Cloud Database & Analytics Services

IT4090 – Cloud Computing

Database Models

Relational/SQL

- Highly structured table organization
- Rigidly-defined formats
- Dependencies among tables
- Enforce ACID (Atomicity, Consistency, Isolation, Durability)
- Reduces anomalies, enforces integrity
- Use SQL to access data
- Examples – MS SQL, MySQL, Oracle, PostgreSQL, Amazon RDS

Non-relational/No-SQL

- Document oriented
- Large and complex queries
- Supports rapidly changing designs
- Examples – MongoDB, Cassandra, CosmosDB, Redis, CouchDB, Aurora

Database Workloads

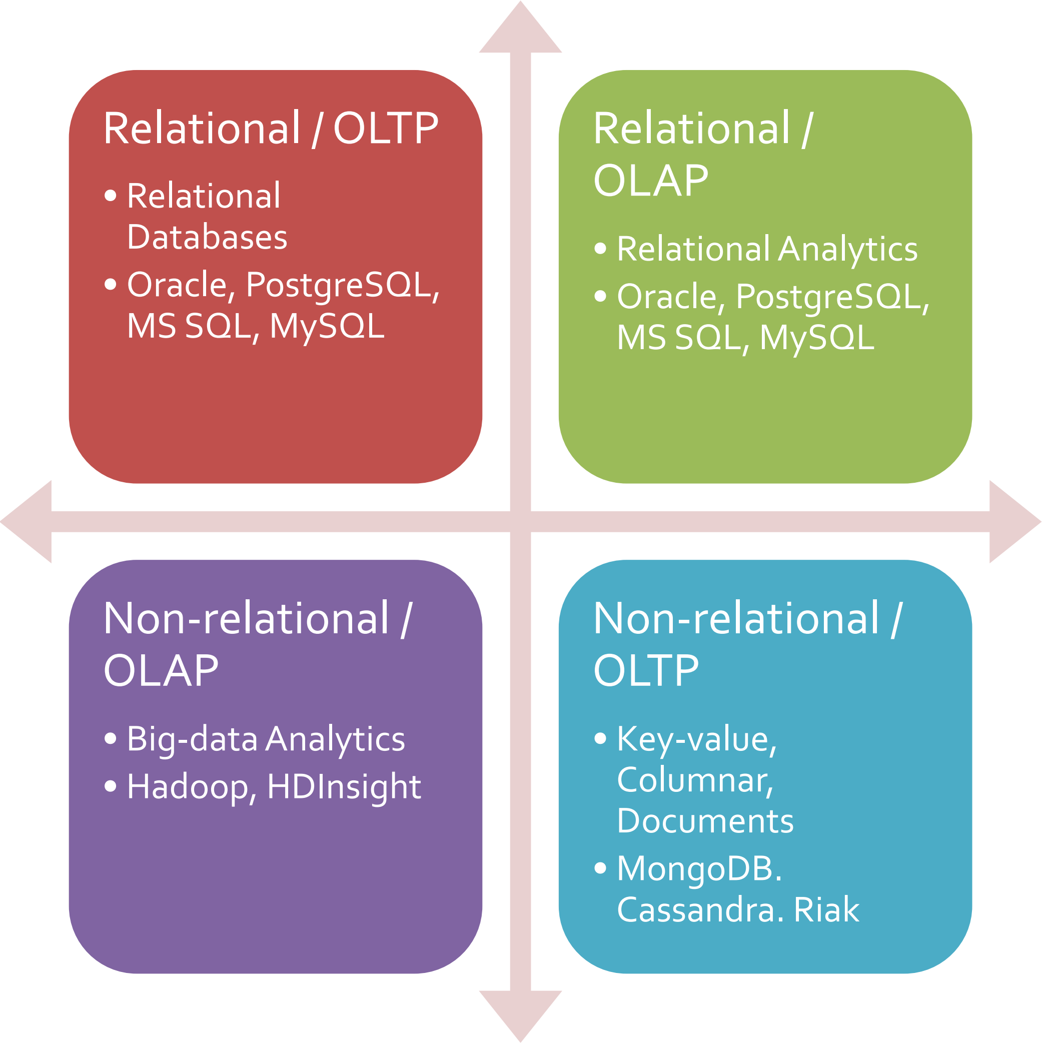
Online Transaction Processing (OLTP)

- Focus is on operational data
- Transaction processing
- Small, simple ad-hoc queries
- Response in milliseconds
- Highly normalized

Online Analytical Processing (OLAP)

- Focus is on historical data
- Data analysis and reporting
- Large, complex queries
- Data warehouses
- Responses times from seconds to hours
- Typically denormalized

Database Models & Workloads



Relational / OLTP

- Relational Databases
- Oracle, PostgreSQL, MS SQL, MySQL

Relational / OLAP

- Relational Analytics
- Oracle, PostgreSQL, MS SQL, MySQL

Non-relational / OLAP

- Big-data Analytics
- Hadoop, HDInsight

Non-relational / OLTP

- Key-value, Columnar, Documents
- MongoDB. Cassandra. Riak

Non-relational / No-SQL Databases

Key-Value

- Indexed keys and values
- Use case – session data, shopping cart

Document Store

- Store data in documents (XML, JSON etc.)
- Schema less
- Documents contain key-value pairs
- Use case – E-commerce, analytics

