SQL AI Assistant in Data Warehouse Public Cloud (Preview)

Date published: 2023-11-20
Date modified: 2023-11-28

Legal Notice

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

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. SQL AI assistant is available with the Hue image version 2023.0.16.0 and higher on the public cloud. Both Hive and Impala dialects are supported.

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

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

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

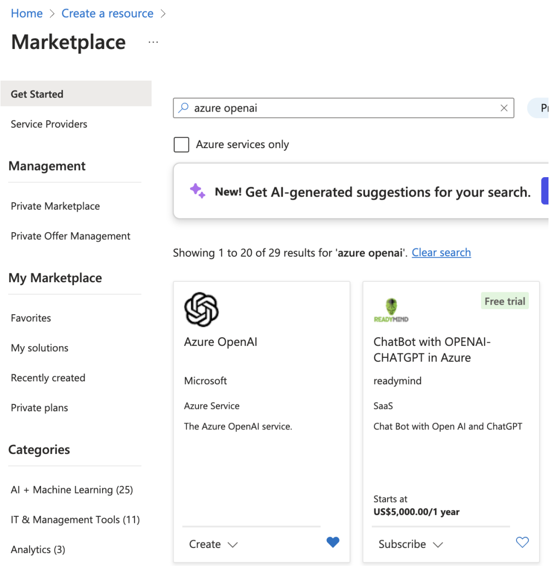
As an administrator, you must obtain clearance from your organization's infosec team to make sure it is safe to use the SQL AI Assistant because some of the table metadata and data, as mentioned in the previous section, is shared with the LLM.

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.



Home > Create a resource >

Marketplace

azure openai

□ Azure services only

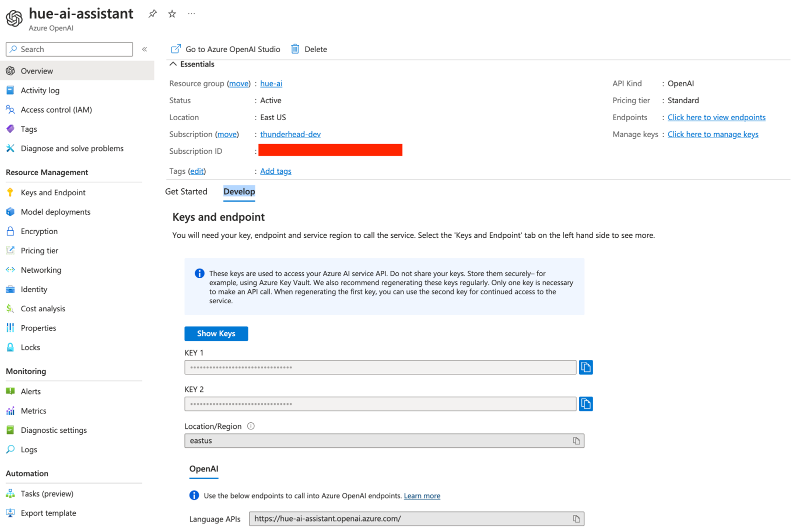
New! Get AI-generated suggestions for your search.

Showing 1 to 20 of 29 results for 'azure openai'. Clear search

Azure OpenAI
Microsoft
Azure Service
The Azure OpenAI service.

ChatBot with OPENAI-CHATGPT in Azure
readymind
SaaS
Chat Bot with Open AI and ChatGPT
Starts at US$5,000.00/1 year

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:



hue-ai-assistant
Azure OpenAI

Essentials

Resource group (move) : hue-ai
Status : Active
Location : East US
Subscription (move) : thunderhead-dev
Subscription ID : [redacted]
Tags (edit) : Add tags

API Kind : OpenAI
Pricing tier : Standard
Endpoints : Click here to view endpoints
Manage keys : Click here to manage keys

Keys and endpoint

You will need your key, endpoint and service region to call the service. Select the 'Keys and Endpoint' tab on the left hand side to see more.

