



การปฏิบัติการฐานข้อมูล
(Database Workshop)



ผู้ช่วยศาสตราจารย์ ดร.ปริศนา มั่งคิมมา

คณะมนุษยศาสตร์และสังคมศาสตร์

มหาวิทยาลัยราชภัฏสวนดุสิต

๒๕๕๕

การปฏิบัติการฐานข้อมูล
(Database Workshop)

ผู้ช่วยศาสตราจารย์ ดร. ปรศนา มณีนิมา
ปร.ด. (เทคโนโลยีสารสนเทศ)

คณะมนุษยศาสตร์และสังคมศาสตร์
มหาวิทยาลัยราชภัฏวไลยอลงกรณ์
2555

คำนำ

ตำราการปฏิบัติการฐานข้อมูลเล่มนี้ เป็นการเขียนในลักษณะที่มีข้อมูลประกอบเชิงทฤษฎี และปฏิบัติ ที่มุ่งเน้นให้ผู้เรียนสามารถจัดการฐานข้อมูลด้วยโปรแกรมสำเร็จรูป Microsoft Access ได้อย่างมีประสิทธิภาพ และสามารถประยุกต์ใช้งานในด้านต่างๆ ได้โดยผ่านกรณีศึกษา ซึ่งตำราเล่มนี้ผู้เขียนได้จัดทำขึ้นเพื่อใช้เป็นตำราประกอบการเรียนการสอนวิชาการปฏิบัติการฐานข้อมูล รหัสวิชา 4122208 ตามหลักสูตรวิทยาศาสตรบัณฑิต ของหลักสูตรเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏสวนดุสิต นอกจากนั้นตำราเล่มนี้ยังสามารถนำไปใช้ศึกษาค้นคว้า ในระดับอุดมศึกษาของสถาบันอื่นๆ ได้อีกด้วย

เนื้อหาในตำราได้แบ่งออกเป็น 7 หัวเรื่อง ซึ่งประกอบด้วย ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล การใช้งานโปรแกรม Microsoft Access การสร้างตาราง แบบสอบถาม ฟอรัม รายงาน และมาโคร

ท่านที่นำตำราเล่มนี้ไปใช้ควรศึกษาเพิ่มเติมจากเอกสารอื่นๆ ประกอบด้วย และหวังว่าตำราเล่มนี้จะเป็นประโยชน์สำหรับนักศึกษาและผู้สนใจ หากมีข้อบกพร่องประการใด ผู้เขียนขอน้อมรับไว้และจะพิจารณาแก้ไขปรับปรุงต่อไป

ปรีศนา มัชฌิมา

25 พฤษภาคม 2555

สารบัญ

	หน้า
คำนำ	(1)
สารบัญ	(3)
บทที่ 1 ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล	1
ขั้นตอนการพัฒนาาระบบฐานข้อมูล	1
แบบจำลองอี-อาร์	4
นอร์มัลไลเซชัน (Normalization)	16
บทที่ 2 การใช้งานโปรแกรม Microsoft Access	25
โครงสร้างของโปรแกรม Microsoft Access	25
เริ่มต้นใช้งานโปรแกรม Microsoft Access 2010	26
ส่วนประกอบของหน้าต่างโปรแกรม Microsoft Access 2010	29
การทำงานของเมนูแบบบริบบอน (Ribbon)	30
การแปลงไฟล์ .mdb ให้เป็นไฟล์ .accdb	32
การจัดเก็บไฟล์ฐานข้อมูล	33
บทที่ 3 การสร้างตาราง (Table)	35
มุมมองของตาราง	35
ส่วนประกอบของตาราง	36
การสร้างตาราง	37
บทที่ 4 การสร้างแบบสอบถาม (Query)	55
มุมมองของแบบสอบถาม	55
ประเภทของแบบสอบถาม	56
วิธีการสร้างแบบสอบถาม	57
สร้างแบบสอบถามโดยใช้ตัวช่วยสร้าง	57

สารบัญ (ต่อ)

	หน้า
การสร้างแบบสอบถามด้วยมุมมองการออกแบบ	59
การสร้างแบบสอบถามในมุมมอง SQL	77
บทที่ 5 การสร้างฟอร์ม (Form)	79
มุมมองของฟอร์ม	79
การสร้างฟอร์มแบบง่าย	80
การสร้างฟอร์มโดยใช้ตัวช่วย (Form Wizard)	81
การสร้างฟอร์มด้วยมุมมองการออกแบบ	83
ปุ่มสร้างคอนโทรลในแท็บ Design	84
ส่วนประกอบในมุมมองออกแบบของฟอร์ม	86
การเชื่อมฟอร์มเข้ากับตารางฐานข้อมูล	92
การปรับแต่งฟอร์ม	94
การจัดเก็บฟอร์มลงฐานข้อมูล	95
บทที่ 6 การสร้างรายงาน (Report)	97
มุมมองของรายงาน	97
วิธีการสร้างรายงาน	97
การสร้างรายงานแบบง่าย	98
สร้างรายงานโดยใช้ตัวช่วยสร้าง (Report Wizard)	100
สร้างรายงานในมุมมองออกแบบ	102
ส่วนประกอบในมุมมองออกแบบของรายงาน	103
ตัวอย่างการสร้างรายงาน	105
การ export รายงาน	110
วิธีสร้างรายงานจากแบบสอบถาม	112

สารบัญ (ต่อ)

	หน้า
บทที่ 7 การสร้างมาโคร (Macro)	113
รู้จักกับมาโคร	113
เริ่มต้นสร้างมาโคร	114
คำสั่งใน Action Catalog	115
วิธีสร้างมาโครโดยเก็บเป็นออบเจ็ค Macro	117
การแก้ไขมาโคร	121
วิธีสั่งรันมาโคร	124
บรรณานุกรม	127

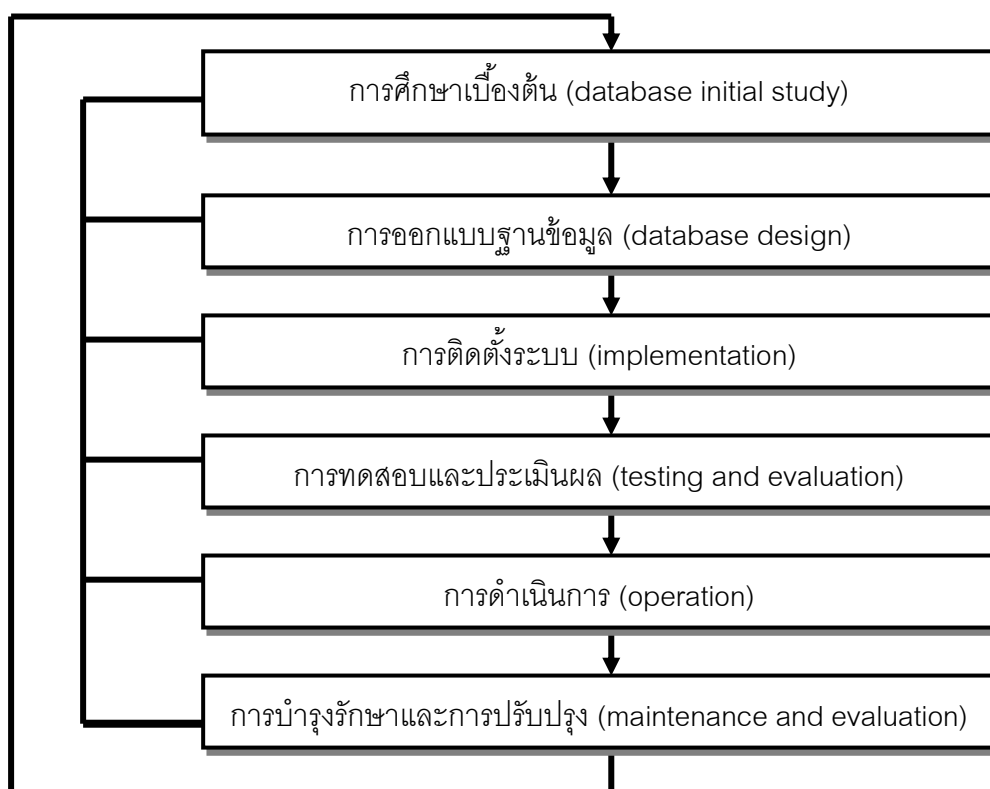
บทที่ 1

ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล

ฐานข้อมูลที่มีประสิทธิภาพและตรงกับความต้องการของผู้ใช้นั้น ต้องอาศัยการพัฒนาฐานข้อมูลที่มีการวางแผนอย่างเป็นระบบและมีขั้นตอนที่ถูกต้อง โดยในบทนี้จะศึกษาถึงขั้นตอนการพัฒนาฐานข้อมูล ซึ่งจะเน้นขั้นตอนการออกแบบฐานข้อมูลเป็นหลัก ตั้งแต่การพัฒนาแบบจำลองอี-อาร์ และการทำนอร์มัลไลเซชัน เพื่อให้ได้ฐานข้อมูลที่สมบูรณ์ที่สุด

ขั้นตอนการพัฒนาฐานข้อมูล

วัฏจักรฐานข้อมูล (The Database Life Cycle: DBLC) เป็นขั้นตอนในการพัฒนาหรือจัดทำระบบฐานข้อมูล ซึ่งประกอบด้วย 6 ขั้นตอน ดังภาพที่ 1.1



ภาพที่ 1.1 วัฏจักรฐานข้อมูล (The Database Life Cycle: DBLC)
ที่มา (Rob and Coronel, 2002, p.326)

1. การศึกษาเบื้องต้น

การศึกษาเบื้องต้นมีวัตถุประสงค์เพื่อวิเคราะห์สถานการณ์ขององค์กร กำหนดปัญหาและข้อจำกัด กำหนดวัตถุประสงค์และขอบเขตของระบบ ดังรายละเอียดต่อไปนี้

1.1 การวิเคราะห์สถานการณ์ขององค์กร

เพื่อศึกษาสภาพแวดล้อมในการทำงานขององค์กร ความต้องการในการปฏิบัติงาน โดยควรรู้ว่าโครงสร้างขององค์กรเป็นอย่างไร ใครเป็นผู้ควบคุมอะไร และใครทำรายงานให้ใคร เป็นต้น

1.2 กำหนดปัญหาและข้อจำกัด

โดยการศึกษาว่า ระบบที่มีอยู่มีการทำงานอย่างไร ข้อมูลที่ป้อนเข้าสู่ระบบมีอะไรบ้าง และระบบสร้างรายงานอะไร มีการใช้รายงานเหล่านี้อย่างไรและใครเป็นผู้ใช้ เพื่อให้ทราบถึงปัญหาและข้อจำกัดในการป้อนข้อมูล หรือการค้นหาข้อมูลเพื่อการทำรายงาน

1.3 กำหนดวัตถุประสงค์และขอบเขตของระบบ

ในการกำหนดวัตถุประสงค์ของระบบฐานข้อมูลควรสอดคล้องกับความต้องการของผู้ใช้ จากคำถามเหล่านี้

- วัตถุประสงค์แรกเริ่มของระบบที่นำเสนอคืออะไร
- ระบบนี้ต้องเชื่อมต่อกับระบบอื่นๆ ที่มีอยู่ในองค์กรหรือไม่
- ระบบนี้จะมีการใช้ข้อมูลร่วมกันกับระบบหรือผู้ใช้อื่นหรือไม่

เมื่อทราบวัตถุประสงค์แล้วก็ทำการกำหนดขอบเขตของระบบโดยการออกแบบตามความต้องการในการปฏิบัติงาน เพื่อใช้ในการออกแบบฐานข้อมูลต่อไป

2. การออกแบบฐานข้อมูล

เมื่อผู้ออกแบบฐานข้อมูลมีความเข้าใจลักษณะขององค์กร ปัญหาและข้อจำกัดรวมทั้งวัตถุประสงค์และขอบเขตของระบบแล้ว ก็ทำการออกแบบฐานข้อมูลดังต่อไปนี้

2.1 การออกแบบเชิงแนวคิด

โดยการพัฒนาแบบจำลองอี-อาร์ (E-R Model) ที่ใช้อธิบายถึงความสัมพันธ์ระหว่างสิ่งที่เราสนใจจะจัดเก็บ ที่เรียกว่า เอนทิตี (entity) และรายละเอียดหรือคุณสมบัติ (attribute) ของสิ่งที่จัดเก็บ แล้วทำการแปลงแบบจำลองอี-อาร์ เป็นโครงสร้างตารางฐานข้อมูล จากนั้นก็ทำการนอร์มัลไลเซชัน (normalization) เพื่อให้ได้โครงสร้างของตารางที่ดี สามารถควบคุมความซ้ำซ้อนของข้อมูล หลีกเลี่ยงความผิดปกติของข้อมูล

2.2 การเลือกโปรแกรมจัดการฐานข้อมูล

ในการตัดสินใจเลือกซื้อโปรแกรมจัดการฐานข้อมูลขององค์กรใด ควรพิจารณาถึงสิ่งต่อไปนี้

2.2.1 ค่าใช้จ่ายต่างๆ เช่น ราคาการซ่อมบำรุง การปฏิบัติงาน ลิขสิทธิ์ การติดตั้ง การฝึกอบรม และค่าใช้จ่ายในการเปลี่ยนไปใช้ระบบใหม่

2.2.2 คุณลักษณะและเครื่องมือของระบบจัดการฐานข้อมูล โปรแกรมฐานข้อมูลบางตัวจะรวมเอาเครื่องมือต่างๆ ที่ให้ความสะดวกในงานการพัฒนาโปรแกรมประยุกต์ ตัวอย่างเช่น การออกแบบหน้าจอ การสร้างรายงาน การสร้างโปรแกรมประยุกต์ และพจนานุกรมข้อมูล เป็นต้น ทำให้สะดวกในการบริหารฐานข้อมูล ใช้งาน มีความสามารถในการรักษาความปลอดภัย และการควบคุมการใช้งานพร้อมกัน เป็นต้น

2.2.3 ความสามารถในการใช้ข้าม platforms ข้ามระบบและภาษา

2.2.4 ความต้องการด้านฮาร์ดแวร์ หน่วยความจำ และเนื้อที่ที่ใช้ในการจัดเก็บ

2.3 การออกแบบทางตรรกะ

จะเกี่ยวข้องกับการตัดสินใจใช้รูปแบบเฉพาะของฐานข้อมูล (แบบลำดับชั้น แบบเครือข่าย และแบบเชิงสัมพันธ์ เป็นต้น) การกำหนดรูปแบบของฐานข้อมูล ซึ่งการออกแบบเชิงตรรกะจะเป็นการแปลงการออกแบบระดับเชิงแนวคิด ให้เป็นแบบจำลองของฐานข้อมูลในระดับภายใน (internal model) ตามระบบการจัดการฐานข้อมูล (DBMS) เช่น MS-Access และ Oracle โดยการสร้างตาราง ฟอรั่ม คิวรี และรายงาน เป็นต้น

2.4 การออกแบบทางกายภาพ

การออกแบบทางกายภาพ คือ กระบวนการในการเลือกหน่วยจัดเก็บข้อมูล และลักษณะการเข้าถึงข้อมูลของฐานข้อมูล การสร้างดรรชนี (index) การจัดทำคลัสเตอร์ (clustering) ซึ่งเป็นการจัดเก็บข้อมูลที่มีการใช้งานบ่อยๆ ไว้ในหน่วยเก็บข้อมูลเดียวกัน หรือการใช้เทคนิคแฮชซิง (hashing technique) ในการจัดตำแหน่งที่อยู่ของข้อมูลภายในหน่วยเก็บข้อมูล เป็นต้น

3. การติดตั้งระบบ

ขึ้นอยู่กับระบบจัดการฐานข้อมูลที่ใช้ โดยเริ่มต้นจากการสร้างฐานข้อมูล กำหนดผู้จัดการฐานข้อมูล กำหนดพื้นที่ๆ ที่ต้องการใช้ และการสร้างตารางต่างๆ ในระบบ

4. การทดสอบและประเมินผล

เพื่อการตรวจสอบดูว่าระบบที่พัฒนามาสามารถทำงานได้ตามที่ต้องการหรือไม่ ซึ่งควรมีการเตรียมข้อมูลทดสอบไว้ล่วงหน้า

5. การดำเนินการ

เมื่อฐานข้อมูลผ่านขั้นตอนการทดสอบและประเมินผล ต่อไปก็เป็นขั้นตอนการดำเนินการ หรือการติดตั้งระบบ ซึ่งต้องเป็นระบบที่สมบูรณ์พร้อมให้ผู้ใช้ได้ใช้งานนั่นเอง ซึ่งอาจรวมไปถึงการฝึกอบรมให้แก่ผู้ใช้ ที่เป็นพนักงานที่ต้องใช้งานจริงด้วย

6. การบำรุงรักษาและการปรับปรุง

หลังจากระบบได้เริ่มดำเนินการ ผู้จัดการฐานข้อมูลจะต้องเตรียมการบำรุงรักษาฐานข้อมูลโดยการสำรองข้อมูลไว้ เพื่อสะดวกในการกู้คืนข้อมูล เมื่อระบบมีปัญหา และหากมีการใช้งานไปนานๆ อาจต้องทำการปรับปรุงแก้ไขโปรแกรมให้เหมาะสมกับเหตุการณ์ และความต้องการของผู้ใช้ที่เปลี่ยนแปลงไป

เมื่อเราทราบขั้นตอนการพัฒนาฐานข้อมูลทั้งหมดแล้ว ในบทนี้จะเน้นถึงรายละเอียดของขั้นตอนการออกแบบฐานข้อมูลเชิงแนวคิด ตั้งแต่การพัฒนาแบบจำลองอี-อาร์ (E-R Model) และการทำนอร์มัลไลเซชัน (normalization) ดังรายละเอียดที่จะกล่าวต่อไป

แบบจำลองอี-อาร์

แบบจำลองอี-อาร์ (Entity-Relationship Model: E-R Model) เป็นแบบจำลองข้อมูลที่ประยุกต์มาจากแนวคิดเรื่อง Semantic Model และมีการพัฒนามาเป็น E-R Model โดย Peter Pin Shan Chen จาก Massachusetts Institute of Technology ในปี ค.ศ. 1976 และได้รับความนิยมมาจนถึงปัจจุบัน

1. ความหมายและความสำคัญของแบบจำลองอี-อาร์

แบบจำลองอี-อาร์ เป็นเครื่องมือที่ใช้ในการออกแบบฐานข้อมูล ที่แสดงความสัมพันธ์ระหว่างเอนทิตีหรือสิ่งที่เราต้องการจะจัดเก็บไว้ในฐานข้อมูล โดยนำเสนอในรูปแบบของแผนภาพที่เรียกว่า อี-อาร์ไดอะแกรม (E-R Diagram) ด้วยการใช้สัญลักษณ์ต่างๆ

แบบจำลองอี-อาร์ มีความสำคัญในการเป็นสื่อกลางเพื่อสื่อสารกับบุคลากรต่างๆ ที่เกี่ยวข้องกับระบบฐานข้อมูล ไม่ว่าจะเป็นในระดับผู้บริหาร นักเขียนโปรแกรม และผู้ใช้ในระดับปฏิบัติการ เป็นต้น ทำให้เข้าใจระบบได้อย่างถูกต้องตรงกัน เนื่องจากมีการแสดงภาพรวมของระบบในลักษณะของรูปภาพหรือแผนภาพ ทำให้เข้าใจง่าย ดังนั้นระบบที่ออกแบบมาจึงมีความถูกต้องและเป็นไปตามวัตถุประสงค์ขององค์กร

2. องค์ประกอบของแบบจำลองอี-อาร์

แบบจำลองอี-อาร์ ประกอบด้วย เอนทิตี แอตทริบิวต์ คีย์ และความสัมพันธ์ ดังรายละเอียดต่อไปนี้

2.1 เอนทิตี

เอนทิตี (entity) คือ สิ่งต่างๆ ที่ผู้ใช้งานฐานข้อมูลต้องการจะจัดเก็บ ซึ่งมีลักษณะเป็นคำนาม ทั้งรูปธรรมและนามธรรม เช่น บุคคล สถานที่ วัตถุสิ่งของ และเหตุการณ์ต่างๆ เป็นต้น ตัวอย่างของเอนทิตีใน “ระบบการลงทะเบียนเรียนของนักศึกษา” ประกอบด้วย รายวิชา นักศึกษา การลงทะเบียน ผลการเรียนประจำเทอม สาขาวิชา คณะ และโปรแกรมวิชา เป็นต้น

เอนทิตีที่รวบรวมได้จากระบบสามารถแยกแยะและจัดเป็นหมวดหมู่ได้ตามชนิดของเอนทิตี ดังต่อไปนี้

- หมวดบุคคล ได้แก่ เอนทิตี → นักศึกษา พนักงาน ประชาชน ผู้ป่วย และลูกค้า เป็นต้น
- หมวดสถานที่ ได้แก่ เอนทิตี → รัฐ ประเทศ จังหวัด ภาค สาขา และวิทยาเขต เป็นต้น
- หมวดวัตถุ ได้แก่ เอนทิตี → อาคาร เครื่องจักร ผลผลิต หนังสือ วัตถุดิบ และรถยนต์ เป็นต้น
- หมวดเหตุการณ์ ได้แก่ เอนทิตี → การขาย การลงทะเบียน การเดินทาง การสั่งซื้อของ การออกไปเสิร์ฟรับเงิน และการให้รางวัล เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้า แทนหนึ่งเอนทิตี โดยใช้ชื่อของเอนทิตีนั้นๆ กำกับอยู่ภายใน เช่น

นักศึกษา

แทน เอนทิตีนักศึกษา

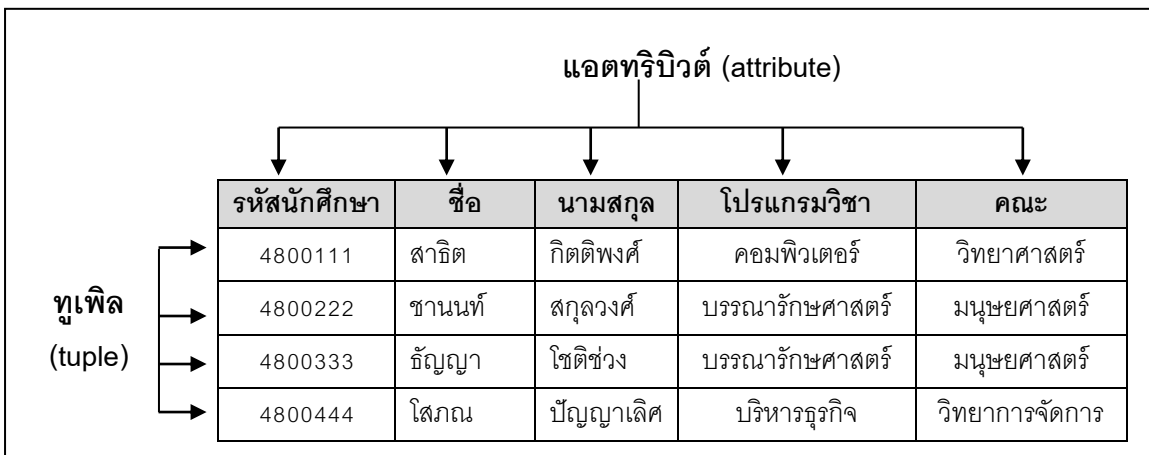
2.2 แอตทริบิวต์

แอตทริบิวต์ (attribute) คือ คุณสมบัติต่างๆ ของเอนทิตีที่เราต้องการจัดเก็บในฐานข้อมูล ตัวอย่างเช่น

- เอนทิตีบัตรประชาชน ประกอบด้วยแอตทริบิวต์ หรือสิ่งที่บ่งบอกคุณสมบัติของประชาชนแต่ละคน ได้แก่ หมายเลขบัตรประชาชน ชื่อ นามสกุล วันเดือนปีเกิด ภูมิลำเนา วันที่ออกบัตร วันที่บัตรหมดอายุ ส่วนสูง น้ำหนัก และกรุ๊ปเลือด เป็นต้น
- เอนทิตีพนักงาน ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสพนักงาน ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ สถานภาพสมรส และเงินเดือน เป็นต้น

- เอนทิตีสินค้า ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสสินค้า ชื่อสินค้า ราคา และจำนวน เป็นต้น
- เอนทิตีนักศึกษา ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสนักศึกษา ชื่อ นามสกุล เพศ วันเดือนปีเกิด ที่อยู่ และเบอร์โทรศัพท์ เป็นต้น
- เอนทิตีวิชา ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสวิชา ชื่อวิชา และจำนวนหน่วยกิต เป็นต้น

ค่าของข้อมูลในแต่ละแอตทริบิวต์ประกอบกัน เรียกว่า **ทูเพิล (tuple)** ซึ่งเป็นแถวของข้อมูลในตาราง โดยแต่ละแถวหรือแต่ละทูเพิลจะประกอบด้วยหลายแอตทริบิวต์หรือหลายคอลัมน์ของข้อมูล จำนวนแถวของข้อมูลในตารางเรียกว่า **Cardinality** และจำนวนแอตทริบิวต์ทั้งหมดในตารางเรียกว่า **Degree** อย่างเช่น จากภาพที่ 6.2 มี 4 Cardinality 5 Degree



ภาพที่ 1.2 ตัวอย่างแอตทริบิวต์ ทูเพิล และเอนทิตีนักศึกษา

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์ **รูปวงรี** แทนหนึ่งแอตทริบิวต์ โดยใช้ชื่อของแอตทริบิวต์นั้นๆ กำกับอยู่ภายใน เช่น



2.3 คีย์

คีย์ (key) คือ แอตทริบิวต์ที่สามารถใช้บ่งบอกความแตกต่างของแต่ละทิวเพิลได้ อาจเป็นแอตทริบิวต์เดี่ยวๆ หรือ กลุ่มของแอตทริบิวต์ก็ได้

ประเภทของคีย์ประกอบด้วย

2.3.1 **ซูเปอร์คีย์ (super key)** คือ แอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ที่สามารถบ่งบอกความแตกต่างของแต่ละทิวเพิลได้

ตารางที่ 1.1 ข้อมูลในเอนทิตีนักศึกษา

รหัสนักศึกษา	ชื่อ	นามสกุล	เลขที่บัตรประชาชน
48001	สามารถ	ประเสริฐกุล	3120100475991
48002	วิชา	ปัญญาเลิศ	3120100475992
48003	น้ำทิพย์	วิเศษศิริ	3120100475993

จากตารางที่ 1.1 ประกอบไปด้วยซูเปอร์คีย์ดังต่อไปนี้

- รหัสนักศึกษา
- รหัสนักศึกษา, ชื่อ
- รหัสนักศึกษา, ชื่อ, นามสกุล
- เลขที่บัตรประชาชน

2.3.2 **คีย์คู่แข่ง (candidate key)** คือ ซูเปอร์คีย์ที่น้อยที่สุด ที่สามารถบ่งบอกความแตกต่างของแต่ละทิวเพิลได้

จากตารางที่ 1.1 ประกอบไปด้วยคีย์คู่แข่งดังต่อไปนี้

- รหัสนักศึกษา
- เลขที่บัตรประชาชน

2.3.3 **คีย์หลัก (primary key)** คือ คีย์คู่แข่งที่ถูกเลือก เพื่อใช้บ่งบอกความแตกต่างของแต่ละทิวเพิล

จากตารางที่ 1.1 คีย์หลัก คือ รหัสนักศึกษา หรือเลขที่บัตรประชาชน อย่างไม่อย่างหนึ่ง

คุณสมบัติของคีย์หลัก

- 1) คีย์หลักซ้ำกันไม่ได้
- 2) คีย์หลักอาจเป็นแค่หนึ่งแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ก็ได้

อย่างเช่น ในตารางที่ 1.1 มีแอตทริบิวต์เดียวที่เป็นคีย์หลัก ซึ่งอาจจะเป็น “รหัสนักศึกษา” หรือ “เลขที่บัตรประชาชน” ก็ได้ แต่ข้อมูลบางตารางอาจต้องอาศัยแอตทริบิวต์ตั้งแต่ 2 ตัวขึ้นไปมาประกอบกันเป็นคีย์หลัก เพื่อให้เกิดความแตกต่างระหว่างทูเพิล ดังเช่นในตารางที่ 1.2

ตารางที่ 1.2 ข้อมูลในเอนทิตีการลงทะเบียนเรียนของนักศึกษา ปีการศึกษา 1/55

รหัสนักศึกษา	ชื่อ	รหัสนวิชา	ชื่อวิชา	ปีการศึกษา
5520249001	ปรีชา	111	คอมพิวเตอร์	1/55
5520249001	ปรีชา	222	ภาษาไทย	1/55
5520249002	เกรียงไกร	111	คอมพิวเตอร์	1/55
5520249003	ฉัตรชัย	333	ภาษาอังกฤษ	1/55
5520249003	ฉัตรชัย	222	ภาษาไทย	1/55
5520249003	ฉัตรชัย	444	สิ่งแวดล้อม	1/55

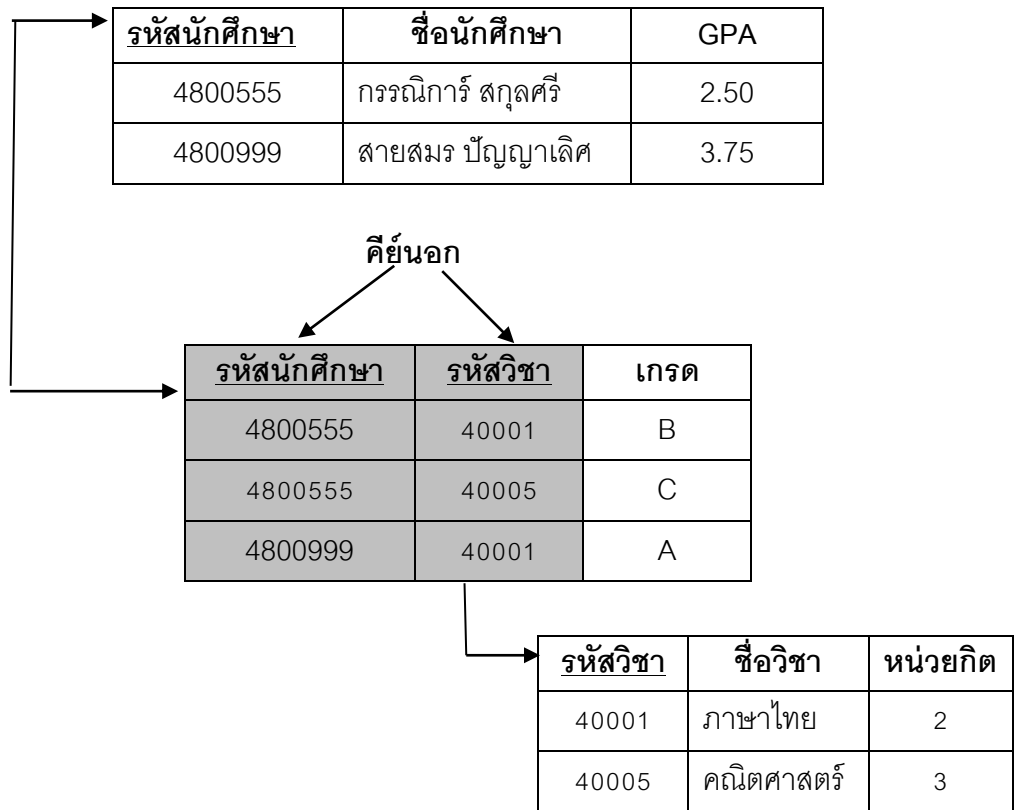
จากตารางที่ 1.2 ไม่สามารถให้แอตทริบิวต์รหัสนักศึกษา เป็นคีย์หลักเพียงแอตทริบิวต์เดียวได้ เพราะจะเห็นว่า รหัสนักศึกษา 5520249001 ของทูเพิลหรือแถวที่ 1 จะไปซ้ำกับแถวที่ 2 แต่ถ้าให้แอตทริบิวต์ “รหัสนักศึกษา” และ “รหัสนวิชา” เป็นคีย์หลัก แล้วพิจารณาข้อมูลของ 2 แอตทริบิวต์นี้ จะเห็นว่าข้อมูลไม่ซ้ำกันแล้ว ดังนั้นตารางที่ 1.2 จึงมีคีย์หลักซึ่งประกอบด้วยแอตทริบิวต์ 2 ตัวประกอบกัน คือ “รหัสนักศึกษา” และ “รหัสนวิชา”

3) คีย์หลักจะเป็นค่าว่าง (null) ไม่ได้ เพราะฉะนั้นในการกรอกข้อมูลต่างๆ ลงในตาราง แอตทริบิวต์ใดที่เรากำหนดให้เป็นคีย์หลักต้องกรอกข้อมูลให้ครบ คือ จะไม่มีค่าไม่ได้ แต่แอตทริบิวต์อื่นอาจจะปล่อยเว้นว่างไว้ก็ได้ถ้าไม่ทราบค่า

2.3.4 คีย์นอก (foreign key) คือ แอตทริบิวต์ที่ใช้ในการเชื่อมต่อกับเอนทิตีอื่นๆ เพื่อแสดงความสัมพันธ์

คุณสมบัติของคีย์นอก คือ

- คีย์นอกสามารถมีค่าซ้ำกันได้
- คีย์นอกสามารถเป็นค่าว่างได้
- คีย์นอกที่ไม่เป็นค่าว่างจะเป็นค่าที่ชี้ไปยังคีย์หลักของเอนทิตีที่สัมพันธ์กัน



ภาพที่ 1.3 แอตทริบิวต์ที่เป็นคีย์นอกที่ใช้ในการเชื่อมต่อกับเอนทิตีอื่น

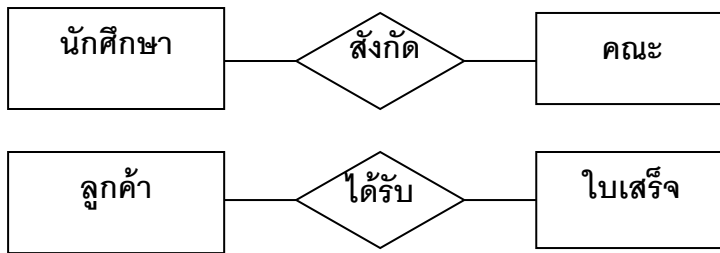
2.3.5 **คีย์รอง (secondary key)** คือ แอตทริบิวต์ที่ไม่เป็น key หลัก แต่สามารถใช้ในการค้นหาข้อมูลนั้นๆ ได้ โดยคีย์รองจะมีค่าซ้ำกันได้ ตัวอย่างเช่น ในตารางที่ 1.3 มีรหัสนักศึกษาเป็นคีย์หลัก แต่หากต้องการค้นหาข้อมูลจากชื่อนักศึกษา แอตทริบิวต์ชื่อก็จะเป็นคีย์รอง หรือถ้าต้องการค้นหาข้อมูลจากนามสกุลนักศึกษา แอตทริบิวต์นามสกุลก็จะเป็นคีย์รอง เป็นต้น

ตารางที่ 3.3 คีย์รองที่ใช้ในการค้นหาข้อมูล

รหัสนักศึกษา	ชื่อ	นามสกุล	เบอร์โทรศัพท์
55111	สามารถ	ประเสริฐกุล	0-1111-1111
55112	วิชา	ปัญญาเลิศ	0-2222-2222
55113	น้ำทิพย์	วิเศษศิริ	0-5555-5555
55114	สมจิตร	สมสกุลวงศ์	0-6666-6666
55115	วิชา	รักศักดิ์ศรี	0-9999-9999

2.4 ความสัมพันธ์

ความสัมพันธ์ (relationship) เป็นการอธิบายความสัมพันธ์ระหว่างเอนทิตีที่มีความสัมพันธ์กัน ว่ามีความสัมพันธ์กันอย่างไร โดยในอี-อาร์ไดอะแกรมใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัด ที่มีชื่อของความสัมพันธ์นั้นกำกับอยู่ภายใน และเชื่อมต่อกับเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์ด้วยเส้นตรง ดังตัวอย่างด้านล่าง

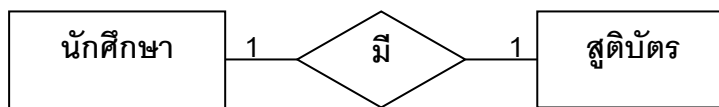


ภาพที่ 1.4 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ระหว่างเอนทิตีแบ่งเป็น 3 ประเภท คือ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง ความสัมพันธ์แบบหนึ่งต่อกลุ่ม และความสัมพันธ์แบบกลุ่มต่อกลุ่ม ดังรายละเอียดต่อไปนี้

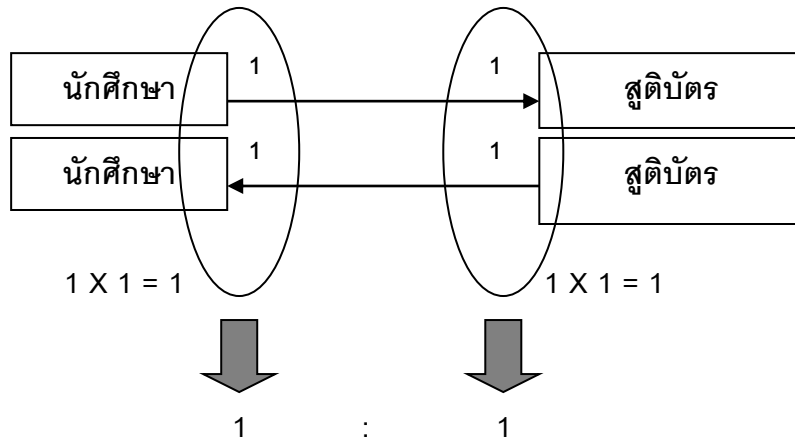
2.4.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one relationship หรือ 1:1)

หมายถึง ข้อมูลในเอนทิตีหนึ่ง มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตีเพียงข้อมูลเดียว ตัวอย่างเช่น นักศึกษาแต่ละคนจะมีสูติบัตรได้เพียงใบเดียวเท่านั้น และสูติบัตรหนึ่งใบก็เป็นของนักศึกษาได้เพียงคนเดียวเท่านั้นเช่นกัน



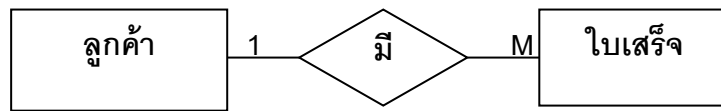
ภาพที่ 1.5 ความสัมพันธ์ของข้อมูลแบบหนึ่งต่อหนึ่ง

ในการพิจารณาความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง ต้องมองสองทิศ คือ มองจากซ้ายไปขวา และก็ต้องมองจากขวาไปซ้าย แล้วจึงนำความสัมพันธ์ทั้งสองทิศมาพิจารณารวมกัน ดังภาพที่ 1.6



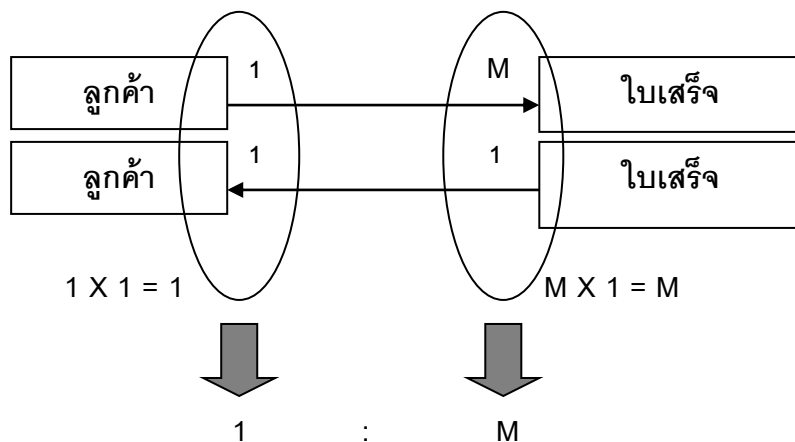
ภาพที่ 1.6 วิธีการพิจารณาความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2.4.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many relationship หรือ 1:M) หมายถึง ข้อมูลในเอนทิตีหนึ่ง มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตีมากกว่าหนึ่งข้อมูล ตัวอย่างเช่น ลูกค้าหนึ่งคนมีใบเสร็จได้หลายใบ เนื่องจากลูกค้าหนึ่งคนอาจมาซื้อสินค้าหลายครั้ง แต่ใบเสร็จหนึ่งใบต้องเป็นของลูกค้าเพียงคนเดียวเท่านั้น



ภาพที่ 1.7 ความสัมพันธ์ของข้อมูลแบบหนึ่งต่อกลุ่ม

ในการพิจารณาความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่ม ต้องมองสองทิศ คือ มองจากซ้ายไปขวา และก็ต้องมองจากขวาไปซ้าย แล้วจึงนำความสัมพันธ์ทั้งสองทิศ มาพิจารณารวมกัน ดังภาพที่ 1.8



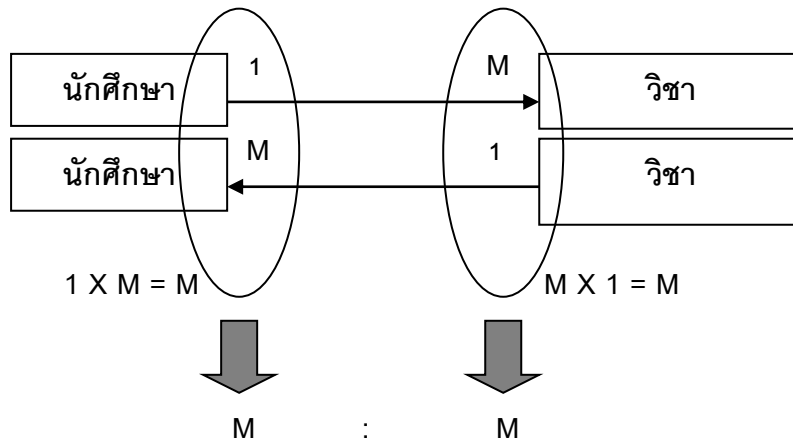
ภาพที่ 1.8 วิธีการพิจารณาความสัมพันธ์แบบหนึ่งต่อกลุ่ม

2.4.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many relationship หรือ M:M) หมายถึง ข้อมูลมากกว่าหนึ่งข้อมูลในเอนทิตีหนึ่ง มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตีมากกว่าหนึ่งข้อมูล ตัวอย่างเช่น นักศึกษาหนึ่งคนสามารถลงทะเบียนเรียนได้หลายวิชา และวิชาแต่ละวิชามีนักศึกษาลงทะเบียนเรียนได้หลายคน



ภาพที่ 1.9 ความสัมพันธ์ของข้อมูลแบบกลุ่มต่อกลุ่ม


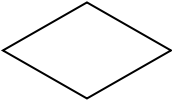
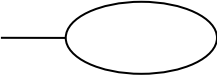

ในการพิจารณาความสัมพันธ์ระหว่างเอนทิตีแบบกลุ่มต่อกลุ่ม ต้องมองสองทิศ คือ มองจากซ้ายไปขวา และก็ต้องมองจากขวาไปซ้าย แล้วจึงนำความสัมพันธ์ทั้งสองทิศมาพิจารณารวมกัน ดังภาพที่ 1.10



ภาพที่ 1.10 วิธีการพิจารณาความสัมพันธ์แบบกลุ่มต่อกลุ่ม

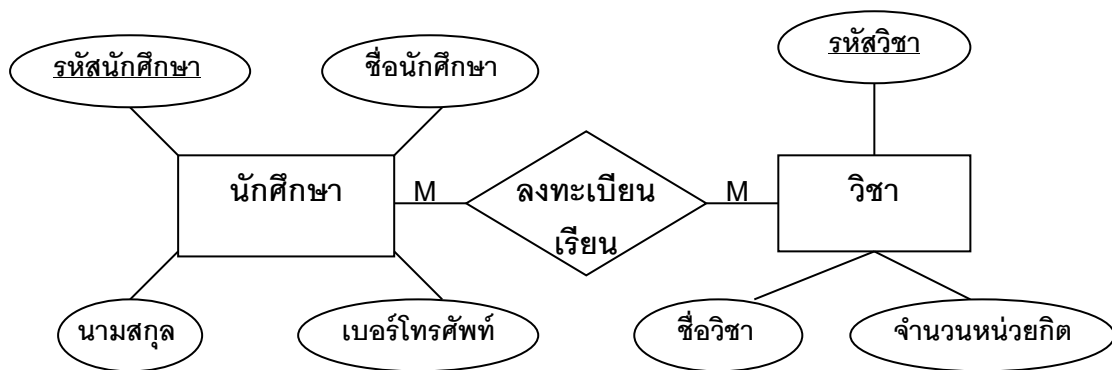
3. สัญลักษณ์ในแบบจำลองอี-อาร์

ตารางที่ 1.4 สัญลักษณ์ที่สำคัญๆ ในแบบจำลองอี-อาร์

สัญลักษณ์	ความหมาย
	เอนทิตี
	ความสัมพันธ์
	แอตทริบิวต์
	คีย์หลัก

ตัวอย่าง

นักศึกษา และ วิชา เป็นเอนทิตีที่เราสนใจจะจัดเก็บ ซึ่งเอนทิตีนักศึกษาคจะประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสนักศึกษา ชื่อนักศึกษา นามสกุล และเบอร์โทรศัพท์ เป็นต้น โดยมีรหัสนักศึกษาเป็นคีย์หลัก ส่วนเอนทิตีวิชาจะประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสวิชา ชื่อวิชา และจำนวนหน่วยกิต เป็นต้น โดยมีรหัสวิชาเป็นคีย์หลัก ซึ่งความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีวิชา เป็นแบบกลุ่มต่อกลุ่ม คือ นักศึกษาหนึ่งคนสามารถลงทะเบียนเรียนได้หลายวิชา และวิชาแต่ละวิชามีนักศึกษาลงทะเบียนเรียนได้หลายคน ดังนั้นเราสามารถนำเสนอในรูปแบบของแผนภาพ ที่เรียกว่า อี-อาร์ไดอะแกรม (E-R Diagram) ด้วยการใช้องค์ประกอบต่างๆ ดังนี้



ภาพที่ 1.11 ตัวอย่างของ อี-อาร์ไดอะแกรม (E-R Diagram)

