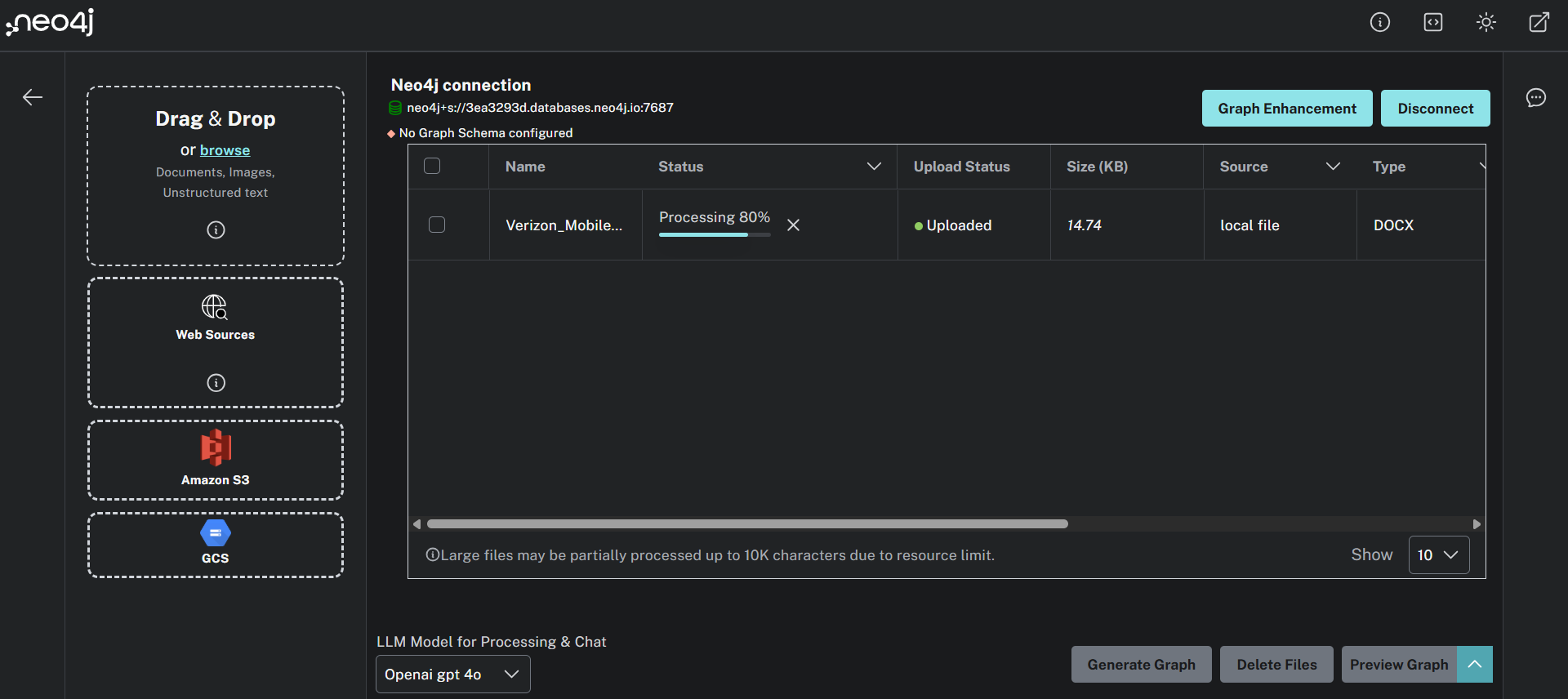
