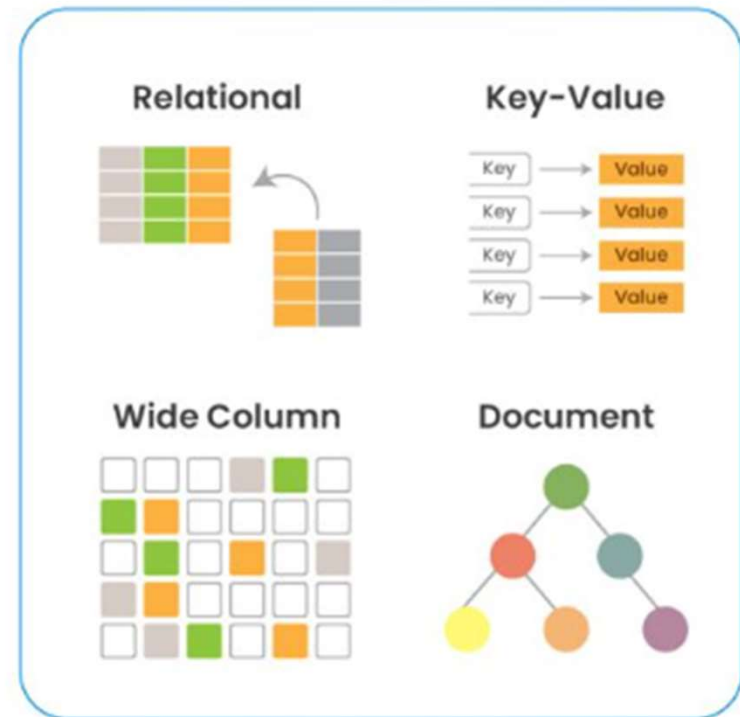
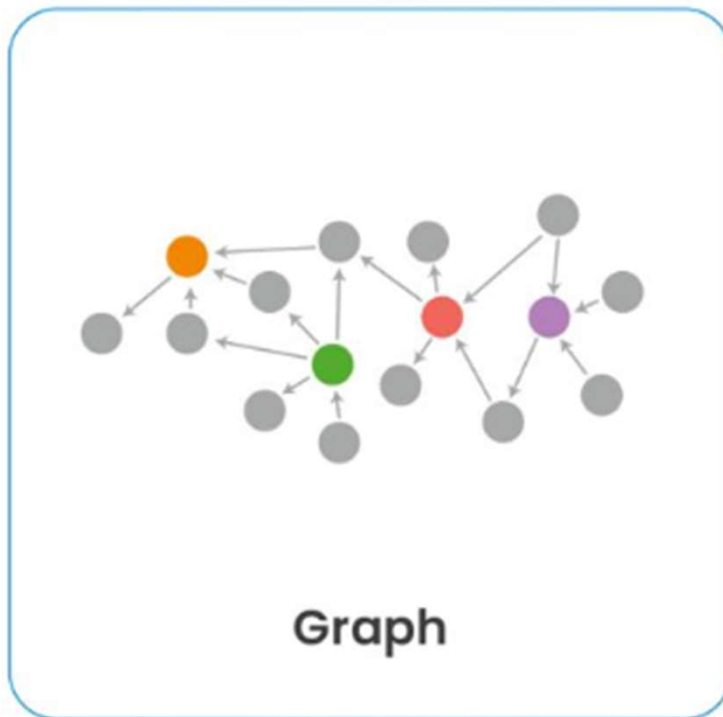


