1. <u>เกริ่นนำเกี่ยวกับธุรกิจและระบบสารสนเทศที่สนใจ</u>

➢ <u>ธุรกิจ SCB คืออะไร?</u>

รูปภาพที่ 1.1 – โลโก้ SCB หรือ ธนาคารไทยพาณิชย์

SCB หรือธนาคารไทยพาณิชย์ เป็นธนาคารแห่งแรกของประเทศไทย ที่มีบทบาทสำคัญในการให้บริการทางการเงินภายในประเทศ โดยในปัจจุบัน มีสาขาภายในประเทศมากกว่า 800 สาขา ซึ่งในปัจจุบันธนาคารไทยพาณิชย์ได้ให้บริการทางการเงินครบวงจร ไม่ว่าจะเป็น “การเปิดบัญชีเงินฝาก” , “การทำบัตรเดบิต/เครดิต” , “การลงทุนภายในและต่างประเทศ” , “บริการสินเชื่อ” , “บริการแลกเปลี่ยนเงินตราระหว่างประเทศ” , “Mobile Banking” , “CardX” เป็นต้น

SCB ในปัจจุบัน นอกเหนือจากการให้บริการทางการเงินที่ธนาคารสาขาต่าง ๆ ทั่วประเทศแล้ว ยังสามารถทำบริการทางการเงินผ่าน Application “SCB Easy” ได้อีกด้วย ซึ่งสามารถทำได้ทั้งการธุรกรรม รวมถึงขอข้อมูลการลงทุนและสินเชื่อต่าง ๆ เปรียบเหมือน**ธนาคารออนไลน์**ที่อยู่เพียงปลายนิ้วคุณ อีกทั้งยังมีบริการเว็บไซต์ <u>https://www.scb.co.th/</u> ที่ช่วยอำนวยความสะดวกให้กับลูกค้าในด้านของการหาข้อมูล โดยที่ไม่ต้องไปถามที่ธนาคารได้อีกด้วย

ซึ่งในกรณีศึกษานี้ จะมุ่งเน้นไปที่การศึกษาในกลุ่มลูกค้าทั่วไป ที่มีการใช้ Application “SCB Easy” เนื่องจากเป็นสิ่งที่เพิ่มความสะดวกสบายให้กับลูกค้าและมักใช้ในชีวิตประจำวัน รวมถึงมีกระบวนการทางธุรกิจที่น่าสนใจในหลาย ๆ มุมอีกด้วย

2. <u>กระบวนการทางธุรกิจ (Business processes) และ กฎเกณฑ์ทางธุรกิจ (Business Rules)</u>

ในธุรกิจด้านธนาคาร มีกระบวนการทางธุรกิจที่เกี่ยวข้องกับการให้บริการทางการเงินแก่ลูกค้า หลากหลายกระบวนการ ไม่ว่าจะเป็นกระบวนการสินเชื่อ , กระบวนการเปิดบัญชี , กระบวนการลงทุน และ 1 ในกระบวนการทางธุรกิจที่มีความสำคัญในธุรกิจธนาคาร นั่นก็คือ **กระบวนการที่เกี่ยวข้องกับการทำธุรกรรมต่าง ๆ**

โดยในกรณีศึกษานี้ จะเจาะลึกไปที่ Mobile Banking ของทาง SCB เป็นหลัก ซึ่งจะหยิบยกมาทั้งหมด **3 กระบวนการ**ด้วยกัน ได้แก่

1. กระบวนการทำธุรกรรมโอนเงินผ่าน Mobile Banking

2. กระบวนการสมัครบัตรเดบิตผ่าน Mobile Banking
3. กระบวนการถอนเงินไม่ใช้บัตรผ่าน Mobile Banking

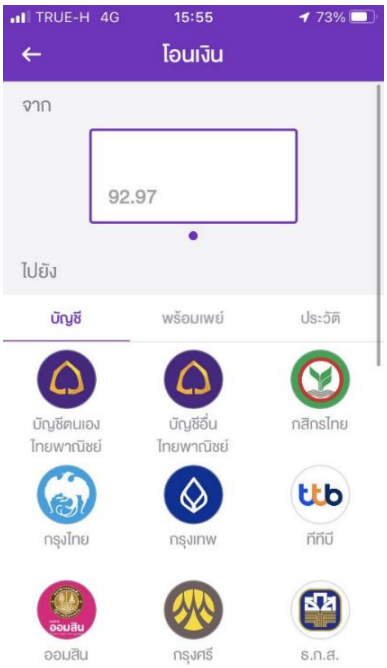
➢ <u>กระบวนการที่ 1 : กระบวนการทำธุรกรรมโอนเงินผ่าน Mobile Banking</u>

กระบวนการทำธุรกรรมโอนเงินผ่าน Mobile Banking ประกอบด้วยกระบวนการ ได้แก่ 1) การเลือกธนาคารของบัญชีปลายทาง 2) การกรอกข้อมูลเลขบัญชีและจำนวนเงิน 3) การตรวจสอบข้อมูลและยืนยันการทำธุรกรรม 4) การทำธุรกรรมระหว่างบัญชี

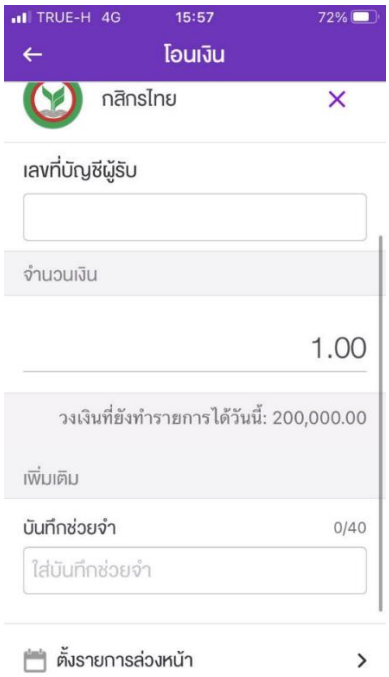
ลูกค้าสามารถทำธุรกรรมโอนเงินไปยังบัญชีธนาคารใดก็ได้ โดยไม่มีค่าธรรมเนียมเพิ่มเติม เริ่มต้นจากการเลือกธนาคารของบัญชีปลายทาง (รูปภาพที่ 1)

เมื่อลูกค้าเลือกธนาคารเสร็จสิ้น จึงจะสามารถกรอกข้อมูลเลขบัญชีปลายทางและจำนวนเงินที่ต้องการโอน ซึ่งการโอนในรูปแบบนี้ สามารถตั้งเวลาการโอนล่วงหน้าได้ (รูปภาพที่ 2) จากนั้นจะเข้าสู่กระบวนการตรวจสอบข้อมูลและยืนยันการทำธุรกรรม (รูปภาพที่ 3)

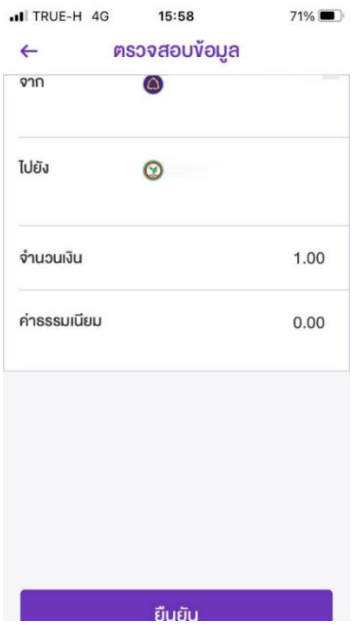
เมื่อลูกค้ากดยืนยัน ระบบของทางธนาคารจะทำธุรกรรมไปยังบัญชีปลายทางให้อัตโนมัติ และลูกค้าจะได้หลักฐานเป็น E-Slip พร้อม QR Code ในการยืนยันว่าเป็นการทำธุรกรรมจริง (รูปภาพที่ 4)



รูปภาพที่ 2.1 : การเลือกธนาคารของบัญชีปลายทาง



รูปภาพที่ 2.2 : การกรอกเลขบัญชีปลายทางและจำนวนเงินที่ต้องการ (สามารถเลือกตั้งรายการล่วงหน้าได้)



รูปภาพที่ 2.3 : การตรวจสอบข้อมูลและยืนยันการทำธุรกรรม

รูปภาพที่ 2.4 : หลักฐานยืนยันการทำธุรกรรม

กฎเกณฑ์ทางธุรกิจ (Business Rules) ของกระบวนการทำธุรกรรมโอนเงินผ่าน Mobile Banking

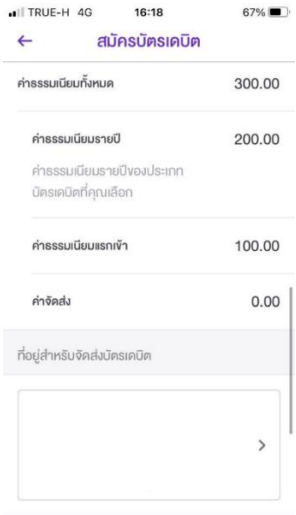
1. **ลูกค้า** สามารถ**ทำธุรกรรมการโอนเงิน** ได้อย่างไม่จำกัดครั้ง
   และในแต่ละครั้ง**ธุรกรรมการโอนเงิน**สามารถถูก**ทำ**ได้โดย**ลูกค้า**
2. การทำธุรกรรมแต่ละครั้ง**ลูกค้า**สามารถ**เลือกประเภทของธุรกรรมโอนเงิน** (บัญชี / พร้อมเพย์) ได้แค่ 1 ประเภท
   การทำธุรกรรมแต่ละครั้ง**ประเภทการทำธุรกรรมโอนเงิน**สามารถถูก**เลือก**โดย**ลูกค้า**เพียง 1 คน
3. การทำธุรกรรมแต่ละครั้ง**ลูกค้า**สามารถ**เลือกธนาคาร**ของบัญชีปลายทางได้เพียง 1 ธนาคาร
   การทำธุรกรรมแต่ละครั้ง**ธนาคาร**ของบัญชีปลายทางสามรถถูก**เลือก**โดย**ลูกค้า**ได้เพียง 1 ธนาคาร
4. **ธุรกรรมโอนเงิน**แต่ละครั้งจะ**โอน**ไปยัง**บัญชีปลายทาง**ได้เพียง 1 บัญชี
   บัญชีปลายทางแต่ละบัญชีสามารถถูก**โอน**จากการทำ**ธุรกรรมโอนเงิน**ได้หลายครั้ง
5. ในแต่ละครั้ง**ลูกค้า**จะต้อง**ระบุ**จำนวน**เงิน** 1 จำนวน
   ในแต่ละครั้ง**เงิน** 1 จำนวนจะต้องถูก**ระบุ**โดย**ลูกค้า**
6. แต่ละครั้ง**ลูกค้า**จะต้อง**ระบุ**จำนวน**เงิน**อย่างน้อย 1 สตางค์
7. **ประเภทการทำธุระกรรมโอนเงิน**บางประเภทจะ**มี**ค่าธรรมเนียมการทำ**ธุรกรรมโอนเงิน**
8. ในแต่ละครั้ง**ลูกค้า**จะ**ได้รับ**ข้อมูลประวัติการทำ**ธุรกรรมโอนเงิน**เพียง 1 ธุรกรรม
   ในแต่ละครั้งการทำ**ธุรกรรมโอนเงิน**จะมีประวัติ**ให้ลูกค้า**ได้เพียง 1 ประวัติ
9. **บัญชี** 1 บัญชีสามารถ**ขอประวัติธุรกรรมการโอนเงิน**ได้หลายครั้ง
   ประวัติการทำธุรกรรมการโอนเงิน 1 ประวัติสามารถถูก**ขอ**ได้โดย**บัญชี**เพียง 1 บัญชี
10. **ธุรกรรมการโอนเงิน**จะต้อง**มี**ข้อมูลประกอบไปด้วย transfer date, transfer_id, Sender acc, remain_balance, Reciever acc, transfer_fee, transfer amount

