet	Srinilta	Co-Advisor

Academic Year 2014

Abstract

Data warehouse is a massive database system that keeps data from the past to present. The main objective is to process the stored raw data into the information that can summarize and analyze the outcome of the organization. It is so much important for the big organization that keeps massive loads of data, such as our institute's registrar. Contrarily, the registrar still doesn't manage to develop data warehouse of their own yet, and it cause the separation of the data localization, bring difficulty to the data analysis. Therefore, we had the opportunity to bring ourselves to develop a testing data warehouse system for the registrar so that can use as a base system for the future improvement for developing a practical data warehouse system to use in the real situation.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่นนี้ สามารถสำเร็จลุล่วงได้ด้วยความกรุณาของ ผศ.ดร.อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษาและ ผศ.ดร. ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษาร่วมของปริญญานิพนธ์ฉบับนี้ ท่านคอยให้คำแนะนาตรวจสอบแก้ไขข้อบกพร่องต่างๆรวมถึงช่วยในการวางแผนการดำเนินงาน ข้าพเจ้าขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยประสิทธิ์ประสาทวิชาความรู้ และอบรมสั่งสอนในเรื่องต่างๆ

ขอขอบคุณบิดา มารดา ครอบครัว และเพื่อนๆ ที่คอยเป็นกาลังใจ และให้การช่วยเหลือสนับสนุนข้าพเจ้าอย่างเต็มที่ตลอดมา

ชนัญญ บุญสิริสัมพันธ์

สรสิช ด่านสถาพร

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญรูป

รูป **หน้า**

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ปัจจุบันระบบคลังข้อมูลที่หลายๆองค์กรใช้อยู่นั้นได้ถูกพัฒนาและออกแบบมาให้เหมาะสมกับการใช้งานขององค์กรนั้นๆ แต่เนื่องจากเวลาผ่านไปเทคโนโลยีมีการเปลี่ยนแปลง อาจทำให้โครงสร้างขององค์กรนั้นๆอาจมีการเปลี่ยนแปลงไปไม่มากก็น้อย คลังข้อมูลที่ใช้อยู่อาจจะไม่สามารถรองรับการทำงานบางอย่างได้ จึงต้องมีการปรับเปลี่ยนและเพิ่มเติมเพื่อให้สามารถใช้งานในส่วนนั้นๆ และอาจเพิ่มขีดจำกัดให้แก่ระบบคลังข้อมูลนั้น จึงได้เกิดมาเป็นโครงงานนี้ขึ้นมาเพื่อพัฒนาเพิ่มเติมขีดความสามารถของระบบที่มีอยู่

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อสร้างระบบคลังข้อมูลให้กับระบบสำนักทะเบียนซึ่ง สามารถนำไปพัฒนาต่อจนใช้งานจริงได้

2) เพื่อจัดระเบียบให้กับข้อมูลต่างๆให้สามารถนำมาใช้วิเคราะห์ทางสถิติได้

3) เพื่อลดเวลาที่ใช้ในการค้นหาคำตอบจากฐานข้อมูลของสำนักทะเบียนได้

1.3 ขอบเขตของโครงงาน

1) พัฒนาโดยใช้โปรแกรม Pentaho ร่วมกับ MySQL เป็นหลัก

2) นำข้อมูลเข้ามาจาก MySQL server ที่ติดตั้งไว้บนเครื่อง

3) นำเสนอข้อมูลตามโจทย์ที่กำหนดไว้ได้

4) เปลี่ยนแปลงตัวแปรข้อมูลใหม่ที่จะใช้นำเสนอได้

1.4 วิธีการดำเนินการ

1) ศึกษาข้อมูลเกี่ยวกับ Data Warehouse

2) ศึกษากระบวนการการทำ Data Integration

3) ศึกษาและทำความเข้าใจโปรแกรม Open Source “Pentaho”

4) วิเคราะห์และออกแบบรายละเอียดสำหรับการทำ Schema

5) ศึกษาวิธีการสร้าง Cube สำหรับข้อมูลที่ต้องการวิเคราะห์

6) ศึกษาวิธีการเขียนภาษา XML เพื่อใช้ในการสร้าง Cube

7) วิเคราะห์ผลที่ได้จากโปรแกรมที่พัฒนาขึ้น และแก้ไขโปรแกรมจากผลการทดลองที่ผิดพลาด

8) จัดทำเอกสารและสร้างคู่มือการใช้งานของระบบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถนำมาพัฒนาให้เกิดความถูกต้องมากยิ่งขึ้น

2) สามารถนำระบบไปต่อยอดในการทำงานจริงได้

3) สามารถนำไปประยุกต์ใช้กับองค์กรอื่นๆได้

4) สามารถนำความรู้ในการออกแบบ Data Warehouse ไปใช้ในอนาคตได้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Data Warehouse

นิยามของ Data Warehouse ได้กล่าวไว้ว่า เป็นระบบคลังข้อมูลขนาดใหญ่ที่เอื้ออำนวยให้ผู้ใช้งานสามารถใช้ข้อมูลได้อย่างมีประสิทธิภาพ ข้อมูลในคลังข้อมูลต้องมีปริมาณเพียงพอ และเป็นข้อมูลที่มีคุณภาพต่อการนำไปวิเคราะห์เพื่อหาคำตอบที่เหมาะสม การจัดเก็บข้อมูลต้องเอื้ออำนวยให้สามารถใช้งานได้ง่าย สามารถวิเคราะห์ข้อมูลได้อย่างรวดเร็ว ไม่ยุ่งยากซับซ้อน โดยที่ Data Warehouse มีหัวใจหลักสำคัญที่มีความแตกต่างจากระบบสารสนเทศทั่วไป คือ คุณลักษณะของข้อมูล โดยแบ่งได้ดังนี้

2.1.1 Subject-oriented Data

เป็นการจัดโดยแบ่งตามเนื้อหาของข้อมูล โดยพิจารณาจากข้อมูลที่มีอยู่ว่าข้อมูลในระบบมีข้อมูลใดบ้างที่คล้ายคลึงกัน แล้วทำการจัดให้อยู่ในกลุ่มเดียวกัน

