CADS
FUTURE PROOFING THE WORLD!

NoSQL

Introduction to NoSQL
Problem of RDBMS

- Relational Database Management System (RDBMS) allows user to **interact with relational databases**.
- Most commercial RDBMS use **SQL** to access the databases.
- Example of RDBMS functions:
  - Create
  - Read
  - Update
  - Delete

  **collectively known as CRUD*

Introduction to NoSQL
Problem of RDBMS

CADS
FUTURE PROOFING THE WORLD!

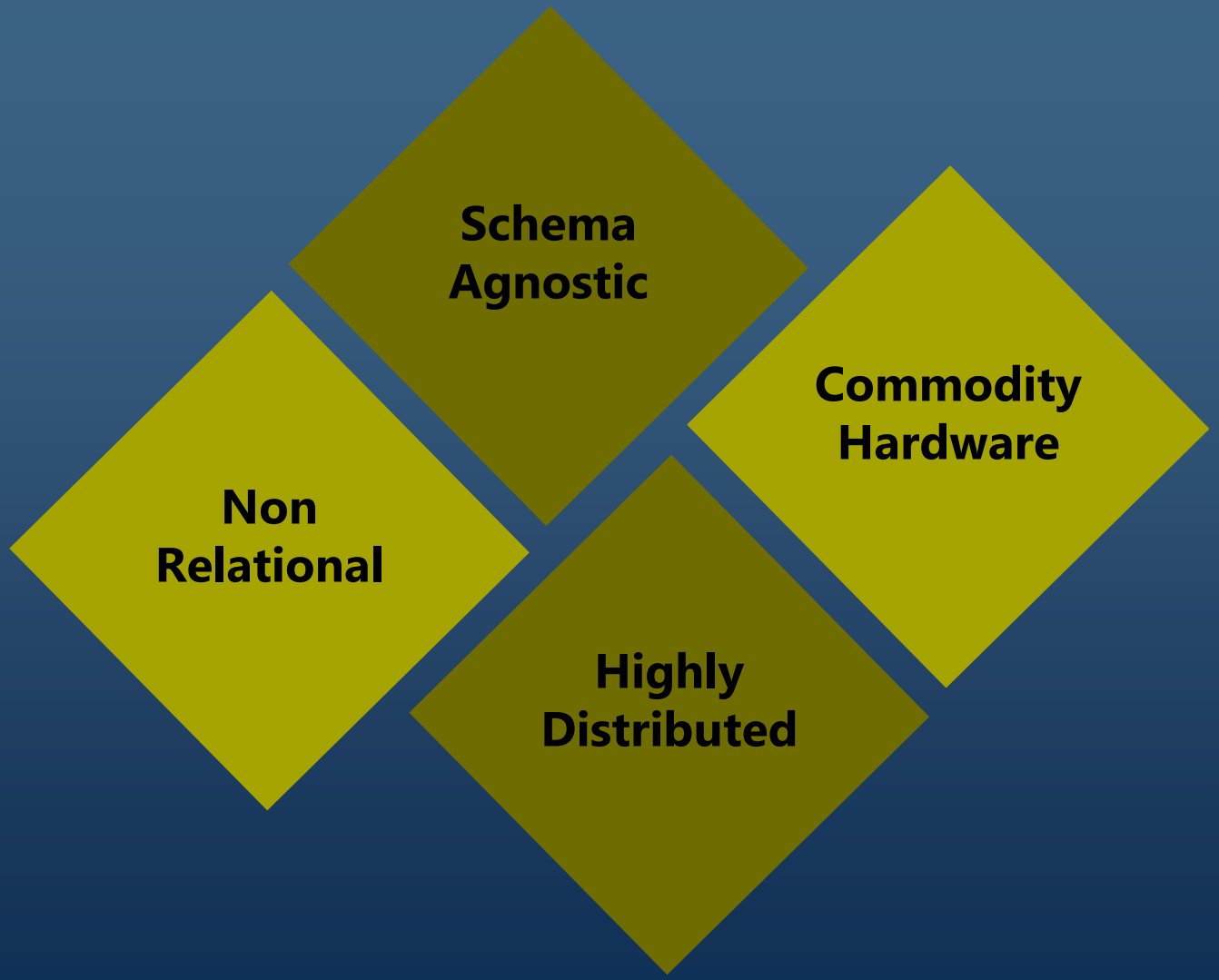
Sparse data problem	Query speed is slow	Designed to run on a single server	Not everything fits well into rows and columns
This happens when few records have values in a column, whereas other records have null values in that columns.	Join information and add constraints across tables at query time, hence the database engine has to evaluate many tables.	Consider the case of Big Data: Largest available single server couldn't even store or process all the Big Data.	For example: semi-structured data such as JSON and XML documents

Introduction to NoSQL
Background of NoSQL

- NoSQL coined in 1998 to name a relational database that did not have a SQL (Structured Query Language) interface, i.e. **N**ot **O**nly **SQL**.
- Brought back in 2009 during a meetup about distributed database.

Introduction to NoSQL

Common Features of NoSQL



CADS
FUTURE PROOFING THE WORLD!

Schema Agnostic

- A schema is not required in NoSQL, hence provide the freedom to store data without designing any up-front schema design.

CADS
FUTURE PROOFING THE WORLD!

Non-Relational

- Data is stored as an aggregate in NoSQL, in which an information could be duplicated in different rows that use the information.

CADS
FUTURE PROOFING THE WORLD!

Commodity Hardware

- NoSQL can perform well with cheap off-the-shelf servers. In fact, it adds scalability to NoSQL databases by adding more of these cheap servers.

CADS
FUTURE PROOFING THE WORLD!

Highly Distributed

- A cluster of servers can be used as a single large database server. NoSQL allows the servers to talk to each other to handle the distributed queries.

Introduction to NoSQL
Consistency vs Availability

- NoSQL loses the support for **ACID** principles as trade-off for increased **scalability** and **availability**.
- NoSQL systems is inherently bound to the **CAP Theorem**.

CADS
FUTURE PROOFING THE WORLD!

WHAT IS ACID?

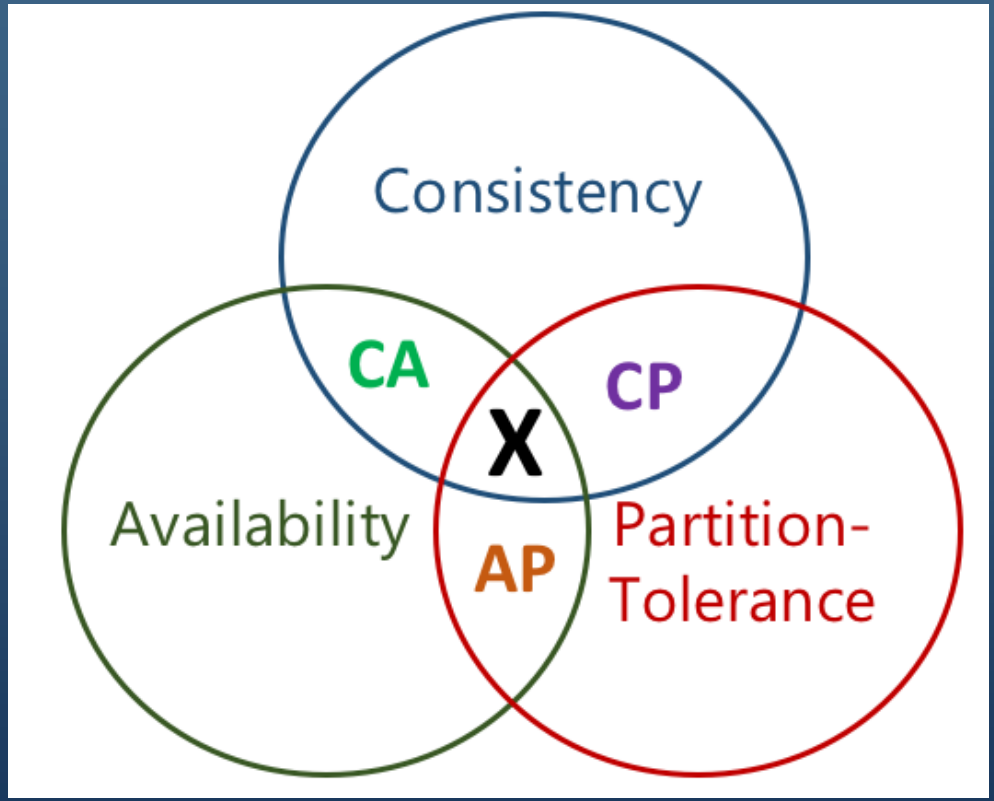
- **Atomicity:** Each operation affects the specified data, and no other data, in the database.
- **Consistency:** Each operation moves the database from one consistent state to another.
- **Isolation:** One operation in-flight does not affect the others.
- **Durability:** The database will not lose your data once the transaction reports success.

WHAT IS CAP?

- **Consistency:** Data is the same for all parts of the cluster eventually.
- **Availability:** Data is still available after partitioning.
- **Partition-Tolerance:** Database is still able to function if some parts of the cluster are not communicating with each other and will correct itself when communication is restored.

Introduction to NoSQL
CAP Theorem

- Described the **trade-offs** involved in distributed system.
- CAP theorem states that a distributed system **cannot have all three features at the same time**..



CADS
FUTURE PROOFING THE WORLD!

CA

Consistency & Availability

- Some NoSQL databases provide consistency by adding read-only replicas of data on additional nodes.
- Advantage:
  - Improve read performance
  - Provide greater availability (at least for read, not write)

CADS
FUTURE PROOFING THE WORLD!

AP

Availability & Partition-Tolerance

- Some NoSQL databases ensure the nodes in the cluster remain online even if they are unable to communicate with each other.
- As restoration step, data would be synchronized once the partition is resolved (However, the consistency of data is not guaranteed either during or after the partition).

CADS
FUTURE PROOFING THE WORLD!

