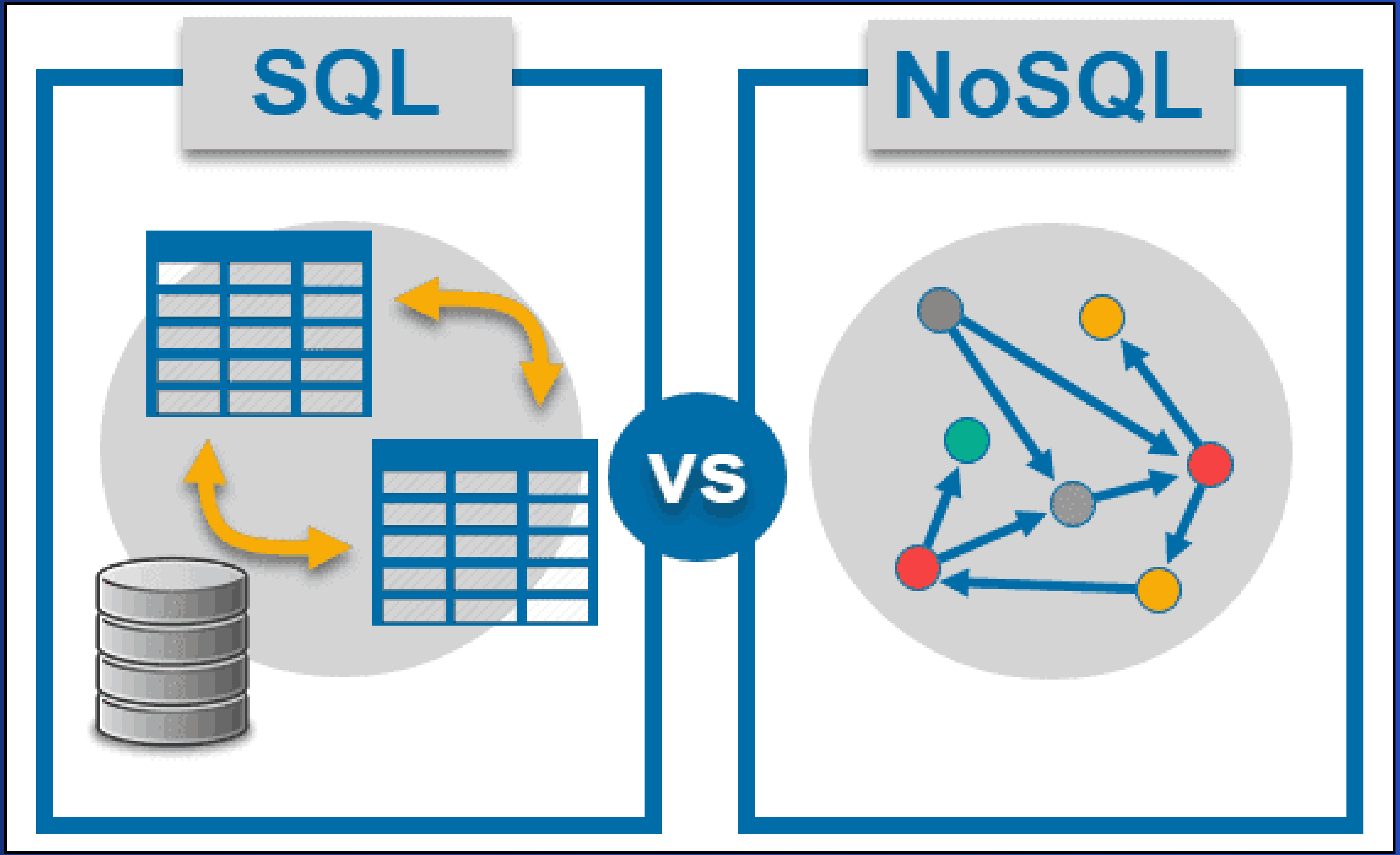
print("Hello, Armenia!")



CORE DIFFERENCES

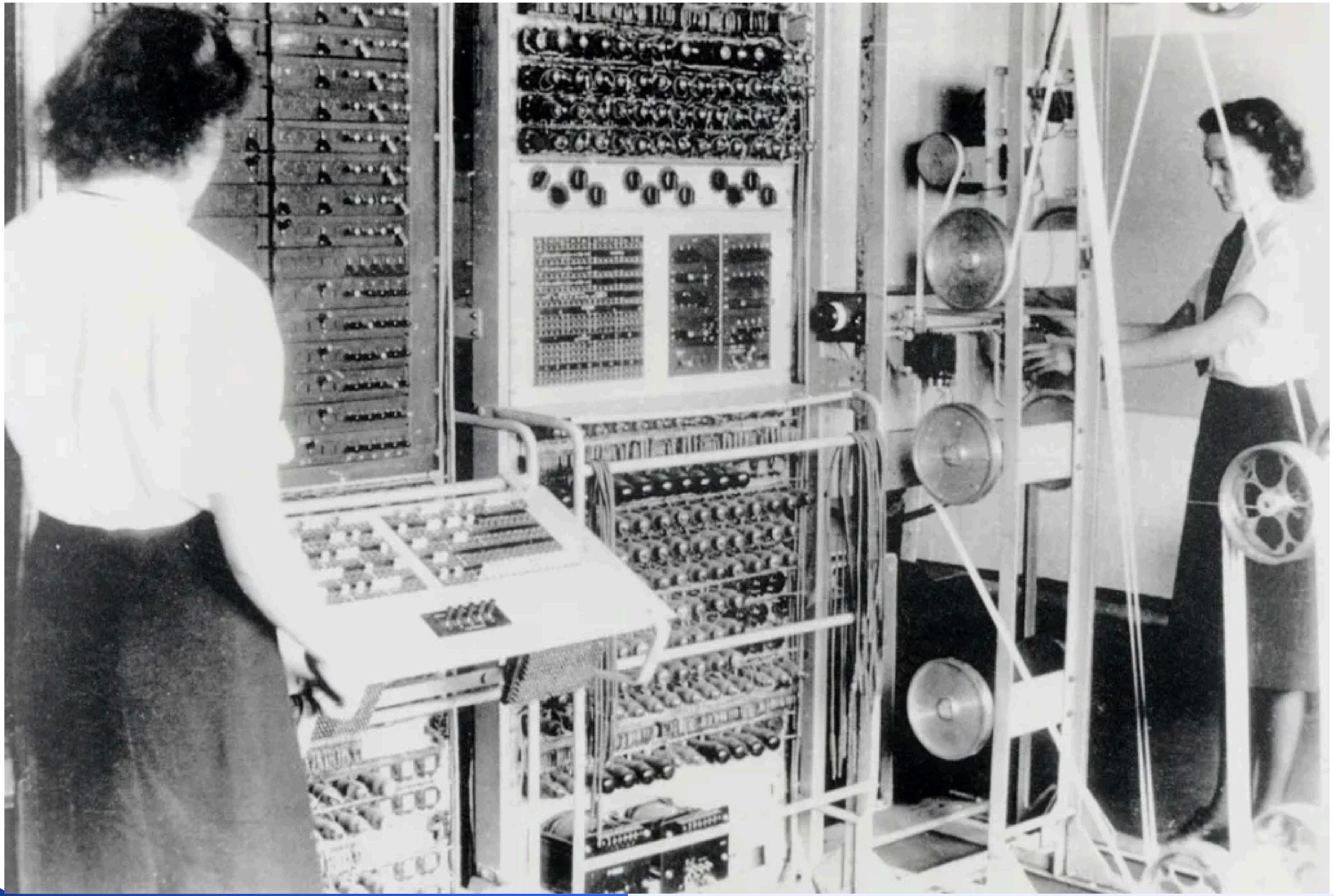
Ms. Aya Guseinova

August 28, 2025

Overview

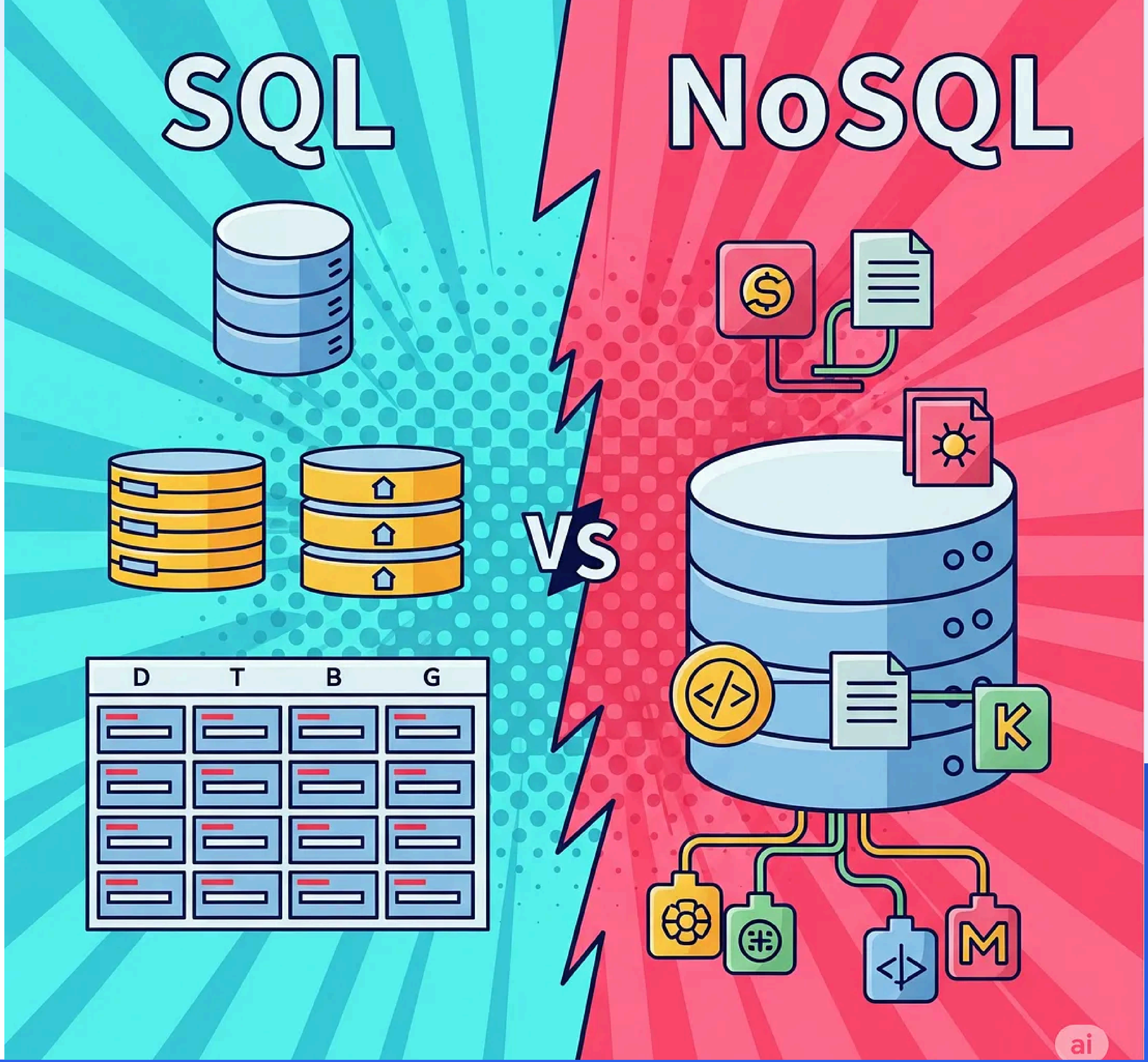
- Big Data (History)
- Understanding Databases
- Definition of SQL & NoSQL
- SQL & NoSQL Databases
- Understanding Scalability

