

ผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกต่อผลสัมฤทธิ์การเรียนรู้จากใบงานปฏิบัติการ
ของนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ในรายวิชา Advanced Database
Effects of Rubric-Based Assessment on Laboratory Assignment Learning
Achievement of Second-Year Computer Science Students in the
Advanced Database Course

ผู้ช่วยศาสตราจารย์ ดร.พิเชษ วยะยลุน
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

บทคัดย่อ

การวิจัยในชั้นเรียนครั้งนี้มีวัตถุประสงค์เพื่อศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูบริก (Rubric Score) ในใบงานปฏิบัติการที่มีต่อผลสัมฤทธิ์การเรียนรู้ของนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ในรายวิชา Advanced Database โดยเปรียบเทียบผลการเรียนระหว่างกลุ่มที่ใช้ใบงานแบบเดิมและกลุ่มที่ใช้ใบงานที่มีการระบุเกณฑ์การให้คะแนนรายข้ออย่างชัดเจน กลุ่มตัวอย่างประกอบด้ว้นิสิตที่ลงทะเบียนเรียน รายวิชา Advanced Database จำนวน 2 กลุ่ม ได้แก่ กลุ่มควบคุมซึ่งใช้ใบงานแบบเดิม จำนวน 55 คน และ กลุ่มทดลองซึ่งใช้ใบงานที่มี Rubric Score รายข้อ จำนวน 55 คน เครื่องมือที่ใช้ในการวิจัย ได้แก่ ใบงานปฏิบัติการจำนวน 5 ใบงาน แบบบันทึกคะแนน และเกณฑ์การให้คะแนนแบบรูบริก การวิเคราะห์ข้อมูลใช้ ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน และการทดสอบความแตกต่างของค่าเฉลี่ยระหว่างกลุ่มด้วยสถิติ Independent Samples t-test

ผลการวิจัยพบว่า กลุ่มที่ใช้ใบงานที่มี Rubric Score มีคะแนนเฉลี่ยผลสัมฤทธิ์ทางการเรียนสูงกว่า กลุ่มที่ใช้ใบงานแบบเดิม โดยมีคะแนนเฉลี่ยเท่ากับ 9.29 คะแนน และ 7.11 คะแนน ตามลำดับ เมื่อทดสอบ ความแตกต่างของค่าเฉลี่ยพบว่าคะแนนของทั้งสองกลุ่มแตกต่างกันอย่างมีนัยสำคัญทางสถิติที่ระดับ .05 ($p < .001$) แสดงให้เห็นว่าการระบุเกณฑ์การให้คะแนนรายข้ออย่างชัดเจนช่วยให้นิสิตเข้าใจเป้าหมายการเรียนรู้ สามารถวางแผนการทำงานได้อย่างเหมาะสม และพัฒนาคุณภาพของชิ้นงานได้ดียิ่งขึ้น

ผลการวิจัยชี้ให้เห็นว่าการใช้เกณฑ์การให้คะแนนแบบรูบริกเป็นแนวทางหนึ่งที่สามารถส่งเสริม ผลสัมฤทธิ์ทางการเรียนในรายวิชาปฏิบัติการด้านฐานข้อมูลได้อย่างมีประสิทธิภาพ และสามารถนำไป ประยุกต์ใช้ในการจัดการเรียนการสอนรายวิชาอื่นที่มีลักษณะการประเมินจากชิ้นงานหรือใบงานปฏิบัติการได้

บทนำ

ปัจจุบันการจัดการเรียนการสอนในระดับอุดมศึกษามุ่งเน้นการพัฒนาผู้เรียนให้สามารถประยุกต์ใช้ความรู้เพื่อแก้ปัญหาและสร้างชิ้นงานได้จริง โดยเฉพาะในสาขาวิทยาการคอมพิวเตอร์ที่ผู้เรียนจำเป็นต้องมีทั้งความรู้ทางทฤษฎีและทักษะปฏิบัติในการพัฒนาระบบสารสนเทศและฐานข้อมูล การจัดการเรียนการสอนรายวิชา Advanced Database จึงเน้นการเรียนรู้ผ่านการปฏิบัติ (Learning by Doing) โดยให้นิสิตฝึกปฏิบัติผ่านใบงาน (Laboratory Assignment) เพื่อพัฒนาความรู้และทักษะด้านการออกแบบและพัฒนาระบบฐานข้อมูลขั้นสูง

อย่างไรก็ตาม จากประสบการณ์การสอนรายวิชา Advanced Database ของผู้วิจัย พบว่านิสิตจำนวนหนึ่งยังไม่สามารถทำใบงานได้ตามเป้าหมายที่กำหนด แม้ว่าจะได้รับโจทย์และคำอธิบายจากผู้สอนแล้วก็ตาม โดยนิสิตมักมีข้อสงสัยเกี่ยวกับขอบเขตของคำตอบ ระดับความสมบูรณ์ของงาน และเกณฑ์ที่ใช้ในการประเมิน ส่งผลให้คุณภาพของชิ้นงานและคะแนนใบงานแตกต่างกันค่อนข้างมาก นอกจากนี้ยังพบว่านิสิตบางส่วนไม่สามารถประเมินความครบถ้วนของงานตนเองก่อนส่งได้ ทำให้สูญเสียคะแนนในส่วนที่สามารถปรับปรุงได้หากได้รับข้อมูลเกี่ยวกับเกณฑ์การประเมินอย่างชัดเจน

แนวคิดการประเมินผลการเรียนรู้ด้วยเกณฑ์การให้คะแนนแบบรูบริก (Rubric Assessment) เป็นเครื่องมือที่ได้รับความนิยมในการศึกษาปัจจุบัน เนื่องจากช่วยให้ผู้เรียนเข้าใจเป้าหมายการเรียนรู้ เกณฑ์การประเมิน และระดับคุณภาพของผลงานที่คาดหวังได้อย่างชัดเจน (Andrade, 2005) โดยรูบริกเป็นเครื่องมือที่อธิบายรายละเอียดของเกณฑ์การประเมินในแต่ละระดับคะแนน ทำให้ผู้เรียนสามารถใช้เป็นแนวทางในการวางแผนและตรวจสอบคุณภาพของงานตนเองก่อนส่งได้ (Brookhart, 2013)

และงานวิจัยหลายฉบับรายงานว่าการใช้รูบริกช่วยส่งเสริมผลสัมฤทธิ์ทางการเรียนและคุณภาพของผลงานผู้เรียนได้อย่างมีนัยสำคัญ โดยเฉพาะในรายวิชาที่เน้นการปฏิบัติและการสร้างชิ้นงาน (Reddy and Andrade, 2010) นอกจากนี้ยังช่วยเพิ่มความโปร่งใสและความยุติธรรมในการประเมินผล ทำให้ผู้เรียนสามารถเข้าใจเหตุผลของคะแนนที่ได้รับและเกิดแรงจูงใจในการพัฒนาตนเองมากขึ้น (Panadero and Jonsson, 2013)

นอกจากนั้น Andrade และ Du พบว่าการใช้รูบริกร่วมกับการประเมินตนเองช่วยให้ผู้เรียนมีความเข้าใจต่อความคาดหวังของผู้สอนมากขึ้น และสามารถปรับปรุงคุณภาพของผลงานได้อย่างต่อเนื่อง (Andrade and Du, 2005) ขณะที่ Jonsson และ Svingby ได้สังเคราะห์งานวิจัยเกี่ยวกับการใช้รูบริกและพบว่ารูบริกช่วยเพิ่มความน่าเชื่อถือของการประเมิน รวมทั้งส่งเสริมการเรียนรู้ของผู้เรียนผ่านการให้ข้อมูลย้อนกลับที่ชัดเจนและเป็นระบบ (Jonsson and Svingby, 2007)

ในด้านการศึกษาระดับอุดมศึกษา Stevens และ Levi ระบุว่า การใช้รูบริกในการประเมินงานที่มีลักษณะซับซ้อน เช่น โครงงาน รายงาน หรือการปฏิบัติการทางวิชาชีพ ช่วยให้ผู้เรียนเข้าใจมาตรฐานของงานที่มีคุณภาพและสามารถวางแผนการทำงานได้อย่างมีประสิทธิภาพมากขึ้น (Stevens and Levi, 2013) ขณะที่ Dawson ชี้ให้เห็นว่ารูบริกเป็นเครื่องมือสำคัญในการสนับสนุนการเรียนรู้เชิงลึก (Deep Learning) เนื่องจากช่วยให้ผู้เรียนมองเห็นความเชื่อมโยงระหว่างผลลัพธ์การเรียนรู้และเกณฑ์การประเมินได้อย่างชัดเจน (Dawson, 2017)

สำหรับสาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ มีงานวิจัยที่พบว่าการใช้รูปrikในการประเมินงานเขียนโปรแกรม งานพัฒนาระบบ และโครงการด้านคอมพิวเตอร์ ช่วยให้ผู้เรียนสามารถพัฒนาคุณภาพของผลงานได้ดีขึ้น ลดความคลุมเครือในการประเมิน และเพิ่มความสอดคล้องของการให้คะแนนระหว่างผู้ประเมิน (Becker, 2016) นอกจากนี้การเปิดเผยเกณฑ์การประเมินล่วงหน้ายังช่วยให้ผู้เรียนสามารถกำหนดเป้าหมายการเรียนรู้และวางแผนการทำงานได้อย่างเหมาะสม ส่งผลต่อผลสัมฤทธิ์ทางการเรียนในภาพรวม

ในบริบทของการเรียนรู้ที่เน้นการปฏิบัติ งานวิจัยของ Fraile, Pardo และ Panadero พบว่าการใช้รูปrikร่วมกับการประเมินตนเองและการประเมินโดยเพื่อนช่วยส่งเสริมความรับผิดชอบต่อการเรียนรู้ของผู้เรียน และทำให้ผู้เรียนมีความสามารถในการสะท้อนคิด (Reflection) และปรับปรุงผลงานของตนเองได้ดียิ่งขึ้น ซึ่งเป็นทักษะสำคัญสำหรับนิสิตในสาขาวิทยาการคอมพิวเตอร์ที่ต้องเรียนรู้ผ่านการลงมือปฏิบัติและการแก้ปัญหาจริง (Fraile et al., 2017)

แม้ว่าจะมีการนำรูปrikมาใช้ในหลายสาขาวิชาและมีหลักฐานเชิงประจักษ์สนับสนุนประสิทธิผลของรูปrikอย่างต่อเนื่อง แต่ยังมีงานวิจัยเกี่ยวกับการใช้รูปrikในรายวิชาด้านฐานข้อมูลระดับปริญญาตรีค่อนข้างจำกัด โดยเฉพาะในบริบทของการเรียนการสอนรายวิชา Advanced Database สำหรับนิสิตสาขาวิทยาการคอมพิวเตอร์ ซึ่งเป็นรายวิชาที่เน้นการปฏิบัติและการประยุกต์ใช้ความรู้ในสถานการณ์จริง ดังนั้น ผู้วิจัยจึงสนใจศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูปrikในใบงานปฏิบัติการที่มีต่อผลสัมฤทธิ์การเรียนรู้ของนิสิต เพื่อใช้เป็นแนวทางในการพัฒนาการจัดการเรียนการสอนและการประเมินผลให้มีประสิทธิภาพมากยิ่งขึ้น

ผลการวิจัยที่ได้คาดว่าจะจะเป็นข้อมูลเชิงประจักษ์ที่สามารถนำไปใช้ในการปรับปรุงรูปแบบการจัดการเรียนการสอนในรายวิชาปฏิบัติการทางด้านคอมพิวเตอร์ ตลอดจนเป็นแนวทางสำหรับอาจารย์ผู้สอนในการออกแบบเกณฑ์การประเมินที่ช่วยส่งเสริมการเรียนรู้ของผู้เรียนได้อย่างมีประสิทธิภาพ

วัตถุประสงค์การวิจัย

1. เพื่อศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูปrikในใบงานปฏิบัติการที่มีต่อผลสัมฤทธิ์การเรียนรู้ของนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ในรายวิชา Advanced Database
2. เพื่อเปรียบเทียบผลสัมฤทธิ์การทำใบงานปฏิบัติการระหว่างนิสิตที่ได้รับใบงานแบบเดิมและนิสิตที่ได้รับใบงานที่มีเกณฑ์การให้คะแนนแบบรูปrik

ความสำคัญของผลงาน

รายวิชา Advanced Database เป็นรายวิชาที่มุ่งพัฒนาความรู้และทักษะด้านการออกแบบ พัฒนา และประยุกต์ใช้ระบบฐานข้อมูลขั้นสูง โดยเน้นการเรียนรู้ผ่านการปฏิบัติจริงด้วยใบงานปฏิบัติการ (Laboratory Assignment) ซึ่งเป็นเครื่องมือสำคัญในการส่งเสริมการเรียนรู้และการประเมินผลสัมฤทธิ์ของผู้เรียน อย่างไรก็ตาม จากประสบการณ์การจัดการเรียนการสอนที่ผ่านมา พบว่านิสิตบางส่วนยังขาดความเข้าใจเกี่ยวกับขอบเขตของงาน ระดับคุณภาพของคำตอบที่คาดหวัง และเกณฑ์การให้คะแนน ส่งผลให้คะแนนและคุณภาพของผลงานมีความแตกต่างกันค่อนข้างมาก

ผลงานวิจัยนี้มีความสำคัญต่อการพัฒนาการเรียนการสอนในรายวิชาปฏิบัติการทางด้านคอมพิวเตอร์ เนื่องจากการนำเกณฑ์การให้คะแนนแบบรูบริก (Rubric Score) มาใช้เป็นเครื่องมือในการสื่อสารความคาดหวังและเกณฑ์การประเมินให้ผู้เรียนรับทราบล่วงหน้าอย่างชัดเจน ช่วยให้ผู้เรียนสามารถวางแผนการทำงาน ประเมินคุณภาพผลงานของตนเอง และปรับปรุงงานให้สอดคล้องกับผลลัพธ์การเรียนรู้ที่กำหนดไว้ในด้านผู้เรียน ผลงานนี้ช่วยส่งเสริมการเรียนรู้เชิงรุก (Active Learning) และการเรียนรู้แบบกำกับตนเอง (Self-Regulated Learning) โดยผู้เรียนสามารถใช้เกณฑ์การประเมินเป็นแนวทางในการตรวจสอบความครบถ้วนและคุณภาพของงานก่อนส่ง ส่งผลให้เกิดความเข้าใจในเนื้อหาวิชามากขึ้น และช่วยยกระดับผลสัมฤทธิ์ทางการเรียนจากการปฏิบัติจริง

ในด้านผู้สอน ผลงานนี้ช่วยเพิ่มประสิทธิภาพในการประเมินผล ลดความคลุมเครือในการให้คะแนน และเพิ่มความโปร่งใสในการประเมิน เนื่องจากผู้เรียนสามารถรับทราบหลักเกณฑ์การให้คะแนนได้อย่างชัดเจน ตั้งแต่ก่อนเริ่มทำใบงาน ส่งผลให้การประเมินมีความเป็นธรรมและสามารถอธิบายผลการประเมินได้อย่างเป็นระบบ