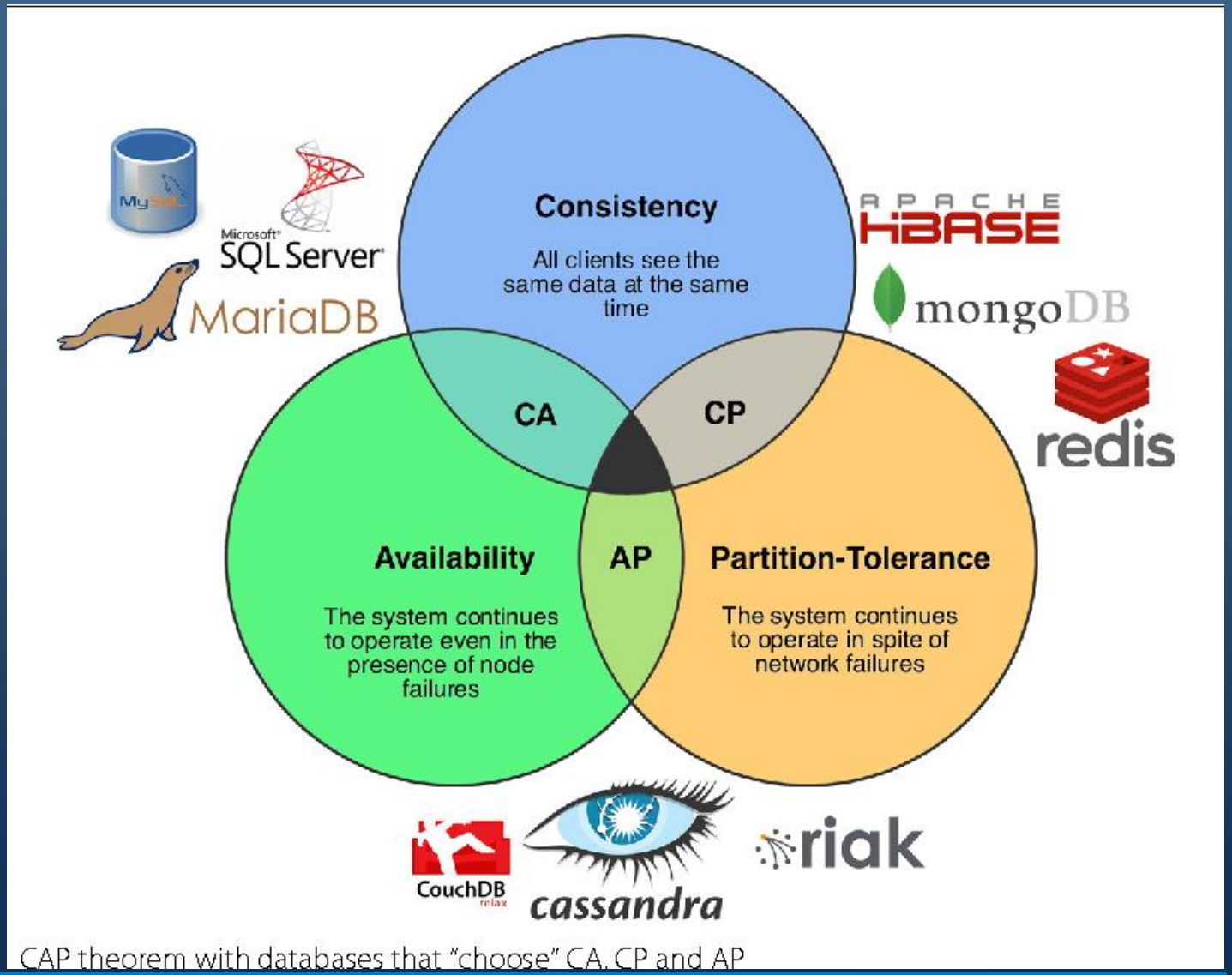
CP

Consistency & Partition-Tolerance

- Some NoSQL databases maintain the consistency of data from all nodes and preserve the partition-tolerance by making data unavailable when a node goes down.

CADS
FUTURE PROOFING THE WORLD!

EXAMPLES OF DATABASE IN DIFFERENCE COMBINATION OF CAP



CAP theorem with databases that "choose" CA, CP and AP

Source: https://www.semanticscholar.org/paper/Choosing-the-right-NoSQL-database-for-the-job%3A-a-Louren%C3%A7o-Cabral/1371cf364fddbf01f3e404fca203cad5e49cec9a

CAP Theorem: Main Takeaway

CADS
FUTURE PROOFING THE WORLD!

CAP Theorem is not “binary” decision. For example:

- AP systems focus on offering Availability and Partitioning – but are not inconsistent in some extent.
- CP systems focus on Consistency and Partitioning - but are not unavailable in some extent.

Hence, CAP Theorem offers certain degree in tolerating the Consistency, and Availability, as well as Partition-Tolerance.

CADS
FUTURE PROOFING THE WORLD!

The Center of Applied Data Science

Introduction to NoSQL
Advantage of NoSQL in Big Data

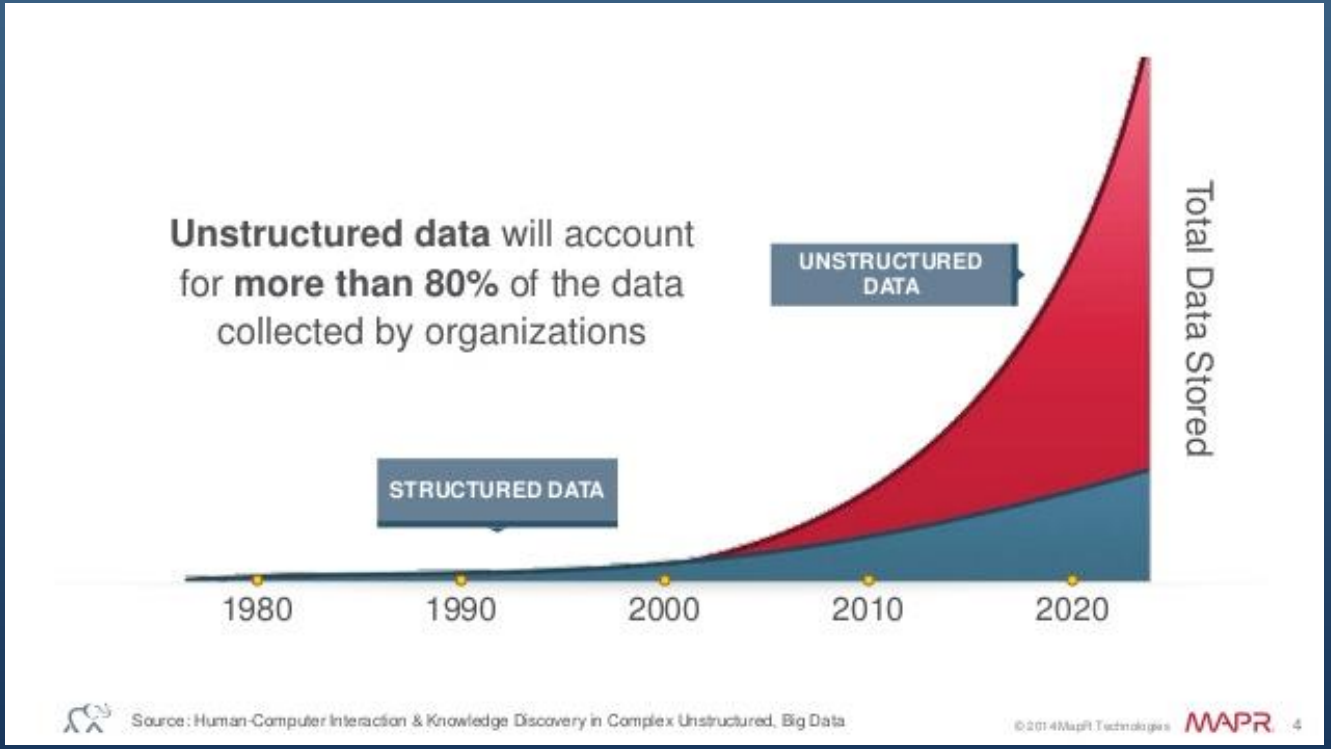
"Most people who choose NoSQL as their primary data storage are trying to solve two main problems: scalability and simplifying the development process," *said Danil Zburivsky, solutions architect at Pythian*

Source: https://sdtimes.com/big-data/number-of-nosql-options-

Advantage of NoSQL in Big Data

1

NoSQL Databases ease the storage of unstructured data due to the feature of Schema Agnostic



Source: https://www.huffingtonpost.com/sriram-krishnan2/ex-summly-team-launches-c_b_9412276.html

Advantage of NoSQL in Big Data

CADS
FUTURE PROOFING THE WORLD!

2

NoSQL Databases can store and process Big Data in real time

Introduction to NoSQL

Advantage of NoSQL in Big Data

3

NoSQL Databases ease the scalability of handling the volume of Big Data **(Horizontal Scalability)**

Introduction to NoSQL
Industrial Example

CADS
FUTURE PROOFING THE WORLD!

Graph database are used to store all the contacts and keywords of Gmail account.

INDUSTRY:
Internet-Related Services

Google built Bigtable and use it to store data of few applications such as Google Earth, Google Finance, and web indexing

USE CASE:
Specialized Storage for Unstructured Data

CADS
FUTURE PROOFING THE WORLD!

The Center of Applied Data Science

Introduction to NoSQL
NoSQL Data Modeling

Although schema is not required in NoSQL, there may be an issue of using bad assumptions of data and without structure planning properly. Hence, data modelling is still necessary for NoSQL.

CADS
FUTURE PROOFING THE WORLD!

