labbit

RDF Knowledge Graph Databases

A Better Choice for Life Science Lab Software

BY EBAN TOMLINSON AT LABBIT

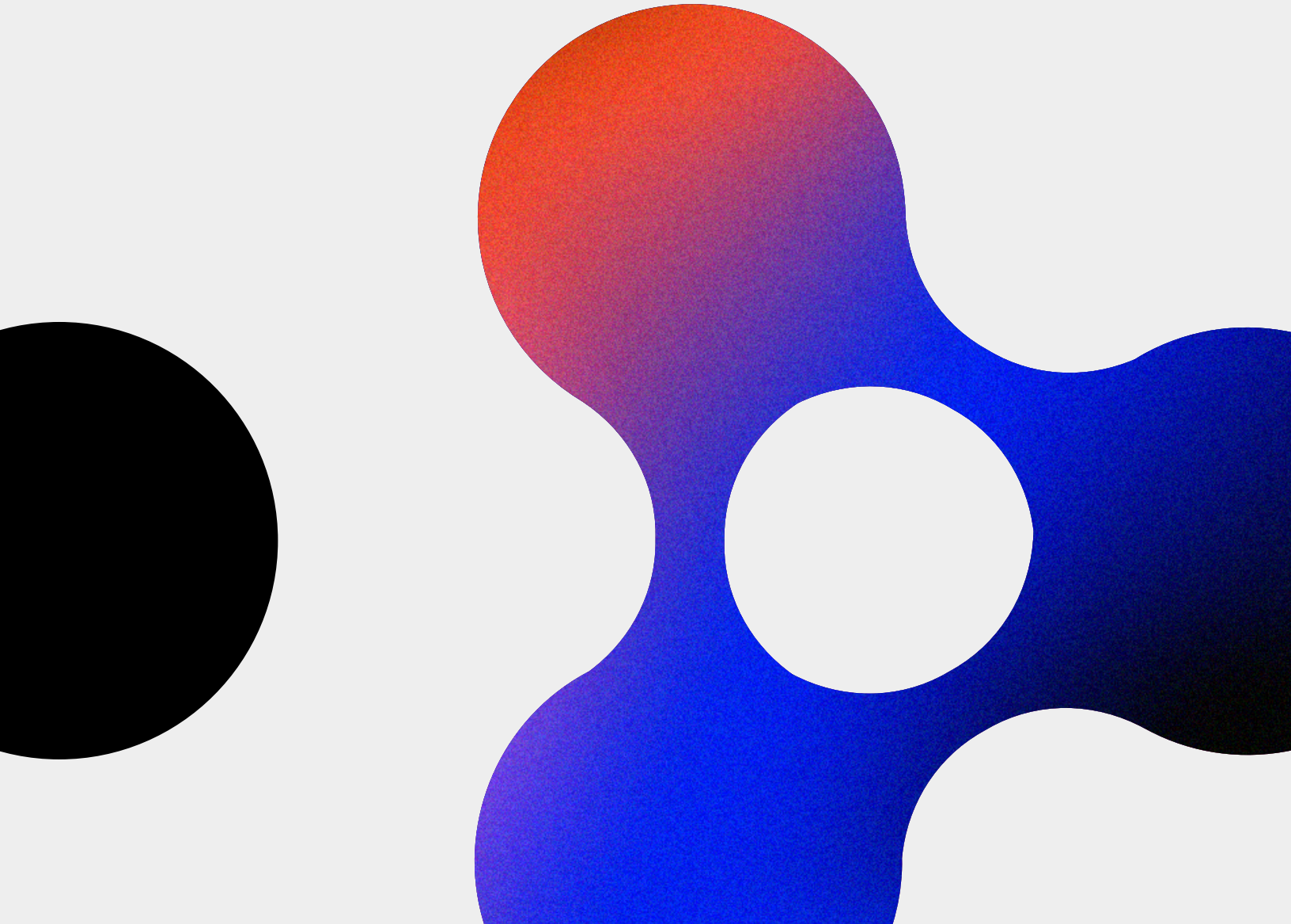


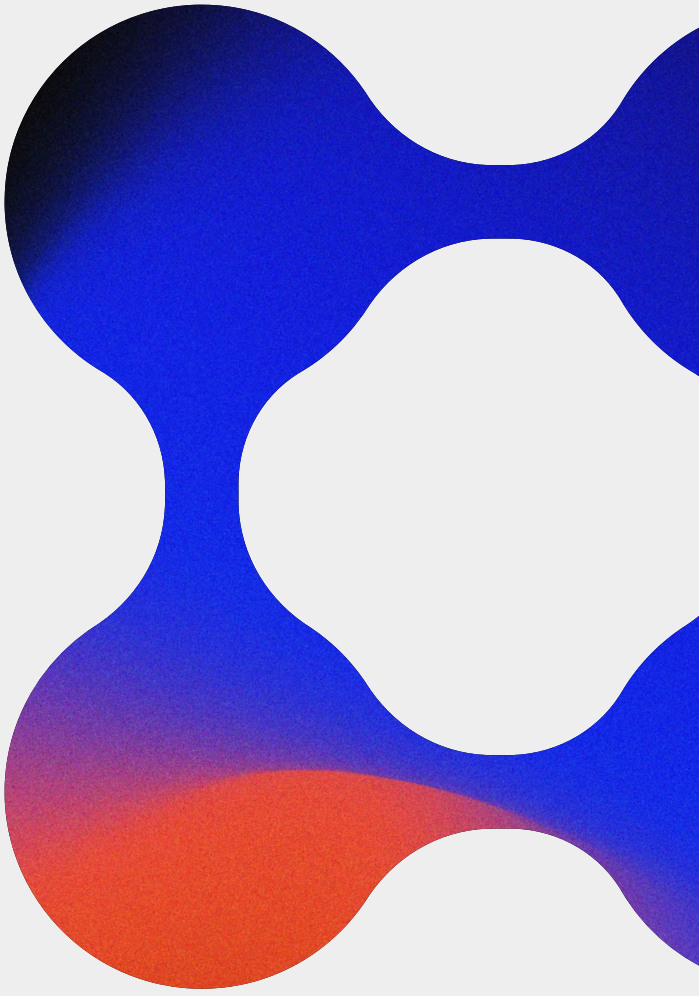
Table of Contents

Every laboratory that uses informatics software like a laboratory information management system (LIMS), laboratory information system (LIS), or manufacturing execution system (MES) is using a database to store and access the data produced by their laboratory. But what laboratory managers might not be aware of is that the underlying type of database — relational or non-relational — can impact how effectively a laboratory can innovate and scale.

Because life science domains like molecular diagnostics, animal studies, and drug discovery produce highly interconnected and complex data, laboratories that aim to put new workflows into production rapidly, and iterate on them, need an informatics platform that supports four primary functions:

- Storage of data with its context, in the form of metadata.
- The ability to naturally model complex real-world relationships and workflows.
- Auditability and provenance of the captured data.
- Easy reuse and searchability.

Most systems on the market today do not support these critical functions. They continue to use a relational database, even though studies show that graph databases — a type of non-relational database — are much more suitable for biomedical applications. In a survey of the literature, Timón-Reina, Rincón, and Martínez-Tomás (2021) concluded that "graph database management systems are fit to support data-intensive, integrative applications, targeted at both basic research and exploratory tasks closer to the clinic."[1]



Relational vs. non-relational databases: what's the difference?

Databases are categorized as relational or non-relational, depending on their underlying data structures.

TYPE	EXAMPLES	DESCRIPTION
Relational ("SQL") database	· PostgreSQL · Microsoft SQL Server · Oracle Database · MySQL	Databases that consist of multiple related tables, with data stored in rows and columns. They use structured query language (SQL) invented in the 1970s to read and make modifications to data in the database.
Non-relational ("NoSQL") database	· Document datastore · Column-oriented database · Key-value store · Graph database (RDF or Property)	Databases that do not use tables, allowing a more flexible, suitable data format.

Table 1. Comparison of differences between relational and non-relational databases.