2.1.2 Integrated Data

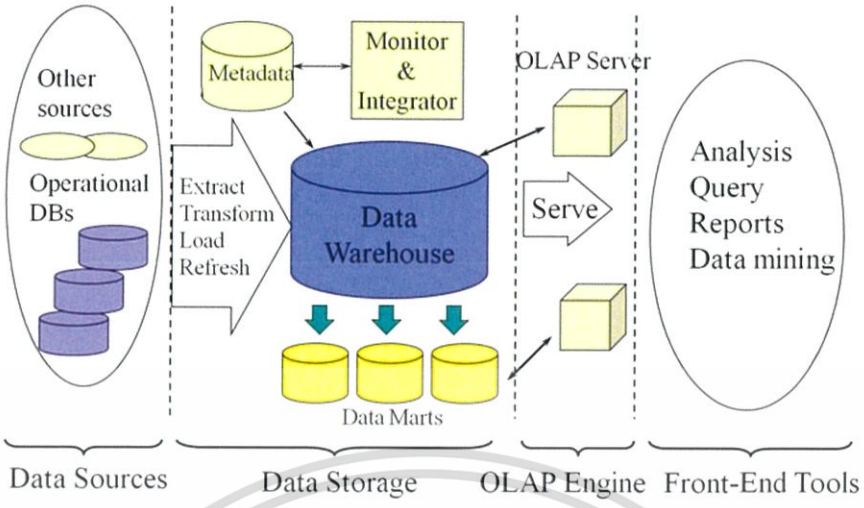
หน้าที่หนึ่งของคลังข้อมูลคือการกำจัดความซ้ำซากของข้อมูล หรือทำข้อมูลให้มีคุณสมบัติเป็น Integrated Data ทั้งนี้เพื่อไม่ให้เกิดความสับสนในการวิเคราะห์ข้อมูล หรือในบางครั้ง เราอาจจะต้องยอมให้เกิดการที่ข้อมูลซ้ำซ้อนเพื่อให้สะดวกแก่การแก้ไขข้อมูล แต่จะต้องไม่เกิดปัญหา Data inconsistency (ความไม่สอดคล้องกันของข้อมูล)

2.1.3 Time Referenced Data

ข้อมูลในคลังข้อมูลจะแตกต่างจากข้อมูลในระบบปฏิบัติการตรงที่ระบบปฏิบัติการจะสนใจข้อมูลในปัจจุบัน แต่คลังข้อมูลจะเน้นไปที่การเก็บข้อมูลเพื่อวิเคราะห์ข้อมูลตามช่วงเวลา (Time-Series Data Analysis)

2.1.4 Non-Volatile Data

เพื่อที่จะรักษาคุณสมบัติ Time Referenced Data ข้อมูลที่อยู่ในคลังข้อมูลจึงเป็นข้อมูลที่คงที่อยู่ตลอดไป ไม่ว่าจะเก่าแค่ไหน ก็จะยังอยู่ในคลังข้อมูล ไม่ถูกลบออก ทั้งนี้เพื่อให้การวิเคราะห์ข้อมูลแบบ Time-Series Data Analysis ให้ผลลัพธ์ที่มีประสิทธิภาพมาก แต่เพื่อการจัดการพื้นที่ในคลังข้อมูลให้มีประสิทธิภาพ Non-Volatile Data สามารถเปลี่ยนแปลงรูปแบบ เพื่อให้ข้อมูลมีขนาดเล็กลงได้ โดยเรียกกระบวนการนี้ว่า “Data Packing”



รูป 2.1 การทำงานของ Data Warehouse

2.2 ETL

ในการทำงานของคลังข้อมูลเราจะสามารถแบ่งการทำงานหลักได้เป็น 3 ขั้นตอนใหญ่ๆคือ

1) การได้มาซึ่งข้อมูล (Data acquisition)

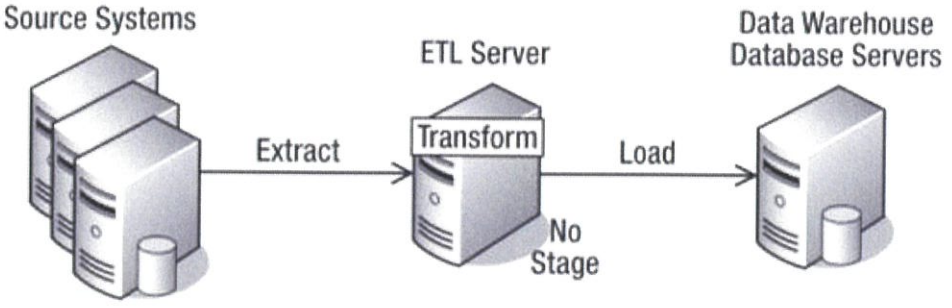
2) การจัดเก็บข้อมูล (Data storage)

3) การส่งต่อข้อมูลสารสนเทศ (Information Delivery)

จาก 3 การทำงานหลักของคลังข้อมูลที่ได้กล่าวมาข้างต้น ETL จะทำงานในส่วนของการได้มาซึ่งข้อมูลและการจัดเก็บข้อมูล ซึ่งเป็นการทำงานที่ไม่ได้มีการติดต่อกับผู้ใช้ ETL ย่อมาจาก Extract, Transform และ Load ซึ่งเป็นกระบวนการทำงาน 3 ขั้นต้น ทำงานเรียงต่อกัน การทำงานจะเริ่ม จากการค้นคืน (Retrieving/Extracting) และ เปลี่ยนแปลง/เปลี่ยนรูป (Transformation) ข้อมูลจาก ระบบการดำเนินงาน (Operational systems) หรือ แหล่งข้อมูลอื่นๆ (Source systems) จากนั้นนำเข้ามูลที่ได้ถ่ายโอน (โหลด: Load) เข้าสู่คลังข้อมูล

2.3 OLAP

OLAP เป็นเครื่องมือที่ใช้ในการวิเคราะห์ข้อมูลใน มุมมองหลากหลายมิติ (Multi-Dimensional) โดยที่ผู้ใช้สามารถที่จะ Drill Down ข้อมูลตามโครงสร้างของปัจจัย (Dimension) และยังสามารถที่จะทำการปรับเปลี่ยนมุมมองหรือ rotate ได้ตามต้องการ นอกจากนี้ OLAP Tools ยังสนับสนุนเครื่องมือในการคำนวณ และวิเคราะห์เข้าด้วย เช่น การพยากรณ์ข้อมูล (Forecasting) หรือการวิเคราะห์การถดถอยของข้อมูล (Regression) เป็นต้น



รูปที่ 2.2 กระบวนการทำ OLAP

OLAP จึงเป็นกระบวนการประมวลผลข้อมูลทางคอมพิวเตอร์ที่ช่วยให้ผู้ใช้สามารถวิเคราะห์ข้อมูลในมิติต่าง ๆ (Multidimensional Data Analysis) ได้ หลักการของ OLAP คือการ denormalize ข้อมูล ซึ่งต้องมีการสร้างโครงสร้างของข้อมูล 2 แบบที่นิยมกัน คือ star schema และ snowflake schema เพื่อใช้ทำ cube โดยข้อมูลที่เก็บอยู่ภายในลูกบาศก์ (Cube) จะถูก Consolidate และคำนวณทำให้เราสามารถมองภาพกลุ่มข้อมูลในแต่ละมุมมองได้

2.4 สถาปัตยกรรมของคลังข้อมูล

2.4.1 Centralized Architecture

เป็นรูปแบบที่ Data Warehouse Database นั้นถูกเก็บเป็นกลุ่มก้อนเดียวกัน แต่ไม่ได้หมายความว่า Data Warehouse Database จะถูกเก็บอยู่บน Hard Disk ตัวเดียวหรือหลายๆตัว แต่ดูจากจำนวนฐานข้อมูลที่ใช้สำหรับ Data Warehouse Database ซึ่งรูปแบบนี้จะมีฐานข้อมูลเพียงตัวเดียวที่ถูกใช้เป็น Data Warehouse Database