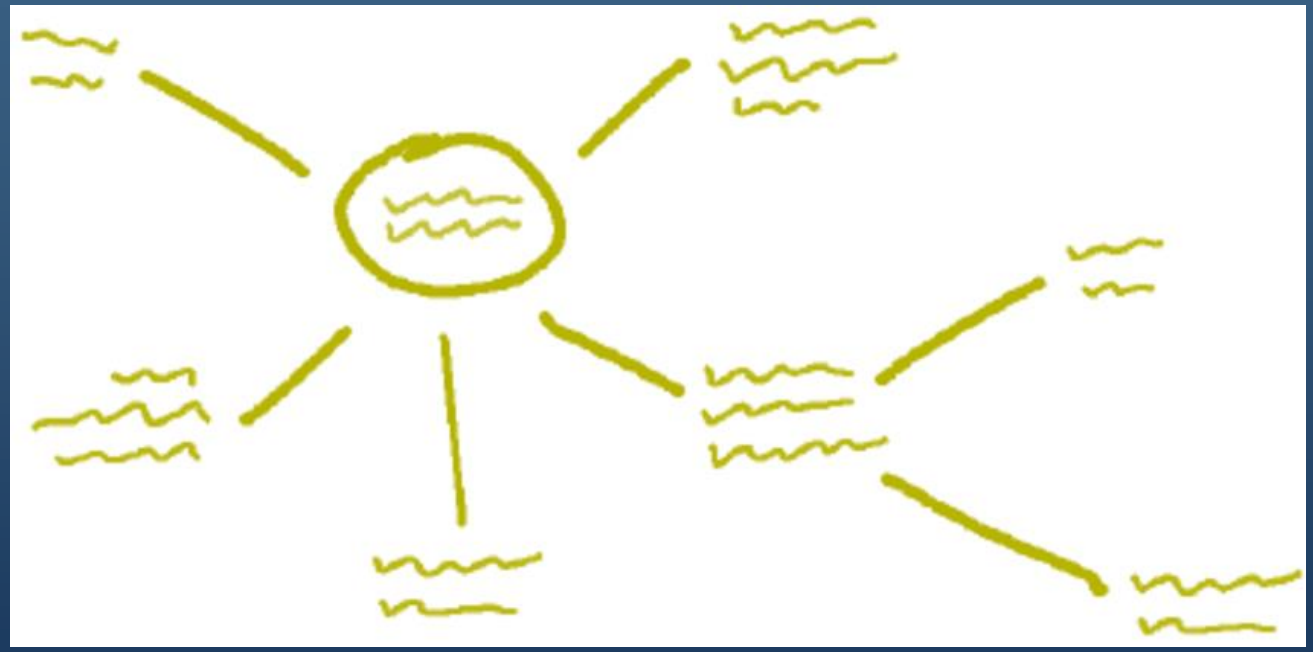
Two Ways of Performing Data Modeling:

- Mind Map
- JSON Notation

Introduction to NoSQL

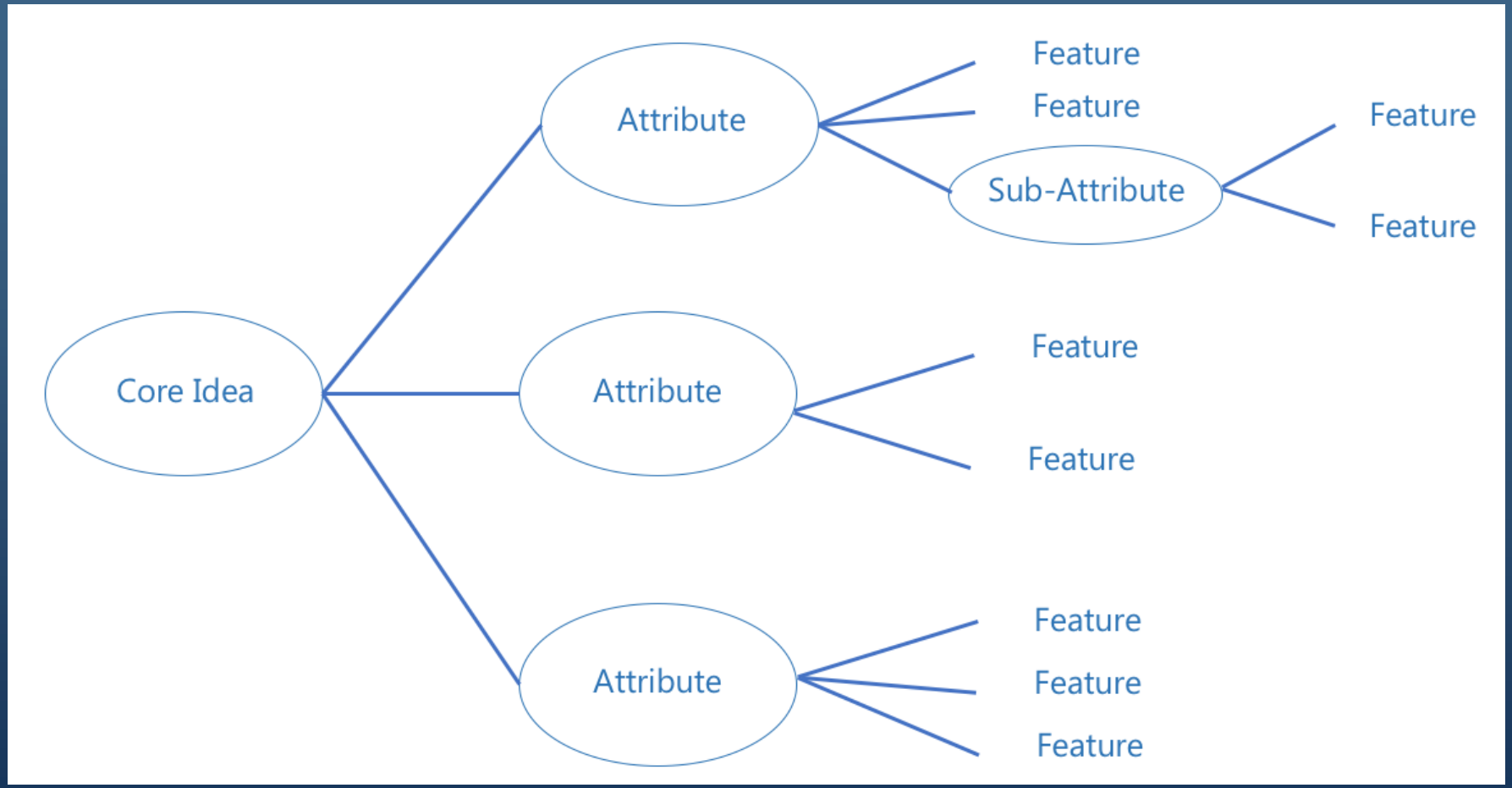
NoSQL Data Modeling: Mind Map

- Mind map is a diagram to represent ideas visually.
- It allows users to sort through different details and establish relationships among the details easily.
- In NoSQL, it helps to determine the structure of databases.



Introduction to NoSQL

NoSQL Data Modeling: Mind Map



Case Study: Customer 360 View

An online shopping system would like to aggregate customer profiles with NoSQL database. They hire you to be the developer to design and employ the NoSQL database.

The company will provide the html web interfaces and you have to store all the entries or data in the database. In general, they would like to create a database that stores customer-centric data, and the data are mostly collected from customer information, order, and payment information.

Case Study: Customer 360 View

Customer Information

ID	11897263	
Name	Jammie	
Email	jammie@gmail.com	
Phone	6012-3456 789	
Shipping Address	Street	123, Jalan Kerinchi
	City	Kerinchi
	State	Selangor
	Zip	59200
	Country	Malaysia

Case Study: Customer 360 View

My Order

Order Number	Date Created	Tracking Number	Courier Company	Order Amount
20181371740-9008	2018-01-23 21:32:11	4003454341	BlackCat	130.16
20171321585-6892	2017-12-27 12:43:31	613283823	WhiteRabbit	220.85
...	...	...	...	...

CADS
FUTURE PROOFING THE WORLD!

Case Study: Customer 360 View

Order Number: 20181371740-9008

Item Number	Quantity	Unit Price
X1234567	1	12.00
X9087573	2	4.80
...	...	...

Shipping Fee	Total Amount
8.00	130.16

CADS
FUTURE PROOFING THE WORLD!

Case Study: Customer 360 View

Payment Information

Payment Type	Debit Card
Bank	ABC Bank
Account Number	1234512345

CADS
FUTURE PROOFING THE WORLD!

