Introduction to Database

204202 – IT II

Based on material by Jeff Ullman

Database

- เป็นที่รวบรวมข้อมูล (data) ไว้อย่างเป็นระบบ
- ระบบจะถูกสร้างขึ้นบนพื้นฐานของแบบจำลอง (model) ของข้อมูล
  - แบบจำลองนี้เน้นจำลองความเกี่ยวข้องสัมพันธ์กันของข้อมูล
- ในคลาสนี้เราจะเรียนแบบจำลองของข้อมูลที่เรียกว่าEntity-Relationship Model

Entity-Relationship Model

- แบบจำลองข้อมูลที่แสดงความสัมพันธ์ (Relationship) ระหว่าง Entity (สิ่งของ คน สถานที่ และอื่นๆ)
- E/R model ช่วยในการออกแบบฐานข้อมูล
- แบบจำลองนี้สามารถวาดออกมาเป็นไดอะแกรม เรียกว่า

  entity-relationship diagrams.
- ซึ่งเราสามารถจะแปลงไดอะแกรมที่ออกแบบไปเป็นฐานข้อมูลได้

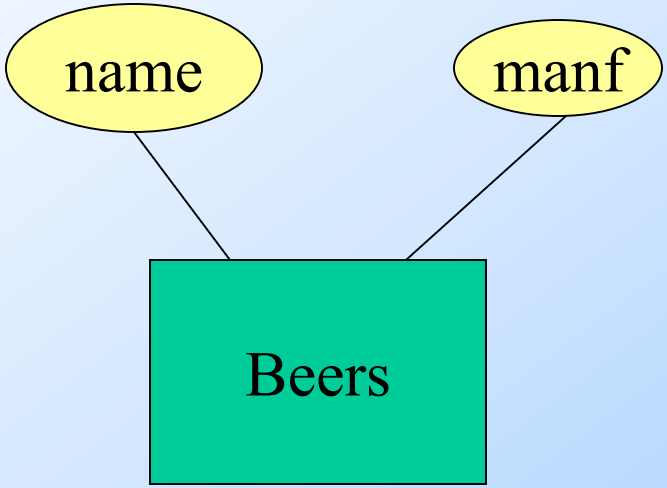
Entity Sets

- *Entity* = “thing” or object.
- *Entity set* = กลุ่มหรือเซทของ entity ที่เหมือนกัน
- *Attribute* = คุณลักษณะเฉพาะของ entity นั้นๆ

E/R Diagrams

- ใน entity-relationship ไดอะแกรม:
  - Entity set จะแสดงด้วยรูปสี่เหลี่ยม.
  - Attribute แสดงด้วยวงรีซึ่งมีเส้นต่อไปยัง Entity set

Example

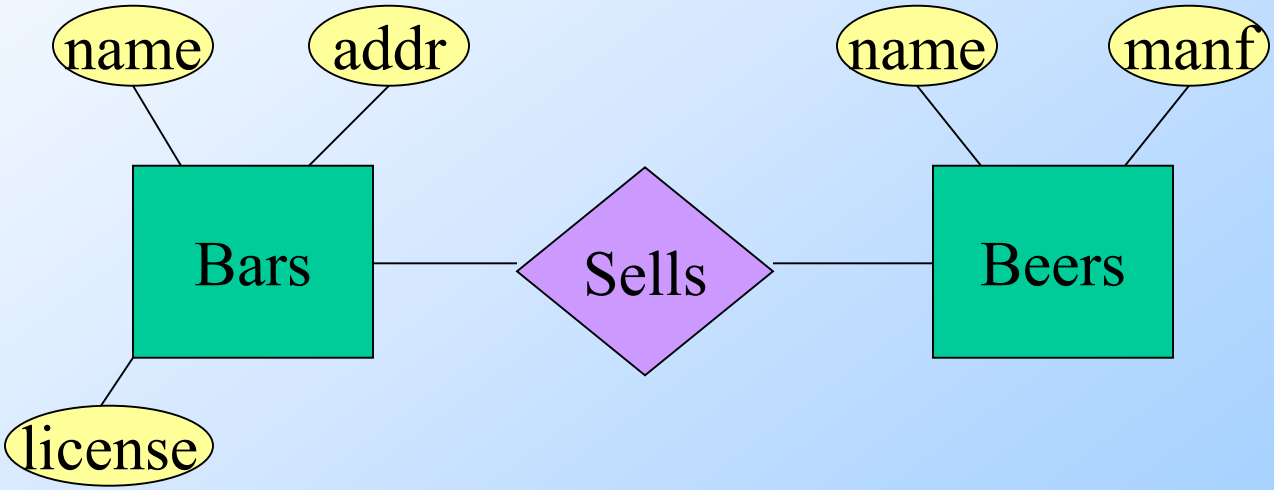


- Entity set Beers มีคุณลักษณะสองตัวคือ,ชื่อ name และ manf (ผู้ผลิต).
- เบียร์แต่ละยี่ห้อก็ต้องมีคุณลักษณะสองตัวนี้ติดอยู่ด้วย

