



จุดเด่นหลักของฐานข้อมูลเชิงสัมพันธ์ (Relational Database) คือ ความสามารถในการเชื่อมโยงความสัมพันธ์ระหว่างข้อมูล อย่างไรก็ตาม ผู้ใช้งานฐานข้อมูลหลายคนยังขาดความเข้าใจในการใช้งานความสามารถของฐานข้อมูลนี้ให้เกิดประสิทธิภาพ ในหน่วยนี้จะมุ่งเน้นไปที่วิธีการสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูลของโปรแกรม Microsoft Access



1. ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์
2. Database key ที่ใช้ในการสร้างความสัมพันธ์
3. การสร้างความสัมพันธ์ระหว่างตาราง



1. แสดงความรู้เกี่ยวกับการสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูลเชิงสัมพันธ์
2. มีทักษะการใช้ Database key ในการสร้างความสัมพันธ์ระหว่างตาราง
3. ประยุกต์ใช้โปรแกรม Microsoft Access ในการสร้างความสัมพันธ์ระหว่างตาราง



## จุดประสงค์การเรียนรู้

1. อธิบายเกี่ยวกับความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์เพื่อการสร้างความสัมพันธ์ระหว่างตารางได้
2. ใช้ Database key เพื่อสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูลได้
3. มีเจตคติและกิจนิสัยที่ดีในการปฏิบัติงานด้วยความละเอียด รอบคอบ และถูกต้องในการสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูล
4. ประยุกต์ใช้โปรแกรม Microsoft Access เพื่อสร้างความสัมพันธ์ระหว่างตารางได้อย่างมีประสิทธิภาพ



# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

**ฐานข้อมูลเชิงสัมพันธ์** หมายถึง ฐานข้อมูลที่มีการเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลในแต่ละวัตถุ (ตาราง, การสืบค้น) เข้าไว้ด้วยกัน ทำให้สามารถดึงข้อมูลจากหลายตารางมาใช้งานพร้อมกันได้ รวมถึงเมื่อมีการแก้ไขข้อมูล หรือการลบข้อมูลในตารางใด ข้อมูลในตารางอื่นที่มีความสัมพันธ์กับข้อมูลที่ถูกเปลี่ยนแปลงจะได้รับการแก้ไขตามไปด้วย

## ลักษณะของฐานข้อมูลเชิงสัมพันธ์

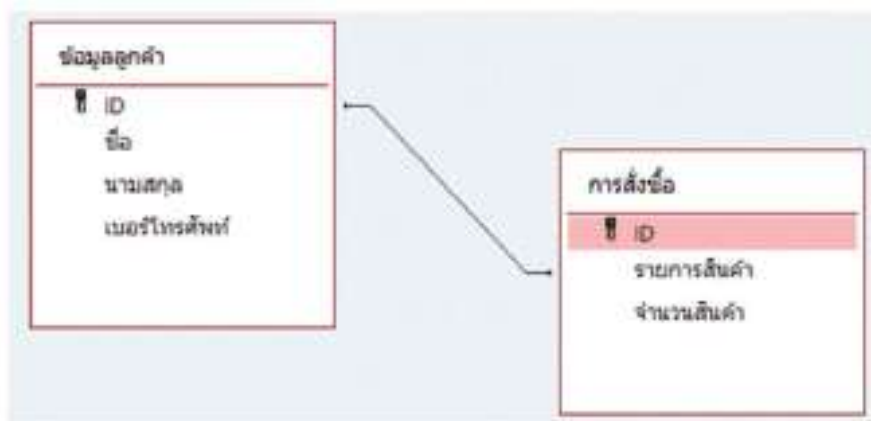
ตารางที่สามารถสร้างความสัมพันธ์ระหว่างกันได้จะต้องมี Common Field หรือ Field ที่เหมือนกันอยู่ในทั้งสองตาราง โดยในตารางหนึ่งจะมี Field ที่เป็นคีย์หลัก (Primary key) และอีกตารางหนึ่งจะมี Field ที่เป็นคีย์นอก (Foreign key) ซึ่งใช้เชื่อมโยงกับคีย์หลักอยู่ Common field ของทั้งสองตารางจะต้องมีชนิดของข้อมูลที่เหมือนกันและโดยทั่วไปแล้วควรตั้งชื่อ Common field ให้เหมือนกัน



# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

## ลักษณะของฐานข้อมูลเชิงสัมพันธ์

ตัวอย่างของฐานข้อมูลเชิงสัมพันธ์ เช่น ฐานข้อมูลประกอบด้วยตาราง 2 ตาราง คือ ตารางข้อมูลลูกค้าและอีกตาราง คือ ตารางการสั่งซื้อเมื่อผู้ใช้สร้างความสัมพันธ์ระหว่างข้อมูลในตารางทั้งสองนี้แล้วทำการปรับเปลี่ยนข้อมูลในตารางหนึ่ง ข้อมูลในอีกตารางซึ่งมีความสัมพันธ์อยู่กับข้อมูลในตารางแรกจะมีการเปลี่ยนแปลงตามไปด้วย





# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

## ลักษณะของฐานข้อมูลเชิงสัมพันธ์

จากตัวอย่างเป็นการสร้างความสัมพันธ์ระหว่างชื่อลูกค้ากับจำนวนสินค้าที่สั่งซื้อ เมื่อทำการเปลี่ยนแปลงข้อมูลในตารางข้อมูลลูกค้า ข้อมูลในตารางการสั่งซื้อจะมีการเปลี่ยนแปลงตามไปด้วยตามลักษณะของความสัมพันธ์ที่กำหนด

## รูปแบบของความสัมพันธ์ของข้อมูล

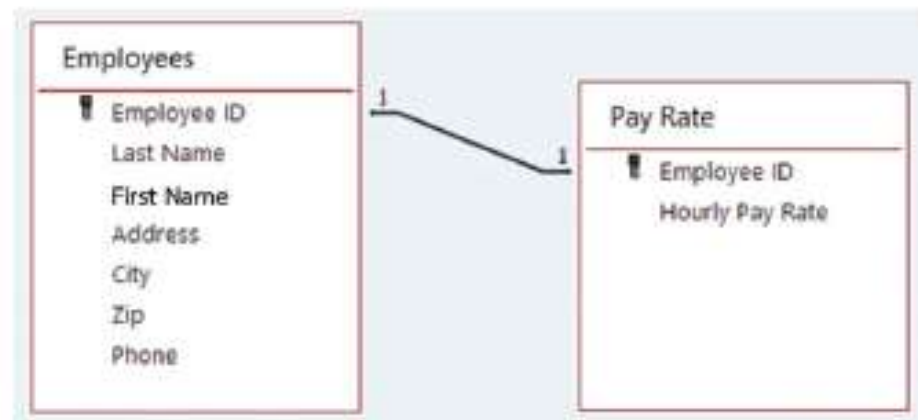
โปรแกรม Microsoft Access รองรับรูปแบบของความสัมพันธ์ของข้อมูลทั้งหมด 3 แบบดังได้กล่าวมาแล้วในหน่วยที่ 1



# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

## รูปแบบของความสัมพันธ์ของข้อมูล

1. ความสัมพันธ์แบบ 1 ต่อ 1 (One - to - One) เป็นความสัมพันธ์ที่ไม่ค่อยใช้งานมาก คือการที่ Record ในตารางหลักมีความสัมพันธ์กับ Record ในตารางอื่นเพียง Record เดียวเท่านั้น ตัวอย่างเช่นการเก็บข้อมูลบางส่วนของ Record ไว้เป็นความลับ โดยการนำข้อมูลที่เป็นความลับนั้นใส่ไว้ในอีกตารางหนึ่งซึ่งมีความสัมพันธ์กับตารางหลัก และมีการจำกัดการเข้าถึงข้อมูลในตารางนี้





# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

## รูปแบบของความสัมพันธ์ของข้อมูล

จากรูปเป็นความสัมพันธ์ระหว่างตาราง Employees และ Pay Rate โดยข้อมูลค่าจ้างรายชั่วโมง(Hourly Pay Rate) นั้นถูกเก็บไว้เป็นความลับ จึงมีการกำหนดความสัมพันธ์ระหว่างคีย์หลักและคีย์นอกเป็นแบบ 1 ต่อ 1

