



รายงานสหกิจศึกษาฉบับสมบูรณ์

การคัดเลือกข้อมูลเบื้องต้นของลูกค้า สำหรับการกู้ยืมเงินผ่านระบบดิจิทัล
Customer Foundation for Digital Lending

นางสาวกิตติมา กาวิละ

สาขาวิชาวิศวกรรมสารสนเทศ
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา การคัดเลือกข้อมูลเบื้องต้นของลูกค้า สำหรับการกู้ยืมเงินผ่านระบบดิจิทัล
ชื่อ-สกุลนักศึกษา นางสาวกิตติมา กาวิละ
คณะ วิศวกรรมศาสตร์
ภาควิชา วิศวกรรมคอมพิวเตอร์
สาขาวิชา วิศวกรรมสารสนเทศ
ชื่อ-สกุล อาจารย์วิเทศ รศ.ดร.ชวลิต เบญจางคประเสริฐ
ชื่อ-สกุล ผู้นิเทศงาน คุณมนตร์วี เชียรชัยนริตชัย
ชื่อสถานประกอบการ บริษัททสิกร เทคโนโลยี กรู๊ป เซเครเทรียต จำกัด

บทคัดย่อ

การให้ยืมแบบดิจิทัลเป็นกระบวนการของการเสนอสินเชื่อที่ใช้สำหรับการให้เงิน และจัดการผ่านช่องทางดิจิทัล ซึ่งผู้ให้กู้จะใช้ข้อมูลดิจิทัลเพื่อแจ้งการตัดสินใจในการเรื่องของสินเชื่อ และยังสร้างข้อตกลงของลูกค้าที่มีความละเอียดรวดเร็ว

การกู้ยืมเงินผ่านระบบดิจิทัล เป็นการให้กู้ยืมเงินที่ทำให้เข้าถึงลูกค้าที่ต้องการสินเชื่อได้อย่างสะดวก ง่ายตาย และมีความโปร่งใสมากขึ้น เนื่องจากธนาคารจะทำการเชื่อมต่อ และดึงข้อมูลของลูกค้ามาจากหลากหลายส่วน เพื่อยืนยันตัวตนของลูกค้าทางดิจิทัลได้ ทำให้สามารถคัดกรองลูกค้าได้อย่างมีประสิทธิภาพมากขึ้น

Co-operation Title: Customer Foundation for Digital Lending
Student intern Name: Kittima Kawila
Faculty: Engineering
Department: Computer Engineering
Program: Information Engineering
Advise Name: Assoc.Prof.Dr. Chawalit Benjangkprasert
Mentor Name: Monrawee Chainchainirattisai
Company: Kasikorn Business Group Secretariat Co., Ltd

ABSTRACT

Digital lending is the process of offering loans that are applied for, disbursed, and managed through digital channels, in which lenders use digitized data to inform credit decisions and build intelligent customer engagement.

It is a loan that makes it easy to reach customers who want to make loans with greater transparency, because the bank will connect and retrieve customer information from many parts to verify the identity of the customer digitally. Makes it able to screen customers more efficiently.

กิตติกรรมประกาศ

รายงานสหกิจฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์ด้วยความกรุณา ตลอดจนตรวจทานแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่เป็นอย่างยิ่ง จาก รศ.ดร.ชวลิต เบญจางคประเสริฐ อาจารย์ที่ปรึกษา ที่ได้สละเวลาอันมีค่าแก่ข้าพเจ้า เพื่อให้คำปรึกษาและคำแนะนำ ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณ คุณเบญจวรรณ แก้วศรี ที่คอยดูแล ให้คำปรึกษา ตลอดจนแนะนำในการแก้ปัญหาจนเกิดมาเป็นชิ้นงานนี้

ขอขอบพระคุณ คุณวิลาสินี ศรีมณี ที่คอยให้คำปรึกษาและคอยสอนอย่างใจเย็น รวมถึงให้ข้อมูลอันเป็นประโยชน์ต่อชิ้นงานโดยตลอดจนสำเร็จลุล่วง

ขอขอบพระคุณ คุณน้ำเพชร สุขเรื่อนสุวรรณ ที่คอยสอน ดูแล และช่วยเหลือในการทำงานทางด้านเทคนิคต่าง ๆ จนสำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณ คุณอากาศรี เรื่องเทพ ที่ดูแลข้าพเจ้าในเรื่องของการพัฒนาที่เก็บข้อมูล ช่วยสอนในเรื่องของโค้ดดิ้ง ทำให้งานสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ คุณกิตติพงศ์ ประพันธ์สิริ ที่คอยดูแล และดำเนินการเกี่ยวกับการฝึกงาน ตลอดจนสหกิจที่ข้าพเจ้าปฏิบัติงานที่บริษัทเคบีทีจีมาโดยตลอด

สุดท้ายนี้ ข้าพเจ้าขอขอบพระคุณบิดา มารดา และครอบครัวของข้าพเจ้า ที่คอยช่วยเหลือ และให้กำลังใจแก่ข้าพเจ้าเสมอมาตลอดจนสำเร็จการศึกษา

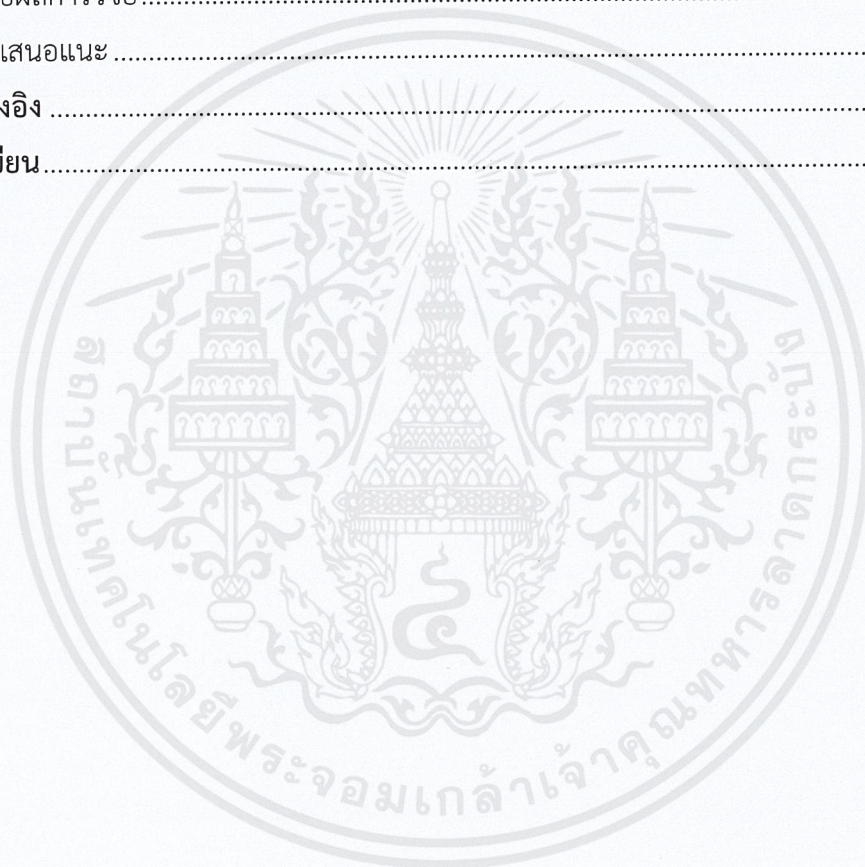
กิตติมา กาวิละ

สารบัญ

หน้า

บทคัดย่อ.....	I
ABSTRACT	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูปภาพ.....	VI
สารบัญตาราง.....	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของการวิจัย	3
1.3 ขอบเขตของการวิจัย.....	3
1.4 วิธีการดำเนินการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	5
1.5.1 ประโยชน์ต่อตนเอง.....	5
1.5.2 ประโยชน์ต่อธนาคาร	5
1.5.3 ประโยชน์ต่อลูกค้า.....	5
บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง	6
2.1 กระบวนการที่เกี่ยวข้องกับการทำงาน	6
2.1.1 อีทีแอล (ETL : Extract-Transform-Load).....	6
2.2 ภาษาที่ใช้สำหรับเขียนโปรแกรม	8
2.2.1 ภาษาเอสคิวแอล (SQL : Structured Query Language)	8
บทที่ 3 วิธีการดำเนินการวิจัย	37
3.1 ศึกษาข้อมูล และวิธีการใช้งานโปรแกรมเพื่อสร้างที่เก็บข้อมูล.....	37
3.2 สร้างเอกสารเพื่อใช้เป็นคู่มือในการพัฒนาที่เก็บข้อมูล.....	37
3.3 การสร้างเอกสารดีดีแอล	40
3.4 การกำหนดไฟล์ข้อมูลให้ตรงตามรูปแบบที่กำหนด	41
3.5 สร้างที่เก็บข้อมูลด้วยโปรแกรมตาต้า สแดง	41

3.6 การทดสอบความเท่ากันของข้อมูลในขาเข้าและขาออก	46
3.7 สร้างเอกสารยูไอเอส	47
3.8 เตรียมข้อมูลขึ้นระบบจริง	50
บทที่ 4 ผลการวิจัย	51
4.1 เครดิต ทีสคอร์ด (Credit T Score)	51
4.2 นอน เครดิต ทีสคอร์ด (Non Credit T Score).....	51
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	53
5.1 สรุปผลการวิจัย.....	53
5.2 ข้อเสนอแนะ	53
เอกสารอ้างอิง	55
ประวัติผู้เขียน.....	57



สารบัญรูปรภาพ

	หน้า
รูปที่ 1 แผนผังงานย่อยของบริษัททกสิกร บิสซิเนส-เทคโนโลยี กรุ๊ป.....	2
รูปที่ 2 กระบวนการทำงานของระบบอีทีแอล	6
รูปที่ 3 ส่วนประกอบต่าง ๆ ของภาษาเอสคิวแอล.....	9
รูปที่ 4 หน้าต่างของโปรแกรมตาต้าสเตจ	32
รูปที่ 5 หน้าต่างการทำงานของโปรแกรมวิน เอสซีพี	33
รูปที่ 6 หน้าต่างของดีบีวีซีวีไลเซอร์	34
รูปที่ 7 ระบบฮาดูป	35
รูปที่ 8 การหน้าต่างทำงานของฮิว	36
รูปที่ 9 โฟลวของกระบวนการทำงาน	37
รูปที่ 10 ข้อมูลที่ผู้ใช้งานต้องการ (Requirements).....	38
รูปที่ 11 การกำหนดแหล่งที่มาของข้อมูล.....	39
รูปที่ 12 ข้อมูลที่ได้จากฝั่งผู้ใช้งาน.....	39
รูปที่ 13 ข้อมูลที่กำหนดลงฐานข้อมูล.....	39
รูปที่ 14 ตารางข้อมูลในขาวิว	40
รูปที่ 15 การกำหนดค่าเอกสารดีดีแอล	41
รูปที่ 16 ไฟล์ข้อมูลที่ได้จากผู้ใช้.....	41
รูปที่ 17 หน้าต่างจ็อบในโปรแกรมตาต้า สเตจ	42
รูปที่ 18 หน้าต่างการกำหนดการอ่านไฟล์ข้อมูล.....	42
รูปที่ 19 หน้าต่างการกำหนดตัวแปร	43
รูปที่ 20 หน้าต่างการกำหนดเงื่อนไขให้กับข้อมูลที่มีค่าว่าง	43
รูปที่ 21 หน้าต่างการกำหนดข้อมูลค่าลงขาเข้า	44
รูปที่ 22 หน้าต่างของซีควนซ์ จ็อบ.....	44
รูปที่ 23 ค่าโกลบอลพารามิเตอร์	45
รูปที่ 24 ค่าวาเรียเบิล (Variable)	45
รูปที่ 25 การกำหนดพารามิเตอร์ในจ็อบย่อยเร็กคอนไซล์.....	46
รูปที่ 26 การตรวจสอบจำนวนข้อมูลในขาเข้าและขาออก	46
รูปที่ 27 ซีทจ็อบในตาต้าสเตจ.....	47

รูปที่ 28	ซีทตัวอย่างข้อมูลในตาราง	48
รูปที่ 29	ซีทหน้าสถานะรายละเอียดการรันจีอบ.....	48
รูปที่ 30	ซีทการทดสอบจำนวนแถวของข้อมูล	49
รูปที่ 31	ซีทการตรวจสอบไอดีของข้อมูลต้นทาง	49
รูปที่ 32	ซีทการทดสอบความถูกต้องของชื่อคอลัมน์และประเภทข้อมูล.....	50
รูปที่ 33	รายการในการขึ้นโปรดักชัน.....	50
รูปที่ 34	ตารางของที สคอร์ด (1)	52
รูปที่ 35	ตารางของที สคอร์ด (2)	52



สารบัญตาราง

	หน้า
ตารางที่ 1 ชนิดของข้อมูลบนฐานข้อมูล	10
ตารางที่ 2 ขนาดของข้อมูลที่เป็นตัวเลขในฐานข้อมูล	11
ตารางที่ 3 ขนาดของข้อมูลที่เป็นตัวเลขค่าประมาณในฐานข้อมูล.....	11
ตารางที่ 4 รูปแบบของข้อมูลที่เป็นวันและเวลาในฐานข้อมูล.....	11
ตารางที่ 5 ชนิด และขนาดของตัวอักษรแบบไม่เป็นยูนิโค้ดของข้อมูลในฐานข้อมูล	12
ตารางที่ 6 ชนิด และขนาดของตัวอักษรแบบเป็นยูนิโค้ดของข้อมูลในฐานข้อมูล	12
ตารางที่ 7 ชนิด และขนาดของตัวอักษรแบบไบนารีของข้อมูลในฐานข้อมูล	12
ตารางที่ 8 ตัวอย่างการคำนวณทางคณิตศาสตร์ของภาษาเอสคิวแอล	13
ตารางที่ 9 การเปรียบเทียบของภาษาเอสคิวแอล	13
ตารางที่ 10 ตรรกศาสตร์ของภาษาเอสคิวแอล	14
ตารางที่ 11 ตัวอย่างการใช้ LIKE Clause	26

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ธนาคารกสิกรไทย จำกัด (มหาชน) ได้ก่อตั้งบริษัทกสิกร บิซิเนส-เทคโนโลยี กรุ๊ป เมื่อวันที่ 25 เมษายน 2559 เพื่อพัฒนาเทคโนโลยีโลกการเงินใหม่ โดยร่วมกับพันธมิตรทางเทคโนโลยี ซึ่งมีการจับมือกับฟินเทค และเทค สตาร์ทอัป สร้างนวัตกรรมการเงินรองรับดิจิทัลแบงก์กิ้ง เพิ่มขีดความสามารถในการแข่งขันท่ามกลางการเปลี่ยนแปลงของเทคโนโลยีทางการเงินของโลก โดยบริษัทกสิกร บิซิเนส-เทคโนโลยี กรุ๊ป จะแบ่งกลุ่มเป็นบริษัทย่อยได้ทั้งหมด 5 บริษัท นั่นคือ

1. บริษัทกสิกร เทคโนโลยี กรุ๊ป เซเคเทรเรียต จำกัด วางแผน และติดตามการทำงานของบริษัทในกลุ่ม KBTG สนับสนุนการจัดการด้านการเงินและด้านบุคลากรของกลุ่มบริษัท และประสานการทำงานร่วมกัน (Alignment) ระหว่างกลุ่มบริษัทและธนาคารกสิกรไทย

2. บริษัทกสิกร แล็บส์ จำกัด ค้นคว้าเทคโนโลยี และรูปแบบการดำเนินธุรกิจใหม่ ๆ ที่จะเป็นประโยชน์ต่อระบบธนาคารดิจิทัล (Digital Banking) และระบบเศรษฐกิจดิจิทัล (Digital Economy) สร้างและทดลองระบบต้นแบบก่อนประยุกต์ใช้ในผลิตภัณฑ์ของธนาคารกสิกรไทย

3. บริษัทกสิกร ซอฟต์แวร์ จำกัด ออกแบบ และสร้างระบบเทคโนโลยีสารสนเทศ เพื่อรองรับความต้องการทางธุรกิจ (Business Requirement) ของธนาคารกสิกรไทย และรองรับการนำนวัตกรรม (Innovation) มาใช้ในธนาคารให้มีความรวดเร็วและคุณภาพสูงสุด

4. บริษัทกสิกร โปร์ จำกัด บริหารจัดการโครงสร้างพื้นฐาน และระบบเทคโนโลยีสารสนเทศของธนาคารกสิกรไทย ทั้งในส่วนที่มีการสร้างขึ้นใหม่ ส่วนที่มีการเปลี่ยนแปลง ตลอดจนดูแลรักษาโครงสร้างพื้นฐานให้มีความทันสมัย และมีประสิทธิภาพตลอดเวลา

5. บริษัทกสิกร เซิร์ฟ จำกัด สนับสนุนการทำงานของกลุ่มบริษัท ทั้งทางด้านการพัฒนาระบบเทคโนโลยีสารสนเทศ การทดสอบและการปฏิบัติการระบบเทคโนโลยีสารสนเทศ

Establishment of KASIKORN BUSINESS – TECHNOLOGY GROUP



NOTE:
 * KASIKORN BUSINESS – TECHNOLOGY GROUP established with 5 companies, as a wholly-owned subsidiary of KASIKORNBANK, included in the KASIKORNBANK FINANCIAL CONGLOMERATE, as approved by the Bank of Thailand in October 2015
 * Registered capital in each company at B50m, except for KASIKORN SERVE at B10m
 * KASIKORN SERVE changed names from PROGRESS SOFTWARE COMPANY LIMITED, established March 1993

รูปที่ 1 แผนผังงานย่อยของบริษัทกสิกร บิสซิเนส-เทคโนโลยี กรุ๊ป

เนื่องจากระบบงานของบริษัทเคบีทีจี มีระบบงานหลายส่วน ในที่นี้จะกล่าวถึงงานที่ได้รับมอบหมาย โดยอยู่ภายใต้ส่วนงานของบริษัทกสิกร ซอฟต์แวร์ จำกัด หรือที่เรียกว่า เค ซอฟต์แวร์ ซึ่งภายใต้บริษัทนี้ ยังมีงานย่อย ๆ ที่แบ่งแยกกันออกไปได้อีก โดยโปรเจกต์ที่ได้รับมอบหมายมานั้น ชื่อว่าดิจิทัล เลนด์ดิง ซึ่งเป็นการให้สินเชื่อต่อผู้ที่สนใจผ่านช่องทางออนไลน์ โดยที่ผู้ขอสินเชื่อไม่จำเป็นต้องมาทำธุรกรรมผ่านทางสาขาของธนาคารโดยตรง แต่สามารถขอสินเชื่อได้ด้วยตนเอง ไม่ว่าจะผู้ขอสินค้านั้นจะอยู่ที่ไหน

ซึ่งดิจิทัล เลนด์ดิงนี้ เป็นการทำธุรกรรมที่ยังใหม่สำหรับธนาคารกสิกรมาก แต่ในปัจจุบันหลาย ๆ ธนาคารนั้นได้นำระบบนี้มาใช้งานซักระยะแล้ว แต่สำหรับธนาคารกสิกรนั้น ยังมองว่าเป็นการทำธุรกรรมที่จำเป็นต้องมีการวิเคราะห์ข้อมูลอย่างละเอียด และรอบคอบมากกว่าการให้สินเชื่อแบบปกติ เพราะการให้สินเชื่อแก่ผู้ขอสินเชื่อในแต่ละรายบุคคล จะต้องเป็นผู้ขอสินเชื่อที่ไม่เคยมีประวัติที่เสื่อมเสีย หรือผู้ขอสินเชื่อต้องมีกำลังในการชำระคืนสินเชื่อให้แก่ธนาคาร ถึงจะสามารถทำธุรกรรมผ่านช่องทางออนไลน์ได้

โดยในการคัดเลือกบุคคลที่สนใจทำสินเชื่อ นั้น ไม่ได้หมายความว่าทุกคนจะสามารถขอสินเชื่อได้ทุกคน แต่การขอสินเชื่อผ่านช่องทางออนไลน์นั้นจะมีความแตกต่างกันออกไปตามข้อกำหนดที่ธนาคารได้ตั้งไว้ ซึ่งบุคคลที่ไม่ผ่านเกณฑ์จะถูกปิดตกไป การคัดเลือกบุคคลนั้นจะทำให้ธนาคารสามารถกรองบุคคลได้อย่างละเอียดและตรงไปตรงมามากขึ้น เพราะหน้าที่ในการคัดเลือกบุคคลจะขึ้นอยู่กับระบบที่เป็นแบบดิจิทัลเท่านั้น

ดังนั้น ปัญหาในการคัดเลือกบุคคลที่ไม่เหมาะสมในการขอสินเชื่อลดลงไปได้ ทำให้การคัดเลือกบุคคลของธนาคารมีประสิทธิภาพมากขึ้น ทั้งยังช่วยลดค่าใช้จ่ายในการทำธุรกรรมมากขึ้นด้วย

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อคัดเลือกบุคคลตามข้อมูลขั้นพื้นฐานเพื่อเสนอสินเชื่อให้กับบุคคลที่มีความต้องการการ
กู้ยืม

1.2.2 เพื่อเสนอการขอสินเชื่อผ่านช่องทางออนไลน์ ทำให้สะดวกในการขอสินเชื่อ

1.2.3 เพื่อเพิ่มประสิทธิภาพให้แก่ธนาคารในการคัดเลือกบุคคล

1.2.4 เพื่อช่วยลดขั้นตอนในการขอสินเชื่อ

1.2.5 เพื่อช่วยลดค่าใช้จ่ายในการขอสินเชื่อผ่านช่องทางออนไลน์

1.3 ขอบเขตของการวิจัย

1.3.1 ศึกษาการทำงานของดิจิทัล เลนดิ่ง

1.3.2 ศึกษาการทำงานของอีทีแอล (ETL : Extract Transform Load) โดยเข้าร่วมฟังจากการ
ประชุมของแต่ละทีม

1.3.3 ศึกษาการทำเอกสารซีบีอาร์ (CBR : Common Business Requirement) จากตัวอย่างที่
ได้จากพี่ในทีม

1.3.4 ศึกษาการนำเอกสารความต้องการของผู้ใช้ผ่านโปรแกรมวิน เอสซีพี (WinSCP)

1.3.5 ศึกษาวิธีการสร้างที่เก็บข้อมูล โดยการใช้โปรแกรมดาต้า สเตจ (Designer Client)

