Output Hub Data Relationships

Core Modules

1. Custom GPTs

The `customgpt` table is central to the system, representing custom GPT models.

Key relationships:

- One-to-many with `lookupgptmodels`: Each custom GPT is based on a specific GPT model.
- One-to-many with `lookupgptrating`: Custom GPTs can be rated for performance or quality.
- One-to-many with `lookupgptresponsetimes`: Response times are tracked for each custom GPT.
- One-to-many with `lookupllmlist`: Custom GPTs are associated with specific language models.

Many-to-many relationships:

- With `lookupagentgroups` via `customgpt_agentgroups`: Custom GPTs can belong to multiple agent groups, allowing for categorization and organization.
- With `lookupgptcats` via `customgpt_gptcats`: Custom GPTs can be categorized into multiple GPT categories.
- With `lookupgptoutputreviewsdone` via `customgpt_outputreviewsdone`: Tracks which output reviews have been performed on each custom GPT.
- With `lookupprojecttags` via `customgpt_projecttags`: Custom GPTs can be tagged with multiple project tags for organization and searching.
- With `promptlibrary` via `customgpt_promptlibrary`: Custom GPTs can be associated with multiple prompts from the prompt library.
- With `promptoutput` via `customgpt_promptoutput`: Links custom GPTs to the outputs they generate.

2. Prompts

The `promptlibrary` table stores reusable prompts for the system.

Key relationships:

- One-to-many with `lookuppromptdevstages`: Each prompt is associated with a development stage.

Many-to-many relationships:

- With `lookupprojecttags` via `promptlibrary_projecttags`: Prompts can be tagged with multiple project tags.
- With `customgpt` via `customgpt_promptlibrary`: Prompts can be associated with multiple custom GPTs.
- With `promptoutput` via `promptoutput_promptlibrary`: Tracks which prompts generated which outputs.

3. Outputs

The `promptoutput` table stores the outputs generated by the system.

Key relationships:

- One-to-many with `promptlibrary`: Each output is generated from a specific prompt.
- One-to-many with `lookupaccuracylevel`: Outputs can be rated for accuracy.
- One-to-many with `lookupactionabilitylevel`: Outputs can be rated for actionability.
- One-to-many with `lookupllmlist`: Each output is associated with a specific language model.

Many-to-many relationships:

- With `lookupknowledgetypes` via `promptoutput_knowledgetypes`: Outputs can be categorized by knowledge types.
- With `lookupmediatypes` via `promptoutput_mediatypes`: Outputs can be associated with different media types.
- With `lookupprojecttags` via `promptoutput_projecttags`: Outputs can be tagged with multiple project tags.
- With `outputtag` via `promptoutput_tags`: Allows for flexible tagging of outputs.

Lookup Tables and Relationships

1. `lookupllmlist`: Stores information about language models.
    - Referenced by `customgpt` and `promptoutput`
2. `lookupgptmodels`: Contains different GPT models available.
    - Referenced by `customgpt`
3. `lookupgptrating`: Stores possible ratings for GPTs.
    - Referenced by `customgpt`
4. `lookupgptresponsetimes`: Defines response time categories for GPTs.
    - Referenced by `customgpt`
5. `lookupagentgroups`: Defines groups of agents or GPTs.
    - Many-to-many with `customgpt`
6. `lookupgptcats`: Stores categories for GPTs.

    - Many-to-many with `customgpt`
7. `lookupgptoutputreviewsdone`: Tracks types of output reviews.
    - Many-to-many with `customgpt`
8. `lookupprojecttags`: Stores project-related tags.
    - Many-to-many with `customgpt`, `promptlibrary`, and `promptoutput`
9. `lookuppromptdevstages`: Defines development stages for prompts.
    - Referenced by `promptlibrary`
10. `lookupaccuracylevel`: Defines accuracy levels for outputs.
    - Referenced by `promptoutput`
11. `lookupactionabilitylevel`: Defines actionability levels for outputs.
    - Referenced by `promptoutput`
12. `lookupknowledgetypes`: Defines types of knowledge for outputs.
    - Many-to-many with `promptoutput`
13. `lookupmediatypes`: Defines media types for outputs.
    - Many-to-many with `promptoutput`
14. `outputtag`: Stores tags for flexible output categorization.
    - Many-to-many with `promptoutput`

Additional Tables

1. `users`: Stores user information for the system.
2. `accessui`: Manages UI access for custom GPTs.
    - One-to-many with `customgpt`: Each UI access entry is associated with a specific custom GPT.

System Overview

The Output Hub system is designed to manage and analyze interactions with Large Language Models (LLMs), particularly focusing on custom GPT models, prompts, and their outputs. The system allows for extensive categorization, tagging, and relationship mapping between these core components.

Custom GPTs form the central entity, representing specialized language models that can be associated with various attributes such as model type, rating, response time, and language model. They can be grouped, categorized, and tagged in multiple ways, allowing for flexible organization and retrieval.

Prompts are stored in a library, each associated with a development stage and capable of being tagged for project relevance. These prompts are used to generate outputs, which form the third core component of the system.

Outputs are linked to the prompts that generated them and the custom GPTs that produced them. They can be evaluated for accuracy and actionability, categorized by knowledge and media types, and flexibly tagged for easy retrieval and analysis.

The extensive use of lookup tables and many-to-many relationships throughout the system allows for a high degree of flexibility in categorizing and organizing the data. This structure supports complex queries and analyses, enabling users to gain deep insights into the performance and characteristics of their custom GPT models and the quality of the outputs they produce.

The inclusion of user management and UI access control adds a layer of security and personalization to the system, allowing for controlled access to specific custom GPTs and their associated data.

Overall, this database structure provides a robust foundation for a sophisticated system to manage, analyze, and improve LLM interactions, supporting use cases ranging from content creation and research to software development and customer support.