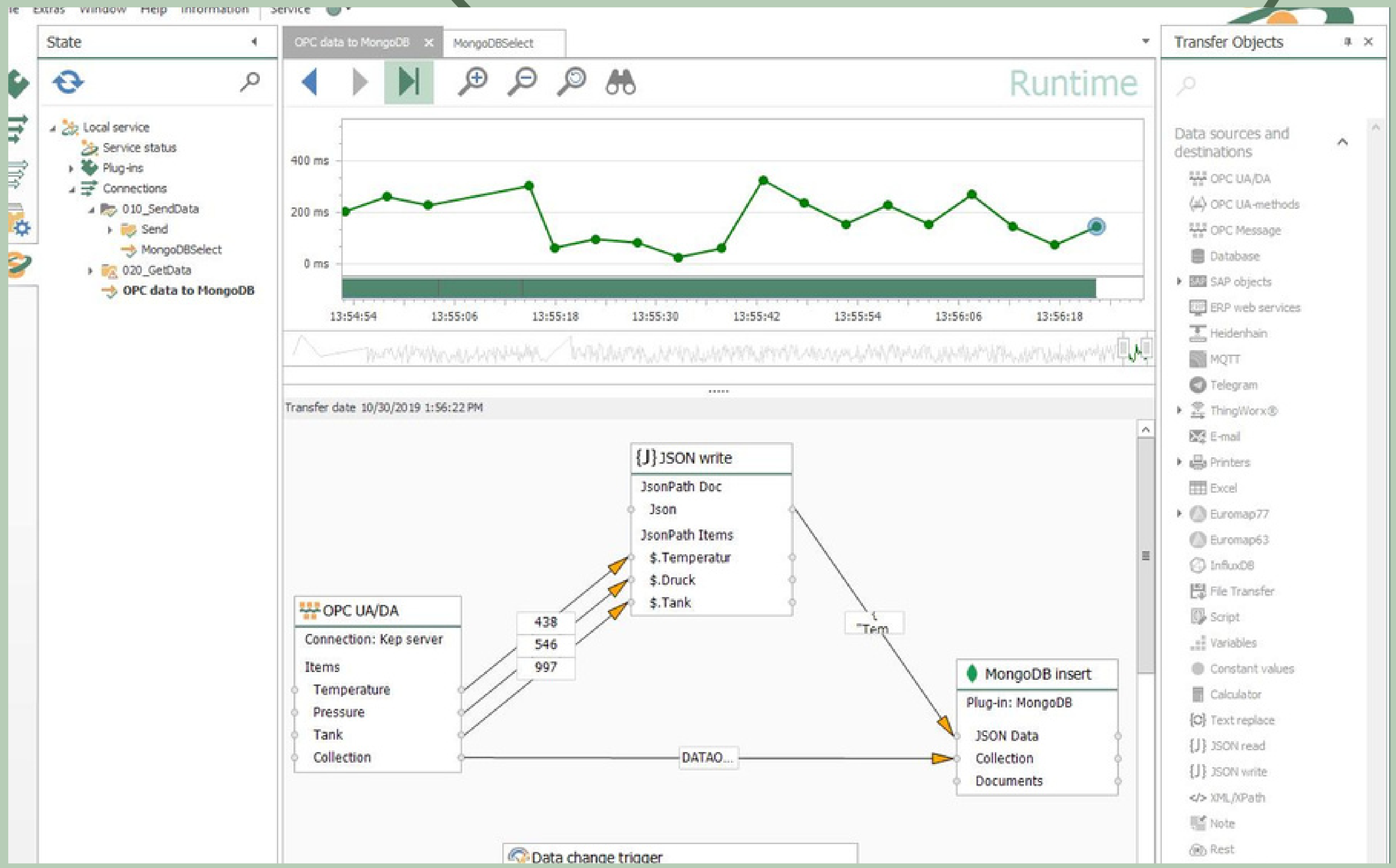
NoSQL to SQL

presented by :Hana Ben Amor

Introduction to MongoDB

MongoDB is a very popular open-source Document Database that operates as a NoSQL database. It is popularly used in collaboration with AWS, Azure, and many other data sources for application development and functioning. Allowing the storage and querying of high volumes of data, it offers the following robust features:

- Better query executions with proper indexing and processing features.
- Real-time analytics and optimized data handling with the use of ad-hoc queries.
- Improved data availability and flexibility with robust replication features.
- Data Sharding allows the splitting of large data chunks for a distributed and faster query execution process.



Transfer date 10/30/2019 1:56:22 PM

Introduction to SQL Databases

SQL Databases are typically Relational Database services that are popularly used for the deployment of Cloud applications. Along with high-performance analytics, they offer various features for accessing, appending, managing, and processing data within the database.

A SQL Database stands out owing to its ease of use, robust classification features, and unhindered reliability as a Database System.

Some prominent SQL Database features are as follows:

- On-demand scalability along with comprehensive and reliable transactional support for businesses.
- Increased workflow control and easy analysis of structured data in a flexible open-source environment.
- It is free to download and implements a straightforward client-server architecture for operation.
- It offers a solid data security layer that helps maintain the integrity of data while also protecting sensitive data from infiltration.

SQL VS MongoDB

- MySQL uses a structured query language (SQL) for entry, like many relational databases. The MongoDB Query Language (MQL), developed for efficient use by developers, is used by MongoDB.
- SQL supports structured data in form of tables whereas MongoDB is used to store unstructured data in BSON format.
- MongoDB supports various drivers like DART, GO, JAVA, JAVASCRIPT, etc. whereas SQL supports .NET, PHP, Python, JAVA, etc.
- The schemas in SQL DBs are fixed whereas in MongoDB it is flexible. (A schema database is the skeleton structure that represents the logical view of entire database.)
- Queries in SQL are very much slower as compared to NoSQL DBs like MongoDB because queries for a single data or object might require joins to take place which could be very expensive with large data in DBs .
- As MongoDB documents map naturally to modern, object-oriented programming languages, development is simplified.
- Since joins are not possible in MongoDB, hence the concept of foreign keys is also not there.
- MongoDB servers are compatible with windows, Linux and mac whereas SQL servers are only compatible with windows machines.
- MongoDB is open source and SQL DB is commercial.

SQL DATABASE

PROS(+)

- Easy to write- In SQL the basic keywords are SELECT, INSERT, INTO, UPDATE which are pretty self-explanatory and direct and also, the syntactical rules are not complex in SQL.
- Large amount of data is retrieved quickly and efficiently.
- It has a table-based schema
- A Mature technology, known by a lot of developers and used regularly.
- Great support and user experience.

CONS(-)

- Although it makes retrieving data easy even for complex queries, sometimes it can be restrictive.
- Changing the structure can create a lot of confusion
- Requires predefined schemas
- Difficult to Scale

MONGODB

PROS(+)

- Has a dynamic schema
- Data stored in many ways such as document-oriented, column-oriented, etc.
- Flexibility- documents can be created without having a defined structure.
- Large scalability making it more powerful
- Horizontally scalable
- High speed

CONS(-)

- A newer database, a lot of unanswered questions
- No standard schema definition
- No join (A JOIN clause is used to combine rows from two or more tables, based on a related column between them)
- Memory limitation- MongoDB needs more storage than other popular databases. Although the introduction of WiredTiger in latest versions has solved this issue, It is not ideal in most applications.
- MongoDB only supports ACID transactions for a single document.(ACID is a set of properties of database transactions intended to guarantee data validity despite errors and other mishaps.)
- In RDBMSs, We have the luxury of triggers, which has saved us in many cases. This luxury is missing in MongoDB.