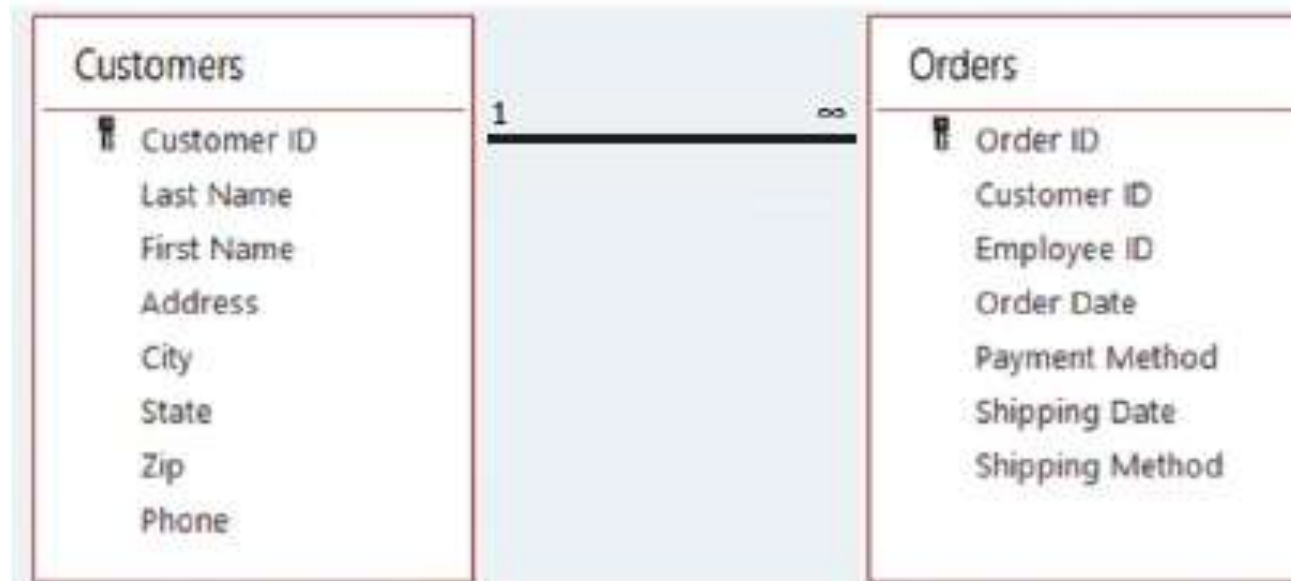
**2. ความสัมพันธ์แบบ One - to - Many** เป็นความสัมพันธ์ที่พบได้ทั่วไป โดยหนึ่ง Record ในตารางหลักสามารถเชื่อมโยงกับหลาย ๆ Record ในตารางอื่น ๆ ได้ แต่ในทางกลับกัน Record ในตารางอื่น ๆ สามารถเชื่อมโยงกับ Record ในตารางหลักได้เพียง Record เดียวเท่านั้น ตัวอย่างเช่นลูกค้า 1 คน สามารถสั่งซื้อได้หลาย Order แต่ Order แต่ละ Order จะถูกสั่งซื้อจากลูกค้าคนเดียว





# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

## รูปแบบของความสัมพันธ์ของข้อมูล

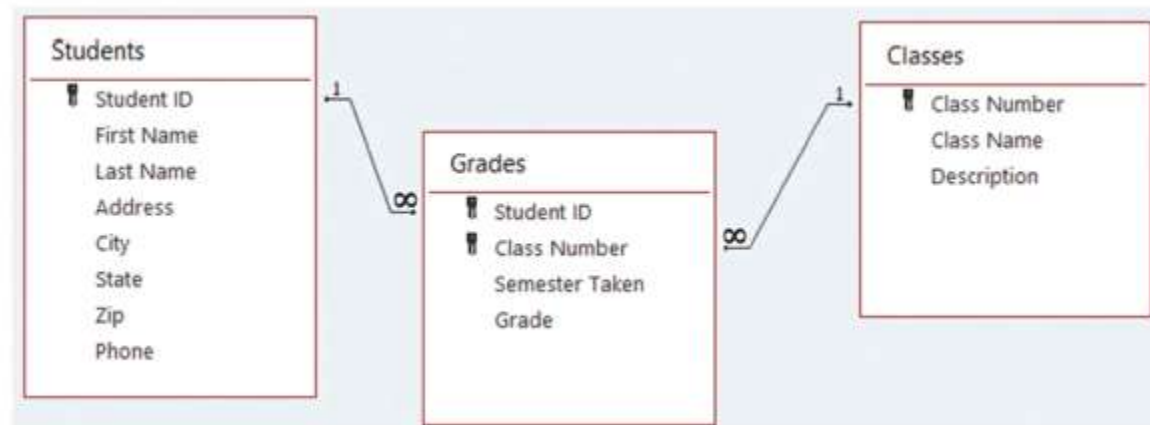




# ความหมายและลักษณะของฐานข้อมูลเชิงสัมพันธ์

## รูปแบบของความสัมพันธ์ของข้อมูล

3. ความสัมพันธ์แบบ Many - to - Many เป็นความสัมพันธ์ที่พบบ่อย ตัวอย่างเช่น นักเขียนสามารถเขียนหนังสือได้หลายเล่ม ในขณะเดียวกัน หนังสือหนึ่งเล่มอาจมีผู้แต่งหลายคน เป็นต้น ความสัมพันธ์รูปแบบนี้ไม่สามารถสร้างขึ้นได้โดยตรง จะต้องใช้ “Junction table” ในการสร้างความสัมพันธ์





# Database Key ที่ใช้ในการสร้างความสัมพันธ์

สิ่งที่ใช้ในการเชื่อมโยงความสัมพันธ์ของข้อมูลระหว่างตาราง คือ Database key โดย Database key ที่ใช้มีดังนี้

## คีย์หลัก (Primary Key)

ในทุก ๆ ตารางควรมีอย่างน้อย 1 Field ที่เป็นคีย์หลัก โดยค่าของคีย์หลักนี้ต้องเป็นค่าเฉพาะของแต่ละ Record ซึ่งไม่ซ้ำกับค่าอื่น ๆ ในฐานข้อมูล ตัวอย่างเช่น สมมุติว่ามีฐานข้อมูลของพนักงานในบริษัทและต้องการเลือก Field ในตาราง Employee เพื่อใช้เป็นคีย์หลักในการระบุข้อมูลของพนักงานแต่ละคนการเลือกใช้ชื่อของพนักงานอาจจะเป็นตัวเลือกที่ไม่เหมาะสม เพราะมีความเป็นไปได้ที่พนักงานของบริษัทอาจมีชื่อที่เหมือนกัน ส่งผลให้การใช้ชื่อพนักงานเป็นคีย์หลักไม่มีความเฉพาะตัวของข้อมูล ทางเลือกที่เหมาะสมกว่า คือ การใช้รหัสพนักงานเป็นคีย์หลัก เนื่องจากรหัสของพนักงานแต่ละคนจะมีค่าเฉพาะสำหรับพนักงานแต่ละคน



# Database Key ที่ใช้ในการสร้างความสัมพันธ์

## คีย์หลัก (Primary Key)

เมื่อทำการเลือกค่าคีย์หลักแล้ว ระบบจัดการฐานข้อมูลจะบังคับใช้ค่าเฉพาะของคีย์หลัก หากพยายามเพิ่ม Record ที่ซ้ำกันลงในตารางที่มีคีย์หลักจะไม่สามารถทำได้ โปรแกรมจัดการฐานข้อมูลส่วนมากสามารถสร้างคีย์หลักขึ้นมาเองได้ เช่น Microsoft Access จะใช้ข้อมูลชนิด AutoNumber ในการสร้างคีย์หลักให้กับแต่ละ Record

The screenshot shows the Microsoft Access interface. The 'Fields' ribbon is active, displaying options like 'Name & Caption', 'Default Value', 'Field Size', 'Data Type', and 'Format'. The 'Data Type' is set to 'AutoNumber'. Below the ribbon, a table named 'Student Info' is displayed with the following data:

ID	First Name	Last Name	e-mail Address	Student ID	Click to Add
1	Kasira	Prapasnobol	kasira2538@gmail.com	1504596	
2	Chananyou	Jao	sp22.scha@gmail.com	1505087	
3	Maliwan	Yondun	dow.ruk.toon@gmail.com	1504595	
4	Rungnapa	Uhnha chote	yingziijibijibieiei.56@gmail.com	1504600	
5	Sornsuk	Kedsuke	gognza1111@gmail.com	1505094	