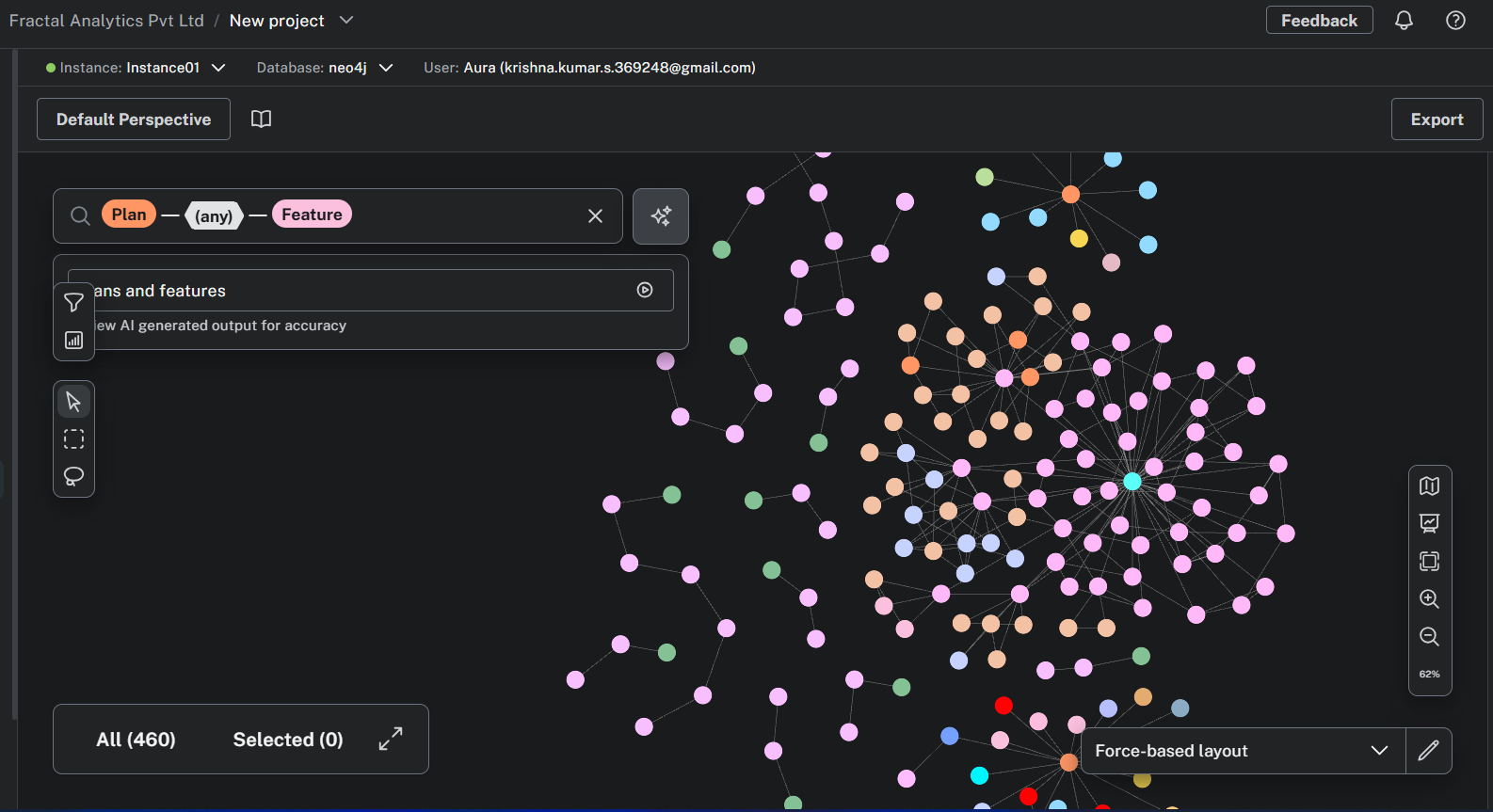