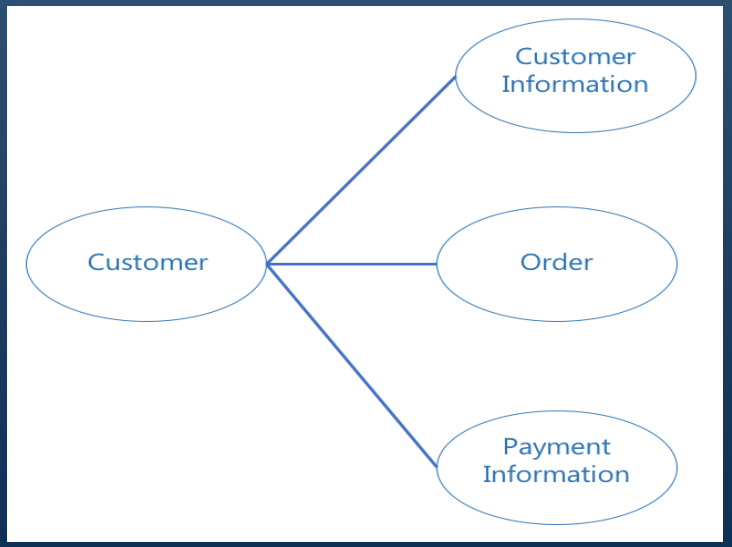
Case Study: Customer 360 View

STEP 1: Define the core idea/ business need

Case Study: Customer 360 View

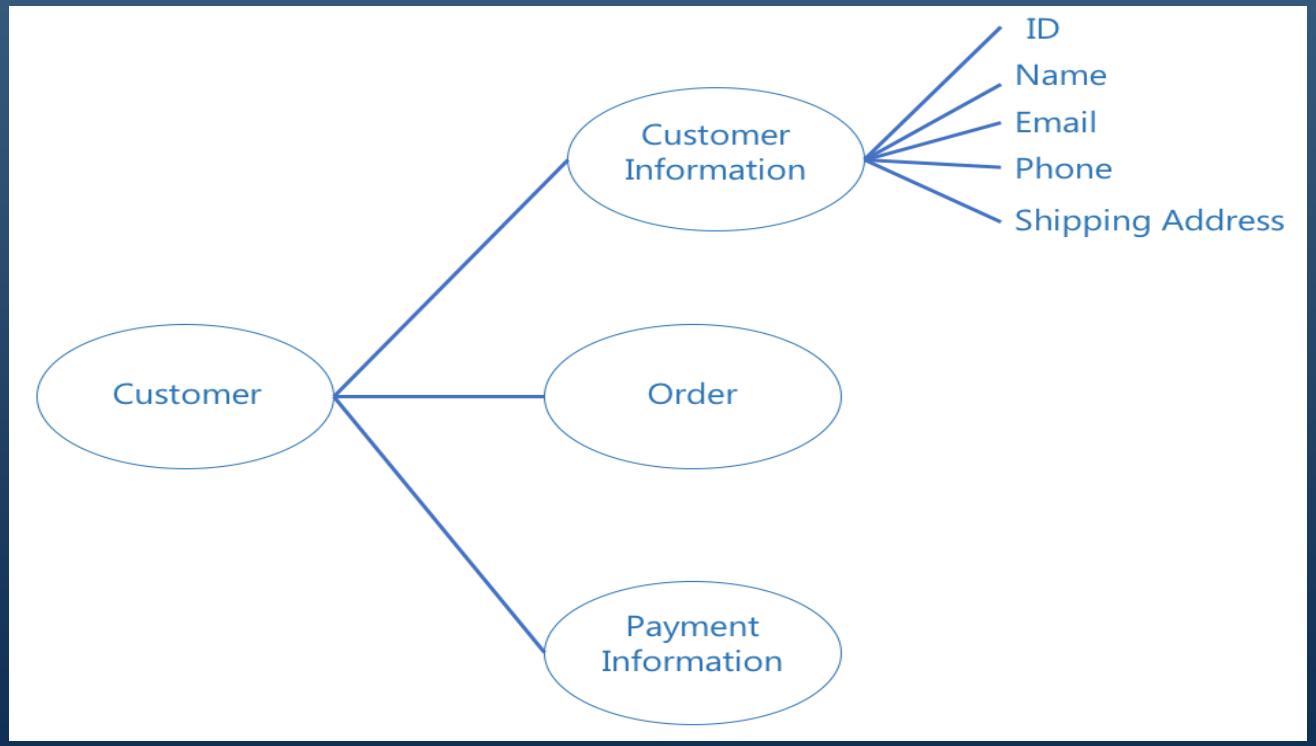
STEP 2: Define the attributes that describe the core idea

Clue: In general, they would like to create a database that stores customer-centric data, and the data are mostly collected from customer information, order, and payment information.



Case Study: Customer 360 View

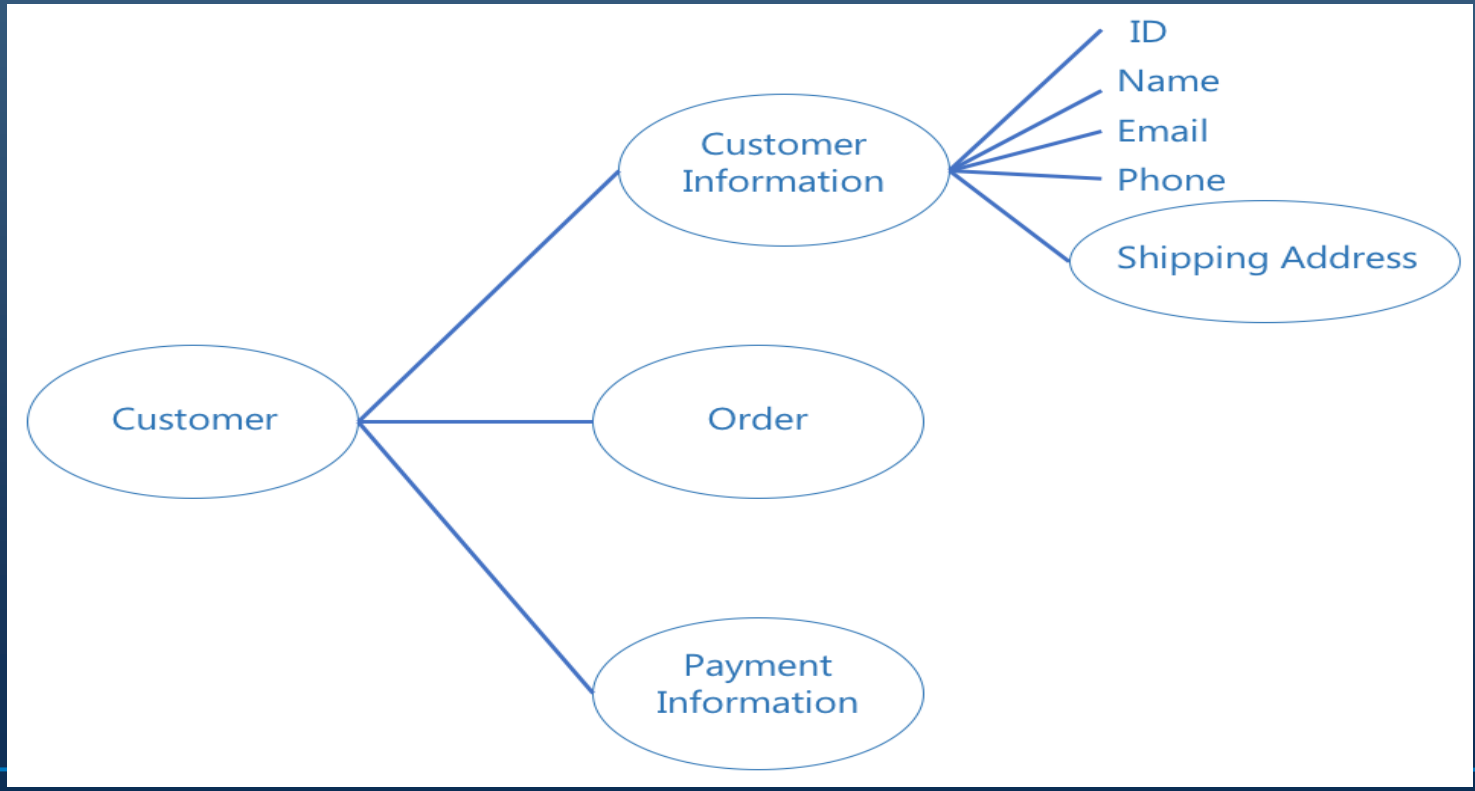
STEP 3: Define the sub-attributes/features for each attribute



Something Wrong?

Case Study: Customer 360 View

STEP 3: Define the sub-attributes/features for each attribute

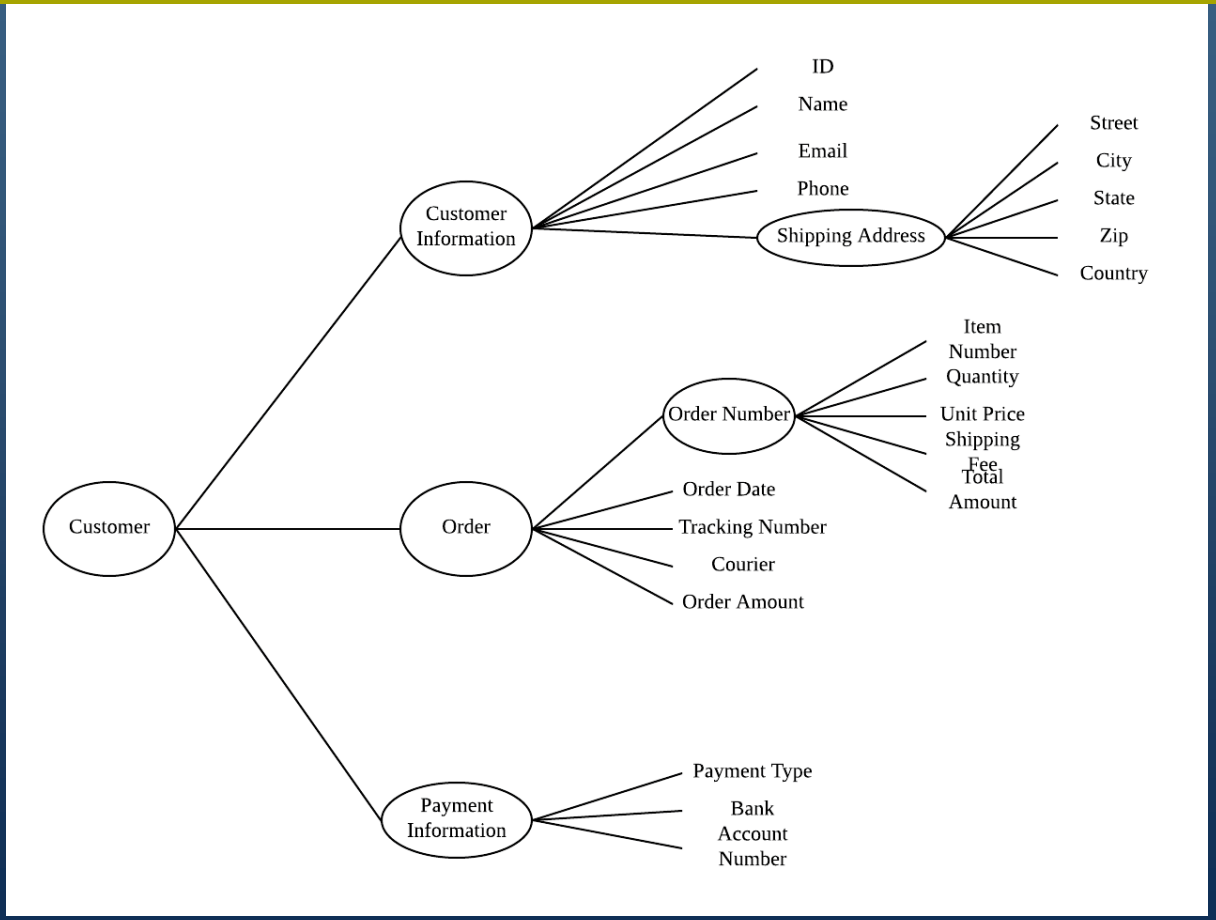


CADS
FUTURE PROOFING THE WORLD!