# Database Key ที่ใช้ในการสร้างความสัมพันธ์

## คีย์นอก (Foreign key)

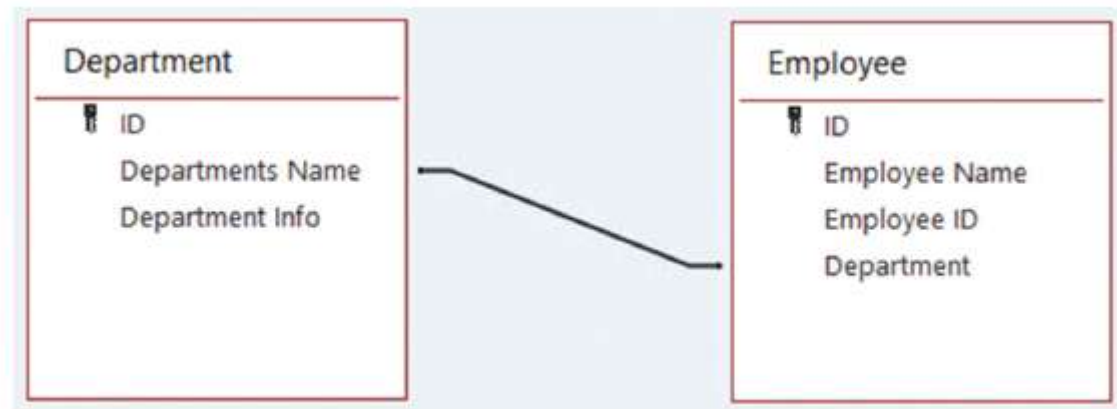
โดยทั่วไปแล้วโครงสร้างฐานข้อมูลจะมีความสัมพันธ์ระหว่างตาราง สามารถใช้คีย์นอกเป็นตัวเชื่อมความสัมพันธ์ระหว่างตารางนั้น ย้อนกลับไปยังตัวอย่างฐานข้อมูลของพนักงานบริษัทข้างต้น สมมติว่าต้องการเพิ่มตารางข้อมูลของแผนกลงในฐานข้อมูลโดยใช้ชื่อตาราง Department และในตารางนี้มีข้อมูลของทุก ๆ แผนก หากต้องการเพิ่มข้อมูลของพนักงานแต่ละแผนกในตารางนี้ก็สามารถทำได้ แต่วิธีการนี้จะเป็นการทำงานที่ซ้ำซ้อนเนื่องจากต้องเพิ่มข้อมูลลงในทั้งสองตาราง ในกรณีนี้เราควรใช้การสร้างความสัมพันธ์ระหว่างตาราง Employee และตาราง Department จึงจะเหมาะสมกว่าสมมติให้ Field ชื่อ Department Name ในตาราง Department เป็นคีย์หลัก การสร้างความสัมพันธ์ระหว่างตารางทำได้โดยเพิ่ม Field ชื่อ Department ลงในตาราง Employee จากนั้นใส่ข้อมูลของแผนกที่สังกัดให้แก่พนักงานแต่ละคน หลังจากใส่ข้อมูลแล้วให้ทำการแจ้งกับระบบจัดการฐานข้อมูลว่า Field Department ในตาราง Employee คือ คีย์นอก ที่อ้างอิงไปยังตาราง Department



# Database Key ที่ใช้ในการสร้างความสัมพันธ์

## คีย์นอก (Foreign key)

ฐานข้อมูลจะบังคับใช้ “Referential integrity” คือ เมื่อเปลี่ยนแปลงหรือลบข้อมูลในตาราง Department ระบบจะทำการเปลี่ยนแปลงหรือลบข้อมูลในตาราง Employee ไปด้วย ทำให้ค่าของข้อมูลใน Field Department ของตาราง Employee มีค่าตรงกับข้อมูลในตาราง Department





# Database Key ที่ใช้ในการสร้างความสัมพันธ์

## คีย์นอก (Foreign key)

ฐานข้อมูลจะบังคับใช้ “Referential integrity” คือ เมื่อเปลี่ยนแปลงหรือลบข้อมูลในตาราง Department ระบบจะทำการเปลี่ยนแปลงหรือลบข้อมูลในตาราง Employee ไปด้วย ทำให้ค่าของข้อมูลใน Field Department ของตาราง Employee มีค่าตรงกับข้อมูลในตาราง Department จะสังเกตได้ว่า คีย์นอกนั้นจะไม่มีการบังคับค่าเฉพาะตัว (Uniqueness) หมายความว่าในหนึ่งแผนกสามารถมีพนักงานได้หลายคน







# การสร้างความสัมพันธ์ระหว่างตาราง

วิธีการสร้างความสัมพันธ์ระหว่างตาราง มีดังนี้

1. เปิดฐานข้อมูลในใบงานที่ 4.1 ขึ้นมา
2. ไปที่ Database Tools tab แล้วเลือกที่ Relationships