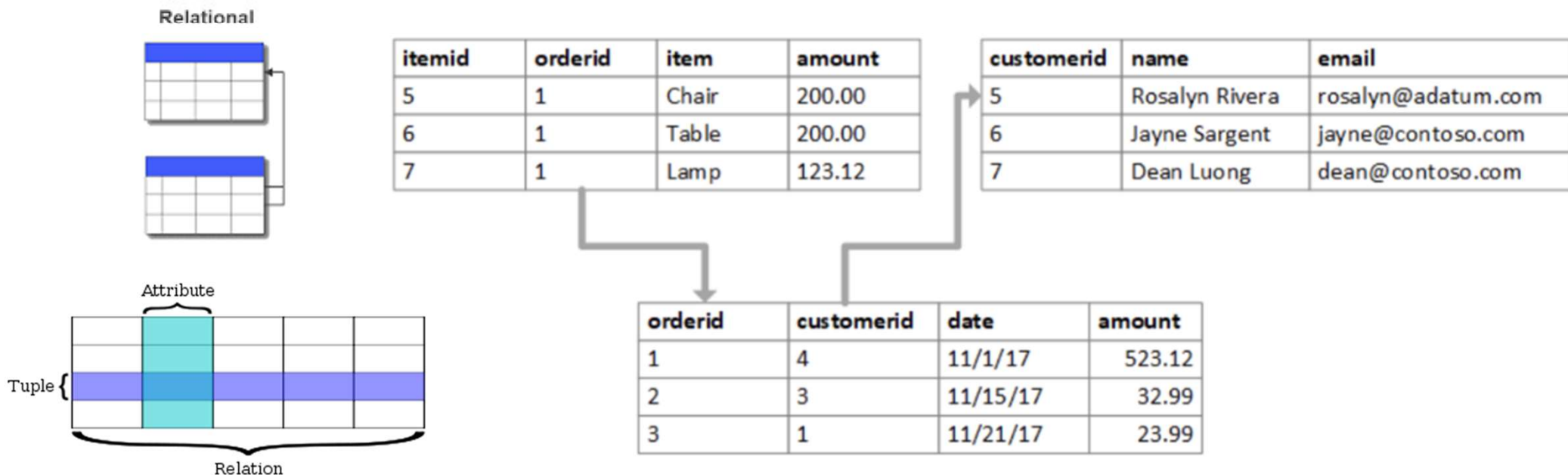
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยหอการค้าไทย
ผศ.เฉลิมชนม์ ไวศยดำรง

SQL vs NoSQL



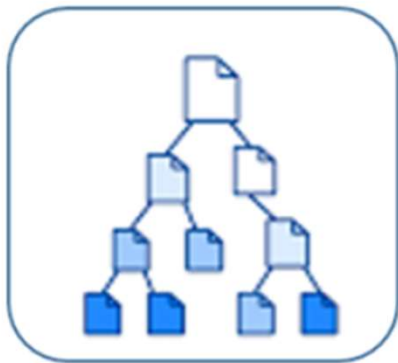
RDBMS (Relational Database Management System)

- SQL is a programming language designed to manage data stored in a relational database management system (RDBMS).
- มีการเก็บข้อมูลในรูปแบบ **Tables** (ตาราง) มีองค์ประกอบเป็น **Rows** และ **Columns**
- RDBMS ที่นิยมในตลาดที่เรารู้จักกัน ได้แก่ Oracle, Microsoft SQL Server, MySQL, PostgreSQL, IBM DB2, Microsoft Access, SQLite และ MariaDB

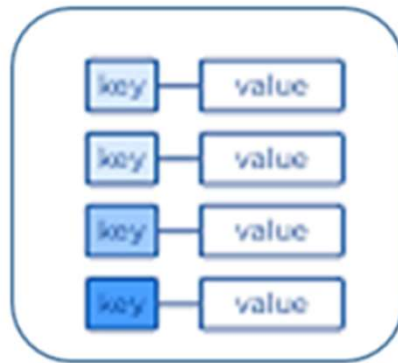


No SQL

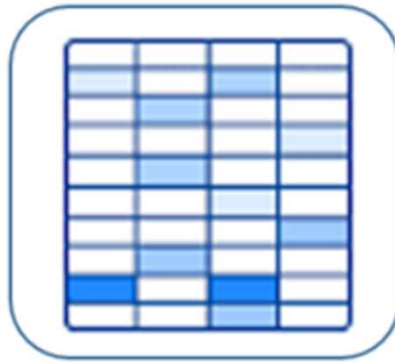
- เป็น Database อื่น ๆ ที่ไม่ได้เป็นแบบ Relational หรือมีความสัมพันธ์กันชัดเจนแบบ Pattern เหมาะสำหรับการใช้งานจำพวก Big Data และ Real-time Web Application