ในระดับหลักสูตร ผลงานวิจัยนี้สามารถใช้เป็นแนวทางในการพัฒนารูปแบบการจัดการเรียนการสอน และการประเมินผลในรายวิชาปฏิบัติการอื่น ๆ ของหลักสูตรวิทยาการคอมพิวเตอร์ ตลอดจนสนับสนุนแนวทางการจัดการศึกษาแบบมุ่งผลลัพธ์การเรียนรู้ (Outcome-Based Education: OBE) และการประกันคุณภาพการศึกษาตามแนวทาง AUN-QA ที่ให้ความสำคัญกับความสอดคล้องระหว่างผลลัพธ์การเรียนรู้ กิจกรรมการเรียนรู้ และการประเมินผล

ผลการวิจัยที่ได้จะเป็นข้อมูลเชิงประจักษ์ที่ช่วยยืนยันประสิทธิผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกในการพัฒนาผลสัมฤทธิ์ทางการเรียนของนิสิต และสามารถนำไปประยุกต์ใช้ในการออกแบบการประเมินผลสำหรับรายวิชาปฏิบัติการด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศในอนาคตได้อย่างมีประสิทธิภาพ

สมมติฐาน

สมมติฐานศูนย์: H_0

คะแนนเฉลี่ยผลสัมฤทธิ์การทำใบงานปฏิบัติการของนิสิตที่ได้รับใบงานแบบเดิมและนิสิตที่ได้รับใบงานที่มีเกณฑ์การให้คะแนนแบบรูบริกไม่แตกต่างกันโดยแสดงดังสมการที่ 1

$$H_0: \mu_1 = \mu_2 \quad (1)$$

โดยที่

μ_1 = ค่าเฉลี่ยคะแนนของกลุ่มที่ใช้ใบงานแบบเดิม

μ_2 = ค่าเฉลี่ยคะแนนของกลุ่มที่ใช้ใบงานที่มี Rubric Score รายข้อ

สมมติฐานทางเลือก: H_1

คะแนนเฉลี่ยผลสัมฤทธิ์การทำใบงานปฏิบัติการของนิสิตที่ได้รับใบงานที่มีเกณฑ์การให้คะแนนแบบรูบริกสูงกว่านิสิตที่ได้รับใบงานแบบเดิมโดยแสดงดังสมการที่ 2

$$H_1: \mu_2 > \mu_1 \quad (2)$$

โดยที่

μ_1 = ค่าเฉลี่ยคะแนนของกลุ่มที่ใช้ใบงานแบบเดิม

μ_2 = ค่าเฉลี่ยคะแนนของกลุ่มที่ใช้ใบงานที่มี Rubric Score รายข้อ

ขอบเขตการวิจัย

1. ขอบเขตด้านประชากรและกลุ่มเป้าหมายได้แก่ นิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ที่ลงทะเบียนเรียนรายวิชา Advanced Database จำนวน 110 คน แบ่งเป็น 2 กลุ่มเรียน กลุ่มละ 55 คน โดยมีรายละเอียดดังนี้

- กลุ่มที่ 1 จำนวน 55 คน ใช้ใบงานปฏิบัติการแบบเดิมที่ไม่มีการระบุเกณฑ์การให้คะแนนแบบรูบริก
- กลุ่มที่ 2 จำนวน 55 คน ใช้ใบงานปฏิบัติการที่มีการระบุเกณฑ์การให้คะแนนแบบรูบริกในทุกข้อของใบงาน

2. ขอบเขตด้านเนื้อหาเป็นเนื้อหาในรายวิชา Advanced Database ในส่วนของใบงานปฏิบัติการจำนวน 5 ใบงาน ประกอบด้วย

1. การสร้าง Stored Procedure ชั้นแนะนำ (Introduction to Stored Procedure)
2. การสร้างและการใช้งาน Parameters
3. Declare Variable และ IF...ELSE
4. LOOP และ CASE WHEN
5. View Table on Database

โดยใบงานแต่ละชุดมีคะแนนเต็ม 10 คะแนน และใช้เกณฑ์การประเมินเดียวกันทั้งสองกลุ่มเรียน

3. ขอบเขตด้านตัวแปรที่ศึกษาประกอบด้วย

1. ตัวแปรต้น (Independent Variable) โดยการใช้เกณฑ์การให้คะแนนแบบรูบริก (Rubric Score) ในใบงานปฏิบัติการคือ

- กลุ่มที่ 1 ใช้ใบงานแบบเดิมที่ไม่มี Rubric Score
- กลุ่มที่ 2 ใช้ใบงานที่มี Rubric Score ระบุเกณฑ์การให้คะแนนในแต่ละข้ออย่างชัดเจน

2. ตัวแปรตาม (Dependent Variable) คือผลสัมฤทธิ์การเรียนรู้จากคะแนนใบงานปฏิบัติการของนิสิตในรายวิชา Advanced Database

4. ขอบเขตด้านการจัดการเรียนการสอน เพื่อควบคุมตัวแปรแทรกซ้อน ผู้วิจัยกำหนดให้ทั้งสองกลุ่มเรียนได้รับการจัดการเรียนการสอนในรูปแบบเดียวกัน ได้แก่

- ผู้สอนคนเดียวกัน
- เนื้อหาการสอนเดียวกัน
- เอกสารประกอบการสอนเดียวกัน

- ชั่วโมงเรียนภาคบรรยายและภาคปฏิบัติเท่ากัน
- วิธีการสอนและกิจกรรมการเรียนรู้เหมือนกันทุกประการ

โดยมีความแตกต่างเพียงประเด็นเดียว คือ การระบุเกณฑ์การให้คะแนนแบบรูบริกในใบงานปฏิบัติการของกลุ่มที่ 2

5. ขอบเขตด้านระยะเวลา โดยการวิจัยดำเนินการในภาคการศึกษาที่ 2 โดยเก็บรวบรวมข้อมูลจากคะแนนใบงานปฏิบัติการทั้ง 5 ใบงานตลอดภาคการศึกษา

ประโยชน์ที่ได้รับ

1. ได้ทราบผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกในใบงานปฏิบัติการที่มีต่อผลสัมฤทธิ์การเรียนรู้ของนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ในรายวิชา Advanced Database

2. นิสิตสามารถเข้าใจเกณฑ์การประเมินและความคาดหวังของผู้สอนได้ชัดเจนมากขึ้น ส่งผลให้สามารถวางแผนการทำงาน ตรวจสอบความครบถ้วนของงาน และพัฒนาคุณภาพของใบงานปฏิบัติการได้อย่างมีประสิทธิภาพ

3. ผู้สอนสามารถใช้เกณฑ์การให้คะแนนแบบรูบริกเป็นเครื่องมือในการประเมินผลการเรียนรู้ที่มีความชัดเจน โปร่งใส และเป็นธรรมมากยิ่งขึ้น ลดความคลุมเครือในการให้คะแนนและการอธิบายผลการประเมินแก่ผู้เรียน

4. ผลการวิจัยสามารถใช้เป็นแนวทางในการออกแบบใบงานปฏิบัติการและเกณฑ์การประเมินสำหรับรายวิชาอื่นที่มีลักษณะการเรียนรู้ผ่านการปฏิบัติในสาขาวิทยาการคอมพิวเตอร์และสาขาที่เกี่ยวข้อง

5. หลักสูตรวิทยาการคอมพิวเตอร์สามารถนำผลการวิจัยไปใช้ในการพัฒนากระบวนการจัดการเรียนการสอนและการประเมินผลให้สอดคล้องกับแนวทางการจัดการศึกษาแบบมุ่งผลลัพธ์การเรียนรู้ (Outcome-Based Education: OBE)

6. ได้ข้อมูลเชิงประจักษ์ที่สนับสนุนการพัฒนาคุณภาพการเรียนการสอนและการประกันคุณภาพการศึกษา รวมทั้งสามารถใช้เป็นแนวทางในการจัดทำผลงานทางวิชาการด้านการวิจัยในชั้นเรียนและการพัฒนาการเรียนการสอนในอนาคต

ระเบียบวิธีวิจัย

การวิจัยครั้งนี้เป็นการวิจัยเชิงปริมาณ (Quantitative Research) โดยใช้รูปแบบการวิจัยกึ่งทดลอง (Quasi-Experimental Research) เพื่อศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกในใบงานปฏิบัติการที่มีต่อผลสัมฤทธิ์การเรียนรู้ของนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ในรายวิชา Advanced Database

1) ประชากรและกลุ่มเป้าหมาย โดยประชากรที่ใช้ในการวิจัย ได้แก่ นิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ที่ลงทะเบียนเรียนรายวิชา Advanced Database ในภาคการศึกษาที่ 2 จำนวน 110 คน กลุ่มเป้าหมายแบ่งออกเป็น 2 กลุ่มเรียน ดังนี้

- กลุ่มที่ 1 จำนวน 55 คน ใช้ใบงานปฏิบัติการแบบเดิมที่ไม่มีเกณฑ์การให้คะแนนแบบรูบริก

- กลุ่มที่ 2 จำนวน 55 คน ใช้ใบงานปฏิบัติการที่มีเกณฑ์การให้คะแนนแบบรูบริกในทุกข้อของใบงาน

2) รูปแบบการวิจัย โดยการวิจัยครั้งนี้ใช้รูปแบบ Nonequivalent Control Group Design โดยข้อกำหนดดังตารางที่ 1

ตารางที่ 1 ข้อกำหนดรูปแบบการวิจัย

กลุ่ม	วิธีการจัดการเรียนรู้	การใช้ Rubric
กลุ่มที่ 1	เรียนตามปกติ	ไม่มี Rubric
กลุ่มที่ 2	เรียนตามปกติ	มี Rubric

ทั้งสองกลุ่มได้รับการสอนโดยผู้สอนคนเดียวกัน ใช้เอกสารประกอบการสอน เนื้อหา ชั่วโมงเรียน และกิจกรรมการเรียนรู้เหมือนกันทุกประการ

3) เครื่องมือที่ใช้ในการวิจัย โดยเครื่องมือที่ใช้ในการวิจัยประกอบด้วย

3.1 ใบงานปฏิบัติการรายวิชา Advanced Database จำนวน 5 ใบงาน ได้แก่

1. การสร้าง Stored Procedure ขึ้นแนะนำ
2. การสร้างและการใช้งาน Parameters
3. Declare Variable และ IF...ELSE
4. LOOP และ CASE WHEN
5. View Table on Database

ใบงานแต่ละชุดมีคะแนนเต็ม 10 คะแนน โดยแสดงดังภาคผนวก ก

3.2 เกณฑ์การให้คะแนนแบบรูบริก (Rubric Score) ใช้สำหรับกลุ่มทดลอง โดยระบุรายละเอียดการให้คะแนนในแต่ละข้อของใบงานอย่างชัดเจน เพื่อให้นิสิตทราบเกณฑ์การประเมินก่อนปฏิบัติงาน โดยแสดงตัวอย่าง Rubric Score ใบงานที่ 1 – 5 ดังตารางที่ 2 – 6

ตารางที่ 2 ตัวอย่าง Rubric Score ใบงานที่ 1 ข้อที่ 9

คะแนน	เกณฑ์
2.0	โค้ดถูกต้องทุกส่วน ตรงตามเงื่อนไข WHERE + เลือกเฉพาะ field
1.5	โค้ดถูกต้องแต่มี formatting หรือการจัดประกอบคำสั่งที่ไม่เป็นมาตรฐาน
1.0	โค้ดทำงานได้แต่เงื่อนไขไม่ครบ เช่น เลือกคอลัมน์ผิด
0.5	โค้ดสร้าง SP แต่รันไม่ได้
0	ไม่ส่ง

ตารางที่ 3 ตัวอย่าง Rubric Score ใบงานที่ 2 ข้อที่ 4

ระดับ	เกณฑ์
0.4	ใช้ FOUND_ROWS() ถูกต้อง
0.3	โค้ดถูกต้องแต่ format ผิด
0.1	ใช้ FOUND_ROWS ไม่ถูกต้อง
0	ไม่ส่ง

ตารางที่ 4 ตัวอย่าง Rubric Score ใบงานที่ 3 ข้อที่ 3

คะแนน	เกณฑ์
1.5	SP ถูกต้อง คำนวณส่วนลด 30% เงื่อนไข IF >300 ถูกต้อง แสดงผลลัพธ์ถูกต้อง
1.0	SP ถูกต้องแต่ logic มี error เล็กน้อย เช่น ชื่อ variable ไม่ตรง
0.5	เงื่อนไขผิดหรือไม่คำนวณส่วนลด
0	ไม่ส่ง

ตารางที่ 5 ตัวอย่าง Rubric Score ใบงานที่ 4 ข้อที่ 2

ระดับ	เกณฑ์
3.0	Temp Table ถูกต้อง LOOP ทำงานครบ
2.0	LOOP ทำงานแต่ ItemDetail ผิดเล็กน้อย
1.0	Temp Table หรือ LOOP ผิด
0	ไม่ส่ง

ตารางที่ 6 ตัวอย่าง Rubric Score ใบงานที่ 5 ข้อที่ 3

ช่วงคะแนน	เกณฑ์ละเอียด
0.77-0.90 (A)	สร้าง View/คำสั่งตามโจทย์ได้ครบ: JOIN 3 ตาราง (customer05 + country + audit) + เงื่อนไข USED ≤ 300,000 และประเทศ United States + GROUP BY ลูกค้า + COUNT(audit) ตั้งชื่อคอลัมน์ ITEM_AUDIT + ORDER BY CUSTOMER_ID DESC
0.63-0.76 (B)	โค้ดทำงานได้และเกือบครบ ผิดเล็กน้อย 1 จุด (เช่น alias ITEM_AUDIT ไม่ตรง, order ผิด)
0.36-0.62 (C)	โค้ดรันได้แต่ขาดส่วนสำคัญ (เช่นไม่ group by/ไม่ count/กรองประเทศผิด)
0.00-0.35 (D)	สร้างไม่ได้/ผิดทั้งหมด/ไม่ส่ง

3.3 แบบบันทึกคะแนนใบงาน โดยใช้สำหรับบันทึกผลคะแนนของนิสิตในแต่ละใบงาน และใช้ในการวิเคราะห์ข้อมูลทางสถิติ โดยแสดงดังภาคผนวก ข

4) การเก็บรวบรวมข้อมูล โดยผู้วิจัยดำเนินการจัดการเรียนการสอนตามแผนการสอนรายวิชา Advanced Database ตลอดภาคการศึกษาที่ 2 โดยดำเนินการดังนี้

1. จัดการเรียนการสอนให้กับทั้งสองกลุ่มเรียนด้วยเนื้อหาและกิจกรรมเดียวกัน
2. มอบหมายใบงานปฏิบัติการจำนวน 5 ใบงาน
3. กลุ่มที่ 1 ใช้ใบงานแบบเดิม
4. กลุ่มที่ 2 ใช้ใบงานที่มี Rubric Score
5. บันทึกคะแนนใบงานของนิสิตทุกคน
6. รวบรวมคะแนนเพื่อใช้ในการวิเคราะห์ผลการวิจัย