Case Study: Customer 360 View

What Next?

Case Study: Customer 360 View



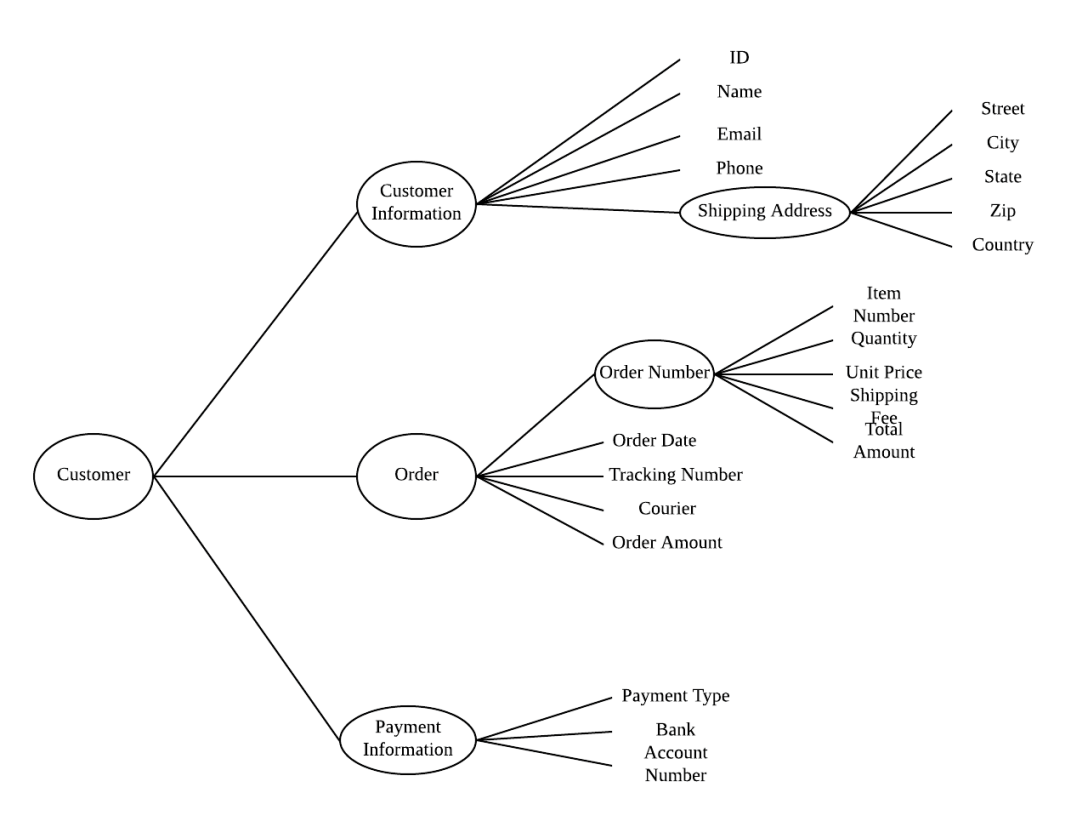
*Created with Lucidchart

Case Study: Customer 360 View

- JSON is a tagged language and is used to describe the instances.
- It is a text-based method to define the structure of a NoSQL database.

```
{
  "employees": {
    "employee": [
      {
        "id": "1",
        "firstName": "Tom",
        "lastName": "Cruise",
        "photo": "https://jsonformatter.org/img/tom-cruise.jpg"
      },
      {
        "id": "2",
        "firstName": "Maria",
        "lastName": "Sharapova",
        "photo": "https://jsonformatter.org/img/Maria-Sharapova.jpg"
      }   ]
  }
}
```

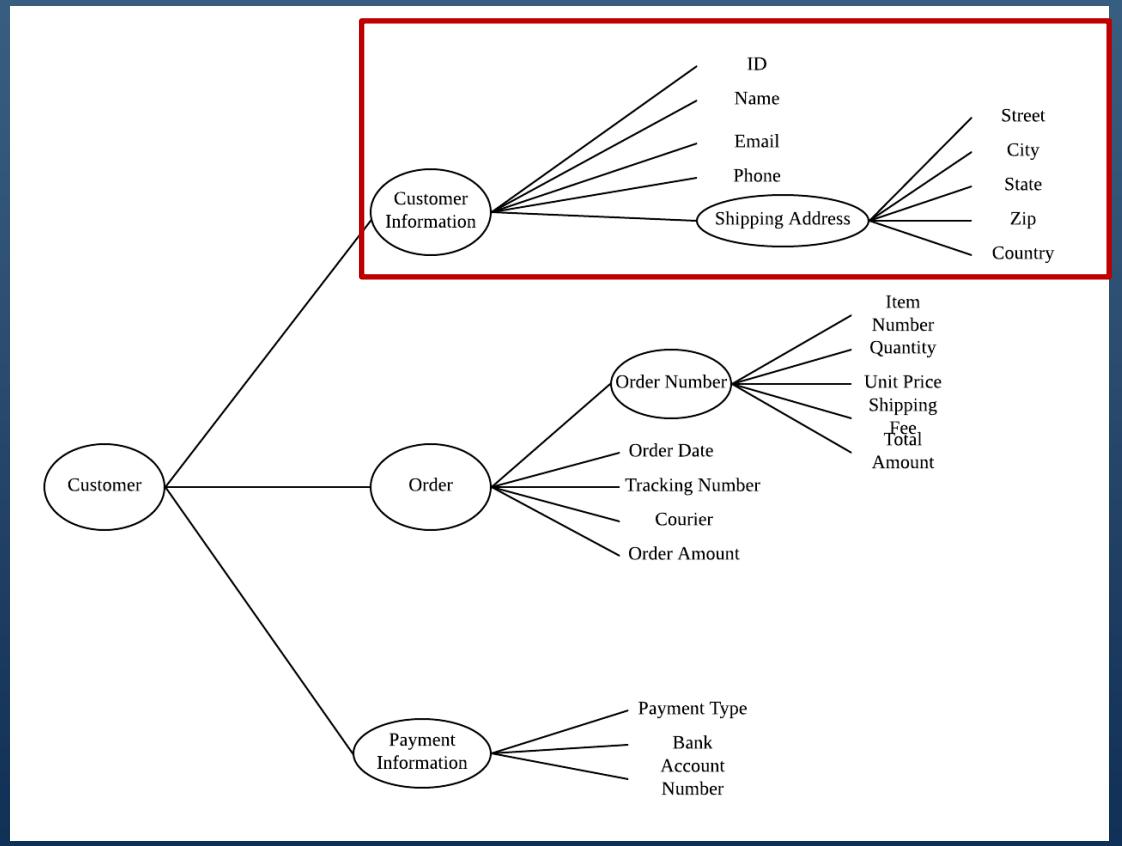
Case Study: Customer 360 View

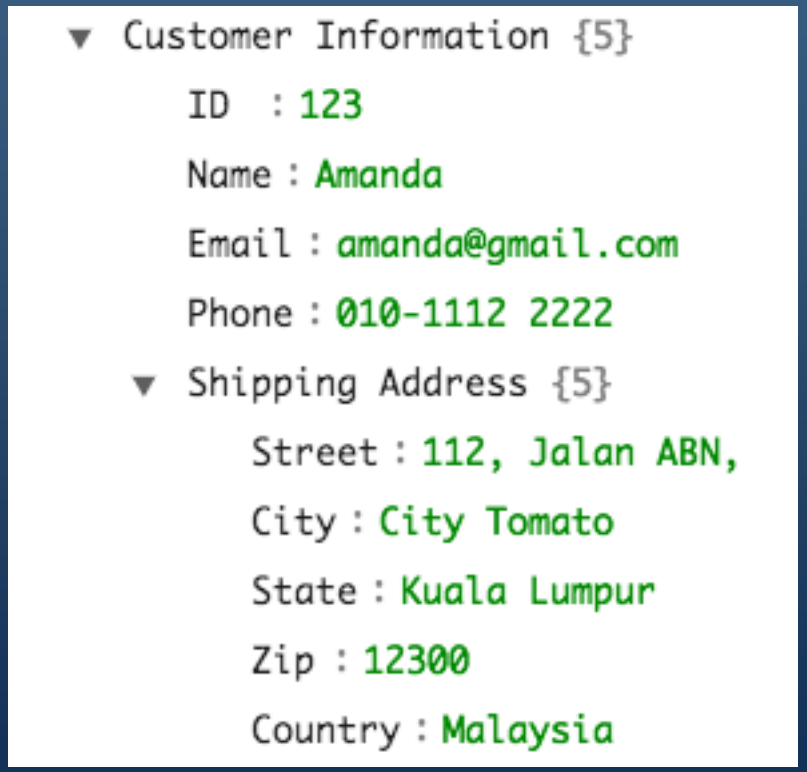



```
▼ Customer {3}
  ▼ Customer Information {5}
      ID : 123
      Name : Amanda
      Email : amanda@gmail.com
      Phone : 010-1112 2222
    ▶ Shipping Address {5}
  ▼ Order {5}
    ▶ Order Number {5}
      Order Date : 2018-01-23 21:32:11
      Tracking Number : 4003454341
      Courier : BlackCat
      Order Amount : 130.16
  ▶ Payments {3}
```

