

# Distributed Data Systems

---

MAHESH CHAUDHARI, PH.D

# Announcement

---

## Programming Assignment - 1

- MongoDB Queries, Due: February 12, 2026

## Group Project Task 1 - Finalize your storyline and datasets

- Due : February 7, 2026

# Recap

---

Course Overview

Workflow Management

Apache Airflow

Airflow DAG Creation

# NoSQL Learning Objectives

---

Understand the needs and characteristics of schemaless non-relational databases for storing and querying data in distributed settings.

Gain insights to make judgments to choose SQL or NoSQL (and which NoSQL) with various hands-on exercises.

Gain experience with NoSQL databases including MongoDB through in-class exercises and homework.

Be competent to work with Spark and MongoDB in a distributed environment including Amazon Web Services, MongoDB Atlas.

# NoSQL Interview Questions

---

What is NoSQL?

Relational Database vs. NoSQL

Impedance mismatch

Polyglot persistence

Aggregate-oriented database

Key-value database

Document database

Column family database

Graph database

Replication vs sharding

CAP Theorem

Eventual Consistency

See the Interview Questions module on Canvas

# MongoDB Interview Questions

---

MongoDB's type

Consistency

MongoDB's characteristics

Alternative databases

Supported programming languages

Index

Aggregation Operations(aggregation pipeline)

Sharding

Replication

ObjectId

**See the Interview Questions module on Canvas**

# Contents

---

Why NoSQL?

What is NoSQL?

Types of NoSQL

Choice of DBMS (SQL and NoSQL)

Document Database

# Contents

---

## Why NoSQL?

What is NoSQL?

Types of NoSQL

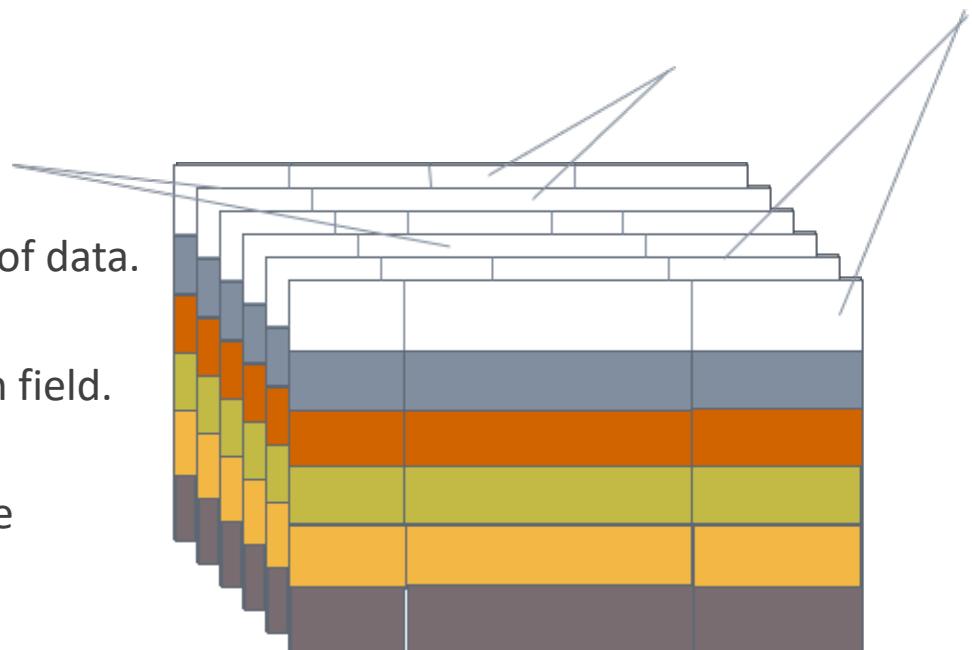
Choice of DBMS (SQL and NoSQL)

Document Database

# Review - Relational Database

A DBMS based on the relational data model.

- Relational Model
- Relation = Records
- Every relation has a schema : Description of data.
  - Specifies its name, name of each fields (column/attribute), and the type of each field.  
Ex. Students(sid : string, name: string)
  - Integrity constraints : conditions that the records in a relation must satisfy
  - Each relation has a collection of tuples



# Review - Relational Database

---

SQL (Structured Query Language)

- Manage data in a relational database
- Basic operations (CRUD)

1. Create/Insert : Define a new table, Insert a tuple into a table.

2. Read : Retrieve data from a table.

3. Update : Change an existing table and tuples.

4. Delete : Remove the tuple(s) from a table or the table.

ACID Properties?



UNIVERSITY OF SAN FRANCISCO  
CHANGE THE WORLD FROM HERE

# Review - Relational Database

---

## Pros

ACID (Atomic, Consistent, Isolated and Durable) **Transaction Management**

- **Atomicity** : Actions in transactions are carried out all or none.
- DBMS logs all actions so that it can undo the actions of aborted transaction
- **Consistency**: A transaction must change affected data only in allowed ways, according to all defined rules.
- **Isolation** : When there are several transactions happen concurrently, users should be able to understand a transaction without considering the effect of other concurrently existing transaction.
- **Durability** : Once a transaction has been successfully completed, its effects should persist even if the system crashes before all its changes are reflected on disk.