4. การแปลงแบบจำลองอี-อาร์เป็นโครงสร้างตารางฐานข้อมูล

ขั้นตอนในการแปลงแบบจำลองอี-อาร์เป็นโครงสร้างของตารางในฐานข้อมูล มีขั้นตอนดังต่อไปนี้

4.1 แปลงเอนทิตีปกติในแบบจำลองอี-อาร์เป็น 1 ตาราง ซึ่งประกอบด้วยแอตทริบิวต์ของเอนทิตีนั้นๆ โดยชื่อของตารางก็คือชื่อของเอนทิตี และแอตทริบิวต์ของเอนทิตี ก็คือแอตทริบิวต์ของตาราง สำหรับแอตทริบิวต์ที่เป็นคีย์หลักของตาราง ให้ขีดเส้นใต้ที่แอตทริบิวต์นั้น เช่นเดียวกับในแบบจำลองอี-อาร์ ซึ่งจากภาพที่ 1.11 นำมาแปลงเป็นตารางได้ 2 ตาราง คือ

นักศึกษา			
รหัสนักศึกษา	ชื่อ	นามสกุล	เบอร์โทรศัพท์
55111	สามารถ	ประเสริฐกุล	0-1111-1111
55112	วิชา	ปัญญาเลิศ	0-2222-2222

วิชา		
รหัสวิชา	ชื่อวิชา	หน่วยกิต
40001	ภาษาไทย	2
40005	คณิตศาสตร์	3

ภาพที่ 1.12 โครงสร้างของตารางในฐานข้อมูลจากการแปลงเอนทิตีปกติในแบบจำลองอี-อาร์

4.2 แปลงความสัมพันธ์เป็นตาราง

4.2.1 แปลงความสัมพันธ์แบบ 1 : M นั้นไม่ต้องสร้างตารางใหม่ แต่ให้นำแอตทริบิวต์ ที่เป็นคีย์หลักของเอนทิตีที่อยู่ด้านความสัมพันธ์ที่เป็น 1 ไปเพิ่มเป็นแอตทริบิวต์ของตารางด้านที่มีความสัมพันธ์เป็น M

4.2.2 แปลงความสัมพันธ์แบบ M : M จะได้ตารางใหม่ 1 ตาราง ซึ่งประกอบด้วยแอตทริบิวต์ของความสัมพันธ์นั้นรวมกับแอตทริบิวต์ที่เป็นคีย์หลักของ 2 เอนทิตีที่มีความสัมพันธ์ แบบ M : M

จากแบบจำลองอี-อาร์ในภาพที่ 1.11 สามารถสร้างตารางตามขั้นตอนนี้ได้อีก 1 ตาราง คือ ตารางการลงทะเบียน ซึ่งประกอบด้วยแอตทริบิวต์ **รหัสนักศึกษา** (คีย์หลักของเอนทิตีนักศึกษา) และ **รหัสวิชา** (คีย์หลักของเอนทิตีวิชา) ฉะนั้นตารางใหม่ที่เกิดขึ้น ซึ่งก็คือ ตารางการลงทะเบียน มี รหัสนักศึกษาและรหัสวิชา เป็นคีย์หลัก ดังนี้

การลงทะเบียน	
รหัสนักศึกษา	รหัสวิชา
48111	40001
48111	40005
48112	40001

ภาพที่ 1.13 โครงสร้างของตารางที่ได้จากการแปลงความสัมพันธ์แบบ M:M

จากการแปลงแบบจำลองอี-อาร์ตามขั้นตอนข้างต้น สรุปตารางที่ได้ทั้งหมด 3 ตาราง ดังต่อไปนี้

นักศึกษา			
รหัสนักศึกษา	ชื่อ	นามสกุล	เบอร์โทรศัพท์
48111	สามารถ	ประเสริฐกุล	0-1111-1111
48112	วิชา	ปัญญาเลิศ	0-2222-2222

วิชา		
รหัสวิชา	ชื่อวิชา	หน่วยกิต
40001	ภาษาไทย	2
40005	คณิตศาสตร์	3

การลงทะเบียน	
รหัสนักศึกษา	รหัสวิชา
48111	40001
48111	40005
48112	40001

ภาพที่ 1.14 โครงสร้างฐานข้อมูลการลงทะเบียนเรียนของนักศึกษา

ทั้งนี้โครงสร้างฐานข้อมูลที่ได้จากการแปลงแบบจำลองอี-อาร์นั้นจะอยู่ใน 1NF ดังนั้นจึงจำเป็นต้องนำมาทำนอร์มัลไลเซชันต่อ เพื่อให้ได้ฐานข้อมูลที่ปราศจากความซ้ำซ้อนหรือซ้ำซ้อนน้อยที่สุด แต่ถ้าได้ทำการออกแบบฐานข้อมูลโดยใช้แบบจำลองอี-อาร์ มาอย่างถูกต้องแล้ว เมื่อแปลงเป็นโครงสร้างฐานข้อมูลแบบสัมพันธ์ จะได้โครงสร้างความสัมพันธ์ที่จัดกลุ่มของแอตทริบิวต์มาเป็นอย่างดี และบางทีโครงสร้างของความสัมพันธ์ที่ได้นั้นอาจอยู่ในนอร์มัลฟอร์มที่สูงกว่านอร์มัลฟอร์มที่ 1 แล้ว อย่างไรก็ตามขั้นตอนถัดมาจำเป็นต้องวิเคราะห์ความสัมพันธ์ระหว่างแอตทริบิวต์ ซึ่งก็คือ วิธีนอร์มัลไลเซชันที่จะกล่าวต่อไป

นอร์มัลไลเซชัน (Normalization)

การออกแบบฐานข้อมูลที่ดี ต้องมีความซ้ำซ้อนในการจัดเก็บข้อมูลน้อยที่สุด หรือไม่มี ความซ้ำซ้อนเลย ซึ่งต้องอาศัยหลักการในการทำนอร์มัลไลเซชันดังรายละเอียดต่อไปนี้

1. แนวคิดเกี่ยวกับนอร์มัลไลเซชัน

นอร์มัลไลเซชันเป็นวิธีการที่ใช้ในการตรวจสอบและแก้ไขปัญหาเกี่ยวกับความซ้ำซ้อนของข้อมูล โดยดำเนินการให้ข้อมูล ในแต่ละรีเลชัน (relation) อยู่ในรูปที่เป็นหน่วยที่เล็กที่สุดที่ไม่สามารถแตกออกเป็นหน่วยย่อยๆ ได้อีก โดยยังคงความสัมพันธ์ระหว่างข้อมูลในรีเลชันต่างๆ ไว้ตามหลักการที่กำหนดไว้ใน relational model การทำนอร์มัลไลเซชันนี้ เป็นการดำเนินการอย่างเป็นลำดับ ที่กำหนดไว้ด้วยกันเป็นขั้นตอน ตามปัญหาที่เกิดขึ้นในขั้นต่อนั้นๆ ซึ่งแต่ละขั้นตอนจะมีชื่อตามโครงสร้างข้อมูลที่กำหนดไว้ดังนี้ 1. First Normal Form (1NF) 2. Second Normal Form (2NF) 3. Third Normal Form (3NF) 4. Boyce-Codd Normal Form (BCNF) 5. Fourth Normal Form (4NF) และ 6. Fifth Normal Form (5NF) ในการออกแบบฐานข้อมูลเพื่อลดความซ้ำซ้อนในการจัดเก็บข้อมูลอย่างน้อยต้องมีคุณสมบัติเป็น 3 NF เพราะจริงๆ แล้ว ในการทำงานทั่วไป แค่ 3 NF ก็สามารถใช้งานได้แล้ว แต่สำหรับ BCNF ไปจนถึง 5NF เป็นฐานข้อมูลชนิดพิเศษจริงๆ ที่แทบจะไม่มีในชีวิตประจำวัน โอกาสพบประมาณ 0.01 % ดังนั้นในที่นี้จะศึกษาเพียงแค่ 1NF 2NF และ 3NF เท่านั้น เพื่อใช้เป็นแนวทางในการศึกษาระดับอื่นต่อไป

2. รูปแบบของนอร์มัลฟอร์ม (Normal Form : NF)

2.1 First Normal Form (1NF)

ตารางที่ผ่านการทำนอร์มัลไลเซชันระดับที่ 1 หรือ First Normal Form ต้องมีคุณสมบัติดังนี้

ไม่มีคอลลัมน์ใดในตารางที่มีค่ามากกว่า 1 ค่า คือ ค่าในแต่ละคอลลัมน์ต้องเป็น atomic หรือไม่อยู่ในรูปของ repeating group

หมายความว่าข้อมูลที่เก็บในแต่ละคอลลัมน์จะต้องมีลักษณะเป็นค่าเดียว (single valued) ไม่สามารถแบ่งย่อยได้อีก

ในการทำนอร์มัลไลเซชันจะต้องดูข้อมูลในตารางเป็นหลัก ตัวอย่างเช่น ข้อมูลในตารางที่ 1.5 แสดงการเก็บข้อมูลเกี่ยวกับนักศึกษา ซึ่งแต่ละคนสามารถอยู่ชมรมและมีงานอดิเรกได้มากกว่า 1 อย่าง

ตารางที่ 1.5 ข้อมูลนักศึกษา

รหัสนักศึกษา	ชื่อ	ที่อยู่	ชมรม	งานอดิเรก
55001	นารี ศิริพร	กรุงเทพ	ดนตรี อาสาพัฒนาชนบท อนุรักษ์สิ่งแวดล้อม	เล่นกีตาร์
55002	ศรีสมร อมรชัย	นนทบุรี	พระพุทธศาสนา	สะสมพระเครื่อง ร้องเพลง
55003	อรอนงค์ สมประสงค์	กรุงเทพ	ดนตรี	ตีกลอง

จากตารางที่ 1.5 จะเห็นว่าข้อมูลในคอลลัมน์ชมรมและงานอดิเรกมีค่ามากกว่า 1 ค่า แสดงว่าไม่เป็น atomic หรืออยู่ในรูปของ repeating group ดังนั้นตารางที่ 1.5 จึงไม่เป็น 1NF โดยเราจะเรียกตารางที่ยังไม่ผ่านแม้แต่ 1NF ว่า Unnormalized Form (UNF) ซึ่งมีวิธีการที่จะทำให้เป็น 1NF คือ

- 1) แยกคอลลัมน์ที่มีค่ามากกว่า 1 ค่าออกเป็นแถวใหม่
- 2) เพิ่มข้อมูลที่เหมาะสมเข้าไปในคอลลัมน์ที่ว่างอยู่ของแถวที่เกิดขึ้นใหม่

จากตารางที่ 1.5 ที่ไม่มีคุณสมบัติเป็น 1NF สามารถทำให้มีคุณสมบัติเป็น 1NF ได้ดังตารางที่ 1.6 ซึ่งมีรหัสนักศึกษา ชมรม และงานอดิเรก เป็นคีย์หลัก

ตารางที่ 1.6 ข้อมูลนักศึกษาที่ผ่านการทำนอร์มัลไลเซชัน ระดับที่ 1 แล้ว

รหัสนักศึกษา	ชื่อ	ที่อยู่	ชมรม	งานอดิเรก
55001	นารี ศิริพร	กรุงเทพ	ดนตรี	เล่นกีตาร์
55001	นารี ศิริพร	กรุงเทพ	อาสาพัฒนาชนบท	เล่นกีตาร์
55001	นารี ศิริพร	กรุงเทพ	อนุรักษ์สิ่งแวดล้อม	เล่นกีตาร์
55002	ศรีสมร อมรชัย	นนทบุรี	พระพุทธศาสนา	สะสมพระเครื่อง
55002	ศรีสมร อมรชัย	นนทบุรี	พระพุทธศาสนา	ร้องเพลง
55003	อรอนงค์ สมประสงค์	กรุงเทพ	ดนตรี	ตีกลอง

สาเหตุที่แยกคอลลัมน์ที่มีค่ามากกว่า 1 ค่าออกเป็นแถวใหม่ เนื่องจากไม่รู้จำนวนที่แน่นอนของค่าที่มีอยู่ในคอลลัมน์นั้น เช่น ไม่ทราบว่านักศึกษาแต่ละคนจะมีงานอดิเรกกันคนละไม่เกินไปอย่าง แต่ถ้าเราทราบจำนวนที่แน่นอนของคอลลัมน์ที่มีหลายค่า นั้น เราอาจแยกเป็นคอลลัมน์ใหม่ได้เลย ตัวอย่างเช่น การเก็บชื่อผู้แต่งของหนังสือในห้องสมุด ซึ่งหนังสือเล่มหนึ่งอาจจะมีผู้แต่งหลายคน แต่ในการเก็บชื่อผู้แต่งจะเก็บเพียง 3 คนเท่านั้น ในกรณีนี้ควรแบ่งคอลลัมน์ซึ่งเก็บชื่อผู้แต่งออกเป็นหลายคอลลัมน์โดยขึ้นกับจำนวนผู้แต่งที่มากที่สุดที่มีอยู่หรือเราต้องการเก็บข้อมูลเอาไว้ ซึ่งจะทำให้ค่าของแต่ละคอลลัมน์ เป็น Atomic ดังตารางต่อไปนี้

ตารางที่ 1.7 การเก็บข้อมูลหนังสือ ที่มีคุณสมบัติเป็น 1NF

ISBN	ชื่อหนังสือ	ผู้แต่ง 1	ผู้แต่ง 2	ผู้แต่ง 3
9749151001	การจัดการฐานข้อมูล	วาสนา ทรัพย์แก้ว		
9749151002	ระบบสารสนเทศ	วิเชียร เขียวชัย	นฤมล สมสกุล	
9749151003	เทคโนโลยีสารสนเทศ	สุชุม เฉลยทรัพย์	ปริศนา มัชฌิมา	กาญจนา เผือกคง

ถึงแม้ว่าตารางที่ 1.6 จะได้รับการออกแบบให้อยู่ในรูปแบบ 1NF แล้ว แต่ลักษณะของข้อมูลภายในอาจก่อให้เกิดปัญหาขึ้นได้อีก เช่น ข้อมูลที่เกี่ยวกับนักศึกษารหัส 55001 ถูกจัดเก็บไว้ในแถวที่ 1, 2 และ 3 ได้แก่ ชื่อ และที่อยู่ โดยจะเห็นว่าเป็นการเก็บข้อมูลที่ซ้ำซ้อนกัน ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ และก่อให้เกิดปัญหาในการเปลี่ยนแปลงข้อมูลด้วย เช่น ถ้านักศึกษารหัส 55001 มีการเปลี่ยนชื่อ หรือที่อยู่ ก็ต้องทำการแก้ไขข้อมูลหลายแถว ซึ่งถ้ามีการแก้ไขข้อมูลไม่ครบ ก็อาจทำให้ข้อมูลภายในตารางเกิดความขัดแย้งกันได้ ดังนั้นจึงต้องมีการนอร์มัลไลเซชันระดับที่ 2 ต่อไป

2.2 Second Normal Form (2NF)

ตารางที่ผ่านการทำนอร์มัลไลเซชันระดับที่ 2 หรือ Second Normal Form ต้องมีคุณสมบัติดังนี้

- 1) ต้องมีคุณสมบัติของ 1NF
- 2) ทุก nonprime attribute จะต้องขึ้นกับ prime (primary key) ทุกตัว

นั่นคือแอตทริบิวต์ที่ไม่ใช่คีย์หลักจะต้องมีค่าขึ้นอยู่กับคีย์หลักเท่านั้น โดยถ้าคีย์หลักประกอบด้วยแอตทริบิวต์ที่มากกว่า 1 ตัวก็จะต้องขึ้นอยู่กับแอตทริบิวต์ทั้งหมดที่เป็นคีย์หลัก ไม่ใช่ขึ้นอยู่กับบางตัว

การที่จะรู้ว่าแอตทริบิวต์ใดขึ้นอยู่กับแอตทริบิวต์ใดนั้น ต้องใช้ความรู้ในเรื่องฟังก์ชันการขึ้นต่อกัน หรือ functional dependency ซึ่งเป็นสิ่งที่ใช้แสดงความสัมพันธ์ระหว่างแอตทริบิวต์ ในรูปแบบฟังก์ชัน เพื่อช่วยในการตัดสินใจว่าแอตทริบิวต์ที่ไม่ใช่คีย์หลักควรจะปรากฏเป็นคอลัมน์อยู่ในตารางหรือควรจะแยกออกมาสร้างเป็นตารางใหม่

สมมติว่า X และ Y เป็นแอตทริบิวต์ในตารางหนึ่ง ถ้า Y ขึ้นอยู่กับ X จะสามารถเขียนฟังก์ชันการขึ้นต่อกันได้ดังนี้

$$X \rightarrow Y$$

การที่ Y ขึ้นอยู่กับ X หมายความว่า ทุกๆค่าของ X ที่เราเลือกขึ้นมา จะสามารถหาค่าของ Y มา 1 ค่าที่สอดคล้องกับค่าของ X ได้เสมอ เช่น จากตารางที่ 1.8 เป็นตารางที่เก็บข้อมูลเกี่ยวกับนักศึกษา ถ้าถามว่านักศึกษาคอนโดที่มีรหัสนักศึกษาเป็น 55111 ก็สามารถตอบได้ทันทีก็คือ นักศึกษาที่ชื่อว่า สามารถ ประเสริฐกุล ดังนั้นชื่อนักศึกษาจึงขึ้นอยู่กับรหัสนักศึกษา ซึ่งเขียนเป็นฟังก์ชันการขึ้นต่อกันได้ว่า **รหัสนักศึกษา \rightarrow ชื่อ** นั่นเอง

ตารางที่ 1.8 ตารางนักศึกษา

รหัสนักศึกษา (คีย์หลัก)	ชื่อ	นามสกุล	เบอร์โทรศัพท์
55111	สามารถ	ประเสริฐกุล	0-1111-1111
55112	วิชา	ปัญญาเลิศ	0-2222-2222
55113	น้ำทิพย์	ปัญญาเลิศ	0-2222-2222
55114	สมจิตร	สมสกุลวงศ์	0-6666-6666
55115	วิชา	รักศักดิ์ศรี	0-9999-9999

เมื่อเข้าใจในเรื่องฟังก์ชันการขึ้นต่อกันแล้ว เราลองมาพิจารณาว่าตารางที่ 1.9 จะมีคุณสมบัติเป็น 2NF หรือไม่

ตารางที่ 1.9 ข้อมูลการสั่งซื้อสินค้าของลูกค้า

รหัสลูกค้า	ชื่อลูกค้า	ระดับ	ประเภท	รหัสสินค้า	ชื่อสินค้า	จำนวนสินค้า
001	นารี	A	ชั้นดี	P111	ปากกา	10
001	นารี	A	ชั้นดี	P222	ดินสอ	12
001	นารี	A	ชั้นดี	P333	ยางลบ	10
002	ศรีสมร	B	ปานกลาง	P222	ดินสอ	15
003	อรอนงค์	C	พอใช้	P333	ยางลบ	16

ก่อนอื่นเราต้องพิจารณาว่าตารางที่ 1.9 มีคุณสมบัติเป็น 1NF หรือไม่ จากข้อมูลในตารางจะเห็นว่าไม่มีคอลัมน์ใดในตารางที่มีค่ามากกว่า 1 ค่า แสดงว่าผ่านคุณสมบัติเป็น 1NF จากนั้นต้องพิจารณาต่อว่ามีแอตทริบิวต์ใดเป็น prime หรือคีย์หลัก ส่วนที่เหลือก็จะเรียกว่า nonprime จากตารางที่ 1.9 จะมีแอตทริบิวต์ รหัสลูกค้า และ รหัสสินค้า เป็น prime ส่วน ชื่อลูกค้า ที่อยู่ ชื่อสินค้า และ จำนวนสินค้า เป็น nonprime

จากคุณสมบัติของ 2NF คือ nonprime ต้องขึ้นกับ prime ทุกตัว ในการพิจารณาว่าเป็น 2NF หรือไม่ จะต้องพิจารณา nonprime ทีละตัว ซึ่งมีผลสรุปการขึ้นต่อกันดังนี้

รหัสลูกค้า, ~~รหัสสินค้า~~ → ชื่อลูกค้า, ระดับ, ประเภท

จากข้อมูลในตารางที่ 1.9 จะสังเกตว่าชื่อลูกค้า, ระดับ และประเภทจะขึ้นกับรหัสลูกค้าเพียงอย่างเดียว ไม่ขึ้นกับรหัสสินค้าเลย ทำให้ไม่เป็นไปตามคุณสมบัติของ 2NF และนอกจากนี้ยังมีกรณีอื่นอีกที่ทำให้ตารางที่ 1.9 ไม่เป็นไปตามคุณสมบัติของ 2NF ได้แก่

~~รหัสลูกค้า~~, รหัสสินค้า → ชื่อสินค้า

จากข้อมูลในตารางที่ 1.9 จะสังเกตว่าชื่อสินค้าจะขึ้นกับรหัสสินค้าเพียงอย่างเดียว ไม่ขึ้นกับรหัสลูกค้าเลย ทำให้ไม่เป็นไปตามคุณสมบัติของ 2NF มีเพียงจำนวนสินค้าอย่างเดียวที่ขึ้นกับรหัสลูกค้า และรหัสสินค้า สรุปว่า

รหัสลูกค้า,รหัสสินค้า → จำนวนสินค้า

รหัสลูกค้า → ชื่อลูกค้า, ระดับ, ประเภท

รหัสสินค้า → ชื่อสินค้า

ดังนั้นถ้าต้องการให้ตารางที่ 1.9 มีคุณสมบัติเป็น 2NF จะต้องทำการแตกตารางออกมา ตามความสัมพันธ์ของฟังก์ชันการขึ้นต่อกัน เป็น 3 ตาราง ดังนี้

ตารางที่ 1.10 ข้อมูลลูกค้า

รหัสลูกค้า	ชื่อลูกค้า	ระดับ	ประเภท
001	নারী	A	ชั้นดี
002	ศรีสมร	B	ปานกลาง
003	อรอนงค์	C	พอใช้

ตารางที่ 1.11 ข้อมูลสินค้า

รหัสสินค้า	ชื่อสินค้า
P111	ปากกา
P222	ดินสอ
P333	ยางลบ

ตารางที่ 1.12 ข้อมูลการสั่งซื้อสินค้า

รหัสลูกค้า	รหัสสินค้า	จำนวนสินค้า
001	P111	10
001	P222	12
001	P333	10
002	P222	15
003	P333	16

จากตารางที่ 1.10 หากต้องการเพิ่มข้อมูลลูกค้าขึ้นมาอีก 1 คน อาจมีปัญหาเกิดขึ้น คือ ถ้าลูกค้าที่จะเพิ่มเข้าไปเป็นลูกค้าในระดับ A แสดงว่าต้องเป็นลูกค้าชั้นดี ในชื่อประเภทต้องใส่ว่า “ชั้นดี” เท่านั้น ถ้าหากใส่ว่า ปานกลาง หรือพอใช้ ข้อมูลก็จะขัดแย้งกัน เพราะฉะนั้นตารางที่ผ่าน 2NF บางตารางอาจเกิดปัญหาในเรื่องของการเพิ่มข้อมูลได้ ดังนั้นจึงต้องมีการนอร์มัลไลเซชันระดับที่ 3 ต่อไป

2.3 Third Normal Form (3NF)

ตารางที่ผ่านการทำนอร์มัลไลเซชันระดับที่ 3 หรือ Third Normal Form ต้องมีคุณสมบัติดังนี้

- 1) ต้องมีคุณสมบัติของ 2NF
- 2) nonprime ต้องไม่ขึ้นกับ nonprime

ตารางที่ 1.13 ข้อมูลใบเสร็จรับเงินจากลูกค้า

เลขที่ใบเสร็จ	รหัสลูกค้า	ชื่อลูกค้า	ที่อยู่
1001	43	กฤษณา	121 อ.เมือง จ.นนทบุรี
1002	55	ศักดิ์สิทธิ์	222 อ.ปากเกร็ด จ.นนทบุรี
1003	43	กฤษณา	121 อ.เมือง จ.นนทบุรี

จากตารางที่ 1.13 จะมีแอดทริบิวต์ เลขที่ใบเสร็จ เป็น prime ส่วน รหัสลูกค้า ชื่อลูกค้า และที่อยู่ เป็น nonprime นักศึกษาต้องพิจารณาก่อนว่าตารางที่ 1.13 มีคุณสมบัติเป็น 2NF หรือไม่ ซึ่งจากข้อมูลจะสังเกตว่ามีคุณสมบัติเป็น 2NF เนื่องจากไม่อยู่ในรูป repeating group และ nonprime ทุกตัวขึ้นกับ prime ทุกตัว คือ รหัสลูกค้า ชื่อลูกค้า และ ที่อยู่ ขึ้นกับ เลขที่ใบเสร็จ ซึ่งเป็น prime เพียงตัวเดียว

เลขที่ใบเสร็จ → รหัสลูกค้า, ชื่อลูกค้า, ที่อยู่

เมื่อมีคุณสมบัติเป็น 2NF แล้ว ก็พิจารณาต่อว่า nonprime ขึ้นกับ nonprime หรือไม่ ถ้าไม่มี nonprime ตัวใดขึ้นต่อกัน ก็แสดงว่ามีคุณสมบัติเป็น 3NF แต่จากข้อมูลในตารางที่ 1.13 มี nonprime บางตัวที่ขึ้นต่อกัน ได้แก่ ชื่อลูกค้าและที่อยู่ ขึ้นกับรหัสลูกค้า ทำให้ไม่เป็นไปตามคุณสมบัติของ 2NF

รหัสลูกค้า → ชื่อลูกค้า, ที่อยู่

ดังนั้นจึงต้องมีการแตกตารางที่ 1.13 ออกมาเป็น 2 ตาราง ตามความสัมพันธ์ของฟังก์ชันการขึ้นต่อกัน ดังนี้

ตารางที่ 1.14 ข้อมูลลูกค้าจากตารางที่ 1.13

รหัสลูกค้า	ชื่อลูกค้า	ที่อยู่
43	กฤษณา	121 อ.เมือง จ.นนทบุรี
55	ศักดิ์สิทธิ์	222 อ.ปากเกร็ด จ.นนทบุรี

ตารางที่ 1.15 ข้อมูลใบเสร็จ

เลขที่ใบเสร็จ	รหัสลูกค้า
1001	43
1002	55
1003	43

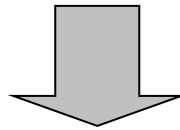
อีกตัวอย่างหนึ่ง จากตารางที่ 1.10 ซึ่งผ่านการนอร์มัลไลเซชันระดับที่ 2 แล้ว แต่ยังมีปัญหาการเพิ่มข้อมูลอยู่ ดังที่กล่าวมาแล้วข้างต้น จึงต้องมีการทำให้เป็น 3NF เนื่องจาก nonprime บางตัวขึ้นกับ nonprime คือ

ระดับ	→	ประเภท
-------	---	--------

 ดังนั้นจึงต้องแยก เป็น 2 ตาราง ดังต่อไปนี้

ตารางที่ 1.10 ข้อมูลลูกค้า

รหัสลูกค้า	ชื่อลูกค้า	ระดับ	ประเภท
001	นารี	A	ชั้นดี
002	ศรีสมร	B	ปานกลาง
003	อรอนงค์	C	พอใช้



ตารางที่ 1.16 ข้อมูลลูกค้าจากตารางที่ 1.10

รหัสลูกค้า	ชื่อลูกค้า	ระดับ
001	นารี	A
002	ศรีสมร	B
003	อรอนงค์	C

ตารางที่ 1.17 ข้อมูลระดับลูกค้า

ระดับ	ประเภท
A	ชั้นดี
B	ปานกลาง
C	พอใช้

บทที่ 2

การใช้งานโปรแกรม Microsoft Access

เมื่อทำการออกแบบฐานข้อมูลด้วยแผนภาพที่เรียกว่า แบบจำลองอี-อาร์ และทำการนอร์มัลไลเซชันแล้ว จึงนำฐานข้อมูลที่ออกแบบไว้นั้นมาสร้างให้ใช้งานได้จริงด้วยโปรแกรมสำเร็จรูปในการจัดการฐานข้อมูล ซึ่งก็คือ โปรแกรม Microsoft Access ที่เลือกโปรแกรมนี้ เนื่องจากเป็นโปรแกรมที่สามารถฝึกใช้งานได้ง่าย และเครื่องคอมพิวเตอร์ส่วนใหญ่จะมีโปรแกรมนี้ อยู่แล้ว เพราะเป็นโปรแกรมหนึ่งในชุด Microsoft Office ซึ่งสามารถศึกษาการใช้งานเบื้องต้นได้ด้วยตนเอง ตั้งแต่การสร้างตาราง ฟอรั่ม แบบสอบถาม และรายงาน เป็นต้น

โครงสร้างของโปรแกรม Microsoft Access

โปรแกรม Microsoft Access เป็นโปรแกรมที่ใช้ในการจัดการฐานข้อมูลอย่างหนึ่ง เริ่มตั้งแต่จัดเก็บข้อมูล ประมวลผล และออกสารสนเทศ ที่สามารถนำไปใช้ประโยชน์ได้อย่างมีประสิทธิภาพ

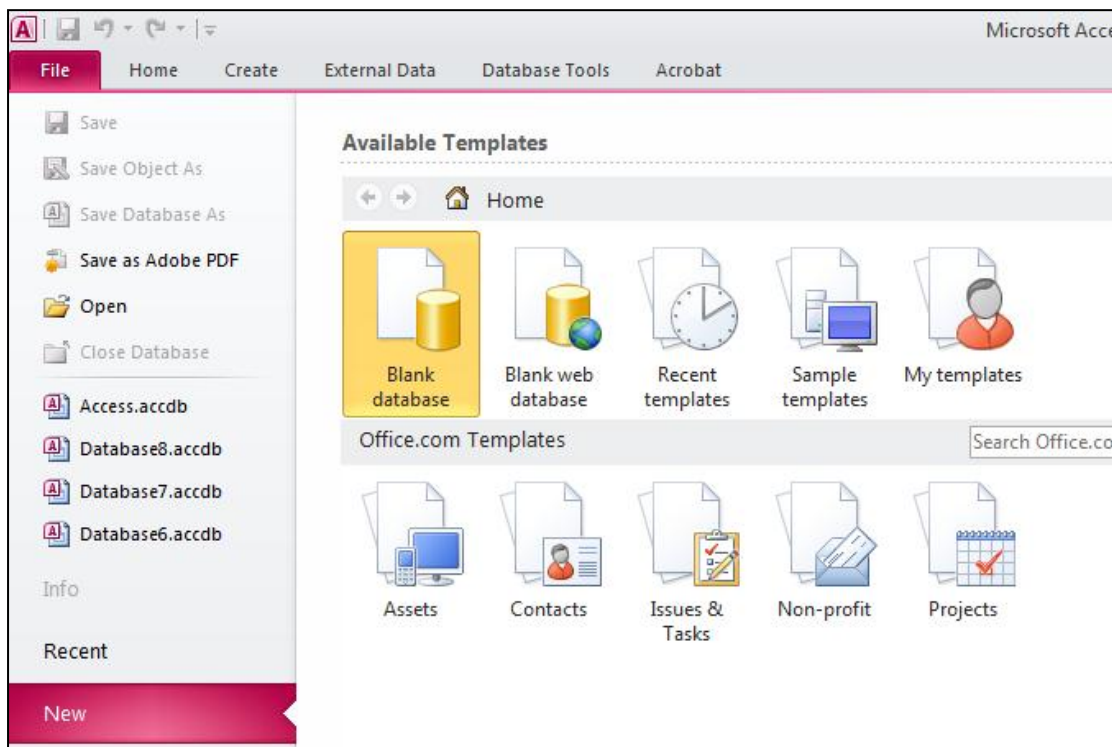
โครงสร้างของโปรแกรม Microsoft Access ประกอบด้วย

1. ตาราง (table) ใช้สำหรับเก็บข้อมูลต่างๆ ซึ่งแต่ละตารางจะประกอบไปด้วยเขตข้อมูล (Field) และระเบียน (Record)
2. แบบสอบถาม (queries) ใช้สำหรับดูข้อมูล ค้นหาข้อมูล ตรวจสอบแก้ไข เพิ่มเติมหรือลบข้อมูลที่ต้องการ
3. ฟอรั่ม (form) ออกแบบมาเพื่อเป็นส่วนติดต่อกับผู้ใช้สำหรับป้อนข้อมูลและแสดงข้อมูล
4. รายงาน (report) ออกแบบมาให้มีการจัดรูปแบบ มีการคำนวณและพิมพ์ออกมาเป็นรายงานได้ รวมทั้งมีการสรุปผลของข้อมูล queiry ที่เลือก ซึ่งสามารถดูรายงานก่อนพิมพ์ได้
5. มาโคร (macro) เป็นภาษาโปรแกรมง่ายๆ เพื่อการกำหนดโครงสร้างลำดับขั้นตอนให้ปฏิบัติงานตามที่ต้องการเพื่อตอบสนองกับเหตุการณ์ที่กำหนดขึ้น

6. โมดูล (module) เป็นการเขียนโปรแกรมด้วยภาษา VBA (Visual Basic for Application) ซึ่งใช้ในการทำงานที่ค่อนข้างซับซ้อนกว่าการใช้แมโคร เพื่อให้ได้ฟอร์มหรือรายงานตามความต้องการของผู้ใช้

เริ่มต้นใช้งานโปรแกรม Microsoft Access 2010

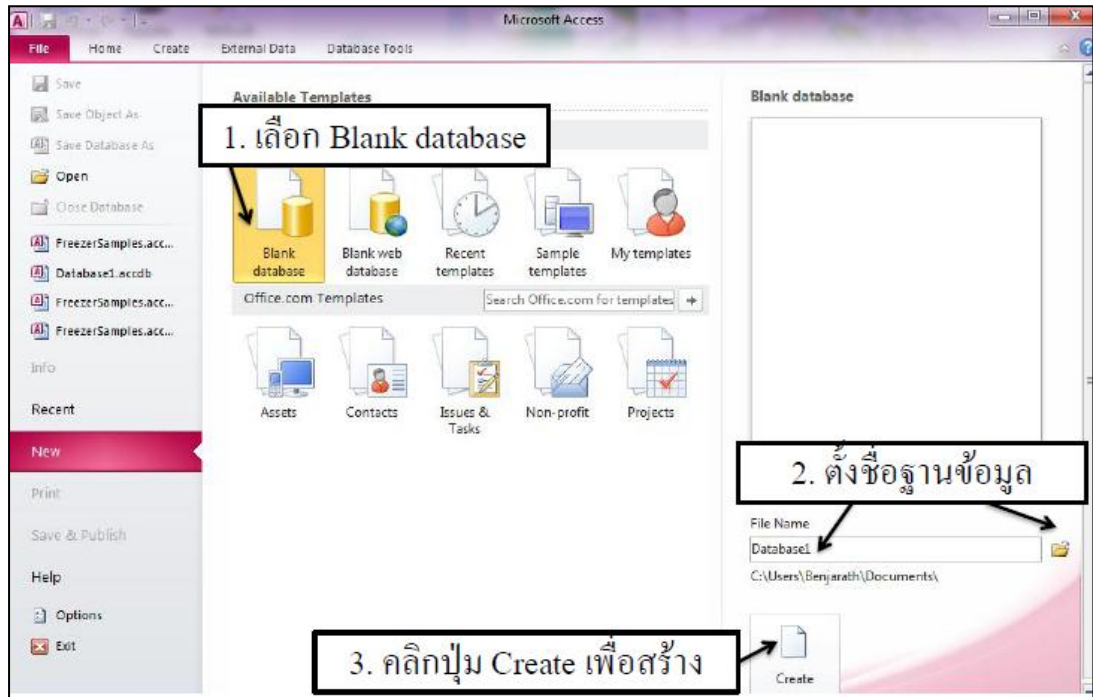
เรียกใช้งาน Microsoft Access 2010 ได้โดยการคลิกที่ปุ่ม Start → All Programs → Microsoft Access 2010 หรือคลิกที่ icon  จะปรากฏหน้าจอ ดังภาพที่ 2.1



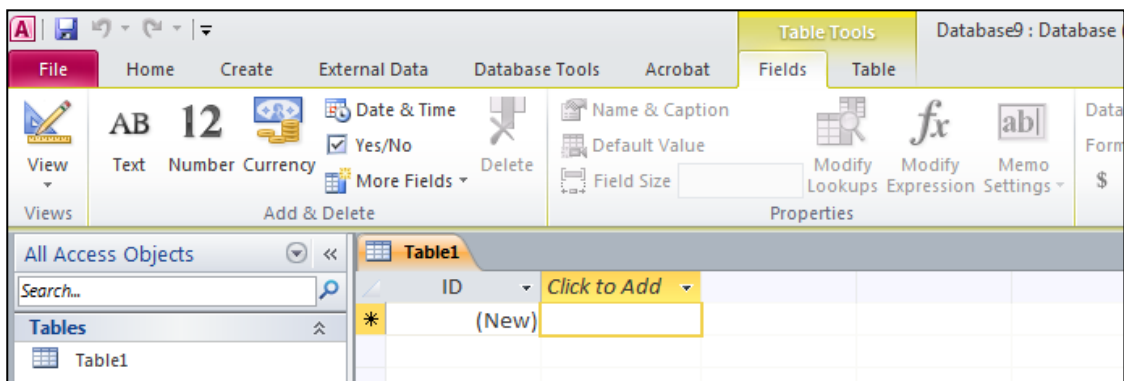
ภาพที่ 2.1 หน้าจอเมื่อเปิดโปรแกรม Microsoft Access 2010

วิธีการสร้างฐานข้อมูลมี 2 แบบ คือ

1. สร้างฐานข้อมูลเปล่า (Blank database) เพื่อกำหนดรูปแบบของโครงสร้างฐานข้อมูลด้วยตนเอง โดยการคลิกเลือก Blank database ตั้งชื่อฐานข้อมูลที่จะสร้างขึ้นใหม่ และคลิกปุ่ม Create เพื่อสร้างฐานข้อมูล ดังภาพที่ 2.2 จะปรากฏภาพที่ 2.3

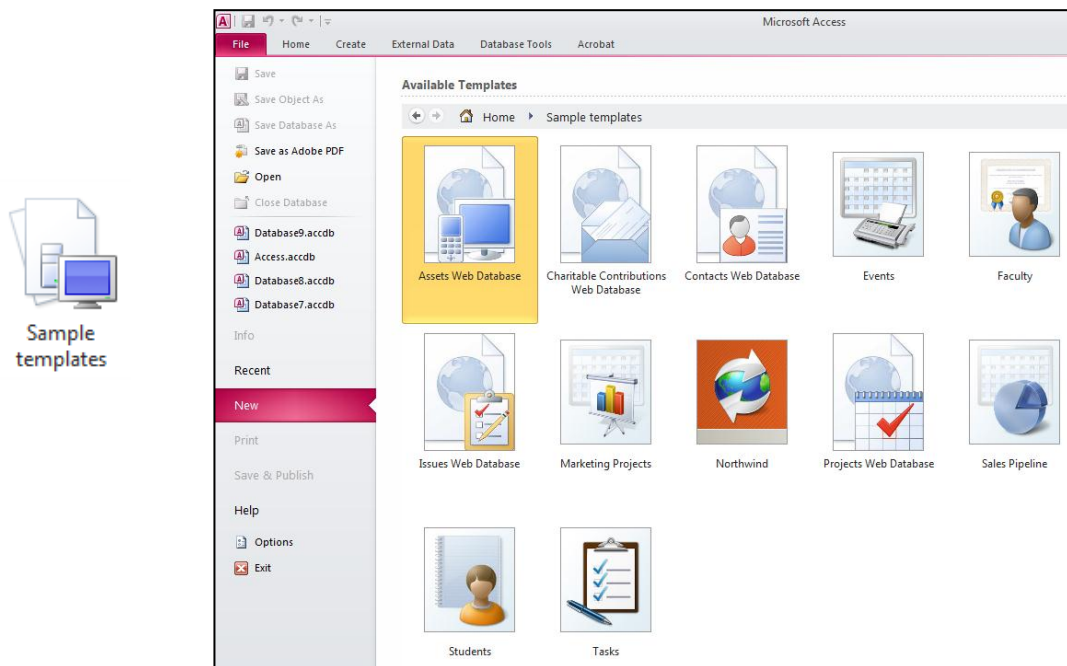


ภาพที่ 2.2 วิธีการสร้างฐานข้อมูลเปล่า (blank database)



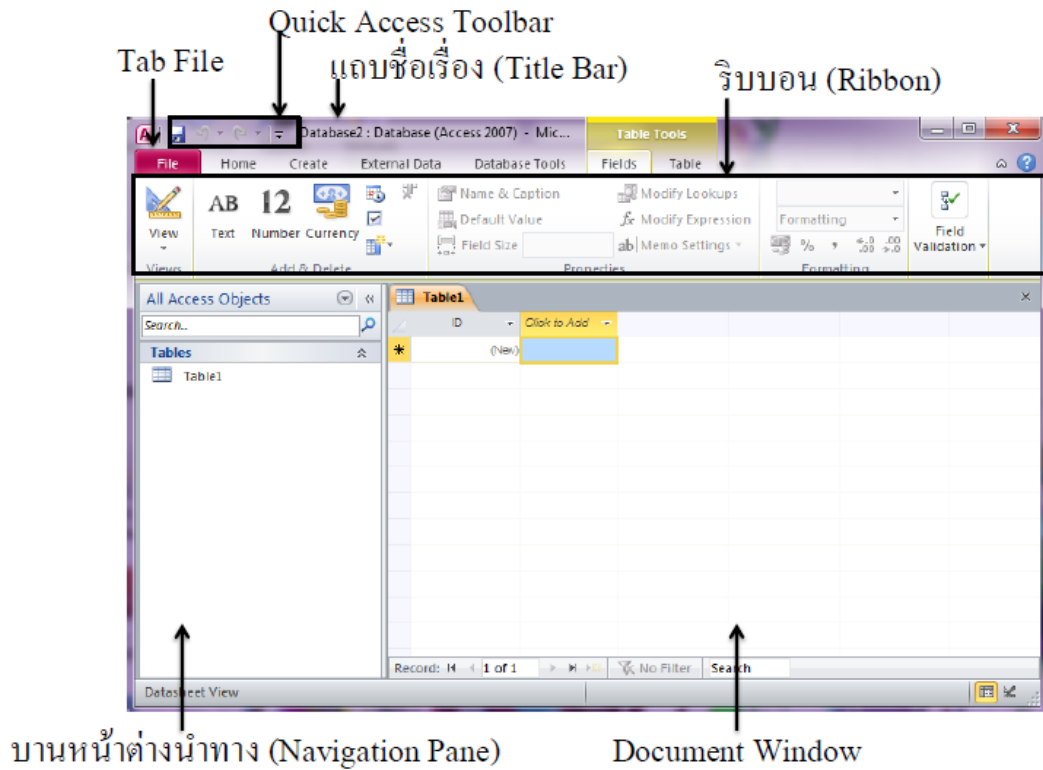
ภาพที่ 2.3 ฐานข้อมูลเปล่า (blank database)

2. สร้างฐานข้อมูลจากแม่แบบ (Sample templates) ที่โปรแกรมเตรียมไว้หลายรูปแบบสำหรับการใช้งานในด้านต่างๆ เช่น ฐานข้อมูลบุคคลากร (Personal) ฐานข้อมูลทางการศึกษา (Education) และฐานข้อมูลการเงิน (Finance) เป็นต้น โดยทำตามขั้นตอนไปที่ละขั้น เมื่อเสร็จแล้วจะได้ไฟล์ฐานข้อมูลที่น่าไปใช้งานได้ทันที ถ้าต้องการแก้ไขปรับปรุงก็สามารถทำได้ในภายหลัง ซึ่งมีวิธีการสร้างโดยการคลิกเลือก Sample templates จะปรากฏภาพที่ 2.4 แต่ต้องมีการเชื่อมต่ออินเทอร์เน็ตด้วย เพื่อเชื่อมต่อกับเว็บ Office.com



ภาพที่ 2.4 วิธีการสร้างฐานข้อมูลจากแม่แบบ (Sample templates)

ส่วนประกอบของหน้าต่างโปรแกรม Microsoft Access 2010



ภาพที่ 2.5 ส่วนประกอบของหน้าต่างโปรแกรม Microsoft Access 2010

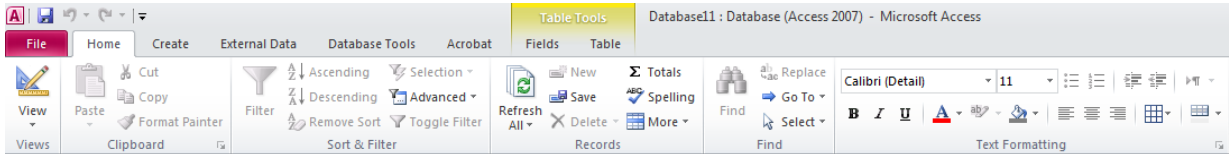
ตารางที่ 2.1 หน้าที่ของส่วนประกอบของหน้าต่างโปรแกรม Microsoft Access 2010

ส่วนประกอบ	หน้าที่การทำงาน
Tab File	รวบรวมชุดคำสั่งในการจัดการฐานข้อมูล เช่น การสร้าง การบันทึก การพิมพ์ การสำรอง การตั้งค่า Access 2010
Quick Access Toolbar	แถบเครื่องมือที่ได้รวบรวมปุ่มที่ใช้งานบ่อยๆเอาไว้ เพื่อช่วยให้ทำงานสะดวกรวดเร็วขึ้น และสามารถเพิ่มหรือลดปุ่มเครื่องมือเหล่านี้ได้
แถบชื่อเรื่อง (Title Bar)	แถบแสดงชื่อเรื่องและเวอร์ชันของโปรแกรม
ริบบอน (Ribbon)	แบ่งกลุ่มการทำงานหลักๆ ไว้ด้วยกัน มีริบบอน Home, Create, External Data, Database Tools, Fields, Tables
บานหน้าต่างนำทาง (Navigation Pane)	แสดงสิ่งต่างๆ ที่ได้มีการสร้างขึ้น เช่น Table, Query, Form, Report เป็นต้น สามารถซ่อนและเปิดใหม่ได้
Document Window	พื้นที่หลักที่ใช้ในการทำงาน

การทำงานของเมนูแบบริบบอน (Ribbon)