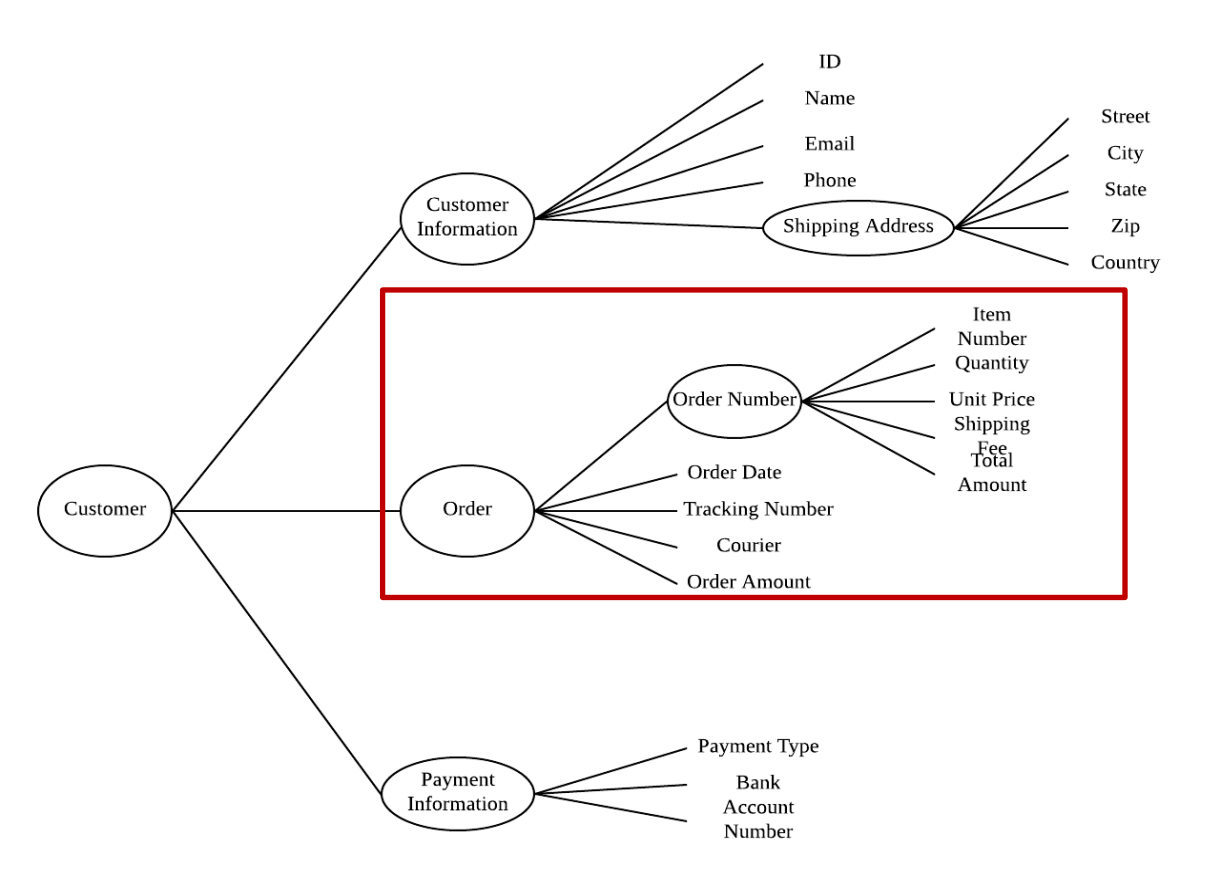

*Created with JSON Formatter

Case Study: Customer 360 View



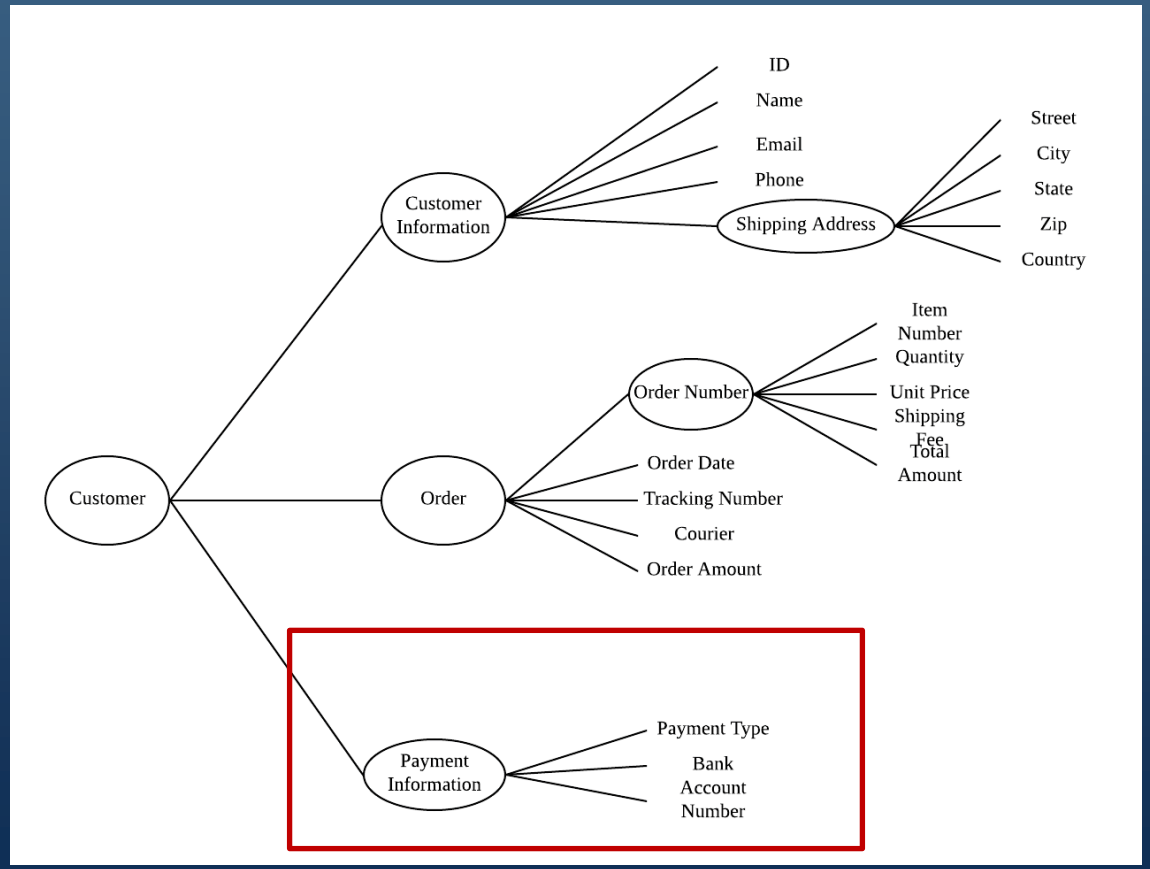


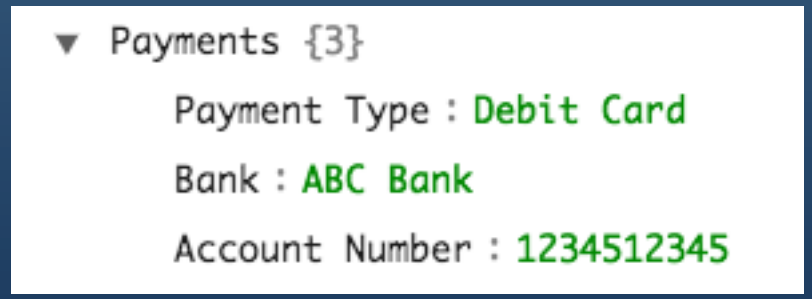
Case Study: Customer 360 View



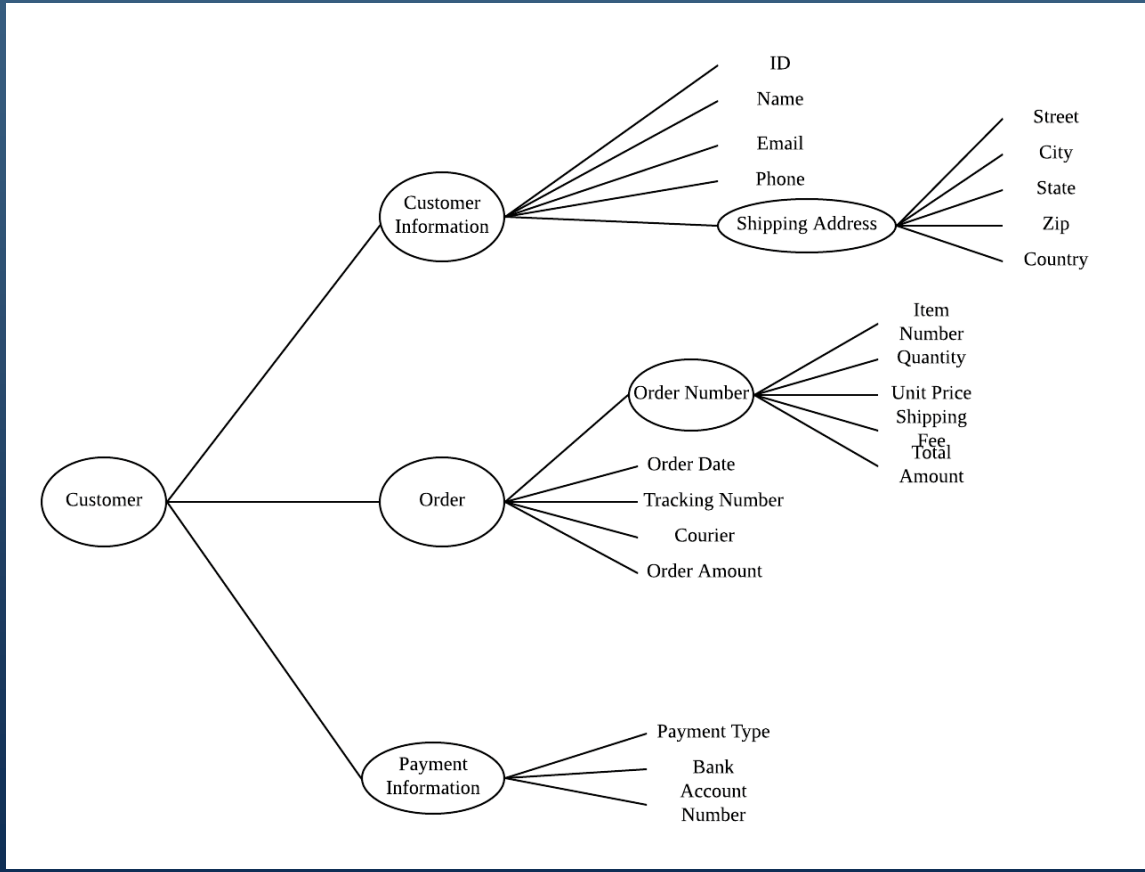

```
▼ Order {5}
  ▼ Order Number {5}
      Item Number : X1234567
      Quantity : 1
      Unit Price : 12.00
      Shipping Fee : 8.00
      Total Amount : 130.16
    Order Date : 2018-01-23 21:32:11
    Tracking Number : 4003454341
    Courier : BlackCat
    Order Amount : 130.16
```