Note that within the non-relational database grouping, there are two types of graph databases:

- **RDF knowledge graph database** (e.g., OntoText GraphDB). This type of database supports RDF (resource description framework[2]) and conforms to certain W3C standards.
- **Property graph database** (e.g., Neo4j). This type of database does not support RDF and is less precise.[3]

Oracle's article, Graph Database Defined, provides a comprehensive overview of the differences between RDF and property graphs.[4]

Why are relational databases insufficient for laboratories?

When building a software application with a relational database, data modeling must be performed ahead of time so that database tables can be set up for the specific types of data to be stored. Laboratories, therefore, must know upfront what types of data they need to store and what types of queries will be performed.

A major limitation of this is that if data has to be restructured at any point, a database migration must be performed. This is a process that becomes increasingly high-risk as the volume of data grows — every new addition to a table requires a corresponding change in the software code and decisions must be made about existing data with respect to the new addition. The more changes made, the more chance there is of introducing an error.

As a life science laboratory's operation evolves, what it stores is likely to change based both on what the laboratory wants to query and the new products it wants to create. In our experience, laboratories tend to use a lot of unstructured and user-defined data, which is not easy to deal with in a structured database table. Software architects and engineers are forced to manage this challenge of evolving data models by making explicit assumptions about the data domain, and hard coding those into the relational data models and the software applications that use them.

Non-relational databases, however, are less likely to need major database restructuring due to the inherent flexibility of their structures and the demands they place on the software applications that use them. For example, each piece of data is explicitly stored with its own data type, unlike in a relational database, where entire table columns need to be a uniform inferred type.

What is an RDF knowledge graph database?

As a type of non-relational database, RDF knowledge graph databases use a data model that conceptually consists of:

- Nodes representing entities, such as a person, sample, or reagent.
- Edges representing relationships between entities. For example, a sample is processed by a person, and an aliquot is taken from a sample.

If we were to represent this as an image, it would look something like the figure on the right.

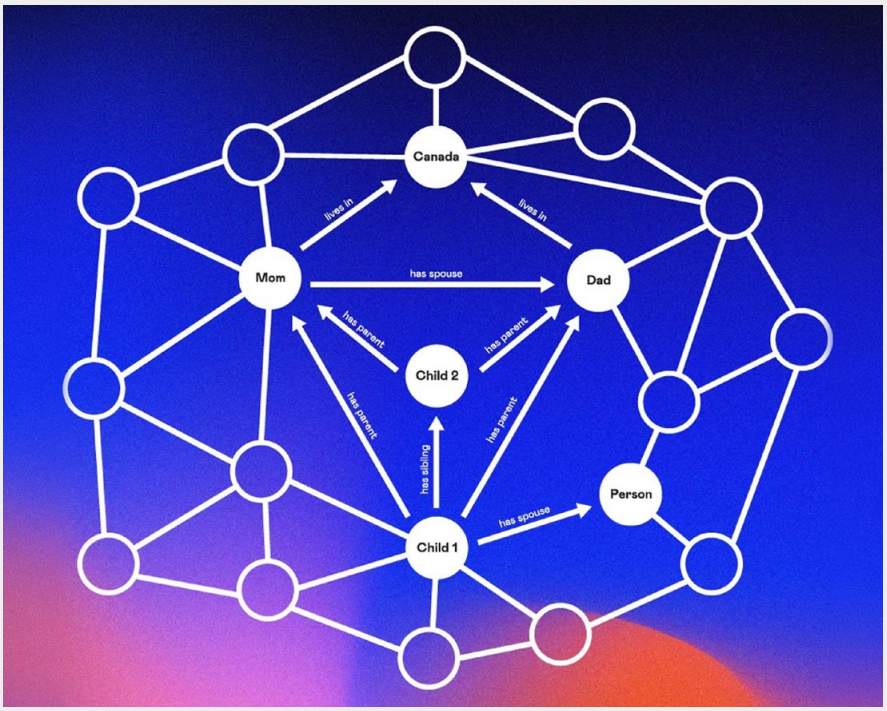


Figure 1. Example of an RDF knowledge graph database data model showing nodes and edges.