Document
Store



Key-Value
Store



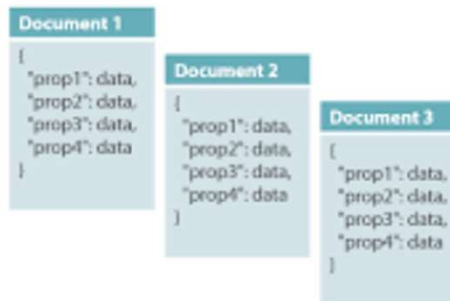
Wide-Column
Store



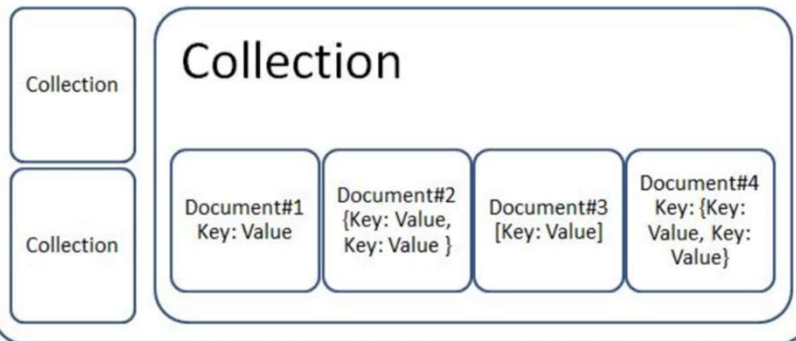
Graph
Store

- Document -> ข้อมูลและ Metadata จะเก็บเป็นลำดับชั้นในรูปแบบ Semi-structure data เช่น JSON หรือ XML ใน Database ตัวอย่าง Database Software ที่ใช้งานลักษณะนี้ได้แก่ Cosmos DB, IBM Domino, MongoDB, Couchbase, ArangoDB

| Col1 | Col2 | Col3 | Col4 |
|------|------|------|------|
| Data | Data | Data | Data |
| Data | Data | Data | Data |
| Data | Data | Data | Data |



Database



```

first_name: 'Paul',
surname: 'Miller',
cell: 447557505611,
city: 'London',
location: [45.123, 47.232],
Profession: ['banking', 'finance', 'trader'],
cars: [
  { model: 'Bentley',
    year: 1973,
    value: 100000, ... },
  { model: 'Rolls Royce',
    year: 1965,
    value: 330000, ... }
]