การวิเคราะห์ข้อมูล

การวิเคราะห์ข้อมูลผู้วิจัยวิเคราะห์ข้อมูลด้วยโปรแกรมสำเร็จรูปทางสถิติ โดยใช้สถิติดังต่อไปนี้

1. สถิติเชิงพรรณนา (Descriptive Statistics)

- ค่าเฉลี่ย (Mean)
- ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)
- ค่าร้อยละ (Percentage)

2. สถิติเชิงอนุมาน (Inferential Statistics)

ใช้สถิติ Independent Samples t-test เพื่อเปรียบเทียบคะแนนเฉลี่ยของนิสิตระหว่าง

- กลุ่มที่ใช้ใบงานแบบเดิม
- กลุ่มที่ใช้ใบงานที่มีเกณฑ์การให้คะแนนแบบรูบริก

โดยกำหนดระดับนัยสำคัญทางสถิติที่ .05

ผลการวิจัย

ผลการวิเคราะห์ข้อมูลตามวัตถุประสงค์ของการวิจัย ดังนี้

1. จากการวิเคราะห์ข้อมูลคะแนนใบงานปฏิบัติการของนิสิตชั้นปีที่ 2 สาขาวิชาการคอมพิวเตอร์ รายวิชา Advanced Database จากตารางที่ 7 พบว่ากลุ่มที่ได้รับใบงานปฏิบัติการที่มีเกณฑ์การให้คะแนนแบบรูบริกมีคะแนนเฉลี่ยเท่ากับ 9.28 คะแนน (SD = 0.56) สูงกว่ากลุ่มที่ได้รับใบงานแบบเดิมซึ่งมีคะแนนเฉลี่ยเท่ากับ 7.38 คะแนน (SD = 1.66)

2. ผลการทดสอบความแตกต่างของคะแนนเฉลี่ยด้วยสถิติ Independent Samples t-test จากตารางที่ 8 พบว่าคะแนนของทั้งสองกลุ่มแตกต่างกันอย่างมีนัยสำคัญทางสถิติที่ระดับ .05 ($t = -8.048$, $p < .001$) แสดงให้เห็นว่าการใช้เกณฑ์การให้คะแนนแบบรูบริกช่วยส่งเสริมผลสัมฤทธิ์ทางการเรียนของนิสิตได้อย่างมีประสิทธิภาพ

ตารางที่ 7 เปรียบเทียบคะแนนเฉลี่ยใบงานระหว่างกลุ่ม

รายการ	กลุ่มที่ 1 (ไม่มี Rubric)	กลุ่มที่ 2 (มี Rubric)
จำนวน (n)	55	55
คะแนนเฉลี่ย (Mean)	7.38	9.28
ส่วนเบี่ยงเบนมาตรฐาน (SD)	1.66	0.56
คะแนนต่ำสุด	0.70	7.30
คะแนนสูงสุด	9.20	10.00

ตารางที่ 8 ผลการทดสอบ t-test

ตัวแปร	t	p-value
คะแนนใบงาน	-8.048	< .001

สรุปและอภิปรายผล

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกในใบงานปฏิบัติการที่มีต่อผลสัมฤทธิ์การเรียนรู้ของนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ในรายวิชา Advanced Database โดยเปรียบเทียบผลการเรียนระหว่างนิสิตที่ได้รับใบงานแบบเดิมและนิสิตที่ได้รับใบงานที่มีการระบุเกณฑ์การให้คะแนนแบบรูบริกอย่างชัดเจน กลุ่มเป้าหมายประกอบด้วยนิสิตจำนวน 110 คน แบ่งเป็น 2 กลุ่ม กลุ่มละ 55 คน ซึ่งได้รับการจัดการเรียนการสอน เนื้อหา เอกสารประกอบการสอน และกิจกรรมการเรียนรู้เหมือนกันทุกประการ แตกต่างกันเฉพาะการใช้เกณฑ์การให้คะแนนแบบรูบริกในใบงานปฏิบัติการ ผลการวิจัยพบว่านิสิตกลุ่มที่ได้รับใบงานที่มีเกณฑ์การให้คะแนนแบบรูบริกมีคะแนนเฉลี่ยใบงานปฏิบัติการเท่ากับ 9.28 คะแนน สูงกว่านิสิตกลุ่มที่ได้รับใบงานแบบเดิมซึ่งมีคะแนนเฉลี่ยเท่ากับ 7.38 คะแนน โดยผลการทดสอบด้วยสถิติ Independent Samples t-test พบว่าคะแนนของทั้งสองกลุ่มแตกต่างกันอย่างมีนัยสำคัญทางสถิติที่ระดับ .05 ($p < .001$) และมีขนาดอิทธิพลในระดับสูงมาก สะท้อนให้เห็นว่าการใช้เกณฑ์การให้คะแนนแบบรูบริกสามารถส่งเสริมผลสัมฤทธิ์ทางการเรียนของนิสิตได้อย่างมีประสิทธิภาพ

ผลการวิจัยดังกล่าวสามารถอธิบายได้ว่าการระบุเกณฑ์การให้คะแนนรายข้ออย่างชัดเจนช่วยให้นิสิตเข้าใจเป้าหมายการเรียนรู้ ขอบเขตของงาน และระดับคุณภาพของผลงานที่คาดหวังได้ดียิ่งขึ้น ส่งผลให้นิสิตสามารถวางแผนการทำงาน ตรวจสอบความครบถ้วนของงาน และปรับปรุงคุณภาพของชิ้นงานก่อนส่งได้อย่างเหมาะสม นอกจากนี้ยังช่วยลดความคลุมเครือในการตีความโจทย์และลดความไม่แน่นอนเกี่ยวกับหลักเกณฑ์การประเมิน ทำให้นิสิตสามารถมุ่งเน้นไปยังองค์ประกอบสำคัญที่ผู้สอนต้องการวัดได้อย่างตรงจุด ผลการศึกษานี้สอดคล้องกับแนวคิดของ Andrade (2005) ที่กล่าวว่ารูบริกเป็นเครื่องมือที่ช่วยสื่อสารความคาดหวังของผู้สอนและส่งเสริมการเรียนรู้ของผู้เรียนผ่านการกำหนดเกณฑ์การประเมินที่ชัดเจน รวมทั้งสอดคล้องกับงานวิจัยของ Reddy and Andrade (2010) ที่พบว่าการใช้รูบริกในระดับอุดมศึกษาช่วยเพิ่มผลสัมฤทธิ์ทางการเรียนและคุณภาพของผลงานผู้เรียนได้อย่างมีนัยสำคัญ อีกทั้งยังสอดคล้องกับการศึกษาของ Panadero and Jonsson (2013) ที่รายงานว่ารูบริกช่วยส่งเสริมการประเมินตนเองและการเรียนรู้แบบกำกับตนเองของผู้เรียน นอกจากนี้ยังพบว่ากลุ่มที่ได้รับ Rubric Score มีการกระจายตัวของคะแนนน้อยกว่ากลุ่มที่ไม่ได้รับ Rubric Score แสดงให้เห็นว่าการใช้รูบริกไม่เพียงแต่ช่วยเพิ่มคะแนนเฉลี่ยของผู้เรียน แต่ยังช่วยให้คุณภาพของผลงานมีความสม่ำเสมอมากขึ้น เนื่องจากผู้เรียนทุกคนได้รับข้อมูลเกี่ยวกับเกณฑ์การประเมินเดียวกันและสามารถใช้เป็นแนวทางในการพัฒนางานของตนเองได้อย่างเป็นระบบ ดังนั้น การใช้เกณฑ์การให้คะแนนแบบรูบริกจึงเป็นแนวทางที่มีประสิทธิภาพในการพัฒนาการจัดการเรียนการสอนรายวิชาปฏิบัติการด้านคอมพิวเตอร์ และสามารถนำไปประยุกต์ใช้กับรายวิชาปฏิบัติการอื่นในหลักสูตรเพื่อยกระดับผลสัมฤทธิ์ทางการเรียนและคุณภาพการเรียนรู้ของผู้เรียนในอนาคตได้

ข้อจำกัดการวิจัย

1. การวิจัยนี้เป็นการวิจัยในชั้นเรียนที่ดำเนินการกับนิสิตชั้นปีที่ 2 สาขาวิทยาการคอมพิวเตอร์ ที่ลงทะเบียนเรียนรายวิชา Advanced Database ในภาคการศึกษาที่ 2 จำนวน 110 คน แบ่งเป็น 2 กลุ่มเรียน กลุ่มละ 55 คน ซึ่งแม้ว่าทั้งสองกลุ่มจะได้รับการจัดการเรียนการสอน เนื้อหา เอกสารประกอบการสอน และกิจกรรมการเรียนรู้ในรูปแบบเดียวกัน แต่เนื่องจากการเป็นกลุ่มเรียนตามสภาพจริงของรายวิชา จึงไม่

สามารถควบคุมความแตกต่างด้านพื้นฐานความรู้ ความสามารถ และประสบการณ์เดิมของผู้เรียนได้ทั้งหมด อาจส่งผลต่อผลสัมฤทธิ์ทางการเรียนที่เกิดขึ้น

2. การวิจัยนี้มุ่งศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกที่มีต่อผลสัมฤทธิ์ทางการเรียนจากคะแนนใบงานปฏิบัติการเพียงอย่างเดียว โดยยังไม่ได้ศึกษาปัจจัยด้านอื่น เช่น ความพึงพอใจของผู้เรียน ความเข้าใจในเกณฑ์การประเมิน แรงจูงใจในการเรียน หรือทักษะการเรียนรู้แบบกำกับตนเอง ซึ่งอาจเป็นปัจจัยสำคัญที่ส่งผลต่อการพัฒนาผลสัมฤทธิ์ทางการเรียนเช่นกัน

ข้อเสนอแนะ

1. จากผลการวิจัยพบว่าการใช้เกณฑ์การให้คะแนนแบบรูบริกช่วยเพิ่มผลสัมฤทธิ์ทางการเรียนของนิสิตได้อย่างมีนัยสำคัญทางสถิติ ดังนั้นผู้สอนควรนำเกณฑ์การให้คะแนนแบบรูบริกมาใช้ในการออกแบบใบงานปฏิบัติการหรือการประเมินชิ้นงานในรายวิชาที่มีลักษณะการเรียนรู้เชิงปฏิบัติ เพื่อช่วยให้ผู้เรียนเข้าใจเป้าหมายการเรียนรู้และสามารถพัฒนาคุณภาพของผลงานได้ดียิ่งขึ้น

2. ควรศึกษาผลของการใช้เกณฑ์การให้คะแนนแบบรูบริกร่วมกับตัวแปรด้านอื่น เช่น ความพึงพอใจของผู้เรียน ความโปร่งใสในการประเมินผล แรงจูงใจในการเรียน หรือทักษะการประเมินตนเอง เพื่อให้เข้าใจกลไกที่ส่งผลต่อการพัฒนาผลสัมฤทธิ์ทางการเรียนได้อย่างรอบด้านมากขึ้น

เอกสารอ้างอิง

- Andrade, H. G. (2005). Teaching with rubrics: The good, the bad, and the ugly. *College Teaching*, 53(1), 27–31.
- Brookhart, S. M. (2013). *How to create and use rubrics for formative assessment and grading*. Alexandria, VA: ASCD.
- Panadero, E., & Jonsson, A. (2013). The use of scoring rubrics for formative assessment purposes revisited: A review. *Educational Research Review*, 9, 129–144.
- Reddy, Y. M., & Andrade, H. (2010). A review of rubric use in higher education. *Assessment & Evaluation in Higher Education*, 35(4), 435–448.
- Andrade, H., & Du, Y. (2005). Student perspectives on rubric-referenced assessment. *Practical Assessment, Research, and Evaluation*, 10(3), 1–11.
- Becker, K. (2016). The use of rubrics in computer science education. *Journal of Information Technology Education: Innovations in Practice*, 15, 173–184.
- Dawson, P. (2017). Assessment rubrics: Towards clearer and more replicable design, research and practice. *Assessment & Evaluation in Higher Education*, 42(3), 347–360.
- Fraile, J., Pardo, R., & Panadero, E. (2017). How does student self-assessment influence learning? A systematic review. *Assessment in Education: Principles, Policy & Practice*, 24(4), 1–24.

Jonsson, A., & Svingby, G. (2007). The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, 2(2), 130–144.

Stevens, D. D., & Levi, A. J. (2013). *Introduction to rubrics: An assessment tool to save grading time, convey effective feedback, and promote student learning* (2nd ed.). Sterling, VA: Stylus Publishing.

ภาคผนวก ก

ใบงานปฏิบัติการรายวิชา Advanced Database จำนวน 5 ใบงาน

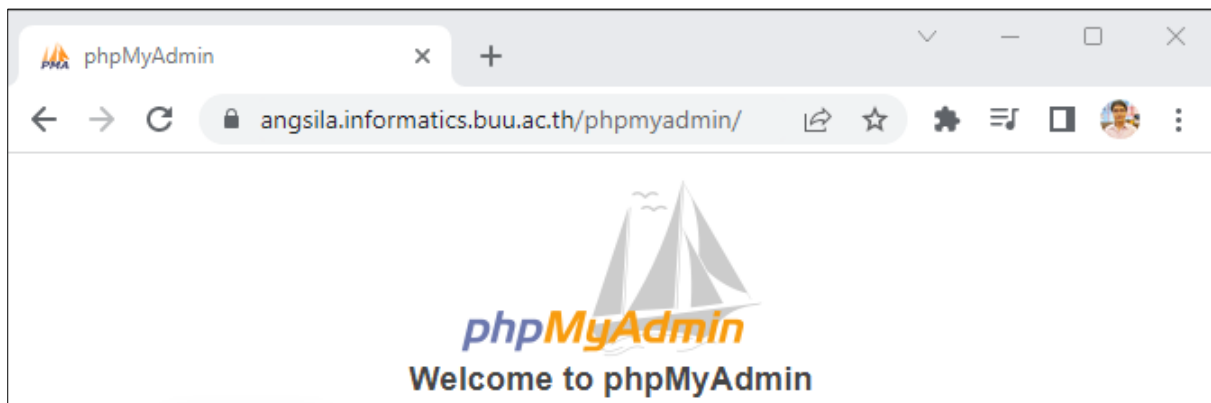
ใบงานที่ 1 การสร้าง Stored Procedure ขึ้นแนะนำ

วัตถุประสงค์

- เพื่อให้นิสิตเข้าใจงานโปรแกรม phpMyAdmin เพื่อใช้ในการจัดการฐานข้อมูล MySQL ได้
- เพื่อให้นิสิตสามารถสร้าง Stored Procedure เบื้องต้นด้วยโปรแกรม phpMyAdmin ได้
- เพื่อให้นิสิตเรียกใช้ Stored Procedure เบื้องต้นได้
- เพื่อให้นิสิตสามารถแก้ไข Stored Procedure เบื้องต้นได้

1. กำหนดให้นิสิตเข้าใช้งานฐานข้อมูล MySQL Server: angvila.informatics.buu.ac.th โดยจะการฐานข้อมูลด้วยโปรแกรม phpMyAdmin โดยมีขั้นตอนดังนี้

- เข้าไปยัง Link: <https://angvila.informatics.buu.ac.th/phpmyadmin/> ดังรูปที่ 1



