Lecture 03: Databases and Big data: Persistence, transactions, querying, indexing and SQL.

KD44103 Big Data Analytics
Faculty of Computing & Informatics,
Universiti Malaysia Sabah

UMS
UNIVERSITI MALAYSIA SABAH

Learning outcomes

1. Explain Traditional Data Systems: RDBMS data, such as MySQL DB2
2. Provides hands-on, real world database experience using data from the City of Edmonton Open Data Portal
3. Differentiate the various traditional and Big Data analytics architecture reference model
4. Explain the basic concepts of NOSQL database

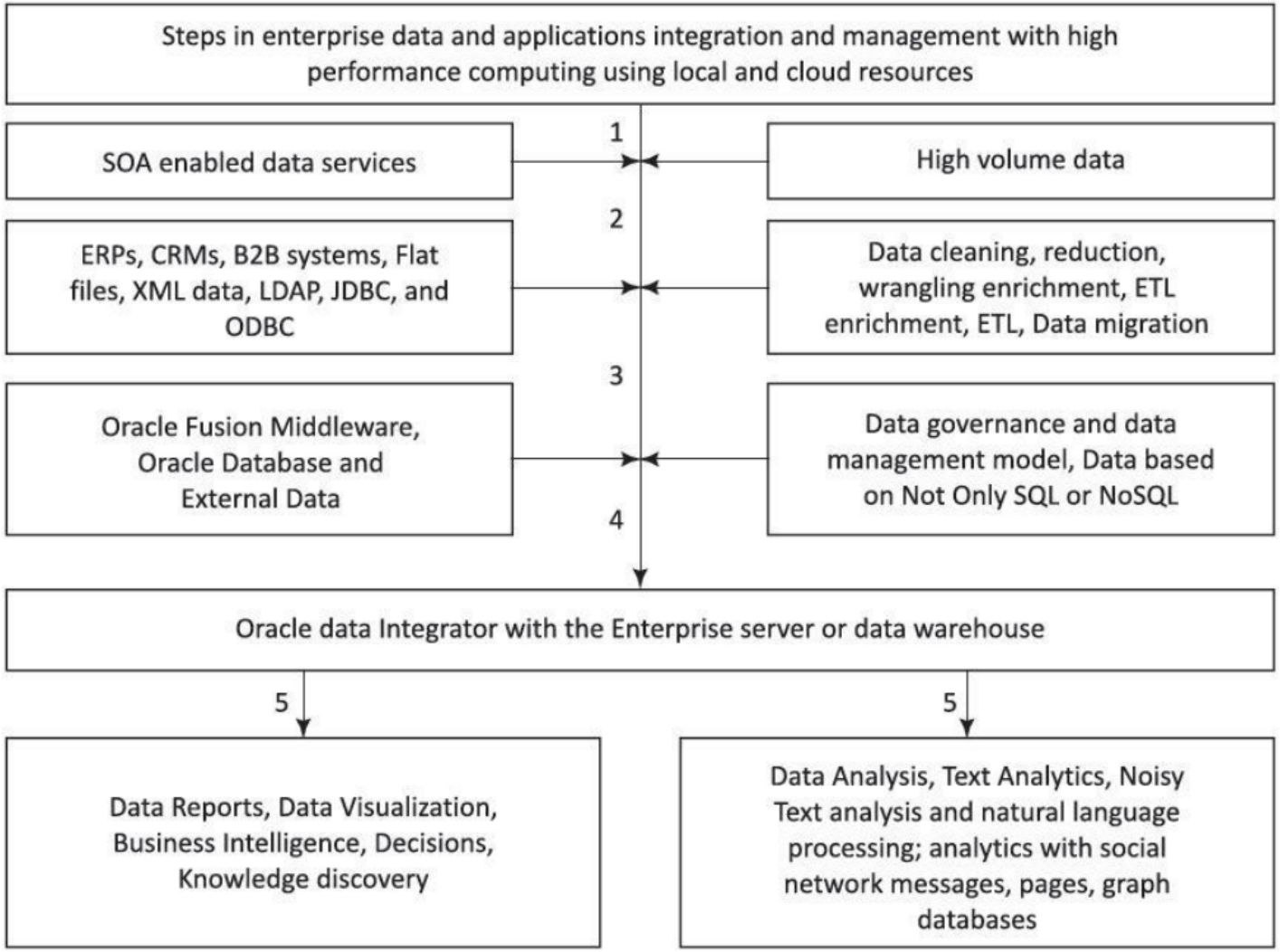
Traditional systems

- Structured Data
- RDBMS data, such as MySQL DB2, enterprise server and data warehouse
- SQL– a language for managing the RDBMS Relational database examples are MySQL PostGreSQL Oracle database, Informix, IBM DB2 and Microsoft SQL server.
- Semi-Structured Data
- XML and JSON
- A comma-separated values (CSV) file