3. เมื่อเลือกที่ Relationships แล้วจะปรากฏหน้าต่างขึ้น ให้เลือกตารางที่ต้องการสร้างความสัมพันธ์โดยสามารถเลือกหลาย ๆ ตารางพร้อมกันได้ด้วยการกดแป้น Ctrl แล้วคลิกที่ตารางที่ต้องการ เมื่อเลือกตารางเสร็จแล้วคลิกที่ Add เพื่อเพิ่มตาราง เมื่อเพิ่มตารางจบครบแล้วคลิกที่ Close



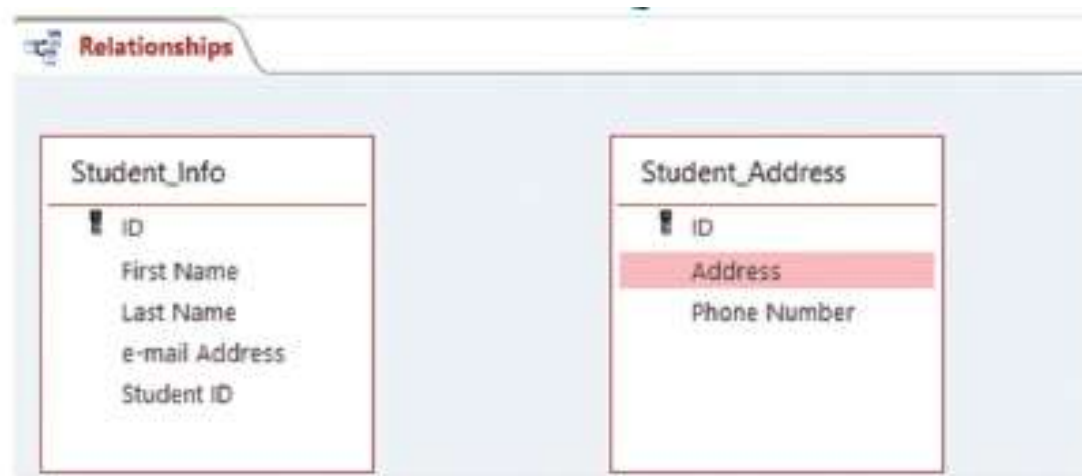




# การสร้างความสัมพันธ์ระหว่างตาราง

วิธีการสร้างความสัมพันธ์ระหว่างตาราง มีดังนี้

4. เมื่อคลิกที่ Close จะปรากฏหน้าต่างความสัมพันธ์ขึ้น จากตัวอย่างจะเป็นการสร้างความสัมพันธ์ระหว่างตาราง Student\_Info และตาราง Student\_Address จะสังเกตได้ว่ายังไม่มีเส้นเชื่อมระหว่างตารางทั้งสอง เนื่องจากยังไม่มีการสร้างความสัมพันธ์นั่นเอง





# การสร้างความสัมพันธ์ระหว่างตาราง

วิธีการสร้างความสัมพันธ์ระหว่างตาราง มีดังนี้

5. ทำการเลือกคีย์หลักและคีย์นอกที่จะใช้ในการสร้างความสัมพันธ์ ในตัวอย่างนี้จะใช้ ID ในตาราง Student\_Info เป็นคีย์หลัก และ Phone Number ในตาราง Student\_Address เป็นคีย์นอก

6. การสร้างความสัมพันธ์ทำได้โดยคลิกลากคีย์หลักไปยังคีย์นอก จะปรากฏกล่องข้อความ Edit Relationships ในขั้นตอนี้ต้องทำการเลือกว่าจะบังคับใช้ Referential Integrity หรือไม่ ในตัวอย่างนี้จะไม่บังคับใช้ Referential Integrity จากนั้นให้คลิกที่ Create

