MongoDB雲端資料庫簡介(一)

日期 2016 06/07

地點 校務研究辦公室

傳統資料庫技術原理-基本觀念 01

孰輕孰重

程式

資料庫

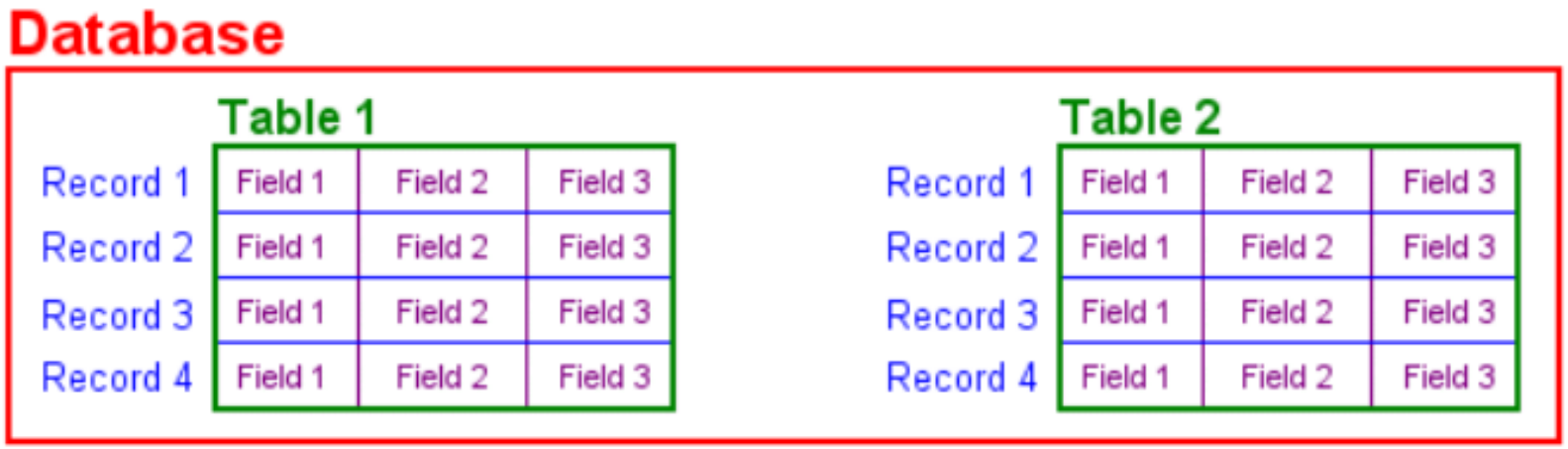
```php
<?php
    $docRoot = $_SERVER['DOCUMENT_ROOT'];
    $uploadDir = $docRoot . "/" . "upload";
    if(!is_dir($uploadDir)){
        mkdir($uploadDir);
    }

    if( $_FILES['file']['name'] != "" ){
        move_uploaded_file( $_FILES['file']
           die( "檔案上傳失敗");
    } else {
        die("開啟檔案失敗!");
    }
```

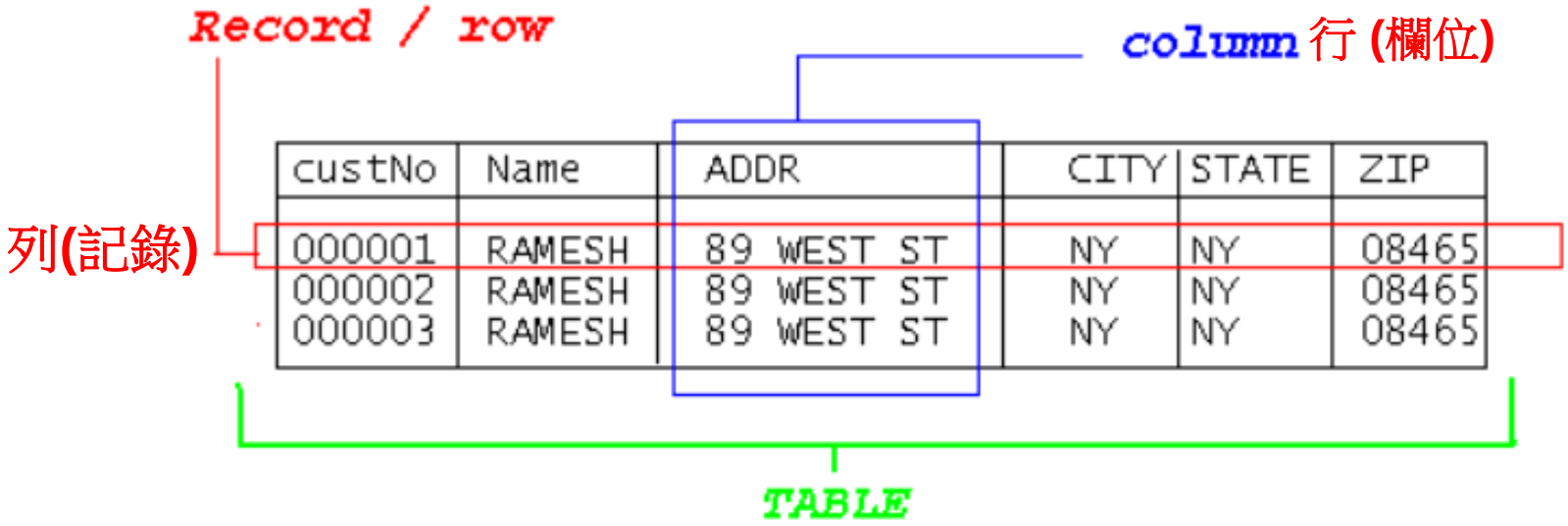
傳統資料庫技術原理-基本觀念 02

關聯式資料庫管理系統 (RDBMS)

---Oracle RDBMS
---SQL RDBMS
---MySQL RDBMS ---Database1
設備資料庫 ---Database2
人事資料庫 ---Database3---Table1
職員表格 ---Table2
教師表格 ---Table3---Record1
王老師記錄 ---Record2
張老師記錄 ---Record3---Field1
姓名欄位 ---Field2
住址欄位 ---Field3

傳統資料庫技術原理-基本觀念 03





傳統資料庫技術原理-基本觀念 04

```
mysql> create table performance
    -> ( emp_id tinyint unsigned,
    -> units tinyint unsigned,
    -> hours tinyint unsigned,
    -> shift tinyint unsigned,
    -> supervisor varchar(25),
    -> date date,
    -> name varchar(25),
    -> index (emp_id, name)
    -> ) engine innodb;
Query OK, 0 rows affected (0.02 sec)
```

Schema

表格欄位定義

```
mysql> explain performance;
+------------+---------------------+------+-----+---------+-------+
| Field      | Type                | Null | Key | Default | Extra |
+------------+---------------------+------+-----+---------+-------+
| emp_id     | tinyint(4) unsigned | YES  | MUL | NULL    |       |
| units      | tinyint(3) unsigned | YES  |     | NULL    |       |
| hours      | tinyint(3) unsigned | YES  |     | NULL    |       |
| shift      | tinyint(3) unsigned | YES  |     | NULL    |       |
| supervisor | varchar(25)         | YES  |     | NULL    |       |
| date       | date                | YES  |     | NULL    |       |
| name       | varchar(25)         | YES  |     | NULL    |       |
+------------+---------------------+------+-----+---------+-------+
```

Transaction - ACID

Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, the entire transaction fails, and the database state is left unchanged.（不可分割性）

Consistency ensures that any transaction will bring the database from one valid state to another. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers.（一致性）

Isolation ensures that the concurrent execution of transactions result in a system state that would be obtained if transactions were executed serially, i.e. one after the other.（隔離性）

Durability means that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.（永久性）

傳統資料庫技術原理–合併資料表

Passenger Table (旅客表格)

NAME	SSN	FLIGHT#
Alice	555–55–5555	F101
Bob	222–22–2222	F313
Carol	123–45–6789	F123
Bob	444–44–4444	F313

Flight Table (航班表格)

FLIGHT#	START	END
F101	DCA	JFK
F313	JFK	DCA
F123	JFK	LAX

合併航班欄位

NAME	SSN	FLIGHT#	START	END
Alice	555–55–5555	F101	DCA	JFK
Bob	222–22–2222	F313	JFK	DCA
Carol	123–45–6789	F123	JFK	LAX
Bob	444–44–4444	F313	JFK	DCA

傳統資料庫技術原理-表格正規化 01

學號	導師	導師辦公室	課程 1	課程 2	課程 3
1022	Jones	412	101-07	143-01	159-02
4123	Smith	216	201-01	211-02	214-01

一級正規化-避免重複欄位（三維表格）

傳統資料庫技術原理-表格正規化 02

學號	導師	導師辦公室	課程 #
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	201-01
4123	Smith	216	211-02
4123	Smith	216	214-01