Name	**Manf**
Chang	ThaiBev
Archa	ThaiBev
Signha	Bunrawd

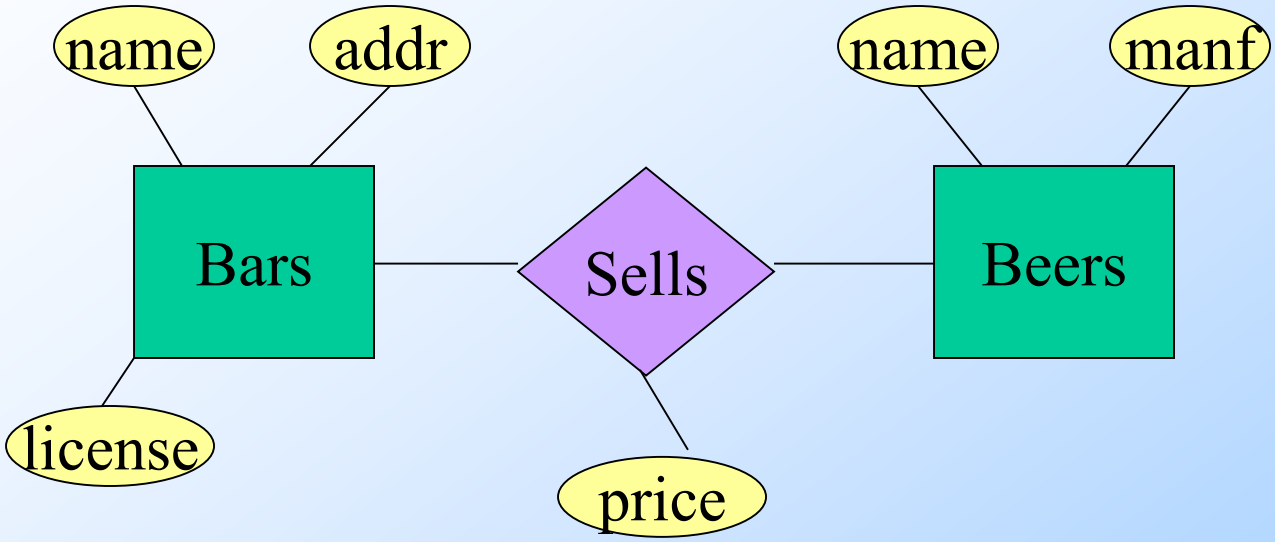
ความสัมพันธ์ (Relationships)

- ความสัมพันธ์ใช้เชื่อมต่อ Entity set สองเซทขึ้นไป
- ปกติใช้สัญลักษณ์รูปเพชรในการแสดงความสัมพันธ์



Bars sell some beers.

