

# Ch. 1

## Introduction to Database

สู้ๆ กันอีกปี!!

### • What's "Machine"

ເກີດມືອນທີ່ໄດ້ປະກາດພວມຈາກບໍລິຫານທີ່ນີ້

### • What's "Thinking"

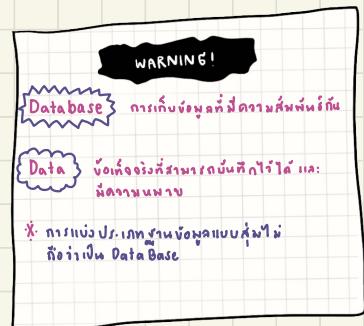
ຄວາມຮັບຮັດຂອງມີຄວາມສົດຍຸດໃນຕົວພົນທັນ

## Common ground of how people communicate

- Mutual knowledge (ຄວາມເຊື່ອມືດີແລະ ກົດ)
- Mutual Beliefs (ຄວາມເຊື່ອມືດີແລະ ກົດ)
- Mutual Assumptions (ຄວາມເຊື່ອມືດີຮັບຮັດ)

## Databases and Database Users

- Type of Database and Database Application
  - ▶ Traditional Application (ແບບດົ່ວ ເລີນ)
    - ຕົວລາວ, text database
  - ▶ More Recent Application
    - Multimedia databases
    - Geographic Information Systems (GIS)
    - Biological and Genome databases
    - Data Warehouses
    - Mobiles databases
    - Realtime and Active databases



## Recent Development

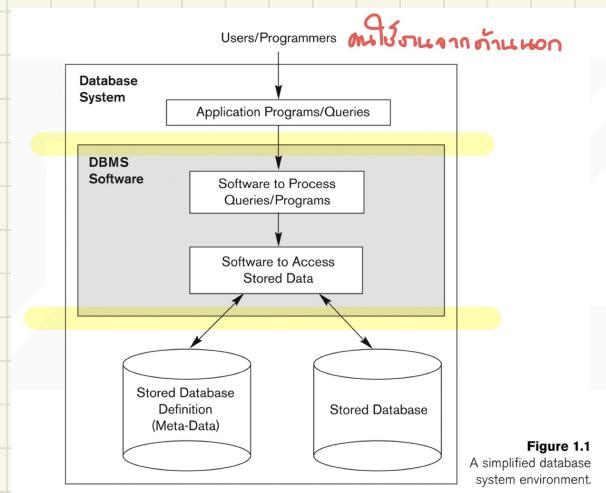
- Social Networks : ລັດຖະບານບໍລິຫານຜູ້ອັນ ແລະ ດາວໂຫຼນນາມ  
ໃນການຜູ້ອັນທີ່ຢູ່, text, Post, Video ໃນ facebook, twitter
- Search - engine ໃນ Google, Yahoo ລັດຖະບານເວັບໄຕ້ງໆ ໄດ້ໃຫ້ໂຄງການຄົນເນັດ
- Non - Database Software vendors → ທີ່ກົດຖານບໍລິຫານທີ່ໄດ້ Relational Database  
ເກີດບໍລິຫານທີ່ໄດ້ອັນດີໃນການນັກງານໂຄງການໂຄງການ ໃນ ການ, vdo, ເພີ້ນ, text  
→ Big Data : ການຄຣີ: ການ comp. ເປັນກົມດ້ວຍນາດໃໝ່
- NoSQL : ແນວດ: ກົດ Big Data, Real time, Web-application
  - ↳ ສຳເນົາແປງຕ່າງ
  - ↳ ສຳເນົາບັດບຸນ
  - ↳ ສຳເນົາສາມາດຮັດໃນມາດປັບປຸງ
- \* Data ໜີ້ທີ່ມີຄວາມເຊື່ອມືດີໃນ cloud ສື່ນາທິການກ່າວ  
Data ວິທາດໃນງ່າຍ໌ machine ອ່ານານນັກ

## Basic Definitions

- Data ข้อมูลที่สามารถบันทึกได้, มีการนับอย่าง
- Database การเก็บข้อมูลที่มีความเกี่ยวข้องกัน
- Mini-world โลกของและองค์กรทางด้านของโลกจริง (UoD : universe of discourse)
- Database Management System (DBMS) ระบบจัดการฐานข้อมูล, Software ที่ใช้สำหรับ Data เป็นสื่อสารระหว่าง User  $\leftrightarrow$  DB user define, descriptive หมาย DB ที่ถูกเก็บใน DBMS ทำให้สะดวก, รวดเร็ว, Share สำหรับผู้ใช้งาน catalog, dictionary  $\rightarrow$  "Meta - Data"
- Database System การสนับสนุน DBMS, Database

ex. university  
student grade  
transcript

## Simplified database system environment



## Typical DBMS Functionality

- Define
  - Data type : ลักษณะของข้อมูล
  - Structure : โครงสร้างของ datatype ตัวอย่าง ตารางห้าช่อง
  - Constraints : ข้อจำกัด กฎต่างๆ
- Construct (load) : นำข้อมูลเข้ามาไว้ใน Database
- Manipulating : นำร่องการchanging Database
  - Retrieval : Query, Generate report นำข้อมูล
  - Modification : แก้ไขข้อมูล Insert, delete, update
  - Accessing : นำข้อมูลมาใช้งาน
- Processing , Sharing : รองรับการทำงานของ multiple user, program พร้อมกัน
  - Valid, Consistent (สมดุล)