2.4.1.1 ข้อดีของ Centralized Architecture

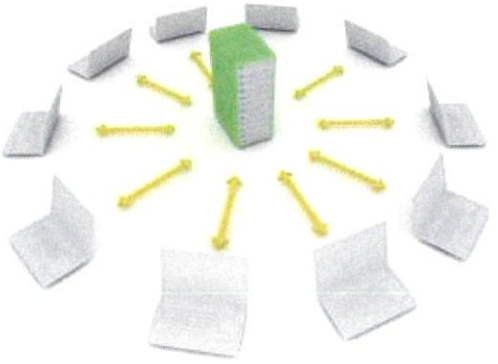
1) การรักษาความปลอดภัยและบำรุงรักษาง่าย

2) สามารถสร้างความเป็นปึกแผ่นของข้อมูลได้ง่ายที่สุด

2.4.1.2 ข้อเสียของ Centralized Architecture

1) มีความเสี่ยงในการสูญเสียข้อมูลมากกว่ารูปแบบอื่น

2) ออกแบบและพัฒนาได้ยากที่สุด



รูปที่ 2.3 โครงสร้าง Centralized Architecture

2.4.2 Distributed Architecture

เป็นรูปแบบที่ Data Warehouse Database มีการกระจายตัวออก โดยอาจจะกระจายอยู่บน Disk ตัวเดียวกันหรือคนละตัวก็ได้ แต่มีฐานข้อมูลมากกว่า 1 ตัวที่ใช้เป็น Data Warehouse Database ทั้งนี้ Data Warehouse Database จะถูกแยกออกเป็นหลายตัว เพื่อสร้างความคล่องตัวในการใช้งาน โดยในการกระจายออกของ Data Warehouse Database นั้น จะส่งผลให้ข้อมูลตัวใดตัวหนึ่งอาจจะมีอยู่ใน Data Warehouse Database เพียงตัวเดียวหรือหลายตัวก็ได้

2.4.2.1 ข้อดีของ Distributed Architecture

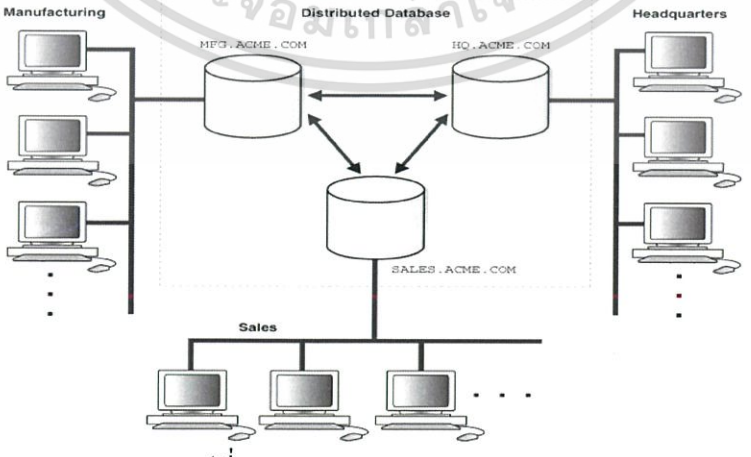
1) สร้างได้ง่ายกว่า Centralized Architecture

2) สามารถกระจายความเสี่ยงในกรณีที่อาจจะเกิดความเสียหายขึ้นกับข้อมูล

2.4.2.2 ข้อเสีย ของ Distributed Architecture

1) มีโอกาสที่ข้อมูลจะขาดความเป็นอันหนึ่งอันเดียวกัน

2) การรักษาความปลอดภัยทำได้ยากว่า Centralized Architecture



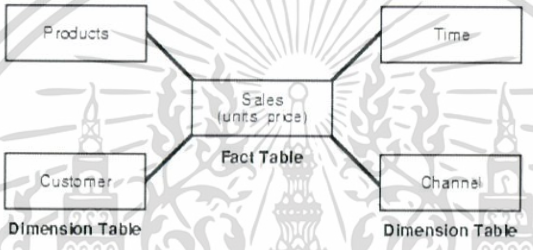
รูปที่ 2.4 Distributed Architecture

2.5 Schema

Schema คือโครงสร้างของฐานข้อมูลที่ประกอบด้วย object ของ database รวมถึงตาราง รูปแบบการจัดเก็บข้อมูล ซึ่งเราสามารถแบ่งได้เป็น 2 ประเภทด้วยกันได้แก่

2.5.1 Star Schema

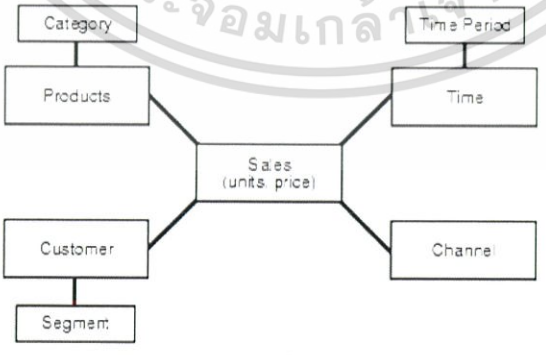
เป็น schema ที่เรียบง่ายและนิยมใช้กันมากที่สุดใน data warehouse โดยจะจัดเก็บโครงตารางในรูปแบบ fact (ข้อมูลหลัก) และ dimension (ข้อมูลที่ใช้อธิบาย fact นั้นๆ) โดยลักษณะของ star schema ตามหลักการ คือ จะให้ตาราง fact เป็นศูนย์กลางแล้วล้อมรอบด้วยตาราง dimension เมื่อทำเป็นไดอะแกรมออกมาจะมีรูปร่างหน้าตาเหมือนกับรูปดาว จึงได้ชื่อว่า star schema



รูปที่ 2.5 รูปแบบของ Star Schema

2.5.2 Snowflake Schema

เป็น schema ที่มีความซับซ้อนมากกว่า Star Schema โดยจะมี dimension เพิ่มมาจำนวนหนึ่งเพื่อกำหนดมุมมองของ dimension นั้นๆ โดยจำนวนมุมมองที่มองได้จะเท่ากับจำนวน dimension table ที่อยู่รอบๆ fact table



รูปที่ 2.6 รูปแบบของ Snowflake Schema

2.6 แนวทางในการพัฒนาคลังข้อมูล

แนวทางในการออกแบบและสร้างคลังข้อมูลนั้น เราสามารถแบ่งได้ 3 วิธีการด้วยกันดังต่อไปนี้

1.) Top – Down

เป็นหลักการที่เหมาะสมในการใช้พัฒนา Centralized Data Warehouse โดยหลักการนี้จะทำให้ได้ Data Model และคลังข้อมูลรวมของทั้งองค์กรในคราวเดียว โดยเริ่มจากการวิเคราะห์ธุรกิจและออกแบบ Data Model ที่เป็นภาพรวมของธุรกิจทั้งหมดขององค์กร แล้วจึงค่อยวิเคราะห์ถึง Input และ Output เพื่อสร้าง Data Mart ต่อไป

