

# Database Systems

## 2024

### overview

Proadpran Punyabukkana  
**Chula Computer Engineering**

มีครั้งหน้า ...

อ.ธาราทิพย์



อ.ดวงดาว



อ.โซชิติรัตน์



อ.โปรดปران



อ.วิวัฒน์



+ กองทัพพี่ๆ เอ

เรียนยังไงน้าา ...

- 7 January - 17 February: Tuesday and Thursday ONSITE
- **ATTENDANCE CHECK for lectures and activities (15%)**
- Exams
  - Exam 1 (30%)
  - Exam 2 (15%)
  - Exam 3 (20%)
- Assignment (5%)
- Term Project (15%)

my  
**CourseVille**



มาดู syllabus กัน 

มาดู project หน่อย my CourseVille

วันแรกนี่เรียนอะไรในน้าา ...

# To cover

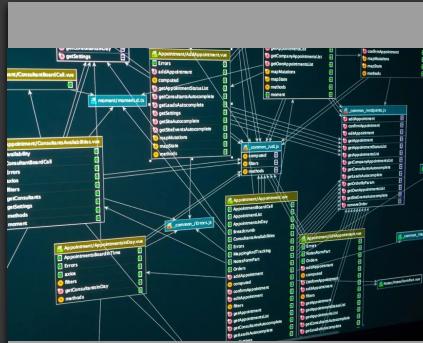
- ★ Know the terminologies
- ★ Data center in Thailand
- ★ History and Trends
- ★ DBMS characteristics
- ★ Relational DBMS
  - Environment
  - Three-schema architecture
  - Model representations
  - Languages
  - Set Theory
- ★ SQL, NoSQL and NewSQL

มาตรฐานคุณภาพกันก่อน  
ห้ามใช้ผิด  
~~อาจ~~ มีสอบเด้อ



## Data

facts and statistics collected.



## Database

An organized collection of data stored and accessed electronically from a computer system.

The data in the same the database should be related and have “entities” along with their “relationships”.

[https://www.kindpng.com/png/hhbRhb1\\_big-data-icon-big-data-analytics-logo-hd/](https://www.kindpng.com/png/hhbRhb1_big-data-icon-big-data-analytics-logo-hd/)

[https://en.wikipedia.org/wiki/Database\\_machine](https://en.wikipedia.org/wiki/Database_machine)

[http://www.tape-storage.net/en/about\\_tape\\_storage/](http://www.tape-storage.net/en/about_tape_storage/)

<https://www.itsguru.com/the-considerations-and-perks-of-choosing-a-dbms/>



## Database machines and storages

**DB Machines or back end processor** is a computer or special hardware that stores and retrieves data from a database. The **storage** may be tape, disks, optical disks, etc.



## Database Management System (DBMS)

A software package designed to store and manage databases.

**DBMS** enables users to define, create, maintain and control access to the database.

## Entities

STUDENT  
FACULTY\_MEMBER  
COURSE  
CLASSROOM

SId, SName,  
SEmail,  
SPhone, ...  
  
CourseNo,  
CourseName,  
Level,  
Description, ...

FId, FName,  
FEmail,  
FPhone,  
FOffice, ...

## Relationships

STUDENT “TAKES” COURSE  
FACULTY\_MEMBER “TEACHES” COURSE  
COURSE “IS\_IN” CLASSROOM  
FACULTY “ADVISES” STUDENT  
  
...  
...  
...  
...

Semester, Year, Grade, ...

# ตัวอย่าง DB ขนาดใหญ่



The screenshot shows a web browser with a dark theme. The address bar at the top right contains a yellow hexagon icon followed by the text "Kostenlose Cloud-Computing-S...". Below the address bar, there are standard navigation icons (back, forward, refresh, home) and a URL field showing "https://www.marketingoops.com/de/free/?trk=ps\_a131L000". The main content area features the AWS logo prominently. Below the logo, the text reads: "Amazon ลงทุน 1.9 แสนล้านเปิด Data Center ในไทย เพิ่มความสามารถในการแข่งขันของประเทศไทย". At the bottom of the page, the source is cited as "MARKETINGOOPS.COM".

Kostenlose Cloud-Computing-S...

https://www.marketingoops.com/de/free/?trk=ps\_a131L000

aws

Amazon ลงทุน 1.9 แสนล้านเปิด Data Center ในไทย  
เพิ่มความสามารถในการแข่งขันของประเทศไทย

MARKETINGOOPS.COM

# เรียกหัวใจด้วย? เปิดซื้อบริษัทเทคโนโลยี ลงทุน Data Center & Cloud Service ในไทย

กี่นา : ล้านบาทจากงบประมาณการต่อสิ่งก่อสร้างในปี 22 พ.ย. 67)

ผู้ให้บริการ

สัญชาติ

มูลค่าลงทุน

 Amazon Web Service (AWS)



200,000 ล้านบาท

 Google



36,000 ล้านบาท

 Microsoft

ยังไม่เปิดเผย  
ตัวเลขลงทุน

 GDS



28,000 ล้านบาท

 Equinix



16,500 ล้านบาท

 NextDC



13,700 ล้านบาท

 CtrlS



5,000 ล้านบาท

 STT GDC



4,500 ล้านบาท

 Evolution Data Center

4,000 ล้านบาท  
(รวมทุกรูปแบบ เช่น เครือข่ายพัฒนา  
และ NT ที่ครอบคลุมทั่วโลก)

 Alibaba Cloud



4,000 ล้านบาท

 Huawei Technologies



3,000 ล้านบาท

 Supernap (Switch)



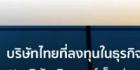
3,000 ล้านบาท

 Telehouse



2,700 ล้านบาท

 One Asia



2,000 ล้านบาท

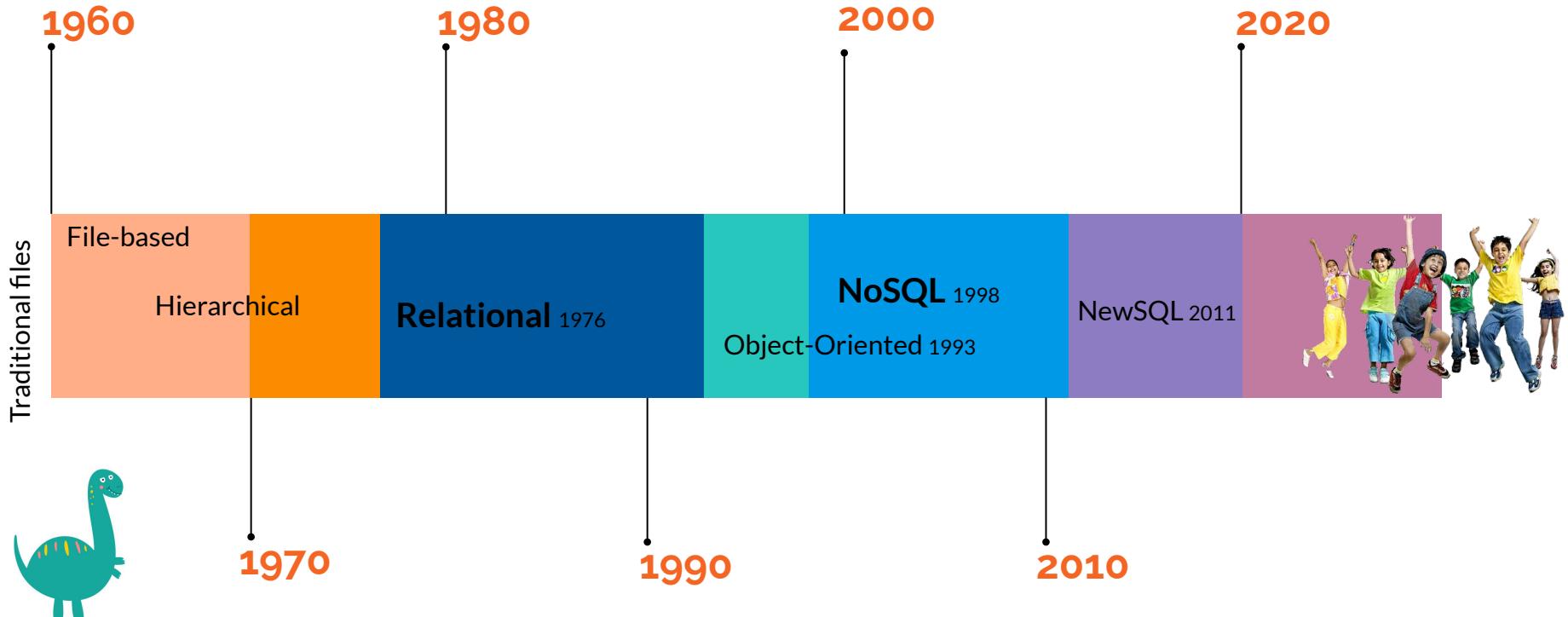
บริษัทไทยที่ลงทุนในธุรกิจ Data Center & Cloud Service