WHAT IS AN ENTITY?

In laboratory applications, common entities are samples, containers, reagents, instruments, and files. Entities typically have a lifecycle: they are generated, used, and invalidated. PROV, described below, defines an entity as "a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary."[5]

In data science terms, this structure is referred to as a directed graph. Directed graphs are stored using triple statements of subject-predicate-object. A triple statement (or 'triple' for short) consists of:

- A node for the subject.
- An edge that describes the connection from the subject to the object, known as the predicate.
- A node for the object.

Additionally, an object can be the subject for another triple, or a literal value to enable the storage of a simple number, for example.

This data model is much more flexible than a table, as it does not constrain the type of data that can be added to the graph. It also explicitly records the relationships between entities, which is a form of human-and machine-readable metadata not explicitly stored in relational databases.

Three advantages of an RDF knowledge graph database for clinical software

RDF knowledge graph databases offer laboratories a number of benefits compared to other types of databases. The three main advantages are:

1

MORE FLEXIBILITY AND AGILITY

RDF knowledge graphs simultaneously manage both the data and metadata (ontology[6]) at the same time. They therefore can evolve as the business and data requirements change or emerge. They enable the capture of any object or concept in a laboratory, at any time, in a way that is understood by the humans using them without adding technical debt to the data model.

In non-RDF graphs and relational databases, that tech debt makes data visualization and reporting convoluted, and thus necessitates an extract/transform/load (ETL)[7] — a three-phase process for aggregating data into a human rather than machine-readable form.

2

EASIER QUERYING AND MORE PRECISE DATA CAPTURE

Graph-stored data represents reality more accurately than the normalized forms required by relational databases. When information is needed from the database, a graph can provide the data quickly and with full context.

This capacity supports two relevant use cases for laboratories:

- The ability to create and update the data model within the informatics system to match the evolving conceptual ontology that laboratory staff hold in their minds about the laboratory, rather than forcing them to adapt to legacy software's narrowly defined view of it.
- Significantly faster knowledge inference for machine learning applications and other ad-hoc querying applications. Consider sample history auditing, for example. An RDF knowledge graph enables an informatics system to capture all context about a sample and display it on a single screen.

IMPROVED DATA SHAREABILITY BY ENABLING DATA INTEROPERABILITY

Paired with the right software and planning, an RDF knowledge graph enables a laboratory's data to be stored, maintained, and shared in accordance with the FAIR data principles.[8] Each piece of data includes rich metadata and has a unique identifier[9] (in an RDF knowledge graph database, this is an internationalized resource identifier or IRI[10]).

RDF knowledge graph database data also meets data provenance PROV standards[11] so it can be traced throughout a workflow. Meeting these principles and standards enables a laboratory to make connections to other sources of data and share data with external users. Effectively, software that is built on top of a FAIR RDF knowledge graph data store will allow its users — the subject matter experts — to control their ontologies directly. At the same time, the application handles mapping the ontologies to other ontologies both known and unknown, rather than requiring a data scientist expert to enable data sharing applications between different systems.

For instance, imagine two large academic research institutions, University A and Laboratory B, wanting to share their data or work together. Both run mouse vivariums, drug discovery units, and genomic sequencing labs. But to collaborate, they need to do some serious work to make their data and vocabularies match. Potentially, it could take months of meetings and years of data clean up effort to match up their production databases.

In this context, that type of collaboration is either not commercially possible or is made possible by using a data store that is extraneous to both organizations, further proliferating disconnected databases.

On the other hand, if both organizations were FAIR compliant (even if using different systems), they would simply need to point their data queries at one another and the recursive nature of the FAIR principle "I2. (Meta)data use vocabularies that follow FAIR principles" would enable seamless communication.

The practical result of this would be that University A could view Laboratory B's data in A's system (translated into their understood vocabulary), and Laboratory B could do the same with University A's data in B's systems.

Critically, when data is FAIR compliant, this type of collaboration is simple; it just works, without additional effort.

