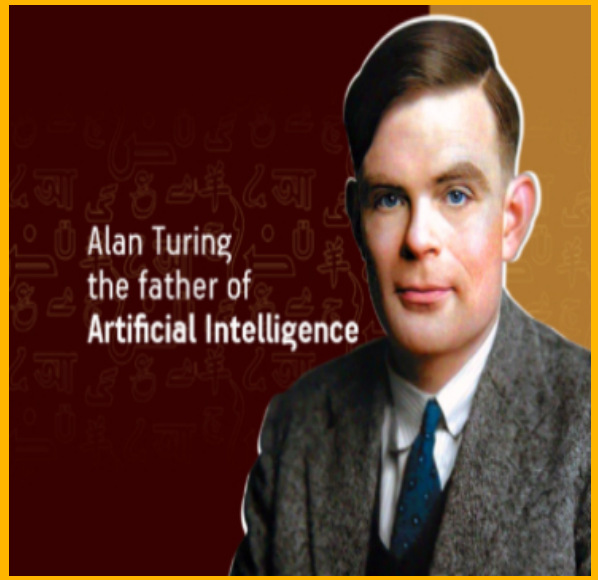


Introduction to Computer Science Fall 2022 #11

Chi-Jen Wu

Database Systems

- The history of database systems
- Database fundamentals
- Traditional File Structures
- Maintaining Database Integrity
- The Relational Model
- Object-Oriented Databases
- Data Mining

儲存和連結
資料和
資料之間的關係

A database is a collection of related, logically coherent, data used by the application programs in an organization.

The advantages for a database system

- Advantages of databases
  - **Less redundancy**
    - A lot of redundancy in files
  - **Inconsistency avoidance**
    - Any changes in the data need to occur in all places

教授A	課程1
教授A	課程2
教授B	課程3
教授B	課程4
教授C	課程5
教授C	課程6

Database fundamentals

- A collection of data
  - **Multidimensional**
  - Internal links (**relationship**)
    - make the information accessible from a variety of **perspectives**
- **File-oriented database system**
  - Traditional File Structures
- **Database system**

Traditional File Structures

- **Sequential file**
  - A file whose contents can only be **read in order**
- Reader must be able to detect end-of-file (**EOF**)
- Data can be stored in **logical records**, sorted by a **key field**
  - Greatly **increases** the speed of batch updates

P檔

1	教授A	皮卡丘
2	教授B	雷丘
3	教授C	妙蛙

C檔

1	課程1	微積分
2	課程2	計算機概論
3	課程3	人工智慧概論
4	課程4	基礎英文
5	課程5	普通物理學
6	課程6	體育大一

S檔

1	學生1	小智
2	學生2	大智
3	學生3	大木
4	學生4	大大
5	學生5	小弱
6	學生6	大弱

6 P.3 C6

1	P.1	C.1
2	P.1	C.2
3	P.1	C.3
4	P.2	C.4
5	P.2.	C.5

教授教學檔

1	C.1	S.1
2	C.1	S.2
3	C.1	S.3
4	C.1	S.4
5	C.1.	S.5
6	C.1.	S.6

學生修課檔

7 C.2 S.6

新增一個老師課程學生關係

CRUD 操作:

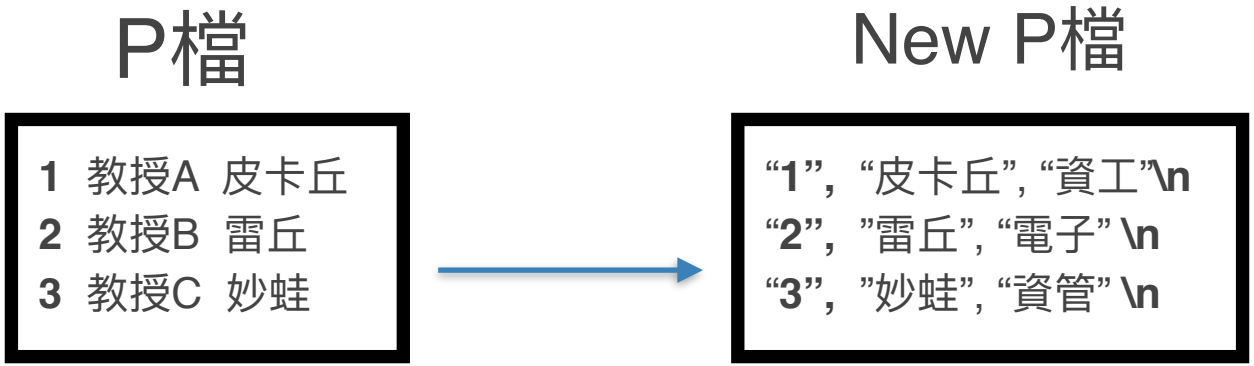
Create

Read

Update

Delete

Text based logical records



我們可以**定義**一個老師應該有什麼**屬性**

在P檔裡 目前一個教師會有

屬性

1. id (獨一無二的編號)
2. 姓名
3. 系所

Logical record

電腦科學中的 index

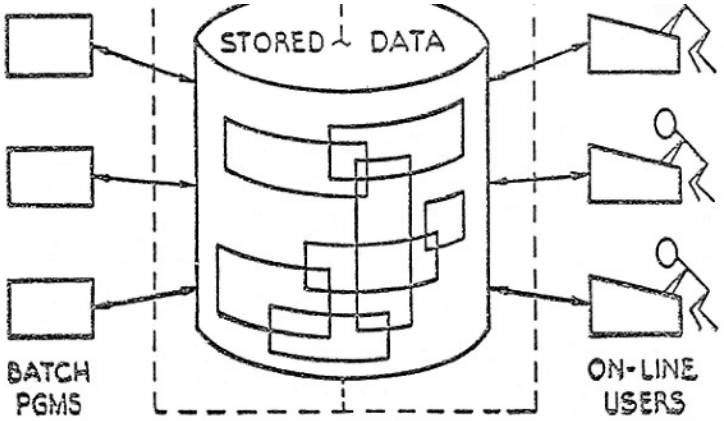
- Index 技術在電腦科學中非常重要
- 應用範圍非常廣
  - 你每天用的google 也是靠特別的 index 技術
- 大概也是整個電腦科學都會使用的技術
- 先記住他是一種有**排序的資料結構**
  - **利用索引鍵值來指向一個資料區塊**

The Role of Schemas

- **Schema**
  - A description of the **structure** of an database
    - 表格結構，欄位格式，資料關聯
- **Subschema**
  - A description of the portion of the database to a particular user's needs
    - to prevent **sensitive** data from being accessed by **unauthorized personnel**

Topics

- An Introduction to Computer Science
- The Shapes of Computers Today
- Computer Organization and Architecture
- Operating system
- Networking & The Internet
- **Database Systems**
- Software Engineering
- Cloud Platform/Cloud Shell Editor
- Cloud Platform/Cloud Source Repositories (git)
- C/C++ Programming



Database Systems

- The history of database systems
- Database fundamentals
- Traditional File Structures
- Maintaining Database Integrity
- The Relational Model
- Object-Oriented Databases
- Data Mining

儲存和連結
資料和
資料之間的關係

A database is a collection of related, logically coherent, data used by the application programs in an organization.

File-oriented 問題

- 單人在CRUD情況下都算是正常
  - 在**大部分情況**下是正常
- 所以現在要討論在**多人**情況下的情況
  - **Multiprogramming**
    - 還記得嗎？又要**分享**
    - Execute multiple processes concurrently
  - **Multi-user / Multi-host**

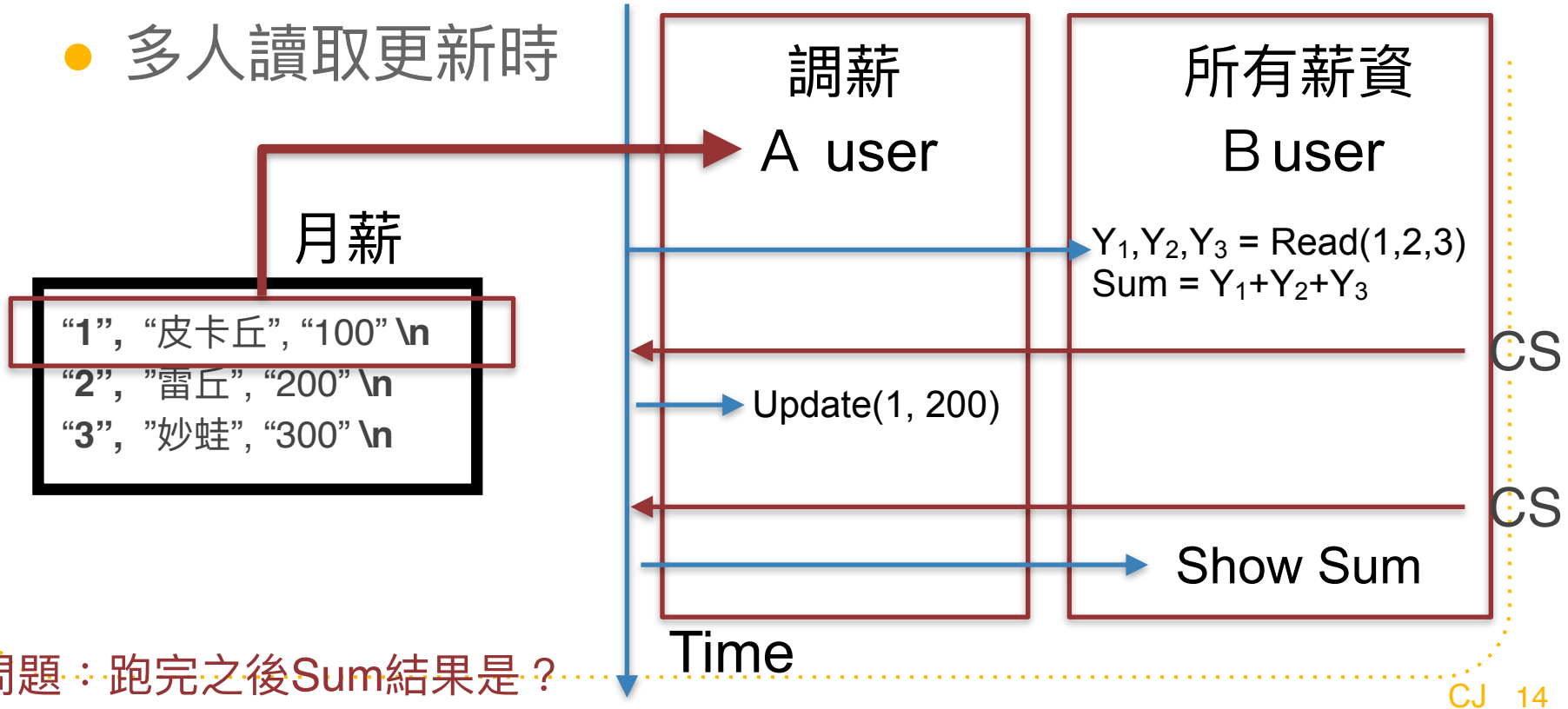
CRUD 操作:
Create
Read
Update
Delete

Maintaining data **integrity**

- Simultaneous access problems
  - Incorrect summary problem
  - Lost update problem
- **Locking**
  - preventing others from accessing data being used by a transaction
    - **Shared lock**: used when reading data
    - **Exclusive lock**: used when altering data