```

Fields

String

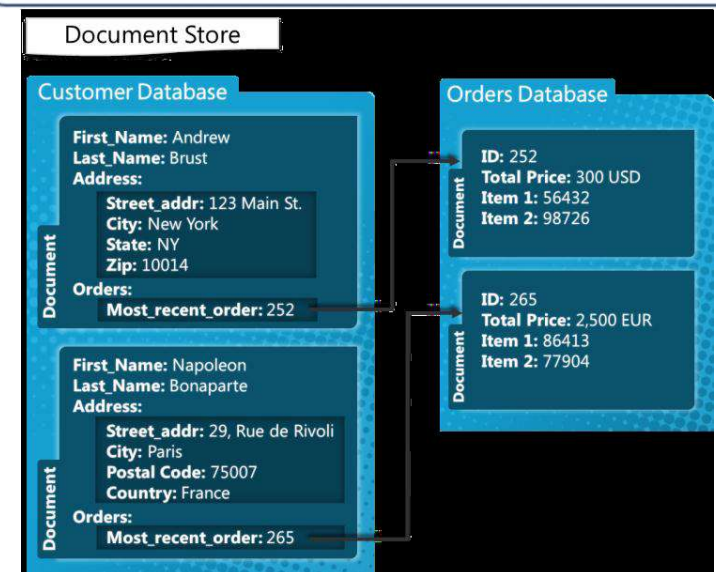
Number

Geo-Coordinates

Typed field values

Fields can contain arrays

Fields can contain an array of sub-documents



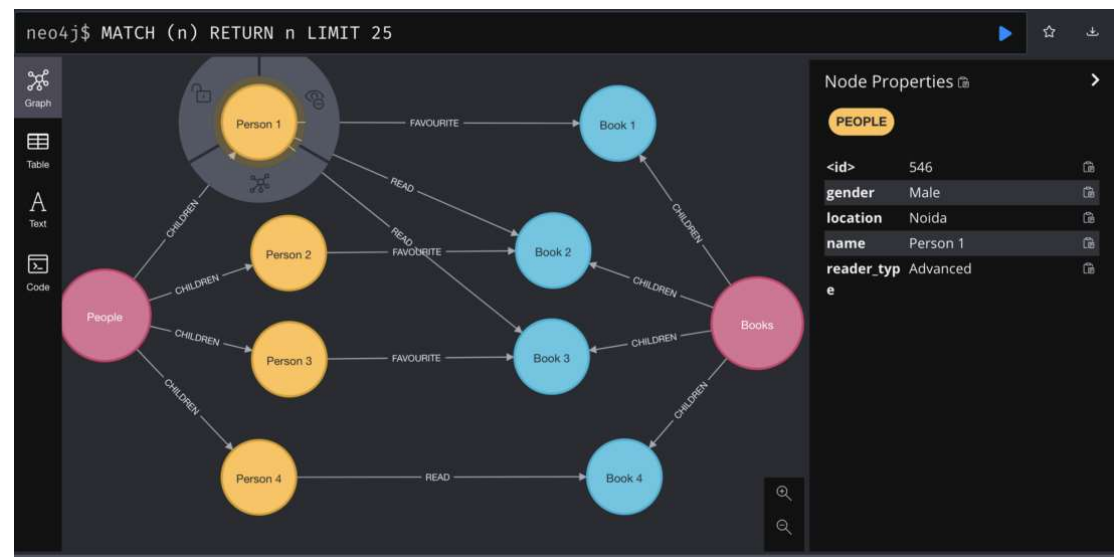
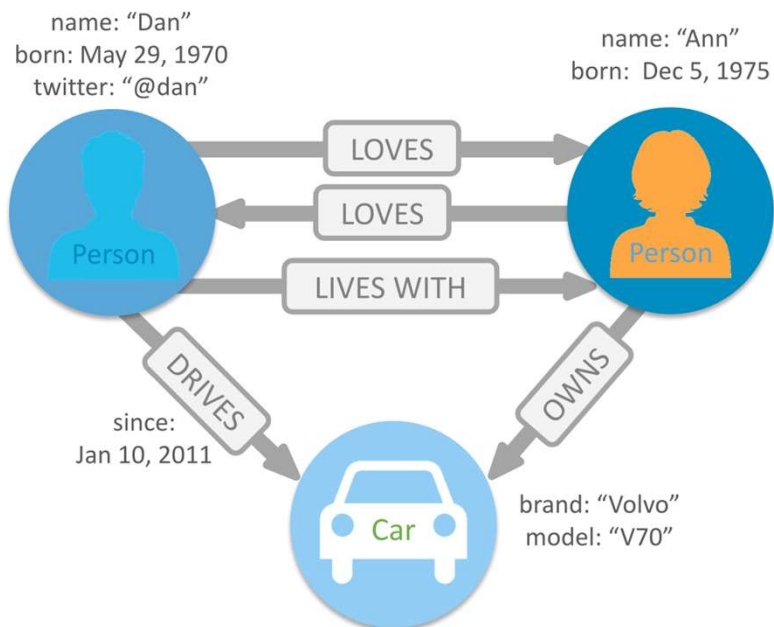
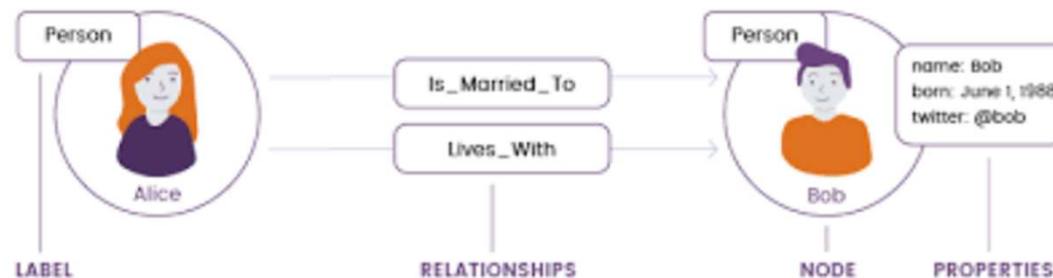
2. Key-Value -> เป็นการเก็บ Record ที่ไม่มีอะไรซับซ้อน มีแค่ Key และ Value ทำให้สามารถเข้าถึงข้อมูลได้รวดเร็ว โดยการเข้าถึงข้อมูลก็ให้ใช้ Key ก็จะได้ Value ที่ต้องการ ตัวอย่าง Database Software ที่ใช้งานลักษณะนี้ ได้แก่ Redis, Memcached, Apache Ignite, Couchbase, Dynamo

| Key | Value |
|-----|------------------|
| K1 | AAA,BBB,CCC |
| K2 | AAA,BBB |
| K3 | AAA,DDD |
| K4 | AAA,2,01/01/2015 |
| K5 | 3,ZZZ,5623 |