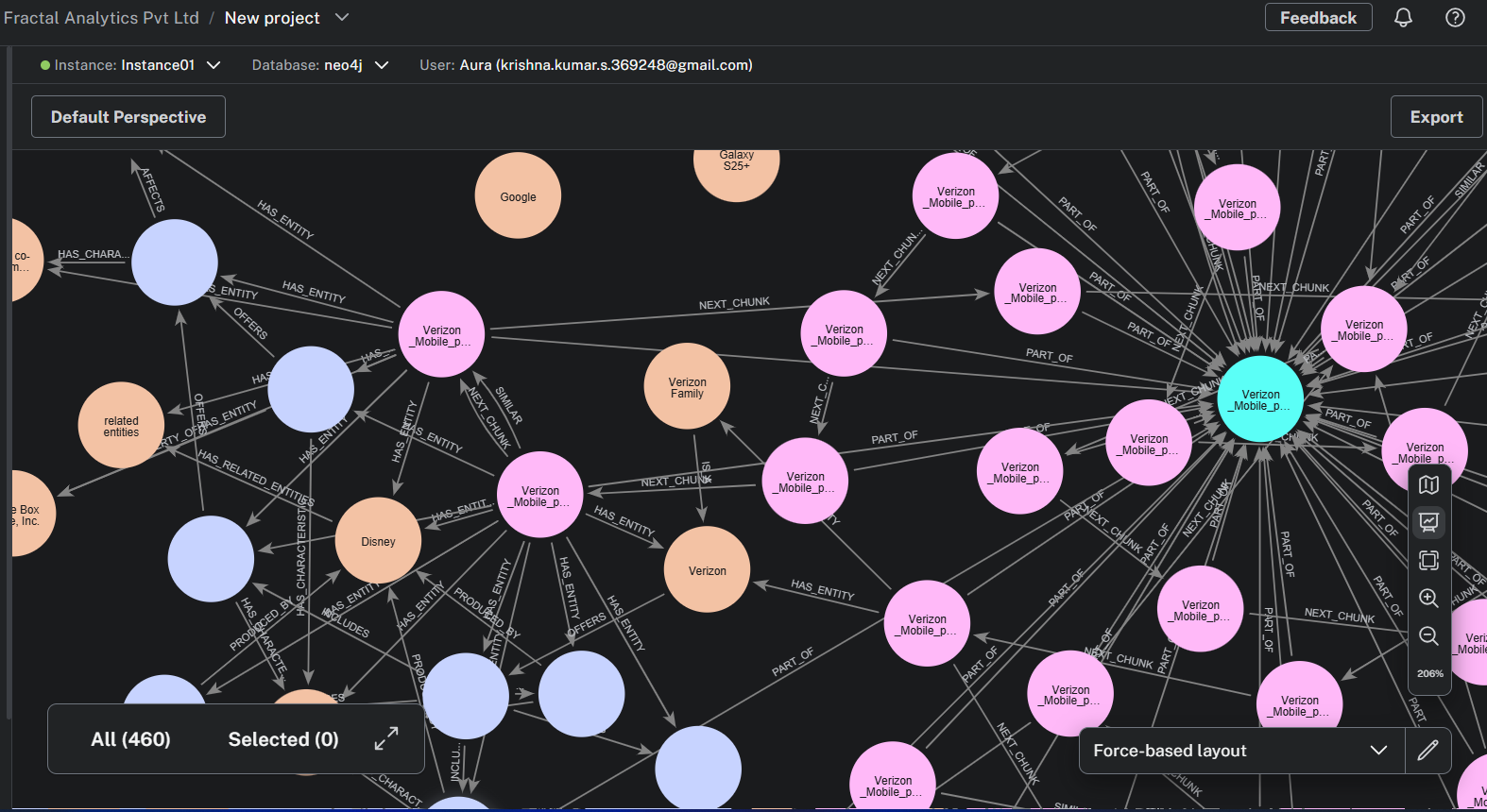
# Neo4j LLM Graph Builder