1.3.6 ศึกษาการคิวรีข้อมูลผ่านการใช้งานฮิว (Hue Hadoop) และโปรแกรมดีบี วิช่วไลเซอร์
(DbVisualizer)

1.3.7 ศึกษาการนำที่เก็บข้อมูลไปทดสอบบนระบบงานจริง

1.4 วิธีการดำเนินการวิจัย

การวางแผนการดำเนินการในแต่ละขั้นตอนนั้น แบ่งออกตามลักษณะการทำงาน ดังนี้

1.4.1 เรียนรู้ และศึกษาของการขอสินเชื่อผ่านดิจิทัล เลนดิ่ง

1.4.1.1 เก็บข้อมูลตามความต้องการของผู้ใช้

1.4.1.2 ทำความเข้าใจเกี่ยวกับกฎเกณฑ์ของการคัดเลือกบุคคลในการขอสินเชื่อ

1.4.1.3 ทำความตกลงกับผู้ใช้งานของการใช้ที่เก็บข้อมูลในระบบดิจิทัล เลนดิ่ง

1.4.2 การศึกษาการทำงานของอีทีแอล

1.4.2.1 ศึกษาข้อมูล และโครงสร้างการทำงานของแต่ละทีม

1.4.2.2 เข้าร่วมประชุมระหว่างทีม ภายใต้การทำงานของระบบดิจิทัล เลนดิ่ง เพื่อให้

เข้าใจการทำงานร่วมกัน

- 1.4.3 การศึกษาการสร้างเอกสารซีปียาร์
 - 1.4.3.1 เรียนการสร้าง และการอ่านเอกสารซีปียาร์จากพีในทีม
 - 1.4.3.2 ศึกษาการสร้างเอกสารซีปียาร์จากตัวอย่างเอกสารเก่าที่ได้มา
 - 1.4.3.3 ทำความเข้าใจเกี่ยวกับค่าต่าง ๆ ในการสร้างเอกสารว่ามีความหมายว่าอย่างไร
 - 1.4.3.4 สร้างเอกสารซีปียาร์เพื่อเป็นคู่มือในการบอกคุณลักษณะของข้อมูลให้กับผู้ที่ต้องการจะใช้ข้อมูลที่สร้างขึ้น
 - 1.4.3.5 ปรับปรุง และแก้ไขเอกสารซีปียาร์ ให้ตรงตามข้อกำหนดระหว่างภายในทีม และตรงตามเอกสารความต้องการของผู้ใช้ที่ได้รับมา
- 1.4.4 ศึกษาการนำเอกสารความต้องการของผู้ใช้ผ่านโปรแกรมวิน เอสซีพี
 - 1.4.4.1 ดาวนโหลดโปรแกรมวิน เอสซีพี ลงเครื่องคอมพิวเตอร์
 - 1.4.4.2 ทำความเข้าใจของการทำงานของโปรแกรมวิน เอสซีพี
 - 1.4.4.3 นำเอกสารความต้องการของผู้ใช้เข้าไปในโปรแกรมวิน เอสซีพี
- 1.4.5 ศึกษาการทำงาน และทดลองการใช้งานของโปรแกรมตาต้า สเตจ
 - 1.4.5.1 ลงโปรแกรมที่เกี่ยวข้องกับการพัฒนาที่เก็บข้อมูลทั้งหมด
 - 1.4.5.2 เรียนรู้การสร้างที่เก็บข้อมูลพื้นฐานจากคนที่ทำหน้าที่พัฒนา (Dev.)
 - 1.4.5.3 ทดลองใช้โปรแกรมตาต้า สเตจ เพื่อทดสอบว่าโปรแกรมจะทำงานอย่างไร
 - 1.4.5.4 เริ่มการพัฒนาที่เก็บข้อมูลโดยดูจากตัวอย่างที่เก็บข้อมูลที่พีในทีมเคยทำไว้
 - 1.4.5.5 สร้างที่เก็บข้อมูลเพื่อให้ได้คุณลักษณะที่สอดคล้องกับเอกสารซีปียาร์
 - 1.4.5.6 ปรับปรุง และแก้ไขข้อผิดพลาด (Error) ที่เกิดขึ้นให้สามารถโหลดข้อมูลเข้าฐานข้อมูลได้
- 1.4.6 ศึกษาการคิวรีข้อมูลผ่านการใช้งานฮิว และโปรแกรมดีบี วิช่วไลเซอร์
 - 1.4.6.1 ทดสอบการโหลดข้อมูลเข้าตาต้าเบส โดยการคิวรีผ่านฮิว
 - 1.4.6.2 ตรวจสอบความถูกต้องของข้อมูลในฮิวว่ามีข้อมูลตรงกับตัวอย่างเอกสารของผู้ใช้หรือไม่ ถ้าในกรณีข้อมูลไม่เท่ากัน จำเป็นต้องกลับไปแก้ไขที่ตาต้า สเตจที่ได้สร้างไว้
 - 1.4.6.3 คิวรีข้อมูลผ่านการใช้งานโปรแกรมดีบี วิช่วไลเซอร์ เพื่อตรวจสอบดูว่าข้อมูลก่อนที่จะทำการเปลี่ยนแปลงรูปแบบกับข้อมูลหลังจากจากเปลี่ยนแปลงรูปแบบเท่ากันหรือไม่
- 1.4.7 ทดสอบความถูกต้องของการโหลดข้อมูลฝั่งขาเข้าและข้อมูลฝั่งขาออก
 - 1.4.7.1 นำข้อมูลตัวอย่างของผู้ใช้ที่มีปริมาณมากแล้ว มาทดสอบการคิวรีว่าได้ข้อมูลตามที่คิวรีแล้วหรือไม่
- 1.4.8 การเตรียมงานที่พัฒนาก่อนขึ้นระบบจริง หรือที่เรียกว่าแพ็คโค้ด (Pack code)

1.4.8.1 สร้างรายการสำหรับงานที่ต้องการจะขึ้นระบบจริง

1.4.8.2 สร้างดีดีแอลที่ต้องการใช้ขึ้นบนระบบจริง โดยต้องเป็นพาร์ทที่ใช้เก็บข้อมูลบน

ระบบจริง

1.4.9 การนำที่เก็บข้อมูลไปทดสอบบนระบบงานจริง

1.4.9.1 นำที่เก็บข้อมูลที่สร้างดีดีพลอย (Deploy) ขึ้นโปรดักชัน (Production) เพื่อนำเข้าไปใช้ในงานจริงบนระบบงานของธนาคาร

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ได้รับจากการทำโปรเจกต์ที่ได้รับมอบหมายตลอดระยะเวลาภายใต้โครงการสหกิจศึกษากับทางบริษัทเค ซอฟต์ จำกัด สามารถแบ่งได้เป็น 3 ส่วน ดังต่อไปนี้

1.5.1 ประโยชน์ต่อตนเอง

1.5.1.1 ได้รับความรู้ คำศัพท์ และความหมายที่เกี่ยวข้องทางด้านธุรกิจ

1.5.1.2 ได้เรียนรู้และเข้าใจในเรื่องของระบบการทำงานจริง การออกแบบ และพัฒนาโปรแกรมตามความต้องการของผู้ใช้งานที่ได้รับ

1.5.1.3 ได้เข้าร่วมประชุมข้ามทีม เพื่อให้แต่ละฝ่ายสื่อสารได้อย่างถูกต้องและเข้าใจตรงกัน

1.5.1.4 ได้เรียนรู้วิธีการจัดการบริหารเวลา การวางแผนการทำงานการแลกเปลี่ยน และพูดคุยถึงปัญหาที่ได้พบของงาน ตามกระบวนการของอาโใจล์

1.5.1.5 ได้เรียนรู้วิธีการจัดการเกี่ยวกับข้อมูล

1.5.1.6 ได้รับความรู้เกี่ยวกับวิธีการโหลดข้อมูลเข้าดาต้าเบสผ่านโปรแกรมดาต้า สเตจ

1.5.1.7 ได้เรียนรู้เกี่ยวกับการควิรีข้อมูลผ่านเว็บอินเทอร์เฟซ ชื่อว่าฮิว

1.5.2 ประโยชน์ต่อธนาคาร

1.5.2.1 ทำให้จัดข้อมูลได้อย่างเป็นระเบียบ ทำให้ลดระยะเวลาหาข้อมูลได้ง่ายขึ้น

1.5.2.2 บริษัทได้ลดค่าใช้จ่ายในการจ้างบุคคลากร

1.5.2.3 ลดขั้นตอนในการขอสินเชื่อ

1.5.2.4 เพิ่มประสิทธิภาพให้กับธนาคาร ในการคัดเลือกลูกค้าอย่างเหมาะสม เนื่องจากต้องมีการยืนยันตัวตนต่อธนาคาร

1.5.3 ประโยชน์ต่อลูกค้า

1.5.3.1 เพิ่มช่องทางในการขอสินเชื่อ ทำให้ลูกค้าสะดวกในการขอสินเชื่อมากขึ้น

1.5.3.2 ลูกค้าสามารถรับเงินได้อย่างรวดเร็วและง่าย

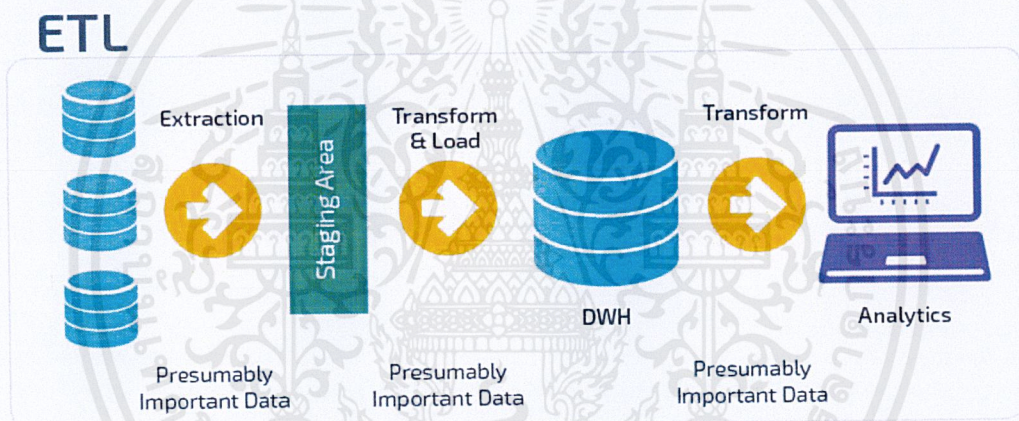
บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 กระบวนการที่เกี่ยวข้องกับการทำงาน

2.1.1 อีทีแอล (ETL : Extract-Transform-Load)

คือกระบวนการหนึ่งในระบบคลังข้อมูล (Data Warehouse) โดยระบบที่ออกแบบเอาไว้จะดึงข้อมูลออกมาจากหลาย ๆ แหล่ง แล้วนำกระบวนการตรวจสอบคุณภาพของข้อมูลมาประยุกต์ใช้ มีการเชื่อมโยง และปรับข้อมูลให้เป็นไปในรูปแบบเดียวกัน เพื่อให้ข้อมูลจากหลาย ๆ แหล่งสามารถใช้งานร่วมกันได้ และทำการส่งมอบ (Delivery) ข้อมูลเหล่านั้นในรูปแบบที่ง่ายต่อการใช้งาน เพื่อใช้ในการตัดสินใจขององค์กรโดยมีกระบวนการหลัก ๆ ที่เกี่ยวข้องกับกระบวนการต่าง ๆ ดังต่อไปนี้



รูปที่ 2 กระบวนการทำงานของระบบอีทีแอล

2.1.1.1 กระบวนการดึงข้อมูล (Extract)

เป็นกระบวนการเริ่มต้นของระบบอีทีแอลจากแหล่งของข้อมูล โดยทั่วไปแล้วระบบคลังข้อมูลจะประกอบด้วย ข้อมูลจากหลาย ๆ แหล่งข้อมูลที่อยู่ต่างที่กันนั้น อาจอยู่ในรูปแบบที่แตกต่างกัน เช่น อาจอยู่ในรูปแบบของฐานข้อมูลคนละชนิด หรือไม่ใช่ฐานข้อมูลแท้จริงซึ่งอาจจะเป็นระบบไฟล์ข้อมูลธรรมดา (Flat Files) หรือในอีกกรณี คือ เป็นข้อมูลในฐานข้อมูลที่ไม่ใช่ระบบอาร์ดีบีเอ็มเอส (RDBMS : Relational Database System) เช่น ข้อมูลในรูปแบบของ Information Management System (IMS) หรือข้อมูลอื่นใดนอกกระบบ เช่น Virtual Storage Access Method (VSAM) หรือ Indexed Sequential Access Method (ISAM) หรือแม้กระทั่งการดึงจากข้อมูลรูปแบบอื่น ๆ ที่ดึงกันโดยตรงจากระบบอื่นเลย เช่น Web Spider หรือ Screen scraping ข้อมูลจะถูกดึงเข้าสู่ระบบแบบ on-the-fly เป้าหมายของกระบวนการดึงข้อมูลนี้คือ ดึงข้อมูลเข้ามาสู่รูปแบบมาตรฐานเดียวกัน เพื่อให้

เหมาะสมต่อการแปลงรูปร่างของข้อมูลในขั้นตอนถัดไป เนื้อแท้ของกระบวนการนี้จะยุ่งเกี่ยวกับกระบวนการอ่านข้อมูลที่ถูกต้องเข้าสู่ ระบบ ส่งผลให้เกิดกระบวนการตรวจสอบข้อมูลว่า ตรงตามรูปแบบที่กำหนดไว้เบื้องต้นหรือไม่ ทั้งรูปแบบของข้อมูล และโครงสร้างของข้อมูล ซึ่งถ้าหากเกิดข้อมูลนั้นไม่ได้มาตรฐานหรือรูปแบบที่กำหนดเอาไว้เบื้องต้น ก็จะทำให้เกิดกระบวนการปฏิเสธ (Reject) ข้อมูลนั้น ๆ

2.1.1.2 กระบวนการแปลงข้อมูล (Transform)

ขั้นตอนการแปลงข้อมูลนี้จะมีการใช้กฎ หรือฟังก์ชันมากมาย เพื่อที่จะแปลงข้อมูลให้ได้ตามที่ต้องการก่อนที่จะนำข้อมูลเข้าไปยังปลายทาง ข้อมูลจากต้นทางบางแหล่งข้อมูลมีความจำเป็นน้อยมาก หรือแทบจะไม่ต้องมีการแปลงข้อมูลเลย แต่ในบางแหล่งอาจจะต้องมีการกระบวนการที่ซับซ้อน ซึ่งจะใช้ทรัพยากรระบบ และเวลาในการประมวลผล

ทั้งนี้ ความซับซ้อนของข้อมูลนั้นขึ้นอยู่กับความต้องการของธุรกิจ หรือเป้าหมายของการนำข้อมูลไปใช้งาน โดยมีกระบวนการตัวอย่างต่อไปนี้

-Selection คือ เลือกเฉพาะคอลัมน์ที่ต้องการที่จะโหลด ยกตัวอย่าง เช่น ถ้าต้นทางข้อมูลมี 3 คอลัมน์ หรือจะเรียกว่ามี 3 แอททริบิวต์ เช่น roll_no age และ salary จะมีการแปลงข้อมูล และเลือกที่จะไม่มีการแปลงข้อมูลหากว่าคอลัมน์ salary มีค่าเป็น Null

-Translation คือ การแปลข้อมูล ยกตัวอย่าง เช่น หากข้อมูลต้นทางมีการเก็บข้อมูลเพศว่า 1 เป็นเพศชาย และ 2 เป็นเพศหญิง จะต้องมีการแปลจากโค้ดที่กำหนดก่อนหน้านี้ให้ 1 = Male และ 2 = Female กระบวนการนี้เรียกว่า กระบวนการทำความสะอาดข้อมูลหรือ data cleaning

-Encoding free form ยกตัวอย่าง เช่น การ mapping จาก “Male” ไปเป็น “1” และ “Mr” ไปเป็น “M”

-Deriving a new calculated value คือ กระบวนการคำนวณเพื่อให้ได้ข้อมูลที่ ต้องการ ยกตัวอย่าง เช่น $sale_amount = qty * unit_price$

-Filtering คือ กระบวนการกรองเฉพาะข้อมูลที่กำหนด

-Sorting คือ กระบวนการเรียงข้อมูลที่ต้องการ

-Joining คือ กระบวนการเชื่อมโยงข้อมูลระหว่างตารางข้อมูล

-Aggregation คือ กระบวนการรวบรวมและสรุปชุดข้อมูล ยกตัวอย่าง เช่น การรวมยอด (summarize) ข้อมูลจากหลาย ๆ ระเบียบจนได้มาเป็น ยอดขายรวม เป็นต้น

-Generating surrogate key

-Transposing or pivoting คือการสลับทิศทางตำแหน่งของการแสดงข้อมูล เช่น การย้าย ระเบียบไปเป็นคอลัมน์ หรือย้ายคอลัมน์มาเป็นระเบียบ เพื่อให้ง่ายต่อการนำข้อมูลไปใช้

-Splitting column into multiple columns คือ กระบวนการหั่นข้อมูลออกเป็นคอลัมน์ย่อย ยกตัวอย่าง เช่น ถ้าข้อมูลเป็นข้อมูลวันที่ เช่น 2010/08/13 10:58:02 ก็จะมีการหั่นเป็นคอลัมน์ย่อย ๆ ได้เป็น 2010,08,13,10,58,02

-Disaggregation คือ กระบวนการยกเลิกการรวมข้อมูล ยกตัวอย่าง เช่น แทนที่จะเก็บข้อมูลเป็นระเบียบ แต่นำข้อมูลนั้นมาเชื่อมต่อกันเป็นระเบียบเดียว

-Lookup and validate data กระบวนการ Lookup ข้ามตารางเพื่อตรวจสอบความถูกต้องของข้อมูล

-Applying any form of simple or complex data validation กระบวนการนี้หากพบว่าการตรวจสอบข้อมูลผิดพลาดจะมีการปฏิเสธข้อมูลบางส่วน หรือทั้งหมดของข้อมูล ขึ้นอยู่กับการกำหนดของผู้ที่ออกแบบการประมวลผลนี้

2.1.1.3. การนำข้อมูลเข้าสู่ระบบ (Loading)

กระบวนการโหลดข้อมูลเข้า โดยทั่วไปจะนำข้อมูลเข้าไปในระบบคลังของข้อมูล โดยขึ้นอยู่กับความต้องการขององค์กร หรือธุรกิจว่าจะให้ข้อมูลไหลไปในทิศทางใด บางองค์กร หรือบางงานจะมีการสะสมของข้อมูล ความถี่ของการนำข้อมูลเข้าสู่ระบบ อาจจะมีการล้างข้อมูลแล้วทับข้อมูลใหม่ โดยทั่วไปแล้วข้อมูลของคลังของข้อมูล จะมีการใช้กันปีต่อปี เมื่อขึ้นปีใหม่แล้วจะมีการล้างข้อมูลของปีเก่า และเก็บไว้ในระบบข้อมูลสำรอง เนื่องจากว่ากระบวนการนำข้อมูลเข้าจะต้องปฏิสัมพันธ์กับฐานข้อมูล ดังนั้นจะต้องมีประเด็นเรื่องของ Database Constraints Referential Integrity และ Database Trigger เข้ามาเกี่ยวข้องด้วย ในกระบวนการนำข้อมูลเข้า ซึ่งสิ่งเหล่านี้รวม ๆ แล้วเรียกว่า กระบวนการควบคุมคุณภาพของข้อมูล (Data Quality performance of ETL process)

2.2 ภาษาที่ใช้สำหรับเขียนโปรแกรม

2.2.1 ภาษาเอสคิวแอล (SQL : Structured Query Language)

เอสคิวแอล เป็นภาษาทางโปรแกรมที่ออกแบบมาเพื่อทำการจัดการข้อมูลที่อยู่ใน Relational Database Management System (RDBMS) หรือไว้สำหรับค้นหาข้อมูล เปลี่ยนแปลง เพิ่ม และลดข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลในรูปแบบตารางที่มีลักษณะเป็นคอลัมน์ (column) และแถว (row) โดยเรียกข้อมูลเหล่านี้ว่าถูกเก็บอยู่ในตาราง (table) ด้วยความสามารถของเอสคิวแอล สามารถสร้างตารางขึ้นมาใหม่ (create) รวมถึงลบ (drop) และเปลี่ยนแปลงค่า (alter) ของตารางได้ ซึ่งในความเป็นจริงแล้วคำสั่งเอสคิวแอลประกอบไปด้วย

-DDL : Data definition language ใช้ในการสร้างฐานข้อมูล

-DML : Data manipulation language ใช้เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง

-DCL : Data control language ใช้ในการกำหนดสิทธิการอนุญาต

ทั้งหมดนี้อยู่ในขอบเขตของการทำ insert query update delete schema creation and modification และ data access control

ภาษาเอสคิวแอล ถูกแบ่งออกมาเป็นส่วนประกอบต่าง ๆ เรียกส่วนเหล่านี้ตามรูปแบบ เช่น

-Clauses คือองค์ประกอบหนึ่งของ statement และ query (ส่วนนี้เป็น Optional)

-Expressions คือการสร้างผลลัพธ์ออกมาในรูปแบบตาราง ที่ประกอบด้วยคอลัมน์ และแถว

จากข้อมูล

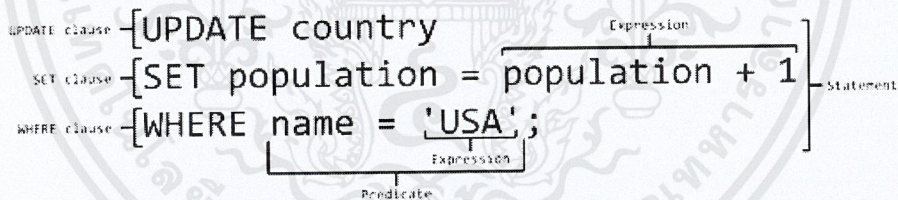
-Predicates คือรูปแบบเงื่อนไขที่มีผลลัพธ์เป็น true/false/unknown หรือก็คือ

Boolean

-Queries คือการดึงข้อมูลตามเงื่อนไข (clause) เป็นส่วนสำคัญในเอสคิวแอล

-Statements คือสามารถมีผลต่อโครงสร้างข้อมูล จัดการข้อมูล transactions/ program flow/session หรือแม้กระทั่งวิเคราะห์ปัญหา โดยจำเป็นต้องจบด้วย semicolon (;) ซึ่งเป็นสิ่งจำเป็นที่ต้องมีทุกครั้งสำหรับเอสคิวแอล