# Application Activities Against a Database

- ms Interact by generating
    - ↳ Queries : ms รีบก, การดูแลระบบ
    - ↳ Transactions : ก่อตั้ง transaction ดำเนินงานเกี่ยวกับข้อมูลเดียวกัน  
※ ไม่เกิด transaction ว่างดำเนินการ fail ต้อง fail ทุก transaction  
พื้นฐานเรียกว่า "Atomicity property" เป็นคุณลักษณะเดียว
  - ไม่อนุญาตให้ user ห้ามล็อกเพื่อกำหนด
  - ต้องห้ามคนต่างประเทศเข้ามาใช้ user requirement, Database

## Additional DBMS Functionality

- Protection, Security Measures : វាគ្នេងការកិច្ចអនុវត្តន៍នៃ Data
  - "Active" numbers នូវ internal action នៃ data
  - Presentation, Visualization data
  - Maintenance database, program និងការបរិច្ឆេទ

## Main Characteristic of the Database approach

- Self-describing nature of a database system
    - ↳ In DBMS catalog នៃការបញ្ជីតម្លៃទូទៅនៃទិន្នន័យ database  
ការបញ្ជីតម្លៃទូទៅនៃទិន្នន័យ, និងការរំពេលឱ្យនា
      - data structure
      - type
      - constraints
    - និងចុងក្រោយ DBMS នឹងរាយការ Database ដែលមានការពិនិត្យ "Meta-data"
  - Insulation between programs and data : និងការបញ្ជីតម្លៃទិន្នន័យ Data, Program
    - ↳ ការបញ្ជីតម្លៃទិន្នន័យ Data, Program
    - ឯកសារក្នុងការបញ្ជីតម្លៃទិន្នន័យ Data structure, និង organize storage ក្នុងក្នុង DBMS ហើយក្នុង program
  - Data Abstraction : ទទួលយកពីការបញ្ជីតម្លៃទិន្នន័យ, និងរាយការបញ្ជីតម្លៃទិន្នន័យ
    - ↳ Data Model : ពិនិត្យ storage detail និង present user ជាតាម conceptual view ឬ logical concept  
និង properties, relationship ក្នុង program refer to data model constructs អនុវត្តន៍យក storage detail
  - View ឬ subset នៃ database នឹងចុងក្រោយការបញ្ជីតម្លៃទិន្នន័យ view និងក្នុង DBMS
  - Sharing of data and multi-user transaction processing : ឯកសារក្នុង user ឱ្យក្នុង DB, update និងការបញ្ចីតម្លៃទិន្នន័យ
  - Concurrency control : DBMS និងក្នុង transaction រាយការណ៍ធម្មាន execute, aborted (cancel)
  - Recovery : subsystem check ឱ្យក្នុង transaction និងក្នុងការបញ្ចីតម្លៃទិន្នន័យ ឬក្នុង DB មិនការពារ
  - OLTP (Online Transaction Processing) : ឯកសារក្នុងការបញ្ចីតម្លៃទិន្នន័យ, និងការបញ្ចីតម្លៃទិន្នន័យ online ឱ្យក្នុងការពិនិត្យ, check ឱ្យក្នុងការពិនិត្យ, ការផ្តល់ទិន្នន័យ



# Ch.2 Database System Concepts and Architecture

## Data Models

ນີ້ແມ່ນມີການສໍາຜົນການໃຫ້ຈຳນວນຂອງ structure, operations, constraints.

## Data Model Structure and Constraints

- Constructs to define database structure

Elements : ມົງກັນເປົ້າກ່ຽວກົດໄສຮັບຊື່  
Ex. un. 

Group of elements (entity, record, table)

Relationships : ອານວິທີກົດລົບອະນຸຍາດ:element

ວິທີກົດລົບລູກ database ລົບກົດຢູ່ Valid

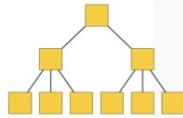
## Data Model Operations

basic model operations : add, delete, insert, update

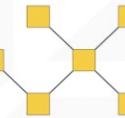
user-defined operations : user-defined functions like compute\_student\_gpa

## Categories of Data Models

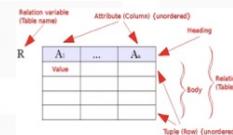
Conceptual (high-level, semantic) data models : ອີ່ນັ້ນຈະຖືກໃຫ້ໃຫ້ມີຄວາມຄຸນຫຼັງຈາກຈະໄດ້ຕົກລົງ, ຈະມີຄວາມສໍາພັນ, ສໍາມັນເກີນຫຼັງຈາກ



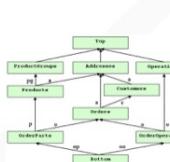
Hierarchical model



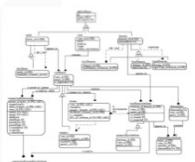
Network model



Relational model



Concept-oriented model



Object-Role model(ORM)  
ອີ່ນັ້ນມີການ

Implementation (representational) (logical schema) data models : ອີ່ນັ້ນໄດ້ຮັບອະນຸຍາດ, ມີຄວາມສໍາພັນ

Physical (low-level, internal) data models : ອີ່ນັ້ນຈະມີການໃໝ່ໃນ Database ສຳເນົາ

X: logical, conceptual ມີການ  
implemented ອີ່ນັ້ນເກີນກັນ

## Schemas

ມີການ Database ຈະໄດ້ຮັບຊື່ໃຫ້ຈຳນວນ "schema diagram"