Enterprise Data

- Enterprise Data-Store Server
- Data Warehouse
- Enterprise data warehouse store the databases, and data stores after integration, using tools from number of sources

Enterprise data integration and management with Big-Data for high performance computing



Big Data Platform Data management, storage and Analytics

1. New innovative non-traditional methods of storage, processing and analytics
2. Distributed Data Stores
3. Creating scalable as well as elastic virtualized platform (cloud computing)
4. Huge volume of Data Stores

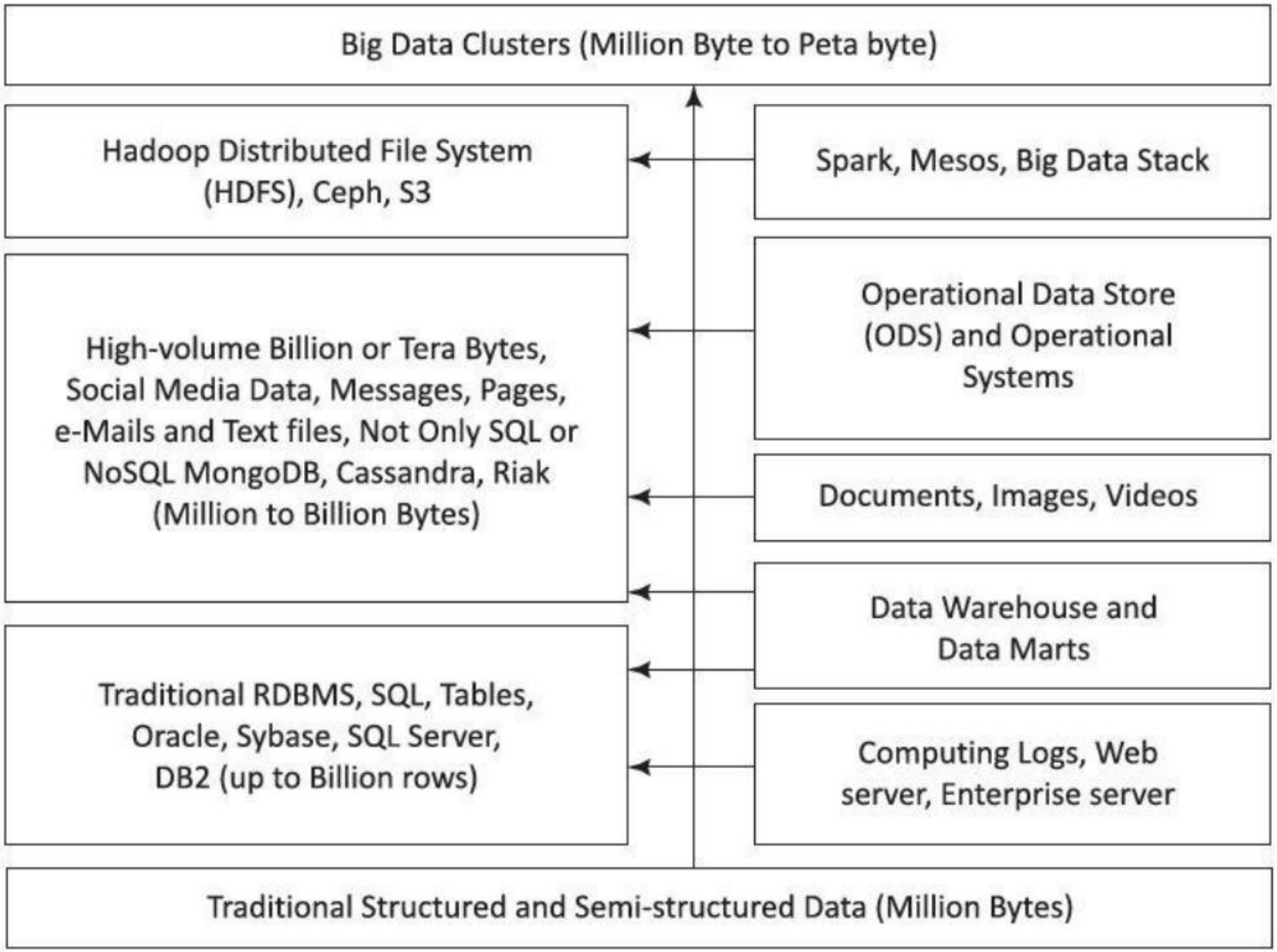
Big Data Platform Data management, storage and Analytics ..cont