二級正規化-刪除重複資料

傳統資料庫技術原理-表格正規化 03

學號	導師	導師辦公室
1022	Jones	412
4123	Smith	216

學號	課程 #
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

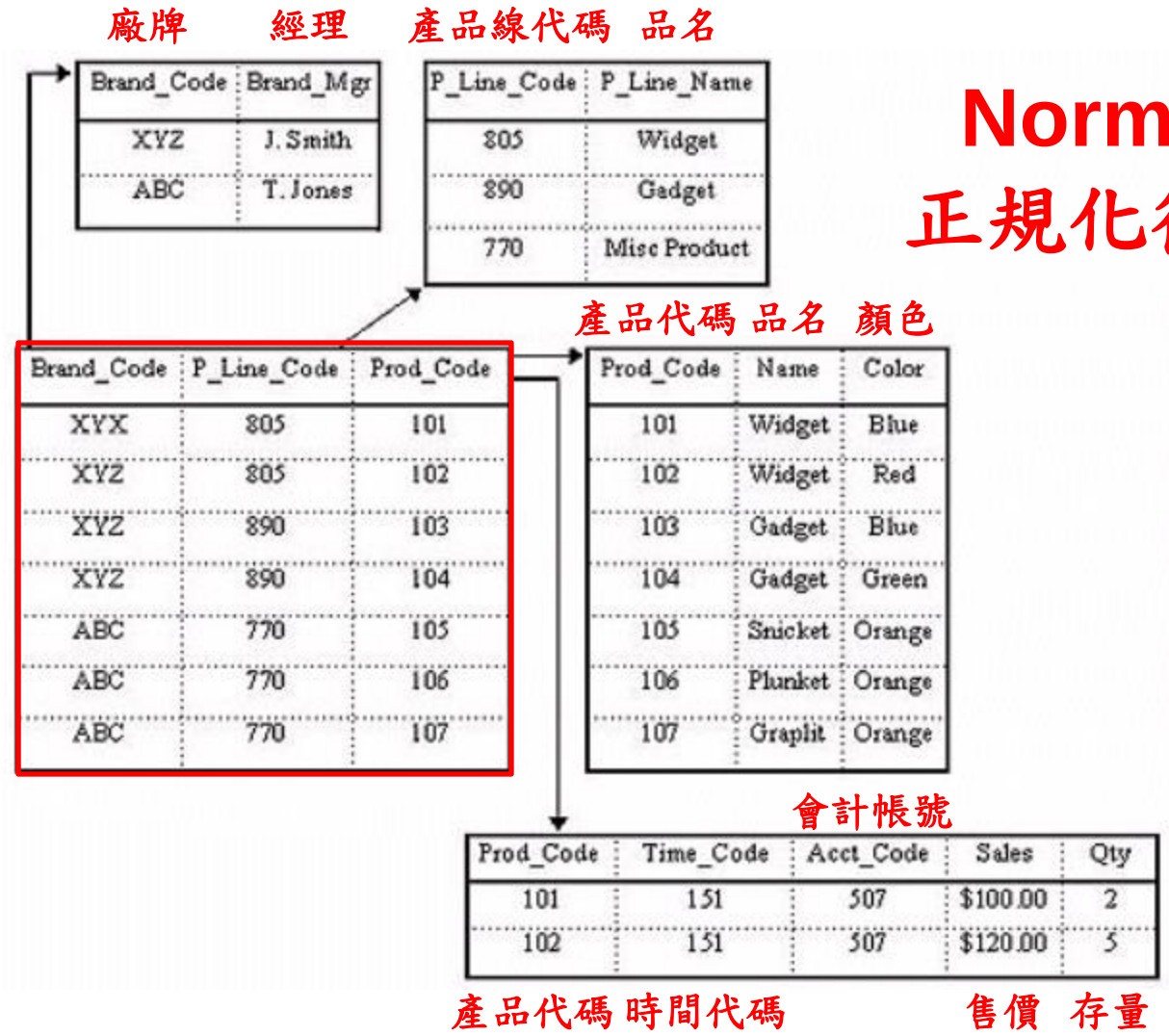
三級正規化-排除遞移相關性

傳統資料庫技術原理-表格正規化 04

學號	導師
1022	Jones
4123	Smith

名稱	辦公室	部門
Jones	412	42
Smith	216	42

傳統資料庫技術原理-表格正規化 05



Normalization
正規化後的資料表

傳統資料庫技術原理-表格正規化 06

雪花式的資料表？

DimDate
- DateKey
- Date
- Day
- MonthNumber
- MonthName
- Quarter
- Year

DimCity
- CityKey
- CityId
- CityName

DimCustomer
- CustomerKey
- CustomerId
- CustomerName
- CityKey

FactBookSales
- DateKey
- CustomerKey
- ProductKey
- TransRef
- SalePrice
- SaleCost
- Quantity
- SalesAmount

DimProduct
- ProductKey
- LineKey
- AuthorKey
- PublisherKey
- ProductId
- Title
- SaleCost
- SelePrice

DimLine
- LineKey
- LineId
- LineName

DimAuthor
- AuthorKey
- AuthorId
- AuthorName

DimPublisher
- PublisherKey
- PublisherId
- PublisherName

傳統資料庫技術原理–RDBMS 的困境 01

How About Schema Change?

如果需要變更欄位怎麼辦?

How About Database Performance?

如何提升資料庫效能?

傳統資料庫技術原理-RDBMS 的困境 02

Limitations for SQL database

擴充困難 Scability

Scalability: Users have to scale relational database on powerful servers that are expensive and difficult to handle. To scale relational database it has to be distributed on to multiple servers. Handling tables across different servers is a chaos.

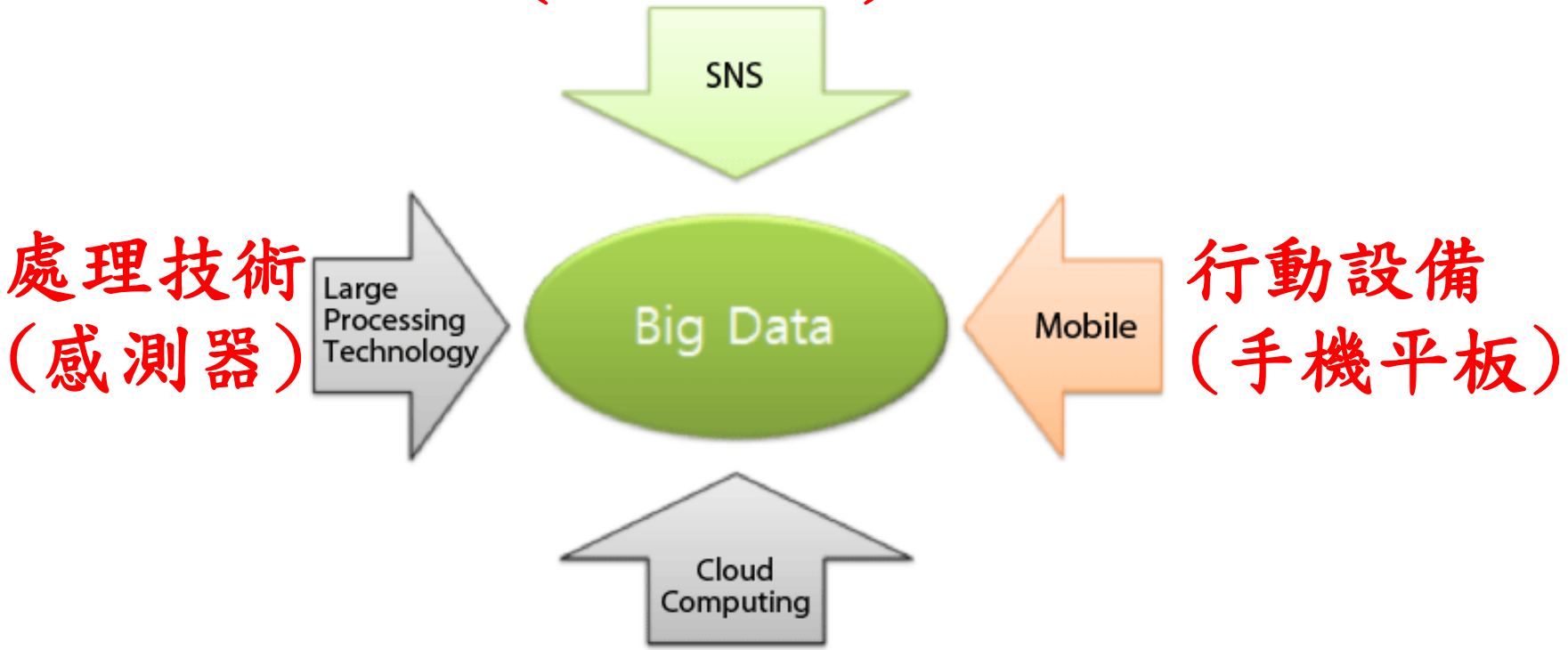
設計複雜 Complexity

Complexity: In SQL server's data has to fit into tables anyhow. If your data doesn't fit into tables, then you need to design your database structure that will be complex and again difficult to handle.

雲端資料庫基本原理–Big Data 01

雲端資料庫基本原理–Big Data 02

社群網路服務
(Facebook)



雲端運算資料

雲端資料庫基本原理-Big Data 03

巨量-水平擴充容易

多樣-欄位定義彈性

快速-分散運算能力

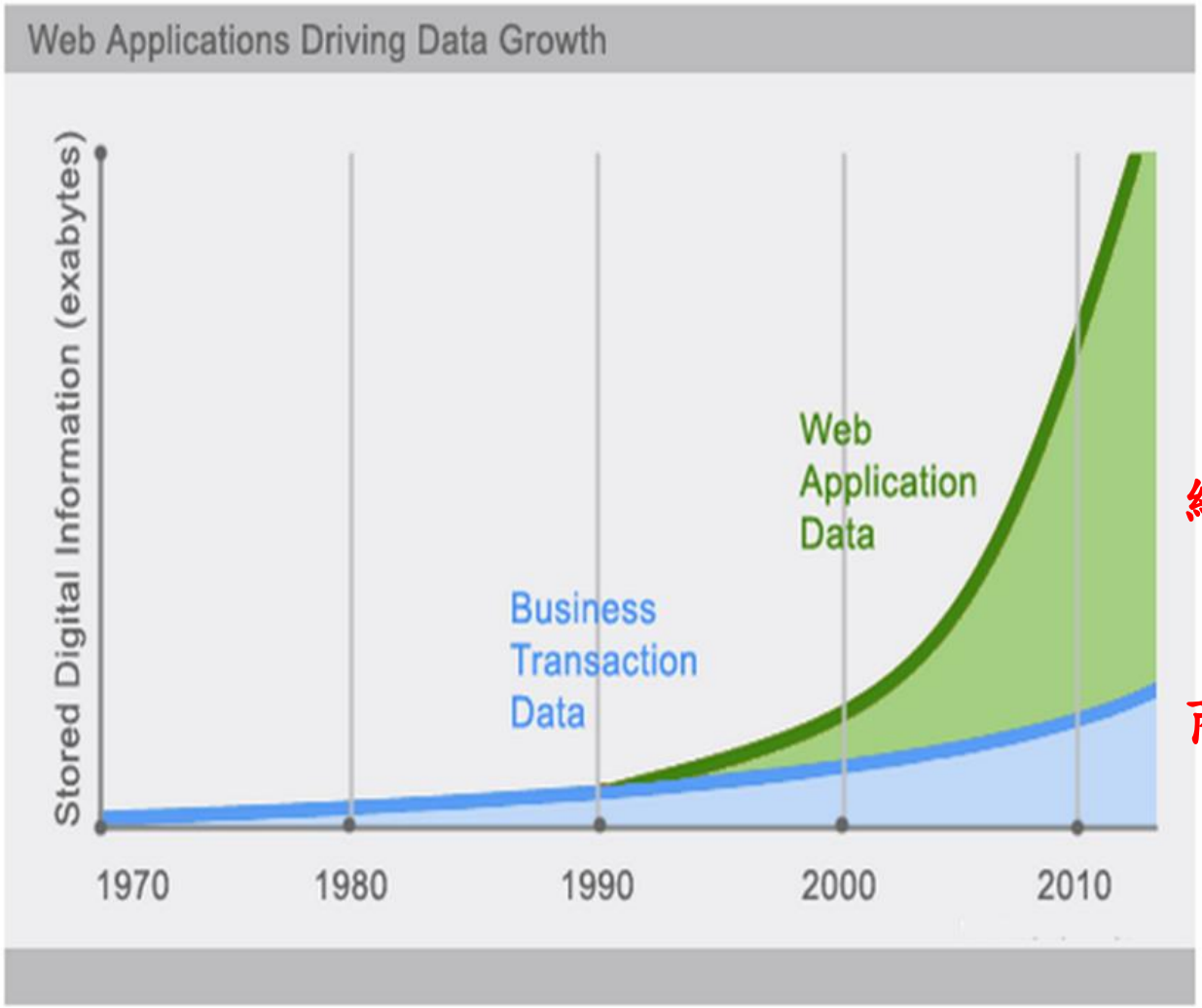
What is Big Data?