# Review - Relational Database

---

## Pros

- Standard Model
  - Different vendors' query languages are similar.
  - Transaction operations work in the similar way.



Isn't RDBMS  
perfect enough?



UNIVERSITY OF SAN FRANCISCO  
CHANGE THE WORLD FROM HERE

# Big Data

---

Volume, Variety and Velocity is beyond the ability of traditional data-processing application software can adequately handle.

- Volume : Constantly moving target - as of 2025 ranging from terabytes (TB) to Zettabytes (ZB).
- Variety : Combined structured and unstructured data.
- Velocity : High speed in creation and required speed for ingestion and analysis.
  
- Example domains
  - Internet of Things (IoT) – Smart homes, Electronic toll collection, Smart grid, Wearable, etc.
  - Health Care – Electronic health records (EHR)
  - Marketing – Consumer data

**Storing and Processing** Big Data became an important issue!

[https://en.wikipedia.org/wiki/Big\\_data](https://en.wikipedia.org/wiki/Big_data)

<https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data/>

# Big Data and Relational Databases

---

New Data Types and Large Volumes of Data

- Data and traffic increase.
- As a solution, started to store data in the clusters.

Most relational databases are not designed to run on clusters.

Reliability issues.

- Schema changes.
- Doesn't fit in a table-like structure.

# NoSQL (Not Only SQL)

Why NoSQL?

- **Impedance Mismatch**
  - Relational Model ≠ In-memory data structure (Object)
    - For application development productivity, it is better to map with in-memory data structure.
- **Large volume of data (2000s)**
  - Scale-up vs. Scale-out?
    - Run large data on a cluster of many smaller and cheaper machines. → Cheaper and Reliable.
  - Example of NoSQL database : Google BigTable, Amazon DynamoDB



<https://cloud.google.com/bigtable>

<https://aws.amazon.com/dynamodb/>



UNIVERSITY OF SAN FRANCISCO  
CHANGE THE WORLD FROM HERE

# Contents

---

Why NoSQL?

**What is NoSQL?**

Types of NoSQL

Choice of DBMS (SQL and NoSQL)

Document Database

# NoSQL (Not Only SQL)

---

Generally,

- Take schemaless data.
- Non-relational.
- Open-source.
- Trade off traditional consistency for other properties.
- Run on clusters.

# Summary - SQL and NoSQL

## SQL

### Pros

Persistent data storage.  
Concurrency.  
Standard Model.

### Cons

Impedance mismatch.  
Hard to scale.  
Fixed schema.

### Example

MySQL, PostgreSQL,  
SQL Server, Oracle

VS

## NoSQL

### Pros

Mostly open-source.  
Schemaless.  
Good for non-relational data.  
Scalable.

### Cons

Installation, toolsets still maturing.

### Example

Redis, MongoDB, Cassandra,  
OrientDB



# Postgres and NoSQL

---

Postgres supports a JSON data type.

**However,**

- Postgres does not offer any native mechanisms for data availability or scaling the database beyond a single server.
  - Lacks mechanisms for automatic failover and recovery between database replicas.
  - No native mechanisms to partition (shard) the database across a cluster of nodes.
- It is not as natural to work with JSON data in Postgres.
  - The non-standard extensions to SQL to query and manipulate JSON are not supported by most tools.

<https://www.postgresql.org/docs/9.3/static/functions-json.html>

<https://www.mongodb.com/compare/mongodb-postgresql>

# Contents

---

Why NoSQL?

What is NoSQL?

**Types of NoSQL**

Choice of DBMS (SQL and NoSQL)