➢ กระบวนการที่ 2 : กระบวนการสมัครบัตรเดบิตผ่าน Mobile Banking

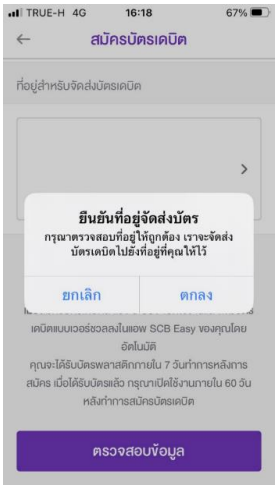
กระบวนการสมัครบัตรเดบิตผ่าน Mobile Banking ประกอบด้วยกระบวนการ ได้แก่ 1) การเลือกชนิดบัตรเดบิตที่ลูกค้าต้องการสมัคร 2) การกรอกข้อมูลที่อยู่จัดส่งบัตรเดบิต 3) การตรวจสอบและยืนยันข้อมูลการสมัครบัตร 4) การชำระเงินค่าทำบัตร 5) การจัดส่งสินค้า

ลูกค้าสามารถสมัครบัตรเดบิตผ่านแอปพลิเคชั่น SCB Easy ได้ โดยการเลือกชนิดบัตรเดบิตที่ลูกค้าต้องการ (รูปภาพที่ 1) เมื่อลูกค้าเลือกเสร็จสิ้น จะต้องกรอกข้อมูลที่อยู่จัดส่งบัตร (รูปภาพที่ 2) จากนั้นจะเข้าสู่กระบวนการตรวจสอบข้อมูลที่อยู่ และค่าทำบัตร (รูปภาพที่ 3) และเมื่อตรวจสอบเสร็จแล้ว ลูกค้าจะต้องกดยืนยัน เพื่อดำเนินการชำระค่าทำบัตร และหลังการชำระเงินเสร็จสิ้น ลูกค้าจะได้รับอีเมล์ยืนยันการสมัครบัตรเดบิตพร้อมรายละเอียดการสมัครบัตร เช่น หมายเลขบัตร, ประเภทบัตร, ชื่อ-นามสกุลผู้สมัคร และระยะเวลาการจัดส่งและหลังจากนั้นทางธนาคารจะแจ้งข้อมูลการดำเนินการจัดส่งต่อไป

รูปภาพที่ 2.5 : การเลือกชนิดบัตรเดบิตที่ลูกค้าต้องการ



รูปภาพที่ 2.6 : การกรอกข้อมูลที่อยู่จัดส่งบัตร



รูปภาพที่ 2.7 : การตรวจสอบข้อมูลที่อยู่และค่าทำบัตร

กฎเกณฑ์ทางธุรกิจ (Business Rules) ของกระบวนการสมัครบัตรเดบิตผ่าน Mobile Banking

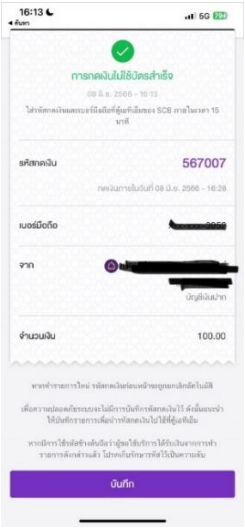
1. **ลูกค้า** สามารถ**สมัครบัตรเดบิต** ได้ครั้งละ 1 ใบ
   บัตรเดบิตแต่ละใบสามารถถูก**สมัคร**โดย**ลูกค้า**ได้เพียง 1 คน
2. ในแต่ละครั้ง**ลูกค้า** สามารถ**เลือกประเภทของบัตรเดบิต**ได้เพียง 1 ประเภท
   ในแต่ละครั้ง**บัตรเดบิต**แต่ละใบสามารถถูก**เลือก**โดย**ลูกค้า**ได้เพียง 1 คน
3. ในแต่ละครั้ง**ลูกค้า** จะสามารถ**ระบุที่อยู่จัดส่งบัตร**ได้เพียง 1 ที่อยู่
   ในแต่ละครั้ง **ที่อยู่จัดส่งบัตร** จะต้องถูก**ระบุ**โดย**ลูกค้า**เพียง 1 คน
4. **ลูกค้า**ต้อง**ยืนยัน**ที่จะยินยอมเงื่อนไขการสมัครทุกเงื่อนไข
5. **ลูกค้า**ต้อง**เสียค่าธรรมเนียม**ในการสมัครบัตร 300 บาท
6. ในแต่ละครั้ง**ลูกค้า**สามารถ**เปิดใช้งานบัตรเดบิต**ได้ 1 ใบ
   บัตรเดบิต แต่ละใบ สามารถถูก**เปิดใช้งาน**จาก**ลูกค้า**ได้เพียง 1 ครั้ง
7. **บัตรเดบิต** 1 ใบสามารถ**ผูก**กับ**บัญชี**ได้ 1 บัญชี
   บัญชี 1 บัญชีสามารถถูก**ผูก**กับ**บัตรเดบิต**ได้หลายใบ
8. **ลูกค้า**จำเป็นต้อง**เปิดใช้งานบัตรเดบิต**ภายใน 60 วันหลังการสมัคร
   ภายใน 60 วันหลังการสมัครบัตร **บัตรเดบิต**ต้องถูก**เปิดใช้งาน**โดย**ลูกค้า**
9. แต่ละครั้งที่มี**การสมัครบัตรเดบิต** จะมีการ**ส่งข้อมูลการยืนยันการสมัคร**บัตรเดบิตผ่านทางอีเมล
   ข้อมูลการยืนยันการสมัครจะถูก**ส่ง**ผ่านทางอีเมลในแต่ละครั้งที่มี**การสมัครบัตรเดบิต**
10. **บัตรเดบิต**จะต้อง**มี**ข้อมูลประกอบไปด้วย card_num, CVV, card_type, exp_date, cardPIN

➢ กระบวนการที่ 3 : กระบวนการถอนเงินไม่ใช้บัตรผ่าน Mobile Banking

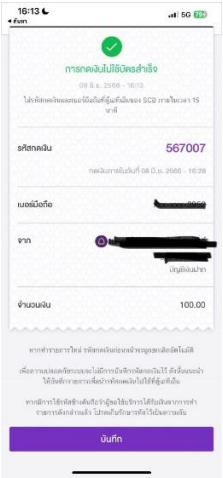
กระบวนการถอนเงินไม่ใช้บัตรผ่าน Mobile Banking ประกอบด้วยกระบวนการ ได้แก่ 1) การระบุจำนวนเงินที่ลูกค้าต้องการถอน 2) การตรวจสอบและยืนยันข้อมูลการถอนเงิน 3) รับรหัสถอนเงินเพื่อนำไปถอนที่ตู้ ATM 4) การกดเงินไม่ใช้บัตรที่ตู้ ATM

ลูกค้าสามารถกดเงินโดยไม่ใช้บัตรที่ตู้ ATM ผ่านแอปพลิเคชัน SCB Easy ได้ โดยการระบุจำนวนเงินที่ลูกค้าต้องการถอน (รูปภาพที่ 1) ซึ่งมีข้อบังคับว่าเงินที่ถอนจะถอนหารด้วย 100 ลงตัวและขั้นต่ำ 100 บาท สูงสุด 30,000 บาทเท่านั้น (เนื่องจากตู้ ATM มีธนบัตร 100 บาทเป็นขั้นต่ำ) หลังจากที่ลูกค้าระบุจำนวนเงินที่ต้องการถอนเสร็จ จะเข้าสู่กระบวนการที่ลูกค้าจะต้องตรวจสอบข้อมูลการถอนเงิน (รูปภาพที่ 2) ถ้าข้อมูลถูกต้อง ให้กดยืนยัน และหลังจากนั้น ทางระบบธนาคารจะทำการ generate รหัสกดเงินเพื่อให้นำไปถอนที่ตู้ ATM ได้ (รูปภาพที่ 3) ลูกค้าจะต้องนำรหัสกดเงินที่ได้ ไปดำเนินการทำธุรกรรมที่ตู้ และเมื่อทำธุรกรรมสำเร็จ จะมีการแจ้งเตือนขึ้นบนแอป และถือว่าเป็นการกดเงินไม่ใช้บัตรที่เสร็จสมบูรณ์

รูปภาพที่ 2.8 : การระบุจำนวนเงินที่ต้องการถอน



รูปภาพที่ 2.9 : การตรวจสอบข้อมูลการถอนเงิน



รูปภาพที่ 2.10 : รับรหัสถอนเงินเพื่อนำไปถอนที่ตู้ ATM

กฎเกณฑ์ทางธุรกิจ (Business Rules) ของกระบวนการถอนเงินโดยไม่ใช้บัตร

1. **ลูกค้า** สามารถ**ทำการถอนเงินโดยไม่ใช้บัตร**ได้ไม่จำกัดจำนวนครั้ง
   ในแต่ละครั้ง**การถอนเงินโดยไม่ใช้บัตร**สามารถถูก**ทำ**โดย**ลูกค้า**ได้เพียง 1 คน
2. ในแต่ละครั้ง**ลูกค้า**จะต้อง**ถอนเงิน**ในช่วง 100 – 30,000 บาท
   ในแต่ละครั้ง**เงิน**ในช่วง 100 - 30,000 บาท จะถูก**ถอน**โดย**ลูกค้า**
3. **ลูกค้า**จะ**ถอนเงิน**ได้เฉพาะจำนวนที่หารด้วย 100 ลงตัว
4. **ลูกค้า**สามารถ**ถอนเงิน**ภายในระยะเวลา 1 วันได้ไม่เกิน 50,000 บาท
5. **บัญชี**บางประเภทไม่สามารถใช้**การถอนเงินโดยไม่ใช้บัตร**ได้
6. **พนักงาน** 1 คนสามารถ**เติมเงิน**เข้า**ตู้ ATM** ได้หลายตู้
   ตู้ ATM 1 ตู้สามารถถูก**เติมเงิน**โดย**พนักงาน**ได้หลายคน
7. ใน**การถอนเงินโดยไม่ใช้บัตร** 1 ครั้ง จะ**ได้รับรหัสกดเงิน** 1 รหัส
   รหัสกดเงิน 1 รหัสจะ**ถูกส่ง**เพื่อนำไปใช้ใน **การถอนเงินโดยไม่ใช้บัตร** ได้เพียง 1 ครั้ง
8. **ลูกค้า**สามารถ**ค้นหาตู้ ATM** ได้หลายแห่ง
   ตู้ ATM สามารถถูก**ค้นหา**ได้จาก**ลูกค้า** หลายคน
9. ในแต่ละครั้ง**ลูกค้า**สามารถ**ยกเลิกรหัสกดเงิน**ได้
   ในแต่ละครั้ง**รหัสกดเงิน**สามารถถูก**ยกเลิก**โดย**ลูกค้า**
10. ในแต่ละครั้ง**ลูกค้า**สามารถ**เลือกบัญชี**ที่จะใช้ทำธุรกรรมได้
    ในแต่ละครั้ง**บัญชี**สามารถถูก**เลือก**ในการใช้ทำธุรกรรมโดย**ลูกค้า**

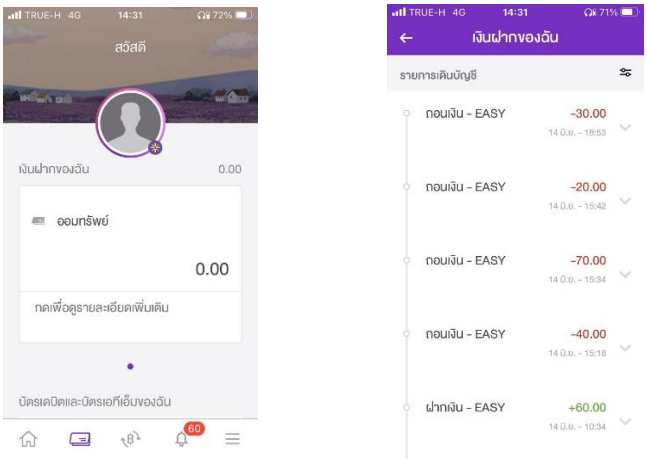
3. ความต้องการข้อมูลของธุรกิจ (Business data requirements)