Big data means immense amount of data, so much so that it is difficult to collect, store, manage, and analyze via general database software. In general, the meaning of "immense amount of data" is classified into three types as follows:

- **Volume:** There is too much data to be stored and require too many processes—semantic analysis/data processing. These are the two elements that we need to understand.
- **Velocity:** It means storage and processing speed.
- **Variety:** The demand for unstructured data, such as text and images is increasing as well as refined-type data that can be standardized and previously defined like the RDBMS table record.

Value(價值)？Veracity(正確)？

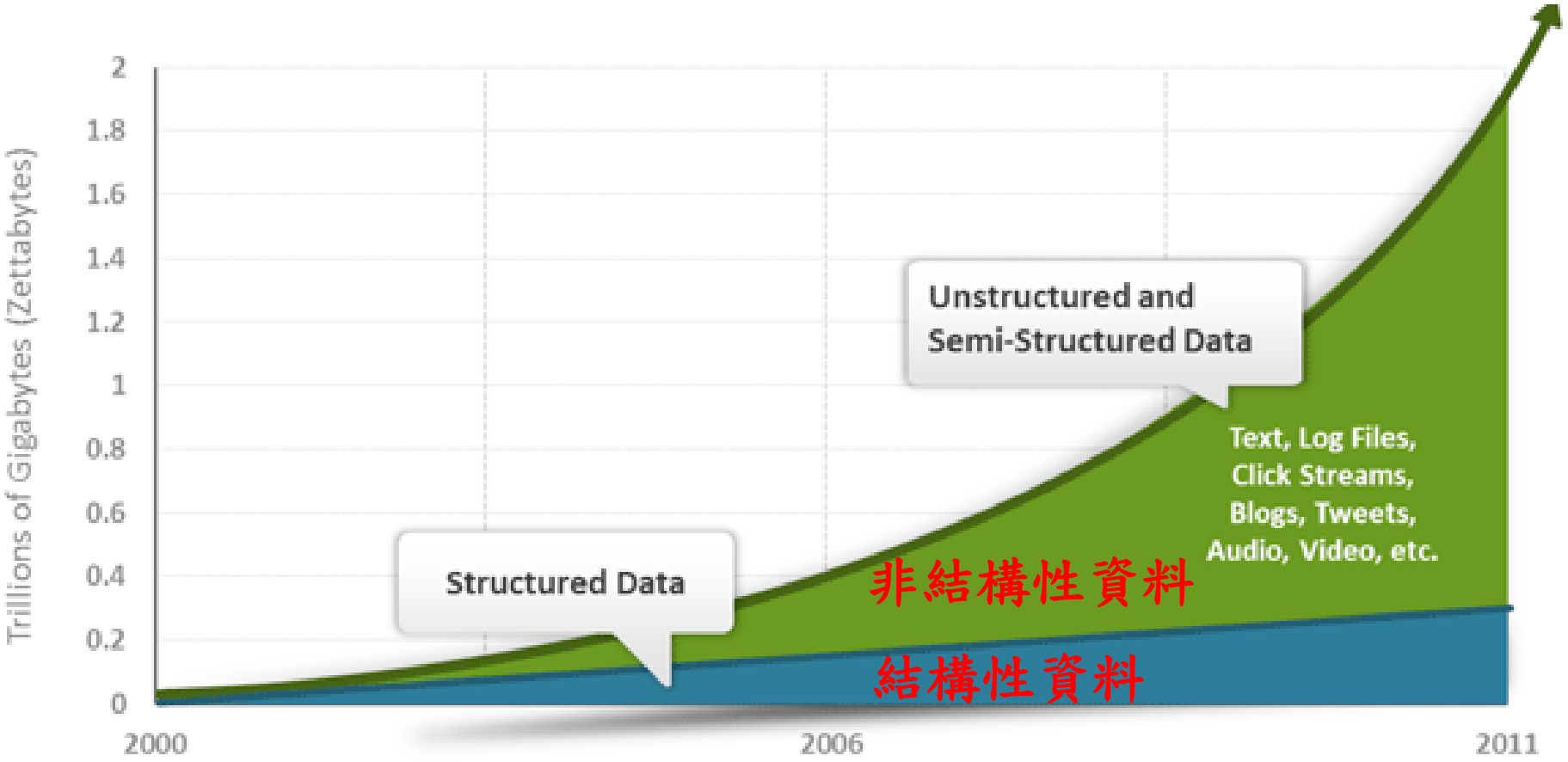
雲端資料庫基本原理–Big Data 04



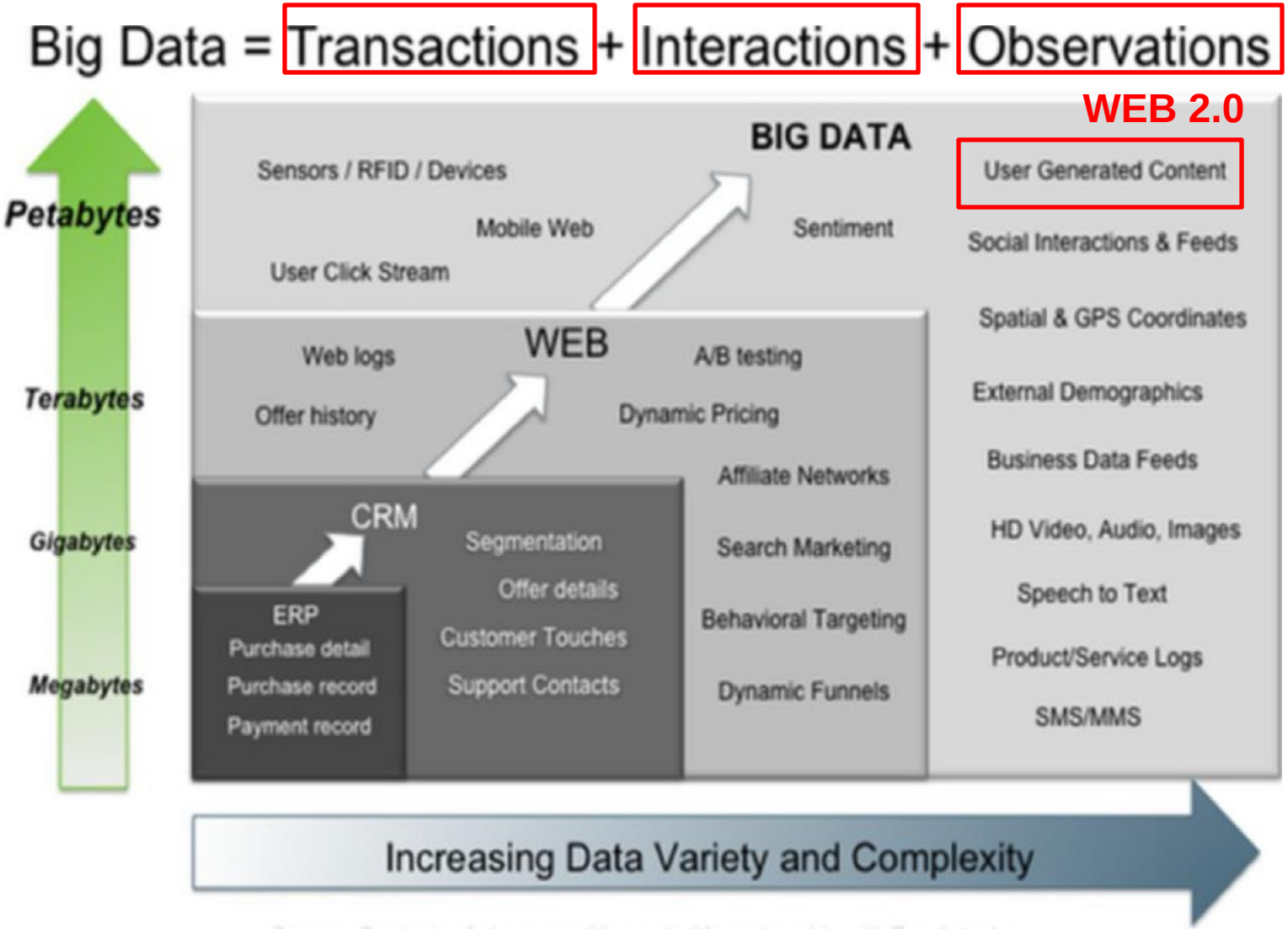
網站應用資料

商業交易資料

雲端資料庫基本原理–Big Data 05



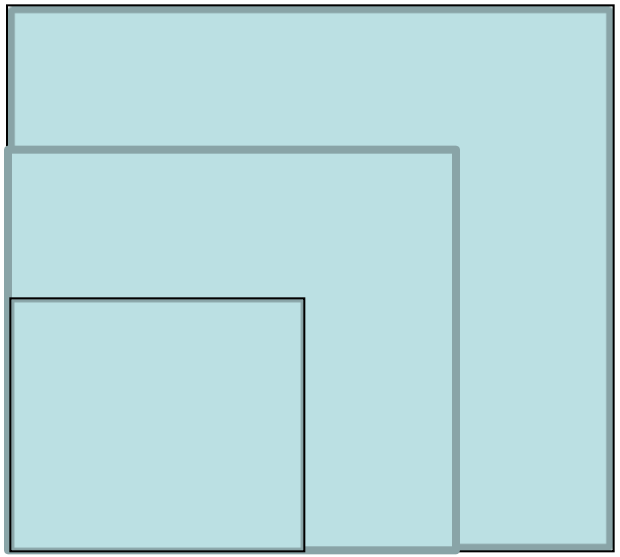
雲端資料庫基本原理–Big Data 06



雲端資料庫基本原理–系統擴充 01

Scale Up ➡ Vertical Scaling (垂直擴充)