Document Database

# NoSQL Database Types

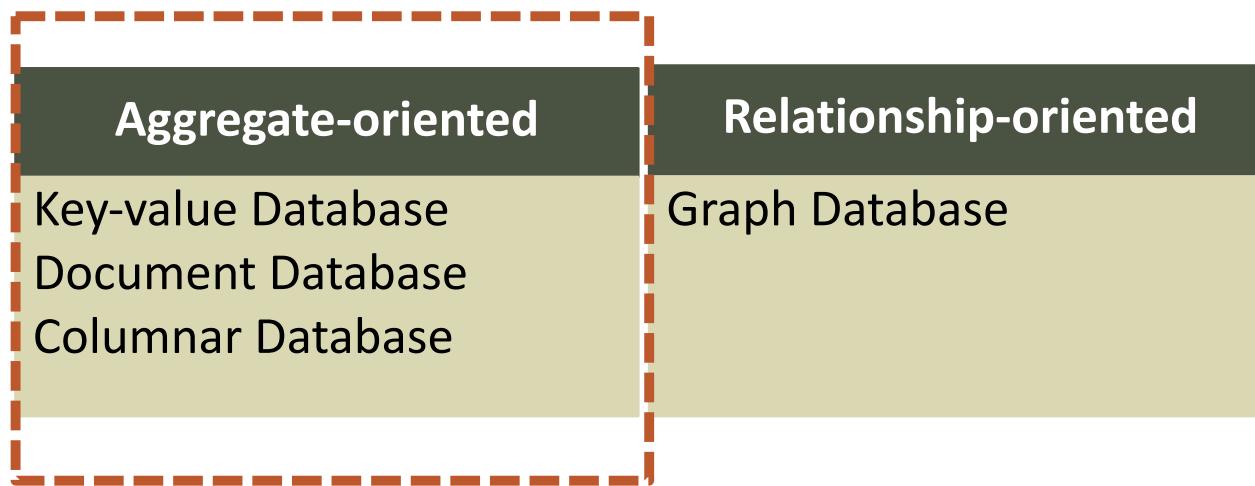
---

Aggregate-oriented	Relationship-oriented
Key-value Database Document Database Columnar Database	Graph Database



# NoSQL Database Types

---



Aggregate-oriented	Relationship-oriented
Key-value Database Document Database Columnar Database	Graph Database

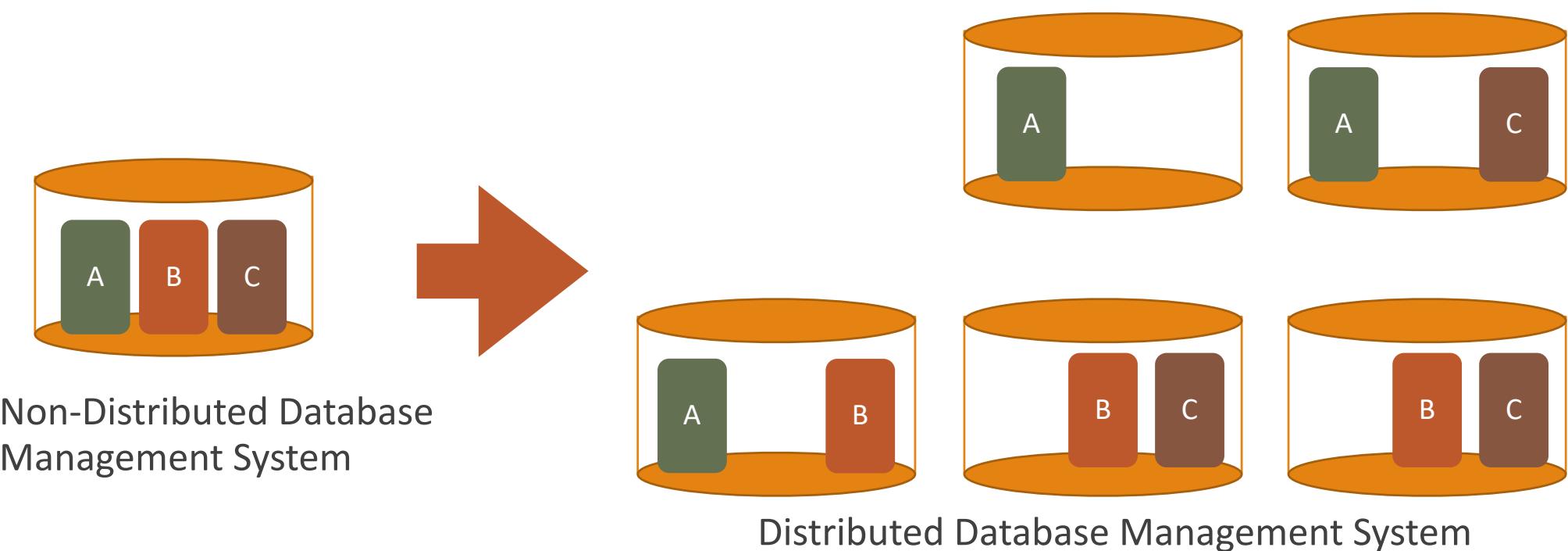
# Aggregate-oriented NoSQL Databases

Aggregate: Collection of related objects treated as a unit.

- For analyzing data, you might want to place some data together as a unit.
- On a cluster, an aggregate is stored together on a node.
- A single aggregate is a unit of atomic updates.
- Aggregate-oriented databases use aggregates indexed by **key** for data lookup.
- Aggregate-oriented databases don't have ACID transactions that span multiple aggregates.
- cf. RDBMS is aggregate-ignorant.

```
{
  "_links": {
    "self": {
      "href": "/member/109087/cart"
    },
    "http://example.com/rels/payment": {
      "href": "/payment"
    }
  },
  "_embedded": {
    "http://example.com/rels/cart-item": [
      {
        "_links": {
          "self": {
            "href": "/member/109087/cart/14418796"
          }
        },
        "id": "14418796",
        "quantity": 1,
        "expire_time": "2009-09-11T08:00:00-07:00",
        "_embedded": {
          "http://example.com/rels/sku": [
            {
              "_links": {
                "self": {
                  "href": "/skus/654654"
                },
                "http://example.com/images/cart/item": "http://example.com/product/6895/thumbnail.jpg"
              },
              "color": "Black",
              "size": "S",
              "returnable": false,
              "price": 49
            }
          ]
        }
      }
    ]
  }
}
```