- ข้อมูลของลูกค้า ประกอบด้วย ชื่อและนามสกุล , เลขที่บัญชี , ธนาคารที่ใช้เปิดบัญชี และหมายเลขโทรศัพท์
- ข้อมูลบัญชี ประกอบด้วย เลขที่บัญชี , ประเภทของบัญชี , ยอดเงินในบัญชี และ หมายเลขพร้อมเพย์
- ข้อมูลการทำธุรกรรมโอนเงิน ประกอบด้วย ชื่อและนามสกุลผู้โอนกับผู้รับ , เลขที่บัญชีผู้โอนกับผู้รับ , ธนาคารของผู้โอนกับผู้รับ , จำนวนเงินที่โอน , วัน/เดือน/ปี/เวลาการโอน , ประเภทการโอนเงิน และ หมายเลขอ้างอิงการทำธุรกรรม
- ข้อมูลการสมัครบัตรเดบิต ประกอบด้วย ชื่อและนามสกุลของผู้ถือบัตร , หมายเลขบัตร , ที่อยู่การจัดส่งบัตร และ ชนิดของบัตร
- ข้อมูลการทำธุรกรรมถอนเงินไม่ใช้บัตร ประกอบด้วย เลขบัญชีที่ใช้ถอนเงิน , เบอร์โทรศัพท์ , วัน/เดือน/ปี/เวลาการที่ทำการถอนเงิน , จำนวนเงินที่ถอน และ รหัสการถอนเงิน
- ข้อมูลที่อยู่การจัดส่ง ประกอบด้วย ชื่อและนามสกุล ที่อยู่ ตำบล อำเภอ จังหวัด ประเทศ รหัสไปรษณีย์ และ เบอร์โทรศัพท์
- ข้อมูลสาขา ประกอบด้วย ชื่อสาขา , ที่อยู่ และจำนวนตู้ ATM

- ข้อมูลตู้ ATM ประกอบด้วย Serial Number ของตู้ , จำนวนเงินภายในตู้ , สถานะความพร้อมในการถอนเงิน
- ข้อมูลพนักงาน ประกอบด้วย ชื่อและนามสกุลของพนักงาน และ ตำแหน่ง

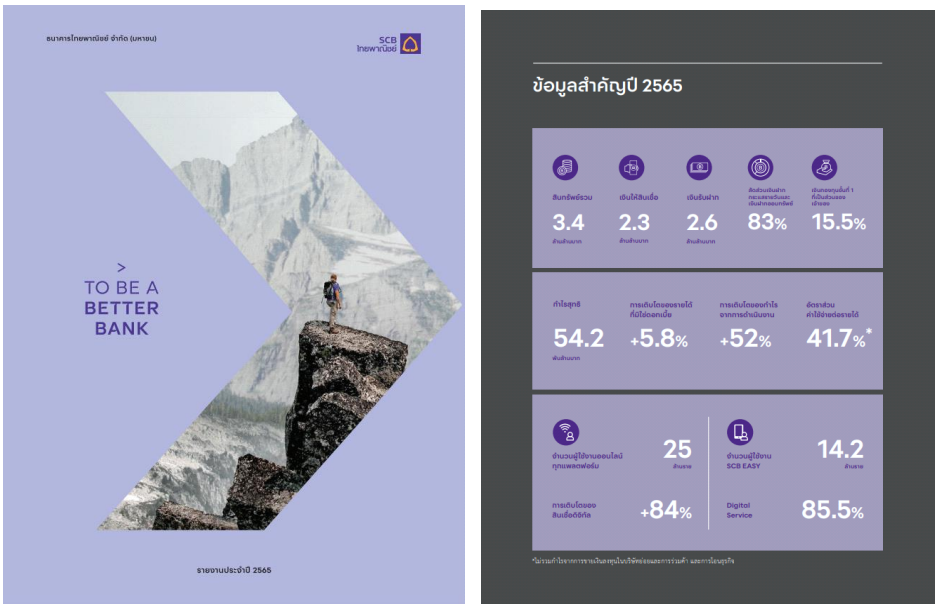
4. <u>ตัวอย่างรายงานสารสนเทศที่น่าสนใจ</u>

รายงานสารสนเทศที่น่าสนใจใน Application “SCB Easy” นี้ คือ **ข้อมูลประวัติการเข้า-ออก**เงิน ที่ผู้ใช้สามารถตรวจสอบได้ทันที โดยไม่จำเป็นต้องใช้สมุดบัญชีอีกต่อไป



รูปภาพที่ 4.1 & 4.2 หน้าสรุปการเข้า-ออกเงินในบัญชี

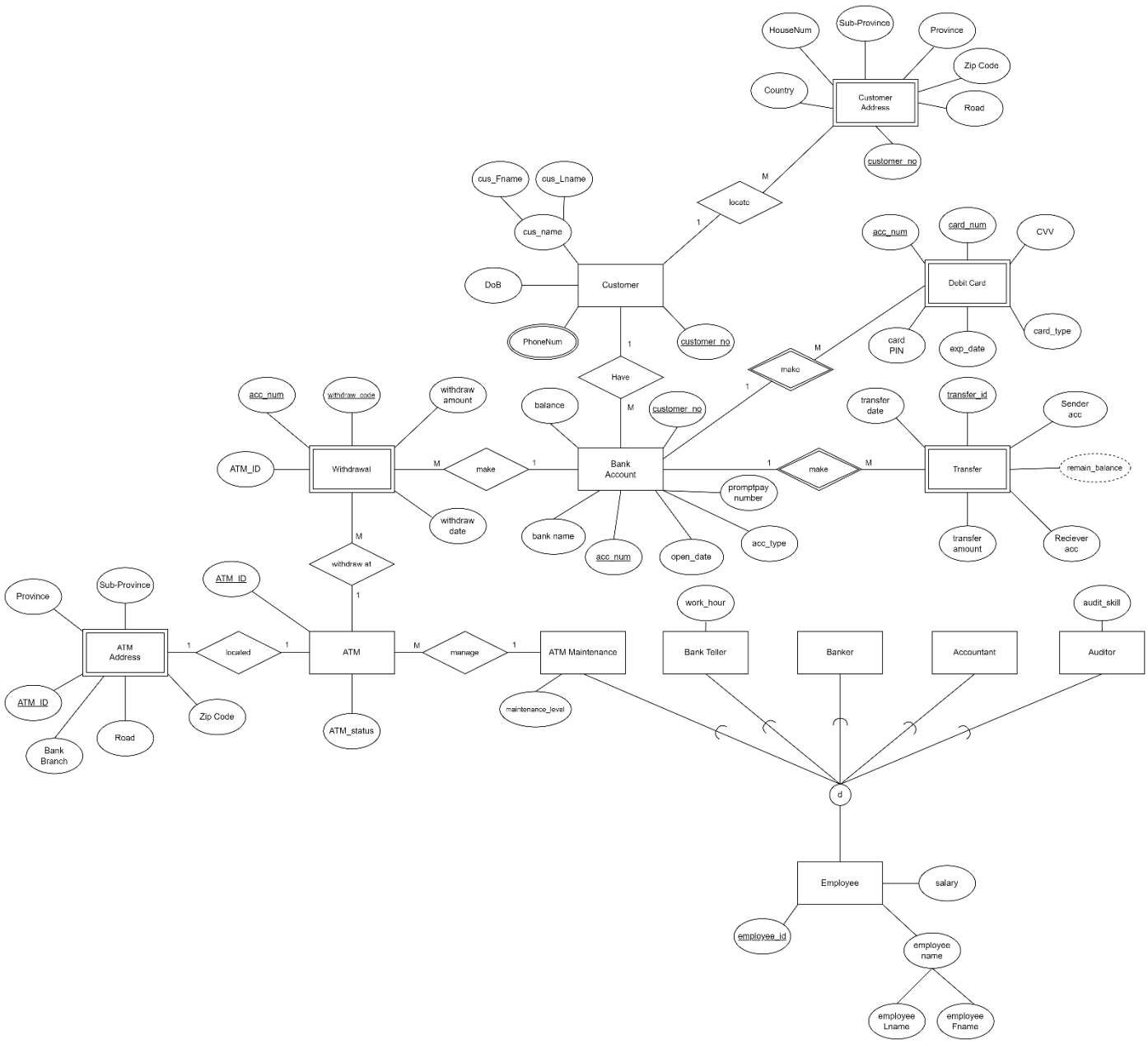
และในมุมมองของ**ธนาคาร**ที่ต้องการนำข้อมูลจากลูกค้ามาใช้งาน จะมีการเก็บข้อมูลเป็น**รายงานประจำปี** โดยนำข้อมูลลูกค้า เช่น เลขที่บัญชี, ยอดเงินในบัญชี, ประวัติการทำธุรกรรม เป็นต้น และนำมาทำเป็นข้อมูลในแบบสารสนเทศ