- 1) บริษัท อุบลฯ เป็นบอร์ดประเทศไทย จำกัด (มหาชน) (INET)
- 2) บริษัท ทุก วินาบรุณดา จำกัด (True IDC)
- 3) บริษัท จีเอ็ม ดาต้า เซ็นเตอร์ จำกัด (GSA) รวมทุกรูปแบบ Gulf, Singtel และ AIS



ເລອ db ມັນເກີດມາໄຈ  
ແລ້ວໄປສຶ່ງໃຫນກັນນໍາາ ...

# A brief history of DBMS



# ເວັະ ບຣີ່ຈັກໃໝ່ ၅ ເຂົາໃຊ້ dbms ຕ້ວ່າຫນ ກັນນະ

## DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



423 systems in ranking, January 2025

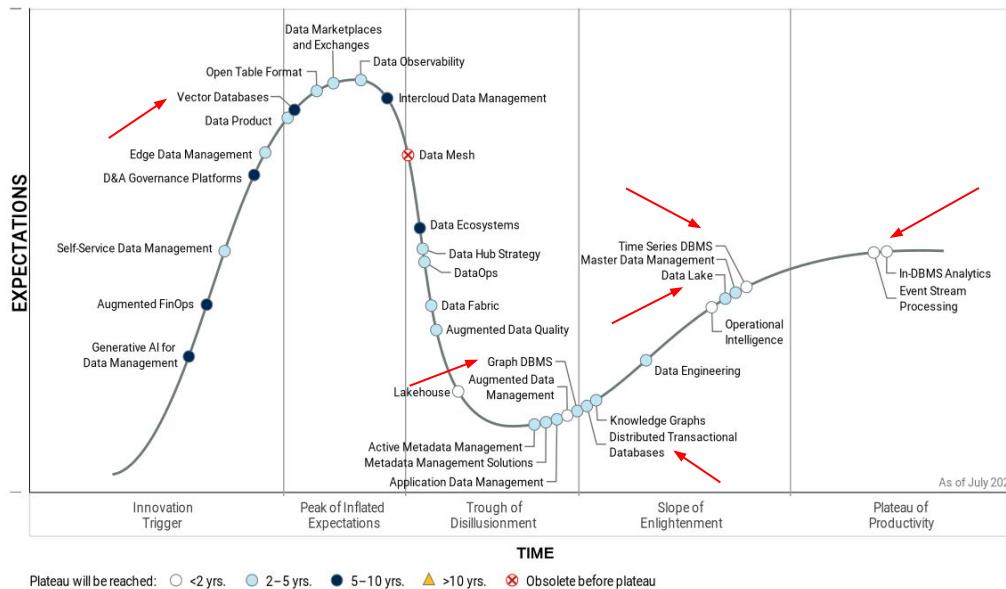
Rank Jan 2025	DBMS	Database Model	Score		
			Jan 2025	Dec 2024	Jan 2024
1.	1. Oracle +	Relational, Multi-model <a href="#">i</a>	1258.76	-5.03	+11.27
2.	2. MySQL +	Relational, Multi-model <a href="#">i</a>	998.15	-5.61	-125.31
3.	3. Microsoft SQL Server	Relational, Multi-model <a href="#">i</a>	798.55	-7.14	-78.05
4.	4. PostgreSQL +	Relational, Multi-model <a href="#">i</a>	663.41	-2.97	+14.45
5.	5. MongoDB +	Document, Multi-model <a href="#">i</a>	402.50	+2.12	-14.98
6.	↑ 7. ↑ 9. Snowflake +	Relational	153.90	+6.54	+27.98
7.	↓ 6. ↓ 6. Redis +	Key-value, Multi-model <a href="#">i</a>	153.36	+3.08	-6.03
8.	8. ↓ 7. Elasticsearch	Multi-model <a href="#">i</a>	134.92	+2.60	-1.15
9.	9. ↓ 8. IBM Db2	Relational, Multi-model <a href="#">i</a>	122.97	+0.19	-9.43
10.	10. ↑ 11. SQLite	Relational	106.69	+4.97	-8.51
11.	11. ↑ 12. Apache Cassandra +	Wide column, Multi-model <a href="#">i</a>	99.19	+1.26	-11.84
12.	12. ↓ 10. Microsoft Access	Relational	92.70	+1.88	-24.97
13.	13. ↑ 17. Databricks +	Multi-model <a href="#">i</a>	87.85	+0.16	+7.31
14.	↑ 15. ↓ 13. MariaDB	Relational, Multi-model <a href="#">i</a>	85.58	+1.81	-13.65
15.	↓ 14. ↓ 14. Splunk	Search engine	83.09	-2.27	-9.63
16.	16. ↓ 15. Microsoft Azure SQL Database	Relational, Multi-model <a href="#">i</a>	73.78	-2.59	-7.29
17.	17. ↓ 16. Amazon DynamoDB +	Multi-model <a href="#">i</a>	73.00	+0.27	-7.94
18.	18. 18. Apache Hive	Relational	56.87	+3.78	-10.08
19.	19. 19. Google BigQuery +	Relational	53.04	+0.75	-10.44
20.	20. ↑ 22. Neo4j	Graph	43.69	+0.62	-4.49

So...

We now focus on both  
**relational & NOSQL**.