คำสั่งทั้งหมดที่ใช้งานจะอยู่บนแถบสีเหลี่ยมผืนผ้าตรงส่วนบนที่เรียกว่า ริบบอน (Ribbon) ซึ่งในการเรียกใช้งานคำสั่งต่างๆ จะสะดวกและรวดเร็วกว่าการเรียกใช้คำสั่งในเมนูแบบ ลาดับชั้นหรือเมนูแบบ pull down

1. Tab Home

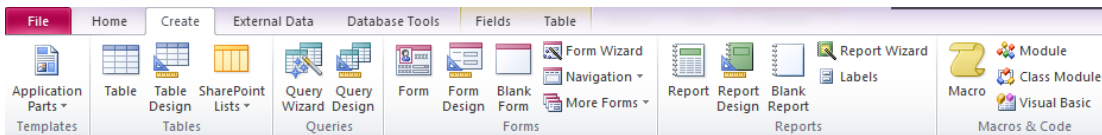


ภาพที่ 2.6 คำสั่งใน Tab Home

Tab Home เป็นแท็บที่รวบรวมคำสั่งเกี่ยวกับการจัดการข้อมูลบนหน้าจอ ซึ่งแบ่งออกเป็นกลุ่มต่างๆ ดังนี้

- View: ใช้แสดงมุมมองต่างๆ เช่น มุมมองการแสดงข้อมูล และมุมมองการออกแบบ เป็นต้น
- Clipboard: คัดลอก เคลื่อนย้าย และวางข้อมูลในตำแหน่งที่ต้องการ
- Sort & Filter: เรียงข้อมูลและกรองข้อมูลตามเงื่อนไขที่กำหนด
- Record: กลุ่มคำสั่งทำงานกับเรคคอร์ด
- Find: ค้นหาและแทนที่ข้อมูล
- Text Formatting: จัดแบบตัวอักษร ย่อหน้า เป็นต้น

2. Tab Create



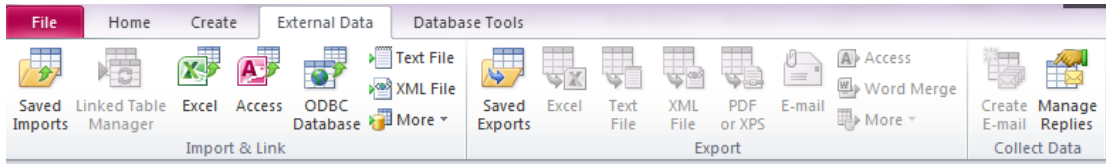
ภาพที่ 2.7 คำสั่งใน Tab Create

Tab Create ประกอบด้วยคำสั่งที่ใช้ในการเริ่มต้นสร้างส่วนต่างๆ ของฐานข้อมูล ได้แก่

- Tables: เป็นกลุ่มคำสั่งในการสร้างตารางข้อมูล
- Queries: เป็นกลุ่มคำสั่งในการสร้างแบบสอบถามข้อมูล
- Forms: เป็นกลุ่มคำสั่งในการสร้างฟอร์ม

- Reports: เป็นกลุ่มคำสั่งในการสร้างรายงาน
- Macro & Module: เป็นกลุ่มคำสั่งในการสร้าง Macro และ Module

3. Tab External Data

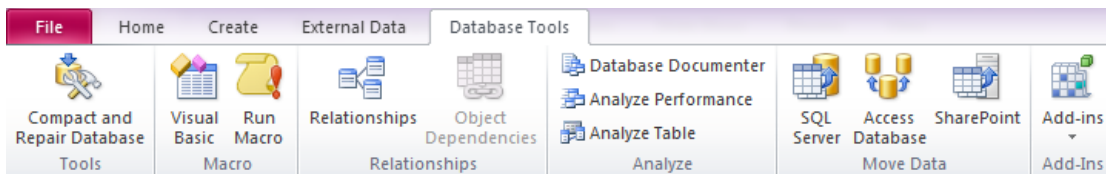


ภาพที่ 2.8 คำสั่งใน Tab External Data

Tab External Data รวบรวมคำสั่งในการนำเข้าหรือส่งออกข้อมูล ซึ่งแบ่งออกเป็นกลุ่มต่างๆ ดังนี้

- Import & Link: นำเข้าข้อมูลจากฐานข้อมูล Microsoft Access หรือ Microsoft Excel
- Export: ส่งฐานข้อมูลออกเป็น Microsoft Access, Microsoft Excel, HTML เป็นต้น
- Collect Data: สร้างและจัดการ E-mail ร่วมกับโปรแกรม Microsoft Outlook

4. Tab Database Tools



ภาพที่ 2.9 คำสั่งใน Tab Database Tools

Tab Database Tools รวบรวมคำสั่งในการจัดการฐานข้อมูล ได้แก่

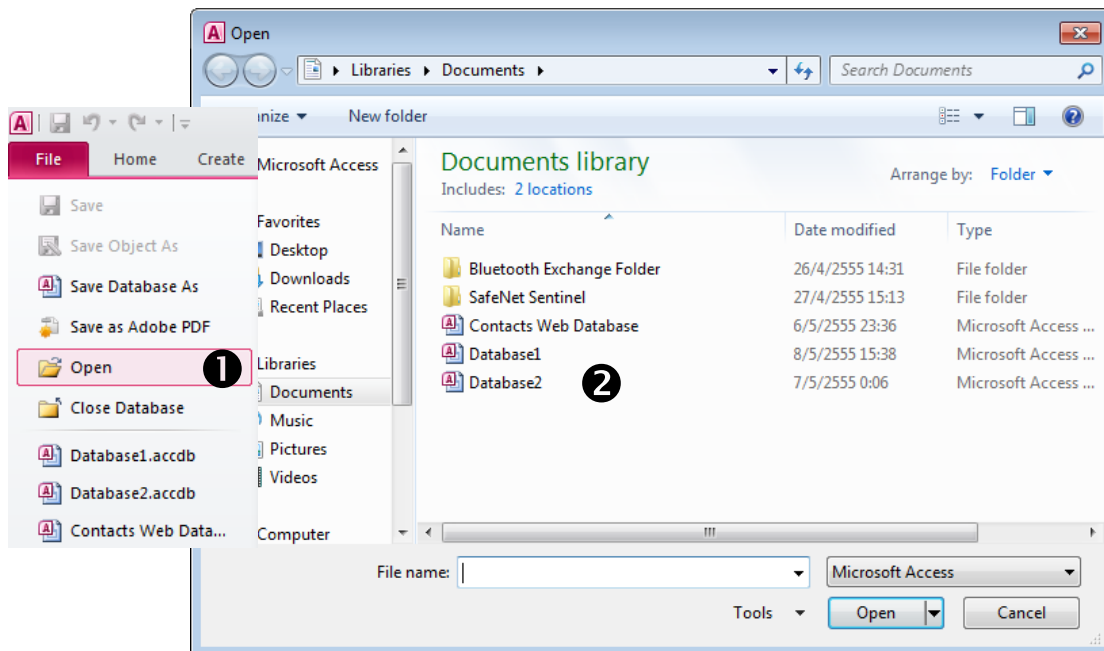
- Tools: คำสั่งที่ใช้ในการบีบอัดและซ่อมแซมฐานข้อมูล
- Macro: คำสั่งที่ใช้ร่วมกับ Macro การแปลง Macro ให้เป็นคำสั่งในภาษา Visual Basic
- Relationships: สร้างความสัมพันธ์ระหว่างตาราง
- Analyze: ทำหน้าที่ตรวจสอบและวิเคราะห์ประสิทธิภาพในการทำงานของฐานข้อมูล
- Move Data: คำสั่งที่ใช้ในการติดต่อระหว่าง Access กับ SQL Server

- Add-Ins: ทำหน้าที่เพิ่มคำสั่งแบบกำหนดเองและคุณลักษณะใหม่ๆ เพื่อปรับปรุงและเพิ่มประสิทธิภาพในการทำงานร่วมกับโปรแกรม Access

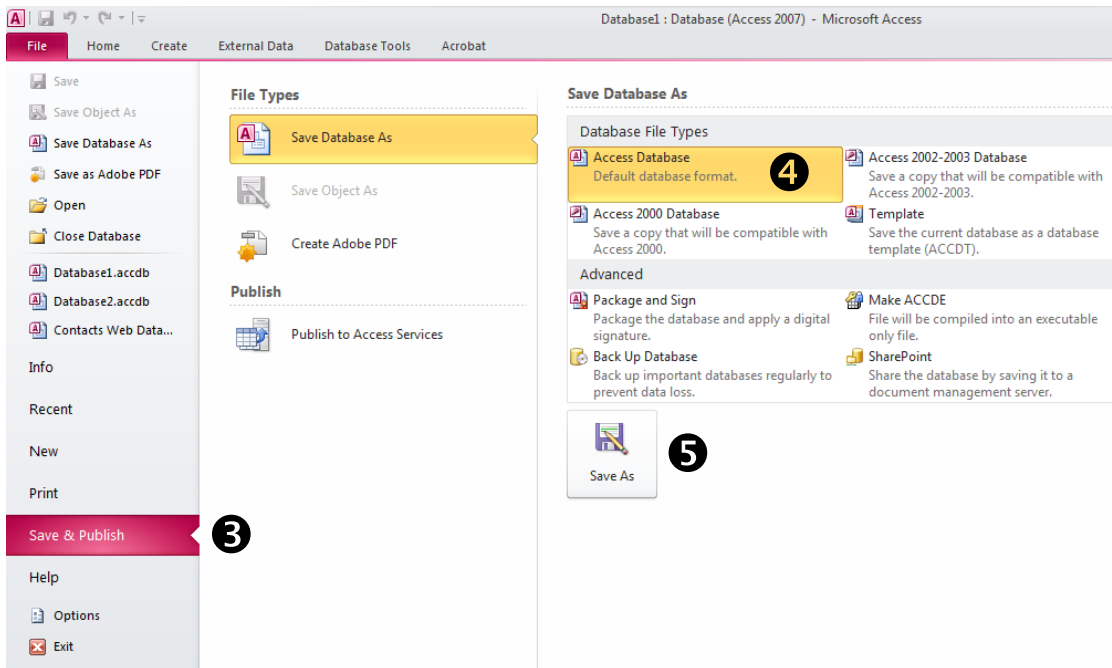
การแปลงไฟล์ .mdb ให้เป็นไฟล์ .accdb

ไฟล์ฐานข้อมูลเวอร์ชันเก่า เช่น Access 97, 2000, 2002-2003 (ไฟล์ .mdb) จะเปิดใช้ในโปรแกรม Access 2010 ได้ แต่จะทำงานร่วมกับความสามารถใหม่ๆ ของ Access 2010 ไม่ได้ จึงต้องแปลงให้เป็นไฟล์ .accdb (Access 2007) ก่อน โดย

1. เลือก Open จากเมนู File
2. เลือกไฟล์ที่ต้องการจะแปลง
3. เลือก Save & Publish จากเมนู File
4. เลือก Access Database
5. คลิก Save As
6. หลังจากแปลงไฟล์เสร็จแล้ว จะมีข้อความแจ้งว่า ไฟล์ที่แปลงแล้วจะใช้กับโปรแกรม Access เวอร์ชันเก่าไม่ได้ ให้คลิก OK
7. เข้าสู่วินโดว์ Database ของไฟล์ใหม่ที่ห้ววินโดว์จะบอกว่ารูปแบบไฟล์ คือ Access 2007 ซึ่งเป็นไฟล์แบบ .accdb



ภาพที่ 2.10 เปิดไฟล์ที่ต้องการจะแปลงไฟล์



ภาพที่ 2.11 ขั้นตอนการแปลงไฟล์.mdb ให้เป็นไฟล์ .accdb

การจัดเก็บไฟล์ฐานข้อมูล

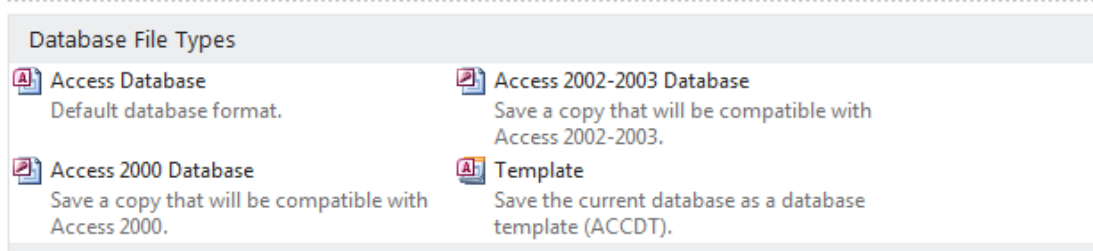
1. การจัดเก็บไฟล์เป็น .accdb และ .mdb

ถ้าจะเก็บไฟล์แทนที่ไฟล์เดิม ให้ใช้คำสั่ง Save ถ้าจะเก็บไฟล์ใหม่โดยชนิดไฟล์เหมือนเดิม ใช้ Save Database As แต่ถ้าจะเลือกชนิดไฟล์ต่างจากเดิม ใช้คำสั่ง Save & Publish โดยสามารถเลือกรูปแบบที่ต้องการจัดเก็บได้ดังนี้

- Access Database จะจัดเก็บเป็นไฟล์ .accdb (Access2007)
- Access 2002-2003 Database จะจัดเก็บเป็นไฟล์ .mdb
- Access 2000 Database จะจัดเก็บเป็นไฟล์ .mdb

2. การจัดเก็บไฟล์เป็น .accdt (template)

ไฟล์ฐานข้อมูลที่จะนำมาสร้าง template ต้องเป็นไฟล์ .accdb เท่านั้น ในการจัดเก็บไฟล์ให้ใช้คำสั่ง Save & Publish แล้วคลิกเลือกรูปแบบ Template โปรแกรมจะจัดเก็บเป็นไฟล์ .accdt



ภาพที่ 2.12 ประเภทของไฟล์ฐานข้อมูล

บทที่ 3

การสร้างตาราง (Table)

ตาราง (Table) คือ องค์ประกอบหลักอย่างหนึ่งในฐานข้อมูล เป็นส่วนที่ต้องสร้างขึ้นเป็นอันดับแรก เพื่อใช้ในการเก็บข้อมูลทั้งหมดที่อยู่ในฐานข้อมูล โดยในการสร้างตารางนั้นมีหลายรูปแบบด้วยกัน ซึ่งเราสามารถจัดการกับข้อมูลในตารางเพื่อให้เกิดความสะดวกและรวดเร็วในการทำงานมากยิ่งขึ้นได้ เช่น การเรียงลำดับข้อมูล การกรองข้อมูล และการค้นหาข้อมูล เป็นต้น

มุมมองของตาราง

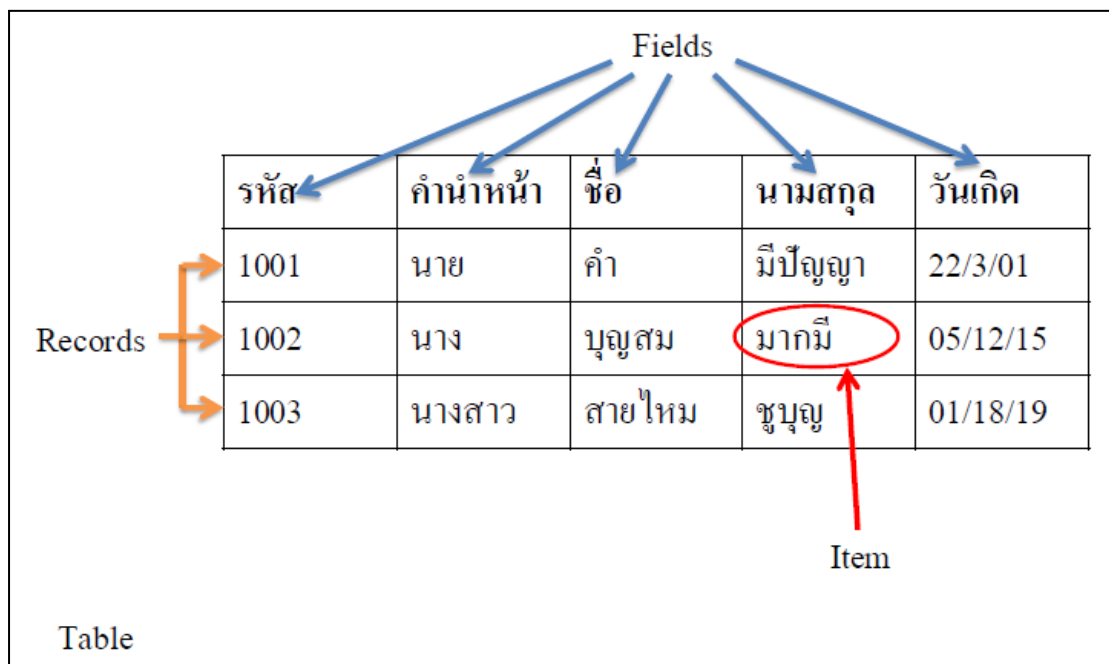
มุมมอง (View) คือ รูปแบบการทำงานกับ Table ซึ่งมีอยู่ด้วยกันทั้งหมด 4 แบบ ตามลักษณะและวัตถุประสงค์ของการใช้งาน คือ

1. **มุมมอง Design** ใช้ในการออกแบบ และแก้ไขโครงสร้างของตาราง เช่น เพิ่มลบเขตข้อมูล (field) แก้ไขชื่อเขตข้อมูล กำหนดชนิดข้อมูล กำหนดคีย์ กำหนดคุณสมบัติต่างๆ ของเขตข้อมูล (Properties) เป็นต้น
2. **มุมมอง Datasheet** ใช้ในการป้อนข้อมูล หรือแสดงข้อมูลที่เก็บไว้ในตาราง โดยแสดงในรูปแบบของตาราง ในมุมมองนี้สามารถเพิ่ม ลบ หรือแก้ไขเรคอร์ดได้
3. **มุมมอง PivotTable** ใช้วิเคราะห์และสรุปผลข้อมูลในตารางโดยแสดงในรูปแบบของตารางแจกแจงรายละเอียดข้อมูลและสรุปผลข้อมูล
4. **มุมมอง PivotChart** ใช้วิเคราะห์และสรุปผลข้อมูลในตารางโดยแสดงในรูปแบบของแผนภูมิหรือ Chart

ส่วนประกอบของตาราง

1. ไอเท็ม (Item) คือ ข้อมูลที่เก็บอยู่ในแต่ละเซลล์ของตาราง
2. field (Field) คือ ข้อมูลในแนวคอลัมน์ (แนวตั้ง)
3. เรคอร์ด (Record) คือ ข้อมูลในแต่ละแถวของตาราง
4. ตาราง (Table) คือ ส่วนของตารางทั้งหมดที่ใช้ในการเก็บข้อมูล ซึ่งถ้ามีหลายๆ

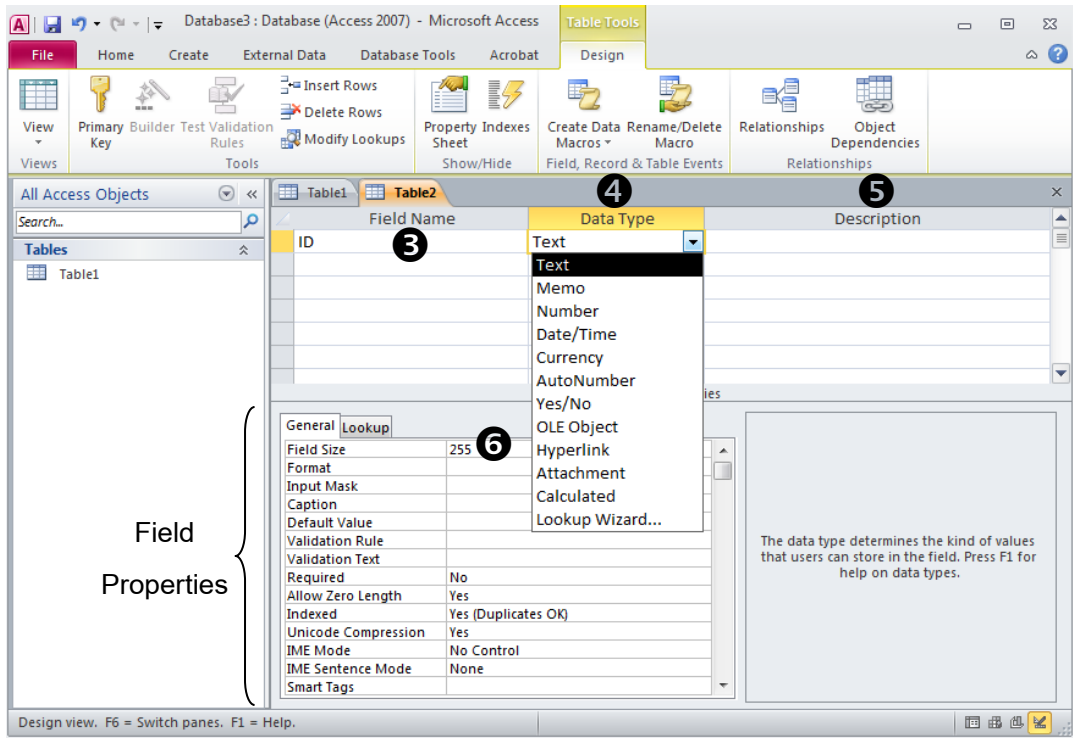
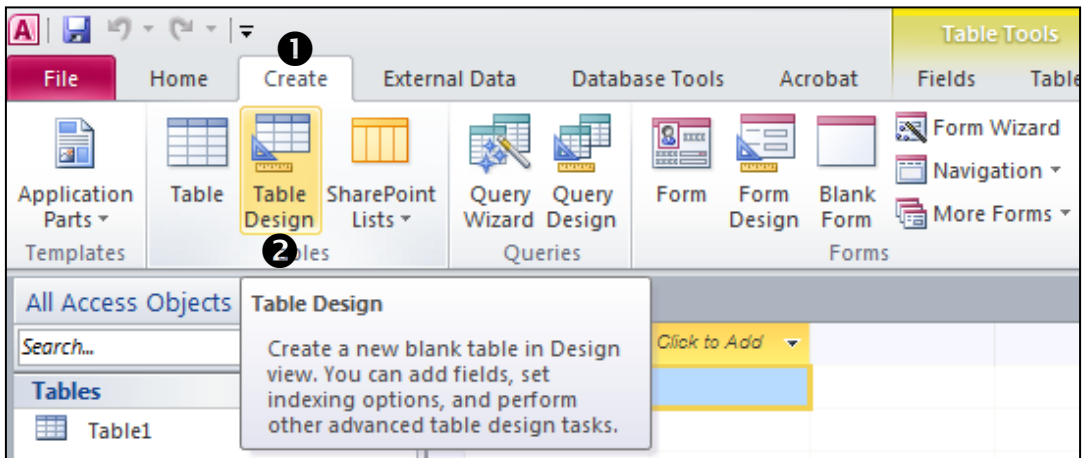
ตารางมารวมกันทั้งหมด จะเรียกว่า ฐานข้อมูล หรือ Database



ภาพที่ 3.1 ตัวอย่างแสดงส่วนประกอบของตาราง

การสร้างตาราง

1. คลิกแท็บ Create
2. เลือกรูปแบบ Table Design
3. ตั้งชื่อ Field ในช่อง Field Name
4. เลือกประเภทของข้อมูลในช่อง Data Type
5. ใส่รายละเอียดย่อๆ ของแต่ละ field ในช่อง Description (จะใส่หรือไม่ก็ได้)
6. กำหนดขนาดของ field ในช่องคุณสมบัติ Field Size



ภาพที่ 3.2 ขั้นตอนการสร้างตาราง

การตั้งชื่อ field

- ชื่อ field ต้องไม่ยาวเกิน 64 ตัวอักษร รวมทั้งช่องว่างด้วย
- ห้ามตั้งชื่อ field ซ้ำกัน
- สามารถใช้ตัวอักษร ตัวเลข ช่องว่าง ในการตั้งชื่อ field ได้
- ห้ามใช้เครื่องหมาย จุด (.) อักเจี๋ย (!) และก้ามปู ([]) ในการตั้งชื่อ field
- ห้ามเริ่มชื่อ field ด้วยช่องว่าง

ชนิดข้อมูล

ชนิดข้อมูล (data type) เป็นส่วนที่ใช้กำหนดชนิดของข้อมูลในแต่ละ field โดยต้องเลือกให้สอดคล้องกับข้อมูลที่จะจัดเก็บในแต่ละ field ซึ่งมีทั้งหมด 11 ชนิด คือ

1) Text ข้อความที่ประกอบไปด้วยตัวอักษรที่อาจเป็นตัวอักษร สัญลักษณ์พิเศษ ช่องว่าง หรือตัวเลขประกอบกัน ซึ่งถ้าเป็นตัวเลขอย่างเดียวจะต้องเป็นตัวเลขที่ไม่ใช้ในการคำนวณ เช่น หมายเลขบัตรประจำตัวประชาชน รหัสนักศึกษา เบอร์โทรศัพท์ บ้านเลขที่ และรหัสไปรษณีย์ เป็นต้น โดยสามารถบรรจุข้อมูลได้สูงสุดจำนวน 255 ตัวอักษร

2) Memo ข้อความที่เป็นตัวอักษรที่มีความยาวมากๆ และมากกว่า 255 ตัวอักษร แต่ไม่เกิน 65,535 ตัวอักษร ส่วนมากใช้ในการเก็บข้อมูลและรายละเอียดที่มีความยาวของข้อมูลมากๆ นั้นเอง

3) Number ข้อมูลที่เป็นตัวเลข สามารถนำไปคำนวณได้ เช่น จำนวนสินค้า และจำนวนหน่วยกิต เป็นต้น

4) Date/Time ข้อมูลที่เป็นวันที่และเวลา

5) Currency ข้อมูลที่เป็นตัวเลข ใช้ในการคำนวณข้อมูลเกี่ยวกับการเงิน เช่น ราคา สินค้า ค่าหน่วยกิต เงินเดือน รายรับ และรายจ่าย เป็นต้น

6) Auto Number เป็นตัวเลขจำนวนเต็มแบบลำดับที่ ซึ่งโปรแกรมจะกำหนดค่าให้โดยอัตโนมัติเมื่อมีการเพิ่มเรคอร์ดใหม่เข้ามาในตาราง

7) Yes/No เป็นข้อมูลทางตรรกะ ซึ่งมีสถานะเป็นจริงหรือเท็จ เช่น True/False, Yes/No, หรือ On/Off และจะเป็นค่าว่างไม่ได้

8) OLE Object ข้อมูลที่เป็นการเชื่อมโยงหรือนำเข้าข้อมูลจากโปรแกรมอื่นๆ มาเก็บไว้ เช่น รูปภาพ ตาราง กราฟ และเสียง เป็นต้น OLE ย่อมาจาก Object Linking and Embedding เป็นเทคนิคของการประยุกต์ใช้งานบนวินโดวส์ ที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างกัน เช่น สามารถนำรูปภาพจากโปรแกรม PhotoShop มาใช้ใน Microsoft Word ได้ เป็นต้น

9) Hyperlink เป็นข้อมูลหรือแอดเดรสที่ใช้อ้างอิงไปยังข้อมูลอื่นๆ หรือระบุมการเชื่อมโยงในแบบของเว็บเพจ ซึ่งแสดงด้วย URL (Uniform Resource Locator) โดยการอ้างอิงไปยังแหล่งข้อมูลอื่น ซึ่งอาจเป็นได้ทั้งไฟล์ฐานข้อมูลของโปรแกรมไมโครซอฟต์เอกเซลเอง หรือไฟล์ของโปรแกรมอื่นที่อยู่ในเครื่องเดียวกัน นอกจากนี้ยังสามารถเชื่อมโยงไปยังเว็บไซต์บนอินเทอร์เน็ตได้

10) Attachment เหมือนไฟล์แนบในอีเมล เป็นไฟล์ชนิดใดก็ได้ เช่น Word, Excel

11) Calculated ผลของการคำนวณจาก field อื่นในตาราง

Lookup Wizard ... ที่แสดงในเมนู Data Type นั้น ไม่ใช่ชนิดข้อมูลแต่เป็นเครื่องมือช่วยป้อนข้อมูลและนำเข้าข้อมูลจาก Table อื่นๆ ของฐานข้อมูล

คุณสมบัติของ field

คุณสมบัติของ field (Field Properties) เป็นส่วนที่ใช้กำหนดคุณสมบัติของแต่ละ field เช่น กำหนดขนาดของ field รูปแบบที่ใช้ในการป้อนข้อมูล รูปแบบการแสดงผลข้อมูล และกำหนดเงื่อนไขที่ใช้ตรวจสอบค่าใน field เป็นต้น

- **Field Size** คือ ขนาดของ field หรือความยาวของข้อมูลที่สามารถป้อนเข้าไปได้ เช่น รหัสไปรษณีย์ จะมีความยาว 5 อักขระ ค่า default ของ Field Size คือ 255 อักขระ ดังนั้นถ้าเรากำหนดให้เล็กลงได้ ก็จะช่วยลดการเก็บพื้นที่ข้อมูล

- **Format** ใช้ในการกำหนดรูปแบบในการแสดงข้อมูลที่หน้าจอ แต่ไม่ได้เปลี่ยนข้อมูลจริง เช่น

ตารางที่ 3.1 ตัวอย่างรูปแบบในการแสดงข้อมูล

สัญลักษณ์	คำอธิบาย	ตัวอย่าง
@	อักขระ 1 ตัว และทุกช่องจะต้องมีข้อมูล แต่ถ้าไม่ป้อน Access จะใส่ช่องว่างให้	@@-@@@ แสดงเป็น 43-001
<	แสดงตัวอักษรเป็นตัวพิมพ์เล็กทั้งหมด	แสดงเป็น bangkok
>	แสดงตัวอักษรเป็นตัวพิมพ์ใหญ่ทั้งหมด	แสดงเป็น BANGKOK

- **Decimal Place** ใช้กำหนดว่าจะให้แสดงจุดทศนิยมกี่ตำแหน่ง ซึ่งจะใช้ได้กับข้อมูลชนิด Number และ Currency เท่านั้น

- **Input Mask** ใช้กำหนดรูปแบบในการป้อนข้อมูล เพื่อความสะดวกรวดเร็ว เช่น หมายเลขโทรศัพท์ และหมายเลขบัตรประชาชน เป็นต้น

- **Caption** ใช้กำหนดข้อความที่แสดงในส่วนหัวคอลัมน์ใน Datasheet View หรือ เป็นชื่อที่จะปรากฏใน Form หรือ Report
- **Default Value** ใช้กำหนดค่าเริ่มต้นของข้อมูลใน field พิมพ์ใหม่ได้ แต่ถ้าไม่พิมพ์ใหม่ก็จะมีค่าเป็น Default Value
- **Validation Rule** ใช้กำหนดเงื่อนไขสำหรับค่าของข้อมูล เช่น ใน field นี้ จะต้องป้อนข้อมูลเพียง 2 ค่า คือ หญิง หรือ ชาย เท่านั้น ถ้าป้อนนอกเหนือจากนี้แล้ว จะแสดงกรอบหน้าต่างเตือน และไม่สามารถป้อนนอกเหนือจากที่กำหนดไว้ได้ เช่น

ตารางที่ 3.2 ตัวอย่างการกำหนดเงื่อนไขสำหรับค่าของข้อมูล

กำหนดค่า	ความหมาย
“หญิง” or “ชาย”	ป้อนได้เป็นหนึ่งในสองค่านี้เท่านั้น
Between #01/01/2007# and #31/12/2007#	ป้อนได้ตั้งแต่วันที่ 1 ม.ค. 2007 ถึง 31 ธ.ค. 2007 เท่านั้น
>0	ป้อนด้วยค่าที่มากกว่าศูนย์เท่านั้น
>= Date()	ป้อนด้วยวันที่เป็นวันที่ปัจจุบันเป็นต้นไป

- **Required** ถ้าเลือก Yes จะต้องป้อนข้อมูลลงใน field เสมอ ปล่อยให้ว่างไว้ไม่ได้
- **Allow Zero Length** ถ้าจะให้ field แบบ Text หรือ Memo รับข้อมูลที่เป็นค่าว่าง (Null) หรือข้อความที่มีความยาวเป็นศูนย์ เช่น "" (ไม่มีช่องว่างในเครื่องหมายคำพูด) ให้ตั้งค่าเป็น Yes
- **Indexed** จะให้ field นั้นเป็นดัชนีหรือไม่ No (ไม่เป็นดัชนี), Yes (Duplicates OK) (เป็นดัชนีที่มีค่าซ้ำกันได้), Yes (No Duplicates) (เป็นดัชนีที่มีค่าซ้ำกันไม่ได้)
- **Unicode Compression** ใช้กับข้อมูล Text, Memo, และ Hyperlink เพื่อบีบอัดข้อมูลที่เข้ารหัสแบบ Unicode (Unicode จะใช้ 2 ไบต์แทนอักขระ 1 ตัว ทำให้ใช้พื้นที่เก็บข้อมูลมากกว่าปกติ) ค่าเริ่มต้นของคุณสมบัตินี้เป็น Yes เพื่อให้อักขระทุกตัวที่ไบต์แรกมีค่าเป็น 0 เช่น ภาษาอังกฤษ, สเปน, เยอรมัน ถูกบีบอัดบนอุปกรณ์เก็บข้อมูล และคลายออกเมื่อนำไปใช้ การบีบอัดจะไม่เกิดขึ้นถ้าขนาดข้อมูล Memo น้อยกว่า 4,096 ไบต์ หรือบีบอัดแล้วไม่ทำให้ขนาดข้อมูลเล็กลง
- **IME Mode และ IME Sentence Mode** ใช้กับข้อมูล Text, Memo, Date/Time และ Hyperlink พบคุณสมบัตินี้ใน Control Text Box, Combo Box และ List Box ในมุมมอง

Design ของ Form ด้วย ใช้ในกรณีติดตั้งโปรแกรม Input Mode Editors (IME) เพื่อเปลี่ยน layout ของคีย์บอร์ดให้สามารถคีย์ตัวอักษรในภาษาแถบเอเชียตะวันออก เช่น จีน ญี่ปุ่น และเกาหลี เป็นต้น

- **Smart Tags** ช่วยในการทำงานระหว่างฐานข้อมูล Access กับโปรแกรมภายนอกสะดวกขึ้น เช่น การเรียกใช้โปรแกรมรับ-ส่งอีเมล การนัดหมาย การติดต่อกับบุคคล เมื่อผู้ใช้คลิก field ข้อมูลที่จะติดต่อกับโปรแกรมภายนอกจะมีไอคอน Smart Tags ที่มุมล่างของ field ให้เลือกการทำงานได้จากเมนู

- **Text Format** ใช้กับข้อมูลแบบ Memo ถ้าตั้งค่าเป็น Plain Text หมายถึงแสดงข้อความโดยไม่ต้องจัดรูปแบบ ส่วน Rich Text หมายถึงแสดงข้อความแบบจัดรูปแบบ เช่น ตัวหนา ตัวเอน ย่อหน้า เป็นต้น

- **Text Align** ใช้กับข้อมูลทุกชนิดยกเว้น Attachment ทำหน้าจัดตำแหน่งข้อมูล เช่น ซิดซ้าย ซิดขวา เป็นต้น

- **Show Date Picker** ใช้กับข้อมูล Date/Time เท่านั้น มี 2 ทางเลือก คือ For Dates หมายถึงให้แสดงปฏิทินเล็กๆ เพื่อเลือกวัน/เดือน/ปี จากปฏิทินโดยไม่ต้องคีย์เอง และ Never หมายถึงไม่ต้องแสดงปฏิทิน ผู้ใช้จะคีย์วัน/เดือน/ปี ลงไปเอง

การกำหนดคีย์หลัก (Primary Key)

- คีย์หลัก หรือ Primary Key คือ field ที่มีข้อมูลใน record ที่ไม่ซ้ำกัน เพื่อเป็นตัวกำหนดให้ทุก record ต่างกัน

- ประโยชน์ คือ เมื่อมีการป้อนข้อมูลใน field ที่กำหนดเป็น Primary Key ซ้ำกันแล้ว ก็เกิดค่าเตือนขึ้น และให้ป้อนข้อมูลใหม่

- Primary Key จะเป็น field ที่ไม่ว่าง จะต้องมีค่าอยู่เสมอ ตัวอย่าง เลขบัตรประจำตัวประชาชน และรหัสผู้ป่วย เป็นต้น

สัญลักษณ์กุญแจ แสดงว่าเป็น คีย์หลัก

The screenshot shows the Microsoft Access interface. In the ribbon, the 'Primary Key' icon is highlighted with a red box. Below it, the 'Subject' table is displayed with a list of fields. The 'SubjectID' field is highlighted with a red box, and its data type is 'AutoNumber'. A red arrow points from the 'Primary Key' icon to the 'SubjectID' field. Another red arrow points from the 'SubjectID' field to the 'AutoNumber' data type.

Field Name	Data Type	รหัสผู้ช่วย
SubjectID	AutoNumber	
Title	Text	ตำแหน่งงาน
Firstname	Text	ชื่อ
Lastname	Text	นามสกุล
Birthday	Date/Time	วันเกิด
Gender	Text	เพศ
Province	Text	ภูมิภาค
ZipCode	Text	รหัสไปรษณีย์
Height	Number	ส่วนสูง
Weight	Number	น้ำหนัก

Data Type เป็น AutoNumber เพื่อจะได้ไม่ซ้ำกัน
โดยเลขจะเพิ่มขึ้นทีละ 1 ตามจำนวนเรคคอร์ด

ภาพที่ 3.3 ขั้นตอนการสร้างคีย์หลัก (Primary Key)

การเพิ่มข้อมูลในตาราง (เรคคอร์ด)

การเพิ่มข้อมูลในตาราง สามารถทำได้หลายวิธี ดังต่อไปนี้

3. กดปุ่มเพิ่มเรคคอร์ดใหม่ตรงนี้ได้เช่นกัน

4. คลิกขวาที่เรคคอร์ดที่มีอยู่ก่อนหน้านี้ แล้วกด New Record

The screenshot shows the Microsoft Access interface with a table containing two records. A context menu is open over the first record, with 'New Record' selected. A yellow box highlights the 'New Record' button in the ribbon. Another yellow box highlights the 'New Record' option in the context menu. A third yellow box highlights the 'New Record' button in the ribbon. A fourth yellow box highlights the 'New Record' option in the context menu.

1. คลิกที่แถวใหม่ (สุดท้าย) แล้วพิมพ์ต่อไปเลย

2. กดปุ่มเพิ่มเรคคอร์ดใหม่ตรงนี้ได้เช่นกัน

ภาพที่ 3.4 ขั้นตอนการเพิ่มข้อมูลในตาราง

การลบข้อมูลในตาราง

1. เลือก record ที่ต้องการจะลบ
2. การลบข้อมูลในตาราง สามารถทำได้หลายวิธี ดังต่อไปนี้

3. กดปุ่ม Delete ตรงนี้

1. กดปุ่ม Delete บนแป้นพิมพ์

2. คลิกขวา แล้วเลือก Delete Record

โดย Access จะถามให้แน่ใจก่อน ในทุกช่องทาง เพราะลบแล้วจะเอาคืนกลับมาไม่ได้ ต้องพิมพ์ใหม่

ภาพที่ 3.5 ขั้นตอนการลบข้อมูลในตาราง

การทำงานกับ Record Navigator

Record Navigator

กลับไปที่เรคคอร์ดอันก่อนหน้า

กลับไปที่เรคคอร์ดแรกสุด

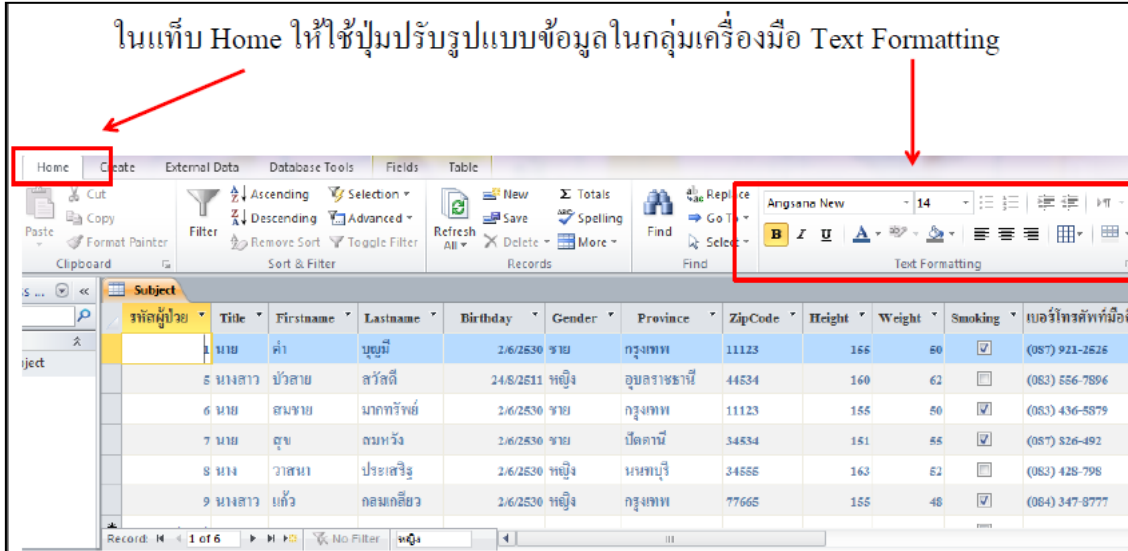
กลับไปที่เรคคอร์ดแรกสุด

กลับไปที่เรคคอร์ดอันก่อนหน้า

เพิ่มเรคคอร์ดใหม่

ภาพที่ 3.6 การทำงานกับ Record Navigator

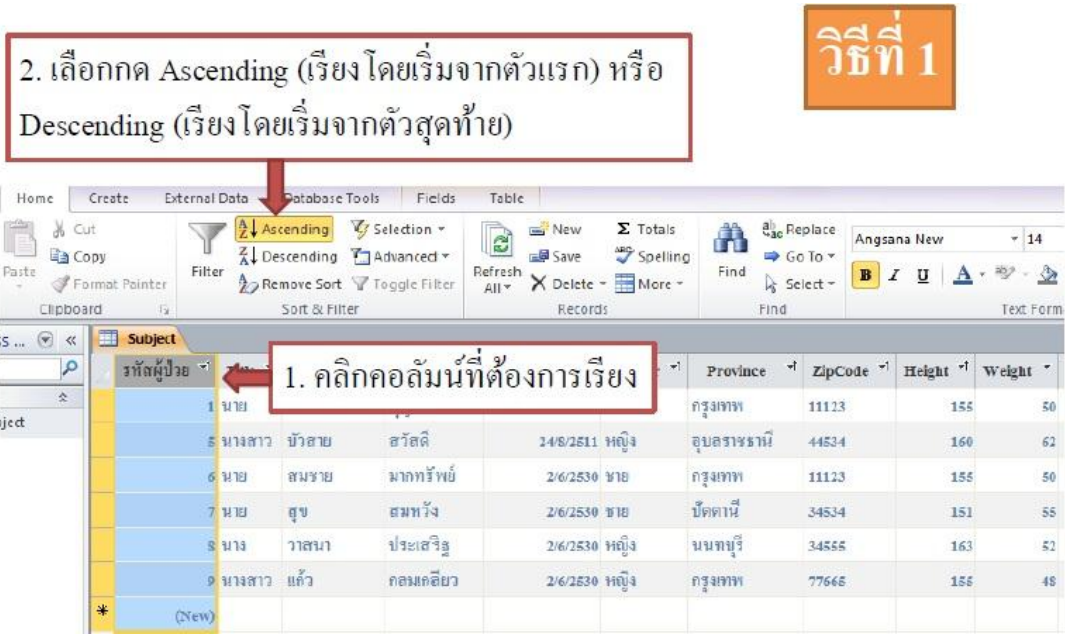
การจัดรูปแบบข้อมูลในตาราง



ภาพที่ 3.7 เครื่องมือการจัดรูปแบบในตาราง

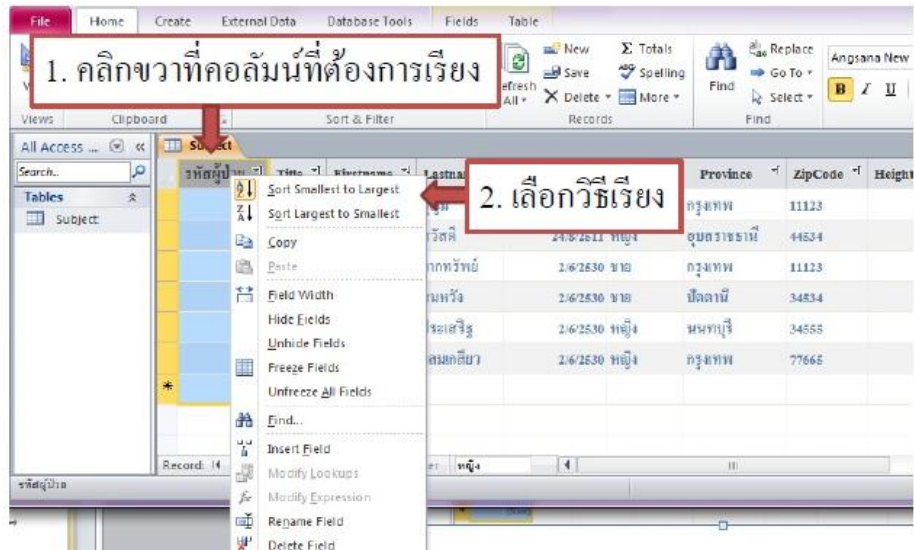
การเรียงลำดับข้อมูล (Sort)