5. Massive parallelism
6. High speed networks
7. High performance processing, optimization and tuning
8. 8. Data management model based on Not Only SQL or NoSQL
9. In-memory data column-formats transactions processing or dual in memory data columns as well as row formats for OLAP and OLTP

Big Data Platform Data management, storage and Analytics ..cont

10. Data retrieval, mining, reporting, visualization and analytics
11. Graph databases to enable analytics with social network messages, pages and data analytics
12. Machine learning or other approaches
13. Big data sources: Data storages, data warehouse, Oracle Big Data, MongoDB NoSQL, Cassandra NoSQL
14. Data sources: Sensors, Audit trail of Financial transactions data, external data such as Web, Social Media, weather data, health records data.

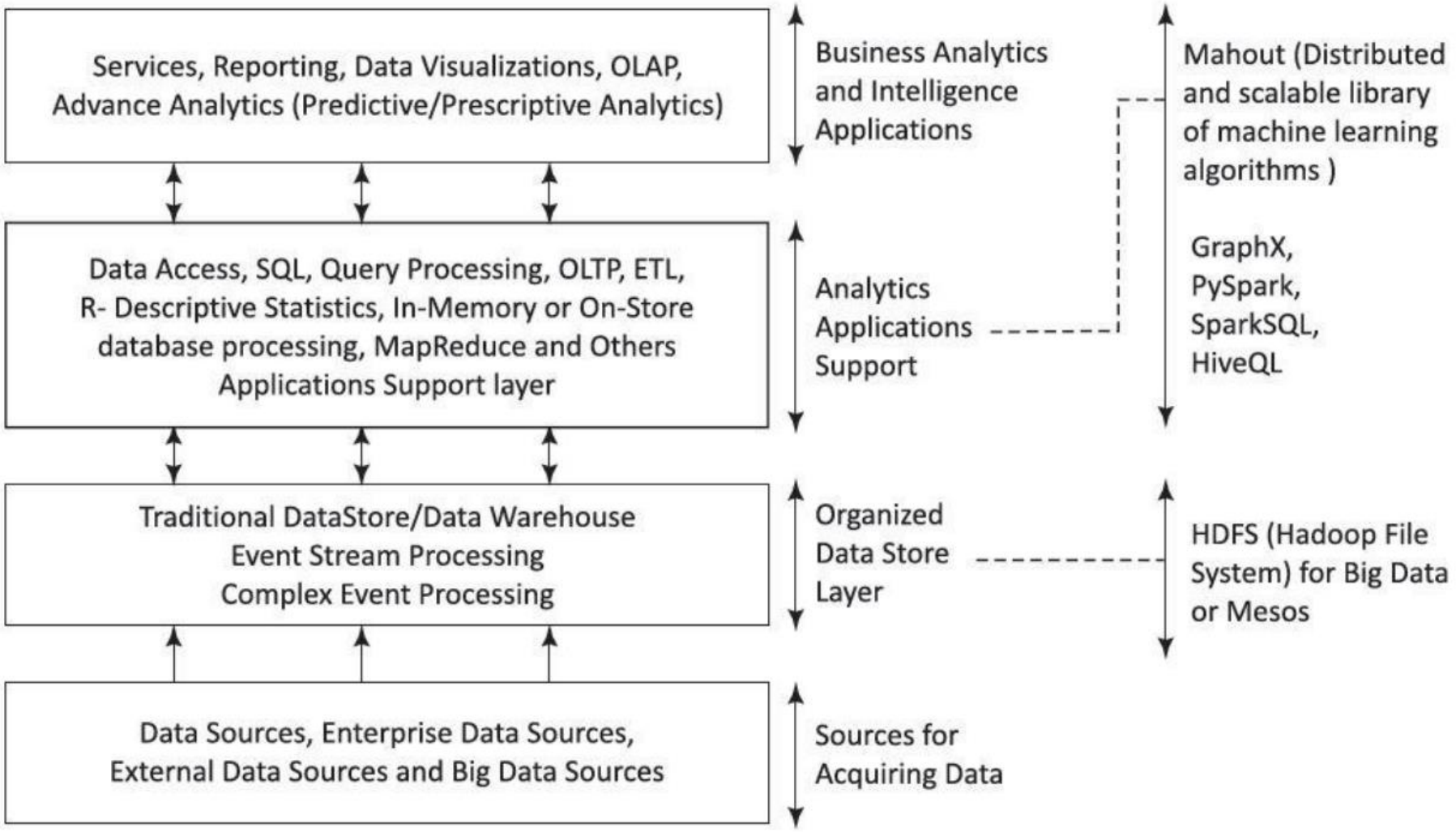
Coexistence of traditional server data, NoSQL and Hadoop, Spark and compatible Big Data Clusters



