TrackMan Data Engineering Challenge: Table Dependencies Graph

Zhiwen Shi (Clark)

06/04/2023

GitHub repository for checking README.md, module and unit test.

In [1]:
```python
import os
import json

def get_queries(directory: str) -> list:
    """Get queries from the target directory

    Parameters
    ----------
    directory : str
        The directory that stores all the configuration files in
        JSON. The directory should be in the same path of the script.

    Returns
    -------
    queries : list
        A list contains all the queries from each configuration
        JSON file.

    """
    queries = []
    for filename in os.listdir(directory):
        if filename.endswith('.json'):
            file_path = os.path.join(directory, filename)
            with open(file_path, 'r', encoding='utf-8') as file:
                query = json.load(file)
                queries.append(query)
    return queries
```

In [2]:
```python
def get_table_dependency(query: dict) -> list:
    """Get table dependency from the query.

    Parameters
    ----------
    query : dict
        The query that extract from the configuration JSON file.

    Returns
    -------
    table_dependency : list
        A list contains two elements. First element is the new table
        which is created by the query. Second element is a list of
        all the tables that construct the new table. They are the
```

```
        dependencies of the new table.

    """

    table_dependency = []

    schema = query['schema']['S']
    table = query['table']['S']
    # get the name of the created table from query
    new_table = schema + '.' + table

    from_tables = []

    # Get query `from statement` according to the JSON file
    # Can be optimized if get more information about query strcutre in JSON
    try:
        from_query = query['query']['L'][0]['M']['from']['S']
    except KeyError:
        from_query = query['query']['M']['from']['S']

    from_query = from_query.split()

    # First element is the created table
    from_tables.append(from_query[0])
    for string in from_query:
        if string.lower() == 'join':
            # Dependent table is following the `join``
            table = from_query[from_query.index(string)+1]
            # Only take unique names of the dependencies
            if table not in from_tables:
                from_tables.append(table)
            # Delete current `join` to insure we check next `join` by `index()`
            del from_query[from_query.index(string)]

    table_dependency = [new_table, from_tables]

    return table_dependency
```

In [3]:

```
def print_table_dependency(dependencies_list: list) -> None:
    """Print table dependency from the list of dependencies.

    Parameters
    ----------
    dependencies_list : list
        A list contains all the created tables from the query and
        the tables they are depends on.

    Returns
    -------
    None

    """
    for table_dependency in dependencies_list:
        print('\n')
        print(table_dependency[0])
        for from_table in table_dependency[1]:
```

```
            depth = 1
            print(' '*depth*3, '|')
            print(' '*depth*3, f'|+{from_table}')
            print_dependency_of_dependency(
                from_table, dependencies_list, depth)
```

In [4]:
```
def print_dependency_of_dependency(
        check_table: str, dependencies_list: list, depth: int) -> None:
    """If the table has dependent table in the dependencies list,
       print the table dependency from the list of dependencies.

    Parameters
    ----------
    check_table: str
        The name of the table for checking the dependency.
    dependencies_list : list
        A list contains all the created tables from the query and
        the tables they are depends on.
    depth: int
        The depth of the dependency chain.

    Returns
    -------
    None

    """
    for table in dependencies_list:
        if table[0] == check_table:
            # increase the depth for indentation of the graph
            depth += 1
            for from_table in table[1]:
                print(' '*depth*3, '|')
                print(' '*depth*3, f'|+{from_table}')
                print_dependency_of_dependency(
                    from_table, dependencies_list, depth)
```

In [5]:
```
def table_dependencies_graph(directory: str) -> None:
    """Print the graph of table dependencies from the given directory

    Parameters
    ----------
    directory : str
        The directory that stores all the configuration files in
        JSON. The directory should be in the same path of the script.

    Returns
    -------
    None

    """
    table_dependencies = []
    queries = get_queries(directory)

    for query in queries:
        table_dependencies.append(get_table_dependency(query))
```

```
    print_table_dependency(table_dependencies)
```

