



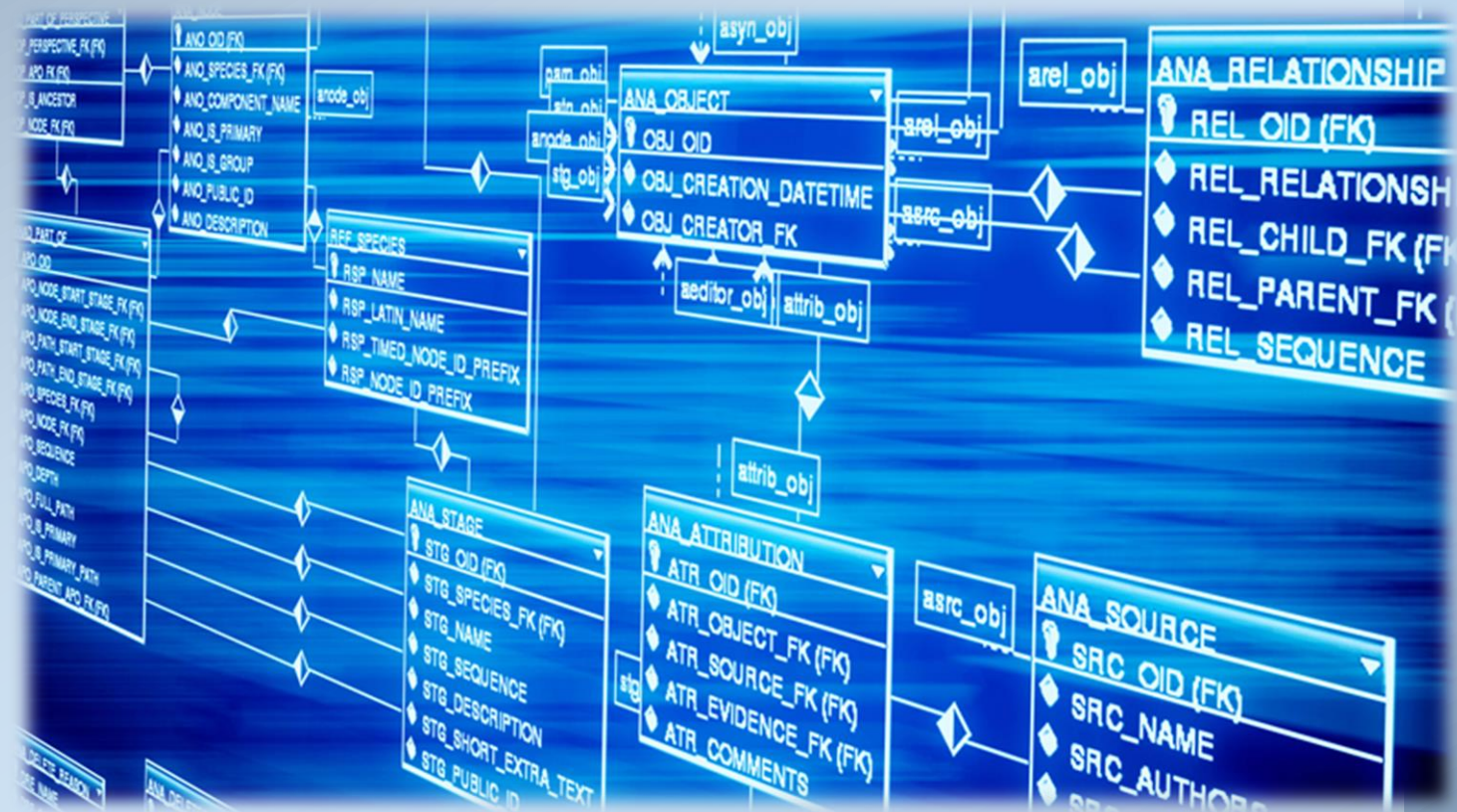
# บทที่ 4 โครงสร้างฐานข้อมูลแบบ Relational (Relational Database Model)

