**ขนาดความจุ**

**บิต (Bit) Binary Digit เป็นหน่วยที่เล็กที่สุดของข้อมูลที่ใช้อยู่ในคอมพิวเตอร์ มี 0 กับ 1**

**ไบต์ (Byte) ตัวเลขจำนวน 8 บิต จะรวมกันเข้าเป็น 1 ไบต์**

**กิโลไบต์ (Kilobyte) ใช้ย่อว่า KB โดย 1 KB มีค่าเท่ากับ 1,024 ไบต์**

**เมกะไบต์ (Megabyte) ใช้ย่อว่า MB โดย 1 MB มีค่าเท่ากับ 1,048,576 หรือ (1,024 x 1,024 ) มักใช้ในการวัดหน่วยความจำหลัก (RAM)**

**กิกะไบต์ (Gigabyte)  ใช้ย่อว่า GB โดย 1 GB มีค่าเท่ากับ 1,073,741,824 หรือ (1,024  x 1,024 x 1,024)**

**เทราไบต์ (Terabyte) ใช้ย่อว่า TB โดย 1 เทราไบต์จะเท่ากับ 1,099,511,627,776 หรือ (1024 x 1024 x 1024 x 1024) บิต (Bit) Bin**

**หน่วยวัดความเร็ว**

**เฮิรตซ์ (**[**อังกฤษ**](https://th.wikipedia.org/wiki/%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%E0%B8%AD%E0%B8%B1%E0%B8%87%E0%B8%81%E0%B8%A4%E0%B8%A9)**: hertz ย่อว่า Hz) เป็นหน่วย**[**SI**](https://th.wikipedia.org/wiki/SI)**ของค่า**[**ความถี่**](https://th.wikipedia.org/wiki/%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%96%E0%B8%B5%E0%B9%88)**โดย 1 Hz คือความถี่ที่เท่ากับ 1 ครั้ง ต่อวินาที**

**1 Hz. = 1ครั้ง/วินาที**

**1 KHz กิโลเฮิรตซ์ = 1000 ครั้ง/วินาที**

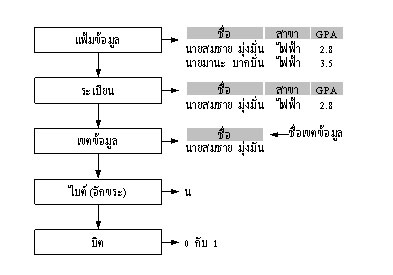
**1 MHz เมกะเฮิรตซ์ =1 ล้านครั้ง/วินาที**

**1 GHz กิกะเฮิรตซ์ =1 พันล้านครั้ง/วินาที**

**1 THz เทราเฮิรตซ์ =1 ล้านล้านครั้ง/วินาที**

**หน่วยในการจัดเก็บข้อมูล**  
ข้อมูลที่จัดเก็บในคอมพิวเตอร์ โดยแท้จริงแล้วมีลักษณะการจัดเก็บเป็นสัญญาณดิจิตอล คือมีค่า 0 กับ 1 เท่านั้น แต่เพื่อให้มองภาพข้อมูลได้ง่าย จึงแบ่งหน่วยในการจัดเก็บข้อมูลออกเป็นส่วนย่อย ต่างๆ ดังนี้