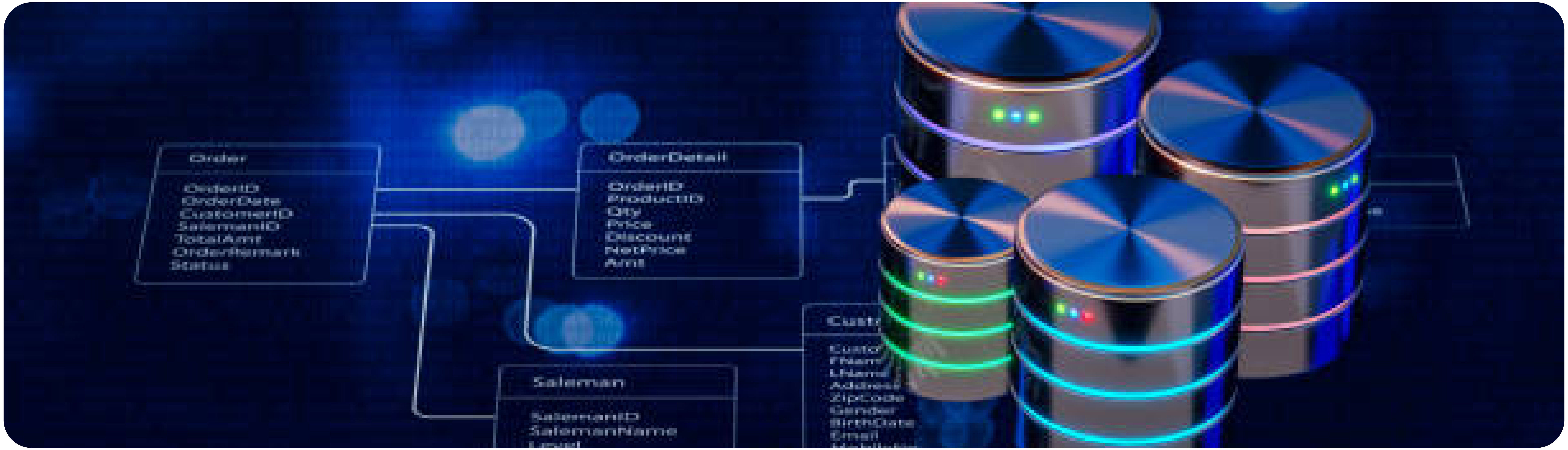
Big Data in 20th century



A Colossus Mark 2 codebreaking computer being operated by Dorothy Du Boisson (left) and Elsie Booker (right), 1943

Databases Activity





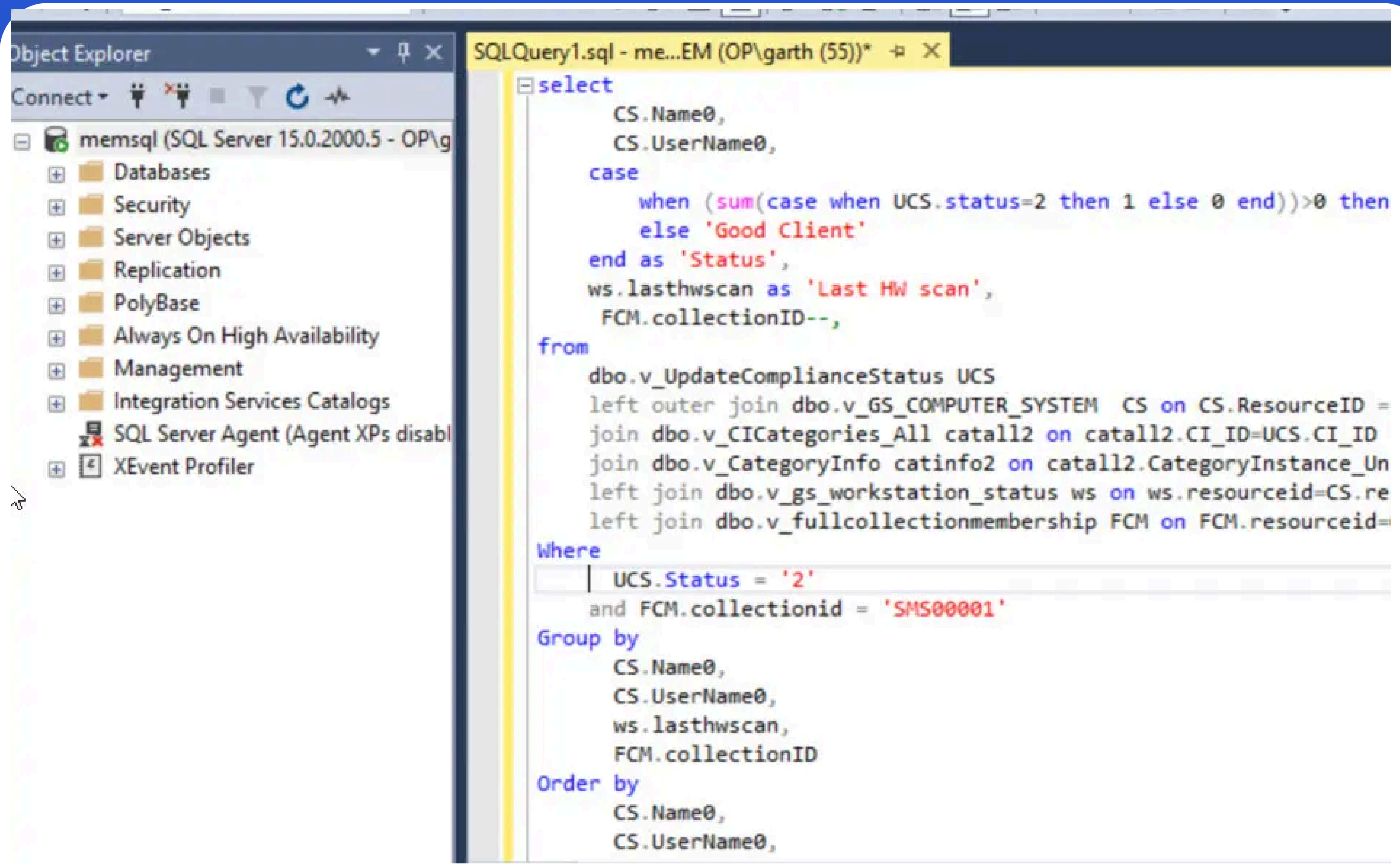
Understanding Databases:

The Backbone of Data Management

Databases are structured collections of data that allow for efficient storage, retrieval, and management of information. They play a crucial role in enabling organizations to handle large volumes of data and support decision-making processes. With the rise of digital information, databases have become essential for businesses, applications, and services across various industries.

Definition of SQL

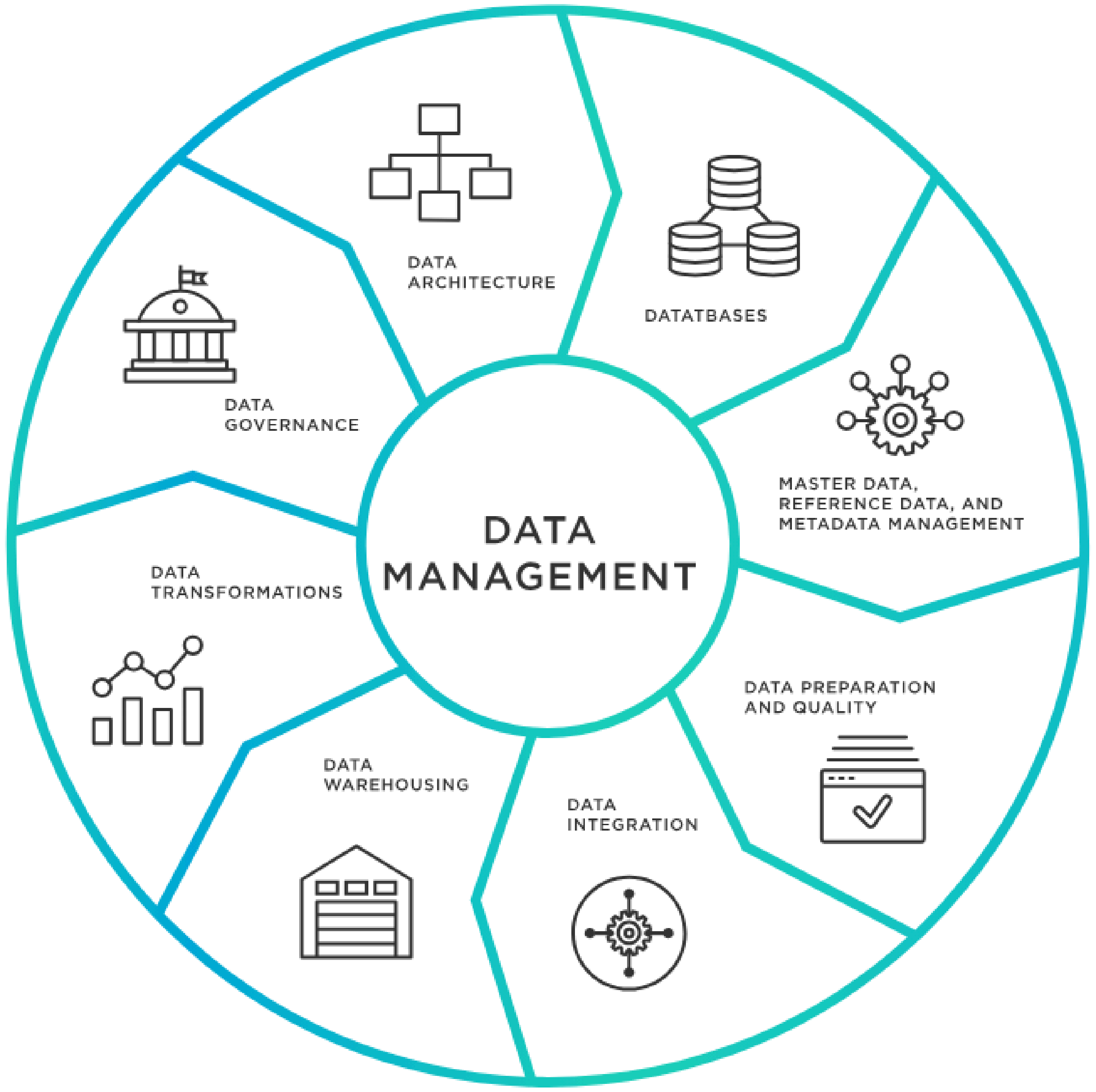
SQL, or Structured Query Language, is a standardized programming language used to manage and manipulate relational databases. It allows users to perform various operations such as querying, updating, and managing data efficiently.