รูปที่ 1 Link: <https://angvila.informatics.buu.ac.th/phpmyadmin/>

- ใช้ username และ password ที่แจกให้ ดังรูปที่ 2

รูปที่ 2 ใช้ username และ password

- คลิกเลือกฐานข้อมูลชื่อรหัสสินค้า 67160xxx พร้อมถ่ายภาพหน้าจอผลลัพธ์วางในตารางด้านล่างตัดเฉพาะ Icon รูป database โดยภาพต้องแสดงฐานข้อมูลชื่อรหัสสินค้า 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
1			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.5		

เกณฑ์การประเมิน Item 1

ระดับ	เกณฑ์การประเมิน
0.5	แสดงภาพฐานข้อมูลรหัสสินค้าถูกต้อง ชัดเจน และเป็นฐานข้อมูลของตนเอง
0.3	แสดงภาพถูกต้องแต่ตัดไม่ครบหรือชื่อฐานข้อมูลไม่ชัด
0.1	ภาพไม่ตรงตามข้อกำหนด เช่น ไม่เห็นชื่อฐานข้อมูลชัดเจน
0	ไม่ส่ง / เป็นฐานข้อมูลของผู้อื่น

2. กำหนดให้นี้สร้างตารางฐานข้อมูล “customer” ด้วยโปรแกรม phpMyAdmin โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้ พร้อมทำการรันคำสั่งโดยแสดงผลลัพธ์ลงในตารางด้านล่าง

- SQL statement 1

```
CREATE TABLE IF NOT EXISTS `customer` (
  `CUSTOMER_ID` varchar(4) NOT NULL,
  `NAME` varchar(50) NOT NULL,
  `EMAIL` varchar(50) NOT NULL,
  `COUNTRY_CODE` varchar(2) NOT NULL,
  `BUDGET` decimal(18,2) NOT NULL,
  `USED` decimal(18,2) NOT NULL,
  PRIMARY KEY (`CUSTOMER_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

- ภาพหน้าจอผลลัพธ์การสร้างสำเร็จลงในตารางด้านล่างตัดเอาเฉพาะคอลัมตาราง โดยภาพต้องแสดง ฐานข้อมูลชื่อรหัสสินค้า 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
2			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.5		

เกณฑ์การประเมิน Item 2

คะแนน	เกณฑ์
0.5	ตารางถูกต้องทุกคอลัมน์ แสดงภาพเฉพาะคอลัมน์ตามโจทย์
0.3	ตารางถูกต้องแต่ภาพไม่ชัด / ตัดภาพกว้างเกินไป
0.1	โครงสร้างตารางผิดบางส่วน
0	คำสั่งผิดทั้งหมด หรือไม่ส่ง

3. กำหนดให้ผลิตเพิ่มข้อมูลลงในตารางฐานข้อมูล “customer” ด้วยโปรแกรม phpMyAdmin โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้พร้อมทำการรันคำสั่งโดยแสดงผลพร้อมในตารางด้านล่าง

- SQL statement 2

```
INSERT INTO `customer` (`CUSTOMER_ID`, `NAME`, `EMAIL`, `COUNTRY_CODE`, `BUDGET`, `USED`) VALUES
('C001', 'Win Weerachai', 'win.weerachai@thaicreate.com', 'TH', 1000000.00, 600000.00),
('C002', 'John Smith', 'john.smith@thaicreate.com', 'UK', 2000000.00, 800000.00),
('C003', 'Jame Born', 'jame.born@thaicreate.com', 'US', 3000000.00, 600000.00),
('C004', 'Chalee Angel', 'chalee.angel@thaicreate.com', 'US', 4000000.00, 100000.00);
```

- ภาพหน้าจอผลลัพธ์การสร้างสำเร็จลงในตารางด้านล่างตัดเอาเฉพาะตารางพร้อมข้อมูลในตาราง โดยภาพต้องแสดง ฐานข้อมูลชื่อรหัสผลิตภัณฑ์ 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
3			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.5		


เกณฑ์การประเมิน Item 3

คะแนน	เกณฑ์
0.5	เพิ่มข้อมูลครบ ถูกต้อง แสดงภาพตารางพร้อมข้อมูลถูกต้อง
0.3	เพิ่มข้อมูลครบแต่ภาพไม่ชัด หรือ missing 1 row
0.1	เพิ่มข้อมูลไม่ครบ / ข้อมูลผิด
0	ไม่ส่ง

4. จงสร้าง Stored Procedure ชื่อ getCustomer เพื่อใช้แสดงข้อมูลทั้งหมดในตารางชื่อ customer โดยให้นิสิตนาคำสั่งการสร้างต่อไปนี้เป็นไปใช้สร้างด้วยโปรแกรม phpMyAdmin

- Stored Procedure 1

```
DELIMITER //
CREATE PROCEDURE getCustomer()
BEGIN
    SELECT * FROM customer;
END //
DELIMITER;
```

- ภาพหน้าจอผลลัพธ์การสร้าง Stored Procedure getCustomer ที่สำเร็จจากแท็บ  Routines ลงในตาราง โดยภาพต้องแสดงฐานข้อมูลชื่อรหัสนิสิต 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
4			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.5		

เกณฑ์การประเมิน Item 4

คะแนน	เกณฑ์
0.5	สร้าง SP ถูกต้องตามตัวอย่าง แสดงภาพแท็บ Routine
0.3	SP ทำงานได้แต่มี warning หรือโค้ด formatting ผิดเล็กน้อย
0.1	สร้าง SP แต่ไม่ทำงานจริง
0	ไม่ส่ง


5. จงสร้าง Stored Procedure ชื่อ `getCustomerName` เพื่อใช้แสดงข้อมูล 'NAME' ในตารางชื่อ customer โดยให้ชนิดนำคำสั่งการสร้างต่อไปนี้เป็นไปใช้สร้างด้วยโปรแกรม phpMyAdmin

- นำโค้ด Stored Procedure `getCustomerName` ลงในตารางด้านล่าง

Item	โค้ด Stored Procedure		
5			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	1		

เกณฑ์การประเมิน Item 5

คะแนน	เกณฑ์
1.0	โค้ดถูกต้อง 100% ตรงตามโจทย์
0.7	โค้ดถูกต้องแต่ formatting/non-critical detail ผิดเล็กน้อย
0.3	โค้ดมีข้อผิดพลาดบางส่วน
0	ไม่ส่งหรือโค้ดผิดทั้งหมด

- ภาพหน้าจอผลลัพธ์การสร้าง Stored Procedure `getCustomerName` ที่สำเร็จจากแท็บ  ลงในตารางด้านล่าง โดยภาพต้องแสดงฐานข้อมูลชื่อรหัสสินค้า 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
6			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	1		

เกณฑ์การประเมิน Item 6

คะแนน	เกณฑ์
1.0	แสดงภาพผลลัพธ์จาก Routine ถูกต้อง ชัดเจน เป็นฐานข้อมูลรหัสสินค้า
0.7	แสดงภาพถูกต้องแต่ไม่ชัด / cropping ผิดรูปแบบ
0.3	แสดงภาพแต่ไม่ใช่ SP ที่สร้าง
0	ไม่ส่ง

6. จงเรียกใช้ Stored Procedure ชื่อ getCusName โดยเติมคำสั่งที่ถูกต้องลงในตารางด้านล่าง

- คำสั่งเรียกใช้ Stored Procedure ชื่อ getCusName

Item	คำสั่ง Stored Procedure		
7			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	1		

เกณฑ์การประเมิน Item 7

คะแนน	เกณฑ์
1.0	คำสั่ง CALL ถูกต้อง แสดงรูปแบบ syntax มาตรฐาน
0.7	คำสั่งถูกต้องแต่พิมพ์ format ไม่เรียบร้อย
0.3	คำสั่งผิดบางส่วน
0	ไม่ส่ง

- ภาพหน้าจอผลลัพธ์การเรียกใช้ Stored Procedure getCusName ที่สำเร็จลงในตารางด้านล่าง โดยภาพต้องแสดง ฐานข้อมูลชื่อรหัสสินค้า 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
8			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	1		

เกณฑ์การประเมิน Item 8

คะแนน	เกณฑ์
1.0	ภาพผลลัพธ์ถูกต้อง แสดงข้อมูล NAME ถูกต้อง
0.7	ภาพถูกต้องแต่ครอปไม่ดี / ไม่เห็นชื่อฐานข้อมูลชัด
0.3	ภาพไม่ครบ / ข้อมูลไม่ถูกต้อง
0	ไม่ส่ง

7. จงแก้ไข Stored Procedure ชื่อ getCustomerName โดยกำหนดให้แสดงข้อมูลเฉพาะชื่อ และ Email โดยมีเงื่อนไขให้แสดงเฉพาะคนที่มี COUNTRY_CODE = 'US'

- โค้ด Stored Procedure getCustomerName ที่แก้ไขสำเร็จลงในตารางด้านล่าง โดยใช้คำสั่งในกล่อง Edit เท่านั้น

Item	โค้ด Stored Procedure		
9			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	2		

เกณฑ์การประเมิน Item 9

คะแนน	เกณฑ์
2.0	โค้ดถูกต้องทุกส่วน ตรงตามเงื่อนไข WHERE + เลือกเฉพาะ field
1.5	โค้ดถูกต้องแต่มี formatting หรือการจัดประกอบคำสั่งที่ไม่เป็นมาตรฐาน
1.0	โค้ดทำงานได้แต่เงื่อนไขไม่ครบ เช่น เลือกคอลัมน์ผิด
0.5	โค้ดสร้าง SP แต่รันไม่ได้
0	ไม่ส่ง

- ภาพหน้าจอผลลัพธ์การเรียกใช้ Stored Procedure getCustomerName ที่แก้ไขสำเร็จลงในตารางด้านล่าง โดยภาพต้องแสดงฐานข้อมูลชื่อรหัสสินค้า 67160xx เสมอ

Item	ภาพหน้าจอผลลัพธ์		
10			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	2		

เกณฑ์การประเมิน Item 10

คะแนน	เกณฑ์
2.0	ภาพผลลัพธ์ถูกต้อง แสดงเฉพาะ NAME + EMAIL และเฉพาะ US
1.5	ข้อมูลถูกต้องแต่ภาพครอบไม่ดี / ไม่ชัด
1.0	ข้อมูลไม่ครบ เช่น แสดงหลายประเทศ
0.5	ภาพไม่ตรงตาม SP ที่แก้ไข
0	ไม่ส่ง

ใบงานที่ 2 Stored Procedure : การสร้างและการใช้งาน Parameters

วัตถุประสงค์

- เพื่อให้คุณสามารถสร้าง Parameters ในรูปแบบต่างๆ ได้ใน Stored Procedure ได้
- เพื่อให้คุณสามารถสร้างและเรียกใช้ Stored Procedure ในรูปแบบการรับค่า Parameters ได้
- เพื่อให้คุณสามารถสร้างและเรียกใช้ Stored Procedure ในรูปแบบการส่งค่า Parameters ได้

กำหนดให้

A. กำหนดให้นักเรียนสร้างตารางฐานข้อมูล “buyer” ด้วยโปรแกรม phpMyAdmin โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้ พร้อมทำการรันคำสั่งโดยแสดงผลพร้อมทั้งในตารางด้านล่าง

- SQL statement 1

```
CREATE TABLE IF NOT EXISTS `buyer` (  
  `BUYER_ID` varchar(4) NOT NULL,  
  `NAME` varchar(50) NOT NULL,  
  `EMAIL` varchar(50) NOT NULL,  
  `COUNTRY_CODE` varchar(2) NOT NULL,  
  `BUDGET` decimal(18,2) NOT NULL,  
  `USED` decimal(18,2) NOT NULL,  
  PRIMARY KEY (`BUYER_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

B. กำหนดให้นักเรียนเพิ่มข้อมูลลงในตารางฐานข้อมูล “buyer” ด้วยโปรแกรม phpMyAdmin โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้พร้อมทำการรันคำสั่งโดยแสดงผลพร้อมทั้งในตารางด้านล่าง

- SQL statement 2

```
INSERT INTO `buyer` (`BUYER_ID`, `NAME`, `EMAIL`, `COUNTRY_CODE`, `BUDGET`, `USED`) VALUES  
(  
'A001', 'Win Kai', 'win.weerachai@buu.com', 'TH', 8000000.00, 500000.00),  
(  
'A002', 'John Zion', 'john.smith@ buu.com', 'TH', 7000000.00, 800000.00),  
(  
'A003', 'Jame Luca', 'jame.born@ buu.com', 'US', 6000000.00, 600000.00),  
(  
'A004', 'Chalee Maeve', 'chalee.angel@ buu.com', 'US', 5000000.00, 100000.00),  
(  
'A005', 'Fannie Rothfuss', 'Fannie.Rothfuss@zoozoo.nl', 'GQ', 1000000.00, 500000.00),  
(  
'A006', 'Patti Barrowman', 'Patti.Barrowman@yale.co.uk', 'GQ', 2000000.00, 100000.00),  
(  
'A007', 'Marybeth Runyon', 'Marybeth.Runyon@goobal.nl', 'GQ', 3000000.00, 200000.00),  
(  
'A008', 'Cyndi Budrene', 'Cyndi.Budrene@xeexee.de', 'KR', 4000000.00, 300000.00),  
(  
'A009', 'Letitia Pantilla', 'Letitia.Pantilla@xeebal.com', 'GS', 2000000.00, 200000.00),  
(  
'A010', 'Tamma Wiles', 'Tamma.Wiles@iaaxee.edu', 'AI', 1000000.00, 300000.00);
```

C. ตารางอธิบายข้อมูลผู้ซื้อ (Buyer) โดยข้อมูลที่เก็บจะเป็นข้อมูลการใช้จ่ายบัตรเครดิตรายบุคคลโดยมีรายละเอียดดังนี้

Data Dictionary

Data Item	Data Type	Description
BUYER_ID	varchar(4)	รหัสผู้ซื้อ
NAME	varchar(50)	ชื่อผู้ซื้อ
EMAIL	varchar(50)	อีเมลผู้ซื้อ
COUNTRY_CODE	varchar(2)	รหัสประเทศที่ผู้ซื้ออาศัยอยู่
BUDGET	decimal(18,2)	วงเงินบัตร (บาท)
USED	decimal(18,2)	ยอดเงินที่ใช้ไป (บาท)

1. จงสร้าง (Stored Procedure: SP) โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getBuyer01
 2. หากมีการสร้าง getBuyer01 ก่อนนี้ให้ลบออกได้
 3. กำหนดให้ SP แสดงข้อมูลทุกคอลัมน์ในตาราง buyer
 4. กำหนดให้ SP แสดงข้อมูลตามรหัสประเทศที่ผู้ซื้ออาศัยอยู่โดยรับข้อมูลจากผู้ใช้
- จงระบุภาพหน้าจอโค้ด SP getBuyer01 ลงในตาราง Item 1 (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอโค้ด		
1			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.7		