# Aggregate-oriented NoSQL Databases



Aggregate-oriented	Relationship-oriented
Key-value Database Document Database Columnar Database	Graph Database

# Aggregate-oriented NoSQL Databases

## Pros

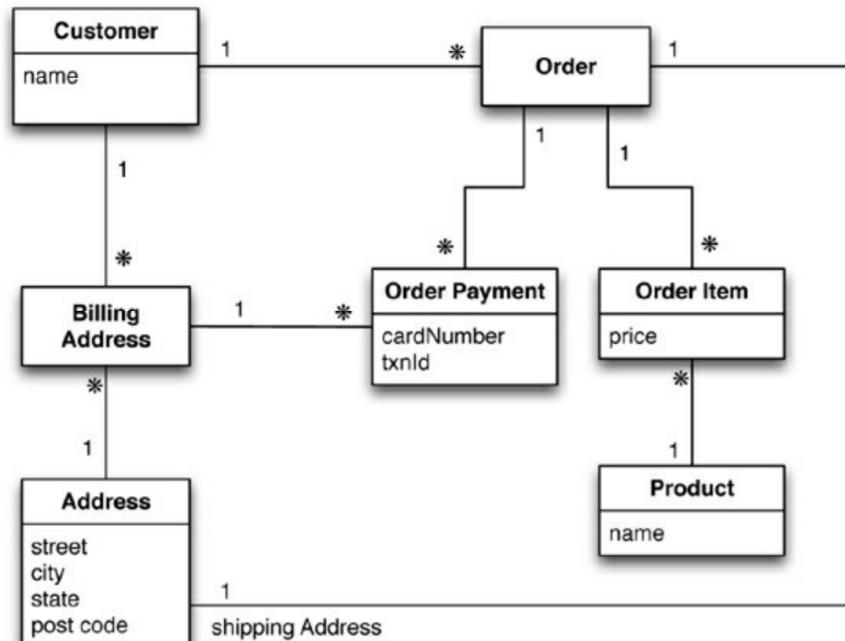
- Provides clearer semantics to consider by focusing on the aggregate unit used by applications.
- Better design choice for running on a cluster.

## Cons

- Drawing boundaries of an aggregate is not easy.
- When a goal of data management/analysis is not clear, aggregate models might not be the best choice.
- Doesn't support ACID transactions.

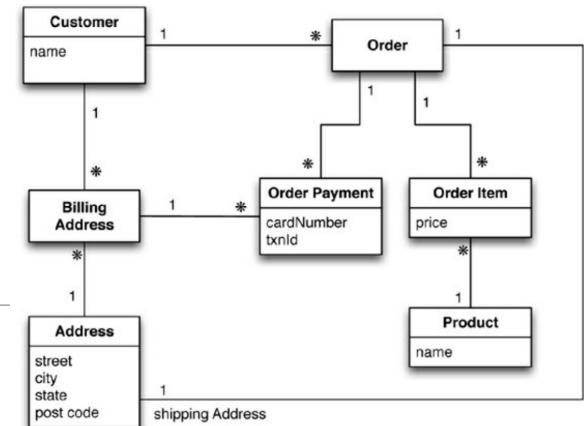
# Example 1

Draw boundaries of an aggregate.



# Example 1

Draw boundaries of an aggregate.

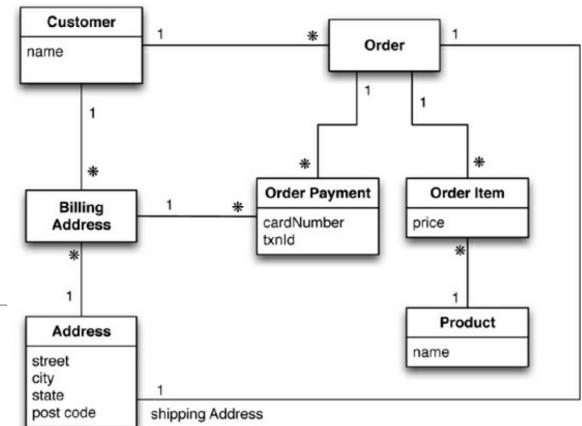
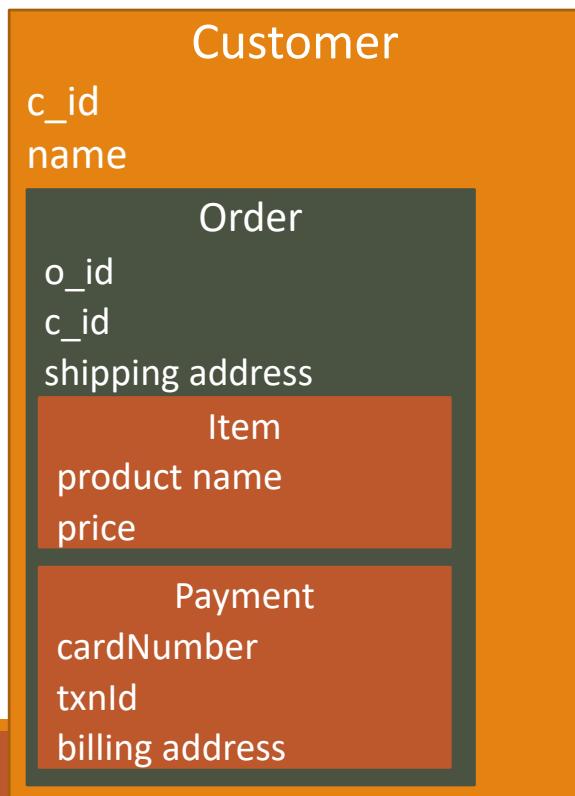


