Target Path:

The target path in dbt refers to the directory where the compiled files, such as compiled models and tests, are written. When you run dbt commands like **dbt run**, **dbt compile**, or **dbt test**, dbt generates and stores the compiled artifacts in the target path directory.

The target path serves as the output location for the compiled files, which are the result of transforming your raw data and applying your defined models and transformations. These compiled files are typically SQL queries or code that can be executed against your data warehouse or database.

By default, dbt uses a predefined target path, which may vary depending on your dbt installation. However, you have the flexibility to specify a custom target path to control where these compiled files are stored.

The target path allows you to separate the compiled files from your source code or raw data, providing a clear separation between your development environment and the generated output. It also enables you to organize and manage the compiled files independently, making it easier to deploy, share, or version control your dbt projects.

When specifying a custom target path, make sure to choose a directory that has appropriate permissions and sufficient storage capacity. It's also a good practice to specify a target path that is different from your source code directory to avoid mixing the compiled artifacts with your source files.

Overall, the target path is a crucial component in dbt that determines where the compiled files are written, allowing you to leverage the output of your dbt transformations and analytics processes effectively.

target path precedence:

1. CLI flag: If you specify the **--target-path** flag in dbt commands, it takes the highest precedence. For example:

dbt run --target-path=/path/to/target

1. Environment variable: If the **--target-path** flag is not provided, dbt checks for the presence of an environment variable named **DBT\_TARGET\_PATH**. If the variable is set, its value is used as the target path. For example:

export DBT\_TARGET\_PATH=/path/to/target

**dbt run**

1. **dbt\_project.yml** file: If neither the **--target-path** flag nor the environment variable is provided, dbt falls back to the **target-path** specified in the **dbt\_project.yml** file. The **dbt\_project.yml** file allows you to set a default target path that will be used if no other target path is explicitly specified. For example:

# dbt\_project.yml

...

target-path: /path/to/target

By following this precedence order, you can override the default target path specified in the **dbt\_project.yml** file by using the **--target-path** flag or environment variable, providing flexibility in selecting different target paths for your dbt commands.