

File-based System

ระบบไฟล์มีชื่อว่า ไมโครซอฟต์ เอ็กซ์เพรส ขายเครื่องคอมพิวเตอร์ → Database System

Problem of File Based System

- ข้อจำกัดคือการแยก Separation and Isolation data

เพื่อให้ผู้ใช้งานสามารถเข้าถึงข้อมูลได้โดยตรง

- Duplication of Data 造成冗余数据

→ Data inconsistency ข้อมูลไม่สอดคล้องกัน

ex. บุคคล นำเข้าไปขายของในห้อง GPA
ห้อง GPA ไม่ได้รับการจัดการ

→ Data anomalies ข้อผิดพลาด

- Modification anomalies : แก้ไขข้อมูลไม่ถูกต้อง
- Insertion
- Deletion

- Incompatible file formats ไม่สามารถใช้งาน

ต้องแปลงรูปแบบไฟล์ให้เข้ากัน ก่อนจะสามารถใช้งาน

- Fixed Queries/Proliferation of Application Programs

Programmer ต้องเขียนโปรแกรมตรวจสอบ

รองรับทุกกรณี

Data Base

- End user data (Raw facts) ; ข้อมูลที่บันทึกไว้
 - Meta Data (Data about data) ; รายละเอียด เช่น รูปแบบ นามสกุล
- Type Data → Non-numeric Data type : ตัวอักษร ลักษณะ string char
 ID, Name, Professor
 → Numeric Data type : ตัวเลข int, double, float
 GPA

Why Store Data IN Database?

- + Efficiency
- + Convenience
- + Reliability
- + Security

Data Base Characteristics

- Persistent data store in Database

ไม่สามารถลบออก ex. Disk

อาจมี Disk Failure

- Shared can Multiclient ผู้ใช้งาน

สามารถเข้าถึงข้อมูลเดียวกัน

ให้ผู้ใช้งานร่วมกัน

- Interrelated data relate ที่มีความสัมภาระ

ทำให้ใช้efficiency

Identity: student, Professor, Course

Relate: สารสนเทศที่เกี่ยวข้องกันในทางเดียว

เช่น อาจารย์สอนนักเรียน

INTRODUCTION TO DATA BASE SYSTEM

CHAPTER 1



- Client Server Teleprocessing & File Server

- Client Server
 - 2 Tier
 - 3 Tier
 - Database Server
- Increased for Forwarded Requests
- Reduced Hard ware cost
- Reduce Communication Cost
- Increase Consistency

- Client user
- 2: Application Server Logic
- 3: Data Base Server

Advantage of Data System

+ Program Data Dependency

- Unintended dependency between data and program

- Inconsistency Data Structure, Characteristics

between program

Program → Data Dependent, Inconsistent

+ Minimal Data Redundancy

among various data

+ Improved Data Consistency

ข้อมูลเดียวกันในทุกๆ สถานที่ที่มี

DBMS

Data Base Management System

Software for manage Data Base System

DBMS คืออะไร สำคัญอย่างไร ประโยชน์

→ Data Base Definition สร้าง Data Base ด้วย Entity + Relationship

→ Data base Access Store, Update, Query

→ Transaction Processing จัดการ transaction

↓ ลักษณะที่มีใน transaction

↓ ลักษณะของ transaction

↓ Transaction คือสิ่งที่ต้องดำเนินการตามขั้นตอน

Database Architecture

Abstract View คือวิธีที่ Store, Manage

= easy to use system and efficiency =

ex. Student Data Abstract View (User view)

- High Level → studentID, Name, Major
 ex. 00010824, Kavikara-Prathum, ComSci

- Intermediate Level → structure

ID → char(3)

Firstname → varchar(20)

Lastname → varchar(20)

Major → varchar(20)

- Low Level → ข้อมูลระดับ Bit, Byte, Block

THREE LEVEL ARCHITECTURE ANSI

External Level (application)

Intermediate Level (Abstract View)

Data from databases

ex. 00010824, Kavikara-Prathum, ComSci

Conceptual Level (User view)

conceptual schema show in Data Model

ex. เอกสารที่เขียนใน กระดาษ hardware

Internal Level (Physical Level)

