

Database Systems

資料庫系統

國立政治大學

資訊科學系

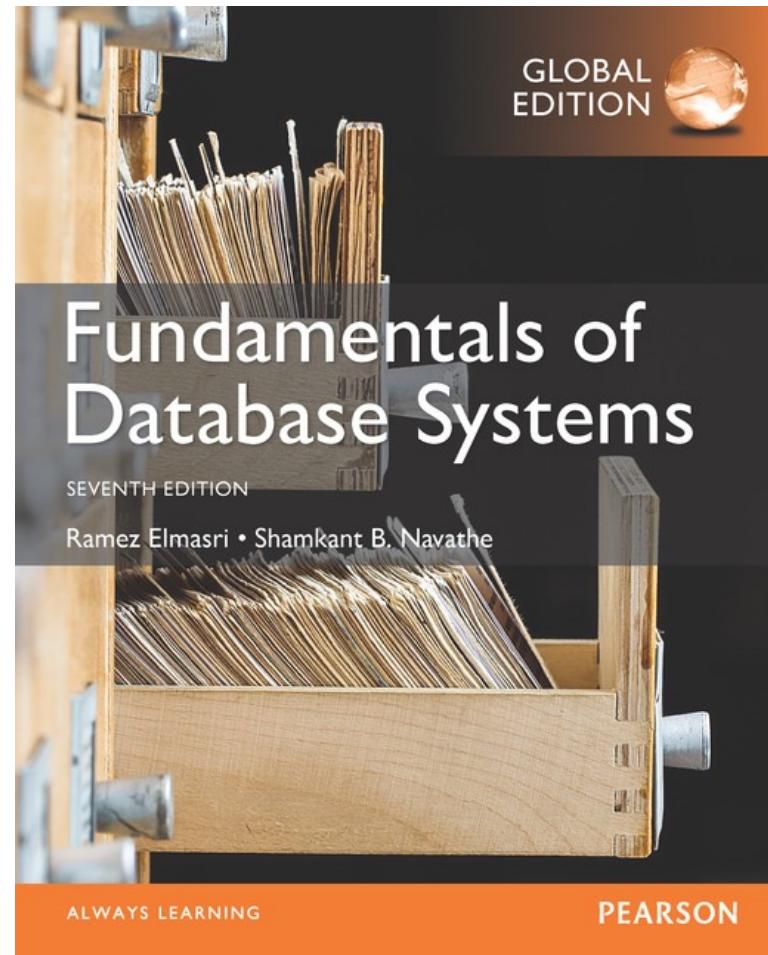
沈錨坤

課程資訊

- ◆ Instructor: 沈錦坤 (Man-Kwan Shan)
 - Email: mkshan@nccu.edu.tw
 - Office Hour: Fri. 16:00~17:00 or by Appointment
- ◆ Teaching Assistant
 - 許軒祥 (111753122@nccu.edu.tw)
 - 陳筱詩 (111753201@nccu.edu.tw)

Text Book

- Fundamentals of Database Systems, 7th Ed., R. Elmasri & S. B. Nanathe, Pearson, 2021. (歐亞)



Reference Book

- ◆ Database System Concepts, A. Silberschatz, H. F. Korth, & S. Sudarshan, 2006.
- ◆ An Introduction to Database Systems, C. J. Date, 2000.
- ◆ Database System Implementation, E. Garcia-Molina, J. D. Ullman, J. Widom, 2009.

The SD Times news article discusses the 2020 ACM A.M. Turing Award winners, Alfred Vaino Aho and Jeffrey David Ullman. The article highlights their work on fundamental algorithms and theory, which underlies programming language compilers and algorithm innovators. It includes portraits of Alfred Aho and Jeffrey Ullman, and links to their profiles.

SD Times SOFTWARE DEVELOPMENT

ITOps Times Features Subscribe Learning Center Podcasts Events

API AGILE CONTAINERS DATA DEVOPS DEVSECOPS LOW CODE MICROSERVICES MONITOR

Programming language compilers and algorithm innovators awarded ACM A.M. Turing Award