These keys are used to access your Azure AI service API. Do not share your keys. Store them securely– for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

KEY 1: ••••••••••••••••••••••••••••••••

KEY 2: ••••••••••••••••••••••••••••••••

Location/Region: eastus

Use the below endpoints to call into Azure OpenAI endpoints. Learn more

Language APIs: https://hue-ai-assistant.openai.azure.com/

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 in CDW as follows:
   a. Log in to the Data Warehouse service as DWAdmin.
   b. Go to the Virtual Warehouse tab, locate the Virtual Warehouse on which you want to enable this feature, and click **Edit**.
   c. Go to **CONFIGURATIONS** > **Hue**, select **hue-safety-valve** from the **Configuration files** drop-down menu, and add the following lines under the [desktop] section:

```
[desktop]
```

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

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

d. Click **Apply Changes**.

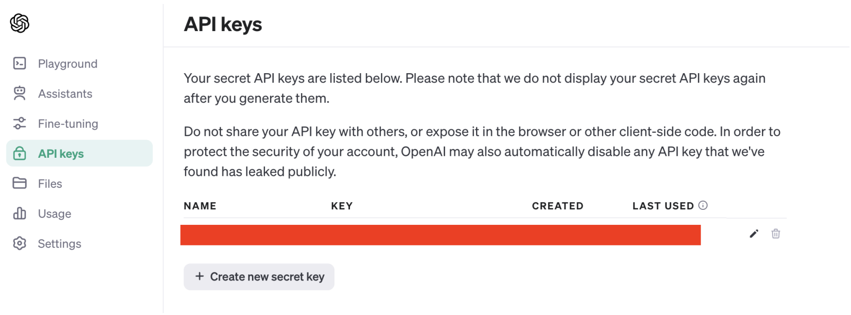
Configuring the SQL AI Assistant on the OpenAI Service

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.



API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED
[redacted]			

3. Enable the Hue SQL AI Assistant in CDW as follows:
   a. Log in to the Data Warehouse service as DWAdmin.
   b. Go to the Virtual Warehouse tab, locate the Virtual Warehouse on which you want to enable this feature, and click **Edit**.
   c. Go to **CONFIGURATIONS** > **Hue**, select **hue-safety-valve** from the **Configuration files** drop-down, and add the following lines under the [desktop] section:

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

   d. Click **Apply Changes**.

