Data Storage Models

Independent Seminar

Guide:

Dr. Shelly Sachdeva

Associate Professor

Presented By :

Prabhat Pushp

212211012

March 2023

NIT Delhi

Table of Contents

1. INTRODUCTION
2. FLAT FILE-BASED DATABASE
3. RELATIONAL BASED DATABASE
4. KEY-VALUE-BASED DATABASE
5. DOCUMENT-BASED DATABASE
6. GRAPH-BASED DATABASE
7. COLUMN-BASEDATABASE
8. NATIVE XML-BASED DATABASE
9. HIERARCHICAL DATABASES (4th sem)
10. NETWORK DATABASES (4th sem)
11. TIME SERIES DATABASES (4th sem)
12. NEWSQL DATABASES (4th sem)
13. MULTI-MODEL DATABASES (4th sem)
14. COMPARATIVE ANALYSIS
15. REFERENCES

Introduction

A data model is a representation that we use to perceive and manipulate our data.

It allows us to :

- represent the data elements under analysis
- how these are related to each others

The Various data storage models available are:

- Flat File based database
- Relational based database
- Key-Value based database
- Document based database
- Graph based database
- Column based database
- Native XML based database
- Hierarchical database
- Network database
- Time series database
- NewSQL database
- Multi-model database

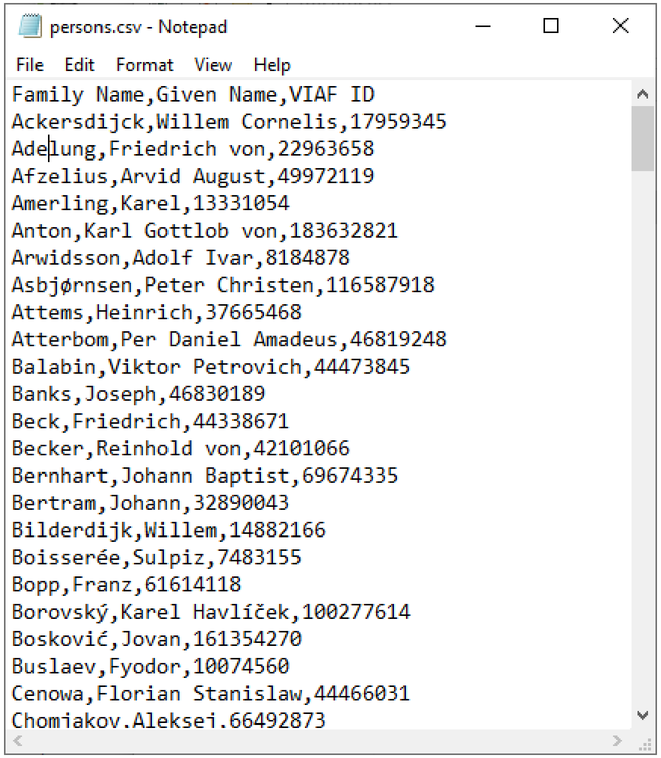
Flat File Based Databases

In this type of database data stored in file. A flat file can be a **plain text file** or a **binary file**. The term has generally implied a small, simple database.

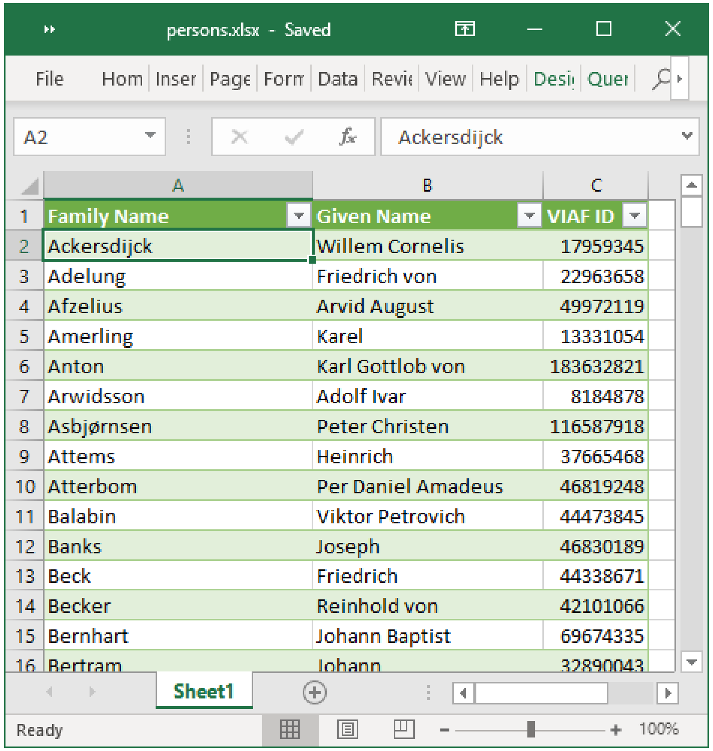
- Advantage of Flat file database
  - Ideal for small amount of database
  - Stored many data in file
  - Less Software and Hardware Required
  - Less Skills are required to handle flat database systems.

- Disadvantage of Flat file database
  - Files are not linked
  - Security
  - Not stored images
  - Slow for huge database

Flat File Based Databases



```
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov.Aleksei.66492873
```



	A	B	C
1	Family Name	Given Name	VIAF ID
2	Ackersdijck	Willem Cornelis	17959345
3	Adelung	Friedrich von	22963658
4	Afzelius	Arvid August	49972119
5	Amerling	Karel	13331054
6	Anton	Karl Gottlob von	183632821
7	Arwidsson	Adolf Ivar	8184878
8	Asbjørnsen	Peter Christen	116587918
9	Attems	Heinrich	37665468
10	Atterbom	Per Daniel Amadeus	46819248
11	Balabin	Viktor Petrovich	44473845
12	Banks	Joseph	46830189
13	Beck	Friedrich	44338671
14	Becker	Reinhold von	42101066
15	Bernhart	Johann Baptist	69674335
16	Bertram	Johann	32890043

Fig 1 : Flat File based databases (CSV and XML file) [1]