2.) Bottom – Up

จะตรงกันข้ามกับ Top – Down โดยเริ่มวิเคราะห์และออกแบบผลลัพธ์และข้อมูลนำเข้าทีละส่วนแล้วจึงออกแบบ Data Model ทีละส่วนไปพร้อมๆกับการออกแบบ Data Acquisition, Data Staging Area และ Data Mart ของข้อมูลส่วนย่อยๆ จากนั้นค่อยๆทำกระบวนการพัฒนาแบบเดียวกันนี้จนกว่าจะครบทุก Data Mart ที่จำเป็นต้องมี อาจมีการนำข้อมูลที่มีอยู่ในแต่ละ Data Mart มารวมกันเพื่อสร้าง Data Model และ Data Warehouse Database ของทั้งองค์กรในภายหลัง

3.) Mix Data

เป็นการพัฒนาโดยแยกข้อมูลออกเป็นส่วนๆ แล้วพิจารณาเลือกวิธีการวิเคราะห์และออกแบบ (Top – Down หรือ Bottom – Up)ที่เหมาะสมสำหรับข้อมูลแต่ละส่วน แล้วจึงนำเอาแต่ละส่วนที่ได้พัฒนามารวมกันในภายหลัง

2.7 พื้นฐานของภาษา XML

Xml เป็นภาษาที่ใช้เน้นส่วนที่เป็นข้อมูล โดยสามารถกำหนดชื่อแท็ก (Element) และชื่อแอตทิบิวต์ ได้ตามความต้องการของผู้สร้างเอกสาร xml โดยเอกสารนั้นจะต้องมีความเป็น Well-formed ส่วน DTD และ Schema จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับว่ามีผู้ใช้เอกสารนั้นมากน้อยแค่ไหน เอกสาร xml จึงเป็นแค่เท็กซ์ไฟล์ชนิดหนึ่ง ที่มีแท็กเปิดและเท็กปิดครอบข้อมูลไว้ตรงกลางเท่านั้น ทำให้เอกสาร xml ถูกใช้ในการติดต่อกับระบบที่ต่างกัน เนื่องจากความง่ายในการสร้างเอกสาร การนำเอกสาร xml ไปใช้งาน จะสนใจแต่ข้อมูลที่ถูกเน้นด้วยแท็กมากกว่า

Well-formed เป็นไวยากรณ์พื้นฐานของเอกสาร xml อย่างเช่น เอกสาร xml ต้องเริ่มต้นด้วย <?xml version="1.0" ?> เอกสาร xml 1 เอกสาร จะต้องมีแท็กรูทเพียงแท็กเดียว หมายความว่า แท็กและข้อมูลต่างๆ จะต้องอยู่ภายในแท็กแรกสุดเพียงแท็กเดียว การเปิดและปิดแท็กจะต้องไม่มีการคร่อมกัน เช่น ตัวหนาและเอียง จะไม่ Well-formed เนื่องจากเอกสาร xml สามารถกำหนดชื่อแท็ก และชื่อแอตทิบิวต์ได้ตามความต้องการของผู้สร้างเอกสาร ทำให้ในการเน้นข้อมูลใดข้อมูลหนึ่ง สามารถมีเอกสาร xml หลายรูปแบบ (ผู้เขียนอาจใช้ชื่อแท็กต่างกัน ทั้งที่สื่อ

ความหมายไปที่สิ่งเดียวกัน) หากว่าเอกสาร xml นั้น ถูกนำไปใช้ติดต่อกับระบบอื่นๆ อาจทำให้สื่อความหมายไม่ตรงกัน ดังนั้นจึงต้องมีการกำหนดรูปแบบที่เป็นมาตรฐานขึ้น (ตกลงรูปแบบระหว่างกัน) โดย DTD และ Schema จะเป็นตัวกำหนดว่าเอกสาร xml นั้น จะต้องมีแท็กอะไรบ้าง ภายในแท็กนั้นจะมีแท็ก แอตทิบิวต์ หรือข้อมูลอะไรได้บ้าง โดย DTD จะต่างกับ Schema ตรงที่ Schema เป็นเอกสาร xml ด้วย

2.6.1 Element

ประกอบไปด้วย แท็กเปิด ข้อมูล และแท็กปิด ยกตัวอย่างเช่น

```
<student> Example_sudent </student>
```

ในที่นี้ <student> คือแท็กเปิด Example_sudent คือข้อมูล และ </student> คือแท็กปิด โดยแท็กปิดนั้นจะต้องมีชื่อเหมือนแท็กเปิดของมันแต่ตามหลังจากเครื่องหมาย '/' จะสังเกตได้ว่า XML นั้นคล้ายกับ HTML เป็นอย่างมากสำหรับข้อแตกต่างที่ชัดเจนคือ HTML ได้กำหนดแท็กไว้ล่วงหน้าแล้วแต่ XML ไม่ว่าใครๆก็สามารถกำหนดแท็กของเราเองได้ XML นั้นไม่ใช่ภาษาโดยสมบูรณ์มันเป็น*มาตรฐาน*ข้อมูลมากกว่า โดยตัวโปรแกรมประยุกต์จะเป็นผู้กำหนดรูปแบบของตัวเองขึ้นและจะสามารถใช้ได้ กับโครงสร้างข้อมูลที่ถูกอนุญาต (เพราะว่ามีรูปแบบของข้อมูลที่เข้ากันได้) XML นั้นเป็นภาษาที่ case sensitive ดังนั้นการที่เราเขียนว่า <student> กับ <Student> จึงถือว่าเป็นคนละแท็กกัน นอกจากนี้แล้ว element ใน XML สามารถบรรจุอยู่ใน element อื่นๆได้ ยกตัวอย่างเช่น

```
<student>
        <name>example name</name>
        <id>123456789</id>
</student>
```

จะเห็นว่า element <name> บรรจุอยู่ภายใน element <student> element ไม่สามารถคร่อมกันได้ เช่น

```
<student>
        <name>example name
        <id></name>123456789</id>
</student>
```

แบบนี้ถือว่าไม่ถูกต้อง แต่สามารถมี element ว่างแบบนี้ได้

```
<book></book>
```

2.6.2 Attribute

นอกจากแท็กแล้วยังมี สิ่งที่เรียกว่า attribute ด้วยซึ่งจะเป็นค่าที่อยู่ใน Element โดยที่มีรูปแบบดังนี้

```
<student name="example_name"></student>
<student name='example_name'></student>
```

2.6.3 โครงสร้างของเอกสาร XML

ถูกกำหนดขึ้นโดยลำดับชั้น โดยเอกสารใดๆนั้นต้องมี root element หนึ่งตัวเสมอ เช่นในที่นี้คือ <student> ดังตัวอย่าง

```
<?xml version="1.0" ?>
        <student>
                <name>example name</name>
                <id>123456789</id>
        </student>
```