การเรียงลำดับข้อมูล ทำได้สองทางดังนี้



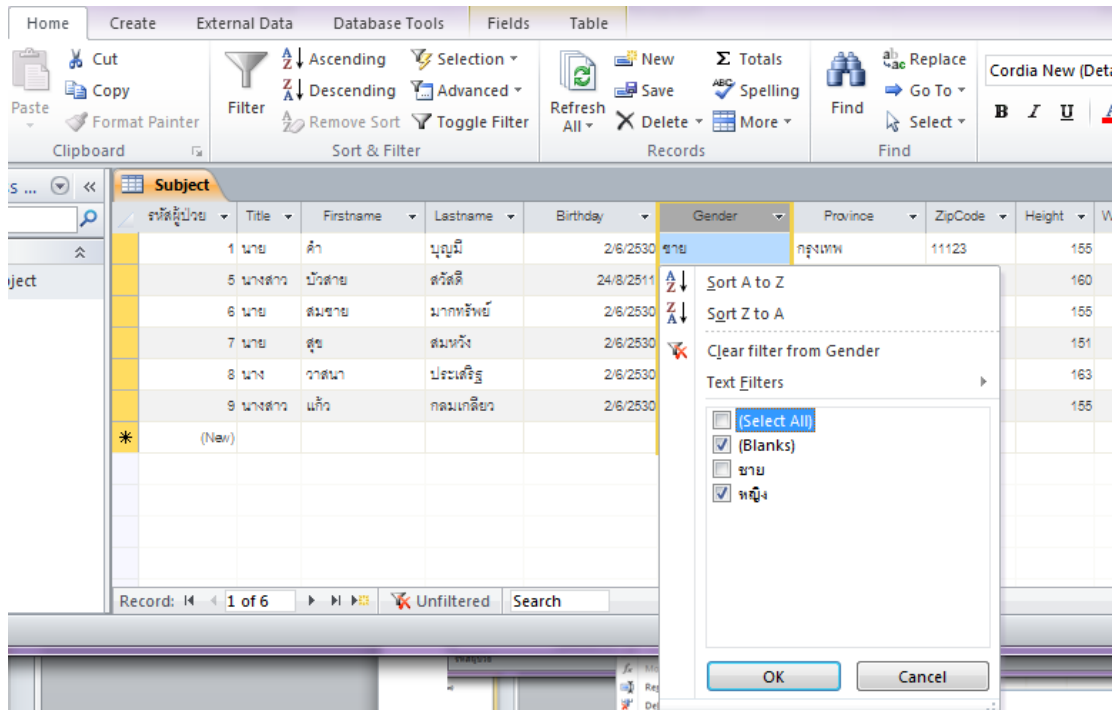
ภาพที่ 3.8 วิธีการเรียงลำดับข้อมูล (วิธีที่ 1)

วิธีที่ 2

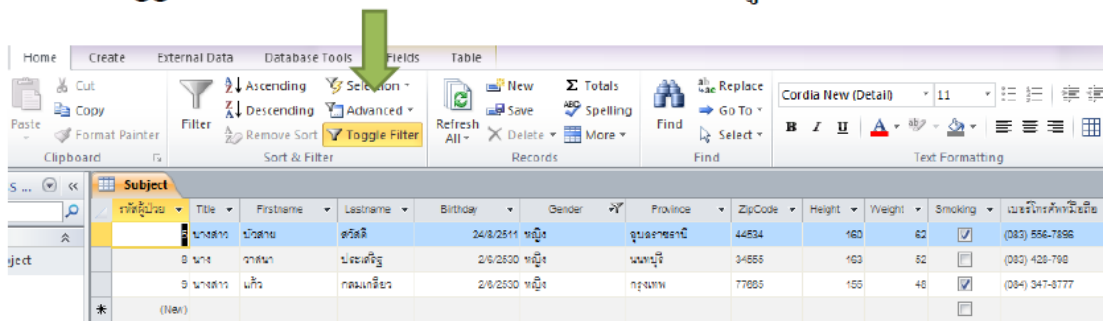


ภาพที่ 3.9 วิธีการเรียงลำดับข้อมูล (วิธีที่ 2)

การกรองข้อมูลใน field



กด Toggle Filter เพื่อกลับไปกลับมาระหว่างข้อมูลก่อนและหลังกรอง

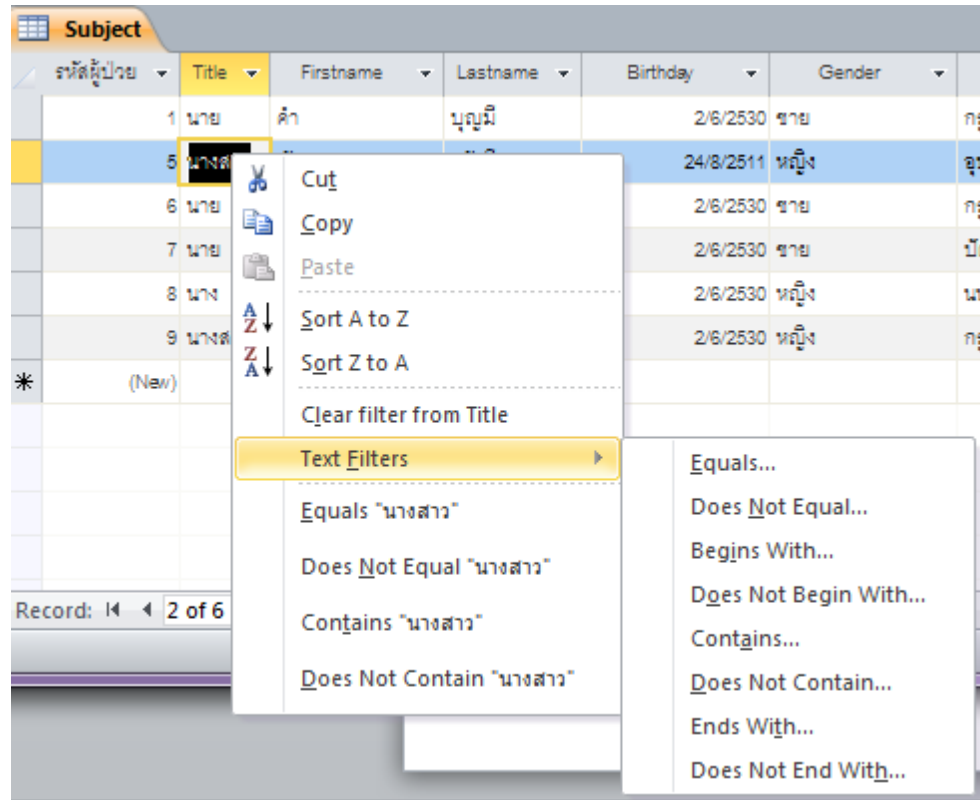


หลังจากคลิก OK แล้ว จะเหลือแต่เรคอร์ดที่ไม่มีคำว่า ชาย

ภาพที่ 3.10 วิธีการกรองข้อมูลใน field

การกรองข้อมูลแบบเลือกรายการ

1. คลิกขวาในช่องข้อมูลของใดก็ได้ของคอลัมน์ที่ต้องการกรอง

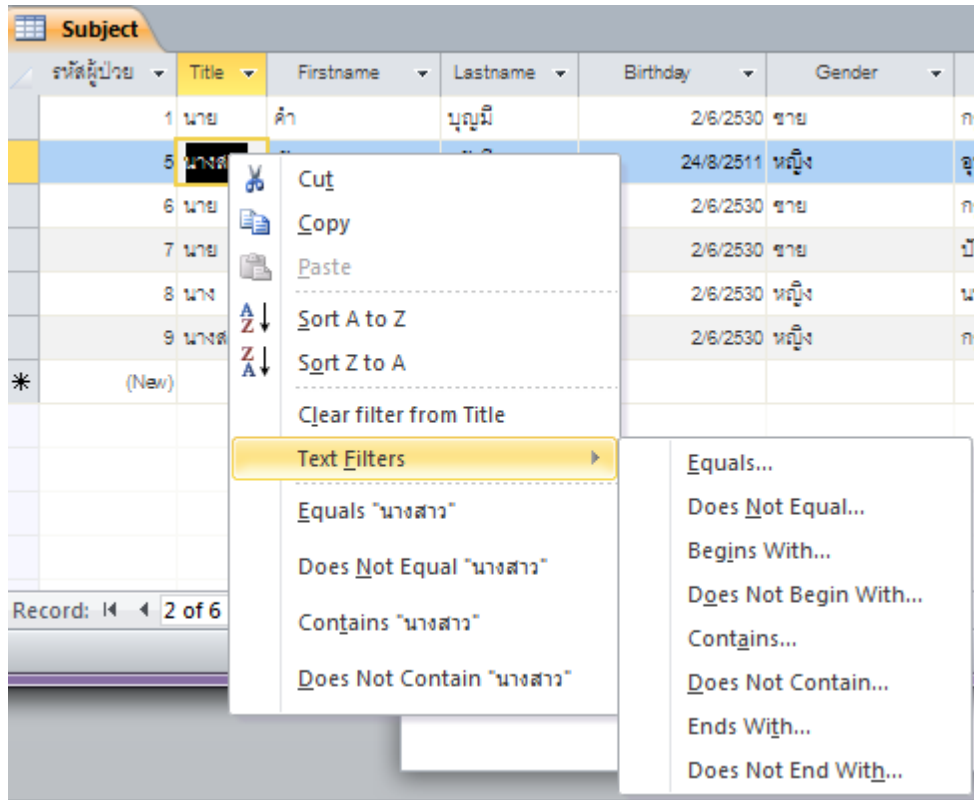


2. เลือก Text Filters

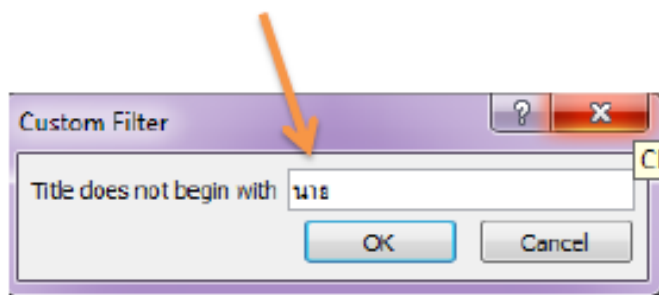
Equals	= เท่ากับ
Does Not Equal	= ไม่เท่ากับ
Begins With	= เริ่มต้นด้วย
Does Not Begin With	= ไม่เริ่มต้นด้วย
Contains	= มี
Does Not Contain	= ไม่มี
Ends With	= จบด้วย
Does Not End With	= ไม่จบด้วย

ภาพที่ 3.11 วิธีการกรองข้อมูลแบบเลือกรายการ

ตัวอย่างการกรองข้อมูลแบบเลือกรายการ



1. เลือก Does Not Begin With
2. เติม **นาย** แล้วคลิก OK



ภาพที่ 3.12 ตัวอย่างการกรองข้อมูลแบบเลือกรายการ

ตัวอย่างผลลัพธ์การกรองข้อมูลแบบเลือกรายการ

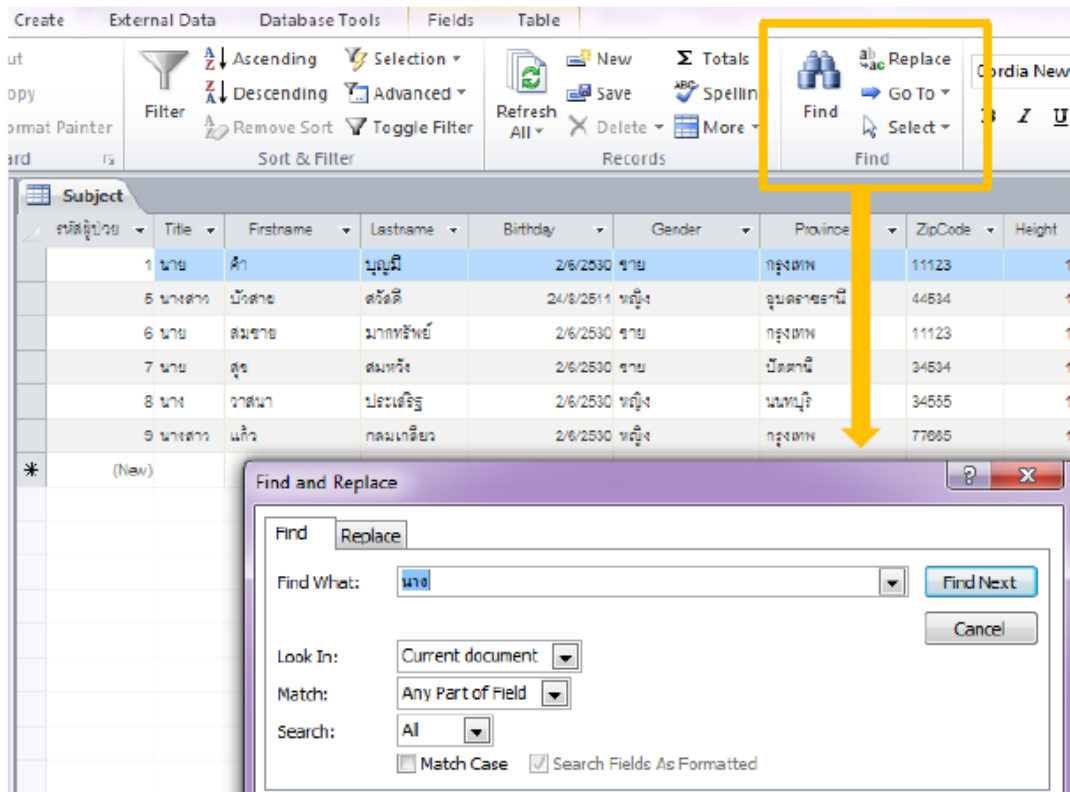
ชื่อผู้ปวย	Title	Firstname	Lastname	Birthday	Gender	Province	ZipCode	Height	Weight	Smoking	เบอร์โทรศัพท์มือถือ
นางสาว	นางสาว	ศิริณี		24/8/2511	หญิง	อุบลราชธานี	44534	180	62	<input checked="" type="checkbox"/>	(083) 556-7896
นาง	นาง	วาสนา	ประเสริฐ	2/6/2530	หญิง	นนทบุรี	34555	163	52	<input type="checkbox"/>	(083) 428-798
นางสาว	นางสาว	แก้ว	กลมเกลียว	2/6/2530	หญิง	กรุงเทพฯ	77665	155	48	<input checked="" type="checkbox"/>	(084) 347-8777

ภาพที่ 3.13 ตัวอย่างผลลัพธ์การกรองข้อมูลแบบเลือกรายการ

การกรอง record โดยเก็บไว้แต่เรคอร์ดที่ข้อมูลในช่อง Title ไม่เริ่มต้นด้วยคำว่า นาย ออกไปนั้น จะเหลือเรคอร์ดดังที่เห็นนี้ กด Toggle Filter เพื่อดูกลับไปกลับมา ระหว่างข้อมูลก่อนและหลังกรอง

การค้นหาข้อมูล

กดปุ่ม Find



ภาพที่ 3.14 ขั้นตอนการค้นหาข้อมูล

การกำหนดรายละเอียดในการค้นหา

- ช่อง Find What พิมพ์ค่าที่ต้องการค้นหา
- ช่อง Look In เลือกแหล่งค้นหา (Current Document, Current Field)
- ช่อง Match กำหนดความสัมพันธ์ของข้อมูลที่ต้องการค้นหา
 - Any Part of Field ข้อมูลที่ค้นเป็นส่วนหนึ่งของ field
 - Whole Field ข้อมูลที่ค้นเป็นทั้งหมดของ field
 - Start Field ข้อมูลที่ค้นเป็นค่าเริ่มต้นของ field
- ช่อง Search กำหนดทิศทางการค้นหา (All, Up, Down)
- ที่ Match Case กำหนดให้หาตรงตามที่พิมพ์เข้าไปเท่านั้น เช่น ถ้าหา MISS จะไม่เจอ record ที่เขียนว่า Miss

การค้นหาแบบเร็ว

ลำดับ	Title	Firstname	Lastname	Birthday	Gender	Province	ZipCode	Height	Weight	Smoking	เบอร์โทรศัพท์มือถือ
1	นาย	คำ	บุญมี	2/6/2530	ชาย	กรุงเทพฯ	11123	155	50	<input checked="" type="checkbox"/>	(087) 921-2525
5	นางสาว	บัวสาย	สวัสดิ์	24/8/2511	หญิง	อุดรธานี	44534	160	62	<input type="checkbox"/>	(083) 556-7896
6	นาย	สมชาย	มาจรีพันธ์	2/6/2530	ชาย	กรุงเทพฯ	11123	155	50	<input checked="" type="checkbox"/>	(083) 436-5879
7	นาย	สุธ	สมหวัง	2/6/2030	ชาย	ปัตตานี	34034	101	50	<input checked="" type="checkbox"/>	(087) 826-492
8	นาง	วาสนา	ประเสริฐ	2/6/2530	หญิง	นนทบุรี	34555	163	52	<input type="checkbox"/>	(083) 428-798
9	นางสาว	แก้ว	กมลเกียรติ	2/6/2530	หญิง	กรุงเทพฯ	77666	100	48	<input checked="" type="checkbox"/>	(084) 347-8777
*	(New)									<input type="checkbox"/>	

Record: 2 of 6 | No Filter | หญิง

↑
ให้ลองใส่คำว่า หญิง แล้วกด Enter ดู

ภาพที่ 3.15 ขั้นตอนการค้นหาข้อมูลแบบเร็ว

การคำนวณข้อมูลในตาราง

1. คลิกปุ่ม Totals

2. แถว Total จะปรากฏขึ้น

3. เลือกสิ่งที่ต้องการทำได้ตามต้องการ โดยข้อมูลแต่ละชนิดจะมีสิ่งที่ทำได้ไม่เหมือนกัน

Subject	Title	Firstname	Lastname	Birthday	Gender	Province	ZipCode	Height	Weight
1	นาย	คำ	บุญมี	2/6/2530	ชาย	กรุงเทพ	11123	155	50
5	นางสาว	บัวสาย	ศรีสวัสดิ์	24/8/2511	หญิง	อุบลราชธานี	44534	160	62
6	นาย	สมชาย	มากทรัพย์	2/6/2530	ชาย	กรุงเทพ	11123	155	50
7	นาย	สุข	สมหวัง	2/6/2530	ชาย	ปทุมธานี	34534	151	55
8	นาง	วาสนา	ประเสริฐ	2/6/2530	หญิง	นนทบุรี	34555	163	52
9	นางสาว	แก้ว	กลมเกลียว	2/6/2530	หญิง	กรุงเทพ	77665	155	48
* (New)									
Total									

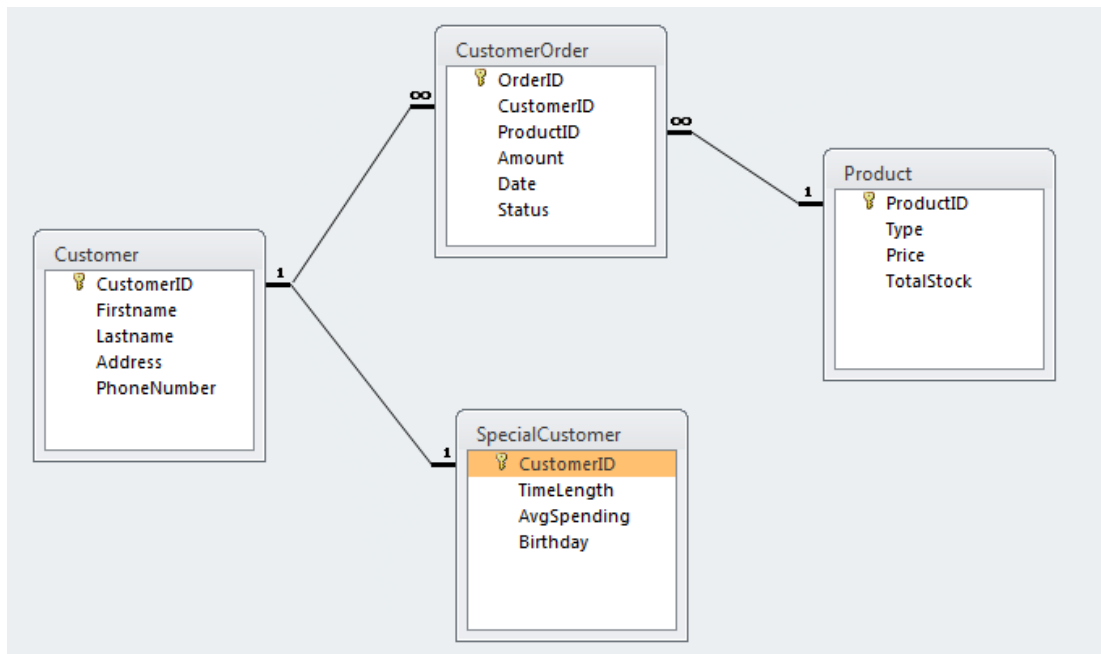
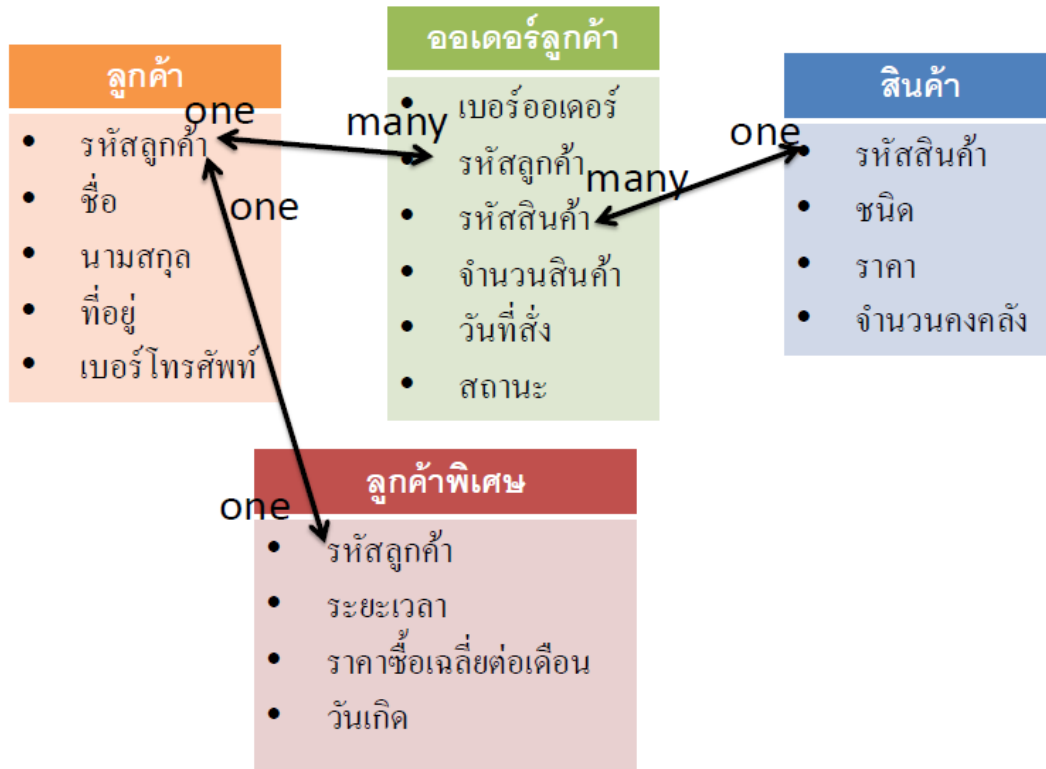
ภาพที่ 3.16 ขั้นตอนการคำนวณข้อมูลในตาราง

การสร้างความสัมพันธ์ (Relationship)

การสร้างฐานข้อมูลดีนั้น ส่วนมากจะมีมากกว่า 1 ตาราง เพราะในแต่ละงานที่เราต้องการเก็บข้อมูล ก็มักมีส่วนประกอบมากกว่า 1 ส่วน เช่น ข้อมูลโกดังเก็บสินค้า อาจจะมีข้อมูลส่วนของลูกค้า ส่วนของออเดอร์ ส่วนของสินค้า ซึ่งอาจแยกไปเป็นสินค้าขายส่ง ขายปลีก สินค้าลดราคา เป็นต้น ซึ่งผู้สร้างฐานข้อมูลต้องคิดให้รอบคอบตั้งแต่แรก ว่าในงานที่ต้องการเก็บข้อมูลนี้ มีข้อมูลที่แบ่งเป็นชนิดต่างๆ อะไรได้บ้าง โดยยึดว่าต้องแบ่งอย่างมีเอกภาพ ส่วนเดียวกันต้องมีเนื้อหาเป็นเรื่องเดียวกัน ในการสร้างฐานข้อมูล ข้อมูลแต่ละส่วน จะมีตารางเป็นของตัวเอง

ตัวอย่างความสัมพันธ์

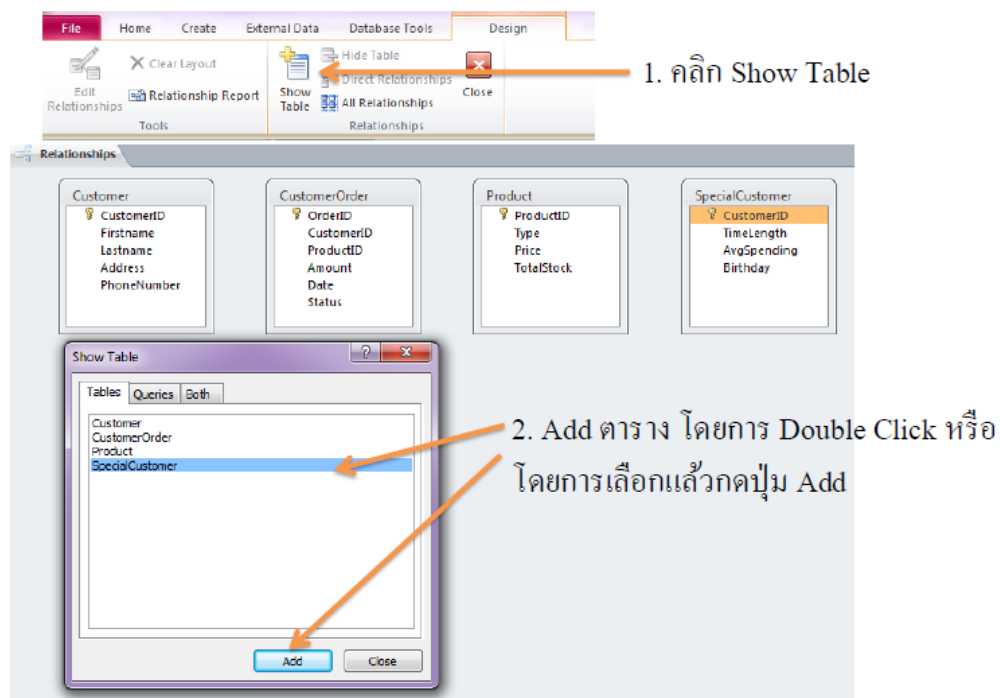
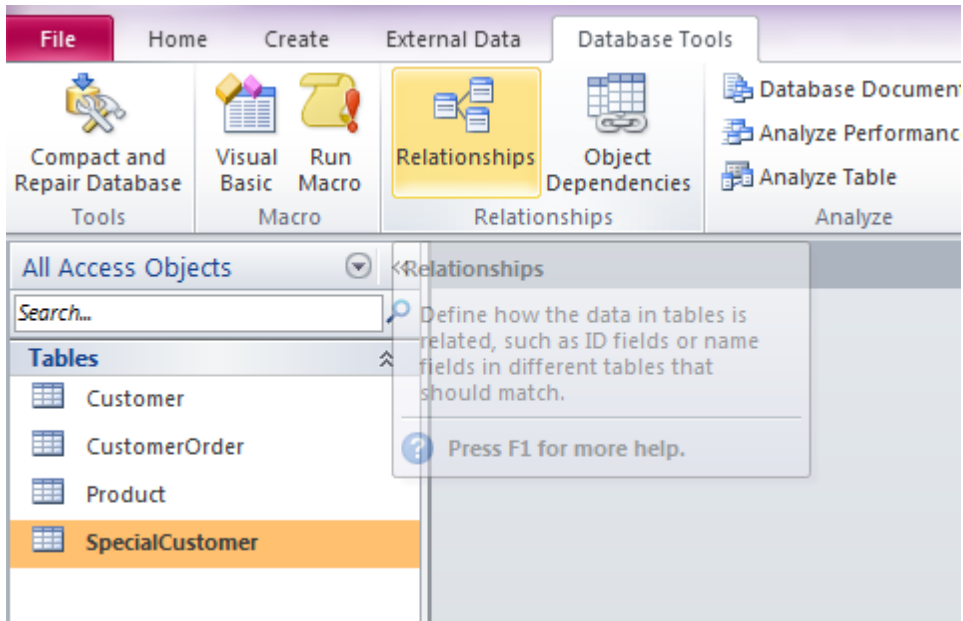
- ลูกค้า: ชื่อ, นามสกุล, ที่อยู่, เบอร์โทรศัพท์, รหัสลูกค้า
- สินค้า: รหัสสินค้า, ชนิด, ราคา, จำนวนคงคลัง
- ออเดอร์ลูกค้า: เบอร์ออเดอร์, รหัสลูกค้า, รหัสสินค้า, จำนวนสินค้า, วันที่สั่ง, สถานะ
- ลูกค้าพิเศษ: รหัสลูกค้า, ระยะเวลาที่เป็นลูกค้ามา, ราคาที่ซื้อเฉลี่ยต่อเดือน, วันเกิด



ภาพที่ 3.17 ตัวอย่างความสัมพันธ์ระหว่างตาราง

ตัวอย่างการสร้างความสัมพันธ์ของตาราง

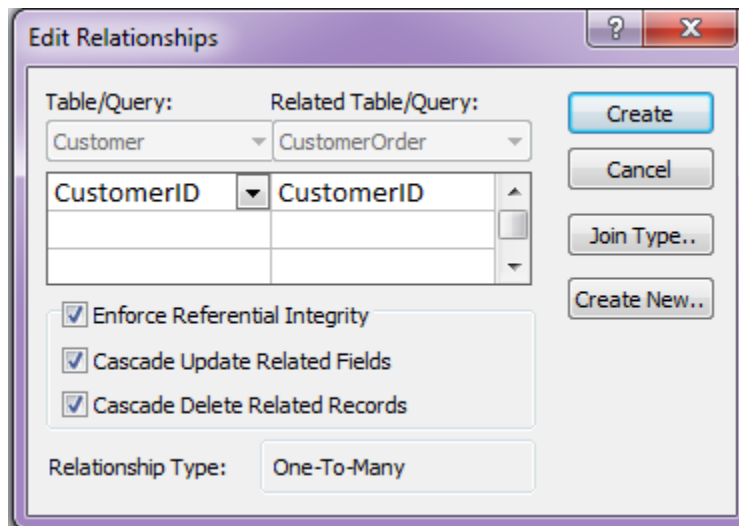
1. คลิกแท็บ Database Tools และคลิกปุ่ม Relationships



ภาพที่ 3.18 ขั้นตอนการความสัมพันธ์ระหว่างตาราง

2. ใช้ Mouse ลากให้เกิดความสัมพันธ์ระหว่าง CustomerID ในตาราง Customer และ CustomerOrder

3. จะมีหน้าจอขึ้นมาถาม โดย Access จะกำหนดชนิดความสัมพันธ์ให้โดยอัตโนมัติ
4. ให้เลือก Enforce Referential Integrity, Cascade Update Related Fields, และ Cascade Delete Related Fields เพื่อที่ว่า เมื่อใดก็ตามที่ record มีการเปลี่ยนแปลงที่ตารางใดตารางหนึ่ง ก็ให้ update กับตารางอื่นที่มีความสัมพันธ์ด้วย



ภาพที่ 3.19 หน้าจอ Edit Relationships

บทที่ 4

การสร้างแบบสอบถาม (Query)

การสร้างแบบสอบถาม (query) เป็นเครื่องมือที่ใช้ในการค้นหาข้อมูลตามเงื่อนไขที่กำหนด เรียงลำดับหรือจัดกลุ่มข้อมูลให้อยู่ในรูปแบบที่ต้องการ แสดงผลข้อมูลที่มาจกหลายตารางที่มีความสัมพันธ์กัน เปลี่ยนแปลงรูปแบบของการแสดงผล ปรับปรุงข้อมูลในตาราง คำนวณผลลัพธ์หรือสรุปผลข้อมูลจากตารางต่างๆ และสามารถกำหนดตัวแปรเพื่อรับค่าที่ใช้เป็นเงื่อนไขในการค้นหา คำนวณผลลัพธ์ หรือวิเคราะห์สรุปผลข้อมูลได้ด้วย

มุมมองของแบบสอบถาม

มุมมองของแบบสอบถาม มี 5 ลักษณะ คือ

1. **มุมมองออกแบบ (design)** เป็นมุมมองที่ใช้ในการออกแบบและสร้างแบบสอบถามซึ่งประกอบด้วยเงื่อนไขและกฎเกณฑ์ต่างๆ ที่จะนำไปใช้ค้นหาหรือแสดงข้อมูลที่ต้องการ
2. **มุมมองแผ่นข้อมูล (datasheet)** เป็นมุมมองที่ใช้แสดงข้อมูลทั้งหมดที่ค้นหาได้ตามเงื่อนไขที่กำหนดไว้ใน แบบสอบถาม โดยแสดงในลักษณะ datasheet เหมือนกับ table
3. **มุมมอง SQL** เป็นมุมมองที่ใช้สร้างหรือแสดงเงื่อนไขที่กำหนดไว้ในแบบสอบถามด้วยรูปแบบของภาษาเอสคิวแอล (Structured Query Language : SQL)
4. **มุมมอง pivot table** เป็นมุมมองที่ใช้แสดงข้อมูลที่ได้จากการรันแบบสอบถามตามเงื่อนไขที่กำหนดไว้ โดยแสดงในรูปของตารางแจกแจงรายละเอียดและสรุปผลข้อมูล
5. **มุมมอง pivot chart** เป็นมุมมองที่ใช้แสดงข้อมูลที่ได้จากการรันแบบสอบถามโดยวิเคราะห์หรือสรุปผล ในรูปของแผนภูมิหรือ chart

ประเภทของแบบสอบถาม

Access แบ่งแบบสอบถามเป็น 5 ประเภท คือ

1. **Select Query** สอบถามข้อมูลจาก 1 หรือหลายๆ ตาราง
2. **Crosstab Query** สอบถามข้อมูลและแสดงผลลัพธ์แบบ 2 มิติ โดยสลับข้อมูลระหว่างแนวแถวและแนวคอลัมน์
3. **Action Query**
 - **Make-Table Query** สร้างตารางใหม่ ซึ่งประกอบด้วย field ที่เลือกจาก Table หรือ Query อื่นที่สร้างไว้
 - **Update Query** แก้ไขข้อมูลในตารางนั้นให้มีค่าตามที่กำหนด
 - **Append Query** เพิ่ม record หรือกลุ่มของ record ใหม่ต่อท้าย record สุดท้ายของตาราง
 - **Delete Query** ลบ record หรือกลุ่มของ record ออกจากตาราง
4. **Parameter Query** เป็น Query ที่ใช้ใส่ค่าพารามิเตอร์ได้ระหว่างการประมวลผล เพื่อนำไปใช้ค้นหาข้อมูลหรือคำนวณค่า เช่น ใส่อัตราการเปลี่ยนแปลงต้นทุนและราคาสินค้า เพื่อคำนวณหากำไร ณ อัตราต่างๆ
5. **SQL Query** สร้างด้วยภาษา SQL ซึ่งเป็นภาษามาตรฐานที่ใช้จัดการฐานข้อมูล
 - **Union Query** รวมข้อมูลใน field ชนิดเดียวกันจากหลายๆ ตาราง มาไว้ใน 1 field ของ 1 ตาราง
 - **Pass-Through Query** ส่งคำสั่งไปยัง ODBC Database Server โดยใช้คำสั่งภาษา SQL ของ Server นั้น
 - **Data Definition Query** สร้างหรือเปลี่ยนแปลงโครงสร้างตารางในฐานข้อมูล (คำสั่ง SQL ที่ทำงานในลักษณะนี้จะมีชื่อเฉพาะว่า Data Definition Language (DDL))

วิธีการสร้างแบบสอบถาม

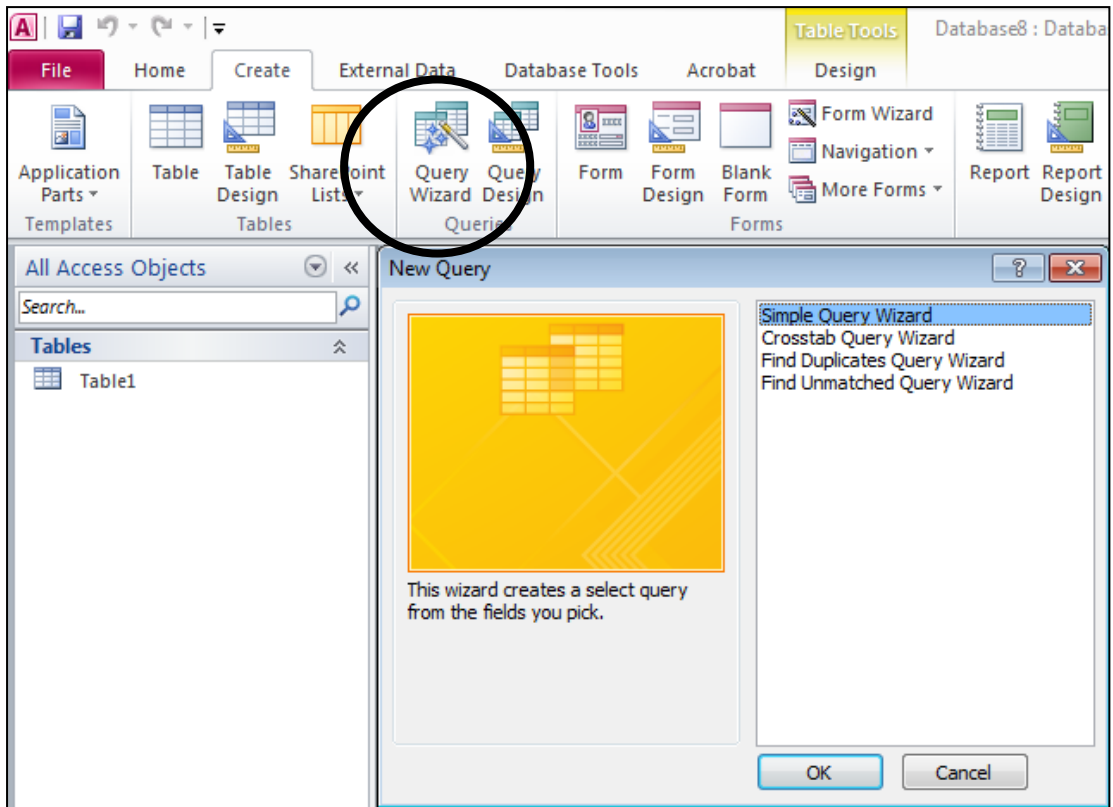
การสร้างแบบสอบถาม ทำได้ 3 วิธี คือ

1. **สร้างแบบสอบถามโดยใช้ตัวช่วยสร้าง (Query Wizard)** เป็นวิธีการสร้างแบบสอบถาม แบบง่ายๆ และรวดเร็ว โดยตอบคำถามและให้ข้อมูลในการสร้าง แบบสอบถามผ่านทางไดอะล็อกบ็อกซ์ที่โปรแกรมไมโครซอฟต์แอกเซสเตรียมไว้ให้ทำตามอย่างเป็นลำดับขั้นตอน
2. **สร้างแบบสอบถามในมุมมองออกแบบ (Design View)** เป็นวิธีการสร้างแบบสอบถาม ด้วยตนเอง โดยผู้สร้างจะเป็นผู้กำหนดรายละเอียดต่างๆ เองทั้งหมด ซึ่งทำให้ได้แบบสอบถามที่เหมาะสมและตรงกับความต้องการมากที่สุด แต่ผู้สร้างต้องมีความเชี่ยวชาญในการสร้างและออกแบบแบบสอบถามมากกว่าวิธีแรก
3. **สร้างแบบสอบถามในมุมมอง SQL** สร้างและแก้ไขแบบสอบถามด้วยภาษา SQL ใช้กับแบบสอบถามบางประเภทที่สร้างด้วย Wizard หรือมุมมอง Design ไม่ได้ เช่น Union Query และ Pass-Through Query เป็นต้น

สร้างแบบสอบถามโดยใช้ตัวช่วยสร้าง

เครื่องมือช่วยสร้าง Query มี 4 ประเภท คือ

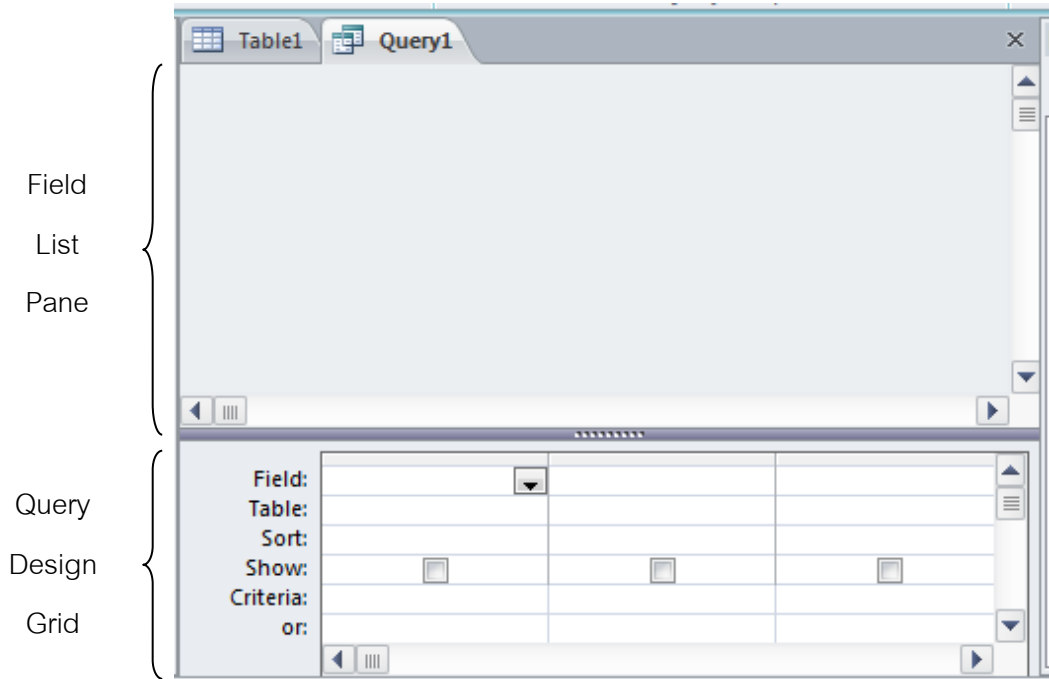
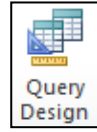
- 1.1 Simple Query Wizard สร้าง Select Query ค้นหาและประมวลผลข้อมูล โดยเลือกแสดงผลลัพธ์แบบรายละเอียดหรือผลสรุปได้
- 1.2 Crosstab Query Wizard สร้าง Crosstab Query ค้นหาและประมวลผลข้อมูล โดยแสดงตารางแจกแจงข้อมูลที่สลับแนวแถวและแนวคอลัมน์พร้อมค่าผลสรุปของข้อมูลนั้น
- 1.3 Find Duplicate Query Wizard สร้าง Select Query ค้นหาว่ามี record ใดบ้างที่มีข้อมูลซ้ำกันใน field ที่กำหนด
- 1.4 Find Unduplicated Query Wizard สร้าง Select Query ค้นหาข้อมูลในตารางหนึ่ง โดยแสดงเฉพาะ record ที่ไม่สามารถจับคู่กับ record ในตารางอื่นที่สัมพันธ์กัน



ภาพที่ 4.1 เครื่องมือช่วยสร้าง Query

การสร้างแบบสอบถามด้วยมุมมองการออกแบบ

1. ส่วนประกอบในมุมมองออกแบบ



ภาพที่ 4.2 ส่วนประกอบในมุมมองออกแบบ

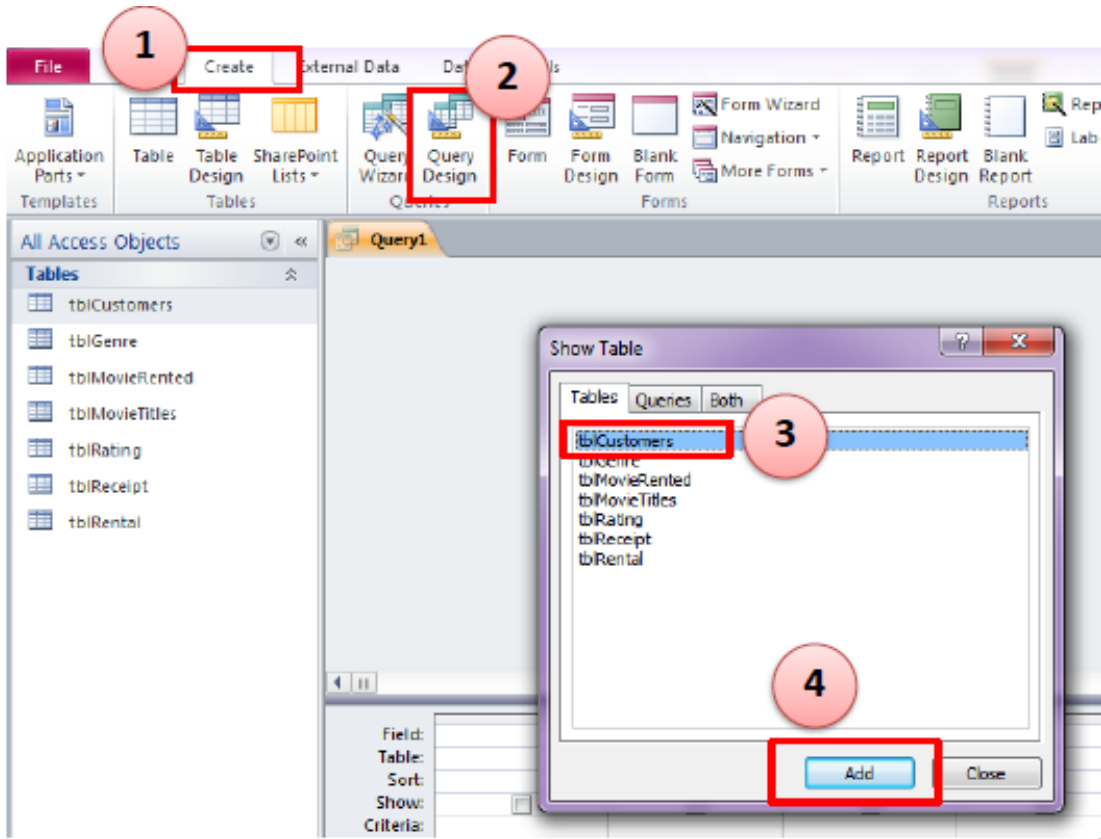
1.1 Field List Pane แสดงชื่อ Table / Query ที่เป็นแหล่งข้อมูลของ Query แสดงชื่อ field และเส้นความสัมพันธ์ระหว่างตาราง