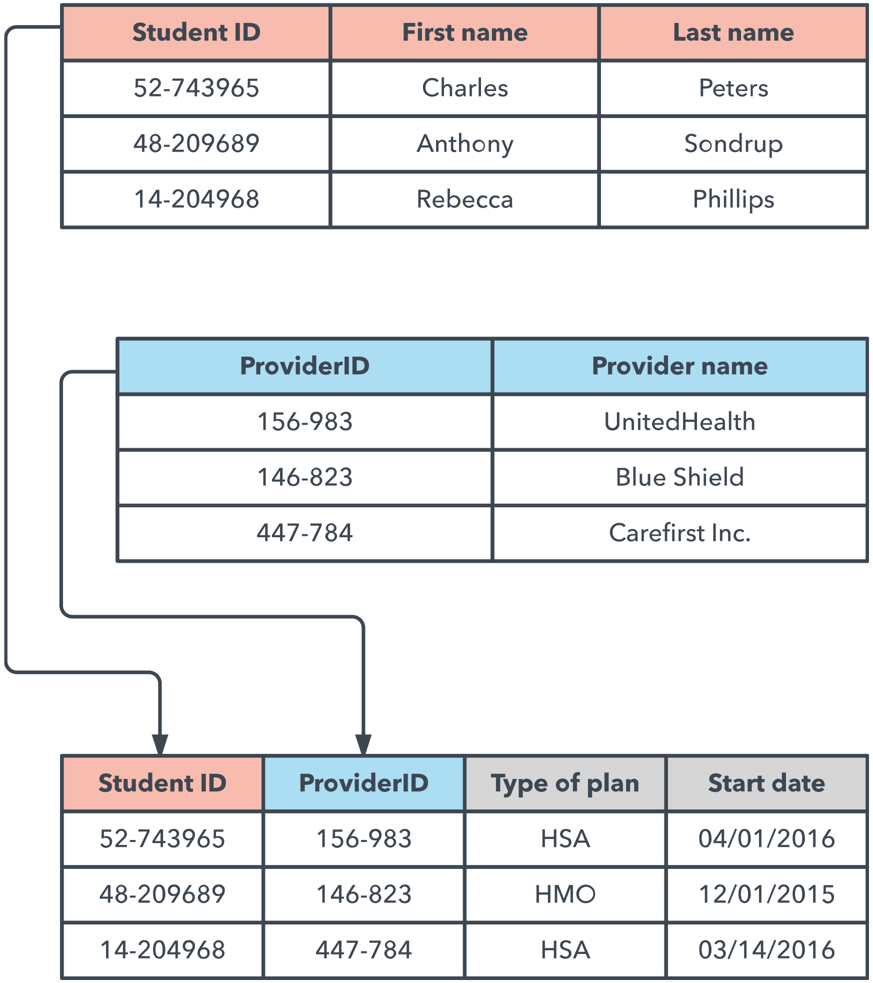
Relational Based Databases

A relational database organizes data into **tables** which can be linked or related based on data common to each. Tables contains **tuples(rows)** and **attributes(columns).** You can communicate with relational databases using **Structured Query Language (SQL)**

- **Advantages of Relational based databases:**
  - Strict ACID rules
  - Strict Schema

- **Disadvantages of Relational based databases:**
  - Strict ACID rules
  - Strict Schema
  - Vertical Scalability

Relational Based Databases



Student ID	First name	Last name
52-743965	Charles	Peters
48-209689	Anthony	Sondrup
14-204968	Rebecca	Phillips

ProviderID	Provider name
156-983	UnitedHealth
146-823	Blue Shield
447-784	Carefirst Inc.

Student ID	ProviderID	Type of plan	Start date
52-743965	156-983	HSA	04/01/2016
48-209689	146-823	HMO	12/01/2015
14-204968	447-784	HSA	03/14/2016

Fig 2 : Relational Databases [2]

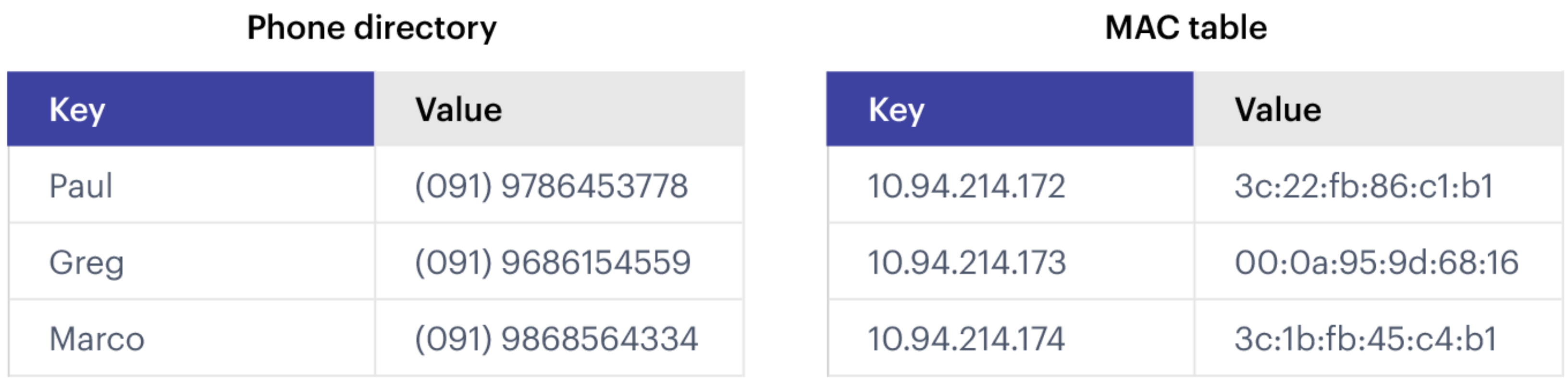
Key-Value Based Databases

Key-value databases are the simplest type of NoSQL database. It has simplicity, they are also the most scalable, allowing horizontal scaling of large amounts of data.

- **Advantages of Key-Value based databases:**
  - Very Fast
  - Good for session stores and shopping carts

- **Disadvantages of Key-Value based databases:**
  - Generally in-memory databases
  - Not suitable for storing complex data
  - Bad for large data

Key-Value Based Databases



Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334

MAC table

Key	Value
10.94.214.172	3c:22:fb:86:c1:b1
10.94.214.173	00:0a:95:9d:68:16
10.94.214.174	3c:1b:fb:45:c4:b1

Fig 3 : Key-Value based database (Redis) [3]