In [6]:
```
table_dependencies_graph('tables')
```

```
crosscheck.calibration_maintenance
    |
    |+base.calibration_maintenance
    |
    |+crosscheck.calibrations
    |
    |+crosscheck.games
       |
       |+base.games
       |
       |+base.locations
       |
       |+rundown.location_history
          |
          |+rundown.location_history_rows
             |
             |+rundown.locations
       |
       |+crosscheck.tags
          |
          |+scout.tags
             |
             |+base.tags
             |
             |+dict.player_dedup
          |
          |+base.games


crosscheck.games
    |
    |+base.games
    |
    |+base.locations
    |
    |+rundown.location_history
       |
       |+rundown.location_history_rows
          |
          |+rundown.locations
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


crosscheck.tags
    |
    |+scout.tags
```


|
|+base.tags
|
|+dict.player_dedup
|
|+base.games
games.autorecalibration
|
|+crosscheck.calibrations
|
|+crosscheck.games
|
|+base.games
|
|+base.locations
|
|+rundown.location_history
|
|+rundown.location_history_rows
|
|+rundown.locations
|
|+crosscheck.tags
|
|+scout.tags
|
|+base.tags
|
|+dict.player_dedup
|
|+base.games
|
|+crosscheck.calibration_maintenance
|
|+base.calibration_maintenance
|
|+crosscheck.calibrations
|
|+crosscheck.games
|
|+base.games
|
|+base.locations
|
|+rundown.location_history
|
|+rundown.location_history_rows
|
|+rundown.locations
|
|+crosscheck.tags
|
|+scout.tags
|

```
                |+base.tags
                |
                |+dict.player_dedup
             |
             |+base.games


games.latency
    |
    |+advance.merged_measurement
    |
    |+sdkjson.live_pitch_data
    |
    |+sdkjson.measurement
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


games.live_final
    |
    |+merged.measurement
    |
    |+merged.live_pitch_data
    |
    |+merged.batting_launch_data
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


games.metadata
    |
    |+base.games
    |
    |+base.locations
    |
    |+rundown.location_history
       |
       |+rundown.location_history_rows
          |
          |+rundown.locations
```

```
games.nulls
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


games.oem
    |
    |+radar.measurement
    |
    |+radar.live_pitch_data
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


games.vision
    |
    |+vision.final_pitch_data
    |
    |+merged.measurement
    |
    |+vision.live_pitch_data
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


games.vision_oem
    |
    |+radar.measurement
    |
    |+radar.live_pitch_data
```

```
    |
    |+merged.measurement
    |
    |+vision.live_pitch_data
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games


locations.latency
    |
    |+merged.measurement
    |
    |+sdkjson.live_pitch_data
    |
    |+sdkjson.measurement
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
       |
       |+base.games
    |
    |+base.games


locations.vision_oem
    |
    |+radar.measurement
    |
    |+radar.live_pitch_data
    |
    |+merged.measurement
    |
    |+vision.live_pitch_data
    |
    |+base.locations
    |
    |+crosscheck.tags
       |
       |+scout.tags
          |
          |+base.tags
          |
          |+dict.player_dedup
```

```
       |
       |+base.games


report.delivered_combined_hist
    |
    |+report.delivered_no_exceptions_hist
       |
       |+report.delivered_new_pk
          |
          |+broadcast.delivered_games
       |
       |+base.games
       |
       |+report.exp_access_rules_hist
          |
          |+rundown.access_rules_end
          |
          |+broadcast.priority_lists
    |
    |+report.delivered_exceptions_hist
       |
       |+report.delivered_new_pk
          |
          |+broadcast.delivered_games
       |
       |+base.games
       |
       |+report.exp_access_rules_hist
          |
          |+rundown.access_rules_end
          |
          |+broadcast.priority_lists


report.delivered_exceptions_hist
    |
    |+report.delivered_new_pk
       |
       |+broadcast.delivered_games
    |
    |+base.games
    |
    |+report.exp_access_rules_hist
       |
       |+rundown.access_rules_end
       |
       |+broadcast.priority_lists


report.delivered_new_pk
    |
    |+broadcast.delivered_games


report.delivered_no_exceptions
```

```
    |
    |+report.delivered_new_pk
       |
       |+broadcast.delivered_games
    |
    |+base.games
    |
    |+report.exp_access_rules
       |
       |+broadcast.access_rules
       |
       |+broadcast.priority_lists


report.delivered_no_exceptions_hist
    |
    |+report.delivered_new_pk
       |
       |+broadcast.delivered_games
    |
    |+base.games
    |
    |+report.exp_access_rules_hist
       |