-Insignificant whitespace หรือช่องว่าง สำหรับใน SQL statement และ query จะไม่สนใจ ทำให้เอสคิวแอลสามารถเขียนในรูปแบบที่หลากหลายสวยงาม



รูปที่ 3 ส่วนประกอบต่าง ๆ ของภาษาเอสคิวแอล

2.2.1.1 การกำหนดชนิดค่าตัวแปรในตารางข้อมูล

เป็นการกำหนดชนิดของข้อมูล (data types) ในตารางว่าเป็นข้อมูลแบบใด เช่น ข้อมูลตัวเลข ตัวอักษร วันเวลา หรือแบบไม่มีโครงสร้าง ซึ่งสิ่งเหล่านี้จำเป็นตั้งแต่เริ่มสร้างตารางในฐานข้อมูล (database table) เพื่อให้ข้อมูลที่ใส่ลงสู่ตารางมีความถูกต้องตามที่วางเอาไว้ อีกทั้งยังช่วยให้ฐานข้อมูล ทำงานได้ง่ายขึ้นในการจัดเก็บ และการทำดัชนี (index) ได้เหมาะสมกับข้อมูลที่ใช้งาน โดยชนิดของข้อมูลบนฐานข้อมูลมีด้วยกันหลายชนิด ขึ้นอยู่กับชนิดของฐานข้อมูลที่ต้องการใช้งาน

ตัวอย่างชนิดของข้อมูลบนฐานข้อมูล

Data type	Description
CHARACTER(n)	Character string จำกัดจำนวนตัวอักษรเท่ากับ n ตัว
VARCHAR(n) or CHARACTER VARYING(n)	Character string จำกัดจำนวนตัวอักษรไม่มากกว่า n ตัว
BINARY(n)	Binary string จำกัดจำนวนตัวอักษรเท่ากับ n bytes
VARBINARY(n) or BINARY VARYING(n)	Binary string จำกัดจำนวนตัวอักษรไม่มากกว่า n bytes
BOOLEAN	TRUE หรือ FALSE คือข้อมูลแบบตรรกศาสตร์
INTEGER(p)	ตัวเลขจำนวนเต็ม p หลัก
SMALLINT	ตัวเลขจำนวนเต็ม 5 หลัก
INTEGER	ตัวเลขจำนวนเต็ม 10 หลัก
BIGINT	ตัวเลขจำนวนเต็ม 19 หลัก
DECIMAL(p,s)	ตัวเลขที่มีจำนวนทั้งหมด p หลัก เป็นทศนิยม s หลัก และจำนวนเต็ม p-s หลัก
NUMERIC(p,s)	ตัวเลขที่มีจำนวนทั้งหมด p หลัก เป็นทศนิยม s หลัก และจำนวนเต็ม p-s หลัก
FLOAT(p)	ตัวเลขค่าประมาณรูปทศนิยมจำนวน p หลัก
REAL	ตัวเลขค่าประมาณรูปทศนิยมจำนวน 7 หลัก
FLOAT	ตัวเลขค่าประมาณรูปทศนิยมจำนวน 16 หลัก
DOUBLE PRECISION	ตัวเลขค่าประมาณรูปทศนิยมจำนวน 16 หลัก
DATE	ข้อมูล วัน/เดือน/ปี
TIME	ข้อมูล ชั่วโมง นาที วินาที
TIMESTAMP	ข้อมูล วัน เดือน ปี ชั่วโมง นาที วินาที
INTERVAL	ข้อมูลความต่างในทางเวลา
ARRAY	ชุดของข้อมูลที่มีการจัดเรียง
MULTISET	ชุดของข้อมูลที่ไม่มีการจัดเรียง อาจจะมีข้อมูลซ้ำได้
XML	ข้อมูลชนิด XML

ตารางที่ 1 ชนิดของข้อมูลบนฐานข้อมูล

สามารถจัดกลุ่มประเภทของข้อมูลเอสคีสแอล (SQL Data Types) ที่มีลักษณะเดียวกัน

ได้ 6 กลุ่ม คือ

1. ตัวเลข

DATA TYPE	FROM
bigint	-9.22E+18
int	-2147483648
smallint	-32768
tinyint	0
bit	0
decimal	1E+38
numeric	1E+38
money	-9.22337E+14
smallmoney	-214748.3648

ตารางที่ 2 ขนาดของข้อมูลที่เป็นตัวเลขในฐานข้อมูล

2. ตัวเลขค่าประมาณ

DATA TYPE	FROM
float	-1.79E+308
real	-3.40E+38

ตารางที่ 3 ขนาดของข้อมูลที่เป็นตัวเลขค่าประมาณในฐานข้อมูล

3. วันและเวลา

DATA TYPE	FROM	TO
datetime	Jan 1, 1753	Dec 31, 9999
smalldatetime	Jan 1, 1900	Jun 6, 2079
date	รูปแบบวัน June 30, 1991	
time	รูปแบบเวลา 12:30 P.M	

ตารางที่ 4 รูปแบบของข้อมูลที่เป็นวันและเวลาในฐานข้อมูล

4. ตัวอักษร (Non Unicode)

DATA TYPE	Description
char	สูงสุด 8,000 ตัวอักษร (จำนวนตัวอักษรต้องเท่ากัน และไม่ใช้ Unicode)
varchar	สูงสุด 8,000 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และไม่ใช้ Unicode)
text	สูงสุด 2,147,483,647 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และไม่ใช้ Unicode)

ตารางที่ 5 ชนิด และขนาดของตัวอักษรแบบไม่เป็นยูนิโค้ดของข้อมูลในฐานข้อมูล

5. ตัวอักษร (Unicode)

DATA TYPE	Description
char	สูงสุด 8,000 ตัวอักษร (จำนวนตัวอักษรต้องเท่ากัน และไม่ใช้ Unicode)
varchar	สูงสุด 8,000 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และไม่ใช้ Unicode)
text	สูงสุด 2,147,483,647 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และไม่ใช้ Unicode)

ตารางที่ 6 ชนิด และขนาดของตัวอักษรแบบเป็นยูนิโค้ดของข้อมูลในฐานข้อมูล

6. ตัวอักษร (Binary)

DATA TYPE	Description
binary	สูงสุด 8,000 bytes (Fixed-length binary data)
varbinary	สูงสุด 8,000 bytes (Variable length binary data)
image	สูงสุด 2,147,483,647 bytes (Variable length Binary Data)

ตารางที่ 7 ชนิด และขนาดของตัวอักษรแบบไบนารีของข้อมูลในฐานข้อมูล

2.2.1.2 การกำหนดกระบวนการในการเลือกข้อมูล

คือกระบวนการทำงานเพื่อเลือกข้อมูลด้วยเงื่อนไขหรือวิธีการที่ต้องการ โดยระบบฐานข้อมูลจะมีการจอง key word บางคำ หรือตัวอักษรไว้สำหรับให้ SQL statement ใช้ระบุเงื่อนไข (Where clause) เพื่อเป็นการสั่งระบบให้ทำตามเงื่อนไขที่วางไว้ เช่น การเปรียบเทียบการคำนวณทางคณิตศาสตร์ การทำกระบวนการเหล่านี้มักจะใช้ระบุเงื่อนไขภายใต้ SQL statement และสามารถกำหนดได้หลายเงื่อนไขภายใน statement นั้น ๆ

2.2.1.2.1 ตัวอย่างรูปแบบกระบวนการ (SQL Operator)

1. คำนวณทางคณิตศาสตร์ (Arithmetic operators)

จากตัวอย่าง a = 5 และ b = 10

Operator	Description	Example
บวก +	เพิ่มค่าของข้อมูล	a + b = 5
ลบ -	ลดค่าของข้อมูล	a - b = 5
คูณ *	เพิ่มค่าของข้อมูลเป็นจำนวนเท่า	a * b = 50
หาร /	ลดค่าของข้อมูลเป็นจำนวนเท่า	b / a = 2
โมดูลัส %	หารค่าเพื่อเอาเศษ	b % a = 0

ตารางที่ 8 ตัวอย่างการคำนวณทางคณิตศาสตร์ของภาษาเอสคิวแอล

2. เปรียบเทียบ (Comparison operators)

จากตัวอย่าง a = 5 และ b = 10

Operator	Description	Example
=	เงื่อนไขข้อมูลด้านซ้าย และด้านขวาเท่ากัน	(a = b) -> false
!=	เงื่อนไขข้อมูลด้านซ้าย และด้านขวาไม่เท่ากัน	(a != b) -> true
<>	เงื่อนไขข้อมูลด้านซ้าย และด้านขวาไม่เท่ากัน	(a <> b) -> true
>	เงื่อนไขข้อมูลด้านซ้าย มากกว่าด้านขวา	(a > b) -> false
<	เงื่อนไขข้อมูลด้านซ้าย น้อยกว่าด้านขวา	(a < b) -> true
>=	เงื่อนไขข้อมูลด้านซ้าย มากกว่าหรือเท่ากับด้านขวา	(a >= b) -> false
<=	เงื่อนไขข้อมูลด้านซ้าย น้อยกว่าหรือเท่ากับด้านขวา	(a <= b) -> true
!<	เงื่อนไขข้อมูลด้านซ้าย ไม่น้อยกว่าด้านขวา	(a !< b) -> false
!>	เงื่อนไขข้อมูลด้านซ้าย ไม่มากกว่าด้านขวา	(a !> b) -> true

ตารางที่ 9 การเปรียบเทียบของภาษาเอสคิวแอล

3. ตรรกศาสตร์ (Logical operators)

Operator	Description
ALL	เปรียบเทียบข้อมูลภายในชุดทั้งหมดระหว่าง 2 ชุด
AND	เชื่อมเงื่อนไขตั้ง 2 เงื่อนไขขึ้นไป ต้องเป็นจริงทั้งหมดถึงจะจริง

ANY	เปรียบเทียบข้อมูลกับชุดข้อมูล โดยถ้ามีอย่างน้อย 1 ค่าที่เหมือนกัน จะมีค่าเป็นจริง
BETWEEN	เปรียบเทียบข้อมูลว่าอยู่ระหว่างค่าต่ำสุด และสูงสุด
EXISTS	เปรียบเทียบข้อมูลว่ามีปรากฏอยู่ในแถวที่กำหนด
IN	เปรียบเทียบข้อมูลกับชุดข้อมูล โดยถ้ามีอย่างน้อย 1 ค่าที่เหมือนกัน จะมีค่าเป็นจริง
LIKE	เปรียบเทียบข้อมูลว่าเป็นส่วนประกอบภายในข้อมูลอีกค่าหนึ่ง
NOT	เงื่อนไขปฏิเสธ
OR	เชื่อมเงื่อนไข 2 เงื่อนไขขึ้นไป ถ้าเป็นจริงอันหนึ่งทั้งหมดจะเป็นจริง

ตารางที่ 10 ตรรกศาสตร์ของภาษาเอสคิวแอล

2.2.1.3 การสร้างเงื่อนไขจากกระบวนการเพื่อเลือกข้อมูล

คือการรวมเอาค่าตัวแปร หรือข้อมูลตั้งแต่ 1 ตัวขึ้นไป รวมถึง SQL Operator SQL function ที่สามารถกำหนดค่าของข้อมูลได้ โดย SQL Expression คือ เปรียบเสมือนสูตร หรือ ประโยคในภาษาเขียน ซึ่งใช้ในการดึงข้อมูล (query) ทำให้เลือกข้อมูลที่ต้องการจากฐานข้อมูลได้

พิจารณาจากรูปแบบพื้นฐาน (Syntax) ของ SQL select statement ต่อไปนี้

```
SELECT column1, column2, columnN
FROM table_name
WHERE [CONDITION|EXPRESSION];
```

SQL expression มีด้วยกันหลายชนิด ดังนี้

1. การตัดสินใจ (Boolean)

SQL Boolean Expression จะดึงเอาข้อมูลจากฐานข้อมูลด้วยเงื่อนไขที่ตรงตามค่าที่กำหนด

```
SELECT column1, column2, columnN
FROM table_name
WHERE SINGLE VALUE MATCHING EXPRESSION;
```

2. ตัวเลข (Numeric)

SQL Numeric Expression คือการคำนวณค่าทางคณิตศาสตร์ บางฟังก์ชันใช้กระบวนการคำนวณแบบคณิตศาสตร์ เช่น avg(), sum(), count()

```
SELECT numerical_expression as OPERATION_NAME
[FROM table_name
```

WHERE CONDITION];

3. วันเวลา (Date)

SQL Date Expression ใช้แสดงข้อมูลเวลา และวันที่ปัจจุบัน

2.2.1.4 คำสั่งเอสคิวแอลพื้นฐาน

2.2.1.4.1 คำสั่งสำหรับฐานข้อมูล

1. การสร้างฐานข้อมูล

รูปแบบของการสร้างฐานข้อมูล คือ SQL Create database statement โดยชื่อฐานข้อมูลในระบบจะต้องไม่ซ้ำกันภายในอาร์ตบีเอ็มเอสที่ใช้งาน สามารถเขียนได้ตามรูปแบบข้างล่างนี้

```
CREATE DATABASE <DatabaseName>;
```

2. การลบตารางออกจากฐานข้อมูล

รูปแบบของลบตารางออกจากฐานข้อมูล คือ SQL Drop table statement ข้อมูลที่ต้องระบุคือ table name

```
DROP TABLE <table_name>;
```

คำสั่ง SQL Drop table statement เมื่อทำการรันสำเร็จแล้ว จะไม่สามารถเรียกข้อมูลภายในตารางกลับคืนมาได้อีก

3. การเลือกใช้งานฐานข้อมูลที่ต้องการ

รูปแบบ SQL user database statement หรือการ select database สามารถระบุชื่อของฐานข้อมูลตามที่ต้องการ เพราะชื่อฐานข้อมูลจะไม่มีซ้ำกัน

```
USE <DatabaseName>;
```

สมมติ หากต้องการใช้งานฐานข้อมูล phpmyadmin ให้ใช้คำสั่ง SQL user database statement ตามด้วยชื่อฐานข้อมูล จะได้

```
SQL> USE phpmyadmin;
```

2.2.1.4.2 คำสั่งสำหรับตาราง (Table)

1. การสร้างตารางข้อมูลในฐานข้อมูล

การสร้างตารางที่เอาไว้เก็บข้อมูลที่ต้องการ เรียกคำสั่งว่า SQL Create table statement ซึ่งข้อมูลแบบตารางจำเป็นต้องกำหนดค่าของคอลัมน์ รวมถึงประเภทของข้อมูลของแต่ละคอลัมน์ว่าจะใช้เก็บข้อมูลชนิดใด

รูปแบบในการสร้างตารางในฐานข้อมูลนั้น ข้อมูลที่ต้องระบุคือ table name, column name, data type เขียนได้เป็น

```
CREATE TABLE table_name(  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    .....  
    columnN datatype,  
    PRIMARY KEY(one or more columns)  
);
```

คำสั่ง CREATE TABLE เป็นคีย์เวิร์ดเพื่อบอกกับฐานข้อมูลว่าต้องการจะทำอะไร ในที่นี้คือสร้างตารางใหม่ โดยที่ชื่อของตารางต้องไม่ซ้ำกับตารางเดิมที่มีอยู่ในฐานข้อมูล ซึ่งต้องใช้รูปแบบตามตัวอย่างที่กำหนดไว้ หลังจากนั้นในส่วนของวงเล็บที่ตามมา จะเป็นการกำหนดข้อมูลแต่ละคอลัมน์ของตาราง ว่าประเภทของข้อมูลที่จะใช้นั้น คือข้อมูลประเภทใด

ตัวอย่างของการสร้างตาราง สมมติหากต้องการสร้างตาราง CUSTOMERS ที่ใช้ ID เป็น primary key ซึ่งต้องมีค่าไม่ซ้ำกัน และไม่มีค่าว่าง (NOT NULL)

```
SQL> CREATE TABLE CUSTOMERS(  
    ID INT NOT NULL,  
    NAME VARCHAR (20) NOT NULL,  
    AGE INT NOT NULL,  
    ADDRESS CHAR (25) ,  
    SALARY DECIMAL (18, 2),  
    PRIMARY KEY (ID)  
);
```

สามารถตรวจสอบตารางที่สร้างเสร็จสมบูรณ์ โดยดูจากข้อความที่แสดงกลับมาจากฝั่ง SQL server หรือสามารถใช้คำสั่ง DESC ในการดูข้อมูลของ table ที่สร้างไว้ได้

```
SQL> DESC CUSTOMERS;
```

```
+-----+-----+-----+-----+-----+  
| Field | Type      | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+
```

```

| ID      | int(11)      | NO | PRI |      |      |
| NAME    | varchar(20)  | NO |      |      |      |
| AGE     | int(11)      | NO |      |      |      |
| ADDRESS | char(25)     | YES|      | NULL |      |
| SALARY  | decimal(18,2)| YES|      | NULL |      |
+-----+-----+-----+-----+-----+-----+

```

การแสดงผลรายละเอียดของตารางว่ามีคอลัมน์ชื่ออะไรบ้าง รวมถึงแต่ละคอลัมน์มีข้อมูลชนิดใด ซึ่งสามารถเริ่มใช้ตารางนี้ในการเก็บข้อมูลที่ต้องการ

2. การเพิ่มแถวข้อมูลลงในตารางของฐานข้อมูล (Insert Query)

การเพิ่มแถวข้อมูลลงในตารางของฐานข้อมูล ด้วยการกำหนดประเภทของข้อมูลแต่ละคอลัมน์ที่ต้องกำหนดให้ถูกต้อง ข้อมูลที่เพิ่มลงไปจะอยู่ในรูปแบบที่ละแถวข้อมูล จะเรียกคำสั่งเพิ่มข้อมูลนี้ว่า SQL Insert into statement

รูปแบบ SQL Insert into statement ข้อมูลที่ต้องระบุคือ table name/column name/value โดยคำสั่งการเพิ่มแถวข้อมูล เป็นดังนี้

```

INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);

```

การเรียงลำดับของชื่อคอลัมน์จะต้องเหมือนกันกับลำดับของข้อมูล (column) เพราะการเพิ่มแถวข้อมูลลงในตาราง จะทำการเอาข้อมูลใส่ลงแต่ละคอลัมน์ตามลำดับที่ระบุ

สำหรับกรณีที่ระบุข้อมูลครบทุกคอลัมน์ ไม่จำเป็นต้องระบุชื่อคอลัมน์ก็ได้ แต่ข้อมูลจะต้องกำหนดให้ตรงตามลำดับของคอลัมน์ในตารางโดยมีคำสั่ง เป็นดังนี้

```

INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...valueN);

```

ตัวอย่างการเพิ่มแถวข้อมูลในตาราง สมมติ ถ้าลองใส่ข้อมูลด้วย SQL Insert into statement ลงในตาราง CUSTOMERS ที่ได้สร้างไว้ จำนวน 6 แถวข้อมูลตามด้านล่าง

```

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00);
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (2, 'Khilan', 25, 'Delhi', 1500.00);
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (3, 'kaushik', 23, 'Kota', 2000.00);
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

```

```
VALUES (4, 'Chaitali', 25, 'Mumbai', 6500.00);
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (5, 'Hardik', 27, 'Bhopal', 8500.00);
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (6, 'Komal', 22, 'MP', 4500.00);
```

ถ้าลองใส่ข้อมูล SQL Insert into statement แบบไม่ต้องระบุชื่อคอลัมน์ จะได้ว่า

```
INSERT INTO CUSTOMERS
VALUES (7, 'Muffy', 24, 'Indore', 10000.00);
```

ผลลัพธ์ที่ได้จะพบข้อมูลทั้งหมด 7 เรคคอร์ดหรือ 7 แถว ในตาราง 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 |
+---+-----+---+-----+-----+
```

3. การลบตารางออกจากฐานข้อมูล (Drop table)

เรียกคำสั่งว่า SQL Drop table statement ซึ่งการ drop table นี้จะทำการลบข้อมูลภายในตาราง รวมถึงโครงสร้างของตารางนี้ออกจากฐานข้อมูล และไม่สามารถดึงข้อมูลกลับมาได้อีก รูปแบบของคำสั่งนี้ จะมีข้อมูลที่ต่อระบุคือ table name

```
DROP TABLE <table_name>;
```

คำสั่งการลบตารางออกจากฐานข้อมูล เมื่อทำการรันสำเร็จแล้วจะไม่สามารถเรียกข้อมูลภายในตารางกลับคืนมาได้

ตัวอย่างการลบตารางออกจากฐานข้อมูล สมมติว่ามีข้อมูลในตารางที่ชื่อ CUSTOMERS ในฐานข้อมูลโดยมีโครงสร้างคอลัมน์ ดังนี้

```
SQL> DESC CUSTOMERS;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI		
NAME	varchar(20)	NO			
AGE	int(11)	NO			
ADDRESS	char(25)	YES		NULL	
SALARY	decimal(18,2)	YES		NULL	

หมายความว่าตอนนี้มีตารางชื่อ CUSTOMERS ที่มีข้อมูล และพร้อมที่ใช้งานอยู่ ถ้าทำการลบตาราง จะใช้คำสั่งตามนี้

```
SQL> DROP TABLE CUSTOMERS;
```

Query OK, 0 rows affected (0.01 sec)

หลังจากทำการลบตารางเรียบร้อยแล้ว ถ้าลองดูข้อมูลในตารางอีกครั้งด้วยคำสั่ง DESC จะได้รับการแจ้งเตือนว่าไม่พบข้อมูลตารางบนระบบ

```
SQL> DESC CUSTOMERS;
```

ERROR 1146 (42S02): Table 'TEST.CUSTOMERS' doesn't exist

4. การแสดงข้อมูลในตารางของฐานข้อมูล (Select Query)

หลังจากที่ได้ทำการสร้างฐานข้อมูล รวมถึงสร้างตาราง และนำข้อมูลใส่ในตารางเรียบร้อยแล้ว ถ้าต้องการดึงข้อมูลที่อยู่ในตารางกลับออกมาเพื่อใช้ตามเงื่อนไขที่ระบุกัน โดยคำสั่งที่ใช้คือ SQL Select statement ซึ่งผลลัพธ์ที่ได้จะเรียกว่า result set ที่มีลักษณะเป็นข้อมูลตาราง ที่ประกอบด้วยคอลัมน์และแถว

