บทที่ 7
การสร้างแบบจำลองความสัมพันธ์ของเอนทิตี้
(Entity Relation Modeling)

7.1 แนวคิดพื้นฐานในการสร้างแบบจำลอง (Basic Modeling Concepts)

ขั้นตอนแรกในการทำ Database Design ก็คือการทำ Data Modeling เพราะ Data Modeling เป็นเสมือนสะพานเชื่อมข้อมูลจริง (Real-World Object) กับ รูปแบบข้อมูลที่จะจัดเก็บใน ฐานข้อมูล

สาเหตุที่ต้องทำ Data Modeling ก็เพราะในการทำ Database Design นั้นปัญหาสำคัญที่ เกิดขึ้นก็คือ มุมมองข้อมูลของ User Designer และ Programmer จะแตกต่างกัน ทำให้เกิดผลต่อการออกแบบ ดังนั้นเราจึงต้องทราบรูปแบบที่ถูกต้องจริงของข้อมูล ซึ่งเราทำได้โดยการวิเคราะห์ข้อมูลและสร้าง Data Modeling ขึ้นมาอธิบายถึงรูปแบบข้อมูลที่ถูกต้องที่ใช้ในองค์กร

Data Model ที่นิยมใช้ในการออกแบบ Database Entity Relationship Model ซึ่งจะใช้อธิบายรูปแบบต่างๆ ของข้อมูลที่จะมีใน ฐานข้อมูล โดยจะเป็นการกำหนดรูปแบบในระดับที่เรียกว่า Conceptual Level (ระดับของแนวคิด)

"Model" หมายถึง รายละเอียดที่ใช้ในการอธิบายให้เห็นภาพพจน์ของสิ่งที่ไม่สามารถจะสังเกตุเห็นได้โดยตรง

"Data Model" หมายถึง การแสดงอย่างง่ายและชัดเจนของโครงสร้างข้อมูลจริงที่ ซับซ้อน (โดยการใช้ Graphic ในการอธิบาย)

Data Model จะแสดงถึง โครงสร้างข้อมูล (Data Structure) ลักษณะข้อมูล (Characteristics) ความสัมพันธ์ (Relation) ,ข้อจำกัด (Constraint) และ การเปลี่ยนแปลงของข้อมูล(Transformation)

Database Designer จะใช้ Data Model เป็นเครื่องมือในการติดต่อระหว่าง Designers, Programmer และ End Users ถ้า Data Model ถูกสร้างขึ้นมาอย่างถูกต้องจะทำให้การพัฒนาระบบฐานข้อมูล ง่ายขึ้น และการใช้ข้อมูลขององค์กรจะมีประสิทธิภาพด้วย

CT 316 (S)

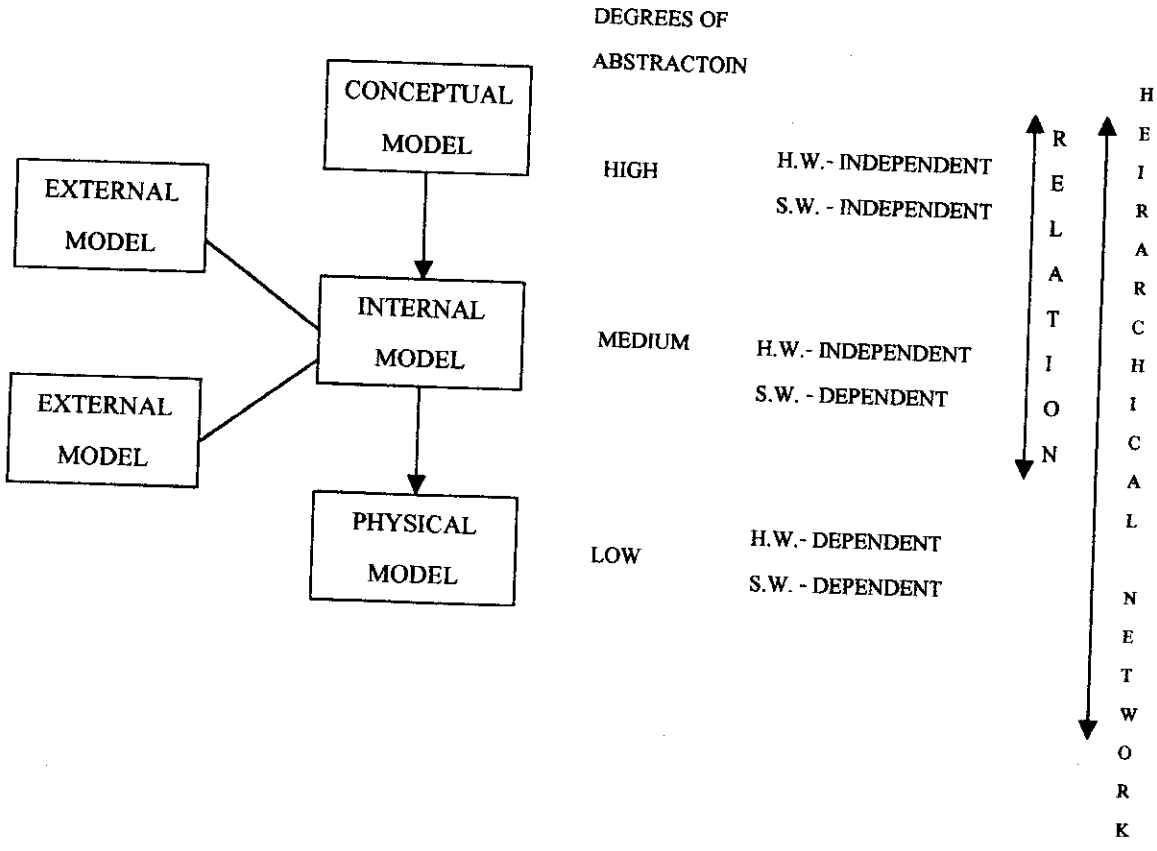
7.2 ระดับของข้อมูลเชิงนามธรรม (Degrees of Data Abstraction)

The American National Standard Institute/Standards Planning and Requirements Committee (ANSI/SPARC) กำหนด Data Model ใช้ 3 ระดับตาม Abstraction :

Conceptual Data Model

External Data Model

Internal Data Model



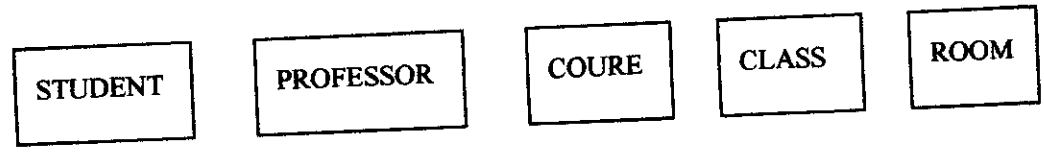
แบบจำลองของแนวคิด (Conceptual Model)

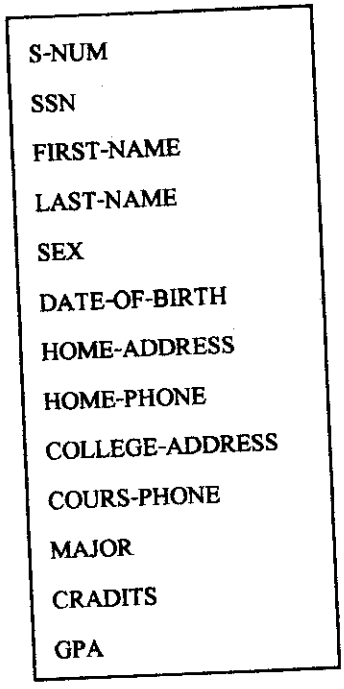
Conceptual Model : มีความเป็น Abstraction สูงสุด แสดงให้เห็นถึง Global View ของข้อมูล คือ แสดงถึงข้อมูลทั้งหมดขององค์กร

Conceptual Model : เป็นพื้นฐานของการระบุและกำหนดรายละเอียดของข้อมูลหลักที่มีในองค์กร ซึ่งแสดงในรูปของ E-R Model