Object Explorer
Connect ▾
memsql (SQL Server 15.0.2000.5 - OP\g
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - me...EM (OP\garth (55))*

```
select
      CS.Name0,
      CS.UserName0,
    case
        when (sum(case when UCS.status=2 then 1 else 0 end))>0 then
        else 'Good Client'
    end as 'Status',
    ws.lasthwscan as 'Last HW scan',
     FCM.collectionID--,
from
    dbo.v_UpdateComplianceStatus UCS
    left outer join dbo.v_GS_COMPUTER_SYSTEM  CS on CS.ResourceID =
    join dbo.v_CICategories_All catall2 on catall2.CI_ID=UCS.CI_ID
    join dbo.v_CategoryInfo catinfo2 on catall2.CategoryInstance_Un
    left join dbo.v_gs_workstation_status ws on ws.resourceid=CS.re
    left join dbo.v_fullcollectionmembership FCM on FCM.resourceid=
Where
      UCS.Status = '2'
    and FCM.collectionid = 'SMS00001'
Group by
      CS.Name0,
      CS.UserName0,
      ws.lasthwscan,
      FCM.collectionID
Order by
      CS.Name0,
      CS.UserName0,
```

Data Querying

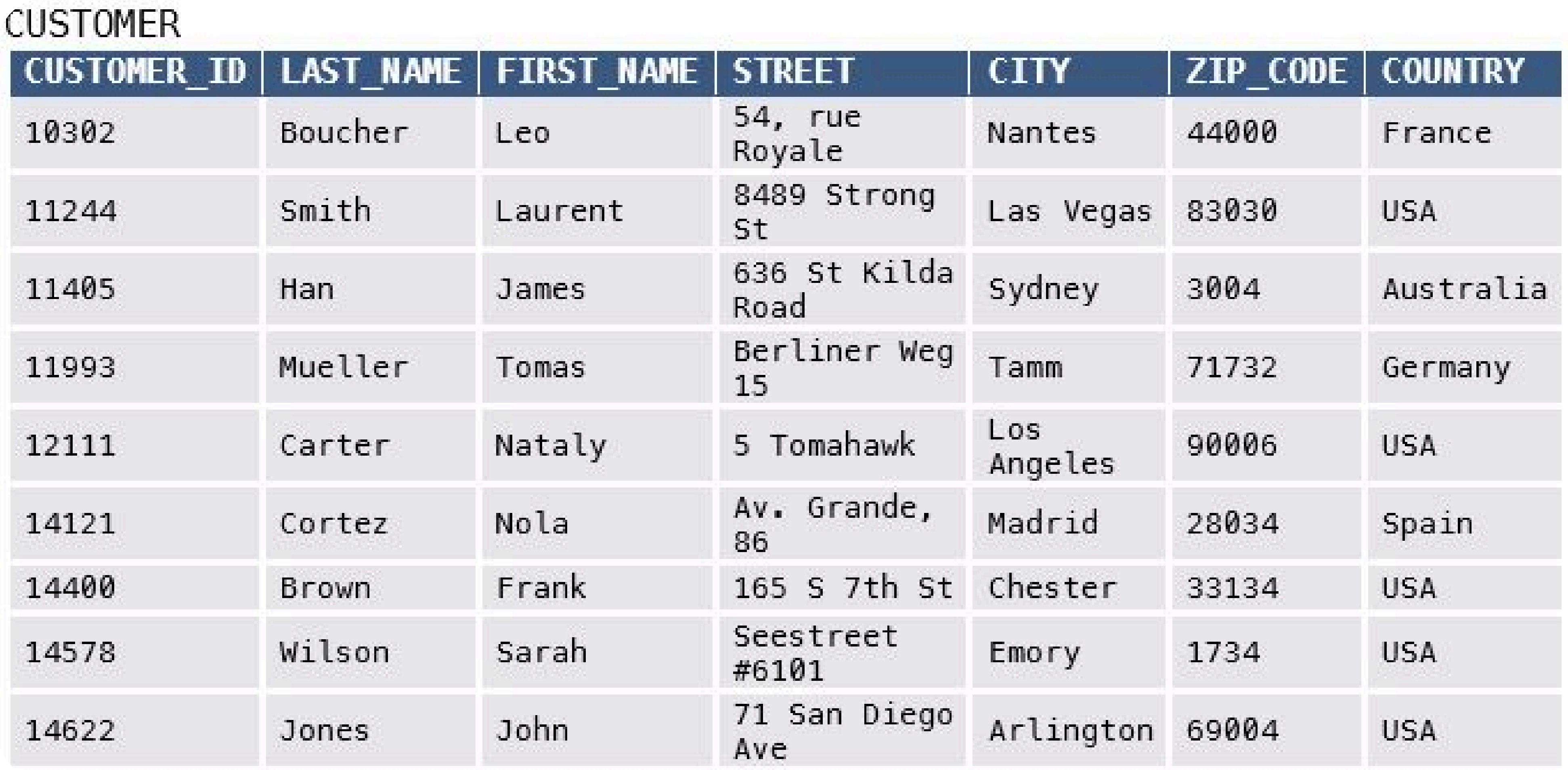


Data Management

Purpose of SQL

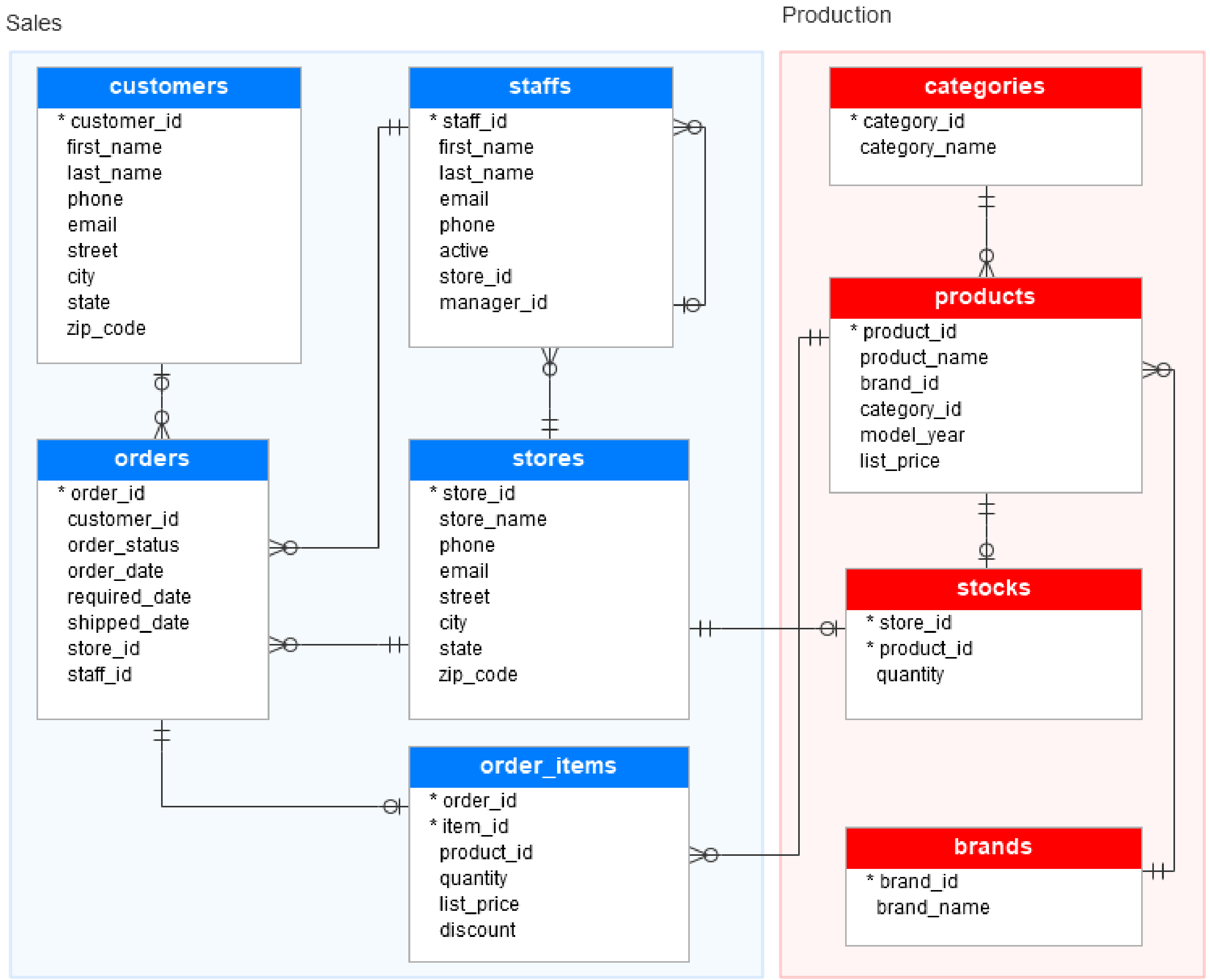
SQL is primarily used for tasks like data retrieval, insertion, updating, and deletion in relational database management systems **(RDBMS)**. It provides a systematic way to handle structured data, making it essential for applications requiring complex queries and transactions.

Common Use Cases



CUSTOMER

CUSTOMER_ID	LAST_NAME	FIRST_NAME	STREET	CITY	ZIP_CODE	COUNTRY
10302	Boucher	Leo	54, rue Royale	Nantes	44000	France
11244	Smith	Laurent	8489 Strong St	Las Vegas	83030	USA
11405	Han	James	636 St Kilda Road	Sydney	3004	Australia
11993	Mueller	Tomas	Berliner Weg 15	Tamm	71732	Germany
12111	Carter	Nataly	5 Tomahawk	Los Angeles	90006	USA
14121	Cortez	Nola	Av. Grande, 86	Madrid	28034	Spain
14400	Brown	Frank	165 S 7th St	Chester	33134	USA
14578	Wilson	Sarah	Seestreet #6101	Emory	1734	USA
14622	Jones	John	71 San Diego Ave	Arlington	69004	USA



Sales

customers
* customer_id
first_name
last_name
phone
email
street
city
state
zip_code

staffs
* staff_id
first_name
last_name
email
phone
active
store_id
manager_id

orders
* order_id
customer_id
order_status
order_date
required_date
shipped_date
store_id
staff_id

stores
* store_id
store_name
phone
email
street
city
state
zip_code

order_items
* order_id
* item_id
product_id
quantity
list_price
discount

Production

categories
* category_id
category_name

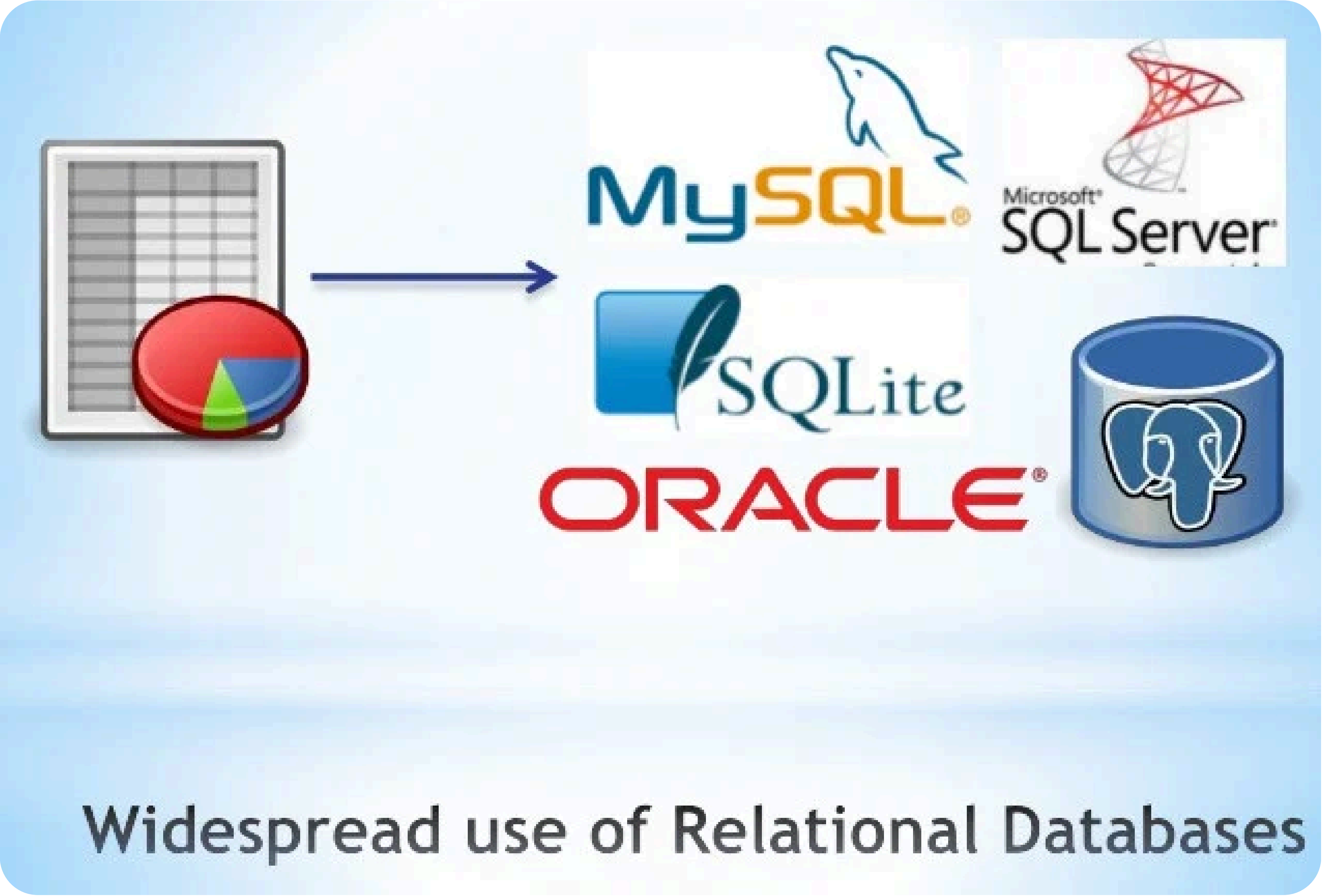
products
* product_id
product_name
brand_id
category_id
model_year
list_price

stocks
* store_id
* product_id
quantity

brands
* brand_id
brand_name

Common use cases for **SQL** include managing customer relationship management **(CRM)** systems, e-commerce platforms, and financial applications, where a structured data format is crucial for business operations and reporting.

