SQL AI Assistant in DataHub (Preview)

Date published: 2024-02-29
Date modified: 2024-02-29

Legal Notice

Contents

About the Hue SQL AI Assistant

A SQL AI Assistant has been integrated into Hue with the capability to leverage the power of Large Language Models (LLMs) for various SQL tasks. It helps you to create, edit, optimize, fix, and succinctly summarize queries using natural language and makes SQL development faster, easier, and less error-prone. The Hue AI assistant is available with Cloudera Runtime 7.2.18 and higher.

Note: *The Hue SQL AI assistant is in technical preview and not recommended for use in production deployments. Cloudera recommends that you try this feature in test and development environments.*

Which AI models and services does Hue use?

The AI Assistant supports various LLMs and hosting services. The model can run in the infrastructure of a service provider, and the AI Assistant can be configured to use them remotely. Cloudera has tested with GPT running in Azure OpenAI, and the following service-model combinations are supported

Service Provider	Model	Model versions
OpenAI	OpenAI GPT	gpt-3.5-turbo gpt-3.5-turbo-16k
Microsoft Azure	OpenAI GPT	gpt-3.5-turbo gpt-3.5-turbo-16k
Amazon Bedrock	Anthropic Claude	anthropic.claude-v1 anthropic.claude-v2
Amazon Bedrock	Amazon Titan	amazon.titan-text-express-v1

Note: *Cloudera recommends using the Hue AI assistant with the Azure OpenAI service.*

For augmenting the results, the SQL AI Assistant uses a Retrieval Augmented Generation (RAG)-based architecture. It uses the sentence-transformer library for semantic search, and Hue can be configured with any of these pre-trained models for better multi-lingual support. By default, **all-MiniLM-L6-v2** models are used.

Embedding Model	Language Support
all-MiniLM-L6-v2	English only

distiluse-base-multilingual-cased-v1	Arabic, Chinese, Dutch, English, French, German, Italian, Korean, Polish, Portuguese, Russian, Spanish, and Turkish.

What data is shared with the LLM models?

The following details are shared with the Large Language Models (LLMs):

- Everything that a user inputs
- Dialect in use
- Table details such as table name, column names, column data types and related keys, partitions, and constraints that the logged-in user has access to.
- Three sample rows from the tables (as per the best practices specified in *Rajkumar et al, 2022*)

Prerequisites for enabling the SQL AI Assistant

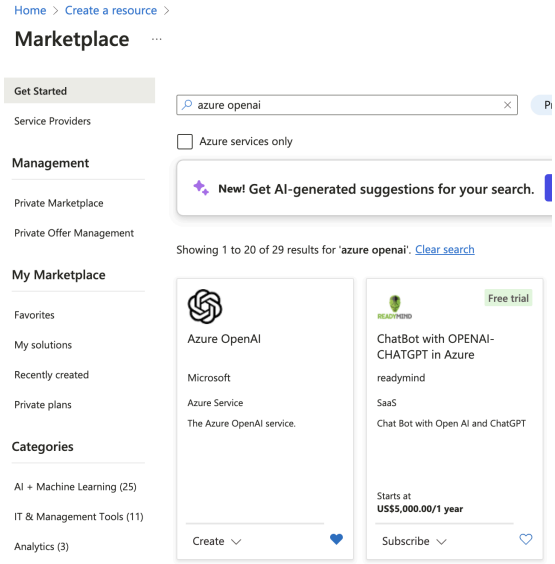
As an administrator, you must obtain clearance from your organization's infosec team to make sure it is safe to use the SQL AI Assistant because some of the table metadata and data, as mentioned in the previous section, is shared with the LLM.

