<https://github.com/tsyscom5112/gppd-dbt>

gppd-dbt # コメントにて各フォルダの目的を簡潔に記載。

├─ analysis # モデルに適合しないsql格納ディレクトリ

|

├─ data # ローカル生データ格納ディレクトリ

|

├─ macros # マクロ関数格納ディレクトリ

| └- utils

|

├─ models # モデルファイル格納ディレクトリ

| ├─ marts # Mart用モデルファイル格納ディレクトリ

| │ ├─ marts.yml

| │ ├─ intermediate # intermediate用モデルファイル格納ディレクトリ

| │ ├─ core # core用モデルファイル格納ディレクトリ

| │ └- dim\_common # dim\_cmmon用モデルファイル格納ディレクトリ

| |

| └- staging # Stage用モデルファイル格納ディレクトリ

| ├─ gsedi # GS\_EDI用モデルファイル格納ディレクトリ

| | └- stg\_revise.sql

| |

| └- tlswms # 豊物用モデルファイル格納ディレクトリ

| └- stg\_inbound\_outbound.sql

|

├─ snapshots # スナップショット保存ディレクトリ

|

├─ snowchange # snowchange関連格納ディレクトリ

|

├─ snowflake/scripts/ddl/gs-edi # rawデータ用Sql格納ディレクトリ

|

└- tests # テストファイル保存ディレクトリ

構造はこれで

Macros/utils/generate\_schema\_name.sql

に関して

“ジンジャのマクロは何度も再利用できるコードで、他のプログラミング言語の「機能」に似ており、複数のモデルにわたってコードを繰り返す場合に非常に便利です。 マクロは.sqlファイル（通常はマクロディレクトリ）で定義されます。”

と言う説明がありました。

書き方がJINJAなので難しくて理解ができませんでした。

Snowflake/scripts/ddl

のところもraw dataのものだと書いてますがちゃんと理解ができませんでした。

dbt\_project.yml

databaseと各folderのsqlで作るもののschemaとtable? View?どっちで作るのか確認

models/

staging : sql文のものをviewで作る

mart : sql文のものをtableで作る

＊モデルはコンパイル中に生成された従属グラフで定義された順序で実行されます。

＊dataの構造と作ったdataの形を確認できる

Profile.ymlはfolder移動する必要あると思う　〜/.dbt/profile.yml

{{env\_var()}} 環境で設定したものを利用、無いときdefaultのものも設定できる

{{doc()}} .md　fileの中に設定している名前でその文句を呼び出す

{{source(,)}} 文書化とモデル選択に有用なソースと現在のモデル間の依存関係を作成します。

Source keyと　その下のtable key

<https://docs.getdbt.com/docs/introduction>

<https://docs.getdbt.com/reference/dbt_project.yml>

stagingにあるsqlは大体ymlで設定しているtable名とか利用しますが、

martsにあるsqlはtable名がstagingで作ったものを利用します。