จัดเก็บใน กระดาษ หรือแม้แต่

File organization ในระบบ operating system

⇒ Fast & efficient to access data

ex. MySQL (Physical Level)



logical



Data Independence

ไม่ต้องเปลี่ยน data level แล้ว ยังคง run ที่เดิม

• Logical

• Physical



Multi-User DBMS Architecture

• Tele processing ดำเนินการทางไกล



Client Server Work & Work Station

Client Server LAN Local Area Network

- Large amount of network traffic

- High cost server

- Complexity because concurrency control



Evolution OF Database

Gen 1	1960s	File	File Structure
Gen 2	1970s	Network Navigation	Network and hierarchies of related records
Gen 3	1980s	Relational	Non-procedural languages
Gen 4	1990s	Object	Multi-media, distributed processing

2000 and Beyond

2000 & Beyond

• relational database ฐานข้อมูลที่สามารถเข้าถึงฟรี

• Not Only SQL - MongoDB, Apache Cassandra, Hadoop



File Based System

↓
ກາງຈະບໍາໄວ໌
ປີລັບປິດເປົ້າບຸນຫຼາຍໆຂອງສູລະເການ



Data Base System *

ຳ ACID Atomicity Consistency Isolation Durability

ເພື່ອຄວບຄຸມ Data Base ອັນ Transaction ມີ User

Atomicity: ຂອງບັນດາມີຄວາມຕິດພາດ Transaction ສີໂລດືບ → completed
→ Cancelled All

Consistency: Data ທີ່ເປັນຂອດ Data Base ທີ່ສູງຄູກເຫັນອຽນກົນເປັນໃນການເງື່ອນໄໝ
ທີ່ຖືກຕໍ່າງໆໄດ້

Isolation: ທັນສົກມີຄວາມຕິດພາດ 1 Transaction ຮະຫວັງໄນ້ການກາບ Transaction ທີ່ໆ

Durability: Transaction ສີໂລດືບ ຕົວດູກເຫັນທັກລົງ Data Base ກົບເຊົ້າ
ເປົ້າຮ່ວມມືນຂຶ້ນມູນລົດທີ່ໄຟ້ຢູ່ ex. ປັບກັບລົງ

ໃນ Hard disk

DataBase = Schema + Instance

ID	Name	Lastname	Address	Major	Schema
650710524	Kanisara	Prasitnui	Songkhla	CS	
650710077	Natthaphon	Ruengwarapit	Bangkok	CS	Instance

Data Model : Data + ຄວາມສ້າງໃໝ່ + ຄວາມຂອງຍຸ + ເງື່ອນໄໄລ & ດັບຕິປະກອບ

Relation Model: show tables & column

Object-Based Model: ພິເສດວິຊາກາ Relation "OOP" ex. Entity-Relationship Model or ER Model

Semi-Structured Model: XML (Extensible Markup Language) ດາວໂຫຼວດໂຫຼວດ ບໍ່ມີຕົວແທນ
<body></body> ...

SQL ສັບກັນທີ່ໂຄງສະຕິງ ໃຫ້ສົດຂອງອັນດີເກີນ

+ Data Definition Language "DDL": schema w; DataBase, ms w; Table ex. create, delete, alter...

+ Data Manipulation Language "DML": form data in DB ex. insert, delete, update

+ Data Query Language "DQL": find data, query data ex. select, count, sum, min, max, average...

+ Data Control Language "DCL": control user right DB ex. GRANT, REVOKE or transaction ex. commit, rollback

Ex. SQL select name, age, email from TB_member where age <

Will you miss me when
I'm gone?