Columnar

- Optimized to retrieve columns of data
- Use case – CMS, Blogging platforms

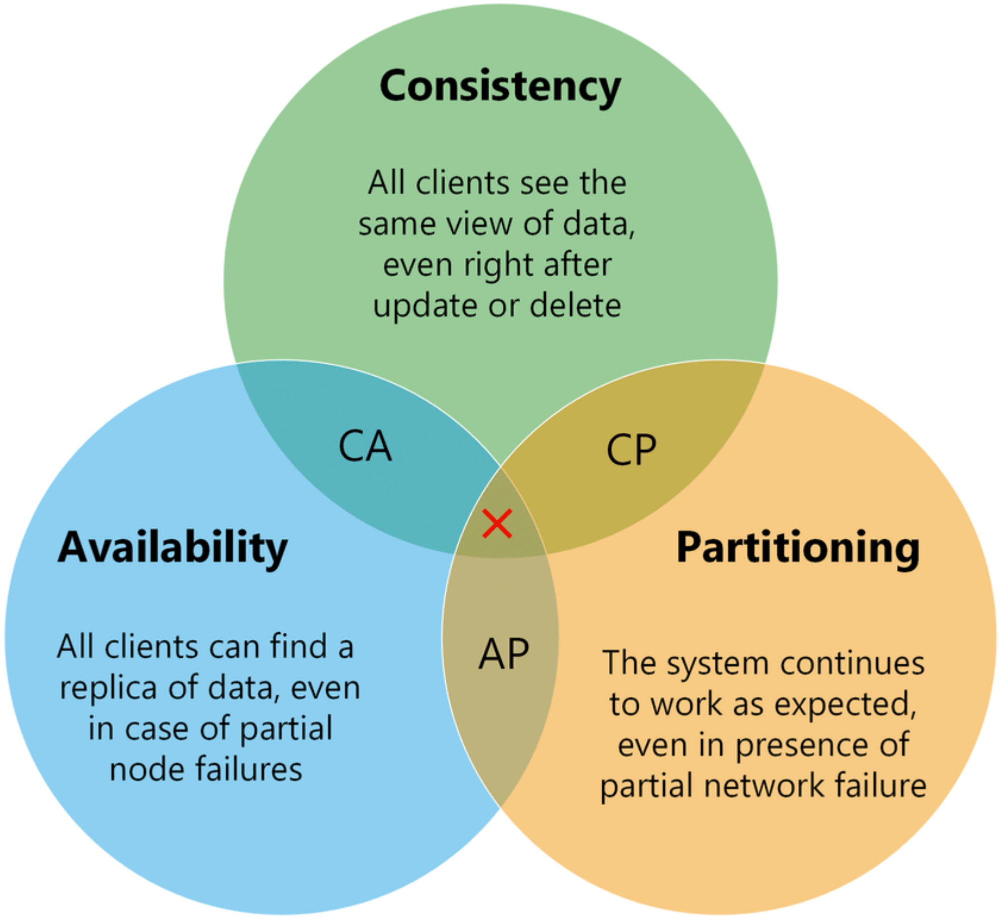
Graph

- Presents interconnected data as logical graphs
- Focus on relationships

Non-relational / No-SQL Databases Comparison

TYPES	FLEXIBILITY	COMPLEXITY	PERFORMANCE	SCALABILITY
KEY-VALUE STORE	High	None	High	High
COLUMN SOTRE	Moderate	Low	High	High
DOCUMENT	High	Low	High	Variable (High)
GRAPH DB	High	High	Variable	Variable

CAP Theorem



Consistency

All clients see the same view of data, even right after update or delete

Availability

All clients can find a replica of data, even in case of partial node failures

Partitioning

The system continues to work as expected, even in presence of partial network failure

Database Caching

Caching is a buffering technique that stores frequently requested data in temporary memory. Facilitates data access and reduces database workloads.

Two popular caching systems

- Redis
- Memcached

Redis vs Memcached

Redis

- Open source, in-memory, key-value data store
- Sub-millisecond response times
- Supports various data structures (strings, lists, sets etc.)
- Persistent – cache survives reboots
- Supports read replicas, atomic operations, backup/restore, HA

Memcached

- Open source, in-memory, object store
- Sub-millisecond response times
- Supports strings and objects
- Not persistent – cache does not survive reboots
- Supports scaling out, multithreading

Data Warehouse

A data warehouse is a type of data management system designed to enable and support business intelligence (BI) activities, especially analytics.

Data warehouses are intended for querying and analysis only and often contain large amounts of historical data.

A repository for structured, filtered data that has already been processed for a specific purpose

Data in a data warehouse is typically derived from a wide variety of sources such as application log files and transaction applications.

Two approaches

- ETL – Extract, Transform, Load (Source -> Staging -> Destination)
- ELT – Extract, Load, Transform (Source -> Destination)

Data Lake

A data lake is a storage repository that holds a large amount of raw data in its native format until it is needed.

While a hierarchical data warehouse stores data in files or folders, a data lake uses a flat architecture to store the data.

Data Warehouse vs Data Lake

Data Warehouse

- Processed data
- Data currently in use
- Used by business professionals

Data Lake

- Raw data
- Purpose of data not determined yet
- Used by data scientists

Real Time Data Processing

Real-time data processing is the quickest data processing technique that executes data in a short period of time and provides the most accurate output.

The processing is done as the data is inputted, so it needs a continuous stream of input data in order to provide a continuous output.

Also known as stream processing.