รูปแบบของ SQL Select statement จำเป็นต้องใช้ข้อมูล column name และ table name ในการระบุชุดของข้อมูลที่จะแสดง

```
SELECT column1, column2, columnN
```

```
FROM table_name;
```

column1, column2 คือชื่อคอลัมน์ที่เป็น field ในตารางที่ต้องการดึงข้อมูล ซึ่งสามารถแทนชื่อคอลัมน์ได้ด้วยตัวเลขลำดับของคอลัมน์ในตารางนั้น ๆ ได้เช่นกัน เช่น column1 มีค่าเท่ากับ 1 และ column มีค่าเท่ากับ 2

```
SELECT 1, 2
```

FROM table_name;

ในกรณีที่ต้องการดึงข้อมูลทุกคอลัมน์ในตารางนั้น สามารถใช้รูปแบบตามด้านล่างนี้

SELECT * FROM table_name;

ตัวอย่างการแสดงผลข้อมูลในตาราง จากข้อมูลที่ได้ทำการเพิ่มลงในตาราง CUSTOMERS ไปเรียบร้อยแล้วจำนวน 7 แถว

```
+---+-----+---+-----+-----+
| 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 |
+---+-----+---+-----+-----+
```

ถ้าต้องการแสดงผลข้อมูลของตาราง CUSTOMERS โดยเอาเฉพาะ column ID/NAME/SALARY มาแสดง จะใช้คำสั่ง SQL Select statement ดังนี้

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS;
```

ผลลัพธ์ที่ได้จะพบข้อมูลทั้งหมด 7 เรคคอร์ดหรือ 7 แถวในตาราง CUSTOMERS และมีข้อมูลคอลัมน์ตามลำดับที่ต้องการ

```
+---+-----+-----+
| ID | NAME   | SALARY |
+---+-----+-----+
| 1 | Ramesh | 2000.00 |
| 2 | Khilan | 1500.00 |
| 3 | kaushik | 2000.00 |
| 4 | Chaitali | 6500.00 |
| 5 | Hardik | 8500.00 |
| 6 | Komal  | 4500.00 |
+---+-----+-----+
```

```
| 7 | Muffy | 10000.00 |
```

```
+-----+
```

แต่ถ้าต้องการดึงข้อมูลจากตาราง CUSTOMERS โดยเอาคอลัมน์ที่มีอยู่ทั้งหมด จะสามารถแสดงเป็น SQL Select statement ดังนี้

```
SQL> SELECT * FROM CUSTOMERS;
```

ผลลัพธ์ที่ได้

```
+-----+
```

```
| 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 |
```

```
+-----+
```

5. การสร้างเงื่อนไขเพื่อเลือกข้อมูลที่ต้องการ (Where Clause)

ในกรณีที่ข้อมูลในตารางมีจำนวนมาก และต้องการข้อมูลเฉพาะแถวที่ต้องการตามเงื่อนไข โดยจะเพิ่มส่วนของ SQL Where Clause statement เข้ามาช่วยในการเลือกข้อมูลจากตาราง หลักการของ Where Clause ก็คือการนำเอา SQL Operator และตัวแปรมาสร้างเป็น SQL Expression ในรูปแบบเงื่อนไข

ไม่เพียงแต่ SQL Select statement เท่านั้นที่สามารถใช้เงื่อนไข Where Clause ได้ แต่การลบข้อมูล รวมถึงการเปลี่ยนแปลงข้อมูลก็สามารถใช้ Where Clause มาช่วยได้ด้วยเช่นกัน

รูปแบบ SQL Select statement สามารถนำเอา Where condition เพิ่มต่อเข้าไปได้ เพื่อให้ข้อมูลที่แสดงถูกรองให้เหลือเฉพาะข้อมูลที่อยู่ในเงื่อนไขที่ต้องการ

```
SELECT column1, column2, columnN
```

```
FROM table_name
```

```
WHERE [condition]
```

ในส่วนของเงื่อนไข สามารถนำเอา SQL Operator เช่น = > < LIKE NOT มาสร้างเป็น SQL Expression เพื่อเปรียบเทียบข้อมูลในตาราง และข้อมูลที่ต้องการได้

สมมติข้อมูลในตาราง CUSTOMERS เป็น

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

สามารถใช้ Where Condition เพิ่มเข้ามาต่อจาก SQL Select statement ได้ปกติ เช่น ต้องการดึงข้อมูล column ID/NAME/SALARY จากตาราง CUSTOMERS ที่มีเงื่อนไขว่า SALARY ต้องมากกว่า 2,000 ขึ้นไป

```
SQL> SELECT ID, NAME, SALARY
FROM CUSTOMERS
WHERE SALARY > 2000;
```

ผลลัพธ์ที่ได้ มีด้วยกัน 4 แถวที่ SALARY > 2000

ID	NAME	SALARY
4	Chaitali	6500.00
5	Hardik	8500.00
6	Komal	4500.00
7	Muffy	10000.00

สำหรับข้อมูล ที่มีประเภทของข้อมูลแบบตัวอักษร หรือ text จำเป็นต้องใส่ keyword single quote ('text') ข้างหน้าและข้างหลังของคำนั้น เพื่อบ่งบอกให้ทราบว่าข้อมูลต่อไปนี้เป็นข้อมูลชนิดตัวอักษร ไม่ใช่ keyword SQL

```
SQL> SELECT ID, NAME, SALARY
FROM CUSTOMERS
WHERE NAME = 'Hardik';
```

ผลลัพธ์ที่ได้ คือ

ID	NAME	SALARY
5	Hardik	8500.00

6. การเชื่อมความสัมพันธ์ของเงื่อนไข (And Or Clause)

ถ้าต้องการให้มีเงื่อนไขของ where clause ที่มากกว่า 1 เงื่อนไข โดยอาศัยการเชื่อมความสัมพันธ์ด้วย And & Or Clause ซึ่งเป็น SQL Operator อย่างหนึ่งเหมือนกัน

6.1 AND Operator

ใช้ทำการเชื่อมเงื่อนไขหลาย ๆ เงื่อนไข ภายใต้ Where Clause statement รูปแบบของการเชื่อมหลาย ๆ เงื่อนไข ด้วยการใส่ keyword “AND” นำหน้าแต่ละเงื่อนไขที่เพิ่มเข้ามา

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition1] AND [condition2]...AND [conditionN];
```

ภายใต้ Where condition ใน SQL Select statement ทุกเงื่อนไขที่เชื่อมกันด้วย And จำเป็นต้องได้ค่าเป็น TRUE ทั้งหมด Where Clause statement นั้นถึงจะเป็นจริงได้

ตัวอย่างการแสดงผลข้อมูลในตาราง สมมติให้มีข้อมูลในตาราง CUSTOMERS เป็น

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 |
+----+-----+----+-----+-----+
```

สามารถดึงข้อมูลจาก 2 เงื่อนไขที่ต้องเป็นจริงทั้งคู่ เช่น SALARY > 2000 และ age < 25

```
SQL> SELECT ID, NAME, SALARY
FROM CUSTOMERS
WHERE SALARY > 2000 AND age < 25;
```

ผลลัพธ์ที่ได้ มีด้วยกัน 2 แถวที่ SALARY > 2000 และ age < 25

```
+----+-----+-----+
| ID | NAME | SALARY |
+----+-----+-----+
| 6 | Komal | 4500.00 |
| 7 | Muffy | 10000.00 |
+----+-----+-----+
```

6.2 OR Operator

ใช้ทำการเชื่อมเงื่อนไขหลาย ๆ เงื่อนไข ภายใต้ Where Clause statement รูปแบบในการเชื่อมหลาย ๆ เงื่อนไขต่อกัน ด้วยการใช้ keyword “OR” นำหน้าแต่ละเงื่อนไขที่เพิ่มเข้ามา

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition1] OR [condition2]...OR [conditionN];
```

ภายใต้ Where condition ใน SQL Select statement ทุกเงื่อนไขที่เชื่อมกันด้วย Or มีอย่างน้อย 1 เงื่อนไขที่ได้ค่าเป็น TRUE จะทำให้ Where Clause statement นั้นเป็นจริงได้ทันที

ตัวอย่างการแสดงผลข้อมูลในตาราง สมมติข้อมูลในตาราง CUSTOMERS เป็น

```
+----+-----+----+-----+-----+
| 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 |
```

```
+---+-----+-----+-----+
```

สามารถดึงข้อมูลจาก 2 เงื่อนไขที่ต้องเป็นจริงทั้งคู่ เช่น SALARY > 2000 หรือ age < 25

```
SQL> SELECT ID, NAME, SALARY
FROM CUSTOMERS
WHERE SALARY > 2000 OR age < 25;
```

ผลลัพธ์ที่ได้ มีด้วยกัน 5 แถวที่ SALARY > 2000 หรือ age < 25

```
+---+-----+-----+
```

```
| ID | NAME | SALARY |
```

```
+---+-----+-----+
```

```
| 3 | kaushik | 2000.00 |
```

```
| 4 | Chaitali | 6500.00 |
```

```
| 5 | Hardik | 8500.00 |
```

```
| 6 | Komal | 4500.00 |
```

```
| 7 | Muffy | 10000.00 |
```

```
+---+-----+-----+
```

7. การเปรียบเทียบค่าแบบส่วนประกอบ (Like Clause)

ในการทำงานของ Where Clause โดยอาศัย SQL Operator ต่าง ๆ ในการเปรียบเทียบข้อมูล ซึ่งหนึ่งในนั้นคือ LIKE operator ที่สามารถเปรียบเทียบข้อมูลกับตัวแปรในลักษณะที่ข้อมูลหนึ่ง เป็น ส่วนประกอบ (contain) ของอีกข้อมูลหนึ่งหรือไม่ เช่น คำว่า "DAY" เป็นส่วนประกอบภายใต้คำว่า "TODAY" ซึ่งสามารถเอา Like clause มาใช้ทำการเปรียบเทียบแบบนี้ได้ โดยหลักการ wildcard ซึ่งมีด้วยกัน 2 แบบคือ

- Percent (%) มีค่าเท่ากับตัวอักษรอะไรก็ได้ และจำนวนกี่ตัวก็ได้

- Underscore (_) มีค่าเท่ากับตัวอักษรอะไรก็ได้ เพียงแค่ 1 ตัว

รูปแบบของการนำเอา LIKE operator มาใช้งาน โดยจะใช้ Percent (%) หรือ Underscore (_) เป็นตัวระบุค่า wildcard ยกตัวอย่างการเขียนเงื่อนไขนี้ จะได้เป็น

SELECT FROM table_name
WHERE column LIKE 'XXXX%'

หรือ

SELECT FROM table_name
WHERE column LIKE '%XXXX%'

หรือ

SELECT FROM table_name
WHERE column LIKE 'XXXX_'

หรือ

SELECT FROM table_name
WHERE column LIKE '_XXXX'

หรือ

SELECT FROM table_name
WHERE column LIKE '_XXXX_'

ค่า XXX สามารถเป็นได้ทั้งข้อมูลตัวเลขหรือตัวอักษรก็ได้ และยังสามารถนำเอา AND OR Clause มาเพิ่มเงื่อนไขได้ด้วย

ตัวอย่างการใช้ LIKE Clause ได้ดังนี้

LIKE Clause	Description
WHERE SALARY LIKE '200%'	ค้นหาข้อมูลที่ขึ้นต้นด้วย 200
WHERE SALARY LIKE '%200%'	ค้นหาข้อมูลที่มีส่วนประกอบ 200
WHERE SALARY LIKE '_00%'	ค้นหาข้อมูลที่ขึ้นต้นด้วยอะไรก็ได้ 1 ตัว แล้วตามด้วย 00
WHERE SALARY LIKE '2_%_'	ค้นหาข้อมูลที่ขึ้นต้นด้วย 2 และข้อมูลต้องมีอย่างน้อย 3 ตัวอักษรขึ้นไป
WHERE SALARY LIKE '%2'	ค้นหาข้อมูลบที่ลงท้ายด้วย 2
WHERE SALARY LIKE '_2%3'	ค้นหาข้อมูลที่ตัวอักษรตัวที่ 2 มีค่าเท่ากับ 2 และลงท้ายด้วย 3
WHERE SALARY LIKE '2__3'	ค้นหาข้อมูลที่ขึ้นต้นด้วย 2 และลงท้ายด้วย 3 ข้อมูลทั้งหมดรวมกันต้องมี 5 ตัว

ตารางที่ 11 ตัวอย่างการใช้ LIKE Clause

ตัวอย่างข้อมูลในตาราง CUSTOMERS จะได้เป็น

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

คำสั่งแสดงข้อมูลบนจากตาราง CUSTOMER สำหรับข้อมูลที่มี SALARY ขึ้นต้นด้วย 200

```
SQL> SELECT * FROM CUSTOMERS  
WHERE SALARY LIKE '200%';
```

ผลลัพธ์ที่ได้ จะเห็นว่าบรรทัดที่ 1 และ 3 ที่ SALARY ขึ้นต้นด้วย 200 หรือก็คือมีค่าเป็น 2000

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
3	kaushik	23	Kota	2000.00

8. การจัดเรียงข้อมูลที่แสดงผลตามลำดับ (ORDER BY Clause)

สามารถใช้คำสั่ง SQL ORDER BY Clause ต่อหลัง SQL Select เพื่อให้ข้อมูลที่แสดงมีการจัดลำดับเรียงจากน้อยไปมาก หรือมากไปน้อยตามที่ต้องการได้ ซึ่งคำสั่ง ORDER BY จะทำการเรียงข้อมูลจากน้อยไปมากเป็นค่าเริ่มต้น (Default)

รูปแบบ SQL ORDER BY Clause ที่ใช้งานภายใต้ SQL Select query statement สามารถเขียนได้ว่า

```
SELECT column-list  
FROM table_name
```

[WHERE condition]

[ORDER BY column1, column2, .. columnN] [ASC | DESC];

สามารถใช้ 1 คอลัมน์ หรือหลายคอลัมน์ในการจัดเรียงข้อมูลก็ได้ และสามารถจัดเรียงจากน้อยไปมาก หรือมากไปน้อยด้วยการใส่

-ASC (Default) เรียงจากน้อยไปมาก

-DESC เรียงจากมากไปน้อย

ตัวอย่างการใช้ ORDER BY Clause สมมติถ้ามีข้อมูลในตาราง CUSTOMERS แสดงได้เป็น

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

การดึงข้อมูลจากตาราง CUSTOMERS ทุกคอลัมน์ โดยจัดเรียงข้อมูลด้วยคอลัมน์ที่ชื่อ NAME และ SALARY จากน้อยไปมาก

```
SQL> SELECT * FROM CUSTOMERS
```

```
ORDER BY NAME, SALARY;
```

ผลลัพธ์ที่ได้

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
3	kaushik	23	Kota	2000.00
2	Khilan	25	Delhi	1500.00

```
| 6 | Komal   | 22 | MP       | 4500.00 |
| 7 | Muffy   | 24 | Indore   | 10000.00 |
| 1 | Ramesh  | 32 | Ahmedabad | 2000.00 |
```

```
+-----+-----+-----+-----+-----+
```

การดึงข้อมูลจากตาราง CUSTOMERS ทุกคอลัมน์ โดยจัดเรียงข้อมูลด้วยคอลัมน์ที่ชื่อ NAME จากมากไปน้อย

```
SQL> SELECT * FROM CUSTOMERS
ORDER BY NAME DESC;
```

ผลลัพธ์ที่ได้

```
+-----+-----+-----+-----+-----+
| ID | NAME    | AGE | ADDRESS  | SALARY |
+-----+-----+-----+-----+
| 1 | Ramesh  | 32 | Ahmedabad | 2000.00 |
| 7 | Muffy   | 24 | Indore    | 10000.00 |
| 6 | Komal   | 22 | MP        | 4500.00 |
| 2 | Khilan  | 25 | Delhi     | 1500.00 |
| 3 | kaushik | 23 | Kota      | 2000.00 |
| 5 | Hardik  | 27 | Bhopal    | 8500.00 |
| 4 | Chaitali | 25 | Mumbai    | 6500.00 |
+-----+-----+-----+-----+

```

9. การจัดกลุ่มข้อมูลตามคอลัมน์ (GROUP BY)

การใช้งาน SQL Select query statement นอกจากจัดเรียงข้อมูลโดย ORDER BY แล้ว ยังสามารถนำเอาข้อมูลทั้งหมดมาทำการจัดกลุ่มข้อมูลที่มีลักษณะซ้ำกัน โดยจะใช้คำสั่งที่เรียกว่า SQL Group by statement

รูปแบบ SQL GROUP BY มักจะใช้งานตามหลัง SQL Select query statement ที่อาจจะใช้ SQL Where Clause ร่วมด้วย และจะใช้นำหน้า SQL ORDER BY หากต้องการจัดเรียงลำดับข้อมูลหลายการจัดกลุ่มด้วย SQL GROUP BY

```
SELECT column1, column2
```

```
FROM table_name
```

```
WHERE [ conditions ]
```

GROUP BY column1, column2

ORDER BY column1, column2;

ตัวอย่างการใช้ GROUP BY สมมติถ้ามีตัวอย่างข้อมูลในตาราง 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 |
+-----+-----+-----+-----+
```

การดึงข้อมูลจากตาราง CUSTOMERS เพื่อแสดงคอลัมน์ NAME รวมถึง SALARY รวมทั้งหมดของ NAME เดียวกันจากการจัดกลุ่ม จะได้เป็น

```
SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS
```

```
GROUP BY NAME;
```

ผลลัพธ์ที่ได้

```
+-----+-----+
| NAME | SUM(SALARY) |
+-----+-----+
| Chaitali | 6500.00 |
| Hardik | 8500.00 |
| kaushik | 2000.00 |
| Khilan | 1500.00 |
| Komal | 4500.00 |
| Muffy | 10000.00 |
| Ramesh | 2000.00 |
+-----+-----+
```

ซึ่งจะเห็นการทำงานของ SQL GROUP BY ได้ชัดเจนยิ่งขึ้น จากตัวอย่างที่มี NAME ซ้ำกันหลายบรรทัด เช่น ตัวอย่างข้อมูลต่อไปนี้

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Ramesh	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	kaushik	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

การดึงข้อมูลจากตาราง CUSTOMERS เพื่อแสดงคอลัมน์ NAME รวมถึง SALARY รวมทั้งหมดของ NAME เดียวกันจากการจัดกลุ่ม