1.2 Query Design Grid ประกอบด้วย

- Field: ใส่ชื่อ Field ที่จะใช้สอบถามข้อมูล
- Table: ใส่ชื่อ Table / Query ที่เป็นแหล่งข้อมูล
- Total: ใช้จัดกลุ่มและคำนวณค่าผลสรุป (จะแสดงแถบนี้เมื่อคลิกปุ่ม Totals ในแท็บ Design)
- Sort: กำหนดรูปแบบการเรียงข้อมูล โดย Ascending จะเรียงจากน้อยไปมาก ส่วน Descending จะเรียงจากมากไปน้อย และ not Sorted คือ ไม่ต้องจัดเรียง
- Show: ซ่อน field ที่ไม่ให้แสดงค่าในตารางผลลัพธ์ โดยคลิกเอาเครื่องหมาย ✓ ในช่องสี่เหลี่ยมออก
- Criteria: ใส่เงื่อนไขที่ใช้สอบถามข้อมูล

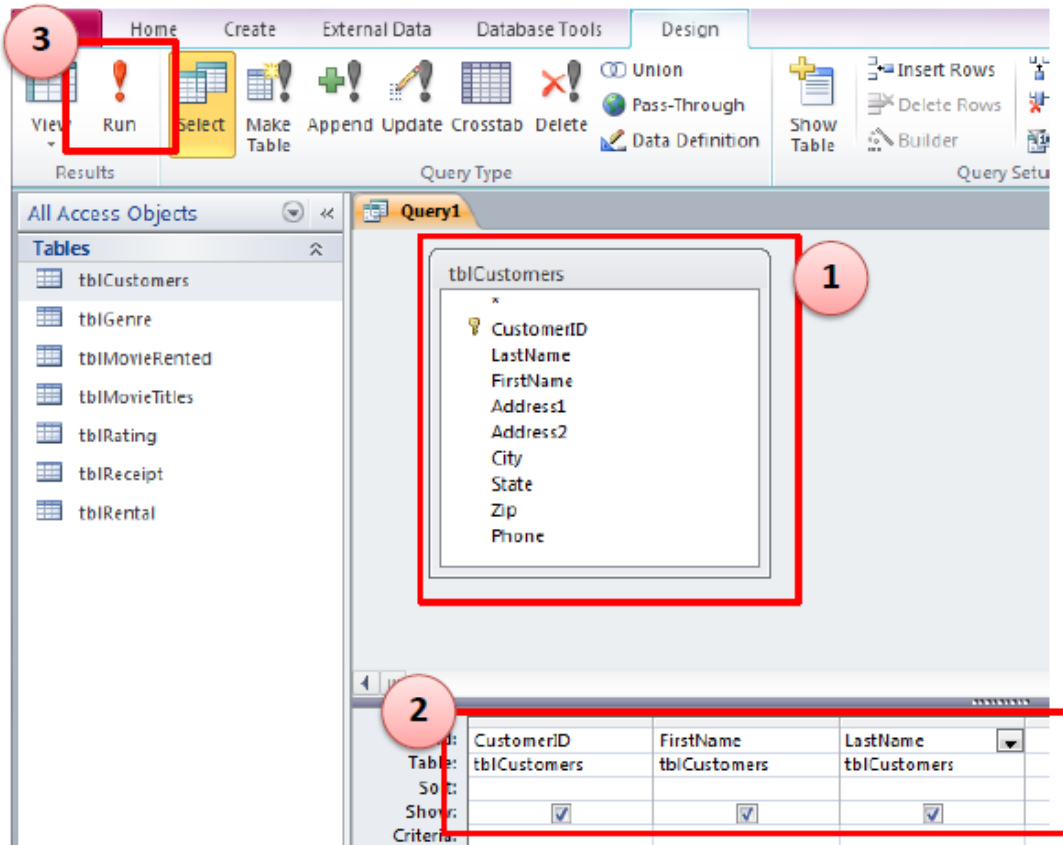
2. วิธีการสร้างแบบสอบถามด้วย Query Design

การสร้างแบบสอบถามด้วยมุมมองการออกแบบเป็นการสอบถามข้อมูลด้วยการออกแบบด้วยตนเอง ซึ่งมีขั้นตอนดังนี้



ภาพที่ 4.3 ขั้นตอนการเริ่มต้นสร้างแบบสอบถามด้วย Query Design

หากต้องการสอบถามตารางลูกค้า (tblCustomers) โดยถามถึง รหัสลูกค้า (CustomerID), ชื่อ (FirstName), และนามสกุล (LastName) เท่านั้น มีขั้นตอนและผลลัพธ์ดังนี้



คำตอบที่ได้

Customer	FirstNan	LastName
1	James	Accardo
2	James	Adams
3	Lisa	Alexander
4	Beverly	Allen
5	Timothy	Allen
6	Velma	Allen
7	Michael	Allen
8	Catherine	Andersen
9	Dawne	Anger
10	Jeffrey	Archer

ภาพที่ 4.4 ขั้นตอนและผลลัพธ์การสร้างแบบสอบถามด้วย Query Design

การบันทึกผลของแบบสอบถามข้อมูล

- หลังจากผลของแบบสอบถามได้แสดงออกมาในรูปแบบ Datasheet หรือตารางแล้ว ให้กดปุ่ม Save ที่ด้านมุมบนซ้าย
- Access จะให้ตั้งชื่อของแบบสอบถามนี้ สมมติตั้งว่า myQuery1
- หลังจาก Save แล้ว จะปรากฏหน้าจอดังนี้

Customer	FirstName	City	State
1	James	Commerce Township	MI
2	James	Adamsville	MI
3	Lisa	MyTown	MI
4	Beverly	MyTown	MI
5	Timothy	MyTown	MI
6	Velma	MyTown	MI
7	Michael	MyTown	MI
8	Catherine	MyTown	MI
9	Dawne	Commerce Township	MI
10	Jeffrey	Adamsville	MI
11	Elizabeth	Adamsville	MI
12	John	MyTown	MI
13	Mary	MyTown	MI

ภาพที่ 4.5 หน้าจอการบันทึกผลของแบบสอบถามข้อมูล

การสร้างแบบสอบถามข้อมูลจากตารางมากกว่า 1 ตาราง

1 เลือกสอบถามจากตาราง tblMovieRented และ tblMovieTitles

2 สร้างความสัมพันธ์

3 เลือกสิ่งที่ต้องการถามถึง

Field:	MovieID	DateRented	Movie	Year
Table:	tblMovieRented	tblMovieRented	tblMovieTitles	tblMovieTitles
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				

คำตอบที่ได้

MovieID	DateRented	Movie	Year
198	24/9/2550	Lord of the Rings: The Fellowship of the Ring (2001)	2001
200	24/9/2550	Lord of the Rings: The Two Towers (2002)	2002
199	24/9/2550	Lord of the Rings: The Return of the King (2003)	2003
352	24/9/2550	Women in Film (2001)	2001
294	24/9/2550	Shawshank Redemption (1994)	1994
42	24/9/2550	Aviator, The (2005)	2005
100	24/9/2550	Fantastic Four (2005)	2005
125	30/9/2550	General, The (1998)	1998
342	30/9/2550	Where the Truth Lies (2005)	2005
203	30/9/2550	Loverboy (2005)	2005
*			

ภาพที่ 4.6 ขั้นตอนการสร้างแบบสอบถามข้อมูลจากตารางมากกว่า 1 ตาราง

การแก้ไขแบบสอบถามข้อมูล

- ให้บันทึกแบบสอบถามที่เพิ่งทำเสร็จ (แบบมีความสัมพันธ์) ด้วยชื่อว่า myQuery2 หลังจากนั้นให้เปิดโปรแกรม MS Access
- เปิดโปรแกรม MS Access ใหม่ และเปิดฐานข้อมูล VDO ใหม่อีกครั้ง
- คลิก myQuery2 ในหน้าต่างนำทางด้านซ้ายมือเพื่อเปิดแบบสอบถามนี้ขึ้นมาอีกครั้ง โดยครั้งนี้มีจุดประสงค์เพื่อทำการแก้ไขแบบสอบถามข้อมูล
- เข้า Design View

1 เพิ่มตาราง tblCustomers โดยที่ความสัมพันธ์ได้เคยถูกสร้างไว้แล้วตอน Design ตารางครั้งแรก

2 เลือกสิ่งที่ต้องการถามถึง

Field	MovieID	DateRented	Movie	Year	FirstName	LastName
Table:	tblMovieRented	tblMovieRented	tblMovieTitles	tblMovieTitles	tblCustomers	tblCustomers
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:						

MovieID	DateRented	Movie	Year	FirstName	LastName
198	24/9/2550	Lord of the Rings: The Fellowship of the Ring (2001)	2001	Timothy	Allen
200	24/9/2550	Lord of the Rings: The Two Towers (2002)	2002	Timothy	Allen
199	24/9/2550	Lord of the Rings: The Return of the King (2003)	2003	Timothy	Allen
352	24/9/2550	Women in Film (2001)	2001	Timothy	Allen
294	24/9/2550	Shawshank Redemption (1994)	1994	Deeter	Poohbah
42	24/9/2550	Aviator, The (2005)	2005	Deeter	Poohbah

ภาพที่ 4.7 ขั้นตอนการแก้ไขแบบสอบถามข้อมูล

คำตอบที่ได้: แสดงรายการชื่อ ปี และรหัสของหนังที่มีการเช่า รวมถึงวันที่เช่า ชื่อและนามสกุลของลูกค้า ทั้งหมด

* เมื่อได้คำตอบแล้วให้ Save ไปด้วย

การกำหนดเงื่อนไขในการค้นหาข้อมูล

- ในการกำหนดเงื่อนไขในการค้นหาข้อมูลนั้น เราสามารถกำหนดได้ที่ช่อง criteria ใน Design View หลังจากที่เราสร้างแบบสอบถามแล้ว
- เราสามารถกำหนดเงื่อนไขต่างๆที่ต้องการได้ โดยการใช้ operator ต่างๆ ซึ่งได้แก่
 - 1) โอเปอเรเตอร์ทางคณิตศาสตร์ (เช่น บวก ลบ คูณ หาร เป็นต้น)
 - 2) โอเปอเรเตอร์ทางการเปรียบเทียบและกำหนดค่า (เช่น เท่ากับ น้อยกว่า มากกว่า เป็นต้น)
 - 3) โอเปอเรเตอร์ทางตรรกศาสตร์ (เช่น และ หรือ ไม่ เป็นต้น)
 - 4) โอเปอเรเตอร์เกี่ยวกับการเชื่อมข้อความ

โอเปอเรเตอร์ (Operator) ทางคณิตศาสตร์

Operator	คำอธิบาย
+	เครื่องหมายบวก เช่น $1 + 1 = 2$
-	เครื่องหมายลบ เช่น $2 - 1 = 1$
*	เครื่องหมายคูณเช่น $3 + 2 = 5$
/	เครื่องหมายหารเช่น $5/2 = 2.5$
\	เครื่องหมายหารแบบเอาแต่จำนวนเต็ม ไม่สนใจเศษ เช่น $5 \setminus 2 = 2$
Mod	เครื่องหมายหารแบบเอาแต่เศษที่เหลือ เช่น $5 \text{ Mod } 2 = 1$
^	เครื่องหมายยกกำลัง เช่น $2^3 = 6$

โอเปอเรเตอร์ (Operator) ทางการเปรียบเทียบและกำหนดค่า

Operator	คำอธิบาย
<	น้อยกว่า
<=	น้อยกว่าหรือเท่ากับ
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
=	เท่ากับ
<>	ไม่เท่ากับ

Operator	คำอธิบาย
Between	ใช้ในการตรวจสอบว่ามีค่าอยู่ในระหว่าง 2 ค่าที่กำหนดมาหรือไม่ เช่น Between 10000 And 20000
Like	ใช้ในการตรวจสอบว่าอักขระนั้นตรงกับที่กำหนดหรือไม่ โดยที่ -* แทนอักขระใดๆ ก็ได้ -? แทนอักขระ 1 ตัว -# แทนตัวเลข 1 ตัว -! แทนการยกเว้นอักขระที่อยู่หลังเครื่องหมายนี้ -[] แทนช่วงตัวอักษรหรือตัวเลข เช่น Like "สม*"
In	ใช้ในการกำหนดว่ามีค่าใดตรงกับค่าในรายการหรือไม่ เช่น In ("บัญชี", "สารสนเทศ")

โอเปอเรเตอร์ (Operator) ทางตรรกศาสตร์

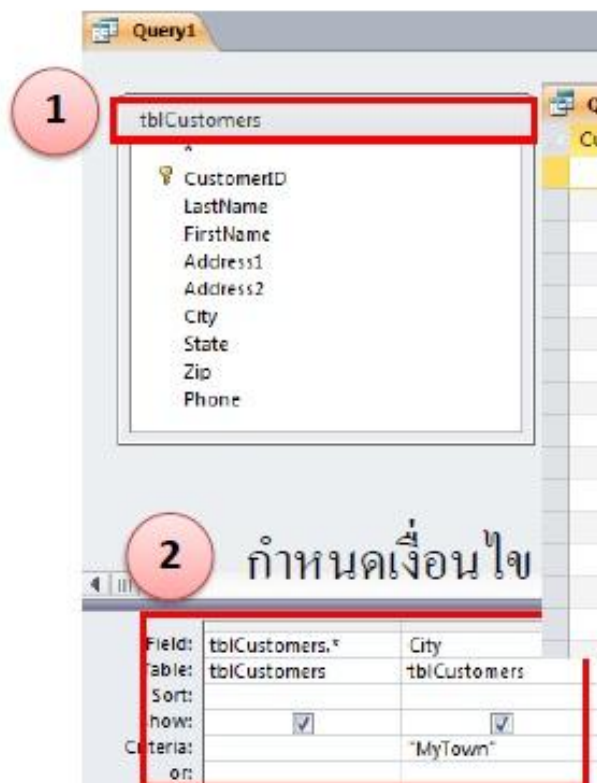
Operator	คำอธิบาย
And	ให้ค่าจริงต่อเมื่อทั้ง 2 ค่าเป็นจริง
Or	ให้ค่าจริงเมื่อค่าใดค่าหนึ่งเป็นจริง
Xor	ให้ค่าจริงเมื่อเป็นจริงเพียง 1 ค่า
Not	ให้ค่าเป็นตรงกันข้าม

โอเปอเรเตอร์ (Operator) ในการเชื่อมต่อข้อความ

Operator	คำอธิบาย
&	ใช้ในการเชื่อมต่อข้อความ เช่น [FirstName]&" "&[LastName]

ตัวอย่างการกำหนดเงื่อนไข

ตัวอย่างที่ 1 เลือกทุกคอลัมน์ของตาราง tblCustomerID ที่ City เท่ากับ “MyTown”



คำตอบที่ได้: สังเกตว่า city เป็น MyTown เท่านั้น

CustomerID	LastName	FirstNan	Address1	Address2	tblCustomers.City	State	Z
3	Alexander	Lisa	2478 Marjorie Lane		MyTown	MI	485
4	Allen	Beverly	914 Mckeighan		MyTown	MI	485
5	Allen	Timothy	3015 Grandblanc Rd		MyTown	MI	485
6	Allen	Velma	4101 Sheridan Rd		MyTown	MI	485
7	Allen	Michael	12345		MyTown	MI	485
8	Andersen	Catherine	3729 Palmer		MyTown	MI	485
12	Assaf	John			MyTown	MI	485
13	Attarian	Mary	2406 Broadway		MyTown	MI	485
14	Averill	Patricia	2163 Mackinaw		MyTown	MI	485
15	Bachman	Madison	2082 Five Lakes Rd		MyTown	MI	485
17	Bailey	Gerald	603 Peirson St		MyTown	MI	485
19	Bair	Danielle	5371 Worchester Dr		MyTown	MI	485
20	Baker	Corey	4234 Cascade Ct		MyTown	MI	485
21	Balcom	Peggy	1397 Lake Nepessing Rd		MyTown	MI	485
23	Bancroft	Patsy	4481 Rainbow Lane		MyTown	MI	485

ภาพที่ 4.8 ขั้นตอนและผลลัพธ์การกำหนดเงื่อนไขของตัวอย่างที่ 1

ตัวอย่างที่ 2 เลือกทุกคอลัมน์ของตาราง tblCustomerID ที่ LastName ขึ้นต้นด้วย Ba

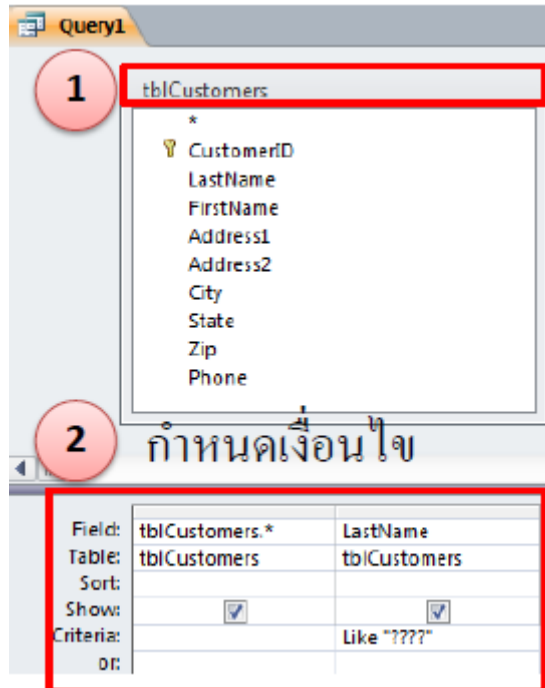


คำตอบที่ได้: สังกัด LastName

CustomerID	tblCustomers.LastName	FirstName	Address1	Address2
15	Bachman	Madison	2082 Five Lakes Rd	MyTown
16	Bachman	Brandon	8387 Neff	Adams*
17	Bailey	Gerald	603 Peirson St	MyTown
18	Bailey	Bryan	3314 Dillon Rd	Adams*
19	Bair	Danielle	5371 Worchester Dr	MyTown
20	Baker	Corey	4234 Cascade Ct	MyTown
21	Balcom	Peggy	1397 Lake Nepessing Rd	MyTown
22	Balyeat	Ellen	1016 Willowood	Adams*
23	Bancroft	Patsy	4481 Rainbow Lane	MyTown
24	Barker	Jamilda	G 2381 Pearl Ann	MyTown
25	Barnum	Sandra	5038 N Seymour Rd	Comme
26	Barrett	Raymond	One Citizens Bank Bldg	MyTown
27	Barroso	Paula	5410 Maple Park Dr Apt 6	Adams*
28	Barry	Denise	2110 Briar Lane	Adams*
29	Baryo	Dawn	1315 G	MyTown
30	Baryo I	James	6131 Woodmoor Dr	Adams*
31	Baryo II	Dave	201 Harriet Apt 1	Adams*

ภาพที่ 4.9 ขั้นตอนและผลลัพธ์การกำหนดเงื่อนไขของตัวอย่างที่ 2

ตัวอย่างที่ 3 เลือกทุกคอลัมน์ของตาราง tblCustomerID ที่ LastName มีเพียง 4 อักขระ

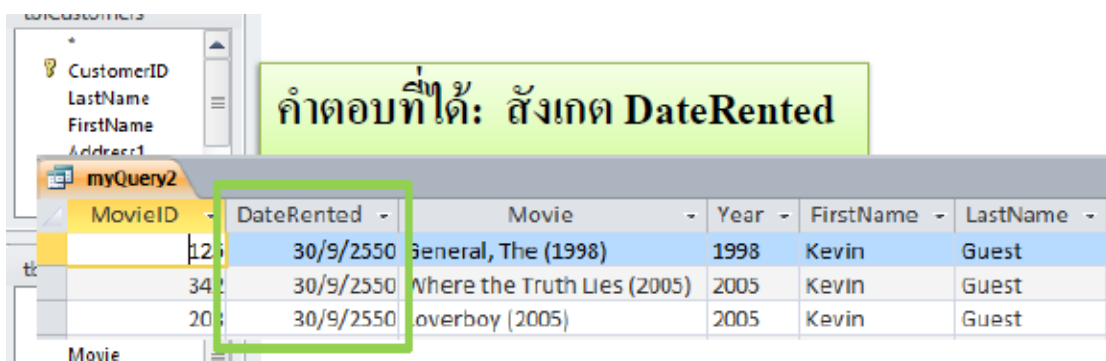
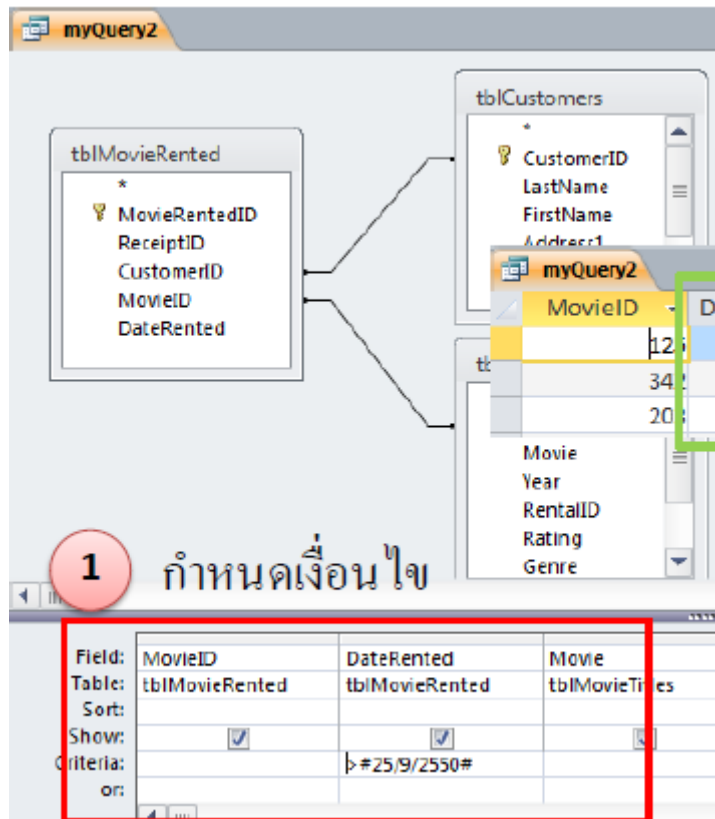


คำตอบที่ได้: สังกัด LastName

CustomerID	tblCustomers.LastName	FirstName	Address1	Address2	City
1	Bair	Danielle	5371 Worchester Dr		MyTown
3	Beck	Noreen	6262 Richfield Rd		MyTown
4	Bell	Stephen	6262 Richfield Rd		MyTown
6	Bosh	Justin	6344 Squire Lake Dr		MyTown
6	Boys	Vicki	G 3073 West Dayton		Adamsvi
9	Bush	Sandy	1114 Ida Ave		MyTown
9	Bush	Wendy	2171 E Francis Rd		MyTown
9	Bush	Pamela	9219 Beecher Rd		Commer
9	Buss	Robert	5144 Sunburst Ct		MyTown
10	Carr	Louis	4163 Manor Dr		Adamsvi
11	Cole	Harold	5241 E Coldwater Rd Lot 14		MyTown
12	Cole	George	11442		Adamsvi
12	Cole	Diane	10252 W Stanley Rd		Commer
12	Cook	Gail	342 North Dr		MyTown
12	Cook	Cynthia	2439 W N Union Rd		MyTown
12	Coon	Nathan	4511 Pendragon		MyTown
13	Cord	Doug	2465 Eaastwood Dr		MyTown
16	Deal	Raynetta	3353 Gernada		MyTown

ภาพที่ 4.10 ขั้นตอนและผลลัพธ์การกำหนดเงื่อนไขของตัวอย่างที่ 3

ตัวอย่างที่ 4 เปิด myQuery2 ออกมา คงไว้เหมือนเดิมทุกอย่าง เว้นแต่ว่าต้องการรายการที่เช่าหนังสือหลังจากวันที่ 25 กันยายน 2550 เท่านั้น



ภาพที่ 4.11 ขั้นตอนและผลลัพธ์การกำหนดเงื่อนไขของตัวอย่างที่ 4

ตัวอย่างที่ 5 จงค้นหาข้อมูลดังนี้ ชื่อลูกค้า รหัสของหนังที่ลูกค้าคนนั้นเช่าไป และชื่อหนัง โดยกาหนดว่า หนังต้องออกมาในระหว่างปี 1990 – 2000 หรือ มากกว่า 2002 เท่านั้น

1

2 กำหนดเงื่อนไข

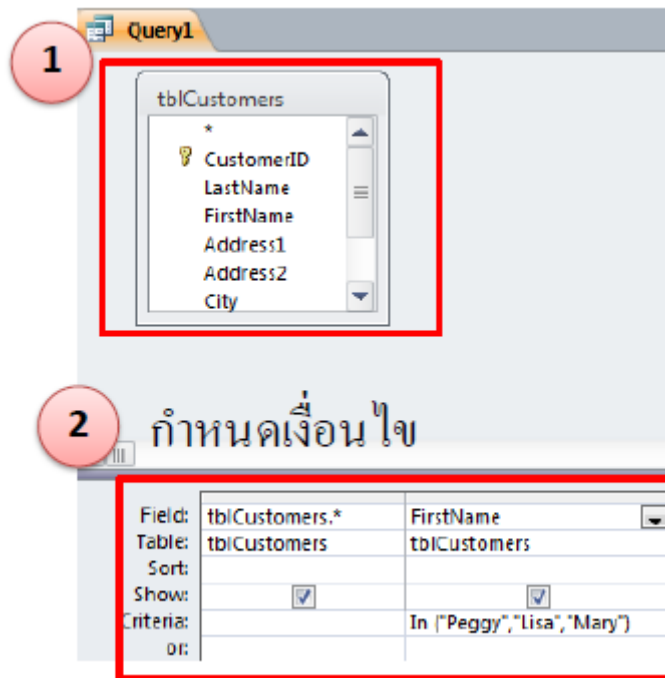
Field:	FirstName	MovieID	Movie	Year
Table:	tblCustomers	tblMovieRented	tblMovieTitles	tblMovieTitles
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				Between "1990" And "2000"
on:				> "2002"

คำตอบที่ได้: สังกศปี

FirstNan	MovieID	Movie	Year
Timothy	199	Lord of the Rings: The Return of the King	2003
Deeter	294	Shawshank Redemption (1994)	1994
Deeter	42	Aviator, The (2005)	2005
Deeter	100	Fantastic Four (2005)	2005
Kevin	125	General, The (1998)	1998
Kevin	342	Where the Truth Lies (2005)	2005
Kevin	203	Loverboy (2005)	2005

ภาพที่ 4.12 ขั้นตอนและผลลัพธ์การกำหนดเงื่อนไขของตัวอย่างที่ 5

ตัวอย่างที่ 6 จงค้นหาข้อมูลทั้งหมดจากตาราง tblCustomers โดยมีเงื่อนไขว่า ลูกค้าต้องมีชื่อจริงว่า Lisa, Peggy, หรือ Mary เท่านั้น



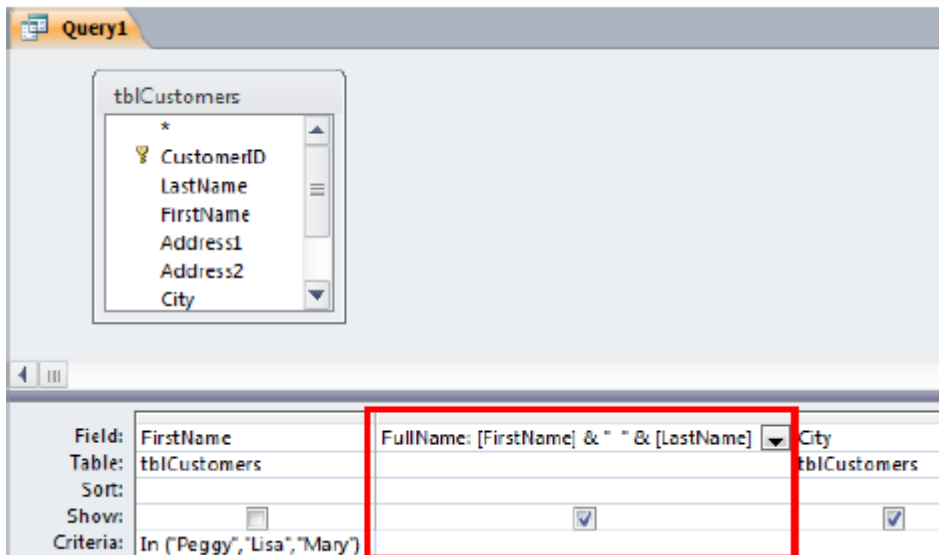
คำตอบที่ได้: สังกัดชื่อ

CustomerID	LastName	tblCusto	Address1	Address2	CI
3	Alexander	Lisa	2478 Marjorie Lane		MyTown
13	Attarian	Mary	2406 Broadway		MyTown
21	Balcom	Peggy	1397 Lake Nepessing Rd		MyTown
162	Delbridge	Lisa	2507 Covert Rd		Commer
247	Gitre	Mary	9500 Fuller Dr		Adamsvil
302	Hodo	Mary	540 Oakbrook Circle	4112	MyTown
324	Isaac	Mary	832 Moore St		MyTown
412	Lindsay	Mary	5337 Westchester		MyTOWN
473	Miller	Lisa	4350 Warbler		MyTown
492	Morgan	Mary	1022 E Alma St		MyTown
516	Nolfs	Mary	10418 Richfield Road		Adamsvil
581	Raslich	Lisa	9337 W Stanley Rd		MyTown
680	Stevens	Peggy	4342 Falcon Dr		MyTown
704	Tillman	Lisa	399 Jackson Circle		Commer
707	Towery	Mary	2333 Valley Vista		Adamsvil
711	Truss	Lisa	14130 Landingsway		Adamsvil

ภาพที่ 4.13 ขั้นตอนและผลลัพธ์การกำหนดเงื่อนไขของตัวอย่างที่ 6

การสร้างฟิลด์ใหม่จากการรวมฟิลด์

สามารถทำได้โดยการเติมฟิลด์ใหม่ที่ต้องการในรูปแบบชื่อฟิลด์ใหม่:
[ฟิลด์ที่ 1]&[ฟิลด์ที่ 2]& ... เช่น



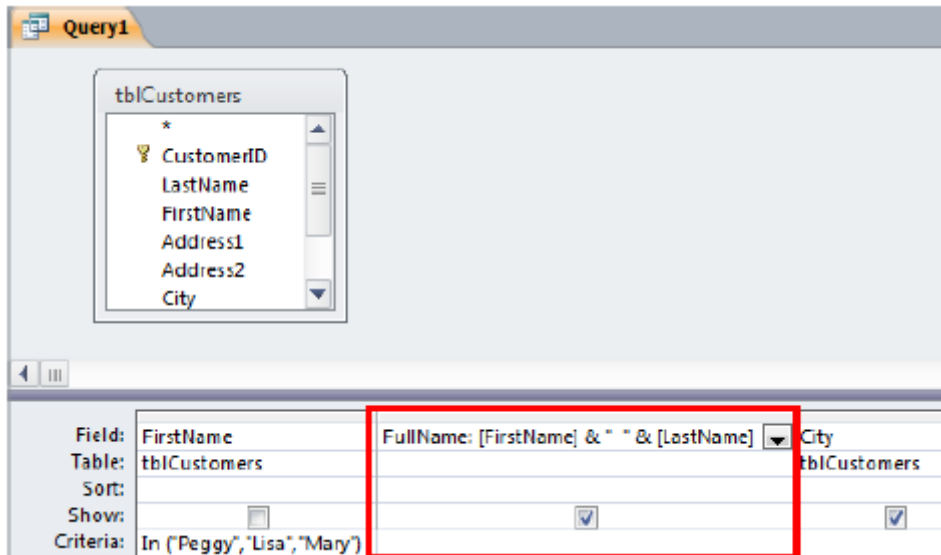
คำตอบที่ได้:

FullName	City
Lisa Alexander	MyTown
Mary Attarian	MyTown
Peggy Balcom	MyTown
Lisa Delbridge	Commerce Township
Mary Gitre	Adamsville
Mary Hodo	MyTown
Mary Isaac	MyTown
Mary Lindsay	MyTown
Lisa Miller	MyTown
Mary Morgan	MyTown
Mary Nolfs	Adamsville
Lisa Raslich	MyTown
Peggy Stevens	MyTown
Lisa Tillman	Commerce Township
Mary Towery	Adamsville
Lisa Truss	Adamsville

ภาพที่ 4.14 ขั้นตอนและผลลัพธ์การสร้างฟิลด์ใหม่จากการรวมฟิลด์

การเรียงฟิลด์ (Sort)

- ใช้ตัวอย่างที่แล้วแต่จะเรียงจากตัวอักษรก่อนไปหลัง หรือ Ascending
- Descending = จากหลังมาหน้า (ถ้าเป็นตัวเลข มากกว่า ไปน้อยกว่า)



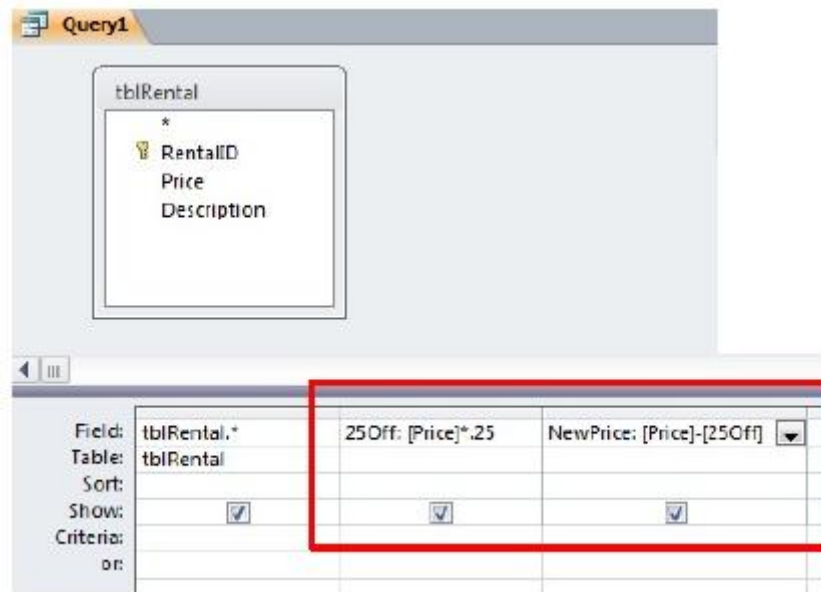
คำตอบที่ได้:

FullName	City
Lisa Alexander	MyTown
Mary Attarian	MyTown
Peggy Balcom	MyTown
Lisa Delbridge	Commerce Township
Mary Gitre	Adamsville
Mary Hodo	MyTown
Mary Isaac	MyTown
Mary Lindsay	MyTown
Lisa Miller	MyTown
Mary Morgan	MyTown
Mary Noffs	Adamsville
Lisa Raslich	MyTown
Peggy Stevens	MyTown
Lisa Tillman	Commerce Township
Mary Towery	Adamsville
Lisa Truss	Adamsville

ภาพที่ 4.15 ขั้นตอนและผลลัพธ์การเรียงฟิลด์

การสร้างฟิลด์คำนวณ

- เราสามารถสร้างฟิลด์ใหม่เป็นฟิลด์คำนวณคิดเลขให้เราได้ โดยใช้รูปแบบดังนี้ ชื่อฟิลด์ใหม่: สูตร
- ตัวอย่างเช่น จากตาราง tblRental จงหาว่าถ้าลดราคา 25% จะลดเท่าไร และราคาหลังลดแล้วเท่ากับเท่าไร



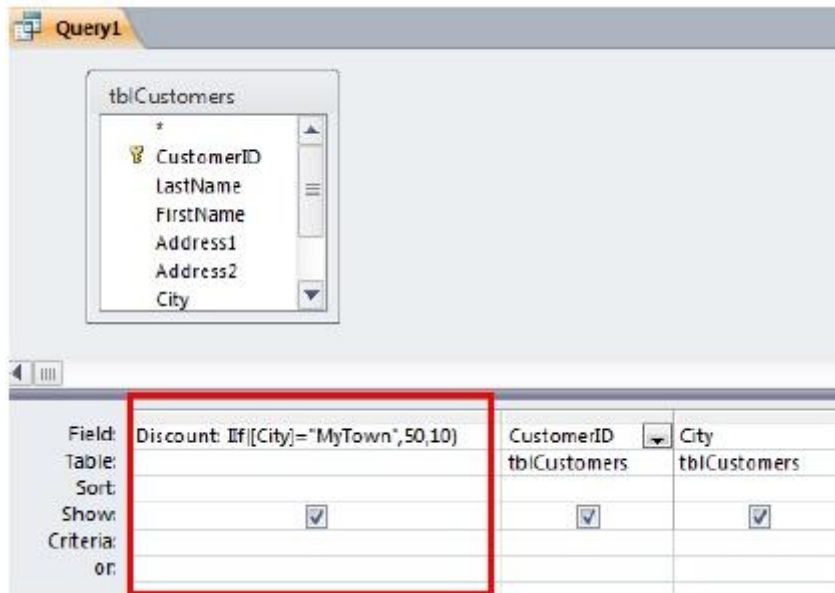
คำตอบที่ได้:

RentalID	Price	Description	25Off	NewPrice
1	\$4.50	1-Day Rental	1.125	฿3.38
2	\$3.50	3-Day Rental	.875	฿2.63
3	\$2.50	5-Day Rental	.625	฿1.88

ภาพที่ 4.16 ขั้นตอนและผลลัพธ์การสร้างฟิลด์คำนวณ

การใช้ IIF

รูปแบบ IIF(เงื่อนไข, ค่าที่ได้เมื่อเงื่อนไขเป็นจริง, ค่าที่ได้เมื่อเงื่อนไขเป็นเท็จ) เช่น



คำตอบที่ได้:

Discount	CustomerID	City
10	1	Commerce Township
10	2	Adamsville
50	3	MyTown
50	4	MyTown
50	5	MyTown
50	6	MyTown
50	7	MyTown
50	8	MyTown
10	9	Commerce Township
10	10	Adamsville
10	11	Adamsville
50	12	MyTown
50	13	MyTown
50	14	MyTown
50	15	MyTown
10	16	Adamsville

ภาพที่ 4.17 ขั้นตอนและผลลัพธ์การใช้อีไอเอฟ

การสร้างแบบสอบถามในมุมมอง SQL

ในการสร้าง Query ด้วย Wizard และสร้างด้วยตนเองในมุมมอง Design จะเหมือนกัน ตรงที่ไม่จำเป็นต้องทราบโครงสร้างภาษา SQL ที่ซ่อนอยู่เบื้องหลังการทำงานของ Query นั้น แม้จะทำงานร่วมกับ Query ได้โดยไม่จำเป็นต้องรู้ภาษา SQL เลยก็ตาม แต่ถ้าได้ทราบการทำงานของภาษา SQL บ้าง ก็จะช่วยให้งานร่วมกับ Query บางประเภทที่ไม่สามารถสร้างด้วย Wizard หรือมุมมอง Design ได้ เช่น Union Query, Pass-Through Query และ Data Definition Query

SQL ย่อมาจาก Structured Query Language เป็นภาษามาตรฐานที่ใช้ทำงานร่วมกับระบบฐานข้อมูล ชุดคำสั่งพื้นฐานที่สำคัญในภาษา SQL ที่ใช้จัดการข้อมูลมี 4 คำสั่ง คือ

1. SELECT ค้นหาข้อมูล

คำสั่งที่ใช้ร่วมกับคำสั่ง SELECT คือ

- DISTINCT แสดงกลุ่ม record ที่ค่าใน field นั้นไม่ซ้ำกัน
- DISTINCTROW แสดงกลุ่ม record ที่ข้อมูลทุก field ไม่ซ้ำกัน
- JOIN กำหนดลักษณะการรวมกันของข้อมูลจากต่าง Table แบ่งออกเป็น 3 ลักษณะ คือ INNER JOIN, LEFT-OUTER JOIN และ RIGHT-OUTER JOIN
- FROM ระบุชื่อ Table / Query ที่จะนำมาค้นหาข้อมูล
- WHERE ระบุเงื่อนไขที่จะใช้ค้นหาข้อมูล
- GROUP BY กำหนด field ที่จะใช้จัดกลุ่มข้อมูล
- ORDER BY กำหนด field ที่จะใช้จัดเรียงข้อมูล
- AS ตั้งชื่อ field ใหม่ในตารางผลลัพธ์

2. UPDATE ปรับปรุงข้อมูล

3. INSERT เพิ่ม record ใหม่ในฐานข้อมูล

4. DELETE ลบ record ออกจากฐานข้อมูล

บทที่ 5

การสร้างฟอร์ม (Form)

นอกจากใช้มุมมองของแผ่นข้อมูล (datasheet) ในการป้อนข้อมูล แก้ไขและแสดงผลข้อมูลแล้ว ยังมีเครื่องมือ อีกชนิดหนึ่ง คือ ฟอร์ม (Form) ที่ช่วยให้สามารถจัดระบบข้อมูลบนจอภาพได้อย่างมีประสิทธิภาพ เนื่องจากรูปแบบการใช้งานจะไม่ถูกกำหนดให้อยู่ในรูปของตารางเท่านั้น แต่สามารถจัดรูปแบบได้อย่างอิสระ เหมาะสมและสะดวกต่อการใช้งานมากขึ้น

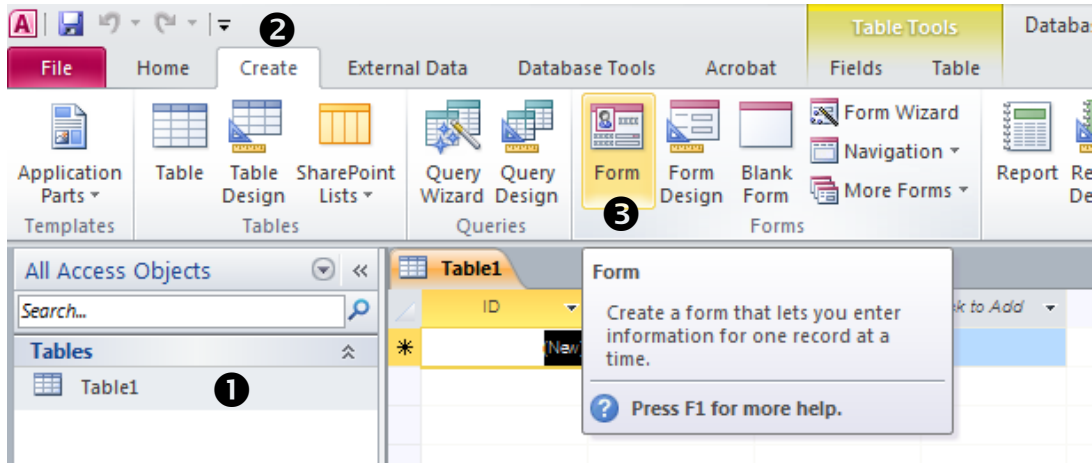
มุมมองของฟอร์ม

ในการทำงานร่วมกับฟอร์มจะมีมุมมองในเลือก 5 แบบ คือ

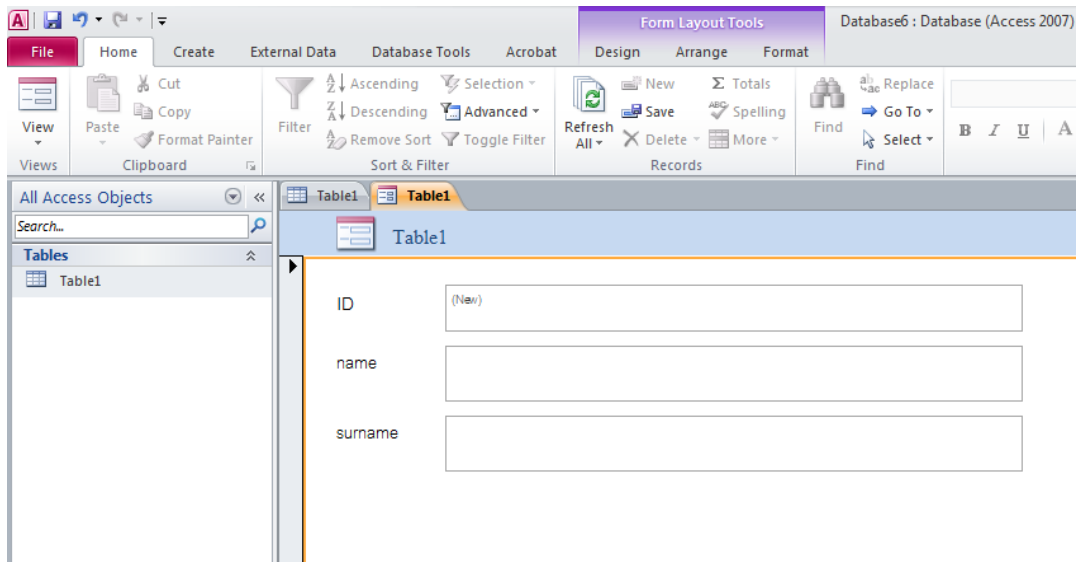
1. **มุมมองการออกแบบ (form design)** ใช้ในการสร้าง ออกแบบ และแก้ไขฟอร์ม
2. **มุมมองฟอร์ม (form view)** ใช้แสดงผลฟอร์มเพื่อติดต่อกับผู้ใช้ตามรูปแบบที่สร้างในมุมมองออกแบบ หรือสร้างด้วยตัวช่วยสร้าง
3. **มุมมองแผ่นข้อมูล (datasheet)** ใช้แสดงผลฟอร์มในรูปของแผ่นข้อมูล หรือตารางข้อมูล
4. **มุมมอง PivotTable** เป็นมุมมองที่แสดงผลฟอร์มในรูปของตารางแจกแจงรายละเอียดและสรุปผลข้อมูล
5. **มุมมอง PivotChart** เป็นมุมมองที่แสดงผลฟอร์มเพื่อการวิเคราะห์และสรุปผลข้อมูลในรูปของแผนภูมิหรือ Chart

การสร้างฟอร์มแบบง่าย

1. เลือก Table/Query ที่จะสร้างฟอร์ม
2. คลิกแท็บ Create
3. คลิก Form



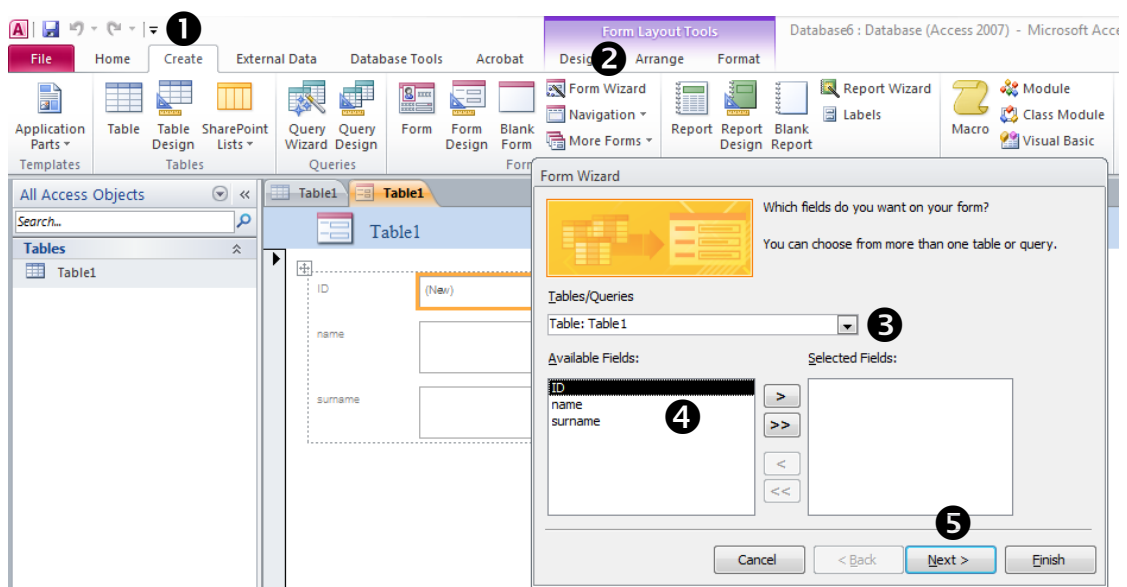
ภาพที่ 5.1 ขั้นตอนการสร้างฟอร์มแบบง่าย



ภาพที่ 5.2 ผลการสร้างฟอร์มแบบง่าย

การสร้างฟอร์มโดยใช้ตัวช่วย (Form Wizard)

1. คลิกแท็บ Create
2. คลิก Form Wizard
3. เลือก Table/Query ที่จะป็นแหล่งข้อมูลของฟอร์ม
4. ดับเบิลคลิกเลือก field ที่จะใช้ หรือคลิก >> เพื่อเลือก field ทั้งหมด
5. คลิก Next
6. เลือกว่าจะสร้างฟอร์มแบบไหน
7. คลิก Next
8. ตั้งชื่อฟอร์ม
9. เลือกว่าจะเปิดฟอร์มในมุมมองใด
10. คลิก Finish



Form Wizard

What layout would you like for your form?