Incorrect summary problem

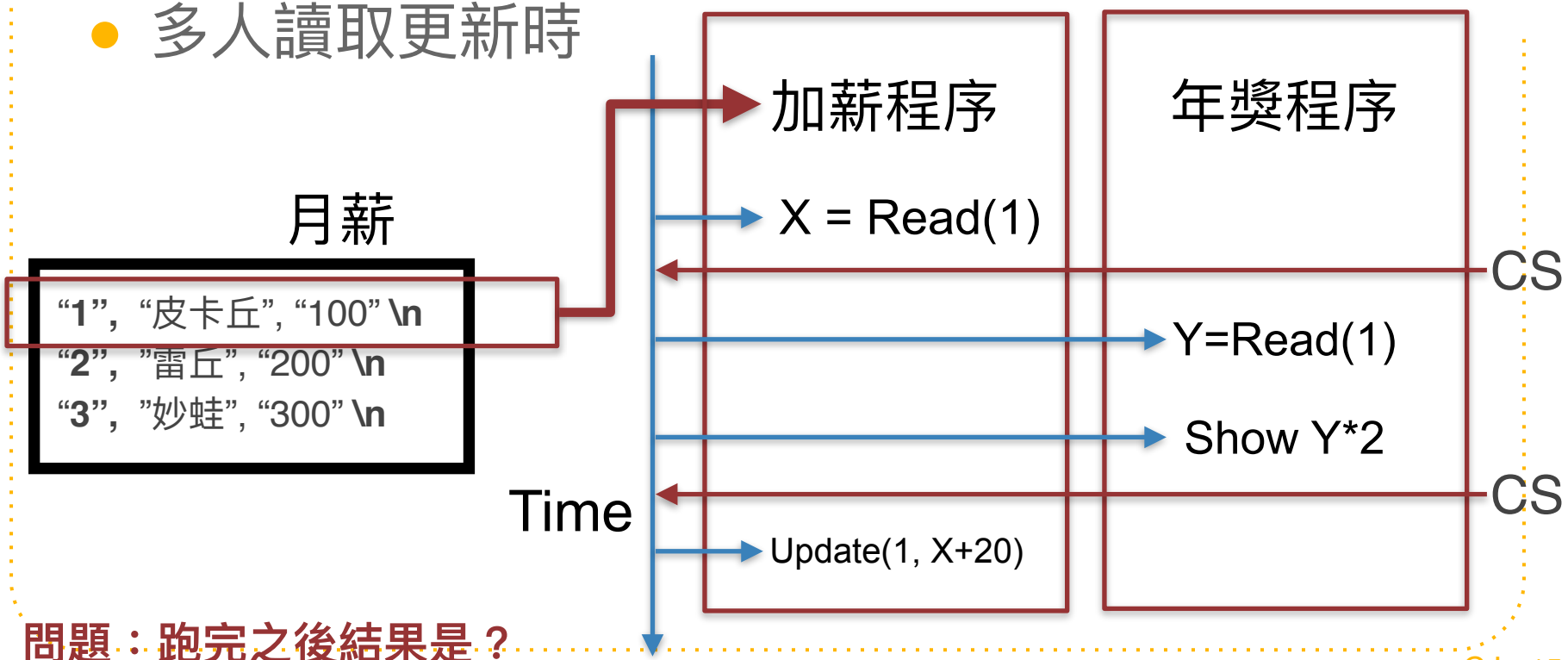
- 多人讀取更新時



問題：跑完之後Sum結果是？

Lost update problem

- 多人讀取更新時

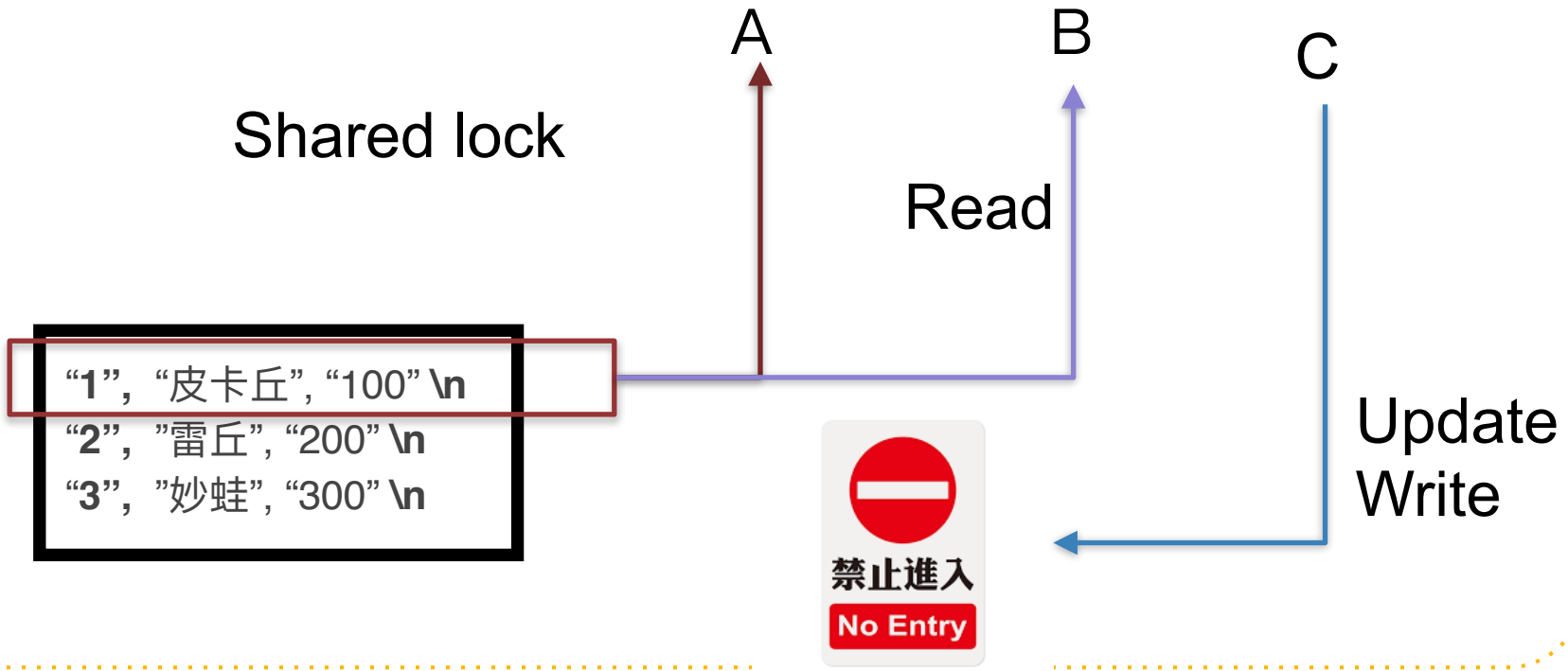


問題：跑完之後結果是？

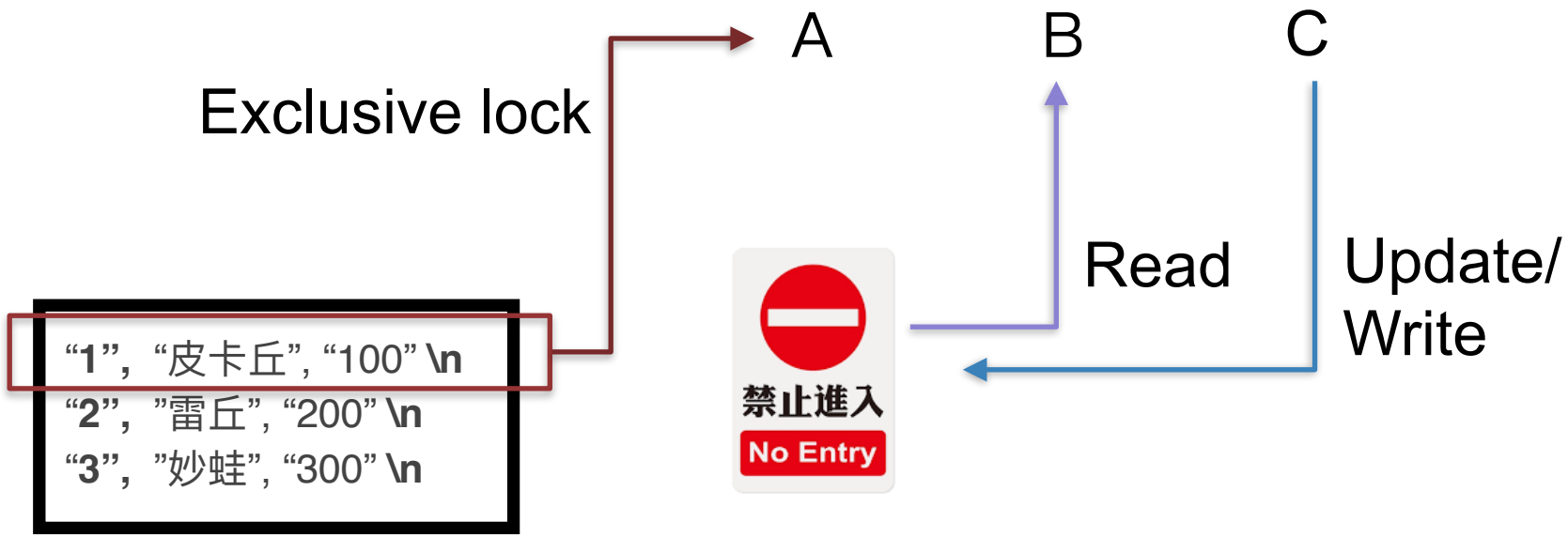
Locking

- Preventing others from accessing data being used by **a transaction**
  - **Shared lock (Read lock)**
    - used when reading data
    - Only **read**
  - **Exclusive lock (Write lock)**
    - used when altering data (Update data)
    - **No one can access**

Shared lock

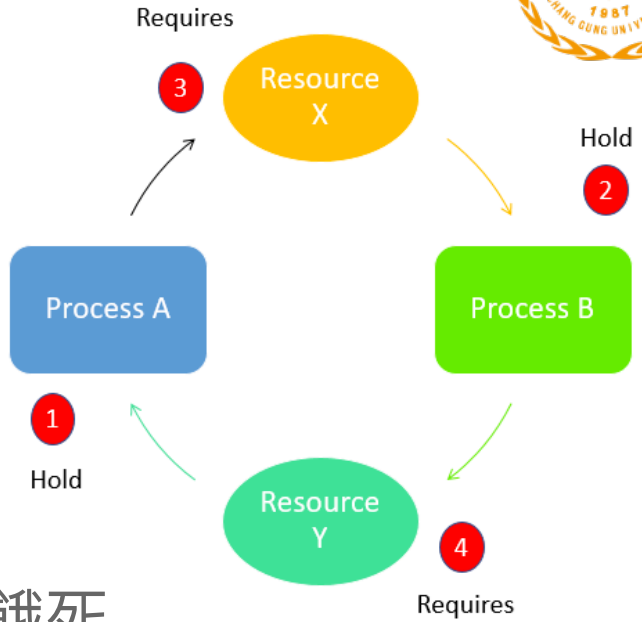

A
B
C
Shared lock
Read
“1”, “皮卡丘”, “100” \n
“2”, ”雷丘”, “200” \n
“3”, ”妙蛙”, “300” \n
禁止進入
No Entry
Update
Write