Attribute ของ Relationship



Bars sell some beers for xxx Baht

Relationship Set

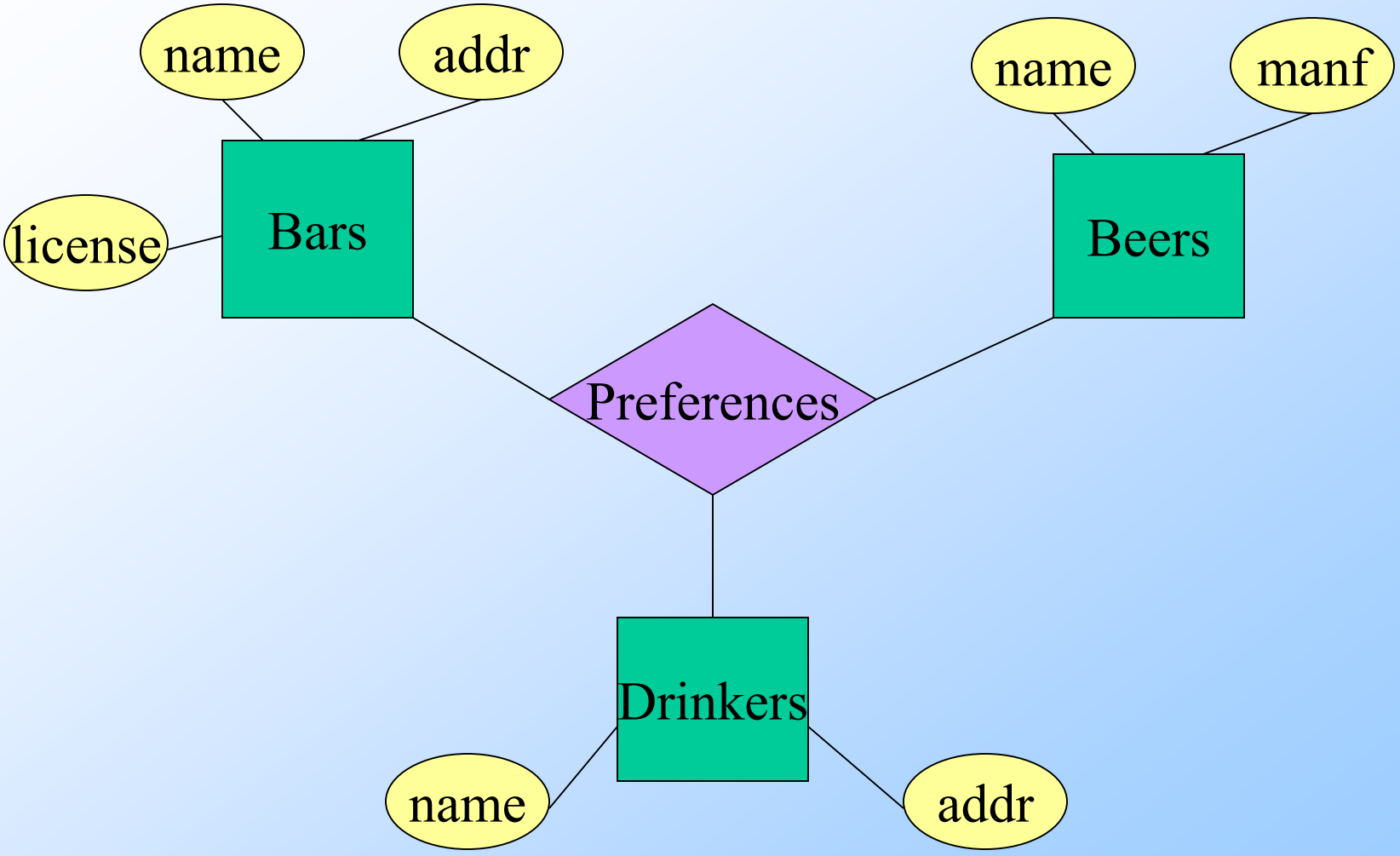
- คือเซทของ Entities ที่มีอยู่ในฐานข้อมูลที่เกี่ยวข้องกันด้วยความสัมพันธ์นี้ ณ เวลาปัจจุบัน
- ตัวอย่างRelationship set ของความสัมพันธ์ข้างต้น

Bar	Beer
Joe's Bar	Bud
Joe's Bar	Miller
Sue's Bar	Bud
Sue's Bar	Pete's Ale
Sue's Bar	Bud Lite

Relationships แบบหลายทาง

- บางครั้ง เราต้องการความสัมพันธ์ที่ต้องเชื่อมต่อEntity มากกว่าสองตัว
- เช่น การแสดงความสัมพันธ์ที่คนบางคนชอบดื่มเบียร์บางยี่ห้อที่ร้านบางร้านเท่านั้น
  - เราต้องปรับปรุงความสัมพันธ์ข้างต้นให้เป็นแบบ หลาย ทาง

ตัวอย่าง


name
addr
Bars
license
name
manf
Beers
Preferences
Drinkers
name
addr

Relationship Set ของไดอะแกรมข้างต้น

Bar	Drinker	Beer
Joe's Bar	Ann	Miller
Sue's Bar	Ann	Bud
Sue's Bar	Ann	Pete's Ale
Joe's Bar	Bob	Bud
Joe's Bar	Bob	Miller
Joe's Bar	Cal	Miller
Sue's Bar	Cal	Bud Lite

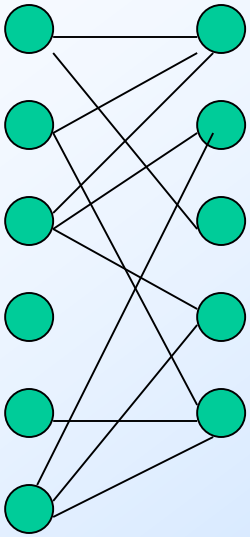
แต่!