1. **บิต (Bit)** คือหน่วยของข้อมูลที่เล็กที่สุด คือการจัดเก็บข้อมูลที่มีอยู่เพียง 2 สถานะ คือ 0 กับ 1 ซึ่งเป็น ลักษณะการทำงานของคอมพิวเตอร์ที่ทำงานด้วยระดับสัญญาณดิจิตอล
2. **ไบต์ (Byte)** คือหน่วยของข้อมูลซึ่งเกิดจากการนำข้อมูล 8 บิตมารวมกัน โดยใช้ แทน ตัวอักขระ 1 ตัว
3. **เขตข้อมูล (Field)** หรือฟิลด์ คือหน่วยของข้อมูลซึ่งเกิดจากการนำข้อมูลหลายๆ ไบต์ หรือหลายๆ อักขระมารวม กัน เพื่อใช้แทนความหมายของสิ่งใดสิ่งหนึ่ง เช่น รหัสนักศึกษา, ชื่อ, ที่อยู่ เป็นต้น
4. **ระเบียน (Record)** หรือ เรคคอร์ด คือหน่วยของข้อมูลซึ่งเกิดจากการนำข้อมูล หลายๆ เขตข้อมูลมารวมกัน ซึ่งเขตข้อมูล ที่นำมารวมกันนี้ จะมีความสัมพันธ์เกี่ยวเนื่องกัน เช่น รหัสนักศึกษา, ชื่อ, ที่อยู่ รวมกันเป็นระเบียนข้อมูล ของนักศึกษา เป็นต้น
5. **แฟ้มข้อมูล (File)** หรือไฟล์ คือ หน่วยของข้อมูลซึ่งเกิดจากการนำข้อมูลหลายๆ ระเบียน ที่มีลักษณะ ของเขต ข้อมูลเหมือนกันมาจัดเก็บรวมกัน เช่นการจัดเก็บข้อมูลระเบียนของนักศึกษา หลายๆ คน รวมกันเป็น แฟ้มข้อมูลนักศึกษาเป็นต้น ซึ่งข้อมูลแต่ละระเบียนที่นำมารวมกันจะต้องมี เขตข้อมูล อย่างน้อย 1 เขต ข้อมูลที่แยกความแตกต่างของข้อมูลในแต่ละระเบียนได้หลายๆ คน รวมกันเป็น แฟ้มข้อมูลนักศึกษา เป็นต้น ซึ่งข้อมูลแต่ละระเบียนที่นำมารวมกันจะต้องมี เขตข้อมูล อย่างน้อย 1 เขต ข้อมูลที่แยกความแตกต่างของข้อมูลในแต่ละระเบียนได้ ตัวอย่างของหน่วยข้อมูลต่างๆ แสดงดังรูปที่ 1.4



**ประโยชน์ของระบบฐานข้อมูล**

ระบบฐานข้อมูล จะช่วยแก้ปัญหาของระบบแฟ้มข้อมูล และมีประโยชน์ หลายๆ ด้านดังนี้

**1.4.1 ลดความซ้ำซ้อนของข้อมูล** เนื่องจากการนำข้อมูลที่จำเป็นต่อการใช้งานมาจัดเก็บไว้ รวมกันเป็นฐานข้อมูลส่วนกลาง ทำให้แต่ละหน่วยงานที่จำเป็นต้องใช้ข้อมูล ไม่ต้องจัดเก็บข้อมูลไว้ที่หน่วย งานของตนเองอีก นอกจากลดความสิ้นเปลืองในการจัดเก็บแล้ว ยังช่วยแก้ปัญหาต่างๆ ที่จะเกิดตามมา เนื่องจากความซ้ำซ้อนของข้อมูลได้  
**1.4.2 แก้ปัญหาความขัดแย้งกันของข้อมูล** ซึ่งเป็นปัญหาที่เกิดเนื่องมาจากความซ้ำซ้อนของข้อมูล เมื่อมีข้อมูลที่ซ้ำๆ กันอยู่หลายที่ หากมีการปรับปรุงแก้ไขข้อมูลในที่หนึ่งแล้ว แต่ไม่ได้แก้ไขข้อมูลในที่ อื่นๆ ตามด้วย ก็จะทำให้ข้อมูลในแต่ละที่เกิดความขัดแย้งกันขึ้น ดังนั้นการจัดเก็บข้อมูลไว้เพียงที่เดียวจึง ช่วยแก้ปัญหาความขัดแย้งกันของข้อมูลได้  
**1.4.3 การบริหารจัดการฐานข้อมูลทำได้ง่าย** เนื่องจากมีการจัดเก็บข้อมูลไว้ที่ส่วนกลาง ทำให้การ จัดการข้อมูลทำได้ง่ายขึ้น โดยผู้ที่ทำหน้าที่ในการบริหารจัดการฐานข้อมูลเรียกว่า ผู้บริหารฐานข้อมูล (Database Administrator: DBA)  
**1.4.4 กำหนดมาตรฐานของข้อมูลได้**เนื่องจากการจัดเก็บข้อมูลไว้ที่ส่วนกลางที่เดียวดังนั้น DBA จะเป็นผู้กำหนดโครงสร้างในการจัดเก็บข้อมูล ทำให้โครงสร้างของข้อมูลต่างๆ  
**1.4.5 สามารถใช้งานฐานข้อมูลร่วมกันได้**เนื่องจากโครงสร้างการจัดเก็บข้อมูลในฮาร์ดดิสก์จะ ถูก กำหนดด้วย DBMS และผู้ใช้แต่ละคนจะต้องใช้งานผ่าน DBMS เท่านั้น ดังนั้นจึงสามารถใช้งาน ฐาน ข้อมูลร่วมกันได้โดยไม่ต้องกังวลถึงความแตกต่างของภาษาคอมพิวเตอร์ที่ใช้ นอกจากนี้ข้อมูลอื่นๆ ที่ผู้ใช้ไม่ได้เป็นผู้สร้างขึ้นมา ก็สามารถใช้งานได้ถ้าหากได้รับสิทธิในการใช้งานข้อมูลดังกล่าว  
**1.4.6 เกิดความเป็นอิสระระหว่างข้อมูลกับโปรแกรม** จากปัญหาของระบบแฟ้มข้อมูล ซึ่งการ แก้ไขโครงสร้างของแฟ้มข้อมูล เช่นการเพิ่มฟิลด์ ซึ่งโปรแกรมที่มีอยู่เดิมไม่จำเป็นต้องนำไปใช้งาน แต่ต้องทำการแก้ไขโปรแกรมเนื่องจากการเขียนโปรแกรมจะยึดติดกับโครงสร้างของแฟ้มข้อมูล หากใช้งานเป็นระบบฐานข้อมูล จะสามารถแก้ไขปัญหานี้ได้ เนื่องจากการใช้งานต่างๆ จะต้องใช้งานไว้เพียงที่เดียวจึง ช่วยแก้ปัญหาความขัดแย้งกันของข้อมูลได้  
**1.4.7 กำหนดระบบรักษาความปลอดภัยให้กับข้อมูลได้**เนื่องจากข้อมูลแต่ละข้อมูลจะมีความ สำคัญไม่เท่ากัน ดังนั้นจึงต้องมีการกำหนดสิทธิในการใช้งาน ข้อมูลแต่ละส่วน ซึ่งเป็นหน้าที่ของ ผู้บริหารฐานข้อมูล เป็นผู้กำหนดว่าใครมีสิทธิใช้งานข้อมูลส่วนไหนได้บ้าง