ระดับ	เกณฑ์
0.7	โค้ดถูกต้องครบถ้วน ใช้ parameter ถูกต้อง และแสดงภาพฐานข้อมูลรหัสสินค้า
0.5	โค้ดถูกต้องแต่ format ไม่เรียบร้อย / มี warning
0.3	เงื่อนไขไม่ครบ / parameter ผิด
0	ไม่ส่งหรือผิดทั้งหมด

- คำสั่งเรียกใช้ SP getBuyer01 กำหนดให้แสดงเฉพาะข้อมูลรหัสประเทศที่ผู้ซื้ออาศัยอยู่คือ 'TH' เท่านั้น (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอคำสั่ง		
2			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.6		

ระดับ	เกณฑ์
0.6	CALL SP ถูกต้อง ใช้ 'TH' ตามเงื่อนไข
0.4	CALL ถูกต้องแต่ syntax สะกดผิดเล็กน้อย
0.2	ส่งคำสั่งแต่ไม่ทำงาน
0	ไม่ส่ง

- จงระบุภาพหน้าจอผลลัพธ์การเรียก SP getBuyer01 ใช้จาก Item 2 (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอผลลัพธ์		
3			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	0.7		

ระดับ	เกณฑ์
0.7	แสดงผลเฉพาะ TH ถูกต้อง และชัดเจน
0.5	ผลถูกต้องแต่ภาพครอบไม่ดี
0.3	ผลลัพธ์ไม่ครบ/คลาดเคลื่อน
0	ไม่ส่ง

2. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getBuyer02
2. หากมีการสร้าง getBuyer02 ก่อนนี้ให้ลบออกได้
3. กำหนดให้ SP แสดงข้อมูล รหัสผู้ซื้อ ชื่อผู้ซื้อ รหัสประเทศที่ผู้ซื้ออาศัยอยู่ และยอดเงินที่ใช้ไป
4. กำหนดให้แสดงเฉพาะข้อมูลตามรหัสประเทศที่ผู้ซื้ออาศัยอยู่โดยรับข้อมูลจากผู้ใช้
5. กำหนดให้แสดงเฉพาะข้อมูลยอดเงินที่ใช้ไปที่มีค่ามากกว่าหรือเท่ากับยอดเงินที่ใช้ไปที่รับข้อมูลจากผู้ใช้

- จงระบุภาพหน้าจอโค้ด SP getBuyer02 (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอโค้ด		
4			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	1		

ระดับ	เกณฑ์
1.0	โค้ดถูกต้องทุกเงื่อนไข (ประเทศ + ยอด \geq input)
0.7	โค้ดถูกต้องแต่ format ผิด
0.4	ขาดเงื่อนไขหนึ่งเงื่อนไข
0	ไม่ส่ง

- จงระบุภาพหน้าจอคำสั่งเรียกใช้ SP getBuyer02 โดยกำหนดเงื่อนไขการแสดงผลเฉพาะข้อมูลที่มีรหัสประเทศที่ผู้ซื้ออาศัยอยู่คือ 'TH' และ ยอดเงินที่ใช้ไปคือ '300000' (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอคำสั่ง		
5			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	1		

ระดับ	เกณฑ์
1.0	CALL SP ถูกต้องตาม 2 parameter
0.7	CALL ถูกต้องแต่พิมพ์สะกดไม่เรียบร้อย
0.4	CALL ผิดรูปแบบ
0	ไม่ส่ง

- จงระบุภาพหน้าจอผลลัพธ์การเรียกใช้ SP getBuyer02 จาก **Item 5** (ในภาพต้องเห็น Database รหัสนิติ)

Item	ภาพหน้าจอผลลัพธ์		
6			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	1		

ระดับ	เกณฑ์
1.0	แสดงผลถูกต้องตาม TH + ≥ 300000
0.7	ข้อมูลถูกต้องแต่ภาพตัดไม่ดี
0.4	ข้อมูลบางส่วนผิด
0	ไม่ส่ง

3. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP Count03 และ getBuyer03 เพื่อนับจำนวนข้อมูลผู้ซื้อที่ตรงตามเงื่อนไขที่ค้นหา
2. หากมีการสร้าง Count03 ก่อนนี้ให้ลบออกได้
3. Count03 รับค่าข้อมูลรหัสประเทศที่ผู้ซื้ออาศัยอยู่และยอดเงินที่ใช้ไปจากผู้ใช้
4. Count03 จะแสดงข้อมูลผู้ซื้อที่มีรหัสประเทศที่ผู้ซื้ออาศัยอยู่ตามที่กำหนด
5. Count03 จะแสดงข้อมูลผู้ซื้อที่มียอดเงินที่ใช้จ่ายที่มีค่ามากกว่าหรือเท่ากับยอดเงินที่ใช้ไปที่ได้รับข้อมูลจากผู้ใช้
6. Count03 ส่งค่าจำนวนข้อมูลผู้ซื้อที่ตรงตามเงื่อนไขที่ค้นหากลับ
7. getBuyer03 โดยเรียกใช้งาน Count03 ในการนับจำนวนข้อมูลผู้ซื้อที่ตรงตามเงื่อนไขที่ค้นหา
8. หากมีการสร้าง getBuyer03 ก่อนนี้ให้ลบออกได้
9. getBuyer03 รับค่าข้อมูลรหัสประเทศที่ผู้ซื้ออาศัยอยู่จากผู้ใช้
10. getBuyer03 รับค่าข้อมูลยอดเงินที่ใช้ไปจากผู้ใช้
11. กำหนดให้ getBuyer03 แสดงผลลัพธ์การนับจำนวนข้อมูลผู้ซื้อที่ตรงตามเงื่อนไขที่ค้นหาจาก Count03 ออกหน้าจอตัง
ตัวอย่างรูปนี้

Buyer Row Count
Buyer Count = 3

- จงระบุภาพหน้าจอโค้ด Count03 (ในภาพต้องเห็น Database รหัสนิสิต)

Item	ภาพหน้าจอโค้ด		
7			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	1		

ระดับ	เกณฑ์
1.0	ใช้ OUT parameter ถูกต้อง และนับจำนวนได้
0.7	โค้ดทำงานแต่ format หรือ variable ไม่เรียบร้อย
0.4	OUT parameter ผิด / count ผิด
0	ไม่ส่ง

- จงระบุภาพหน้าจอโค้ด getBuyer03 (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอโค้ด		
8			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	1		

ระดับ	เกณฑ์
1.0	เรียกใช้ getBuyerCount03 ถูกต้อง
0.7	เรียกใช้ถูกต้องแต่ print/display ผลลัพธ์ผิดเล็กน้อย
0.4	ไม่สามารถแสดงผลตาม OUT
0	ไม่ส่ง

- จงระบุภาพหน้าจอคำสั่งเรียกใช้ SP getBuyer03 กรณีต้องการแสดงเฉพาะจำนวนข้อมูลที่มีรหัสประเทศที่ผู้ซื้ออาศัยอยู่คือ 'TH' และยอดเงินที่ใช้ไปคือ '500000' (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอคำสั่ง		
9			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	1		

ระดับ	เกณฑ์
1.0	CALL SP ถูกต้อง ใช้ 2 parameter
0.7	CALL ถูกต้องแต่ format ไม่เรียบร้อย
0.4	คำสั่งผิด
0	ไม่ส่ง

- จงระบุภาพหน้าจอผลลัพธ์การสร้างสำเร็จลงในตารางด้านล่าง (ในภาพต้องเห็น Database รหัสสินค้า)

Item	ภาพหน้าจอผลลัพธ์		
10			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	1		

ระดับ	เกณฑ์
1.0	แสดงจำนวน (Buyer Count) ถูกต้อง
0.7	ข้อมูลถูกต้องแต่ภาพไม่ชัด
0.4	ข้อมูลไม่ครบ
0	ไม่ส่ง

4. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getBuyerCount04 เพื่อนับจำนวนข้อมูลผู้ซื้อที่ตรงตามเงื่อนไขที่ค้นหา
2. หากมีการสร้าง getBuyerCount04 ก่อนนี้ให้ลบออกได้
3. getBuyerCount04 รับค่าข้อมูลรหัสประเทศจากผู้ใช้
4. getBuyerCount04 รับค่าข้อมูลยอดเงินที่ใช้จ่ายจากผู้ซื้อ
5. getBuyerCount04 จะแสดงข้อมูลผู้ซื้อที่มีรหัสประเทศที่ผู้ซื้ออาศัยอยู่ตามที่ผู้ซื้อกำหนด
6. getBuyerCount04 จะแสดงข้อมูลผู้ซื้อที่มียอดเงินที่ใช้จ่ายมากกว่าหรือเท่ากับยอดเงินที่ใช้จ่ายที่ผู้ซื้อกำหนด
7. getBuyerCount04 ต้องใช้ฟังก์ชัน FOUND_ROWS() ในการนับจำนวนข้อมูลผู้ซื้อตามเงื่อนไข
8. getBuyerCount04 แสดงผลลัพธ์การนับจำนวนข้อมูลผู้ซื้อที่ตรงตามเงื่อนไขที่ค้นหาจากหน้าจอจัดตัวอย่างรูปนี้

Buyer Found Row
Buyer Count = 3

- จงระบุภาพหน้าจอโค้ด SP getBuyerCount04 (ในภาพต้องเห็น Database รหัสนิสิต)

Item	ภาพหน้าจอโค้ด		
11			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	0.4		

ระดับ	เกณฑ์
0.4	ใช้ FOUND_ROWS() ถูกต้อง
0.3	โค้ดถูกต้องแต่ format ผิด
0.1	ใช้ FOUND_ROWS() ไม่ถูกต้อง
0	ไม่ส่ง

- จงระบุหน้าจอคำสั่งเรียกใช้ SP getBuyerCount04 กรณีต้องการแสดงเฉพาะจำนวนข้อมูลที่มีรหัสประเทศที่ผู้ซื้ออาศัยอยู่ คือ 'TH' และยอดเงินที่ใช้ไปคือ '500000' (ในภาพต้องเห็น Database รหัสนิติ)

Item	ภาพหน้าจอคำสั่ง		
12			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	0.3		

ระดับ	เกณฑ์
0.3	CALL ถูกต้อง parameter ครบ
0.2	CALL ถูกต้องแต่ format ผิดเล็กน้อย
0.1	CALL ผิด
0	ไม่ส่ง

- จงระบุภาพหน้าจอผลลัพธ์การเรียกใช้ SP getBuyerCount04 (ในภาพต้องเห็น Database รหัสนิติ)

Item	ภาพหน้าจอผลลัพธ์		
13			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	0.3		

ระดับ	เกณฑ์
0.3	แสดงจำนวนข้อมูลถูกต้อง
0.2	ข้อมูลถูกต้องแต่ภาพผิดรูปแบบ
0.1	ข้อมูลไม่ครบ
0	ไม่ส่ง

ใบงานที่ 3 Declare Variable & IF...ELSE...

วัตถุประสงค์

- เพื่อให้นิสิตเข้าใจการประกาศตัวแปรพร้อมเรียกใช้ตัวแปรบน (Stored Procedure: SP)
- เพื่อให้นิสิตเข้าใจการประกาศตัวแปรตารางพร้อมเรียกใช้
- เพื่อให้นิสิตเข้าใจและเรียกใช้โครงสร้างแบบกำหนดเงื่อนไข IF แบบต่าง ๆ

กำหนดให้

A. กำหนดให้นิสิตสร้างตาราง “client” เพิ่มลงในฐานข้อมูล S66160xxx โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้พร้อมทำการรันคำสั่งโดยแสดงผลพร้อมลงในตารางด้านล่าง

- SQL statement 1

```
CREATE TABLE IF NOT EXISTS `client` (
  `CLIENT_ID` varchar(4) NOT NULL,
  `NAME` varchar(50) NOT NULL,
  `EMAIL` varchar(50) NOT NULL,
  `COUNTRY_CODE` varchar(2) NOT NULL,
  `BUDGET` decimal(18,2) NOT NULL,
  `USED` decimal(18,2) NOT NULL,
  PRIMARY KEY (`CLIENT_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

B. กำหนดให้นิสิตเพิ่มข้อมูลลงในตาราง “client” โดยกำหนดให้คัดลอก SQL statement ต่อไปนี้พร้อมทำการรันคำสั่งโดยแสดงผลพร้อมลงในตารางด้านล่าง

- SQL statement 2