Columnar **6**
 Tabular
 Datasheet
 Justified

7

Form Wizard

What title do you want for your form?

Table2 **8**

That's all the information the wizard needs to create your form.
Do you want to open the form or modify the form's design?

Open the form to view or enter information. **9**
 Modify the form's design.

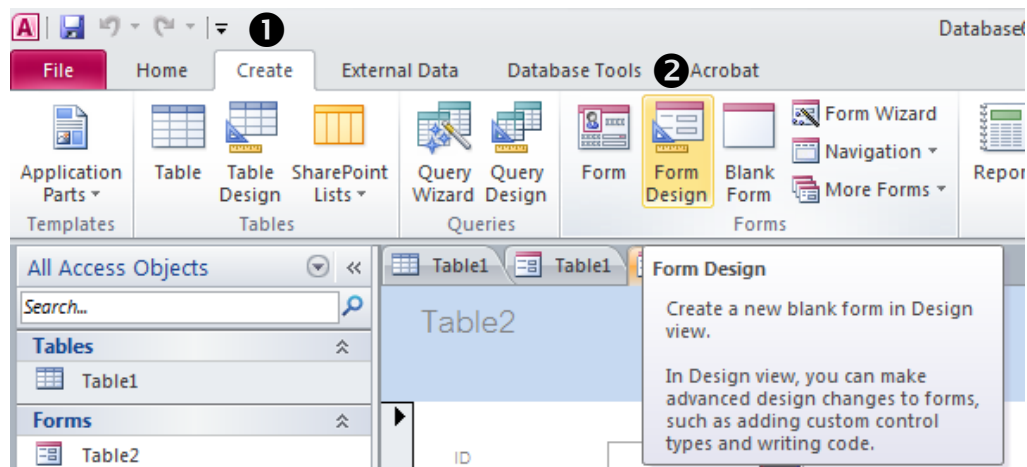
10

ภาพที่ 5.3 ขั้นตอนการสร้างฟอร์มโดยใช้ตัวช่วย (Form Wizard)

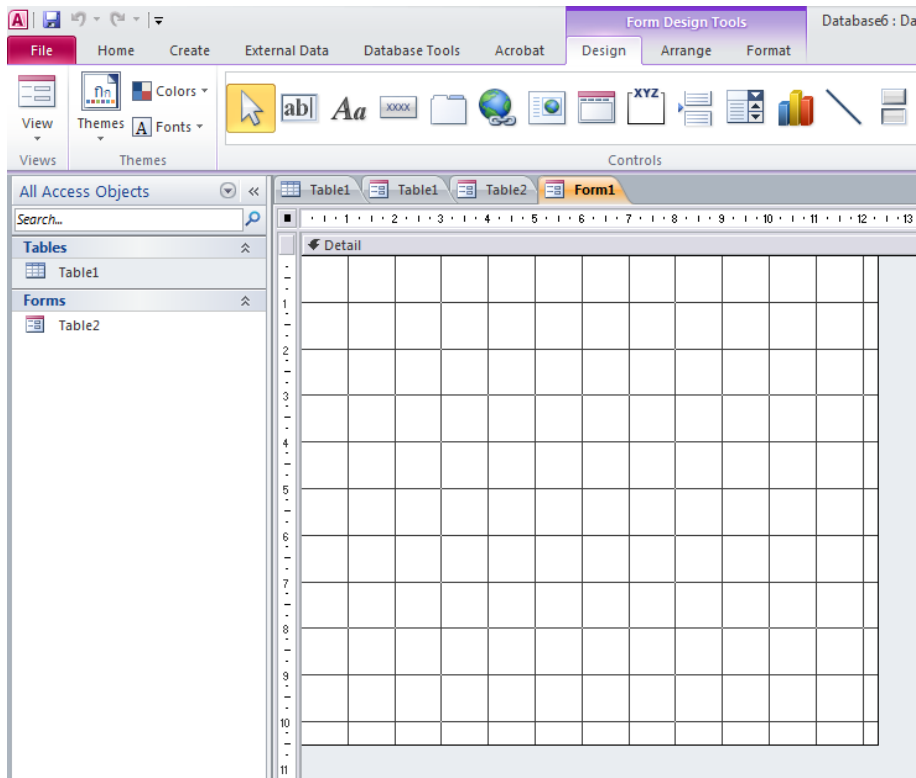
การสร้างฟอร์มด้วยมุมมองการออกแบบ

การสร้างฟอร์มด้วยมุมมองการออกแบบ (Form Design) จะได้ฟอร์มที่ตรงกับวัตถุประสงค์ในการใช้งานมากที่สุด สามารถใช้แก้ไขฟอร์มที่สร้างด้วย Form Wizard ได้ วิธีการสร้างมีดังนี้

1. คลิกแท็บ Create
2. คลิก Form Design

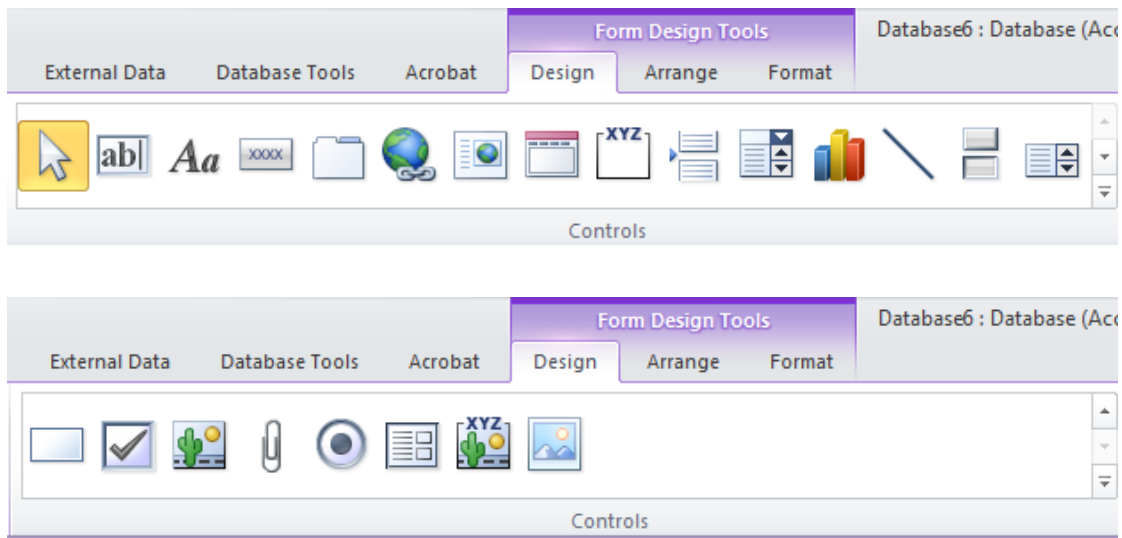


ภาพที่ 5.4 ขั้นตอนการสร้างฟอร์มด้วยมุมมองการออกแบบ (Form Design)






























ภาพที่ 5.5 หน้าจอการสร้างฟอร์มด้วยมุมมองการออกแบบ (Form Design)

ปุ่มสร้างคอนโทรลในแท็บ Design



ภาพที่ 5.6 ปุ่มสร้างคอนโทรลในแท็บ Design

ตารางที่ 5.1 แสดงปุ่มสร้างคอนโทรลในการสร้างฟอร์ม

เครื่องมือ	หน้าที่
	เลือกสิ่งที่ต้องการปรับ เช่น สีเหลี่ยม ตัวหนังสือ ช่องให้กรอก เป็นต้น
	สร้างช่องให้กรอก (ช่อง Text Box)
	สร้างเลเบล (Label)
	สร้างปุ่มกด
	สร้างตัวควบคุมแท็บ เพื่อแบ่งกลุ่มคำสั่งให้เป็นหมวดหมู่
	Hyperlink
	Web browser control
	Navigation control
	สร้างกลุ่มรายการให้เลือก
	ใช้แบ่งหน้าและยกเลิกการแบ่งหน้า
	สร้างกล่องรายการให้เลือกแบบบรรทัดเดียว
	แทรกกราฟลงไปในฟอร์ม
	สร้างเส้นตรง
	สร้างตัวควบคุมข้อมูลว่าจะใช้หรือไม่ใช้
	สร้างกล่องรายการให้เลือกแบบหลายบรรทัด
	สร้างสี่เหลี่ยม
	สร้างตัวเลือกซึ่งจะเลือกหรือไม่ก็ได้
	สร้างกรอบเก็บรูปภาพซึ่งจะไม่เก็บลงตาราง
	เพิ่มไฟล์แนบเข้ามาในฟอร์ม
	สร้างตัวเลือกซึ่งเลือกได้ 1 เดียวเท่านั้น
	สร้างฟอร์มย่อยหรือรายงานย่อย
	สร้างกรอบเก็บรูปหรือข้อมูลลงในตาราง
	แทรกรูปภาพ
	ใส่โลโกตรงส่วนหัว (Header) ของฟอร์ม
	ใส่หัวเรื่องตรงส่วนหัวของฟอร์ม
	ใส่วันและเวลาในส่วนหัวของฟอร์ม
	ใส่รูปในฟอร์ม

ส่วนประกอบในมุมมองออกแบบของฟอร์ม

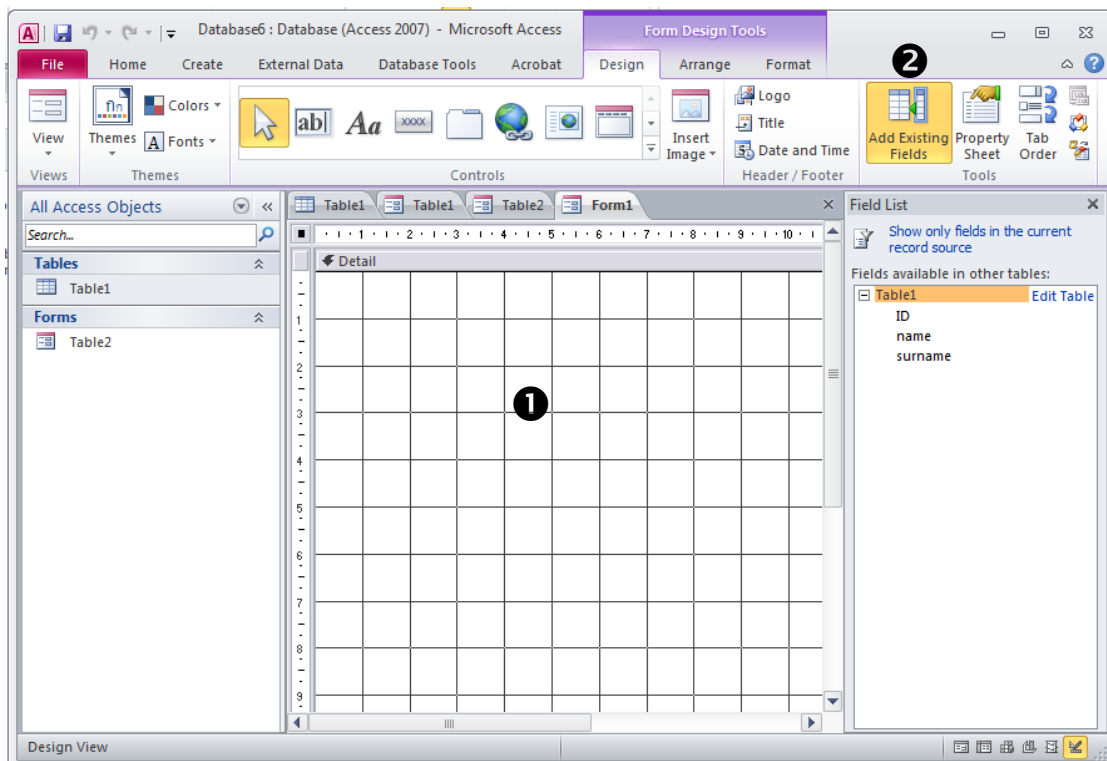
1. Form Design Area

พื้นที่ที่ใช้สร้างและออกแบบฟอร์ม

2. Field List

ส่วนที่แสดงชื่อ Table/Query ที่เป็นแหล่งข้อมูลของฟอร์ม จากการคลิกที่ปุ่ม Add

Existing Fields ในแท็บ Design



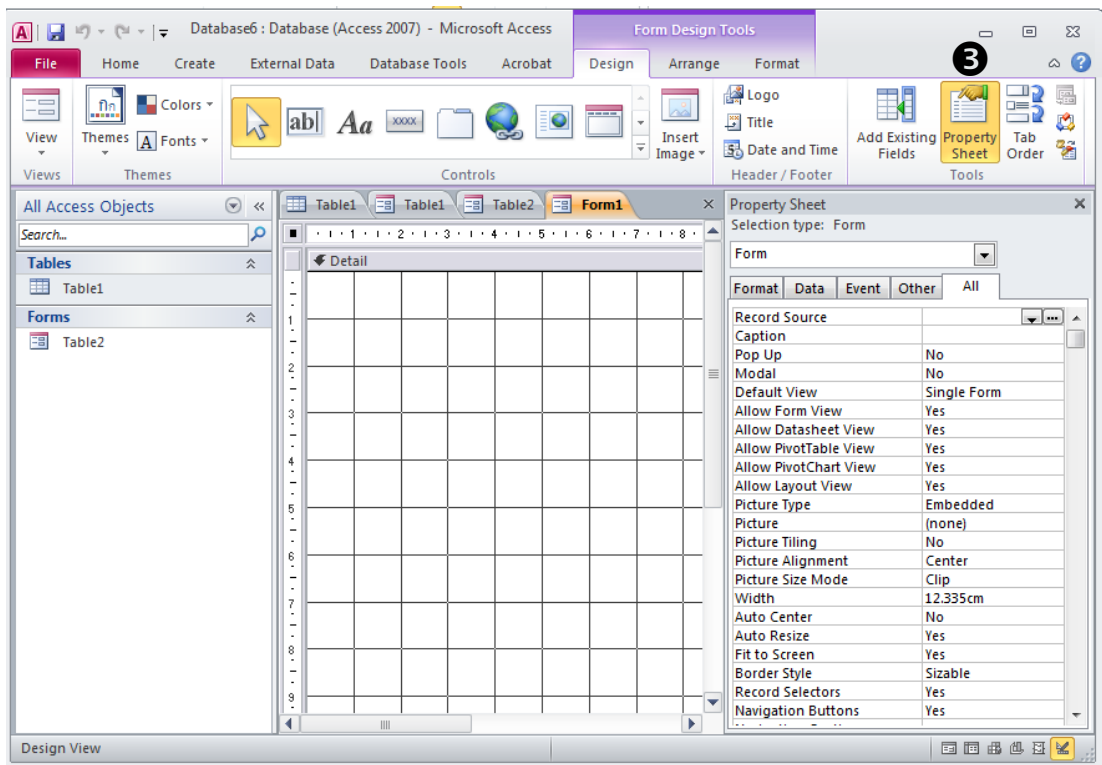
ภาพที่ 5.7 ส่วนประกอบในมุมมองออกแบบ

3. Property Sheet

Property Sheet เป็นส่วนที่ใช้กำหนดคุณสมบัติต่างๆ ให้กับฟอร์มหรือคอนโทรลต่างๆ ในฟอร์ม

3.1 คุณสมบัติของฟอร์ม

คุณสมบัติที่ใช้กำหนดรูปแบบและการทำงานของฟอร์มจะแบ่งออกเป็น 4 กลุ่ม คือ Format, Data, Event และ Other



ภาพที่ 5.8 คุณสมบัติที่ใช้กำหนดรูปแบบและการทำงานของฟอร์ม (Property Sheet)

ตารางที่ 5.2 แสดงคุณสมบัติของฟอร์ม

คุณสมบัติ	การทำงาน
<p>แท็บ Format</p> <p>ใช้กำหนดรูปแบบการแสดงผลฟอร์ม เช่น กำหนดมุมมอง จัดตำแหน่งฟอร์มบนจอ และกำหนดเส้นขอบ เป็นต้น</p>	
<p>Default View</p>	<p>กำหนดรูปแบบการแสดงผลในมุมมองฟอร์ม มีให้เลือก 6 แบบ คือ</p> <ol style="list-style-type: none"> 1. Single Form แสดง 1 record ต่อ 1 หน้า 2. Continuous Forms แสดงหลายๆ record ต่อเนื่องกันไปใน 1 หน้า 3. Datasheet แสดงข้อมูลในรูปแบบตารางเช่นเดียวกับมุมมอง Datasheet ของ Table/Query 4. PivotTable แสดงข้อมูลในรูปแบบของตารางสรุปผลข้อมูล 5. PivotChart แสดงข้อมูลในรูปแบบของกราฟ 6. Split Form เป็นฟอร์มแบบผสมที่มี 2 มุมมองในฟอร์มเดียวกัน คือ มุมมอง Datasheet และมุมมอง Form
<p>Border Style</p>	<p>กำหนดลักษณะเส้นขอบของฟอร์ม มี 4 แบบ คือ</p> <ol style="list-style-type: none"> 1. None ไม่แสดงเส้นขอบ แถบชื่อฟอร์ม เมนูควบคุม ปุ่มปรับขนาดและปุ่มปิดฟอร์ม และผู้ใช้ไม่สามารถปรับขนาดฟอร์มได้ 2. Thin แสดงเส้นขอบแบบบางและสวดส่วนประกอบอื่นๆ ข้างต้นที่ None ไม่แสดง ถ้าจะปรับขนาดของฟอร์มต้องใช้ปุ่มปรับขนาดเท่านั้น ไม่สามารถปรับเองแบบอิสระได้ 3. Sizable แสดงเส้นขอบแบบปกปิดและส่วนประกอบอื่นๆ ข้างต้นสามารถปรับขนาดได้อย่างอิสระโดยคลิกค้างไว้ที่เส้นขอบ ลากไปมาจนได้ขนาดที่ต้องการ 4. Dialog แสดงเส้นขอบแบบบางและส่วนประกอบอื่นๆ (ยกเว้นปุ่มปรับขนาด) ถ้ากำหนดคุณสมบัตินี้ จะปรับขนาดฟอร์มไม่ได้ <p>ให้กำหนดคุณสมบัติ Pop Up เป็น Yes ก่อนใช้คุณสมบัติ Border Style</p>
<p>Record Selectors</p>	<p>ถ้าตั้งเป็น Yes จะแสดงแถบเลือก record ในแนวตั้งท้ายสุดของฟอร์ม</p>

คุณสมบัติ	การทำงาน
Navigation Buttons	ถ้าตั้งเป็น Yes จะแสดงแถบเลือก record ที่ส่วนล่างสุดของฟอร์ม
Dividing Lines	ถ้าตั้งเป็น Yes จะแสดงเส้นแบ่ง record หรือเส้นแบ่ง section ในฟอร์มออก จากกัน จะเห็นได้ชัดเจนถ้าเลือกแสดงฟอร์มแบบ Continuous ในคุณสมบัติ Default View
Auto Resize	ถ้าตั้งเป็น Yes จะปรับขนาดของฟอร์มให้พอดีกับพื้นที่ใช้งานจริงใน section ต่างๆ โดยอัตโนมัติเมื่ออยู่ในมุมมองฟอร์ม
Auto Center	ถ้าตั้งเป็น Yes เมื่อเปิดใช้ฟอร์ม ตำแหน่งของฟอร์มจะอยู่กึ่งกลางจอโดยอัตโนมัติ
แท็บ Data	
กำหนดคุณสมบัติการทำงานร่วมกับข้อมูล เช่น กำหนดแหล่งข้อมูล และผู้ใช้สามารถแก้ไขข้อมูล ได้หรือไม่ เป็นต้น	
Record Source	ระบุแหล่งข้อมูลของฟอร์มนั้น ซึ่งอาจเป็น Table/Query หรือประโยคคำสั่ง SELECT ในภาษา SQL
Allow Additions	ถ้าตั้งเป็น Yes หมายถึง ให้เพิ่ม record ใหม่ผ่านทางฟอร์มนี้ได้ โดยแสดงทั้ง record เดิมและ record ใหม่ว่างๆ สำหรับป้อนข้อมูล
Allow Deletions	ถ้าตั้งเป็น Yes หมายถึง ให้ลบ record ใดๆ ในแหล่งข้อมูลนั้นได้
Allow Edits	ถ้าตั้งเป็น Yes หมายถึง ให้แก้ไข record ในแหล่งข้อมูลนั้นผ่านทางฟอร์มได้
Data Entry	ถ้าตั้งเป็น Yes หมายถึง ให้เพิ่ม record ใหม่ได้ โดยแสดงเฉพาะ record ใหม่ว่างๆ สำหรับป้อนข้อมูลเท่านั้น จะไม่แสดง record เดิมที่มีอยู่แล้ว
แท็บ Event	
ใช้กำหนดเหตุการณ์ที่จะให้เกิดการกระทำ (Action) บนฟอร์มหรือคอนโทรลต่างๆ บนฟอร์มนั้น ใช้ร่วมกับ Macro และ Module ได้แก่ คุณสมบัติที่ขึ้นต้นด้วย On ทั้งหมด เช่น On Open, On Load, On Close และ On Click เป็นต้น และคุณสมบัติอื่นๆ เช่น Before Insert, Before Update, After Insert และ After Update เป็นต้น	
แท็บ Other	
Modal	ถ้าตั้งเป็น Yes หมายถึง ถ้าเปิดฟอร์มนี้หลังจากเปิดฟอร์มอื่นไว้ก่อนแล้ว ฟอร์มเหล่านั้นจะทำงานต่อไม่ได้ จนกว่าจะปิดฟอร์มแบบ Modal นี้ก่อน

คุณสมบัติ	การทำงาน
Pop-up	ถ้าตั้งเป็น Yes หมายถึง เส้นขอบของฟอร์มจะถูกเปลี่ยนให้เป็นแบบบางและแสดงลอยตัวเหนือฟอร์มอื่นๆ ทั้งหมดที่เปิดใช้งานอยู่ก่อนแล้ว

3.2 คุณสมบัติของคอนโทรล

คุณสมบัติ	การทำงาน
คุณสมบัติทั่วไปของคอนโทรล	
คอนโทรลทุกประเภทจะมีคุณสมบัติพื้นฐานบางตัวที่เหมือนกันที่มักจะถูกใช้งานบ่อย เช่น คุณสมบัติในการซ่อน/แสดงคอนโทรล และคุณสมบัติในการตอบสนองการทำงานของผู้ใช้ เป็นต้น	
Visible	คุณสมบัติในแท็บ Format ของคอนโทรลนั้น ถ้าตั้งค่าเป็น No หมายถึง ให้ซ่อนคอนโทรลนั้นไว้เพื่อไม่ให้ผู้ใช้มองเห็นขณะทำงานในมุมมองฟอร์ม แต่ถ้าอยู่ในมุมมอง Design จะมองเห็นและแก้ไขได้
Enabled	กำหนดว่าจะให้คอนโทรลนั้นตอบสนองการทำงานจากผู้ใช้หรือไม่ - Yes ให้คอนโทรลนั้นทำงานตามคำสั่ง เมื่อผู้ใช้คลิกหรือกำหนดเหตุการณ์ใดๆ ให้กับคอนโทรล - No ผู้ใช้ไม่สามารถคลิกหรือทำงานใดๆ ร่วมกับตัวคอนโทรลนั้นได้
Locked	ถ้าตั้งเป็น Yes ผู้ใช้จะไม่สามารถแก้ไขข้อมูลที่อยู่กับคอนโทรลนั้นได้
Tab Order	ใช้กำหนดลำดับแท็บของคอนโทรลให้สอดคล้องกับลำดับการจัดวางคอนโทรลในฟอร์มหรือสอดคล้องกับลำดับการป้อนข้อมูล (เป็นตัวกำหนดลำดับการเลื่อนเคอร์เซอร์ไปยังคอนโทรลต่างๆ ในมุมมองฟอร์ม เมื่อผู้ใช้กดคีย์บอร์ด Tab หรือ Enter)
คุณสมบัติของ Text Box	
Name	ตั้งชื่อให้คอนโทรล เพื่อนำไปใช้อ้างอิงในออบเจ็ค Macro, Module หรือออบเจ็คอื่นๆ ได้
Control Source	ระบุชื่อ field ที่นำมาผูกหรือเป็นแหล่งข้อมูลของคอนโทรลนั้น
Format	กำหนดรูปแบบการแสดงผลข้อมูลในคอนโทรลนั้น
Input Mask	กำหนดรูปแบบการป้อนข้อมูล
Enter Key Behavior	กำหนดว่าจะให้ทำอะไรต่อไปหลังกดคีย์บอร์ด Enter ใน Text Box นั้น ถ้ากำหนดเป็น Default หมายถึง ให้ไปทำงานใน Text Box ที่อยู่ลำดับถัดไป ถ้ากำหนดเป็น New Line in Field หมายถึง ให้ขึ้นบรรทัดใหม่ใน Text Box นั้น

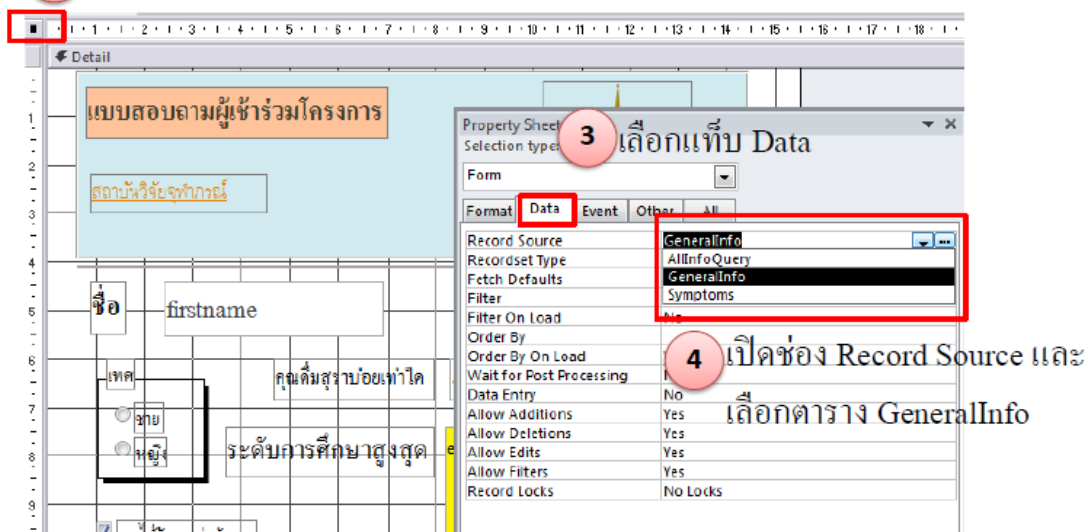
คุณสมบัติ	การทำงาน
คุณสมบัติของ Option Group	
คุณสมบัติของคอนโทรลชนิดนี้จะแยกออกเป็น 2 ส่วน คือ ส่วนของ Option Group และ Option Button	
- Option Group Properties	
Name	ตั้งชื่อคอนโทรล Option Group เพื่อนำไปใช้อ้างอิงในออบเจ็ค Macro, Module หรือออบเจ็คอื่นๆ ได้
Control Source	ระบุชื่อ field ที่นำมาผูกกับคอนโทรล
Default Value	กำหนดค่าเริ่มต้นให้คอนโทรล
- Option Button Properties	
Name	ตั้งชื่อคอนโทรล Option Group เพื่อนำไปใช้อ้างอิงในออบเจ็ค Macro, Module หรือออบเจ็คอื่นๆ ได้
Option Value	กำหนดค่าให้กับปุ่ม Option แต่ละปุ่ม เช่น ปุ่มเลือกเพศชายมีค่าเป็น 1 และ ปุ่มเลือกเพศหญิงมีค่าเป็น 2
- Combo Box Properties	
Row Source Type	ระบุแหล่งข้อมูลของลิสต์ ค่าเริ่มต้นที่ Access กำหนดไว้ คือ Table/Query
Row Source	ระบุชื่อ Table/Query ที่เป็นแหล่งข้อมูล หรือระบุประโยคคำสั่ง SELECT ใน ภาษา SQL
Column Count	กำหนดจำนวนคอลัมน์ที่จะแสดงในลิสต์ เช่น 1 หมายถึง แสดง 1 คอลัมน์
Column Heads	กำหนดว่าจะแสดงชื่อ field ที่ส่วนหัวของคอลัมน์ในลิสต์หรือไม่ ถ้ากำหนดเป็น No หมายถึง ไม่ต้องแสดง
Column Widths	กำหนดความกว้างของแต่ละคอลัมน์ในลิสต์
Bound Column	ระบุหมายเลขคอลัมน์ที่จะนำข้อมูลไปเก็บใน Table (เรียงลำดับหมายเลขจกซ้ายไปขวา คือ 1, 2, 3,...)
Limit To List	ถ้าเป็น Yes หมายถึง จะคีย์ข้อมูลที่ไม่มีในลิสต์ลงช่องว่างของ Combo Box ไม่ได้ ถ้าเป็น No หมายถึง คีย์ข้อมูลที่ไม่มีในลิสต์ลงช่องว่างได้ (ก่อนกำหนดเป็น No ต้องยกเลิกการซ่อนคอลัมน์ในลิสต์ก่อน)

คุณสมบัติ	การทำงาน
- Subform Properties	
Name	ตั้งชื่อให้คอนโทรล Subform เพื่อนำไปใช้อ้างอิงในออบเจ็ค Macro, Module หรืออื่น ๆ
Source Object	ระบุชื่อฟอร์มย่อยที่นำมาผูกกับคอนโทรล Subform
Link Child Fields	ระบุชื่อ field ในฟอร์มย่อยที่จะนำมาเชื่อมโยงกับ field ในฟอร์มหลัก
Link Master Fields	ระบุชื่อ field ในฟอร์มหลักที่จะนำมาเชื่อมโยงกับ field ในฟอร์มย่อย

การเชื่อมฟอร์มเข้ากับตารางฐานข้อมูล

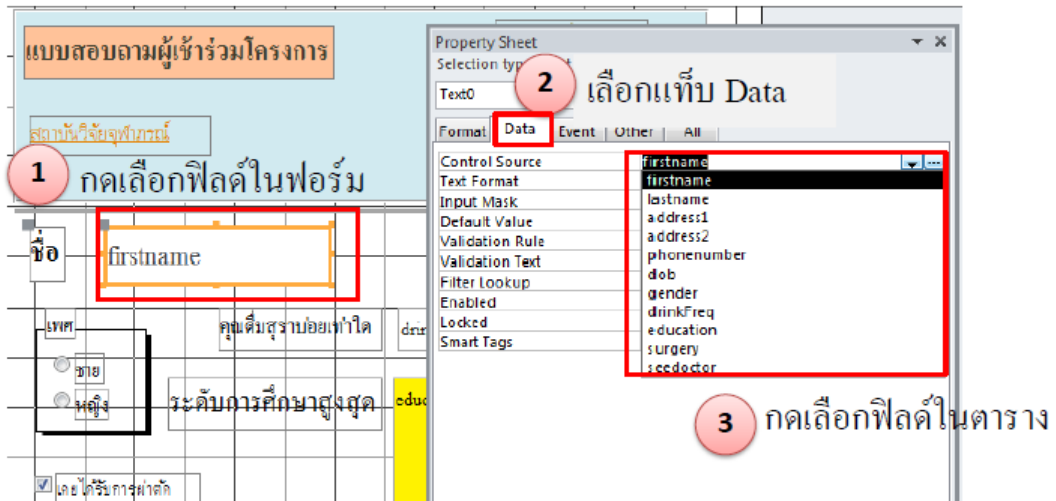
1. เชื่อมทั้งฟอร์มเข้ากับทั้งตาราง

- 1 เปิด TestForm ใน Design View
- 2 กดสี่เหลี่ยมตรงมุมบนซ้าย



ภาพที่ 5.9 ขั้นตอนการเชื่อมฟอร์มทั้งฟอร์มเข้ากับตารางฐานข้อมูล

2. เชื่อมแต่ละ field ของฟอร์มเข้ากับแต่ละ field ของตาราง
- การเชื่อม 1 ฟอร์มกับหลายๆ ตารางเข้าด้วยกันต้องสร้าง query เพื่อเชื่อมตารางก่อน จึงจะเชื่อมฟอร์มเข้ากับตารางหลายๆตารางได้

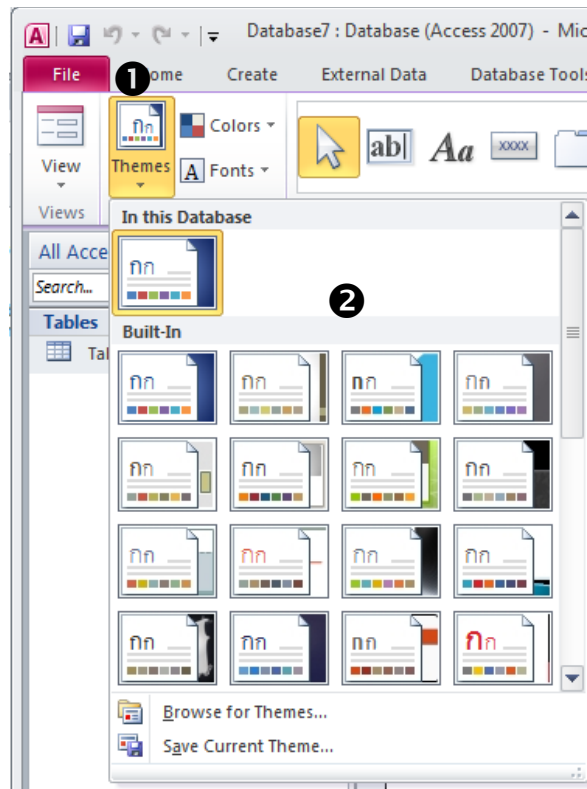


ภาพที่ 5.9 ขั้นตอนการเชื่อมฟอร์มเข้ากับตารางฐานข้อมูลโดยเชื่อมแต่ละ field

การปรับแต่งฟอร์ม

คำสั่ง Themes จะปรับรูปแบบตัวอักษรและสีที่ใช้ในอบเจ็ค Table, Form และ Report โดยมีขั้นตอนดังนี้

1. คลิก Themes ในแท็บ Design
2. คลิกเลือกรูปแบบที่จะใช้

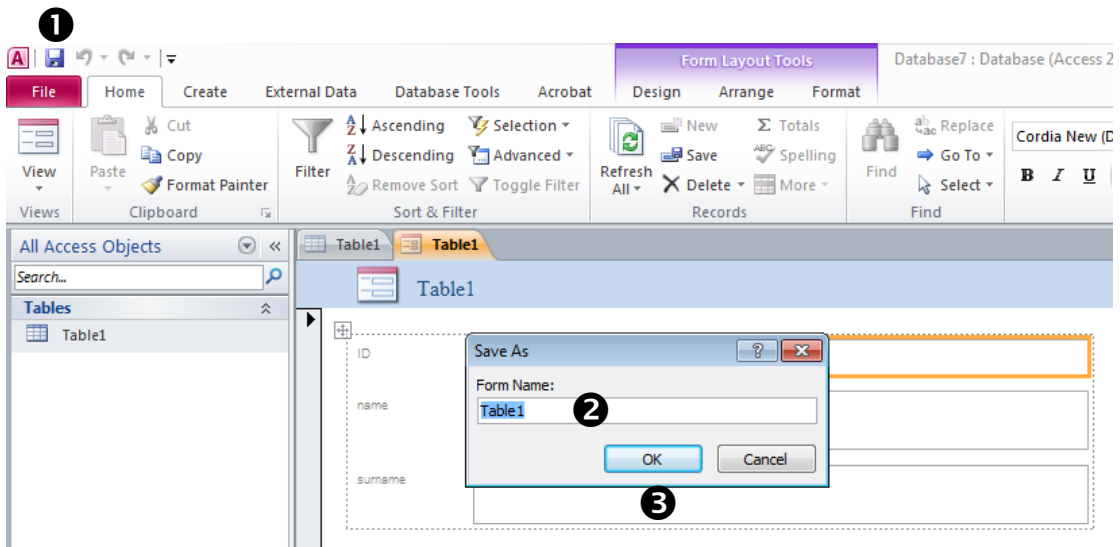


ภาพที่ 5.10 ขั้นตอนการปรับแต่งฟอร์มด้วยคำสั่ง Themes

การจัดเก็บฟอร์มลงฐานข้อมูล

การจัดเก็บหรือบันทึกฟอร์มลงฐานข้อมูลให้ทำดังนี้

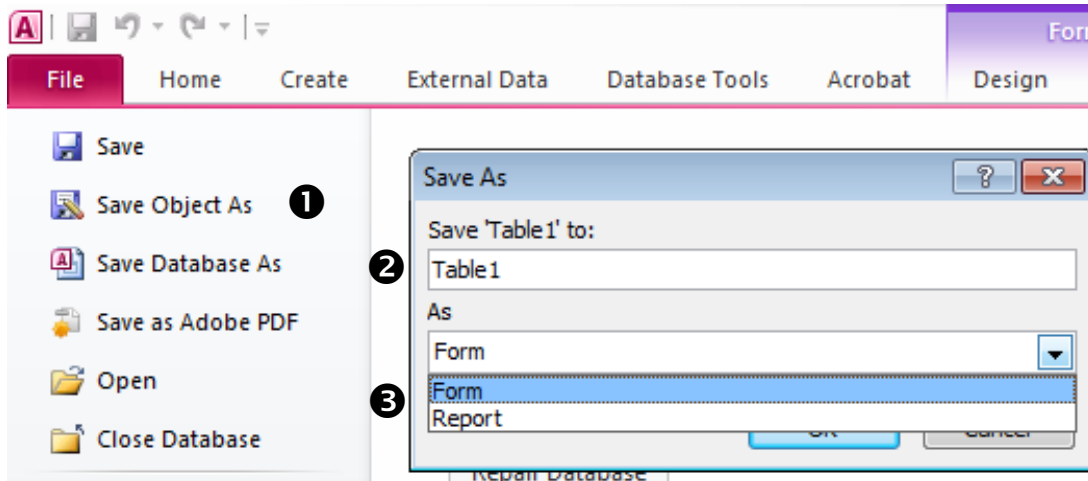
1. คลิก Save บนทูลบาร์ Quick Access
2. ตั้งชื่อฟอร์มในช่อง Form Name
3. คลิก OK



ภาพที่ 5.11 ขั้นตอนการจัดเก็บฟอร์มลงฐานข้อมูล

ถ้าจัดเก็บฟอร์มให้เป็นออบเจ็คประเภทอื่น เช่น Report ทำได้ดังนี้

1. คลิกแท็บ File แล้วคลิกคำสั่ง Save Object as
2. ตั้งชื่อออบเจ็คที่จะจัดเก็บ
3. ถ้าจะจัดเก็บเป็นออบเจ็ค Report ให้คลิกเลือก Report ในช่อง As แล้วคลิกปุ่ม OK



ภาพที่ 5.12 ขั้นตอนการจัดเก็บฟอร์มให้เป็นออบเจ็คประเภทอื่น

บทที่ 6

การสร้างรายงาน (Report)

การสร้างรายงาน (report) เป็นการนำข้อมูลจากตาราง หรือแบบสอบถาม ผลลัพธ์ที่ได้จากการค้นหาข้อมูล หรือผลลัพธ์ที่ได้จากการคำนวณ เช่น ผลสรุปยอดรวม ค่าเฉลี่ย ค่าทางสถิติ และค่าที่ได้จากการวิเคราะห์ด้วยฟังก์ชันทางการเงิน เป็นต้น รวมทั้งการนำเสนอข้อมูลในรูปแบบแผนภูมิหรือ chart ซึ่งสามารถแสดงข้อมูลได้ทั้งบนจอภาพและพิมพ์รายงานออกมาทางเครื่องพิมพ์

มุมมองของรายงาน

มุมมองของรายงาน มี 3 แบบ คือ

1. **มุมมองออกแบบ (design)** เป็นมุมมองที่ใช้ในการสร้างและแก้ไขรายงาน จะมีลักษณะคล้ายมุมมอง design ของฟอร์ม
2. **มุมมองตัวอย่างก่อนพิมพ์ (print preview)** เป็นมุมมองที่ใช้แสดงตัวอย่างรายงานที่สร้างไว้ในมุมมอง design เพื่อดูผลก่อนสั่งพิมพ์ออกทางเครื่องพิมพ์ ถ้าผลที่ได้ยังไม่เป็นที่พอใจ หรือมีข้อผิดพลาดจะได้แก้ไขก่อนพิมพ์จริง ซึ่งจะช่วยประหยัดกระดาษและเวลาได้มาก
3. **มุมมองแสดงตัวอย่างเค้าโครง (layout previews)** เป็นมุมมองที่ใช้แสดงตัวอย่างโครงสร้างรายงานในลักษณะเดียวกับมุมมอง print preview ต่างกันตรงจะแสดงข้อมูลตัวอย่างเพียงคร่าวๆ เท่านั้น จะไม่แสดงข้อมูลจริงทั้งหมดในรายงานนั้นเพื่อให้ผู้ใช้งานเห็นตัวอย่างรายงานอย่างรวดเร็วก่อนพิมพ์จริง

วิธีการสร้างรายงาน

การสร้างรายงาน ทำได้ 3 วิธี คือ

1. **สร้างรายงานง่ายๆ จาก 1 Table / Query อย่างรวดเร็ว** เป็นวิธีที่เร็วและง่ายที่สุด เพียงคลิกเลือก Table / Query ที่จะให้เป็นแหล่งข้อมูลของรายงานนั้น แล้วคลิกปุ่มคำสั่ง Report ในแท็บ Create