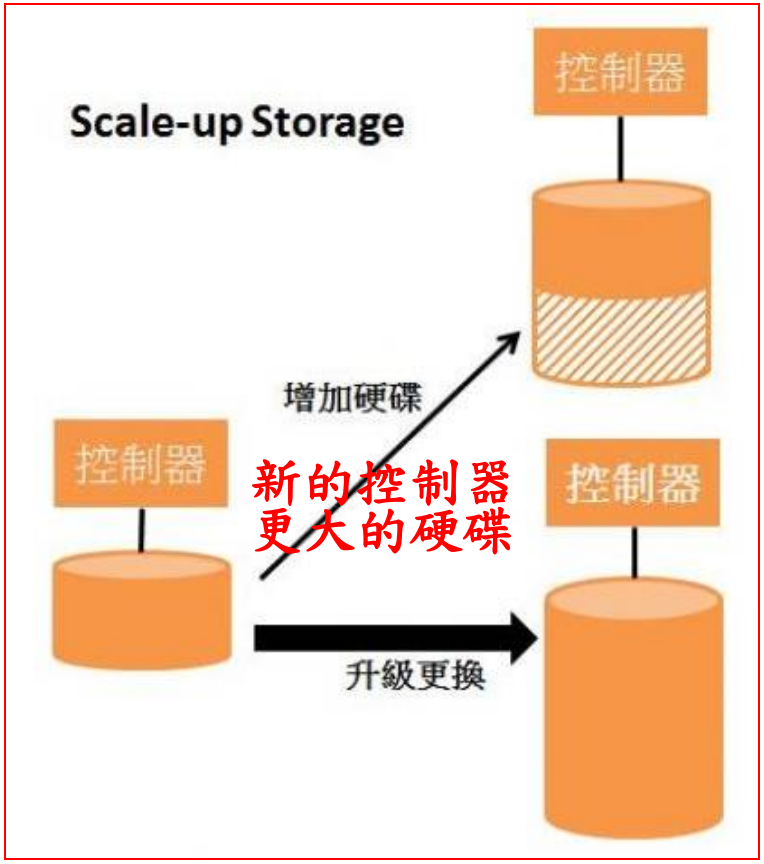
Scale Out ➡ Horizontal Scaling (水平擴充)