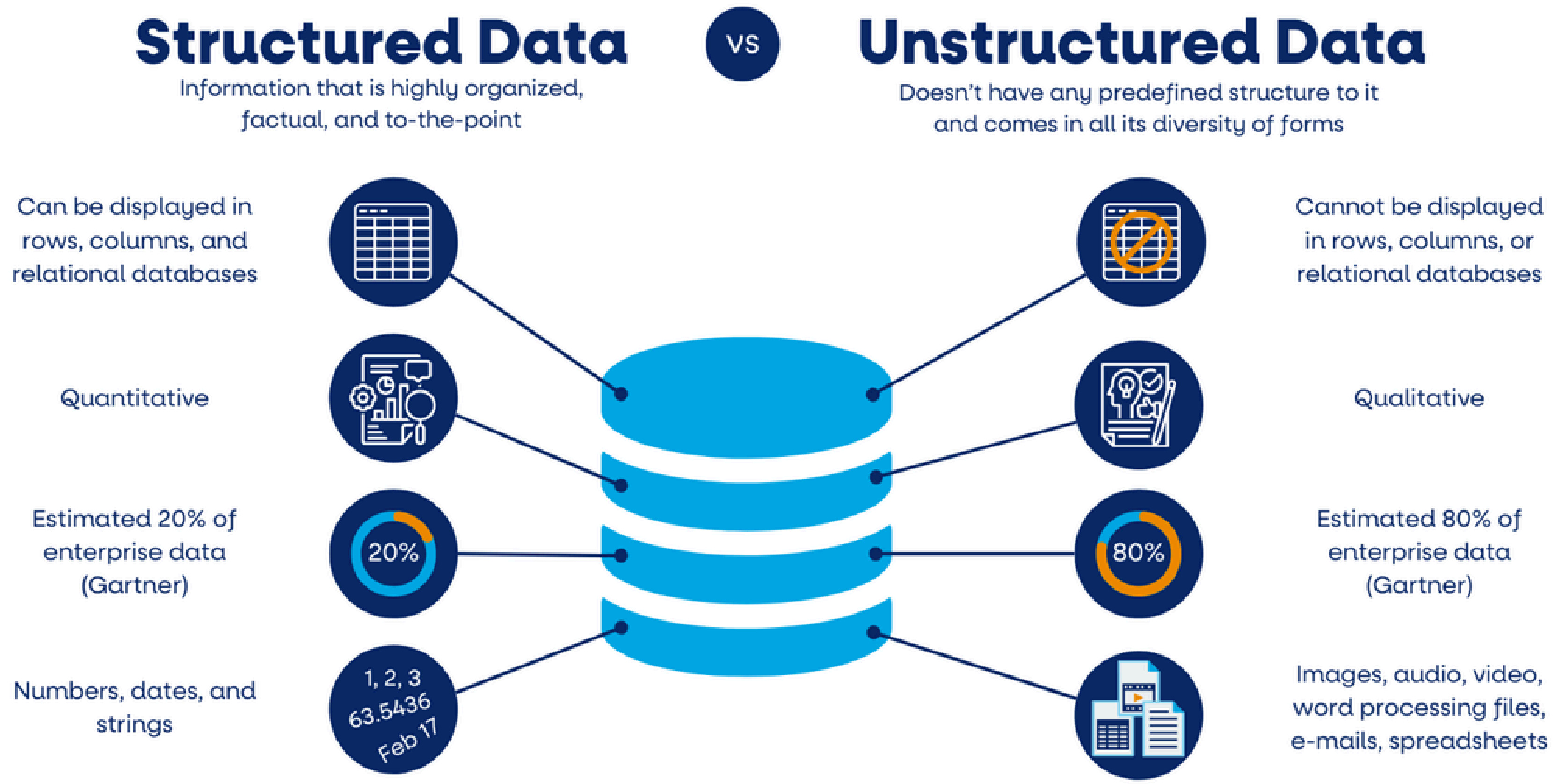
Most popular SQL databases



Widespread use of Relational Databases

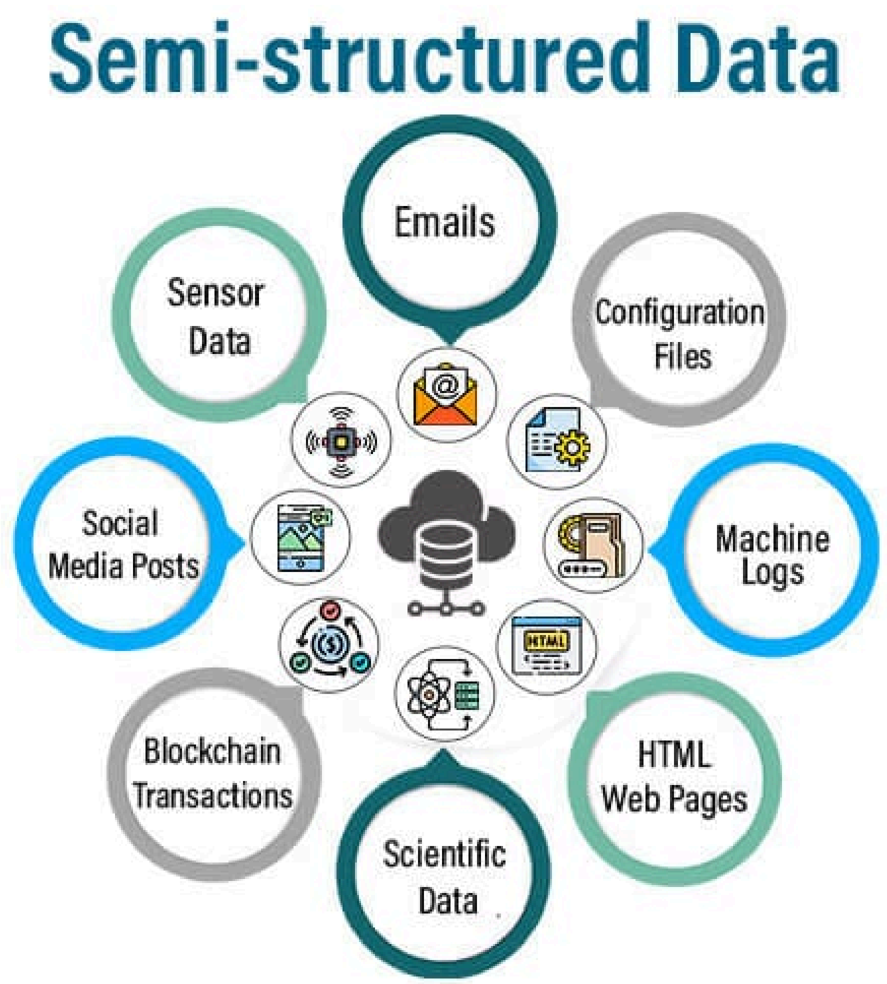
Definition of NoSQL

NoSQL, which stands for **'Not Only SQL',** refers to a category of database management systems that do not use traditional relational database structures. They are designed to handle large volumes of unstructured or semi-structured data, making them suitable for modern applications that require flexibility and scalability.



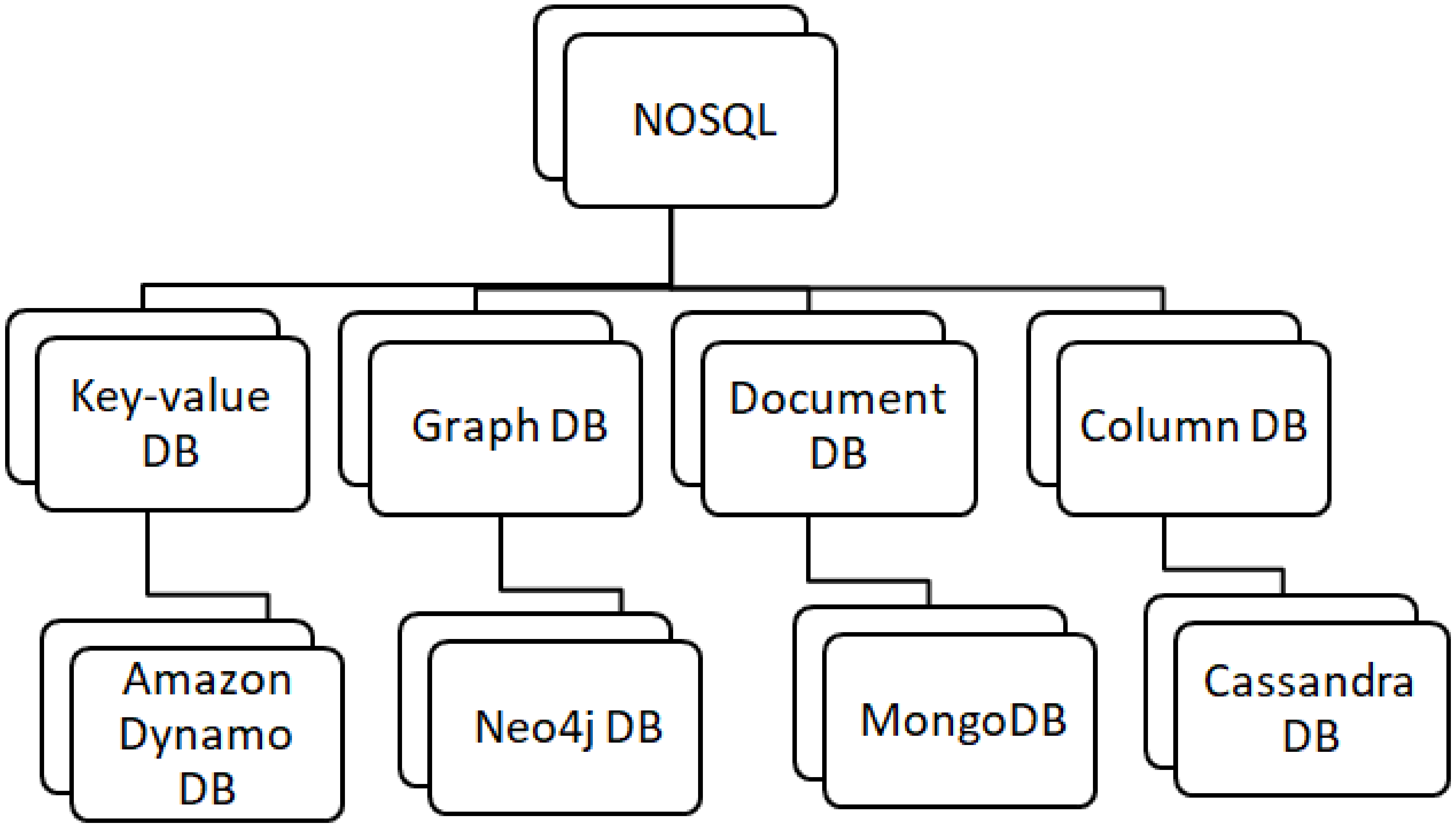
Structured Data
Information that is highly organized, factual, and to-the-point
VS
Unstructured Data
Doesn't have any predefined structure to it and comes in all its diversity of forms
Can be displayed in rows, columns, and relational databases
Cannot be displayed in rows, columns, or relational databases
Quantitative
Qualitative
Estimated 20% of enterprise data (Gartner)
20%
80%
Estimated 80% of enterprise data (Gartner)
Numbers, dates, and strings
1, 2, 3
63.5436
Feb 17
Images, audio, video, word processing files, e-mails, spreadsheets