รูปภาพที่ 4.3 & 4.4 หน้ารายงานประจำปี

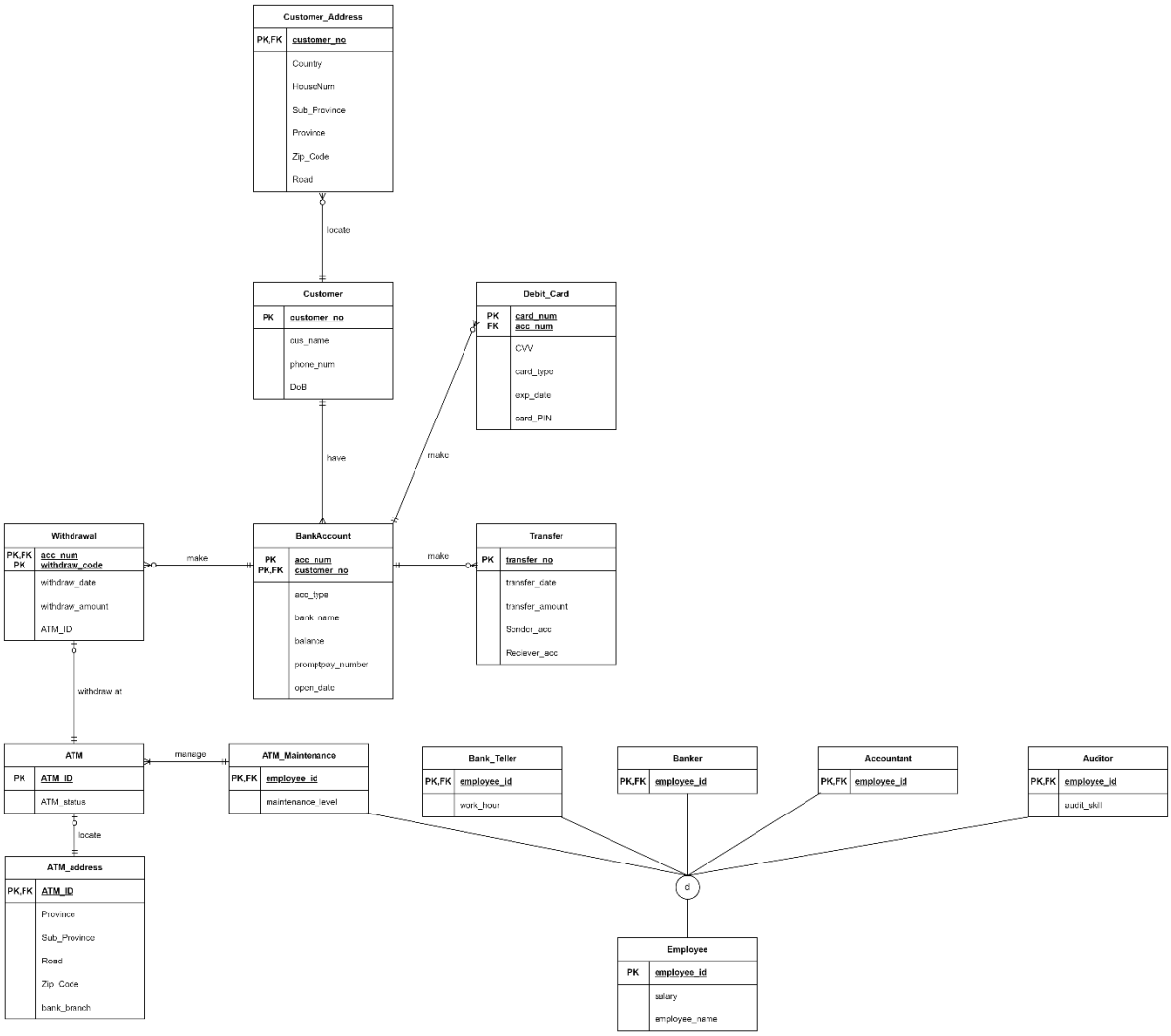
5. แผนภาพ ER/EER Diagram

5.1. แผนภาพ ER/EER Diagram แบบ Chen



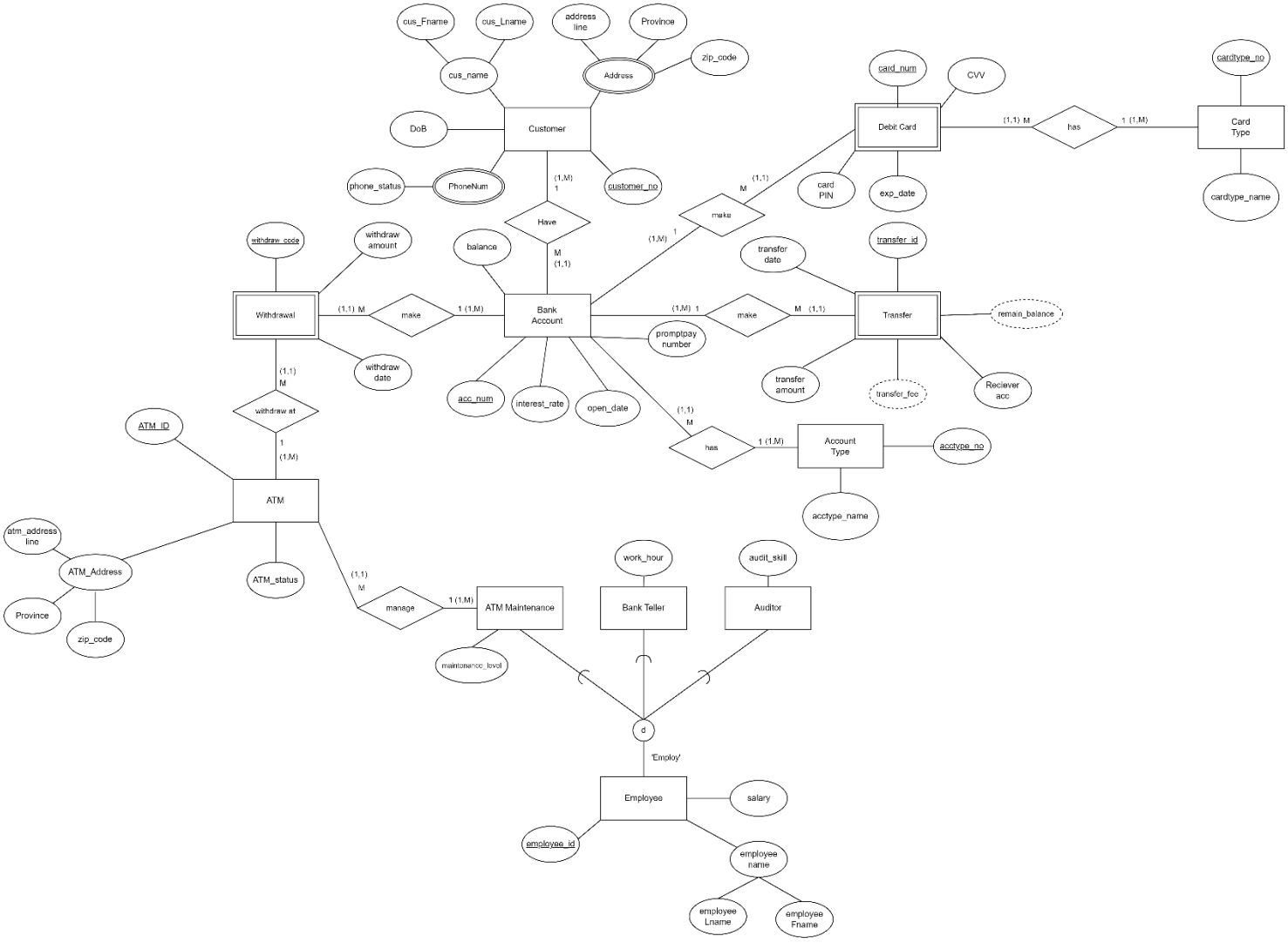
รูปภาพที่ 5.1 แผนภาพ ER Diagram แบบ Chen

5.2. แผนภาพ ER/EER Diagram แบบ Crow-Foot



รูปภาพที่ 5.2 แผนภาพ ER Diagram แบบ Crow-Foot

6. แผนภาพ ER/EER Diagram ฉบับปรับปรุง



รูปภาพที่ 6 แผนภาพ ER Diagram ฉบับปรับปรุง

ER Diagram ข้างต้นนี้ มีการปรับปรุงจาก**แบบเดิมในข้อที่ 5.1.** เนื่องจากมีการใส่ Entity และ Attribute ที่ไม่ถูกต้อง และมีการพบว่าข้อมูลบางส่วนในกฎเกณฑ์ทางธุรกิจตกหล่นไป จึงมีการปรับปรุงดังนี้:

1. นำ Entity ของ ATM Address และ Customer Address ออก เนื่องจาก Address หรือที่อยู่นั้น ควรเป็น Attribute และที่อยู่ของลูกค้า สามารถมีได้มากกว่า 1 ที่อยู่ จึงปรับ Customer Address เป็น Multi-Valued Attribute และ ATM Address เป็น Composite Attribute
2. ลด Subtype Entity ตัวที่ไม่จำเป็นลงในส่วนของ Supertype : Employee เพื่อลดปัญหาความคลุมเครือของข้อมูล

3. เพิ่ม Entity ของประเภทบัตร (Card type) และ ประเภทบัญชี (Account type) เพื่อลดความซ้ำซ้อนของข้อมูลใน Table เมื่อแปลงเป็น Database บน MySQL
4. เพิ่ม attribute interest_rate (อัตราดอกเบี้ยของบัญชี) ใน Entity Bank Account และ derived attribute transfer_fee (ค่าธรรมเนียมในการโอน) ใน Entity Transfer
5. เพิ่ม Cardinality เข้าไปใน ER Diagram เพื่อบอกถึงค่า min,max ในแต่ละเส้นความสัมพันธ์

7. การดำเนินกระบวนการ Transform Mapping ในแต่ละขั้นตอน

ขั้นตอนที่ 1 : สร้างตารางความสัมพันธ์ (Relation) ของแต่ละ Strong หรือ Regular Entity โดยรวม Attribute ที่เป็น Single หรือ Composite Attribute และ Key Attribute