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

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 in CDW as follows:
    a. Log in to the Data Warehouse service as DWAdmin.
    b. Go to the Virtual Warehouse tab, locate the Virtual Warehouse on which you want to enable this feature, and click **Edit**.
    c. Go to **CONFIGURATIONS** > **Hue**, select **hue-safety-valve** from the **Configuration files** drop-down menu, and add the following lines:

```
[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 **Apply Changes**.

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

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

You can configure the AI services and models you want to use by going to **CONFIGURATIONS** > **Hue** > **hue-safety-valve** 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:

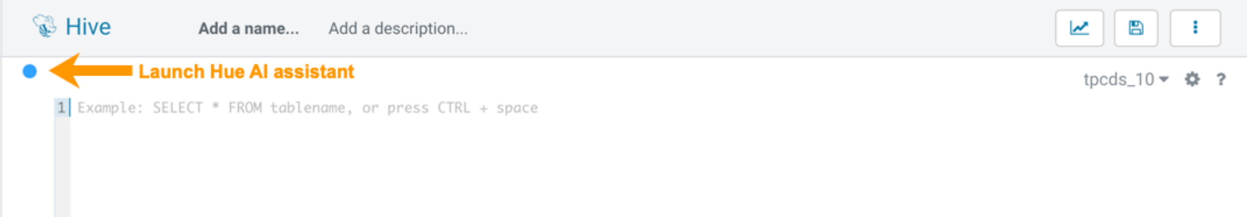
AI interface config key	Description
service	API service to be used for AI tasks. AI is disabled when a service is not configured. For example, `azure`, `openai`, `bedrock`, and `ai_assistant`.
trusted_service	Indicates whether the LLM is trusted or not. Turn on to disable the warning. The default value 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?

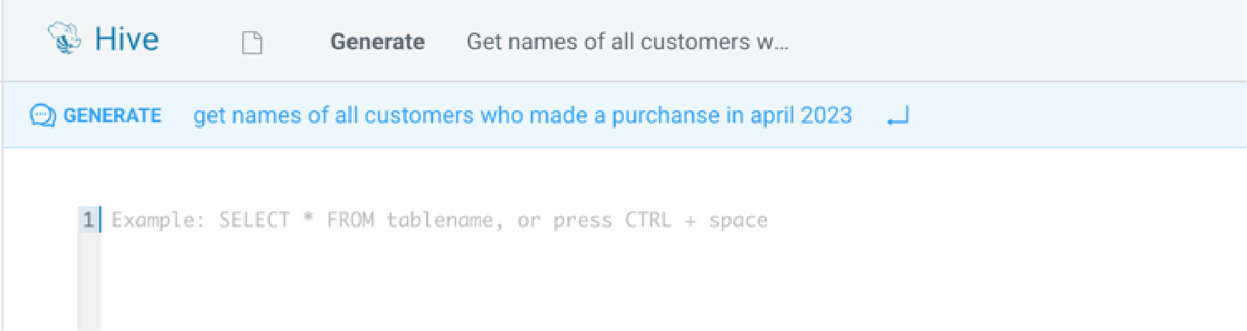
To launch the Hue AI assistant, click the blue dot on the Hue web interface interface as shown in the following image:



```
Example: SELECT * FROM tablename, or press CTRL + space
```

Generating SQL from NQL

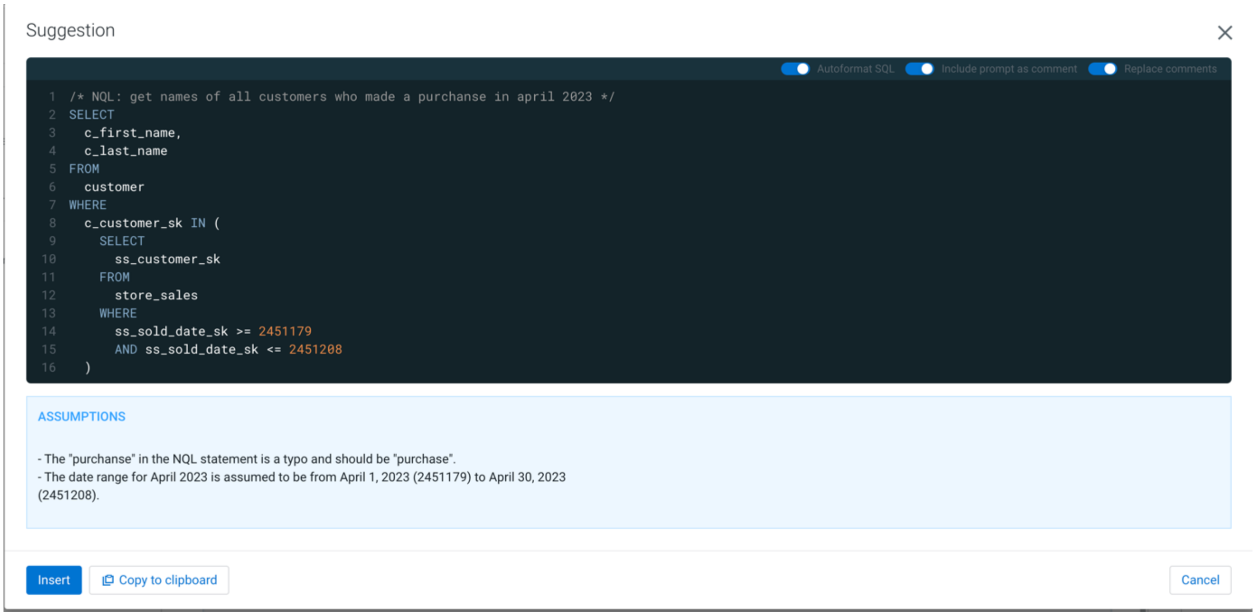
Click **GENERATE** and type the query in natural language as shown in the following image:



get names of all customers who made a purchase in april 2023