Unstructured Data



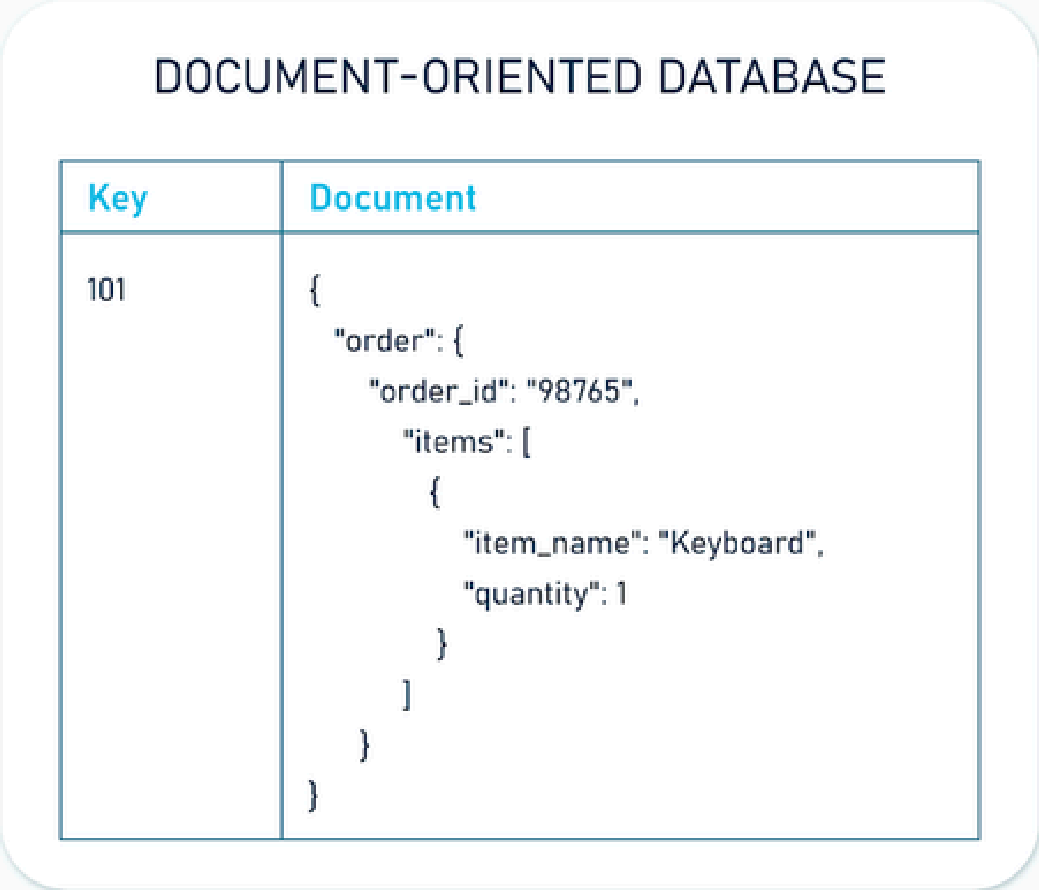
Semi-Structured Data (XML, JSON)

Purpose of NoSQL



NoSQL databases are designed for distributed data architecture, allowing horizontal scaling across multiple servers. This enables them to handle increased loads and larger datasets without compromising performance, making them ideal for big data applications.

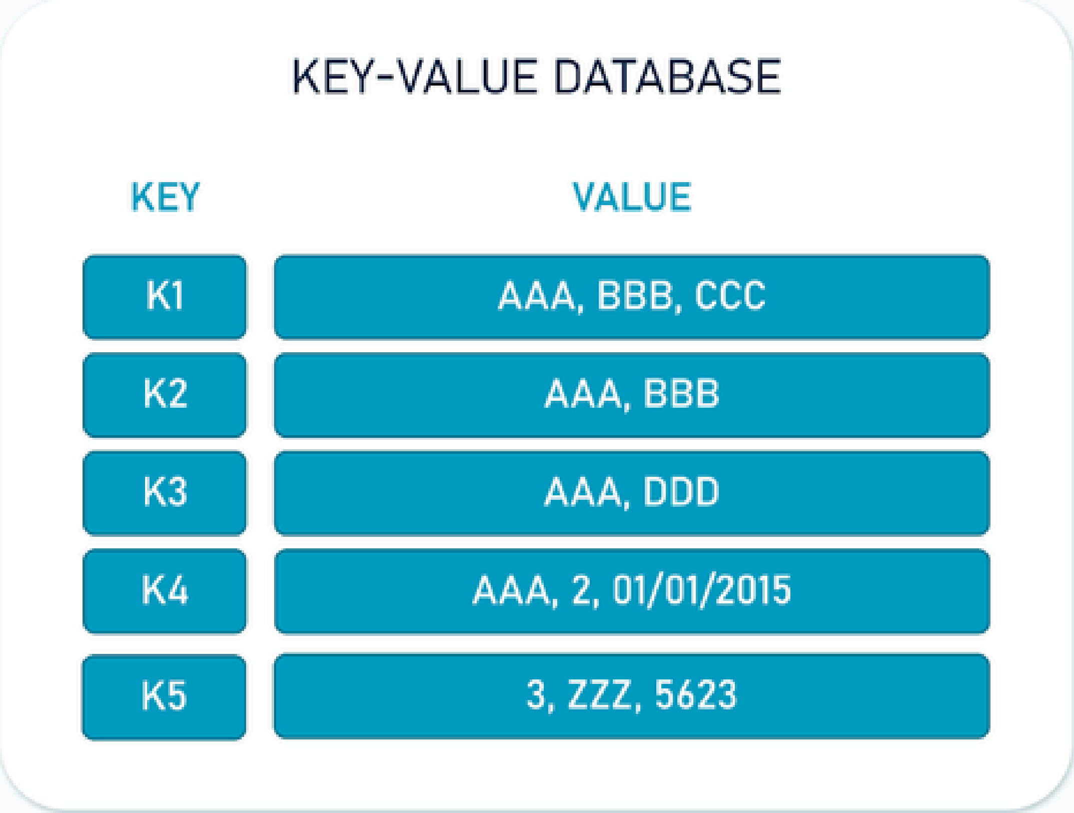
NoSQL Database Types



DOCUMENT-ORIENTED DATABASE

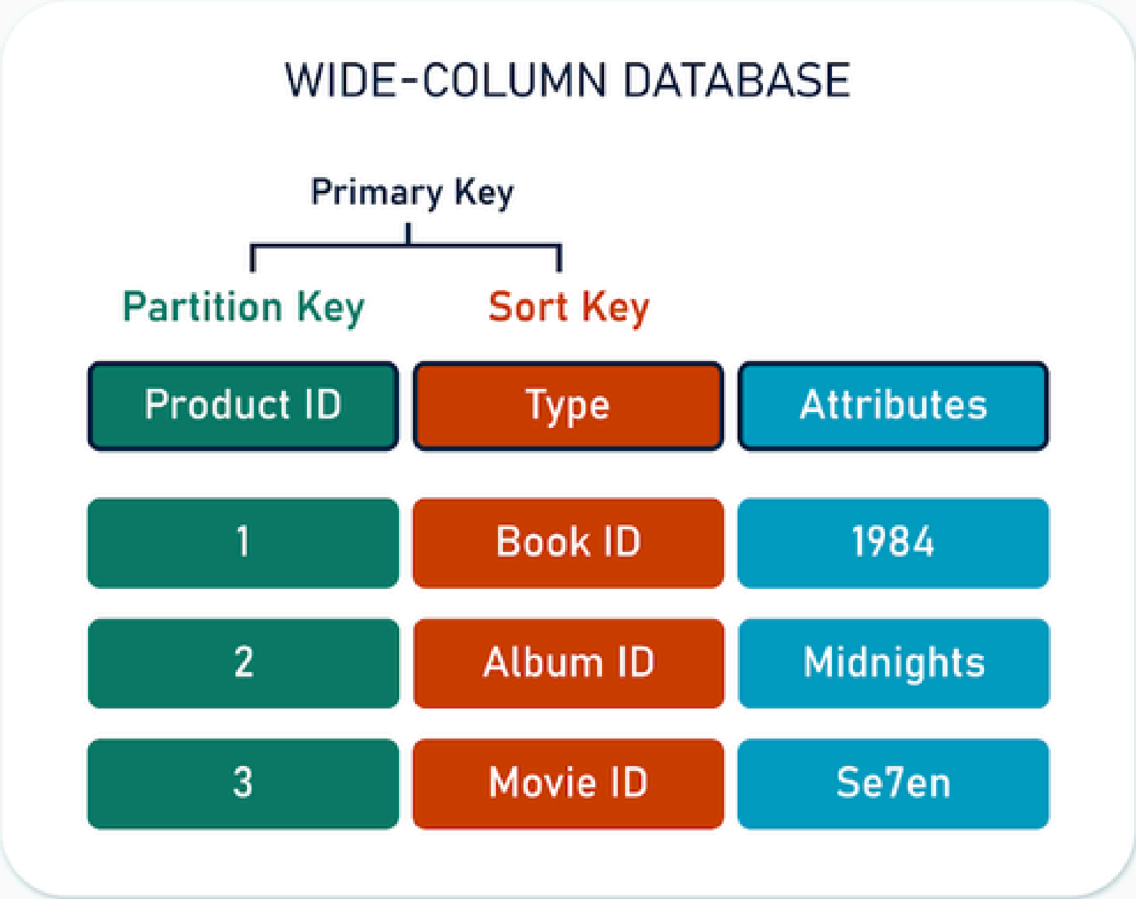
Key	Document

101	{
	"order": {
	"order_id": "98765",
	"items": [
	{
	"item_name": "Keyboard",
	"quantity": 1
	}
	]
	}
	}