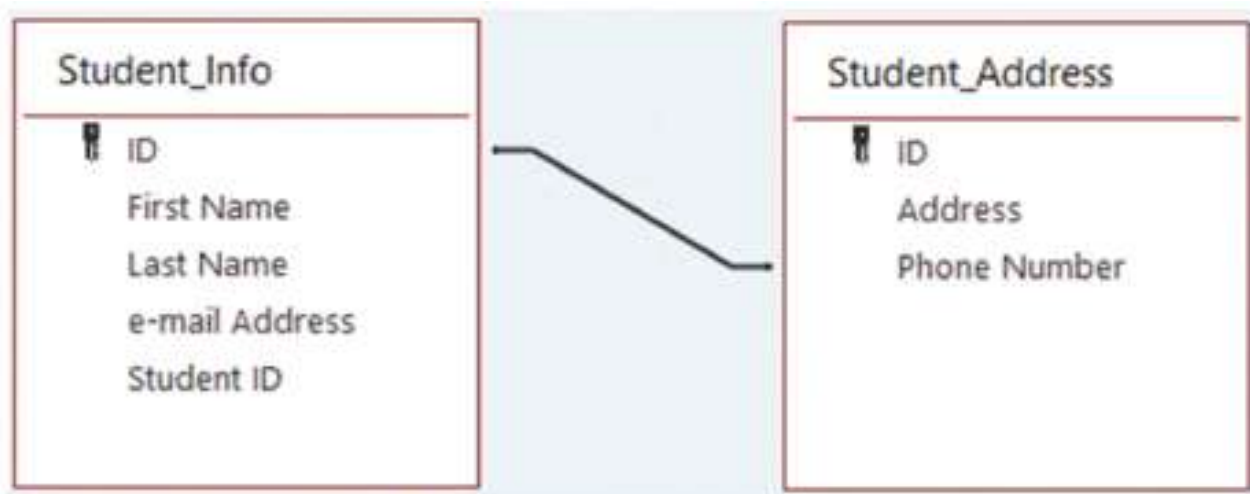
Dialog box titled "Edit Relationships" showing the relationship between "Student\_Info" (Table/Query) and "Student\_Address" (Related Table/Query). The primary key "ID" is linked to the foreign key "Phone Number". The "Enforce Referential Integrity" checkbox is unchecked. The "Relationship Type" is set to "One-To-Many". Buttons for "Create", "Cancel", "Join Type..", and "Create New.." are visible.



# การสร้างความสัมพันธ์ระหว่างตาราง

วิธีการสร้างความสัมพันธ์ระหว่างตาราง มีดังนี้

7. เมื่อคลิกที่ Create แล้วจะได้ความสัมพันธ์ระหว่างตารางดังรูป





# การสร้างความสัมพันธ์ระหว่างตาราง

## Referential Integrity

เพื่อเป็นการทำความเข้าใจ Referential Integrity จึงขอยกตัวอย่างเหตุการณ์ดังนี้ ฐานข้อมูลหนึ่งประกอบด้วยตาราง Customer ซึ่งใช้เก็บข้อมูลของลูกค้า และตาราง Order ใช้สำหรับเก็บรายการสั่งซื้อจากลูกค้า หากเกิดเหตุการณ์บางอย่างทำให้ Record ในตาราง Customer หายไป (ถูกลบหรือแก้ไข) จะส่งผลให้รายการสั่งซื้อจากลูกค้าที่หายไปนั้นกลายเป็น “Orphan record”

**Orphanrecord** หมายถึง Record ที่มีการอ้างอิงไปยัง Record ในตารางอื่นที่ไม่มีตัวตน เช่นเดียวกับตัวอย่างยอดสั่งซื้อที่ไม่มีผู้สั่งซื้อ (ลูกค้า) การบังคับใช้ Referential Integrity มีขึ้นเพื่อป้องกันการเกิด Orphan record นี้ขึ้น เมื่อมีการบังคับใช้ Referential integrity จะมีผลดังนี้



# การสร้างความสัมพันธ์ระหว่างตาราง

## Referential Integrity

1. ในการเพิ่ม Record ลงในตารางซึ่งมีคีย์นอกอยู่นั้น จะต้องมีการมี Record ที่เหมือนกันอยู่ในตารางคีย์หลักด้วย ตัวอย่างเช่น ในการสั่งซื้อสินค้าของลูกค้านั้นจะต้องมีข้อมูลของลูกค้าก่อนจึงจะสามารถทำการสั่งซื้อได้
2. ป้องกันการแก้ไขข้อมูลในตารางที่มีคีย์หลักอยู่หากมี Record ที่ตรงกันกับ Record ในตารางคีย์นอก ตัวอย่างเช่น ไม่สามารถแก้ไขข้อมูลการสั่งซื้อของลูกค้าในตาราง Customer ได้ หากมีข้อมูลการสั่งซื้อในตาราง Order แล้ว
3. ป้องกันการลบข้อมูลในตารางที่มีคีย์หลักอยู่หากมี Record ที่เกี่ยวข้องกันกับ Record ในตารางคีย์นอก ตัวอย่างเช่น ไม่สามารถลบข้อมูลของลูกค้าในตาราง Customer ได้ หากมีข้อมูลการสั่งซื้อในตาราง Order แล้ว