WHAT ARE FAIR DATA PRINCIPLES?

FAIR means that data is:

- **Findable:** Data and its metadata need to be easy to find by humans and computers.
- **Accessible:** Once the data has been found, there has to be a way to access it — including authentication and authorization.
- **Interoperable:** Data must be in a form that allows it to be integrated with other data, and within other applications or workflows — for storage, analysis, and processing without any extra transformation required.
- **Reusable:** The ultimate goal is that data is optimized for reuse so that results can be replicated or data can be combined with other datasets to discover new insights all while remaining compliant with whatever data usage license is defined for that data.

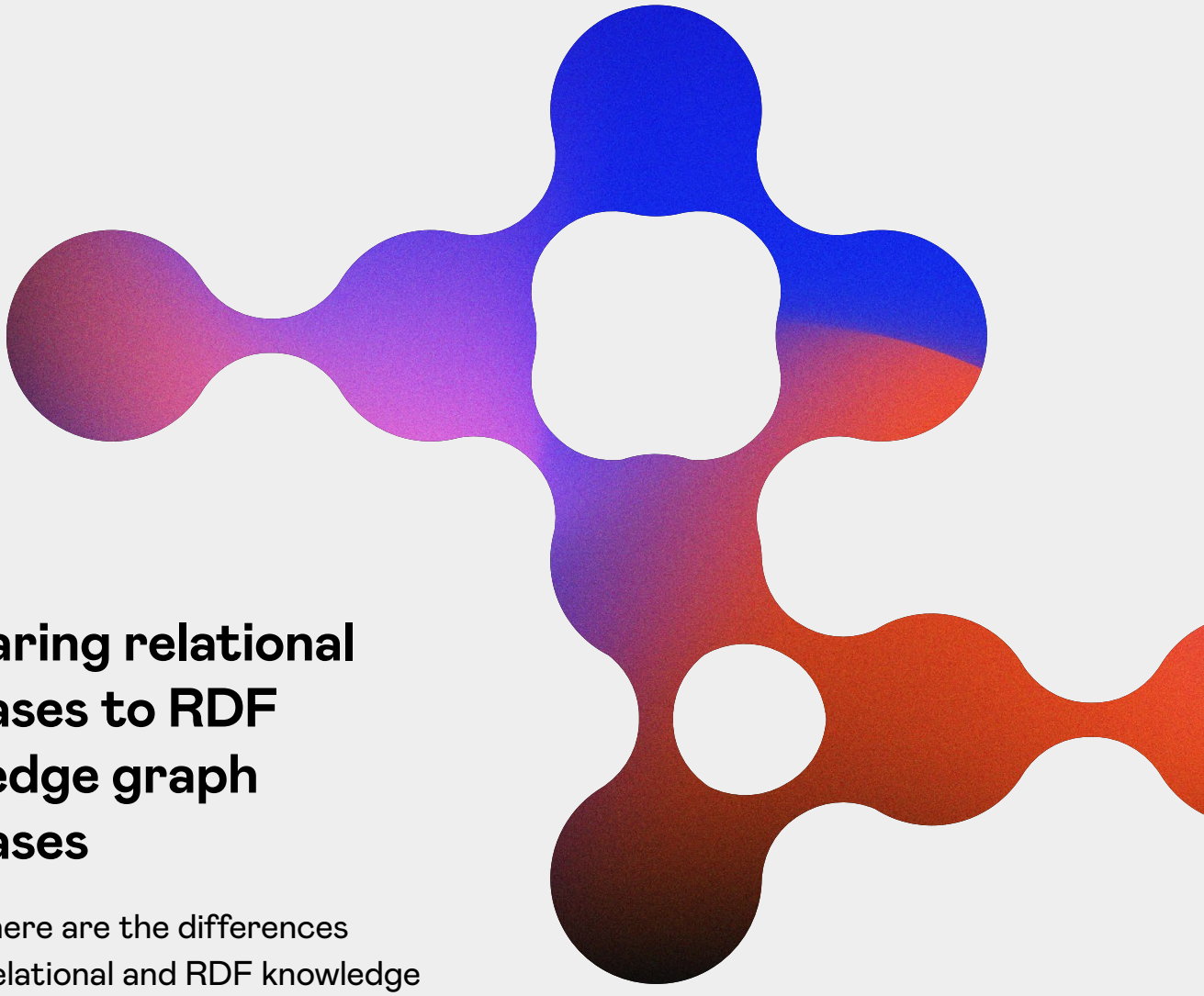
Laboratories that meet FAIR principles facilitate the exchange of both data and metadata, not only within the laboratory itself but also with other laboratories and organizations.