Exclusive lock



Locking 問題？

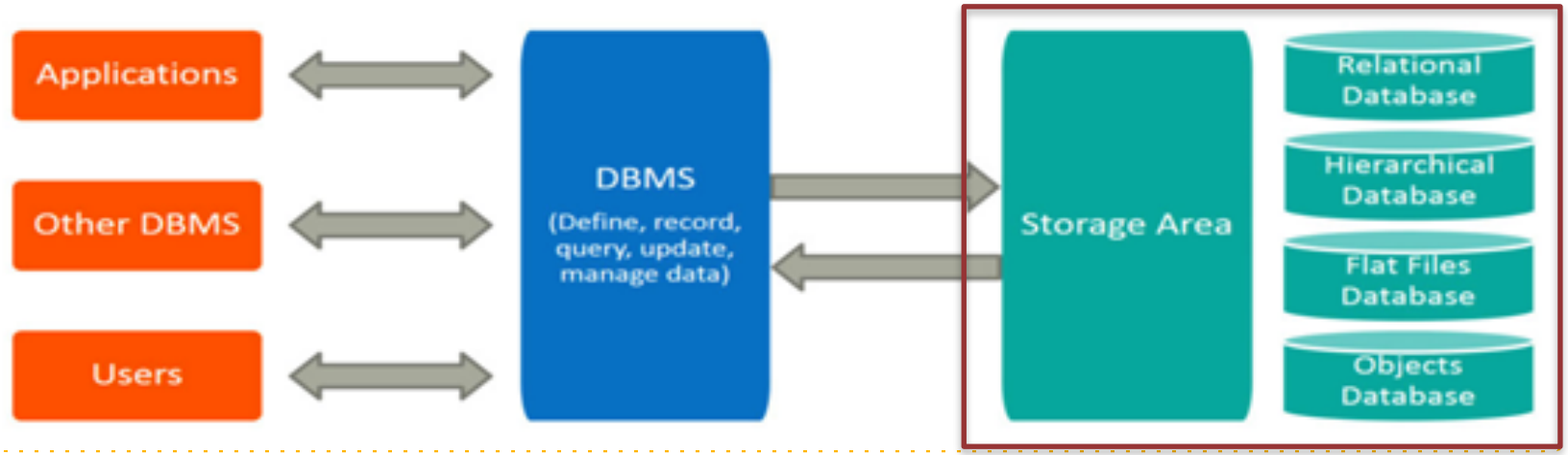
- **Deadlock**
  - 作業系統中要存取共同的資源
  - 有了鎖，就有機會產生死結
  - 問題越來越多！
- **Starvation**
  - 產生了死結，就有可能會有人會餓死
  - **系統就好像不動了**，程序也無法進行



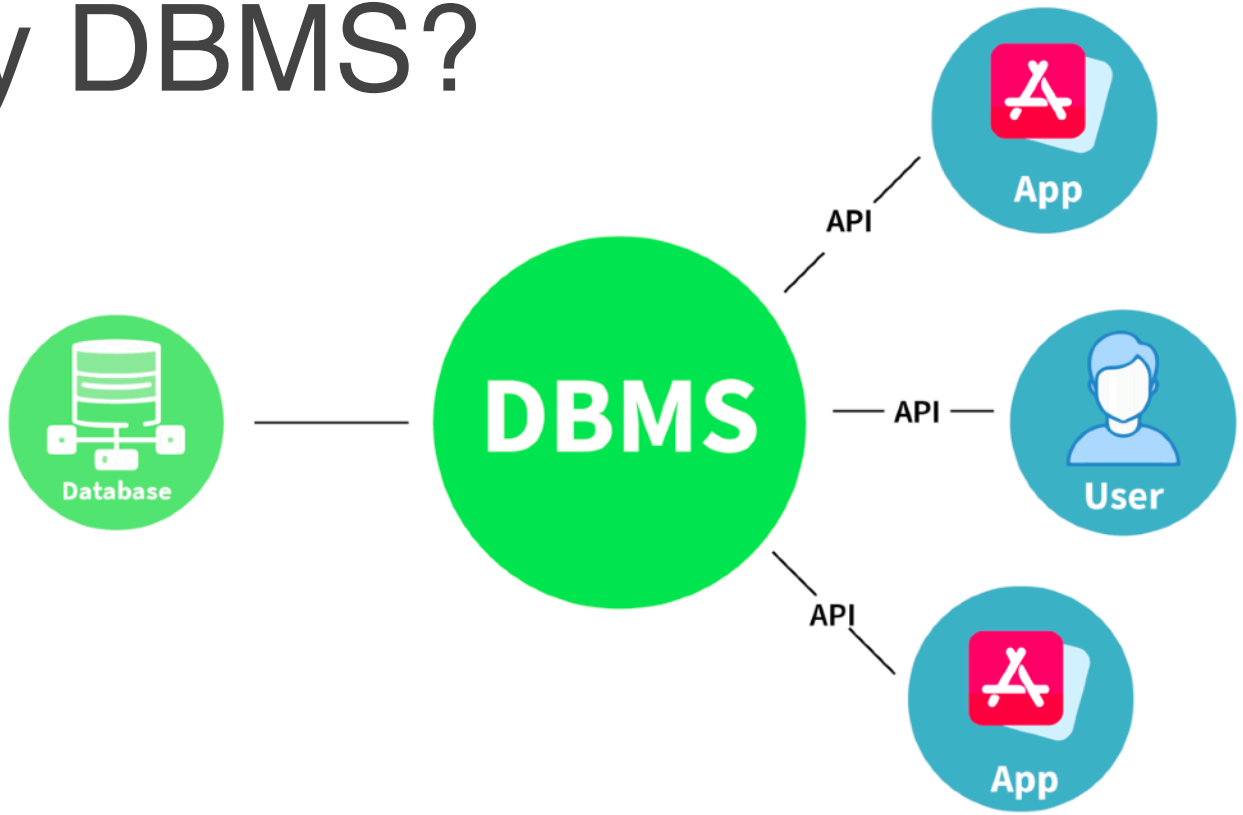
問題：這時候怎麼辦呢？電腦怎麼處理死結？

DataBase Management System (DBMS)

- 所以我們需要一個程式來控制**資料庫**的分類及數據的存取，解決**多人讀取**的問題



Why DBMS?



Maintaining Database Integrity

- 在資料庫中，我們把一個不能分割的資料操作稱為
- **Transaction (交易或事務)**
  - **A sequence of operations that must all happen together**