KEY-VALUE DATABASE

KEY	VALUE
K1	AAA, BBB, CCC
K2	AAA, BBB
K3	AAA, DDD
K4	AAA, 2, 01/01/2015
K5	3, ZZZ, 5623

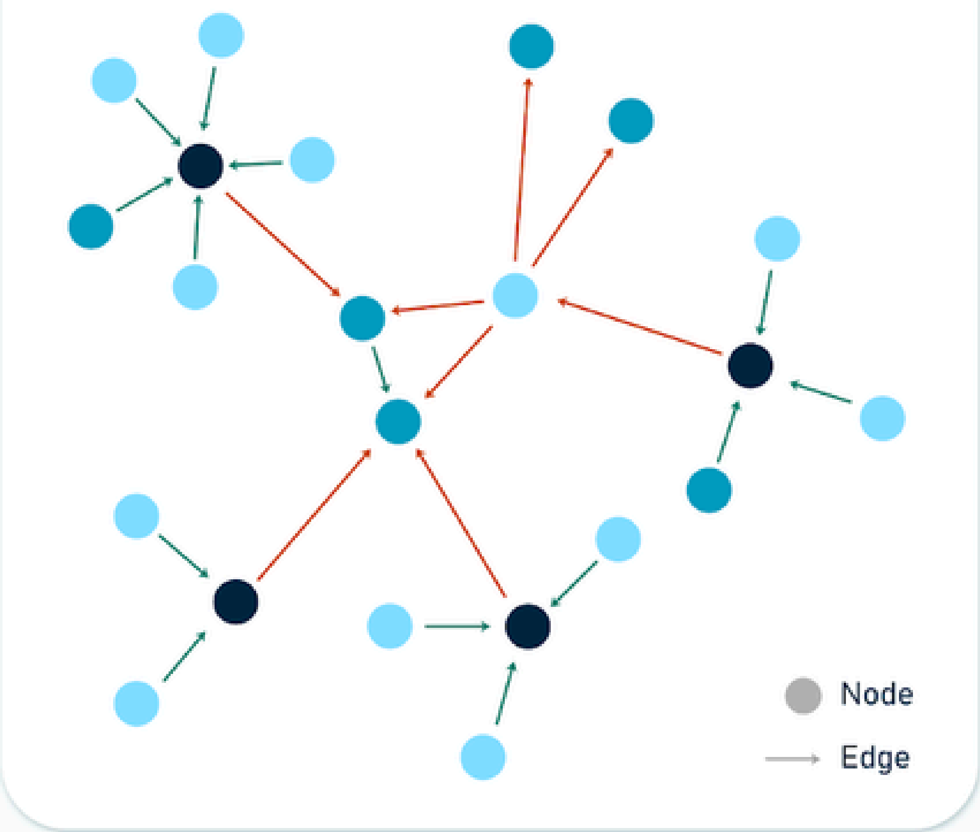


WIDE-COLUMN DATABASE

Primary Key

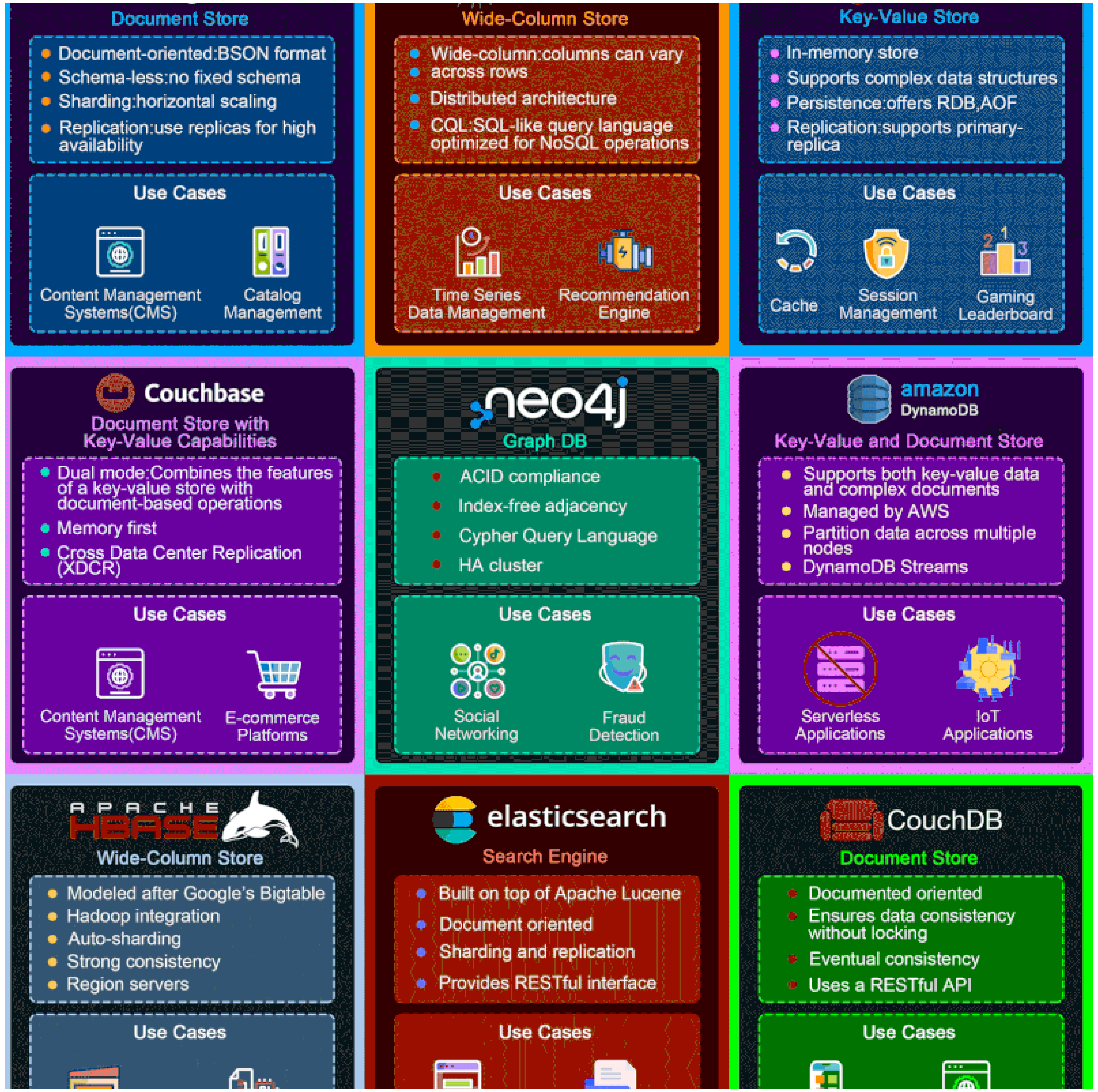
Partition Key	Sort Key	
Product ID	Type	Attributes
1	Book ID	1984
2	Album ID	Midnights
3	Movie ID	Se7en

GRAPH DATABASE



Common Use Cases

Common use cases for **NoSQL** databases include real-time web applications, big data analytics, content management systems, and IoT applications. They excel in scenarios where data structures are constantly evolving or are not well-defined.



Document Store
- Document-oriented:BSON format
- Schema-less:no fixed schema
- Sharding:horizontal scaling
- Replication:use replicas for high availability

Use Cases

Wide-Column Store
- Wide-column:columns can vary across rows
- Distributed architecture
- CQL:SQL-like query language optimized for NoSQL operations

Use Cases

Key-Value Store
- In-memory store
- Supports complex data structures
- Persistence:offers RDB,AOF
- Replication:supports primary-replica

Use Cases

Couchbase
Document Store with Key-Value Capabilities
- Dual mode:Combines the features of a key-value store with document-based operations
- Memory first
- Cross Data Center Replication (XDCR)

Use Cases

neo4j
Graph DB
- ACID compliance
- Index-free adjacency
- Cypher Query Language
- HA cluster

Use Cases

amazon DynamoDB
Key-Value and Document Store
- Supports both key-value data and complex documents
- Managed by AWS
- Partition data across multiple nodes
- DynamoDB Streams

Use Cases

APACHE HBASE
Wide-Column Store
- Modeled after Google's Bigtable
- Hadoop integration
- Auto-sharding
- Strong consistency
- Region servers

Use Cases

elasticsearch
Search Engine
- Built on top of Apache Lucene
- Document oriented
- Sharding and replication
- Provides RESTful interface

Use Cases

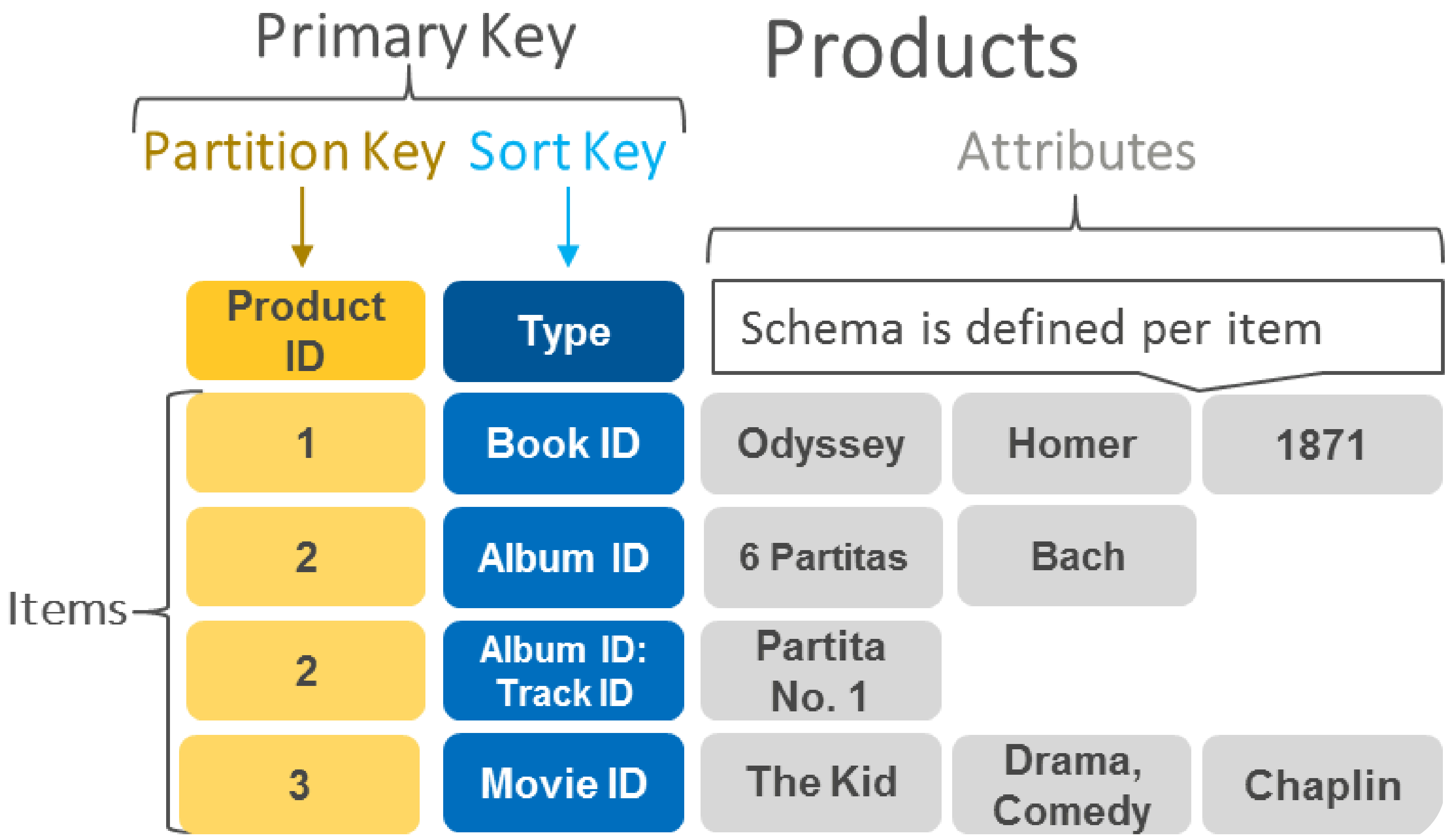
CouchDB
Document Store
- Documented oriented
- Ensures data consistency without locking
- Eventual consistency
- Uses a RESTful API

