Project : Bridge Competition Management System

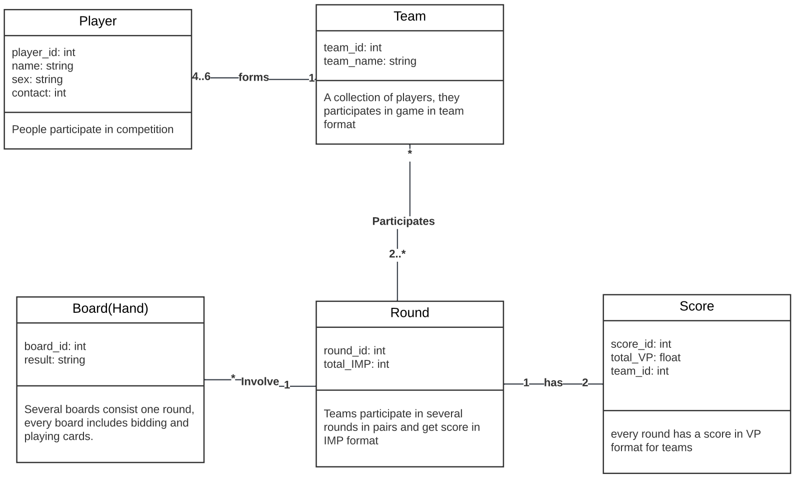
1. The system aims to develop a monitor that could help anticipants and spectators of a Bridge competition find relative information quickly especially regarding live scores, and detailed performance statistics.

Nouns & Actions

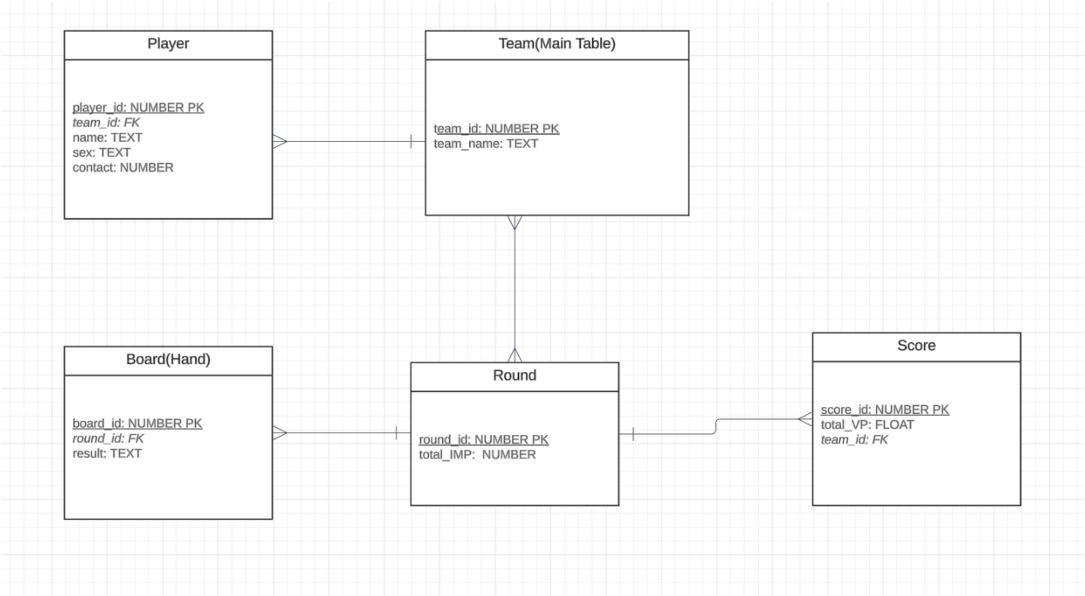
Rules:

1. Players must register for the competition, providing their personal details and contact information.
2. Players form teams often with regular partners (in pairs). A team contains 4 or 6 players so that there are 2 or 3 pairs.
3. The competition consists of multiple rounds which teams all participate in, each involving a set number of boards (hands) to be played.
4. At the beginning of each round, cards are randomly distributed to players according to the rules of the game. In the big tournaments, the machine deals the cards.
5. After every round, system shows real-time scores for teams, enabling participants and spectators to follow the progress of the competition.
6. Teams engage in bidding to determine the contract and play the hands according to the contract's specifications.
7. International Match Point (IMP) scores are calculated based on the results of each round, taking into account the difference in performance between competing teams.
8. Victory Points (VP) are computed to establish rankings among teams in the competition.