```
SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS
GROUP BY NAME;
```

ผลลัพธ์ที่ได้จะเห็นว่ามีการจัดกลุ่มข้อมูล NAME ที่ซ้ำกันให้เหลือบรรทัดเดียว และทำการรวมค่าในคอลัมน์ SALARY จากคำสั่ง SUM(SALARY)

NAME	SUM(SALARY)
Hardik	8500.00
kaushik	8500.00
Komal	4500.00
Muffy	10000.00
Ramesh	3500.00

2.3 เครื่องมือที่ใช้การสร้างที่เก็บข้อมูล

2.3.1 โปรแกรมดาต้าสเตจ (IBM InfoSphere DataStage)

IBM InfoSphere DataStage เป็นผู้นำทางอุตสาหกรรมเกี่ยวกับ ETL และให้บริการแพลตฟอร์มที่เป็นมัลติ-คลาวด์ที่รวบรวมข้อมูลข้ามระบบหลาย ๆ องค์กร โดยแพลตฟอร์มนี้สามารถปรับขนาดได้ ซึ่งมีคุณสมบัติและความสามารถในเรื่องของ

- เฟรมเวิร์กขนานประสิทธิภาพสูงพร้อมใช้งานที่เป็นออฟไลน์หรือในคลาวด์
- เตรียมการในการพัฒนาได้อย่างรวดเร็ว และง่ายตายหากต้องการทำงานบนคลาวด์
- สามารถขยายงานในการจัดการข้อมูลที่มีขนาดใหญ่มาก และการเชื่อมต่อภายในองค์กร
- ให้ผลที่เพิ่มขึ้นอย่างมากในการสร้างมากกว่าการเข้ารหัส โดยการจัดการความแตกต่างของ

อุปกรณ์ปลายทาง

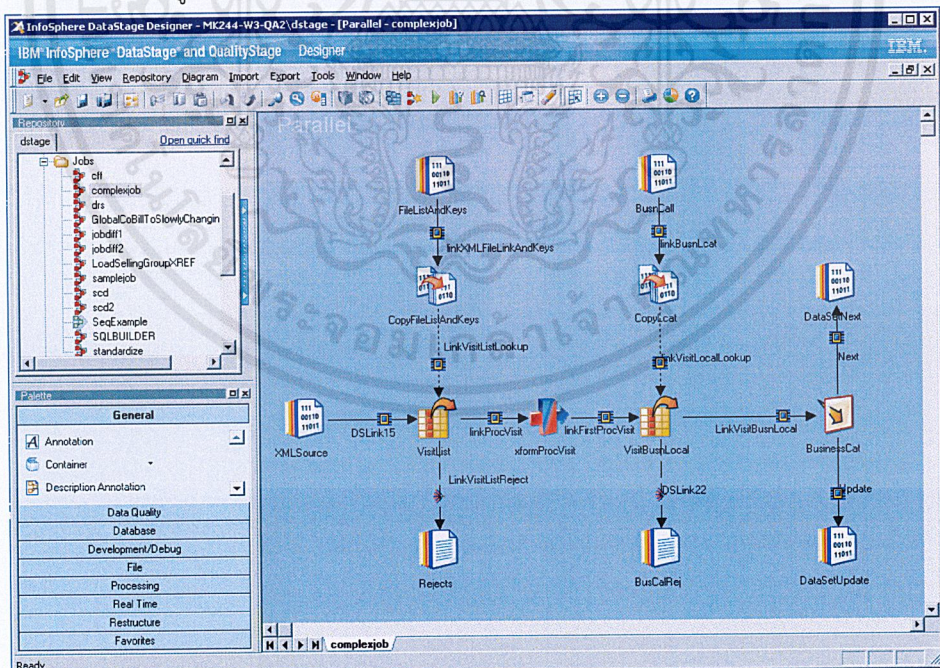
-การรวบรวมข้อมูลที่แตกต่างกัน รวมไปถึงบิกดาต้าที่เก็บไว้ (Hadoop) หรือบิกดาต้าที่มีการเคลื่อนย้ายตลอดเวลา (stream) โดยทั้งสองแบบอยู่บนแพลตฟอร์มแบบกระจายและแบบเมนเฟรม

-สนับสนุน IBM Db2 Z และ Db2 สำหรับ z / OS

-การประยุกต์ใช้กับภาระงาน และกฎเกณฑ์ทางธุรกิจ

-ให้วงจรการพัฒนาที่รวดเร็วโดยใช้การออกแบบอย่างอัตโนมัติ และรูปแบบที่สร้างไว้ล่วงหน้า

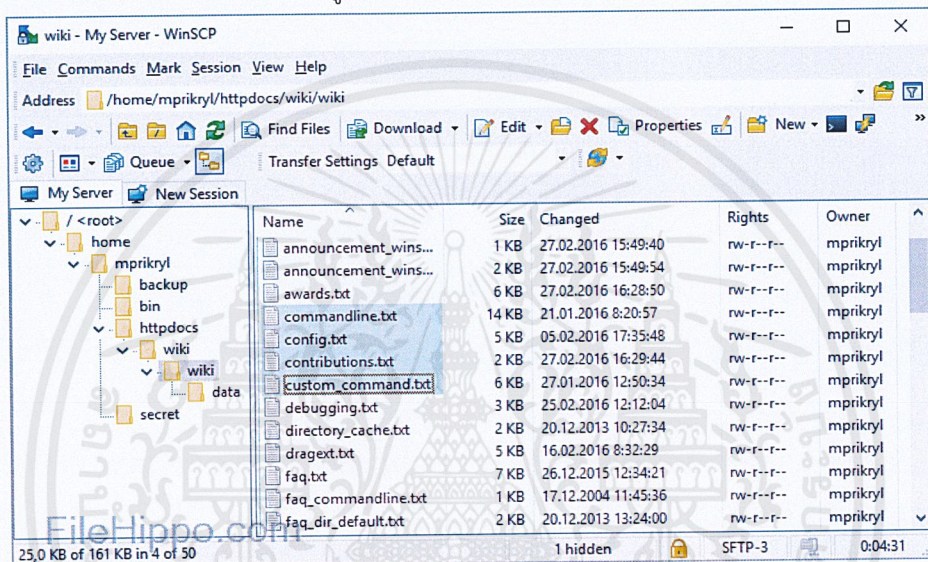
-การรวบรวมข้อมูลแบบเรียลไทม์ และแพลตฟอร์มที่ออกแบบมาเพื่อให้ใช้งานง่าย



รูปที่ 4 หน้าต่างของโปรแกรมดาต้าสเตจ

2.3.2 โปรแกรมวินเอสซีพี (WinSCP)

วินเอสซีพี (WinSCP) เป็นโปรแกรมสำเร็จรูปเอพีทีพี โคลแอนท์ (FTP Client) ที่เอาไว้ใช้ถ่ายโอนข้อมูล คัดลอกไฟล์ข้อมูล อัปโหลด ดาวน์โหลดจากเครื่องลูกข่ายหรือโคลแอนท์ (Client) หรือคอมพิวเตอร์ของผู้ใช้ไปยังเครื่องแม่ข่ายหรือเซิร์ฟเวอร์ (Server) สำหรับผู้ดูแลระบบ หรือดูแลเว็บไซต์ จัดเป็นโปรแกรมเอพีทีพี เอสเอฟทีพี (SFTP) โปรแกรมวินเอสซีพี สามารถดาวน์โหลดไปใช้ได้ฟรีเพราะเป็นโอเพนซอร์ส (Open Source) บนระบบปฏิบัติการวินโดวส์ โดยใช้มาตรฐานเอพีทีพี (FTP : File Transfer Protocol) ในการถ่ายโอนข้อมูล



รูปที่ 5 หน้าต่างการทำงานของโปรแกรมวิน เอสซีพี

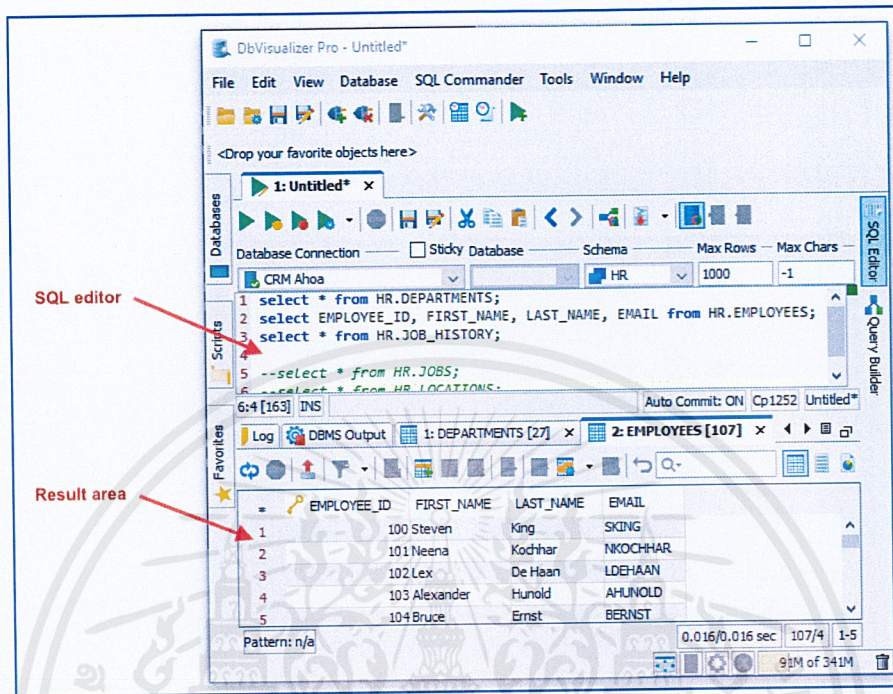
โปรแกรมเอพีทีพี จะแบ่งเป็น 2 ส่วน คือ

1. เอพีทีพี เซิร์ฟเวอร์ (FTP Server) ติดตั้งไว้บนเซิร์ฟเวอร์เพื่อให้บริการโคลแอนท์ที่ติดต่อเข้ามา
2. เอพีทีพี โคลแอนท์ (FTP Client) จะติดตั้งไว้บนเครื่องโคลแอนท์ หรือเครื่องของผู้ใช้ทั่วไปเพื่อติดต่อไปยังเซิร์ฟเวอร์ในการถ่ายโอนข้อมูลจากโคลแอนท์ ไปยังเซิร์ฟเวอร์

2.3.3 ดีบีวีซิวไลเซอร์ (DbVisualizer)

เป็นเครื่องมือฐานข้อมูลสำหรับนักพัฒนา และผู้บริหารฐานข้อมูล ช่วยในการพัฒนาและการบำรุงรักษาฐานข้อมูล นอกจากนี้ยังมีการสนับสนุนอย่างเรียลไทม์ในตัวแก้ไขเอสคิวแอล ภาพการสร้าง หรือแก้ไขตารางการสนับสนุนการดำเนินการของหลายคำสั่งสคริปต์เอสคิวแอล แผนภูมิของข้อมูลการจัดการข้อมูลหยุด หรือซีแอลโอบี (CLOB) ส่งออกสคีมา (Schema) หรือฐานข้อมูลนำเข้าข้อมูลจากไฟล์ซีเอสวี (CSV) สร้างแบบสอบถาม นาวิกเตอร์คีย์ข้อมูลต่างประเทศอธิบายการสนับสนุนแผน และรวมถึงการบริหารจัดการของมุมมอง ทริกเกอร์ ตารางสรุปคำพ้องความหมาย ฟังก์ชันขั้นตอน

ตารางระบบเรียกคอลัมน์คีย์หลักสิทธิ จำกัด อ้างอิง ดีดีแอล (DDL) โปรแกรมโมดูล เพิ่มข้อมูล และพื้นที่ ตาราง



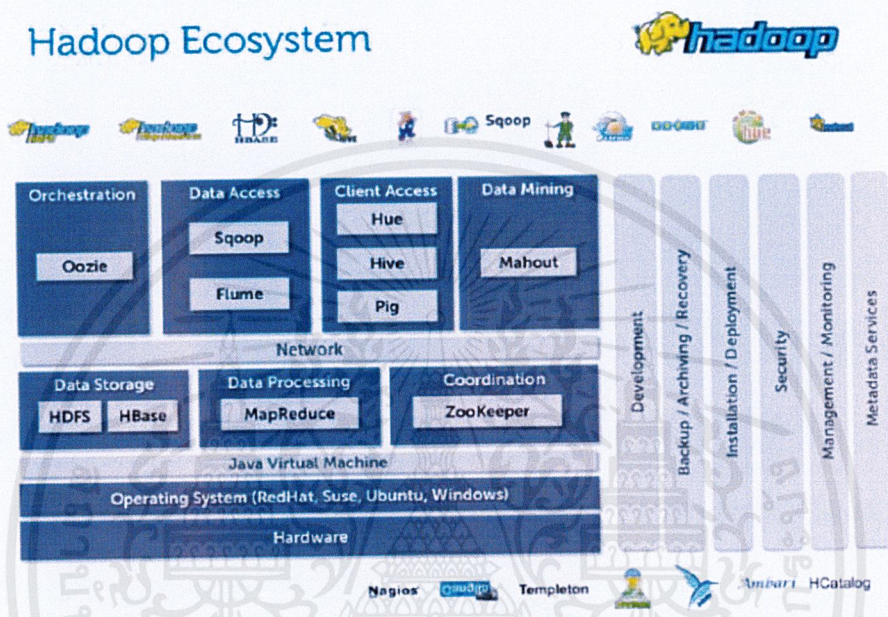
รูปที่ 6 หน้าต่างของดีบีวีซิวไลเซอร์

2.3.4 ฮาดูป (Hadoop)

ฮาดูปเป็นหนึ่งในเครื่องมือ Big Data ที่ได้รับความนิยมอย่างกว้างขวาง เพราะสามารถที่จะจัดการข้อมูลที่ไม่มีโครงสร้างที่ชัดเจน (Unstructured) ขนาดใหญ่ได้ เช่น ข้อมูลที่เป็น Text File XML หรือ JSON และอาจจะมีไฟล์ที่เป็น Web Crawl อยู่ในรูปแบบของไฟล์ Web ARChive (WARC) ซึ่งเป็น Text ขนาดใหญ่ขนาดหลายร้อย TeraByte ซึ่งแน่นอนการจัดการข้อมูลแบบนี้ต้องหาเครื่องมือที่เหมาะสม และ Hadoop ก็คือเครื่องมือเข้ามาช่วยจัดการได้

Hadoop เป็น Open source Project ของ Apache สำหรับการเก็บ และบริหารข้อมูลขนาดใหญ่ ฮาดูปเขียนด้วยโปรแกรมภาษาจาวา มีความสามารถในการทำ Fault Tolerant เพราะจะเก็บข้อมูลซ้ำกันในหลากหลายที่ และเป็นระบบที่เป็น Horizontal Scale ที่รันบนเครื่อง commodity server จำนวนมาก Hadoop Project เริ่มต้นโดย Doug Cutting และ Mike Cafarella ที่เป็นทีมงานของบริษัท Yahoo ซึ่งต่อมาก็มีบริษัทอื่น ๆ นำไปใช้กันอย่างมากมาย ทั้ง eBay Facebook และ Amazon รวมถึงมีบริษัทหลาย ๆ รายที่นำมาฮาดูปมาทำ Commercial Distribution อาทิเช่น Cloudera/MapR/ IBM Infoshpere BigInsight/Hortonwork หรือ Amazon Elastic Map Reduce

ระบบฮาดูป (Hadoop Ecosystem) จะมีองค์ประกอบหลักอยู่แค่สองส่วน คือ HDFS และ Map/Reduce ซึ่งค่อนข้างจะไม่สะดวกกับผู้ใช้งานที่มีความต้องการอื่น ๆ เช่น การประมวลผลโดยใช้ภาษาเอสคิวแอล การเขียน หรืออ่านข้อมูลแบบ Random access หรือการถ่ายโอนข้อมูลจากที่อื่น ดังนั้น จึงมีการพัฒนาโปรเจกต์อื่น ๆ ที่มาทำงานร่วมกับฮาดูป เพื่อให้ได้ประสิทธิภาพดียิ่งขึ้น ดังแสดงตัวอย่างในรูปที่ 7 ซึ่งมีเครื่องมือที่สำคัญดังนี้

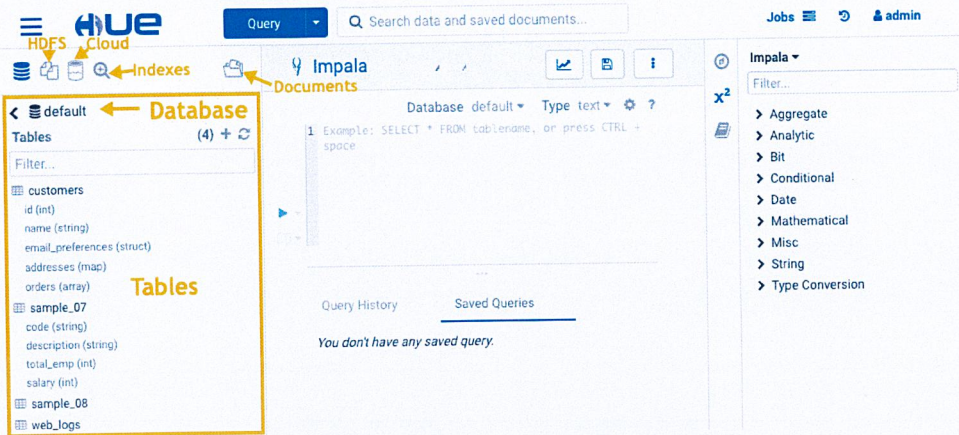


รูปที่ 7 ระบบฮาดูป

เครื่องมือที่ช่วยในการทำงานของโปรเจกต์สทกนี้ จะเป็นเครื่องมือที่ใช้อยู่แล้วภายในบริษัท โดยจะขอยกตัวอย่างเครื่องมือที่ได้ใช้งานจริง ดังนี้

2.3.4.1 ฮิว (HUE)

ฮิว (Hue) ย่อมาจากคำว่า Hadoop User Experience เป็นเครื่องมือช่วยทำ User Interface ของฮาดูป ให้ใช้งานได้ง่ายขึ้นกว่าการต้องใช้ command line ฮิวเป็นหน้าจอ Web UI ของค่าย ทำให้สามารถจัดการ Application ตัวอื่น ๆ เช่น Hive/Pig/Sqoop/Hbase/Files/Security เป็นต้น ผ่านทาง web browser ได้อย่างสะดวก และง่ายกว่าใช้คำสั่ง Command Line



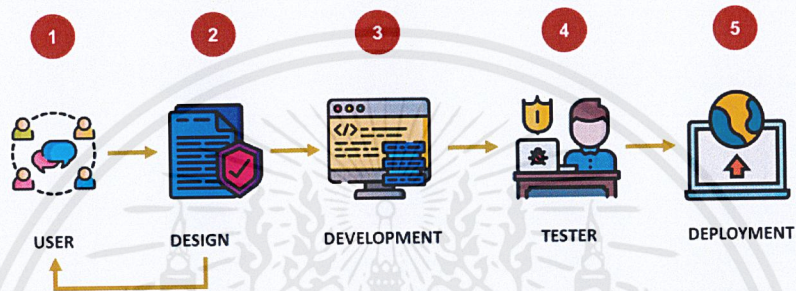
รูปที่ 8 การหน้าต่างทำงานของฮิว



บทที่ 3 วิธีการดำเนินการวิจัย

กระบวนการทำงานนั้น สามารถเขียนเป็นรูปประกอบได้ดังนี้

PROCESS FLOW



รูปที่ 9 โฟลวของกระบวนการทำงาน

3.1 ศึกษาข้อมูล และวิธีการใช้งานโปรแกรมเพื่อสร้างที่เก็บข้อมูล

ขั้นตอนนี้เป็นการศึกษาข้อมูลที่ได้จากความต้องการของผู้ใช้ โดยต้องมีการเก็บรวบรวมข้อมูล และการพูดคุยถึงข้อมูล ให้เป็นไปตามข้อกำหนดระหว่างผู้ใช้งานข้อมูล และนักวิเคราะห์ธุรกิจ (BA) ได้เข้าใจตรงกัน เพื่อให้สามารถใช้งานได้ตามรูปแบบที่วางไว้

ในการสร้างที่เก็บข้อมูลนั้น จำเป็นต้องมีโปรแกรมในการสร้าง โดยโปรแกรมที่ใช้ในการสร้างนั้น ชื่อว่า ดาต้า สเตจ

3.2 สร้างเอกสารเพื่อใช้เป็นคู่มือในการพัฒนาที่เก็บข้อมูล

ในการสร้างที่เก็บข้อมูลนั้น จำเป็นต้องมีเอกสารเพื่อกำหนดคุณลักษณะของข้อมูลให้เป็นไปตามความต้องการของผู้ใช้งาน สำหรับใช้งานในการสร้างที่เก็บข้อมูล โดยเอกสารที่บอกความต้องการของผู้ใช้ นั้นมีลักษณะ ดังรูปล่างข้าง

Field Name	Description	Possible Values	Sample Values
1			
2	Field Name		1234
3	Flag_Status	N/A < 7 N/A > 7 N/ Normal Reschedule 2000 5000 1000 Watch list	
4	AddressRestructuring_F	Flag กำหนดโครงสร้างข้อมูล	N/ Normal Reschedule
5	AddressDelete_F	Flag กำหนดการลบข้อมูล	N/ Normal Reschedule
6	Payment_Bad_201_F	Flag กำหนดค่าเงินคงค้าง	N/ Normal Reschedule
7	Payment_Bad_1201_F	Flag กำหนดค่าเงินคงค้าง	N/ Normal Reschedule
8	FlagScore	Flag Score	200
9	Flag	Flag กำหนดค่าเงินคงค้าง	N/ Normal Reschedule
10	Flag For SM	Flag For SM	N/ Normal Reschedule
11	Flag For PL	Flag For PL	N/ Normal Reschedule
12	Flag For CC	Flag For CC	N/ Normal Reschedule
13	Flag For KIC	Flag For KIC	N/ Normal Reschedule
14	Flag For KPS	Flag For KPS	N/ Normal Reschedule
		E1	
		E2	
		E3	
		E4	

รูปที่ 10 ข้อมูลที่ผู้ใช้งานต้องการ (Requirements)

โดยเอกสารตามรูปที่ 10 จะประกอบไปด้วย

1. ชื่อของข้อมูล หรือที่เรียกว่าชื่อคอลัมน์

โดยข้อมูลในช่องนี้นั้น จะบอกถึงข้อมูล หรือคอลัมน์ที่ผู้ใช้งานต้องการ ตามที่ได้ประชุมร่วมกันกับผู้ที่มีหน้าที่สร้างที่เก็บข้อมูล และการตั้งชื่อคอลัมน์ จำเป็นต้องตั้งชื่อให้เหมาะสมตามรูปแบบที่แต่ละแผนกเข้าใจตรงกัน

2. คำอธิบาย

เพื่อเป็นการบอกว่าข้อมูลในแต่ละคอลัมน์ในแต่ละตัวที่ต้องการใช้นั้น มีความหมายว่าอย่างไร ทำให้คนที่สร้างเอกสารซีปียอร์ สามารถเข้าใจได้ถูกต้องตามที่ตกลงกัน

3. ค่าที่เป็นไปได้

เป็นตัวอย่งค่าของในแต่ละคอลัมน์ เพื่อให้สามารถเข้าใจข้อมูลได้อย่างถูกต้องในแต่ละคอลัมน์ โดยค่าที่เป็นไปได้นั้นจะเป็นตัวบอกว่าข้อมูลควรจะมีชนิดของข้อมูลได้เป็นแบบไหนบ้าง เพื่อที่จะได้สร้างที่เก็บข้อมูลได้ตรงกับข้อมูล

4. ค่าตัวอย่าง

เป็นค่าที่บอกว่า ข้อมูลในแต่ละคอลัมน์นั้นสามารถเป็นค่าอะไรได้บ้าง
 ในการสร้างเอกสารซีปียอร์เพื่อใช้พัฒนาที่เก็บข้อมูลนั้น จำเป็นต้องดูเอกสารความต้องการของผู้ใช้ ตามรูปที่ 10 โดยเอกสารนี้นั้นจะต้องใช้ในการกำหนดค่าต่าง ๆ ของข้อมูล
 รายละเอียดของเอกสารนั้นจะใช้ในการสร้างเอกสารซีปียอร์ ประกอบไปด้วยข้อมูลดังต่อไปนี้

Source Name	src_mnl_tscor_credit_20190731.xlsx				Table Schema	DATATANK
Table (IN)	DATATANK.IN CST CR T SCOR	(Table Path) Shared/EBAN/DATA/MNUL/IN/IP			Table Name	CST CR T SCOR
Physical Table (OUT)	DATATANK.P CST CR T SCOR	(Table Path) Shared/EBAN/DATA/MNUL/OUT/IP			Source App Acronym	MNUL
View Name					Data Level	IP
Partition Key	PTN YYYY, PTN MM					
Format file type	HDFS					
Ingestion Method	Load Append					

รูปที่ 11 การกำหนดแหล่งที่มาของข้อมูล

โดยปกติแล้วเอกสารซีบีอาร์ จะประกอบไปด้วย 3 ส่วน คือ

1. ข้อมูลเริ่มต้นที่มาจากผู้ใช้งาน

		Source			
Source App	Table / File Name	Column Name	Data Type	Size	Mandatory
SPV	src_mnl_tscor_credit_20190731.xlsx	LPM	DECIMAL(10,0)	10	M
SPV	src_mnl_tscor_credit_20190731.xlsx	Flag_Status	STRING	10	O
SPV	src_mnl_tscor_credit_20190731.xlsx	debt restructuring_F	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	defaultdate_F	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Payment_Bad_3M_F	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Payment_Bad_12M_F	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	FicoScore	SMALLINT	5	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Flag	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Fit_For_SM	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Fit_For_HL	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Fit_For_CC	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Fit_For_KEC	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Fit_For_KPL	STRING	1	O
SPV	src_mnl_tscor_credit_20190731.xlsx	T_Score	STRING	5	O
SPV	src_mnl_tscor_credit_20190731.xlsx	PD_T_Score	DECIMAL(10,4)	5	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Limit_Amount	DECIMAL(10,1)	4	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Occupation	STRING	20	O
SPV	src_mnl_tscor_credit_20190731.xlsx	Tier_Income	STRING	20	O

รูปที่ 12 ข้อมูลที่ได้จากฝั่งผู้ใช้งาน

2. ข้อมูล หรือคอลัมน์ที่แสดงบนฐานข้อมูล

Physical Table						
Seq.	Table Name	Column Name	Data Type	Size	Mandatory	Transformation Rules
						Processing Logic:
1	DATATANK.P_CST_CR_T_SCOR	POS_DT	TIMESTAMP	19	M	Current timestamp (YYYY-MM-DD)
2	DATATANK.P_CST_CR_T_SCOR	LPM_CST_ID	DECIMAL(10,0)	10	M	Right trim and direct move
3	DATATANK.P_CST_CR_T_SCOR	AST_CLASS	STRING	15	O	Right trim and direct move
4	DATATANK.P_CST_CR_T_SCOR	DBT_RSTCF_F	STRING	1	O	Right trim and direct move
5	DATATANK.P_CST_CR_T_SCOR	DFLT_DT_F	STRING	1	O	Right trim and direct move
6	DATATANK.P_CST_CR_T_SCOR	DLQ_DYS_3M_F	STRING	1	O	Right trim and direct move
7	DATATANK.P_CST_CR_T_SCOR	DLQ_DYS_12M_F	STRING	1	O	Right trim and direct move
8	DATATANK.P_CST_CR_T_SCOR	FICO_SCORE	SMALLINT	5	O	Right trim and direct move
9	DATATANK.P_CST_CR_T_SCOR	NCB_F	STRING	1	O	Right trim and direct move
10	DATATANK.P_CST_CR_T_SCOR	SM_F	STRING	1	O	Right trim and direct move
11	DATATANK.P_CST_CR_T_SCOR	HL_F	STRING	1	O	Right trim and direct move
12	DATATANK.P_CST_CR_T_SCOR	CC_F	STRING	1	O	Right trim and direct move
13	DATATANK.P_CST_CR_T_SCOR	KEC_F	STRING	1	O	Right trim and direct move
14	DATATANK.P_CST_CR_T_SCOR	KPL_F	STRING	1	O	Right trim and direct move
15	DATATANK.P_CST_CR_T_SCOR	T_SCORE	STRING	5	O	Right trim and direct move
16	DATATANK.P_CST_CR_T_SCOR	PD_T_SCORE	DECIMAL(5,4)	6	O	Right trim and direct move
17	DATATANK.P_CST_CR_T_SCOR	LMT_AMT	DECIMAL(2,1)	4	O	Right trim and direct move
18	DATATANK.P_CST_CR_T_SCOR	CCP	STRING	25	O	Right trim and direct move
19	DATATANK.P_CST_CR_T_SCOR	INOM_LEN	STRING	20	O	Right trim and direct move
20	DATATANK.P_CST_CR_T_SCOR	PPN_TMS	TIMESTAMP	19	M	Current timestamp (YYYY-MM-DD) , mm-MM-SS
21	DATATANK.P_CST_CR_T_SCOR	SRC_STM_ID	STRING	10	M	Set to '710'
22	DATATANK.P_CST_CR_T_SCOR	PTN YYYY	STRING	4	M	Cast POST_DT as String (YYYY)

รูปที่ 13 ข้อมูลที่กำหนดลงฐานข้อมูล

3. แสดงข้อมูลที่ออกฝั่งขาวิว

เป็นข้อมูลที่จะสามารถแสดงได้เฉพาะข้อมูลที่ไม่เป็นความลับ กล่าวคือ ถ้าข้อมูลที่ผู้ใช้งานไม่ต้องการให้แสดง ข้อมูลนั้นจะไม่แสดงผลออกมาได้ ยกตัวอย่าง เช่น ข้อมูลลูกค้าที่เป็นความลับ เฉพาะบุคคล ในกรณีที่มีการขอเรียกดูข้อมูลลูกค้า ข้อมูลบางอย่างจะไม่สามารถแสดงผลออกมาได้ หรือ ข้อมูลอาจมีการทำแฮชซึ่งเกิดขึ้น

แต่เนื่องจากโปรเจกต์ที่ได้รับมอบหมายมานั้น เป็นข้อมูลที่ไม่ได้เป็นความลับของลูกค้า ดังนั้นข้อมูล หรือทุกคอลัมน์นั้นไม่จำเป็นต้องมีการทำแฮชซึ่ง หรือสามารถแสดงข้อมูลลูกค้าได้เลย

View							
Seq.	Schema	Confidential	Column Name	Data Type	Size	Mandatory	Transformation Rules
1	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	POS_DT	TIMESTAMP	19	M	Direct Move
2	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	LPN_CST_ID	DECIMAL(10,0)	10	M	Direct Move
3	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	AST_CLSS_ST	STRING	15	O	Direct Move
4	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	DBT_RSTCF_F	STRING	1	O	Direct Move
5	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	DFLT_DT_F	STRING	1	O	Direct Move
6	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	DLQ_DYS_3M_F	STRING	1	O	Direct Move
7	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	DLQ_DYS_12M_F	STRING	1	O	Direct Move
8	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	FICO_SCOR	SMALLINT	5	O	Direct Move
9	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	NCB_F	STRING	1	O	Direct Move
10	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	SM_F	STRING	1	O	Direct Move
11	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	HLL_F	STRING	1	O	Direct Move
12	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	CC_F	STRING	1	O	Direct Move
13	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	KEC_F	STRING	1	O	Direct Move
14	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	KPL_F	STRING	1	O	Direct Move
15	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	T_SCOR	STRING	5	O	Direct Move
16	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	PD_T_SCOR	DECIMAL(5,4)	6	O	Direct Move
17	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	LMT_AMT	DECIMAL(2,1)	4	O	Direct Move
18	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	OCP	STRING	25	O	Direct Move
19	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	INCH_LEN	STRING	20	O	Direct Move
20	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	PPN_TMS	TIMESTAMP	19	M	Direct Move
21	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	SRC_STM_ID	STRING	10	M	Direct Move
22	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	PTH_YYYY	STRING	4	M	Direct Move
23	DATATANK_VIEW.VP_CST_CR_T_SCOR	No	PTN_MM	STRING	2	M	Direct Move

รูปที่ 14 ตารางข้อมูลในขาวิว

3.3 การสร้างเอกสารดีดีแอล

เป็นเอกสารที่ใช้ในสร้างตารางในฐานข้อมูล โดยใช้ภาษาเอสคิวแอล ในการกำหนดชื่อคอลัมน์ และชนิดของข้อมูลตามเอกสารซีปีอาร์ที่ได้ทำไปก่อนหน้านี้ เพื่อที่จะสามารถนำข้อมูลเข้าไปในพาร์ธอย่างถูกต้องตามที่ได้กำหนดไว้ ดังนั้นการจะสร้างพาร์ธได้นั้นจะต้องมีการสร้างตารางในฐานข้อมูลก่อน ถึงจะสามารถนำดีดีแอลตัวนี้ไปใช้ได้