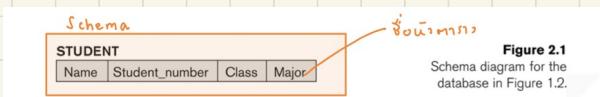
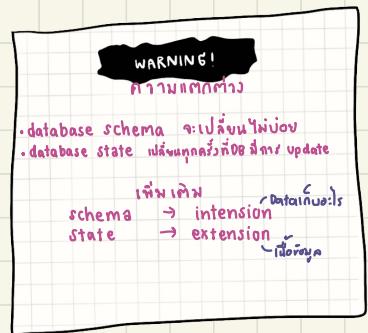


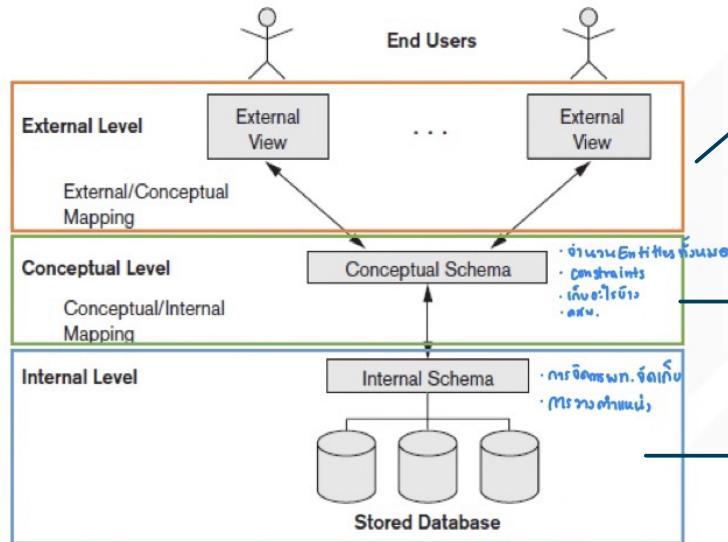
Figure 2.1  
Schema diagram for the  
database in Figure 1.2.

## Database State (database instance, occurrence, snapshot)

- ມີມົດຂອງເຖິງກົດໝາຍເກມນີ້ທີ່ພາກກົມຕື່ບັນໄດ້
- Database State : ມົດຂອງ ຂໍ້ເກມນີ້
- Initial Database state : ມົດຂອງຕົນກໍ່load ນາງໃນ system
- Valid State : Data ອີງ constraint, structure ສໍາເລັດ State ຖະໜຸດ data ນີ້ແຈ້ງວ່າຖືກ



## The Three-Schema Architecture



ດ້ວຍນາມ external schema  
ເຖິງ user view ຮັນນະນຳ: user  
views: ອຳນົດຕາມ ຕາມ ກົດລົງ  
user ມີ

logical level ໂອນໃຫ້ນີ້ຍຸດ  
ມີຄວາມອັນດີ ລົງ entities, data type  
relationship, user operation, constraint  
ດ້ວຍນາມ physical storage structure  
ການເປົ້າໂຄງນູ່ອັນດີ

## Data Independence

- logical Data Independence : ມີມົດຂຶ້ນ conceptual schema ລົງໄວ້ສ່ວນ external schema, program ຖື່ນວົງ
- Physical Data Independence : ມີມົດ Internal schema ລົງໄວ້ສ່ວນ conceptual schema

## User-Friendly DBMS Interfaces

- Menu-based (Web-based) : ສ່ວນສໍາເລັກ browser web
- Forms-based : ໃຫ້ naive user nonform
- Graphics-based : Point, click, Drag, Drop
- Natural language : ຖະໜຸດ English
- Combinations of the above : menu, form ຖະໜຸດ Web DB interface