โดย อ.อภิพงศ์ ปิงยศ

รายวิชา สธ312 ระบบการจัดการฐานข้อมูลทางธุรกิจ

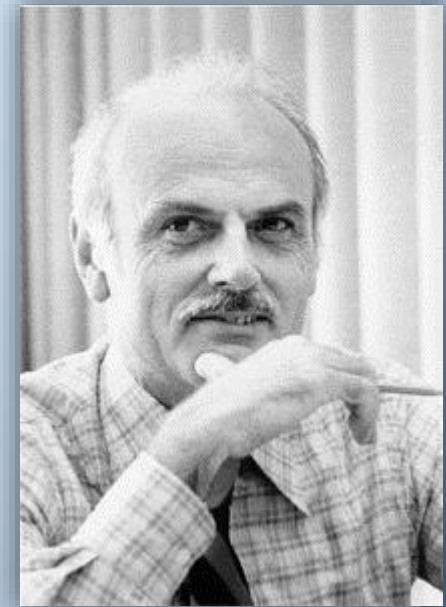
# Overview

- Relation
- Domain
- คุณสมบัติของ Relation
- ประเภทของ Relation
- Degree
- Cardinality
- Catalog ของ Relational Model



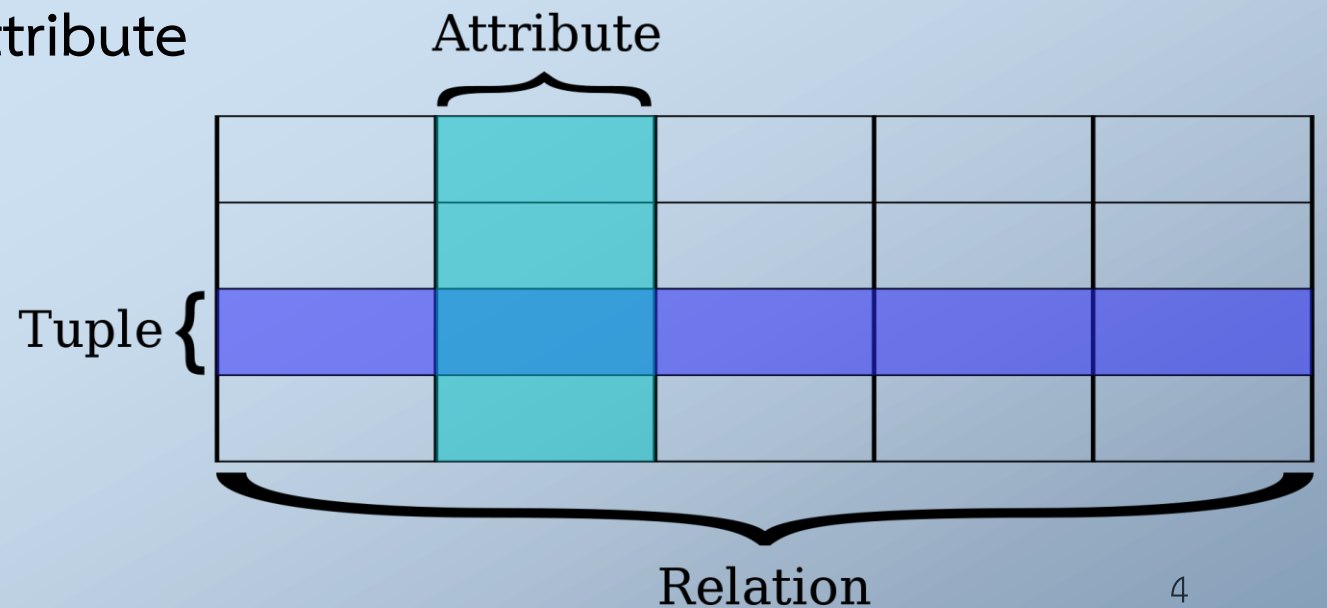
# บทนำ

- แบบจำลองข้อมูลเชิงสัมพันธ์เป็นแบบจำลองข้อมูลที่มีการใช้งานแพร่หลายมากที่สุดในปัจจุบัน จึงมีความจำเป็นอย่างยิ่งที่จะต้องศึกษาให้เข้าใจอย่างถ่องแท้
- แบบจำลองข้อมูลเชิงสัมพันธ์ถูกคิดค้นโดยชาวอังกฤษชื่อ Dr. Edgar F. Codd ในปี 1968 ซึ่งขณะนั้นเขาเป็นนักวิจัยของบริษัท IBM
- ในบทนี้จะนำเสนอส่วนประกอบหลักที่สำคัญของแบบจำลองข้อมูลเชิงสัมพันธ์



# Relation

- Relation มักเรียกว่า “Table” เป็นหน่วยที่จัดเก็บข้อมูลในรูปแบบตาราง 2 มิติ ที่ประกอบไปด้วยแถว (Row) และคอลัมน์ (Column)
- Row ใน Relation จะเรียกว่า **Tuple**
- Column ใน Relation จะเรียกว่า **Attribute**