```

1 CREATE EXTERNAL TABLE DATATANK_BAU_IN_CST_CR_T_SCOR(
2 POS_DT TIMESTAMP,
3 LPH_CST_ID DECIMAL(18,0),
4 AST_CLSS_ST STRING,
5 DBT_RSTCF_F STRING,
6 DFLT_DT_F STRING,
7 DLQ_DYS_3M_F STRING,
8 DLQ_DYS_12M_F STRING,
9 FICO_SCOR SMALLINT,
10 NCB_F STRING,
11 SM_F STRING,
12 HL_F STRING,
13 CC_F STRING,
14 KEC_F STRING,
15 KPL_F STRING,
16 T_SCOR STRING,
17 PD_T_SCOR DECIMAL(5,2),
18 LMT_AMT DECIMAL(2,1),
19 OCP STRING,
20 INCM_LEN STRING,
21 PPH_TSTG TIMESTAMP,
22 SRC_STM_ID STRING
23 )
24
25 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\u0001' LOCATION '/Shared/EBANU/DATA_BAU/HRUL/IN/1P/IN_CST_CR_T_SCOR';

```

รูปที่ 15 การกำหนดค่าเอกสารดีดีแอล

3.4 การกำหนดไฟล์ข้อมูลให้ตรงตามรูปแบบที่กำหนด

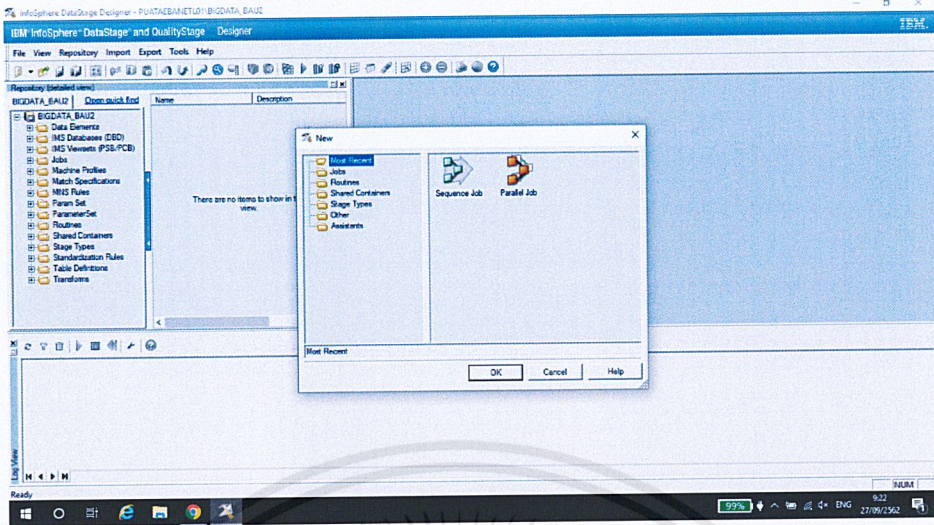
ไฟล์ข้อมูลที่ได้มาจากผู้ใช้อาจจะมีรูปแบบไม่ตรงตามที่กำหนดไว้ ดังนั้นผู้ที่เป็นผู้พัฒนาที่เก็บข้อมูล จำเป็นต้องกำหนดไฟล์ข้อมูลให้ตรงตามที่ต้องการ เนื่องจากเวลาที่นำไฟล์ไปวางไว้ในเซอร์สไฟล์เวิร์ด จะต้องให้ไฟล์ข้อมูลมีรูปแบบที่เหมือนกัน เพราะถ้าหากมีรูปแบบไฟล์ข้อมูลไม่ตรงตามที่กำหนด จะทำให้การอ่านไฟล์เกิดปัญหา และทำให้ข้อมูลไม่สามารถไรท์ลงในฐานะข้อมูลได้ตรงตามที่กำหนดไว้

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R																												
1	1PM	Flag	Status	addr	debt	structuring	F	default	date	F	Payment	Bad	3M	F	Payment	Bad	12M	F	FicoScore	Flag	FR	For	SM	FR	For	HL	FR	For	CC	FR	For	K5C	FR	For	KPL	T	Score	PD	T	Score	Limit	Amount	Occupation	Year	Income
2	2152130	PL	Normal				N				N				N				651	N	Y		Y		Y		Y		Y		Y		Y		6.0	0.04	2	1	Salary	Earnor	15	000	18	000	
3	83528	DMA	Y				N				N				N				760	Y		Y		Y		Y		Y		Y		6.5	0.01	1	Entrepreneur	15	000	18	000	70	000	18	000		
4	36794	PL	Normal				N				N				N				650	Y		Y		Y		Y		Y		Y		6.5	0.065	2	Entrepreneur	Stable	70	000	18	000	70	000	18	000	
5	4879322	PL	Normal				N				N				N				650	Y		Y		Y		Y		Y		Y		6.5	0.040	1	Salary	Earnor	30	000	40	000	30	000	40	000	
6	2839548	PL	Normal				N				N				N				650	Y		Y		Y		Y		Y		Y		6.4	0.120	1	Salary	Earnor	20	000	25	000	20	000	25	000	
7	2632904	PL	Normal				N				N				N				650	Y		Y		Y		Y		Y		Y		6.1	0.002	1	Salary	Earnor	++500	300	++500	300	++500	300			
8	160946	PL	Normal				N				N				N				615	Y		Y		Y		Y		Y		Y		6.0	0.014	4	Entrepreneur	15	000	18	000	15	000	18	000		
9	3930130	PL	Normal				N				N				N				615	Y		Y		Y		Y		Y		Y		6.0	0.014	4	Entrepreneur	15	000	18	000	15	000	18	000		
10	5564616	PL	Normal				N				N				N				600	Y		Y		Y		Y		Y		Y		6.1	0.267	1	Entrepreneur	15	000	18	000	15	000	18	000		
11	78195	Flag	Y				Y				Y				Y				600	Y		Y		Y		Y		Y		Y		6.5	0.062	1	Entrepreneur	15	000	18	000	15	000	18	000		
12	2095237	PL	Normal				Y				Y				Y				600	Y		Y		Y		Y		Y		Y		6.5	0.040	2	Entrepreneur	15	000	18	000	15	000	18	000		
13	1122281	PL	Normal				Y				Y				Y				600	Y		Y		Y		Y		Y		Y		6.5	0.062	1	Entrepreneur	15	000	18	000	15	000	18	000		
14	2651802	UDR	Y				Y				Y				Y				585	Y		Y		Y		Y		Y		Y		6.1	0.267	1	Entrepreneur	15	000	18	000	15	000	18	000		

รูปที่ 16 ไฟล์ข้อมูลที่ได้จากผู้ใ้

3.5 สร้างที่เก็บข้อมูลด้วยโปรแกรมดาต้า สเตจ

การสร้างที่เก็บข้อมูลด้วยโปรแกรมดาต้า สเตจนั้น จะแบ่งออกเป็น 2 จ๊อบ คือ เป็นพาราล์ลอลจ๊อบ (Parallel Job) และซีควเอนซ์ จ๊อบ (Sequence Job)

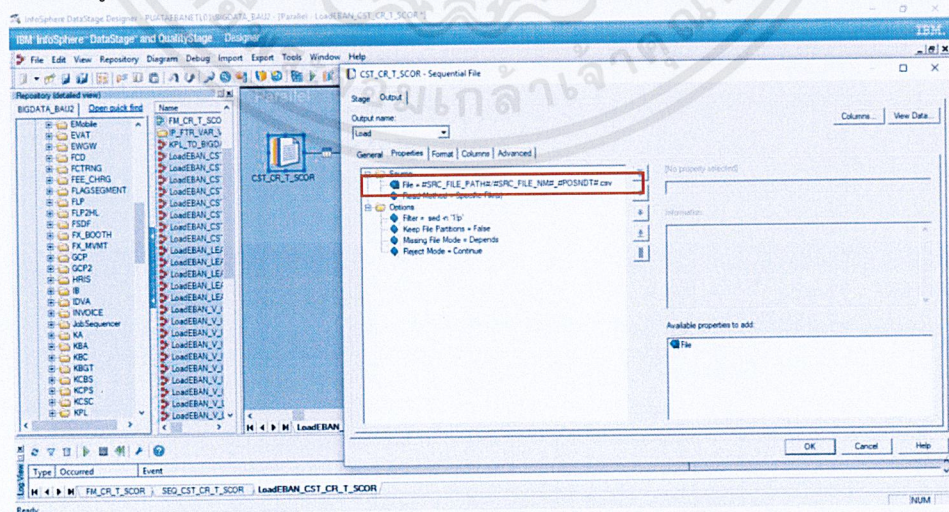


รูปที่ 17 หน้าต่างจ๊อบในโปรแกรมดาต้า สเตจ

3.5.1 การสร้างพารัลเลล จ๊อบ

3.5.1.1 สร้างตัวกำหนดการอ่านไฟล์ข้อมูล

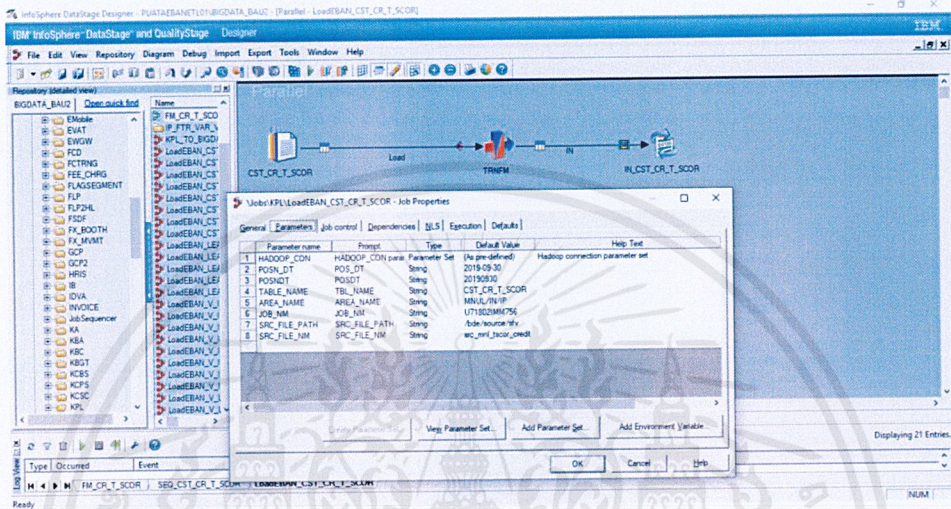
การสร้างจ๊อบนี้เป็นการสร้างที่เก็บข้อมูลซึ่งให้อ่านไฟล์ข้อมูล โดยดึงจากพาร์ธที่อยู๋ในซอร์สไฟล์เวอร์ลที่ใต้ใส่ไว้ โดยไฟล์ข้อมูลที่มีชื่อพาร์ธตามที่กำหนดไว้ โปรแกรมดาต้า สเตจ จะไปอ่านไฟล์ที่มีชื่อพาร์ธตามที่กำหนดไว้ผ่านตัวแปร การกำหนดพาร์ธให้อ่านผ่านตัวแปร เนื่องจากเวลาที่ต้องการแก้ไขไฟล์ข้อมูล จะทำให้ไม่ต้องแก้ไข หรือเปลี่ยนแปลงหลาย ๆ จุด และในการอ่านไฟล์ข้อมูลนั้น จำเป็นต้องกำหนดให้โปรแกรมอ่านไฟล์โดยให้ตัดบรรทัดหัวและท้าย เพื่อให้อ่านเฉพาะข้อมูลที่ต้องการเท่านั้น และเนื่องจากข้อมูลเป็นไฟล์เท็กซ์ ดังนั้นต้องมีการกำหนดตัวค้นด้วยไปป์ (Pipe) เพื่อที่จะสามารถตัดข้อมูลได้อย่างถูกต้อง



รูปที่ 18 หน้าต่างการกำหนดการอ่านไฟล์ข้อมูล

3.5.1.2 กำหนดตัวแปร

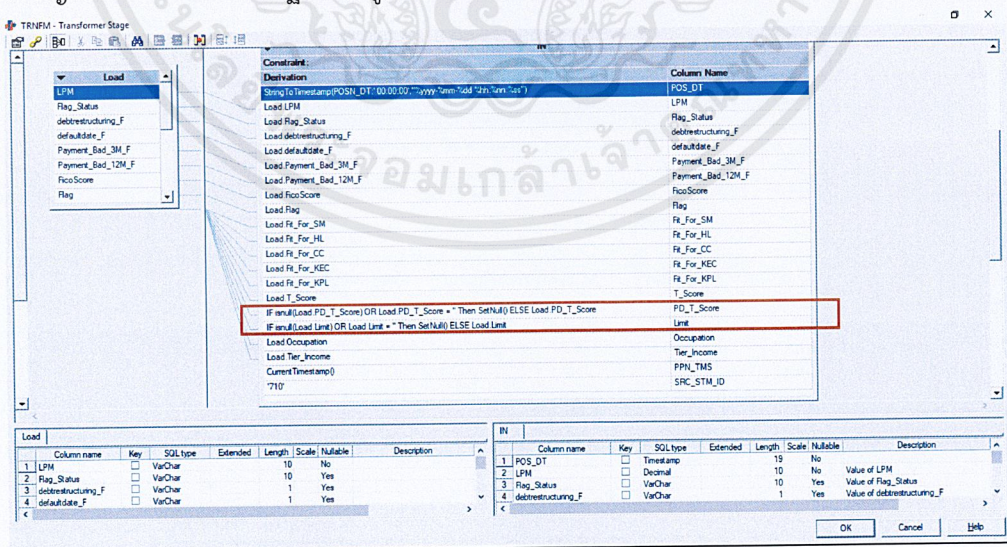
ค่าที่ต้องการจะใช้ จำเป็นต้องกำหนดตัวแปรเมื่อเวลาเรียกใช้ค่า ก็สามารถใช้ชื่อตัวแปรตามที่กำหนดไว้ได้ ซึ่งการกำหนดตัวแปรไว้นั้น เนื่องจากเวลาที่ต้องการแก้ไข หรือเปลี่ยนแปลงค่า สามารถมาแก้ไขได้จากที่กำหนดไว้นี้ โดยการกำหนดตัวแปรไว้ เพราะต้องการจะใช้งานค่าตัวแปรนี้ในโปรแกรมหลาย ๆ จุด



รูปที่ 19 หน้าต่างการกำหนดตัวแปร

3.5.1.3 กำหนดเงื่อนไขให้กับค่าของข้อมูล

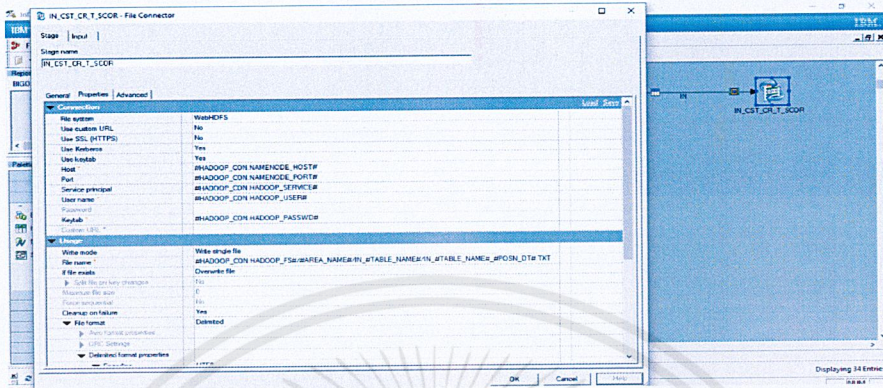
เนื่องจากข้อมูลที่ได้จากไฟล์ข้อมูลนั้น มันมีค่าที่เป็นค่าว่าง (Null) ดังนั้นเพื่อไม่ให้เกิดปัญหาในการไรท์ลงฐานข้อมูล จำเป็นต้องเขียนโค้ดในการจัดการกับข้อมูลที่มีค่าว่าง



รูปที่ 20 หน้าต่างการกำหนดเงื่อนไขให้กับข้อมูลที่มีค่าว่าง

3.5.1.4 โหลดข้อมูลลงขาเข้าในฐานข้อมูล

เนื่องจากได้สร้างพาร์ตชาเข้าเป็นไฟล์เท็กซ์จากการสร้างดีดีแอลไปยังต้น เวลาที่ต้องการจะโหลดข้อมูลลงขาเข้า จำเป็นต้องสร้างพาร์ตให้ตรงกันทั้งจากฮิวและโปรแกรมตาต้า สเตจ

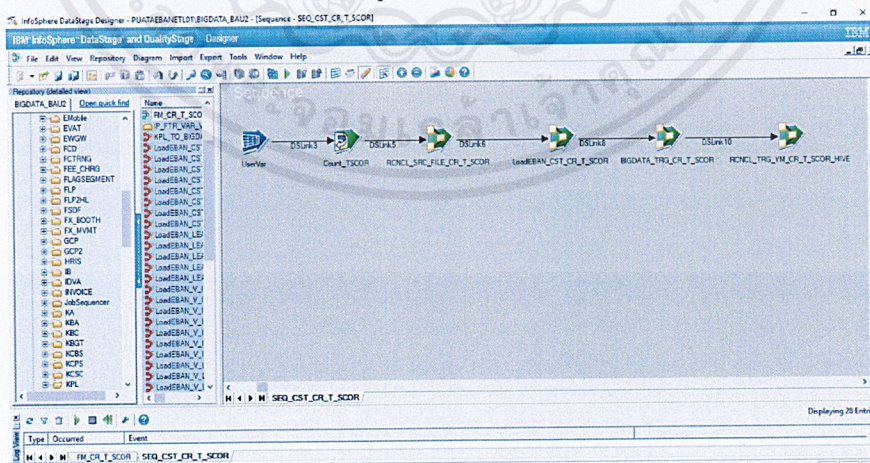


รูปที่ 21 หน้าต่างการกำหนดข้อมูลค่าลงขาเข้า

3.5.2 การสร้างซีควเอนซ์ จ๊อบ

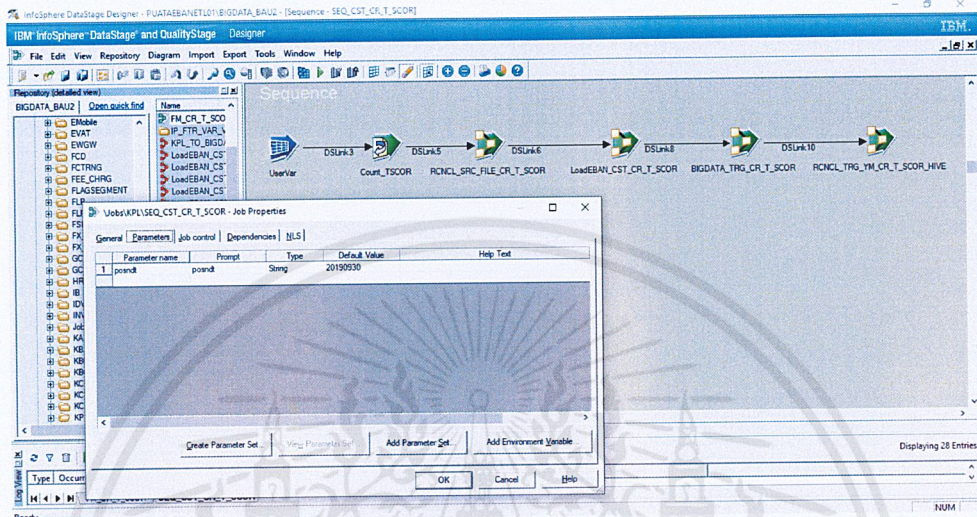
3.5.2.1 สร้างจ๊อบข้างนอกเพื่อดึงข้อมูลจากขาเข้าโหลดลงขาออก

จากข้อ 3.5.1 ที่ได้สร้างพาร์ตเลล จ๊อบ เพื่อที่จะโหลดข้อมูลจากซอร์สลงขาเข้า ขั้นตอนต่อไปนี้จะเป็นการนำข้อมูลจากขาเข้าโหลดลงขาออก เพื่อที่จะนำข้อมูลไปเก็บไว้ โดยขาออกนี้จะมีขั้นตอนที่เรียกว่า เร็กคองไชน์ (Reconcile) คือ การตรวจสอบความเท่ากันของจำนวนข้อมูลในขาเข้าและขาออกว่ามีเท่ากันหรือไม่ และในขาออกนี้จะเก็บข้อมูลที่มีการแบ่งพาร์ติชัน (Partition) เพื่อเวลาที่ต้องการข้อมูลของช่วงเวลาหนึ่ง ๆ ย้อนหลัง ก็ไม่จำเป็นต้องหาข้อมูลทั้งฐานข้อมูล แต่สามารถระบุวันที่ต้องการได้เลย ซึ่งจะช่วยลดระยะเวลาในการหาข้อมูลลง

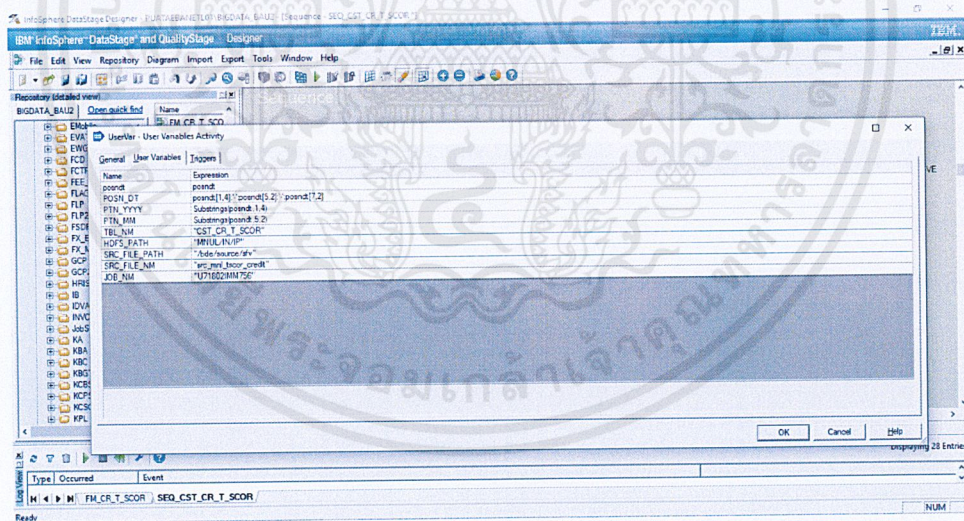


รูปที่ 22 หน้าต่างของซีควเอนซ์ จ๊อบ

จากรูปที่ 23 จะสังเกตได้ว่า มีการทำเรีกคอนโซลค์นทุกครั้งหลังจากที่ทำจ็อบย่อย เพราะต้องการจะตรวจสอบความถูกต้องของข้อมูลทุกครั้งหลังจากที่มีการโหลดข้อมูลลงไป และเนื่องจากการเรียกใช้ค่าที่ควบคุมจ็อบ ดังนั้นจำเป็นต้องมีการกำหนดค่าของตัวแปรให้เรียกใช้งานภายในจ็อบ

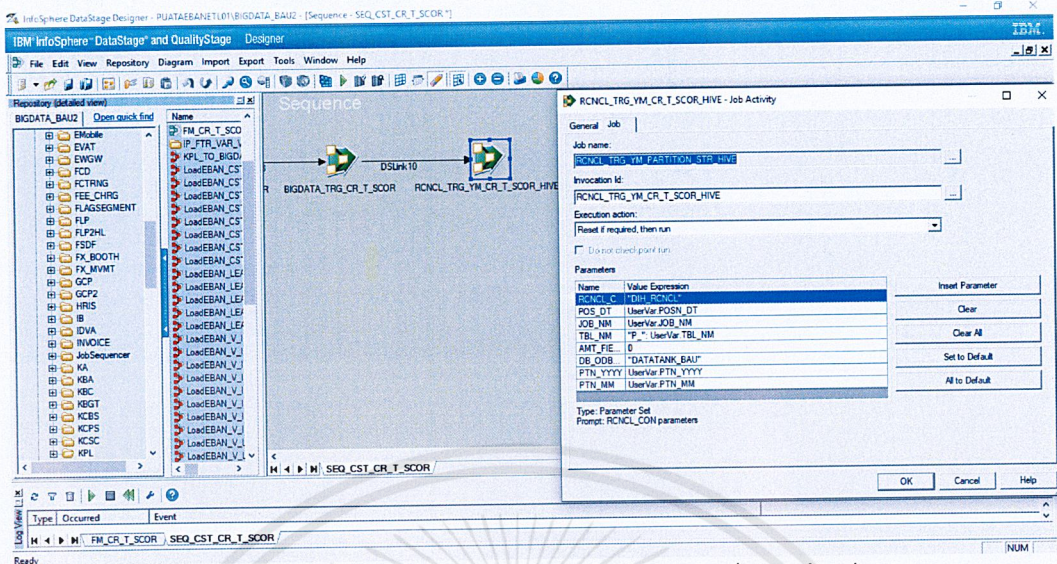


รูปที่ 23 ค่าโกลบอลพารามิเตอร์



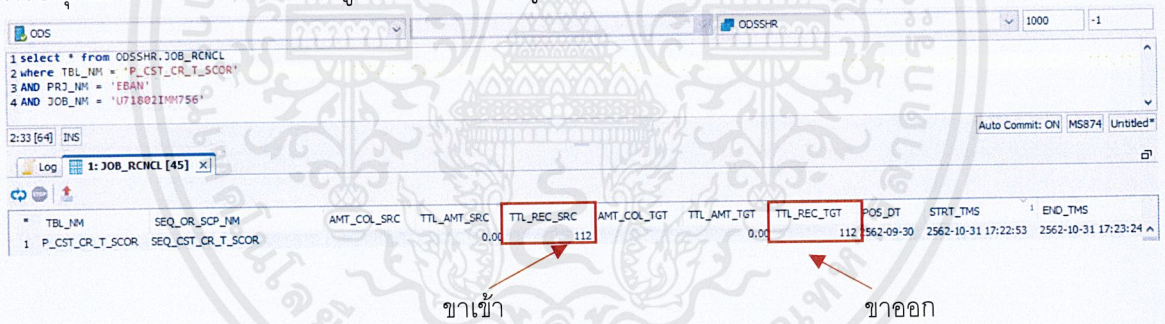
รูปที่ 24 ค่าวาเรียเบิล (Variable)

ถ้ามีการกำหนดค่าวาเรียเบิลแล้ว ขั้นตอนต่อไปต้องมาทำเรีกคอนโซลค์นเพื่อตรวจสอบความเท่ากันของข้อมูล โดยการนำเรีกคอนโซลค์นออกนี้ จำเป็นต้องมีการแบ่งพาร์ทิชัน เนื่องจากข้อมูลมีปริมาณมาก เวลาต้องการดูข้อมูลย้อนหลัง จะสามารถเลือกเป็นช่วงเวลาได้ ซึ่งทำให้การคิวรี่ข้อมูลนั้นเร็วยิ่งขึ้น



รูปที่ 25 การกำหนดพารามิเตอร์ในจ็อบย่อยเรียกคอนโซล

3.6 การทดสอบความเท่ากันของข้อมูลในขาเข้าและขาออก
 เป็นการตรวจสอบให้แน่ใจว่า ข้อมูลที่ไหลตเข้าทั้งขาเข้าและขาออกนั้น เข้าไปอยู่ในฐานข้อมูลครบทุกแถวตามจำนวนของข้อมูลที่ได้รับมาจากผู้ใช้งาน