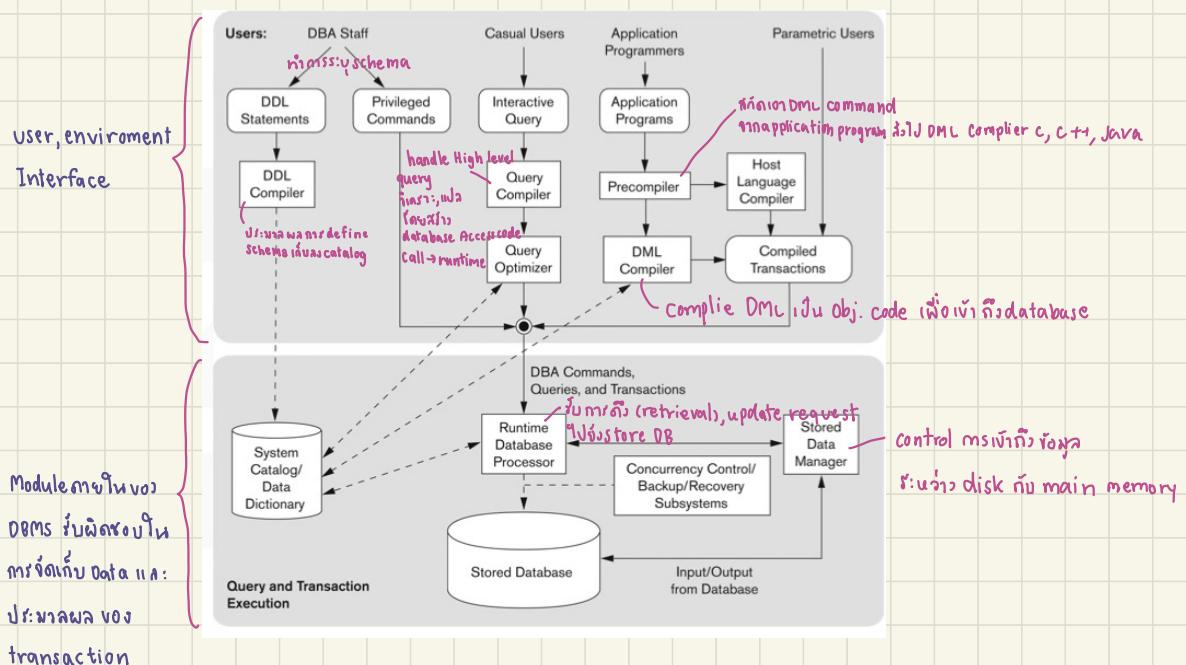
## Other DBMS Interfaces

- Natural language
  - free text as a query
- Speech
  - Input query and Output response
- Web Browser with keyword search
- Parametric interfaces, Interfaces for DBA:  
e.g., bank tellers using function keys.
- Interfaces for the DBA:  
(database administrator)
  - Creating user accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access paths

## Database System Utilities

- ↳ loading data, inserting Database
- ↳ file organization
- ↳ Back up data
- ↳ Performance monitoring utilities

## Typical DBMS component Modules



# Centralized DBMS

Program 11a: Database อยู่ในที่เดียวทั้งหมด User สามารถเข้าถึงต่อ ๆ กัน: remote terminal

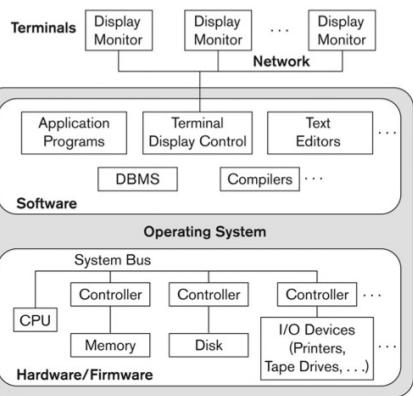


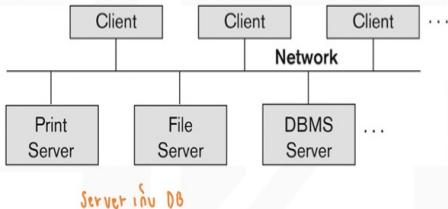
Figure 2.4  
A physical centralized architecture.

## Two-tier Client/Server

มี 2 ชั้นคือ Client / Server servers: ที่หนึ่งที่เก็บDatabase ไว้ ปะกับชั้น 2 ที่เป็น Client โดยที่ชั้น 2 ไม่ต้องมีการติดต่อเก็งจ์: มี install ให้ผู้ใช้ตั้งแต่

### Client

Figure 2.5  
Logical two-tier client/server architecture.

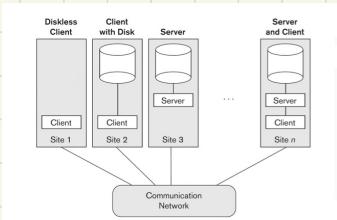


Server ที่อยู่ DB

! ต้องติดตั้งในชั้น 2: มี install ให้ผู้ใช้ตั้งแต่

### ► Specialized Server

- Print server, File server, DBMS server
- Web server, Email server



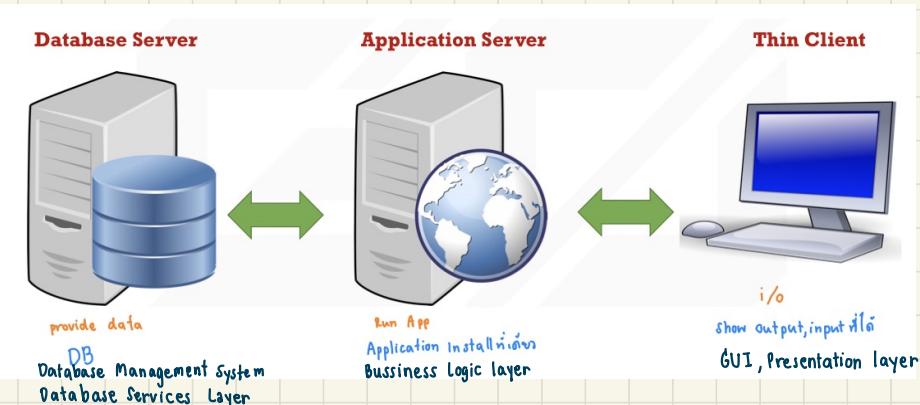
### ► Client

- ไม่ diskless, PCs, Workstations with disk
- connect Server LAN

### ► DBMS Server <SQL server, query server, transaction servers>

- ให้ Database query and transaction services
- ไฟล์ API ที่มี Access server database

# Three-tier Client/Server



## ↳ Three-tier Architecture

- Database server ต้องมี middle tier ให้ได้
- Client ที่สามารถเชื่อมต่อไปยัง Database server
- Client ด้วย UI, Web Browsers
- Client ผ่านทาง PC หรือ mobile device ผ่าน web