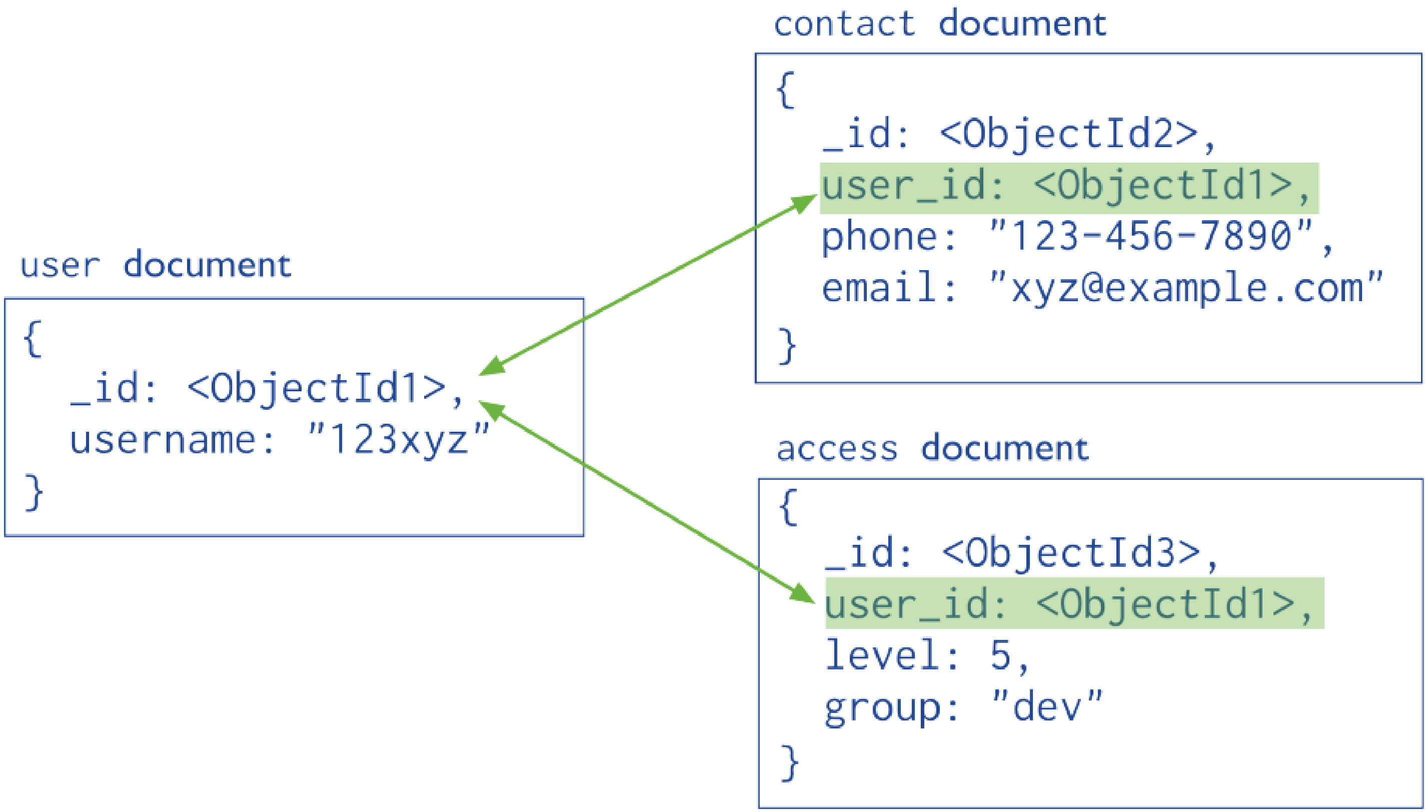
Document Based Databases

A document-oriented database is a computer program designed for storing, retrieving, and managing **document oriented**, or **semi-structured data** information.

- **Advantages of Document based databases:**
  - Schema-less
  - Faster creation and care
  - No foreign keys
  - Open formats
  - Built-in versioning

- **Disadvantages of Document based databases:**
  - Consistency-Check Limitations
  - Atomicity weaknesses
  - Security

Document Based Databases



contact document

```
{
   _id: <ObjectId2>,
   user_id: <ObjectId1>,
   phone: "123-456-7890",
   email: "xyz@example.com"
}
```

user document

```
{
   _id: <ObjectId1>,
   username: "123xyz"
}
```

access document

```
{
   _id: <ObjectId3>,
   user_id: <ObjectId1>,
   level: 5,
   group: "dev"
}
```

Fig 4 : Document based database (MongoDB) [4]

Graph Based Databases

Graph database is a kind of No SQL database that uses graph structures with nodes, edges, and properties to represent and store information.

- **Advantages of Document based databases:**
  - Agile and Flexible Structure
  - Explicit Entity Relationship
  - Realtime results
  - Useful for systems with highly connected relationships

- **Disadvantages of Document based databases:**
  - No Standard Query Language
  - Not for Transactional Based Systems
  - User base is small. Debugging and support is hard to find.

Graph Based Databases

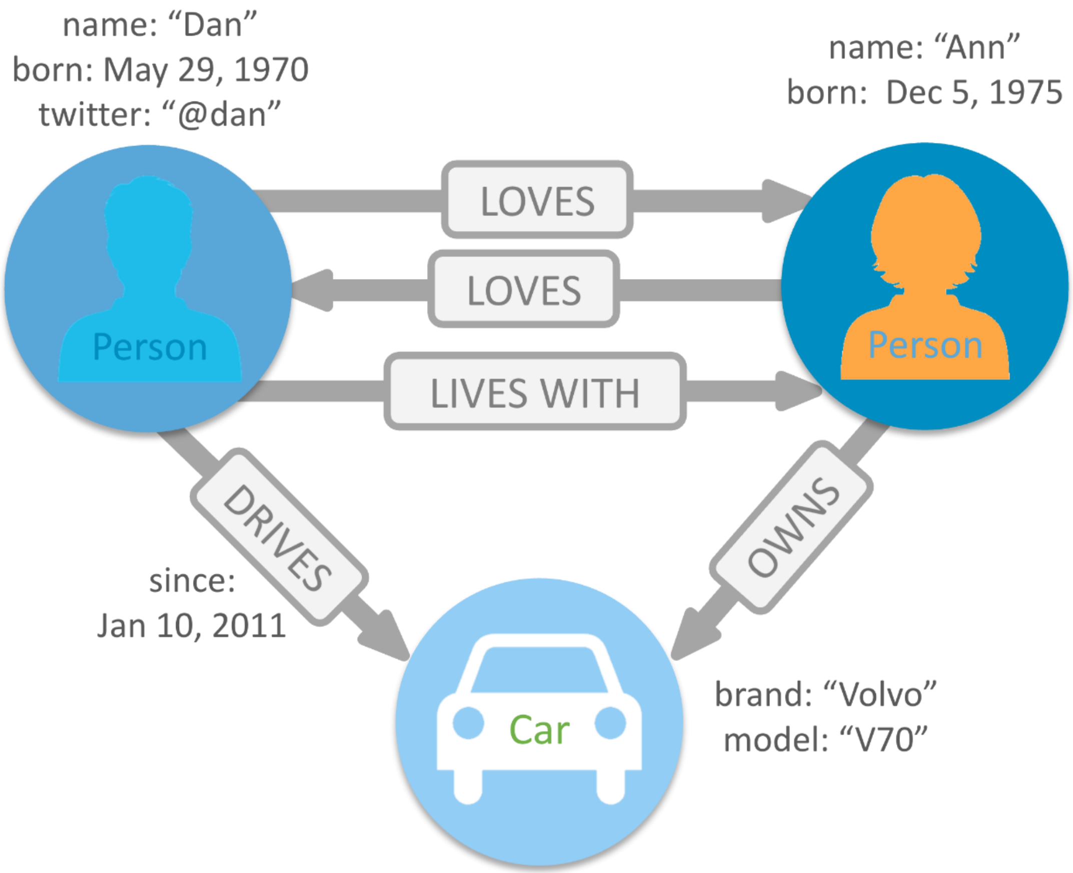


Fig 5 : Graph based database (Neo4j) [5]