# Hype Cycle for Data Management 2024

Hype Cycle for Data Management, 2024



Gartner®

แล้วเจ้า dbms ผสานกันได้มะ

## The end of “1-size-fits-all”

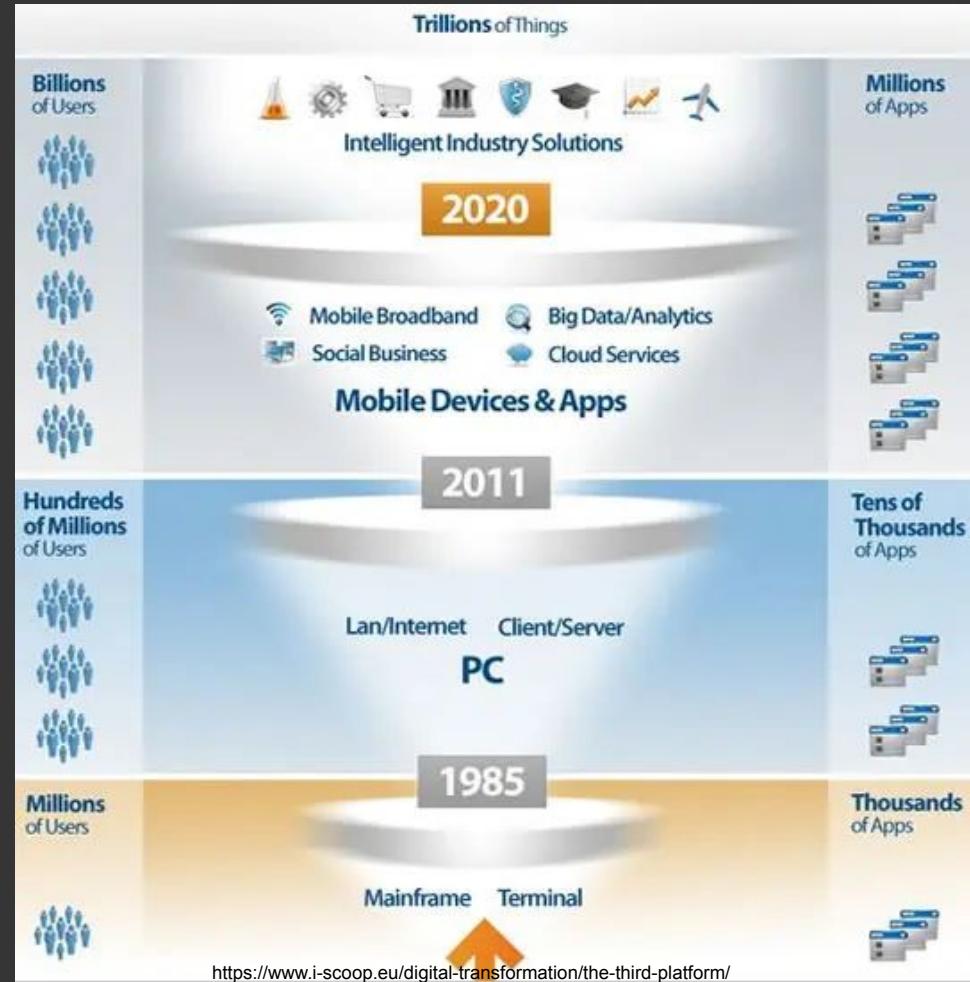
### THE THIRD PLATFORM

is described by International Data Corporation (IDC) as the next generation compute platform that is accessed from **mobile devices**, utilizes **Big Data**, and its **cloud** based.

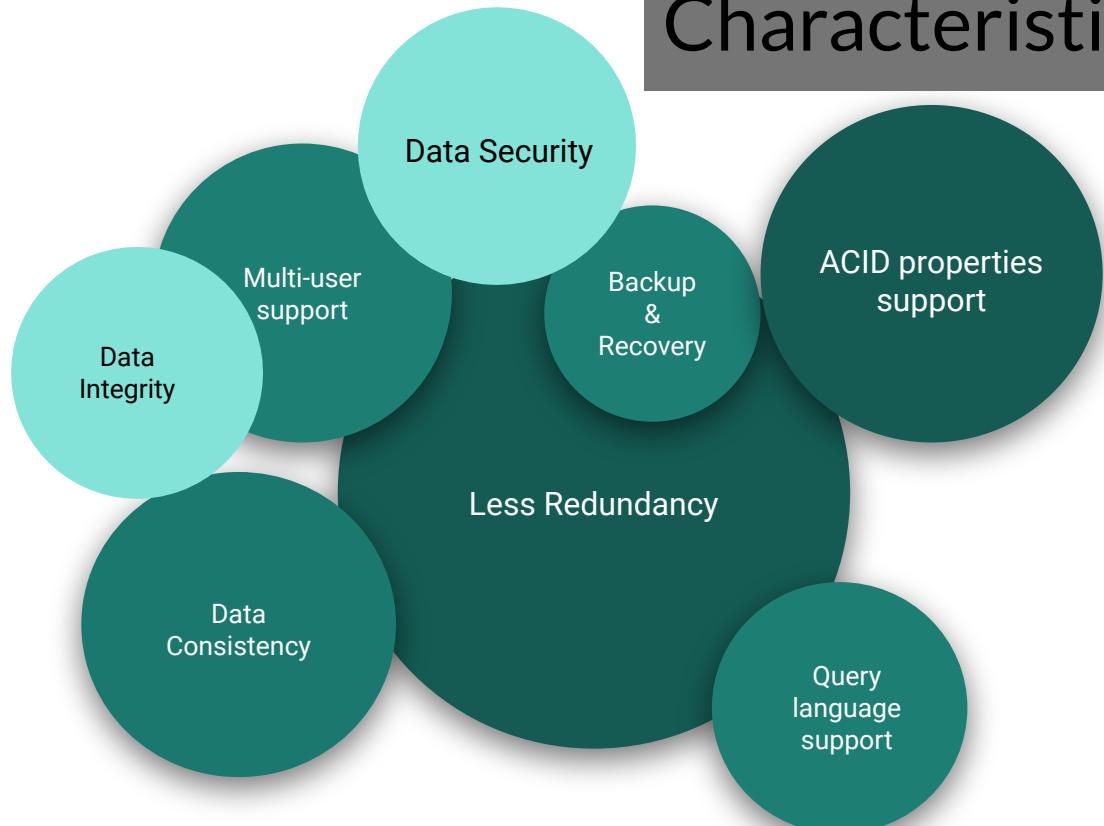
3rd Platform drives new demands on the database:

- Global High Availability
- Data volumes
- Unstructured data
- Transaction rates
- Latency

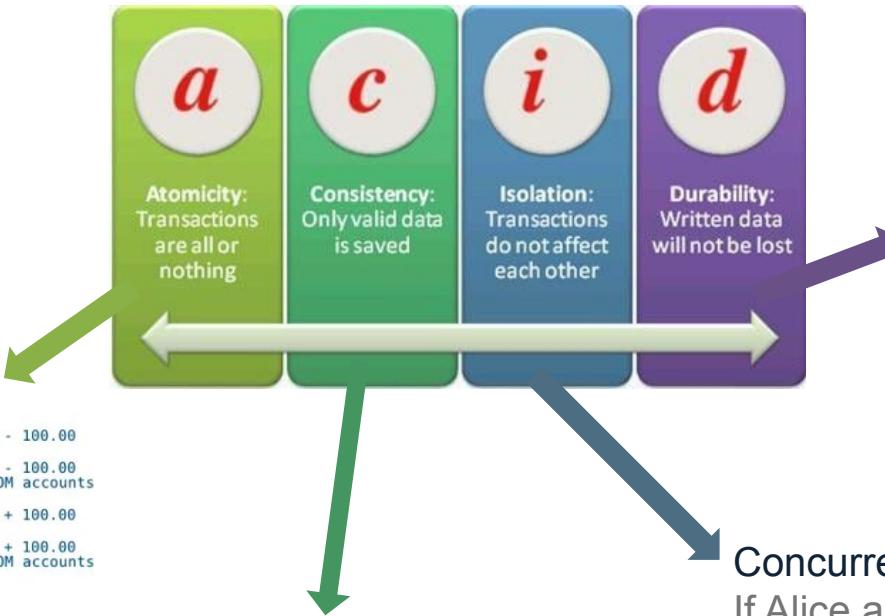
A single architecture cannot meet all those demands



# แล้ว DBMS ดียังไง



# Characteristics of DBMS



```
BEGIN;
UPDATE accounts SET balance = balance - 100.00
WHERE name = 'Alice';
UPDATE branches SET balance = balance - 100.00
WHERE name = (SELECT branch_name FROM accounts
WHERE name = 'Alice');
UPDATE accounts SET balance = balance + 100.00
WHERE name = 'Bob';
UPDATE branches SET balance = balance + 100.00
WHERE name = (SELECT branch_name FROM accounts
WHERE name = 'Bob');
COMMIT;
```

**Rules/Constraints:**  
Do not allow account overdrawn → Alice's balance will never be less than zero.

Data will never be lost:  
Committed transactions are saved to permanent storage, such as disks.