Latest News Published: April 1st, 2021 ~ Christina Cardoza

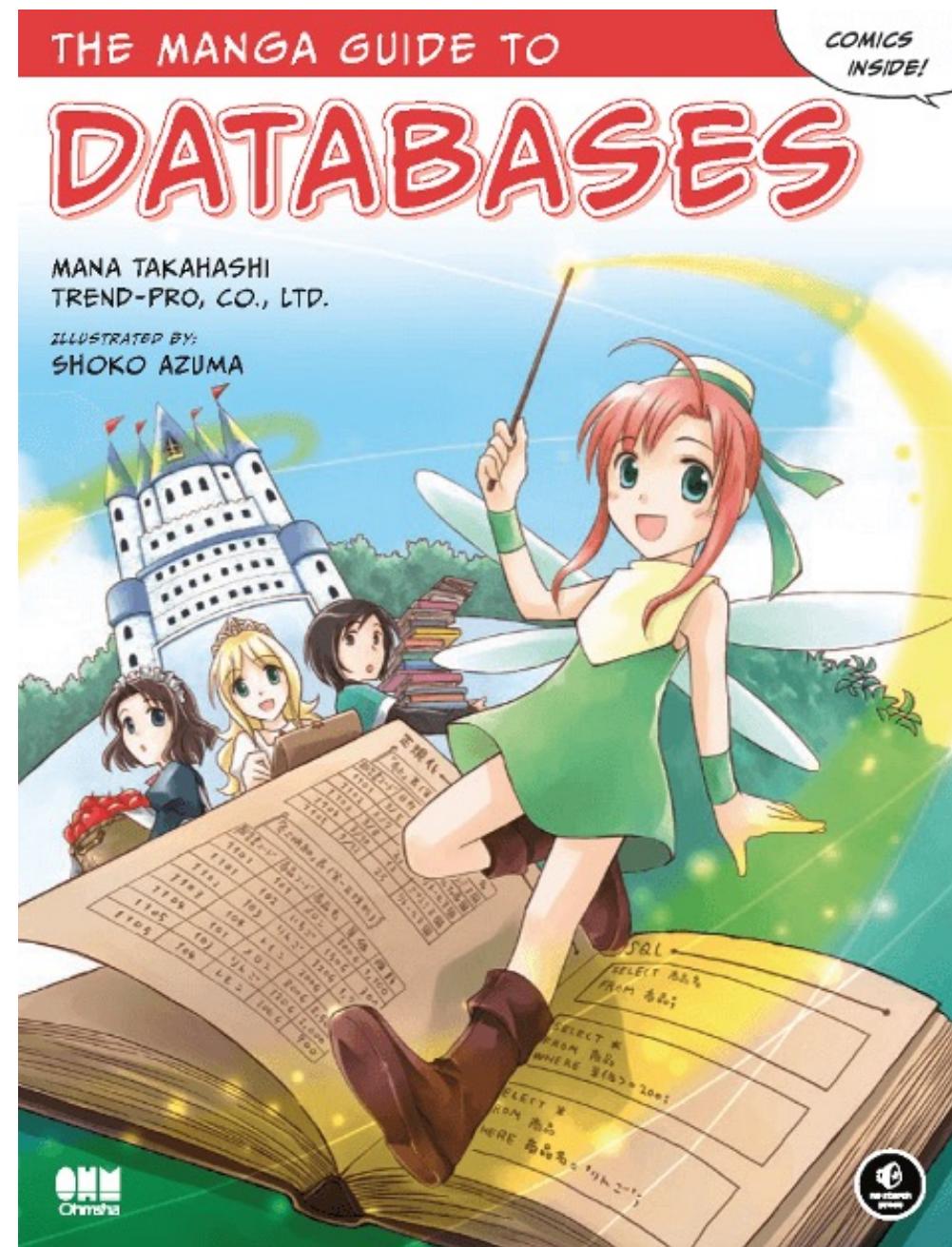
Twitter Email + More

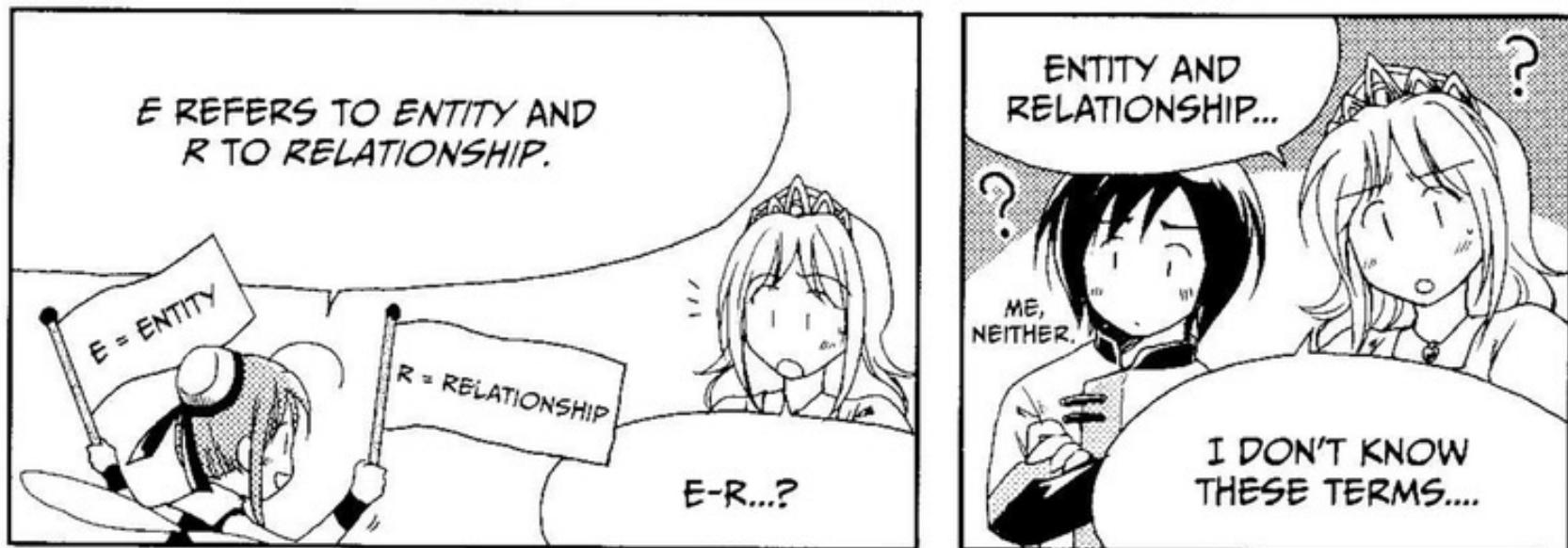
The Association for Computing Machinery (ACM) has announced Alfred Vaino Aho and Jeffrey David Ullman as the [2020 ACM A.M. Turing Award](#) winners for their work on fundamental algorithms and theory. The ACM A.M. Turing Award, also known as the Nobel Prize of Computing, includes a \$1 million prize with financial support from Google. According to the association, Aho and Ullman's work underlies programming language

Alfred Aho and Jeffrey Ullman receive the 2020 ACM A.M. Turing Award

Alfred Aho

Jeffrey Ullman





教材網站

- ◆ 政大遠距教學網wm5

- 投影片
- 參考網站
- 作業公告與繳交
- 分組學期專題公告及報告繳交

- ◆ 投影片

- 上課的輔助工具
- 投影片不是課本，無法取代課本與上課的角色
- 課前：投影片最晚每週上課前10分鐘上傳
- 課後：參考同學上課反應補充更新

課程定位

- ◆ Focus on principles and applications of
 - Database Design (資料庫設計, 7 weeks)
 - DBMS Design (資料庫系統設計, 8 weeks)
- ◆ not a course about
 - how to use a DBMS (資料庫系統操作)
 - how to build database application systems
(資料庫應用系統開發)
 - how to administer a DBMS (資料庫系統管理)

PART 1

資料庫設計

Database Design

資料庫 ?



學生修課資料庫

Student

學號	姓名	性別	生日	系所
1101	陳綺貞	女	1975/06/06	哲學系
2301	黃奇斌	男	1990/09/10	廣告系
1102	張雨生	男	1966/06/07	外交系
1201	林志玲	女	1974/11/29	資料系
1103	謝馨儀	女	1982/04/16	企管系
1301	吳青峰	男	1982/08/30	中文系
2302	林依晨	女	1982/10/29	韓文系

SC

學生	課程	分數
1101	C3501	90
1102	C3001	70
1102	J2010	80
1103	C3001	100
2301	J3020	85
2301	C3001	90
2302	J2010	70
2302	C3020	80
1301	C3501	80
1301	J2010	85

Course

代碼	名稱	學分	人數限制	選課人數
C3001	資料庫系統	3	10	3
J2010	演算法	3	5	3
C3020	人工智慧概論	3	10	2
C3501	資料庫系統	3	5	2

學生修課資料庫報表

國立政治大學學期成績單

學年期: 111/1 (Fall Semester 2022)

姓名: 吳青峰 學號: 1301

科目代號	科目名稱	學分	成績
C3501	資料庫系統	3	80.0
J2020	演算法	3	85.0

修習學分: 6 實得學分: 6 平均成績: 82.5

國立政治大學修課名單

學年期 : 111/1 (Fall 2022)

科目代號 : C3501

科目名稱 : 資料庫系統

學號	姓名
1101	陳綺貞
1301	吳青峰

=====

人數: 2

有Excel資料嗎？



有喔！



有的話
事情就簡單多了！

ID	hoge	fuga	piyo
1	29	2	AB
2	35	0	XY
3	41	-9	EF
4	98	0	LM

テスト	日付	B1	B2	備考
A	今日	829418	23425	
B	昨日	829415fa		
C	先月末			
D		2312		0
E				
F				
G				
H				
I				
J				
K				
L				
M				
N				
O				
P				
Q				
R				
S				
T				
U				
V				
W				
X				
Y				
Z				



氣死工程師

資料庫

- ◆ 關聯式資料模型 (**Relational Data Model**)
- ◆ 資料庫 (**Database**)
 - 多個資料表的集合
- ◆ 資料表 (**Table**)
 - 多筆紀錄(**Record**)的集合
 - 每筆紀錄都有相同的欄位
- ◆ 欄位 (**Field**)
 - 描述每筆紀錄的屬性 (**Attribute**)
 - **Primary Key**: 可以區別不同紀錄的一組欄位
 - **Foreign Key**: 參考其他資料表的 Primary Key 之一組欄位

學生修課資料庫

Student

學號	姓名	性別	生日	系所
1101	陳綺貞	女	1975/06/06	哲學系
2301	黃奇斌	男	1990/09/10	廣告系
1102	張雨生	男	1966/06/07	外交系
1201	林志玲	女	1974/11/29	資料系
1103	謝馨儀	女	1982/04/16	企管系
1301	吳青峰	男	1982/08/30	中文系
2302	林依晨	女	1982/10/29	韓文系

SC

學生	課程	分數
1101	C3501	90
1102	C3001	70
1102	J2010	80
1103	C3001	100
2301	J3020	85
2301	C3001	90
2302	J2010	70
2302	C3020	80
1301	C3501	80
1301	J2010	85

Course

代碼	名稱	學分	人數限制	選課人數
C3001	資料庫系統	3	10	3
J2010	演算法	3	5	3
C3020	人工智慧概論	3	10	2
C3501	資料庫系統	3	5	2

這資料庫設計
符合實際需求嗎？



大家來找碴！



學生修課資料庫

Student

學號	姓名	性別	生日	系所
1101	陳綺貞	女	1975/06/06	哲學系
2301	黃奇斌	男	1990/09/10	廣告系
1102	張雨生	男	1966/06/07	外交系
1201	林志玲	女	1974/11/29	資料系
1103	謝馨儀	女	1982/04/16	企管系
1301	吳青峰	男	1982/08/32	中文系
2302	林依晨	女	1982/10/29	韓文系

SC

學生	課程	分數
1101	C3501	90
1102	C3001	70
1102	J2010	80
1103	C3001	100
2301	J3020	85
2301	C3001	90
2302	J2010	70
2302	C3020	80
1301	C3502	80
1302	J2010	85

Course

代碼	名稱	學分	人數限制	選課人數
C3001	資料庫系統	3	10	2
J2010	演算法	3	5	3
C3020	人工智慧概論	3	10	2
C3501	資料庫系統	3	5	1

資料庫設計 (Database Design)

- ◆ 資料庫有哪些資料表？
- ◆ 每個資料表有哪些欄位？
- ◆ 每個欄位的資料類型為何？有沒有範圍的限制？
- ◆ 不同資料表的欄位之間有什麼關係？
- ◆ 有沒有協助資料庫設計的方法論 (Methodology)？
- ◆ 資料庫設計對於查詢效能 (Efficiency) 有影響嗎？

Data Models

- ◆ Relational Data Model
 - Relational Data Structures
 - Relational Constraints
 - Relational Algebra
- ◆ Entity-Relationship Model
 - Entities
 - Relationships between Entities
- ◆ ...