Case Study: Customer 360 View





Case Study: Customer 360 View




```
{
  "Customer": {
    "Customer Information": {
      "ID": "123",
      "Name": "Amanda",
      "Email": "amanda@gmail.com",
      "Phone": "010-1112 2222",
      "Shipping Address": {
        "Street": "112, Jalan ABN,",
        "City": "City Tomato",
        "State": "Kuala Lumpur",
        "Zip": "12300",
        "Country": "Malaysia"
      }
    },
    "Order": {
      "Order Number": {
        "Item Number": "X1234567",
        "Quantity": "1",
        "Unit Price": "12.00",
        "Shipping Fee": "8.00",
        "Total Amount": "130.16"
      },
      "Order Date": "2018-01-23 21:32:11",
      "Tracking Number": "4003454341",
      "Courier": "BlackCat",
      "Order Amount": "130.16"
    },
    "Payments": {
      "Payment Type": "Debit Card",
      "Bank": "ABC Bank",
      "Account Number": "1234512345"
    }
  }
}
```

Introduction to NoSQL

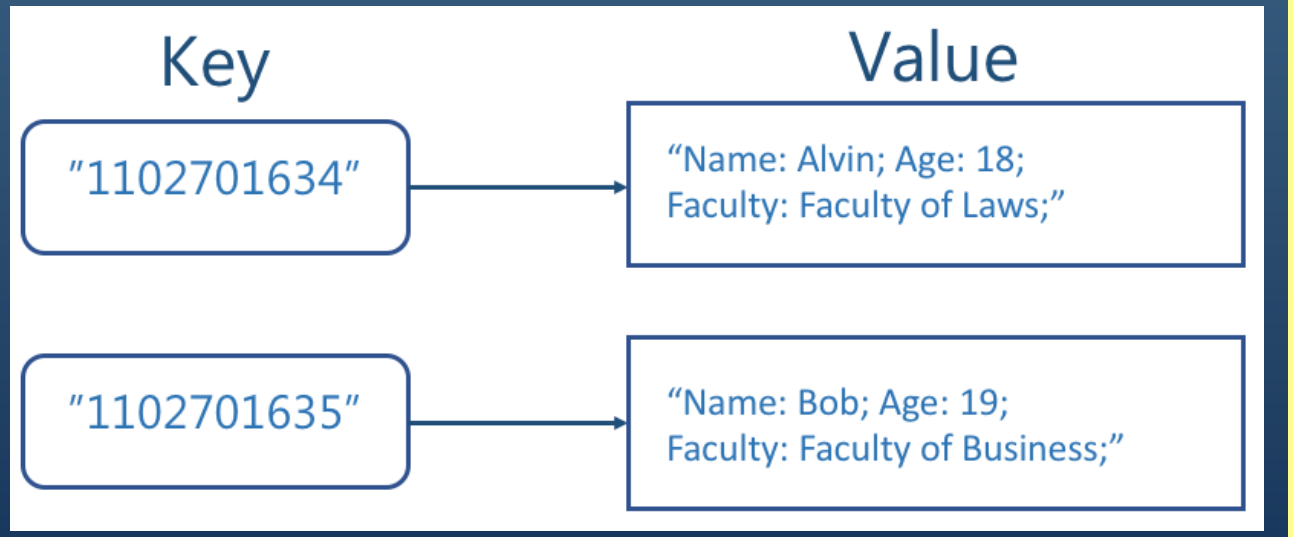
Types of NoSQL Databases

CADS
FUTURE PROOFING THE WORLD!

Key-Value Store

Introduction to NoSQL

Types of NoSQL Databases: Key-Value Store



- Data is stored as a collection of key-value pairs.
- Maps are the simplest module in key-value store. A unique key in map has a single value associated with it.
- Sometimes, the value could be a list of another map. So, user is able to create tree-structures within key-value store.
- Key-value stores are optimized for speed of ingestion and retrieval.

Introduction to NoSQL

Types of NoSQL Databases: Key-Value Store

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Manage transient information
- Handle high-speed retrieval

1

Manage User Information

- Mission-critical data.
- The ability to scale up for user supporting data.

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Manage transient information
- Handle high-speed retrieval

2

Deliver Web Advertisements

- Speed is crucial so that the advertisements do not slow down the user's experience.
- The **key** can be served as the combination of factors that determine what a user interested, whereas the **value** stores every thing that is required to serve the advertisement.

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Manage transient information
- Handle high-speed retrieval

3

Handle User Sessions

- Websites may need to store various types of short-lived session data.
- Key-value stores are ideal to store and retrieve session data at high speeds.

CADS
FUTURE PROOFING THE WORLD!

Industrial Example

INDUSTRY:
Social Networking Services

USE CASE:
Near Real-Time Personalization

Facebook uses Memcache as the distributed key-value store to improve users experience by caching the data requested by clients for further requests.

CADS
FUTURE PROOFING THE WORLD!

Document-Based

Introduction to NoSQL

Types of NoSQL Databases: Document-Based

- A.k.a aggregate database because it stores documents in single logical unit (an aggregate).
- Generally, a document is any unstructured or semi-structured piece of data.
- The value of a document can be retrieved by a key.

Source: https://www.w3resource.com/mongodb/nosql.php

Introduction to NoSQL

Types of NoSQL Databases: Document-Based

CADS
FUTURE PROOFING THE WORLD!

Couchbase

MarkLogic™

mongoDB®

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Modify data structures
- Control unstructured data feeds

1

Manage Content Lifecycle

- It is useful to store the document lifecycle status.
- These information are usually kept as XML schema.

**Content Processing Framework (CPF) of MarkLogic Server*

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Modify data structures
- Control unstructured data feeds

2

Manage Changing Data Structures

- Organizations can collect data from various departments and public data.
- However, these ever-changing data may appear in different formats.
- Document-based database is suitable to handle these varieties.

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Modify data structures
- Control unstructured data feeds

3

Consolidate Related Data

- Document-based datasets is suitable to store data related to the same topic.
- Some document-based databases can add documents to multiple collections.

CADS
FUTURE PROOFING THE WORLD!

Industrial Example

INDUSTRY:
Music Applications

USE CASE:
Near Real-Time Personalization

TuneWiki enhance the user music experience by allowing them to share their favourite photos, music, and lyrics with friends.

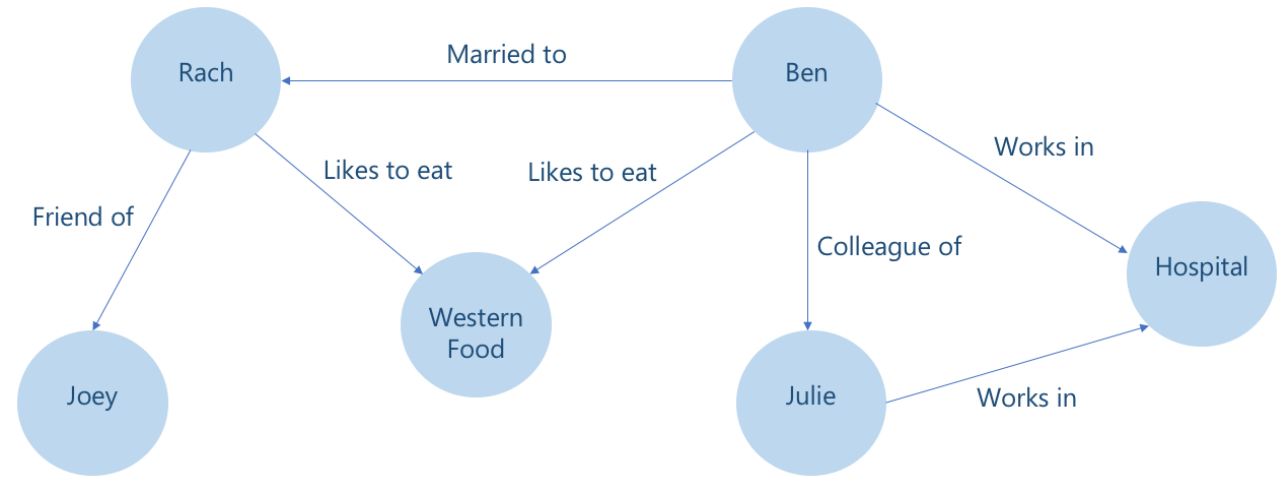
TuneWiki adopts Couchbase as a replacement for Memcached to cache and store everything the application needs such as the photos, lyrics, timelines, and API keys.