```
INSERT INTO `client` (`CLIENT_ID`, `NAME`, `EMAIL`, `COUNTRY_CODE`, `BUDGET`, `USED`) VALUES
('T001', 'Win Kai', 'win.weerachai@buu.com', 'TH', 8000000.00, 600000.00),
('T002', 'John Zion', 'john.smith@buu.com', 'TH', 7000000.00, 800000.00),
('T003', 'Jame Luca', 'jame.born@buu.com', 'TH', 6000000.00, 600000.00),
('T004', 'Chalee Maeve', 'chalee.angel@buu.com', 'US', 5000000.00, 100000.00),
('T005', 'Mogan Lee', 'mogan.lee@buu.com', 'UK', 5000000.00, 100000.00),
('T006', 'Monaleza Go', 'monaleza.go@buu.com', 'UK', 1000000.00, 500000.00),
('T007', 'Ann Toto', 'ann.toto@buu.com', 'US', 5000000.00, 100000.00),
('T008', 'Cyndi Budrene', 'Cyndi.Budrene@xeexee.de', 'UK', 4000000.00, 300000.00),
('T009', 'Letitia Pantilla', 'Letitia.Pantilla@xeebal.com', 'US', 2000000.00, 200000.00),
('T010', 'Tamma Wiles', 'Tamma.Wiles@iaaxee.edu', 'US', 1000000.00, 300000.00);
```

C. กำหนดให้ผลิตสร้างตาราง “country” เพิ่มลงในฐานข้อมูล S67160xxx ดังตารางที่ 1 พร้อมคัดลอกคำสั่ง SQL statement ลงในช่องว่าง

ตารางที่ 1

COUNTRY_CODE	COUNTRY_NAME
TH	Thailand
UK	United Kingdom
US	United States

D. ตารางอธิบายข้อมูลลูกค้าของบริษัทขายตรง The TOTO โดยข้อมูลที่เก็บจะเป็นข้อมูลการใช้จ่ายบัตรเครดิตรายบุคคลของลูกค้าโดยมีรายละเอียดดังนี้

Data Dictionary

client table

Data Item	Data Type	Description
CLIENT_ID	varchar(4)	รหัสลูกค้า
NAME	varchar(50)	ชื่อลูกค้า
EMAIL	varchar(50)	อีเมลล์ลูกค้า
COUNTRY_CODE	varchar(2)	รหัสประเทศ
BUDGET	decimal(18,2)	วงเงินบัตร (บาท)
USED	decimal(18,2)	ยอดเงินที่ใช้ไป (บาท)

country table

COUNTRY_CODE	varchar(2)	รหัสประเทศ
COUNTRY_NAME	varchar(50)	ชื่อประเทศ

1. จงระบุภาพหน้าจอคำสั่งสร้างตาราง country ลงในตาราง Item 1 (ในภาพต้องเห็นก้อน Database รหัสนิสิต)

Item [0.5]	ภาพหน้าจอโค้ดสร้างตารางข้อมูล country		
1			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1.1	0.5		

2. จงระบุภาพหน้าจอคำสั่งเพิ่มข้อมูลลงตาราง country ในตาราง Item 2 (ในภาพต้องเห็นก้อน Database รหัสนิสิต)

COUNTRY_CODE	COUNTRY_NAME
TH	Thailand
UK	United Kingdom
US	United States

Item [0.5]	ภาพหน้าจอโค้ดเพิ่มข้อมูล		
2			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1.2	0.5		

ระดับ	เกณฑ์
1.0	โค้ด CREATE + INSERT ถูกต้อง แสดงภาพฐานข้อมูลชัดเจน
0.7	โค้ดถูกต้องแต่จัด format ไม่ดี / กรอบภาพไม่ครบ
0.4	โค้ดผิดบางส่วน หรือ missing columns
0	ไม่ส่ง / ไม่ถูกต้อง

3. จงสร้าง SP โดยมีเงื่อนไขดังนี้

- จงสร้าง SP myDiscount03
- จงเขียน SP ที่รับค่าจำนวนเงินมา 1 จำนวนจากผู้ใช้ เพื่อคิดส่วนลดโดยมีเงื่อนไข
 - หากเงินที่รับมามีค่ามากกว่า 300 บาทจะได้รับส่วนลด 30%
 - ส่งผลลัพธ์ส่วนลดที่ได้กลับออกจาก SP
- จงระบุภาพหน้าจอผลลัพธ์ลงในตารางด้านล่าง (ในภาพต้องเห็นก้อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
ภาพหน้าจอ Code			
ภาพหน้าจอคำสั่ง เรียกใช้เมื่อรับเงิน : 330 บาท			
ภาพหน้าจอผลลัพธ์			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	1.5		

คะแนน	เกณฑ์
1.5	SP ถูกต้อง คำนวณส่วนลด 30% เงื่อนไข IF >300 ถูกต้อง แสดงผลลัพธ์ถูกต้อง
1.0	SP ถูกต้องแต่ logic มี error เล็กน้อย เช่น ชื่อ variable ไม่ตรง
0.5	เงื่อนไขผิดหรือไม่คำนวณส่วนลด
0	ไม่ส่ง

4. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP `getPerson03`
2. จงสร้างตาราง temp เพื่อเก็บข้อมูลชื่อบุคคล และชื่อประเทศ
3. ตาราง temp ประกอบด้วยคอลัมน์ RowID, Name ,CountryName และ CountryNumber
4. RowID เป็น PRIMARY KEY ชนิดข้อมูล INT UNSIGNED โดยตัวเลขจะเพิ่มค่าขึ้นแบบอัตโนมัติ
5. จงเพิ่มข้อมูลนี้ลงตาราง temp

Name	CountryName	CountryNumber
Weerachai Nukitram	Thailand	34000
Surachai Sirisart	United States	35000
Adisorn Boonsong	United Kingdom	36000

6. แสดงผลข้อมูลทั้งหมดของตาราง temp ออกหน้าจอ
7. ลบตาราง temp

- จงระบุภาพหน้าจอผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสனிสิต)

Condition	ภาพหน้าจอผลลัพธ์		
ภาพหน้าจอ Code			
ภาพหน้าจอคำสั่งเรียกใช้			
ภาพหน้าจอผลลัพธ์			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	2		

คะแนน	เกณฑ์
2.0	สร้าง temp table, insert 3 rows, select แสดงผลครบ & drop table ถูกต้อง
1.4	ทำครบแต่ temp table ไม่ drop หรือข้อมูลไม่ครบ 3 แถว
0.8	สร้าง temp table แต่ข้อมูลผิดหลายจุด
0	ไม่ส่ง

5. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getTH03
2. จงเขียน SP รับค่าจำนวนเต็ม 1 จำนวน โดยมีเงื่อนไขหากตัวเลขที่รับคือ "1" ให้แสดงข้อมูลจากตาราง client และ country โดยแสดงข้อมูลรหัสลูกค้า ชื่อลูกค้า อีเมลลูกค้า และชื่อประเทศ
3. กำหนดให้แสดงข้อมูลที่มีรหัสประเทศคือ TH เท่านั้น
4. กำหนดให้เรียงลำดับรหัสลูกค้าจาก**มากไปน้อย**
5. กำหนดการแสดงผลจำนวน 2 ชุดข้อมูล

- จงระบุภาพหน้าจอผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
ภาพหน้าจอ Code			
ภาพหน้าจอ คำสั่งเรียกใช้ เมื่อกำหนด หมายเลข 1			
ภาพหน้าจอ ผลลัพธ์			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
4	2		

คะแนน	เกณฑ์
2.0	ใช้ IF condition ถูกต้อง (1 = TH), join client+country, order desc, limit 2
1.4	เงื่อนไขถูก แต่ลำดับผลลัพธ์ผิด / แสดงมากกว่า 2 แถว
0.8	join ผิด หรือ filter TH ผิด
0	ไม่ส่ง

6. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP `getUK03`
2. จงเขียน SP รับค่าจำนวนเต็ม 1 จำนวน โดยมีเงื่อนไขตัวเลขที่รับคือ
 - หมายเลข "1" ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลลูกค้า และชื่อประเทศ จากตาราง client และ country โดยกำหนดให้แสดงข้อมูลรหัสประเทศคือ TH
 - หากเป็นตัวเลขใด ๆ ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลลูกค้า และชื่อประเทศ จากตาราง client และ country โดยกำหนดให้แสดงข้อมูลรหัสประเทศคือ 'UK'
3. กำหนดให้เรียงลำดับข้อมูลรหัสลูกค้าจาก**มากไปน้อย**
4. กำหนดการแสดงผลจำนวน 2 ชุด
 - จงระบุภาพหน้าจอผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
ภาพหน้าจอ Code			
ภาพหน้าจอ คำสั่งเรียกใช้ เมื่อกำหนด หมายเลข 8			
ภาพหน้าจอ ผลลัพธ์			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
5	2		

คะแนน	เกณฑ์
2.0	IF 1=TH, ELSE=UK, join ถูกต้อง, limit 2, order desc ถูกต้อง
1.4	ลำดับผลลัพธ์ผิด / ข้อมูลไม่เรียง
0.8	เงื่อนไข IF ผิด
0	ไม่ส่ง

7. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getUS03
2. จงเขียน SP เพื่อจัดการข้อมูลในตาราง client และ country เมื่อรับค่าจำนวนเต็ม 1 จำนวน โดยมีเงื่อนไขตัวเลขที่รับคือ
 - หมายเลข "1" ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลล์ลูกค้า และชื่อประเทศจาก client และ country โดยกำหนดให้แสดงข้อมูลรหัสประเทศคือ TH
 - หมายเลข "2" ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลล์ลูกค้า และชื่อประเทศจกตาราง client และ country โดยกำหนดให้แสดงข้อมูลรหัสประเทศคือ US
 - หากเป็นตัวเลขใด ๆ ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลล์ลูกค้า และชื่อประเทศจกตาราง client และ country โดยกำหนดให้แสดงข้อมูลรหัสประเทศคือ UK
3. กำหนดให้เรียงลำดับข้อมูลตามรหัสลูกค้า
4. กำหนดการแสดงผลจำนวน 2 ชุด
 - จงระบุภาพหน้าจอผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
ภาพหน้าจอ Code			
ภาพหน้าจอคำสั่งเรียกใช้ เมื่อกำหนดหมายเลข 2			
ภาพหน้าจอผลลัพธ์			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
6	2		

คะแนน	เกณฑ์
1.5	IF 1=TH, 2=US, ELSE=UK, join client+country ครบ, order และ limit ถูกต้อง
1.0	เงื่อนไข IF ถูกแต่การจัดผลลัพธ์ผิด
0.5	Logic ผิดหลายจุด
0	ไม่ส่ง

ใบงานที่ 4 LOOP and CASE WHEN

วัตถุประสงค์

- เพื่อให้นิสิตเข้าใจ และประยุกต์ใช้คำสั่ง LOOP
- เพื่อให้นิสิตเข้าใจ และประยุกต์ใช้คำสั่ง CASE WHEN

1. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP getDataLoopLab01
2. กำหนดให้ใช้คำสั่ง WHILE LOOP เท่านั้น
3. กำหนดให้แสดงตัวเลขพร้อมสัญลักษณ์ “=>” โดยกำหนดให้เริ่มต้นที่ 0=> และลงท้ายด้วย =>10
4. กำหนดข้อความหัวตารางคือ “Result_Lab01” โดยแสดงดังรูปผลลัพธ์

Result_Lab01

0=>1=>2=>3=>4=>5=>6=>7=>8=>9=>10

จงเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสนิสิต)

Condition	ภาพหน้าจอผลลัพธ์		
1.1 ภาพหน้าจอCode [0.8 คะแนน]			
1.2 ภาพหน้าจอคำสั่งเรียกใช้ [0.6 คะแนน]			
1.3 ภาพหน้าจอผลลัพธ์ [0.6 คะแนน]			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	2		

ระดับ	เกณฑ์
เต็ม	LOOP ทำงานครบทุกค่า แสดงผลถูกต้อง
ผ่าน	LOOP ถูกต้องแต่ format / label ผิด
พอใช้	LOOP ทำงานไม่ครบ
0	ไม่ส่ง / ผิดทั้งหมด

ตารางที่ใช้ประกอบข้อที่ 2-3

1. customer

CUSTOMER_ID	NAME	EMAIL	COUNTRY_CODE	BUDGET	USED
C001	Win Weerachai	win.weerachai@thaicreate.com	TH	1000000.00	600000.00
C002	John Smith	john.smith@thaicreate.com	UK	2000000.00	800000.00
C003	Jame Born	jame.born@thaicreate.com	US	3000000.00	600000.00
C004	Chalee Angel	chalee.angel@thaicreate.com	US	4000000.00	100000.00

2. country

COUNTRY_CODE	COUNTRY_NAME
TH	Thailand
UK	United Kingdom
US	United States

3. ตารางอธิบายข้อมูลลูกค้าบริษัทขายตรง The Basic โดยข้อมูลที่เก็บจะเป็นข้อมูลการใช้จ่ายบัตรเครดิตรายบุคคลของลูกค้า โดยมีรายละเอียดดังนี้

Data Dictionary

client table

Data Item	Data Type	Description
CUSTOMER_ID	varchar(4)	รหัสลูกค้า
NAME	varchar(50)	ชื่อลูกค้า
EMAIL	varchar(50)	อีเมลลูกค้า
COUNTRY_CODE	varchar(2)	รหัสประเทศ
BUDGET	decimal(18,2)	วงเงินบัตร (บาท)
USED	decimal(18,2)	ยอดเงินที่ใช้ไป (บาท)

country table

COUNTRY_CODE	varchar(2)	รหัสประเทศ
COUNTRY_NAME	varchar(50)	ชื่อประเทศ

2. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP LoopCustomerLab2
2. จงแสดงข้อมูลชื่อลูกค้า และรหัสประเทศ จากตาราง **customer** พร้อมเก็บข้อมูลลงในตาราง temp
3. จงเพิ่มคอลัมน์ RowID เป็น PRIMARY KEY ซึ่งจะเพิ่มจำนวนแบบอัตโนมัติลงในตาราง temp
4. จงเพิ่มคอลัมน์ ItemDetail เก็บข้อมูลข้อความลงในตาราง temp
5. จงใช้ **WHILE LOOP** เก็บข้อมูลข้อความ "Item No : 1" ลงในคอลัมน์ ItemDetail โดยหมายเลขในข้อความจะเริ่มจาก 1

จนถึงจำนวนชุดข้อมูลที่ได้จากการ Query ในข้อ 2

6. จงแสดงข้อมูลในตาราง temp
7. จงลบตาราง temp
8. ตัวอย่างการจัดเรียงคอลัมน์

RowID	NAME	COUNTRY_CODE	ItemDetail
-------	------	--------------	------------

จงเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