Relational Data Models

◆ Relational Data Structures

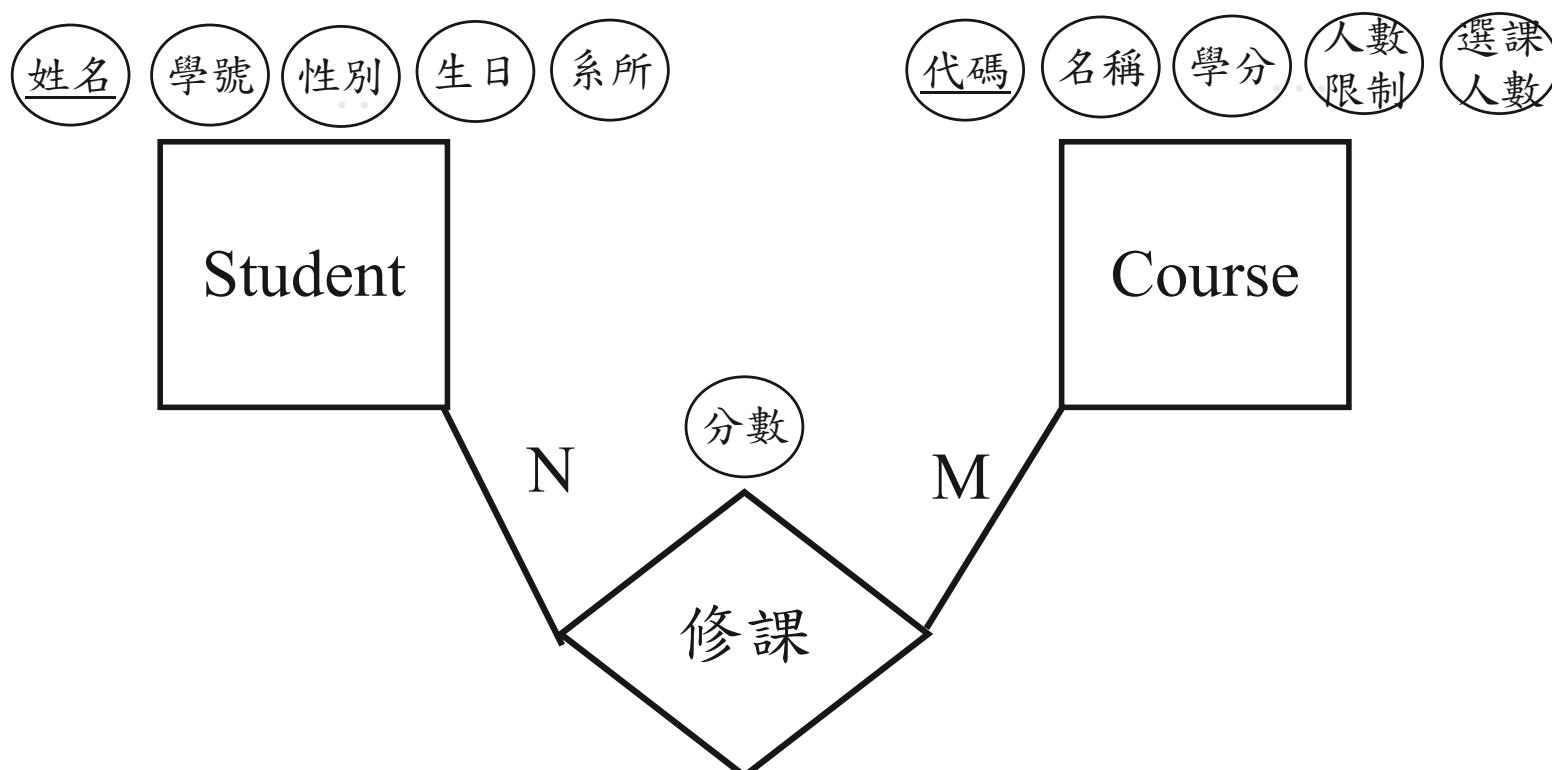
- Student (學號, 姓名, 性別, 生日, 系所)
- Course (代碼, 名稱, 學分, 人數限制, 修課人數)
- SC (學生, 課程, 分數)

◆ Relational Constraints

- Key Constraints: 每個 Table 一定要有 Key
- Domain Constraints: 每個欄位的值必須在值域範圍內
- Entity Integrity Constraints: Primary Key不能是 Null
- Referential Integrity Constraints: Foreign Key

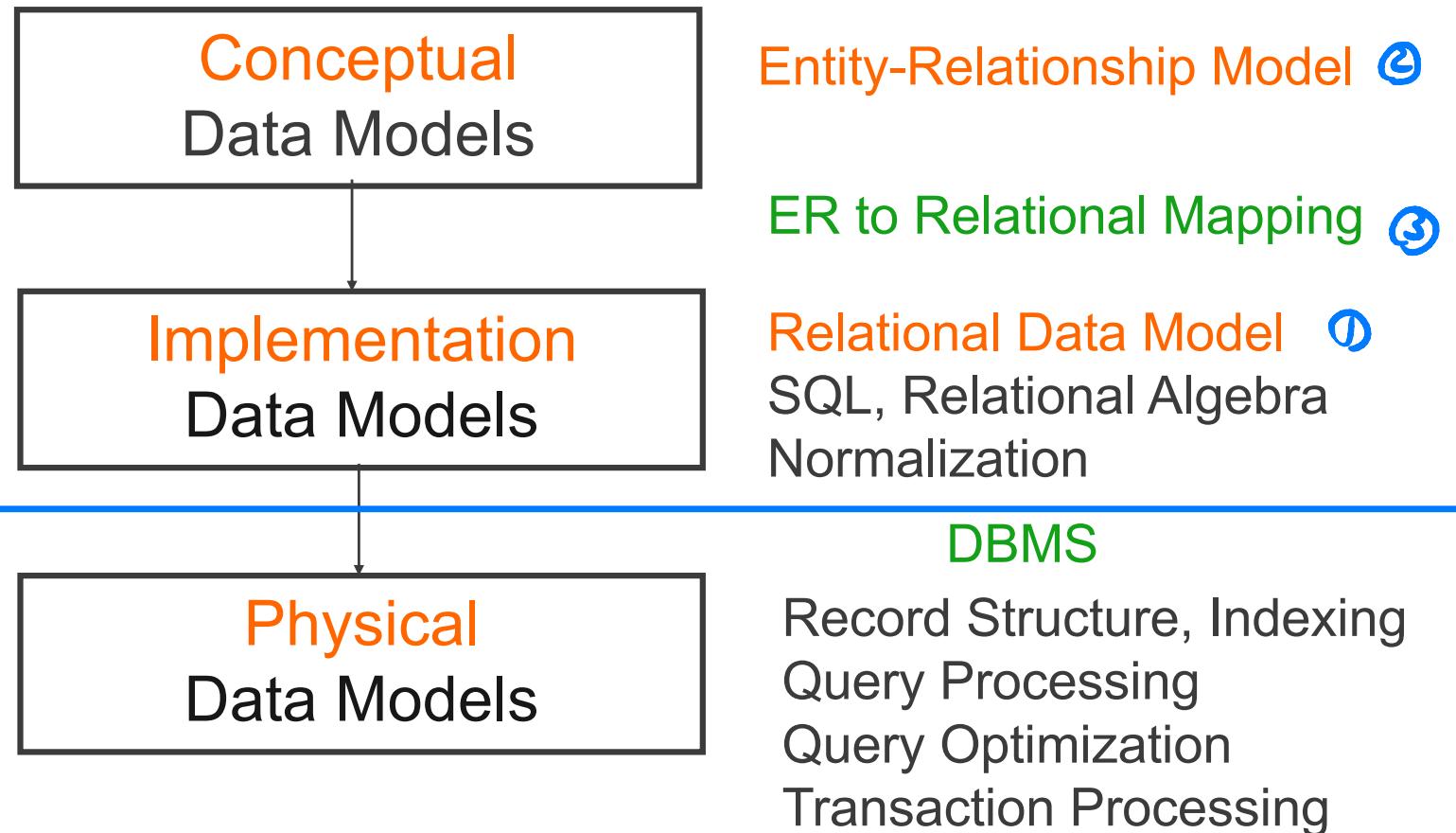
Entity-Relationship Model

- ◆ Entities with Attributes
- ◆ Relationships (between Entities) with Attributes



Data Models

上半學期
下半學期



資料庫如何使用？



資料庫系統的使用

- ◆ **SQL** (Structure Query Language)

- Select 代碼, 名稱

- From Course

- Where 學分=3

- ◆ **QBE** (Query By Example)