## Classification of DBMSs

- Based on the data model used ข้อมูลใน data Model ที่ใช้
  - Legacy: รุ่นเก่าๆ
    - Network, Hierarchical.
  - Currently Used:
    - Relational, Object-oriented, Object-relational
  - Recent Technologies:
    - Key-value storage systems, NOSQL systems: document based, column-based, graph-based and key-value based. Native XML DBMSs.
- Other classifications
  - Single-user (typically used with personal computers) vs. multi-user (most DBMSs).
  - Centralized (uses a single computer with one database) vs. distributed (multiple computers, multiple DBs)

34

## Cost considerations for DBMSs การพิจารณา ราคาสำหรับ DBMS

- Cost Range: ฟรี - ล้าน dollars
  - from free open-source systems to configurations costing millions of dollars
- Examples of free relational DBMSs:
  - MySQL, PostgreSQL, others
- Commercial DBMS offer additional specialized modules,
  - e.g. time-series module, spatial data module, document module, XML module
  - These offer additional specialized functionality when purchased separately
  - Sometimes called cartridges (e.g., in Oracle) or blades
- Different licensing options:
  - site license, maximum number of concurrent users (seat license), single user, etc.

# Ch. 3 Data Modeling Using the Entity-Relationship (ER) Model

## >Data Model

Set of concept (พื้นฐาน) structure, operations, constraints

## Entities

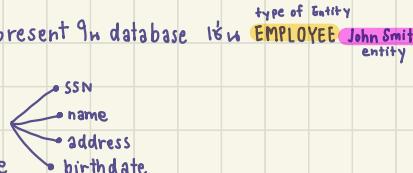
Entity (object law): ที่สามารถ represent ใน database เช่น EMPLOYEE John Smith entity

## Attribute

ต特徵ของ Entity Business

Entity: ของจริงๆ อยู่ในโลก

Simple  $\Rightarrow$  เช่น: entity มี single atomic value  
คือตัวเดียว



Composite  $\Rightarrow$  ประกอบด้วย Name (Firstname, MiddleName, LastName)

Multi-value  $\Rightarrow$  ประกอบด้วย color {red, blue, black}

## Store attributed

เก็บลงไดอะไบ

## Derived attributed

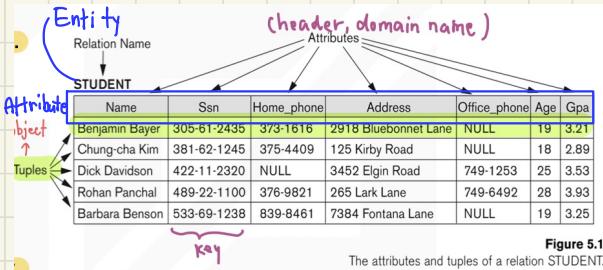
คำนวณโดยเก็บลง

ER model มี 3 concepts  $\Rightarrow$  Entities, Attributed, Relationship

# Ch.

## The Relational Data Model and Relational Database

- Relation  $\Rightarrow$  mathematical concept សម្រាប់នឹងនៅក្នុង the ideas of sets  
ការរក្សាទុកប្រព័ន្ធដែលមានវត្ថុ Table  $\Rightarrow$  Set of row
- Tuples  $\Rightarrow$  row និងជាការណ៍ក្នុង real world entity ឬ relationship, set of value
- Attribute name  $\Rightarrow$  column header, domain name បីន្យានការរក្សាទុកប្រព័ន្ធដែល data item នឹង column



- Key  $\Rightarrow$  uniquely identifies that row in the table
  - \* ការណ៍ seq number ឬ key ឬនឹង artificial key ឬ surrogate key (អាមេរិកសាគិក: អូតូឡូតី)
- Schema  $\Rightarrow$  description  $\Rightarrow R(A_1, A_2, \dots, A_n)$  attributes  
(name of the relation)
- domain  $\Rightarrow$  និងរាយការណ៍ set of atomic value
- Atomic  $\Rightarrow$  នឹង: value នឹង domain ឬនឹងលទ្ធផលឱ្យមួយ
- Relation state  $\Rightarrow$  subset of Cartesian product និង domain និង attributes  
និង: domain នឹង set of All possible values និង attribute ការរាយការណ៍
- $r(R)$   $\Rightarrow$  state, value, population : set of tuples  
 $r(R) = \{t_1, t_2, \dots, t_n\}$ ;  $t_i = \langle v_1, v_2, \dots, v_n \rangle$   
(រាយការណ៍ obj (tuples))
- SK  $\Rightarrow$  superkey ឬ key នឹង attribute និងកំពើ 1 ពីរខ្លះ {ID, Phone-num, age}
- Minimal key  $\Rightarrow$  និង Attribute ដែលត្រូវក្នុងក្រុងក្រោម tuples
  - \* key នឹង superkey ; key មិនជា superkey
- Primary Key (PK)  $\Rightarrow$  ក្នុងតាមលក្ខណនា Key នឹង reference tuple និង tuple នឹង និងត្រូវត្រូវ primary key  
(បានក្នុង)
- Database schema  $\Rightarrow$  set of schemas និង database ធំណ៍ង
- FK  $\Rightarrow$  foreign key និងតើសេចក្តីថ្លែងជា Table ដែល primary key នឹង table  
តើសេចក្តីថ្លែងជា  $t_1 \in R_1$  និង  $t_2 \in R_2 \Rightarrow t_1[FK] = t_2[PK]$