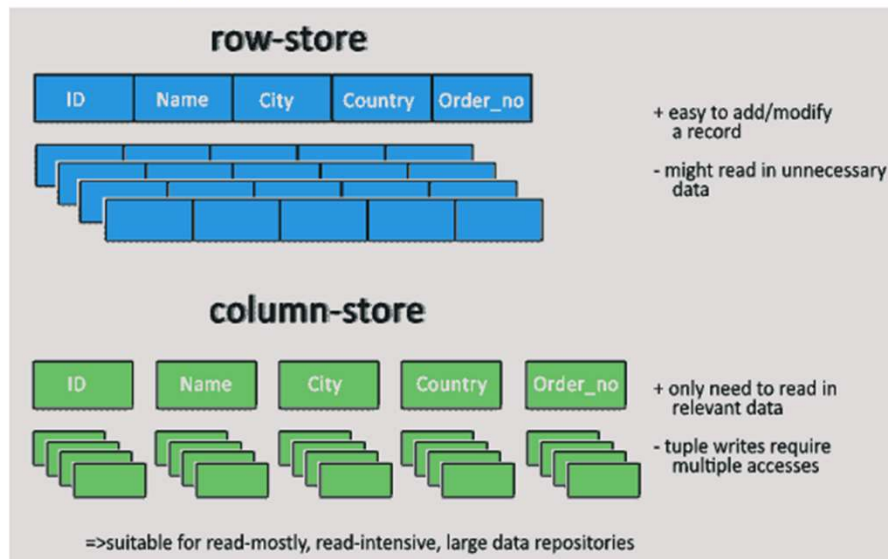
| Key | Document |
|------|---|
| 1001 | { "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" } |
| 1002 | { "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" } |

3. Graph -> ข้อมูลจะเก็บอยู่ในรูปแบบกราฟแผนภูมิ มี Node และ Edge ที่เชื่อมต่อกัน ทำให้ไม่ต้องนำข้อมูลมา JOINS กันเหมือนของ RDBMS ตัวอย่าง Database Software ที่ใช้งานลักษณะนี้ ได้แก่ ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J,

OrientDB, Virtuoso



4. Wide-Column -> รูปแบบของ Wide-Column จะบันทึกข้อมูลในรูปแบบ Tables (Rows และ Columns) แต่จะต่างจาก RDBMS ตรงที่ แต่ละ Rows จะไม่ Fix Column (ถ้าเป็น RDBMS จะ Fix มาเป็น Pattern เดียวกัน) ตัวอย่าง Database Software ที่ใช้งานลักษณะนี้ ได้แก่ Amazon DynamoDB, Cassandra, Azure Tables, Accumulo, HBase



Column family (Table)

| partition key | columns ... | | | |
|---------------|-------------|------|-------|-------|
| 101 | email | name | tel | |
| | ab@c.to | otto | 12345 | |
| 103 | email | name | tel | tel2 |
| | karl@a.b | karl | 6789 | 12233 |
| 104 | name | | | |
| | linda | | | |