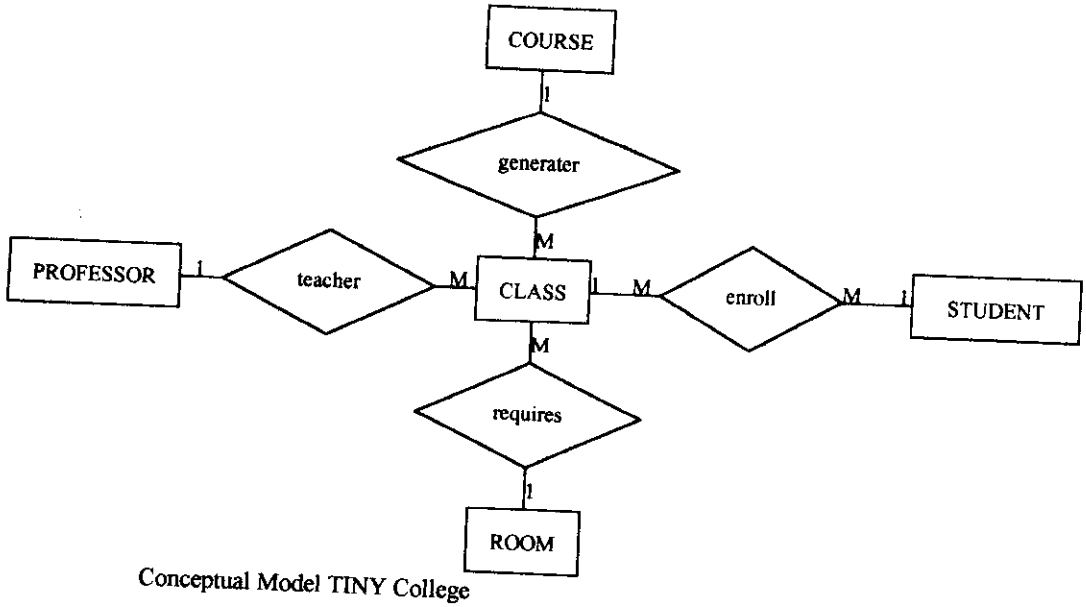
EX Data Environment ของ TINY College

MAIN OBJECTS : STUDENTS , PROFESSORS , COURSES , CLASS และCLASSROOM ทั้งหมดนี้เป็น MAIN ENTITIES ที่ซึ่งรวบรวมและจัดเก็บข้อมูลไว้





STUDENT'S ATTRIBUTES



Conceptual Model TINY College

ข้อดี

1. Conceptual Schema ซึ่งทำให้เข้าใจ Data Environment ขององค์กร ดีขึ้น
2. เน้น S.W. + H.W. Independent

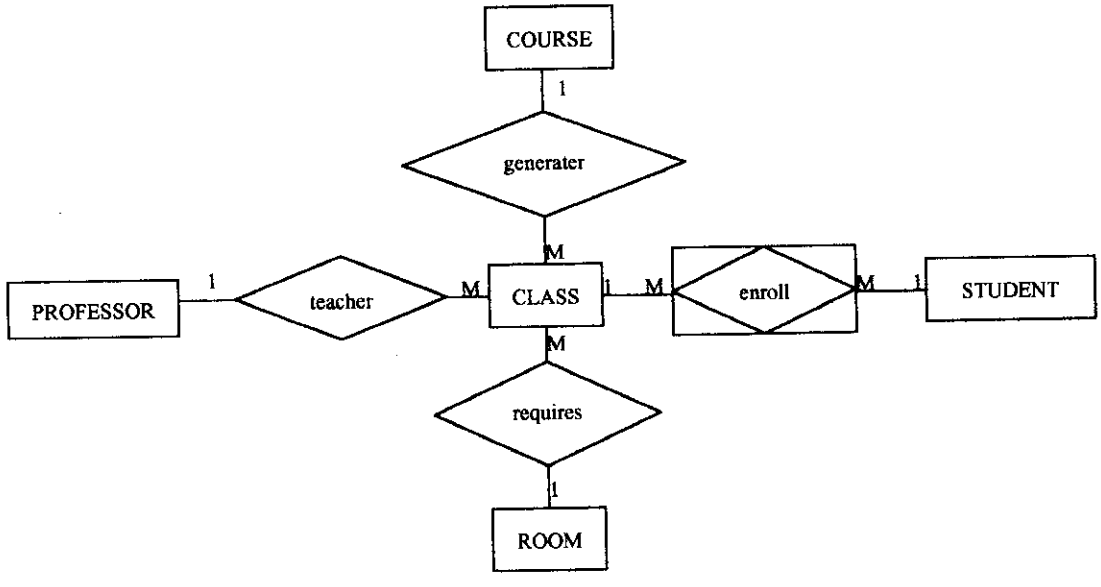
S.W. Independent หมายถึง Model ไม่ขึ้นกับ DBMS ในการที่จะนำ Model มาใช้งาน

H.W. Independent หมายถึง Model ไม่ขึ้นกับ H.W. ในการที่จะนำ Model มาใช้งาน

แบบจำลองภายใน (Internal Model)

เมื่อมีการเลือกใช้ DBMS แล้ว Conceptual Model ที่สร้างขึ้นจะได้รับการปรับเปลี่ยนให้เข้ากับ DBMS นั้น เกิดเป็น Internal Model ขึ้น นั่นคือ Designer จะนำข้อกำหนด (Characteristics) และข้อจำกัด (Constraints) ที่มีใน Conceptual Model มาปรับให้สัมพันธ์กับ DBMS ที่จะใช้ ทำให้ Internal Model นั้นมีลักษณะเป็น S.W. Dependent นั่นคือ ถ้าเราเปลี่ยน DBMS เราต้องเปลี่ยนแปลงบางอย่างของ Internal Model ด้วย

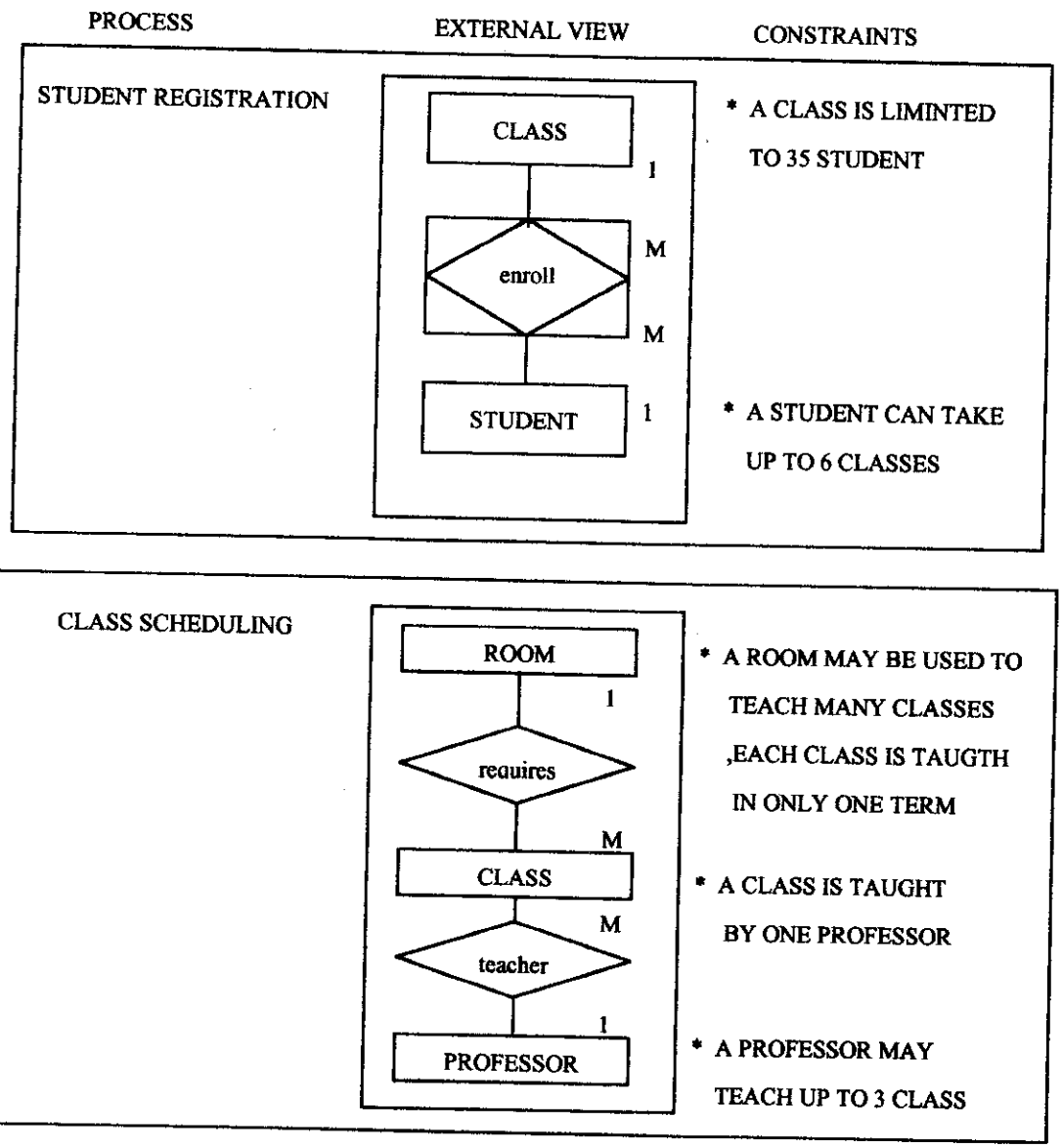
ในการพัฒนารายละเอียดของ Internal Model นั้น ถ้า Designer เลือกใช้ Hierarchical Database Model หรือ Network Database Model แล้วละก็ การกำหนด Internal Model จะต้องเป็นไปอย่างถูกต้องและละเอียด เพราะต้องกำหนดรายละเอียดที่ชัดเจนสำหรับการจัดเก็บข้อมูลและ Path ที่ใช้เข้าถึงข้อมูล แต่ถ้า Designer ใช้ Relational Model ในการทำงาน ก็ไม่ต้องกำหนดรายละเอียดมากมายนักเพราะ RDBMS มีการ Access แบบ "Transparently" นั่นคือ Designer ไม่ต้องกังวลถึง Access Path