WHAT ARE DATA PROVENANCE AND PROV?

PROV is a set of data provenance standards — including a data model, ontology, notation, and other definitions (defining part of the metadata that a robust informatics system should capture) — developed by the W3 Consortium's Provenance Working Group. According to the W3 Consortium, *"Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability, or trustworthiness."*

Data in laboratories is often produced by separate systems, which can make it difficult to trace from end to end. When the systems are not well connected in a data context, it can be even more challenging to maintain the data's chain of custody records.

Laboratories that adhere to PROV standards, however, benefit from improved data auditability, transparency, and trust. For this reason, data provenance also supports collaboration, reproducibility, and patient safety.



Comparing relational databases to RDF knowledge graph databases

To recap, here are the differences between relational and RDF knowledge graph databases.

FEATURES	RELATIONAL DATABASE	RDF KNOWLEDGE GRAPH DATABASE
Relationships	Inferred and enforced using foreign keys between tables.	Stored explicitly and naturally between nodes as data.
Data Structure	Rigid — Must be pre-determined to create the correct tables.	Flexible — New types of data can be added without the need to change a schema or perform a migration.
Complex Querying	Slower and difficult to construct — Requires complex joins on data tables and deep knowledge of both the schema and best practices.	Faster — Does not require joins; follows connections between nodes, allowing for ad-hoc querying on any topic without prior data modeling or knowledge of the data model.

Table 2. Comparison between relational and RDF knowledge graph databases.

RDF knowledge graph databases provide a more flexible foundation for laboratory software

Although relational databases have been the default for laboratory software for many years, advances in technology and data modeling mean that laboratories can now choose a more flexible, proven option — the non-relational RDF knowledge graph database.

Modern RDF knowledge graph databases are at the forefront of data storage.[12] Because they can help future-proof software, they have been widely adopted by cutting-edge organizations with a heavy research focus.

For laboratories facing a replatforming or database migration, we recommend considering one of two options:

- Use an RDF knowledge graph database as part of an overall laboratory management data solution.
- Implement a next-generation informatics platform, such as Labbit, which natively employs an RDF knowledge graph database.

Beyond the use of an RDF knowledge graph database and automatic management of the ontologies necessary for true interoperability required by FAIR, Labbit offers a number of other advantages for molecular laboratories. Benefits include process and data capture in a single system, automatic data traceability and artificial intelligence/machine learning (AI/ML)-readiness, multi-site data sharing (data federation), reliable performance, and scalable productivity.

How do RDF knowledge graph databases perform?

In theory, using an RDF knowledge graph database seems like an obvious choice for laboratories. But, in practice, how do they perform within an informatics application like a LIMS, LIS, or MES?

As part of our performance analysis of the Labbit system, we timed the following processes:

- **Insert time:** The length of time it takes to insert an entire W3-PROV standard record from a task conducted on a typical 96-well plate into the database (including all activities and entities used to create the data; consisting of a few hundred RDF statements).
- **History:** The length of time it takes to get the entire provenance of an entity out of the database.
- **Range queries:** The length of time it takes to query over a range of datetimes, 64-bit double-precision numbers, or 64-bit integers. (i.e., "Give me all entities with a value between A and B")

Method: A simple Labbit testbed was constructed starting with an empty graph. Sample history was inserted in a tight loop. After every 1,000 loops, queries were run and times were recorded. This data formulated the curve of performance time vs. total database size, from 0 to around 3.7 billion. Performance analysis was conducted using an r5a.2xlarge system — a "general purpose" AWS server with 8 AMD EPYC 7000-series processor cores, 64 GB of RAM, with gp2-type storage, on OntoText GraphDB Free 9.2.0.