ผลลัพธ์

CUSTOMER

customer_no	cus_Fname	cus_Lname	DoB

BANK_ACCOUNT

acc_num	acc_balance	interest_rate	open_date	promptpay_num

ATM

ATM_ID	ATM_status	ATM_address_line	ATM_province	ATM_zip_code

EMPLOYEE

emp_id	emp_salary	emp_fname	emp_lname

ACCOUNT_TYPE

acctype_no	acctype_name

CARD_TYPE

cardtype_no	cardtype_name

<u>ขั้นตอนที่ 2 :</u> สร้างตารางความสัมพันธ์ (Relation) ของแต่ละ Weak Entity

<u>ผลลัพธ์</u>

BANK_ACCOUNT

acc_num	acc_balance	interest_rate	open_date	promptpay_num

TRANSFER

transfer_id	acc_num	transfer_date	transfer_amount	receiver_acc

BANK_ACCOUNT

acc_num	acc_balance	interest_rate	open_date	promptpay_num

WITHDRAWAL

withdraw_code	acc_num	withdraw_date	withdraw_amount

BANK_ACCOUNT

acc_num	acc_balance	interest_rate	open_date	promptpay_num

DEBIT_CARD

card_num	acc_num	card_PIN	exp_date	CVV

ขั้นตอนที่ 3 : แปลงความสัมพันธ์แบบ 1:1 Relationship

<u>ผลลัพธ์</u>

ไม่มีความสัมพันธ์แบบ 1:1 ใน ER Diagram นี้

<u>ขั้นตอนที่ 4 :</u> แปลงความสัมพันธ์แบบ 1:N Relationship

<u>ผลลัพธ์</u>

CUSTOMER

customer_no	cus_Fname	cus_Lname	DoB

BANK_ACCOUNT

acc_num	acc_balance	customer_no	interest_rate	open_date	promptpay_num

WITHDRAWAL

withdraw_code	acc_num	ATM_ID	withdraw_date	withdraw_amount

ATM

ATM_ID	ATM_status	ATM_address_line	ATM_province	ATM_zip_code

ATM

ATM_ID	Emp_id	ATM_status	ATM_address_line	ATM_province	ATM_zip_code

EMPLOYEE

emp_id	emp_salary	emp_fname	emp_lname

DEBIT_CARD

card_num	acc_num	cardtype_no	card_PIN	exp_date	CVV

CARD_TYPE

cardtype_name	cardtype_no

BANK_ACCOUNT

acc_num	acctype_no	customer_no	acc_balance	interest_rate	open_date	promptpay_num

ACCOUNT_TYPE

acctype_name	acctype_no

<u>ขั้นตอนที่ 5 :</u> แปลงความสัมพันธ์แบบ M:N Relationship

<u>ผลลัพธ์</u>

ไม่มีความสัมพันธ์แบบ M:N ใน ER Diagram นี้

<u>ขั้นตอนที่ 6 :</u> แปลง Multi-valued Attribute

<u>ผลลัพธ์</u>

CUSTOMER

customer_no	cus_Fname	cus_Lname	DoB

CUSTOMER_ADDRESS

cus_address	customer_no	cus_address_line	cus_province	cus_zip_code	cus_road

CUSTOMER

customer_no	cus_Fname	cus_Lname	DoB

CUSTOMER_PHONENUM

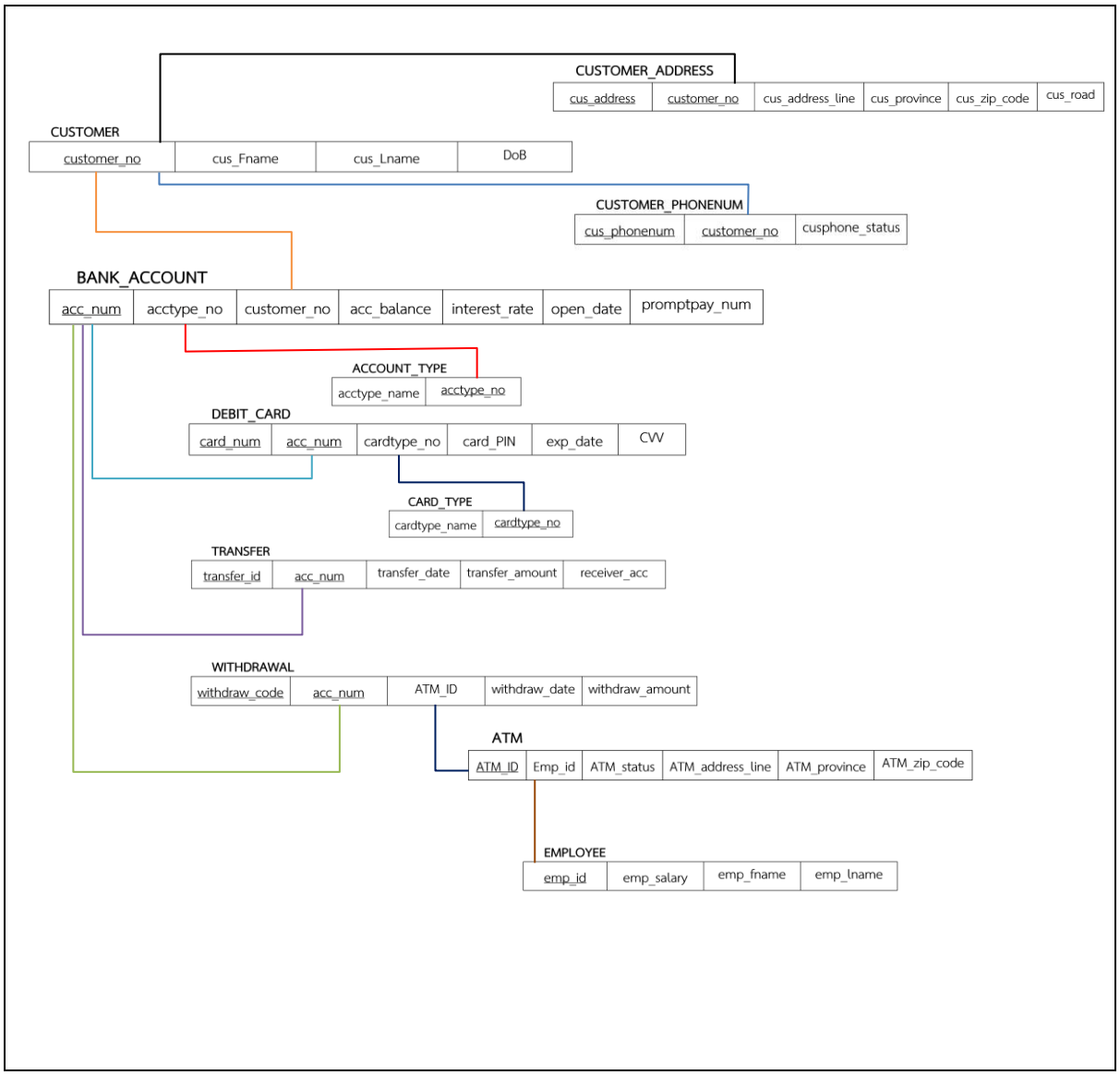
cus_phonenum	customer_no	cusphone_status

<u>ขั้นตอนที่ 7 :</u> แปลงความสัมพันธ์แบบ N-ary Relationship

<u>ผลลัพธ์</u>

ไม่มีความสัมพันธ์แบบ N-ary ใน ER Diagram นี้

8. ภาพ Relational Diagram แสดงภาพรวมการออกแบบ



9. Data Dictionary สำหรับทุก Schema

Table Name (ชื่อตาราง)	Attribute Name (ชื่อ Attribute)	Contents (คำอธิบาย)	Types (ชนิดของข้อมูล)	Format (รูปแบบ)	Nullable (เป็นค่า Null ได้หรือไม่)	Range (ช่วงข้อมูล)	Key (PK)	FK Referenceable Table (FK และตารางที่อ้างมา)
CUSTOMER	customer_no	Customer's Number	int	x			PK	
	cus_Fname	Customer's First Name	nvarchar(50)	xxxxxxx				
	cus_Lname	Customer's Last Name	nvarchar(50)	xxxxxxx				
	DoB	Customer's Date-of-Birth	date	yyyy-mm-dd				
CUSTOMER_ADDRESS	customer_no	Customer's Number	int	x			PK,FK	customer_no [CUSTOMER]
	cus_address_no	Customer's Address No. **(ในกรณีลูกค้ามีหลายที่อยู่)**	int	x			PK	
	cus_address_line	Customer's Address Line.	nvarchar(255)	xxxxxxx				
	cus_province	Customer's Address Province	nvarchar(255)	xxxxxxx				
	cus_zip_code	Customer's Address Zip Code	nvarchar(5)	xxxxxxx				
CUSTOMER_PHONE_NUM	customer_no	Customer's number	int	x			PK,FK	customer_no [CUSTOMER]
	cus_phonenum	Customer's Phone Number	nvarchar(10)	xxxxxxx			PK	
	cusphone_status	Customer's Phone Status	char(1)	x		'Y'/'N'		
BANK_ACCOUNT	acc_num	Account's Number	nvarchar(10)	xxxxxxx			PK	
	acctype_no	Account's type Number	int	x			FK	acctype_no [ACCOUNT_TYPE]
	customer_no	Customer's number	int	x			FK	customer_no [CUSTOMER]
	acc_balance	Account's balance	decimal(12,2)	1234567890.12				
	interest_rate	Account's interest rate	decimal(3,2)	1.23				
	open_date	Account's open date	date	yyyy-mm-dd				
	promptpay_num	Account's promptpay number	nvarchar(10)	xxxxxxx	Y			
ACCOUNT_TYPE	acctype_name	Account's type name	nvarchar(100)	xxxxxxx				
	acctype_no	Account's type number	int	x			PK	
TRANSFER	transfer_id	Transfer's ID	int	x			PK	
	acc_num	Account's Number	nvarchar(10)	xxxxxxx			PK,FK	acc_num [BANK_ACCOUNT]
	transfer_date	Transfer's Date	datetime	yyyy-mm-dd xx:xx:xx				
	transfer_amount	Transfer's Amount	decimal(12,2)	1234567890.12				
	receiver_acc	Receiver Account's	nvarchar(10)	xxxxxxx				

WITHDRAWAL	withdraw_code	Withdraw Code	int	x			PK	
	acc_num	Account's Number	nvarchar(10)	xxxxxxx			PK,FK	acc_num [BANK_ACCOUNT]
	atm_id	ATM's ID	int	x			FK	atm_id [ATM]
	withdraw_date	Withdraw Date	datetime	yyyy-mm-dd xx:xx:xx				
	withdraw_amount	Withdraw Amount	decimal(12,2)	1234567890.12		[100,30000]		
DEBIT_CARD	card_num	Debit Card's number	nvarchar(20)	xxxxxxx			PK	
	acc_num	Account's number	nvarchar(10)	xxxxxxx			PK,FK	acc_num [BANK_ACCOUNT]
	cardtype_no	Debit Card's type number	int	x			FK	cardtype_no [CARD_TYPE]
	card_PIN	Debit Card's Password PIN	nvarchar(6)	xxxxxx				
	exp_date	Debit Card's Expiry Date	date	yyyy-mm-dd				
	CVV	Card Verification Value	nvarchar(3)	xxx				
CARD_TYPE	cardtype_name	Debit Card's type name	nvarchar(100)	xxxxxxx				
	cardtype_no	Debit Card's type number	int	x			PK	
ATM	atm_id	ATM's ID	int	x			PK	
	emp_id	Employee's ID	int	x			FK	emp_id [EMPLOYEE]
	atm_status	ATM's Status	char(1)	x		'Y'/'N'		
	atm_address_line	ATM's Address Line	nvarchar(255)	xxxxxxx				
	atm_province	ATM's Address Province	nvarchar(255)	xxxxxxx				
	atm_zip_code	ATM's Address Zip Code	nvarchar(5)	xxxxxxx				
EMPLOYEE	emp_id	Employee's ID	int	x			PK	
	emp_fname	Employee's First Name	nvarchar(50)	xxxxxxx				
	emp_lname	Employee's Last Name	nvarchar(50)	xxxxxxx				

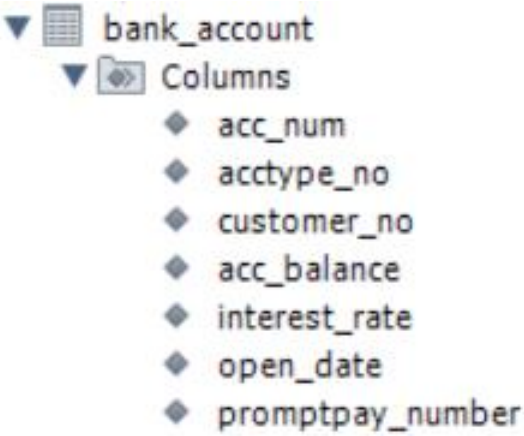
10. SQL (Structured Query Language)

10.1 Data Definition Language (DDL)

BANK_ACCOUNT

acc_num	acc_balance	interest_rate	open_date	promptpay_num

```
-- CREATE BANK ACCOUNT TABLE
CREATE TABLE IF NOT EXISTS BANK_ACCOUNT(
    acc_num NVARCHAR(10),
    acctype_no INT NOT NULL,
    customer_no INT NOT NULL,
    acc_balance DECIMAL(12,2) NOT NULL,
    interest_rate DECIMAL(3,2) NOT NULL,
    open_date DATE NOT NULL,
    promptpay_number NVARCHAR(10),
    PRIMARY KEY (acc_num)
);
```

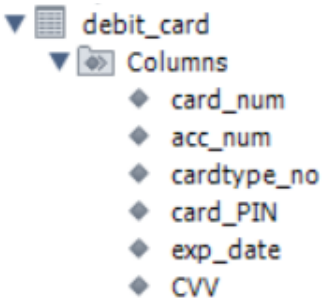


รูปภาพที่ 10.1.1 การสร้าง Table bank_account

DEBIT_CARD

card_num	acc_num	cardtype_no	card_PIN	exp_date	CVV

```
-- CREATE DEBIT CARD TABLE
CREATE TABLE IF NOT EXISTS DEBIT_CARD(
    card_num NVARCHAR(20),
    acc_num NVARCHAR(10),
    cardtype_no INT NOT NULL,
    card_PIN NVARCHAR(6) NOT NULL,
    exp_date DATE NOT NULL,
    CVV NVARCHAR(3) NOT NULL,
    PRIMARY KEY (card_num,acc_num),
    CONSTRAINT acc_con FOREIGN KEY (acc_num) REFERENCES bank_account(acc_num)
);
```

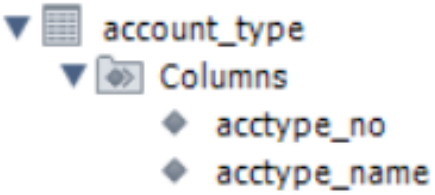


รูปภาพที่ 10.1.2 การสร้าง Table debit_card

ACCOUNT_TYPE

acctype_name	acctype_no

```
-- CREATE ACCOUNT TYPE TABLE
CREATE TABLE IF NOT EXISTS ACCOUNT_TYPE(
    acctype_no INT NOT NULL AUTO_INCREMENT,
    acctype_name NVARCHAR(100) NOT NULL,
    CONSTRAINT acctype_prime PRIMARY KEY (acctype_no)
);
```

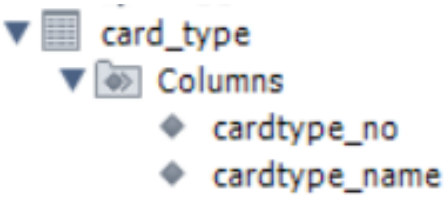


รูปภาพที่ 10.1.3 การสร้าง Table account_type

CARD_TYPE

cardtype_name	cardtype_no

```
-- CREATE CARD TYPE TABLE
CREATE TABLE IF NOT EXISTS CARD_TYPE(
    cardtype_no INT NOT NULL AUTO_INCREMENT,
    cardtype_name NVARCHAR(100) NOT NULL,
    PRIMARY KEY (cardtype_no)
);
```

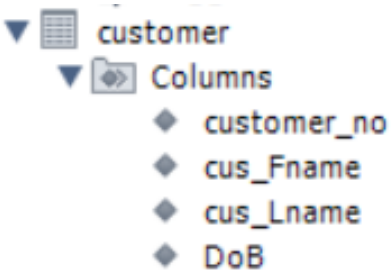


รูปภาพที่ 10.1.4 การสร้าง Table card_type

CUSTOMER

customer_no	cus_Fname	cus_Lname	DoB

```
-- CUSTOMER TABLE
CREATE TABLE IF NOT EXISTS CUSTOMER(
    customer_no INT NOT NULL AUTO_INCREMENT,
    cus_Fname NVARCHAR(50) NOT NULL,
    cus_Lname NVARCHAR(50) NOT NULL,
    DoB DATE NOT NULL,
    CONSTRAINT customer_prime PRIMARY KEY (customer_no)
);
```