- ◆ Embedded SQL + 高階程式語言 = 資料庫應用系統

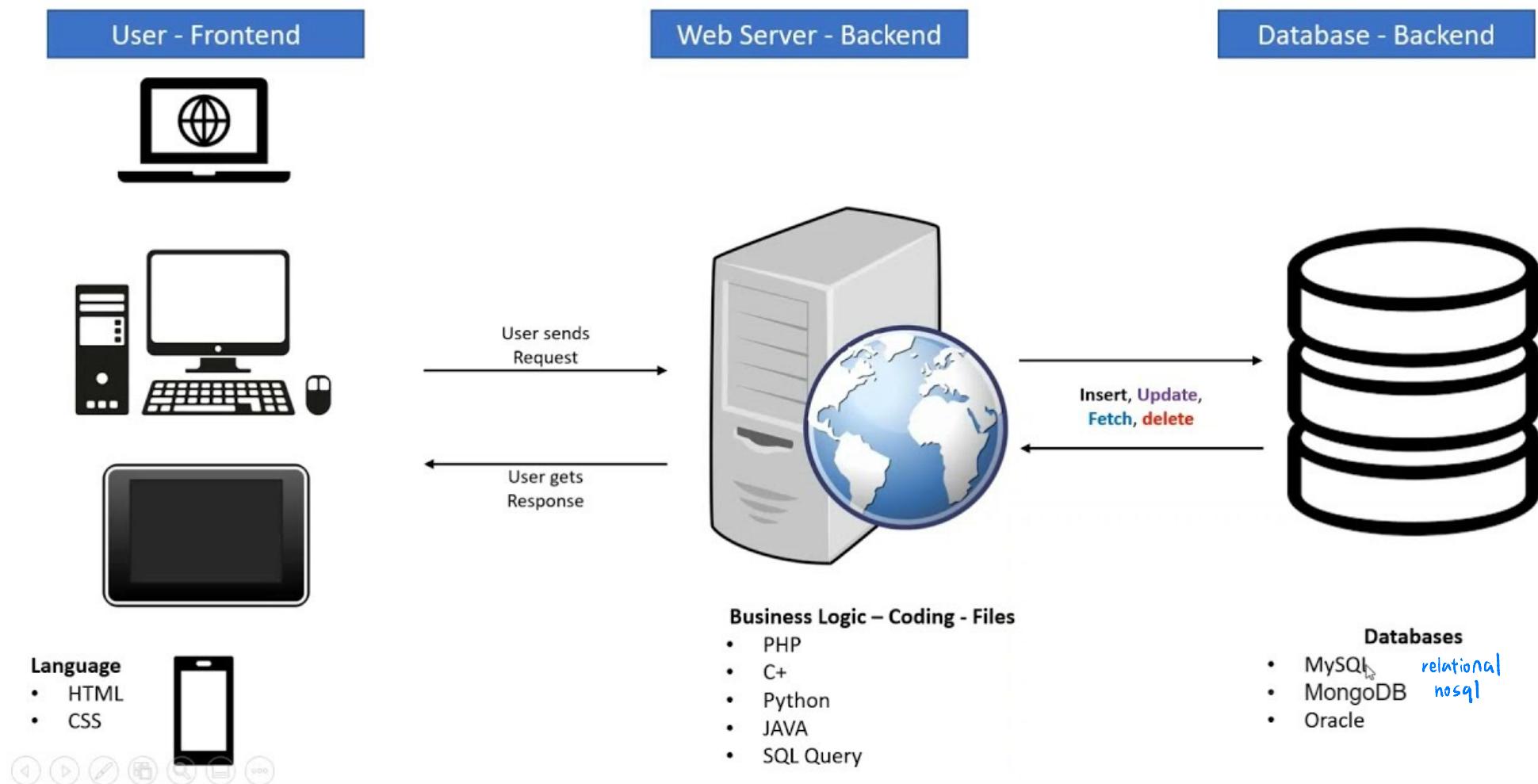
a visual method for retrieving data

資料庫查詢語言 SQL

- ◆ 資料庫的程式語言？
- ◆ 資料庫查詢語言背後有理論基礎嗎？
- ◆ 如何設計資料查詢？
- ◆ 如何設計高效率的查詢？

資料庫應用系統：3-Tier架構

Data Flow in Software



為什麼要分開多個資料表？



學生修課資料庫

Student

學號	姓名	性別	生日	系所
1101	陳綺貞	女	1975/06/06	哲學系
2301	黃奇斌	男	1990/09/10	廣告系
1102	張雨生	男	1966/06/07	外交系
1201	林志玲	女	1974/11/29	資料系
1103	謝馨儀	女	1982/04/16	企管系
1301	吳青峰	男	1982/08/30	中文系
2302	林依晨	女	1982/10/29	韓文系

SC

學生	課程	分數
1101	C3501	90
1102	C3001	70
1102	J2010	80
1103	C3001	100
2301	J3020	85
2301	C3001	90
2302	J2010	70
2302	C3020	80
1301	C3501	80
1301	J2010	85

Course

代碼	名稱	學分	人數限制	選課人數
C3001	資料庫系統	3	10	3
J2010	演算法	3	5	3
C3020	人工智慧概論	3	10	2
C3501	資料庫系統	3	5	2

正規化 (Normalization)

學生	姓名	性別	生日	系所	課程	名稱	學分	分數
1101	陳綺貞	女	1975/06/06	哲學系	C3501	資料庫系統	3	90
1102	張雨生	男	1966/06/07	外交系	C3001	資料庫系統	3	70
1102	張雨生	男	1966/06/07	外交系	J2010	演算法	3	80
1103	謝馨儀	女	1982/04/16	企管系	C3001	資料庫系統	3	100
2301	黃奇斌	男	1990/09/10	廣告系	J3020	人工智慧概論	3	85
2301	黃奇斌	男	1990/09/10	廣告系	C3001	資料庫系統	3	90
2302	林依晨	女	1982/10/29	韓文系	J2010	演算法	3	70
2302	林依晨	女	1982/10/29	韓文系	C3020	人工智慧概論	3	80
1301	吳青峰	男	1982/08/30	中文系	C3501	資料庫系統	3	80
1301	吳青峰	男	1982/08/30	中文系	J2010	演算法	3	85

正規化 (Normalization)

- ◆ 資料庫設計
- ◆ 正規劃：decomposition 資料表的過程
 - 1st Normal Form
 - 2nd Normal Form
 - 3rd Normal Form
 - Boyce-Codd Normal Form
 - 4th Normal Form
 - 5th Normal Form

PART 2

資料庫管理系統設計

Database Management System
Design

管理資料庫的軟體 ?



資料庫系統設計 (DBMS Design)

- ◆ 資料庫管理系統(DataBase Management Systems, DBMS)
- ◆ 資料庫管理系統 如何管理資料庫 (DB) ?
 - 資料庫如何儲存在輔助記憶體 (**Storage Management**) ?
 - DBMS 如何處理查詢 (**Query Processing**) ?
 - 資料儲存方式對於效能的影響為何 ?
 - 如何加快查詢速度 (**Indexing, Query Optimization**) ?
 - 新增、讀取、修改、刪除 (**CRUD**) 對效能的影響為何 ?
 - DBMS 可以調校 (**Database Tuning**)，以提升效能嗎 ?

為什麼會發生重複劃位的情形？



交易處理 (Transaction Processing)

- ◆ 交易進行到一半，系統突然當機，資料庫資料會不會出錯 (Recovery) ?
- ◆ 多工併行的交易，訂票系統後端的資料庫會不會重複劃位 (Concurrency Control) ?