Configuring the SQL AI Assistant on the Microsoft Azure OpenAI Service

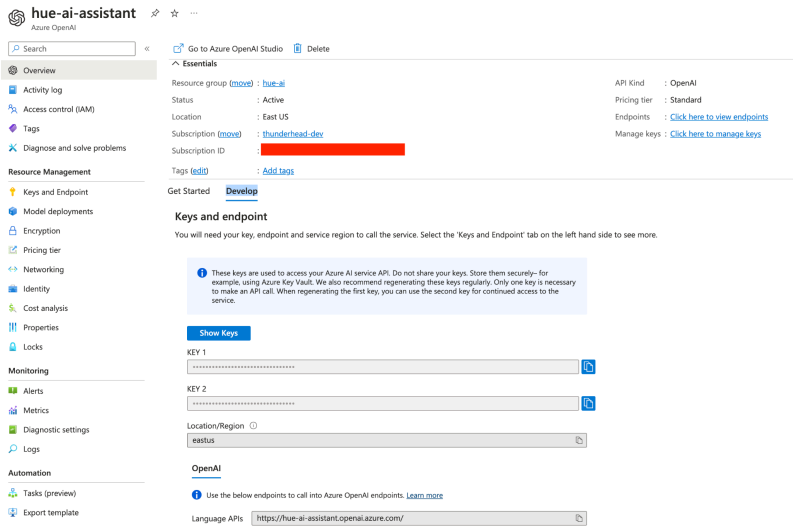
Microsoft Azure allows dedicated deployments of OpenAI GPT models. Using Azure's OpenAI service is much more secure than the publicly hosted OpenAI APIs because the data can be processed in your Virtual Private Cloud (VPC). Due to security considerations, Cloudera recommends that you use GPT models in Hue SQL AI Assistant with Azure's OpenAI service.

Steps

1. Obtain a Microsoft Azure subscription by working with your organization's IT team. Subscriptions vary based on your team and purpose.
2. Register to access the Azure OpenAI service.
   Azure OpenAI requires registration and is currently only available to approved enterprise customers and partners. Customers who wish to use Azure OpenAI are required to submit a registration form.
3. Create an Azure OpenAI resource in the Azure portal.



4. Obtain the resource URL and resource keys from the **Develop** tab under the resource details page.
   You can use any one of the two keys as shown in the following image:



5. Go to Azure OpenAI Studio at https://oai.azure.com/portal and create your deployment under **Management** > **Deployments**. Select **gpt-35-turbo-16k** or higher.
6. Enable the Hue SQL AI Assistant as follows:
   a. Log in to the CDP Management Console as an Administrator.
   b. Go to **Environments** and select your environment.
   c. Go to the **Data Lake** tab and click on the CM URL to open Cloudera Manager.
   d. Go to **Clusters** > **Hue service** > **Configuration** and select add the following lines in the **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini** field:

```
[desktop]
```

```
[[ai_interface]]
    service='azure'
    model_name='[***DEPLOYMENT-NAME***]'
    base_url="https://[***RESOURCE***].openai.azure.com/"
    token="[***RESOURCE-KEY***]”
```

e. Click **Save Changes**.

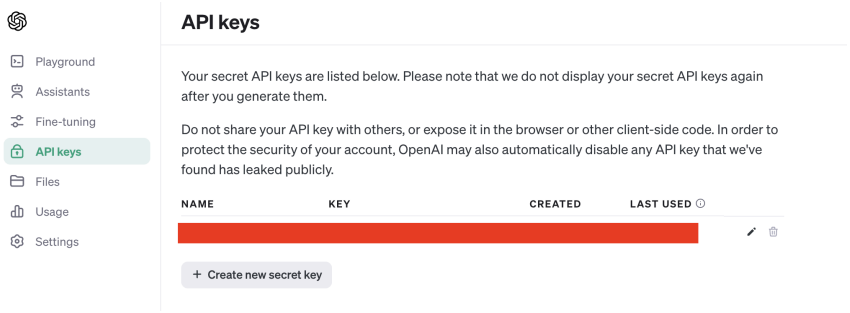
Configuring the SQL AI Assistant on the OpenAI Service

Before you begin

You must have created an account with the OpenAI platform.

Steps

1. Log in to the OpenAI portal.
2. Obtain the API key by navigating to the API keys from the left navigation pane. The key is required to integrate Hue SQL AI Assistant with the OpenAI service.



3. Enable the Hue SQL AI Assistant as follows:
    a. Log in to the CDP Management Console as an Administrator.
    b. Go to **Environments** and select your environment.
    c. Go to the **Data Lake** tab and click on the CM URL to open Cloudera Manager.
    d. Go to **Clusters** > **Hue service** > **Configuration** and select add the following lines in the **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini** field:

```
[desktop]
     [[ai_interface]]
          service='openai'
          token='[***API-KEY***]'
```

    e. Click **Save Changes**.