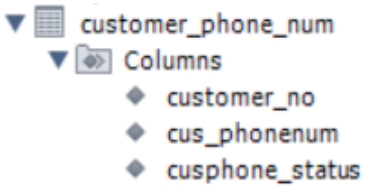


รูปภาพที่ 10.1.5 การสร้าง Table customer

CUSTOMER_PHONENUM

cus_phonenum	customer_no	cusphone_status

```
-- CREATE CUSTOMER PHONE NUMBER TABLE
CREATE TABLE IF NOT EXISTS CUSTOMER_PHONE_NUM(
    customer_no INT NOT NULL,
    cus_phonenum NVARCHAR(10),
    cusphone_status CHAR(1) NOT NULL,
    CONSTRAINT customer_phone_num_con FOREIGN KEY (customer_no) REFERENCES CUSTOMER(customer_no),
    CONSTRAINT customer_phone_num_prime PRIMARY KEY (cus_phonenum,customer_no),
    CONSTRAINT CHK_cus_phonenum CHECK (cusphone_status in ('Y','N'))
);
```

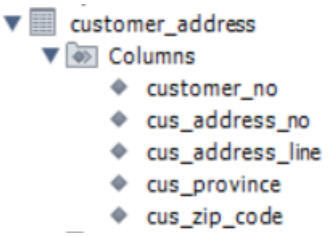


รูปภาพที่ 10.1.6 การสร้าง Table customer_phonenum

CUSTOMER_ADDRESS

cus_address	customer_no	cus_address_line	cus_province	cus_zip_code	cus_road

```
-- CREATE CUSTOMER ADDRESS TABLE
CREATE TABLE IF NOT EXISTS CUSTOMER_ADDRESS(
    customer_no INT NOT NULL,
    cus_address_no INT NOT NULL,
    cus_address_line NVARCHAR(255) NOT NULL,
    cus_province NVARCHAR(255) NOT NULL,
    cus_zip_code NVARCHAR(5) NOT NULL,
    CONSTRAINT customer_address_con FOREIGN KEY (customer_no) REFERENCES CUSTOMER(customer_no),
    CONSTRAINT customer_address_prime PRIMARY KEY (customer_no, cus_address_no)
);
```

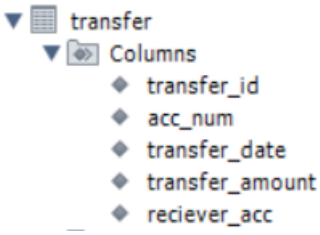


รูปภาพที่ 10.1.7 การสร้าง Table customer_address

TRANSFER

transfer_id	acc_num	transfer_date	transfer_amount	receiver_acc

```
-- CREATE TRANSFER TABLE
CREATE TABLE IF NOT EXISTS TRANSFER(
    transfer_id INT NOT NULL AUTO_INCREMENT,
    acc_num NVARCHAR(10) NOT NULL,
    transfer_date DATETIME NOT NULL,
    transfer_amount DECIMAL(12,2) NOT NULL,
    reciever_acc NVARCHAR(10) NOT NULL,
    CONSTRAINT transfer_con FOREIGN KEY (acc_num) REFERENCES bank_account(acc_num),
    CONSTRAINT transfer_prime PRIMARY KEY (transfer_id, acc_num)
);
```

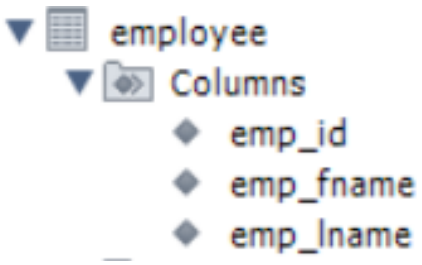


รูปภาพที่ 10.1.8 การสร้าง Table transfer

EMPLOYEE

emp_id	emp_salary	emp_fname	emp_lname

```
-- CREATE EMPLOYEE TABLE
CREATE TABLE IF NOT EXISTS EMPLOYEE(
    emp_id INT NOT NULL AUTO_INCREMENT,
    emp_fname NVARCHAR(50) NOT NULL,
    emp_lname NVARCHAR(50) NOT NULL,
    CONSTRAINT emp_prime PRIMARY KEY (emp_id)
);
```

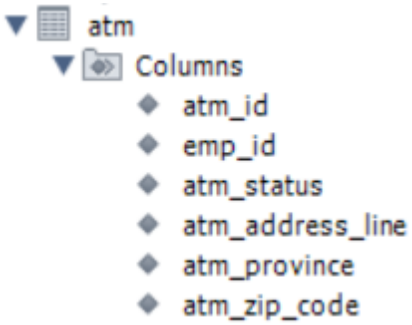


รูปภาพที่ 10.1.9 การสร้าง Table employee

ATM

ATM_ID	Emp_id	ATM_status	ATM_address_line	ATM_province	ATM_zip_code

```
-- CREATE ATM TABLE
CREATE TABLE IF NOT EXISTS ATM(
    atm_id INT NOT NULL AUTO_INCREMENT,
    emp_id INT NOT NULL,
    atm_status CHAR(1) NOT NULL,
    atm_address_line NVARCHAR(255) NOT NULL,
    atm_province NVARCHAR(255) NOT NULL,
    atm_zip_code NVARCHAR(5) NOT NULL,
    CONSTRAINT ATM_con FOREIGN KEY (emp_id) REFERENCES EMPLOYEE(emp_id),
    CONSTRAINT ATM_prime PRIMARY KEY (atm_id)
);
```

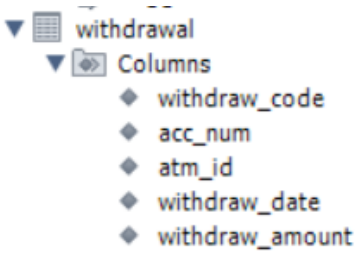


รูปภาพที่ 10.1.10 การสร้าง Table ATM

WITHDRAWAL

withdraw_code	acc_num	ATM_ID	withdraw_date	withdraw_amount

```
-- CREATE WITHDRAWAL TABLE
CREATE TABLE IF NOT EXISTS WITHDRAWAL(
    withdraw_code NVARCHAR(7) NOT NULL,
    acc_num NVARCHAR(10) NOT NULL,
    atm_id INT NOT NULL,
    withdraw_date DATETIME NOT NULL,
    withdraw_amount DECIMAL(7,2) NOT NULL,
    CONSTRAINT withdraw_con FOREIGN KEY (acc_num) REFERENCES bank_account(acc_num),
    CONSTRAINT withdraw_atm FOREIGN KEY (atm_id) REFERENCES ATM(atm_id),
    CONSTRAINT withdraw_prime PRIMARY KEY (withdraw_code, acc_num)
);
```



รูปภาพที่ 10.1.11 การสร้าง Table withdrawal

- ALTER Constraint

```
-- MODIFY cardtype_no in TABLE CARD_TYPE
ALTER TABLE CARD_TYPE
    MODIFY COLUMN `cardtype_no` INT NOT NULL AUTO_INCREMENT ;

-- ADD CONSTRAINT IN TABLE DEBIT_CARD
ALTER TABLE DEBIT_CARD
    ADD CONSTRAINT card_con FOREIGN KEY (cardtype_no) REFERENCES card_type(cardtype_no);

-- ADD CONSTRAINT IN BANK_ACCOUNT
ALTER TABLE bank_account
    ADD CONSTRAINT customer_no_con FOREIGN KEY (customer_no) REFERENCES customer(customer_no),
    ADD CONSTRAINT acc_type_con FOREIGN KEY (acctype_no) REFERENCES account_type(acctype_no);

-- ADD CONSTRAINT IN ATM
ALTER TABLE ATM
    ADD CONSTRAINT atm_status_con CHECK (atm_status in ('Y','N'));

-- ADD CONSTRAINT IN TRANSFER
ALTER TABLE transfer
    ADD CONSTRAINT amount_limit CHECK (transfer_amount BETWEEN 1 AND 200000);

ALTER TABLE withdrawal
    ADD CONSTRAINT withdraw_limit CHECK (withdraw_amount BETWEEN 100 AND 30000);
```

รูปภาพที่ 10.1.12 การเพิ่ม Constraint เข้าไปในตารางโดยใช้ Alter

10.2 Data Manipulation Language (DML)

```
-- INSERT CUSTOMER DATA
INSERT INTO CUSTOMER (cus_Fname,cus_Lname,DoB)
    VALUES ('Saran','Wongthanakit','1990-05-15'),
           ('Napat','Thongchai','1985-12-10'),
           ('Sirima','Khongchai','1982-07-25'),
           ('Tawan','Somchai','1998-02-03'),
           ('Patchara','Panom','1993-11-28'),
           ('Darunee','Rattanakosin','1984-08-22'),
           ('Kunthira','Chaiyapruk','1987-09-12'),
           ('Surachai','Boonsri','1995-06-20'),
           ('Natthapong','Songtham','1989-04-17'),
           ('Niran','Thamrong','1991-03-07');
```

customer_no	cus_Fname	cus_Lname	DoB
1	Saran	Wongthanakit	1990-05-15
2	Napat	Thongchai	1985-12-10
3	Sirima	Khongchai	1982-07-25
4	Tawan	Somchai	1998-02-03
5	Patchara	Panom	1993-11-28
6	Darunee	Rattanakosin	1984-08-22
7	Kunthira	Chaiyapruk	1987-09-12
8	Surachai	Boonsri	1995-06-20
9	Natthapong	Songtham	1989-04-17
10	Niran	Thamrong	1991-03-07
NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.1 การเพิ่มข้อมูลเข้าใน Table customer

```
-- INSERT PHONE NUM DATA
INSERT INTO CUSTOMER_PHONE_NUM
    VALUES (1,'0874512093','Y'),
           (2,'0930925174','N'),
           (2,'0957727772','Y'),
           (3,'0689235410','Y'),
           (4,'0830194725','Y'),
           (5,'0647210935','Y'),
           (6,'0823893019','Y'),
           (7,'0930937451','Y'),
           (8,'0651748923','Y'),
           (9,'0800077799','N'),
           (9,'0930925174','Y'),
           (10,'0809356217','Y');
```

customer_no	cus_phonenum	cusphone_status
1	0874512093	Y
2	0930925174	N
2	0957727772	Y
3	0689235410	Y
4	0830194725	Y
5	0647210935	Y
6	0823893019	Y
7	0930937451	Y
8	0651748923	Y
9	0800077799	N
9	0930925174	Y
10	0809356217	Y
NULL	NULL	NULL