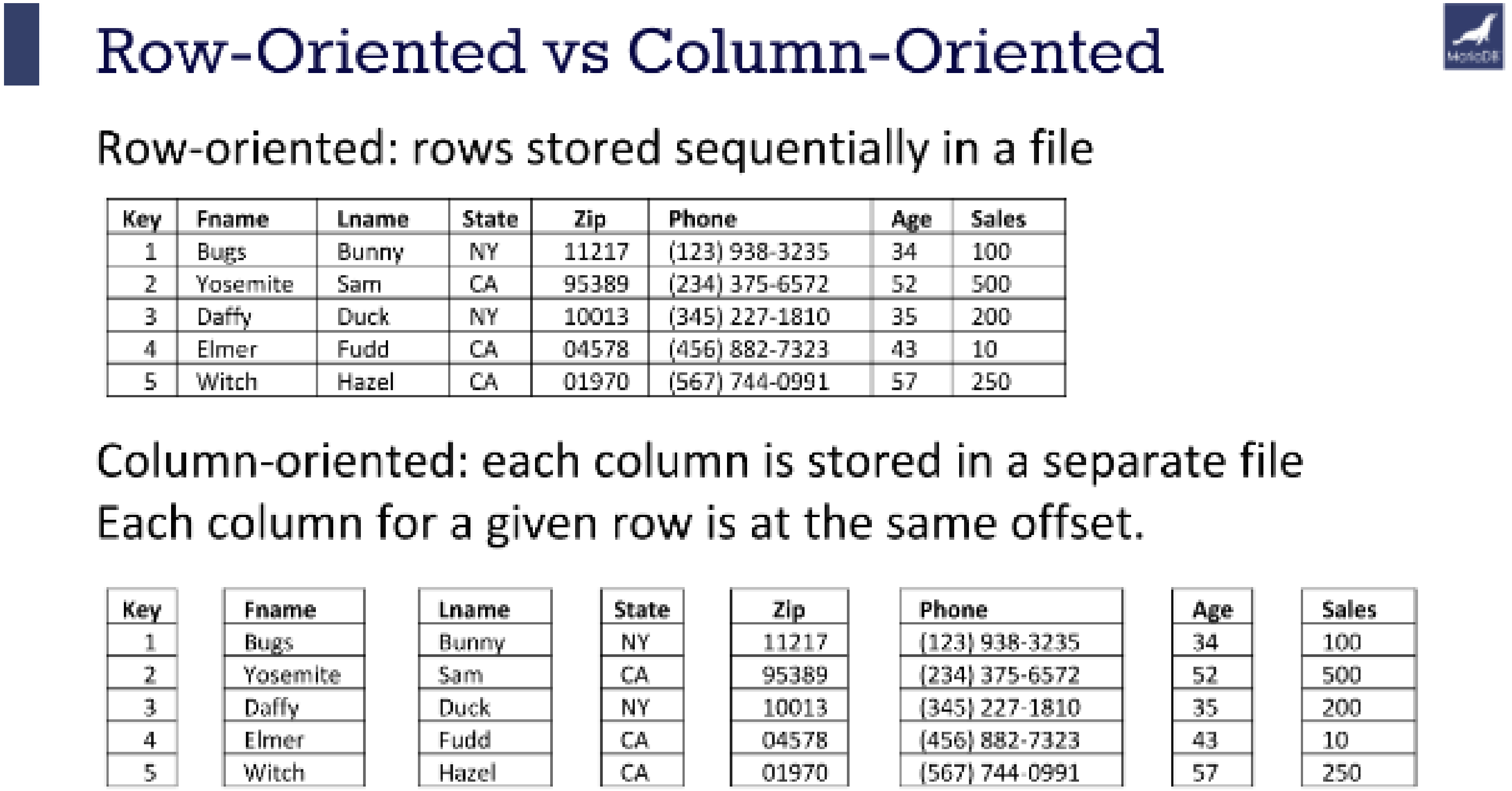
Column Based Databases

Column oriented databases are databases that organize data by field, keeping all of the data associated with a field next to each other in memory. Columnar databases are great for querying data.

- **Advantages of Column based databases:**
  - Self-Indexing
  - Speed and efficiency
  - Compressible data
  - Bulk data

- **Disadvantages of Column based databases:**
  - Bad for transactional data
  - Security

Column Based Databases



Row-Oriented vs Column-Oriented

Row-oriented: rows stored sequentially in a file

Key	Fname	Lname	State	Zip	Phone	Age	Sales
1	Bugs	Bunny	NY	11217	(123) 938-3235	34	100
2	Yosemite	Sam	CA	95389	(234) 375-6572	52	500
3	Daffy	Duck	NY	10013	(345) 227-1810	35	200
4	Elmer	Fudd	CA	04578	(456) 882-7323	43	10
5	Witch	Hazel	CA	01970	(567) 744-0991	57	250

Column-oriented: each column is stored in a separate file

Each column for a given row is at the same offset.

Key
1
2
3
4
5

Fname
Bugs
Yosemite
Daffy
Elmer
Witch

Lname
Bunny
Sam
Duck
Fudd
Hazel

State
NY
CA
NY
CA
CA

Zip
11217
95389
10013
04578
01970

Phone
(123) 938-3235
(234) 375-6572
(345) 227-1810
(456) 882-7323
(567) 744-0991

Age
34
52
35
43
57

Sales
100
500
200
10
250

Fig 6 : Column based database (MariaDB) [6]

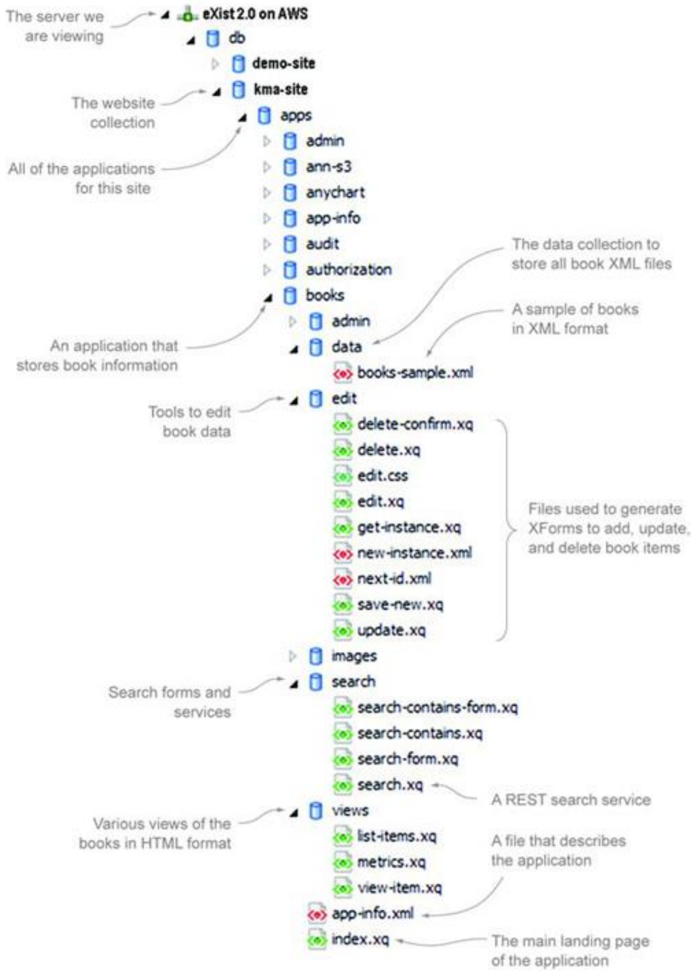
Native XML Based Databases

Native XML database is based on the container rather than table format. It can store large amount of XML document and data. Native XML database is queried by the XPath-expressions.