Transaction

- ◆ Transaction
 - A logical unit of DB processing
 - Each transaction includes one or more DB access operations (insertion, deletion, modification, retrieval operations)
 - e.g. 轉帳: 從帳號X轉帳N元到帳號Y

read(X);

X=X-N;

write(X);

read(Y);

Y=Y+N;

write(Y);

Interleaved Processing

- ◆ Modern computer
 - time-sharing system
 - Interleaved processing

read(X);
X=X-N;
write(X);
read(Y);
Y=Y+N;
write(Y);

read(X);
X=X+M;
write(X);



Serial
processing

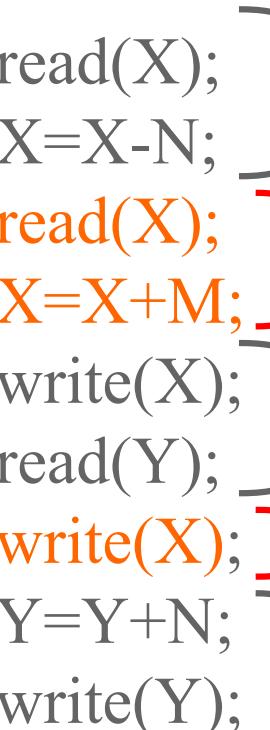
read(X);
X=X-N;

read(X);
X=X+M;

write(X);
read(Y);

write(X);
Y=Y+N;

write(Y);



Interleaved
processing



親愛的客戶大家好：

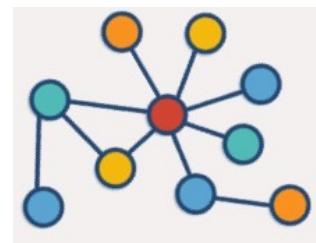
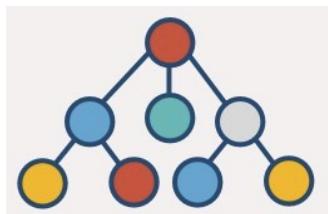
為提供更優質的服務，momo購物網將進行設備維護

造成不便，敬請見諒，感謝您的支持！33

momo 購物網 富邦媒體科技股份有限公司

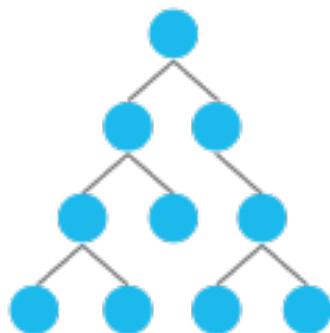
Data

- ◆ Structured data vs. Unstructured Data
- ◆ Text ?
- ◆ Image, Audio, Video content ?
- ◆ Tree ?
- ◆ Graph ?
- ◆ Geographical data ?

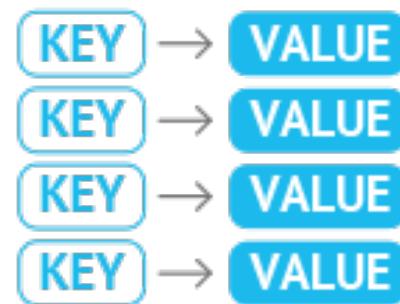


NoSQL (Not Only SQL)

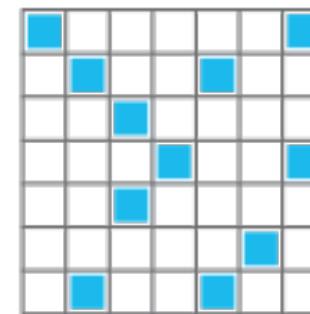
- ◆ Document-based NoSQL Systems
- ◆ Key-value NoSQL Stores
- ◆ Column-based NoSQL Systems
- ◆ Graph-based NoSQL Systems



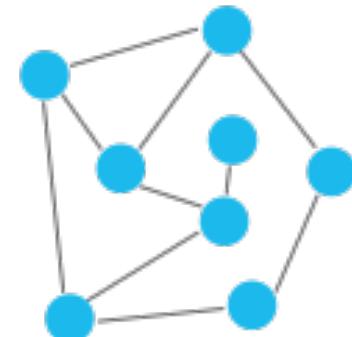
Document



Key-Value



Column-Family



Graph

NoSQL

- ◆ Structured Data vs. Unstructured Data
- ◆ Unstructured Data
 - Document (text)
 - Graph
 - Key-value pairs
 - XML Data (Semi-structured Data)
 - ...

Commercial DBMS

- ◆ Oracle (Oracle Inc., www.oracle.com.tw)
- ◆ Postgres
- ◆ MySQL (MariaDB)
- ◆ SQLite
- ◆ SQL Server 
- ◆ DB 2 
- ◆ Access (Microsoft Office 365)
- ◆



Course Contents

Part 1 Database Design

上半學期

- ◆ Relational Data Model & Relational Database Constraints
- ◆ SQL
- ◆ Relational Algebra & Relational Calculus
- ◆ Entity-Relationship Model
- ◆ ER-to-Relational Mapping
- ◆ Functional Dependencies and Normalization for Relational Databases

Course Content (cont.)

Part 2 DBMS Design 下半學期

- ◆ Disk Storage, Basic File Structures, and Hashing
- ◆ Indexing Structures for Files
- ◆ Algorithms for Query Processing and Optimization
- ◆ Practical Database Design and Tuning
- ◆ Transaction Processing Concepts
- ◆ Concurrency Control
- ◆ NoSQL (*)

Requirements

- ◆ 5 次作業
 - Database Creation (MySQL)
 - SQL Design (MySQL)
 - ER-model design
 - Relational schema design
 - DBMS Performance Evaluation
- * 可以參考 ChatGPT, 但必須註明修改處
- ◆ 學期專題: 分組資料庫應用系統開發
 - 分組
 - 進度報告
 - 學期專題報告 & Demo
- * Front-end PL is preferred, but not required

Project Requirements

- ◆ 分組專題 (每組成員 5~7人)
- ◆ 專題目的：團隊合作開發 Web-Based or 單機資料庫應用系統
- ◆ 專題題目：各組自訂
- ◆ 要求
 - Framework + DBMS 自選 (例如 Flask + SQLite)
 - 至少 5 個 Entity Types, 5 個 Relationship Types
 - 系統功能必須具備 CRUD (Create, Read, Update, Delete)
 - 團隊合作分工
 - 團隊合作討論需求分析、ER Model、Relational Schema
 - 每位成員都必須參與程式設計

Project Grading

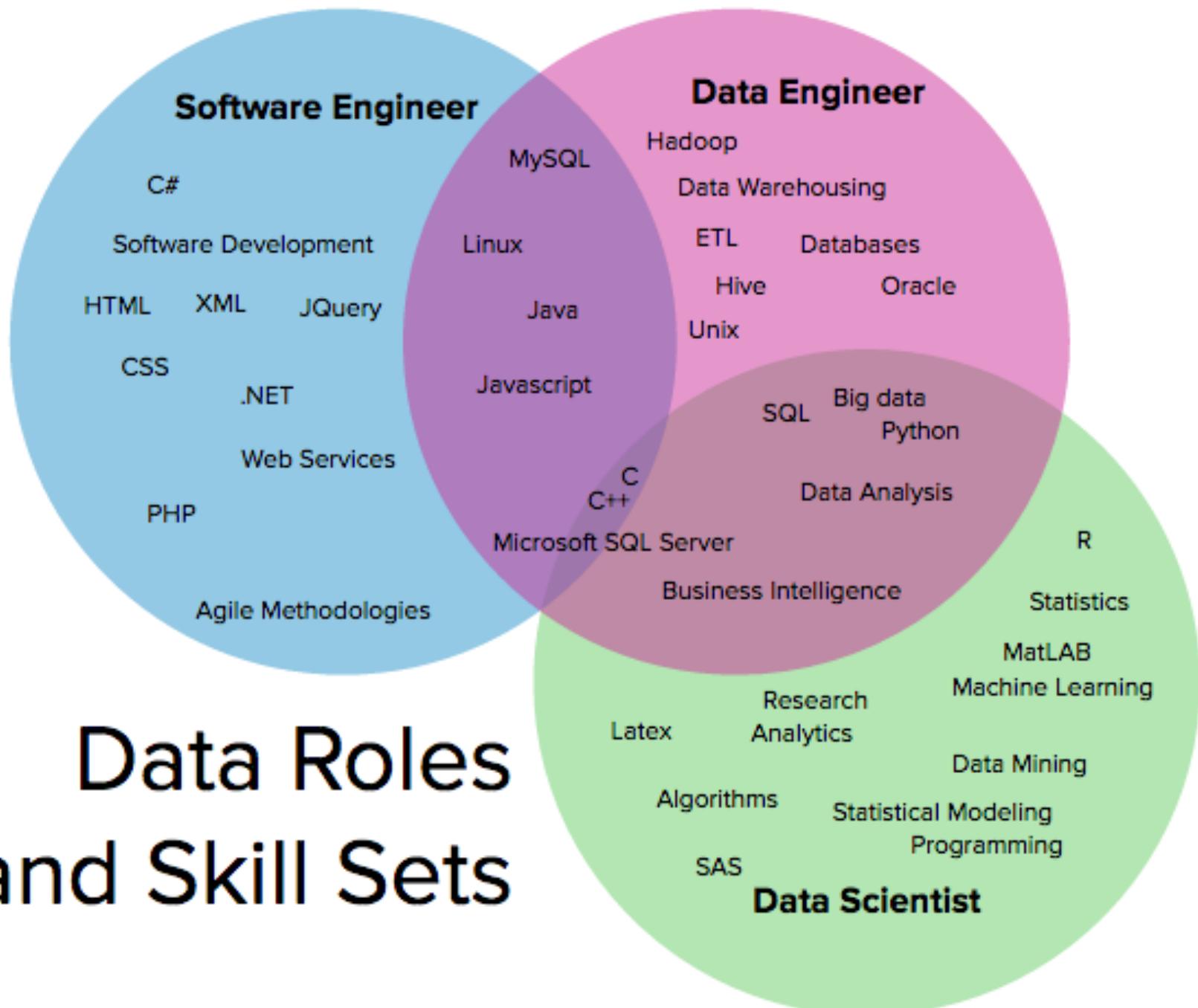
- ◆ 40% Basic Functions (CRUD), Data
- ◆ 30% Data Base Design (Requirement Analysis, ER model, Relational Schema)
- ◆ 10% User Interface (UX/UI, Error-proof)
- ◆ 10% Additional Functions
- ◆ 10% Creativity, Team Work