transferring money between bank accounts

Transaction

Transaction
交易/事務

資料庫執行過程中的一個「邏輯單位」

一個transaction = 一組一連串對資料庫進行存取、讀取的行為

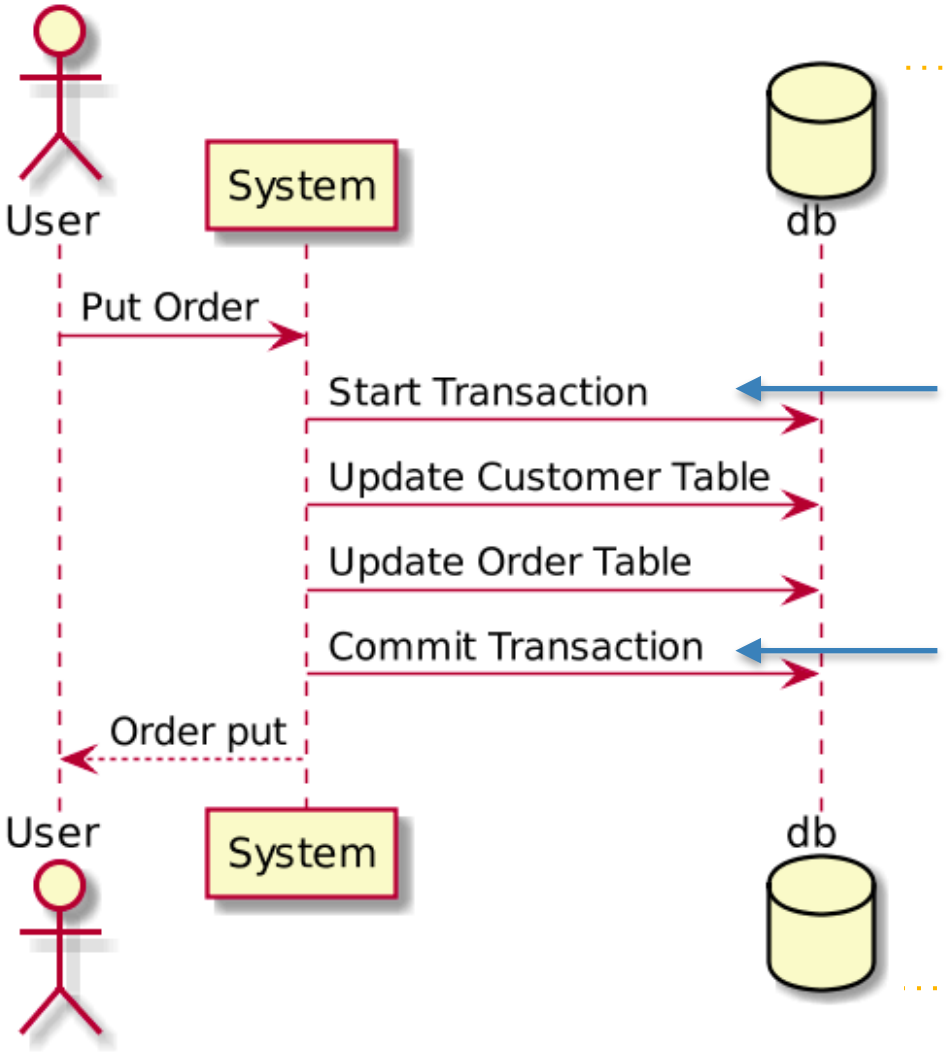
兩種結局

成功 or **失敗**

全部SQL執行成功 -> commit

只要有任一個SQL失敗 -> rollback

Transaction Example



Transaction: **ACID**

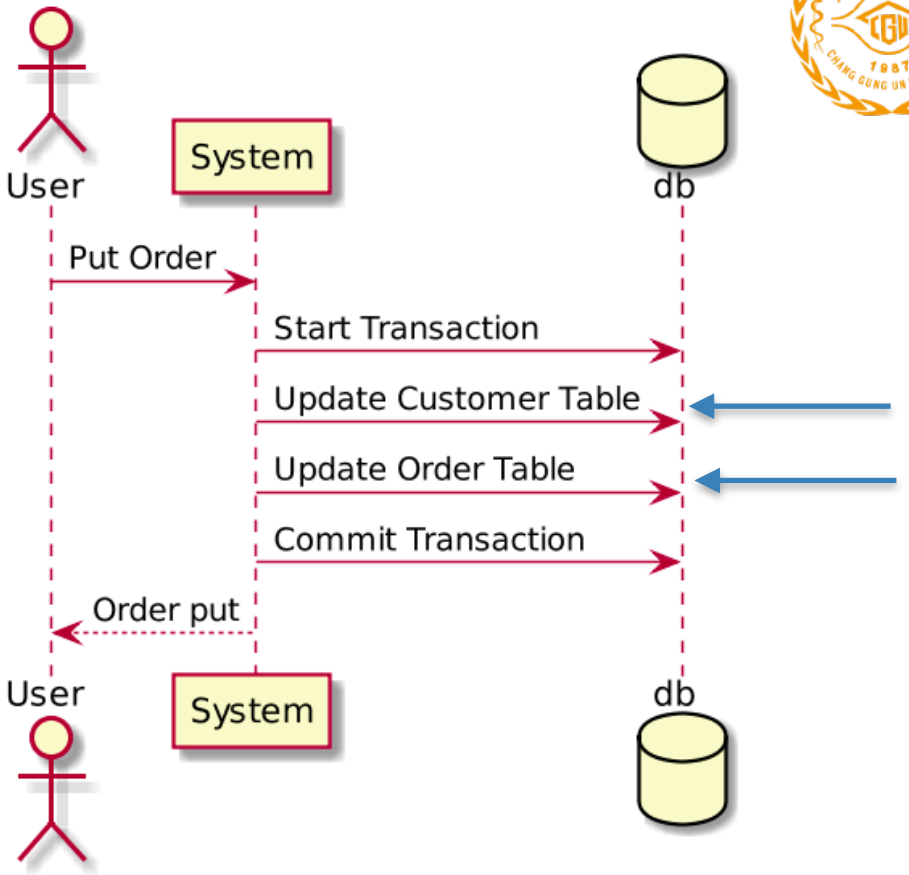
- 原子性 **Atomicity**
  - 所有操作，或者**全部完成**，或者**全部不完成**，不會結束在中間某個環節
- 一致性 **Consistency**
  - 結束以後，資料庫的完整性沒有**被破壞**
- 事務隔離 **Isolation**
  - 防止多個事務並發(同時)執行時由於交叉執行而導致數據的**不一致**
- 持久性 **Durability**
  - 對數據的修改就是**永久**的

Transaction Log (Journal)

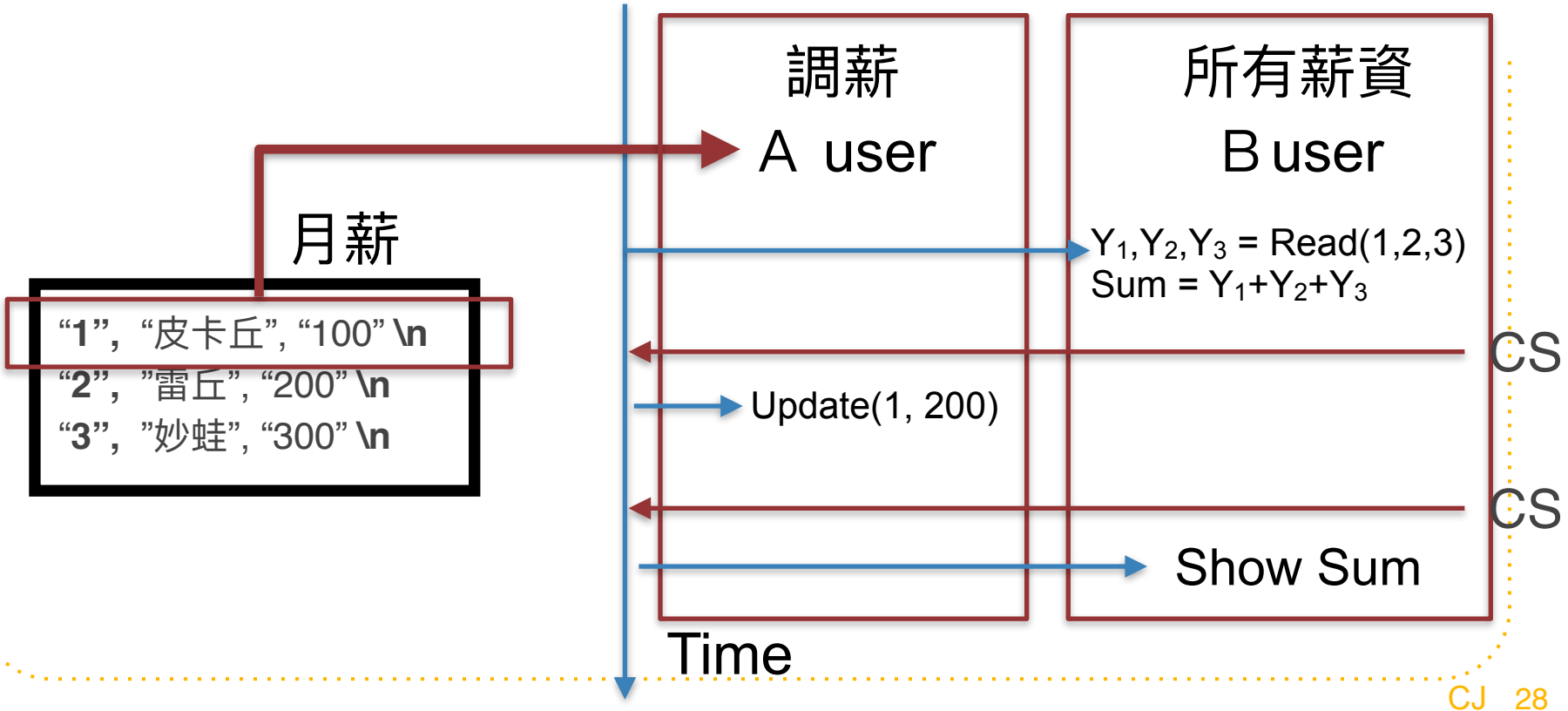
- A **non-volatile record** of each transaction's activities
  - **Commit point**:
    - The point at which a transaction has been recorded in the log
  - **Roll-back**:
    - The process of undoing a transaction

Commit point

- The **point** has been recorded in the log
  - **Write**
  - **Update**



問題：Commit point 在哪裏？



Transaction Log Example

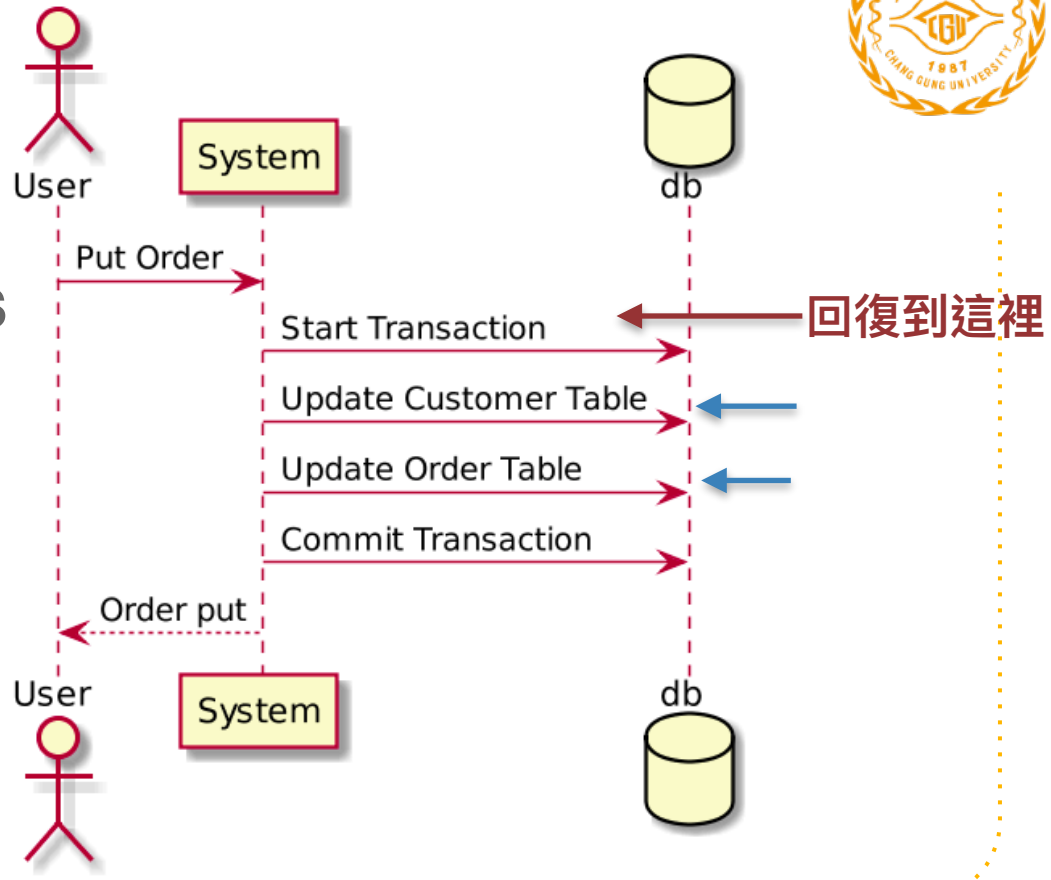
	Current LSN	Operation	Context	Transaction ID	AllocUnitId	AllocUnitName
3576	0000008a:00000198:0010	LOP_EXPUNGE_ROWS	LCX_CLUSTERED	0000:00000000	281474979397632	sys.syscolpars.clst
3577	0000008a:00000198:0011	LOP_EXPUNGE_ROWS	LCX_CLUSTERED	0000:00000000	281474979397632	sys.syscolpars.clst
3578	0000008a:00000198:0012	LOP_EXPUNGE_ROWS	LCX_CLUSTERED	0000:00000000	281474979397632	sys.syscolpars.clst
3579	0000008a:00000198:0013	LOP_EXPUNGE_ROWS	LCX_CLUSTERED	0000:00000000	281474979397632	sys.syscolpars.clst
3580	0000008a:00000198:0014	LOP_SET_BITS	LCX_PFS	0000:00000000	281474979397632	sys.syscolpars.clst
3581	0000008a:00000198:0015	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3582	0000008a:00000198:0016	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3583	0000008a:00000198:0017	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3584	0000008a:00000198:0018	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3585	0000008a:00000198:0019	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3586	0000008a:00000198:001a	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3587	0000008a:00000198:001b	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3588	0000008a:00000198:001c	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3589	0000008a:00000198:001d	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3590	0000008a:00000198:001e	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	562949956108288	sys.syscolpars.nc
3591	0000008a:00000198:001f	LOP_SET_BITS	LCX_PFS	0000:00000000	562949956108288	sys.syscolpars.nc
3592	0000008a:00000198:0020	LOP_EXPUNGE_ROWS	LCX_INDEX_LEAF	0000:00000000	844424932360192	sys.sysschobjs.nc2
3593	0000008a:00000198:0021	LOP_SET_BITS	LCX_PFS	0000:00000000	844424932360192	sys.sysschobjs.nc2