INTERNAL MODEL

แบบจำลองภายนอก (External Model)

External Model สร้างขึ้นมาจาก Internal Model เป็นมุมมอง Data ของ End-User ซึ่งจะมีมุมมองที่เน้นเป็นขอบเขตเฉพาะในธุรกิจแต่ละประเภท ธุรกิจหนึ่งๆ อาจแบ่งเป็นธุรกิจย่อยๆ ได้หลายประเภท เช่น Sales , Finance , Marketing เป็นต้น ซึ่งธุรกิจย่อยแต่ละประเภทจะมี Characteristics และ Constraints ของข้อมูลที่ใช้แตกต่างกันไป และ จะใช้เพียงส่วนของข้อมูลเท่านั้น (ไม่ใช่ข้อมูลทั้งหมด) ดังนั้นมุมมองข้อมูลจึงแตกต่างจาก Internal Model คือกลายเป็น External Model



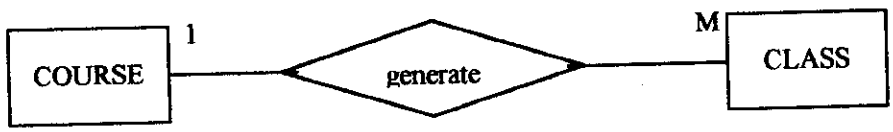
7.3 แบบจำลองของ E-R (E-R Model)

เนื่องจาก Relational Model เป็น Database Model ที่ได้รับความนิยมในการใช้งานมากที่สุด RDB จึงมีบทบาทสำคัญต่อการทำ Database Design ดังนั้น E-R Model ซึ่งใช้ในการออกแบบ Conceptual Model จึงมีบทบาทสำคัญเช่นกันในการทำ Database Design เพราะ RDB นั้น จะเน้นเรื่องการออกแบบ Conceptual Model

Entity Relationship Model เป็นเครื่องมือที่นิยมใช้เพื่อ

- แปลรูปแบบข้อมูลในมุมมองต่างๆ สำหรับ Designer, Programmer และ User
- กำหนดการประมวลผล และ ข้อจำกัดของข้อมูลใน View ต่างๆ
- ช่วยสร้างฐานข้อมูล

ตัวอย่าง เช่น E – R Model ที่แสดงความสัมพันธ์ระหว่าง Course กับ Class โดย Course อาจจะมีได้หลาย Class แต่ Class 1 Class จะมี Course ได้ 1 Course เท่านั้น ซึ่งการบรรยายเช่นนี้ทำให้เข้าใจได้ยาก จึงใช้ E – R Diagram เข้ามาช่วย ก็จะได้ E – R Diagram ที่อธิบายให้เห็นถึง E – R Model นี้ดังนี้

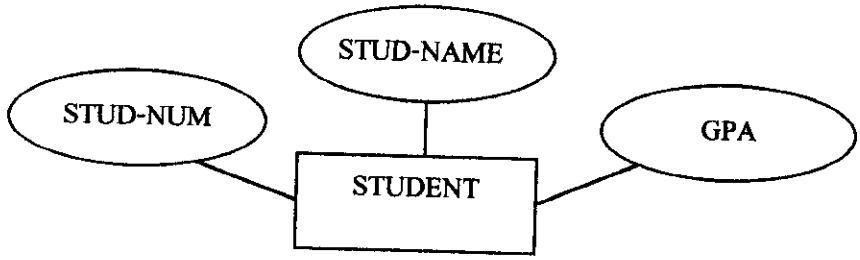


E-R Model ได้รับการพัฒนาขึ้นมาในปี 1976 โดย PETER CHEN

องค์ประกอบของ E-R Model

Entity : ใน E-R Model นั้น Entity จะหมายถึง Entity Set หรือ ก็คือ Table นั่นเอง ถ้าต้องการอ้างถึง Row ใน Table จะเรียกว่า Entity Instance
แสดงโดยใช้รูป 

Attribute : แสดงในรูป  และเชื่อมต่อกับ Entity โดยใช้เส้นตรงเชื่อมโดยจะแสดงถึงบรรดา Attribute ที่มี



STUDENT (STUD-NUM, STUD-NAME, GPA)

โดย Attribute จะมีการกำหนด "Domain" ซึ่งก็คือ ค่าที่เป็นไปได้ของ Attribute นั้น

ตัวอย่างเช่น GPA จะมี DOMAIN เป็น (0.0, 4.0)

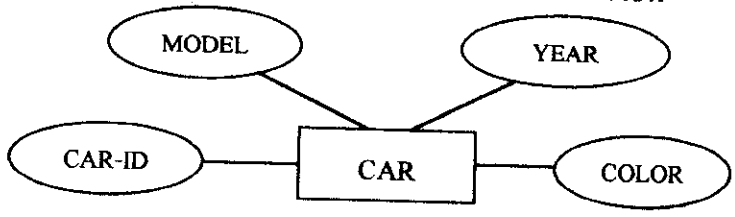
SEX จะมี DOMAIN เป็น (M,F)

Attribute แบ่งเป็น

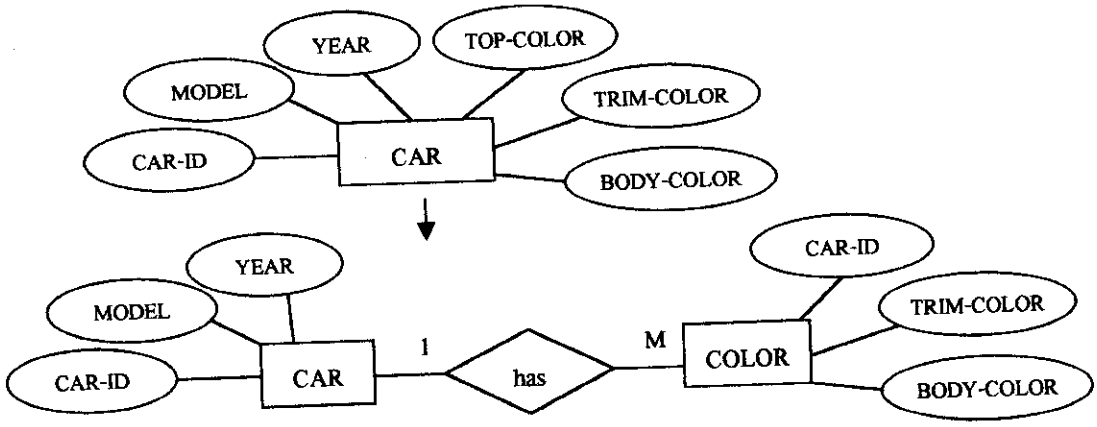
1. Composite Attribute คือATTR ที่สามารถแบ่งย่อยได้อีก ตัวอย่างเช่น STUD-ADDRESS STUD-STRESS, STUD-CITY
2. Simple Attribute คือ ATTR ที่ไม่สามารถแบ่งได้อีก เช่น GPA, SEX เป็นต้น

