Data Engineer Take Home Test

Alongside this document, we will have provided 6 files in the `data` subdirectory. Each file contains a set of balls for a cricket match. The files are named MATCH_X where X indicates an “id” for that match. The balls are represented as JSON objects, one per line. The high-level objectives of this task are:

- Design a SQL database schema for representing these balls.
- Write a Python script that will parse these files, and read them into a database with that schema.
- Write some SQL queries to extract certain information from the database.

Cricket Background

Note: If you understand cricket already, please note there are various additional rules that can be omitted for this exercise, please do keep it simple – there is nothing outside of this section that you need to know!

A cricket match is played between two teams, and they both get a turn (“innings”) to bat and bowl. Any given ball is bowled by one player (the “bowler”) to another player (the “batter”), alongside this there is another player who is playing on the batting team who is referred to as the non-facing batter. Depending on the outcome of the ball, they may or may not be facing the next ball.

For the purpose of this exercise, the outcome of a ball is the combination of two things:

- runs being scored
- whether the batter is out or not

Periods of the game are split into overs (6 balls) and innings. There are two innings in each game. The first innings ends once 20 overs have occurred, or once 10 players have been out. The second innings then begins, and runs until 20 overs have occurred, ten players have been out, or the batting team scores more runs than were scored in the first innings.

Each ball in a match can be uniquely represented by the innings number, the over number and the ball number. For example, the second ball of the third over in the first innings could be represented by the 3-tuple (1, 3, 2).

Data

We have the following objects:

```
Unset
Player = {
        "name": string,
        "player_id": integer,
        "hand": "left" | "right" # "left" or "right"
}
Team = {
        "name": string,
        "team_id": integer
}
Ball = {
        "bowler": Player,
        "batter": Player,
        "non_facer": Player,
        "batting_team": Team,
        "bowling_team": Team,
        "is_out": boolean,
        "runs": integer
} # Each line is one of these objects (with whitespace stripped, ofcourse)
```

An example is below:

```
Unset
{
        "bowler": {
                "player_id": 1,
                "name": "Heinz",
                "hand": "left",
        },
        "batter": {
                "player_id": 2,
                "name": "Colman",
                "hand": "right",
        },
        "non_facer": {
                "player_id": 3,
                "name": "Maille",
                "hand": "right",
        },
        "batting_team": {
                "name": "Mustard Marvels",
                "team_id": 1,
        },
        "bowling_team": {
```

```
                "name": "Ketchup Kids",
                "team_id": 2,
            },
            "runs": 4,
            "is_out": False,
    }
```

We would like you to produce a SQL database schema (ideally for PostgreSQL) that will support storing this data. It is up to you how you do this. We would like you to then write a Python script that parses the data files into this database.

Queries

We would like queries alongside your schema to answer the following questions:

- Which team won each match? (i.e. who had the cumulative highest number of runs in their innings)
- Which “over” had the highest score for each team for each match? (i.e. for balls in this over number, which had the cumulative highest total)
- The average number of runs scored by each batter across all the matches they played in.

Solution

When you have completed your solution, please return to us, by email:

- The complete source of your solution.
- Brief instructions on how we can run the solution. You should assume the solution will be read by a developer working on a Linux desktop. This should include a list of any packages that are required, and which database you were using.
- Example output of your queries.