รูปภาพที่ 10.2.2 การเพิ่มข้อมูลเข้าใน Table customer_phone_num

```
-- INSERT CUSTOMER ADDRESS DATA
INSERT INTO CUSTOMER_ADDRESS
    VALUES (1, 1, '32 Soi Phaholyothin', 'Chatuchak, Bangkok', '10900'),
           (1, 2, '84/33 Bangkluai-Sainoi Rd.', 'Bangbuathong, Nonthaburi', '11110'),
           (2, 1, '7987 Sukhumvit Rd.', 'Watthana, Bangkok', '10110'),
           (3, 1, '456 Phuket Rd.', 'Muang, Phuket', '83000'),
           (4, 1, '93 Chang Klan Rd.', 'Chang Khlan, Chiang Mai', '10110'),
           (5, 1, '21 Charoen Krung Road', 'Bang Rak, Bangkok', '10500'),
           (5, 2, '83 Bang Lamung District', 'Mueang Chon Buri, Chon Buri', '20150'),
           (5, 3, '507 Sukhothai Road', 'Mueang Sukhothai,Sukhothai', '64210'),
           (6, 1, '204 Hat Yai District', 'Mueang Songkhla, Songkhla', '90110'),
           (7, 1, '156 Mueang Chiang Rai District', 'Mueang Chiang Rai, Chiang Rai', '57000'),
           (7, 2, '1015 Mueang Nonthaburi District', 'Mueang Nonthaburi, Nonthaburi', '11000'),
           (8, 1, '77 Mueang Ubon Ratchathani District', 'Mueang Ubon Ratchathani, Ubon Ratchathani', '34000'),
           (9, 1, '390 Hua Hin District', 'Mueang Prachuap Khiri Khan, Prachuap Khiri Khan', '77110'),
           (10, '1', '654 Ratchadamri Road', 'Pathum Wan, Bangkok', '10330');
```

customer_no	cus_address_no	cus_address_line	cus_province	cus_zip_code
1	2	84/33 Bangkluai-Sainoi Rd.	Bangbuathong, Nonthaburi	11110
2	1	7987 Sukhumvit Rd.	Watthana, Bangkok	10110
3	1	456 Phuket Rd.	Muang, Phuket	83000
4	1	93 Chang Klan Rd.	Chang Khlan, Chiang Mai	10110
5	1	21 Charoen Krung Road	Bang Rak, Bangkok	10500
5	2	83 Bang Lamung District	Mueang Chon Buri, Chon Buri	20150
5	3	507 Sukhothai Road	Mueang Sukhothai,Sukhothai	64210
6	1	204 Hat Yai District	Mueang Songkhla, Songkhla	90110
7	1	156 Mueang Chiang Rai District	Mueang Chiang Rai, Chiang Rai	57000
7	2	1015 Mueang Nonthaburi District	Mueang Nonthaburi, Nonthaburi	11000
8	1	77 Mueang Ubon Ratchathani ...	Mueang Ubon Ratchathani, Ub...	34000
9	1	390 Hua Hin District	Mueang Prachuap Khiri Khan, ...	77110
10	1	654 Ratchadamri Road	Pathum Wan, Bangkok	10330
NULL	NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.3 การเพิ่มข้อมูลเข้าใน Table customer_address

```
-- INSERT BANK ACCOUNT TYPE DATA
INSERT INTO ACCOUNT_TYPE
    VALUES (1,'Saving Account'),
           (2, 'Current Account'),
           (3, 'Fixed deposit Account');
```

acctype_no	acctype_name
1	Saving Account
2	Current Account
3	Fixed deposit Account
NULL	NULL

รูปภาพที่ 10.2.4 การเพิ่มข้อมูลเข้าใน Table account_type

```
-- INSERT BANK ACCOUNT DATA
INSERT INTO BANK_ACCOUNT
    VALUES ('0432156870', 1, 1, 5000.75, 2.25, '2020-01-15','0874512093'),
           ('0987564023', 3, 1, 200000.50, 1.50, '2022-11-22', NULL),
           ('0009876543', 1, 2, 12000.90, 2.75, '2023-05-08', NULL),
           ('7001234560', 1, 3, 890.00, 0.75, '2023-02-18','0689235410'),
           ('2109876500', 1, 4, 9800.10, 2.00, '2022-09-30','0830194725'),
           ('8006543219', 3, 4, 7482.55, 1.85, '2018-03-12', NULL),
           ('1098765400', 1, 5, 18250.50, 2.50, '2010-12-05', NULL),
           ('0098765432', 1, 6, 1900.25, 0.50, '2016-08-12','0823893019'),
           ('1023045678', 1, 7, 200000000.00, 2.35, '2022-10-10','0930937451'),
           ('0909876541', 2, 7, 99.99, 2.10, '2023-07-21', NULL),
           ('0765432109', 3, 7, 427500.00, 1.95, '2020-01-02', NULL),
           ('9009876542', 1, 8, 7500.00, 2.15, '2021-04-30','0651748923'),
           ('0000000123', 1, 9, 200.00, 1.75, '2023-04-23','0930925174'),
           ('4321567890', 1, 10, 8490000.00, 2.40, '2005-05-05', NULL),
           ('0032109876', 2, 10, 123456.99, 2.20, '2023-07-20', NULL);
```

acc_num	acctype_no	customer_no	acc_balance	interest_rate	open_date	promptpay_number
0000000123	1	9	200.00	1.75	2023-04-23	0930925174
0009876543	1	2	12000.90	2.75	2023-05-08	NULL
0032109876	2	10	123456.99	2.20	2023-07-20	NULL
0098765432	1	6	1900.25	0.50	2016-08-12	0823893019
0432156870	1	1	5000.75	2.25	2020-01-15	0874512093
0765432109	3	7	427500.00	1.95	2020-01-02	NULL
0909876541	2	7	99.99	2.10	2023-07-21	NULL
0987564023	3	1	200000.50	1.50	2022-11-22	NULL
1023045678	1	7	200000000.00	2.35	2022-10-10	0930937451
1098765400	1	5	18250.50	2.50	2010-12-05	NULL
2109876500	1	4	9800.10	2.00	2022-09-30	0830194725
4321567890	1	10	8490000.00	2.40	2005-05-05	NULL
7001234560	1	3	890.00	0.75	2023-02-18	0689235410
8006543219	3	4	7482.55	1.85	2018-03-12	NULL
9009876542	1	8	7500.00	2.15	2021-04-30	0651748923
NULL	NULL	NULL	NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.5 การเพิ่มข้อมูลเข้าใน Table bank_account

```
-- INSERT CARD TYPE DATA
INSERT INTO CARD_TYPE
    VALUES(1,'VISA'),
          (2,'MasterCard'),
          (3,'UnionPay');
```

cardtype_no	cardtype_name
1	VISA
2	MasterCard
3	UnionPay
NULL	NULL

รูปภาพที่ 10.2.6 การเพิ่มข้อมูลเข้าใน Table card_type

```
-- INSERT DEBIT CARD DATA
INSERT INTO DEBIT_CARD
    VALUES('4556550581359894','0432156870',1,'123456','2024-02-28','358'),
          ('5134801471721117','0009876543',2,'482103','2026-02-28','507'),
          ('6223058109120960','7001234560',3,'769840','2024-10-31','461'),
          ('4916764139281667','2109876500',1,'215367','2025-11-30','938'),
          ('5365414708158424','2109876500',2,'987123','2025-09-30','161'),
          ('5543504073991592','1098765400',2,'938014','2028-09-30','362'),
          ('4002587986033500','0098765432',1,'184250','2026-01-31','109'),
          ('5317215847434546','1023045678',2,'958706','2029-06-30','683'),
          ('4539336639593508','9009876542',1,'453675','2028-04-30','423'),
          ('4532172815285628','0000000123',1,'000000','2026-04-30','834'),
          ('4916346932452065','4321567890',1,'322413','2026-11-30','848'),
          ('5281719951806821','4321567890',2,'123456','2027-03-31','703');
```

card_num	acc_num	cardtype_no	card_PIN	exp_date	CVV
4002587986033500	0098765432	1	184250	2026-01-31	109
4532172815285628	0000000123	1	000000	2026-04-30	834
4539336639593508	9009876542	1	453675	2028-04-30	423
4556550581359894	0432156870	1	123456	2024-02-28	358
4916346932452065	4321567890	1	322413	2026-11-30	848
4916764139281667	2109876500	1	215367	2025-11-30	938
5134801471721117	0009876543	2	482103	2026-02-28	507
5281719951806821	4321567890	2	123456	2027-03-31	703
5317215847434546	1023045678	2	958706	2029-06-30	683
5365414708158424	2109876500	2	987123	2025-09-30	161
5543504073991592	1098765400	2	938014	2028-09-30	362
6223058109120960	7001234560	3	769840	2024-10-31	461
NULL	NULL	NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.7 การเพิ่มข้อมูลเข้าใน Table debit_card

```
-- INSERT EMPLOYEE DATA
INSERT INTO EMPLOYEE(emp_fname,emp_lname)
    VALUES('Nattanan','Srisuk'),
          ('Phumiporn','Orndee'),
          ('Surat','Wisut'),
          ('Wanchai','Misawat'),
          ('Rachnee','Chinsawat'),
          ('Somsri','Jaidee'),
          ('Kritima','Sukjai'),
          ('Natchanon','Inthong'),
          ('Arnanong','Phandang'),
          ('Patcharee','Sombat');
```

emp_id	emp_fname	emp_lname
1	Nattanan	Srisuk
2	Phumiporn	Orndee
3	Surat	Wisut
4	Wanchai	Misawat
5	Rachnee	Chinsawat
6	Somsri	Jaidee
7	Kritima	Sukjai
8	Natchanon	Inthong
9	Arnanong	Phandang
10	Patcharee	Sombat
NULL	NULL	NULL

รูปภาพที่ 10.2.8 การเพิ่มข้อมูลเข้าใน Table employee

```
-- INSERT ATM DATA
INSERT INTO ATM(emp_id,atm_status,atm_address_line,atm_province,atm_zip_code)
    VALUES(8, 'Y', 'Main Branch, 123 Main Street', 'Bangkok, Bangkok', '10100'),
          (8, 'Y', 'Central Latphrao, 428 Phonyothin Rd.', 'Chatuchak Bangkok', '10900'),
          (2, 'Y', 'Central Plaza, 456 Central Rd.', 'Mueng Chiang Mai, Chiang Mai', '50000'),
          (1, 'Y', 'Mega Mall, 789 Mega Avenue', 'Mueng Phuket, Phuket', '83000'),
          (3, 'Y', 'Night Market, 246 Night Market Street', 'Mueng Chiang Rai, Chiang Rai', '50000'),
          (5, 'Y', 'Riverside, 354 Riverside Rd.', 'Mueng Nonthaburi, Nonthaburi', '11000'),
          (4, 'Y', 'Beach Resort, 789 Beach Rd.', 'Hua Hin, Prachuap Khiri Khan', '77110'),
          (9, 'Y', 'Central Plaza, 1201 Central Avenue', 'Pattaya, Chon Buri', '20150'),
          (7, 'Y', 'Paradise Mall, 908 Paradise Street', 'Hatyai, Songkhla', '90110'),
          (6, 'Y', 'Airport Terminal, 101 Mueang Ubon Ratchathani District', 'Mueng Ubon Ratchathani, Ubon Ratchathani', '34000'),
          (10, 'Y', 'Floating Market, 222 Floating Market Lane', 'Mueng Sukhothai, Sukhothai', '13000');
```