Single-Value Attribute คือ ATTR ที่มีค่าได้ค่าเดียว เช่น SIN

Multi-Value Attribute คือ ATTR ที่มีค่าได้หลายค่า เช่น

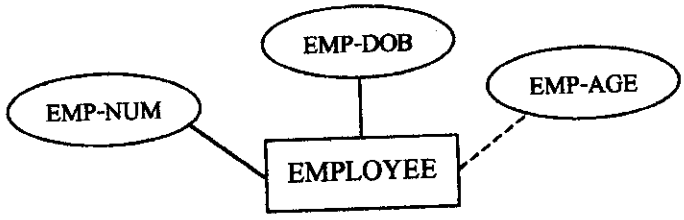


ต้องปรับให้เป็น Single View



SECTION TOP,TRIM,MODEL

Derived Attribute คือ Attribute ที่หาค่าได้โดยใช้ Algorithm ช่วยจึงไม่จำเป็น ต้องเก็บไว้ใน ฐาน ข้อมูล เช่น AGE



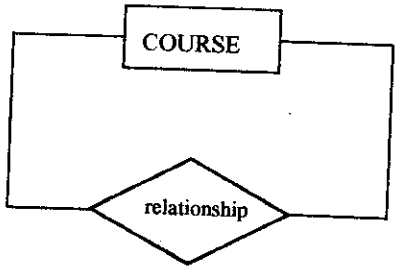
อาจหาได้จาก EMP-DOB

Relationship คือ ความสัมพันธ์ระหว่าง Entity โดยต้องมีการระบุความสัมพันธ์ดังกล่าวในรูปของการ ตั้งชื่ออธิบายความสัมพันธ์ (ใช้Active Verb)

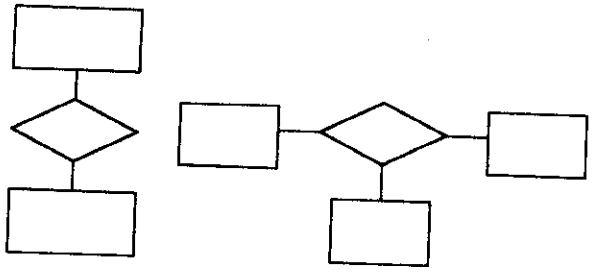
แสดงในรูปของ ◇

Entity ที่เกิด Relationship กับเรียก Participants

Relationship มีการกำหนด Degree



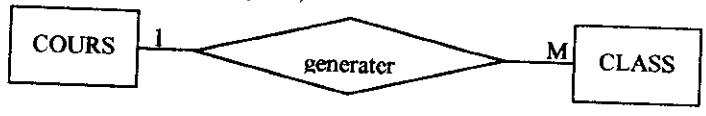
RECURSIVE RELATIONSIHP

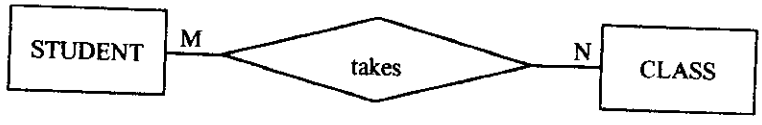


TERNARY

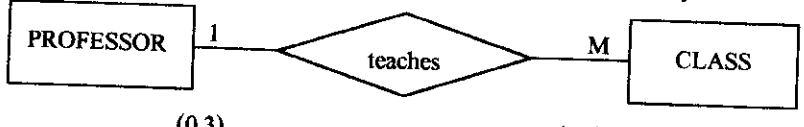
Connectivity ใช้อธิบาย ชนิดของความสัมพันธ์ที่เกิดขึ้นระหว่าง Entity

(1:1,1:M,M:N)



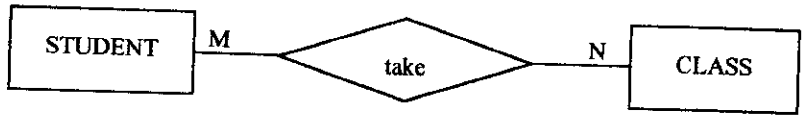


Cardinality ใช้ระบุจำนวนการเกิดของ Entity ที่ซึ่งสัมพันธ์กับอีก Entityหนึ่ง



(0,3) (1,1)

หมายความว่า PROFESSOR 1 คน อาจจะไม่มีการสอน CLASS ใดเลยหรืออาจสอนได้แต่ไม่เกิน 3 CLASS และ CLASS จะต้องมี PROFESSOR สอนเสมอและสอนได้ 1 คนเท่านั้น



(1,6) (0,35)

หมายความว่า นักเรียน (STUDENT) จะต้องลงเรียนอย่างน้อย 1 CLASS แต่ไม่เกิน CLASSES และ CLASS จะมีนักเรียนเรียนได้ไม่เกิน 35 คน หรือ อาจไม่มีนักเรียนเรียนเลยก็ได้

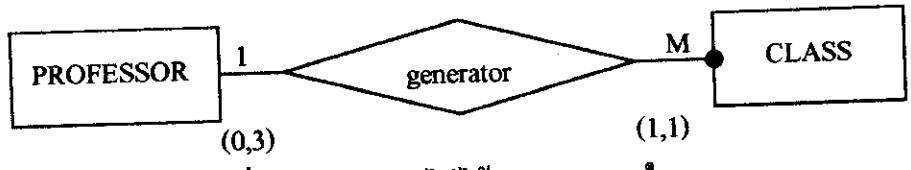
Existence Dependency (การขึ้นอยู่กับ)

ถ้าข้อมูลของ Entity (A) หนึ่งเกิดขึ้นได้เพราะข้อมูลของอีก Entity หนึ่ง (B) เรียกว่าเกิด Existence – Dependent

<u>ตัวอย่าง</u> เช่น CLASS – Entity เกิดขึ้นจาก COURSE-Entity แสดงว่า CLASS เป็น Existence Dependency กับ COURSE เพราะถ้าไม่มีการสร้างข้อมูลของ COURSE ก่อนแล้ว จะสร้าง COURSE ขึ้นมาไม่ได้ ต้องกำหนด Table ของ COURSE ก่อนเสมอ

Relationship Participation (ความสัมพันธ์ที่มีส่วนร่วม)

Entity ที่มี Relationship อาจเป็น Participation แบบ Optional, Mandatoly

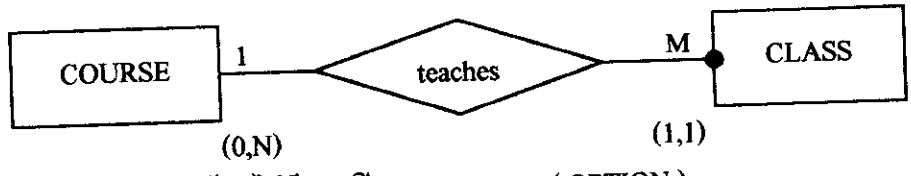


นั่นคือ อาจจะมีความเป็นไปได้ที่ PROFESSOR ไม่ได้สอน CLASS ใดเลย

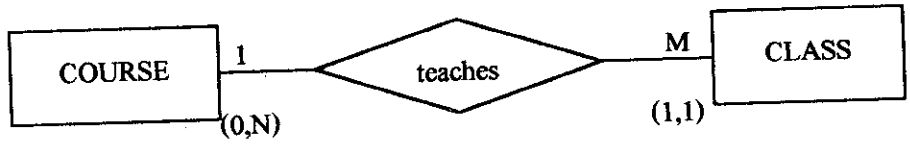
ดังนั้น CLASS เป็น OPTIONAL PARTCIPANTS กับ PROFESSOR

แต่ CLASS ที่เปิดสอนต้องมี PROFESSOR สอนเสมอ

ดังนั้น PROFESSOR เป็น MANDATARY กับ CLASS



คือ อาจเปิด COURSE ใช้แต่ไม่มีการเปิด CLASS สอน (OPTION)

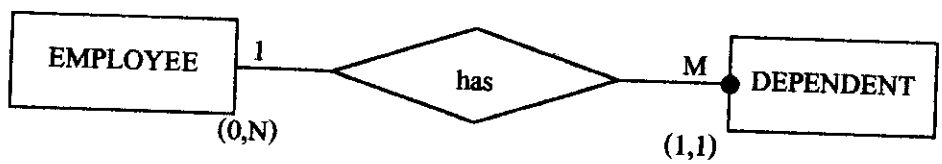


ต้องมีอย่างน้อย 1 CLASS ใน COURSE ที่เปิด (ไม่ OPTION)

Weak Entity

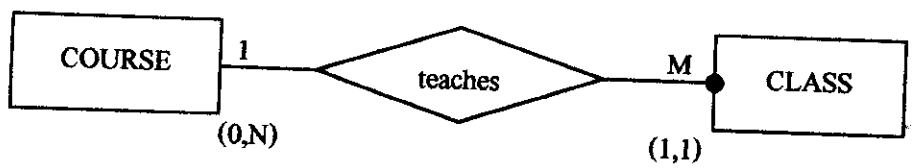
คือ Entity ที่ซึ่ง

1. เป็น Existence – Dependent ของ Entity อื่น
2. มี Primary Key ที่ได้มาทั้งหมด หรือ บางส่วนจาก Entity ที่เป็น Parent



E-NUM	E-NUM
E-NAME	D-NUM
E-JOB	D-NAME
	D-JOB

Dependent เป็น Weak Entity



COURSE (CRS-CODE,…..,……..,…….)