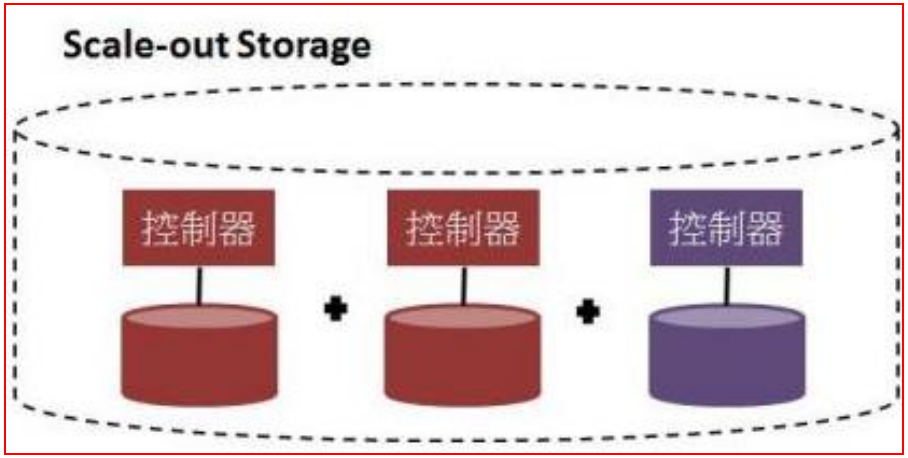


Scale Up

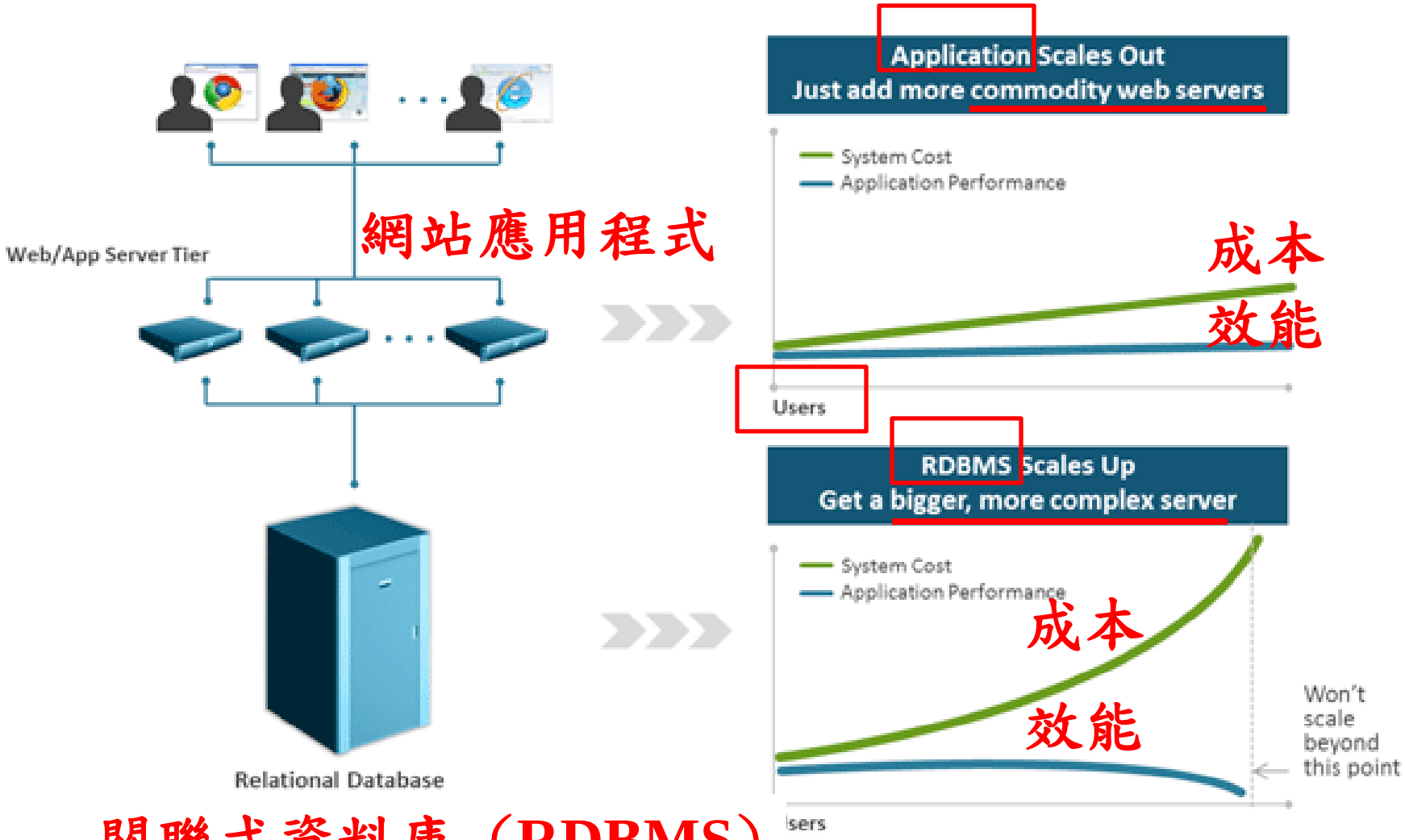
Scale Out

雲端資料庫基本原理-系統擴充 02

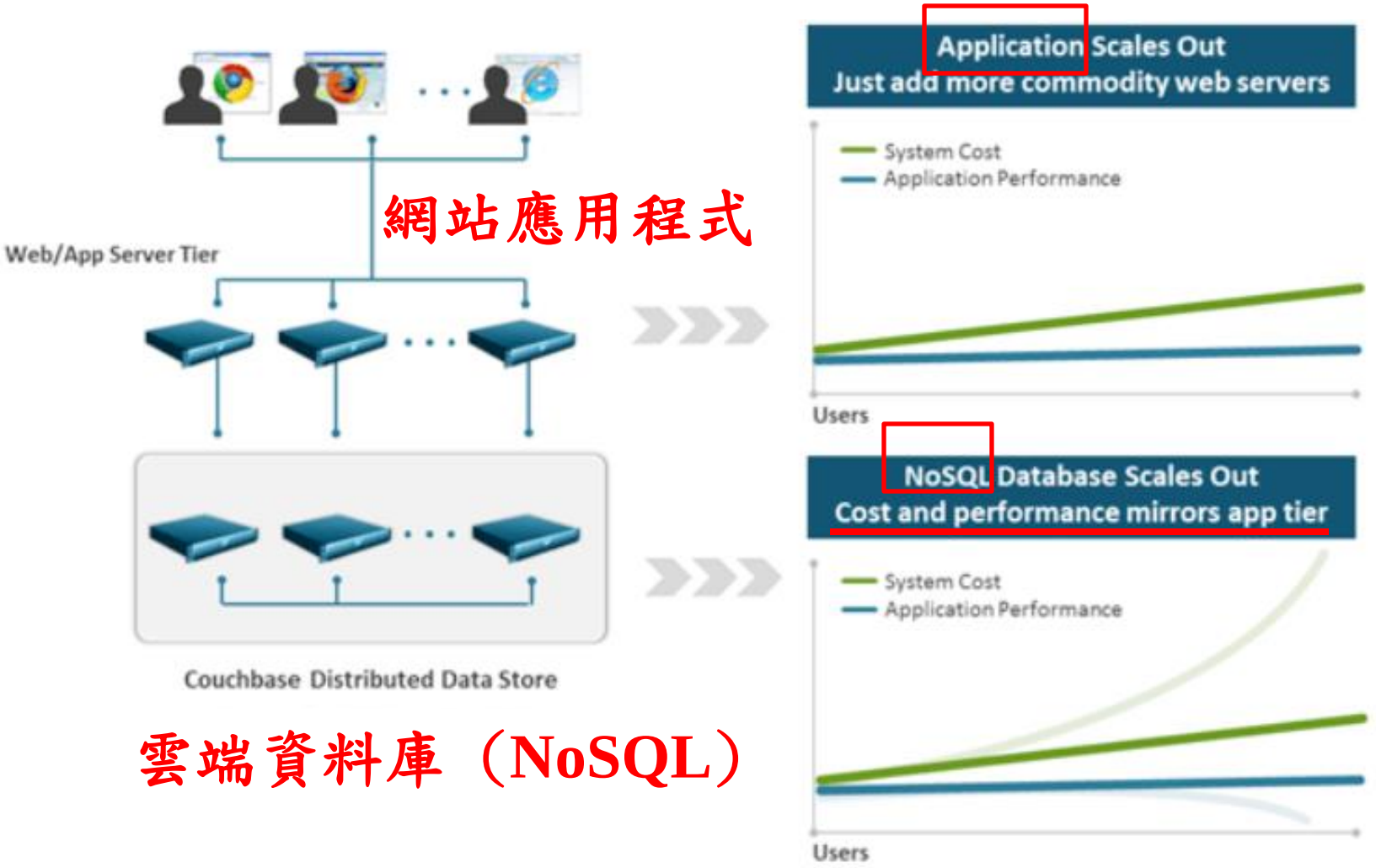




雲端資料庫基本原理-系統擴充 03



雲端資料庫基本原理-系統擴充 04



雲端資料庫（NoSQL）

雲端資料庫基本原理–資料庫分類 01

Key-Value Based
- **Amazon DynamoDB**
- **Redis**

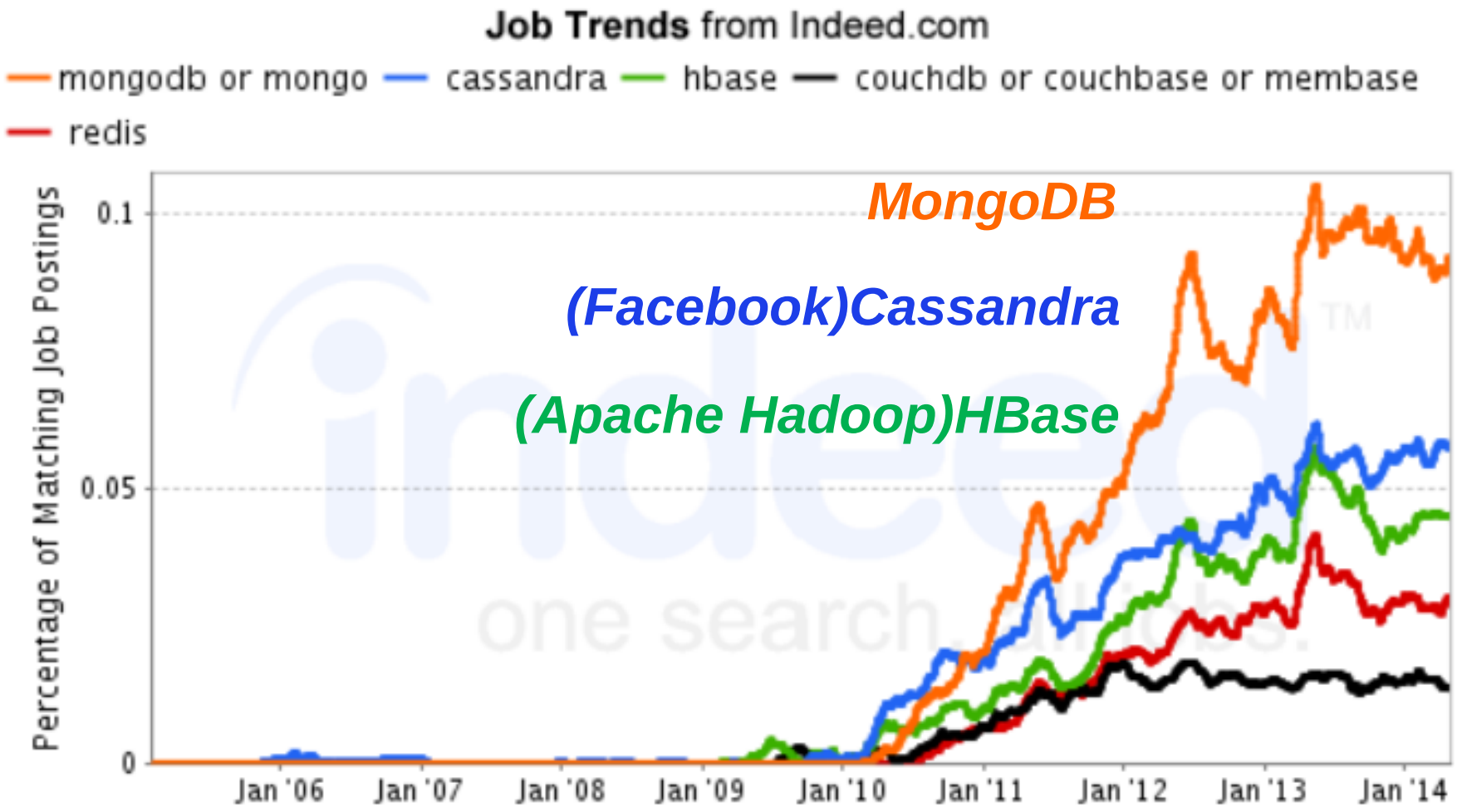
Document Based
- **CouchDB**
- **MongoDB**

Column Based
- **Google BigTable**
- **Facebook Cassandra**

(Google Big Table + Amazon DynamoDB)

Graph Based
- **Neo4j**
- **OrientDB**

雲端資料庫基本原理-資料庫分類 02



雲端資料庫基本原理-CAP 理論 01

According to University of California, Berkeley computer scientist Eric Brewer, the theorem first appeared in fall 1998.[4] It was published as CAP principle in 1999.

In 2002, Seth Gilbert and Nancy Lynch of MIT published a formal proof of Brewer's conjecture, rendering it a theorem.

Conjecture -> Theorem

推測　　理論

雲端資料庫基本原理-CAP 理論 02

資料亂掉

Consistency - This means that the data in the database remains consistent after the execution of an operation.

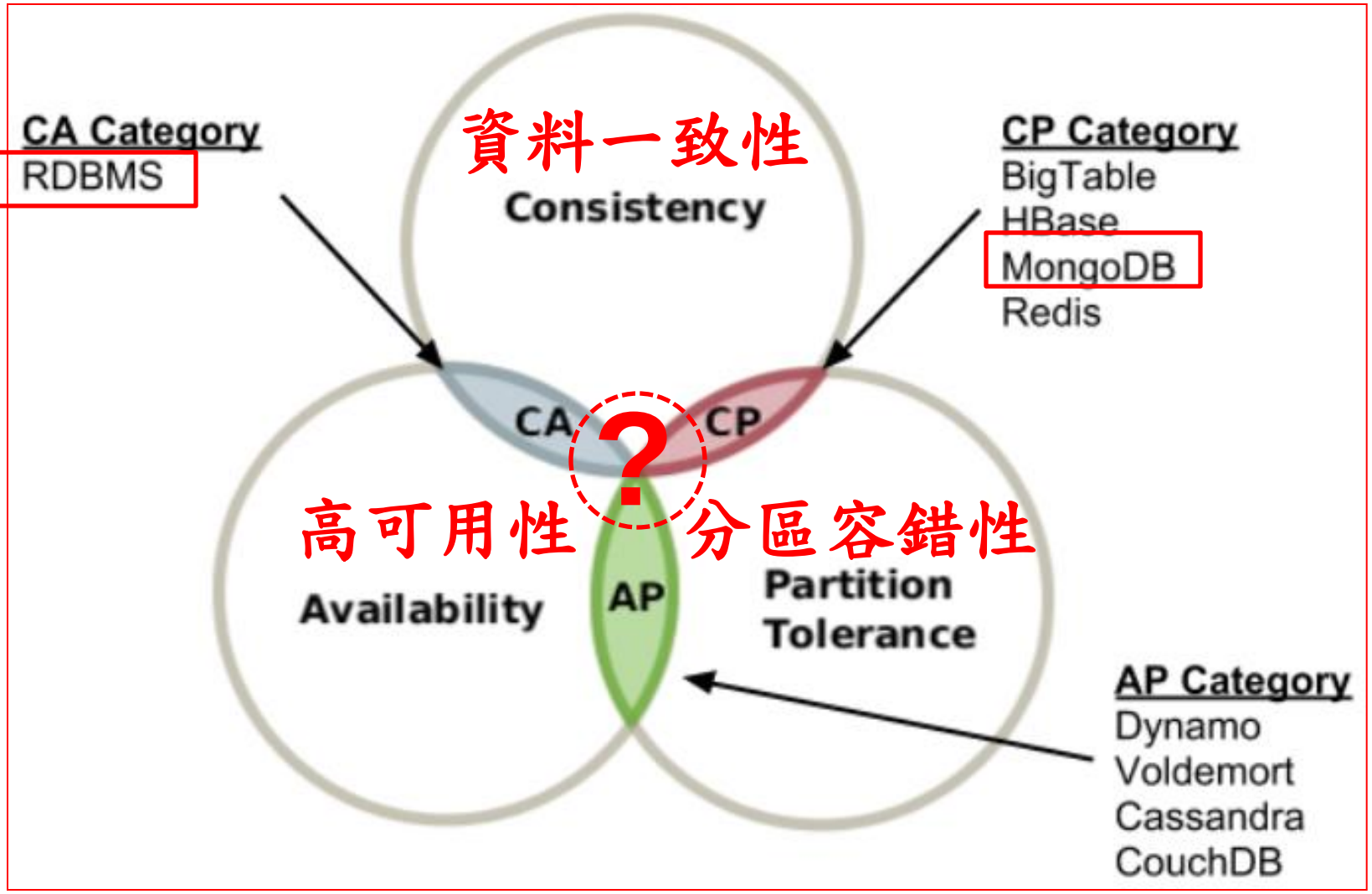
主機當機

Availability - This means that the system is always on (Service guarantee availability), no downtime.