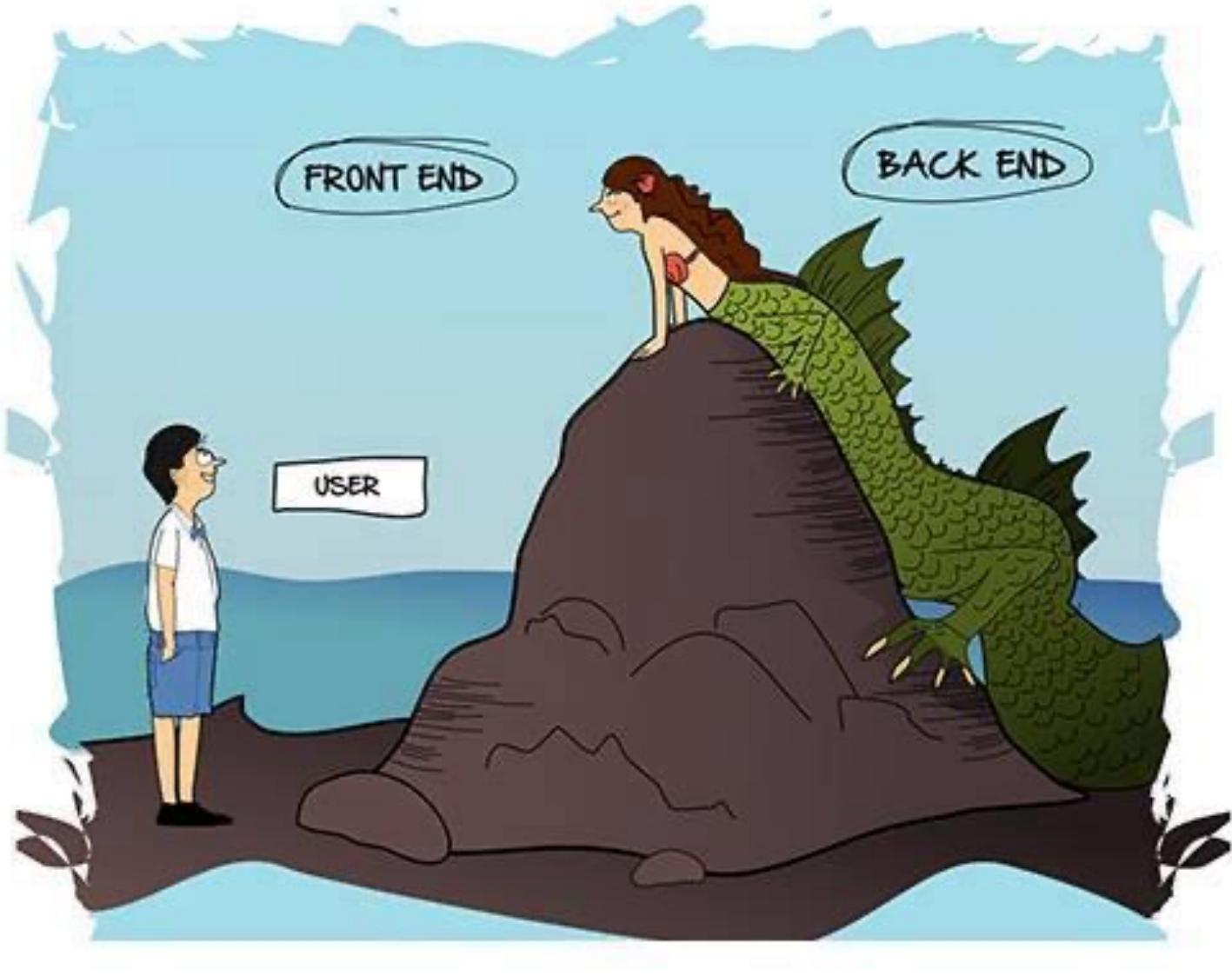
評分標準

- ◆ 評分標準 (Tentative)

- 期中考 : 30~35%
- 期末考 : 30~35%
- 5次個人作業 + 分組學期專題 : 30~40%

Data Roles and Skill Sets





Front end vs. Back end.

作者 blueblueChen (藍藍欠)
標題 [請益] 想問資料庫的問題
時間 Sat Sep 8 23:15:43 2018

看板 Soft_Job

版友晚安，我是代PO

有個朋友想要轉換領域，中文系。想轉戰資料庫。因為她的想法很簡單就是建立一個 table，讓寫程式的人去操作...，然後可以不用碰到程式。

我對資料庫其實也沒很熟，包括大學研所全部都重修過。

而我的經驗是

資料庫可能在 Windows 或者其他 OS 例如 Android，都有支援的套件，然後還是需要C++或者 JAVA 去寫程式邏輯來操作這些資料庫。不可能只有一個 table 表就給其他程式設計師用。

然後這是巨匠課程表，想問資料庫前輩，這些課程對一個外行人來說，有沒有絕對幫助呢？或是這些課程，是目前就業市場不可或缺的呢？畢竟這套課程20萬...

<https://goo.gl/YBcXvW>

作者 littlethe (東周流浪漢)
標題 Re: [請益] 想問資料庫的問題
時間 Mon Sep 10 07:21:01 2018

看板 Soft_Job

HI~

魯叔又來講資料庫和人性啦~

這件事可以分為兩個角度來看，

1. 只是要找到工作
2. 有心把資料庫學好

1. 只是要找到工作 -

如果你朋友是年輕未婚正妹的話，

各位大大你們就別操心了，

因為正妹可以什麼都不會就進到資訊業工作，

幾乎公司都喜歡找正妹當PM或企劃，

2. 有心把資料庫學好 -

講完人性，這個就正題啦！

上一篇大大回文分析得很專業，

我就來講點不一樣的，

有看過足球吧？

DBA就是系統的守門員，

人數很少，但要扛的責任最大，看似簡單，但裡面的水確是最深的，

踢球這件事，就是coding，

大家看足球時，

一定很多人會覺得，

守門員幾乎在用手擋球，

用手比較簡單耶，

所以真要玩足球時，

魯叔不才，
在我目前的人生，從前端，後端，APP，資料庫，甚至接觸過linux和網管，
coding本身真的是最簡單的事了，
不是我在唬人，國中生可以學會coding了，
我是高中就學coding了，
教育部規劃國小學coding，這不無道理呀，國小生學coding是有可能的，
資料庫我不敢說資訊界裡最難，最難應該是AI，
但我敢說資料庫真的很難，
從資工資管畢業的人應該對資料庫這門大刀課很有印象，
一堆人補考還不過，真的很難呀！
不是說開table而已呀，
資料庫是真的要懂理論的東西，
非本科去做資料庫反而是不好做的，

資料庫是真的要懂理論的東西，
非本科去做資料庫反而是不好做的，
光正規化就可以搞死非本科了，
講個實務問題，
如果硬碟滿了，
你的處理方式是什麼？買硬碟嗎？
如果SQL要跑很久怎麼辦？升級電腦嗎？
如果是這樣的回答，那真的不需要你來管資料庫，（還真不少人這樣回答）
老闆自己找時間把主機升級就好了
要寫SQL對吧？
寫SQL其實比一般程式更難哦！因為SQL比較抽象，
不要以為SQL只是寫select * from XXX，
舉例來說如果要把表內容轉字串成a,b,c怎麼寫？

一般程式因為速度很快,所以不精簡還可以感覺不出來,
寫個1加到1百萬,一般人是感覺不到用公式寫和用迴圈寫的差別,真的太快了,
但資料庫效能好不好差太多了,
好SQL和差SQL跑的時間是可以差到幾分鐘,甚至差SQL是跑不出來的,

再講資料庫最恐怖的地方,就是責任TMD有夠重大,
就像醫生開刀,機長開飛機一樣,壓大很大!
出事的話,無法休息,
因為數據都在這,
系統都要用,
資料庫出事系統就不能用,
而資料庫就是非常非常非常容易出事,
因為是中心點,很多問題發生後會往中心跑,
就像足球會一直踢向球門一樣

你朋友知道什麼叫做"資料錯亂"嗎?