**Customer**  
c\_id  
name  
billing address

**Order**  
o\_id  
c\_id  
shipping address  
**Item**  
product name  
price  
**Payment**  
cardNumber  
txnlId  
billing address

# Example 1

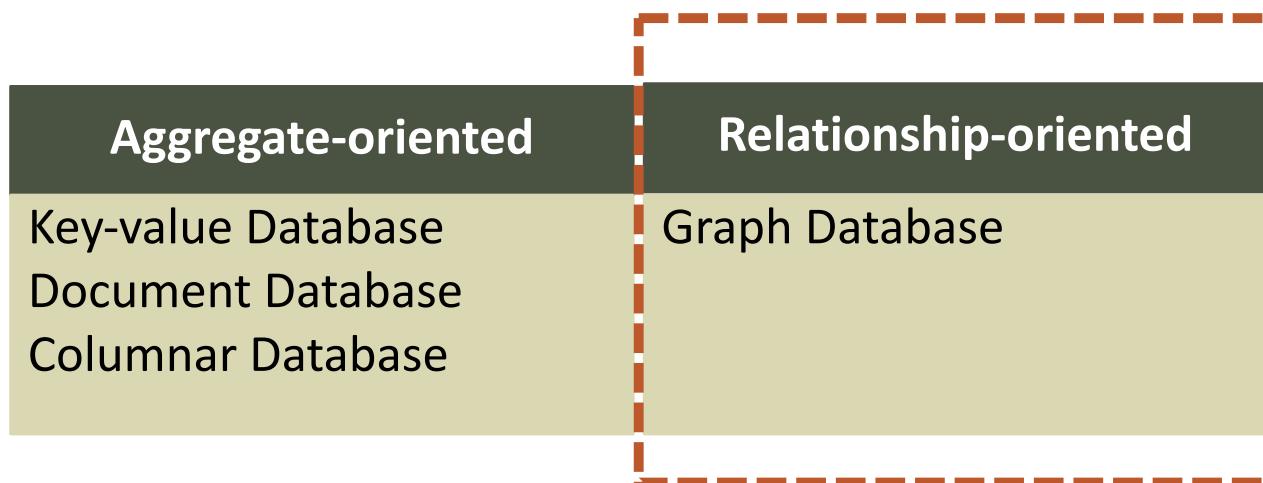
Draw boundaries of an aggregate.



Many ways to draw aggregate boundaries.  
→ Depends on how your application manipulates.

# NoSQL Database Types

---

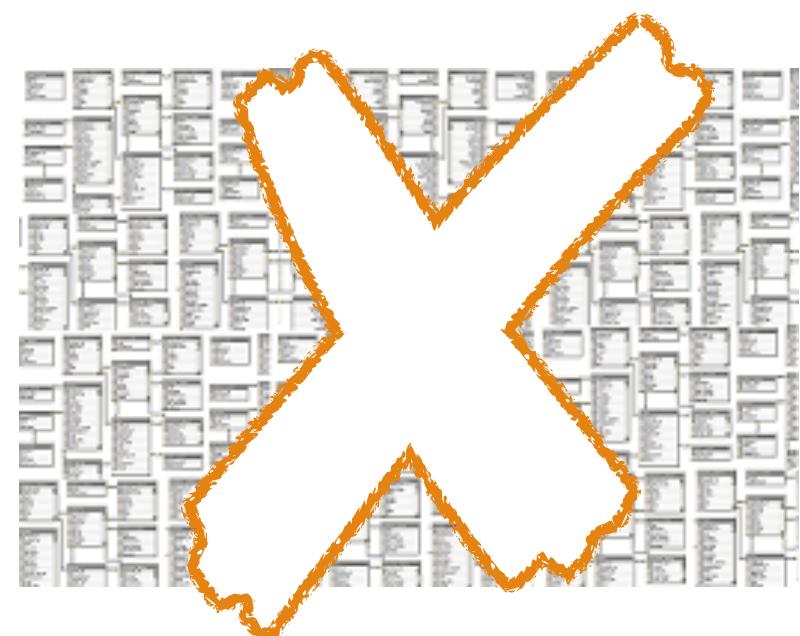


Aggregate-oriented	Relationship-oriented
Key-value Database Document Database Columnar Database	Graph Database

# Relationship-oriented NoSQL Database

Needs of relationship-oriented DB

- Relational database with complex schema
- Hard to understand, query, generalize and integrate data.