2.1 ภาพหน้าจอ Code [1.2 คะแนน]			
2.2 ภาพหน้าจอคำสั่งเรียกใช้ [0.8 คะแนน]			
2.3 ภาพหน้าจอผลลัพธ์ [1.0 คะแนน]			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
2	3		

ระดับ	เกณฑ์
เต็ม	Temp Table ถูกต้อง LOOP ทำงานครบ
ผ่าน	LOOP ทำงานแต่ ItemDetail ผิดเล็กน้อย
พอใช้	Temp Table หรือ LOOP ผิด
0	ไม่ส่ง

3. จงสร้าง SP โดยมีเงื่อนไขดังนี้

1. จงสร้าง SP CaseCustomerLab3 โดยใช้ **CASE WHEN**
2. จงเขียน SP **รับค่าจำนวนเต็ม 1 จำนวน** โดยมีเงื่อนไขตัวเลขที่รับดังข้อที่ 3, 4 และ 5
3. หากเป็นหมายเลข **1** ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลลูกค้า และ ชื่อประเทศ จากตาราง customer และ country โดยกำหนดข้อมูลรหัสประเทศคือ TH
 - 3.1 จงสร้างตาราง temp โดยมีคอลัมน์ในตารางคือ CUSTOMER_ID, NAME, EMAIL, COUNTRY_NAME
 - 3.2 จงเพิ่มคอลัมน์ RowID เป็น PRIMARY KEY ลงในตาราง temp
 - 3.3 คอลัมน์ RowID จะเพิ่มจำนวนแบบอัตโนมัติได้แก่ 1 2 ... N
 - 3.4 จงเพิ่มคอลัมน์ ItemDetail เก็บข้อมูลข้อความลงในตาราง temp
 - 3.5 จงใช้ REPEAT Loop เก็บข้อมูลข้อความ "Stock No: 1" ลงในคอลัมน์ ItemDetail โดยหมายเลขในข้อความจะเริ่มจาก 1 จนถึงจำนวนข้อมูลที่แสดงผลจากการ Query ในข้อที่ 3
 - 3.6 แสดงข้อมูลในตาราง temp
 - 3.7 ลบตาราง temp
4. หากเป็นหมายเลข **2** ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลลูกค้า และ ชื่อประเทศ จากตาราง customer และ country โดยกำหนดข้อมูลรหัสประเทศคือ US
 - 4.1 จงสร้างตาราง temp โดยมีคอลัมน์ในตารางคือ CUSTOMER_ID, NAME, EMAIL, COUNTRY_NAME
 - 4.2 จงเพิ่มคอลัมน์ RowID เป็น PRIMARY KEY ลงในตาราง temp
 - 4.3 คอลัมน์ RowID จะเพิ่มจำนวนแบบอัตโนมัติได้แก่ 1 2 ... N
 - 4.4 จงเพิ่มคอลัมน์ ItemDetail เก็บข้อมูลข้อความลงในตาราง temp
 - 4.5 จงใช้ REPEAT Loop เก็บข้อมูลข้อความ "Stock No: 1" ลงในคอลัมน์ ItemDetail โดยหมายเลขในข้อความจะเริ่มจาก 1 จนถึงจำนวนข้อมูลที่แสดงผลจากการ Query ในข้อที่ 4
 - 4.6 แสดงข้อมูลในตาราง temp
 - 4.7 ลบตาราง temp
5. หากเป็นตัวเลขใด ๆ นอกจากหมายเลข **1** หรือ **2** ให้แสดงข้อมูล รหัสลูกค้า ชื่อลูกค้า อีเมลลูกค้า และ ชื่อประเทศ จากตาราง customer และ country โดยกำหนดข้อมูลรหัสประเทศคือ UK
 - 5.1 จงสร้างตาราง temp กำหนดคอลัมน์ในตาราง CUSTOMER_ID, NAME, EMAIL, COUNTRY_NAME
 - 5.2 จงเพิ่มคอลัมน์ RowID เป็น PRIMARY KEY ซึ่งจะเพิ่มลงในตาราง temp
 - 5.3 คอลัมน์ RowID จะเพิ่มจำนวนแบบอัตโนมัติได้แก่ 1 2 ... N
 - 5.4 จงเพิ่มคอลัมน์ ItemDetail เก็บข้อมูลข้อความลงในตาราง temp
 - 5.5 จงใช้ REPEAT Loop เก็บข้อมูลข้อความ "Stock No: 1" ลงในคอลัมน์ ItemDetail โดยหมายเลขในข้อความจะเริ่มจาก 1 จนถึงจำนวนข้อมูลที่แสดงผลจากการ Query ในข้อที่ 5
 - 5.6 แสดงข้อมูลในตาราง temp
 - 5.7 ลบตาราง temp
6. ตัวอย่างการจัดเรียงคอลัมน์

RowID	CUSTOMER_ID	NAME	EMAIL	COUNTRY_NAME	ItemDetail
-------	-------------	------	-------	--------------	------------

จงเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
3.1 Code [1.5 คะแนน]			
3.2 ภาพ หน้าจอคำสั่ง เรียกใช้ รับค่าเข้า = 1 [1.2 คะแนน]			
3.3 ภาพ หน้าจอผลลัพธ์ รับค่าเข้า = 1 [1.2 คะแนน]			
3.4 ภาพ หน้าจอคำสั่ง เรียกใช้รับค่า เข้า = 8 [1.1 คะแนน]			
3.5 ภาพ หน้าจอผลลัพธ์ รับค่าเข้า = 8 [1.0 คะแนน]			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
3	5		

ระดับ	เกณฑ์
เต็ม	CASE WHEN ถูกต้องทุกกรณี LOOP ทำงานครบ
ผ่าน	ทำงานครบแต่เรียงลำดับ/ItemDetail ผิด
พอใช้	ทำงานได้เพียงบางกรณี
0	ไม่ส่ง

ใบงานที่ 5 View Table on Database

วัตถุประสงค์

- เพื่อให้ห็นสิดเข้าใจ และประยุกต์ใช้ View Table

ตารางข้อมูลที่ใช้ประกอบ

1. ตาราง customer05

CUSTOMER_ID	NAME	EMAIL	COUNTRY_CODE	BUDGET	USED
C001	Win Weerachai	win.weerachai@thaicreate.com	TH	1000000.00	600000.00
C002	John Smith	john.smith@thaicreate.com	UK	2000000.00	800000.00
C003	Jame Born	jame.born@thaicreate.com	US	3000000.00	600000.00
C004	Chalee Angel	chalee.angel@thaicreate.com	US	4000000.00	100000.00
C005	Somnuk Weerachai	somnuk.weerachai@thaicreate.com	TH	1000000.00	600000.00
C006	Adison Smith	adison.smith@thaicreate.com	UK	2000000.00	800000.00
C007	Shusuk Born	shusuk.born@thaicreate.com	US	3000000.00	600000.00
C008	Lee Angel	lee.angel@thaicreate.com	US	4000000.00	100000.00

2. ตาราง country

COUNTRY_CODE	COUNTRY_NAME
TH	Thailand
UK	United Kingdom
US	United States

3. ตาราง audit

AUDIT_ID	CUSTOMER_ID	LOG_DATE	USED
1	C001	2015-08-01	100000.00
2	C001	2015-08-05	200000.00
3	C001	2015-08-10	300000.00
4	C002	2015-08-02	400000.00
5	C002	2015-08-07	100000.00
6	C002	2015-08-15	300000.00
7	C003	2015-08-20	400000.00
8	C003	2015-08-25	200000.00
9	C004	2015-07-04	100000.00
10	C005	2015-08-01	100000.00
11	C005	2015-08-05	200000.00
12	C005	2015-08-10	300000.00
13	C006	2015-08-02	400000.00
14	C006	2015-08-07	100000.00
15	C006	2015-08-15	300000.00
16	C007	2015-08-20	400000.00
17	C007	2015-08-25	200000.00
18	C008	2015-07-04	100000.00

4. Data Dictionary

customer05 table

Data Item	Data Type	Description
CUSTOMER_ID	varchar(4)	รหัสลูกค้า
NAME	varchar(50)	ชื่อลูกค้า
EMAIL	varchar(50)	อีเมลล์ลูกค้า
COUNTRY_CODE	varchar(2)	รหัสประเทศ
BUDGET	decimal(18,2)	วงเงินบัตร (บาท)
USED	decimal(18,2)	ยอดเงินใช้จ่าย (บาท)

country table

COUNTRY_CODE	varchar(2)	รหัสประเทศ
COUNTRY_NAME	varchar(50)	ชื่อประเทศ

audit table

Data Item	Data Type	Description
AUDIT_ID	int(4)	รหัสตรวจสอบบัญชี
CUSTOMER_ID	varchar(4)	ชื่อลูกค้า
LOG_DATE	date	วันที่ใช้จ่าย
USED	decimal(18,2)	ยอดเงินใช้จ่าย (บาท)

5. โค้ดคำสั่งสร้างตาราง customer05 และ audit พร้อมทั้งคำสั่งการเพิ่มข้อมูลอยู่ในช่องใบงานดังรูป

ใบงานที่ 5 Draft

Due Jan 12, 11:59 PM

คำสั่งสร้างตาราง audit และ ...
Google Docs

คำสั่งสร้างตาราง customer ...
Google Docs

[Edit assignment](#)

1. จงสร้าง View โดยมีเงื่อนไขดังนี้

1. จงสร้าง View ชื่อ getCustomer_info01
2. จงแสดงข้อมูลลูกค้าได้แก่ รหัส, ชื่อ-สกุล, email, รหัสประเทศ, วงเงินบัตรเครดิต, ยอดเงินใช้จ่าย
3. กำหนดให้แสดงข้อมูลลูกค้าที่มีรหัสประเทศ US เท่านั้น
4. กำหนดให้แสดงเฉพาะข้อมูลที่มีวงเงินบัตรเครดิตน้อยกว่าหรือเท่ากับ 3,000,000 บาท
5. กำหนดให้เรียงลำดับข้อมูลรหัสลูกค้าจากมากไปน้อย

จงเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสนิติ)

Condition	ภาพหน้าจอผลลัพธ์		
1.1 ภาพหน้าจอ View Code [0.8 คะแนน]			
1.2 ภาพหน้าจอ คำสั่งเรียกใช้ View getCustomer_info01 [0.6 คะแนน]			
1.3 ภาพหน้าจอ ผลลัพธ์ [0.6 คะแนน]			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	2		

เกณฑ์ข้อ 1.1

ช่วงคะแนน	เกณฑ์ละเอียด
0.68–0.80 (A)	สร้าง View ได้จริง โค้ดถูกต้องครบทุกเงื่อนไข: แสดงคอลัมน์ (รหัส/ชื่อ/email/รหัสประเทศ/วงเงิน/ยอดใช้) + WHERE COUNTRY_CODE='US' + BUDGET<=3000000 + ORDER BY CUSTOMER_ID DESC และภาพชัดเจนฐานข้อมูล
0.56–0.67 (B)	โค้ดทำงานได้และเงื่อนไขถูกเกือบครบ แต่ผิดเล็กน้อย 1 จุด (เช่น ORDER BY ผิด ASC/DESC หรือเลือกคอลัมน์ตก 1 ช่อง)
0.32–0.55 (C)	โค้ดรันได้แต่ผิด 2 จุดขึ้นไป (เช่น ขาดเงื่อนไข BUDGET หรือกรองประเทศผิด) หรือสร้าง View ได้แต่ไม่ตรงโจทย์
0.00–0.31 (D)	สร้าง View ไม่ได้ / syntax error / ไม่ส่ง / ภาพไม่ชัดเจนฐานข้อมูล

เกณฑ์ข้อ 1.2

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	คำสั่งเรียกใช้ถูกต้อง เช่น SELECT * FROM getCustomer_info01; และภาพชัดเจนผลการรัน
0.42–0.50 (B)	เรียกใช้ได้ถูกต้องแต่รูปแบบภาพไม่ชัด/ครอบไม่เต็มเล็กน้อย
0.24–0.41 (C)	เรียกผิดเล็กน้อยแล้วแก้ไข/มีความกำกวม (เช่น พิมพ์ชื่อ view ผิดในภาพแรก)
0.00–0.23 (D)	เรียกใช้ไม่ได้ / ไม่ส่ง

เกณฑ์ข้อ 1.3

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	ผลลัพธ์ตรงทุกเงื่อนไข: มีแต่ US, BUDGET ≤ 3,000,000 และเรียง CUSTOMER_ID จากมากไปน้อย
0.42–0.50 (B)	ผลลัพธ์ถูกต้องแต่ผิดเล็กน้อย 1 จุด (เช่น ลำดับไม่ตรง)
0.24–0.41 (C)	ผลลัพธ์ผิดหลายจุด (มีประเทศอื่น/เกินงบ/เรียงไม่ถูก) แต่ยังคงแสดงข้อมูลบางส่วนได้
0.00–0.23 (D)	ไม่มีผลลัพธ์/ไม่ส่ง/ภาพไม่ชัดเจนข้อมูล

2. จงสร้าง View โดยมีเงื่อนไขดังนี้

1. จงสร้าง View ชื่อ getCustomer_info02
2. จงแสดงข้อมูลลูกค้าได้แก่ รหัส, ชื่อ-สกุล, อีเมล, วงเงินบัตรเครดิต, ยอดเงินใช้จ่าย และ ชื่อประเทศ
3. กำหนดให้แสดงข้อมูลลูกค้าที่มีรหัสประเทศ US เท่านั้น
4. กำหนดให้เรียงลำดับข้อมูลตามรหัสลูกค้าจากน้อยไปมาก

จงเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
2.1 ภาพหน้าจอ Code [0.8 คะแนน]			
2.2 ภาพหน้าจอ คำสั่งเรียกใช้ View getCustomer_info02 [0.6 คะแนน]			
2.3 ภาพหน้าจอ ผลลัพธ์ [0.6 คะแนน]			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	2		

เกณฑ์ข้อ 2.1

ช่วงคะแนน	เกณฑ์ละเอียด
0.68–0.80 (A)	สร้าง View ได้จริง ใช้ JOIN customer05 กับ country ถูกต้อง (customer05.COUNTRY_CODE = country.COUNTRY_CODE) เลือกคอลัมน์ครบตามโจทย์ (รหัส/ชื่อ/อีเมล/วงเงิน/ยอดใช้/ชื่อประเทศ) และ กรอง US + ORDER BY CUSTOMER_ID ASC
0.56–0.67 (B)	JOIN ถูกต้องและรันได้ แต่ผิดเล็กน้อย 1 จุด (เช่นเรียงลำดับผิด หรือคอลัมน์ตก 1 ช่อง)
0.32–0.55 (C)	JOIN รันได้แต่ผิดหลายจุด (เช่นไม่กรอง US หรือ join key ผิดแต่ยังได้ผลลัพธ์บางส่วน)
0.00–0.31 (D)	JOIN ไม่ได้ / syntax error / ไม่ส่ง

เกณฑ์ข้อ 2.2

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	SELECT * FROM getCustomer_info02; ถูกต้อง
0.42–0.50 (B)	เรียกถูกต้องแต่ภาพไม่ชัด/ครอบไม่ดี
0.24–0.41 (C)	เรียกผิดเล็กน้อยหรือมีความกำกวม
0.00–0.23 (D)	เรียกไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 2.3

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	ผลลัพธ์มีเฉพาะ US และเรียง CUSTOMER_ID จากน้อยไปมาก พร้อมชื่อประเทศถูกต้อง
0.42–0.50 (B)	ถูกต้องเกือบครบ ผิดเล็กน้อย 1 จุด (เช่นเรียงผิด)
0.24–0.41 (C)	ผลลัพธ์ผิดหลายจุด แต่ยังเห็นการ join ได้
0.00–0.23 (D)	ไม่มีผลลัพธ์/ไม่ส่ง

3. จงเรียกใช้ View โดยมีเงื่อนไขดังนี้

1. จงเรียกใช้ View ชื่อ getCustomer_info03
2. จงแสดงข้อมูลลูกค้าได้แก่ รหัส, ชื่อ-สกุล, ชื่อประเทศ, วงเงินบัตร, ยอดเงินใช้จ่าย และ จำนวนครั้งที่ใช้จ่าย
3. กำหนดให้แสดงเฉพาะข้อมูลลูกค้าที่มียอดเงินที่ใช้ไปน้อยกว่าหรือเท่ากับ 300,000 บาท