CADS
FUTURE PROOFING THE WORLD!

Graph-Based

Introduction to NoSQL

Types of NoSQL Databases: Graph-Based

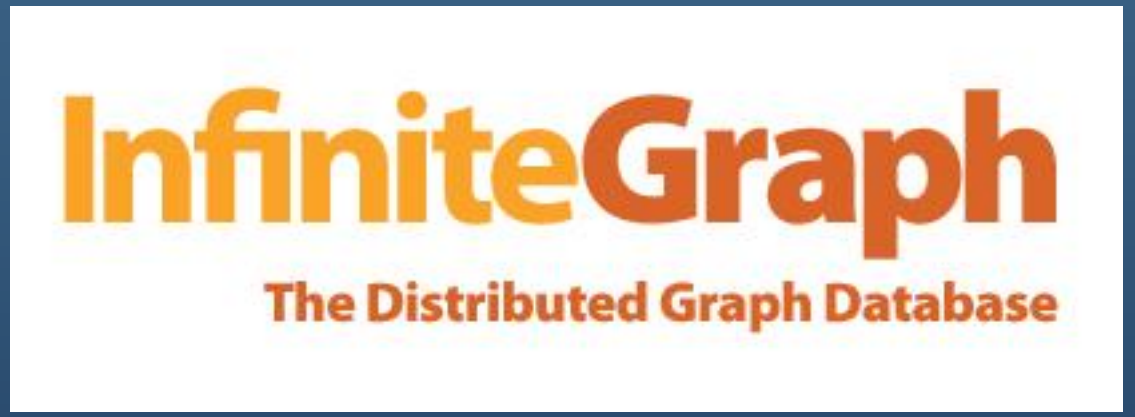


- Designed for data which consists of interconnected elements and well defined relations.
- Every assertion is represented as vertices and edge.
  - **Vertex** is the thing described, could be physical object (eg: a person) or a concept (eg: meeting).
  - **Edge** is the relationship in the graph.

Introduction to NoSQL

Types of NoSQL Databases: Graph-Based

CADS
FUTURE PROOFING THE WORLD!



CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Handle unstructured information
- Establish social relationships

1

Store Semantic Facts from Text

- The relationships between words, phrases, and sentences could be stored with graph-based database.
- Inferencing could be performed with graph-based database.

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Handle unstructured information
- Establish social relationships

2

Track Provenance

- As an example, the possible provenance tracking could be to find collusion in financial markets to trace the pattern of fraud.

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Handle unstructured information
- Establish social relationships

3

Manage Social Graph

- Establishment of social networks as well as professional organizations.

Industrial Example

TalentNet is a social recommendations application to help users to establish professional field networks.

INDUSTRY:
Social Recommendations Application

TalentNet uses graph database to store the data related to users' companies, projects and interests/skills.

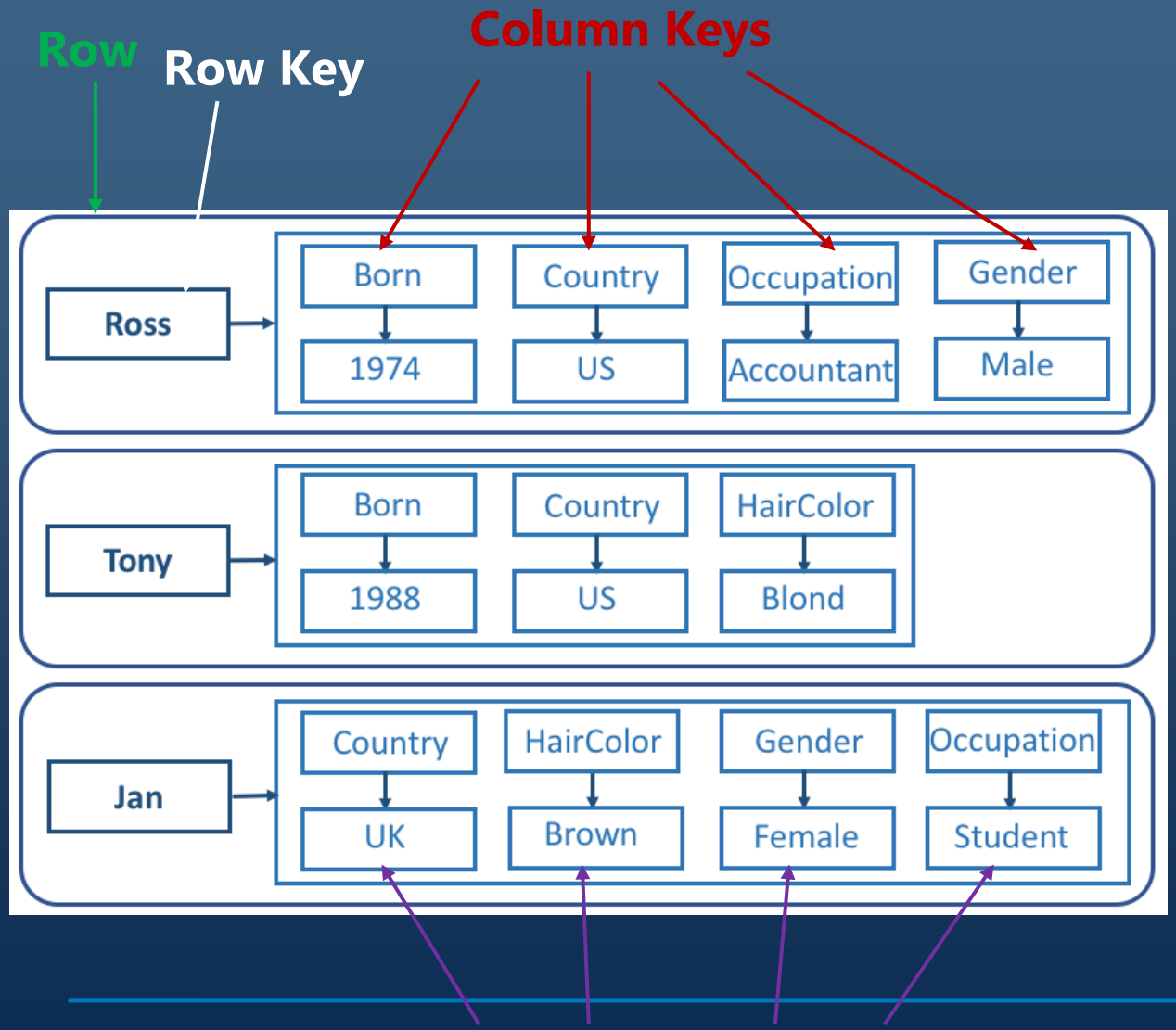
USE CASE:
Establish Connections between Users

CADS
FUTURE PROOFING THE WORLD!

Wide Column Store

Introduction to NoSQL

Types of NoSQL Databases: Wide Column Store



- Can be seen as a two-dimensional key-value store.
- The column names and row keys are not fixed.
- The names and format of columns can vary from different rows in the same table, and the columns can be added to any row at any time without having to add it to other rows.

Introduction to NoSQL

Types of NoSQL Databases: Wide Column Store

Google BigTable

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Restructure data to be stored in columns
- Store data for later analysis

1

Handle Sparse Data

- Wide Column Store is suitable to store sparse data.
- For example, the data related to a social media site where there are users do not provide their profile photos.

CADS
FUTURE PROOFING THE WORLD!

USE CASE

- Restructure data to be stored in columns
- Store data for later analysis

2

Store Log Files

- Wide Column Store can handle flexible columns.
- The same data can be stored in alternative structures at the same time. It is relatively useful to build a quick picture of a day in 5 minutes, 1-hour, or 1-day.

CADS
FUTURE PROOFING THE WORLD!

Industrial Example

INDUSTRY:
Social Networking Application

USE CASE:
More Affordable Scalable Storage

Instagram uses Cassandra to store auditing information related to security and site integrity purposes.

Introduction to NoSQL

Types of NoSQL Databases: Main Takeaway

CADS
FUTURE PROOFING THE WORLD!

- **Key-value store** is the simplest to implement. It is suitable for those data that is fairly **static** and require **high-speed retrieval**.
- **Document-based** is generally having all the **benefits from key-value stores**, and providing the **ease of data processing**.
- **Graph-based** can be used to store data that involves **complex mappings**.
- **Wide column store** can be used for **data reporting usage**.

Reference

1. https://searchdatamanagement.techtarget.com/definition/relational-database
2. https://stackoverflow.com/questions/12346326/cap-theorem-availability-and-partition-tolerance
3. https://www.dezyre.com/article/nosql-vs-sql-4-reasons-why-nosql-is-better-for-big-data-applications/86
4. https://studio3t.com/whats-new/nosql-database-types/
5. https://www.quora.com/What-are-the-main-differences-between-the-four-types-of-NoSql-databases-KeyValue-Store-Column-Oriented-Store-Document-Oriented-Graph-Database
6. https://mindmapsunleashed.com/the-mind-mapping-concept-and-how-you-benefit-from-this
7. https://www.zdnet.com/article/look-at-what-google-and-amazon-are-doing-with-databases-thats-your-future/
8. https://stackoverflow.com/questions/362956/what-database-does-google-use
9. https://dzone.com/articles/couchbase-nosql-tunewiki
10. http://bitnine.net/blog-graph-database/graph-database-real-world-examples/
11. https://medium.com/@shagun/scaling-memcache-at-facebook-1ba77d71c082
12. Scaling Memcache at Facebook
13. https://www.datastax.com/dev/blog/facebooks-instagram-making-the-switch-to-cassandra-from-redis-a-75-insta-savings

CADS
FUTURE PROOFING THE WORLD!

The Center of Applied Data Science

E: info@thecads.org
W : www.thecads.org