- **Advantages of Native XML based databases:**
  - Efficient in storing and managing XML documents than XML-enabled databases
  - Can store more than 2 dimensional data

- **Disadvantages of Native XML based databases:**
  - Relatively new
  - Not easy to learn
  - Small community so support is hard to find.

Native XML Based Databases



```
eXist 2.0 on AWS
  db
    demo-site
    kma-site
      apps
        admin
        ann-s3
        anychart
        app-info
        audit
        authorization
        books
          admin
          data
            books-sample.xml
          edit
            delete-confirm.xq
            delete.xq
            edit.css
            edit.xq
            get-instance.xq
            new-instance.xml
            next-id.xml
            save-new.xq
            update.xq
          images
          search
            search-contains-form.xq
            search-contains.xq
            search-form.xq
            search.xq
          views
            list-items.xq
            metrics.xq
            view-item.xq
          app-info.xml
          index.xq
```

The server we are viewing

The website collection

All of the applications for this site

The data collection to store all book XML files

A sample of books in XML format

An application that stores book information

Tools to edit book data

Files used to generate XForms to add, update, and delete book items

Search forms and services

A REST search service

Various views of the books in HTML format

A file that describes the application

The main landing page of the application

Fig 7 : Native XML based database [7]

Hierarchical Databases

Hierarchical databases revolutionized the field of database management by introducing a new way of organizing data. Hierarchical databases provide a hierarchy of records where each record has a single parent, resulting in a tree-like structure that may be used to classify entries based on their parent record.

- **Advantages of Hierarchical Databases:**
  - Hierarchical structure, like a family tree
  - Simple and easy to use
  - Good for applications that require fast data access and retrieval
  - Efficient for batch processing

- **Disadvantages of Hierarchical Databases:**
  - Limited flexibility
  - Difficult to handle complex relationships between data elements
  - Can lead to data redundancy and inconsistency
  - Not suitable for handling unstructured data
- **Examples:**
  - Filesystems, DNS, LDAP directories

Hierarchical Databases

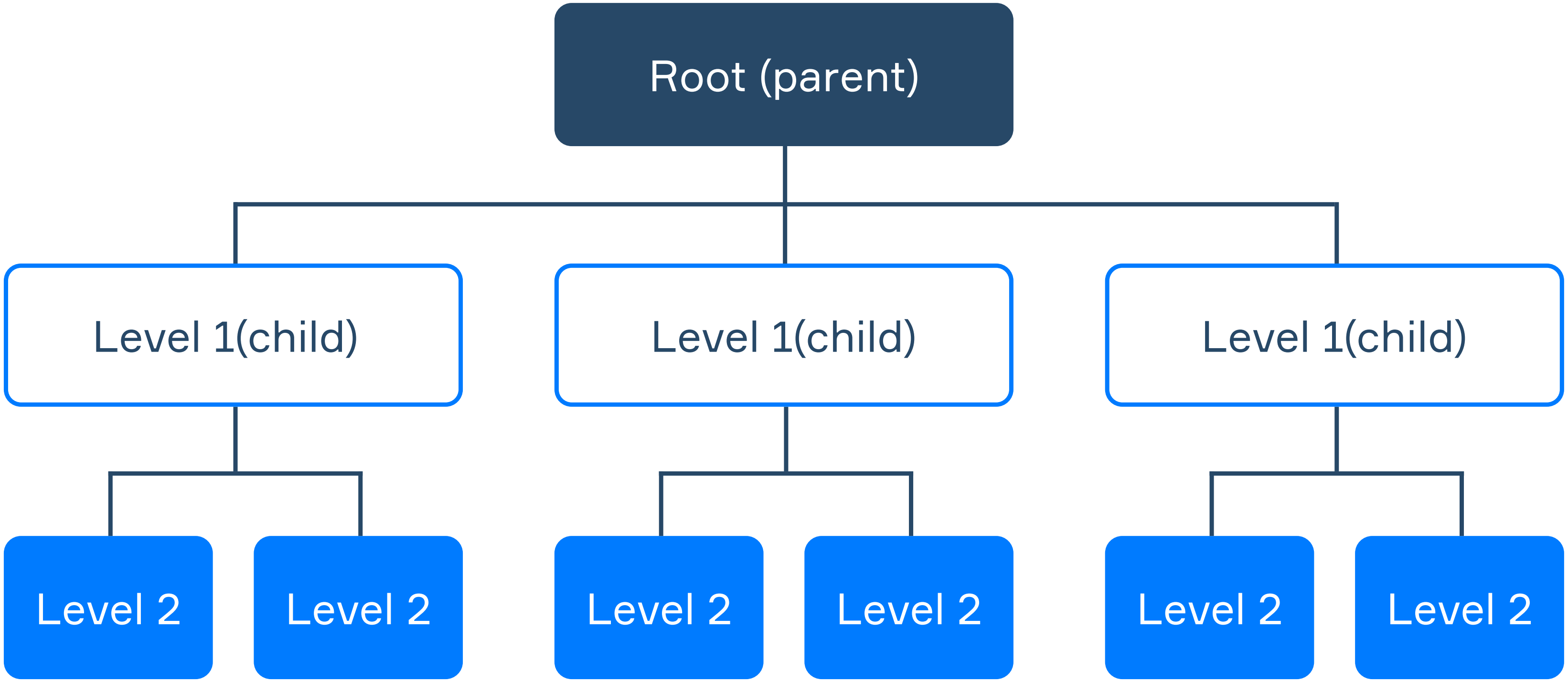


Fig 8 : Hierarchical Databases [8]

Network databases

Network databases are a type of database management system (DBMS) that allows data to be stored in a network-like structure. This structure allows for complex relationships between data elements, making it a popular choice in applications such as engineering, scientific research, and telecommunications.