```
Example: SELECT * FROM tablename, or press CTRL + space
```

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.



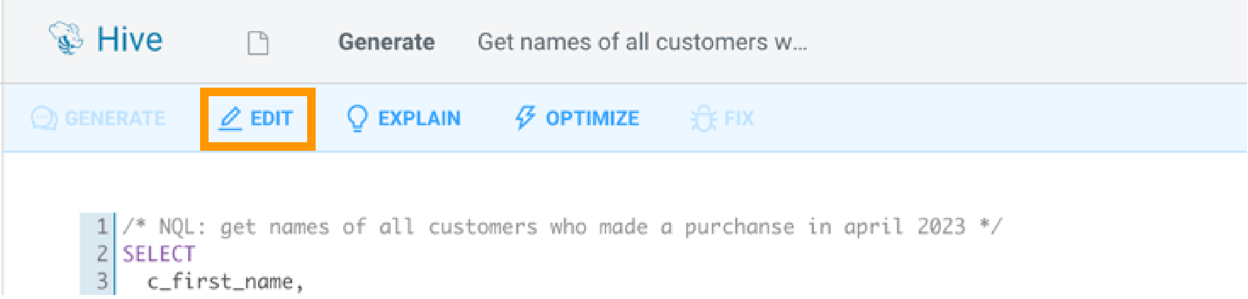
```
/* NQL: get names of all customers who made a purchanse in april 2023 */
SELECT
  c_first_name,
  c_last_name
FROM
  customer
WHERE
  c_customer_sk IN (
    SELECT
      ss_customer_sk
    FROM
      store_sales
    WHERE
      ss_sold_date_sk >= 2451179
      AND ss_sold_date_sk <= 2451208
  )
```

ASSUMPTIONS

- The "purchanse" in the NQL statement is a typo and should be "purchase".
- The date range for April 2023 is assumed to be from April 1, 2023 (2451179) to April 30, 2023 (2451208).

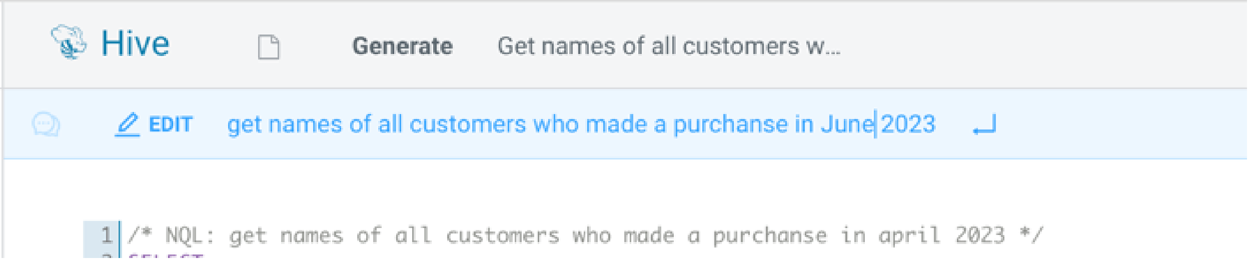
Editing the query in natural language

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