DATA BASE SYSTEM ARCHITECTURE

แบบจำลอง 3 Tier

• Presentation : ลักษณะเชื่อมต่อผู้ใช้ ติดต่อกับ User

UI: user Interface Text mode

Graphic User Interface

• Business Logic : main process user-data-data base

ต้องร่วมกันให้เข้ากันได้ในการทำงาน

soft ware by Programmer

DBMS ระบบจัดการฐานข้อมูล

• Database : ฐานข้อมูล, data ที่ต้องเก็บใน DB

เพื่อ Data Independence

ข้อมูลเชื่อมต่อที่เก็บไว้จะถูกแปลงให้ตรงกับชื่อในระบบ
กระบวนการนั้นๆ

1- Physical โครงสร้างทางกายภาพ schema

ไม่ระบุรายละเอียดของ hardware

ex. หน่วยความจำ Hard disk ของ Server

ไม่ระบุ database อยู่ใน Hard disk

2- Logical ผลลัพธ์ schema ที่ทางเรา

หันมาเก็บ schema ในรูป view

ex. หน้า เว็บ ตารางใน DB ไม่ระบุ hardware

soft ระบุมา หรือลักษณะต้องการกับ User

Architecture of Data Base

- หมายความ การประมวลผลของ Data Base System ที่process ต้องดำเนิน

- Centralize “ศูนย์รวม” ประมวลระบบฐานข้อมูลไว้ที่ Server ก็ได้เช่น

- การประมวลผลเบิกบาน Server, Client

- เป็นยุคสมัยดีปัจจุบันมีความหลากหลายมาก

1 Tier or Centralize

Presentation + Business Logic + Database

อยู่ใน Computer 1 เท่านั้น User ต้องต่อ Server

แล้วรับการประมวลผล

2 Tier or Client-Server

Presentation + Business Logic ; Client

Data Base : Server

ข้อดี = Data base ให้บริการ ผ่าน Client ให้

ทำงานด้วยตัวเอง ให้รวดเร็วและถูกต้อง

Data base ใหญ่

ข้อเสีย = Client ต้องประมวลผลเอง

LAN ต้องเสียบสาย และคอมพิวเตอร์ต้องสูง

Server ต้องเป็นกลาง เชื่อมต่อ Client

ที่ใช้งานร่วมกันเป็น对 → ล้ม, รอดูอีก

3 Tier with Data Independence

Presentation - Client

Business logic - Application Server

“Middle Tier” ต้องต่อ software ประยุกต์ for ประมวลผล

Database - Database Server

ต้องต่อ DBMS data base management system

ข้อดี = ประมวลผลที่พิเศษ ของ Client ต้องต่อ กับ User ได้รวดเร็ว

. Middle Tier หรือ, คอมพิวเตอร์ต้องต่อ:

· ต้องมีความสามารถดูแลภาษาของข้อมูล Data Integrity

ข้อเสีย = . ฝึกอบรมซับซ้อน complex ขนาดการติดต่อต้องมีสาระ

มากกว่า 2-Tier

n Tier or Multitier

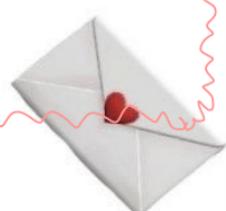
เพิ่มจำนวนห้อง 3 tier ไม่ยังไงที่ต้องทำ

- ไม่ต้อง Business logic & Data Base ให้แยกอย่างเดียว

ให้ต้องมีการประมวลผล

- แยกการสืบค้นข้อมูลใน Database ด้วยคำสั่ง SQL

และการ Stored Procedure ให้คำสั่งที่ใช้ป้อน



♥ Relational Data Model ♥

Relation → Table

branchNo (1)	street (2)	City (3)	Postcode (4)
3003	143 ถนนสีลม	กรุงเทพฯ	10110

+ Tuple = row

+ Degree = column Attribute = 4

+ Cardinality = Primary tuple = 1

Information Table 1 tuple row/column en Schema 400 Data

Ex

Degree = Attribute column

name	major	salary
Phon	CS	50,000
Cin	CS	55,000
Meen	CS	53,000

Table ชื่อของพนักงาน

Table ชื่อของพนักงาน Schema → name, major, salary

If Degree = 3 , ที่ Cardinality = 3

Instance of 3 tuple
- Phon major CS salary 50000
- Cin major CS salary 55000
- Meen major CS salary 53000

Tuple = Attribute คือค่าของแต่ละคอลัมน์

I. Domain: Attribute ต้องมีค่าอยู่ในช่วงจำกัด
ex. Attribute เวลาเดือน ที่ต้องมีค่าอยู่ในช่วงเดือน
ไม่เกิน 12 เดือน

II. Atomic: Attribute ไม่สามารถ拆分成สองค่า
ไม่สามารถต่อไปได้
ex. phone-number & postnumber

III. Unorder: ลำดับ Tuple ไม่สำคัญ Data in Table
แล้วก็จะไม่เป็นปัญหา

IV. Null: ไม่สามารถเป็น Null ใน Table ใดๆ

Data Type

Numeric: จำนวนเต็ม

Character: String: อักษร

Bit-String: Bit หรือ computer Binary

Boolean: TRUE / FALSE / UNKNOWN

Date: DATE / DAY / MONTH / YEAR

HOUR / MINUTE / SECOND

YY-MM-DD , HH-MM-SS

เช่น 2019-09-30, ...