**Concurrency control:**  
If Alice and Bob are withdrawing money at the same time, these two parallel transactions are in reality isolated and seem to be performed sequentially.

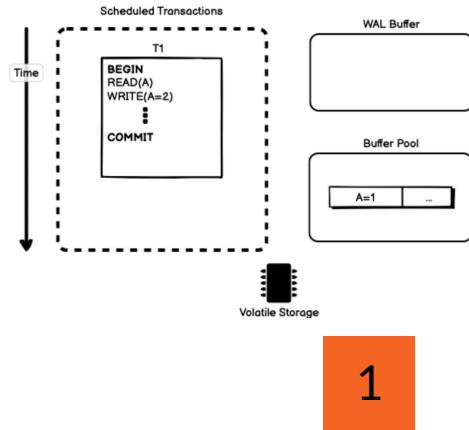
# How are ACID transactions implemented?

The most common implementation of ACID transactions is done via **locks**.

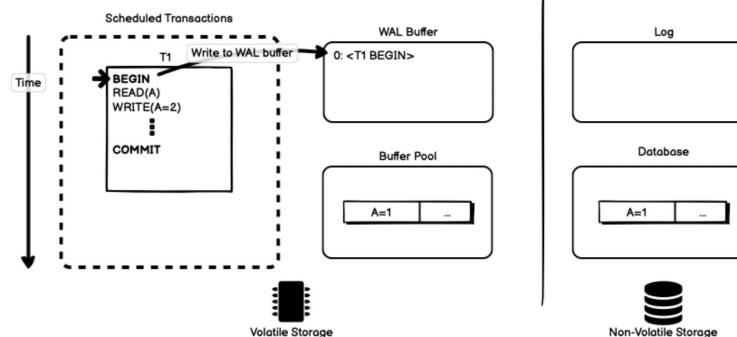
Data is locked until a transaction completes or fails, to guarantee atomicity, isolation, and consistency.

To guarantee durability, databases implement **write-ahead logs**.

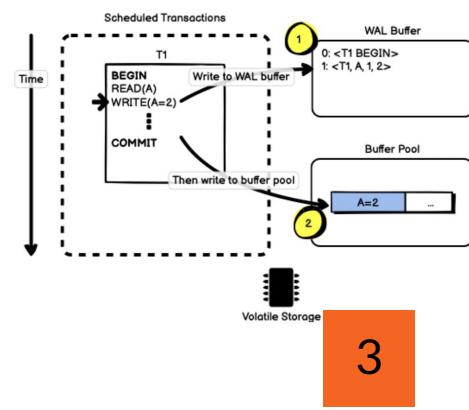
In case of system failure mid-transaction, the transaction is either rolled back or continued from the transaction log left off.



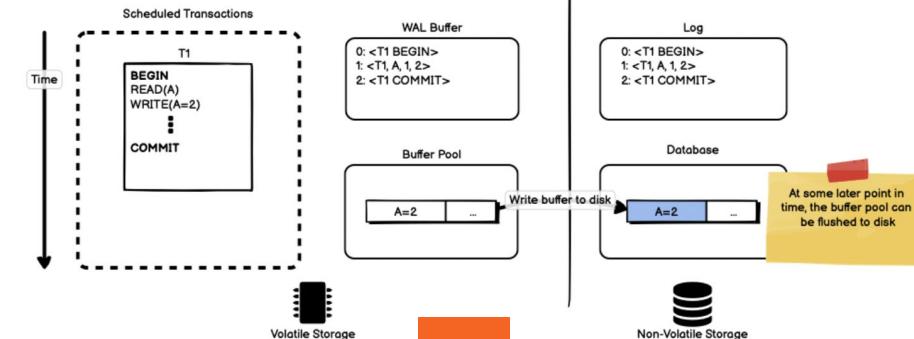
1



2

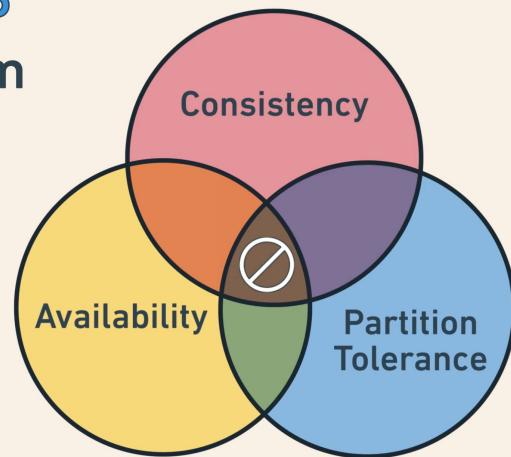


3



4

## The CAP Theorem



### Consistency

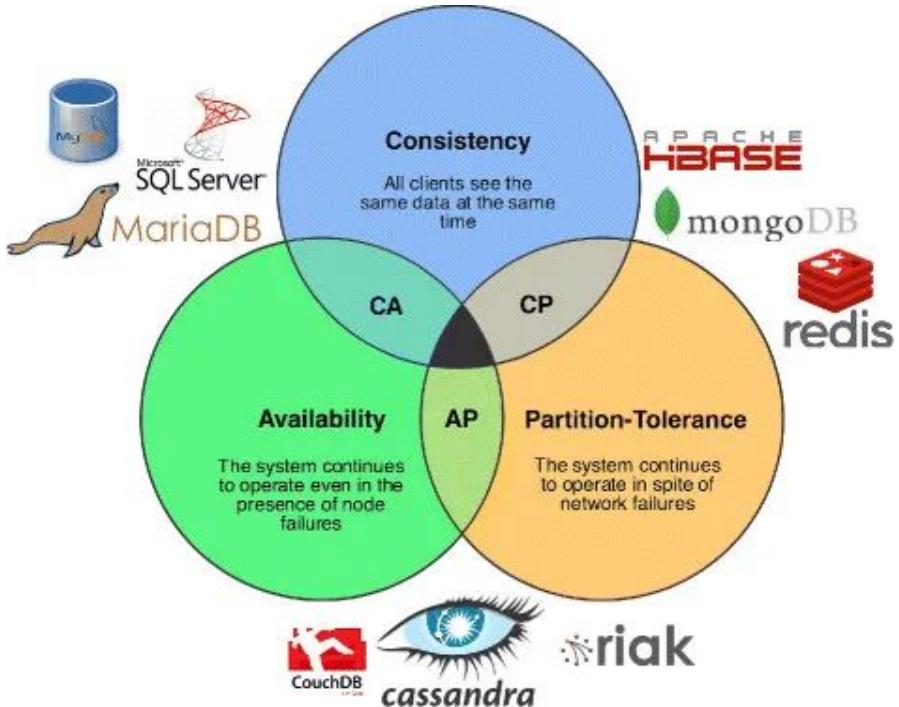
when system returns info, it is always up-to-date