CLASS (C-CODE,……..,……..,……..,CRS-CODE,…….)

เป็น Existence – Dependent แต่ไม่ใช่ Weak เพราะคุณสมบัติข้อ 2 ไม่ตรงเนื่องจาก Primary Key ของ CLASS Table ไม่ได้มาจาก CRS-CODE

แต่ถ้ากำหนดใหม่

COURSE (CRS-CODE,…..,……..)

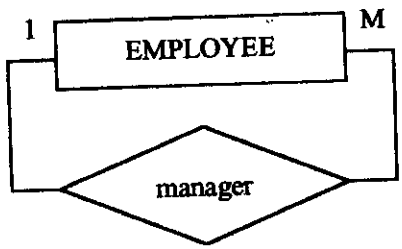
CLASS (CRS-CODE,C-SECTION,……,…….)

จะเป็น Weak Entity ทันทีเพราะตรงทั้งคุณสมบัติข้อ 1 + ข้อ 2

Recursive Entity

คือ Entity ที่ซึ่งมีความสัมพันธ์ได้กับ Entity ชนิดเดียวกัน โดยปกติจะพบใน Unary Relationship

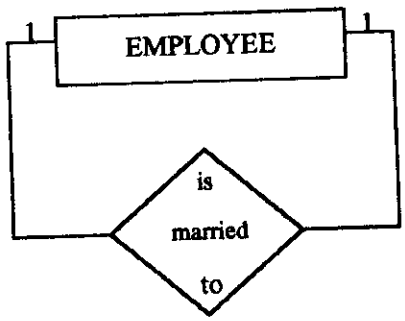
EX: 1 : M



An Employee May Manage Many Employees

And Each Employee is Managed by One Employee

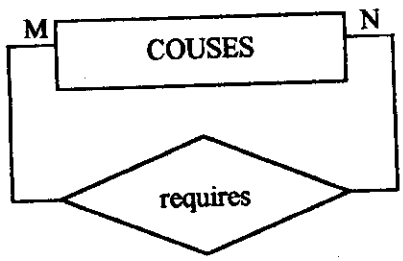
1 : 1



An Employee May be Married to one and

Only on Other Employee

M : N



A Course May be a Prerequirite to Many

Other Courses and Each Course May have

Many Other Courses as Prerequisites

ในความสัมพันธ์แบบ 1 : 1 ใน Recursive Entity เราอาจสร้างเป็น Table ได้ดังนี้

E-NUM	E-NAME	E-ADDRESS	E-SPOUSE
345	JAMES	XXXXX...	347
346	ANN	XXXX...	349
347	ALICS	-------	345
348	ROBERT	------	
349	PAUL	------	346

CT 316 (S)

ความสัมพันธ์แบบ Unary Relationship นั้นพบได้ในการจัดการข้อมูล ของวงการอุตสาหกรรม

<u>ตัวอย่างเช่น</u>

P-CODE	P-DESCRIPTION	P-INSTOCK	P-USED	P-CF-P
AA216	WASHER	432	4	C-130
AB212	COTTER COPPER	1042	12	C-130
C-130	ROTOR ASSEMBLY	36		
E-129	STEEL SHANK	128	2	C-130
DA121	ROTOR SCADE	215	1	C-130

แสดงว่า C-130 เป็นอุปกรณ์ (PART) ที่ประกอบด้วย AA216, AB212, E-129, DA121 แต่ถ้า PART ตัวหนึ่งใช้ในการประกอบ PART อื่นได้หลายๆ PART และ PART นี้ยังประกอบด้วย PART ย่อยหลายๆ PART อีก เราต้องใช้ตาราง 2 ตารางในการเก็บข้อมูลเกี่ยวกับ PART

COMPONENT

COM-CODE	P-CODE	P-USED
C-130	AA216	4
C-130	AB212	12
C-130	E-129	2
C-130	DA121	1
C-131	AA216	3
C-131	E-129	1
C-131	DB110	2

M N

PART

P-CODE	P-DESCRIPTION	P-INSTOCK
AA216	WASHER	432
AB212		
C-130		
E-127		

ดังนั้นความสัมพันธ์แบบ M : N ของ COURSE กับ PREREQUISITE จะเป็น

COURSE

CRS-CODE	CRS-DESC	CRS-CEDIT
AC101	ACCOUNT 1	3
AC102	ACCOUNT 2	3
CT105	INTRO TO COMPUTER	3

PREREQUISITE

CRS-CODE	PRE-CODE
AC201	AC101
AC302	AC101
AC701	AC201
AC401	MA232

ส่วนความสัมพันธ์แบบ 1 : M จะเป็นดังนี้

E-NUM	E-NAME	E-MANAGER
101	JOHN	
102	PAUL	101
103	ALICE	101
104	PETER	
105	ANN	104
106	BOB	104

Composite Entity

ใน E-R Model ตามแนวคิดของ CHEN จะพบว่า Relationship จะไม่มี Attribute แต่เราจะพบว่า ถ้า Entity ของเรามีความสัมพันธ์แบบ M : N เราจะต้องพยายามทำให้เป็น 1 : M ให้ได้ นั่นคือเป็นการสร้าง "BRIDGE" ระหว่าง Entity ทั้ง 2 นั่นเอง โดย BRIDGE จะเป็น Entity ที่ประกอบด้วย Primary Key จากทั้ง 2 Entities ที่มาเชื่อมกันดังนั้นเราจึงเรียก BRIDGE ว่าเป็น Composite Entity