# ตัวอย่างที่ 1 ตัวอย่าง Relation ของ Employee

EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

ใน Relation “EMPLOYEE”

ประกอบไปด้วย 4 **Tuple** ซึ่งเป็นข้อมูลของพนักงานที่มีเลขประจำตัวตั้งแต่ 00001 – 00004

แต่ละ Tuple จะประกอบด้วย 6 **Attribute** ได้แก่ “EmpID”, “Name”, “Surname”, “Sex”, “Salary” และ “DeptID”

# Domain

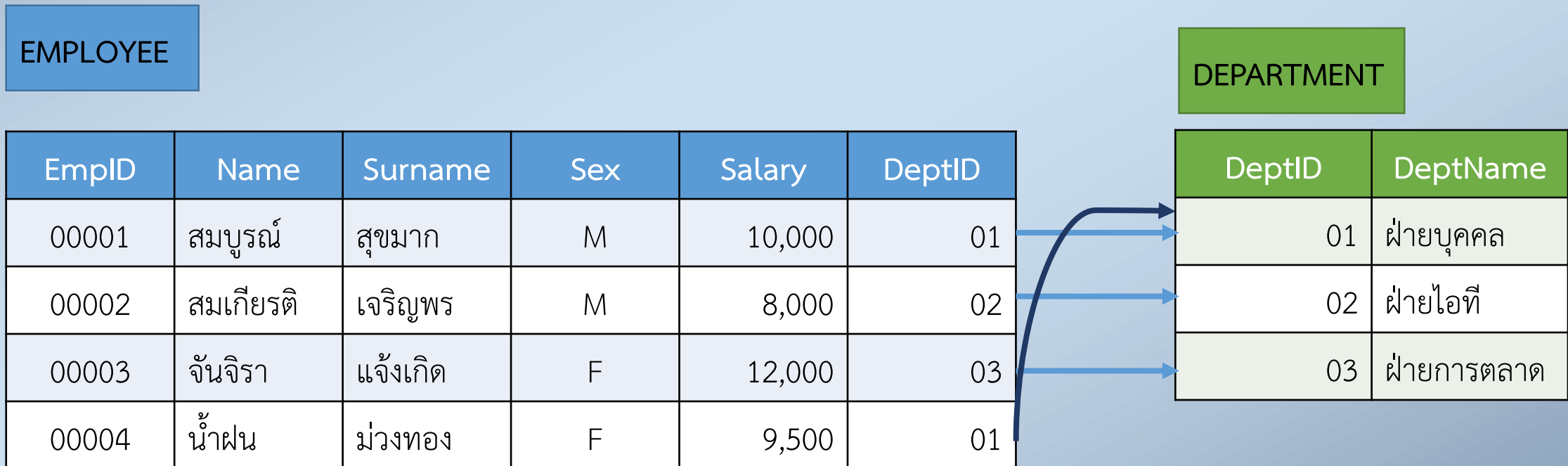
- เป็นการนิยามขอบเขตของค่าที่เป็นไปได้ให้กับข้อมูลแต่ละ Attribute
- เป็นการป้องกันไม่ให้เกิดการป้อนข้อมูลที่เกินขอบเขตที่กำหนด
- เช่น การกำหนดให้เงินเดือนของพนักงานต้องมีค่ามากกว่า 0 เสมอ หรือกำหนดให้เพศของพนักงานจะต้องมีแค่ชาย (M) หรือหญิง (F) เท่านั้น

## คุณสมบัติของ Domain

- 1) จะต้องเป็นค่าที่มีความหมายในหน่วยที่เล็กที่สุด (Atomicity) เช่น “จังหวัด” สามารถนำมาเป็นโดเมนได้ แต่ “ที่อยู่” ไม่สามารถเป็นโดเมนได้เพราะสามารถแยกย่อยได้อีกเป็นบ้านเลขที่ ถนน ตำบล อำเภอ จังหวัด
- 2) ข้อมูลที่สามารถนำมากำหนด Domain ได้ จะต้องเป็นอิสระจากข้อมูลอื่น เช่น รหัสพนักงานของแต่ละคน ชื่อจังหวัด รหัสแผนก เป็นต้น
- 3) ข้อมูลที่นำมากำหนดโดเมนจะต้องเป็นข้อมูลประเภทเดียวกัน เช่น เพศ ควรจะมีแค่ F หรือ M ไม่ควรกำหนดเป็น 0 หรือ 1 ด้วย