- **Advantages of Network databases:**
  - Offers greater flexibility than a hierarchical database
  - Allows for complex relationships between data elements
  - Useful for applications that require a high level of data integrity, such as banking, finance, and scientific research.
- **Disadvantages of Network databases:**
  - Difficult to use for simple data structures
  - Requires extensive programming knowledge to manage
  - Can be slow to retrieve data due to complex relationships
- **Examples:**
  - The Integrated Database Management System (IDMS)

Network databases

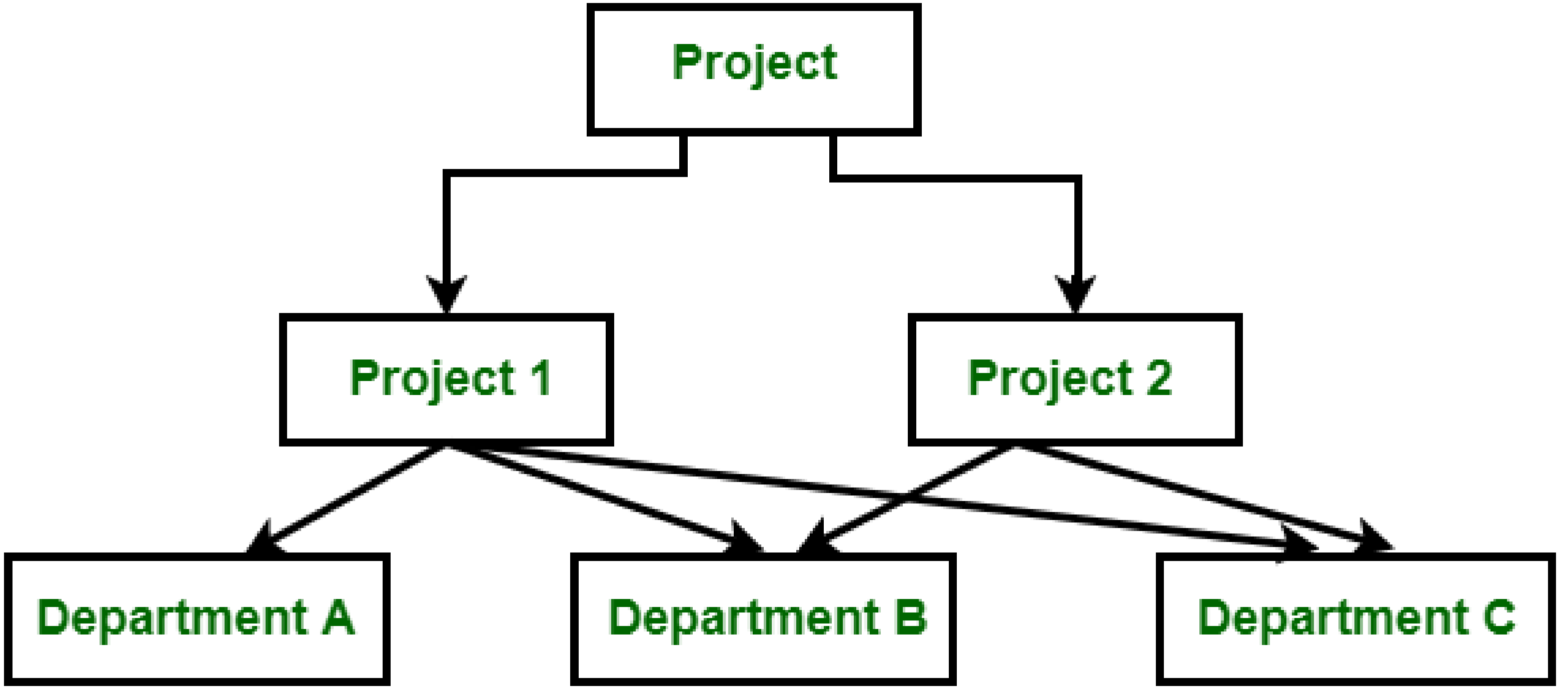


Fig 9 : Network databases [9]

Time series databases

Time series databases are a type of database management system (DBMS) designed to handle time-stamped data. They are used in applications that require storing and analyzing large amounts of data over time, such as financial trading, IoT (Internet of Things), and scientific research.

- **Advantages of Time series databases:**
  - Designed specifically for handling time-stamped data, making it efficient for time-based analysis.
  - High performance and scalability due to the ability to store and query large amounts of data quickly.
  - Provides real-time analytics capabilities, allowing for rapid decision-making.
  - Easy to integrate with other systems, making it an ideal solution for IoT and big data applications.
- **Disadvantages of Time series databases:**
  - Limited flexibility for non-time-based data analysis.
  - Complex data models can be difficult to implement in a time series database.
  - Can be expensive to implement and maintain.
- **Examples:**
  - OpenTSDB, Prometheus, InfluxDB, TimescaleDB, KairosDB

Time series databases

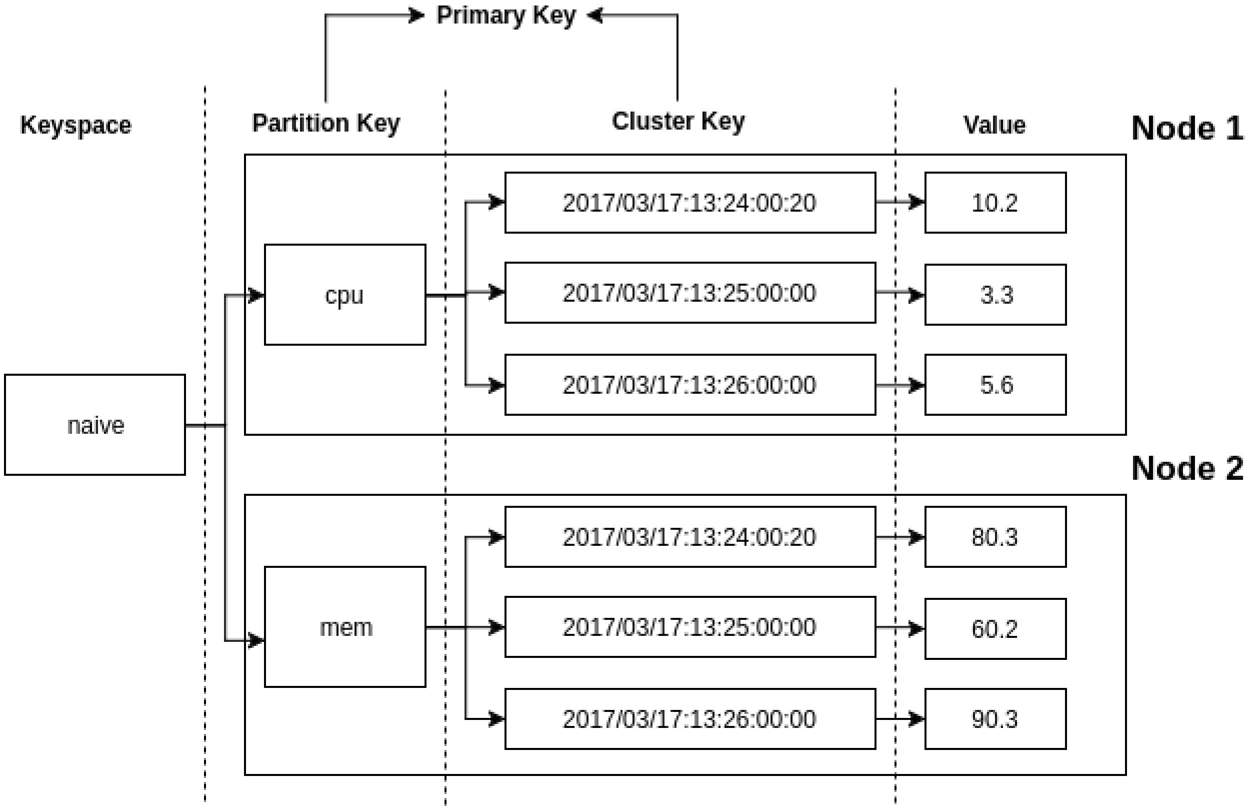


Fig 10 : Time series databases [10]

NewSQL databases

NewSQL databases are a type of database management system (DBMS) that combines the scalability of NoSQL databases with the traditional ACID (Atomicity, Consistency, Isolation, Durability) properties of relational databases. They are designed to handle large amounts of data in distributed environments and are popular in modern, high-performance applications.