Use Cases

Amazon NoSQL Databases

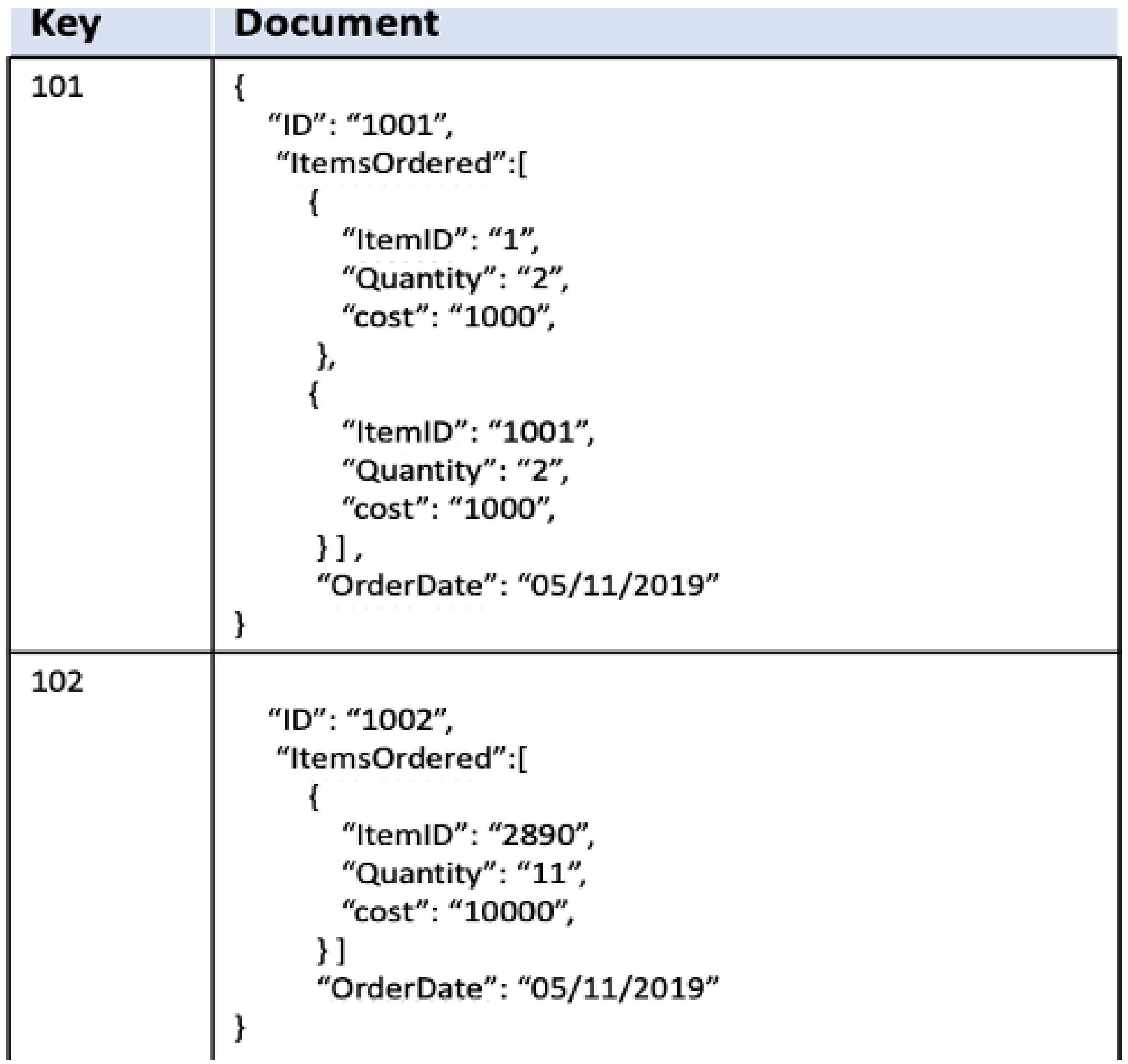
Amazon Dynamo DB

- **Key-Value data models** store as a collection of key-value pairs.
- Each data item is identified by a unique key.



Product ID	Type	Schema is defined per item		
1	Book ID	Odyssey	Homer	1871
2	Album ID	6 Partitas	Bach	
2	Album ID: Track ID	Partita No. 1		
3	Movie ID	The Kid	Drama, Comedy	Chaplin

Amazon DocumentDB (with MongoDB compatibility



Key	Document

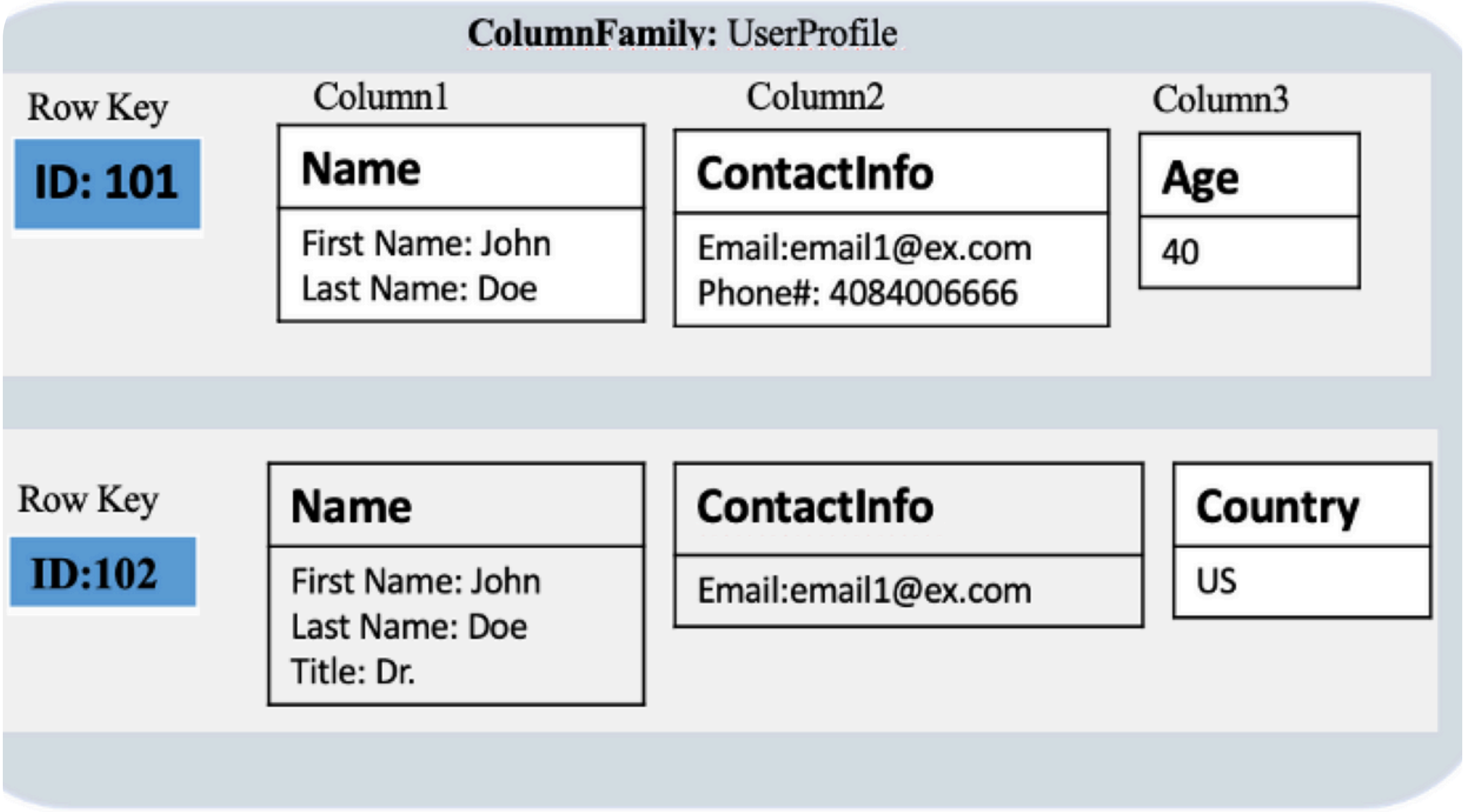
101	{
	"ID": "1001",
	"ItemsOrdered":[
	{
	"ItemID": "1",
	"Quantity": "2",
	"cost": "1000",
	},
	{
	"ItemID": "1001",
	"Quantity": "2",
	"cost": "1000",
	}] ,
	"OrderDate": "05/11/2019"
	}

102	"ID": "1002",
	"ItemsOrdered":[
	{
	"ItemID": "2890",
	"Quantity": "11",
	"cost": "10000",
	}]
	"OrderDate": "05/11/2019"
	}

Amazon NoSQL Databases

Amazon Keyspaces (for Apache Cassandra)

- **Wide-column** stores data in columns, not rows (flexibility, scalability)



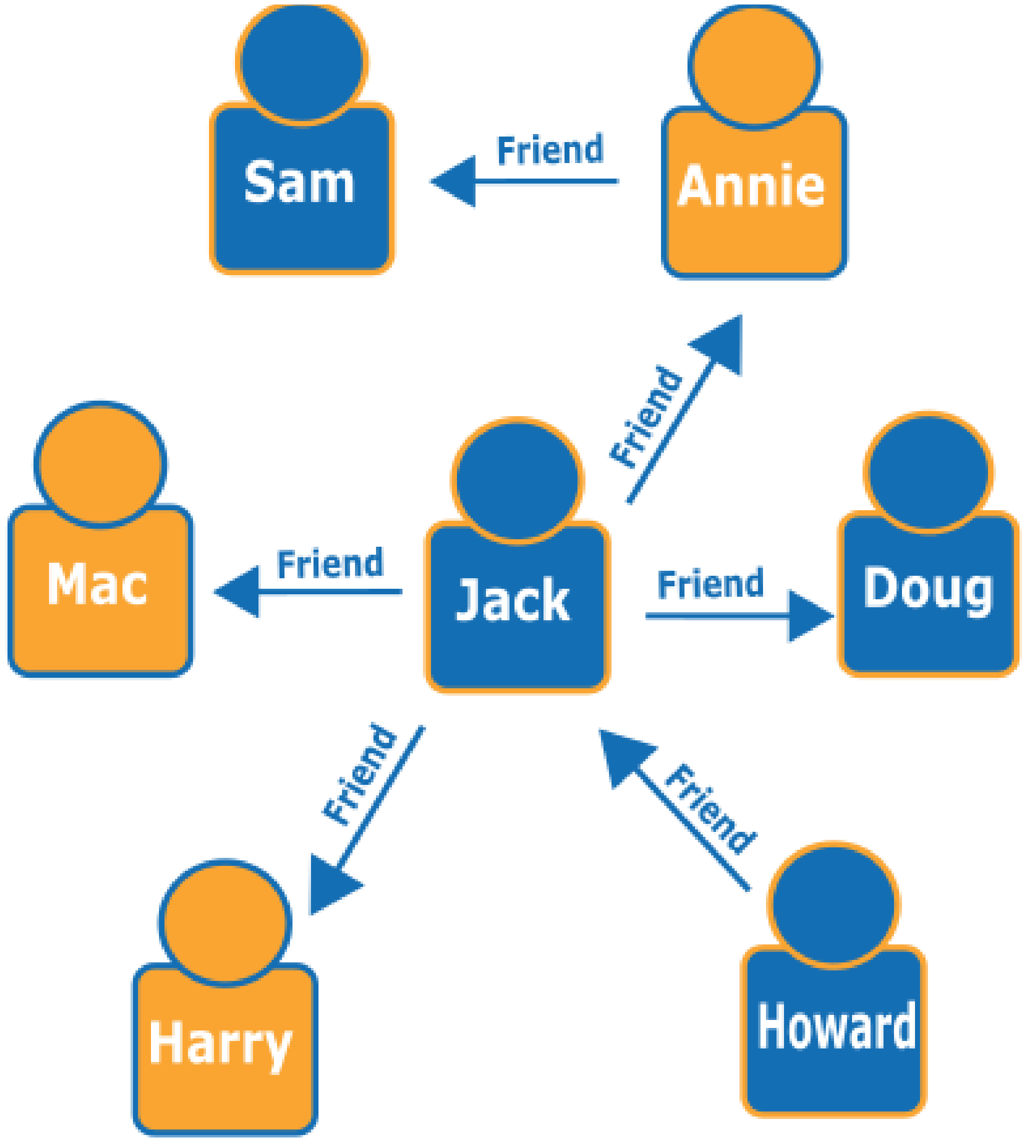
ColumnFamily: UserProfile

Row Key	Column1	Column2	Column3
ID: 101	**Name** First Name: John Last Name: Doe	**ContactInfo** Email:email1@ex.com Phone#: 4084006666	**Age** 40