บทที่ 3
การออกแบบและพัฒนา

3.1 ภาพรวมของการทำงาน

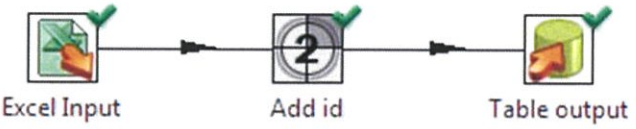
ในส่วนของการออกแบบระบบคลังข้อมูล ส่วนสำคัญคือการวิเคราะห์ข้อมูลที่มีอยู่แล้วภายในฐานข้อมูลของสำนักทะเบียน โดยดูว่าข้อมูลที่ถูกเก็บอยู่ในสำนักทะเบียนนั้นถูกเก็บอยู่ในรูปแบบใด โดยเราจะสังเกตจากผลลัพธ์สุดท้ายที่ทางฝั่งสำนักทะเบียนต้องการจะให้ออกมา ซึ่งก็คือรีพอร์ตที่ทางสำนักทะเบียนต้องการให้สามารถทำออกมาจากคลังข้อมูลได้ โดยเมื่อทำการสำรวจดูว่าภายในฐานข้อมูล มีข้อมูลอะไรที่ทางสำนักทะเบียนต้องการให้นำมาใช้ในระบบคลังข้อมูลบ้าง ก็จะทำการดึงข้อมูลทั้งหมดนั้นออกมา และจากนั้นก็จะทำการออกแบบรูปแบบที่เราจะใช้เก็บข้อมูลเหล่านั้นลงภายในคลังข้อมูล การดึงข้อมูลออกมาจากฐานข้อมูลของสำนักทะเบียนนั้น จะทำการดึงข้อมูลออกมาเป็นข้อมูลดิบทั้งชุดจากนั้นจะนำข้อมูลดิบเหล่านั้นไปจัดรูปแบบและจำลองไว้บนฐานข้อมูลใน MySQL เพื่อเป็นการจำลองว่าได้มีการเชื่อมต่อกับเซิร์ฟเวอร์ภายนอกไม่ใช่การสร้างตารางใหม่หรือเป็นเพียงการ import ตารางเข้ามาประมวลผล โดยการทำงานหลักๆจะอยู่ในส่วนของการทำการทดลองด้าน data integration ซึ่งก็คือการนำข้อมูลมาประมวลผลหรือทำการเปลี่ยนแปลงให้อยู่ในรูปแบบของข้อมูลที่เราต้องการ โดยจะใช้โปรแกรม Pentaho Data Integration (kettle) ในการทดลองการจัดการเปลี่ยนแปลงรูปแบบข้อมูล

รูปที่ 3.1 การทำงานของ Pentaho ใน Data Integration

3.2 หลักการทำงาน

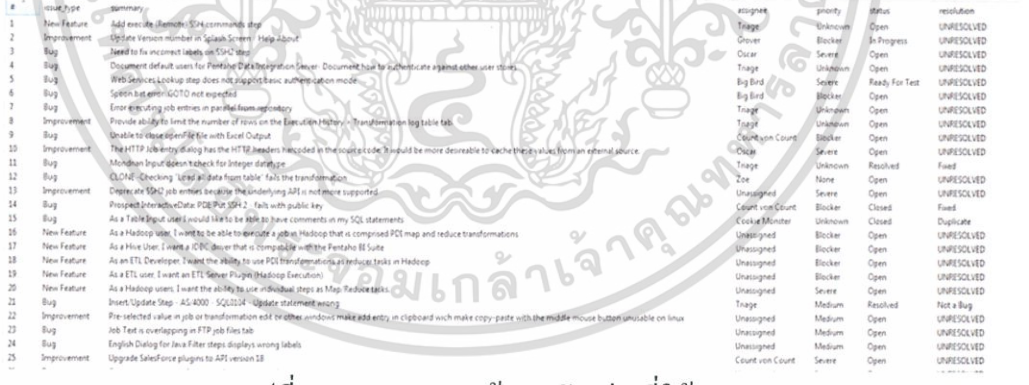
การทำงานจะออกแบบให้ตัวโปรแกรม data integration ทำการดึงข้อมูลจากตัว MySQL server ที่จำลองไว้ จากนั้นจะทำการประมวลข้อมูลที่ดึงมา เปลี่ยนรูปแบบการจัดเก็บให้เป็นเหมือนที่เราทำการออกแบบและกำหนดไว้ แล้วทำการจัดเก็บลงในฐานข้อมูลใหม่โดยจะให้ฐานข้อมูลนั้นเป็น

เสมือนคลังข้อมูล ซึ่งก็คือการทำ ETL (Extract-Transformation-Load) จาก data source ไปสู่ data warehouse โดยจัดเก็บลงใน MySQL server เช่นเดียวกัน



รูปที่ 3.2 ตัวอย่างการทำงานของขั้นตอนการนำข้อมูลเข้า Data Integration

โดยขั้นตอนที่ทำภายในโปรแกรม data integration นั้นทำการดึงข้อมูลมาจากไฟล์ excel แล้วทำการใส่ id ของแต่ละ fact ลงไป จากนั้นทำการ export ไปยังตารางใหม่บน MySQL จากนั้นทำการเชื่อมต่อกับตัว BI server โดยหากจะทำการบอกว่าตารางที่เราทำนั้นจะทำให้เป็น cube (สร้างเพื่อสำหรับการวิเคราะห์ผล ซึ่งจะสามารถทำการวิเคราะห์ผลโดยการแสดงออกเป็นรีพอร์ตหรือ dashboard ได้ โดยจะวิเคราะห์โดยใช้ค่าอะไรเป็นแกนของ cube ก็ได้) จะต้องทำการสร้างไฟล์ xml เพื่อเป็นการกำหนดว่าค่าของคอลัมน์ใดบ้างที่เราจะสามารถให้เป็นแกนในการวิเคราะห์ของ cube ได้ ทั้งนี้ตัว BI server นี้มีตัว server ที่ทำการประมวลผล OLAP cube ไว้ในตัวอยู่แล้ว ซึ่งอยู่ใน pentaho อยู่แล้วคือ Mondrian