**ส่วนในระบบฐานข้อมูล มีคำศัพท์ต่าง ๆ ที่เกี่ยวข้องดังนี้**

**เอนทิตี้ (Entity)** หมายถึง ชื่อของสิ่งใดสิ่งหนึ่ง ได้แก่ คน สถานที่ สิ่งของ การกระทำ ซึ่งต้องการจัดก็บข้อมูล

ไว้ เช่น เอนทิตี้ลูกค้า เอนทิตี้พนักงาน

- เอนทิตี้ชนิดอ่อนแอ (Weak Entity) เป็นเอนทิตี้ที่ไม่มีความหมาย หากขาดเอนทิตี้อื่นในฐานข้อมูล

แอททริบิวต์**(Attribute)** หมายถึง รายละเอียดข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี้หนึ่ง ๆ เช่น

เอนทิตี้นักศึกษา ประกอบด้วย - แอทริบิวต์รหัสนักศึกษา

- แอททริบิวต์ชื่อนักศึกษา

- แอททริบิวต์ที่อยู่นักศึกษา

**ความสัมพันธ์ระหว่างเอนทิตี้ แบ่งออกเป็น 3 ประเภท คือ**

**1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one Relationships)**เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตี้หนึ่ง ในลักษณะหนึ่งต่อหนึ่ง (1 : 1)

**2**. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม**(One-to-many Relationships)**เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่ง ที่มีความสัมพันธ์กับข้อมูลหลาย ๆ ข้อมูลในอีกเอนทิตี้หนึ่ง ในลักษณะ (1:m) ตัวอย่างเช่น

**3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-many Relationships)**เป็นการแสดงความสัมพันธ์ของข้อมูลสองเอนทิตี้ในลักษณะกลุ่มต่อกลุ่ม (m:n)