CAR	License_number	Engine_serial_number	Make	Model	Year
Mass ACB-729	45678-29		Audi	A6	02
Florida TYP-347	B43998		Oldsmobile	Cougar	05
New York MPO-22	X83554		Oldsmobile	Delta	01
California 452-TY	C43742		Mercedes	190-D	99
California RSK-629	Y82935		Toyota	Camry	04
Texas RSK-629	U028365		Jaguar	XJS	04

## Relational Database Schema

$\Rightarrow$  set  $S$  of relation schemas

ສະຖານະ database schema  $S = \{R_1, R_2, R_3, \dots, R_n\}$

ກ່ຽວຂ້ອງ relational schemas

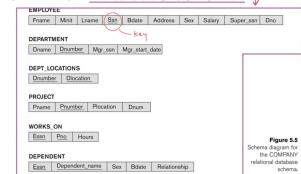
$R_i = (A_1, A_2, \dots, A_n)$

$t = (v_1, v_2, \dots, v_n)$

ກົດໃຫຍ້

ນັບໃຫຍ້

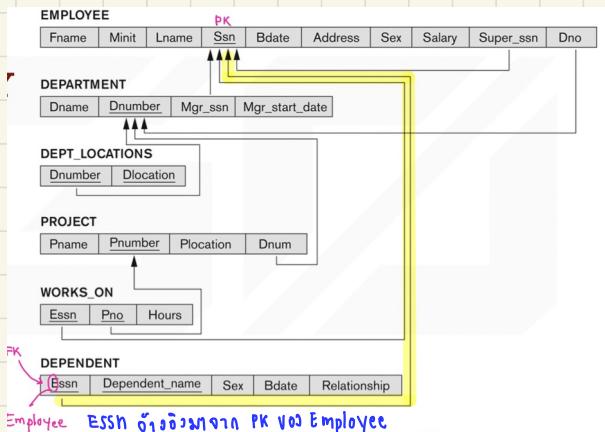
### COMPANY Database Schema



- Relational Database State  $\Rightarrow$  instance, snapshot ສະຖານະ instance ອົບອົບ row (tuple) ເກື່ອງ  
ບັນຍາກ່ອງກູນ DB ມີຄວາມຕະຫຼາດ valid state ສະຖານະກັນທາງການ update
- Populated database state  $\Rightarrow$  Union ມີຄວາມຕິດຕຸກ Individual relational state, database ດີວີຈຸດ  $\Rightarrow$  new state arises  
Basic operation  $\Rightarrow$  Insert, delete, Modify

### Entity Integrity

- ✓ primary key (PK) ມີຄວາມຕິດຕຸກ  $\Rightarrow$  ບໍ່ມີNULL ໃນກົດໃຫຍ້ tuple ມີ
- ✓ ມີFK (foreign key) ຂອງ Table : ຖະແຫຼງເຮັດວຽກ Table ມີຄວາມຕິດຕຸກ PK ນອກ Table  $t_1[PK] = t_2[PK]$
- ✓ Statement of the constraint von FK  $\Rightarrow$  ດ້ວຍມີຄວາມຕິດຕຸກ PK von Table ມີຄວາມຕິດຕຸກ, ບໍ່ມີກູນໃຫຍ້ NULL



### Other types of Constraints

- ✓ ຖະແຫຼງການກ່ຽວຂ້ອງລາຍລຳລັດ ຈົນໃນ Data Base ຢົດ (Semantic Integrity Constraints)
- ✓ ພົມມີກົດ script ມີຄວາມຕິດຕຸກຂອງມີ

- Update Operation on Relations  $\Rightarrow$  INSERT, DELETE, MODIFY
  - Updates  $\Rightarrow$  กรณี update ของปัจจุบันที่ถูกต้อง อย่างไรก็ได้
    - กรณี operation ใดๆ ที่ไม่ถูกต้อง [RESTRICT or REJECT]
    - DBMS จะลบข้อมูลที่ถูกต้องทั้งหมด [CASCADE] หรือ set เป็น NULL [SET NULL]
- Possible violations for each operation
  - INSERT  $\Rightarrow$  กรณี key ซ้ำ , กรณี Key ไม่ถูกต้อง , FK  $\neq$  PK ที่ต้องมี , PK เป็น NULL
  - DELETE  $\Rightarrow$  กรณีลบหัวน้ำที่กำลังมีผู้ใช้งานอยู่ อย่างต่อไปนี้
  - UPDATE  $\Rightarrow$  update PK  $\Rightarrow$  DELETE, INSERT กรณีไม่มีผู้ใช้งาน / กรณี nu domain constraint / NOT NULL constraint

# Ch. 5 Mapping ER To Relational

## Goal During Mapping

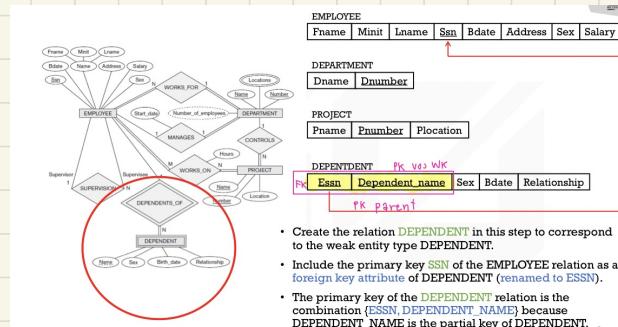
- ↳ Nonattribute
- ↳ Primary constraint ត្រូវការពារណ៍
- ↳ \$NULL Value ត្រូវត្រូវនៅទំនួរ