Aggregate-oriented	Relationship-oriented
Key-value Database Document Database Columnar Database	Graph Database

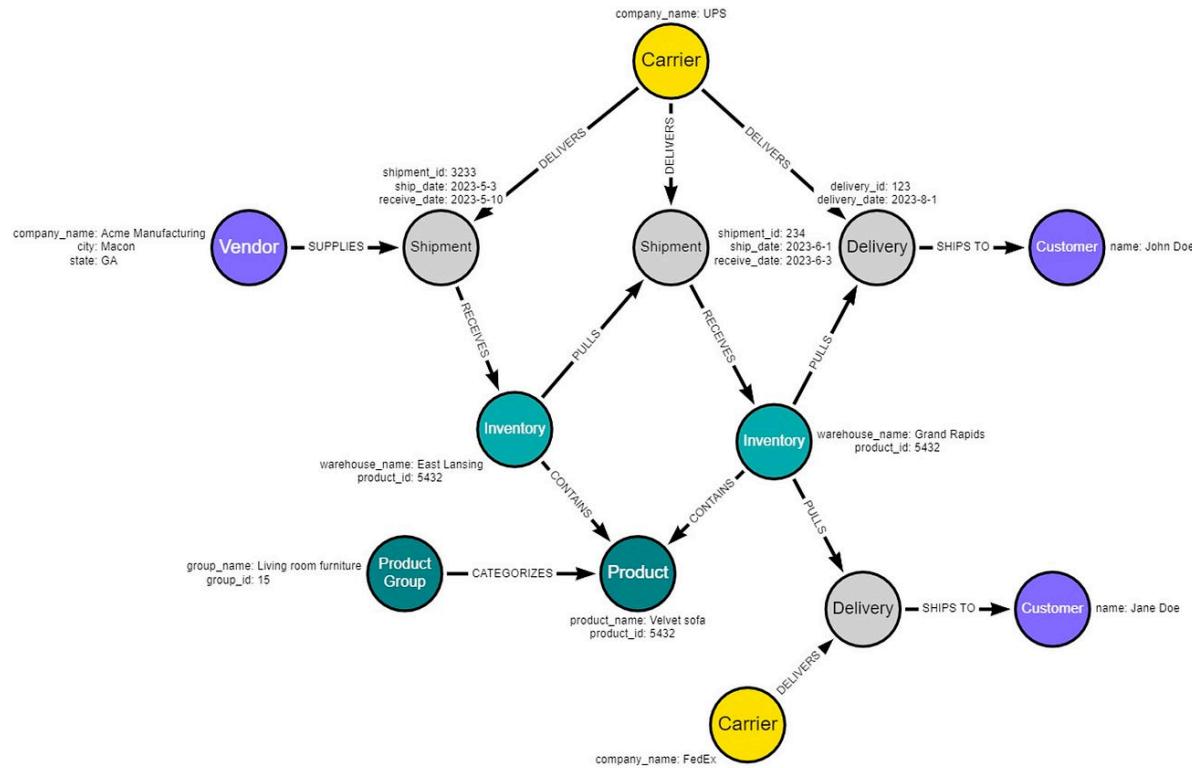
# Relationship-oriented NoSQL Database

## Graph Database

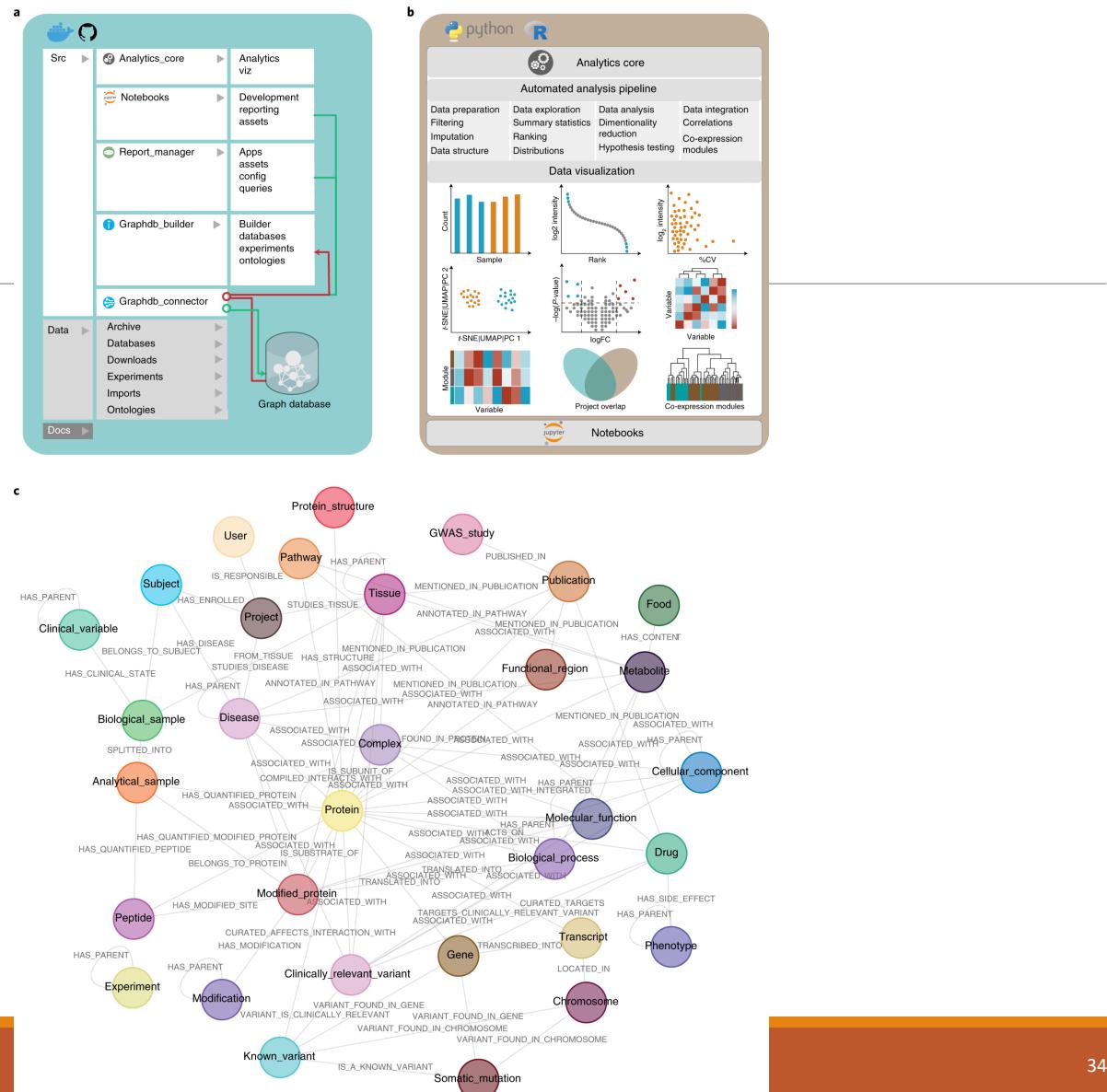
- Nodes (Object) and edges (Relationships) representation
- Relationships and their attributes are explicitly defined .
- For data with complex relationships.
- Focuses on graph traverse (more than insert.).
- cf) RDBMS : Many joins can cause poor performance.
- Running on a single server rather than distributed across clusters.

# Graph Databases

## Graph Model: Supply Chain



# Graph Databases - Healthcare Example



Aggregate-oriented	Relationship-oriented
Key-value Database Document Database Columnar Database	Graph Database

# NoSQL Database Examples

## Key-value

- Redis, Riak, Berkeley DB, etc.

## Document

- MongoDB, CouchDB, OrientDB, RavenDB, etc.