- ข้างต้นเราเห็นความสัมพันธ์แบบหนึ่งต่อหนึ่งไปแล้ว นั่นคือเราสมมุติว่า
  - ร้านแต่ละร้านขายเบียร์ยี่ห้อเดียว
  - และเบียร์แต่ละยี่ห้อจะมีขายในร้านใดร้านหนึ่งเท่านั้น
- ทว่าความสัมพันธ์ข้างต้นไม่สมจริง
  - เพราะ ร้านอาจขายเบียร์หลายยี่ห้อ และ เบียร์แต่ละยี่ห้อก็มีขายอยู่ในหลายๆร้าน

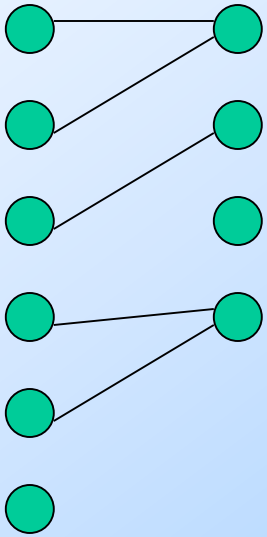
Relationships แบบต่างๆ

- Many-many relationships
  - Entity จากทั้งสองเซทเชื่อมกันได้มากกว่าหนึ่ง
  - ตัวอย่าง ร้านแต่ละร้านขายเบียร์มากกว่าหนึ่งยี่ห้อ
- Many-one relationships
  - Entity จากฝั่งซ้ายเชื่อมไปหา Entity ฝั่งขวาได้ตัวเดียว
  - ตัวอย่าง ความสัมพันธ์"สังกัด" ของนักศึกษา และ คณะ
- One-one relationships
  - Entity จากฝั่งซ้ายและขวาเชื่อมกันได้เพียงตัวเดียว
  - ตัวอย่าง ความสัมพันธ์"มีป้ายทะเบียน" ของ รถกับเลขทะเบียน

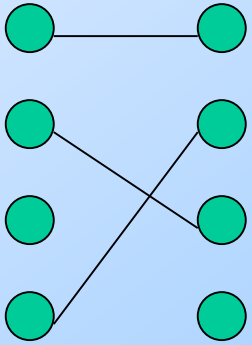
In Pictures:



many-many



many-one



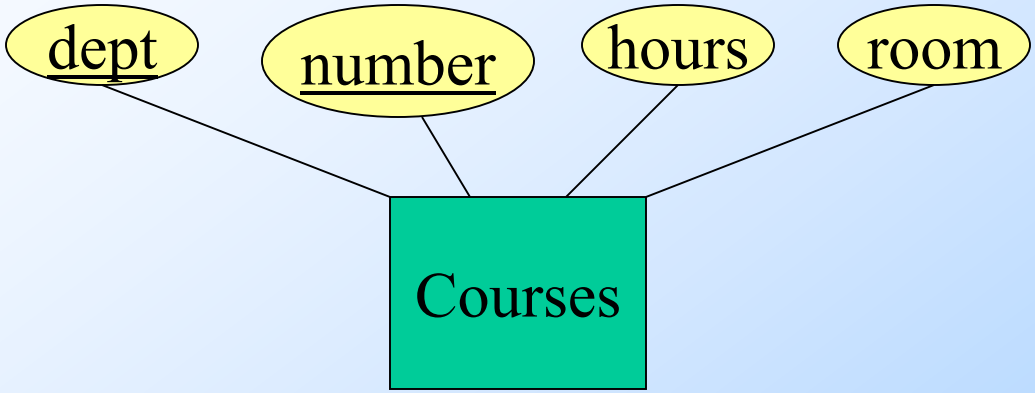
one-one

Keys

- *Key* คือคุณลักษณะ (ตัวเดียวหรือกลุ่มของคุณลักษณะ) ที่ Entity สองตัวใดๆ ใน entity set มี **ไม่ซ้ำกัน**
- เราต้องเป็นคนกำหนดคีย์เหล่านี้
- ตัวอย่าง
  - เลขประจำตัวเสียภาษีของร้าน
  - เลขบัตรประชาชน
  - เลขนักศึกษา
- ใน ER-diagram เราจะขีดเส้นใต้คุณสมบัติที่ทำหน้าที่เป็นคีย์