#	issue_type	summary	assignee	priority	status	resolution
1	New Feature	Add execute /remote/ SQL commands step	Tnage	Unknown	Open	UNRESOLVED
2	Improvement	Update Version number in Splash Screen - Help About	Grover	Blocker	In Progress	UNRESOLVED
3	Bug	Need to fix incorrect labels on SSH2 step	Oscar	Severe	Open	UNRESOLVED
4	Bug	Document default users for Pentaho Data Integration Server. Document how to authenticate against other user stores.	Tnage	Unknown	Open	UNRESOLVED
5	Bug	Web Services Lookup step does not support basic authentication mode	Big Bird	Severe	Ready For Test	UNRESOLVED
6	Bug	Spoon bot error: IGOTO not expected.	Big Bird	Blocker	Open	UNRESOLVED
7	Bug	Error executing job entries in parallel from repository	Tnage	Unknown	Open	UNRESOLVED
8	Improvement	Provide ability to limit the number of rows on the Execution History - Transformation log table tab	Tnage	Unknown	Open	UNRESOLVED
9	Bug	Unable to close openFile file with Excel Output	Count von Count	Blocker	Open	UNRESOLVED
10	Improvement	The HTTP Job entry dialog has the HTTP headers hardcoded in the source code. It would be more desireable to cache these values from an external source.	Oscar	Severe	Open	UNRESOLVED
11	Bug	Mondrian Input doesn't check for Integer data type	Tnage	Unknown	Resolved	Fixed
12	Bug	CLONE - Checking 'Lead all data from table' fails the transformation	Zoe	None	Open	UNRESOLVED
13	Improvement	Deprecate SSH2 job entries because the underlying API is not more supported	Unassigned	Severe	Open	UNRESOLVED
14	Bug	Prospect InteractiveData: PDI Put SSH2 - fails with public key	Count von Count	Blocker	Closed	Fixed
15	Bug	As a Table Input user I would like to be able to have comments in my SQL statements	Cookie Monster	Unknown	Closed	Duplicate
16	New Feature	As a Hadoop user, I want to be able to execute a job on Hadoop that is comprised PDI map and reduce transformations	Unassigned	Blocker	Open	UNRESOLVED
17	New Feature	As a Hive User, I want a JDBC driver that is compatible with the Pentaho BI Suite	Unassigned	Blocker	Open	UNRESOLVED
18	New Feature	As an ETL Developer, I want the ability to use PDI transformations as reducer tasks in Hadoop	Unassigned	Blocker	Open	UNRESOLVED
19	New Feature	As a ETL user, I want an ETL Server Plugin (Hadoop Execution)	Unassigned	Blocker	Open	UNRESOLVED
20	New Feature	As a Hadoop users, I want the ability to use individual steps as Map Reduce tasks	Unassigned	Severe	Open	UNRESOLVED
21	Bug	Insert/Update Step - AS/400 - SQL0104 - Update statement wrong	Tnage	Medium	Resolved	Not a Bug
22	Improvement	Pre-selected value in job or transformation edit or other windows make add entry, in clipboard wich make copy-paste with the middle mouse button unusable on linux	Unassigned	Medium	Open	UNRESOLVED
23	Bug	Job Text is overlapping in FTP job files tab	Unassigned	Medium	Open	UNRESOLVED
24	Bug	English Dialog for Java Filter steps displays wrong labels	Unassigned	Medium	Open	UNRESOLVED
25	Improvement	Upgrade SalesForce plugins to API version 18	Count von Count	Severe	Open	UNRESOLVED

รูปที่ 3.3 แสดงตารางข้อมูลตัวอย่างที่ใช้ทดลอง

โปรแกรม 3.1 ตัวอย่างของXMLที่ใช้ในการกำกับตาราง

```
<?xml version="1.0"?>
<Schema name="IssueSchema">
     <Cube name="Issue">
          <Table name="fact_issue"/>
          <Dimension name="Type">
               <Hierarchy hasAll="true"
allMemberName="All Types">
                    <Level name="Type"
column="issue_type" uniqueMembers="true"/>
               </Hierarchy>
          </Dimension>
          <Dimension name="Assignee">
               <Hierarchy hasAll="true"
allMemberName="All Assignees">
                    <Level name="Assignee"
column="assignee" uniqueMembers="true"/>
               </Hierarchy>
          </Dimension>
          <Dimension name="Priority">
          <Hierarchy  hasAll="true"  allMemberName="All
Priorities">
                         <Level          name="Priority"
column="priority" uniqueMembers="true"/>
                    </Hierarchy>
                </Dimension>
     <Dimension name="Status">
                    <Hierarchy          hasAll="true"
allMemberName="All Status">
                         <Level          name="Status"
column="status" uniqueMembers="true"/>
                    </Hierarchy>
               </Dimension>
               <Dimension name="Resolution">
                    <Hierarchy          hasAll="true"
allMemberName="All Resolution">
                         <Level      name="Resolution"
column="resolution" uniqueMembers="true"/>
                    </Hierarchy>
               </Dimension>
               <Measure name="Issue Count" column="id"
aggregator="count" formatString="Standard"/>
          </Cube>
</Schema>
```

จากโค้ด xml ข้างต้น เราได้กำหนดให้ค่าที่จะเก็บภายใน cube คือจำนวนของ fact ตามแกนที่กำหนด และในแต่ละ dimension นั้นจะมีคอลัมน์เพียงคอลัมน์เดียว ไม่ใช่ตารางที่มีหลายคอลัมน์ เมื่อทำทุกอย่างเสร็จสิ้น เราจะทำการสร้างรีพอร์ตหรือ dashboard จากบนตัว BI server ซึ่งโปรแกรมที่จะทำการสร้างนั้นมีอยู่หลายโปรแกรม ทั้งทางของ pentaho เองหรือจะเป็น third-party plugins ก็ตาม โดยในที่นี้เราเลือกใช้ third-party plugin ของทาง saiku เนื่องจากการจัดการวิเคราะห์ cube นั้นตัว plugin ของ saiku จะทำงานได้ดีกว่าของทาง pentaho ที่ยังมีปัญหาอยู่บ้างเล็กน้อย โดยในรูปที่ 3.4 และ รูปที่ 3.5 จะเป็นตัวอย่างการออกรีพอร์ตจาก plugin ของทาง saiku ในที่นี้กำหนดให้แกน column เป็น type ของ issue แกน row เป็น assignee ของแต่ละ issue

รูป 3.4 แสดงผลการวิเคราะห์ในรูปแบบกราฟ

Columns Type

Rows Assignee

Filter

Assignee	Bug	Improvement	New Feature	Task
Abby Cadabby	5	1		1
Bert	17			
Big Bird	328	22	1	2
Cookie Monster	79	8		
Count von Count	42	5	2	
Elmo	11	3	11	
Ernie	19			
Grover	12	2		
Oscar	26	3		1
Rosita	31	4	1	
Telly Monster	14	1		
Triage	51	11	10	3
Unassigned	183	51	24	
Zoe	13	1	1	

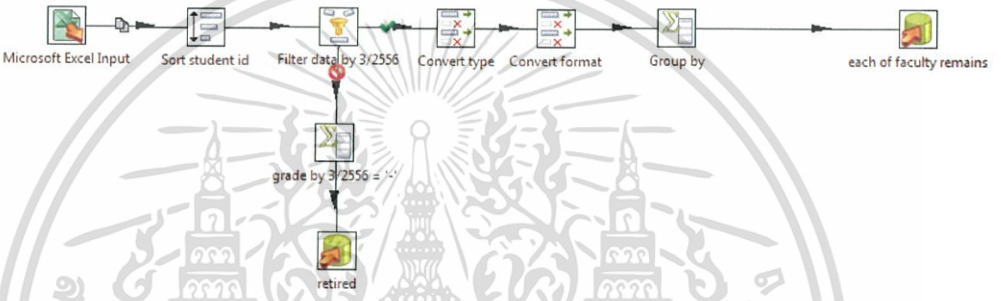
รูปภาพที่ 3.5 แสดงผลการวิเคราะห์ในรูปแบบตาราง