## Column Family

- Cassandra, Hbase, Amazon SimpleDB, etc.

## Graph

- Neo4J, OrientDB, FlockDB, GraphX, etc. (New kid around the block: turingdb) <https://github.com/turing-db/turingdb>

[https://en.wikipedia.org/wiki/Document-oriented\\_database](https://en.wikipedia.org/wiki/Document-oriented_database)

<https://hevodata.com/learn/columnar-databases/>

[https://en.wikipedia.org/wiki/Graph\\_database](https://en.wikipedia.org/wiki/Graph_database)

<https://www.influxdata.com/key-value-database/>



# Contents

---

Why NoSQL?

What is NoSQL?

Types of NoSQL

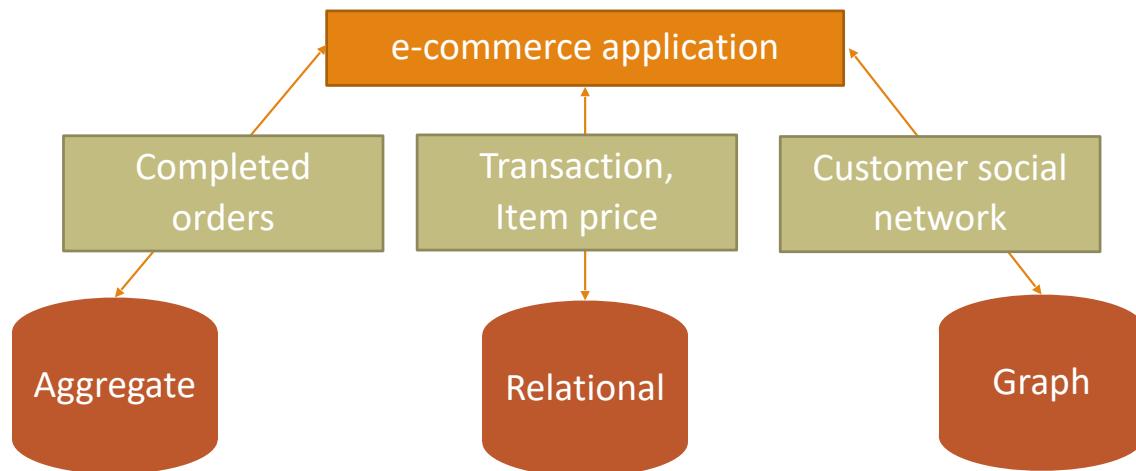
**Choice of DBMS (SQL and NoSQL)**

Document Database

# Choice of DBMS

## Polyglot Persistence

- Using multiple data storage technologies, chosen based on the way data is being used by individual applications.
- **NoSQL data stores do not replace relational databases.**



