# Guidelines for ETL data staging and production

* Author: Hamza Tariq

## Guidance 1

Scenario 1:

For efficient development and deployment processes, it is recommended to conduct all initial development within the Development Environment (DevEnv). This workspace should be seamlessly integrated with GIT for code versioning and history tracking. After the completion of development, thorough testing of data and Power BI (PBI) components should be carried out within the DevEnv. Once validated, the code should be pushed to the designated GIT repository.

Upon successful testing in the DevEnv, the Production Environment (ProdEnv) workspace is then tasked with pulling the verified code from GIT. It is important to note that due to certain restrictions within DevEnv, the movement of data or databases across environments may not be feasible. Consequently, data Extract, Transform, Load (ETL) processes will be executed twice – once on the ProdEnv and again on the DevEnv. This approach will require re-validations of data in the Prod environment after the repeat execution.

Scenario 2:

In the scenario where the development and production environments share the same infrastructure but maintain separate databases (e.g., data staging/dev and Prod), a systematic approach is recommended.

Initiate all primary development within the dev/data staging database, ensuring the workspace is integrated with GIT for effective code versioning and history tracking. Following the completion of the initial development phase, a Python/Pyspark script should be employed in the Prod database to pull the thoroughly tested data from the dev/data staging database. The resulting data can then be distributed to the designated channels.

Similar to Scenario 1, the Prod database code should also be integrated with GIT for version control and collaborative development. This methodology ensures a streamlined workflow, allowing for efficient code development, testing, and deployment while maintaining the necessary separation between development and production environments.

# 

## Guidance 2

The initial development and data storage processes are ideally conducted in the development (dev) environment. Subsequently, the data needs to be migrated to the production (prod) environment, and various techniques, such as data pipelines and Python scripts, can be employed for this purpose.

In the existing AMA Ecomm project, two different cases have been implemented. In the first case, the dev environment's ETL process retrieves data from the data storage (Blob, SharePoint), performs necessary transformations, and stores the output within the same environment. Following validations, the dev environment's ETL is moved to the prod environment using Git. In the prod environment, the ETL process fetches data from the same data storage. This scenario involves the ETL running twice in two distinct environments due to limitations in interaction between the two clusters.

In the second case, there is a singular environment with two different databases: the first is referred to as "Dev" or "Data Staging," and the second as "Prod." In the dev environment, the ETL process retrieves data from the data storage (Blob, SharePoint), performs necessary transformations, and stores the data within the same database. After validations, the Prod database pulls the data from the Dev database using either data pipelines or Python scripts.