```
/* NQL: get names of all customers who made a purchanse in april 2023 */
SELECT
  c_first_name,
```

Make the required changes and press enter:



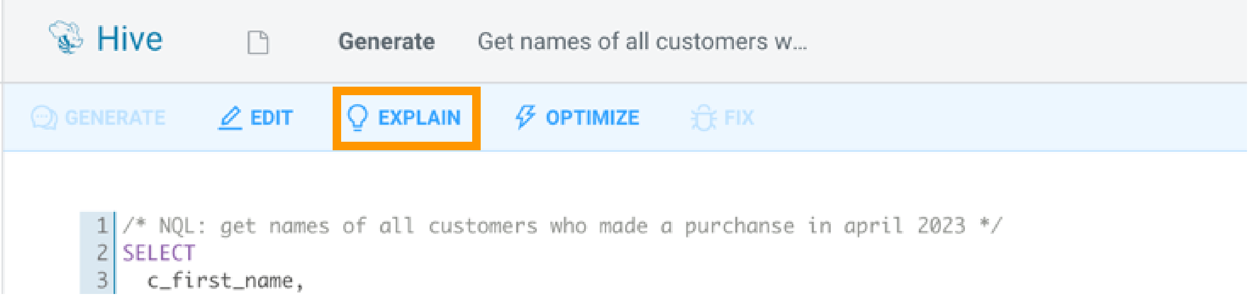
get names of all customers who made a purchase in June 2023

```
/* NQL: get names of all customers who made a purchanse in april 2023 */
```

Getting an explanation of a SQL query in natural language

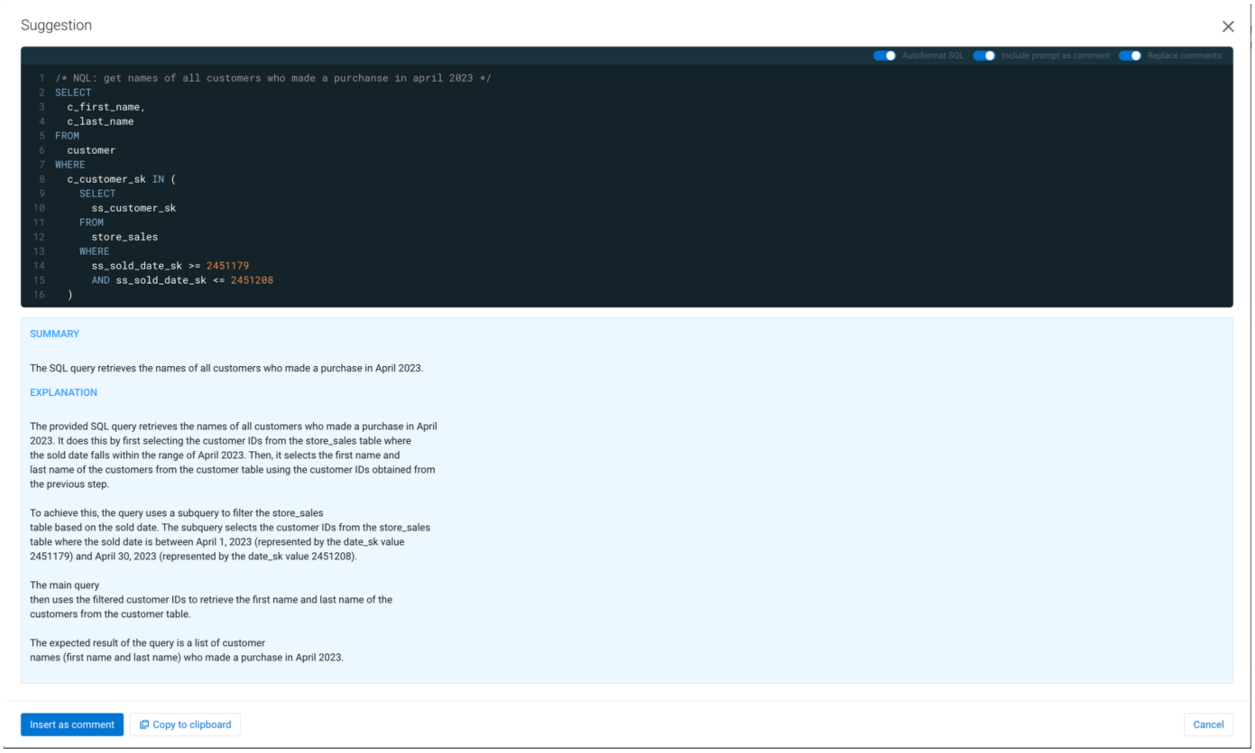
To understand complex queries in natural language, click **EXPLAIN** as shown in the following image:

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.