# Database Ranking (January 2024)

Rank			DBMS	Database Model	Score		
Jan 2024	Dec 2023	Jan 2023			Jan 2024	Dec 2023	Jan 2023
1.	1.	1.	Oracle 	Relational, Multi-model 	1247.49	-9.92	+2.33
2.	2.	2.	MySQL 	Relational, Multi-model 	1123.46	-3.18	-88.50
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	876.60	-27.23	-42.79
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	648.96	-1.94	+34.11
5.	5.	5.	MongoDB 	Document, Multi-model 	417.48	-1.67	-37.70
6.	6.	6.	Redis 	Key-value, Multi-model 	159.38	+1.03	-18.17
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model 	136.07	-1.68	-5.09
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model 	132.41	-2.19	-11.16
9.	↑ 10.	↑ 11.	Snowflake 	Relational	125.92	+6.04	+8.66
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	117.67	-4.08	-15.69

<http://db-engines.com/en/ranking/>

# Contents

---

Why NoSQL?

What is NoSQL?

Types of NoSQL

Choice of DBMS (SQL and NoSQL)

**Document Database**

# Document Databases

Document Databases: For storing and retrieving documents with self-contained schema including XML, JSON, etc.

- Store documents in the value part of the key-value store.

```
<?xml version="1.0" standalone="yes"?>
<BankAccount>
    <Number>1234</Number>
    <Type>Checking</Type>
    <OpenDate>11/04/1974</OpenDate>
    <Balance>25382.20</Balance>
    <AccountHolder>
        <LastName>Singh</LastName>
        <FirstName>Darshan</FirstName>
    </AccountHolder>
</BankAccount>
```

```
{
    "empid": "SJ011MS",
    "personal": {
        "name": "Smith Jones",
        "gender": "Male",
        "age": 28,
        "address": {
            "streetaddress": "7 24th Street",
            "city": "New York",
            "state": "NY",
            "postalcode": "10038"
        }
    },
    "profile": {
        "designation": "Deputy General",
        "department": "Finance"
    }
}
```

[www.kodingmadesimple.com](http://www.kodingmadesimple.com)

# Document Databases

- **Schemaless**

- Schema of the data can differ across documents, but these documents can still belong to the same collection.
- Some attributes do not exist in another document.
- New attributes can be created without defining them in the existing documents.

```
{  
  _id : 1234  
  firsname : Jane,  
  lastname : Smith,  
  address :  
    {state: CA, city: Corte Madera},  
  Interest : Books  
}
```

```
{  
  _id :2345  
  firsname : Jane,  
  lastname : Smith,  
  order :  
    {"Chodorow, Kristina. MongoDB:  
     the definitive guide. " O'Reilly  
     Media, Inc.", 2013."}  
  type : e-copy  
}
```



<http://db-engines.com/en/ranking/document+store>

# Document Databases

include secondary database models

58 systems in ranking, January 2024

Rank			DBMS	Database Model	Score		
Jan 2024	Dec 2023	Jan 2023			Jan 2024	Dec 2023	Jan 2023
1.	1.	1.	MongoDB 	Document, Multi-model 	417.48	-1.67	-37.70
2.	2.	2.	Amazon DynamoDB 	Multi-model 	80.94	-1.47	-0.61
3.	3.	3.	Databricks 	Multi-model 	80.53	+0.22	+19.71
4.	4.	4.	Microsoft Azure Cosmos DB 	Multi-model 	33.47	-1.07	-4.49
5.	5.	5.	Couchbase 	Document, Multi-model 	21.19	-0.74	-4.12
6.	6.	6.	Firebase Realtime Database	Document	16.64	-0.45	-2.12
7.	7.	7.	CouchDB	Document, Multi-model 	13.13	+0.08	-1.76
8.	8.	8.	Google Cloud Firestore	Document	11.12	+0.07	+0.03
9.	9. 	10. 	Realm	Document	8.11	+0.15	-0.22
10.	10. 	9.	MarkLogic	Multi-model 	7.78	-0.01	-1.07



UNIVERSITY OF SAN FRANCISCO  
CHANGE THE WORLD FROM HERE

# Contents

---

Why NoSQL?

What is NoSQL?

Types of NoSQL

Choice of DBMS (SQL and NoSQL)

Document Database

# References

---

Sadalage, Pramod J., and Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2012.

Redmond, Eric, and Jim R. Wilson. *Seven databases in seven weeks: a guide to modern databases and the NoSQL movement*. Pragmatic Bookshelf, 2012.

Date, C. J. "Database systems." *Vols. I & II, Narosa Pub* (1986).