Roll-back

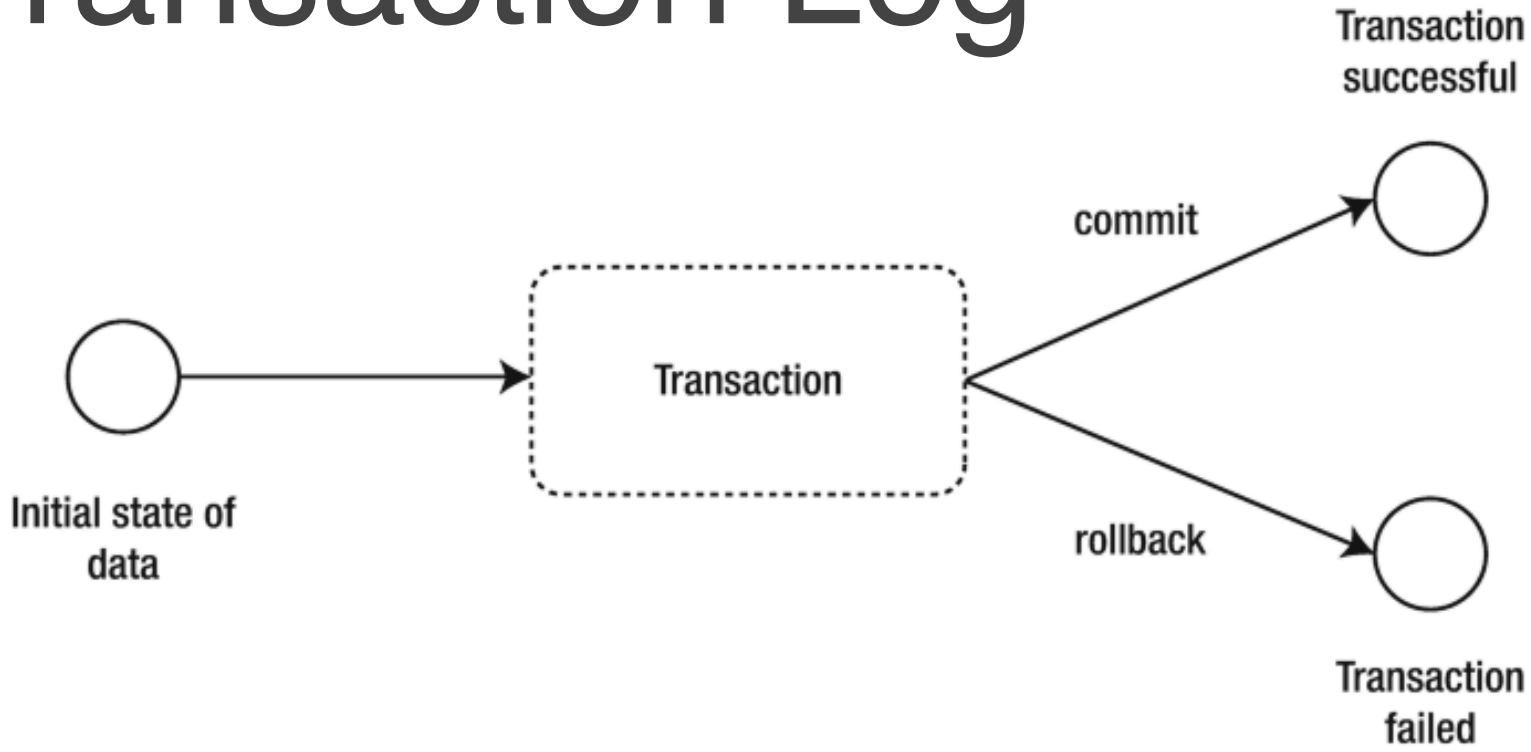
- The process of **undoing** a transaction

Roll-back

- **Undoing the points** have been recorded in the log
  - **Write**
  - **Update**



Transaction Log



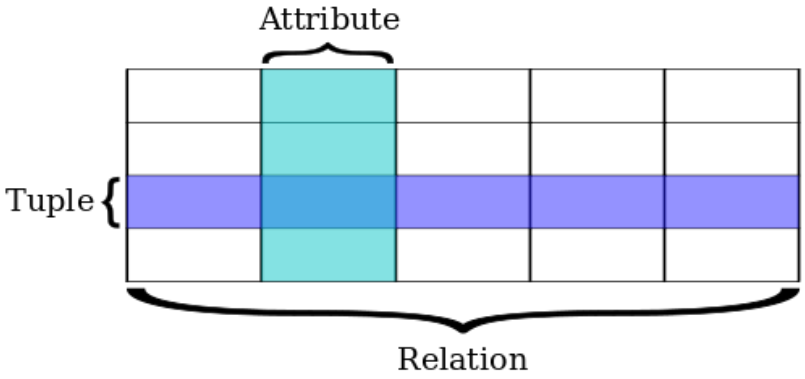
Modern Databases

- Hierarchical Databases
  - Organize and manage data in a **tree-like** structure
- **Relational Databases**
- **Object-Oriented Databases**
- **Column-based databases**
- **Key-Value databases**
- **NoSQL databases**
- Graph databases

Graph

The Relational Databases

- **Entity-Relationship Model**
- Relation: A rectangular table
  - Attribute:
    - A column in the table
  - Tuple:
    - A row in the table



- **Primary key**
  - A primary key uniquely specifies a tuple within a table
- **Foreign key**
  - A field matches the primary key column of another table

Entity-Relationship Model

- 設計資料庫的重要方法
- Entity, Attribute, Relationship

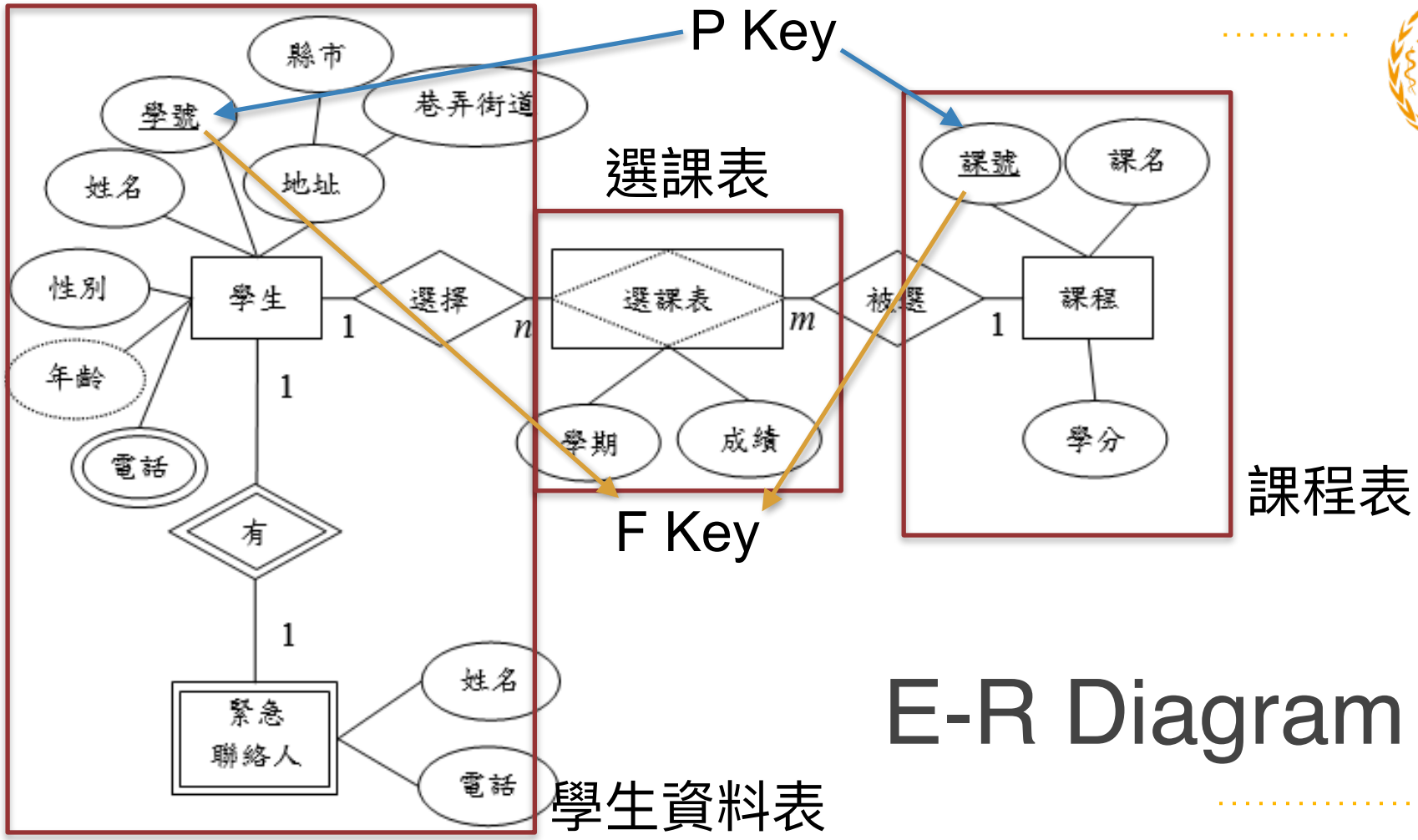
一開始要靠想像的，之後是由需求決定

- Relationship
  - 1 to 1
  - 1 to m
  - n to m

弱關聯

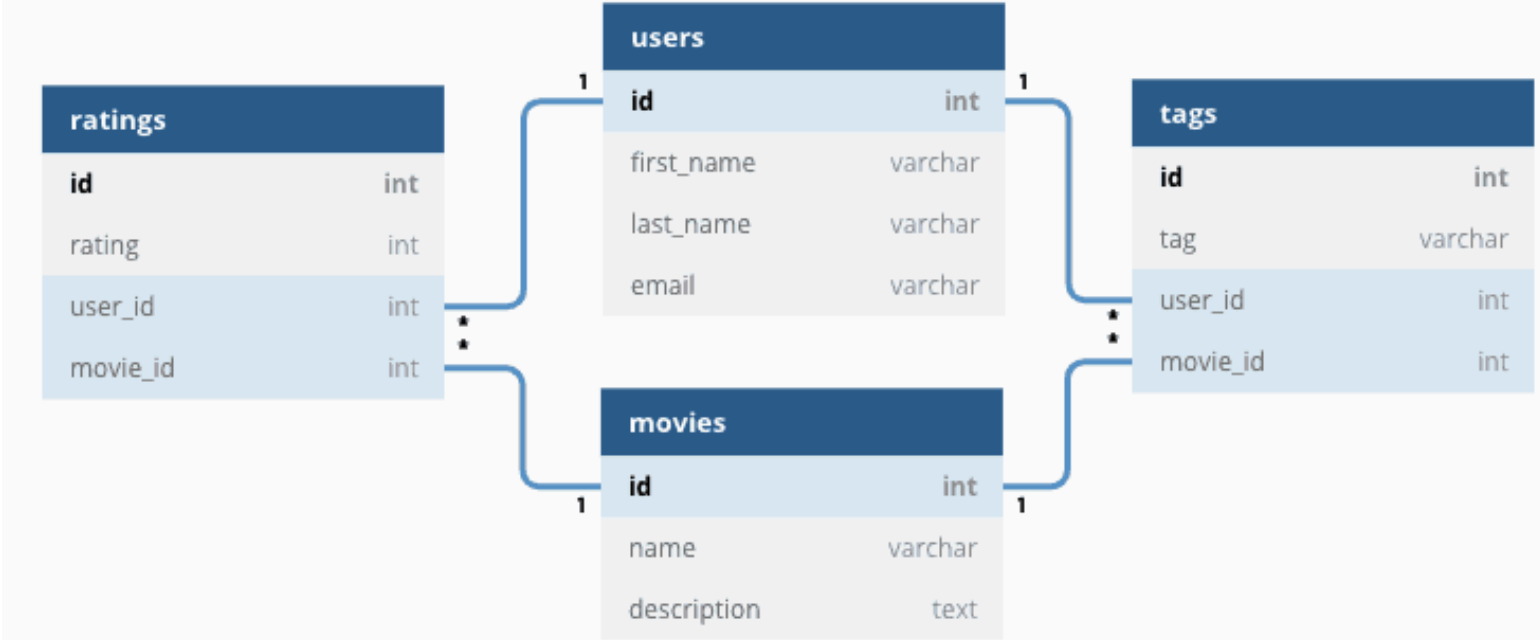
關聯

關聯實體


P Key
縣市
學號
巷弄街道
姓名
地址
性別
學生
選擇
1
n
選課表
m
被選
1
課程
課號
課名
年齡
1
電話
有
1
學期
成績
學分
F Key
課程表
緊急
聯絡人
姓名
電話
學生資料表