Configuring the SQL AI Assistant on the Amazon Bedrock Service

Amazon Bedrock is a fully managed service that makes foundation models from leading AI startups and Amazon available through an API.

Before you begin

You must have an AWS account with Bedrock access.

Steps

1. Log in to the Amazon Bedrock service.
2. Obtain your access key and secret as follows:
    a. Go to the IAM console: https://console.aws.amazon.com/iam
    b. Click on **Users** from the left menu and select the user you want to access.
    c. Click on **Security credentials**.
    d. Go to the **Access keys** section and note the access keys.
3. Establish Anthropic Claude access.
   Claude from Anthropic is one of the best models available in Bedrock for SQL-related tasks. By default, Claude is not available on Bedrock. You need to place a special request for Claude. Once you have access, you can try Claude in the text playground under the Amazon Bedrock service. If you are in the us-east-1 region, this must take you to https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/text-playground.
4. Enable the Hue SQL AI Assistant as follows:
    a. Log in to the CDP Management Console as an Administrator.
    b. Go to **Environments** and select your environment.
    c. Go to the **Data Lake** tab and click on the CM URL to open Cloudera Manager.
    d. Go to **Clusters** > **Hue service** > **Configuration** and select add the following lines in the **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini** field:

```
[aws]
      [[bedrock_account]]
            access_key_id='[***ACCESS-KEY***]'
            secret_access_key='[***SECRET-KEY***]'
            region='us-east-1'
[desktop]
      [[ai_interface]]
            service='bedrock'
            model='claude'
```

5. Click **Save Changes**.

Service and model-related configurations for setting up the SQL AI Assistant in DataHub

You can configure the AI services and models you want to use by going to **Cloudera Manager** > **Clusters** > **Hue service** > **Configurations** > **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini** and adding the following lines:

```
[desktop]
      [[ai_interface]]
            [***CONFIG-KEY1***]='[***VALUE***]'
            [***CONFIG-KEY2***]='[***VALUE***]'
      [[semantic_search]]
            [***CONFIG-KEY1***]='[***VALUE***]'
            [***CONFIG-KEY2***]='[***VALUE***]'
```

Specify the service and model-related configurations under the [[ai_interface]] section as listed in the following table:

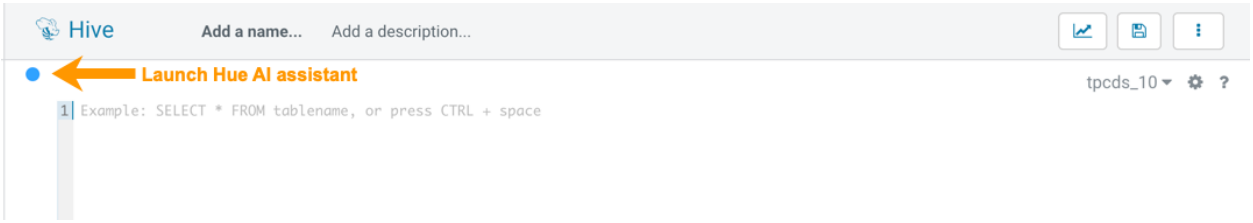
AI interface config key	Description
service	API service to be used for AI tasks. AI is disabled when a service is not configured. For example, `azure`, `openai`, `bedrock`, and `ai_assistant`.
trusted_service	Indicates whether the LLM is trusted or not. Turn on to disable the warning. The default value is "`True`".
model	The AI model you want to use for AI tasks. For example, `gpt`, `llama`.
model_name	The fully qualified name of the model to be used. For example, `gpt-3.5-turbo-16k`.
base_url	Service API base URL.
add_table_data	When enabled, sample rows from the table are added to the prompt. The default value is "`True`".
table_data_cache_size	Size of the LRU cache used for storing table sample data.
auto_fetch_table_meta_limit	Number of tables to load from a database, initially.
token	Service API secret token.
token_script	Provides a secure way to get the service API secret token.

Specify the semantic search-related configurations used for RAG under the [[semantic_search]] section, as listed in the following table:

Semantic search config key	Description
relevancy	The technology you want to use for semantic search. Acceptable values are `vector_search` or `vector_db`.
embedding_model	The model you want to use for data-embedding. This must be compatible with SentenceTransformer.
cache_size	Size of the LRU cache used for storing embedding.