### Availability

system always returns info, even if stale

### Partition Tolerance

system can continue operating during a partition



<https://medium.com/@ibrahimlanre1890/cap-theorem-in-dbms-42027527092e>

## CAP theorem or Brewer's theorem

after computer scientist Eric Brewer, states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:

- **Consistency**: When the system returns info, it is always up-to-date.
- **Availability**: Systems always returns info, even if stale.
- **Partition tolerance**: System continues operating during a partition (eg. network loss)

When a network partition failure happens should we decide to

- Cancel the operation and thus decrease the availability but ensure consistency
- Proceed with the operation and thus provide availability but risk inconsistency

[https://en.wikipedia.org/wiki/CAP\\_theorem](https://en.wikipedia.org/wiki/CAP_theorem)

# เข้าเรื่อง Relational ชีวะที

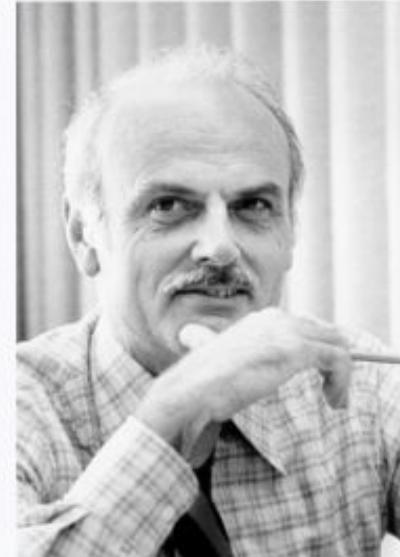


**Edgar Codd**, proposed a new data representation framework called the **relational data model** in 1970.

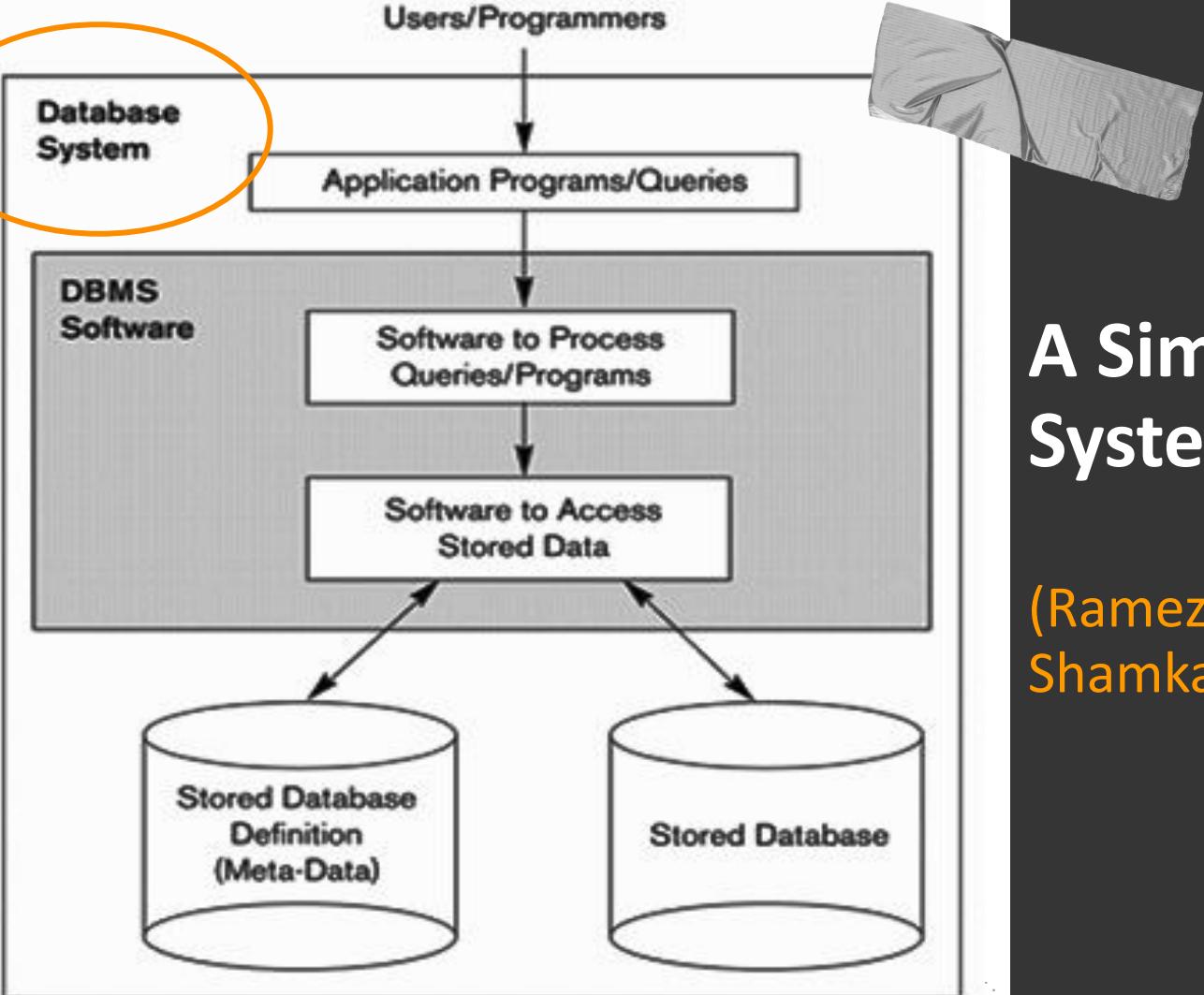
- The model was described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks".
- SQL was one of the first commercial languages to use [Edgar F. Codd's relational model](#).



**Edgar "Ted" Codd**



<b>Born</b>	Edgar Frank Codd 19 August 1923 <a href="#">[1]</a> <a href="#">[2]</a> <a href="#">Fortuneswell, Dorset, England</a>
<b>Died</b>	18 April 2003 (aged 79) <a href="#">Williams Island, Aventura, Florida, USA</a>