Relation → relation(table) ต้องมีข้อจำกัด

- Attribute ต้องมีค่าอยู่ในช่วงจำกัด

- ต้องมีค่าอยู่ในช่วงจำกัด

- ต้องเป็น cell ของเพียง atomic ที่อยู่ในช่วงจำกัด

- Tuple ต้อง Relation ต้องมีค่าอยู่ในช่วงจำกัด

- ต้องไม่ให้ attribute ซ้ำอยู่ใน format เดียวกัน

KEY : คือค่าของ attribute ที่ไม่ซ้ำกันใน tuple ทุกตัว
key หมายความว่า stdID ที่มีค่า unique คือ stdName, value

stdID	stdName
710077	Mr. Phon
710524	Ms. Cin
710572	Ms. Meen

* Super Key: Uniquely identifies

ทุกๆ attribute ใน Relation

- stdID

- stdID, stdName

- stdID, stdMajor

- stdID, stdName, stdMajor

* Candidate Key: super key ที่ไม่ใช่ primary key หรือ attribute ทางเดียวที่ไม่ใช่ primary key

Tuple ที่มีค่า

STUDENT

stdID	stdName	stdMajor	stdSSN
5907010	phon	computerEng	1234567890123
5907012	cin	computerEng	1234567890124
5907018	meen	computerEng	1234567890125

Attribute → stdID, stdName, stdMajor, stdSSN

Tuple ที่มีค่า (stdID, stdName) → stdID, stdSSN

* Primary Key: คือ key ที่ต้องไม่ซ้ำกันใน Relation (Table)
重要な candidate key

stdID → Primary Key หรือ stdID

stdSSN → Primary Key หรือ stdSSN

* Alternate Key: candidate key ที่ต้องไม่ซ้ำกัน หรือ stdSSN

candidate key → stdID, stdSSN

Primary key → stdID ... Alternate key → stdSSN

Primary key → stdSSN ... Alternate key → stdID

* foreign Key: key หนึ่งใน relation Attribute ที่ต้องมีค่า

ใน key ค่าใน Table ใดๆ

STUDENT

PK	stdID	stdName	genderCode
	077	phon	1
	524	cin	2
	572	meen	2

GENDER

PK	genderCode	gender
	1	Male
	2	Female

Foreign Key คือ STUDENT ห้อง gender Code

Integrity Rules ; ถ้า key ใดๆ ก็ตาม ที่ไม่ซ้ำกันใน Table (Relation) ต้อง key

* Entity Integrity : Primary Key ต้อง Table ต้อง unique (ไม่ซ้ำ)

- ห้าม NULL ต้องห้ามเป็น null

- Relation need Entity Integrity ไม่ต้องห้าม null

- หมายความว่า Relation ต้อง

MAJOR	majorCode	majorName
	077	computerEng
	524	computerSci
	572	mathSci

MINOR	minorCode	minorName
	077	mathSci
	524	mathSci
	572	mathSci

Major key + minor code ต้องเป็น实体

Major key + minor code ต้องเป็น entity

* referential Integrity : Foreign key ต้อง relation 2 ต้องมี primary key ของ Relation 2

ต้องไม่ null



♥ Relational Data Model ♥ (mid)

Delete & Update Action for Referenced Rows

- delete / edit data in Relation_1, និងការលើកការពី

Relation_2 ដែលបានស្ថិតនៅ Value

ឬ foreign key

- ត្រូវជាលើក / edit because??

4. **Restrict:** និយាយនៃ value នៃទុក

Primary key នៃ table_1 គឺត្រូវ

នៅ table_2

PKID	attribute	valueCode
500010	university	1
500012	university	2
500018	university	3

នៅលើកការពី "value"

→ **Cascade:** បញ្ចូលការពី បានប្រើបាន Primary key នៃ table_1 នៅលើកការពី foreign key នៃ table_2

→ **Nullify:** និយាយ foreign key នៃ relation_2 ត្រូវ NULL នៅតុកការពី primary key នៃ relation_1 ត្រូវ

→ **Default:** del / edit និយាយ primary key នៃ table_1 នៅលើកការពី foreign key នៃ table_2

NOTE:

Relational Data Model និងការរួមចំណាំ

entityTable

Attribute "Column" Tuple "row"

- At Domain (គុណភាព) និងការប្រើបានក្នុងទឹក Data
- Atomic និងការពី attribute និងពីរ
- Unorder និងការប្រើបានក្នុងការតាមតាមលេខ
- Null និងការប្រើបានក្នុង table