Row Key			
ID:102	**Name** First Name: John Last Name: Doe Title: Dr.	**ContactInfo** Email:email1@ex.com	**Country** US

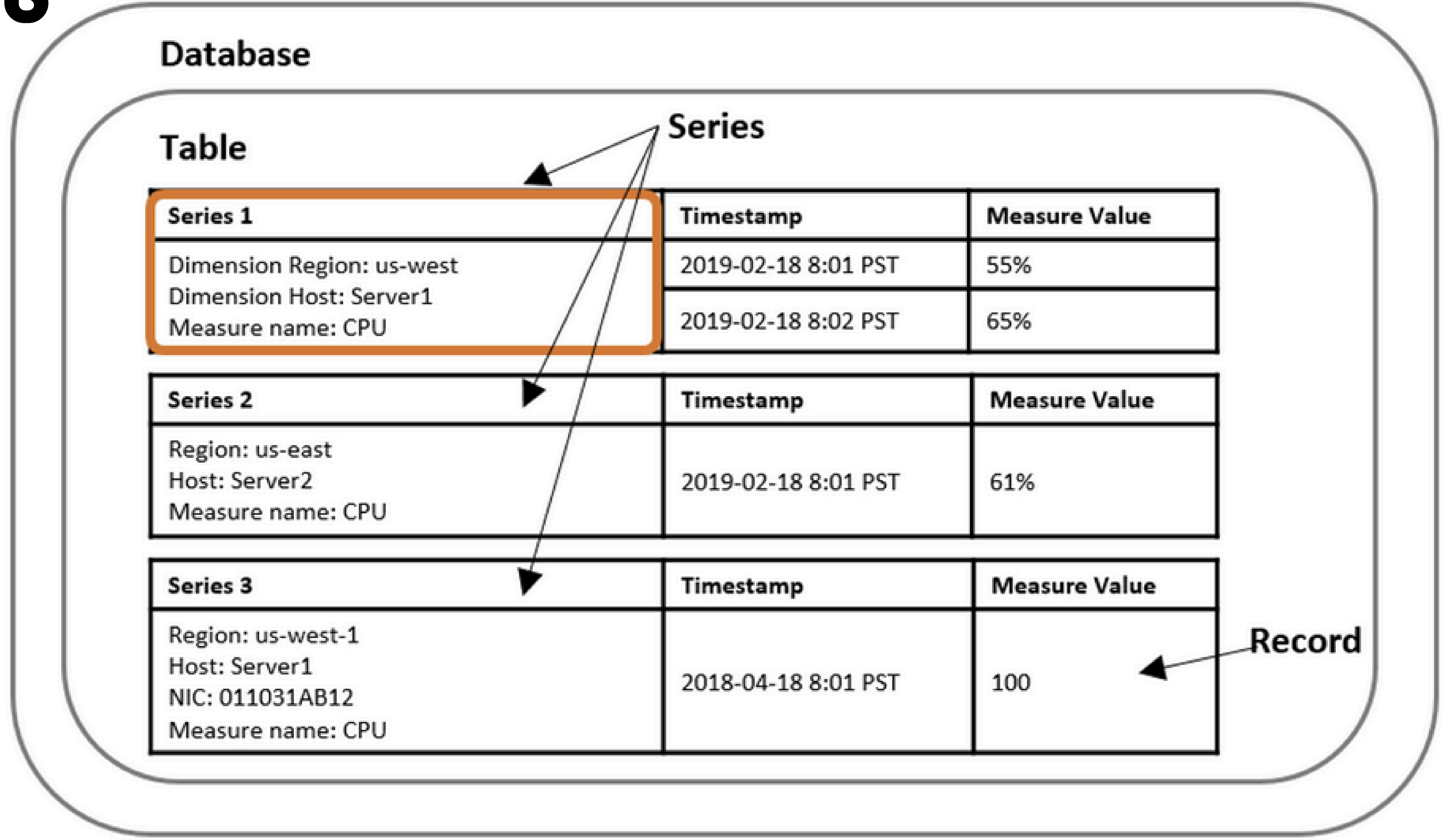
Amazon Neptune

- **Graphs** store and query highly connected data.



Time Series

- Designed to store and retrieve data records that are sequenced by time, which are sets of data points that are associated with timestamps and stored in time sequence order.



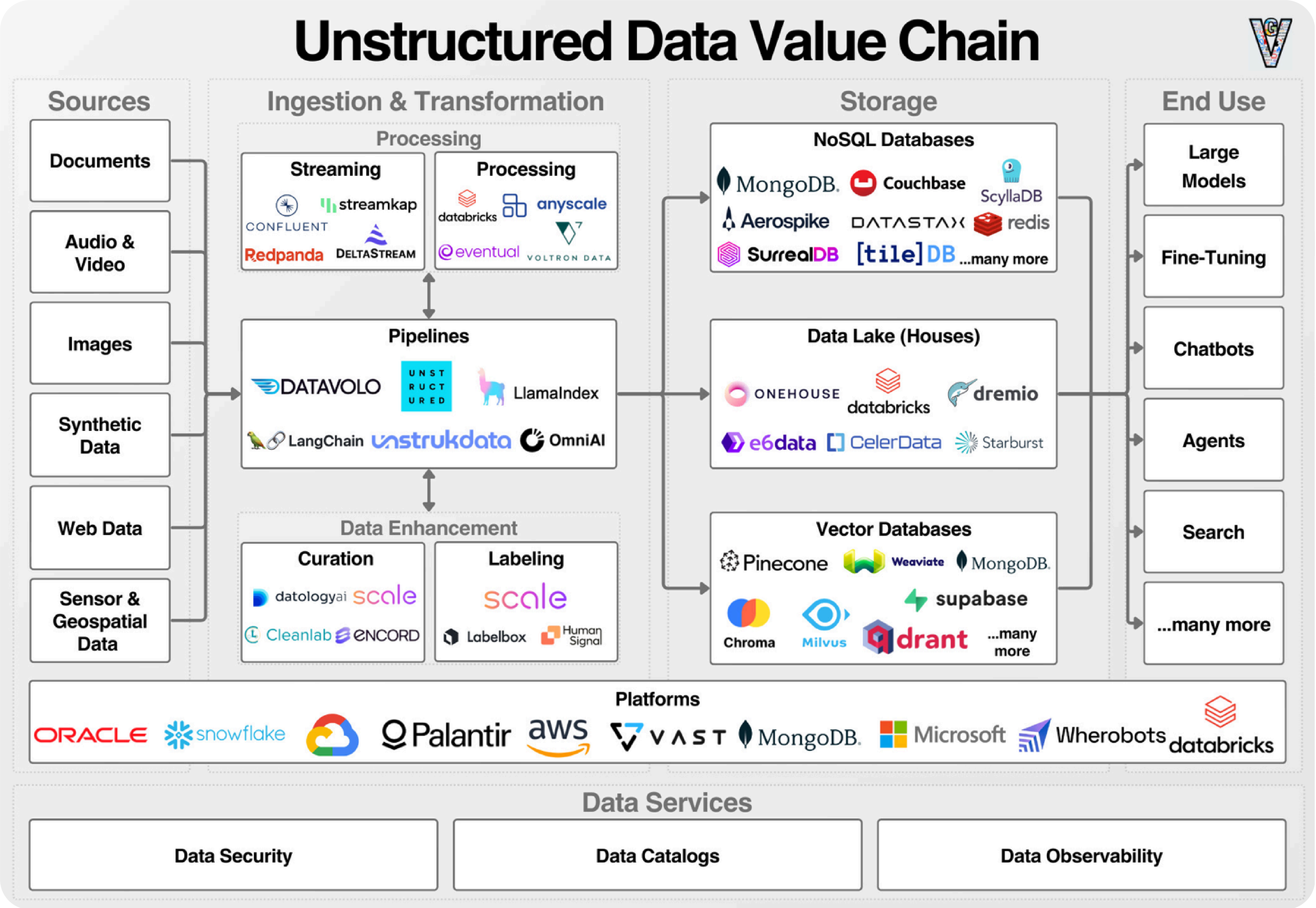
Series 1	Timestamp	Measure Value
Dimension Region: us-west Dimension Host: Server1 Measure name: CPU	2019-02-18 8:01 PST	55%
	2019-02-18 8:02 PST	65%

Series 2	Timestamp	Measure Value
Region: us-east Host: Server2 Measure name: CPU	2019-02-18 8:01 PST	61%

Series 3	Timestamp	Measure Value
Region: us-west-1 Host: Server1 NIC: 011031AB12 Measure name: CPU	2018-04-18 8:01 PST	100

Most popular NoSQL databases

Couchbase
MarkLogic
mongoDB
Neo4j
InfiniteGraph
The Distributed Graph Database
Wide Column Stores
Key-Value Databases
redis
amazon DynamoDB
AEROSPIKE
riak
accumulo
HYPERTABLE INC
Cassandra
APACHE HBASE
Amazon SimpleDB



Unstructured Data Value Chain

Sources

Documents

Audio & Video

Images

Synthetic Data

Web Data

Sensor & Geospatial Data

Ingestion & Transformation

Processing

Streaming: CONFLUENT, streamkap, Redpanda, DELTASTREAM

Processing: databricks, anyscale, eventual, VOLTRON DATA

Pipelines: DATAVOLO, UNSTRUCTURED, LlamaIndex, LangChain, unstrukdata, OmniAI

Data Enhancement

Curation: datologyai, scale, Cleanlab, ENCORD

Labeling: scale, Labelbox, Human Signal

Storage

NoSQL Databases: MongoDB, Couchbase, ScyllaDB, Aerospike, DATASTAX, redis, SurrealDB, [tile]DB, ...many more

Data Lake (Houses): ONEHOUSE, databricks, dremio, e6data, CelerData, Starburst

Vector Databases: Pinecone, Weaviate, MongoDB, Chroma, Milvus, supabase, qdrant, ...many more

End Use

Large Models

Fine-Tuning

Chatbots

Agents

Search

...many more

Platforms: ORACLE, snowflake, Palantir, aws, VAST, MongoDB, Microsoft, Wherobots, databricks

Data Services

Data Security

Data Catalogs

Data Observability

New wave of applications (robotics, autonomy, agents)

Summary

SQL Databases: Structured and Reliable

SQL databases are structured and use a fixed schema, which means the data must fit into a predefined model. They rely on structured query language (SQL) for defining and manipulating data. This structure allows for strong data integrity and ACID compliance, making it suitable for complex queries and transactions.

NoSQL Databases: Flexible and Scalable

NoSQL databases are more flexible, allowing for unstructured or semi-structured data without a predefined schema. They can scale horizontally by distributing data across many servers, making them ideal for handling large volumes of data and allowing for rapid changes. NoSQL databases typically sacrifice some level of consistency for availability and partition tolerance.

Most Popular NoSQL Databases
MongoDB
Cassandra
Redis
Neo4j

Ms. Aigiun (Aya) Guseinova

Lead, Data & Analytics / AI & ML