**Explanation of Data Structure**

**Hierarchical Structuring:**

The GStreamer tutorial for dynamic pipeline creation is organized into sections and subsections to provide a clear, logical flow of information. It is stored in a JSON file format, which supports nested hierarchical storage and easy data retrieval. Each section (such as Introduction, Key Concepts, Tutorial Steps, Examples and Use Cases) contains detailed subsections (like title, content, related sections) that further break down the content.

**Metadata Tagging:**

Each section and subsection within the GStreamer tutorial for dynamic pipeline creation is tagged with relevant keywords, optimizing the RAG model's capability to efficiently retrieve targeted information. Categories are employed to cluster related details, ensuring seamless navigation through concepts, code snippets, and tutorial steps.

**Cross-References:**

*Related Sections:* Each section includes references to related sections. For example, in "GStreamer Elements," there are links to the tutorial steps and code snippets that demonstrate the use of elements.

*Related Concepts:* Tutorial steps link back to key concepts they cover, to link the different elements semantically

*Related Code:* Code snippets are linked to the tutorial steps where they are explained, providing examples.

**Embedding Strategy:**

* *Text Extraction:* Text from each section and subsection is extracted from the JSON file.
* *Preprocessing:* The extracted text is preprocessed by converting it to lowercase and removing punctuation to enhance retrieval and embedding creation.
* *Embedding Creation:* Embeddings for the content text are created using the Mistral Embedding model, which converts text into numerical vectors of embeddings in 1024 dimensions.
* *Storage:* These embeddings are then stored in the "content" column along with section and key columns in the Supabase table database.