บทที่ 4

การทดลองและผลการทดลอง

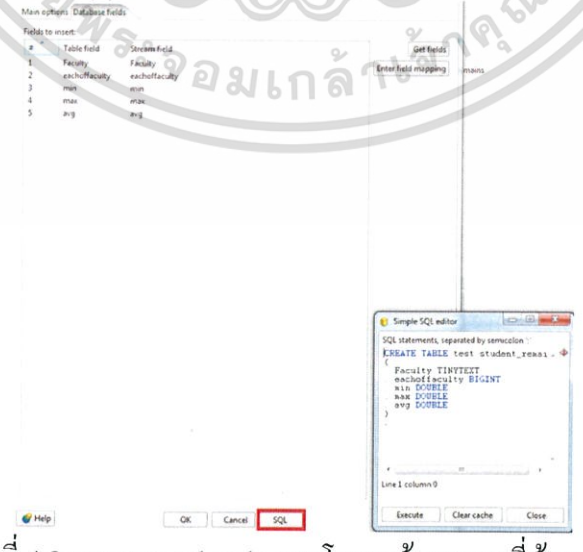
4.1 การทดลองประมวลข้อมูลด้วยโปรแกรม

ทดลองทำการนำข้อมูลเข้าไปทำการ integration ด้วยโปรแกรม Pentaho data integration โดยทดลองนำข้อมูลจากตารางเข้าฐานข้อมูลและทดลองทำการจัดข้อมูลให้อยู่ในรูปแบบตามที่ต้องการโดยแบ่งออกเป็นสองตาราง



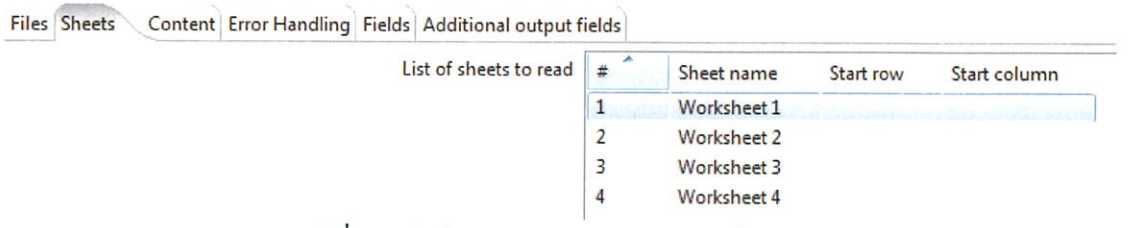
รูปที่ 4.1 นำข้อมูลมาจัดให้อยู่ในรูปแบบตามที่ต้องการ

โดยตารางที่เราต้องการจัดเก็บผลลัพธ์ที่ได้นั้น จะเป็นตารางใหม่ซึ่งยังไม่มีในฐานข้อมูล จึงต้องทำการสร้างตารางก่อน โดยที่ตัวโปรแกรมนี้สามารถจัดการสร้างตารางในฐานข้อมูลได้ โดยจะสร้างโค้ด sql ขึ้นตามที่เรากำหนดให้



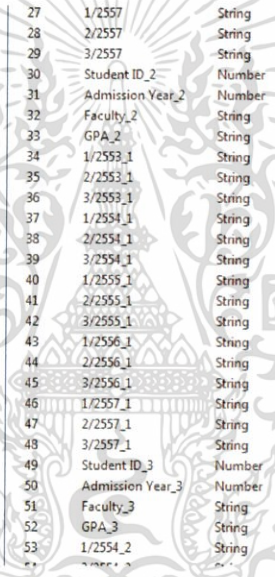
รูปที่ 4.2 generate sql code ตามโครงสร้างตารางที่ต้องการ

4.2 การนำข้อมูลหลาย spreadsheet จากไฟล์ excel เข้า database



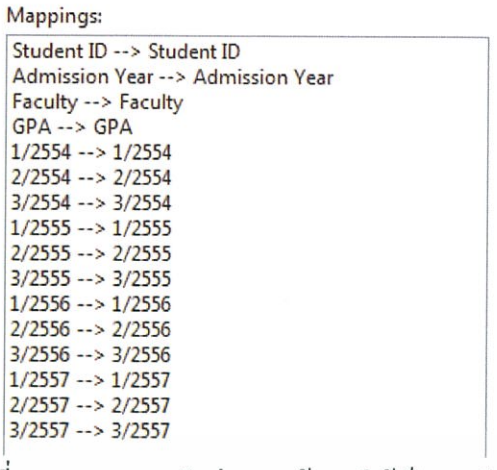
รูปที่ 4.3 นำข้อมูลหลาย spreadsheet เข้าตาราง

แม้ชื่อคอลัมน์จะซื้อกันในแต่ละ spreadsheet แต่เมื่อโปรแกรมทำการอ่านแล้ว จะเสมือนว่าคอลัมน์ของแต่ละ spreadsheet นั้นเป็นคอลัมน์ต่างกัน โดยโปรแกรมจะเพิ่ม _n ต่อท้ายคอลัมน์ที่มีชื่อซ้ำ



รูป 4.4 แสดงการเก็บ field ของตัวโปรแกรมโดยไม่มีการจัดการใดๆ

โดยที่จะต้องทำการจับคู่คอลัมน์ input และ output โดยโปรแกรมนั้นจะมีฟังก์ชั่น guess หรือตัวช่วยคาดเดาอัตโนมัติให้ใช้งาน แต่โปรแกรมจะไม่สามารถเดาให้คอลัมน์ที่ชื่อเดียวกันแต่อยู่ต่าง spreadsheet กันมาอยู่รวมกันได้