In the figure, the linearity demonstrates consistent performance on each of the range queries, a convincing (better than linear) power law trendline on insert time, and a near-flat linear trend on the history query. The overall trendline shows that as the database grows, Labbit's performance remains consistent.

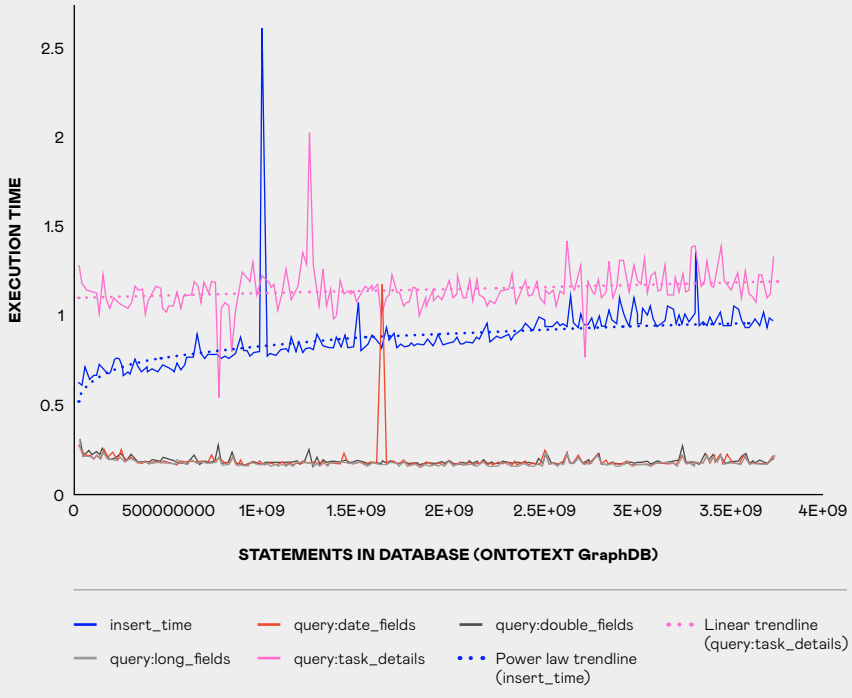


Figure 2. Performance curve generated during system testing.

In conclusion

Life science laboratories need an informatics platform that can support the increasingly complex workflows and studies that are a primary means for innovation. That means software systems must be designed to enable the full life cycle of development, from feasibility and validation all the way through to clinical implementation at scale and eventual retirement.

While there are many informatics choices available, most are built using tools that will limit flexibility in complex use cases over time, challenge operational growth, and lack full data traceability.

A solution built on an RDF knowledge graph database provides a foundation that can scale and evolve with a laboratory over time, meeting current and future requirements. This is particularly important as the laboratory community moves toward better educating the life sciences and medical communities on the significant value that laboratories and the professionals within them provide. Moreover, it will free up laboratory staff to focus on the work that is truly meaningful to them — developing innovative new workflows and products to support the health and wellness of people worldwide.

If you would like to learn more about Labbit, visit **labbit.com** or reach out to Eban Tomlinson at **1 (844) 744-3577**.

labbit

References

1 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8130509/
2 https://www.w3.org/TR/rdf-concepts/
3 https://docs.oracle.com/en/database/oracle/property-graph/21.3/spgdg/what-are-property-graphs.html
4 https://www.oracle.com/autonomous-database/what-is-graph-database/
5 https://www.w3.org/TR/prov-o/#Entity
6 https://en.wikipedia.org/wiki/Ontology_(information_science)
7 https://en.wikipedia.org/wiki/Extract,_transform,_load
8 https://www.go-fair.org/fair-principles/
9 https://www.w3.org/2004/11/uri-iri-pressrelease
10 https://datatracker.ietf.org/doc/html/rfc3987
11 https://www.w3.org/TR/prov-overview/
12 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8130509/