Example: name is Key for Beers

Example: a Multi-attribute Key

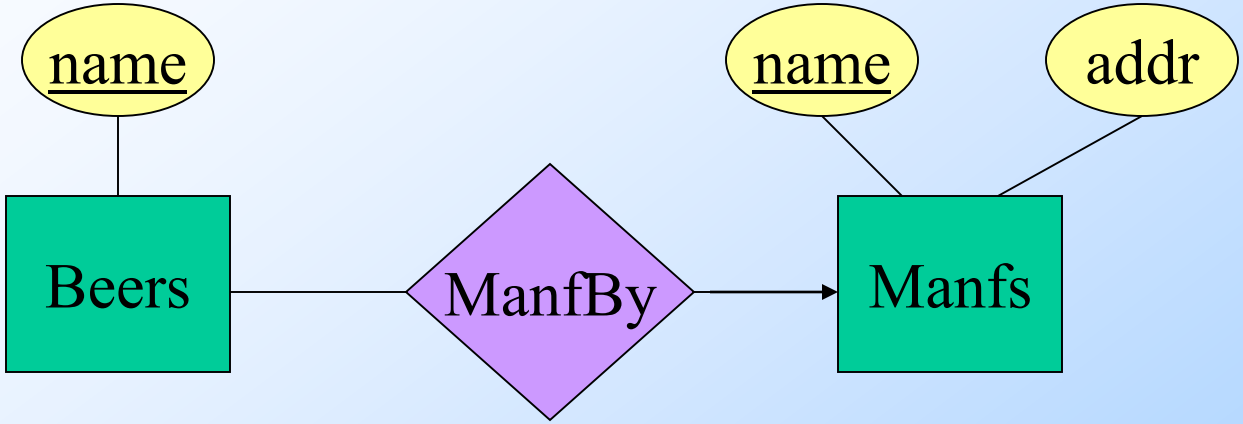


- สังเกตว่าเราสามารถใช้ hours และ room เป็นคีย์ได้ แต่เราเลือกสองตัวแรก

Design Techniques

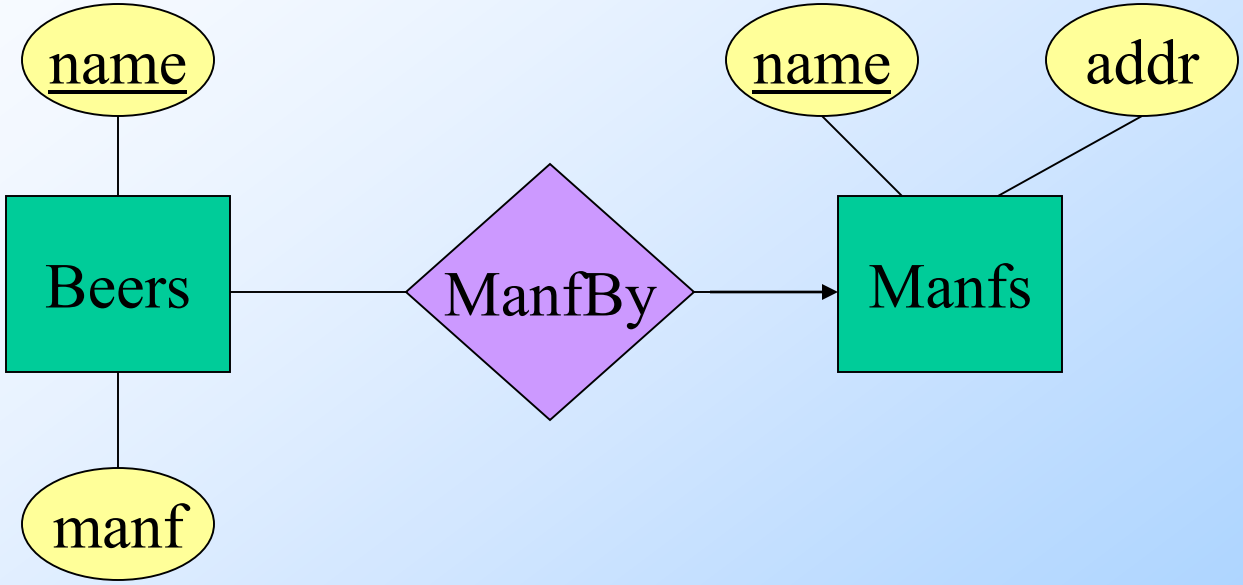
1. เลี่ยงความซ้ำซ้อน
2. จำกัดการใช้ entity set หากสามารถแสดงข้อมูลนั้นได้ด้วยคุณลักษณะ (attribute)