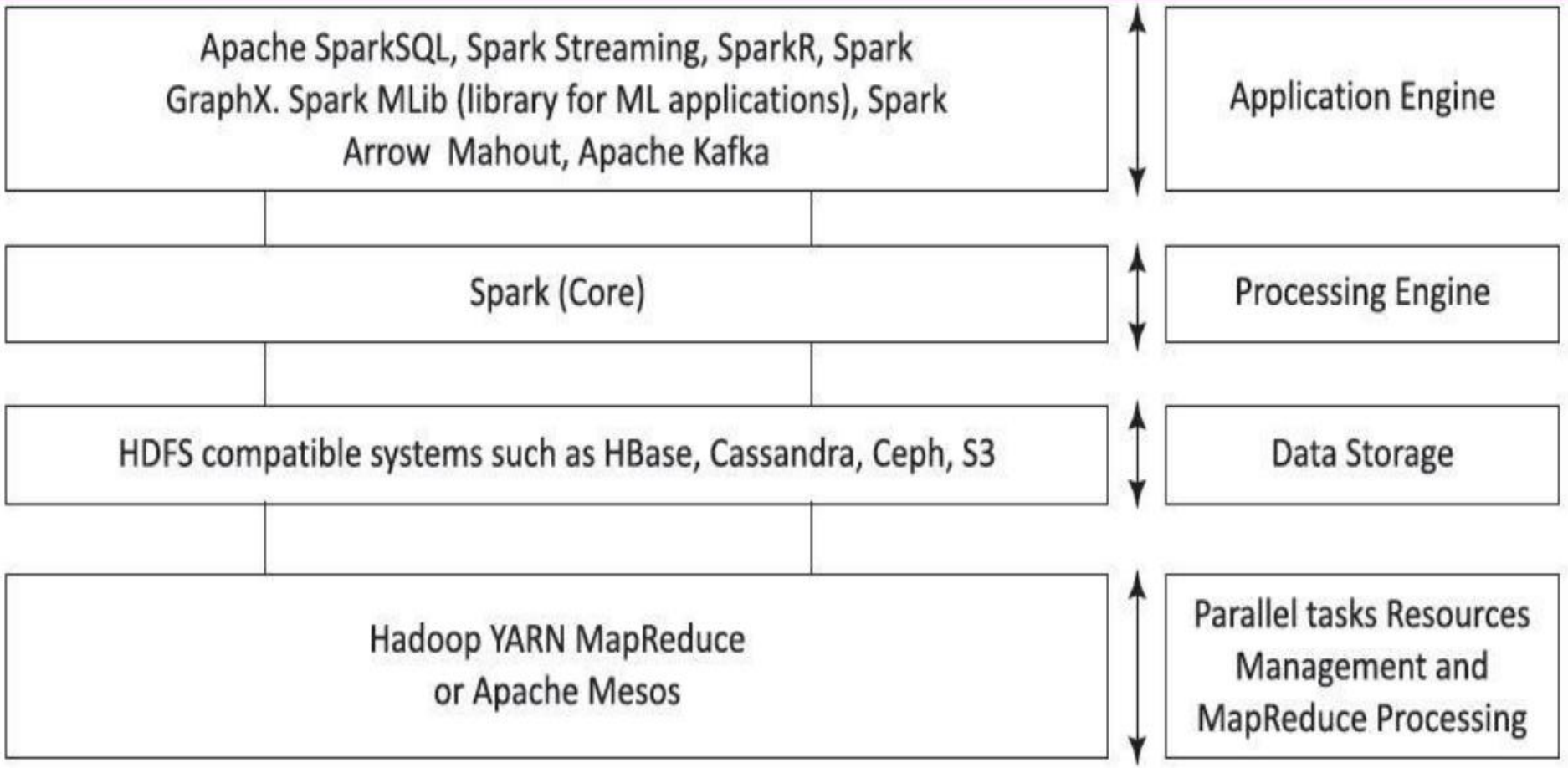
Big Data Stack

- A stack consists of a set of software components and data store units. Applications, machine-learning algorithms, analytics and visualization tools
- Use Big Data Stack (BDS) at a cloud service, such as Amazon EC2, Azure or private cloud. The stack uses cluster of high-performance machines