STUDENT

S-NUM	S-NAME
32145	ROBERT
32147	ANNA

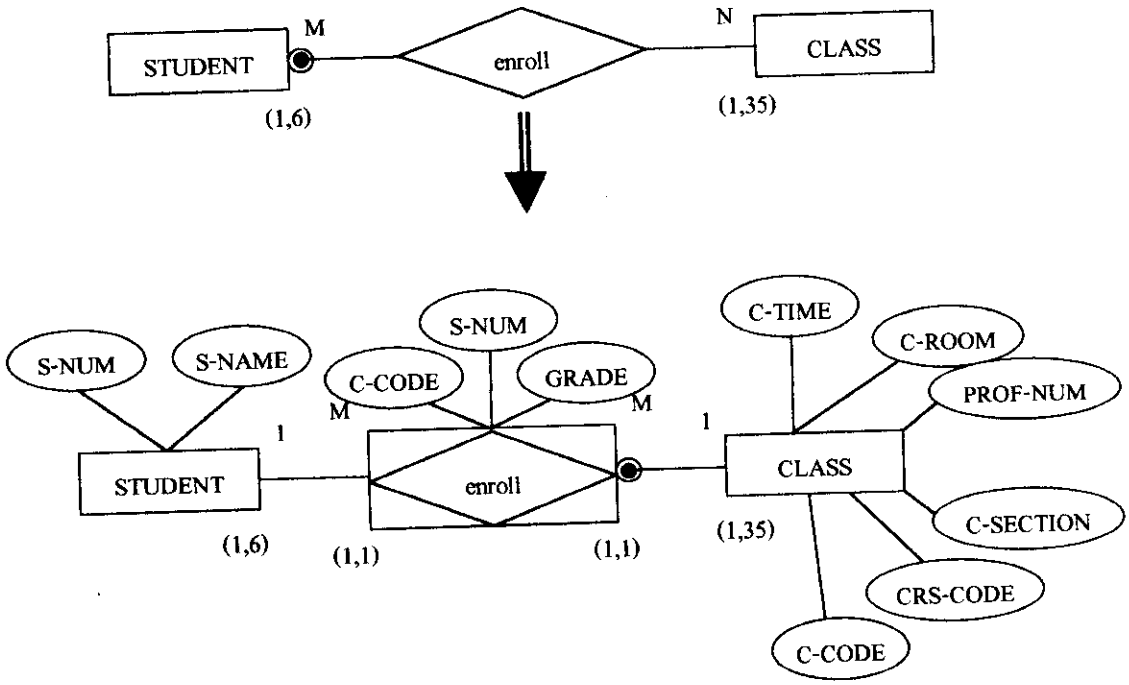
ENROLL

C-CODE	S-NAME	GRADE
CT105	32145	P
CT105	32147	G
CT21	32145	P
CT211	32147	P

CLASS

C-CODE	CRS-CODE	C-SECTION	C-TIME	C-ROOM	PRO-NUM
CT105	CS	1	9:30-11:30	R111	P01
CT211	CS	~			

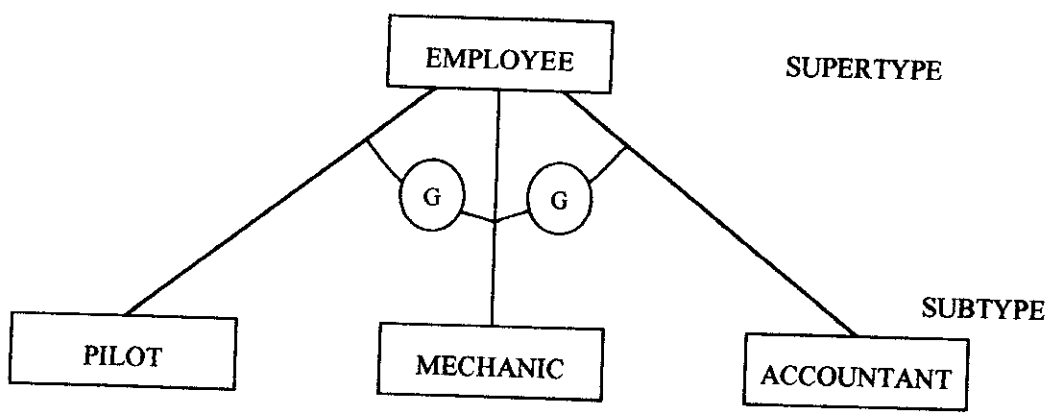
เมื่อเรานำความสัมพันธ์ระหว่าง Entity ที่ชื่อ STUDENT และ CLASS มาพิจารณาเราจะได้ว่า


STUDENT
M
enroll
N
CLASS
(1,6)
(1,35)
S-NUM
S-NAME
S-NUM
C-CODE
GRADE
M
M
C-TIME
C-ROOM
PROF-NUM
1
1
STUDENT
enroll
CLASS
C-SECTION
(1,6)
(1,1)
(1,1)
(1,35)
CRS-CODE
C-CODE

Entity Supertypes and Subtypes

ในบางกรณีข้อมูลที่จัดเก็บไว้ใน Table เดียวกันอาจมีรายละเอียดในการจัดเก็บต่างกัน ตัวอย่างเช่น ข้อมูลของ Table ที่ชื่อ EMPLOYEE ของบริษัทการบินแห่งหนึ่ง เราพบว่า EMPLOYEE ที่เป็นนักบินอาจมีข้อมูลเกี่ยวกับ ทะเบียนการบิน, ชั่วโมงการบิน, การฝึกพิเศษ เป็นต้น ซึ่งข้อมูลนี้ EMPLOYEE คนอื่นๆ จะไม่มี ทำให้เกิดช่องว่างที่สูญเปล่า หรือ หน่วยความจำที่สูญเปล่าในการสร้าง หรือ จัดเก็บ Table ปัญหานี้สามารถแก้ไขได้โดยการใช้ "Generalization Hierarchy" ในการแสดง Entity ที่มีการ Share ข้อมูลร่วมกัน (ใน ตัวอย่างนี้คือ E-NUM, E-NAME, E-ADDRESS, E-DATE)

โดยพื้นฐานแล้ว Generalization Hierarchy จะแสดงถึงความสัมพันธ์แบบ Parent-Child ใน Hierarchical Model แต่ใน Relation Model จะแสดงถึงความสัมพันธ์ระหว่าง Higher-Level Supertype Entity กับ Low-Level Subtype Entity โดยในความเป็นจริง Supertype จะเก็บ Shared Attributes และ Subtype จะเก็บ Unique Attribute



โดย Subtype จะสืบทอด Attribute และ Relationship มาจาก Supertype
ส่วน Subtype แต่ละตัวจะเป็น Disjoin กัน (แสดงโดย (G))

EX

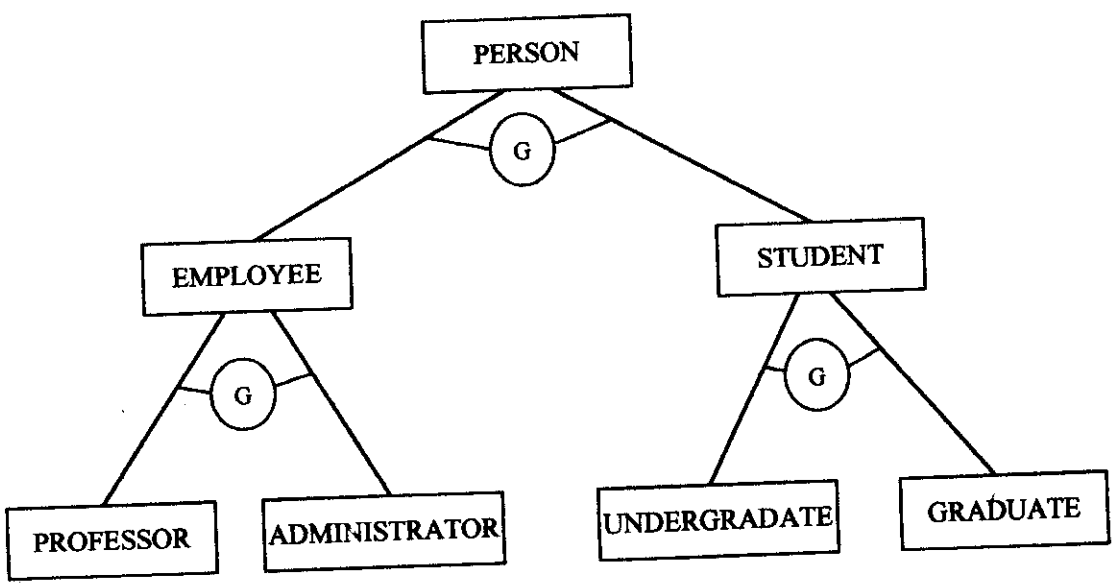
EMPLOYEE

E-NUM	E-NAME	E-ADDRESS	E-DATE	E-POSITION
32100	สมชาย	XXXX-XX	XX-XX-XX	PILOT
:	:	:	:	:
:	:	:	:	:

PILOT

E-NUM	E-LICENSE	P-TRAINED	P-HOURS
32100	XXXXXX	XXX	XXX

ตัวอย่าง เราสามารถพิจารณาข้อมูลในมหาวิทยาลัย ที่เกี่ยวกับ บุคลากร ได้เป็น



7.4 การพัฒนา E-R Diagram

กระบวนการของการออกแบบ ฐานข้อมูล เป็นแบบ Iterative นั่นคือเป็นการทำงานแบบซ้ำๆ ไม่ใช่จะแล้วเสร็จได้ในครั้งเดียว นั่นคือจะเริ่มจากการพิจารณาการทำงานและการปฏิบัติการทั้งหมดที่มีในวงกลมก่อน จากนั้นจะทำการทบทวน และเพิ่มเติมรายละเอียดเข้าไปใน E-R Model จะทำเช่นนี้ไปเรื่อยๆ จนกระทั่งได้รูปแบบที่เป็นที่ยอมรับทั้งฝ่าย Designer และ End-User

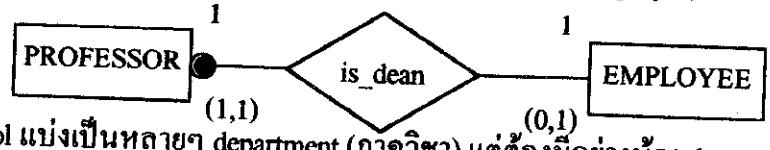
ในขณะที่มีการออกแบบนั้น นักออกแบบจะใช้ข้อมูลที่ได้จากการสอบถามมาใช้ในการออกแบบนอกจากนี้ยังต้องใช้ข้อมูลที่รวบรวมมาจากเอกสารต่างๆ ที่มีในองค์กรมาช่วยในการพิจารณาออกแบบด้วย