Example: Good



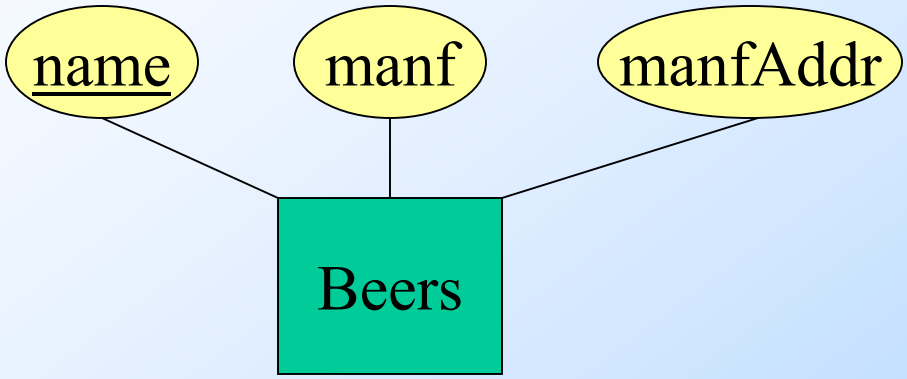
เป็นการออกแบบที่ดี

Example: Bad



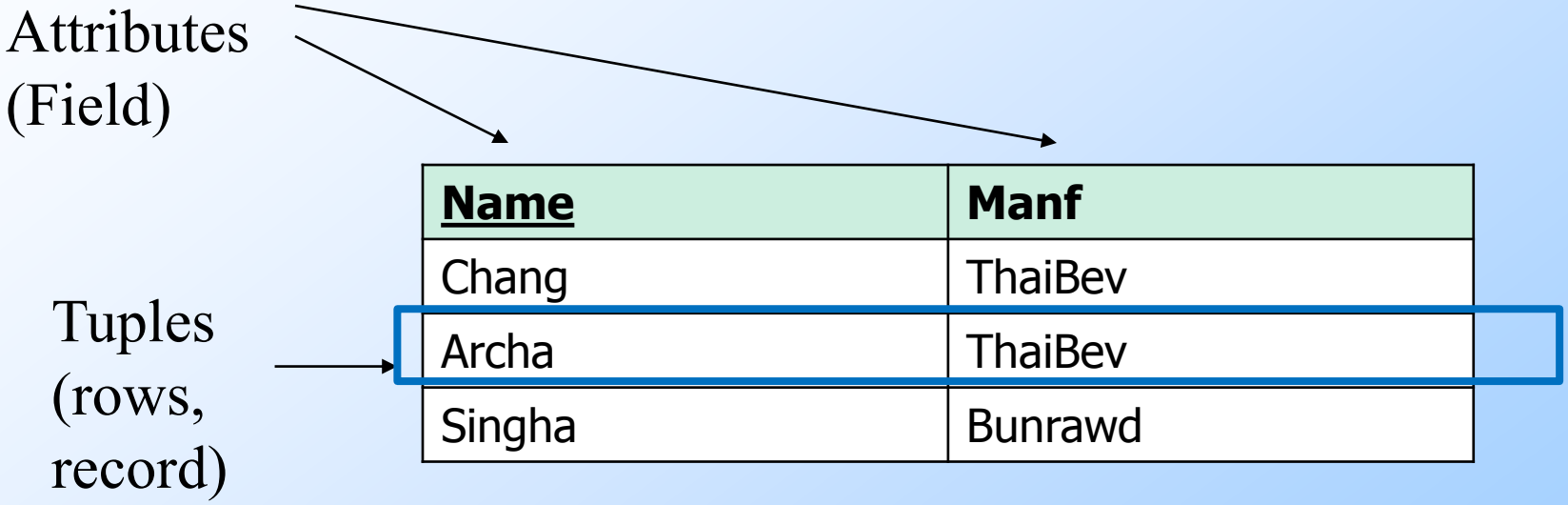
ชื่อผู้ผลิตซ้ำซ้อน (มีการกล่าวถึงสองครั้ง)

Example: Bad



ปัญหา: ชื่อผู้ผลิตจะหายไปหาก ผู้ผลิตนั้นไม่มีผลิตภัณฑ์ (ในขณะนี้)

