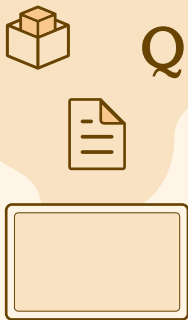


This document is intended to describe some of the issues in QA testing and where AI can help resolve them. It gives you a glimpse into how Datagaps is implementing our AI Copilot to help improve the ease and effectiveness of Automated Testing.

As we see rapid advancement in Artificial Intelligence, especially in the sub-space of Large Language Models, we also see a lot of industries rushing to create integration and automation systems with this new technology. The Data Quality space has always used AI as part of its anomaly detection systems, whether Metric or Profiling Based. The next pillar that has been added is to ease the manual pain of understanding, creating, and translating test cases.

With AI Integration, many pain points QA testers and developers face can be resolved. But first what about privacy?

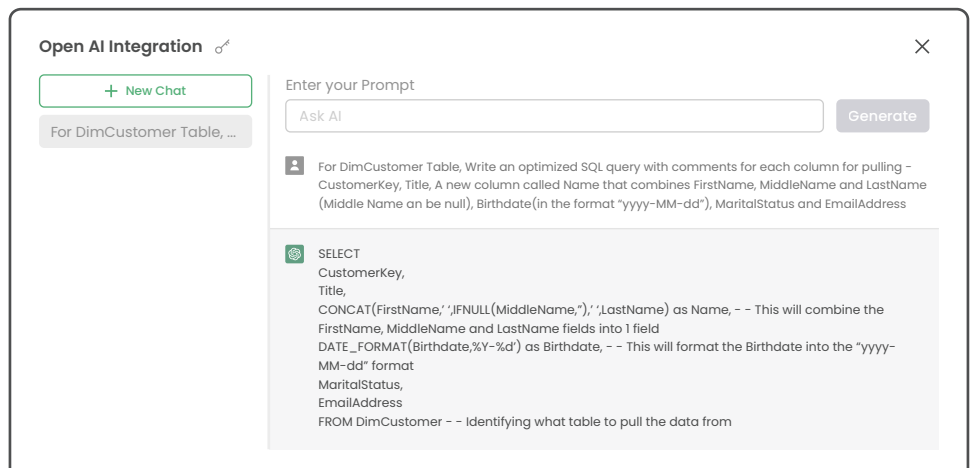


We understand privacy is a top priority when getting AI assistance involved, and only compliant queries and data are to be sent to these systems. Therefore, we have created internal isolation within the Suite with respect to the OpenAI Copilot. DataOps Copilot tools push no metadata, data, test cases, or information ahead.

The application now comes with an OpenAI Assistan Copilot. This can be found in the SQL and Code Components. To use the Copilot, an OpenAI key must first be administered in the "Integrations" Section in the "Admin" Tab.

AI Generated SQL Queries

A multitude of QA testers and developers have expressed their frustrations when working with SQL queries



Open AI Integration 🔗 ✕

For DimCustomer Table, ...

Enter your Prompt

AI For DimCustomer Table, Write an optimized SQL query with comments for each column for pulling - CustomerKey, Title, A new column called Name that combines FirstName, MiddleName and LastName (Middle Name can be null), Birthdate(in the format "yyyy-MM-dd"), MaritalStatus and EmailAddress

```
SELECT
CustomerKey,
Title,
CONCAT(FirstName,'IFNULL(MiddleName,),'LastName) as Name, -- This will combine the
FirstName, MiddleName and LastName fields into 1 field
DATE_FORMAT(Birthdate,'%Y-%d') as Birthdate, -- This will format the Birthdate into the "yyyy-
MM-dd" format
MaritalStatus,
EmailAddress
FROM DimCustomer -- Identifying what table to pull the data from
```



1. Using SQL is a common requirement for QA testers. Often generating SQL queries becomes time-consuming and slows the testing process.