Traditional and Big Data analytics architecture reference mode



Four layers architecture for Big Data Stack



Big Data Persistence

- **Relational Databases** have been the dominant data persistence technology for many years.
- **object persistence** refers to an object that is not deleted until a need emerges to remove it from the memory. Some database models provide mechanisms for storing persistent data in the form of objects.
- **process persistence**, processes are not killed or shut down by other processes and exist until the user kills them. For example, all of the core processes of a computer system are persistent for enabling the proper functioning of the system.

Why SQL

- Simple
- Accessible
- Applicable
- Powerful
- Pervasive
- Valuable
- Universal

SQL, Python and R

- Difficult for beginners
- Complicated syntax
- Requires programming knowledge (logic, algorithms)
- Is SQL better than Python or R?
    - SQL is good for some things
    - Python/R is good for other things
    - Compliment each other
- SQL is a great starting point

Relational Database

- What is a database?
- A relational “database” management system (RDBMS) organizes data
- The logical structure of the database is based upon the information needs of an organization

- Entities (“things” of interest to the organization),

AND

- Relationships (how the Entities are associated with each other)

Advantages of a RDBMS

- Establish a centralized, logical view of data
- Minimizes data duplication (i.e. “redundancy”)
- Promote data accuracy and integrity
- Capacity of database
- Superior multi-user or concurrent access
- Security
- Retrieve information quickly
- Inter-operability

https://www.bespokesoftwaredevelopment.com/blog/advantages-database-development-business/

How to introduce SQL

- Microsoft Access
- Microsoft SQL Server
- MariaDB, MySQL
- Postgresql
- Oracle
- Hadoop, Spark, Hive, Pig

The components of SQL

- As a language, SQL has two main components:

1. a data definition language (DDL) for defining the database structure, implementing the data structure and constraints components of the relational data model

2. a data manipulation language (DML) for CRUD operations on the database, implementing the operations component of the relational data model.

3. a data control language (DCL) for controlling user access to database objects and their contents.

SQL DDL

- SQL is a ‘declarative language’
- A ‘declarative language’ specifies what should be accomplished.
- An ‘imperative language’ such as Java specifies how to accomplish it.
- Example, the following SQL requests to ‘display patients’ details’:

```
SELECT * FROM patient
```

CRUD operations: SQL DML

- In this section you will learn more about how tables can be manipulated using SQL to create, retrieve, update and delete data (CRUD operations).

After completing this section, you should be able to:

- use the SQL INSERT, UPDATE and DELETE statements to add, update and delete rows in tables respectively
- use the SQL SELECT statement to answer queries about the data stored in a single database table
- explain the function and the order of processing of the SQL SELECT statement.

SQL DCL

- Database administrators use DCL to configure security access to relational databases.
- DCL is the simplest of the SQL subsets, as it consists of only three commands: GRANT, REVOKE, and DENY. Combined, these three commands provide administrators with the flexibility to set and remove database permissions in granular fashion.

SQL SELECT statement

- The general form of the SELECT statement is:

```
SELECT [DISTINCT] * | <column list>
FROM <table list>
[WHERE <condition>]
[GROUP BY <column list> [HAVING
<condition>]]
[ORDER BY <column list> [ASC|DESC]]
[LIMIT <number>]
```

The closed operations of relational algebra & SQL

- Both relational algebra operations and the SQL SELECT operation are closed.
- That is, the result of relational algebra operations on relations is another relation, and the result of the SQL SELECT operation on tables is another table.
- The resultant relation or table must adhere to properties of relations or tables respectively.