```
/* NQL: get names of all customers who made a purchanse in april 2023 */
SELECT
  c_first_name,
```

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



```
/* NQL: get names of all customers who made a purchanse in april 2023 */
SELECT
  c_first_name,
  c_last_name
FROM
  customer
WHERE
  c_customer_sk IN (
    SELECT
      ss_customer_sk
    FROM
      store_sales
    WHERE
      ss_sold_date_sk >= 2451179
      AND ss_sold_date_sk <= 2451208
  )
```

SUMMARY

The SQL query retrieves the names of all customers who made a purchase in April 2023.

EXPLANATION

The provided SQL query retrieves the names of all customers who made a purchase in April 2023. It does this by first selecting the customer IDs from the store_sales table where the sold date falls within the range of April 2023. Then, it selects the first name and last name of the customers from the customer table using the customer IDs obtained from the previous step.

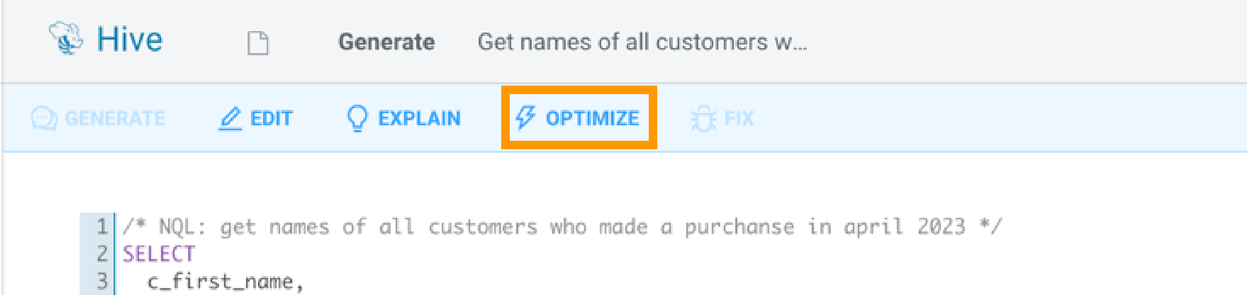
To achieve this, the query uses a subquery to filter the store_sales table based on the sold date. The subquery selects the customer IDs from the store_sales table where the sold date is between April 1, 2023 (represented by the date_sk value 2451179) and April 30, 2023 (represented by the date_sk value 2451208).

The main query then uses the filtered customer IDs to retrieve the first name and last name of the customers from the customer table.

The expected result of the query is a list of customer names (first name and last name) who made a purchase in April 2023.

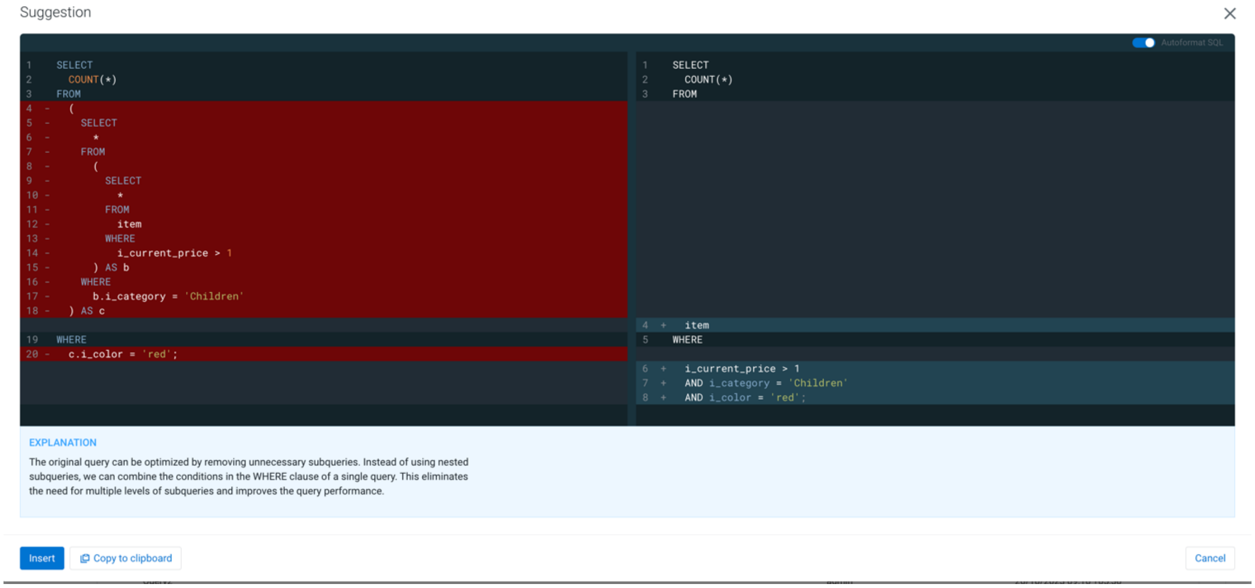
Optimizing a query

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