## Step 1: Mapping of Regular Entity Types

- ↳ សម្រាប់លើកស្ថិតិស្ថាន Entity សម្រាប់ Table ឬនានា Table
- ↳ តាម attribute ដែលជាការពារណ៍នឹង Primary Key នៃ Table

## Step 2: Mapping of Weak Entity Types

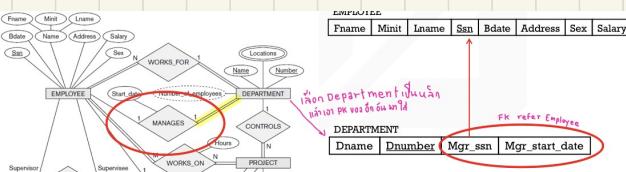
- ↳ តាមការពារណ៍ PK នៃ Parent នឹងធ្វើឱ្យ composite key នៃ weak entity, key នៃ parent



- Create the relation **DEPENDENT** in this step to correspond to the weak entity type **DEPENDENT**.
- Include the primary key **SSN** of the **EMPLOYEE** relation as a foreign key attribute of **DEPENDENT** (renamed to **ESSN**).
- The primary key of the **DEPENDENT** relation is the combination (**ESSN, DEPENDENT\_NAME**) because **DEPENDENT\_NAME** is the partial key of **DEPENDENT**.

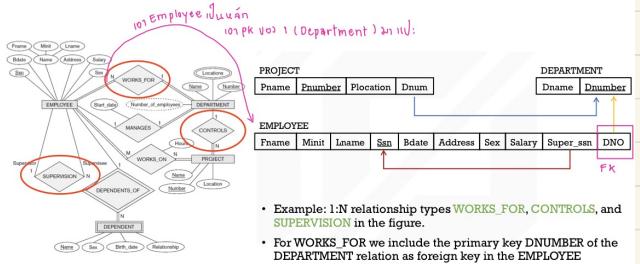
## Step 3: Mapping of Binary 1:1 Relation Types

- ↳ តាមការពារណ៍ total participation ឬបុគ្គលិក ក្នុង 2 ផែនីមួយៗ នឹងការពារណ៍លើកស្ថិតិស្ថាន
- ↳ តាមការពារណ៍ការពារណ៍ក្នុងក្នុង 2 Entities នឹង



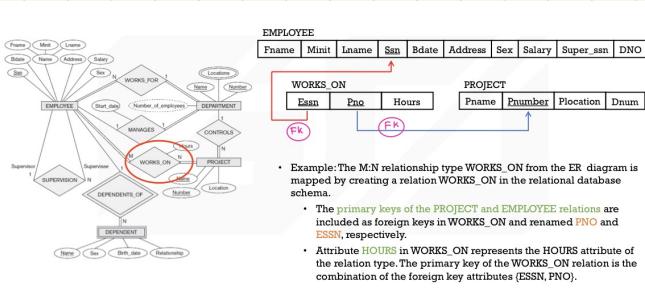
## Step 4 : Mapping of Binary 1:N Relation Types

↪ សម្រាប់ការបង្កើតលើ many នូវការបង្កើតនៃ 1 និង 1 នូវ FK



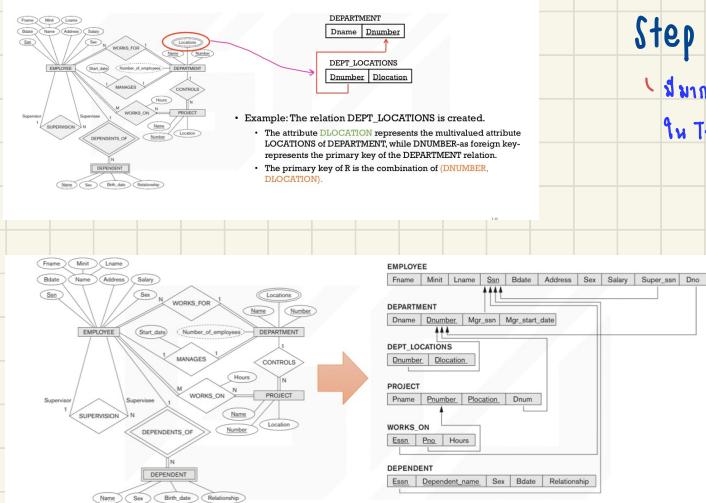
## Step 5 : Mapping of Binary M:N Relation Types

↪ ស្នើសុំការបង្កើតលើ many នូវការបង្កើតនៃ 2 Entity នូវ FK



## Step 6 : Mapping of Multivalued attributes

↪ នឹង Table នៃ 1 នូវ PK ឬ composite Key និង 1 នូវ multivalued attribute នូវ attribute



## Step 7 : Mapping of N-ary Relationship

↪ ស្នើសុំការបង្កើតលើ many នូវ PK នៃ n នូវ Entity និង 1 នូវ Table នៃ 1 នូវ composite Key