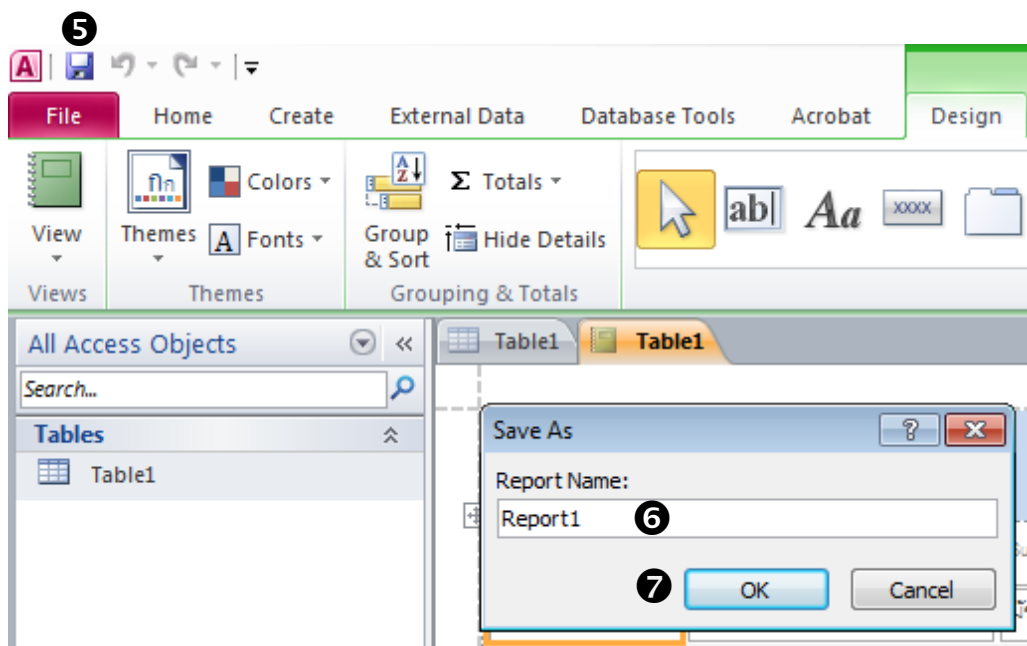
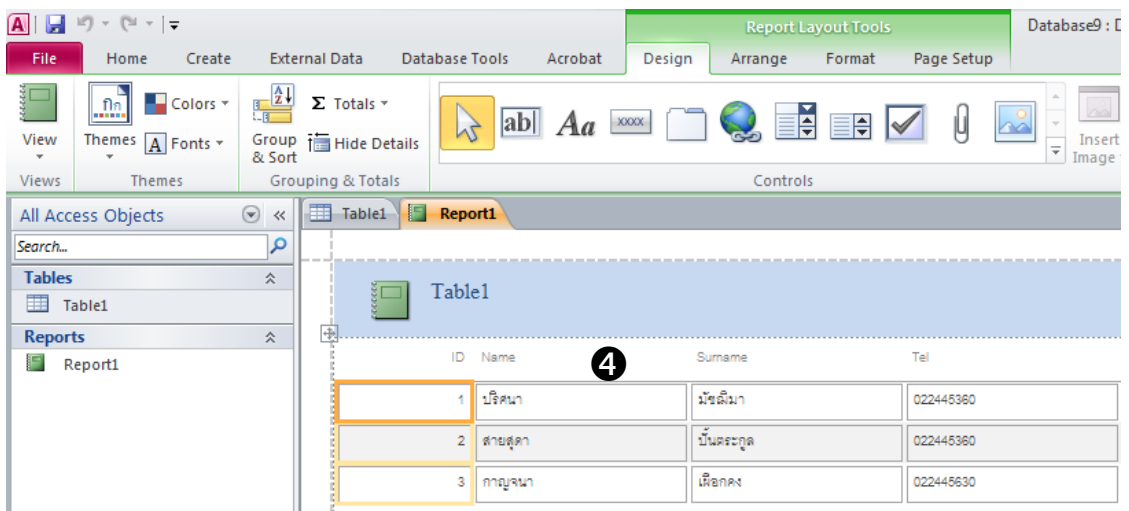
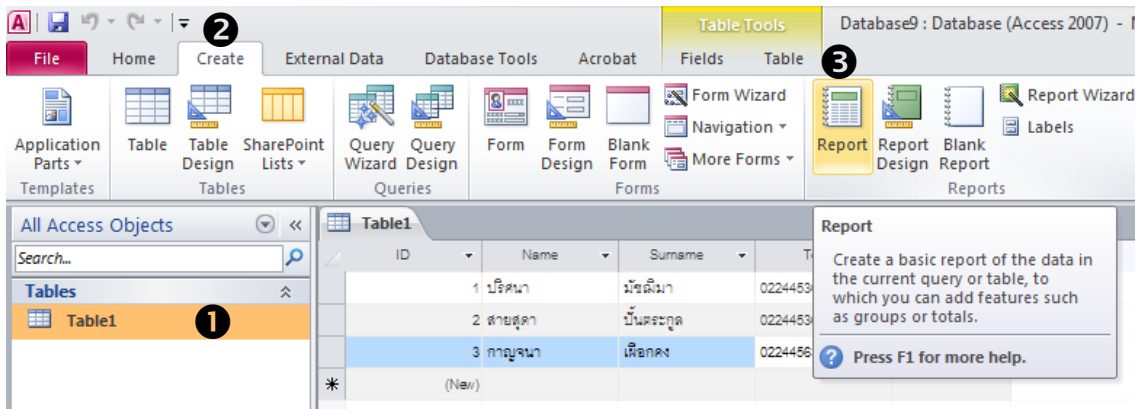
2. **สร้างรายงานโดยใช้ตัวช่วยสร้าง (Report Wizard)** เป็นการสร้างรายงานโดยตอบคำถามหรือเลือกส่วนประกอบต่างๆ ที่ต้องการให้แสดงในรายงาน โดยทำตามลำดับขั้นตอนไปที่ละขั้น ซึ่งสามารถสร้างแผนภูมิหรือ chart ได้ด้วย

3. **สร้างรายงานในมุมมองออกแบบ (Design View)** เป็นการสร้างรายงานด้วยตัวเองทั้งหมด ตั้งแต่การออกแบบ เลือกส่วนประกอบ กำหนดตำแหน่ง เงื่อนไขรายละเอียด และอื่นๆ ที่จำเป็นจนได้รายงานที่ต้องการ

รายละเอียดในการสร้างรายงานจะคล้ายๆ กับการสร้างฟอร์ม ซึ่งถ้าต้องการความสะดวกอาจเลือกสร้างรายงานแบบอัตโนมัติ หรือใช้ตัวช่วยสร้าง แล้วทำการแก้ไขในตอนหลังด้วยตัวเองได้ในมุมมองออกแบบ

การสร้างรายงานแบบง่าย

1. เลือก Table / Query ที่จะใช้สร้างรายงาน
2. คลิกแท็บ Create
3. คลิก Report
4. แสดงรายงานแบบ Tabular
5. คลิก Save
6. ตั้งชื่อรายงาน
7. คลิก OK



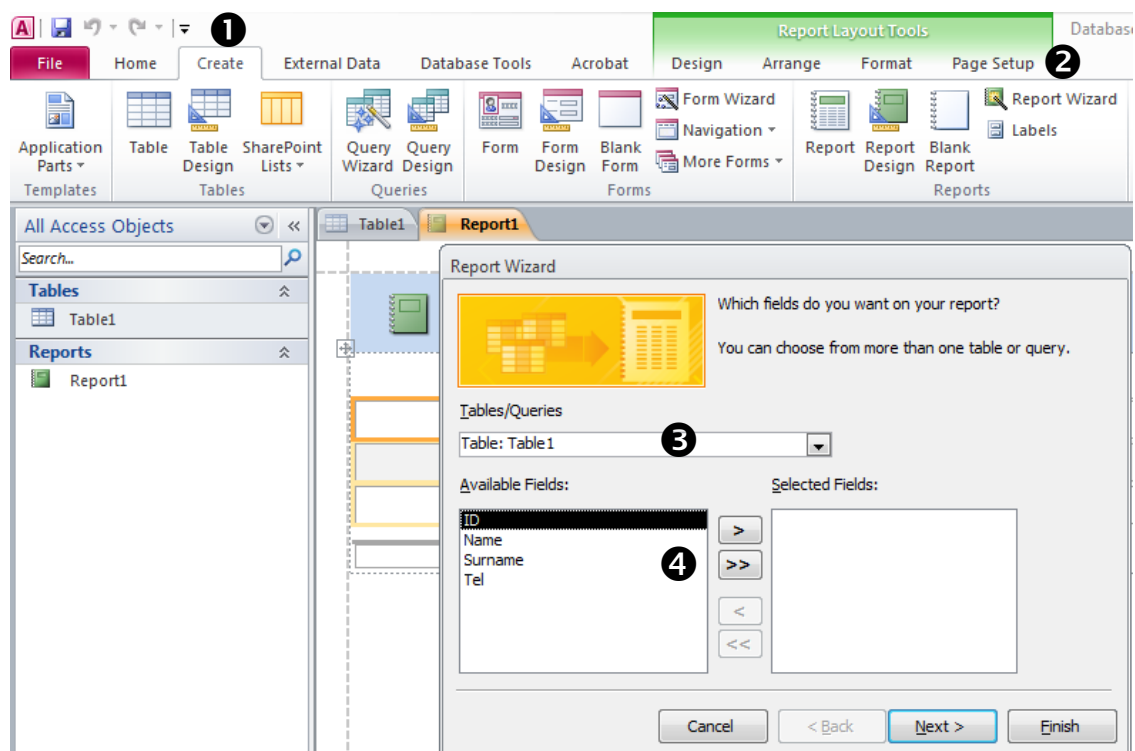
ภาพที่ 6.1 ขั้นตอนการสร้างรายงานแบบง่าย

สร้างรายงานโดยใช้ตัวช่วยสร้าง (Report Wizard)

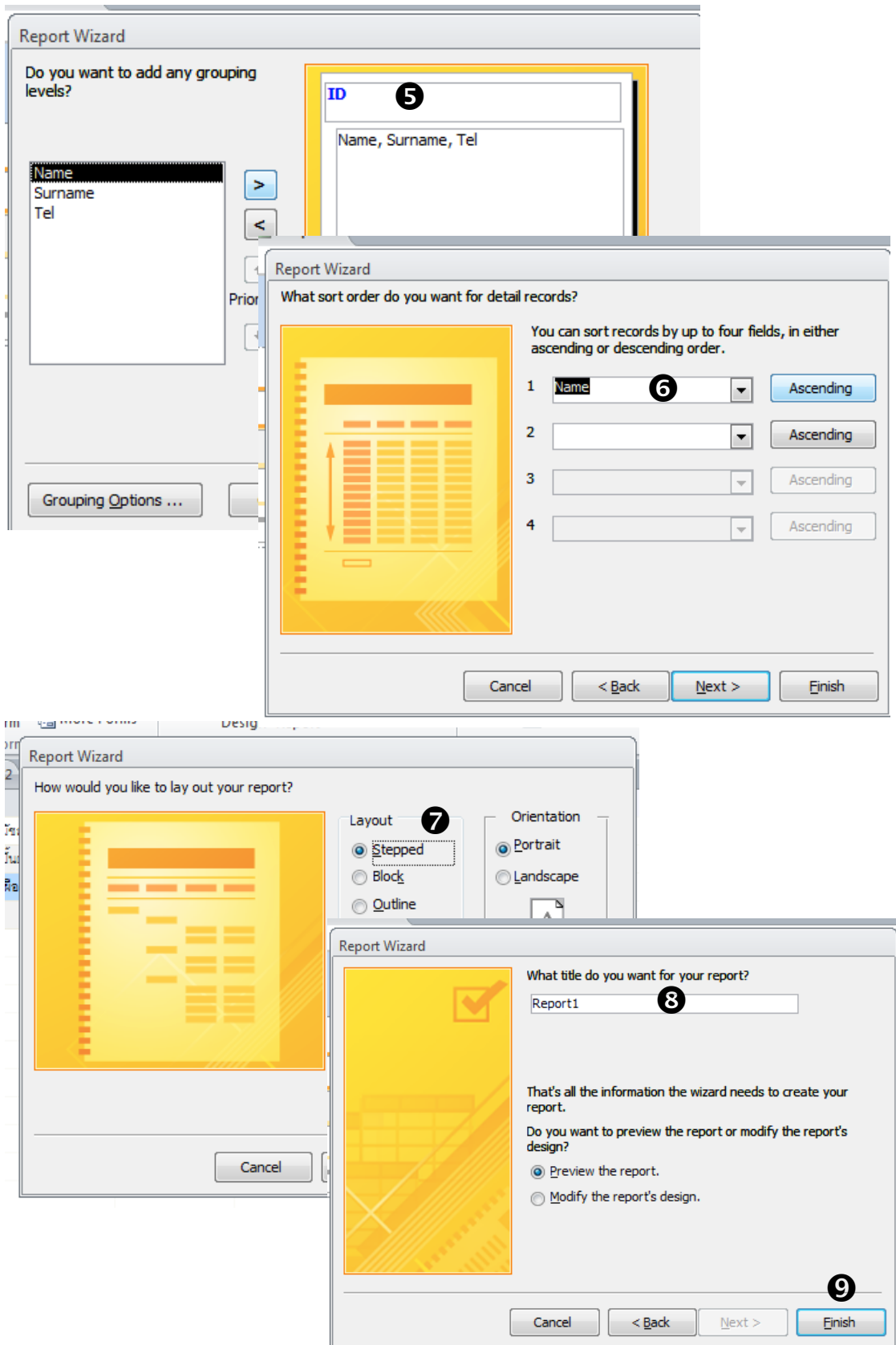
1. คลิกแท็บ Create
2. คลิก Report Wizard
3. เลือก Table / Query ที่จะใช้สร้างรายงาน
4. ดับเบิลคลิกเลือก field ที่จะใช้ หรือคลิกปุ่ม >> เพื่อเลือกทุก field อย่างรวดเร็ว

แล้วจึงคลิกปุ่ม Next

5. เลือก field ที่จะใช้จัดกลุ่ม
6. เลือก field ที่จะใช้เรียงข้อมูล แล้วจึงคลิกปุ่ม Next
7. เลือกรูปแบบการวาง field และการวางแนวกระดาษพิมพ์ แล้วจึงคลิกปุ่ม Next
8. ตั้งชื่อรายงาน
9. คลิก Finish



ภาพที่ 6.2 ขั้นตอนการสร้างรายงานโดยใช้ตัวช่วยสร้าง (Report Wizard)

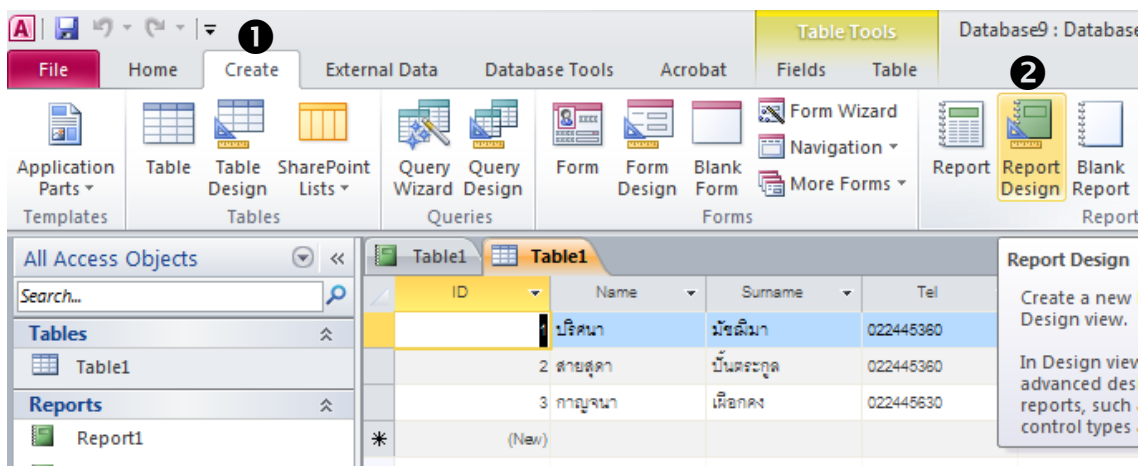


ภาพที่ 6.2 ขั้นตอนการสร้างรายงานโดยใช้ตัวช่วยสร้าง (Report Wizard)

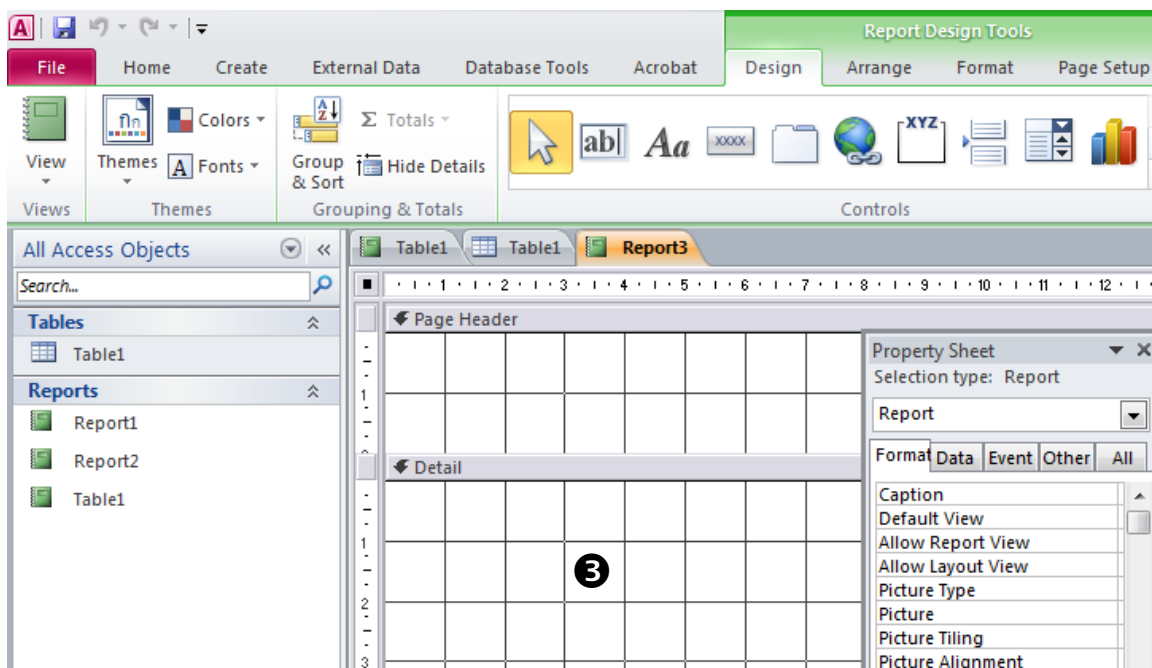
สร้างรายงานในมุมมองออกแบบ

วิธีสร้างรายงานในมุมมองออกแบบ (Design View) จะคล้ายกับการสร้างฟอร์มที่กล่าวมาแล้ว โดยสิ่งที่ต้องพิจารณาเพิ่มเติมจากฟอร์ม คือ การจัดกลุ่มและเรียงลำดับข้อมูล รวมทั้งการเลือกฟังก์ชันเพื่อคำนวณค่าผลสรุปของข้อมูล โดยการเริ่มสร้างรายงานดังนี้

1. เลือกแท็บ Create
2. คลิก Report Design
3. ได้พื้นที่สำหรับสร้างรายงาน



ภาพที่ 6.3 ขั้นตอนการสร้างรายงานด้วยมุมมองการออกแบบ (Report Design)

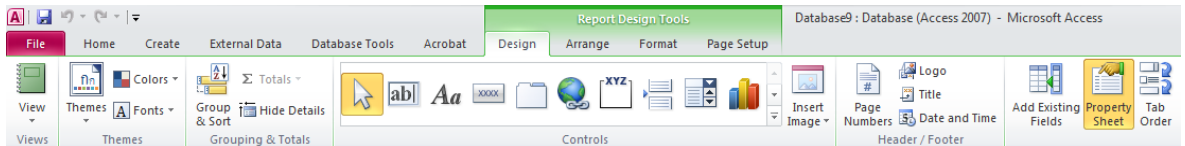


ภาพที่ 6.4 หน้าจอการสร้างรายงานด้วยมุมมองการออกแบบ (Report Design)

ส่วนประกอบในมุมมองออกแบบของรายงาน

แท็บคำสั่งที่ใช้สร้างหรือแก้ไขและจัดรูปแบบรายงาน ได้แก่

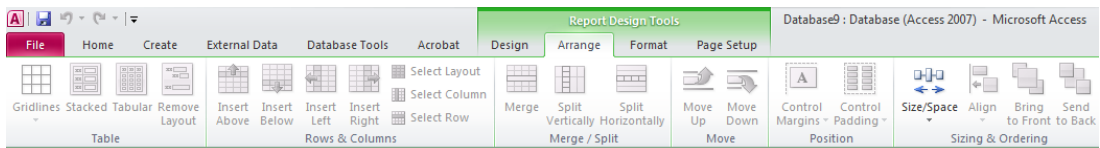
1. แท็บ Design



ภาพที่ 6.5 หน้าจอแท็บ Design ในการสร้างรายงานด้วยมุมมองการออกแบบ

คำสั่งในแท็บ Design ของ Report จะใช้สำหรับการสร้าง/แก้ไขหรือทำงานร่วมกับ Report ซึ่งการทำงานแต่ละคำสั่งจะเหมือนกับคำสั่งในแท็บ Design ของมุมมอง Layout โดยมีคำสั่งที่เพิ่มเข้ามา คือ Tab Order, Subreport in New Window, View Code และ Convert Report's Macros to Visual Basic

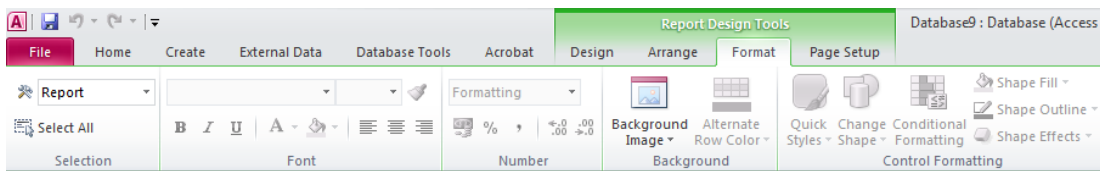
2. แท็บ Arrange



ภาพที่ 6.6 หน้าจอแท็บ Arrange ในการสร้างรายงานด้วยมุมมองการออกแบบ

การทำงานของคำสั่งในแท็บนี้ส่วนใหญ่จะเหมือนกับคำสั่งในแท็บ Arrange ของมุมมอง Layout มีคำสั่งที่เพิ่มเข้ามา คือ Size/Space, Align, Bring to Front และ Send to Back

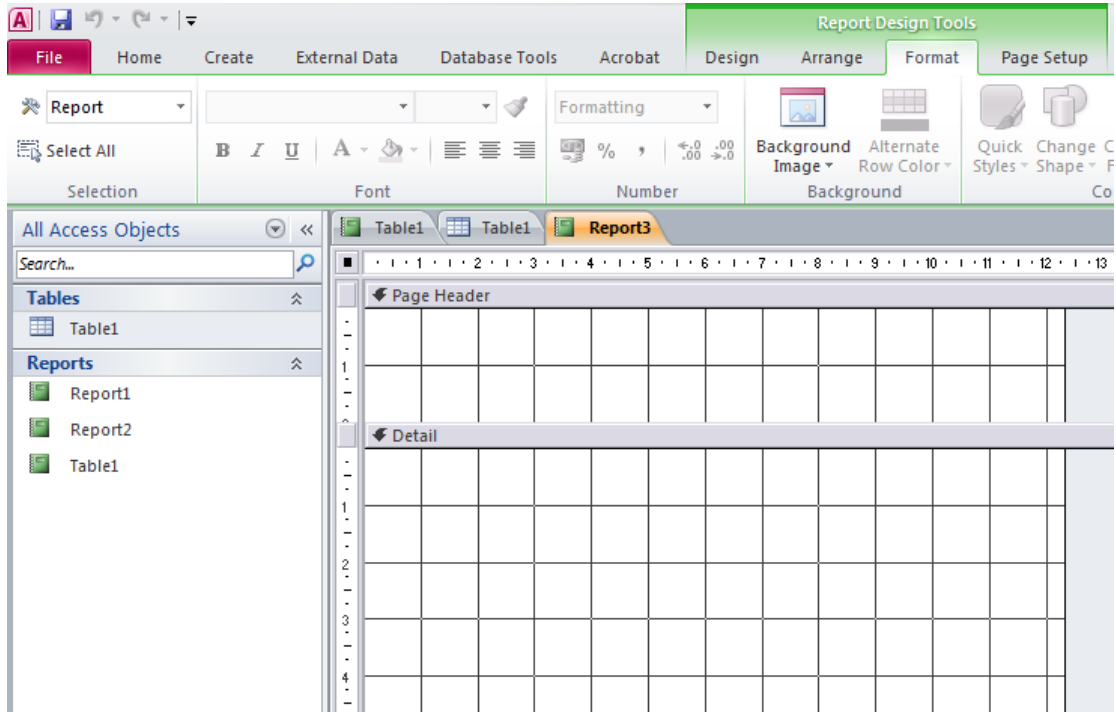
3. แท็บ Format



ภาพที่ 6.7 หน้าจอแท็บ Format ในการสร้างรายงานด้วยมุมมองการออกแบบ

คำสั่งในแท็บนี้จะเหมือนคำสั่งในแท็บ Format ของมุมมอง Layout ทุกคำสั่ง

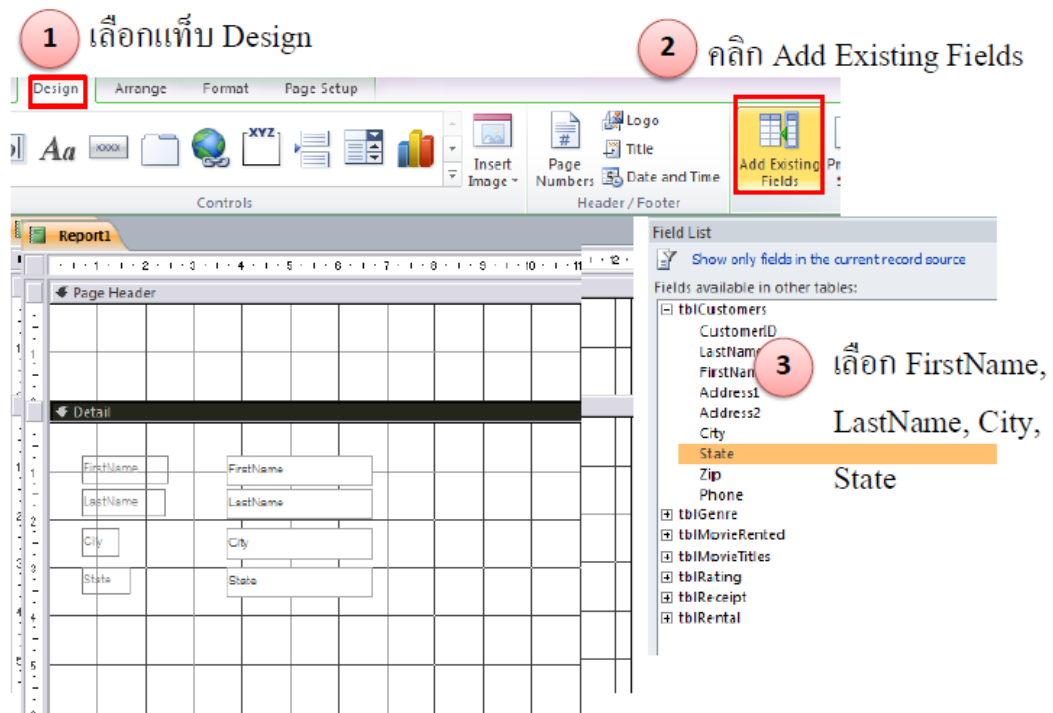
4. Report Design Area



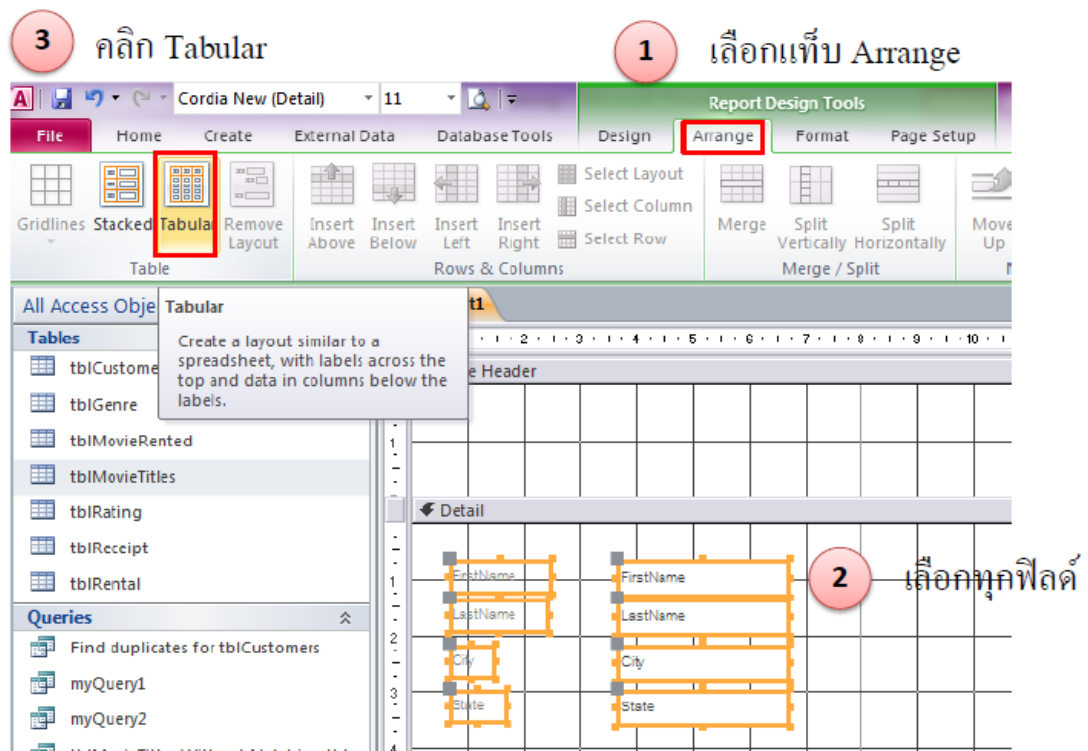
ภาพที่ 6.8 หน้าจอ Report Design Area ในการสร้างรายงานด้วยมุมมองการออกแบบ

ใช้สำหรับสร้างและออกแบบ Report ประกอบด้วยเซคชั่นต่างๆ เหมือนฟอร์ม เมื่อสร้างรายงานใหม่ว่างๆ มุมมองออกแบบ นอกจากเซคชั่น Detail แล้ว จะมีเซคชั่น Page Header และ Page Footer ด้วย นอกจากนี้ยังมีปุ่ม Add a group และ Add a sort เพิ่มเข้ามา เพื่อให้กำหนด field ที่จะใช้จัดกลุ่มและเรียงลำดับข้อมูลในรายงาน

ตัวอย่างการสร้างรายงาน



ภาพที่ 6.9 ตัวอย่างขั้นตอนการสร้างรายงาน



ภาพที่ 6.10 ตัวอย่างขั้นตอนการสร้างรายงานแบบ Tabular

1 หน้ารายงานจะเปลี่ยนไปอยู่ในรูปแบบตาราง

Report1															
Page Header															
FirstName				LastName				City				State			
Detail															
FirstName				LastName				City				State			

2 เปลี่ยนสี ขนาดตัวอักษร ตามต้องการ (เหมือนการเปลี่ยนในฟอร์ม)

Report1															
Page Header															
FirstName				LastName				City				State			
Detail															
FirstName				LastName				City				State			
Page Footer															

ภาพที่ 6.11 ตัวอย่างผลการสร้างรายงานแบบ Tabular

หลังจากลองทำตัวอักษรให้ใหญ่ขึ้นและมีสีคำ

FirstName	LastName	City	State
James	Accardo	Commerce Townshi	MI
James	Adams	Adamsvil	MI
Lisa	Alexander	MyTown	MI
Beverly	Allen	MyTown	MI
Timothy	Allen	MyTown	MI

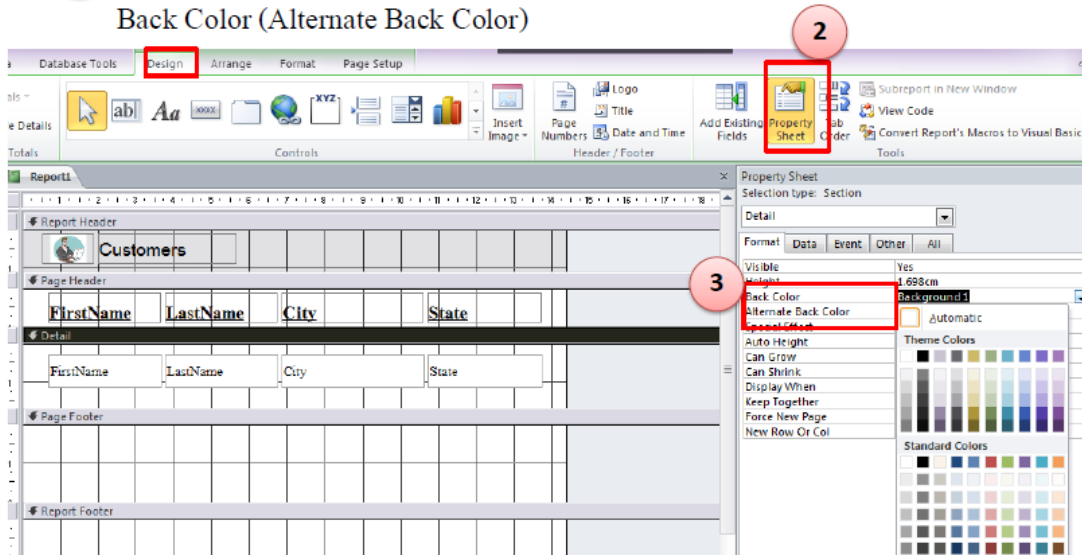
1 ใส่ Logo, Title, และขีดเส้นใต้ ทำตัวหนา

Report1			
Report Header			
Customers			
Page Header			
FirstName	LastName	City	State
Detail			
FirstName	LastName	City	State
Page Footer			
Report1			
Customers			
<u>FirstName</u>	<u>LastName</u>	<u>City</u>	<u>State</u>
James	Accardo	Commerce Township	MI
James	Adams	Adamsvil	MI
Lisa	Alexander	MyTown	MI
Beverly	Allen	MyTown	MI
Timothy	Allen	MyTown	MI
Valma	Allen	MyTown	MI

2

ภาพที่ 6.12 ตัวอย่างผลการแก้ไขเปลี่ยนแปลงรายงานแบบ Tabular

- 1 เปลี่ยนสีพื้นหลังของรายงาน Design > Property Sheet >
Back Color (Alternate Back Color)



ภาพที่ 6.13 ตัวอย่างขั้นตอนการเปลี่ยนสีพื้นหลังของรายงาน

ได้รายงานในรูปแบบ MS Access ให้บันทึกไว้ (Save)

Report1

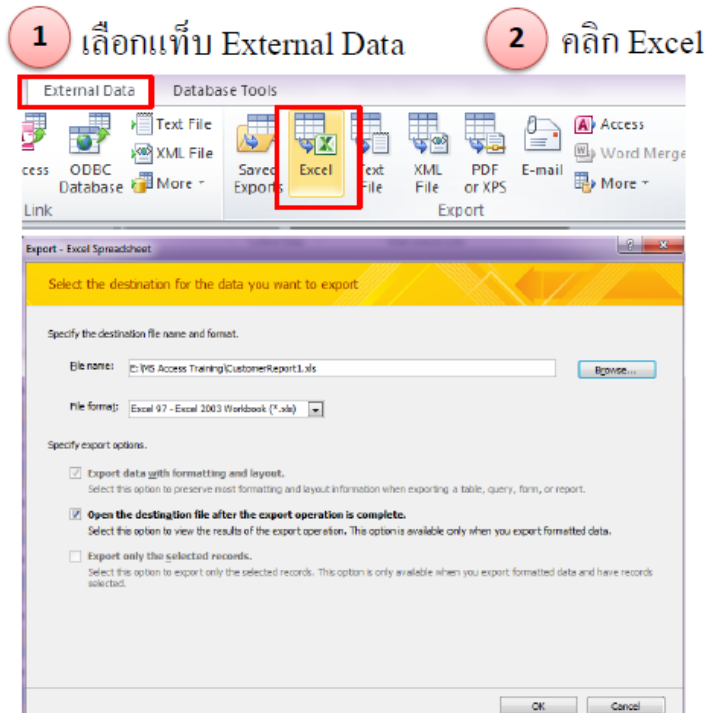
Customers

<u>FirstName</u>	<u>LastName</u>	<u>City</u>	<u>State</u>
James	Accardo	Commerce Township	MI
James	Adams	Adamsvil	MI
Lisa	Alexander	MyTown	MI
Beverly	Allen	MyTown	MI
Timothy	Allen	MyTown	MI
Velma	Allen	MyTown	MI
Michael	Allen	MyTown	MI

ภาพที่ 6.14 ตัวอย่างรายงาน

การ export รายงาน

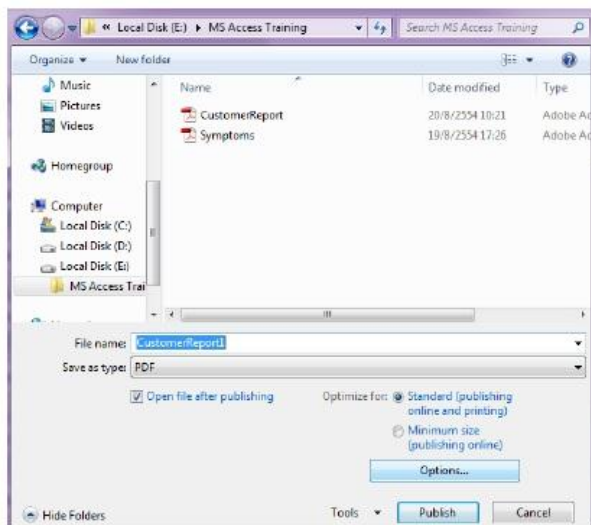
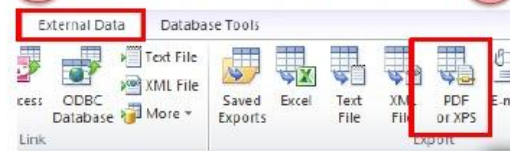
1. การ export รายงานให้อยู่ในรูป Excel



ภาพที่ 6.15 ขั้นตอนการ export รายงานให้อยู่ในรูป Excel

2. การ export รายงานให้อยู่ในรูป PDF

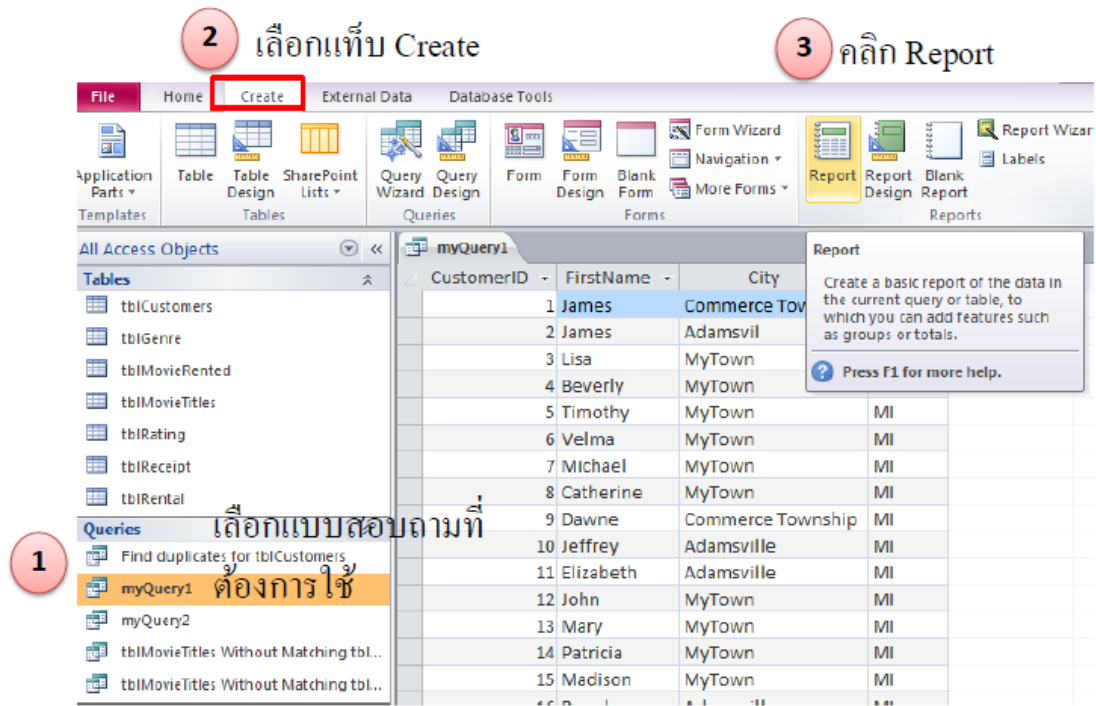
- 1 เลือกแท็บ External Data 2 คลิก PDF or XPS



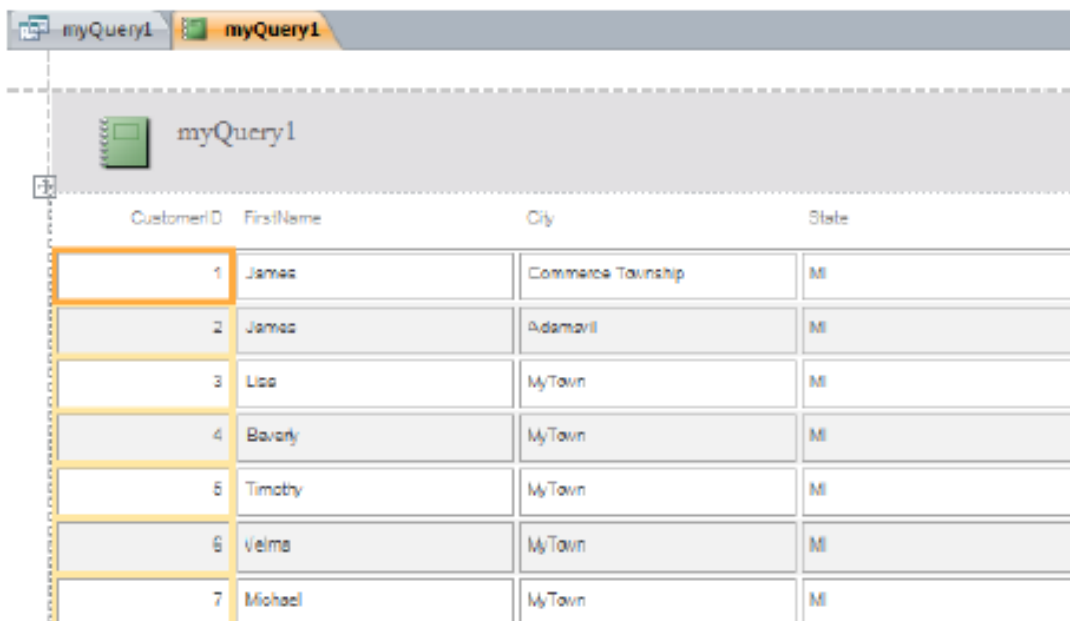
- 3 ใส่ชื่อ และ ที่เก็บ
คลิก “Open file ...” ถ้า
อยากให้เปิด PDF คู่มือทันที
หลังทำเสร็จ

ภาพที่ 6.16 ขั้นตอนการ export รายงานให้อยู่ในรูป PDF

วิธีสร้างรายงานจากแบบสอบถาม



MS Access จะทำรายงานให้โดยรวมเอาทุกๆ ฟิลด์ในแบบสอบถามเอาไว้



ภาพที่ 6.17 ขั้นตอนสร้างรายงานจากแบบสอบถาม

บทที่ 7

การสร้างมาโคร (Macro)

มาโครหรือแมโคร (Macro) เป็นออบเจกต์ฐานข้อมูลอีกตัวหนึ่งที่จะช่วยให้สามารถสร้างกลุ่มคำสั่งที่ทำงานแบบอัตโนมัติได้อย่างรวดเร็ว โดยการทำงานจะเรียงลำดับจากคำสั่งแรกไปจนถึงคำสั่งสุดท้ายในมาโครนั้น เหมาะที่จะนำไปใช้กับการทำงานที่ต้องทำซ้ำๆ กัน และไม่ซับซ้อนมาก โดยมาโครจะช่วยลดภาระในการจดจำคำสั่ง ช่วยให้ผู้ใช้ที่ไม่คุ้นเคยกับการใช้คำสั่งใน Access สามารถนำฐานข้อมูลไปใช้งานได้ง่ายขึ้น เช่น ผู้พัฒนาฐานข้อมูลอาจสร้างปุ่มคำสั่งที่ควบคุมการทำงานด้วยมาโครในลักษณะต่างๆ เช่น ปุ่มคำสั่งที่ใช้เปิดหรือปิดฟอร์ม ปุ่มคำสั่งที่ใช้ค้นหาข้อมูลที่ตรงกับเงื่อนไขที่กำหนดไว้ และปุ่มคำสั่งสำหรับพิมพ์รายงาน เป็นต้น โดยผู้ใช้เพียงคลิกปุ่มคำสั่งก็สามารถทำงานเหล่านั้นได้ โดยไม่ต้องเรียกใช้คำสั่งหลายๆ คำสั่งจาก Ribbon ซ้ำๆ กันทุกครั้ง

รู้จักกับมาโคร

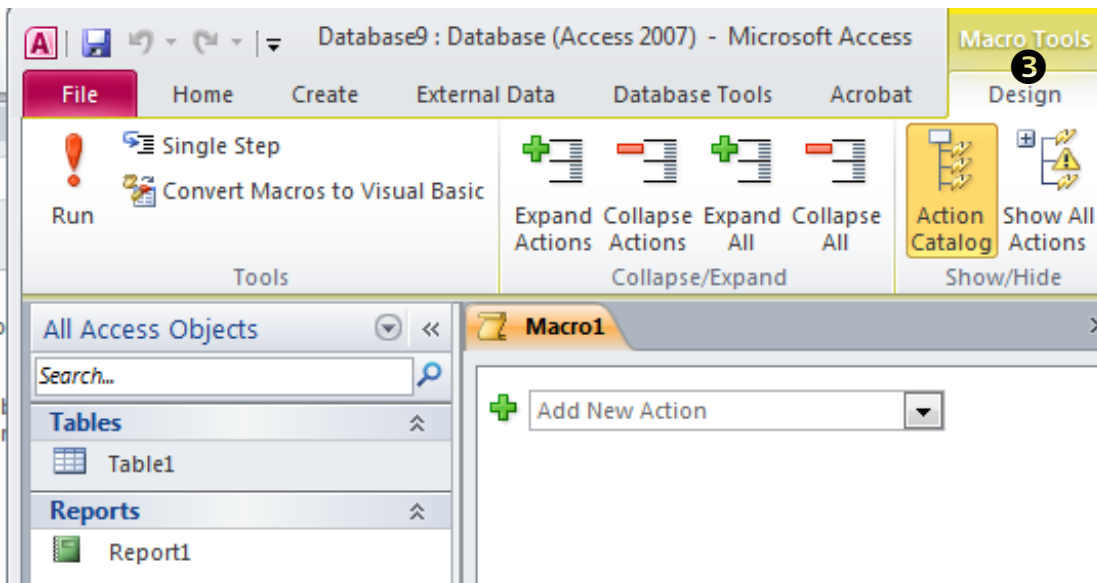
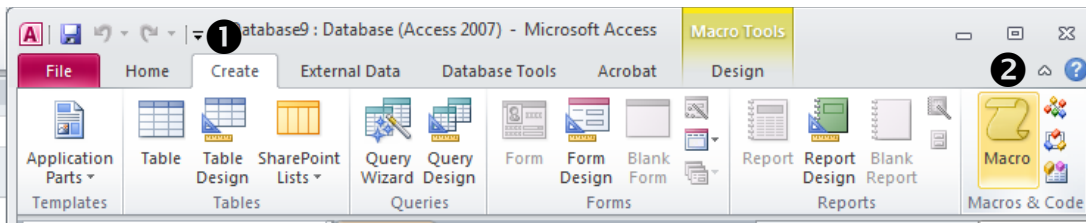
มาโคร (Macro) คือ ชุดคำสั่งที่ทำงานแบบอัตโนมัติ จัดเป็นโปรแกรมประเภทหนึ่งสามารถกำหนดเงื่อนไขการทำงานได้ เช่น ตรวจสอบค่าเพื่อเลือกทำหรือไม่ทำคำสั่งที่กำหนดไว้ หรือสามารถนำมาโครมาช่วยทำงานในลักษณะต่างๆ เช่น