網路斷線

Partition Tolerance - This means that the system continues to function even if the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

雲端資料庫基本原理-CAP 理論 03



雲端資料庫基本原理-CAP 理論 04

Generally, it is not be possible to fulfill all three requirements in a distributed system.

不可能完全滿足 CAP 三項要求

CAP provides the basic requirements for a distributed system to follow two of the three requirements.

最多只能滿足 CAP 其中二項要求

Distributed systems must be partition tolerant (P), so we have to choose between Consistency and Availability.

分散式系統一定要滿足 P 項的要求

Current NoSQL databases follow the different combinations of C and A from the CAP theorem.

NoSQL 可以滿足 AP 項或 CP 項要求

MongoDB 雲端資料庫-基本觀念 01

Humongous：巨大無比

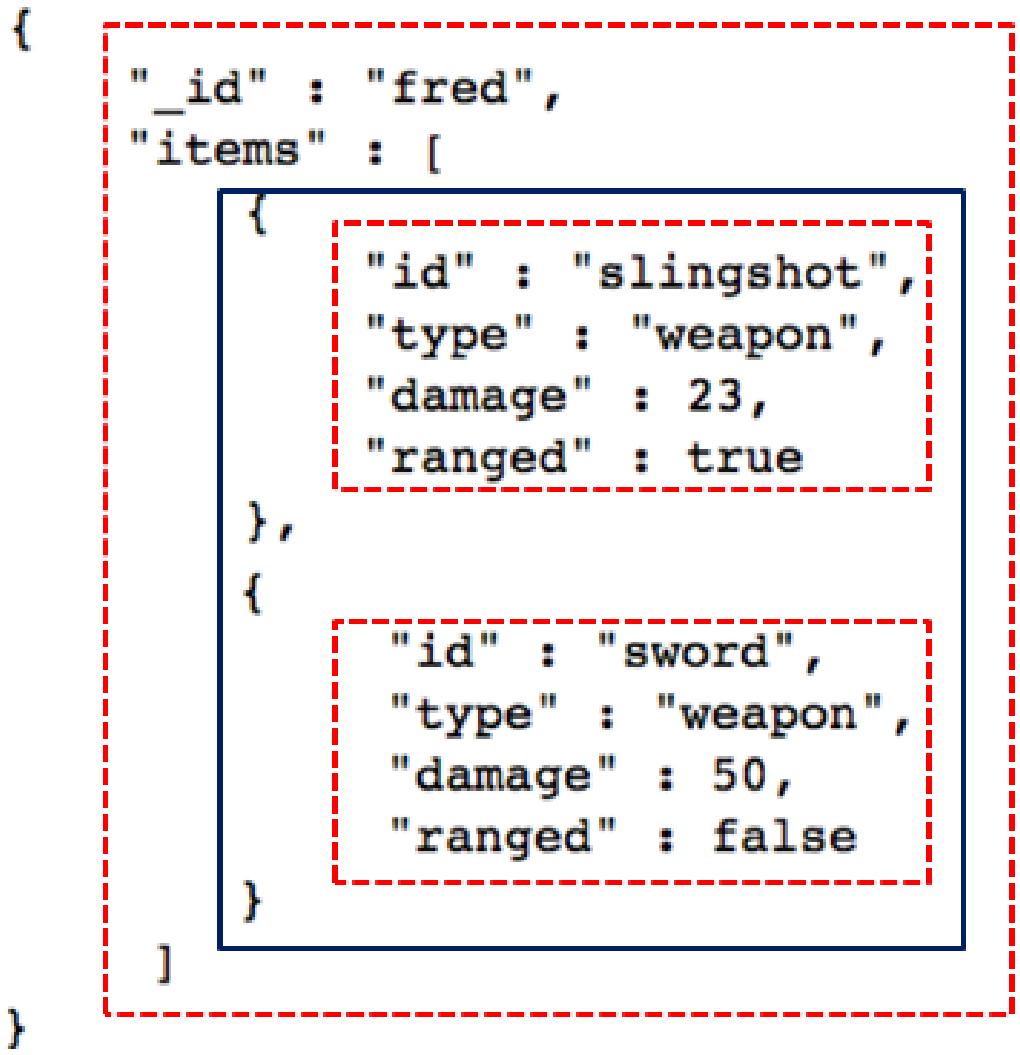
MongoDB 雲端資料庫-基本觀念 02

	關聯式資料庫	MongoDB資料庫
	Relational	MongoDB
資料庫	Database	Database
表格	Table	Collection
資料記錄	Row	(JSON) document

