**Notion to Pinecone Vector Store Integration.txt**

**Workflow: Prod: Notion to Vector Store - Dimension 768**

**Purpose:**  
This production workflow is designed to automate the process of retrieving and processing content from a Notion database and inserting its vector embeddings into a Pinecone vector store. The workflow is configured for a vector dimension of 768. It supports use cases such as creating a searchable knowledge base or enhancing retrieval-augmented generation (RAG) systems.

**Key Components**

1. Notion Trigger

* **Node:** *Notion - Page Added Trigger*
  + **Type:** Notion Trigger (n8n Notion Trigger Node)
  + **Description:**  
    This node watches for newly added pages in a specified Notion database. When a new page is created (e.g., a new article or document), the node triggers the workflow.
  + **Configuration:**
    - **Database ID:** Set to the target Notion database (Embeddings database).
    - **Poll Times:** Configured to check for new pages every minute.

2. Retrieve Notion Page Content

* **Node:** *Notion - Retrieve Page Content*
  + **Type:** Notion (n8n Notion Node)
  + **Description:**  
    This node fetches all the block content of the newly added Notion page. It retrieves detailed content, including paragraphs, headings, lists, etc.
  + **Configuration:**
    - **Block ID:** Dynamically set from the trigger node's output.
    - **Return All:** Configured to retrieve all child blocks.

3. Filter Non-Text Content

* **Node:** *Filter Non-Text Content*
  + **Type:** Filter (n8n Filter Node)
  + **Description:**  
    This node filters out non-text content from the Notion page blocks, such as images and videos, to retain only the textual content for further processing.
  + **Configuration:**
    - **Conditions:** Filters out blocks where the type is "image" or "video".

4. Summarize Notion Content

* **Node:** *Summarize - Concatenate Notion's blocks content*
  + **Type:** Summarize (n8n Summarize Node)
  + **Description:**  
    This node concatenates the text content from the filtered Notion blocks into a single string. The resulting text serves as a summary or document to be processed further.
  + **Configuration:**
    - **Field to Summarize:** The text content of the blocks, separated by newlines.

5. Create Metadata and Load Content

* **Node:** *Create metadata and load content*
  + **Type:** Document Default Data Loader (LangChain Document Loader)
  + **Description:**  
    This node packages the concatenated text with additional metadata about the Notion page, such as its ID, creation time, and title. This structured data is then ready for further processing.
  + **Configuration:**
    - **Metadata:** Includes page ID, creation time, and page title.
    - **JSON Data:** Uses the concatenated content from the previous step.

6. Token Splitting

* **Node:** *Token Splitter*
  + **Type:** Text Splitter (Token Splitter Node)
  + **Description:**  
    To ensure that the text can be effectively processed by downstream AI models, this node splits the concatenated text into chunks. Each chunk has a maximum size of 256 tokens with an overlap of 30 tokens.
  + **Configuration:**
    - **Chunk Size:** 256 tokens.
    - **Chunk Overlap:** 30 tokens.

7. Embeddings Generation

* **Node:** *Embeddings Google Gemini*
  + **Type:** Embeddings (Google Gemini Embeddings Node)
  + **Description:**  
    This node generates vector embeddings for the text chunks using Google Gemini's text embedding model (models/text-embedding-004).
  + **Configuration:**
    - **Model Name:** Set to "models/text-embedding-004".
  + **Credentials:**
    - Uses a Google Gemini (PaLM) API account.

8. Vector Storage in Pinecone

* **Node:** *Pinecone Vector Store*
  + **Type:** Vector Store (Pinecone Vector Store Node)
  + **Description:**  
    This node inserts the generated vector embeddings into a Pinecone vector store. The target vector store is specified with an index named "notion-pages".
  + **Configuration:**
    - **Mode:** Insert.
    - **Pinecone Index:** Set to "notion-pages".
  + **Credentials:**
    - Uses a Pinecone API account.

**Data Flow Overview**

1. **Trigger & Content Retrieval:**
   * The workflow starts when a new page is added to the specified Notion database.
   * The Notion trigger node captures the event, and the page's URL is passed to the "Notion - Retrieve Page Content" node.
2. **Content Processing:**
   * The retrieved blocks are filtered to exclude non-text content.
   * The text from the remaining blocks is concatenated into a single document using the Summarize node.
3. **Metadata Creation:**
   * The concatenated text, along with metadata (page ID, created time, page title), is packaged by the "Create metadata and load content" node.
4. **Token Splitting:**
   * The document is split into manageable token chunks using the Token Splitter node.
5. **Embeddings Generation:**
   * Each text chunk is transformed into vector embeddings using the "Embeddings Google Gemini" node.
6. **Vector Storage:**
   * The embeddings, with their associated metadata, are inserted into the Pinecone vector store via the "Pinecone Vector Store" node.

**Setup Instructions**

1. **Notion Integration:**
   * Create a Notion integration and share the target database with this integration.
   * Add your Notion API credentials to n8n.
   * Configure the Notion trigger and retrieval nodes with the correct database URL/ID.
2. **Google Gemini & Pinecone Setup:**
   * Set up your Google Gemini (PaLM) API credentials.
   * Configure the embeddings node with the appropriate model.
   * Set up your Pinecone vector store and add your credentials to n8n.
3. **Test the Workflow:**
   * Add a new page to your specified Notion database.
   * Monitor the workflow execution to verify that text is properly retrieved, processed, and vector embeddings are stored in Pinecone.
4. **Customization:**
   * Adjust the token splitter parameters if needed (chunk size and overlap).
   * Modify metadata assignments as required for your application.

**Final Output**

The final output of the workflow is the successful insertion of vector embeddings into the Pinecone vector store, enabling efficient retrieval of Notion content for downstream tasks like semantic search or retrieval-augmented generation (RAG).

By following the setup instructions and reviewing the data flow, users can deploy this workflow to automatically convert Notion page content into a searchable vector store, enhancing the capabilities of their AI systems.

Happy Automating with n8n and Pinecone!