Example Automated Graph on the contents in the website <https://www.verizon.com/plans/unlimited/>

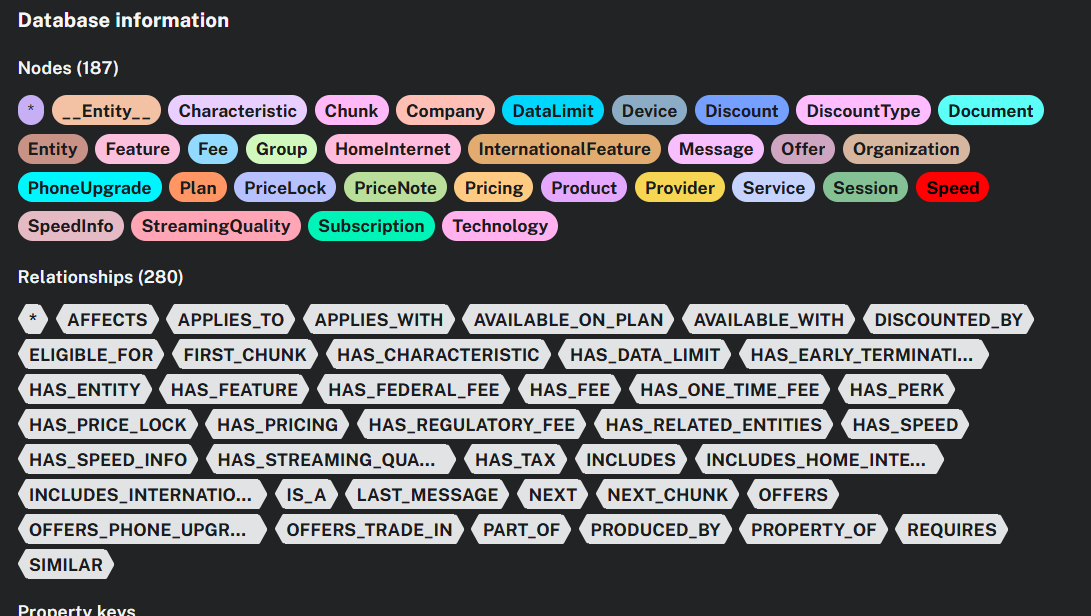


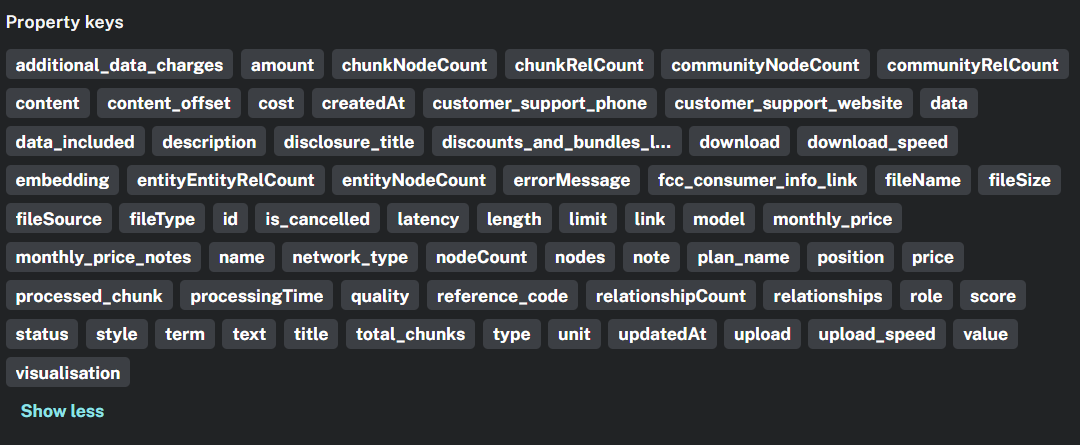
A closer look at the same here.



As we can see, it has performed chunking and entity extraction, the entities are connected and the order of the chunks are also connected. There has been chunk overlap configured too.

So many entities and relationships it will become hard for LLMs to write cypher queries for them.