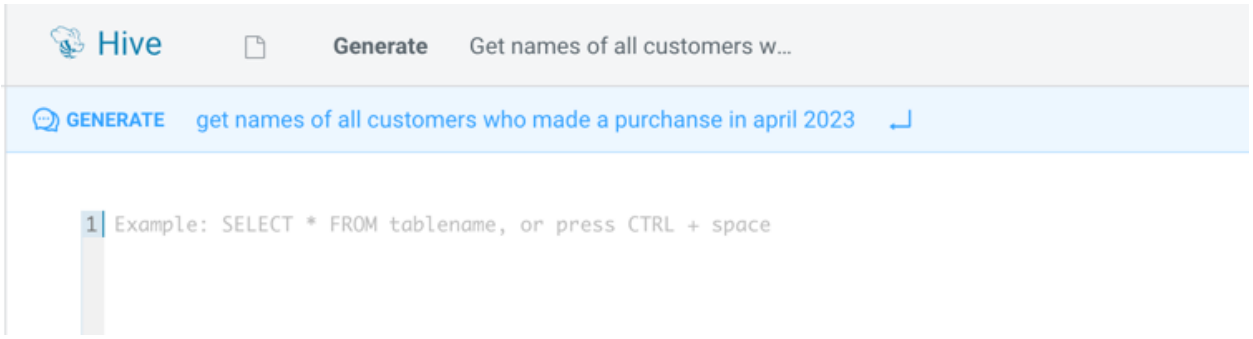
How to use the SQL AI Assistant?

To launch the Hue AI assistant, click the blue dot on the Hue web interface interface as shown in the following image:



Generating SQL from NQL

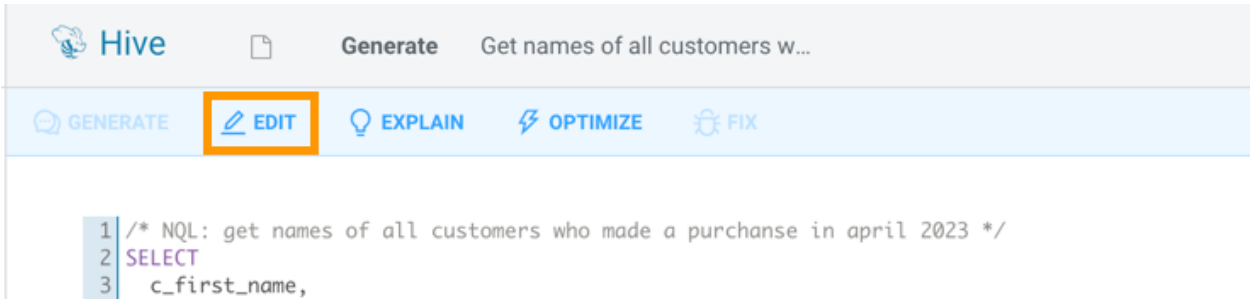
Click **GENERATE** and type the query in natural language as shown in the following image:



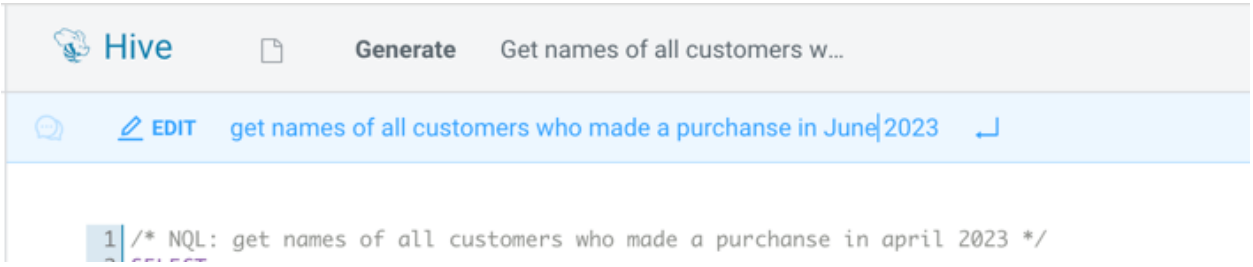
The SQL query is generated as shown in the following image. You can insert it into the editor by clicking **Insert** or you can copy the query into the editor.