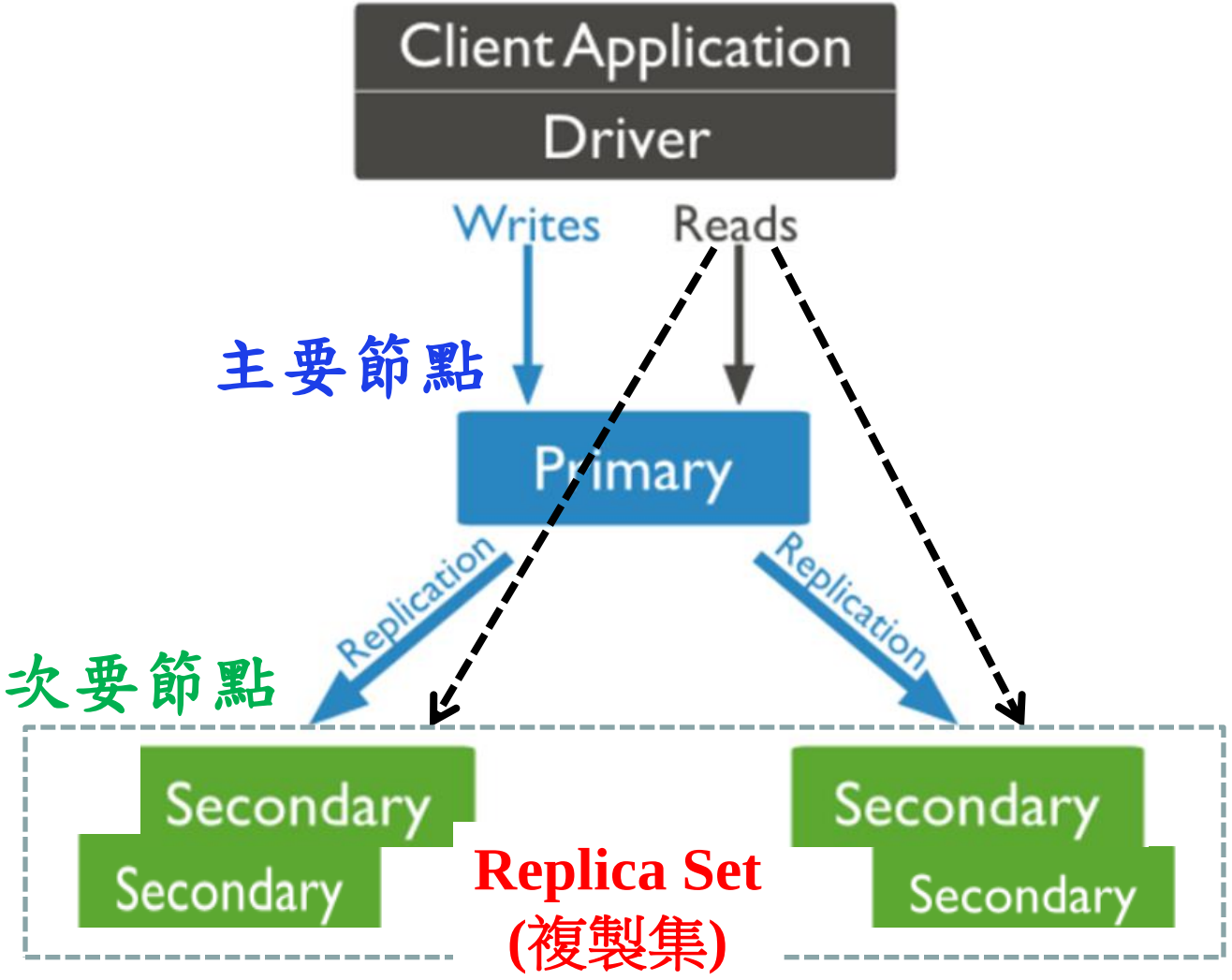
MongoDB 雲端資料庫-基本觀念 03

JSON 資料格式

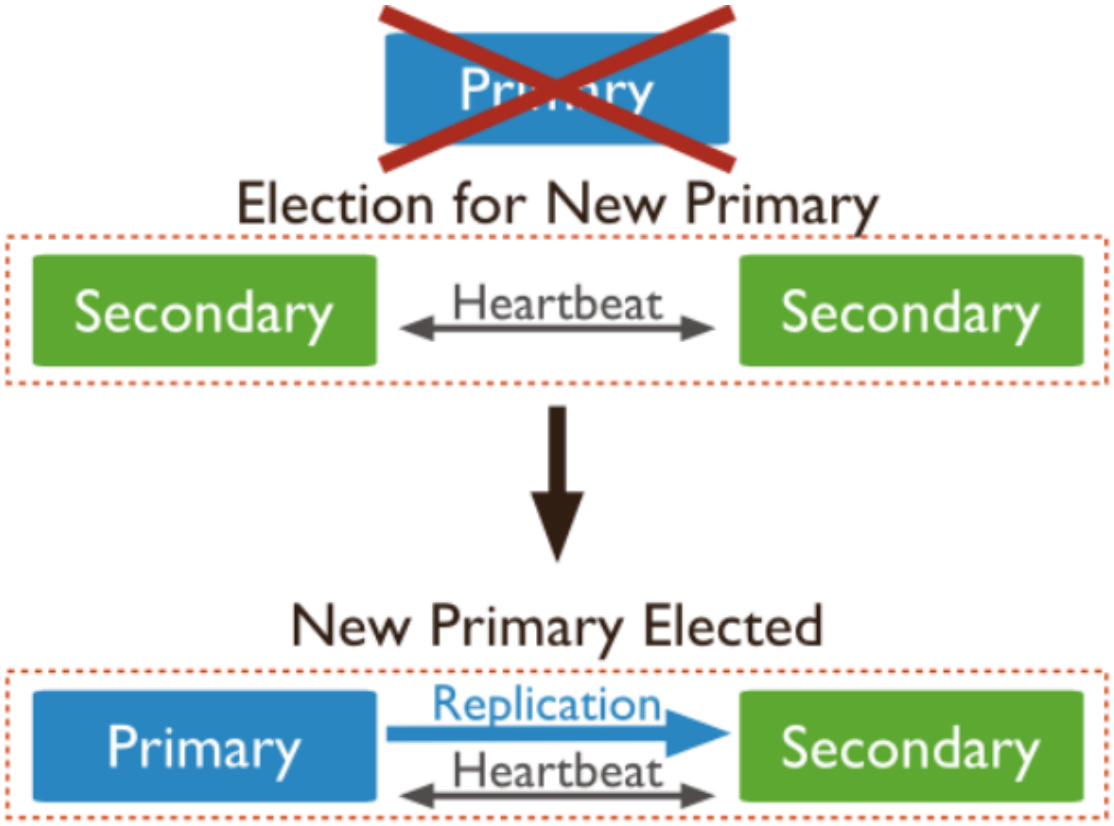
MongoDB 雲端資料庫-基本觀念 04



MongoDB 雲端資料庫-基本觀念 05



MongoDB 雲端資料庫-基本觀念 06

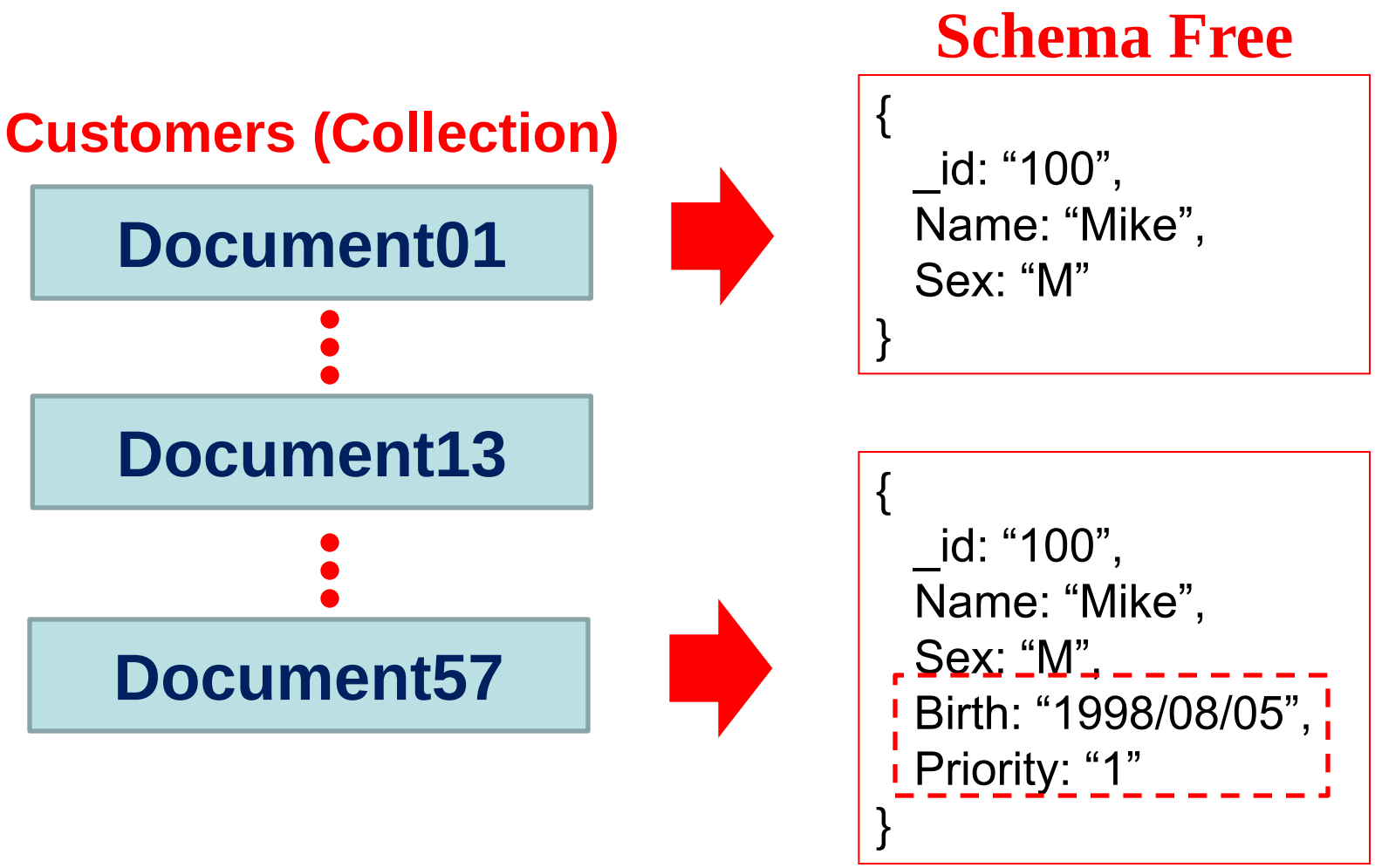


MongoDB 雲端資料庫-基本觀念 07

A replica set can have up to 12 members,
but only 7 voting members.

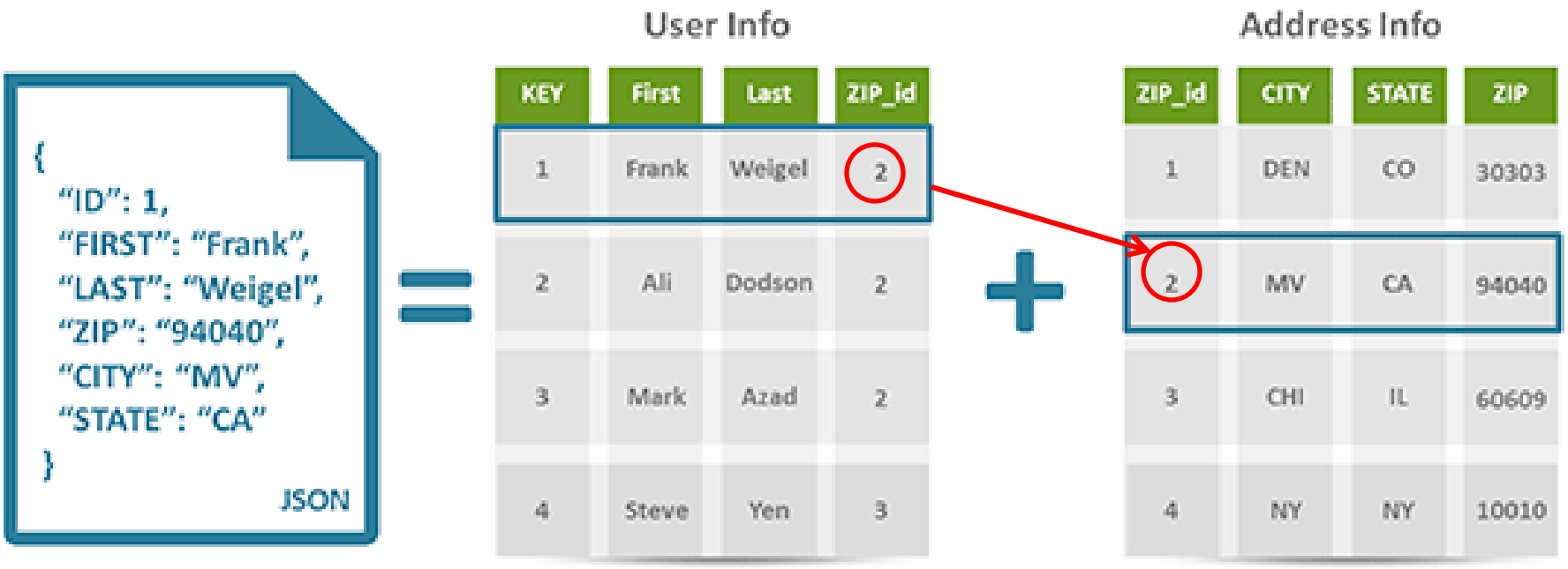

```
{
   "_id" : <num>
   "host" : <hostname:port>,
   "votes" : 0
}
```

MongoDB 雲端資料庫-RDBMS vs NoSQL 01



MongoDB 雲端資料庫–RDBMS vs NoSQL 02

減低多表格查詢次數



MongoDB 雲端資料庫–RDBMS vs NoSQL 03

JSON-format

```
{
  _id : “Key123”,
  name: “Jane”,
  phones: [123, 456],
  …
}
```