## คุณสมบัติของ Domain [cont.]

- 4) โดเมนที่กำหนดให้กับ Attribute อื่นที่อ้างอิง สามารถถ่ายทอดโดเมนของตนเองให้กับ Attribute ในอีก Relation หนึ่งที่อ้างอิงด้วย ดังตัวอย่างต่อไปนี้





## คุณสมบัติของ Domain [cont.]

- 5) ค่าของโดเมนที่กำหนดให้กับข้อมูล ไม่จำเป็นต้องปรากฏในข้อมูลนั้นๆ เช่น กรณีที่กำหนดให้โดเมนของเงินเดือนพนักงานมากกว่า 0 ไม่ได้หมายความว่าทุกจำนวนที่มาค่ามากกว่า 0 จะต้องเป็นข้อมูลใน attribute เงินเดือนพนักงาน

# คุณสมบัติของ Relation

- 1) Relation อยู่ในรูปของเซตทางคณิตศาสตร์ ซึ่งภายในเซตจะต้องประกอบด้วยสมาชิกที่มีค่าไม่ซ้ำกัน ตัวอย่างเช่น

POPULAR

ID	Name	Surname	Sex
00001	สมบูรณ์	สุขมาก	M
00002	สมเกียรติ	เจริญพร	M
00003	สมบูรณ์	สุขมาก	M
00004	น้ำฝน	ม่วงทอง	F

## คุณสมบัติของ Relation [cont.]

- 2) ลำดับที่ของสมาชิกใดๆ จะไม่มีผลต่อ Relation ดังนั้นภายใน Relation จึงไม่มีการกำหนดลำดับที่ให้กับแต่ละ Tuple
- 3) ลำดับที่ของ Attribute จะไม่มีผลต่อ Relation เพราะการอ้างถึง Attribute จะใช้ชื่อของ Attribute ในการอ้างถึงโดยตรง
- 4) ค่าในทุก Attribute จะต้องมีความหมายเพียงความหมายเดียว และจะต้องไม่เป็นข้อมูลที่ซ้ำกัน เช่น ตัวอย่างในหน้าถัดไป

## EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01, 03
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันทิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

จะต้องแยกเป็น Tuple ใหม่

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00001	สมบูรณ์	สุขมาก	M	10,000	03
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันทิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

## คุณสมบัติของ Relation [cont.]

- 5) ชื่อของ Attribute ใน Relation เดียวกัน จะต้องไม่ซ้ำกัน
- 6) ค่าที่ปรากฏในแต่ละ Attribute ใน Relation เดียวกัน จะต้องใช้แทนข้อมูลที่มีความหมายเดียวกัน เช่น “EmpID” จะต้องเก็บข้อมูลรหัสพนักงานเท่านั้น ไม่สามารถเก็บข้อมูลอื่น ๆ ได้ เช่น เพศ หรือเงินเดือน เป็นต้น

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	00001
M	สมเกียรติ	เจริญพร	00002	8,000	02
00003	จันจิรา	แจ้งเกิด	12,000	F	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

# ประเภทของ Relation

- 1) **Named Relation** เป็น Relation ที่สร้างขึ้นด้วยคำสั่ง SQL อาจเป็น Relation จริง หรือ Relation ที่สร้างขึ้นมาด้วยการ Query ก็ได้
- 2) **Base Relation** เป็น Named Relation ที่เป็น Relation จริงที่อยู่ในฐานข้อมูล
- 3) **Derived Relation** เป็น Named Relation ที่ได้มาจากการใช้คำสั่ง Query เรียกข้อมูล มาจาก Base Relation เช่น อาจใช้การ Query “เฉพาะพนักงานชาย” ก็จะได้ Relation ด้านล่าง

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00002	สมเกียรติ	เจริญพร	M	8,000	02



## ประเภทของ Relation [cont.]

- 4) **View** เป็น Derived Relation ประเภทหนึ่ง แต่จะเป็น Relation เสมือน (Virtual Relation)
- 5) **Snapshot** คล้ายกับ View แต่จะแตกต่างกันที่ Snapshot ที่สามารถอ่านข้อมูลได้อย่างเดียว และสามารถกำหนดเวลาในการ Snapshot ได้ เช่น รายวัน รายสัปดาห์ เป็นต้น
- 6) **Query Result** เป็น Relation ชั่วคราว ที่เกิดจากคำสั่ง Query Language โดย Relation จะเกิดขึ้นเมื่อเรียกใช้เท่านั้น และจะหายไปเมื่อเลิกใช้งาน
- 7) **Intermediate Result** เป็น Relation ชั่วคราวที่เกิดขึ้นขณะประมวลผลคำสั่ง Query Language