Editing the query in natural language

Click **Edit** as shown in the following image:

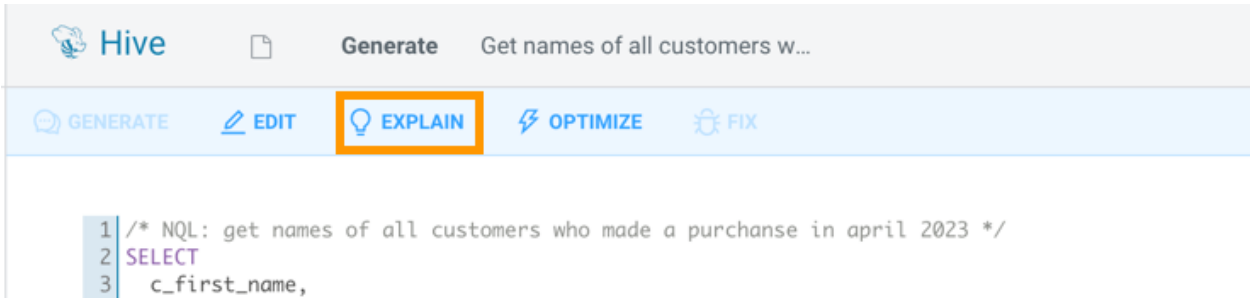


Make the required changes and press enter:

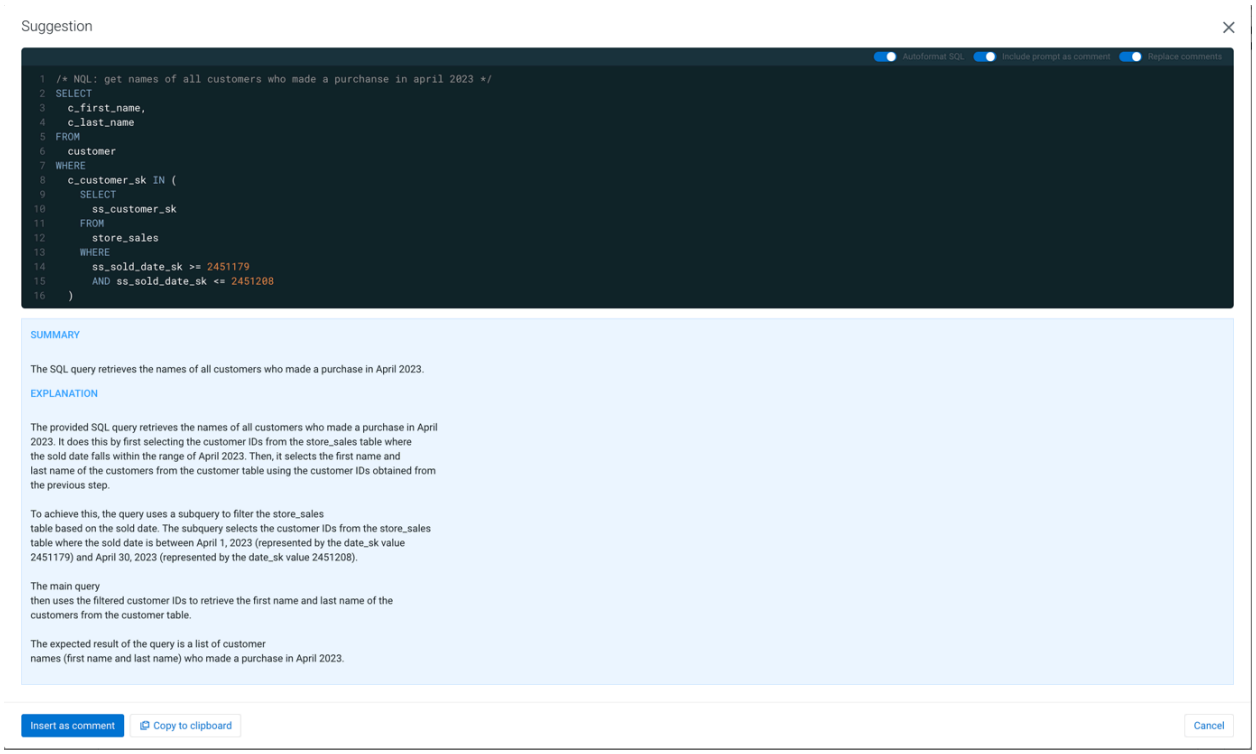


Getting an explanation of a SQL query in natural language

To understand complex queries in natural language, click **EXPLAIN** as shown in the following image:

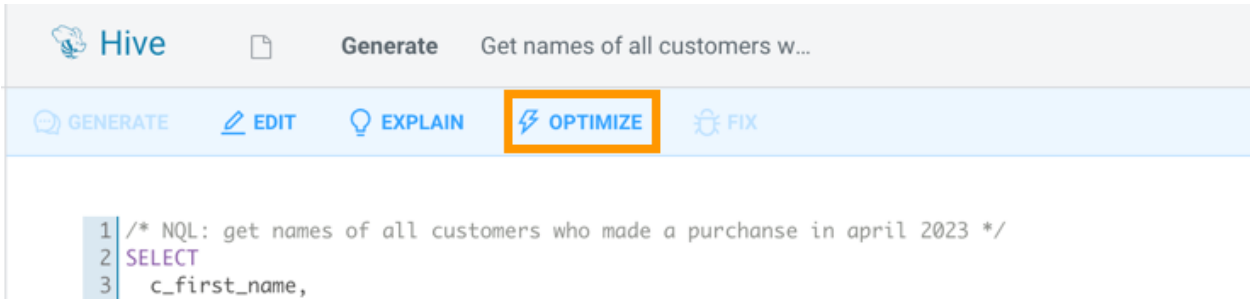


The LLM generates the explanation of the SQL query as shown in the following image:

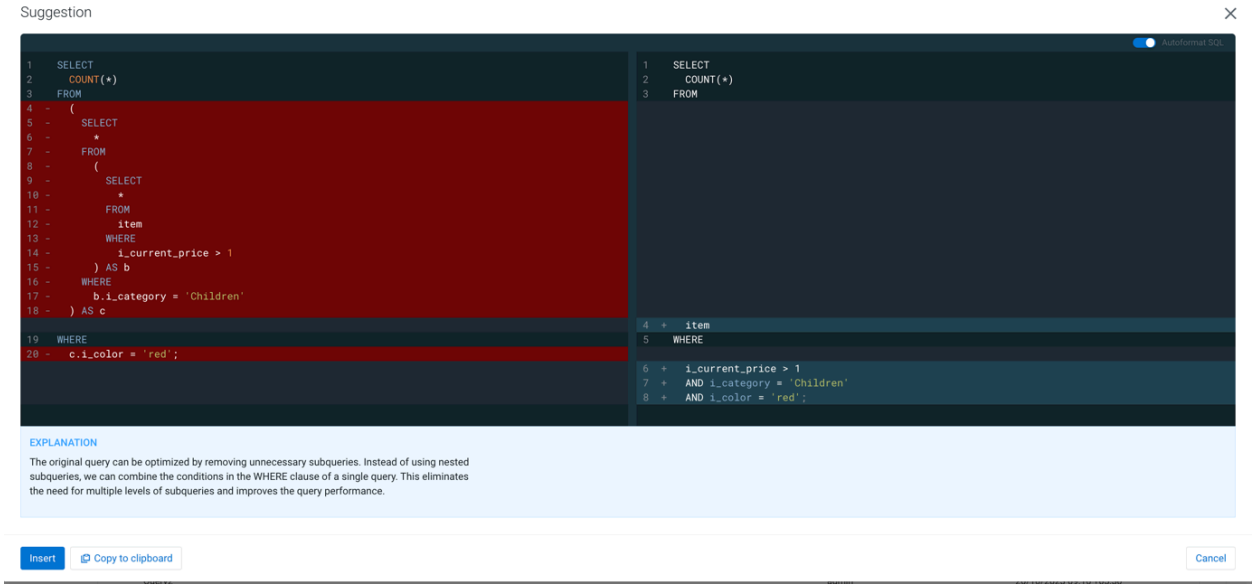


Optimizing a query

Click **OPTIMIZE** to improve an existing query as shown in the following image:

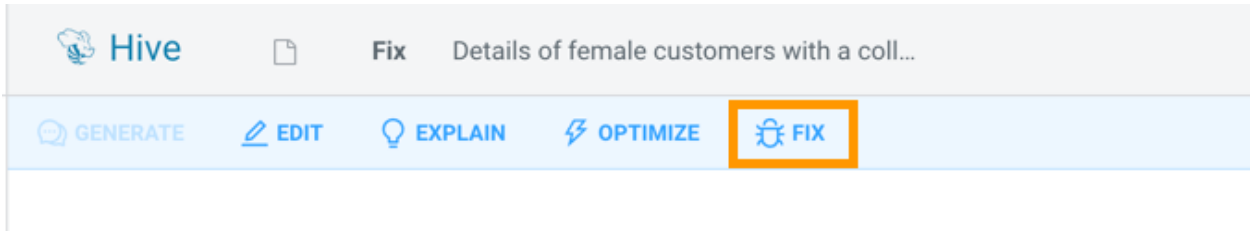


The following example shows how Hue identifies issues, optimizes a query, and provides an explanation:

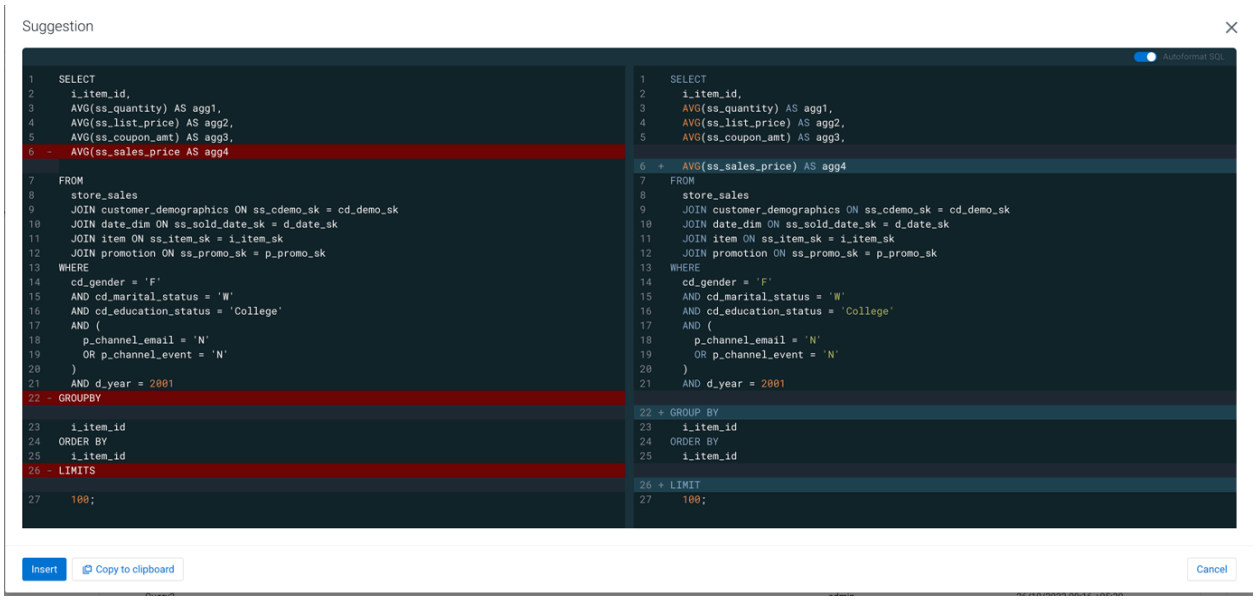


Fixing a SQL query

Click **FIX** to correct any syntax errors present in your SQL statement causing the query to fail as shown in the following image:



For example:



Limitations

- **Non-deterministic**: LLMs are non-deterministic, which means you cannot guarantee the same output for the same input every time, and it can lead to different responses to **similar** queries.
- **Ambiguity**: LLMs may struggle to handle ambiguous queries or contexts. SQL queries often rely on specific and unambiguous language, but LLMs can misinterpret or generate ambiguous SQL queries, leading to incorrect results.
- **Hallucination:** In the context of LLMs, hallucination refers to a phenomenon where these models generate text or responses that are incorrect, nonsensical, or fabricated. Occasionally you might see incorrect identifiers or literals in the response.