"資料錯亂"是資料庫最恐怖又很常見的狀況之一，

因為有bug,或有人資料導錯了,寫入的數據就會不對了，

那一錯往往不會只錯一筆,一錯可能會錯幾萬筆,甚至錯上百萬筆,

想像一下,我們台灣兩千三百萬人一覺醒來,發現我們的身分證都錯了,

你若是DBA,就做到死也得把2300萬人的身份證給更正好,

有些狀況下,如果原記錄是寫在紙上的話,可能會要一筆一筆的去對,

對到眼睛脫窗還對不完,這可不像一般工程師代碼寫錯了,改完就好啦,

因為這些是"數據",而且常常一錯會錯一大片,

你朋友若遇到"資料錯亂"的話,大概會哭著喊說老娘不要管DB了,

coding真的比較簡單輕鬆...

這裡我還沒講到被駭的狀況哩

一般來說,非本科想轉行來當技術人員,
我會建議走前端程式,
因為前端很雜,所以理論不是那麼重要,可以靠經驗去補,
不少設計轉程式的人也是走前端,
越往後端,效能就越吃重,數學就越重要,理論也就越重要,
本科和天份的優勢也就越大了,
真的不想寫程式的話,走網管也行,網管主要靠經驗,本科沒什麼優勢,
但很多網管也是會有"寫程式"的時候,
反正不應該是資料庫,DBA是連工程師都很難當了,
那你朋友的心態...是真的建議不要來當技術人員了,
真的就是不想踢球就想要玩足球的感覺,
那學前面講的做法,去用姿色混個PM還比較好,

上完之後，考完證照...注意注意，
你只是從一般工程師，
變成會用資料庫的工程師，
你還是工程師，還是要coding，
證照只是讓你有機會"管理"DB的藉口，
等幹了雙棲工程師幹了3年，
大概才會對資料庫比較了解，因為很多實務狀況，
上課時不會講到或無法體會，
最好是去大公司的資訊部或做後端工程師練，
因為這類工作接觸DB的機會會很頻繁，
有些工程師缺是很少碰到DB，例如APP或前端工程師，做了就和DBA無緣了...
所以選對工作內容很重要，
練了3年真的對維護DB或SQL很熟了後，才會有DBA面試的機會，



SQL Server 實戰效能調校首部曲：TSQL 優化技巧大公開

帶你深入淺出TSQL 優化技巧

原價 \$8800



SQL Server 實戰效能調校第二部曲：資料表和索引設計一點訣

資料表和索引設計一點訣

原價 \$8800



SQL Server 實戰效能調校第三部曲：決戰執行計畫

決戰執行計畫

原價 \$8800



SQL Server 實戰效能調校第四部曲：縱 橫交易處理

縱橫交易處理

原價 \$8800



RDBMS 資料庫設計討論營 - 資料庫淺談

此次分享會為隨講，主要目的為社群交流，所以不會太正式，會多些互動。我每年為了回饋社群，都會舉辦類似的免費活動，所以請帶著一顆輕鬆、交流的心前來即可。

| 適合會眾

- 想瞭解資料庫開發者或從業人員都在討論什麼。
- 想瞭解資料庫技術選型，例如何時應該採用什麼資料庫。
- 想深入瞭解 MySQL 及 PostgreSQL，可能適時補充 Redis, MongoDB, Elasticsearch。
- 想擺脫被稱為 CRUD 工程師的困擾。

| 內容摘要

- 學習資料庫的原因是什麼？
- 為什麼 ID 設計及索引這麼重要？
- 為什麼 Schema 設計應該放在第一位？
- MySQL 及 PostgreSQL 資料庫應該怎麼選？
- MySQL 及 PostgreSQL 及各資料庫的法律授權有什麼差別？
- MySQL 及 PostgreSQL 近年新的重點特色有什麼？
- 你確定 Prepared Statement 能防止 SQL Injection？
- Prepared Statement 及一般 Statement 的差異你知道多少？
- Connection Pool 是什麼，何時該用？
- 為什麼資料庫與 HDD / SSD 息息相關？SSD 真的快嗎？
- 為什麼資料庫與檔案系統息息相關？我該用 Ext4 還是 XFS？
- 如何最佳化 MySQL / PostgreSQL 參數設定？
- 原來作業系統的參數最佳化，能夠 MySQL / PostgreSQL 更快，但如何做？
- 為什麼學好 MySQL / PostgreSQL 也差不多學好 NoSQL / NewSQL 了？
- 為什麼學好 MySQL / PostgreSQL 對大數據處理的能力會有所提升，即使你的大數據沒有用到 MySQL / PostgreSQL？
- 為什麼學好 MySQL / PostgreSQL，你對作業系統的 Process 及 Thread 的認知能更進一步？
- 為什麼我說大部分人的 Database Migration 都做錯了？

-  Cateyes Lin 我難過...沒搶到
讚 · 回覆 · 6小時
-  Jimmy Tien 沒搶到
讚 · 回覆 · 5小時
-  戴屏川 瞬間就沒票了 😭
讚 · 回覆 · 5小時
-  LienFa Huang 啊啊，來不及 ...
讚 · 回覆 · 5小時
-  邱政憲 錯過了 😳
讚 · 回覆 · 4小時
-  王慕羣 第一分鐘進去，買完就已經是 19 號。應該三分鐘賣光吧 XD
讚 · 回覆 · 4小時
-  Ant Ant Yi-Feng Tzeng 負責報名 Duncan 說最後一張票顯示的是第九秒 Q_Q
讚 · 回覆 · 2小時
-  王慕羣 神！
讚 · 回覆 · 2小時
-  King Yan 難過，賣光了
讚 · 回覆 · 4小時
-  陳典佑 看到時 已經沒有了
讚 · 回覆 · 4小時