UML:

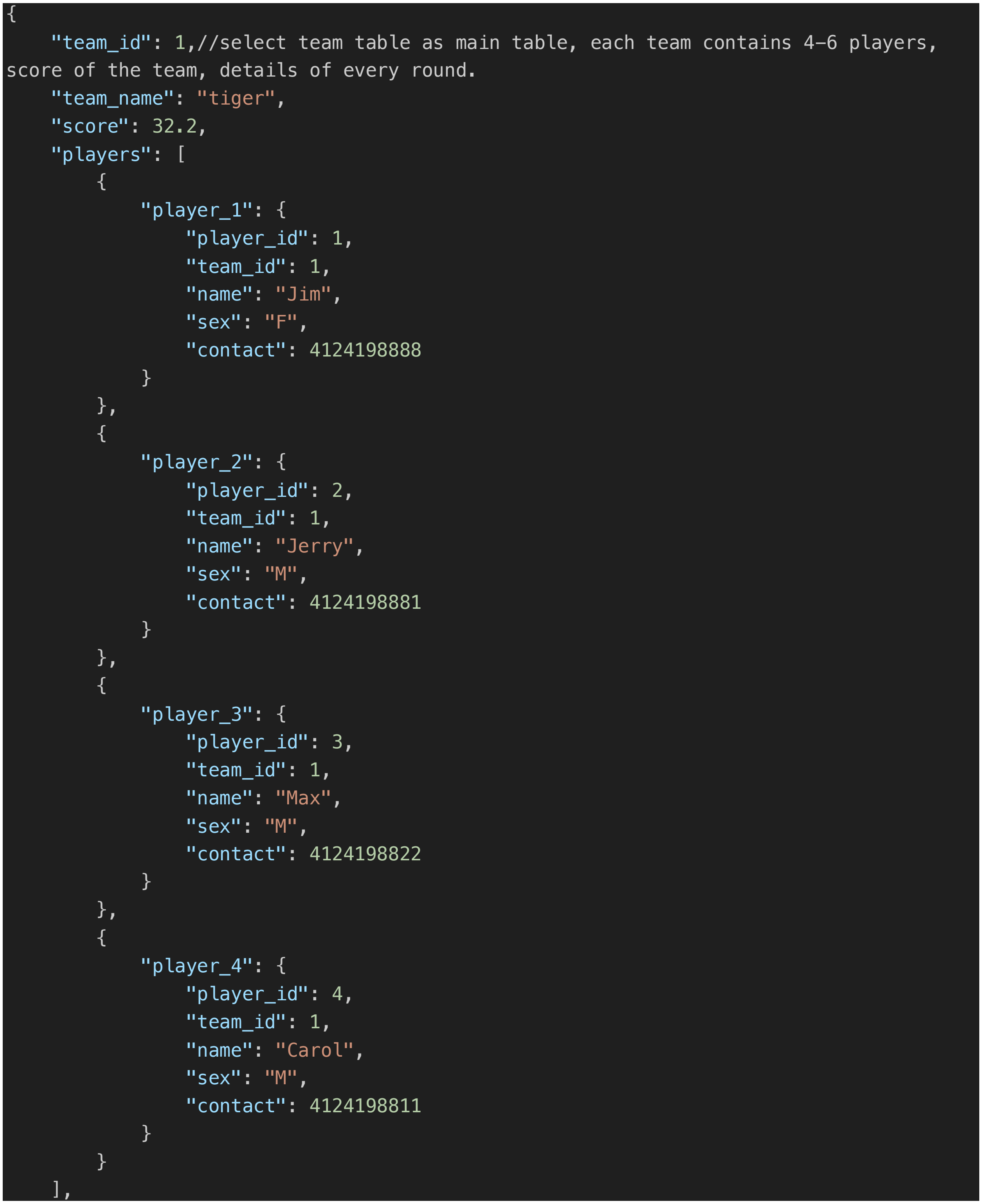

Player
player_id: int
name: string
sex: string
contact: int
People participate in competition
4..6
forms
1
Team
team_id: int
team_name: string
A collection of players, they participates in game in team format
*
Participates
2..*
Board(Hand)
board_id: int
result: string
Several boards consist one round, every board includes bidding and playing cards.
*
Involve
1
Round
round_id: int
total_IMP: int
Teams participate in several rounds in pairs and get score in IMP format
1
has
2
Score
score_id: int
total_VP: float
team_id: int
every round has a score in VP format for teams

