Gold Guiding Documentation

Greetings adventurer,

You have been chosen to embark on a grand quest to build out a dimensional gold layer using Apache Spark and pySpark language. This is a great honor, and a task that will require all of your cunning and skill.

To begin, you must initialize the dimensional model in an Azure Spark notebook. You will need:

* [**sql**](https://spark.apache.org/sql/): to initialize database using SQL and drop tables if exist during initial load – please note it is forbidden to use SQL for anything else. Useful links: [DROP TABLE - Azure Databricks - Databricks SQL | Microsoft Learn](https://learn.microsoft.com/en-us/azure/databricks/sql/language-manual/sql-ref-syntax-ddl-drop-table) and [CREATE DATABASE - Azure Databricks - Databricks SQL | Microsoft Learn](https://learn.microsoft.com/en-us/azure/databricks/sql/language-manual/sql-ref-syntax-ddl-create-database).
* [forPath](https://docs.delta.io/0.4.0/api/python/index.html): to load Delta tables from the storage
* [**alias**](https://spark.apache.org/docs/3.1.2/api/python/reference/api/pyspark.sql.DataFrame.alias.html): this function is used to give an alias or alternate name to a DataFrame or a column within a DataFrame.
* [**distinct**](https://spark.apache.org/docs/3.1.2/api/python/reference/api/pyspark.sql.DataFrame.distinct.html): This function is used to remove duplicates from a DataFrame. It returns a new DataFrame with distinct rows based on all columns. You can also specify a subset of columns to be used in identifying duplicate rows.
* [**withColumnRenamed**](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.withColumnRenamed.html): This function is used to rename a column in a DataFrame. It takes two arguments: the current column name and the new column name.
* [**withColumn**](https://spark.apache.org/docs/3.1.3/api/python/reference/api/pyspark.sql.DataFrame.withColumn.html): This function is used to add a new column to a DataFrame or to modify an existing column. It takes two arguments: the name of the new or existing column and the expression that defines the value of the column.
* [**select**](https://spark.apache.org/docs/3.1.1/api/python/reference/api/pyspark.sql.DataFrame.select.html): This function is used to select one or more columns from a DataFrame. You can pass either column names or expressions as arguments to this function.
* [**monotonically\_increasing\_id**](https://spark.apache.org/docs/3.1.3/api/python/reference/api/pyspark.sql.functions.monotonically_increasing_id.html): This function generates unique and monotonically increasing IDs for rows in a DataFrame. This function does not guarantee unique IDs across multiple Spark sessions or multiple executions of the same Spark job.
* [**join**](https://spark.apache.org/docs/3.1.2/api/python/reference/api/pyspark.sql.DataFrame.join.html): This function is used to join two DataFrames based on a common column or set of columns. There are several types of joins in Spark, such as inner join, outer join, left join, and right join.
* [**Delta merge**](https://docs.delta.io/latest/delta-update.html#upsert-into-a-table-using-merge): This function will allow you to merge two DataFrames based on a common key.

But your quest is not yet complete! You must also build out your fact table. To do this, you must join data with dimensions to get keys you created for your dimensions. This will allow you to link your fact table to your dimensions and create a complete, cohesive data model.

Remember, adventurer, this quest is not for the faint of heart. You must be diligent, focused, and skilled in the ways of Azure Synapse mapping dataflows. But fear not, for I will be with you every step of the way.