SQL SELECT statement processing

- The order of processing of SELECT statement clauses is as follows:

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY
7. LIMIT

Coping with missing or invalid data elements

- We have to be careful when handling columns that can take null
- Any expression or condition that includes a null will evaluate to null (meaning ‘unknown’).
- When a WHERE (or HAVING) clause is processed, if any of the columns referenced by the condition are null for a row, then that row is not selected.
- the functions IS NULL and IS NOT NULL check whether a column (or expression) is null or otherwise.
- Example: the following SQL will ‘display the details of patients whose weight has not been recorded’.

```
- SELECT *
  FROM patient
  WHERE weight IS NULL;
```

Aggregation

- SQL aggregate functions compute a single value from a set of values.
- The following example calculates some summary statistics about the contents of the patient table:

```
- SELECT COUNT(*),
  COUNT(DISTINCT patient_name),
  COUNT(weight),
  MIN(weight),
  MAX(weight),
  CAST(AVG(weight) AS DECIMAL(4,1))
  FROM patient
```

Views

- A view is the resultant table from the execution of an SQL SELECT query on a database
- A view may be used in any SQL statement where a table is expected, for example, on the FROM clause of a SELECT statement.

Example

Let say, we have patients Table which contains patients' information

- CREATE VIEW female_patients
  AS SELECT *
  FROM patient
  WHERE gender = 'F';

- SELECT * FROM female_patients;

What is the results?

Types of views

- There are two types of view: views and materialized views:
- A **view** is a virtual table that results from the execution of the **SELECT** query that defines the view. The virtual table only exists (is materialized) when any SQL statement that references the view is executed.

- A **materialized view** is a physical table that is created when the view is defined using the **CREATE MATERIALISED VIEW** statement and is updated (refreshed) using the REFRESH MATERIALISED VIEW statement.

SQL VS NoSQL

- SQL databases are vertically scalable, while NoSQL databases are horizontally scalable.
- SQL databases are table-based, while NoSQL databases are document, key-value, graph, or wide-column stores.
- SQL databases are better for multi-row transactions, while NoSQL is better for unstructured data like documents or JSON.

NOSQL and Big data storage system

- The data storage block in the big data stack includes distributed filesystems and non-relational (NoSQL) databases, which store the data collected from the raw data sources using the data access connectors.
- For certain analytics applications, it is preferable to store data in a NoSQL database

Some frequently used NoSQL databases

- 1- Key-value databases
- 2- Document Databases
- 3- Column Family Databases
- 4- Graph Databases

Key-Value Databases

- Key-value databases are the simplest form of NoSQL databases. These databases store data in the form of key-value pairs. The keys are used to identify uniquely the values stored in the database. Applications that want to store data generate unique keys and submit the key-value pairs to the database. The database uses the key to determine where the value should be stored.