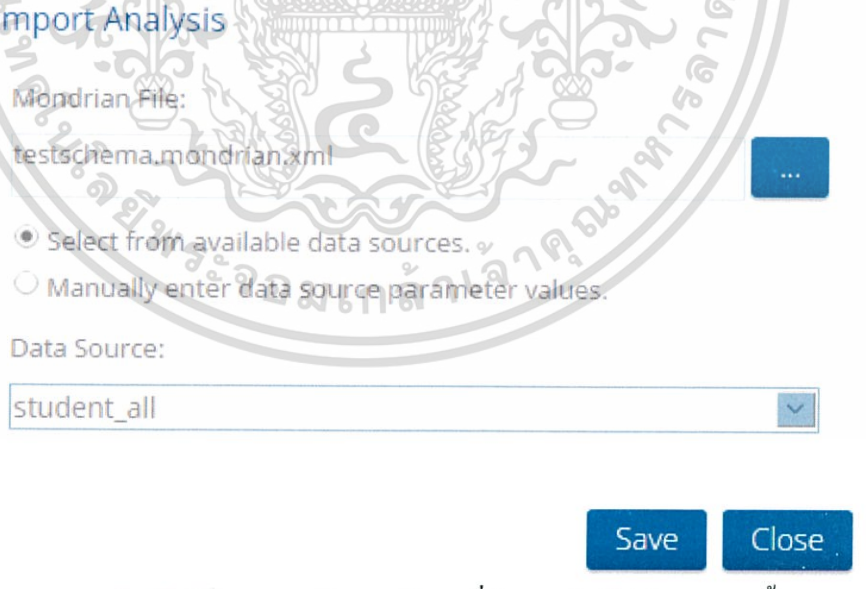


รูปที่ 4.5 แสดงการจับคู่ field ข้อมูลให้เป็นระเบียบ

4.3 การทดลองประมวลผลการ query จาก OLAP cube

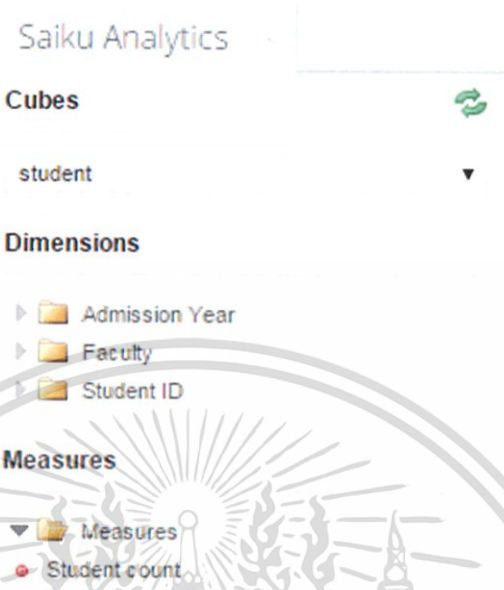
ทดลองสร้าง schema ของ OLAP cube ให้กับ database ตามที่กำหนดไว้ โดยการสร้าง OLAP cube นั้น จะใช้งานตัว Mondrian ซึ่งเป็นตัวจัดการ OLAP cube ซึ่งมีอยู่ใน Pentaho BI server อยู่แล้ว

จากนั้น ทำการรัน Pentaho BI server แล้วจึงทำการกำหนดไฟล์ schema ว่าจะใช้เป็นโครงร่างของฐานข้อมูลไหน



รูป 4.6 แสดงการจับคู่ไฟล์ schema กับฐานข้อมูลที่กำหนดในไฟล์ schema นั้นๆ

จากนั้นทดลองสร้างไฟล์ saiku analytic ขึ้นมาเพื่อทดสอบว่าไฟล์ schema ที่เราทำการกำหนดลงไปนั้นสามารถใช้งานได้หรือไม่



รูป 4.7 แสดงโครงร่างของ OLAP cube ที่ได้กำหนดไว้ใน schema

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

การทำระบบคลังข้อมูลจำเป็นต่อการจัดการข้อมูลปริมาณมากขององค์กรใดองค์กรหนึ่ง เพื่อให้การจัดการข้อมูล ทั้งในเรื่องของการจัดเก็บข้อมูล รวบรวมและประมวลข้อมูล ทำได้อย่างมีประสิทธิภาพและสะดวกมากยิ่งขึ้น ด้วยเหตุดังนี้ จึงได้มีโครงการจัดทำระบบคลังข้อมูลให้กับทางสำนักทะเบียนของสถาบัน โดยได้จัดทำกรณีศึกษาเพื่อประเมินความสามารถของเครื่องมือที่ได้นำมาใช้เพื่อทดสอบประสิทธิภาพของเครื่องมือว่าสามารถจัดการข้อมูลและแสดงผลได้ตามที่ต้องการหรือไม่ โดยได้แบ่งประเภทเครื่องมือที่ศึกษาออกเป็นสองประเภท คือเครื่องมือที่ทำหน้าที่ในส่วนของการนำข้อมูลเข้ามารวบรวมและจัดการข้อมูลดิบที่ได้มา นั่นคือโปรแกรม Pentaho Data Integration และอีกประเภทหนึ่งคือเครื่องมือที่จัดการการประมวลผลข้อมูลแล้วนำไปแสดงผลตามรูปแบบข้อมูลที่ต้องการ ซึ่งคือโปรแกรม Pentaho BI Server โดยให้ตัวโปรแกรมจัดการข้อมูลดิบที่ได้มาแปลงข้อมูลเป็นข้อมูลที่ต้องการ จากนั้นจึงส่งข้อมูลไปที่เซิร์ฟเวอร์จำลองภายในเครื่อง แล้วให้ตัว Pentaho BI Server ดึงข้อมูลจากเซิร์ฟเวอร์ภายในเครื่องไปเข้ากระบวนการการนำข้อมูลไปสร้าง OLAP cube เมื่อได้ OLAP cube ตามที่ต้องการแล้วจึงสั่งให้โปรแกรมแสดงผลออกมาตามรูปแบบที่เรากำหนด

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

1) ต้องใช้เวลาในการศึกษาการทำงานของโปรแกรม เนื่องจากเป็นโปรแกรมที่แพร่หลายในวงแคบเฉพาะกลุ่มผู้ทำงานเฉพาะด้านนี้เท่านั้น จึงทำให้เมื่อเกิดปัญหาการทำงานของโปรแกรมนั้นจะหาวิธีแก้ไขได้ยาก จึงต้องใช้เวลาในการศึกษาตัวโปรแกรมพอสมควร

2) ต้องใช้เวลาในการศึกษาทฤษฎีของระบบคลังข้อมูลพอสมควร เนื่องจากเป็นเรื่องใหม่ที่ผู้จัดทำยังไม่เคยทราบมาก่อน

3) โปรแกรมหลายๆตัวที่ใช้งานร่วมกันเพื่อให้ตัวงานสามารถทำออกมาได้นั้น มีข้อจำกัดในหลายๆส่วนทำให้เมื่อนำมาใช้ร่วมกันแล้วเกิดปัญหาหรือไม่สามารถใช้งานได้ จึงต้องศึกษานอกเหนือจากที่ทางด้านผู้พัฒนาระบุถึงความต้องการของโปรแกรม

5.3 แนวทางในการพัฒนาต่อ

1) ออกแบบระบบคลังข้อมูลใหม่ให้เหมาะสมกับระบบของทางสำนักทะเบียนให้มากยิ่งขึ้น

2) ปรับเปลี่ยนโปรแกรมที่ใช้ในการทำ Data warehouse เป็นชนิดอื่น โดยทำการเชื่อมต่อจากฐานข้อมูลเดิม

3) ออกแบบส่วนของ Data Integration ให้มีประสิทธิภาพมากยิ่งขึ้น

บรรณานุกรม

Ralph Kimball and Margy Ross. 2013. **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition**. Canada. John Wiley and Sons

María Carina Roldán.2013. **Pentaho Data Integration Beginner's Guide Second Edition**. Birmingham. Packt Publishing

Alex Meadows. Adrián Sergio Pulvirenti and María Carina Roldán. **Pentaho Data Integration Cookbook Second Edition**. Birmingham. Packt Publishing.