# Degree

- ดีกรี คือจำนวนของ Attribute ของ Relation
- เช่น Relation “EMPLOYEE” มีดีกรีเท่ากับ 6
- Relation ที่มีดีกรีเท่ากับ 1 เรียกว่า “Unary” ถ้าดีกรี 2 เรียกว่า “Binary”

EMPLOYEE					
EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01, 03
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

# Cardinality

- คาร์ดินัลลิตี้ คือจำนวนของ Tuple ใน Relation หนึ่งที่มีความสัมพันธ์กับ Tuple อีก Relation หนึ่ง
- ใช้ในการกำหนดประเภทของความสัมพันธ์ระหว่างข้อมูลใน Relation ทั้งสอง
- จากตัวอย่างนี้ ความสัมพันธ์ระหว่างสอง Relation นี้ มี Cardinality คือ (1, M)

EMPLOYEE					
EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

DEPARTMENT	
DeptID	DeptName
01	ฝ่ายบุคคล
02	ฝ่ายไอที
03	ฝ่ายการตลาด

# Catalog ของ Relational Model

- โครงสร้างข้อมูลของ Catalog จะมีลักษณะเหมือนกับ Relation แต่ข้อมูลที่ถูกจัดเก็บใน Catalog จะไม่ใช่ข้อมูลทั่วไป แต่จะเป็น **Metadata**
- สมมุติให้ Catalog ประกอบไปด้วย 2 Relation มีชื่อว่า “TABLES” และ “COLUMNS” เพื่อใช้เก็บ Metadata ของ Relation “DEPARTMENT” และ “EMPLOYEE”

TABLES			
TabName	ColCount	RowCount	...
DEPARTMENT	2	3	...
EMPLOYEE	6	4	...
...	...	...	...

TabName	ColName	...
DEPARTMENT	DeptID	...
DEPARTMENT	DeptName	...
EMPLOYEE	EmpID	...
EMPLOYEE	Name	...
EMPLOYEE	Surname	...
...	...	...

COLUMN

## Catalog ของ Relational Model [cont.]

- จากตัวอย่างข้างต้น เมื่อผู้ใช้ต้องการทราบถึงโครงสร้างของ Relation “DEPARTMENT”
- โปรแกรม DBMS จะเข้ามาค้นหารายละเอียดของ Relation “DEPARTMENT” ใน Relation “TABLES” เพื่อแสดงจำนวนของ Tuple กับ Attribute
- ส่วนชื่อของ Attribute ต่าง ๆ ของ Relation “DEPARTMENT” โปรแกรม DBMS จะเข้าไปค้นหาใน Relation “COLUMNS”
- ผลที่ได้ต่อมาคือ Attribute “DeptID” และ “DeptName”

# สรุป

- โครงสร้างของหน่วยในการจัดเก็บข้อมูลของฐานข้อมูลเชิงสัมพันธ์ จะเก็บในรูปแบบตาราง 2 มิติ ที่เรียกว่า Relation ที่ประกอบไปด้วย Tuple และ Attribute
- แต่ละ Attribute สามารถกำหนดขอบเขตของค่าที่เป็นไปได้ เรียกว่า Domain
- จำนวนของ Attribute ใน Relation จะเรียกว่า Degree
- รูปแบบความสัมพันธ์ของ Tuple ระหว่าง Relation จะเรียกว่า Cardinality



## EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01
00005	คฤเดช	ป๋อกันทั้ง	M	15,000	02
00006	เพทชาย	ละดาดก	M	50,000	03

## DEPARTMENT

DeptID	DeptName
01	ฝ่ายบุคคล
02	ฝ่ายไอที
03	ฝ่ายการตลาด
04	ฝ่ายการเงิน

## SITE

EmpID	Site
00001	BKK
00004	CNX
00005	PHK
00006	BKK

**TABLES**

TabName	ColCount	RowCount
DEPARTMENT	2	4
EMPLOYEE	6	6
SITE	2	4

**COLUMN**

TabName	ColName
DEPARTMENT	DeptID
DEPARTMENT	DeptName
EMPLOYEE	EmpID
EMPLOYEE	Name
EMPLOYEE	Surname
EMPLOYEE	Sex
EMPLOYEE	Salary
EMPLOYEE	DeptID
SITE	EmpID
SITE	Site