# Ch. 8 Basic of FD and Normalization

## Information Design Guidelines for Relational Database

- ▷ 2 ระดับของ relation schemas
  - logical "user view" level
  - conceptual "base relation" level
- ▷ Table ที่มีความซ้ำซ้อนอยู่ในส่วนของตัวแปรเดียวกันจะมี redundancy ที่ดู
- ▷ ข้อจำกัด: ไม่ให้ relation นี้เป็นของ Entity ใด Entity นั้น หรือ ไม่เป็นตัวของ Entity ใดๆ
- ▷ กรณีใช้การอ้างอิงกันผ่านตัวแปร (FK)
- ▷ Entity, Relationship อาจมีส่วนกัน เช่น 1:n หรือ 1:1
- ▷ บันทึกต่อไปนี้เป็น



## Redundant Information in tuples and Update Anomalies

- ▷ Update ตัวเลขที่ Jnr: นำเข้าบันทึก
- ▷ Insert ฝ่าฝืน ตรวจสอบตัวเลขที่มีค่าต้องมี 2 key พร้อมกัน จึง Insert ได้
- ▷ Delete ตัวเลขที่ต้องการลบต้องมีบันทึกทั้งหมด
- ※ จึงได้เกิด 3 ข้อด้านบนนี้

## NULL Values in Tuples

- ▷ Null คืออะไร: อยู่ที่ไหนใน Relation ตารางของ Table vor attribute ที่เป็น Multivalue มีค่าเป็น Null

## Spurious Tuples

- ▷ ในการการjoin Table ใหม่ที่มี Join กัน ข้อมูลที่ไม่ถูกเลือกมา อาจทำให้เสียหาย หรือเสียหาย less
- ▷ ตัวอย่าง: สำหรับ 2 ตาราง A และ B ที่มี functional dependency ของ A ที่ B แต่ A ไม่ได้ถูกเลือกมา

↑ กรณี functional dependencies

## Functional Dependencies

- ▷ define normal form ของความซ้ำซ้อนของ attribute ใน Relation, attribute ต้องดูแลโดยตัวตัวเอง ไม่ต้องดูแลตัวอื่นๆ
- ▷  $X \rightarrow Y$  : X determines Y / Y depends on X
- ▷ FD ที่มีกุญแจ key ที่ต้อง FD define ให้ attribute เหล่านี้เป็น key สามารถ define attribute ที่ไม่เป็น key ให้
- ▷ ชนิดของ FD
  - Fully Dependency  $\Rightarrow$  ต้อง pk ของ key กำหนดตัวของกุญแจ non-key attribute ที่จะแนบทิ้งตัวอื่นๆ
  - Partial Dependency  $\Rightarrow$  non-key บางตัวขึ้นอยู่กับ key [composite key] หรือตัวอื่นๆ
  - Transitive Dependency  $\Rightarrow$  non key ตัวหนึ่งขึ้นอยู่กับตัวอื่นๆ ที่ไม่ใช่ key ตัวอื่นๆ

# Normalization of Relations

- Normalization : วิธี relation ที่ไม่ใช่เป็น Table จึงจะ จงหา / Join น้ำหนักสุดห้องไฟ Table เดียว, แก้ปัญหา Anomaly
- Normal form : กี่ key, FD check relation ที่ไม่ใช่ normal form  $\Rightarrow$  2NF, 3NF, BCNF
- Prime attribute  $\Rightarrow$  สามารถใช้ห้องร่างให้ candidate key
- Non prime attribute  $\Rightarrow$  ห้องร่างให้ candidate key

## 1NF

- กำจัด repeat Group
- composite attribute
- nested relation : tuple ที่มี non-atomic

(a) DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	Bellaire, Sugarland, Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

(b) DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	Bellaire, Sugarland, Houston
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

(c) DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Multivalue  
composite

## 2NF

- ห้าม non Prime attribute  $\Rightarrow$  ห้องร่าง pk
- กำจัด partial dependency ของ非主屬性ต่อ pk  
เพื่อให้มี fully dependency : pk ห้องร่าง ห้าม non attribute ที่ไม่ต่อ pk ห้องร่าง

(a) EMP\_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
FD1					
FD2					
FD3					

Prime                          Non-prime

(a) LOTS

Property_id#	County_name	Lot#	Area	Price	Tax_rate
FD1					
FD2					
FD3					
FD4					

Candidate Key                  non-prime

ห้าม partial ให้ 3 Tables

Ssn, Pnumber  $\rightarrow$  Hours

Ssn  $\rightarrow$  Ename

Pnumber  $\rightarrow$  Pname, Plocation

ห้าม Table

Property\_id#  $\rightarrow$  county\_name, Lot#, Area, Price

Country name  $\rightarrow$  tax\_rate

## 3NF

กำจัด ก็ 3 ก้ามห้องห้องกันนี้จะ เป็นการเพิ่มความซับซ้อน

ห้าม Transitive Dependency

(a) EMP\_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
FD1						
FD2						
FD3						
FD4						

nonprime  $\rightarrow$  nonprime  
1NF Normalization

Ssn  $\rightarrow$  Ename, Bdate, address, Dnumber

Dnumber  $\rightarrow$  Dname, Dmgr\_ssn

(a) LOTS1

Property_id#	County_name	Lot#	Area	Price
FD1				
FD2				
FD4				

non-prime  $\rightarrow$  non-prime

ห้าม Table

Property\_id#  $\rightarrow$  County\_name, county\_name, lot#, area

Area  $\rightarrow$  price

## BCNF

សំគាល់នៃ Attribute រួចតារាត្រូវ key នឹង K និងមែន super key នៃ R ទេ