  เอนทิตี้ใบสั่งซื้อแต่ละใบจะสามารถสั่งสินค้าได้มากกว่าหนึ่งชนิด ความสัมพันธ์ของข้อมูลจากเอนทิตี้ใบสั่งซื้อไปยังเอนทิตี้สินค้า จึงเป็นแบบหนึ่งต่อกลุ่ม (1:m) ในขณะที่สินค้าแต่ละชนิด จะถูกสั่งอยู่ในใบสั่งซื้อหลายใบ ความสัมพันธ์ของข้อมูลจากเอนทิตี้สินค้าไปยังอินทิตี้ใบสั่งซื้อ จึงเป็นแบบหนึ่งต่อกลุ่ม (1:n) ดังนั้นความสัมพันธ์ของเอนทิตี้ทั้งสอง จึงเป็นแบบกลุ่มต่อกลุ่ม (m:n)

จากคำศัพท์ต่าง ๆ ที่เกี่ยวข้องกับระบบฐานข้อมูลที่ได้กล่าวมาแล้วข้างต้น จึงอาจให้นิยามของฐานข้อมูลในอีกลักษณะได้ว่า **“ฐานข้อมูล”**อาจหมายถึง โครงสร้างสารสนเทศ ที่ประกอบด้วยหลาย ๆ เอนทิตี้ที่มีความสัมพันธ์กัน

**การออกแบบข้อมูลด้วย E-R Diagram (Entity-Relationship Diagrams)**

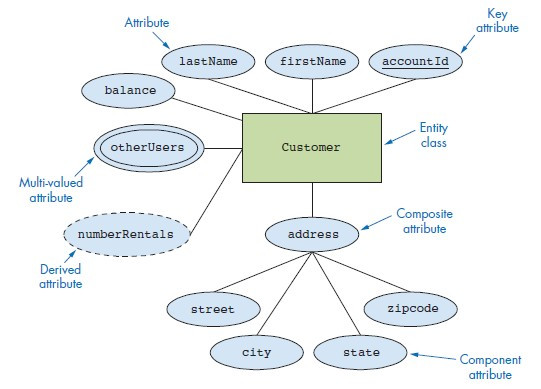
สมมติว่าเรามีโอกาสได้ไปเยือนสถานที่ธรรมชาติแห่งหนึ่งซึ่งยิ่งใหญ่สวยงามมาก ๆ แล้วบังเอิญลืมเอากล้องถ่ายรูปติดไปด้วย  พอเรากลับมาเราก็อยากจะแบ่งปันความประทับใจกับเพื่อนของเราด้วยการเล่าให้เพื่อนฟังโดยบรรยายไปต่าง ๆ นา ๆ แต่พูดเท่าไหร่เพื่อนก็ไม่รู้สึกซาบซึ้งอะไรกับเราด้วยสักที เพราะไม่ว่าจะอธิบายด้วยคำพูดยืดยาวขนาดไหนเพื่อนเราก็ไม่สามารถมองเห็นภาพความสวยงามตรงกับภาพที่เราเห็นมาด้วยตาตนเองได้  เพราะว่าภาพที่เราเห็นนั้นมันประกอบขึ้นด้วยรายละเอียดปลีกย่อยมากมายเกินกว่าที่คำพูดจะอธิบายได้

ปัญหาดังกล่าวสามารถเกิดขึ้นในการออกแบบระบบสารสนเทศเช่นกัน  ถ้านักพัฒนาระบบต้องการจะถ่ายทอดภาพของระบบสารสนเทศที่กำลังจะพัฒนาว่าประกอบด้วยข้อมูลอะไรบ้างซึ่งตรงกับความต้องการที่ยูสเซอร์ที่ได้แจ้งไว้กับนักพัฒนาระบบหรือไม่   การอธิบายด้วยคำพูดของนักพัฒนาระบบไม่สามารถทำให้ยูสเซอร์เข้าใจตรงกับสิ่งที่นักพัฒนาระบบต้องการถ่ายทอดได้   นั่นจึงเป็นที่มาของ E-R Diagrams ซึ่งใช้แสดงข้อมูลในระบบสารสนเทศในรูปของรูปภาพ  ซึ่งทำให้ยูสเซอร์และนักพัฒนาระบบมีความเข้าใจตรงกันในข้อมูลที่จะประกอบขึ้นเป็นระบบที่กำลังพัฒนา

ในการเขียน E-R Diagram จะต้องทำความเข้าใจเกี่ยวกับส่วนประกอบต่าง ๆ ของ E-R ดังต่อไปนี้