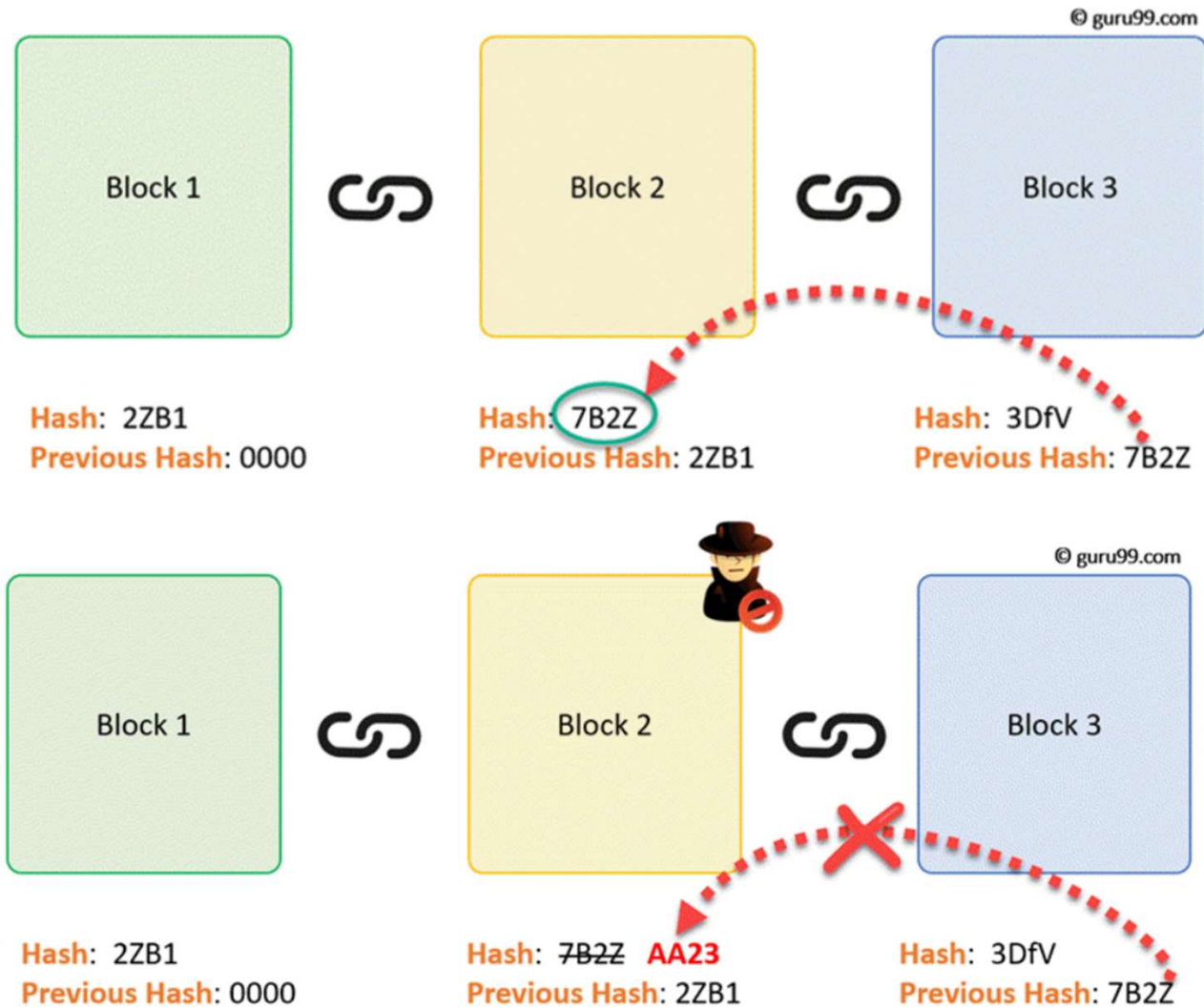
แบบ RDBMS (สีฟ้า) จะเห็นได้ว่า Columns เป็นค่าเดิมตลอด

| | 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 |

| 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 |
| ข้อมูลที่ยืนยันต้องเป็นไปตาม Condition | ต้องการเขียนไวย่ 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) |

<https://andersbrownworth.com/blockchain/>

<https://etherscan.io/blocks>



SQL

NoSQL

insert a new book record

```
INSERT INTO book (  
  `ISBN`, `title`, `author`  
)  
VALUES (  
  '9780992461256',  
  'Full Stack JavaScript',  
  'Colin Ihrig & Adam Bretz'  
);
```

```
db.book.insert({  
  ISBN: "9780992461256",  
  title: "Full Stack  
    JavaScript",  
  author: "Colin Ihrig & Adam  
    Bretz"  
});
```

return all book titles over \$10

```
SELECT title FROM book
WHERE price > 10;
```

```
db.book.find(
  { price: { >: 10 } },
  { _id: 0, title: 1 }
);
```

update a book record

```
UPDATE book
SET price = 19.99
WHERE ISBN = '9780992461256'
```

```
db.book.update(
  { ISBN: '9780992461256' },
  { $set: { price: 19.99 } }
);
```