รูปที่ 26 การตรวจสอบจำนวนข้อมูลในขาเข้าและขาออก

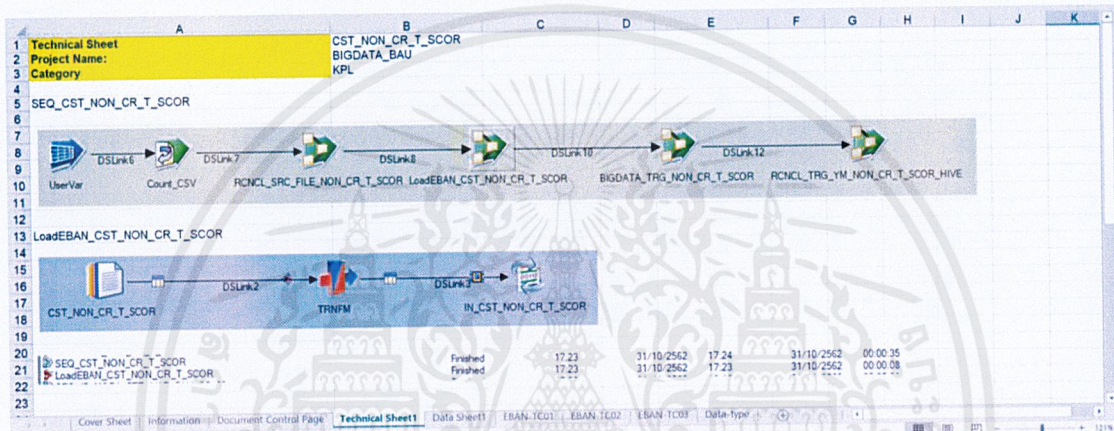
ซึ่งการตรวจสอบจำนวนข้อมูลนั้น จำเป็นต้องตรวจสอบทั้งขาเข้าที่ไหลตข้อมูลจากต้นทางลงฐานข้อมูลในขาเข้า และจำเป็นต้องตรวจสอบจำนวนข้อมูลในขาออกด้วย เพราะทั้งสองขานี้เป็นข้อมูลตัวเดียวกันที่จะต้องมึค่าของข้อมูลเท่ากัน

3.7 สร้างเอกสารยูเอเอส

เป็นเอกสารที่ใช้ประกอบให้กับคนที่มีความหน้าที่ทดสอบข้อมูลที่ได้รับมา และข้อมูลที่ไหลเข้าฐานข้อมูลมีจำนวนเท่ากันหรือไม่ เพราะในการสร้างที่เก็บข้อมูล จำเป็นต้องสร้างให้ทุกคนสามารถคิวรีเพื่อทดสอบข้อมูลที่สร้างนั้นเท่ากัน และข้อมูลต้องไม่มีความผิดพลาดในการคิวรี ซึ่งเอกสารยูเอเอส ประกอบไปด้วยซีทย่อย ดังนี้

1. จ๊อบในดาต้า สเตจ

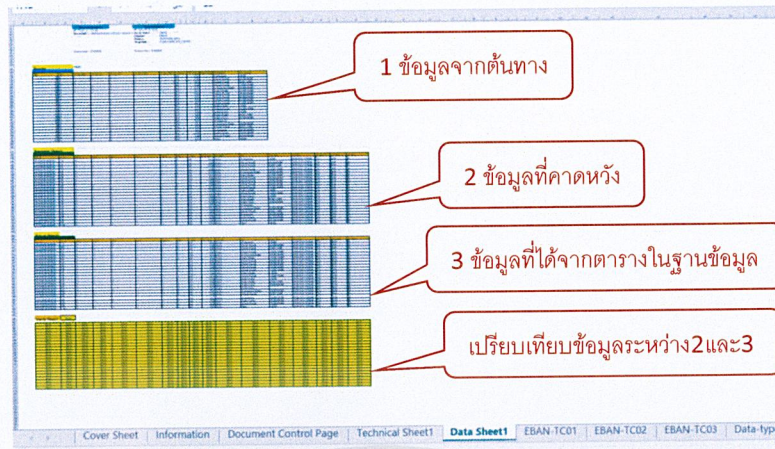
ซึ่งซีทนี้นั้นมันต้องการจะบอกว่า จ๊อบที่ทำไว้มีอะไรบ้าง และแต่ละจ๊อบรันจบกระบวนการแล้วหรือยัง



รูปที่ 27 ซีทจ๊อบในดาต้าสเตจ

2. ข้อมูลในตาราง

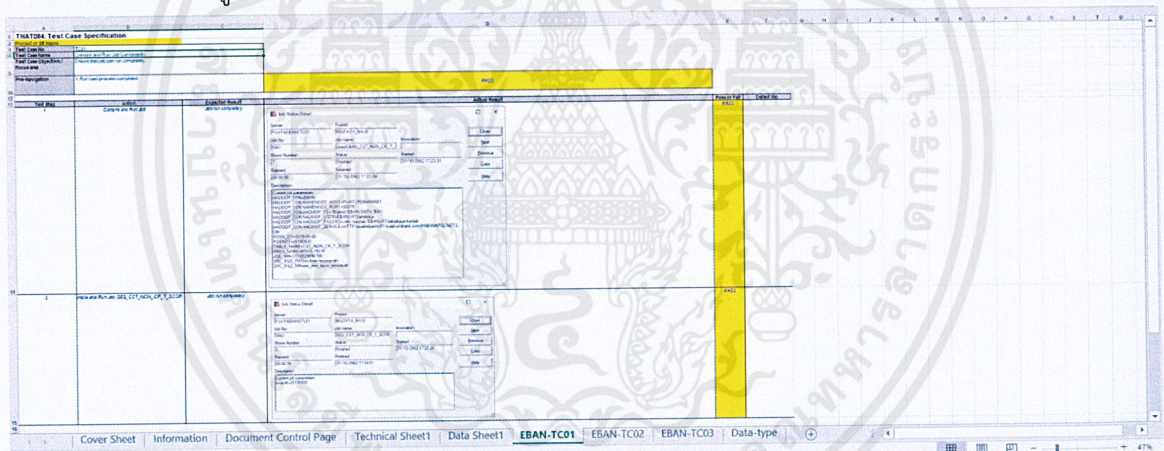
โดยปกติแล้วข้อมูลที่อยู่ในขาเข้า จะเป็นข้อมูลที่มาจากต้นทางโดยตรงมีแถวและคอลัมน์ที่เป็นเหมือนกันกับต้นทาง แต่สำหรับข้อมูลในขาออกนั้นจะเป็นข้อมูลที่ผู้ใช้งานต้องการ ซึ่งบางคอลัมน์ที่เพิ่มเข้ามาจะใช้บอกเรื่องเวลา พาร์ทิชัน และข้อมูลที่เข้ามานั้นเข้ามาจากไอทีของซอร์สอะไร ซึ่งข้อมูลที่นำมาในแต่ละแถวเอามาเปรียบเทียบเพียงบางส่วน เพื่อต้องการทดสอบให้แน่ใจว่า ข้อมูลนั้นตรงกัน



รูปที่ 28 ซิทตัวอย่างข้อมูลในตาราง

3. การแสดงหน้าต่างการรันจ็อบในตาต้า สเตจ

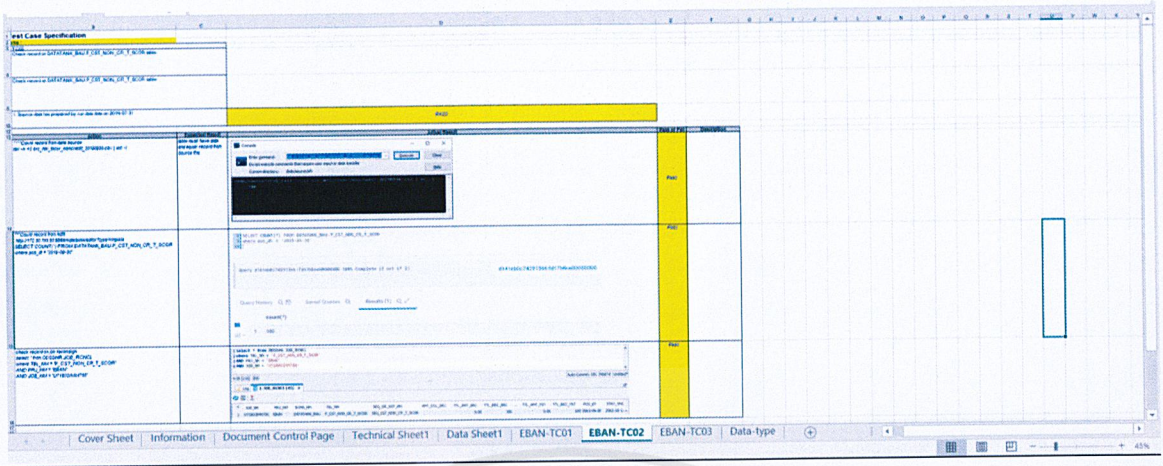
ซิทหน้านี้เป็นการแสดงของหน้าต่างเพื่อตรวจสอบว่า รันจ็อบสำเร็จหรือไม่ แล้วจ็อบนั้นสามารถรันจ็อบเข้ากับพาร์ธที่กำหนดไว้



รูปที่ 29 ซิทหน้าสถานะรายละเอียดการรันจ็อบ

4. การแสดงความเท่ากันของข้อมูลจากต้นทาง ข้อมูลขาเข้า และข้อมูลจากขาออก

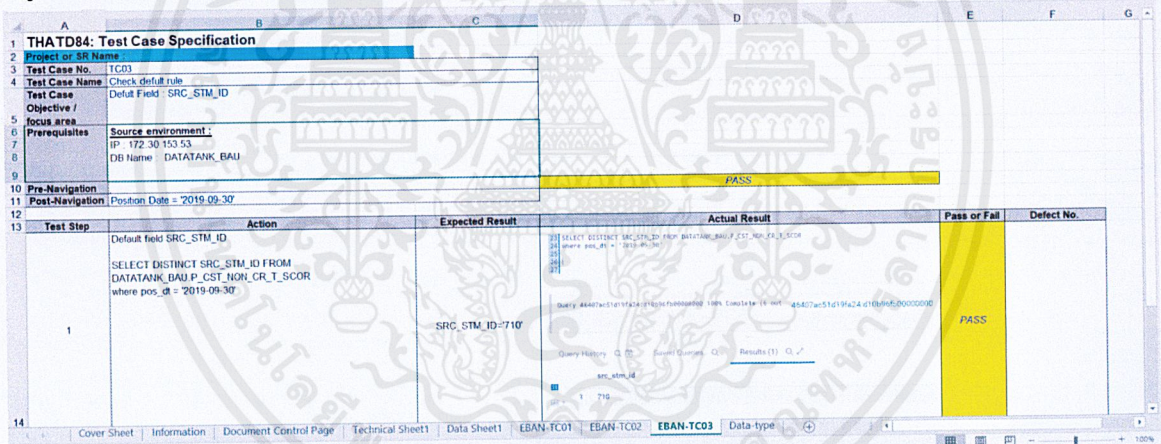
เป็นการตรวจสอบจำนวนแถวของข้อมูล โดยจะเปรียบเทียบจากต้นทาง และเอาไว้ตรวจสอบกับข้อมูลของขาเข้าและขาออก ซึ่งจำเป็นต้องมีข้อมูลเท่ากับข้อมูลจากต้นทาง



รูปที่ 30 ซีทการทดสอบจำนวนแถวของข้อมูล

5. การตรวจสอบข้อมูลจากต้นทาง

ซีทหน้านี้เป็นหน้าที่แสดงเลขไอดีของข้อมูลต้นทางเพื่อตรวจสอบว่า ข้อมูลที่นำมานั้น อยู่ในไอดีหมายเลขใด



รูปที่ 31 ซีทการตรวจสอบไอดีของข้อมูลต้นทาง

6. การตรวจสอบชื่อคอลัมน์ และประเภทของข้อมูล

ในการโหลดข้อมูลลงฐานข้อมูล จำเป็นต้องมีการตรวจสอบดูว่า คอลัมน์ที่กำหนดในเอกสารซีปีอาร์นั้น ตรงกับที่สร้างตารางในฐานข้อมูลหรือไม่

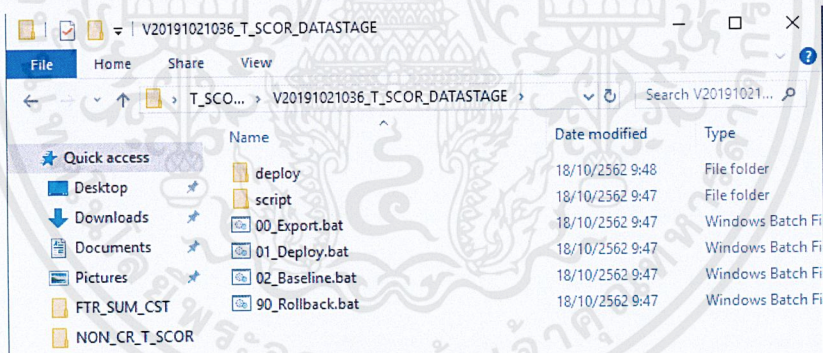
Formula: =IF(TRIM(C7)~"INTEGER",IF(UPPER(H7)~"INT","TRUE","FALSE"),IF(TRIM(C7)~UPPER(H7),"TRUE","FALSE"))

Field Name	Data Type	Field Name	Data Type	Validation
POS_DT	TIMESTAMP	pos_dt	timestamp	TRUE TRUE
IP_ID	INTEGER	ip_id	int	TRUE TRUE
NCB_F	STRING	ncb_f	string	TRUE TRUE
SM_F	STRING	sm_f	string	TRUE TRUE
HL_F	STRING	hl_f	string	TRUE TRUE
CC_F	STRING	cc_f	string	TRUE TRUE
KEC_F	STRING	kec_f	string	TRUE TRUE
KPL_F	STRING	kpl_f	string	TRUE TRUE
T_SCOR	STRING	t_scor	string	TRUE TRUE
PD_T_SCOR	DECIMAL(5,4)	pd_t_scor	decimal(5,4)	TRUE TRUE
LMT_AMT	DECIMAL(2,1)	lmt_amt	decimal(2,1)	TRUE TRUE
OCP	STRING	ocp	string	TRUE TRUE
INCM_LEN	STRING	incm_len	string	TRUE TRUE
PPN_TMS	TIMESTAMP	ppn_tms	timestamp	TRUE TRUE
SRC_STM_ID	STRING	src_stm_id	string	TRUE TRUE
PTN YYYY	STRING	ptn_yyyy	string	TRUE TRUE
PTN_MM	STRING	ptn_mm	string	TRUE TRUE

รูปที่ 32 ซึ่การทดสอบความถูกต้องของชื่อคอลัมน์ และประเภทข้อมูล

3.8 เตรียมข้อมูลขึ้นระบบจริง

เนื่องจากงานที่ได้ทำไปนั้นเป็นงานที่เปรียบเสมือนพื้นที่สำหรับทดลองงาน ซึ่งจะสามารถแก้ไขและเปลี่ยนแปลงงานได้ แต่หลังจากที่ทดสอบความถูกต้องของงานแล้ว จำเป็นต้องนำงานที่ได้สร้างไว้ ขึ้นโปรดักชัน (Production) หรือระบบงานจริง โดยต้องมีการทำรายการในการขึ้นโปรดักชันเพื่อให้คนที่ทำหน้าที่ในการเตรียมขึ้นระบบงานจริงได้ทดลองรันจ็อบก่อนขึ้นงานจริง



รูปที่ 33 รายการในการขึ้นโปรดักชัน

บทที่ 4

ผลการวิจัย

จากผลการสร้างที่เก็บข้อมูลในฐานข้อมูล ซึ่งข้อมูลที่ได้ไหลดเข้าไปนั้นเป็นคะแนน ซึ่งเรียกว่า ทีสกอร์ โดยเป็นข้อมูลส่วนหนึ่งในการคัดเลือกลูกค้าเบื้องต้นที่มีความต้องการในการขอสินเชื่อผ่านทางดิจิทัล และการขอสินเชื่อผ่านทางช่องทางดิจิทัลนี้ เรียกว่า ดิจิทัล เลนดิ่ง การคัดเลือกลูกค้าจำเป็นต้องดูข้อมูลอื่น ๆ ประกอบหลายอย่าง เพื่อให้ได้ลูกค้าที่มีคุณสมบัติเหมาะสมต่อการขอสินเชื่อจริง ๆ ซึ่งในโปรเจกต์ที่ได้รับมอบหมายนี้เป็นการคัดเลือกข้อมูลแบบเบื้องต้น โดยจะรู้แค่ลูกค้ามีเงินเดือนเท่าไร หรือลูกค้านั้นมีการค้างชำระสินเชื่อหรือไม่ เพื่อที่จะคัดเลือกลูกค้าก่อนที่จะนำข้อมูลเหล่านี้ไปจัดกลุ่มของลูกค้า

จากการไหลดข้อมูลลงในฐานข้อมูลนั้น จะแสดงรายละเอียดของข้อมูลลูกค้าแบบเบื้องต้น ซึ่งข้อมูลที่แสดงอยู่นี้จะถูกใช้ในโปรเจกต์ของลาซาด้า (Lazada) โดยทีสกอร์นี้จะแบ่งเป็นลูกค้าออกเป็น 2 กลุ่ม และเกณฑ์ที่ใช้ในการคำนวณคะแนนนั้นใช้วิธีการคิดที่แตกต่างกัน เท่ากับว่าในฐานข้อมูลที่เก็บตารางทีสกอร์นั้น จะแบ่งออกเป็น 2 ตาราง ซึ่งทีสกอร์นั้นจะแบ่งออกเป็น

4.1 เครดิต ทีสกอร์ (Credit T Score)

จะเป็นข้อมูลของบุคคลที่มีเครดิตกับธนาคารกสิกร มีการขอสินเชื่อกับธนาคาร โดยข้อมูลตัวนี้นั้นจะเรียกว่า แอลพีเอ็ม (LPM)

4.2 นอน เครดิต ทีสกอร์ (Non Credit T Score)

เป็นข้อมูลที่ตรงข้ามกับแอลพีเอ็ม คือจะเป็นข้อมูลของลูกค้าที่ไม่ได้เป็นลูกหนี้ของธนาคารกสิกร กล่าวคือเป็นบุคคลที่ไม่ได้มีการขอสินเชื่อ ซึ่งข้อมูลของบุคคลเหล่านี้ เรียกว่า ซีไอเอส (CIS)

โดยปกติแล้ว ตารางทั้งสองนี้จะแสดงข้อมูลเบื้องต้นของลูกค้าคล้ายกัน แต่ตารางทั้งสองนั้นมีข้อแตกต่างตรงที่ตารางของนอน เครดิต ทีสกอร์จะไม่มีแฟล็กที่คอยตรวจสอบประวัติการค้างหนี้ หรือข้อมูลที่เกี่ยวข้องกับสินเชื่อ เนื่องจากข้อมูลในนี้จะเป็นข้อมูลที่ลูกค้ายังไม่มีการทำธุรกรรมประเภทสินเชื่อที่มีเครดิตกับทางธนาคาร

จากรูปภาพข้างล่างนี้เป็นตัวอย่างตารางของเครดิต ทีสกอร์ ซึ่งจะแสดงรายละเอียดเบื้องต้นที่เป็นแฟล็ก และมีการตรวจสอบสถานะของระดับลูกค้าในแต่ละบุคคล

pos_dt	lpm_cst_id	ast_cls_st	dbt_rstcf_f	dflt_d_f	dliq_dys_3m_f	dliq_dys_12m_f	fico_scor	ncb_f	sm_f	hl_f	cc_f	kec_f
1	2019-09-30 00:00:00	PL Normal	N	N	N	N	758	Y	Y	Y	Y	Y
2	2019-09-30 00:00:00	PL Normal	N	N	N	N	724	Y			Y	Y
3	2019-09-30 00:00:00	PL Normal	N	N	N	N	683	Y	Y		Y	Y
4	2019-09-30 00:00:00	PL Normal	N	N	N	N	651	Y	Y	Y	Y	Y
5	2019-09-30 00:00:00	PL Normal	N	N	N	N	763	Y	Y	Y	Y	Y
6	2019-09-30 00:00:00	PL Normal	N	N	N	N	723	Y	Y	Y	Y	Y
7	2019-09-30 00:00:00	PL Normal	N	N	N	N	737	Y	Y	Y	Y	Y
8	2019-09-30 00:00:00	PL Normal	N	N	N	N	654	Y	Y	Y	Y	Y
9	2019-09-30 00:00:00	PL Normal					0	N	Y	Y	Y	Y
10	2019-09-30 00:00:00	PL Normal	N	N	N	N	680	Y	Y	Y	Y	Y
11	2019-09-30 00:00:00	PL Normal					0	N	Y	Y	Y	Y
12	2019-09-30 00:00:00	PL Normal	N	N	N	N	751	Y	Y	Y	Y	Y
13	2019-09-30 00:00:00	PL Normal					0	N	Y	Y	Y	Y
14	2019-09-30 00:00:00	PL Normal	N	N	N	N	681	Y	Y	Y	Y	Y

รูปที่ 34 ตารางของที สคอร์ (1)

kpl_f	t_scor	pd_t_scor	lmt_amt	ocp	incm_len	ppn_tms	src_stm_id	ptn_yyyy	ptn_mm
Y	E8	0.0240	NULL	Salary Stable	50,000-69,999	2019-10-31 17:22:59	710	2019	09
Y	E6	0.0450	NULL	Entrepreneur Stable	100,000-249,999	2019-10-31 17:22:59	710	2019	09