       |+rundown.access_rules_end
       |
       |+broadcast.priority_lists


report.exp_access_rules
    |
    |+broadcast.access_rules
    |
    |+broadcast.priority_lists


report.exp_access_rules_hist
    |
    |+rundown.access_rules_end
    |
    |+broadcast.priority_lists


report.false_delivered_hist
    |
    |+report.delivered_new_pk
       |
       |+broadcast.delivered_games
    |
    |+report.delivered_combined_hist
       |
       |+report.delivered_no_exceptions_hist
          |
          |+report.delivered_new_pk
             |
             |+broadcast.delivered_games
```

```
          |
          |+base.games
          |
          |+report.exp_access_rules_hist
             |
             |+rundown.access_rules_end
             |
             |+broadcast.priority_lists
       |
       |+report.delivered_exceptions_hist
          |
          |+report.delivered_new_pk
             |
             |+broadcast.delivered_games
          |
          |+base.games
          |
          |+report.exp_access_rules_hist
             |
             |+rundown.access_rules_end
             |
             |+broadcast.priority_lists


report.hardball_pitchers
    |
    |+scout.tags
       |
       |+base.tags
       |
       |+dict.player_dedup
    |
    |+base.measurements
    |
    |+base.games


report.mlb_orgs
    |
    |+dict.team_org
    |
    |+scout.teams


report.practice_pitching
    |
    |+practice.plays
    |
    |+practice.games


report.practice_pitching_dev
    |
    |+practice.plays_dev
    |
    |+practice.games_dev
```

```
report.scheduled_combined
    |
    |+report.scheduled_no_exceptions
       |
       |+lineup.schedule
       |
       |+games.metadata
          |
          |+base.games
          |
          |+base.locations
          |
          |+rundown.location_history
             |
             |+rundown.location_history_rows
                |
                |+rundown.locations
       |
       |+report.exp_access_rules
          |
          |+broadcast.access_rules
          |
          |+broadcast.priority_lists
    |
    |+report.scheduled_exceptions
       |
       |+lineup.schedule
       |
       |+games.metadata
          |
          |+base.games
          |
          |+base.locations
          |
          |+rundown.location_history
             |
             |+rundown.location_history_rows
                |
                |+rundown.locations
       |
       |+report.exp_access_rules
          |
          |+broadcast.access_rules
          |
          |+broadcast.priority_lists


report.scheduled_exceptions
    |
    |+lineup.schedule
    |
    |+games.metadata
       |
       |+base.games
```

```
        |
        |+base.locations
        |
        |+rundown.location_history
           |
           |+rundown.location_history_rows
              |
              |+rundown.locations
    |
    |+report.exp_access_rules
       |
       |+broadcast.access_rules
       |
       |+broadcast.priority_lists


report.scheduled_no_exceptions
    |
    |+lineup.schedule
    |
    |+games.metadata
       |
       |+base.games
       |
       |+base.locations
       |
       |+rundown.location_history
          |
          |+rundown.location_history_rows
             |
             |+rundown.locations
    |
    |+report.exp_access_rules
       |
       |+broadcast.access_rules
       |
       |+broadcast.priority_lists


report.scheduled_no_exceptions_hist
    |
    |+report.schedule_new_level
       |
       |+lineup.schedule
    |
    |+games.metadata
       |
       |+base.games
       |
       |+base.locations
       |
       |+rundown.location_history
          |
          |+rundown.location_history_rows
             |
             |+rundown.locations
```

```
    |
    |+report.exp_access_rules_hist
       |
       |+rundown.access_rules_end
       |
       |+broadcast.priority_lists


report.schedule_new_level
    |
    |+lineup.schedule


report.verified_levels_18
    |
    |+base.locations
    |
    |+base.games
    |
    |+dict.game_type


rundown.access_rules_history
    |
    |+rundown.access_rules_rows
       |
       |+rundown.access_rules
       |
       |+broadcast.access_rules
    |
    |+broadcast.access_rules


rundown.access_rules_rows
    |
    |+rundown.access_rules
    |
    |+broadcast.access_rules


rundown.location_history
    |
    |+rundown.location_history_rows
       |
       |+rundown.locations


rundown.location_history_rows
    |
    |+rundown.locations


scout.tags
    |
    |+base.tags
    |
```

```
    |+dict.player_dedup


trout.app_angle
    |
    |+scout.tags
       |
       |+base.tags
       |
       |+dict.player_dedup
    |
    |+base.measurements
    |
    |+base.games
    |
    |+base.players
    |
    |+report.hardball_ref
```