- This could be database connection specific, such as pulling a large aggregated table from a JDBC connection.
- Working with complex functions, aggregations, and joining conditions, especially in conjunction with each other, usually comes with a steep learning curve.
- Poorly formed SQL queries can also cause issues on the server or machine that runs them. Thus, optimization from the start is very important.
- **Thus, Datagaps Copilot can be used to easily create these queries. The application can run individual nodes; therefore, testing and developing these queries is also quick.**
- **For example, we can create a query which is a well-optimized query that creates a formatted dataset with comments.**

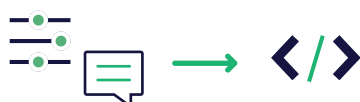
2. Understand existing SQL Queries

- Very often QA testers and developers are tasked with updating, fixing, or validating an existing SQL query.
- A well-maintained data dictionary and ETL documentation can only take one so far as they figure out which functions are used and why.
- **Here, the Copilot can help understand exactly what the query is about, the concerns it sees in it, and how it achieves its transformation logic.**



3. Good code practice

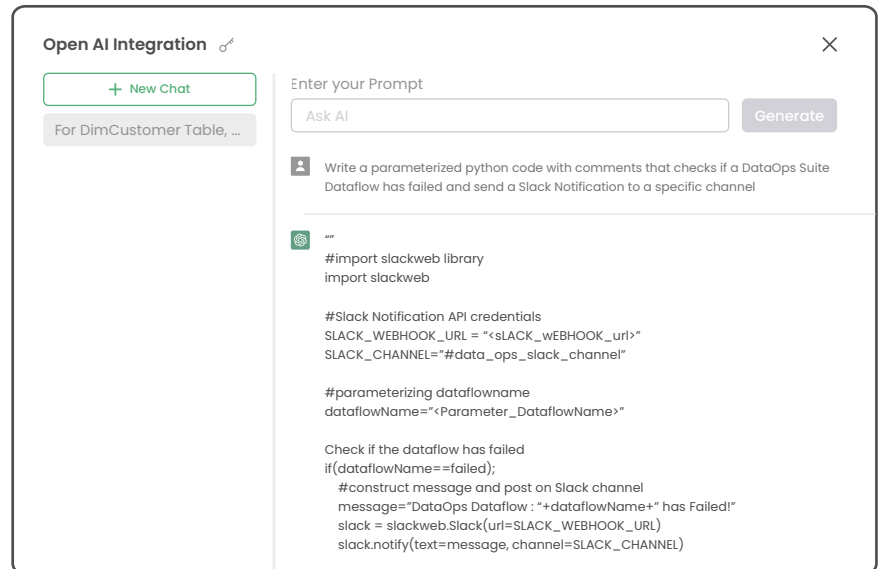
- **A final point that we see internally, is that the application helps with the creation of clean code with standardized annotations and parameterization.**
- **The strain of the addition of comments is also trivialized by the assistance.**





Easy Coding and Integration

While the same pointers we mentioned for SQL applies for the other coding languages supported by the Suite such as Scala - Java, Python, DB Scripts and more, the focus of this section will be ease of integration and automation.



1. A rehash of repeating pointers. The assistance helps with the following

- **Understanding of existing code.**
- **Cleaning up, Optimizing, and Parameterizing the code.**
- **Helping in getting started with a language a user might be unfamiliar with.**

2. Translation of Transformation Logic in Migration Validation or Architecture Changes

- While migrating from systems, a good bit of transformations also have to be transferred to the new system. This usually involves converting or translating the code that works from one system to the other. A direct code copy rarely works due to inherent differences in compilation and processing. **A transformation that works in an On-Prem SQL Server might not necessarily work for a Cloud Snowflake system. The Copilot comes in handy here as well to help with migration of transformations and their validations which is part of both white-box and black- box testing.**
- Architectural Migration Validation is also a place where the Copilot can help you keep track of all the nuances that are changing from one form to the other.

A simplified example would be the changes in transformations required when migrating from a Data Lake architecture to a Data Mesh Architecture. While functions might not change too much, partitions, aggregation conditions, permissions, and other aspects of transformations will be altered. Here, an AI Copilot is of great use to ease the process.