- สามารถค้นหาและสอบถามข้อมูล ผ่านทางฟอร์มหรือรายงานโดยไม่ต้องเรียกใช้อ็อบเจกต์ Query โดยตรง
- สามารถตรวจสอบและแก้ไขข้อมูลที่ป้อนผ่านฟอร์ม การทำงานจะยืดหยุ่นกว่าการใช้กฎการตรวจสอบ (validation rule) ที่กำหนดในโครงสร้างของตาราง
- ช่วยปรับปรุงการทำงานแบบอัตโนมัติ เช่น การคัดลอก เคลื่อนย้าย และแก้ไขข้อมูล เป็นต้น
- ช่วยในการสร้างแอปพลิเคชันโปรแกรมขึ้นใช้งานเอง เช่น สร้างฟอร์มที่ประกอบด้วยปุ่มคำสั่งต่างๆ โดยมาโครเป็นตัวควบคุมการทำงาน

เริ่มต้นสร้างมาโคร

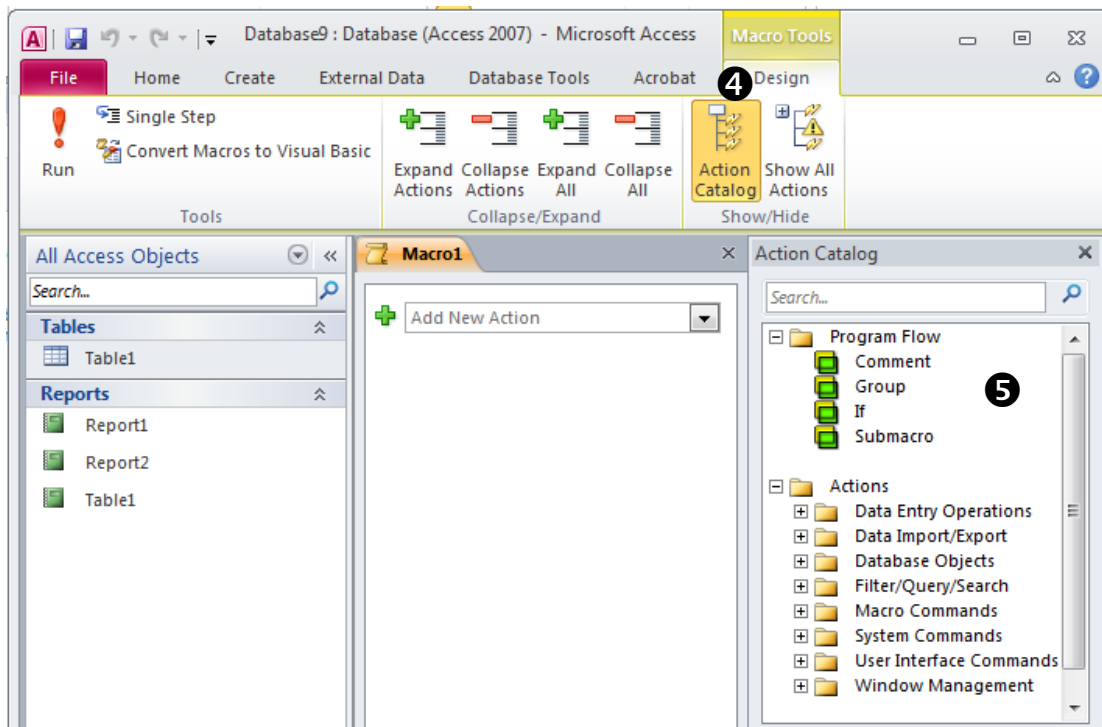
ขั้นตอนการสร้างมาโครเริ่มต้นจาก

1. คลิกแท็บ Create
2. คลิก Macro
3. เข้าสู่มุมมอง Design ของมาโคร



ภาพที่ 7.1 ขั้นตอนการสร้างมาโคร (Macro)

4. คลิกปุ่ม Action Catalog ในแท็บ Design
5. ปกรากฎหน้าต่าง Action Catalog ซึ่งประกอบด้วยกลุ่มคำสั่งที่จะนำมาใช้ในชุดคำสั่งอัตโนมัติ



ภาพที่ 7.2 หน้าต่าง Action Catalog

คำสั่งใน Action Catalog

ใน Action Catalog จะแบ่งคำสั่งออกเป็น 2 กลุ่มดังนี้

1. กลุ่ม Program Flow

- Comment คำสั่งที่ใช้ใส่ข้อความที่ผู้ใช้งานต้องการ เช่น หมายเหตุเตือนความจำ หรือคำอธิบายการใช้งาน

- Group คำสั่งสร้างมาโครแบบกลุ่ม
- If คำสั่งใช้สร้างเงื่อนไขการทำงานในมาโคร
- Submacro คำสั่งสร้างมาโครย่อย

2. กลุ่ม Action

- **Data Entry Operations** กลุ่มคำสั่งที่ใช้จัดการข้อมูล เช่น DeleteRecord (ลบเรคคอร์ด), SaveRecord (จัดเก็บเรคคอร์ด) และ EditItems (เรียกใช้คำสั่งแก้ไขข้อมูลในลิสต์ Lookup) เป็นต้น
- **Data Import/Export** กลุ่มคำสั่งที่ใช้ในการนำเข้าและส่งออกข้อมูลระหว่างฐานข้อมูล Access และโปรแกรมอื่นๆ เช่น EmailDatabaseObject (ส่งออบเจ็คจากฐานข้อมูล Access ไปทางอีเมล), ccExportWithFormatting (ส่งข้อมูลจาก Access ไปโปรแกรมอื่นๆ ตามรูปแบบที่กำหนดไว้), ImportExportSpreadsheet (นำเข้าหรือส่งออกข้อมูล Access กับไฟล์ข้อมูล spread sheet)
- **Database Objects** กลุ่มคำสั่งที่ใช้จัดการออบเจ็คฐานข้อมูล เช่น CopyObject (คัดลอกออบเจ็ค), DeleteObject (ลบออบเจ็คออกจากฐานข้อมูล), RenameObject (เปลี่ยนชื่อออบเจ็ค), PrintObject (พิมพ์ออบเจ็ค)
- **Filter/Query/Search** กลุ่มคำสั่งที่ใช้กรอง สอบถาม และค้นหาข้อมูล เช่น ApplyFilter (กำหนดเงื่อนไขการกรอง), FindNextRecord (ค้นหาเรคคอร์ดถัดไปที่ตรงกับเงื่อนไข), OpenQuery (เปิดใช้ Query)
- **Macro Commands** กลุ่มคำสั่งควบคุมการทำงานของมาโคร เช่น CancelEvent (ยกเลิกเหตุการณ์ที่ทำให้มาโครนั้นทำงาน), RunMacro (เรียกใช้มาโครอื่นๆ จากมาโครที่กำลังทำงาน), StopAllMacros (หยุดการทำงานของมาโครทั้งหมดที่รันค้างอยู่)
- **System Commands** กลุ่มคำสั่งที่เกี่ยวข้องกับการทำงานของระบบฐานข้อมูล เช่น CloseDatabase (ปิดการใช้งานฐานข้อมูล), DisplayHourglassPointer (เปลี่ยนรูปแบบตัวชี้เมาส์จากปกติเป็นนาฬิกาทรายขณะรันมาโคร), QuitAccess (ปิดการทำงานโปรแกรม Access), Sendkeys (ส่งรหัสคีย์บอร์ดโดยไม่ต้องกดคีย์บอร์ดจริงๆ ไปให้ Access หรือโปรแกรมอื่นๆ ที่ใช้งานอยู่), SetWarnings (ซ่อน/แสดงคำเตือนของโปรแกรม Access ระหว่างการทำงาน)
- **User Interface Commands** กลุ่มคำสั่งติดต่อกับผู้ใช้ เช่น AddMenu (เมนูที่ผู้ใช้สร้างเองเพื่อทำงานแทนเมนูหลัก), MessageBox (กล่องข้อความแจ้งข่าวสาร/คำเตือนให้ผู้ใช้ทราบ), ShowToolbars (ซ่อน/แสดงทูลบาร์ที่ผู้ใช้สร้างเอง)
- **Window Management** กลุ่มคำสั่งที่ใช้จัดการหน้าต่างการทำงาน เช่น Maximize (ขยายหน้าต่างที่กำลังใช้งานให้มีขนาดใหญ่ที่สุดหรือเต็มหน้าจอ), Minimize (ย่อ

หน้าต่างที่กำลังใช้งานลงมาให้มีขนาดเล็กที่สุด), MoveAndSizeWindow (ย้ายและปรับขนาดหน้าต่างที่กำลังใช้งาน)

วิธีสร้างมาโครโดยเก็บเป็นออบเจ็ค Macro

เข้ามุมมอง Design โดยคลิก Create ในแท็บ Macro แล้วทำดังนี้

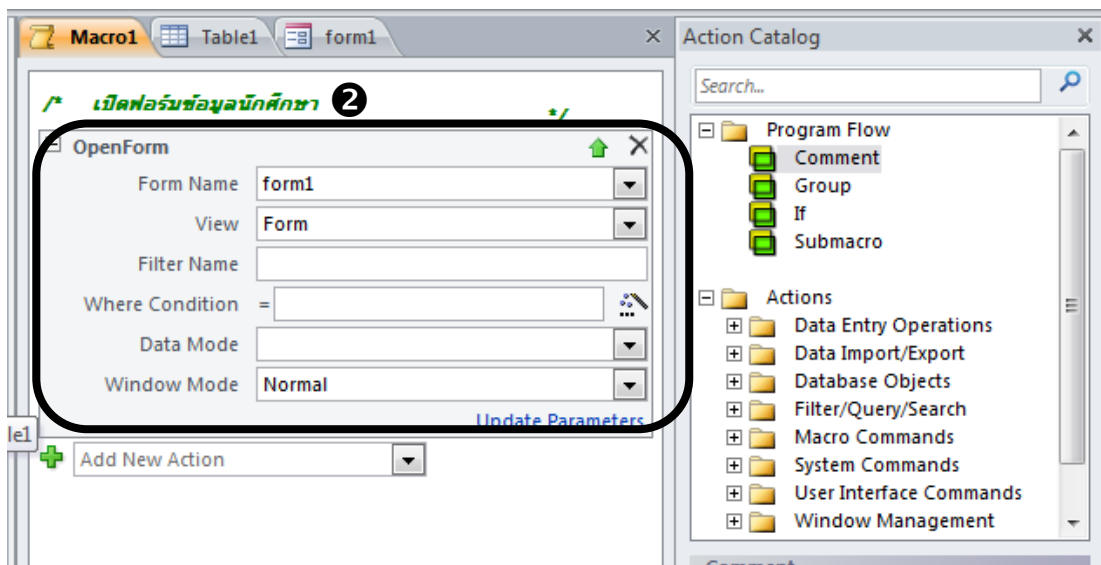
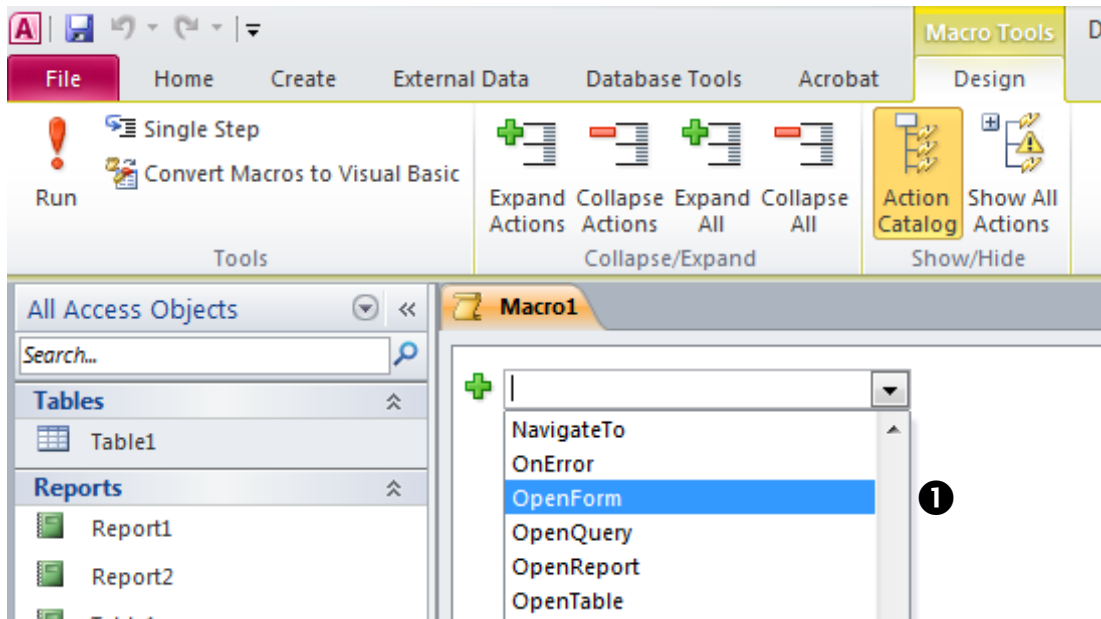
1. เลือกคำสั่งจาก Add New Action ในที่นี้เลือก OpenForm
2. ใส่ค่าอาร์กิวเมนต์ (ถ้ามี) ในที่นี้คือ อาร์กิวเมนต์ของคำสั่ง OpenForm
 - Form Name ใส่ชื่อฟอร์ม
 - View เปิดฟอร์มในมุมมองใด
 - Filter Name ใส่ชื่อตัวกรองข้อมูล (ถ้ามี)
 - Where Condition ใส่เงื่อนไขที่ใช้กรองข้อมูล (ถ้ามี)
 - Data Mode จะแสดงฟอร์มแบบเพิ่มเรคอร์ดได้ (Add) แก้ไขได้ (Edit) หรืออ่านได้

อย่างเดียว (Read Only)

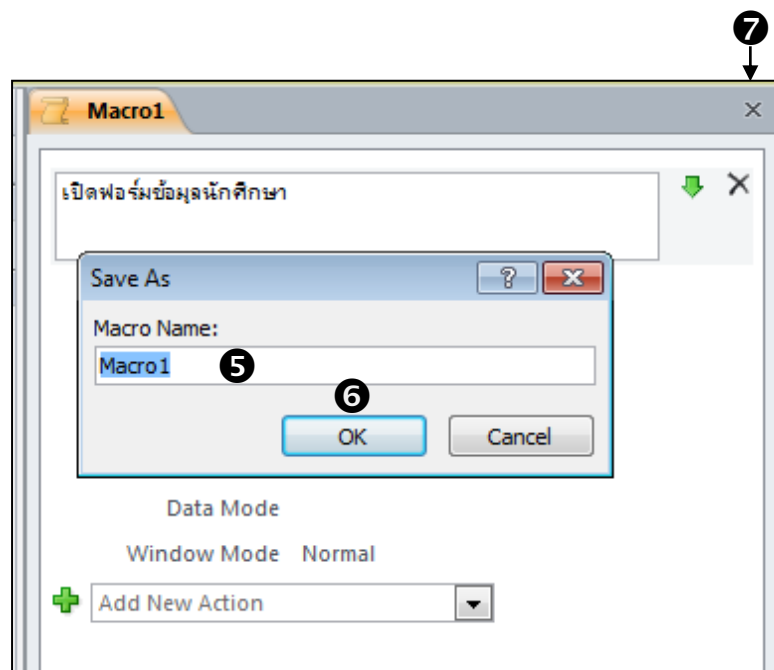
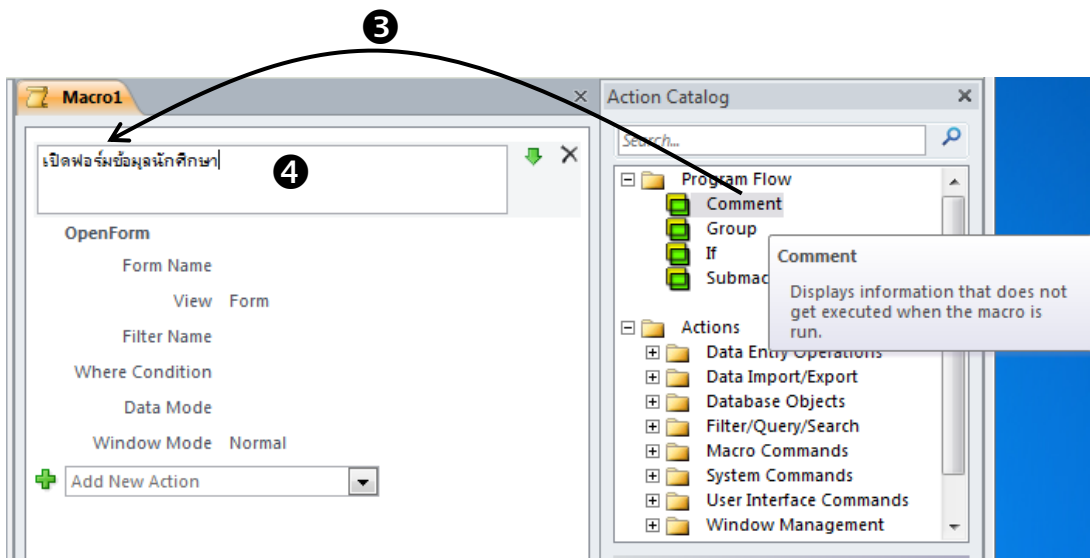
- Window Mode แสดงฟอร์มในมุมมองฟอร์ม (Normal) ซ่อนฟอร์ม (Hidden) ย่อ

ขนาดหน้าต่างให้เล็กที่สุด (Icon) หรือแบบไดอะล็อก (Dialog)

3. คลิกลากลคำสั่ง Comment มาวางถ้าจะใส่หมายเหตุ
4. คีย์ข้อความในช่อง Comment เสร็จแล้วคลิก Save
5. ตั้งชื่อมาโครในช่อง Macro Name: ในที่นี้ตั้งชื่อว่า Macro1
6. คลิก OK
7. คลิก Close เพื่อปิดมุมมองออกแบบ



ภาพที่ 7.3 ขั้นตอนการสร้างมาโครโดยเก็บเป็นออบเจ็ค Macro

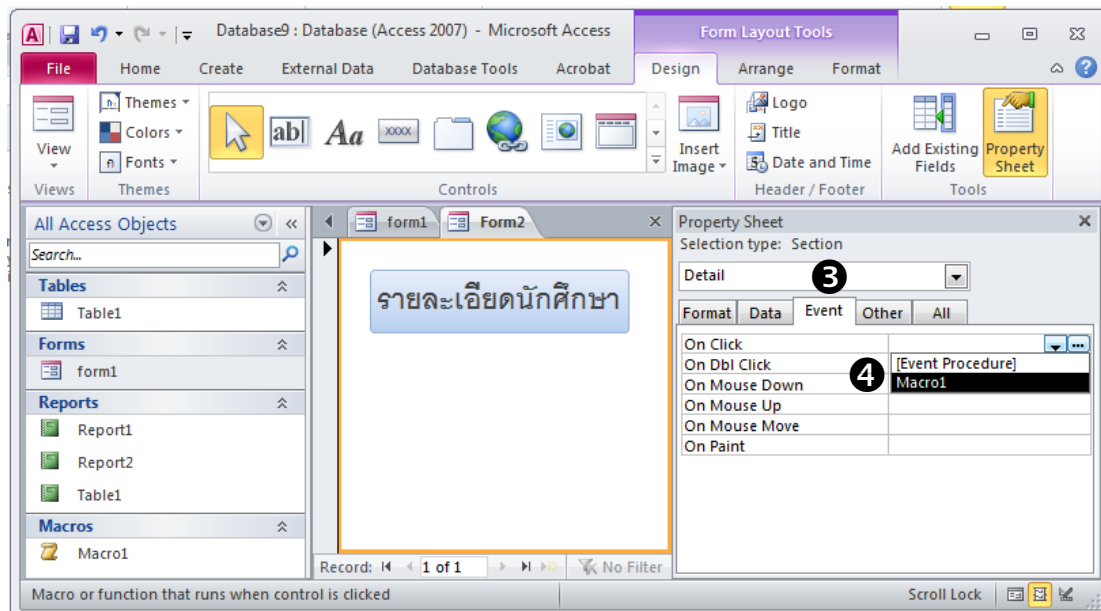
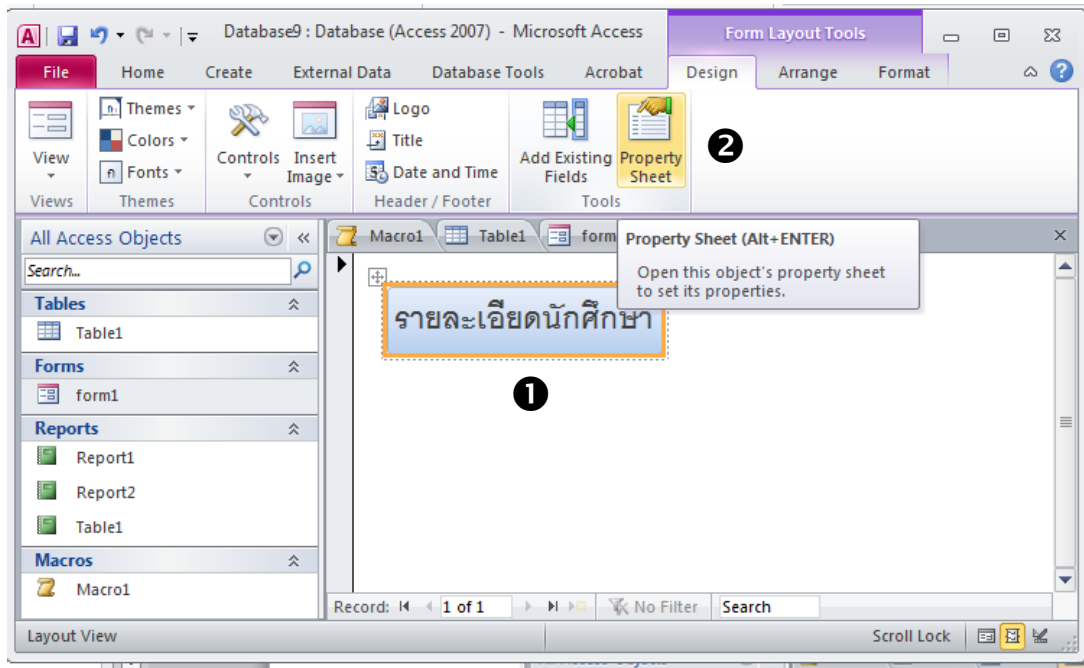


ภาพที่ 7.3 ขั้นตอนการสร้างมาโครโดยเก็บเป็นออบเจ็ค Macro

หากต้องการนำ Macro1 มาผูกกับปุ่มคำสั่ง ที่สร้างขึ้นจากฟอร์ม ทำตามขั้นตอนดังนี้

- 1) คลิกปุ่ม รายละเอียดนักศึกษา
- 2) คลิก Property Sheet
- 3) คลิกแท็บ Event

4) ในแถวคุณสมบัติ On Click ให้คีย์ชื่อ Macro1 หรือจะคลิกปุ่มลูกศรแล้วเลือกจากลิสต์ก็ได้



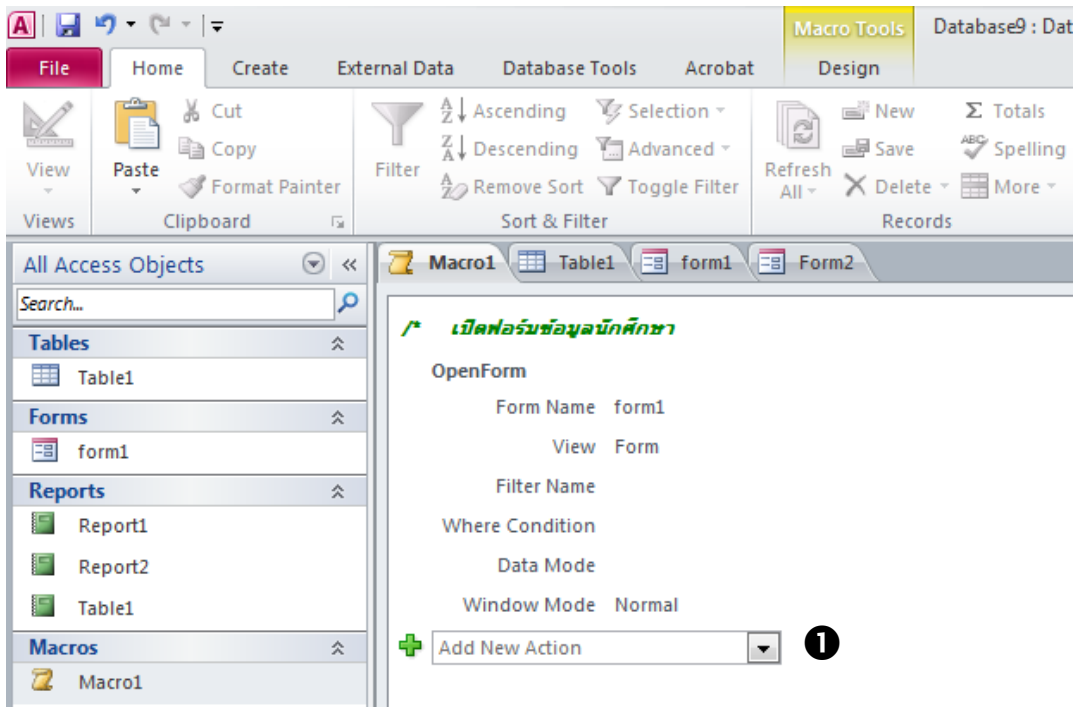
ภาพที่ 7.4 ขั้นตอนการนำ Macro1 มาผูกกับปุ่มคำสั่ง ที่สร้างขึ้นจากฟอร์ม

การแก้ไขมาโคร

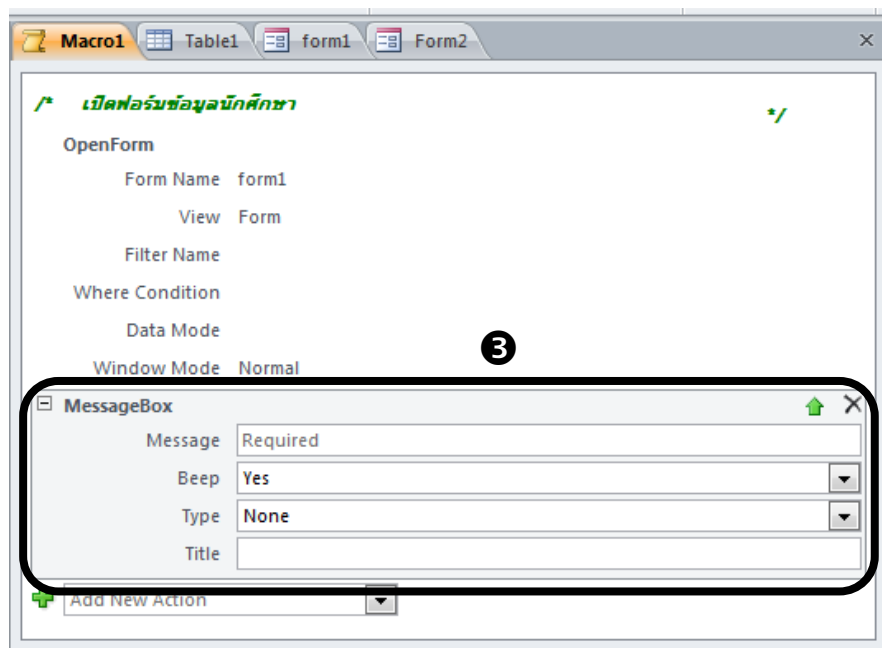
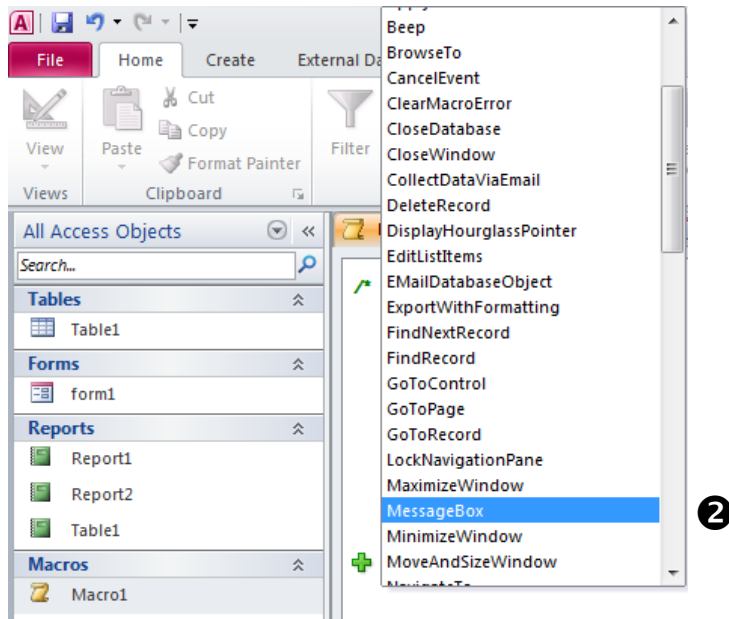
1. เพิ่มคำสั่งใหม่

ตัวอย่างเช่น หากต้องการนำ Macro1 มาเพิ่มคำสั่ง MessageBox ก่อนคำสั่ง OpenForm เพื่อแสดง dialog box ก่อนเปิดฟอร์มข้อมูลนักศึกษา มีขั้นตอนดังนี้

- 1) คลิกเลือกคำสั่งที่จะใช้จากช่อง Add New Action
- 2) เลือกคำสั่ง MessageBox
- 3) คำสั่งใหม่จะอยู่ท้ายสุดของมาโคร



ภาพที่ 7.5 ขั้นตอนเพิ่มคำสั่งใหม่ในมาโคร

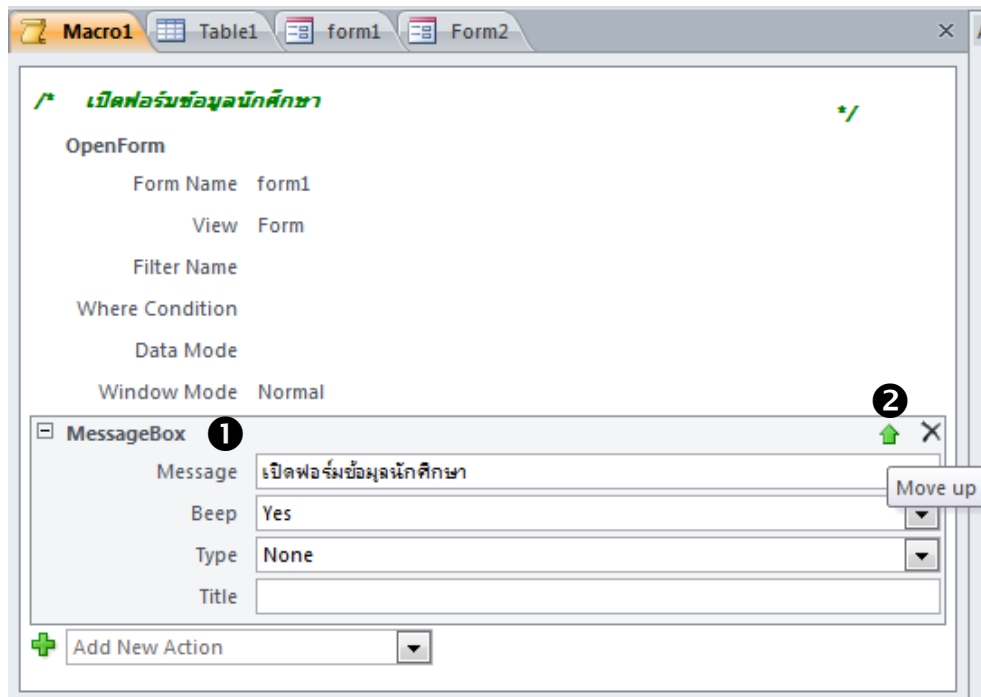


ภาพที่ 7.5 ขั้นตอนเพิ่มคำสั่งใหม่ในมาโคร

2. ย้ายตำแหน่งคำสั่ง

ถ้าเพิ่มคำสั่ง MessageBox ด้วยวิธีคลิกเลือกจากช่อง Add New Action คำสั่ง MessageBox จะอยู่ที่ท้ายสุดของมาโคร ดังนั้นหากต้องการให้รัน MessageBox ก่อนต้องทำการย้ายตำแหน่ง MessageBox ขึ้นไปด้านบน ด้วยวิธีดังนี้

- 1) คลิกคำสั่งที่จะย้ายตำแหน่ง
- 2) คลิกปุ่ม Move up

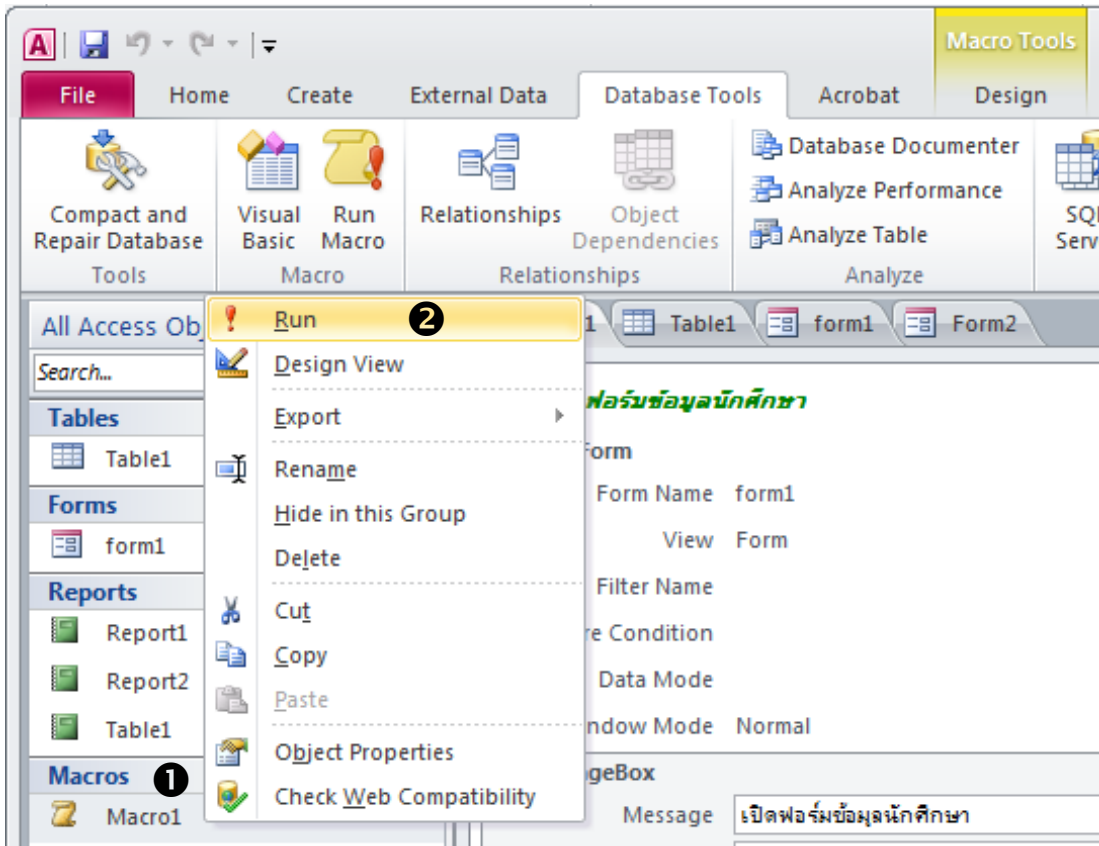


ภาพที่ 7.6 ขั้นตอนการย้ายตำแหน่งคำสั่งในมาโคร

วิธีสั่งรันมาโคร

วิธีที่ 1 สั่งรันมาโครจาก Navigation Pane

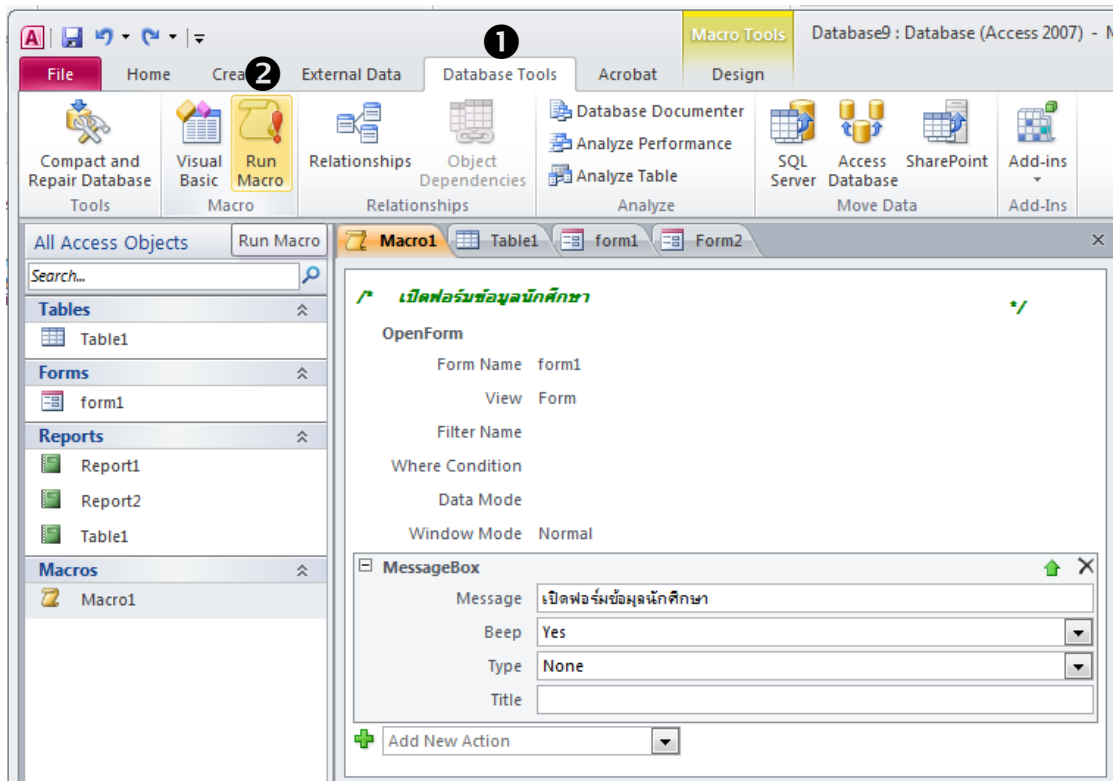
1. คลิกขวาที่ชื่อของแมคโร Macro
2. คลิกคำสั่ง Run จากเมนูลัด



ภาพที่ 7.7 ขั้นตอนการสั่งรันมาโครจาก Navigation Pane

วิธีที่ 2 สั่งรันจากคำสั่ง Run Macro

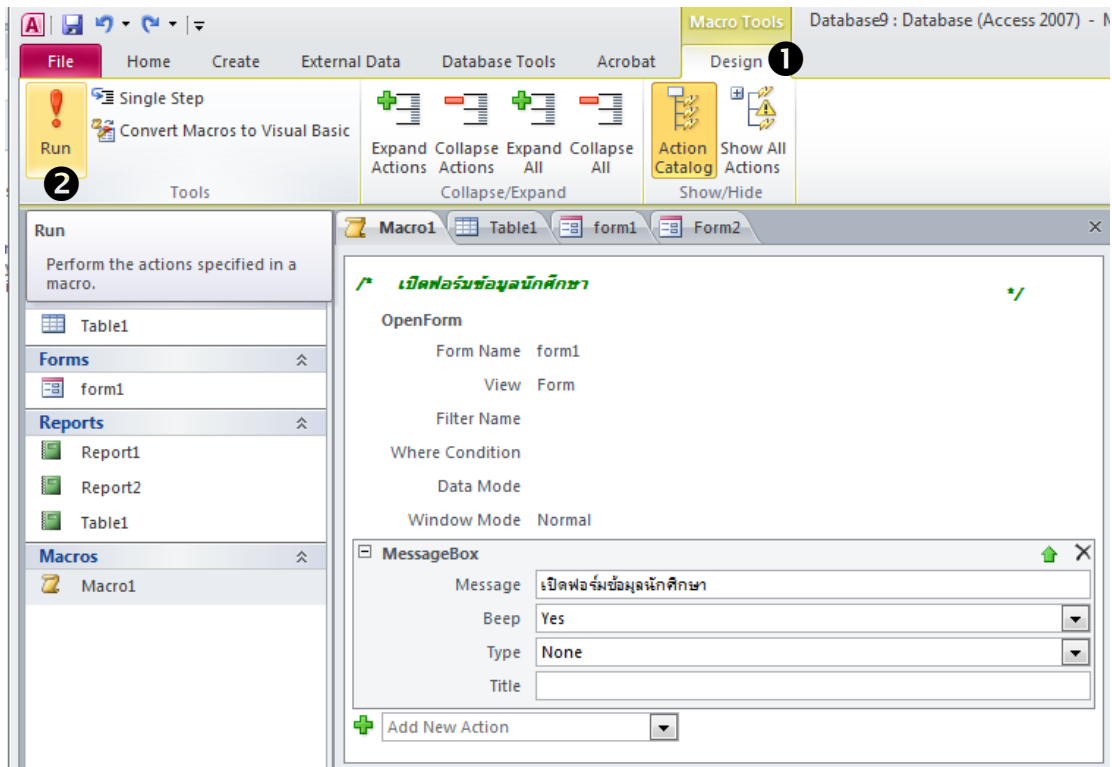
1. คลิกแท็บ Database Tools
2. คลิก Run Macro



ภาพที่ 7.8 ขั้นตอนการสั่งรันมาโครจากคำสั่ง Run Macro

วิธีที่ 3 สั่งรันในมุมมอง Design

1. เลือกมุมมอง Design
2. คลิกคำสั่ง Run ในแท็บ Design



ภาพที่ 7.9 ขั้นตอนการสั่งรันมาโครจากมุมมอง Design

บรรณานุกรม

- กิตติ ภัคดีวัฒนะกุล และจำลอง คุรุอุตสาหะ. **การออกแบบฐานข้อมูล**. กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2546.
- นันทน์ แขวงโสภา. (2554). **คู่มือ Access 2010 ฉบับสมบูรณ์**. กรุงเทพฯ: โปรวิชั่น.
- เบญจรัตน์ ภูภักดี. (2554). **Design, Construction, and Management of Databases with Microsoft Access 2010**. กรุงเทพฯ: สถาบันวิจัยจุฬาภรณ์.
- มณีโชติ สมานไทย. **คู่มือการออกแบบฐานข้อมูลและภาษา SQL ฉบับผู้เริ่มต้น**. นนทบุรี : อินโฟเพรส, 2546.
- วิเชียร เปรมชัยสวัสดิ์. “การออกแบบเชิงแนวคิดโดยใช้ อี-อาร์ โมเดล,” ใน **เอกสารการสอนชุดวิชาการจัดการระบบฐานข้อมูล หน่วยที่ 1-8**. หน้า 241-276. นนทบุรี : มหาวิทยาลัยสุโขทัยธรรมาธิราช, 2546.
- วิเชียร เปรมชัยสวัสดิ์. **ระบบฐานข้อมูล**. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2546.
- วิภา เจริญภัณฑารักษ์. “ทฤษฎีการออกแบบโดยวิธีการนอร์มัลไลเซชัน,” ใน **เอกสารการสอนชุดวิชาการจัดการระบบฐานข้อมูล หน่วยที่ 1-8**. หน้า 279-341. นนทบุรี : มหาวิทยาลัยสุโขทัยธรรมาธิราช, 2546.
- ศิริลักษณ์ โรจนกิจอำนวย. **การออกแบบและบริหารฐานข้อมูล**. กรุงเทพฯ : ดวงกลมสมัย, 2545.
- สมจิตร อาจอินทร์ และงามนิช อาจอินทร์. **ระบบฐานข้อมูล**. ขอนแก่น : ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น, 2543.
- โอบาส เอียมสิริวงศ์. **การออกแบบและจัดการฐานข้อมูล**. กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2546.
- Connolly, Thomas M. and Begg, Carolyn E. **Database systems : a practical approach to design, implementation, and management**. Essex : Addison-Wesley, 2002.
- Kroenke, David M. **Database processing : fundamentals, design and implementation**. N.J. : Prentice Hall, 2002.
- Mannino, Michael V. **Database application development and design**. Boston : McGrae-Hill/Irvin, 2001.

- Post, Gerald V. **Database management systems : designing and building business applications.** Boston : McGraw-Hill/Irwin, 2002.
- Pratt, Philip J. and Adamski, Joseph J. **Concepts of database management.** Australia : Course Technology, 2000.
- Ramakrishnan, Raghu. **Database management systems.** Boston : McGraw-Hill, 2003.
- Rob, Peter and Coronel, Carlos. **Database System: Design, Implementation, and Management.** Australia : Course Technolog/Thomson Learningy, 2002.
- Silberschatz, Abraham, Korth, Henry F. and Sudarshan, S. **Database system concepts.** Boston : McGraw-Hill, 2002.
- Stephens, Ryan K. and Plew, Ronald R. **Database design.** Indianapolis : Sams, 2001.