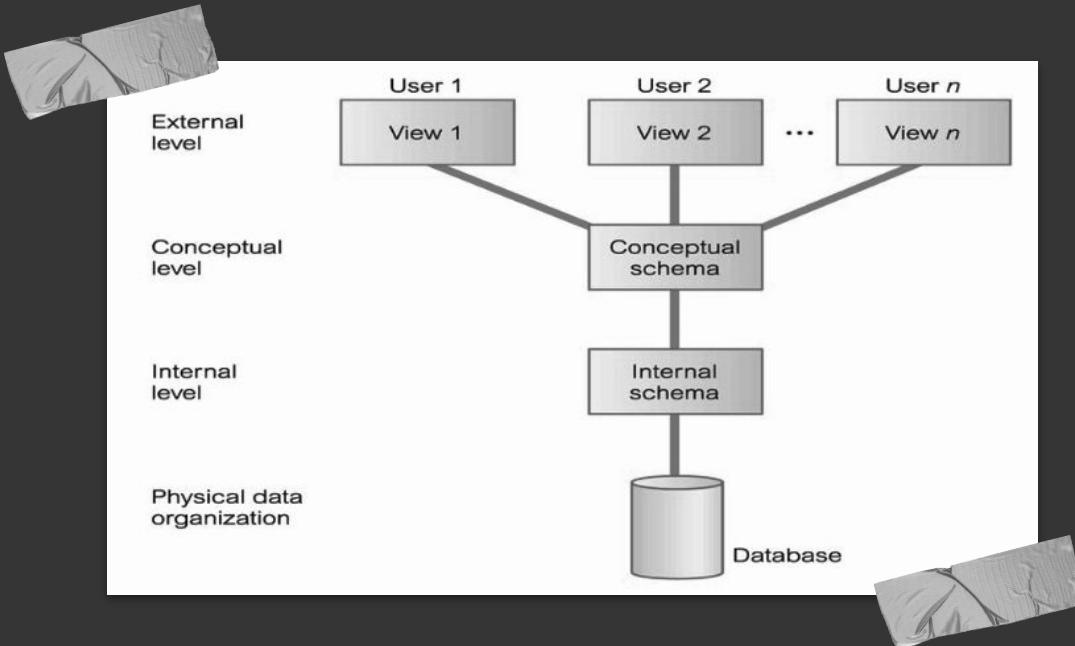


# A Simplified Database System Environment

(Ramez Elmasri and  
Shamkant B. Navathe)

# Level of Abstraction

## Three Schema Architecture



### 1. Internal Level

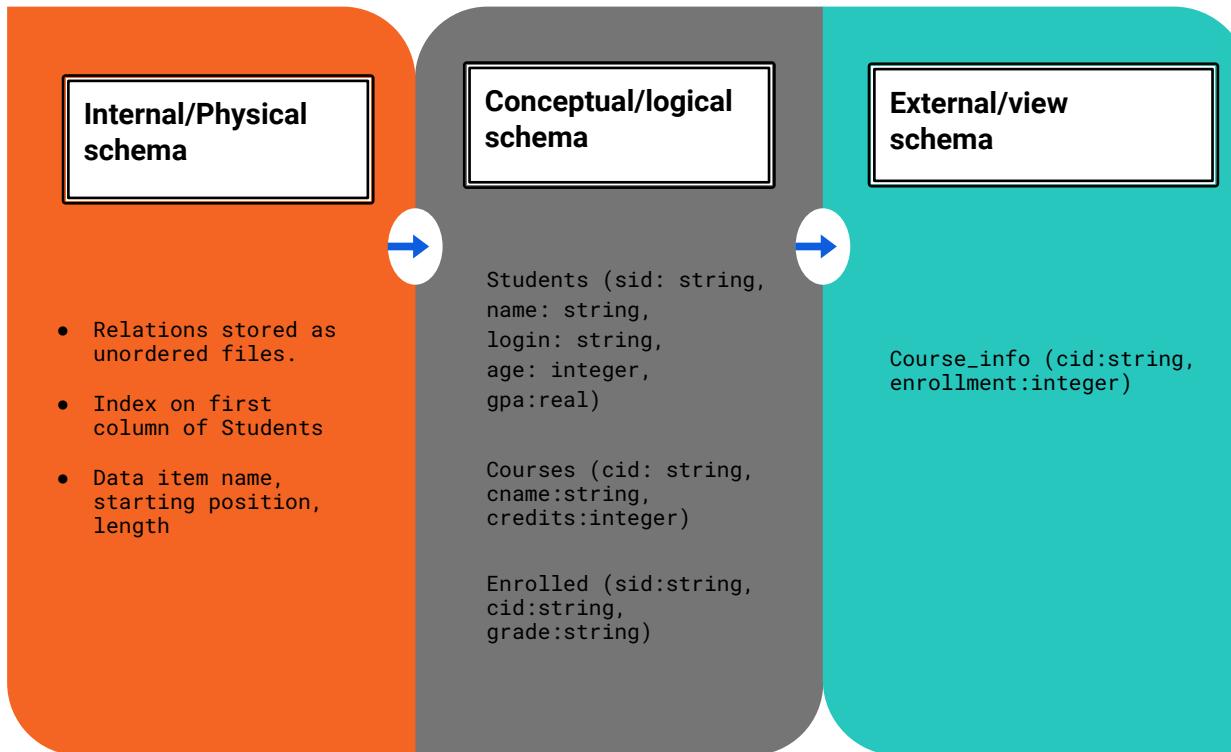
The internal level has an **internal (or physical) schema** which describes the physical storage structure of the database. It uses the **physical data model** to define how the data will be stored in a block to describe complex low-level data structures in detail.

### 2. Conceptual Level

The **conceptual (or logical) schema** describes the design of a database; the structure of the whole database, what data are to be stored in the database, and what relationship exists among those data. In this level, internal details such as an implementation of the data structure are hidden.\* **Programmers and database administrators work at this level.**\*

### 3. External Level

At the external level, a database contains **several external (or view) schemas** that sometimes called **subschema** to describe the different view of the database part that a particular user group is interested and hides the remaining database from that user group.\* **End users only see different views.**\*





employee_id	first_name	last_name	nin	department_id
44	Simon	Martinez	HH 45 09 73 D	1
45	Thomas	Goldstein	SA 75 35 42 B	2
46	Eugene	Comelsen	NE 22 63 82	2
47	Andrew	Petculescu	XY 29 87 61 A	1
48	Ruth	Stadick	MA 12 89 36 A	15
49	Bany	Scardelis	AT 20 73 18	2
50	Sidney	Hunter	HW 12 94 21 C	6
51	Jeffrey	Evans	LX 13 26 39 B	6
52	Doris	Bemdt	YA 49 88 11 A	3
53	Diane	Eaton	BE 08 74 68 A	1
54	Bonnie	Hall	WW 53 77 68 A	15
55	Taylor	Li	ZE 55 22 80 B	1