Y	E7	0.0300	NULL	Entrepreneur Stable	20,000-29,999	2019-10-31 17:22:59	710	2019	09
Y	E7	0.0300	NULL	KBank Payroll	20,000-29,999	2019-10-31 17:22:59	710	2019	09
Y	E9	0.0120	NULL	Entrepreneur	20,000-29,999	2019-10-31 17:22:59	710	2019	09
Y	E10	0.0040	NULL	Entrepreneur Stable	30,000-49,999	2019-10-31 17:22:59	710	2019	09
Y	E10	0.0040	NULL	KBank Payroll	30,000-49,999	2019-10-31 17:22:59	710	2019	09
Y	E9	0.0120	NULL	Salary Stable	20,000-29,999	2019-10-31 17:22:59	710	2019	09
Y	E11	0.0020	NULL			2019-10-31 17:22:59	710	2019	09
Y	E7	0.0300	NULL	KBank Payroll	20,000-29,999	2019-10-31 17:22:59	710	2019	09
Y	E11	0.0020	NULL			2019-10-31 17:22:59	710	2019	09
Y	E10	0.0040	NULL	Salary Earner	20,000-29,999	2019-10-31 17:22:59	710	2019	09
Y	E11	0.0020	NULL			2019-10-31 17:22:59	710	2019	09
Y	E10	0.0040	NULL	Entrepreneur	20,000-29,999	2019-10-31 17:22:59	710	2019	09

รูปที่ 35 ตารางของที สคอร์ (2)

ในการให้สินเชื่อแก่ลูกค้า นั้น ข้อมูลที่จะใช้ในการให้สินเชื่อจะต้องเอาข้อมูลนี้ไปดูควบคู่กับไฟโอสคอร์ (Fico Score) เพื่อดูเครดิตของลูกค้า

ขณะนี้นางขึ้นนี้ได้ถูกนำขึ้นระบบจริง และเริ่มให้บริการแล้ว โดยเป็นการให้สินเชื่อออนไลน์ในแคมเปญ ชื่อว่า “Lazada X Kbank” ซึ่งเป็นการให้บริการแก่ลูกค้าที่เป็นผู้ขายในลาซาด้า และจำเป็นต้องมีบัญชีที่ผูกกับธนาคารกสิกร แม้ว่าที่ผ่านมาจะมีหลายธนาคารที่ร่วมมือกันสร้างแคมเปญนี้ แต่นี่ถือเป็นธนาคารแรกที่มีการนำปุ่มกดขอสินเชื่อเข้าไปอยู่บนแพลตฟอร์มของร้านค้าจริง ๆ ก่อนจะเชื่อมต่อมาสู่แอป K PLUS เพื่อดำเนินการขอวงเงิน และรับส่งข้อมูลเพื่อวิเคราะห์พฤติกรรมจากข้อมูลของลาซาด้า

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

เพื่อให้สอดคล้องและเท่าทัน ปัจจุบันทางธนาคารกสิกรต้องการจะนำระบบดิจิทัลเข้ามาใช้ในงานภายในธนาคารให้ได้มากที่สุด โดยเฉพาะการก้าวเข้าไปหาลูกค้าด้วยตัวเอง และความพยายามของธนาคารในการเอาตัวไปอยู่ที่ ๆ มีผู้ค้าใช้งานจริง ซึ่งจะช่วยให้ลูกค้าไม่พลาดสิทธิประโยชน์ หรือสามารถต่อยอดธุรกิจโดยไม่ต้องเสียเวลาตามหา พร้อมทั้งช่วยผลักดันศักยภาพการแข่งขันของธุรกิจ ขณะเดียวกัน ธนาคารก็ได้ฐานลูกค้าเพิ่มขึ้นด้วย

ดิจิทัล เชนด์ตั้งนั้น ถูกนำมาเข้ามาช่วยในการแก้ปัญหา และเพิ่มประสิทธิภาพให้กับธนาคารในการคัดกรองลูกค้า กล่าวคือการนำระบบดิจิทัลมาใช้ภายในธนาคารจะช่วยให้การใช้งานมีความสะดวก และรวดเร็วมากขึ้น ทั้งยังช่วยให้ธนาคารสามารถคัดกรองบุคคลที่ต้องการขอสินเชื่อได้อย่างถูกต้องตรงกับเงื่อนไขที่กำหนดไว้

จากการทำโปรเจกต์สหกิจในครั้งนี้ ที่เกี่ยวข้องกับทิสเตอร์เป็นเพียงส่วนหนึ่งที่ใช้ในการคัดเลือกบุคคลที่มีความต้องการในการขอสินเชื่อ โดยการคัดเลือกบุคคลเบื้องต้นนั้น ต้องมีการเตรียมข้อมูลก่อนที่จะส่งให้โมเดลเลอร์ในการสร้างโมเดลสำหรับจัดกลุ่มของบุคคลที่ต้องการขอสินเชื่อต่อไป

แต่จากการวิเคราะห์และพิจารณาข้อมูลเบื้องต้นนั้น ทำให้ทราบว่า บุคคลที่ต้องการขอสินเชื่อกับทางธนาคารจะต้องเป็นบุคคลที่ไม่มีการค้างหนี้ ต้องไม่มีหนี้เสีย ไม่มีประวัติที่ส่งผลกระทบต่อทางธนาคาร และบุคคลที่ต้องการขอสินเชื่อจำเป็นต้องมีฐานเงินเดือนที่มั่นคง สามารถที่จะชำระหนี้ให้กับธนาคารได้

สำหรับลูกค้าที่มีความสนใจต้องการจะขอสินเชื่อ ทางธนาคารจะมีวิธีการเสนอให้กับลูกค้าที่มีความประสงค์จริง ๆ ซึ่งการเสนอสินเชื่อให้กับบุคคลที่ต้องการนั้น จะช่วยให้ธนาคารเสนอให้กับลูกค้าได้ตรงตามกลุ่มเป้าหมายอย่างถูกต้อง

5.2 ข้อเสนอแนะ

จากการทำโปรเจกต์สหกิจในครั้งนี้ สิ่งที่พบในงานดิจิทัล เชนด์ตั้ง คือ การให้สินเชื่อแก่บุคคลเฉพาะกลุ่ม เช่น เป็นบุคคลที่ต้องเป็นพ่อค้า แม่ค้าในลาซาต้าอย่างเดียว ซึ่งจริง ๆ แล้วแคมเปญนี้มีจุดประสงค์เพื่อต้องการให้สินเชื่อแบบรวดเร็ว และสะดวกต่อการทำธุรกรรมได้อย่างทันท่วงที แต่ในขณะเดียวกันสำหรับบุคคลทั่วไปที่ต้องการนำเงินมาใช้จ่ายภายในครัวเรือน แต่ไม่มีช่องทางในการขอ

สินเชื่อที่ได้อย่างรวดเร็ว หรือมีแต่อาจจะมียอดดอกเบี้ยที่แพงจนถึงดอกเบี้ยที่มีค่าเท่าเงินกู้ ซึ่งทำให้เพิ่มภาระในการขอสินเชื่อ และเกิดความลำบากในการตัดสินใจมากขึ้น

ดังนั้น ถ้าแคมเปญนี้ประสบความสำเร็จแล้ว อยากให้ทางธนาคารขยายรูปแบบการขอสินเชื่อรูปแบบอื่น ๆ มากขึ้น เพื่อลดปัญหาการกู้ยืมในระบบ หรือการกู้ยืมจากบุคคลใกล้ชิด



เอกสารอ้างอิง

- [1] IBM. “Delivers advanced enterprise ETL”. [Online].
<https://www.ibm.com/products/infosphere-datastage> ค้นข้อมูลเมื่อวันที่ 20 ธันวาคม 2562
- [2] บริษัทโพร์เอ็กซ์ตรีม จำกัด. “ETL”. [Online]. <http://www.4x-treme.com/etl/?lang=th>
ค้นข้อมูลเมื่อวันที่ 12 กันยายน 2562
- [3] บริษัทคอนเทนท์ เบเกอร์ จำกัด. “Lazada x Kbank กู้เงินรู้ผลใน 1 นาที”. [Online].
<https://www.moneybuffalo.in.th/ธุรกิจและเศรษฐกิจ/lazada-x-kbank-กู้เงินรู้ผลใน-1-นาที> ค้นข้อมูลเมื่อวันที่ 20 ธันวาคม 2562
- [4] อาริรัตน์ วงศ์สุวรรณ. “การบริหารจัดการข้อมูลขนาดใหญ่ Big Data โดยใช้ซอฟต์แวร์ Hadoop”. [Online]. <http://www.ubu.ac.th/blog/areerat-153> ค้นข้อมูลเมื่อวันที่ 12 กันยายน 2562
- [5] ธนชาติ นุ่มนนท์. “Hadoop Ecosystem สำหรับการพัฒนา Big Data”. [Online].
<https://thanachart.org/2014/10/18/hadoop-ecosystem-สำหรับการพัฒนา-big-data/>
ค้นข้อมูลเมื่อวันที่ 12 กันยายน 2562
- [6] คุณทวีโชค ปละใจ. “ETL”. [Online]. <https://sites.google.com/site/ectlearning/home/etl-extract-transform-and-load> ค้นข้อมูลเมื่อวันที่ 12 กันยายน 2562
- [7] Suphakit Annopornchai. “SQL คืออะไร Structured Query Language คือ คำสั่งบริหารจัดการ database”. [Online]. <https://saixiii.com/sql-statement/> ค้นข้อมูลเมื่อวันที่ 18 ตุลาคม 2562
- [8] “WinSCP (วินเอสซีพี) คืออะไร เป็นโปรแกรมสำเร็จรูป FTP Client”. [Online].
<https://www.mindphp.com/คู่มือ/73-คืออะไร/4135-what-is-winscp.html> ค้นข้อมูลเมื่อวันที่ 18 ตุลาคม 2562
- [9] “DbVisualizer”. [Online]. <http://th.softwaredownload.com/apps/download-dbvisualizer-for-windows-7-os.html> ค้นข้อมูลเมื่อวันที่ 18 ตุลาคม 2562
- [10] “DbVisualizer Portable”. [Online]. <http://th.softwaredownload.com/database-software/download-dbvisualizer-portable-for-windows.html> ค้นข้อมูลเมื่อวันที่ 18 ตุลาคม 2562
- [11] Suphakit Annopornchai. “รวมคำสั่ง SQL Command พื้นฐานเบื้องต้น (Basic SQL Tutorial)”. [Online]. <https://saixiii.com/sql-command/> ค้นข้อมูลเมื่อวันที่ 18 ตุลาคม 2562

[12] “เกี่ยวกับเรา”. [Online]. <http://www.kbtg.tech/th> ค้นข้อมูลเมื่อวันที่ 6 กันยายน 2562

[13] Techsauce Team. “ขอกู้เงินด่วนผ่าน Lazada x KBank เชื่อมต่อ App to App ไม่ต้องใช้เอกสารอนุมัติไว 1 นาทีรับเงิน”. [Online]. <https://techsauce.co/news/kbank-lazada-online-loan> ค้นข้อมูลเมื่อวันที่ 26 ธันวาคม 2562



ประวัติผู้เขียน

หัวข้อโครงการ การคัดเลือกข้อมูลเบื้องต้นของลูกค้า สำหรับการกู้ยืมเงินผ่านระบบดิจิทัล

ชื่อ - สกุล นางสาวกิตติมา กาวิละ

รหัสนักศึกษา 59010129

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

สาขาวิชา วิศวกรรมสารสนเทศ

ประวัติส่วนตัว

วันเดือนปีเกิด 15 สิงหาคม 2540

ที่อยู่ 54/49 ไลฟ์ สุขุมวิท62 ถนนสุขุมวิท แขวงพระโขนงใต้ เขตพระโขนง กรุงเทพฯ 10260

ประวัติการศึกษา

2559 - ปัจจุบัน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (ระดับปริญญาตรี)

2553 - 2558 โรงเรียนบุญวาทย์วิทยาลัย (ระดับมัธยมศึกษา)

2546 - 2552 โรงเรียนอนุบาลห้างฉัตร (ระดับประถมศึกษา)