E-R Diagram

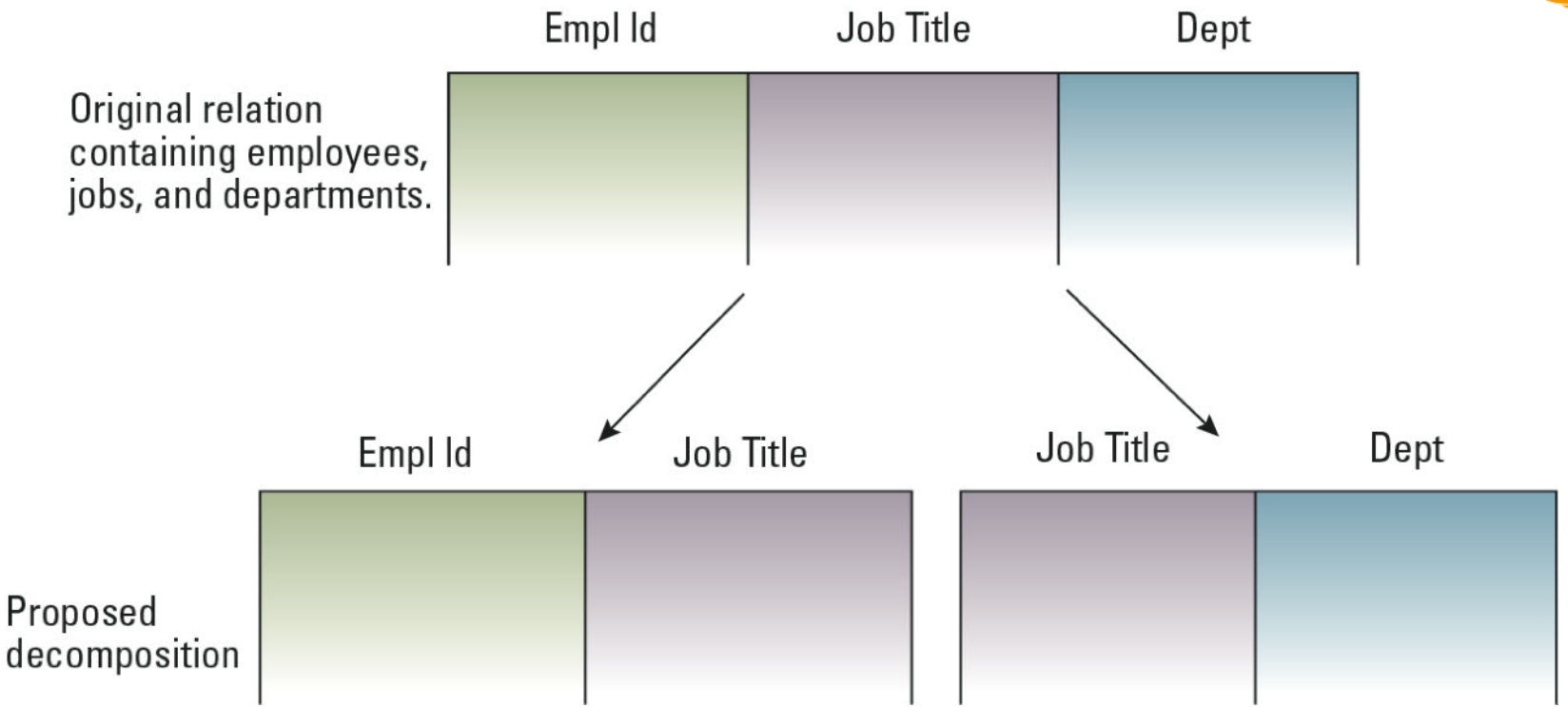
Relational Database Example



A relation containing **redundancy**

Empl Id	Name	Address	SSN	Job Id	Job Title	Skill Code	Dept	Start Date	Term Date
25X15	Joe E. Baker	33 Nowhere St.	111223333	F5	Floor manager	FM3	Sales	9-1-2009	9-30-2010
25X15	Joe E. Baker	33 Nowhere St.	111223333	D7	Dept. head	K2	Sales	10-1-2010	*
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999	F5	Floor manager	FM3	Sales	10-1-2009	*
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S25X	Secretary	T5	Personnel	3-1-1999	4-30-2010
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S26Z	Secretary	T6	Accounting	5-1-2010	*
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•

Decomposition



A relation containing **redundancy**

Empl Id	Name	Address	SSN	Job Id	Job Title	Skill Code	Dept	Start Date	Term Date
25X15	Joe E. Baker	33 Nowhere St.	111223333	F5	Floor manager	FM3	Sales	9-1-2009	9-30-2010
25X15	Joe E. Baker	33 Nowhere St.	111223333	D7	Dept. head	K2	Sales	10-1-2010	*
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999	F5	Floor manager	FM3	Sales	10-1-2009	*
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S25X	Secretary	T5	Personnel	3-1-1999	4-30-2010
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555	S26Z	Secretary	T6	Accounting	5-1-2010	*
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•

EMPLOYEE relation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555

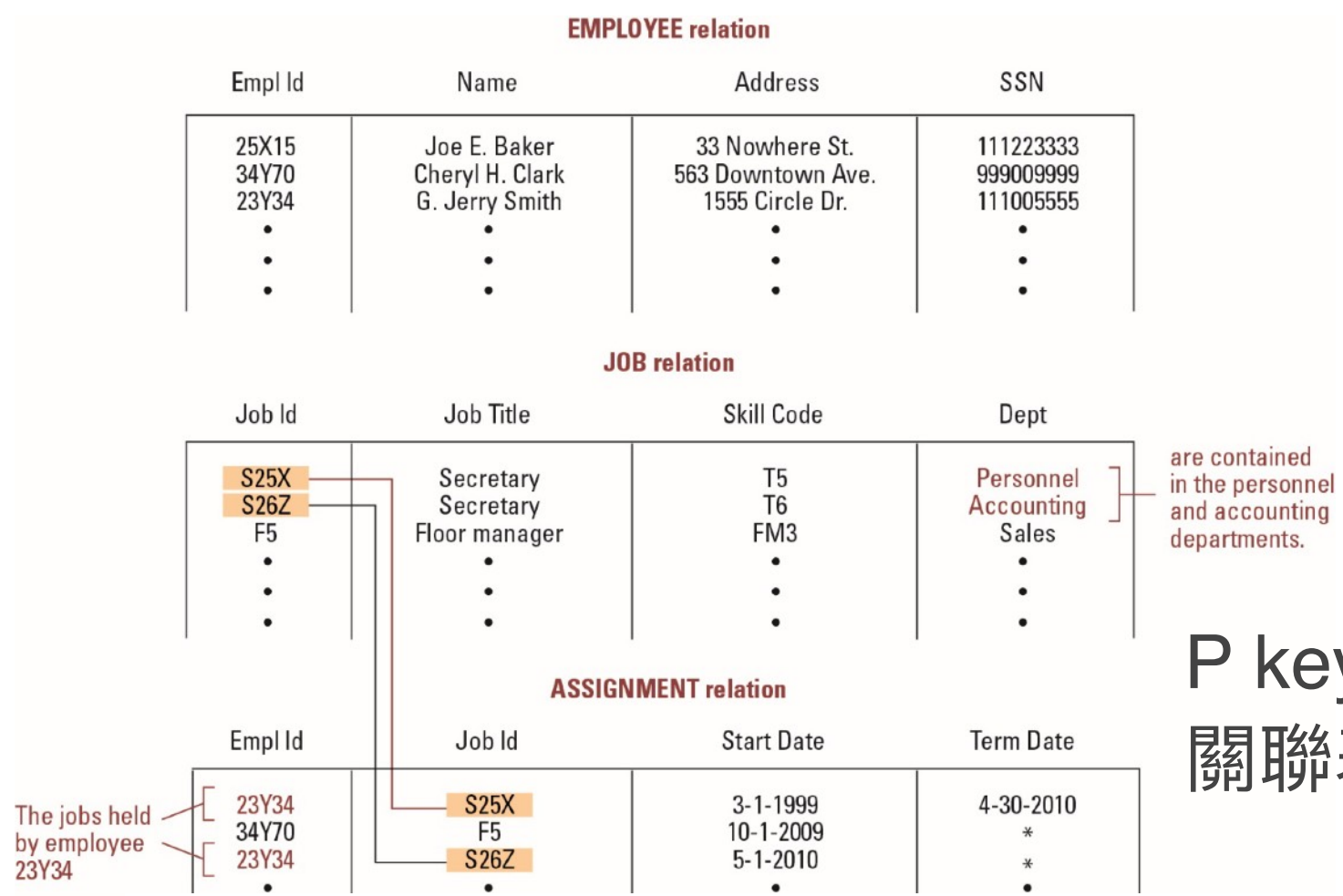
JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•
•	•	•	•

ASSIGNMENT relation

Empl Id	Job Id	Start Date	Term Date
23Y34	S25X	3-1-1999	4-30-2010
34Y70	F5	10-1-2009	*
23Y34	S26Z	5-1-2010	*
•	•	•	•
•	•	•	•
•	•	•	•

Decomposition
變成三張表



P key & F key 關聯表

Finding the departments in which employee 23Y34 has worked

Relational Operations

- **SELECT**:
  - Choose rows
- **PROJECT**:
  - Choose columns
- **JOIN**:
  - Assemble information from relations

EMPLOYEE relation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

SELECT

JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•

are contained in the personn and accountin departments.