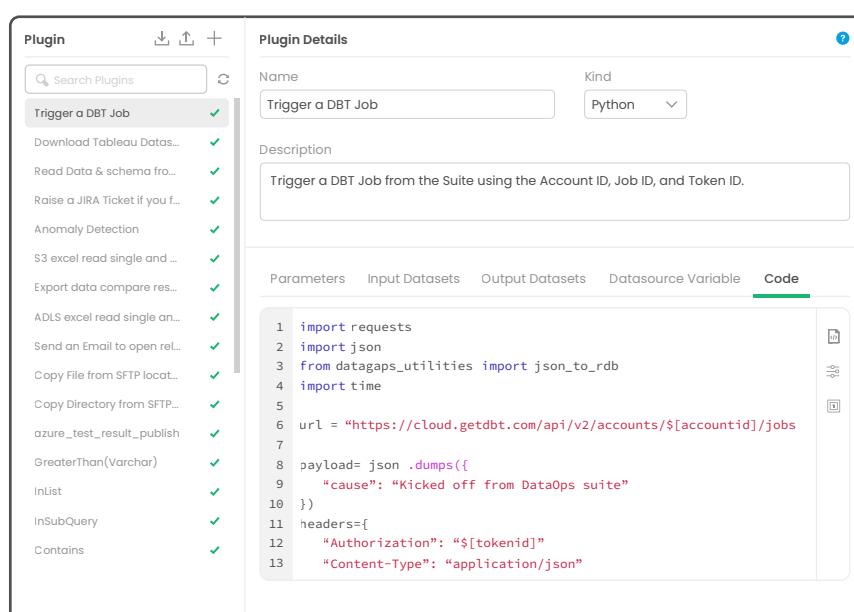




3. Ease of Integration and Automation

- The DataOps Suite is very open-ended and doesn't put any restrictions on the way the user wants to create or run test cases. The application welcomes users to integrate the platform directly with their existing toolsets and pipelines. Whether it's –

1. Automatic JIRA ticket raising for a specific use case.
2. Reports directly sent as a Slack notification.
3. Triggering DBT jobs as part of the DQM pipeline
4. Creating an operator for niche use case
5. Creating test case templates to be imported to the Suite
6. Creating a CSV file out of streaming data
7. or any varied use cases that is needed by your team.



- The application supports user code bases via plugins. These can be created by end users, who define the parameters, datasets and code set in a single repository to be managed. These are then used correspondingly by end QA testers or Developers where the only input needed is the previously defined parameters and datasets. **The code for all the examples being showcased here has been developed with the AI Copilot.**
- **The Plugin functionality along with the AI assistance, have helped reduce our time-to-delivery for new features by a factor of 5. We expect the same for our client spaces who create more intricate plugins for their niche use-cases.**

On the Horizon – What's next?

While we move ahead with further integrations with AI and LLMs, we do want to ensure that compliances are being met and privacy boundaries are being adhered to.



We are preparing the following features for our upcoming releases -



Contextual Processing

When you provide your query to the AI Copilot, a masked version of the associated details can also be sent to help the model yield better responses and results. This is user-controlled, and the degree of masking will also be user-defined.



Automated Data Rules Creation

With the permission of the user, masked or unmasked metadata and context will be sent to the AI Copilot, which can then proceed to create a plethora of test cases that can be directly imported into the Suite. Thus, from an end-user perspective, users only have to define the datasets they want to validate and create test cases for.



Documentation Based Automation

A good practice with Data Pipelines is to maintain data dictionaries and transformation spreadsheets. These are especially useful when migrating architecture or systems. This new feature will allow the user to upload their non-standardized transformation documentation, which will be used by the Suite in conjunction with the AI Copilot to create the transforms and its associated test cases automatically. The end result will be Data Quality Assurance and ETL Testing automated to a huge extent with only the documentation.