Data Type និងការរាយការណ៍ Data

Primary Key for attribute និងការរាយការណ៍, តាមការប្រើបាន row (tuple) និង

- Super Key តើ column / ឱ្យ column សារុបតែងត្រូវ unique និង row (tuple)
- Candidate key តើរាយការណ៍ column តុកការពី
- Primary key និងការណ៍ Candidate Key តុកការពី null
- Foreign key តើរាយការណ៍ primary key នៃទឹក, table

Relational Algebra

1. Selection "σ"

$\sigma_{\text{salary} < 35000}$ (Teacher)

select salary from Teacher where salary < 35000;

2. Projection "Π" និង Π : column

$\Pi_{\text{name}, \text{price}}$ (Product)

select name, price from Product;

3. Union "∪"

$\Pi_{\text{id}, \text{name}}(\text{Depositor}) \cup \Pi_{\text{id}, \text{name}}(\text{Borrower})$

4. Set Difference "Π - Π"

$\Pi_{\text{name}}(\text{Depositor}) - \Pi_{\text{name}}(\text{Borrower})$

5. Intersection "Π ∩ "

$\Pi_{\text{name}}(\text{Depositor}) \cap \Pi_{\text{name}}(\text{Borrower})$

6. Cartesian Product "×" cross

Depositor × Borrower

Join ; $\Pi_{\text{d.name}} = \Pi_{\text{b.name}}$ (Depositor X Borrowers)

7. Natural Join "× oo" ឬនៅក្នុង relation

និយាយនៃ foreign key នៃ T_1 និង Primary Key

នៃ T_2

Product on Category

use $\Pi_{\text{p.ID}, \text{p.name}, \text{p.price}, \text{p.cat_id}, \text{cat.name}}$ ($\Pi_{\text{p.cat_id}} = \Pi_{\text{cat.ID}}$)



Entity-Relationship Model

ER DIAGRAM

Data Base Design

+ Conceptual (โครงสร้าง) → ER Diagram

+ Logical (โครงสร้าง)

+ Physical (รูปแบบ)

Logical คือโครงสร้างที่ใช้ใน DBMS
物理的構造: ความต้องการของผู้ใช้งาน

+ Entity set

• Strong (Regular Entity)
ไม่ต้องมี PK

• Weak Entity
ต้องมี PK ใน Entity ด้วย

• Associative/composite
คู่ๆ Many to Many

+ Attribute

• simple

• composite

• Single value

• Multi Value

• Derived ได้มาจากการคำนวณ Attribute ที่มีอยู่

• Key ห้ามต้องเป็น PK/FK
not null!

Redundant Attribute บันทึกซ้ำ

+ Relationship

NOTE:

Entity - Relationship Model

“ER MODEL”

แบบจำลองความสัมพันธ์ที่สำคัญ
ให้สัมผัสด้วย ER-D

2 แบบที่มี

- Chen: Entity flow Chart

- Crow's foot: ผู้เชี่ยวชาญ

Entity : ตัวเรื่องราว
Attribute : ข้อมูลของ entity
Relationship

* Entity Type

• Strong Entity

• Weak Entity

* Attribute Type

• Simple (single component) ไม่มีอย่างใดอย่างหนึ่ง

• Composite ถ้า Attribute สองตัว ex. name < first last

• Single value ex. สีเสื้อ: แดง ขาว

• Multivalue ex. นามสกุล

• Derived หลักๆ คือ Attribute ใหม่ ex. Age วันเดือนปีเกิด

ER DIAGRAM

Peter Chen กลางๆ Chen → Entity ตัวเรื่อง, Relationship คำว่า

= Entity (Strong/Regular)

= Attribute

= 1:1 and

= Weak Entity

= Key Attribute

= 1:N (Total)
(ทั้งหมด)

= Associative Entity
Composite

= Multi Value Attribute

= 1:N And

= Relationship

= Component Attribute

= 1:N (Partial)
บางส่วน

= Identifying Relationship

= Derived Attribute

Crow's Foot

Entity
Attribute
Attribute
.
.

Entity
key
:
:
:

Entity		
Attribute		
Type		
:	:	:
:	:	:
:	:	:

= one

entity คือ noun

= Many

attribute →

= Zero or One

relationship คำว่า

= One and Only One

= Zero or More

= One or More

= zero or One

= zero or More

mandatory (มีอยู่ใน Entity ต้องมีใน Entity ด้วย)

= One and Only One

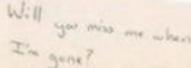
= One and More

optional (ไม่มีอยู่ใน Entity ต้องมีใน Entity ด้วย)

= zero or One

= zero or More

Will you miss me when
I'm gone?



I'm gone?



I'm gone?