**– Entity**  
entity แทนที่ สิ่ง ซึ่งอาจจะเป็นทั้งคน วัตถุ สิ่งของ หรือสิ่งซึ่งเป็นนามธรรมจับต้องไม่ได้ ใช้แทนที่สิ่งในโลกความเป็นจริงแต่ละ entity แทนที่ด้วยชื่อของ entity ในรูปสี่เหลี่ยมผืนผ้า

**– Attribute**  
attributes ใช้แสดงถึงคุณสมบัติของ entity เช่น ชื่อ นามสกุล  เลขประจำตัว  ที่อยู่ ฯลฯ แทนที่ด้วยชื่อของ attribute ในรูปวงรี



จากภาพข้างบนแสดงถึง entity ที่ชื่อ Customer ซึ่งจะแสดงชื่อ entity อยู่ในรูปสี่เหลี่ยมผืนผ้า  
ส่วน attribute จะแสดงชื่อ attribute อยู่ในรูปวงรี ซึ่งเชื่อมโยงกับ entity ด้วยเส้นตรงที่ลากเชื่อมระหว่าง entity และ attribute

**Key attribute**  
คือ attribute ที่ถูกกำหนดให้เป็น key ของ entity โดยแทนที่ด้วย attribute ที่ถูกขีดเส้นใต้ จากในภาพ account Id ถูกขีดเส้นใต้เพื่อแสดงว่า attribute นี้ถูกใช้เป็น key ของ entity Customer

**Multi-valued attribute**  
คือ attribute ที่มีค่าบรรจุอยู่มากกว่าหนึ่งค่า โดยแทนที่ด้วยวงกลมรูปไข่ซ้อนกันสองวง จากในภาพ attribute ที่ชื่อ otherUsers เป็น multi-valued attribute หมายถึง Customer สามารถมีผู้ใช้คนอื่น ๆ ที่ใช้บัญชีของ Customer ได้ (ผู้ใช้คนอื่นอาจจะเป็นญาติกับ Customer เช่น ลูก, ภรรยา, น้อง)

**Derived attribute**  
คือ attribute ที่ค่าของมันได้มาจากการคำนวณของ attribute อื่น โดยแทนที่ด้วยวงกลมรูปไข่ที่เป็นเส้นประ  จากในภาพ attribute ที่ชื่อ numberRentals หรือจำนวนที่เช่าซึ่งได้มาจากการรวมจำนวนสินค้าที่เช่าทั้งหมดเข้าด้วยกัน

**Composite attribute**  
คือ attribute ที่สามารถแยกออกเป็น attribute ย่อย ๆ ได้หลาย attribute แทนที่โดยชื่อ attribute ใน วงกลมรูปไข่ที่มีเส้นตรงลากไปเชื่อมโยงกับ attribute หลัก จากในภาพ attribute ที่ชื่อ address สามารถแยกออกเป็น attribute ย่อยที่ชื่อ street, city, state, zipcode  ได้อีก

**Relationship Types**  
ใช้แสดงความสัมพันธ์ระหว่าง entity โดยแทนที่ด้วยรูปสี่เหลี่ยมข้าวหลามตัด ดังในภาพข้างล่าง Store  Owns (เป็นเจ้าของ)  Video (ในกรณีที่อ่านจากซ้ายไปขวา) หรือ Video IsOwnedBy (ถูกเป็นเจ้าของโดย) Store (ในกรณีที่อ่านจากขวาไปซ้าย)  พึงสังเกตุว่าชื่อของ relationship types จะต้องเป็นคำกริยาเสมอ  และความสัมพันธ์สามารถมี attribute ของตัวเองได้ เช่นในภาพ ความสัมพันธ์ Owns มี attribute คือ purchase Date และ cost

ตัวอย่าง

ตาราง Member

|  |  |  |
| --- | --- | --- |
| Id | รหัสสมาชิก | 1001 |
| name | ชื่อสมาชิก | นาย ก นามสกุล ข |
| Username | ยูสเซอร์เนม | User1 |
| Password | พาสเวิรด์ | 1234 |
| Al | สถานะ | 1 |

Member

ตัวอย่างความสัมพันธ์ของตาราง นักศึกษา อาจารย์ที่ปรึกษา และวิชา อธิบายเป็นภาพ E-R

M

M

1

M

นักศึกษา

อาจารย์ที่ปรึกษา

วิชาเรียน

มี

ลงทะเบียน