# การสร้างความสัมพันธ์ระหว่างตาราง

## ตัวเลือกของ Referential Integrity

1. Cascade Update Related Fields อนุญาตให้มีการเปลี่ยนคีย์หลักในตารางคีย์หลักได้และทำการเปลี่ยนแปลง Record ในตารางคีย์รองให้ตรงกัน ตัวอย่างเช่น กรณีที่มีการระงับการสั่งซื้อผ่านบัตรเครดิต หากลูกค้าทำบัตรเครดิตหาย ก็สามารถเปลี่ยนยอดระงับการสั่งซื้อไปยังบัตรเครดิตใบใหม่ของลูกค้าคนเดิมได้
2. Cascade Delete Related Records อนุญาตให้ลบ Record ที่เกี่ยวข้องในตารางคีย์รองเมื่อมีการลบ Record ในตารางคีย์หลัก โดยปกติแล้วตัวเลือกเหล่านี้จะไม่ได้เปิดใช้งาน หากต้องการใช้งานจะต้องทำการเลือกก่อน



# การสร้างความสัมพันธ์ระหว่างตาราง

## การแก้ไขความสัมพันธ์

การแก้ไขความสัมพันธ์ดังรูปทำได้โดยดับเบิลคลิกที่เส้นแสดงความสัมพันธ์ จะปรากฏกล่องข้อความ Edit Relationship จากนั้นจึงแก้ไขความสัมพันธ์

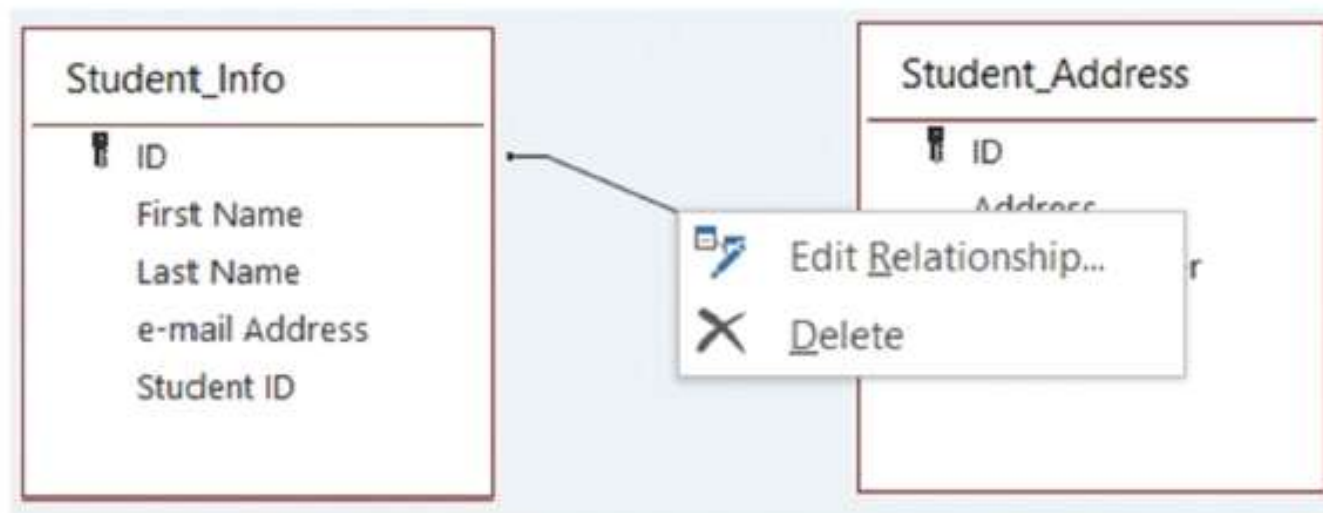




# การสร้างความสัมพันธ์ระหว่างตาราง

## การลบความสัมพันธ์

การลบความสัมพันธ์ทำได้โดยคลิกขวาที่เส้นความสัมพันธ์ แล้วเลือกคำสั่ง Delete จากเมนู







# การสร้างความสัมพันธ์ระหว่างตาราง

## การลบความสัมพันธ์

การลบความสัมพันธ์ทำได้โดยคลิกขวาที่เส้นความสัมพันธ์ แล้วเลือกคำสั่ง Delete จากเมนู

