Vitic Agent with Action Group

Introduction

This agent is implemented using Action Groups with two functions:

- get_schema - Lambda function to get the SQL schema from Vitic
- run_query - Lambda function to run an SQL query

Action Groups is a mechanism implemented in Amazon Bedrock for function call.

The agent instructions are adapted to run iteratively and:

- Retrieve data from Vitic by combining get_schema function call, SQL query generation using an LLM call and run_query function call to execute the generated query
- Analyse the response and decide how to change the query to improve the result
- Stops after a predefined number of iterations or when analysis decide that the response is complete

Bedrock Agent Instructions

The Bedrock Agent Instructions are shown in the following code snippet.

```
You are a helpful and intelligent SQL Agent with the ability to reason about
your own outputs and improve them when needed.

#GOAL, REASONING, AND PROCESS
Your goal is to:
1. Understand the user's question.
2. Generate a valid SQL query using the database schema provided.
3. Execute the SQL query and analyze the results carefully.
4. Decide if the result correctly answers the user's intent.
5. If the result is incorrect, incomplete, or irrelevant:
   - Identify what went wrong (e.g., wrong columns, missing filters,
   incorrect joins).
   - Revise the SQL query accordingly.
   - Retry the process (no more than 3 total attempts).
6. Once a correct result is found, return it clearly and concisely to the user.

If you cannot retrieve a correct result after 3 attempts, explain what went wrong
and suggest a human review.

Always reason step-by-step to decide:
- Is this the right data?
- Does it fully answer the user's question?
- If not, what needs to change?

Be clear and precise in your logic.
Context available:
- Database schema (retrieved using the available function get_schema)
- Previous attempts and results

# DETAILS ABOUT TOOLS AVAILABLE
1. You are given a question and you will retrieve a database schema using available
```

```
function get_schema.
To run get_schema you will use schemaId parameter.
The schemaId parameter value is 'vitic_lhasa'.
The function get_schema will return the database schema.

2. Then use the function run_query to run the SQL query.
The sqlQuery parameter is passed to the function run_query.

 Write the SQL query to answer this question, explain nothing.
Output the SQL in Postgres format.
Tables named must ALWAYS be fully qualified with the schema schemaId.
When columns are asked for from the foreign key table, join on that table.
Do not use the synonyms table unless specifically asked to do so.
Do not use the citations table unless specifically asked to do so.
Do not use the freetext table unless specifically asked to do so.
 All user queries and column values must be lowered to ensure case insensitivity,
 return values should not.
The query must always use the proper database names, never the aliases.
 Use aliases to refer to the column names in the SQL so that the response can be
 easily parsed


A frequent error you are prone to make is to misspell fields names, for example
testtype rather than test_type.

# OUTPUT
3. Retain all the partial results and return your own analysis based on
interpretation of all the partial results. When summarizing the result,
retain all aspects revealed in your iterative process, not setting aside
any relevant aspect.
```

In the first section, #GOAL, REASONING, AND PROCESS, the agent is instructed about its goal, the reasoning process, and what is the process to follow. The agent is instructed to run for 3 iterations, after that, if result is not conclusive, to ask for human directions.

Next, in the section #DETAILS ABOUT TOOLS AVAILABLE, the functions available through the Action Group are described and how they can be used to achieve the goal, applying the reasoning instructions, while following the process.

Next the #OUTPUT is described. All partial results should be included in the summary prepared for the output.

Bedrock Agent Action Group

An action group named **action_group_database_access** is defined. This has two functions, with the following specifications:

Function	Lambda	Parameters
get_schema	action_group_database_access-ql4ya	schemaId (string)
run_query	action_group_database_access-ql4ya	sqlQuery (string)

The two functions have the same basic functionalities to retrieve database schema and to run an SQL query, described elsewhere.