```
/* NQL: get names of all customers who made a purchanse in april 2023 */
SELECT
  c_first_name,
```

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



Suggestion

```
  SELECT
    COUNT(*)
  FROM
-   (
-     SELECT
-       *
-     FROM
-       (
-         SELECT
-           *
-         FROM
-           item
-         WHERE
-           i_current_price > 1
-       ) AS b
-     WHERE
-       b.i_category = 'Children'
-   ) AS c
  WHERE
-   c.i_color = 'red';
```

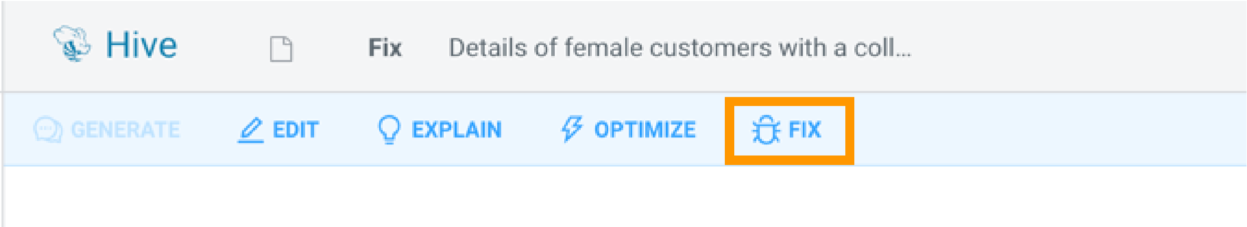
```
  SELECT
    COUNT(*)
  FROM
+   item
  WHERE
+   i_current_price > 1
+   AND i_category = 'Children'
+   AND i_color = 'red';
```

EXPLANATION

The original query can be optimized by removing unnecessary subqueries. Instead of using nested subqueries, we can combine the conditions in the WHERE clause of a single query. This eliminates the need for multiple levels of subqueries and improves the query performance.

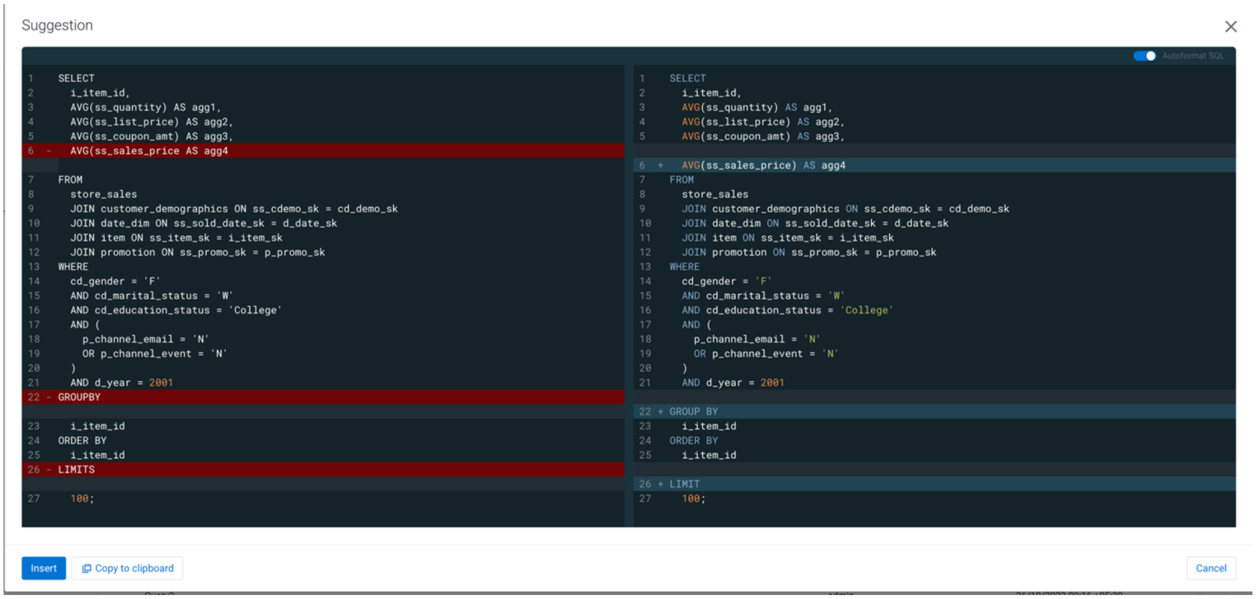
Fixing a SQL query

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:

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.