atm_id	emp_id	atm_status	atm_address_line	atm_province	atm_zip_code
1	8	Y	Main Branch, 123 Main Street	Bangkok, Bangkok	10100
2	8	Y	Central Latphrao, 428 Phonyothin Rd.	Chatuchak Bangkok	10900
3	2	Y	Central Plaza, 456 Central Rd.	Mueng Chiang Mai, Chiang Mai	50000
4	1	Y	Mega Mall, 789 Mega Avenue	Mueng Phuket, Phuket	83000
5	3	Y	Night Market, 246 Night Market Street	Mueng Chiang Rai, Chiang Rai	50000
6	5	Y	Riverside, 354 Riverside Rd.	Mueng Nonthaburi, Nonthaburi	11000
7	4	Y	Beach Resort, 789 Beach Rd.	Hua Hin, Prachuap Khiri Khan	77110
8	9	Y	Central Plaza, 1201 Central Avenue	Pattaya, Chon Buri	20150
9	7	Y	Paradise Mall, 908 Paradise Street	Hatyai, Songkhla	90110
10	6	Y	Airport Terminal, 101 Mueang Ubon Ratchathan...	Mueng Ubon Ratchathani, Ubon Ratchathani	34000
11	10	Y	Floating Market, 222 Floating Market Lane	Mueng Sukhothai, Sukhothai	13000
NULL	NULL	NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.9 การเพิ่มข้อมูลเข้าใน Table ATM

```
-- INSERT TRANSFER DATA
INSERT INTO TRANSFER(acc_num,transfer_date,transfer_amount,reciever_acc)
    VALUES('0432156870','2023-07-01 07:30:00',100.00,'0341758378'),
          ('1098765400','2023-07-01 10:04:20',40.50,'4123097993'),
          ('0009876543','2023-07-01 17:20:42',55.00,'0432156870'),
          ('1098765400','2023-07-02 17:48:20',200.50,'8903246715'),
          ('7001234560','2023-07-02 21:00:05',300.60,'9012345678'),
          ('7001234560','2023-07-02 23:34:45',400.90,'0456789032'),
          ('2109876500','2023-07-03 15:33:12',100.00,'0000000123'),
          ('1098765400','2023-07-04 16:01:55',2950.00,'8765432109'),
          ('4321567890','2023-07-05 07:14:21',30000.00,'4123097993'),
          ('4321567890','2023-07-07 07:17:27',150000.00,'0397725482'),
          ('0000000123','2023-07-09 10:10:10',100.00,'6827539788'),
          ('0909876541','2023-07-10 06:30:00',2.50,'4123097993'),
          ('0009876543','2023-07-10 09:34:30',500.00,'9009876542'),
          ('9009876542','2023-07-10 21:45:10',250.00,'5032968174'),
          ('1098765400','2023-07-12 12:00:00',60.70,'8765432109'),
          ('0098765432','2023-07-13 04:00:00',99.00,'5032968174'),
          ('1098765400','2023-07-13 17:48:59',88.64,'9012345678'),
          ('4321567890','2023-07-18 14:17:25',1.00,'0098765432'),
          ('2109876500','2023-07-19 05:05:05',4200.00,'0956237814'),
          ('4321567890','2023-07-20 22:30:15',49500.00,'3257901846');
```

transfer_id	acc_num	transfer_date	transfer_amount	reciever_acc
1	0432156870	2023-07-01 07:30:00	100.00	0341758378
2	1098765400	2023-07-01 10:04:20	40.50	4123097993
3	0009876543	2023-07-01 17:20:42	55.00	0432156870
4	1098765400	2023-07-02 17:48:20	200.50	8903246715
5	7001234560	2023-07-02 21:00:05	300.60	9012345678
6	7001234560	2023-07-02 23:34:45	400.90	0456789032
7	2109876500	2023-07-03 15:33:12	100.00	0000000123
8	1098765400	2023-07-04 16:01:55	2950.00	8765432109
9	4321567890	2023-07-05 07:14:21	30000.00	4123097993
10	4321567890	2023-07-07 07:17:27	150000.00	0397725482
11	0000000123	2023-07-09 10:10:10	100.00	6827539788
12	0909876541	2023-07-10 06:30:00	2.50	4123097993
13	0009876543	2023-07-10 09:34:30	500.00	9009876542
14	9009876542	2023-07-10 21:45:10	250.00	5032968174
15	1098765400	2023-07-12 12:00:00	60.70	8765432109
16	0098765432	2023-07-13 04:00:00	99.00	5032968174
17	1098765400	2023-07-13 17:48:59	88.64	9012345678
18	4321567890	2023-07-18 14:17:25	1.00	0098765432
19	2109876500	2023-07-19 05:05:05	4200.00	0956237814
20	4321567890	2023-07-20 22:30:15	49500.00	3257901846
NULL	NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.10 การเพิ่มข้อมูลเข้าใน Table transfer

```
-- INSERT WITHDRAWAL DATA
INSERT INTO WITHDRAWAL
    VALUES ('WDR235K','0009876543',2,'2023-07-01 15:15:00',500.00),
           ('P9L4G2R','4321567890',1,'2023-07-01 15:15:00',10000.00),
           ('X7M1N8J','2109876500',3,'2023-07-01 15:15:00',100.00),
           ('3B6YF9A','2109876500',3,'2023-07-01 15:15:00',200.00),
           ('K2D7T4C','1098765400',8,'2023-07-01 15:15:00',400.00),
           ('R1S5W6X','4321567890',1,'2023-07-01 15:15:00',5000.00),
           ('Q8E2V3U','9009876542',10,'2023-07-01 15:15:00',600.00),
           ('6N9H4JL','1023045678',6,'2023-07-01 15:15:00',30000.00),
           ('G5F2P7T','0009876543',2,'2023-07-01 15:15:00',100.00),
           ('4R1D9C6','0098765432',9,'2023-07-01 15:15:00',100.00),
           ('Y8Q3M2B','1098765400',11,'2023-07-01 15:15:00',100.00),
           ('E9K5G7W','0432156870',1,'2023-07-01 15:15:00',300.00),
           ('E9K5G7W','1023045678',5,'2023-07-01 15:15:00',20000.00),
           ('1A3T4N6','7001234560',4,'2023-07-01 15:15:00',100.00),
           ('U2H9J8R','1098765400',8,'2023-07-21 09:30:05',2000.00);
```

withdraw_code	acc_num	atm_id	withdraw_date	withdraw_amount
1A3T4N6	7001234560	4	2023-07-01 15:15:00	100.00
3B6YF9A	2109876500	3	2023-07-01 15:15:00	200.00
4R1D9C6	0098765432	9	2023-07-01 15:15:00	100.00
6N9H4JL	1023045678	6	2023-07-01 15:15:00	30000.00
E9K5G7W	0432156870	1	2023-07-01 15:15:00	300.00
E9K5G7W	1023045678	5	2023-07-01 15:15:00	20000.00
G5F2P7T	0009876543	2	2023-07-01 15:15:00	100.00
K2D7T4C	1098765400	8	2023-07-01 15:15:00	400.00
P9L4G2R	4321567890	1	2023-07-01 15:15:00	10000.00
Q8E2V3U	9009876542	10	2023-07-01 15:15:00	600.00
R1S5W6X	4321567890	1	2023-07-01 15:15:00	5000.00
U2H9J8R	1098765400	8	2023-07-21 09:30:05	2000.00
WDR235K	0009876543	2	2023-07-01 15:15:00	500.00
X7M1N8J	2109876500	3	2023-07-01 15:15:00	100.00
Y8Q3M2B	1098765400	11	2023-07-01 15:15:00	100.00
NULL	NULL	NULL	NULL	NULL

รูปภาพที่ 10.2.11 การเพิ่มข้อมูลเข้าใน Table customer

10.3 Data Query Language (DQL)

➢ **ชุดคำสั่งที่ 1** : SQL Built-in Function

รายงานรายชื่อลูกค้าและเลขบัญชีที่ทำการเปิดบัญชีประเภท **"ออมทรัพย์"**	
จุดประสงค์	เพื่อแสดงข้อมูลของลูกค้าและเลขบัญชีที่เปิดบัญชีประเภท **"ออมทรัพย์"** ซึ่งประกอบด้วย ชื่อลูกค้า, เลขบัญชี, ประเภทของบัญชี โดยให้เรียงลำดับตาม**ชื่อจาก A-Z**

คำสั่ง SQL

```
-- 1st command set (Build-in Function)
SELECT CONCAT(c.cus_fname,' ',c.cus_lname) AS customer_name, b.acc_num, act.acctype_name
FROM bank_account b JOIN customer c ON c.customer_no = b.customer_no
                    JOIN account_type act ON b.acctype_no = act.acctype_no
WHERE b.acctype_no = 1
ORDER BY customer_name;
```

ผลลัพธ์จาก MySQL Server

customer_name	acc_num	acctype_name
Darunee Rattanakosin	0098765432	Saving Account
Kunthira Chaiyapruk	1023045678	Saving Account
Napat Thongchai	0009876543	Saving Account
Natthapong Songtham	0000000123	Saving Account
Niran Thamrong	4321567890	Saving Account
Patchara Panom	1098765400	Saving Account
Saran Wongthanakit	0432156870	Saving Account
Sirima Khongchai	7001234560	Saving Account
Surachai Boonsri	9009876542	Saving Account
Tawan Somchai	2109876500	Saving Account

➢ **ชุดคำสั่งที่ 2** : SQL Aggregate Function

รายงานรายชื่อลูกค้าและเลขบัญชีที่ทำรายการ**โอนเงิน**ในช่วง**วันที่ 1-15**	
จุดประสงค์	เพื่อแสดงข้อมูลของลูกค้าและเลขบัญชีที่รายการในช่วง**วันที่ 1-15** ซึ่งจะมีการนับจำนวนการ**โอนเงิน**และ**ยอดการโอนทั้งหมด** โดยให้เรียงลำดับตาม**ชื่อจาก A-Z**

คำสั่ง SQL

```
-- 2nd command set (Aggregate Function)
SELECT CONCAT(c.cus_fname,' ',c.cus_lname) AS customer_name, t.acc_num, COUNT(t.transfer_id) AS transfer_action, SUM(t.transfer_amount) AS total_transfer_amount
FROM transfer t JOIN bank_account b ON t.acc_num = b.acc_num
                JOIN customer c ON b.customer_no = c.customer_no
WHERE DAY(transfer_date) BETWEEN 1 AND 15
GROUP BY acc_num
ORDER BY customer_name;
```

ผลลัพธ์จาก MySQL Server

customer_name	acc_num	transfer_action	total_transfer_amount
Darunee Rattanakosin	0098765432	1	99.00
Kunthira Chaiyapruk	0909876541	1	2.50
Napat Thongchai	0009876543	2	555.00
Natthapong Songtham	0000000123	1	100.00
Niran Thamrong	4321567890	2	180000.00
Patchara Panom	1098765400	5	3340.34
Saran Wongthanakit	0432156870	1	100.00
Sirima Khongchai	7001234560	2	701.50
Surachai Boonsri	9009876542	1	250.00
Tawan Somchai	2109876500	1	100.00

➢ **ชุดคำสั่งที่ 3** : SQL JOIN

รายงานยอดคงเหลือในบัญชีของลูกค้าหลังการทำธุรกรรม**โอนเงิน**และ**ถอนเงินไม่ใช้บัตร**	
จุดประสงค์	เพื่อแสดงข้อมูลของเลขบัญชีที่มีประวัติการทำธุรกรรม**โอนเงินและถอนเงินไม่ใช้บัตร** ซึ่งประกอบด้วย เลขบัญชี, ชื่อลูกค้า, ยอดเงินคงเหลือ **หลังหักยอดโอนและยอดถอน** โดยให้เรียงลำดับตาม**ชื่อจาก A-Z**

คำสั่ง SQL

```
-- 3rd command set (JOIN)
SELECT b.acc_num, CONCAT(c.cus_fname,' ',c.cus_lname) AS customer_name , b.acc_balance - t.total_transfer - w.total_withdraw AS ramain_balance
FROM bank_account b JOIN customer c ON c.customer_no = b.customer_no
                    JOIN (SELECT acc_num, SUM(transfer_amount) AS total_transfer FROM transfer
                          GROUP BY acc_num) t ON t.acc_num = b.acc_num
                    JOIN (SELECT acc_num, SUM(withdraw_amount) AS total_withdraw FROM withdrawal
                          GROUP BY acc_num) w ON w.acc_num = b.acc_num
ORDER BY customer_name;
```

ผลลัพธ์จาก MySQL Server

acc_num	customer_name	ramain_balance
0098765432	Darunee Rattanakosin	1701.25
0009876543	Napat Thongchai	10845.90
4321567890	Niran Thamrong	8245499.00
1098765400	Patchara Panom	12410.16
0432156870	Saran Wongthanakit	4600.75
7001234560	Sirima Khongchai	88.50
9009876542	Surachai Boonsri	6650.00
2109876500	Tawan Somchai	5200.10

<u>อ้างอิง:</u>

- https://www.scb.co.th/th/about-us.html
- https://app.bot.or.th/BTWS_STAT/statistics/BOTWEBSTAT.aspx?reportID=904&language=th
- https://www.scb.co.th/th/shareholders/financial-information.html
- https://www.w3schools.com/sql/default.asp
- Lecture รายวิชา **ITDS221 – Database Management Systems**
- Lecture รายวิชา **ITDS222 – Database System Lab**
- รายชื่อสมมติจาก **ChatGPT**