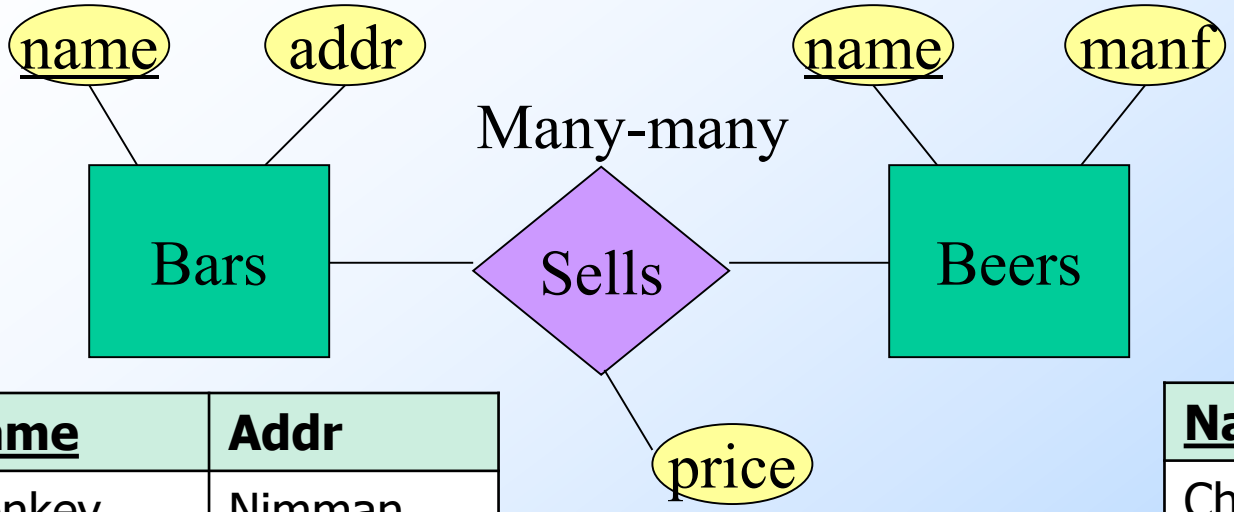
A database (Terminology)



From ER diagram to Database

- Entity แสดงได้ด้วยตารางซึ่งประกอบไปด้วย attributes ของ entity นั้นๆ
- Relationship คือตารางที่แสดง
  - keys ของ entity ที่มาเชื่อมกันด้วยความสัมพันธ์นั้น
  - Attributes ของความสัมพันธ์.
- Database ก็คือ กลุ่มของตารางหลายๆตารางที่เก็บข้อมูลที่เราต้องการ

Example



Bars sell some beers.