- **Advantages of NewSQL databases:**
  - High performance and scalability due to the distributed architecture.
  - ACID compliance, making them suitable for transactional applications.
  - Familiar SQL interface, making them easy to integrate with existing systems and applications.
  - In-memory processing and caching for faster data retrieval.
  - Ability to handle both structured and unstructured data.
- **Disadvantages of NewSQL databases:**
  - Can be complex to set up and maintain.
  - Limited support for advanced analytics and reporting.
- **Examples:**
  - MemSQL, VoltDB, Spanner, Calvin, CockroachDB, FaunaDB, yugabyteDB, PlanetScale

NewSQL databases

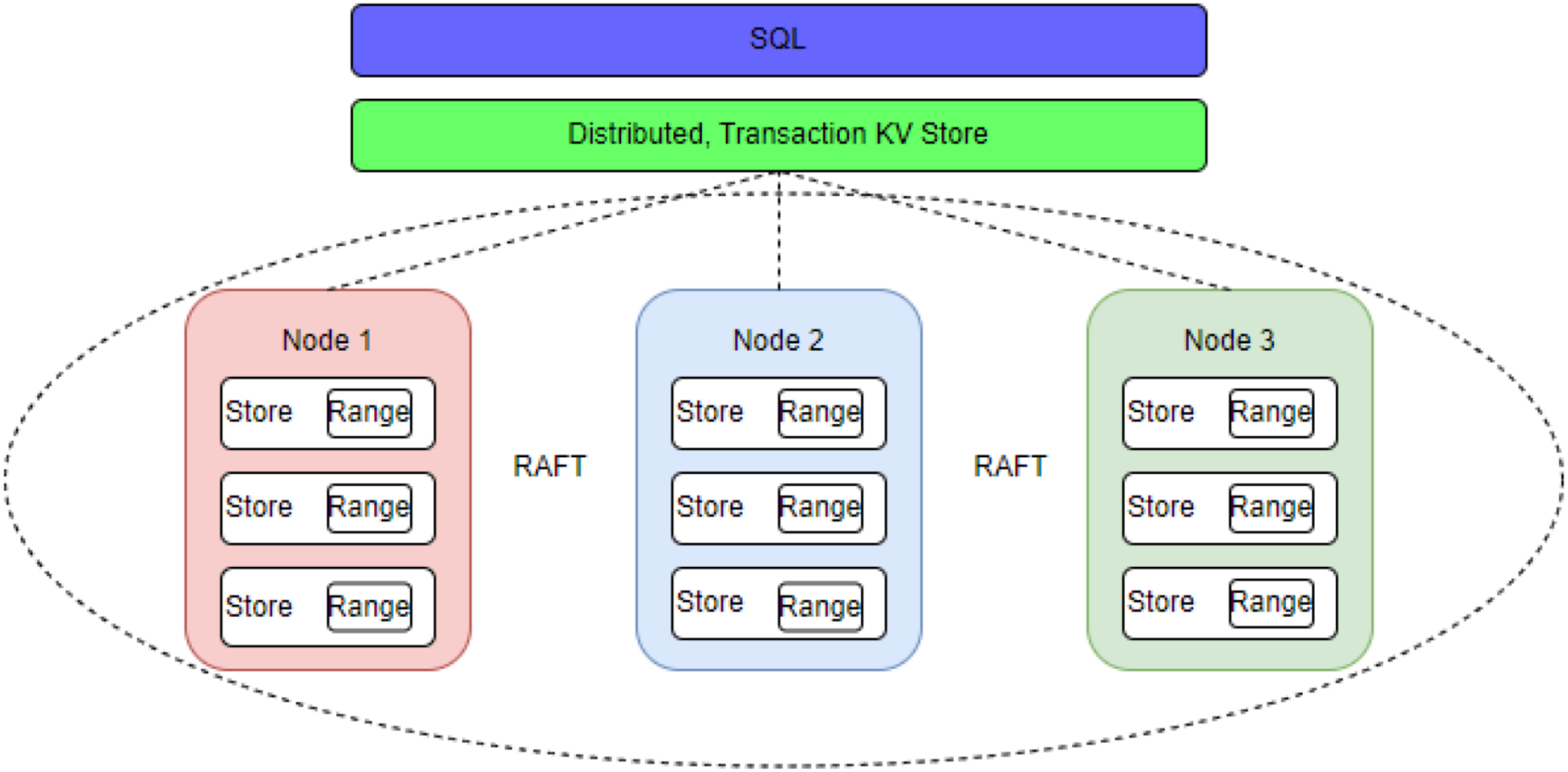


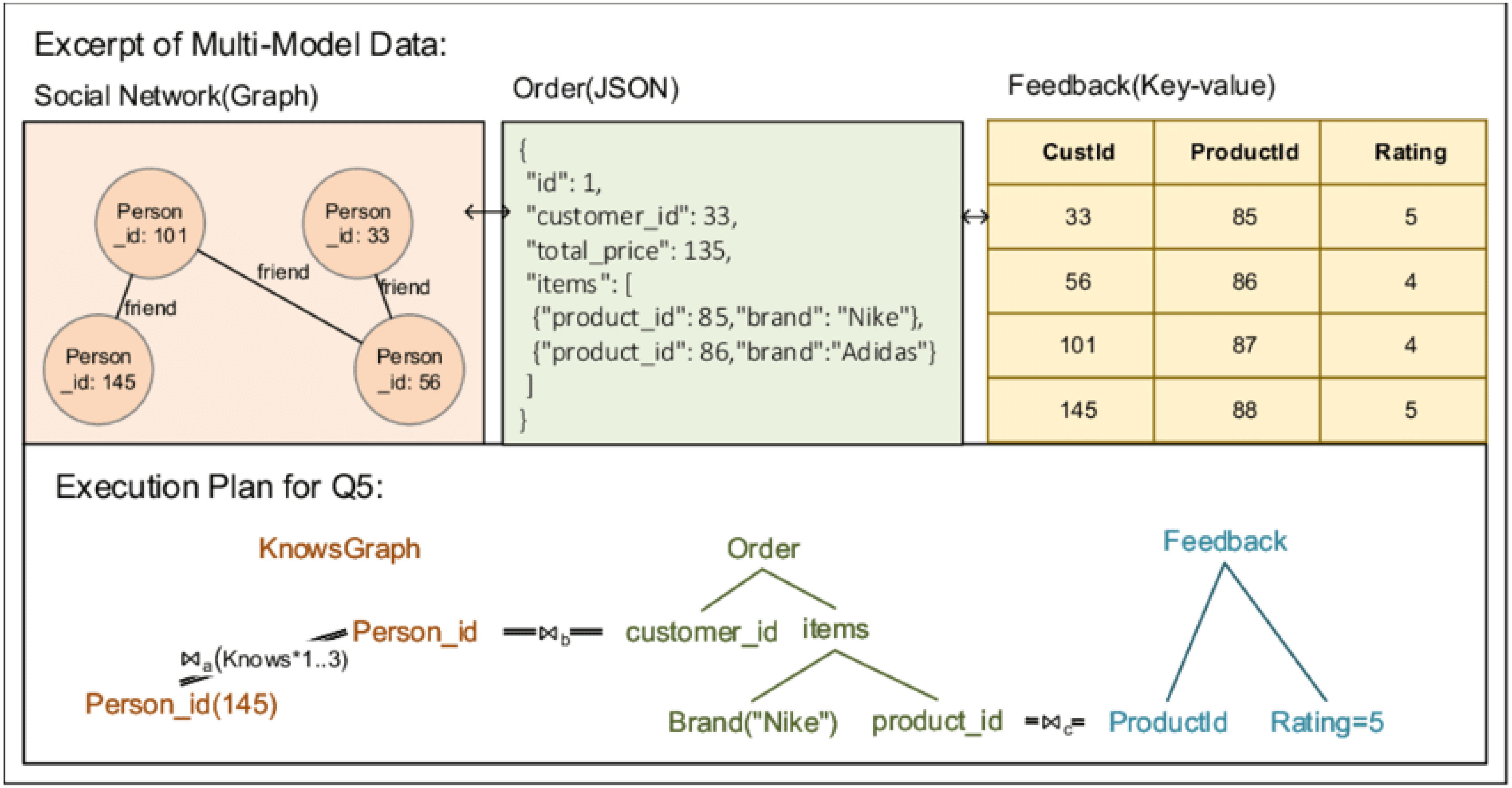
Fig 11 : NewSQL databases (CockroachDB) [11]

