Our lab focuses on researching lymphoma, a type of cancer that, compared to others like breast or lung cancer, has fewer studies and data available in existing databases. This lack of comprehensive data makes it challenging for researchers and clinicians to access the necessary information for advancing lymphoma treatments and understanding. To address this issue, we embarked on creating a specialized lymphoma database aimed at overcoming the data sparsity problem. Our solution leverages information from well-established cancer databases and applies inference techniques from other cancer types to support lymphoma research and clinical decision-making.

To achieve this, we chose to implement a graph database, as it offers several advantages over traditional relational databases. Graph databases are particularly effective at modeling complex, interconnected data, which is common in biological systems where relationships between diseases, drugs, genes, and molecular profiles are intricate. They allow for more flexible querying and faster traversal of relationships between entities, making them ideal for representing the intricate relationships in cancer data. Graph databases also provide a more intuitive way to represent the hierarchical and associative data types that are crucial in oncology research, such as disease ontologies, drug interactions, and genetic mutations.

For the database infrastructure, we selected AuraDB by Neo4j, a fully managed, cloud-based graph database service. Instead of building and maintaining Neo4j locally, the cloud provides a more efficient solution in terms of time, labor, and cost. By leveraging the cloud, we avoid the complexities and overhead associated with maintaining an on-premise infrastructure, allowing us to focus more on data curation and analysis rather than database management.

In building the lymphoma database, we encountered several challenges, the most significant being data harmonization across multiple sources. Different databases often use varying formats, terminologies, and classifications, making it essential to standardize the data for integration. Another challenge was managing synonyms and abbreviations. Many terms, especially in the medical field, can have multiple names or abbreviations, requiring careful mapping to ensure consistency and accuracy across the database.

For our lymphoma database, we extract key information from several reputable sources: CIViC, CGI (for Statements), and the NCI Thesaurus (for Disease Hierarchy and Disease Molecular relationships). The structure of our database is illustrated in the graph model, where the "Statement" node serves as the core unit connecting to critical nodes such as drugs, diseases, and single nucleotide variants (SNVs). Each statement represents a clinically relevant association, such as drug responses (responsive, non-responsive, resistant) and outcomes (poor prognosis, sensitivity, positive response). The "Statement" node also captures the clinical significance, helping users evaluate potential treatment options based on molecular characteristics and their clinical outcomes.

To make the database more user-friendly, we developed a Neo4j query generator with a graphical user interface (GUI) to help users, even those without a coding background, easily generate basic queries. This tool simplifies access to the database, allowing users to retrieve the information they need without writing complex Cypher queries. However, it's worth mentioning that AuraDB itself comes with an excellent GUI that allows users to search for keywords and explore the database by clicking on interesting nodes and edges, making data exploration more intuitive.

Ultimately, our goal is to incorporate data from internal sources to further enhance the database's value in facilitating clinical discoveries, enabling a more comprehensive and actionable understanding of lymphoma and its molecular underpinnings.