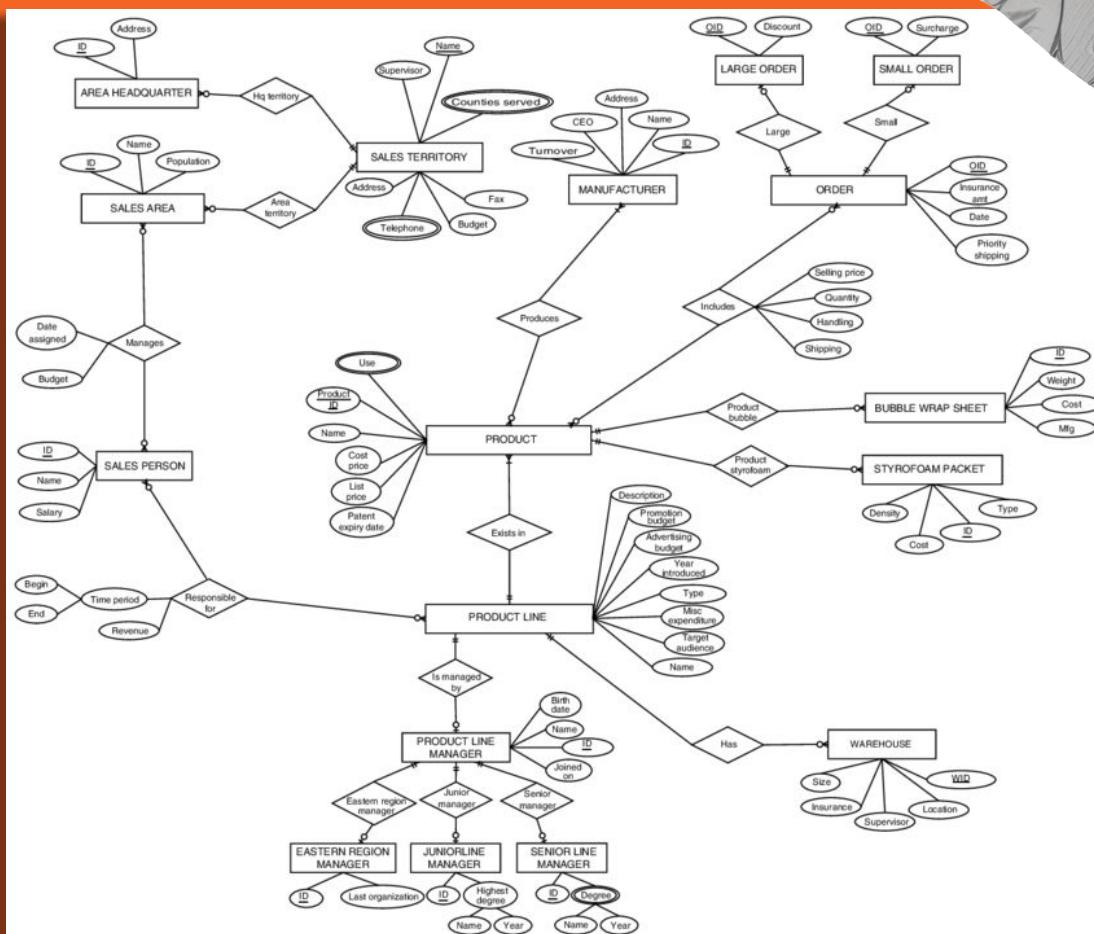
### Metadata

Column	Data Type	Description
employee_id	int	Primary key of a table
first_name	nvarchar(50)	Employee first name
last_name	nvarchar(50)	Employee last name
nin	nvarchar(15)	National Identification Number
position	nvarchar(50)	Current position title, e.g. Secretary
department_id	int	Employee department. Ref: Departments
gender	char(1)	M = Male, F = Female, Null = unknown
employment_start_date	date	Start date of employment in organization.
employment_end_date	date	Employment end date. Null if employee st

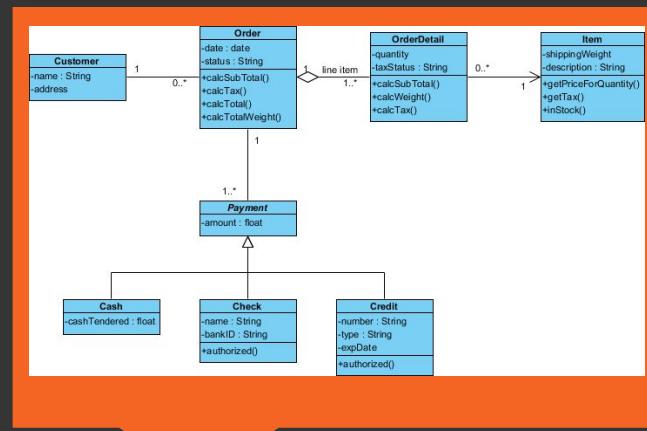
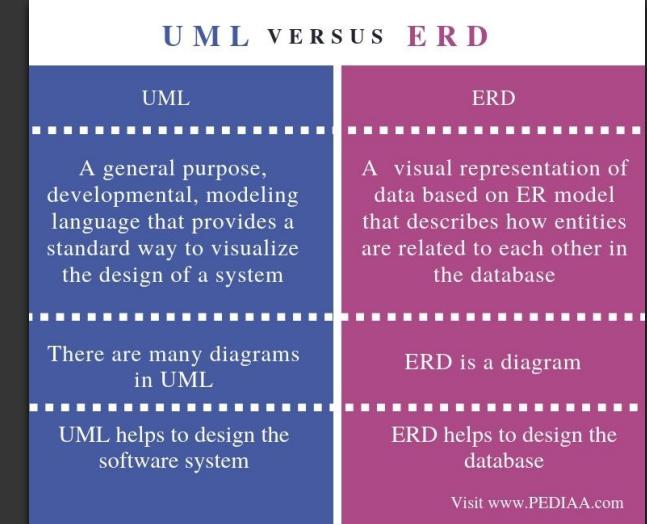
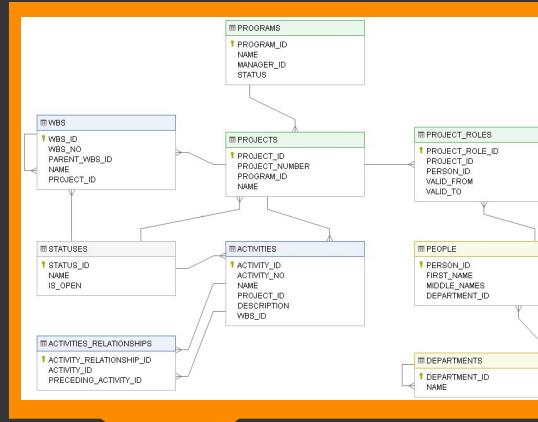
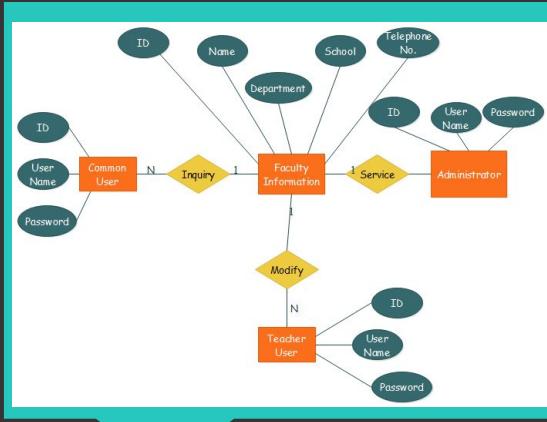
Data

<https://dataedo.com/kb/data-glossary/what-is-metadata>

## Entity-Relationship (ER) Diagram



# Model Representation



แล้วเราร้างฐานข้อมูลและ  
จัดการกับฐานข้อมูลได้ใน



**SQL**  
Structured Query Language  
/ɛs'kju:ə'lɛ/ or /'si:kwəl/

The SQL for relational databases, developed as part of IBM's System R project, became the standard query language. SQL is a domain-specific language used in programming and designed for managing data held in a RDBMS, or for stream processing in a relational data stream management system (RDSMS).

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features.

Arguably, the most widely used form of concurrent programming is the concurrent execution of database programs, called transactions.

<https://en.wikipedia.org/wiki/SQL>

# SQL LANGUAGES

## DDL

### Data Definition Language

- Create
- Drop
- Alter
- Truncate
- Comment

## DML

### Data Manipulation Language

- Select
- Update
- Insert
- Delete
- Merge
- Call
- Lock table
- Explain plan

## DCL

### Data Control Language

- Grant
- Revoke

## TCL

### Transaction Control Language

- Commit
- Rollback
- Savepoint
- Set transaction

# แล้วภาษา SQL มาจากไหน

# Set Theory

# Relational Algebra

# Tuple relational Calculus

The relational algebra uses set union, set difference, and cartesian product from set theory, but adds additional constraints to these operators.