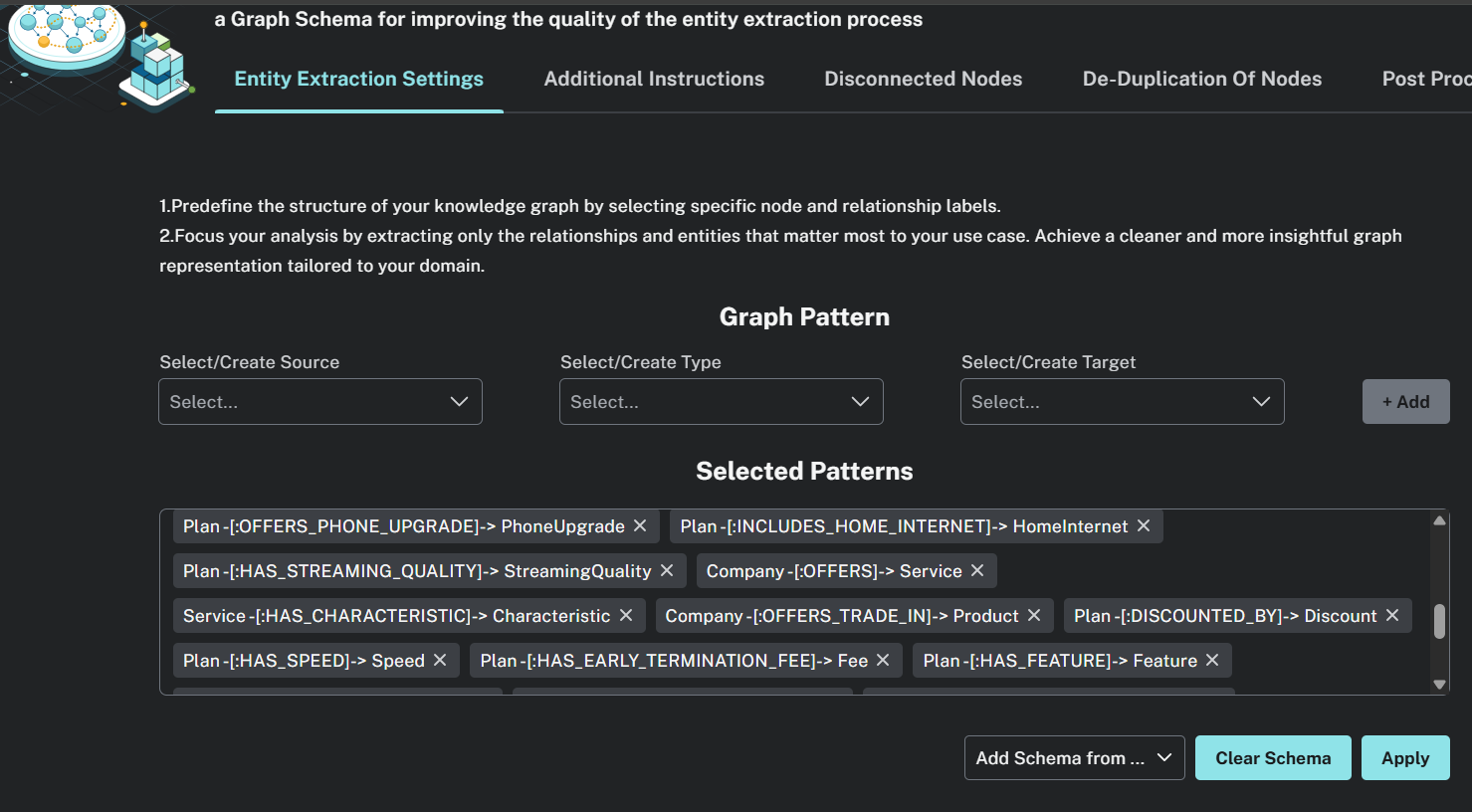




## Complexities involved in using Neo4j Graph Builder

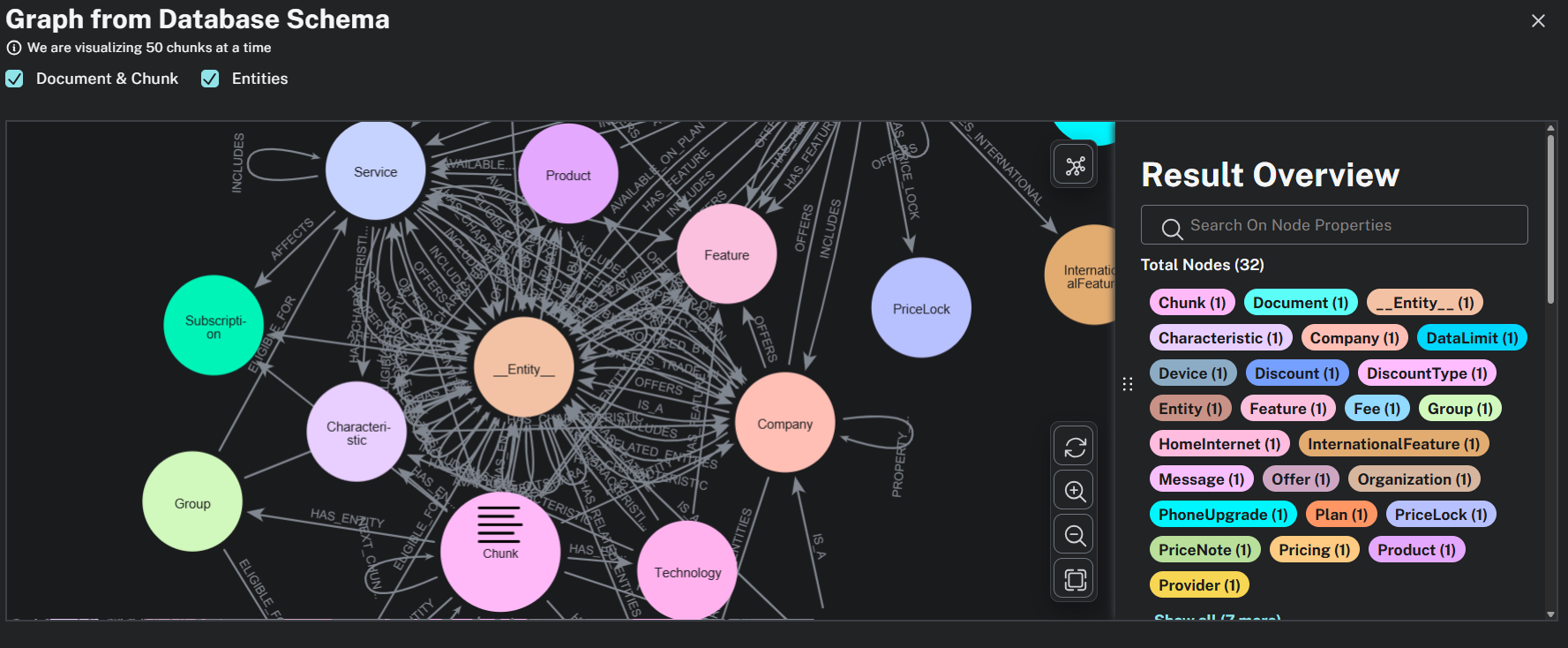
**1. Increased Complexity for Pattern Matching:**

* **Numerous Node Labels and Relationship Types:** A complex graph often involves a high number of distinct node labels and relationship types. To write an accurate Cypher query, the GenAI model needs to understand and correctly utilize these labels and types in the MATCH clause to find the relevant patterns. The more options there are, the higher the chance of the AI making an incorrect selection.
* **Deeply Nested or Multi-Hop Relationships:** Complex queries often require traversing multiple relationships to find the desired information (e.g., friends of friends of colleagues). GenAI models can struggle to accurately chain these relationships in the MATCH clause, especially when there are many possible paths.
* **Varied Properties:** Nodes and relationships in a complex graph can have a wide array of properties. Filtering based on combinations of these properties using the WHERE clause becomes more intricate. The GenAI needs to know which properties exist and how to correctly formulate the filtering conditions.