Name	**Addr**
Monkey	Nimman
Riverside	Ping River

Bars table

Name	**Manf**
Chang	ThaiBev
Archa	ThaiBev
Singha	Bunrawd

Beers table

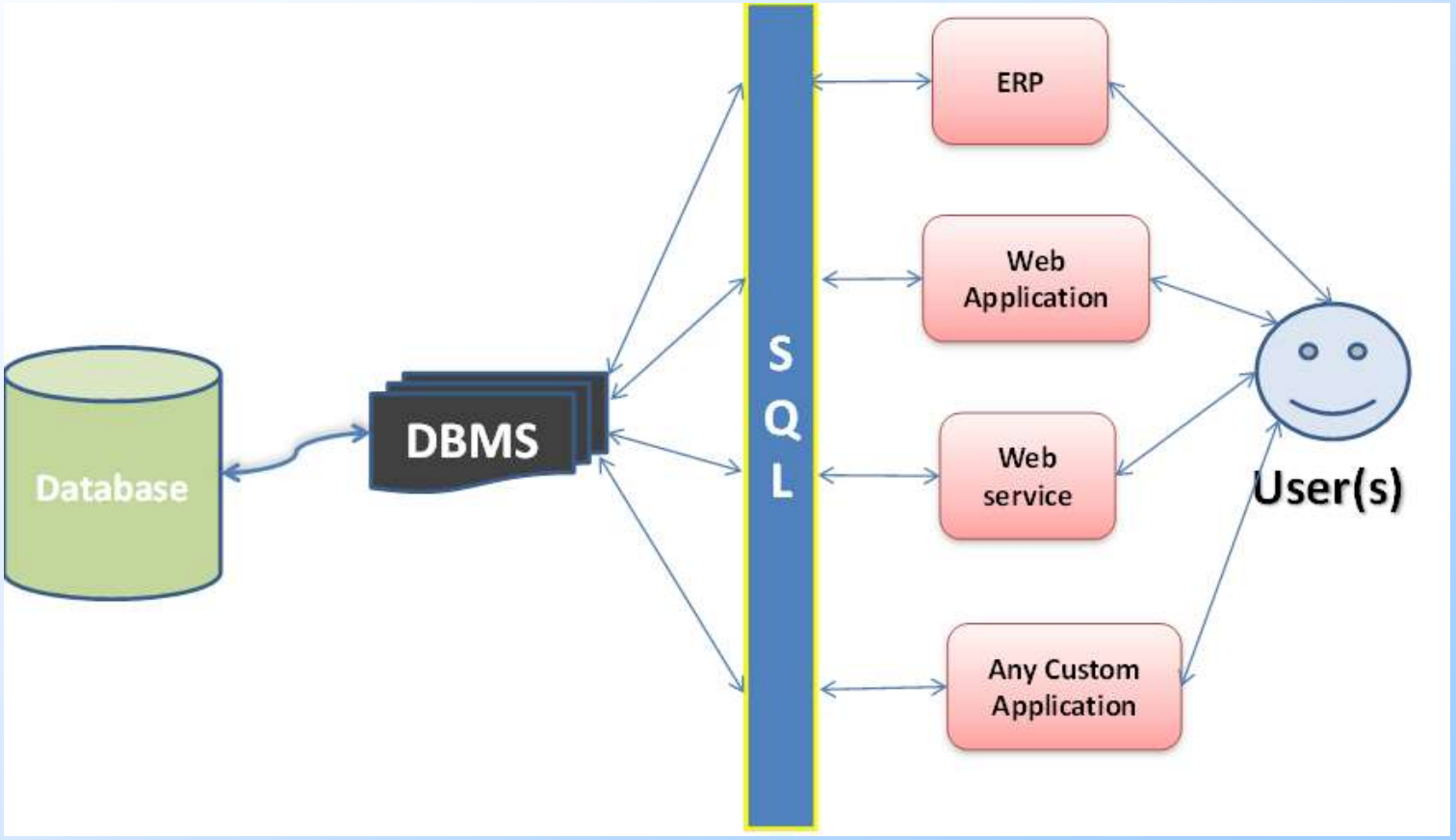
Sells table

Name_bar	**Name_beer**	**Price**
Monkey	Chang	70
Monkey	Singha	90
Monkey	Archa	60
Riverside	Singha	100

Working with the database

- การติดต่อกับ database สามารถทำผ่าน database management system (DBMS)
- DBMS จะทำหน้าที่
  - รับคำสั่งจาก User พร้อมแปลความหมายของคำสั่ง
  - กระทำการกับฐานข้อมูลตามคำสั่ง
  - ส่งข้อมูลกลับมายัง User
- ผู้ใช้จะติดต่อกับ DBMS ผ่านภาษา SQL

Some illustration



SQL Language

- SQL คือภาษาระดับสูงมาก (ใกล้เคียงภาษามนุษย์ที่สุด)
  - เป็นภาษาที่ บอกให้ทำอะไร มากกว่าที่ บอกให้ทำอย่างไร
  - มีความซับซ้อนน้อยกว่าการเขียนภาษาแบบอื่นๆ
- DMBS จะจัดการหาวิธีการดึงข้อมูลที่เราอยากได้ ให้เร็วและมีประสิทธิภาพมากที่สุด
  - เราไม่ต้องบอก dbms ว่าต้องดึงข้อมูลอย่างไร
  - เราแค่บอกว่า เราต้องการอะไร

คำสั่งหลักใน SQL

- SELECT
  - ใช้ในการดึงข้อมูลที่ต้องการจากฐานข้อมูล
- INSERT
  - ใช้ในการใส่ข้อมูลใหม่ เข้าไปในฐานข้อมูล
- DELETE
  - ใช้ในการลบข้อมูลออกจากฐานข้อมูล
- UPDATE
  - ใช้ในการปรับปรุง เปลี่ยนแปลงข้อมูลในฐานข้อมูล

Select-From-Where Statements

SELECT attributes ที่ต้องการ (มีได้มากกว่าหนึ่ง)

FROM ตารางที่ต้องการ (มีได้มากกว่าหนึ่ง)

WHERE เงื่อนไขของ tuple ในตารางเหล่านั้นเป็นจริง

Example

- จากตาราง **beers** ถามว่า เบียร์ยี่ห้อไหนผลิตโดย **ThaiBev**

```
SELECT name
FROM Beers
WHERE manf = 'ThaiBev';
```

สังเกตว่า SQL ใช้ single-quotes สำหรับข้อมูลแบบ strings.
และเป็นภาษาที่ *case-insensitive* ยกเว้นข้อความใน string

Result of Query

name
Chang
Archa

ผลลัพธ์คือตารางแสดงชื่อเบียร์ที่ผลิตโดย ThaiBev

* In SELECT clauses

- * ใช้เป็นสัญลักษณ์แทนการบอกว่า **เอา attribute ทุกตัว (Wildcard)**
- ตัวอย่างจากตารางข้อมูลของเบียร์

```
SELECT *
FROM Beers
WHERE manf = 'ThaiBev';
```

Result of Query:

name	**manf**
Chang	ThaiBev
Archa	ThaiBev
. . .	. . .

กรณีที่ใช้ * ผลลัพธ์จะแสดงตารางที่มี attribute ครบทุกตัว

เงื่อนไขที่ซับซ้อนยิ่งขึ้นใน WHERE Clause

- การหาราคาขายเบียร์ช้างของร้าน Monkey

```
SELECT price
FROM Sells
WHERE Name_bar = 'Monkey' AND
    Name_beer = 'Chang';
```