4. กำหนดให้แสดงเฉพาะข้อมูลลูกค้าที่มีภูมิลำเนาอยู่ที่ประเทศ United States
5. กำหนดให้เรียงลำดับข้อมูลรหัสลูกค้าจากมากไปหาน้อย
6. จงแสดงจำนวนครั้งที่ใช้จ่ายของลูกค้าแต่ละคน โดยกำหนดชื่อคอลัมน์คือ ITEM_AUDIT
7. ตัวอย่างชื่อคอลัมน์

CUSTOMER_ID	NAME	COUNTRY_NAME	BUDGET	USED	ITEM_AUDIT
-------------	------	--------------	--------	------	------------

จงเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์		
3.1 ภาพหน้าจอ Code [0.9 คะแนน]			
3.2 ภาพหน้าจอ คำสั่งเรียกใช้ getCustomer_info03 [0.5 คะแนน]			
3.3 ภาพหน้าจอ ผลลัพธ์ [0.6 คะแนน]			
การทดลองที่	คะแนนเต็ม	คะแนนที่ได้	ผู้ตรวจ
1	2		

เกณฑ์ข้อ 3.1

ช่วงคะแนน	เกณฑ์ละเอียด
0.77-0.90 (A)	สร้าง View/คำสั่งตามโจทย์ได้ครบ: JOIN 3 ตาราง (customer05 + country + audit) + เงื่อนไข USED ≤ 300,000 และประเทศ United States + GROUP BY ลูกค้า + COUNT(audit) ตั้งชื่อคอลัมน์ ITEM_AUDIT + ORDER BY CUSTOMER_ID DESC
0.63-0.76 (B)	โค้ดทำงานได้และเกือบครบ ผิดเล็กน้อย 1 จุด (เช่น alias ITEM_AUDIT ไม่ตรง, order ผิด)
0.36-0.62 (C)	โค้ดรันได้แต่ขาดส่วนสำคัญ (เช่นไม่ group by/ไม่ count/กรองประเทศผิด)
0.00-0.35 (D)	สร้างไม่ได้/ผิดทั้งหมด/ไม่ส่ง

เกณฑ์ข้อ 3.2

ช่วงคะแนน	เกณฑ์ละเอียด
0.43-0.50 (A)	เรียกใช้ view ได้ถูกต้องและภาพชัด
0.35-0.42 (B)	เรียกได้แต่ภาพไม่ชัดเล็กน้อย
0.20-0.34 (C)	เรียกผิดเล็กน้อย/ไม่ชัดเจน
0.00-0.19 (D)	เรียกไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 3.3

ช่วงคะแนน	เกณฑ์ละเอียด
0.51-0.60 (A)	ผลลัพธ์ตรงเงื่อนไขทั้งหมด และมีคอลัมน์ ITEM_AUDIT แสดงจำนวนครั้งถูกต้อง
0.42-0.50 (B)	ผลลัพธ์ถูกต้องเกือบครบ ผิดเล็กน้อย (เช่นการเรียง)
0.24-0.41 (C)	ผลลัพธ์คลาดเคลื่อนหลายจุดแต่ยังแสดงการ join/group ได้
0.00-0.23 (D)	ไม่มีผลลัพธ์/ไม่ส่ง

4. จงสร้าง View โดยมีเงื่อนไขดังนี้

1. จงสร้าง View ชื่อ mst_customer จากตาราง customer05
 - จงสร้าง View เพื่อแสดงข้อมูลลูกค้าได้แก่ รหัส ชื่อ-สกุล อีเมล รหัสประเทศ วงเงินบัตรเครดิต ยอดเงินที่ใช้จ่ายโดยกำหนดชื่อคอลัมน์ใน view mst_customer ได้แก่ CM_ID, CM_NAME, CM_EMAIL, CM_COUNTRY_CODE, CM_BUDGET และ CM_USED
2. จงสร้าง View ชื่อ mst_country จากตาราง country
 - จงสร้าง View เพื่อแสดงข้อมูลประเทศได้แก่ รหัสประเทศ ชื่อประเทศ โดยกำหนดชื่อคอลัมน์ใน view mst_country ได้แก่ CT_COUNTRY_CODE และ CT_COUNTRY_NAME
3. จงสร้าง View ชื่อ trn_audit จากตาราง audit
 - จงสร้าง View เพื่อแสดงข้อมูลการใช้จ่ายของลูกค้าได้แก่ รหัสตรวจสอบบัญชี รหัสลูกค้า วันที่ใช้จ่าย ยอดเงินที่ใช้จ่าย โดยกำหนดชื่อคอลัมน์ใน view trn_audit ได้แก่ CM_AUDIT_ID, CM_ID, CM_LOG_DATE และ CM_USED
4. จงสร้าง View ชื่อ trn_report เรียกใช้ข้อมูลตารางจาก view mst_customer, mst_country และ trn_audit
 - จงแสดงข้อมูลลูกค้าได้แก่ รหัส, ชื่อ-สกุล, ชื่อประเทศ, วงเงินบัตร, ยอดใช้จ่ายรวม และจำนวนครั้งที่ใช้จ่าย
 - กำหนดให้ข้อมูลยอดใช้จ่ายรวมมีชื่อคอลัมน์คือ NUM_USED
 - กำหนดให้ข้อมูลจำนวนครั้งที่ใช้จ่ายมีชื่อคอลัมน์คือ TOTAL_USED
 - จงแสดงข้อมูลลูกค้าที่มียอดใช้จ่ายรวม**สูงสุด** 1 คน

เพิ่มเติมผลลัพธ์การสร้างลงในตารางด้านล่าง (ในภาพต้องเห็นก่อน Database รหัสสินค้า)

Condition	ภาพหน้าจอผลลัพธ์
4.1 ภาพ หน้าจอCode สร้าง view ชื่อ mst_customer [0.8 คะแนน]	
4.2 ภาพ หน้าจอCode สร้าง view ชื่อ mst_country [0.6 คะแนน]	
4.3 ภาพ หน้าจอCode สร้าง view ชื่อ trn_audit [0.6 คะแนน]	

<p>4.4 ภาพ หน้าจอCode สร้าง view ชื่อ trn_report [0.6 คะแนน]</p>			
<p>4.5 ภาพ หน้าจอคำสั่ง เรียกใช้ view ชื่อ trn_report [0.6 คะแนน]</p>			
<p>4.6 ภาพหน้าจอ ผลลัพธ์ view ชื่อ trn_report [0.8 คะแนน]</p>			
<p>การทดลองที่</p>	<p>คะแนนเต็ม</p>	<p>คะแนนที่ได้</p>	<p>ผู้ตรวจ</p>
<p>1</p>	<p>4</p>		

เกณฑ์ข้อ 4.1

ช่วงคะแนน	เกณฑ์ละเอียด
0.68–0.80 (A)	สร้าง view ได้จริงและ alias ครบถูกต้อง: CM_ID, CM_NAME, CM_EMAIL, CM_COUNTRY_CODE, CM_BUDGET, CM_USED
0.56–0.67 (B)	alias ถูกเกือบครบ ผิด 1 จุด (สะกด/ลำดับ)
0.32–0.55 (C)	สร้างได้แต่ alias ผิดหลายจุด/คอลัมน์ตก
0.00–0.31 (D)	สร้างไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 4.2

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	alias ถูกต้อง: CT_COUNTRY_CODE, CT_COUNTRY_NAME
0.42–0.50 (B)	ผิดเล็กน้อย 1 จุด
0.24–0.41 (C)	alias ผิดหลายจุด/ไม่ครบ
0.00–0.23 (D)	สร้างไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 4.3

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	alias ถูกต้องครบ: CM_AUDIT_ID, CM_ID, CM_LOG_DATE, CM_USED
0.42–0.50 (B)	ผิดเล็กน้อย 1 จุด
0.24–0.41 (C)	alias ผิดหลายจุด/ไม่ครบ
0.00–0.23 (D)	สร้างไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 4.4

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	ดึงจาก 3 view (mst_customer + mst_country + trn_audit) ถูกต้อง ใช้ SUM(CM_USED) เป็น NUM_USED และ COUNT(...) เป็น TOTAL_USED พร้อมกลุ่มลูกค้า
0.42–0.50 (B)	โค้ดทำงานได้แต่ผิดเล็กน้อย (เช่น alias NUM_USED/TOTAL_USED ไม่ตรง)
0.24–0.41 (C)	สร้างได้แต่รวม/นับผิด หรือ join view ผิดบางส่วน
0.00–0.23 (D)	สร้างไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 4.5

ช่วงคะแนน	เกณฑ์ละเอียด
0.51–0.60 (A)	เรียกใช้ view ได้ถูกต้องและภาพชัด
0.42–0.50 (B)	เรียกได้แต่ภาพไม่ชัดเล็กน้อย
0.24–0.41 (C)	เรียกผิดเล็กน้อย/ไม่ชัดเจน
0.00–0.23 (D)	เรียกไม่ได้/ไม่ส่ง

เกณฑ์ข้อ 4.6

ช่วงคะแนน	เกณฑ์ละเอียด
0.68–0.80 (A)	แสดง “ลูกค้ายอดใช้จ่ายรวมสูงสุด 1 คน” ถูกต้อง (ใช้ ORDER BY NUM_USED DESC + LIMIT 1 หรือวิธีที่เทียบเท่า) และคอลัมน์ครบ
0.56–0.67 (B)	ได้ Top 1 ถูกต้อง แต่ผิดเล็กน้อย (เช่น ชื่อคอลัมน์/การจัดรูปแบบ)
0.32–0.55 (C)	แสดงผลได้แต่ไม่ใช่ Top 1 หรือการรวมยอดผิด
0.00–0.31 (D)	ไม่มีผลลัพธ์/ไม่ส่ง

ภาคผนวก ข
แบบบันทึกคะแนนใบงาน

ใบลงคะแนนการตรวจใบงานวิชา Advance Database กลุ่มที่ 1

No.	ID	ใบงานที่ [5/week]					total 50	10%
		1	2	3	4	5		
1	66160048	7.5	7.5	9.5	10	9.5	44	8.8
2	66160050	8	5.5	10	8	9.5	41	8.2
3	66160054	5.5	7.5	8	9	5	35	7.0
4	66160060	8	7.5	9	9	4.5	38	7.6
5	66160062	9.5	8	9	8.5	10	45	9.0
6	66160063	6.5	5.5	9	9	0	30	6.0
7	66160064	7	10	9.5	9	10	45.5	9.1
8	66160065	6	5.5	0	9	9	29.5	5.9
9	66160066	9.5	6	9	10	8.5	43	8.6
10	66160067	6	5.5	9	9	9	38.5	7.7
11	66160068	6	8	9.5	7	9	39.5	7.9
12	66160071	5	5	9	10	10	39	7.8
13	66160072	6.5	8	9.5	10	9	43	8.6
14	66160199	6	4	0	7	9	26	5.2
15	66160200	7.5	7	9	9	8	40.5	8.1
16	66160203	9.5	7	10	5	10	41.5	8.3
17	66160204	9	8	10	9	10	46	9.2
18	66160205	7	5.5	9.5	9	9	40	8.0
19	66160206	8.5	8	9.5	8	9	43	8.6

20	66160207	doc	5.5	10	9	10	34.5	6.9
21	66160217	8	7	9	9	9	42	8.4
22	66160290	7.5	8	6.5	4	9	35	7.0
23	66160291	9	7.5	9.5	9	9	44	8.8
24	66160292	8.5	10	9.5	10	7.5	45.5	9.1
25	66160293	9.5	0	8	10	10	37.5	7.5
26	66160295	5.5	10	9.5	10	10	45	9.0
27	66160297	5.5	0	7.5	8	9.5	30.5	6.1
28	66160299	5.5	8	9.5	8	10	41	8.2
29	66160300	7.5	7.5	8.5	6	5	34.5	6.9
30	66160301	7	9.5	8.5	8	8	41	8.2
31	66160302	8	9	6	8	8	39	7.8
32	66160303	8.5	8.5	8.5	9.5	9	44	8.8
33	66160304	8	8	7	7.5	8	38.5	7.7
34	66160305	6.5	6.5	6	8.5	9	36.5	7.3
35	66160306	8	6	6	7.5	8	35.5	7.1
36	66160307	8	7	6	8	9	38	7.6
37	66160309	8	8	9	8.5	7.5	41	8.2
38	66160311	6	8	8.5	8	8	38.5	7.7
39	66160312	9	6.5	6.5	7	7	36	7.2
40	66160313	7	7	6	7	7	34	6.8
41	66160319	0	0	0	0	3.5	3.5	0.7
42	66160321	7	7	7	8	9	38	7.6

43	66160322	7	7	7	8	8.5	37.5	7.5
44	66160323	5	0	5	4	8	22	4.4
45	66160324	8	6.5	7	9	7.5	38	7.6
46	66160326	5	10	7.5	4	7.5	34	6.8
47	66160327	8	9	9.5	8.5	9	44	8.8
48	66160329	9.5	9	6.5	7.5	8.5	41	8.2
49	66160330	0	6.5	0	0	0	6.5	1.3
50	66160332	6	6.5	5.5	8	8.5	34.5	6.9
51	66160334	8	6	5	3	8	30	6.0
52	66160336	8	6	7	8	9	38	7.6
53	66160338	8.5	5	6.5	8.5	8	36.5	7.3
54	66160339	7.5	10	8	9	9	43.5	8.7
55	66160340	4	5	0	6.5	7.5	23	4.6

คะแนนการตรวจใบงานวิชา Advance Database กลุ่มที่ 2

No.	ID	ใบงานที่ [5/week]						
		1	2	3	4	5	total 50	10%
1	67160050	10	9	10	10	9.8	48.8	9.8
2	67160052	10	10	10	9	9	48	9.6
3	67160053	8.5	10	3.9	10	9	41.4	8.3
4	67160055	9	8	10	9	8	44	8.8
5	67160057	10	10	9	10	9.9	48.87	9.8
6	67160064	10	9	10	10	10	49	9.8
7	67160065	10	7.1	9	10	10	46.1	9.2
8	67160066	10	9.6	9	10	9	47.6	9.5
9	67160068	10	9.8	10	8.5	8.7	46.95	9.4
10	67160070	10	9	9.9	10	9.7	48.55	9.7
11	67160071	7	8.1	10	10	9.9	45	9.0
12	67160074	10	9	9	9	9	46	9.2
13	67160076	9	8	10	9	9.3	45.3	9.1
14	67160077	9	7.8	9	8.8	8.8	43.37	8.7
15	67160078	10	10	10	9.5	10	49.5	9.9
16	67160081	9	9.6	9.5	10	8.7	46.77	9.4
17	67160085	10	9.3	10	9.9	9	48.2	9.6
18	67160087	10	9.8	9.5	9.5	9	47.8	9.6
19	67160089	10	9.9	9	9.5	10	48.4	9.7
20	67160098	8	10	10	10	10	48	9.6
21	67160099	10	9.3	10	10	10	49.3	9.9
22	67160101	9.5	9.4	10	9.9	10	48.8	9.8
23	67160104	10	9.2	10	9.8	9	48	9.6

24	67160106	10	10	10	10	10	50	10.0
25	67160107	10	8	10	10	10	48	9.6
26	67160108	10	10	10	10	9.7	49.7	9.9
27	67160110	9	8.7	10	10	9.7	47.37	9.5
28	67160111	10	9.4	9	10	10	48.4	9.7
29	67160245	9	9.7	10	10	10	48.7	9.7
30	67160248	9	10	9.4	10	10	48.4	9.7
31	67160251	10	8.9	9	8.5	10	46.4	9.3
32	67160253	9	9	8.5	9	9	44.5	8.9
33	67160254	9.3	9	10	9	8.1	45.4	9.1
34	67160256	9	9	8.4	6.5	9.6	42.5	8.5
35	67160258	9	8.4	8.5	7.5	8.8	42.2	8.4
36	67160261	10	10	10	9.5	9.9	49.4	9.9
37	67160262	10	10	9	9	9.6	47.6	9.5
38	67160264	10	10	9	10	8.6	47.6	9.5
39	67160265	10	10	7.4	9	9.9	46.3	9.3
40	67160267	9	9	8.7	9	9.6	45.3	9.1
41	67160268	10	8.9	4.5	10	10	43.4	8.7
42	67160271	10	10	9	10	8.6	47.6	9.5
43	67160272	10	9	9	8.5	9.6	46.1	9.2
44	67160274	9	9	9	9	9.4	45.4	9.1
45	67160275	10	9.8	9	5	8.2	42	8.4
46	67160277	10	0	8.5	8	10	36.5	7.3
47	67160417	10	9	9	10	9.6	47.6	9.5
48	67160419	10	9	9	9	9.6	46.6	9.3
49	67160420	10	10	9	9.5	8.6	47.1	9.4

50	67160423	10	9	8.5	9	9	45.5	9.1
51	67160425	9	10	9	9	9.2	46.2	9.2
52	67160426	10	9	0	10	9.7	38.7	7.7
53	67160427	10	10	8.5	9	9.2	46.7	9.3
54	67160428	10	10	10	9.5	9.6	49.1	9.8
55	67160429	9	9	4.5	9.5	9.6	41.6	8.3