No Normalization

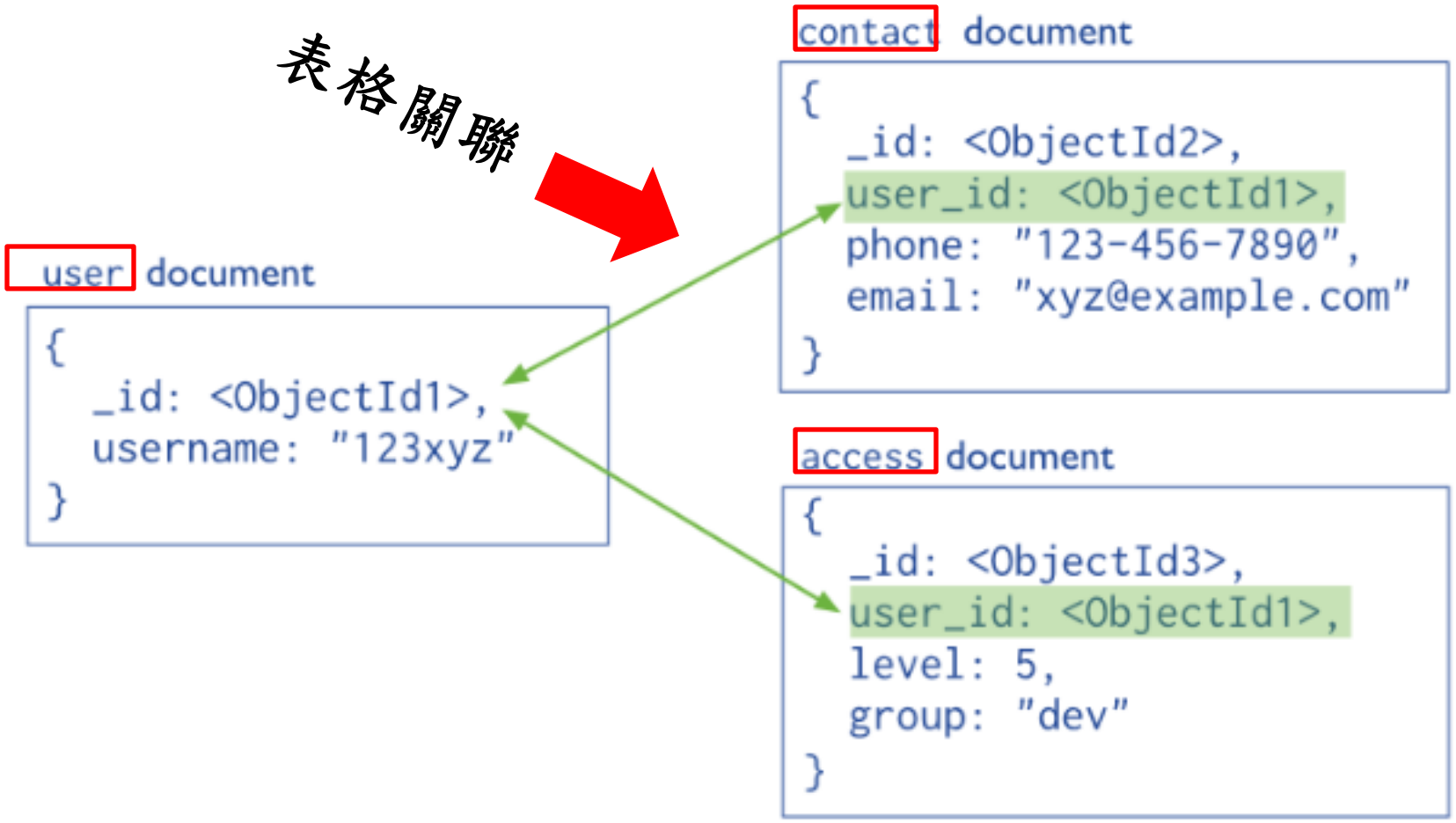
Id	name
Key123	Jane

PersonId	PhoneId
Key123	1
Key123	2

Id	Phone
1	123
2	456

免除正規化操作

MongoDB雲端資料庫–RDBMS vs NoSQL 04

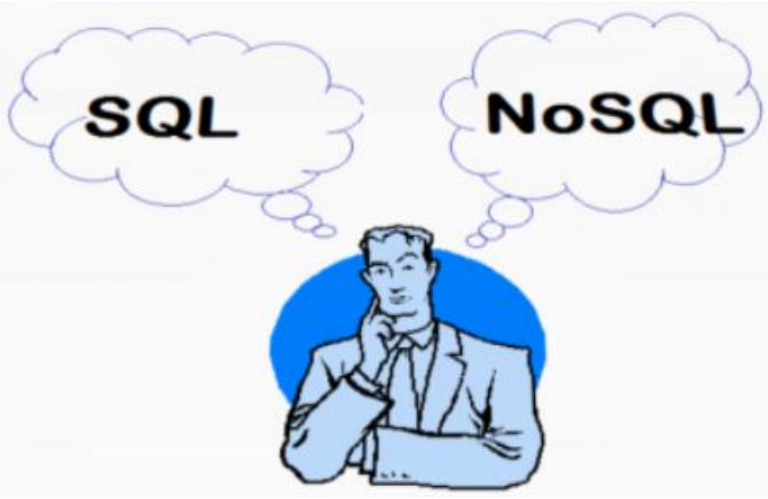


MongoDB 雲端資料庫–RDBMS vs NoSQL 05

RDBMS and NoSQL–選擇資料庫 01

RDBMS (傳統資料庫)
Relational Database Management System (SQL)
(SQL-Structured Query Language)

NoSQL (雲端資料庫)
Not only SQL



RDBMS and NoSQL–選擇資料庫 02

If SQL is your only storage tool, then all problems must look the same – no Big Data.

Using SQL and NoSQL storage technologies where appropriate, systems can have

SQL 高一致性

- **SQL for high consistency where needed**

NoSQL 高可用性與擴充性

- **NoSQL for high availability and scalability**

RDBMS and NoSQL-選擇資料庫 03

In the business world -

實務應用上，高可用性比高一致性更重要
Availability is more valuable than Consistency

可以替代查詢其他表格資料(例如訂單)、道歉、或修補
- **When in doubt, take the customer's order.**
- **Apologize, fix, and compensate later.**

但千萬不要對客戶說很抱歉，機器當掉所以無法向您收費
Do not say
Sorry, we can not take your money.
Our computers are down!

RDBMS and NoSQL-選擇資料庫 04

NoSQL databases relaxed the ACID properties (BASE)

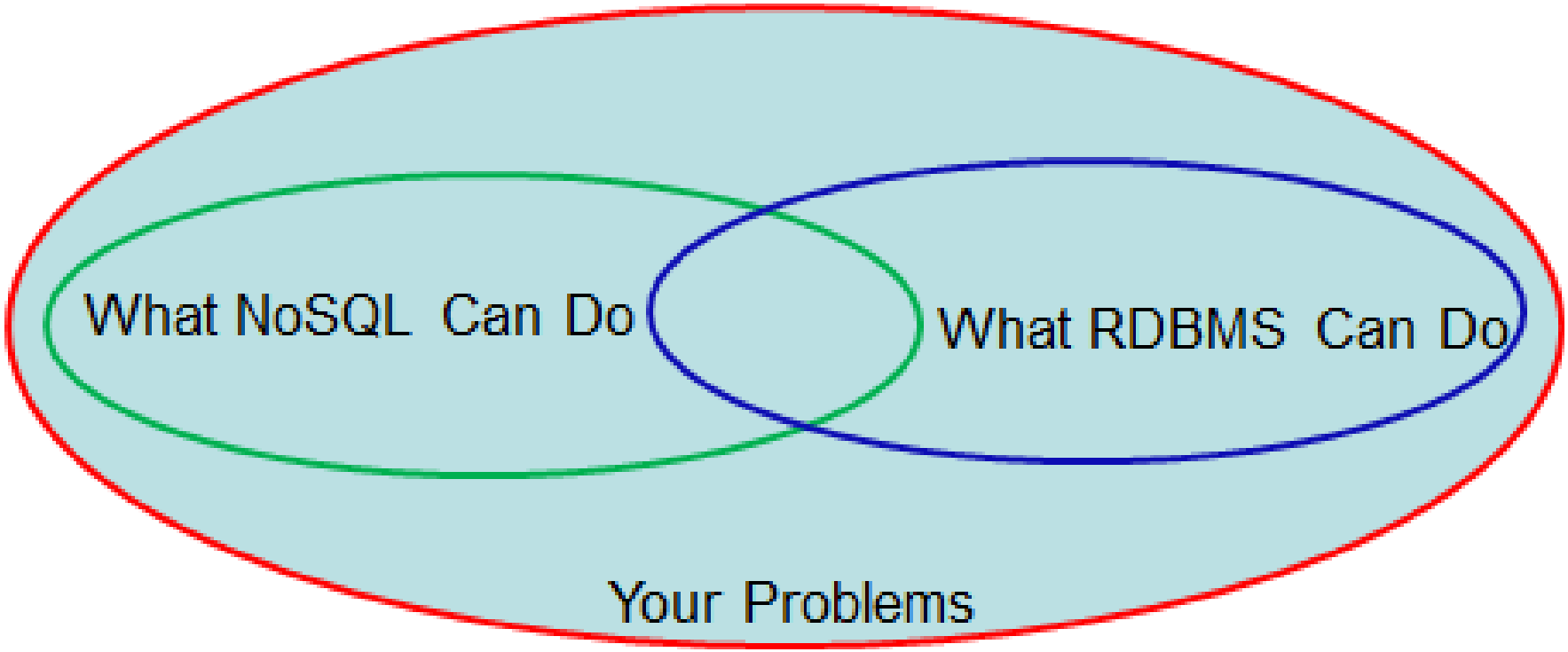
基本上一定可以使用
Basically Available: The database system always seems to work!
資料不一定隨時一致
Soft State: It does not have to be consistent all the time
但資料最後一定一致
Eventually Consistent: The system will eventually become consistent when the updates propagate, in particular, when there are not too many updates

RDBMS and NoSQL-選擇資料庫 05

- Structured and organized data **RDBMS**
- Structured query language (SQL)
- Data and its relationships are stored in separate tables.
- Data Manipulation Language, Data Definition Language
- Tight Consistency and ACID Transaction

- Unstructured and unpredictable data and Big Data **NOSQL**
- No declarative query language
- CAP Theorem and No predefined schema
- Key-Value Store, Column Store, Document Store and Graph Store
- Eventual consistency rather than ACID property
- Prioritizes high performance, high availability and scalability

RDBMS and NoSQL-選擇資料庫 06



NoSQL 與 RDBMS 都是資訊系統設計必備的資料庫系統

謝 謝 指 教