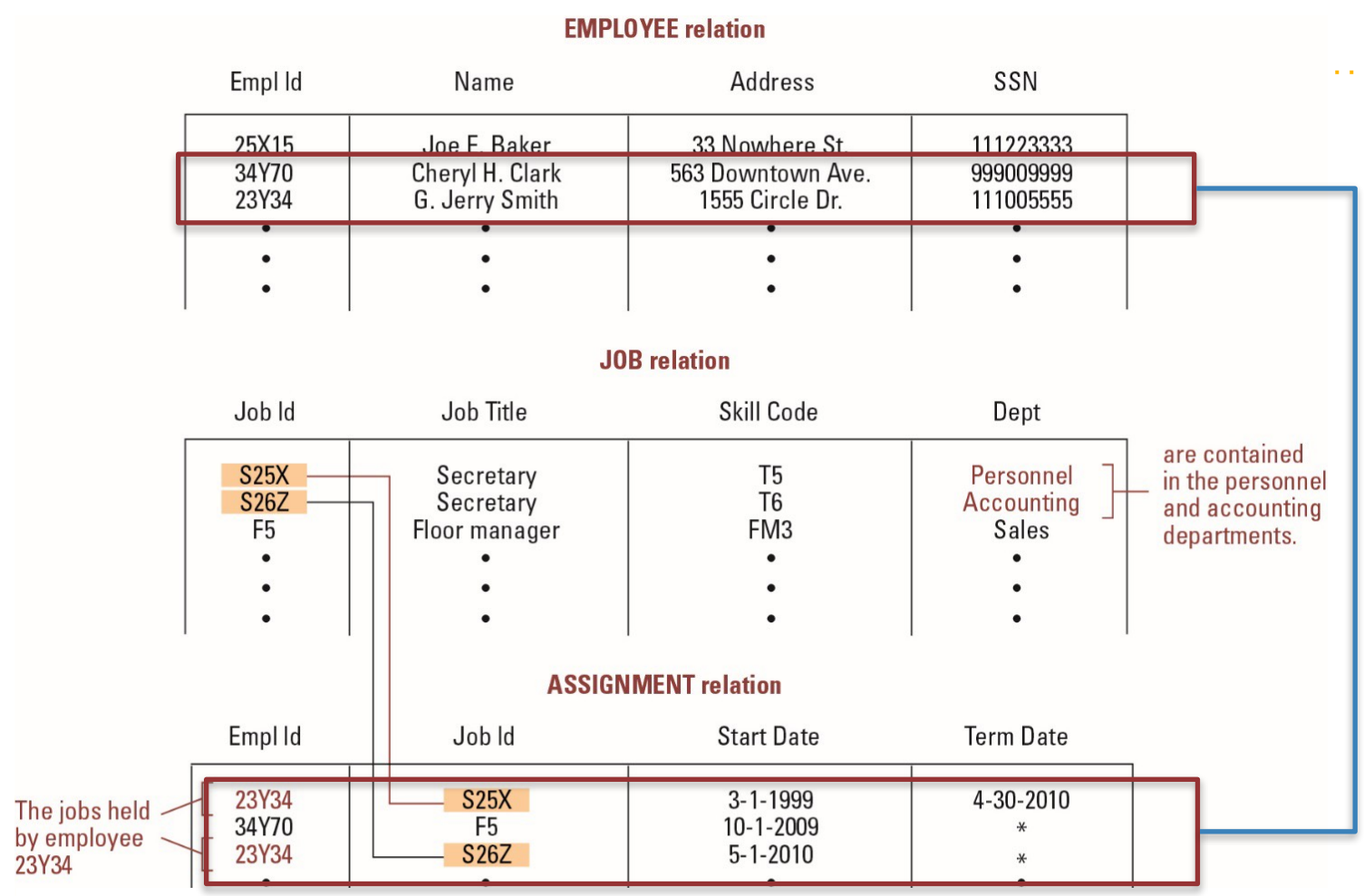
EMPLOYEE relation

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

PROJECT

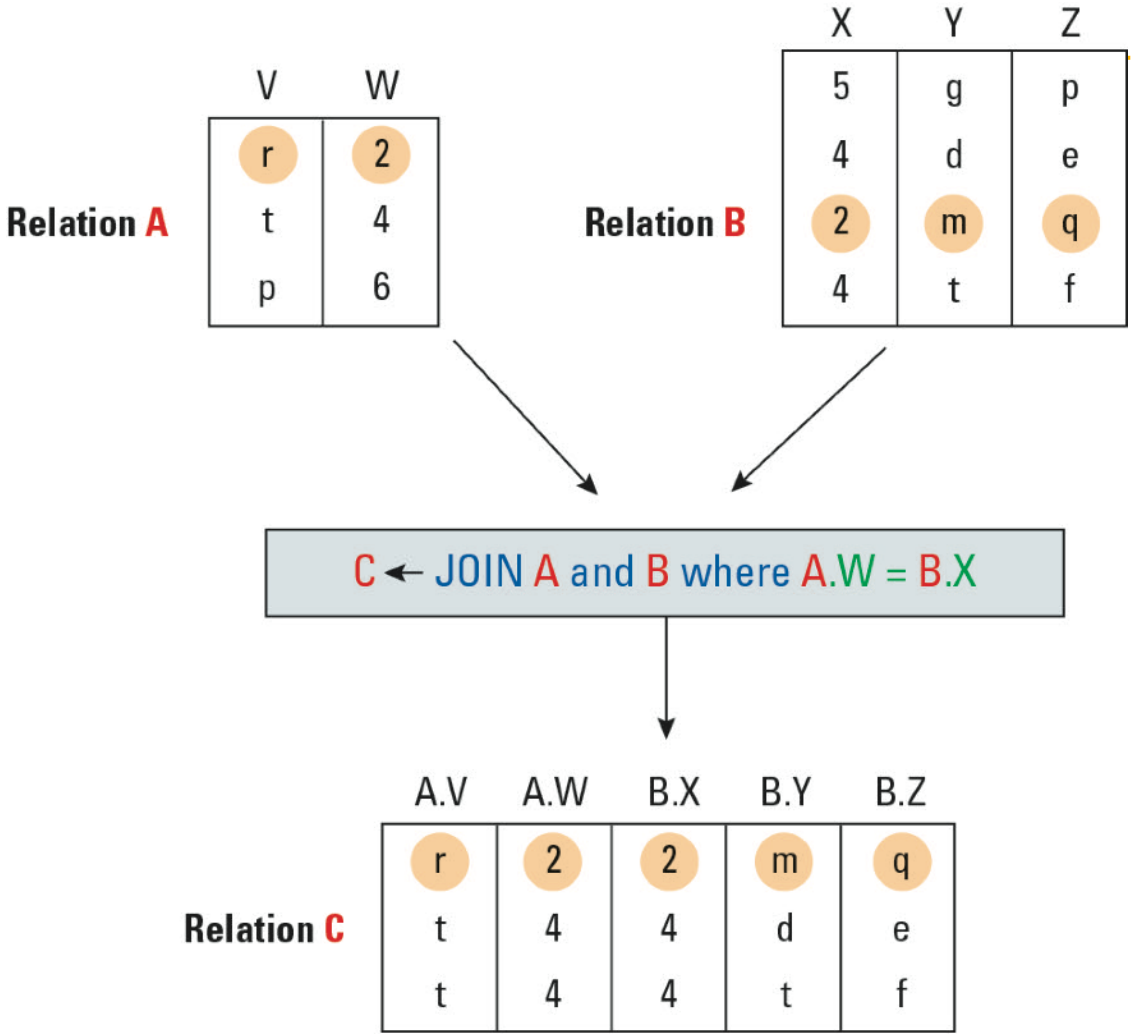
JOB relation

Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting

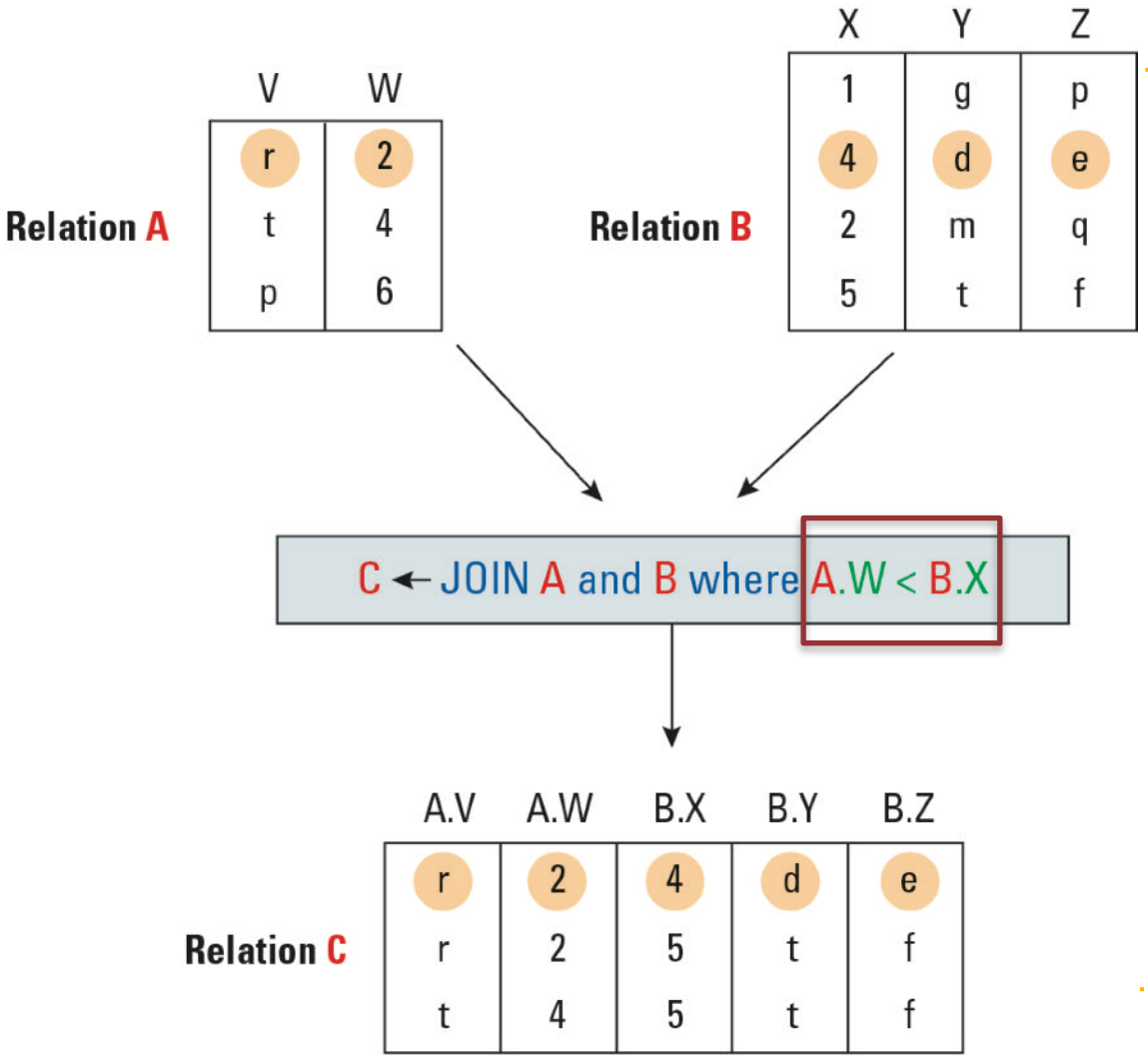


JOIN Tables

Finding the departments in which employee 23Y34 & 34Y70 has worked

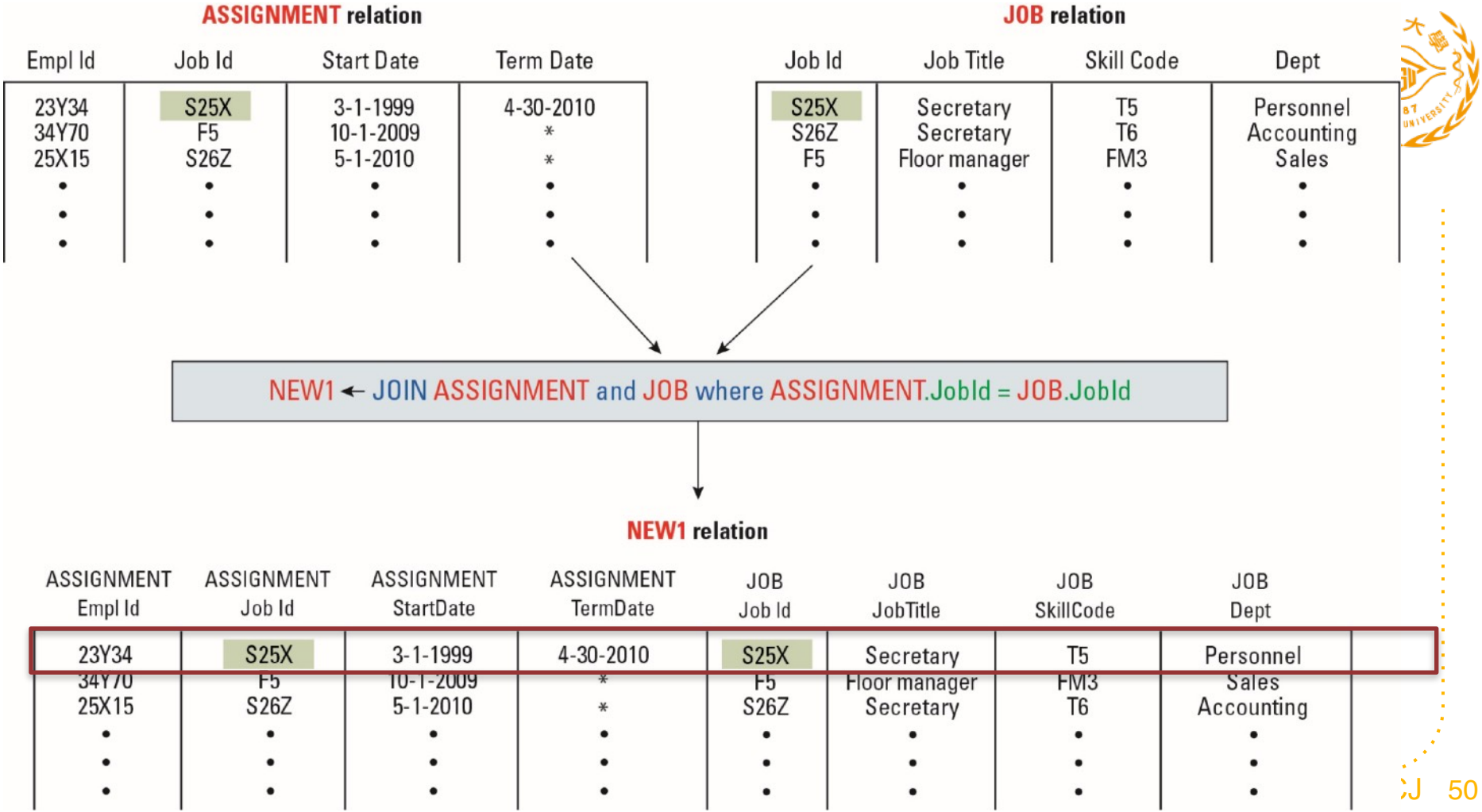


JOIN operation



JOIN operation

=, <=, <


ASSIGNMENT relation
Empl Id
Job Id
Start Date
Term Date
23Y34
34Y70
25X15
S25X
F5
S26Z
3-1-1999
10-1-2009
5-1-2010
4-30-2010
*
*
JOB relation
Job Id
Job Title
Skill Code
Dept
S25X
S26Z
F5
Secretary
Secretary
Floor manager
T5
T6
FM3
Personnel
Accounting
Sales
NEW1 ← JOIN ASSIGNMENT and JOB where ASSIGNMENT.JobId = JOB.JobId
NEW1 relation
ASSIGNMENT Empl Id
ASSIGNMENT Job Id
ASSIGNMENT StartDate
ASSIGNMENT TermDate
JOB Job Id
JOB JobTitle
JOB SkillCode
JOB Dept
23Y34
S25X
3-1-1999
4-30-2010
S25X
Secretary
T5
Personnel
34Y70
F5
10-1-2009
*
F5
Floor manager
FM3
Sales
25X15
S26Z
5-1-2010
*
S26Z
Secretary
T6
Accounting

Structured Query Language (SQL)

- 特定目的**程式語言**，用於管理**關聯式資料庫**
  - IBM, 1974, 1987
- Operations to manipulate tuples
  - **select**
  - **insert**
  - **update**
  - **delete**
- 網站安全

CRUD 操作:
Create
Read
Update
Delete

SELECT and Join Select

- ```
  SELECT EmplId, Dept
  FROM Assignment
  WHERE Assignment.TermData = '*';
  ```

- ```
  SELECT EmplId, Dept
  FROM Assignment, Job
  WHERE Assignment.JobId = Job.JobId
      AND Assignment.TermData = '*';
  ```

INSERT

- ```
  INSERT INTO Employee
  VALUES ('43212', 'Sue A. Burt',
          '33 Fair St.', '444661111');
  ```
  ```

Delete and Update

- ```
  DELETE FROM Employee
  WHERE Name = 'G. Jerry Smith';
  ```

- ```
  UPDATE Employee
  SET Address = '1812 Napoleon Ave.'
  WHERE Name = 'Joe E. Baker';
  ```

https://sqliteonline.com/

SQL 在電腦科學中

- SQL 是**非常重要**的技能！！
- 目前一定要知道
  - **CRUD** 操作所需要的SQL
  - **SELECT**, **INSERT**
  - **UPDATE**, **DELETE**
- https://www.w3schools.com/sql/default.asp
- SQL 語法要背起來！ 應該會用到你孫子生出來！

CRUD 操作:
Create
Read
Update
Delete

Object-Oriented Databases