**2. Difficulty in Understanding Intent and Context:**

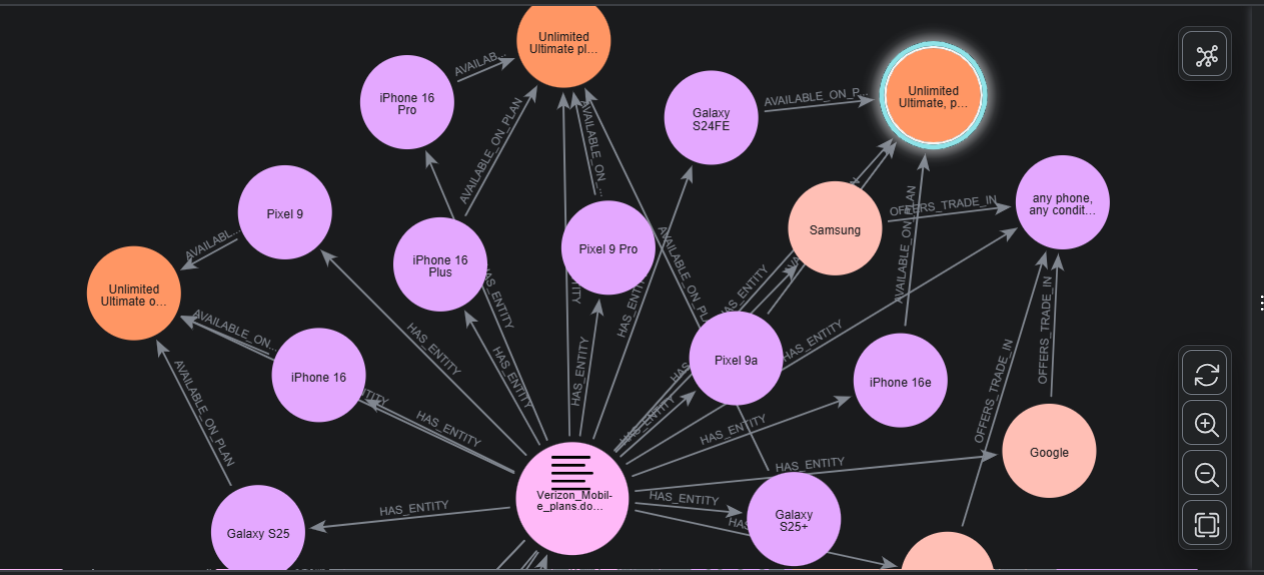
* **Ambiguous Natural Language:** User questions in natural language can sometimes be ambiguous, and in the context of a complex graph, the ambiguity can lead to vastly different Cypher queries. The GenAI might misinterpret which parts of the graph the user is interested in.
* **Lack of Graph Schema Understanding:** While some GenAI integrations can access the graph schema, fully understanding the semantic meaning behind each node label, relationship type, and property can be challenging. This lack of deep understanding can result in semantically incorrect queries, even if the syntax is valid.



This image illustrates how complex the schema itself is for one webpage of verizon, if we were to rely on automated generation. Which is why pre-defining the schema is essential.

**3. Challenges in Query Construction Logic:**

* **Combining Multiple Criteria:** Complex queries often involve combining several matching patterns, filtering conditions, and aggregation functions. GenAI models might struggle to structure these different parts of the Cypher query (MATCH, WHERE, WITH, RETURN, etc.) in the correct logical order.
* **Using Advanced Cypher Features:** Features like subqueries, variable-length paths, and conditional expressions might be necessary for complex data retrieval. GenAI models might not be proficient in using these more advanced aspects of the Cypher language accurately.



Preferable only if we have to depend on continuous text documents for example, privacy policy, network policy, terms and conditions document etc

**4. Output Complexity and Reasoning:**

* **Large and Interconnected Results:** Even if a GenAI can produce a Cypher query, understanding the structure of the expected output from a complex graph and formulating a query to return it in a useful way can be difficult.
* **Explainability of the Graph Structure:** If the GenAI doesn't have a strong internal representation of the complex graph's structure and the relationships between different parts, it will struggle to reason about how to query it effectively.

In essence, the more intricate the web of nodes and relationships becomes, and the more diverse the properties within them, the larger the "search space" for the GenAI to generate a correct Cypher query. This increased complexity makes it harder for the AI to:

* Identify the relevant parts of the graph.
* Construct the correct path or pattern to match.
* Filter the data appropriately.
* Return the desired information in the right format.

Therefore, while Neo4j graph builders are powerful for creating rich and interconnected datasets, the resulting complexity can indeed become a significant hurdle for GenAI models attempting to automatically generate Cypher queries.