SQL was originally based upon relational algebra and tuple relational calculus, it was one of the first commercial languages to use Edgar F. Codd's relational model.

## Relational Algebra

$$\text{FEMALE\_EMPS} \leftarrow \sigma_{\text{SEX}='F'}(\text{EMPLOYEE})$$
$$\text{EMPNAME} \leftarrow \pi_{\text{FNAME}, \text{LNAME}, \text{SSN}}(\text{FEMALE\_EMPS})$$
$$\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAME} \times \text{DEPENDENT}$$
$$\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{SSN}=\text{ESSN}}(\text{EMP\_DEPENDENTS})$$
$$\text{RESULT} \leftarrow \pi_{\text{FNAME}, \text{LNAME}, \text{DEPENDENT\_NAME}}(\text{ACTUAL\_DEPENDENTS})$$

## Relational Calculus

(a)  $\{t \mid \exists u \in R_1 (t[A] = u[A]) \wedge \neg \exists s \in R_2 (t[A] = s[A])\}$

(b)  $\{t \mid \forall u \in R_1 (u[A] = "x" \Rightarrow \exists s \in R_2 (t[A] = s[A] \wedge s[A] = u[A]))\}$

(c)  $\{t \mid \neg(t \in R_1)\}$

(d)  $\{t \mid \exists u \in R_1 (t[A] = u[A]) \wedge \exists s \in R_2 (t[A] = s[A])\}$

# มาเข้าเรื่อง NoSQL กัน

# NoSQL = Not Only SQL or Non-SQL

NoSQL เป็น Non-relational database ที่เก่งในการเก็บข้อมูลที่ไม่ได้อยู่ในรูปแบบ relational หรือมีความสัมพันธ์กันชัดเจน

จึงเหมาะสมสำหรับการใช้งานจำพวก Big Data และ Real-time Web Application ซึ่งมีข้อมูลที่ไร้โครงสร้างที่แน่นอน และมีปริมาณมาก เนื่องจากมีความยืดหยุ่นมากกว่า

	RDBMS	NoSQL
ชนิด	Relational	Non-Relational
รูปแบบของชุดข้อมูล	เป็นแบบโครงสร้างที่เก็บอยู่ใน Table	ไม่เป็นโครงสร้าง เก็บในรูปแบบ JSON (Text) หรือแบบอื่น ๆ
การ Scale	Vertical (เพิ่ม Spec Server)	Horizontal (เพิ่มจำนวน Server)
Schema	เปลี่ยนแปลงไม่ได้	เปลี่ยนแปลงได้ ค่อนข้างยืดหยุ่น
ตัวอย่าง Brand ในตลาด	Oracle, MySQL, Microsoft SQL Server, PostgreSQL	MongoDB, CouchDB, Redis, DynamoDB, Cassandra, HBase, Neo4j, Neptune

แบบเห็นจาก slide หน้าแรก ๆ  
ว่า NoSQL มีหลายแบบ  
แล้วแต่จะแบบต่างกันยังไงนะ

# แล้วจะเลือกใช้ SQL หรือ NoSQL กรณีไหนบ้าง

ใช้ RDBMS ในกรณีที่	ใช้ NoSQL ในกรณีที่
Workload คงที่ คาดเดาได้ และต้องการพื้นที่ปานกลางถึงมาก	มีปริมาณ Workload มหาศาลที่ต้อง Scale ได้เยอะ
ข้อมูลมีรูปแบบที่คาดเดาได้ มีโครงสร้างชัดเจน	ข้อมูลเป็นแบบ Dynamic มีการเปลี่ยนแปลงบ่อย
ข้อมูลต้องมี Relation เชื่อมถึงกันและกัน	ข้อมูลไม่จำเป็นต้อง Relation กันแบบซับซ้อน
ข้อมูลที่เขียนต้องเป็นไปตาม Condition	ต้องการเขียนไว ๆ Condition สำคัญรองลงมา ไม่ได้เน้นมาก
ข้อมูลค่อนข้าง Complex ต้องมีการ Query และ Report ได้	ข้อมูลค่อนข้าง Simple
ต้องการให้มี User ในการควบคุม	ต้องการให้ข้อมูลกระจายให้เข้าถึงกันได้ทุกส่วนใน Environment
จะ Deploy ใส่ Hardware ขนาดใหญ่ หรือของตัวเอง (On-Premise)	จะ Deploy บน Cloud (On-Cloud)



# เลือก Database พิดชีวิตเปลี่ยน

เลือกฐานข้อมูล ที่ใช้ให้กับงานของคุณ

	ข้อได้เปรียบ	ไม่เหมาะสมกับ	เคสการใช้งาน
 <b>Relational</b>	มั่นคงและแม่น้ำสูง ตัว Index ได้เป็นอย่างดี	ข้อมูลที่ซึ่งมีโครงสร้าง	<ul style="list-style-type: none"> <li>ระบบ ERP / CRM</li> <li>Financial Transaction</li> <li>Data Warehousing</li> </ul>
 <b>Key-value</b>	อ่านและเขียนข้อมูลได้ใน เวลากระถูกต่อต้องเร็ว	Query ที่มีความซับซ้อน	<ul style="list-style-type: none"> <li>การประมวล Real Time</li> <li>ระบบติดตามลูกค้า</li> <li>เก็บติดตามลูกค้า</li> </ul>
 <b>Document</b>	เรียบง่ายและอีดหยุ่น กับข้อมูลที่โครงสร้าง	การกำหนดความสัมพันธ์ ระหว่างข้อมูลส่วนต่างๆ	<ul style="list-style-type: none"> <li>เก็บติดตามลูกค้า</li> <li>การจัดทำผู้ใช้งาน</li> <li>ระบบ CMS</li> </ul>
 <b>In-memory</b>	ทำงานร่วมกับฐานข้อมูล ได้หลายรูปแบบ และทำงานได้บันดาลเรียกต่อวินาที	การเก็บข้อมูลตลอดเวลา	<ul style="list-style-type: none"> <li>Caching</li> <li>Real-time streaming</li> <li>กระบวนการแบบ</li> </ul>
 <b>Graph</b>	สร้างความสัมพันธ์กับ ข้อมูลหลายรูปแบบได้ง่าย	การทำงานกับ Transaction จำนวนมาก	<ul style="list-style-type: none"> <li>ระบบ/เครื่องมือแนะนำ</li> <li>Fraud Detection</li> <li>Social Networking</li> </ul>
 <b>Time-series</b>	ปรับขนาดเพื่อเก็บ ข้อมูล Time-series ได้รวดเร็ว	ข้อมูลเปลี่ยน เอกสาร แคตตาล็อก หรือโปรไฟล์	<ul style="list-style-type: none"> <li>แอปพลิเคชัน IoT</li> <li>การติดตามแอปพลิเคชัน</li> <li>DevOps</li> </ul>
 <b>Ledger</b>	เก็บประวัติได้อย่างแม่น้ำ และสามารถปรับ Scale ได้สูง	การใช้กับข้อมูล ที่มีระดับกระจาย	<ul style="list-style-type: none"> <li>เก็บข้อมูลบันทึก</li> <li>ติดตามรายการเดินบัญชี</li> <li>เก็บรักษาข้อมูลสืบทอดคลัง</li> </ul>

<https://web.facebook.com/skooldio/photos/a.457984764545584/1764214350589279/>



The text "THE END" is displayed in large, white, bold letters against a red background. Each letter is suspended by a thin white string from a small black dot at the top center of each letter. The background is a solid red color.

THE END