<u>ตัวอย่าง</u> E-R Model สำหรับ TINY Collage

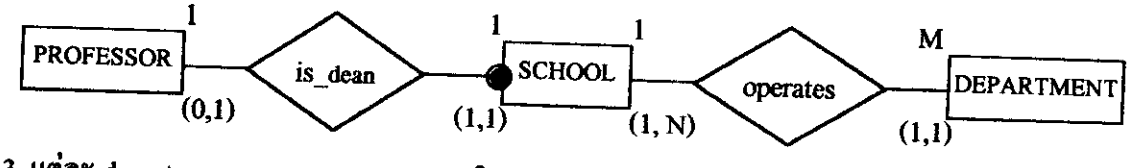
1. Tiny college แบ่งเป็นหลายๆ school เช่น school of business, science ฯลฯ โดยแต่ละ school บริหารงานโดย คณบดี (dean) 1 คน

   dean : school (1 : 1)

   โดยคณบดี (dean) นี้ก็คืออาจารย์ (professor) ซึ่งก็คือ พนักงาน (employee) ชนิดหนึ่งของ Tiny

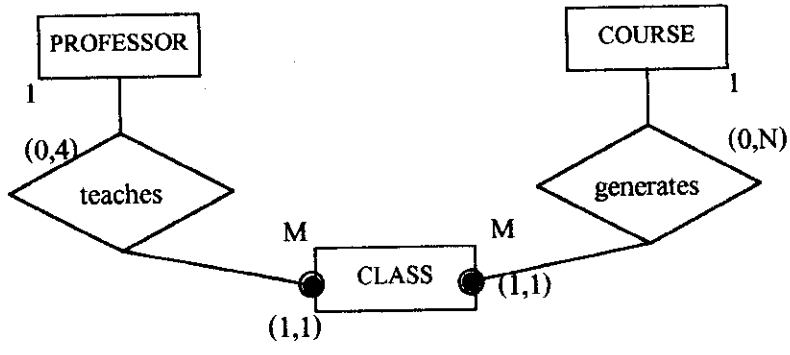


2. แต่ละ school แบ่งเป็นหลายๆ department (ภาควิชา) แต่ต้องมีอย่างน้อย 1 ภาค แต่ department หนึ่งขึ้นอยู่กับ school เดียวกันเท่านั้น

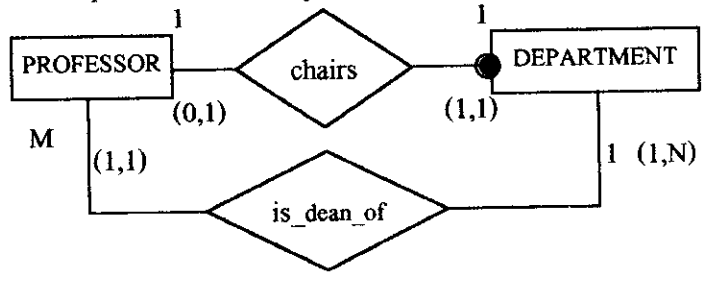


3. แต่ละ department จะเสนอสอบหลายๆ วิชา (coures)

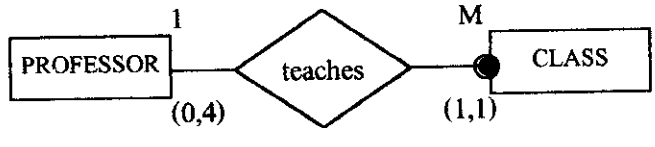
4. แต่ละ department จะเปิดสอนหลายๆ class ของวิชา (course) หนึ่งๆ นั่นคือ class ก็คือ section หนึ่งของ course โดยแต่ละ class จะ class จะสอบโดย professor คนหนึ่งในช่วงเวลาหนึ่งและสถานที่หนึ่งแสดงว่า courses หนึ่งๆ อาจเปิดสอนได้หลายๆ class แต่อาจมี course ที่ไม่มีการเปิด class สอบเลยก็ได้



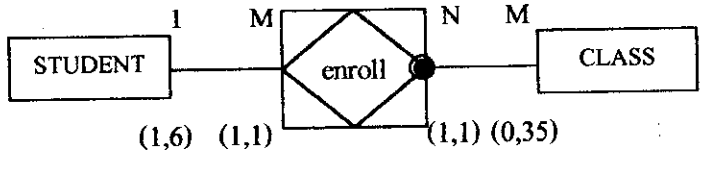
5. แต่ละ department จะมี professor ประจำอยู่หลายๆ คน แต่จะมีหนึ่งคนทำหน้าที่เป็น chair



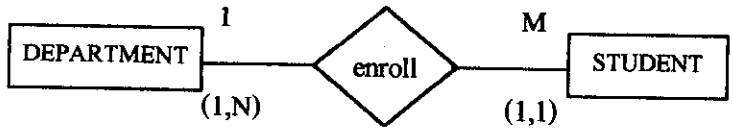
6. แต่ละ PROFESSOR จะสอนได้ไม่เกิน 4 CLASS เป็นหรือบาง PROFESSOR อาจไม่สอนเลยคือทำวิจัยเท่านั้น



7. นักศึกษา STUDENT สามารถสมัครเรียนได้หลายๆ CLASS ในเทอมๆ หนึ่ง (แต่ลงวิชานั้นได้ครั้งเดียว) แต่ไม่เกิน 6 วิชา และแต่ละ CLASS จะมีนักศึกษาได้ไม่เกิน 35 คน (หรืออาจไม่มีเลยก็ได้)

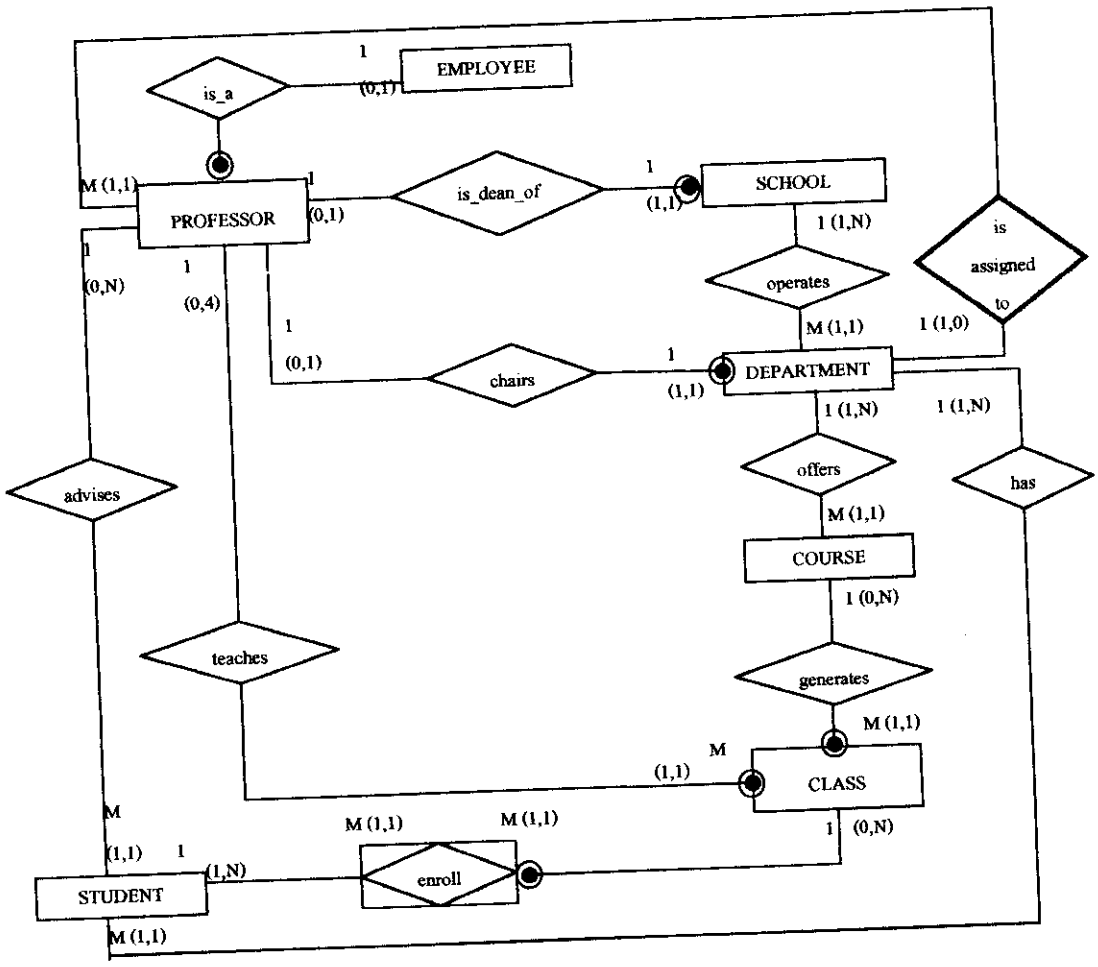


8. แต่ละ DEPARTMENT จะมี STUDENT ได้หลายๆ คนที่มีวิชาเอกอยู่ใน DEPARTMENT นั้น ซึ่ง STUEDNT แต่ละคนจะอยู่ใน DEPARTMENT เดียวเท่านั้น (ตามความเป็นจริง)