Multi-model databases

Multi-model databases are a type of database management system (DBMS) that can support multiple data models, such as relational, document, key-value, graph, and more. They are designed to provide flexibility and efficiency in handling different types of data, making them popular in modern, complex applications.

- **Advantages of Multi-model databases:**
  - Flexibility in handling different types of data, reducing the need for multiple databases.
  - Improved efficiency and cost-effectiveness, as multiple data models can be managed within a single system.
  - Easy integration with existing applications and systems.
  - Scalability and availability through distributed architectures.
  - Better support for complex data relationships and analytics.
- **Disadvantages of Multi-model databases:**
  - Increased complexity compared to single-model databases.
  - Higher learning curve for developers and administrators.
  - Limited support for some data models or query languages.
  - Performance may be affected when using multiple data models in the same database.
- **Examples:**
  - ArangoDB, OrientDB, Couchbase

Multi-model databases



```
{
 "id": 1,
 "customer_id": 33,
 "total_price": 135,
 "items": [
  {"product_id": 85,"brand": "Nike"},
  {"product_id": 86,"brand":"Adidas"}
 ]
}
```

CustId	ProductId	Rating
33	85	5
56	86	4
101	87	4
145	88	5

Fig 12 : Multi-model databases [12]

Comparative Analysis

Database	Speed	Use Case	Support	Examples
Flat File	Slowest (Read)	File Systems	High	CSV, XML
Relational	Slow (Write)	Transactional Databases	High	MySQL
Key-Value	Fastest (Read)	Authentication	Medium	JSON, Redis
Document	Fast (Read)	Single Entity queries	High	MongoDB
Graph	Fastest (relationship)	Social Network	Low	Neo4j, OrientDB
Column	Fast (Write)	Real-time analytics	Medium	Cassandra, HBase
Native XML	Fast (XML)	Hierarchical data, websites	Low	eXist, BaseX

Comparative Analysis

Database	Speed	Use Case	Support	Examples
Hierarchical	Fast (Tree)	File Systems, DNS	Low	DNS, LDAP, IMS
Network	Fast (Tree + networks)	Complex Relationships, Graphs, and Trees	Low	IDMS
Time Series	Fast (storage)	Time-Stamped Data, Sensor data	Medium	OpenTSDB, Prometheus, InfluxDB, TimescaleDB
NewSQL	Fast (Distributed)	High-Volume, High-Velocity Transactions	Medium	MemSQL, VoltDB, Spanner, Calvin, CockroachDB
Multi-Model	Fast (Different Databases)	Multiple Data Models, Complex Data	Medium	ArangoDB, OrientDB, Couchbase

References

1. “Nodegoat” CSV and XML Figures, accessed 02/10/2022 [*https://nodegoat.net/guide.s/22/upload-a-csv-file*]
2. “LucidCharts” Relational database Figure, accessed 02/10/2022 [*https://www.lucidchart.com/pages/database-diagram/database-models*]
3. “Redis” Key-value database Figure, accessed 02/10/2022 [*https://redis.com/nosql/key-value-databases/*]
4. “Future-Processing” Document database Figure, accessed 02/10/2022 [*https://kariera.future-processing.pl/blog/rules-of-data-conversion-from-document-to-relational-databases/*]
5. “Neo4j” Graph Based Databases figure, accessed 02/10/2022, [*https://neo4j.com/developer/graph-database/*]
6. “MariaDB” Column Based Databases figure, accessed 02/10/2022, [https://mariadb.com/resources/blog/why-is-columnstore-important/]
7. “Manning” Native XML Databases, accessed 06/10/2022, [*https://livebook.manning.com/book/making-sense-of-nosql/chapter-5/*]

References

8. “Hyperskill” Hierarchical database model, accessed 05/03/2023, [*https://hyperskill.org/learn/step/17042*]
9. “GeeksForFeeks” Difference between Network and Relational data model, accessed 05/03/2023, [*https://www.geeksforgeeks.org/difference-between-network-and-relational-data-model/*]
10. “Dongyueweb” Introduction to Time Series Database, accessed 06/03/2023, [*https://blog.dongyueweb.com/introduction_to_time_series_database.html]*
11. Pina, E.; Sá, F.; Bernardino, J. NewSQL Databases Assessment: CockroachDB, MariaDB Xpand, and VoltDB. Future Internet 2023, 15, 10. https://doi.org/10.3390/fi15010010
12. Zhang, Chao & Lu, Jiaheng. (2021). Holistic evaluation in multi-model databases benchmarking. Distributed and Parallel Databases. 39. 10.1007/s10619-019-07279-6.
13. “GeeksForGeeks” SQL vs NoSQL, accessed 02/10/2022, [*https://www.geeksforgeeks.org/difference-between-sql-and-nosql/*]

References

14. “GeeksForGeeks” Intro to NoSQL, accessed 02/10/2022, [*https://www.geeksforgeeks.org/introduction-to-nosql/*]
15. “MongoDB” Types of NoSQL Databases, accessed 03/10/2022, [*https://www.mongodb.com/scale/types-of-nosql-databases*]

Thank You