- Object-oriented Database:
  - Constructed by applying the **object-oriented paradigm**
  - Each entity stored as a **persistent** object
  - **Relationships** indicated by links between objects
  - DBMS maintains **inter-object links**

Relational Databases

訂單序號	日期	客戶編號	是否付款
1	2005/7/1	6	1
2	2005/7/1	3	1
3	2005/7/3	2	0

訂單資料表

客戶編號	客戶名稱	聯絡人	性別	地址
1	十全書店	陳圓圓	女	台北市
2	大發書店	陳季暄	女	台北市
3	好看書店	趙飛燕	女	台中市

客戶資料表

被動關聯

	日期	客戶	是否付款
OID 1	2005/7/1	OID 10	1
OID 2	2005/7/1	OID 11	1
OID 3	2005/7/3	OID 12	0

訂單

	客戶名稱	聯絡人	性別	地址
OID 9	十全書店	陳圓圓	女	台北市
OID 10	大發書店	陳季暄	女	台北市
OID 11	好看書店	趙飛燕	女	台中市

客戶

主動關聯

Object-oriented Databases

Advantages of Object-oriented Databases

- Matches design paradigm of **object-oriented applications**
- Intelligence can be built into attribute handlers
- Can handle exotic data types
  - Example
    - multimedia
- 實際上使用的不多，主要是太複雜了，效率也不好

Column-based databases

- A type of NoSQL database
- 適合在數據分析
  - 得到某欄位的max, min

Row Oriented Database

date	price	size
2011-01-20	10.1	10
2011-01-21	10.3	20
2011-01-22	10.5	40
2011-01-23	10.4	5
2011-01-24	11.2	55
2011-01-25	11.4	66
...	...	...
2013-03-31	17.3	100

Column Oriented Database

date	price	size
2011-01-20	10.1	10
2011-01-21	10.3	20
2011-01-22	10.5	40
2011-01-23	10.4	5
2011-01-24	11.2	55
2011-01-25	11.4	66
...	...	...
2013-03-31	17.3	100

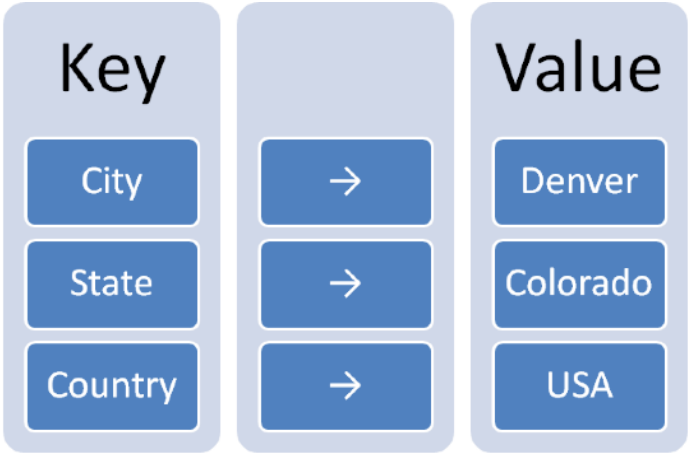
amazon DynamoDB

Google BigTable

SCYLLA

Key-Value databases

- A dictionary
  - Key —> value



Memcached

RocksDB

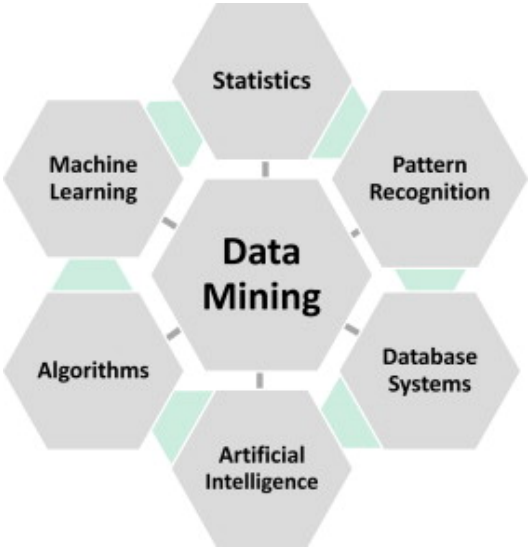
redis

NoSQL databases

- 不同於**關聯式資料庫**的資料庫管理系統的統稱
- Use in **big data** and **real-time web** applications
- May lack true **ACID** transactions
- MongoDB, Cassandra

Data Mining

- 從資料庫中挖掘出潛在、明確、而且非常有用資訊的過程
- 主要方法
  - **統計分析**和**人工智慧**技術
- 改變未來的十大創新技術, 2002, MIT
- 處理問題：
  - 請評估明年長庚資工男女生比例？
  - 請預估下星期林口天氣如何？



Conclusion

- The history of database systems
- Database fundamentals
- Traditional File Structures
- Maintaining Database Integrity
- The Relational Model
- Object-Oriented Databases
- Data Mining

儲存和連結
資料和
資料之間的關係

Thanks!

Open for any questions

CJ Wu

cjwu@mail.cgu.edu.tw