

Case Study



UBS

Data Lineage Tool Improves Risk Management, Drives Compliance

INDUSTRY

Financial Services

USE CASE

Compliance/Data Lineage

GOAL

Comply with BCBS 239 and gain deep visibility into data lineage

CHALLENGE

Oracle database was not suited to storing, querying and visualizing data lineage

SOLUTION

Built a data lineage and data governance solution using Neo4j that syncs with Oracle

RESULTS

- Easier and faster compliance, with real-time queries and visualizations
- Deep understanding of data flows for impact analysis and governance

The UBS Group, one of the largest financial institutions in the world, built a data lineage and data governance platform using Neo4j. The company generated real-time data lineage to comply with banking industry standards for risk management and risk reporting.

The Company

[UBS](#) is a multinational investment bank and financial services company based in Switzerland. Founded in 1862, it is the largest player in the famous Swiss banking industry. UBS maintains offices in more than 50 countries, employs more than 66,000 people, and reported total assets of more than \$958 billion in 2018. UBS operates in five main business areas: personal banking, wealth management, corporate and institutional clients, investment banking and asset management.

The Challenge

UBS needed to comply with regulations put in place to strengthen systems for risk data aggregation and internal risk reporting in the wake of the 2007 global financial crisis. Specifically, UBS sought compliance with the Basel Committee on Banking Supervision issued standard 239 ([BCBS 239](#)).

Under this regulation, banks need to provide transparency into the data flows that feed their risk reporting. This requires broad data governance and detailed data lineage.

Data lineage is an essential component of risk management. Data lineage involves tracking the entire lifecycle of information – its origin, evolution and movement through the organization. With data lineage, organizations can track information as it flows through the enterprise, monitor its quality, discover errors and trace them to the source, minimize damage and reduce data duplication.

UBS built an application called Group Data Dictionary (GDD) as its data lineage and data governance tool.

The first iteration was built on Oracle, but UBS soon discovered limitations with an RDBMS approach, which relies on JOINS to connect data across tables. UBS decided it needed a better solution suited to creating real-time data lineage visualizations and exporting lineage information for ad-hoc analysis via Excel.

Case Study



“Cypher allowed us to much more easily traverse connected data, especially compared to PL/SQL, which relies on JOINS across multiple tables. Cypher and Neo4j are a much more natural fit for the work we’re trying to do.”

– Sidharth Goyal,
Senior Software engineer, UBS

The Solution

Data lineage is a problem that is best solved using connected data.

“Data lineage is a series of highly connected data, and is more naturally persisted in a graph database,” explained Sidharth Goyal, a senior software engineer and technical lead at UBS.

[Neo4j](#) offered several advantages over a relational database, including querying using Neo4j’s [Cypher](#) query language.

“Cypher allowed us to much more easily traverse connected data, especially compared to PL/SQL, which relies on JOINS across multiple tables to generate the lineage in a relational database format, add a processing layer to format this as an object and then visualize it. Cypher and Neo4j are a much more natural fit for the work we’re trying to do,” said Goyal.

The new data lineage and data governance tool would need to smoothly integrate with the legacy system. All UBS workflows and auditing capabilities remained on Oracle, so synchronization was essential.

UBS synchronized Neo4j with the Oracle system, starting with an initial data load and then performing an incremental sync in which transactions were read from an Oracle table and written into Neo4j in real time.

UBS used Neo4j to evaluate data lineages and depict the results in GraphJSON. This information flows into a D3.js visualizer to render the data as a lineage diagram.

Having all the metadata makes for easy reporting. The data can be used for ad hoc reporting when specific questions arise, and entire lineages can be exported to Excel.

The Results

UBS was able to meet regulatory requirements within a short window and limited resources.

Neo4j allows UBS to trace lineage of all metrics for all initiatives across the bank. What’s unique about the solution is the combination of real-time response time and agility. Lineage spans dozens of levels of entities and dependencies, and resolving these kinds of dependencies using traditional technologies would be impossible in real time.

When generating a data lineage, the company no longer faces the headaches involved with traversing multiple tables in a relational database. With Neo4j, results are obtained easily and displayed in an intuitive graph format.

Business analysis requirements are constantly changing. Neo4j does not rely heavily on predefined schema or indexes, and is thus able to keep pace with the changes. Thanks to synchronization, the data lineage tool is always up-to-date with the enterprise system.

“[Neo4j] helps us understand the flow of data through the organization,” said Goyal. “It helps us understand how changes in one application are going to impact the entire organization. It helps us understand how errors can propagate through the system.”

Neo4j is the leader in graph database technology. As the world’s most widely deployed graph database, we help global brands – including [Comcast](#), [NASA](#), [UBS](#), and [Volvo Cars](#) – to reveal and predict how people, processes and systems are interrelated.

Using this relationships-first approach, applications built with Neo4j tackle connected data challenges such as [analytics and artificial intelligence](#), [fraud detection](#), [real-time recommendations](#), and [knowledge graphs](#). Find out more at [neo4j.com](#).

Questions about Neo4j?

Contact us across the globe:
info@neo4j.com
neo4j.com/contact-us