首部曲

TSQL 優化技巧大公開

改善系統調效性能，首先得先寫出高效能TSQL！

內容：生命週期、參數化行為、TSQL改寫法、優化技巧關鍵報告

二部曲

資料表和索引設計一點訣

改善索引與資料表設計，得到真正符合需求的資料庫！

內容：資料表設計和應用、叢集索引、從效能角度談索引設計

三部曲

決戰執行計畫

正確的看懂執行計畫，快速發現問題並找到切入點！

內容：查詢器概述、SSMS 再進化、Join演算法分析

四部曲

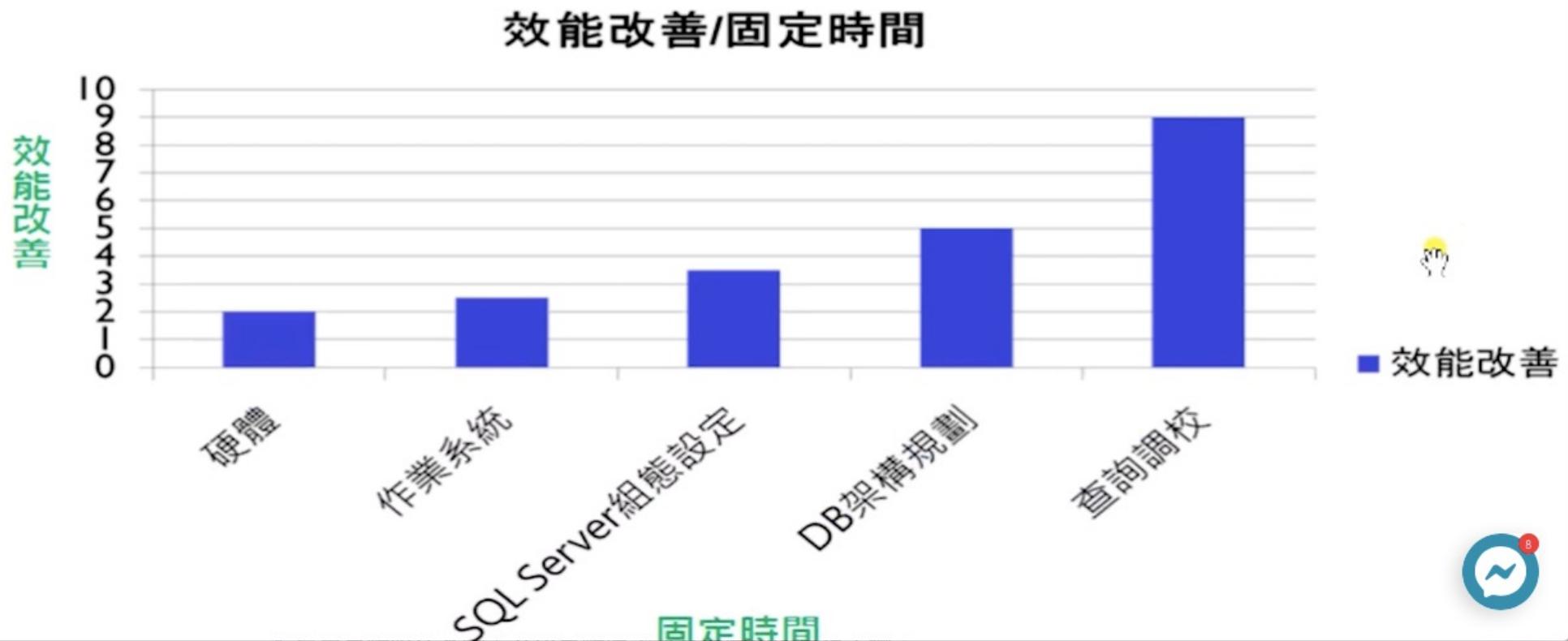
縱橫交易處理

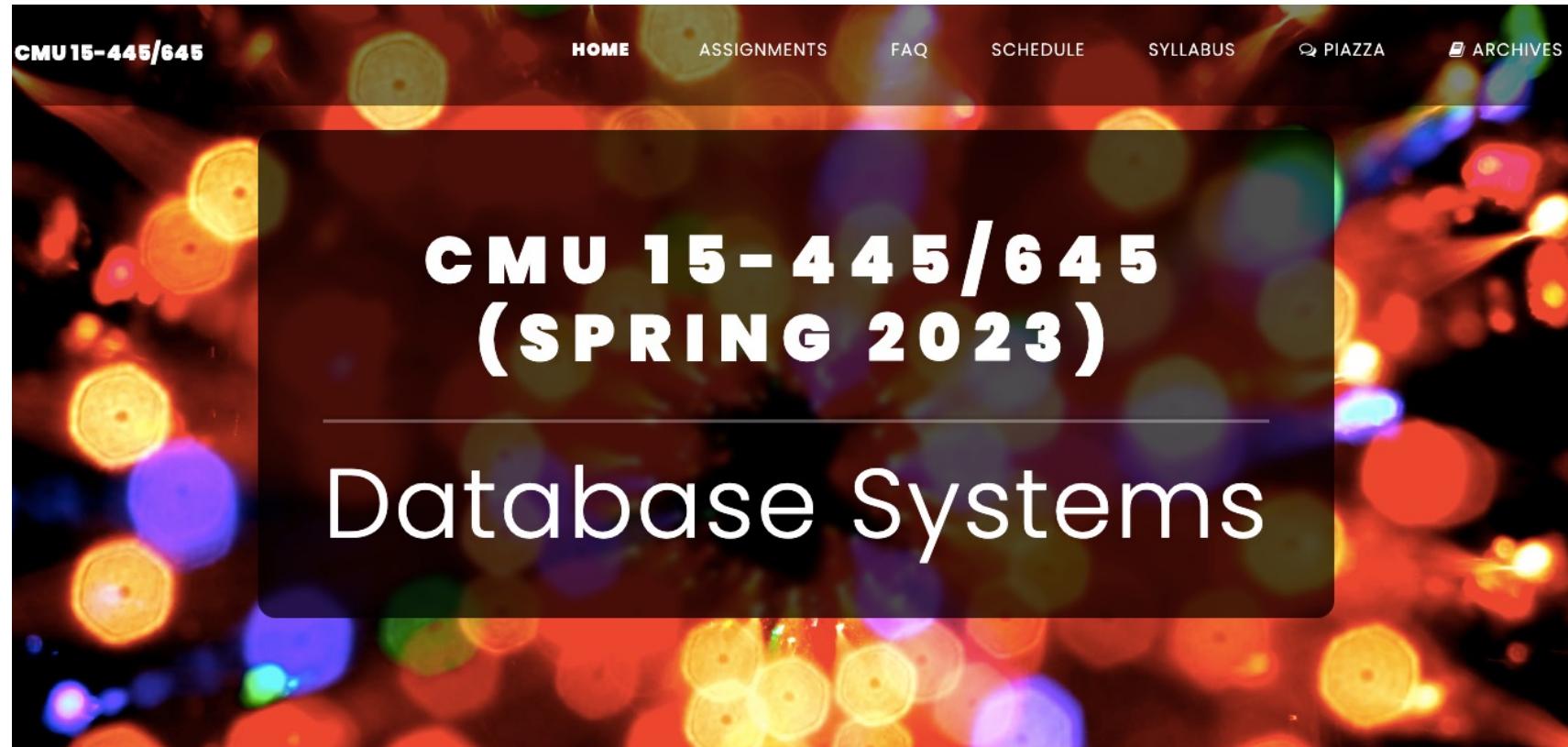
讓系統有效率進行交易處理，同時避免封鎖與資料不一致性！

內容：交易隔離層級介紹、In-Memory OLTP、交易效率加速器



時間成本 VS 效能改善





The image shows a screenshot of a website for a CMU course. The header features a dark background with colorful, blurred bokeh lights. On the left, the text "CMU 15-445/645" is displayed. The top navigation bar includes links for "HOME", "ASSIGNMENTS", "FAQ", "SCHEDULE", "SYLLABUS", "PIAZZA", and "ARCHIVES". The main title "CMU 15-445/645 (SPRING 2023)" is centered in large white font. Below it, the course name "Database Systems" is displayed in a large, white, sans-serif font. A thin horizontal line separates the title from the course name.

Date	Lecture Topic	Slides	Notes	Video
Aug 30	Lecture #01: Course Introduction and the Relational Model 📖 Readings: Chapters 1-2, 27			
Sep 01	Lecture #02: Advanced SQL 📖 Readings: Chapters 3-5 📣 Homework Released: SQL			
Sep 06	Labor Day – No Class			
Sep 08	Lecture #03: Database Storage I 📖 Readings: Chapter 12.1-12.4, 13.2-13.3			
Sep 13	Lecture #04: Database Storage II 📣 Project Released: Buffer Pool Manager			
Sep 15	Lecture #05: Buffer Pools 📖 Readings: Chapter 13.2-13.5			
Sep 20	Lecture #06: Hash Tables 📖 Readings: Chapter 14.5, 24.5			

Sep 22	Lecture #07: Trees Indexes	
	Readings: Chapter 14.1-14.4	
Sep 27	Lecture #08: Index Concurrency Control	
	Readings: Chapter 18.10.2	
	Project Released: Hash Index	
Sep 29	Lecture #09: Sorting + Aggregations	
	Readings: Chapter 15.4-15.5	
Oct 04	Lecture #10: Joins Algorithms	
	Readings: Chapter 15.4-15.6	
Oct 06	Lecture #11: Query Execution I	
	Readings: Chapter 15.1-15.3, 15.7	
Oct 11	Lecture #12: Query Execution II	
	Readings: Chapter 22	
	Homework Released: Join Algorithms	
Oct 13	Mid-Term Exam	

Oct 18	Lecture #13: Query Planning & Optimization I	
	Readings: Chapter 16	
Oct 20	Lecture #14: Query Planning & Optimization II	
	Readings: Chapter 16	
	Project Released: Query Execution	
Oct 25	Lecture #15: Concurrency Control Theory	
	Readings: Chapter 18	
Oct 27	Lecture #16: Two-Phase Locking Concurrency Control	
	Readings: Chapter 18.1-18.3, 18.9	
	Homework Released: Concurrency Control	
Nov 01	Lecture #17: Timestamp Ordering Concurrency Control	
	Readings: Chapter 18.5-18.6	
Nov 03	Lecture #18: Multi-Version Concurrency Control	
	Readings: Chapter 18.7-18.8	
Nov 08	Lecture #19: Logging Protocols + Schemes	

Nov 10	Lecture #20: Crash Recovery Algorithms	  
	 Readings: Chapter 19.1-19.9	
	 Homework Due: Concurrency Control	
Nov 15	Lecture #21: Introduction to Distributed Databases	  
	 Readings: Chapter 20.4-20.5, 21, 23.1-23.4	
	 Project Released: Concurrency Control	
Nov 17	Lecture #22: Distributed OLTP Database Systems	  
	 Readings: Chapter 20.4-20.5, 21, 23.1-23.4	
Nov 22	Lecture #23: Distributed OLAP Database Systems	  
	 Readings: Chapter 20.7, 22.9	
	 Homework Released: Distributed Databases	
Nov 24	Thanksgiving – No Class	
Nov 29	Lecture #24: Guest Speaker: Justin Levandoski (Google BigQuery)	— — —
Dec 01	Lecture #25: Final Review	 — 

Summary

Database Systems =

Database Design + DBMS Design