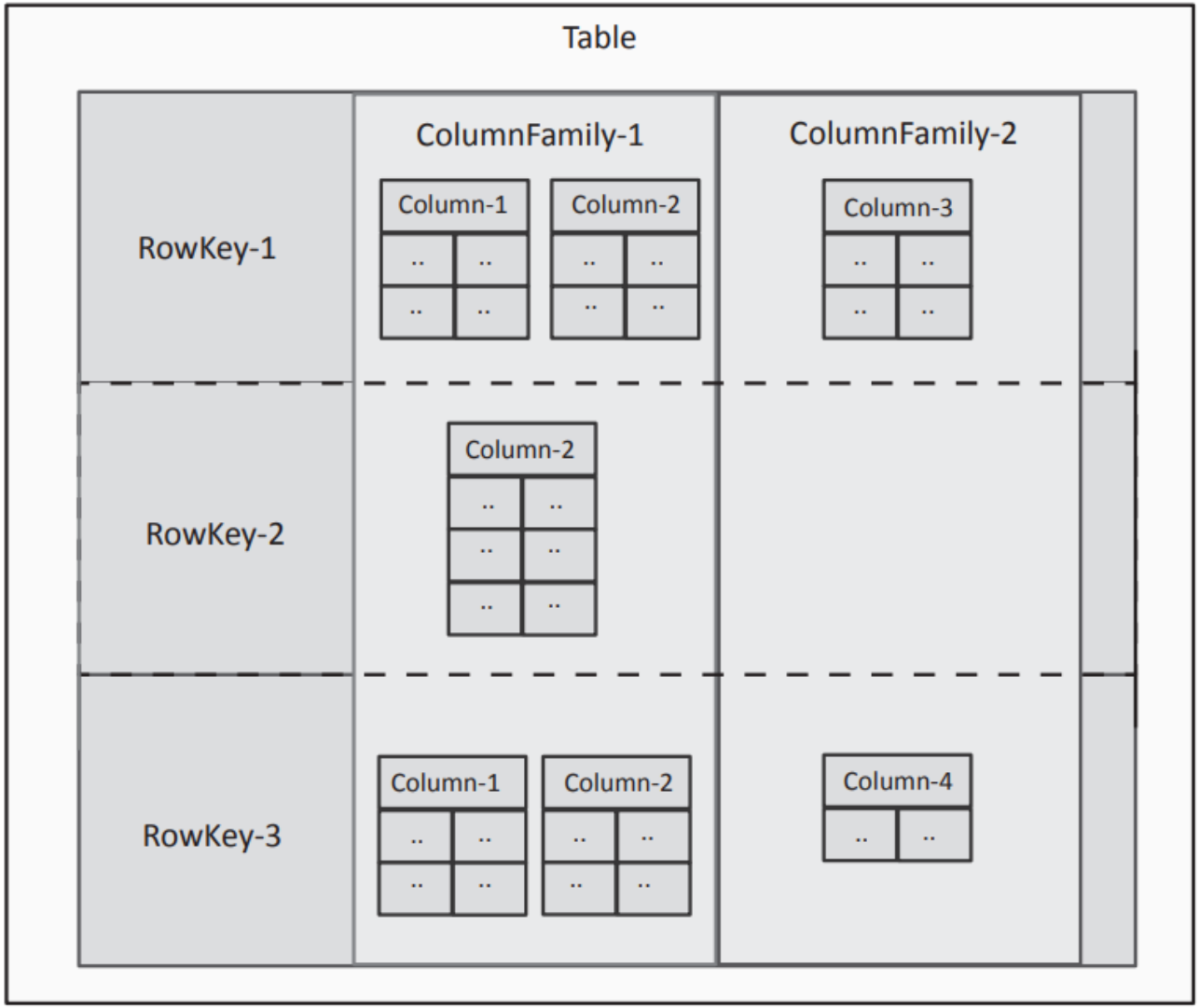
Document Databases

- Document store databases store semi-structured data in the form of documents which are encoded in different standards such as JSON, XML, BSON or YAML.
- By semi-structured data we mean that the documents stored are similar to each other (similar fields, keys or attributes) but there are no strict requirements for a schema.
- Documents are organized in different ways in different document database such in the form of collections, buckets or tags.

Column Family Databases

- In column family databases the basic unit of data storage is a column, which has a name and a value. A collection of columns make up a row which is identified by a row-key. Columns are grouped together into columns families. Unlike, relational databases, the column family databases do not need to have fixed schemas and a fixed number of columns in each row. The number of columns in a column family database can vary across different rows. E.g HBase

HBase table structure



Souce: Bahga, Arshdeep, and Vijay Madisetti. "Big Da

Data Storage & Operation in HBase

Each Region Server has a Block cache, which is an in-memory store that caches the most recently used blocks for fast lookup.

HBase supports the following operations:

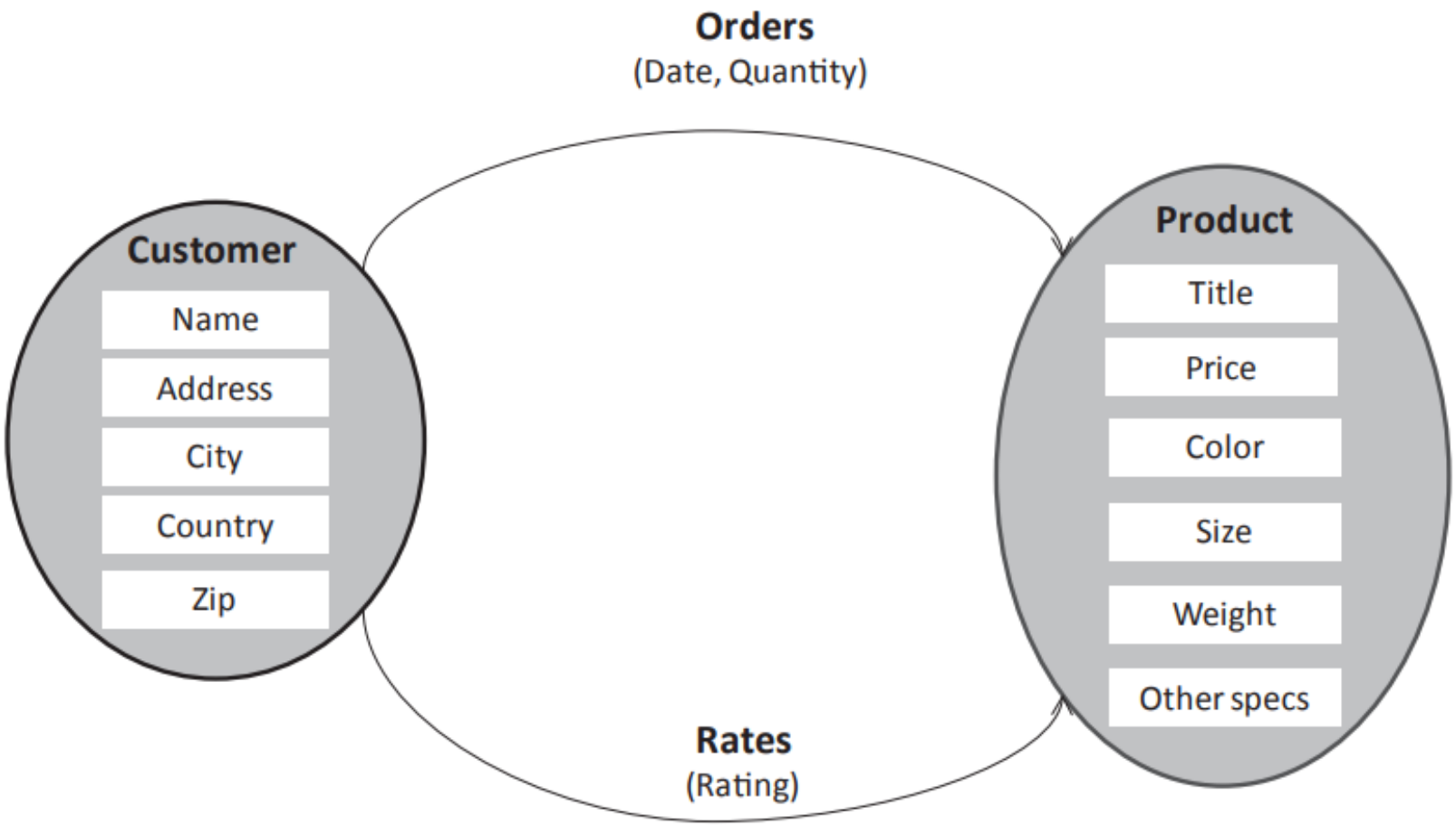
- Get: Get operation is used to return values for a given row key.
- Scan: Scan operation returns values for a range of row keys.
- Put: Put operation is used to add a new entry.
- Delete: Delete operation adds a special marker called Tombstone to an entry.

Graph Databases

- Graph stores are NoSQL databases designed for storing data that has graph structure with nodes and edges.
- The relationships between the entities are represented in the form of links between the nodes.
- E.g. Neo4j

Neo4j

- Neo4j is one the popular graph databases which provides support for Atomicity, Consistency, Isolation, Durability (ACID).
- Neo4j adopts a graph model that consists of nodes and relationships. Both nodes and relationships have properties which are captured in the form of multiple attributes (key-value pairs). Nodes are tagged with labels which are used to represent different roles in the domain being modeled.



Labeled property graph example

Souce: Bahga, Arshdeep, and Vijay Madisetti. "Big Data analytics: A hands-on approach." (2020).

Comparison of NoSQL databases

	Key-Value DB	Document DB	Column Family DB	Graph DB
Data model	Key-value pairs uniquely identified by keys	Documents (having key-value pairs) uniquely identified by document IDs	Columns having names and values, grouped into column families	Graphs comprising of nodes and relationships
Querying	Query items by key, Database specific APIs	Query documents by document-ID, Database specific APIs	Query rows by key, Database specific APIs	Graph query language such as Cypher, Database specific APIs
Use	Applications involving frequent small reads and writes with simple data models	Applications involving data in the form of documents encoded in formats such as JSON or XML, documents can have varying number of attributes	Applications involving large volumes of data, high throughput reads and writes, high availability requirements	Applications involving data on entities and relationships between the entities, spatial data
Examples	DynamoDB, Cassandra	MongoDB, CouchDB	HBase, Google BigTable	Neo4j, AllegroGraph

Souce: Bahga, Arshdeep, and Vijay Madisetti. "Big Data analytics: A hands-on approach." (2020).

The End