```
SELECT
  i_item_id,
  AVG(ss_quantity) AS agg1,
  AVG(ss_list_price) AS agg2,
  AVG(ss_coupon_amt) AS agg3,
- AVG(ss_sales_price AS agg4

FROM
  store_sales
  JOIN customer_demographics ON ss_cdemo_sk = cd_demo_sk
  JOIN date_dim ON ss_sold_date_sk = d_date_sk
  JOIN item ON ss_item_sk = i_item_sk
  JOIN promotion ON ss_promo_sk = p_promo_sk
WHERE
  cd_gender = 'F'
  AND cd_marital_status = 'W'
  AND cd_education_status = 'College'
  AND (
    p_channel_email = 'N'
    OR p_channel_event = 'N'
  )
  AND d_year = 2001
- GROUPBY

  i_item_id
ORDER BY
  i_item_id
- LIMITS

  100;
```

```
SELECT
  i_item_id,
  AVG(ss_quantity) AS agg1,
  AVG(ss_list_price) AS agg2,
  AVG(ss_coupon_amt) AS agg3,

+ AVG(ss_sales_price) AS agg4
FROM
  store_sales
  JOIN customer_demographics ON ss_cdemo_sk = cd_demo_sk
  JOIN date_dim ON ss_sold_date_sk = d_date_sk
  JOIN item ON ss_item_sk = i_item_sk
  JOIN promotion ON ss_promo_sk = p_promo_sk
WHERE
  cd_gender = 'F'
  AND cd_marital_status = 'W'
  AND cd_education_status = 'College'
  AND (
    p_channel_email = 'N'
    OR p_channel_event = 'N'
  )
  AND d_year = 2001

+ GROUP BY
  i_item_id
ORDER BY
  i_item_id

+ LIMIT
  100;
```

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.

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.