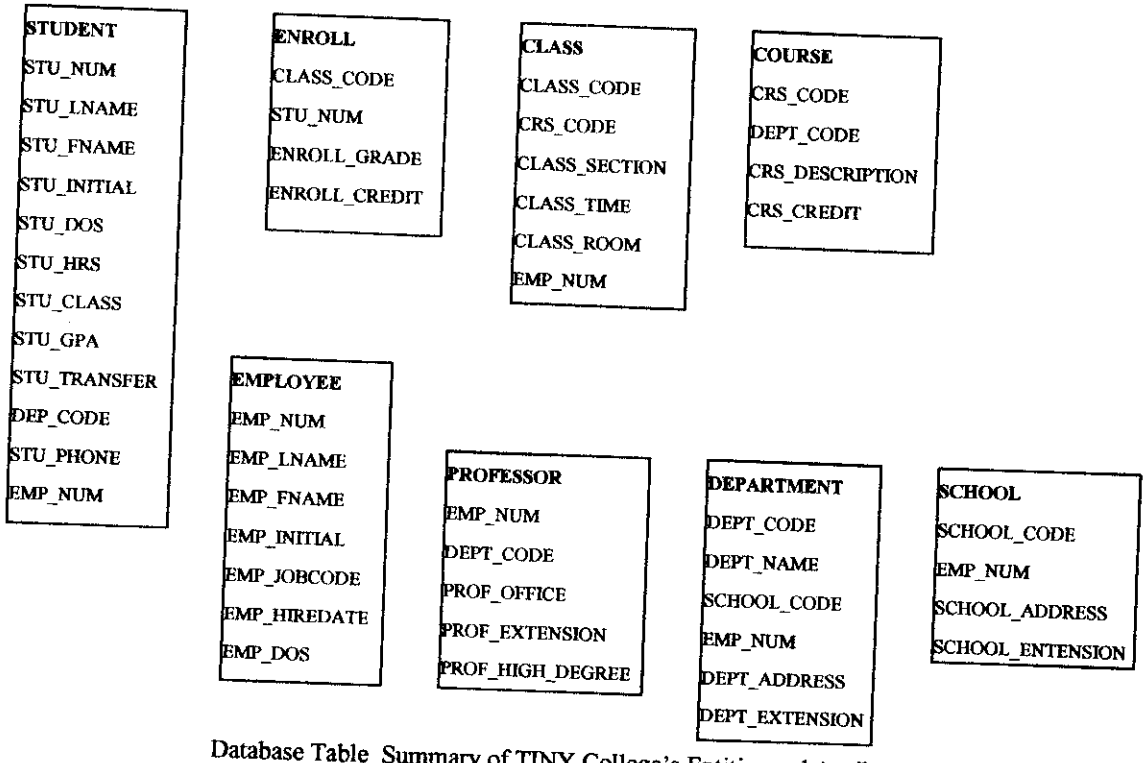


9. STUDENT แต่ละคนจะต้องมีอาจารย์ที่ปรึกษา 1 คน ซึ่งก็คือ PROFESSOR คนใดคนหนึ่งนั้นเอง โดย PROFESSOR บางคนอาจไม่มี STUDENT ให้ปรึกษาเลย

และเมื่อรวมรายละเอียดของความสัมพันธ์ต่างๆ (ที่ได้พิจารณามาแล้ว) เข้าด้วยกัน จะได้ E-R diagrams ของ Tiny college ดังนี้



ซึ่งเมื่อนำมาใช้ในการพัฒนาระบบฐานข้อมูลจะสามารถกำหนดตาราง (table) ที่จะใช้งานได้ดังนี้



Database Table Summary of TINY College's Entities and Attribute

นำมาสร้างตารางจริงในการใช้งานด้วยภาษา DDL ได้เป็น

```
CREATE TABLE STUDENT (
    STU_NUM          INTEGER          NOT NULL
    STU_LNAME        CHAR(15)         NOT NULL
    STU_FNAME        CHAR(15)         NOT NULL
    STU_INITIAL      CHAR(1)
    STU_DOS          DATE
    STU_HRS          SMALLINT
    STU_CLASS        CHAR(5)          NOT NULL
    STU_GPA          DECIMAL(1.1)
    STU_TRANSFER     CHAR(1)
    DEP_CODE         CHAR(5)          NOT NULL
```

```
    STU_PHONE           CHAR(4)
    EMP_NUM             SMALLINT
    PRIMARY KEY (STU_NUM)
    FOREIGN KEY (DEPT_CODE) REFERENCES DEPARTMENT
        ON DFLETE RESTRICT
        ON UPDATE CASCADE
    FOREIGN KEY (EMP_NUM REFERENCES PROFESSOR
        ON DELETE RESTRICT
        ON UPDATE CASCADE);

CREATE TABLE PROFESSOR (
    EMP_NUM             SMALLINT        NOT NULL    UNIQUE
    DEPT_CODE           CHAR(5)         NOT NULL
    ROOM_CODE           CHAR(8)
    PHOF_EXTENSION      CHAR(4)
    PHOF_HIGH_DEGREE    CHAR(5)
    PRIMARY KEY (EMP_NUM)
    FOREIGN KEY (EMP_NUM) REFERENCES EMPLOYEE
        ON DELETE CASCADE
        ON UPDATE CASCADE
    FOREIGN KEY (DEPT_CODD) REFERENCES DEPARTMENT
        ON DELETE RESTRICT
        ON UPDATE CASCADE);

CREATE TABLE ENROLL (
    CLASS_CODE          CHAR(8)         NOT NULL
    STU_NÚM             INTEGER         NOT NULL
    ENROLL              CHAR(1)
```

```
    ENROLL_CREDIT      SMALLINT
    PRIMARY KEY (CLASS_CODE,STU_NUM)
    FOREIGN KEY (CLASS_CODE) REFERENCES CLASS
        ON DELETE RESTRICT
        ON UPDATE CASCADE
    FOREIGN KEY (STU_NUM) REFERENCESS STUDENT
        ON DELETE CASCADE
        ON UPDATE CASCADE);

CREATE TABLE COURSE (
    CRS_CODE           CHAR(8)          NOT NULL UNIQUE
    DEPT_CODE          CHAR(5)          NOT NULL
    CRS_DESCRIPTION    CHAR(15)         NOT NULL
    CRS_CREDIT         SMALLINT         NOT NULL
    PRIMARY KEY (CRS_CODE)
    FOREIGN KEY (DEPT_CODE) REFERENCES DEPARIMENT
        ON DELETE RESTRICT
        ON UPDATE CASCADE);

CREATE TABLE EMPLOYEE (
    EMP_NUM            SMALLINT         NOT NULL UNIQUE
    EMP_LNAME          CHAR(15)         NOT NULL
    EMP_FNAME          CHAR(15)         NOT NULL
    EMP_INITIAL        CHAR(7)
    EMP_JOBCODE        SMALLINT
    EMP_HIREDATE       DATE
    EMP_DOB            DATE
    PRIMARY KEY (EMP_NUM));
```

```
CREAT TABLE DEPARTMENT (
        DEPT_CODE          CHAR(5)          NOT NULL UNIQUE
        DEPT_NAME          CHAR(15)         NOT NULL
        SCHOOL_CODE        CHAR(5)          NOT NULL
        EMP_NUM            SMALLINT         NOT NULL
        DEPT_ADDRESS       CHAR(15)
        DEPT_EXTENSION     CHAR(4)
        PRIMARY KEY (DEPT_CODE)
        FOREIGN KEY (EMP_NUM) REFERENCES SCHOOL
              ON DELETE RESTRICT
              ON UPDATE CASCADE
        FOREIGN KEY (SCHOOL_CODE) REFERENCES SCHOOL
              ON DELETE RESTRICT
              ON UPDATE CASCADE);


CREATE TABLE SCHOOL (
        SCHOOL_CODE        CHAR(5)          NOT NULL UNIQUE
        EMP_NUM            SMALLINT         NOT NULL
        SCHOOL_ADDRESS     CHAR(15)         NOT NULL
        SCHOOL_OFFICE      CHAR(1)
        SCHOOL_EXTENSION   CHAR(4)
        PRIMARY KEY (SCHOOL_CODE)
        FOREIGN KEY (EMP_NUM) REFERENCES EMPLOYEE
              ON DELETE RESTRICT
              ON UPDATE CASCADE);
```