2.ERD:

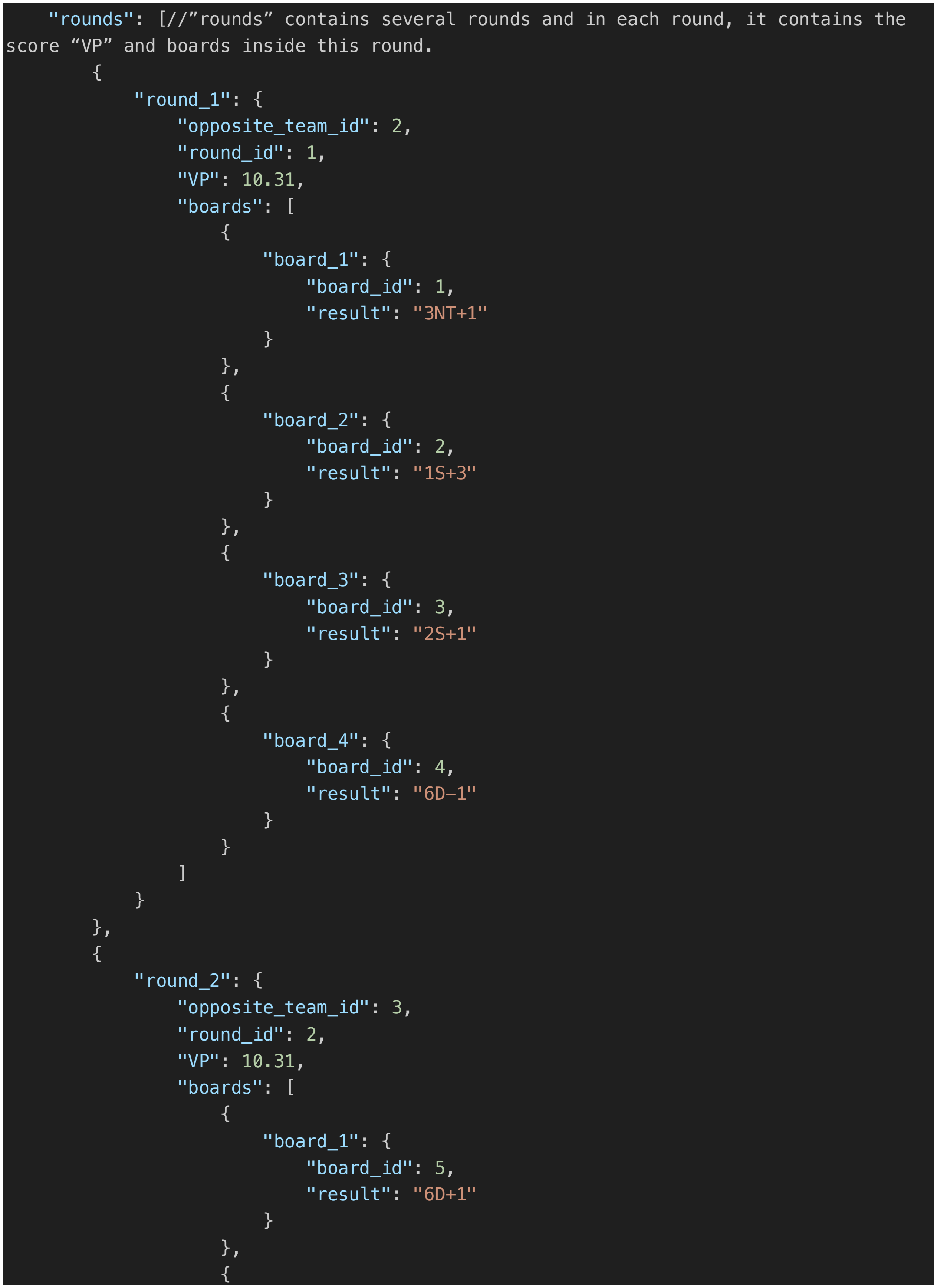


Link:
https://lucid.app/lucidchart/c4572e8a-b059-4304-a448-6d0e230a9d70/edit?viewport_loc=-2089%2C-396%2C2026%2C1104%2C0_0&invitationId=inv_6a6b36f5-d126-4915-8a04-b31b2c3abf09

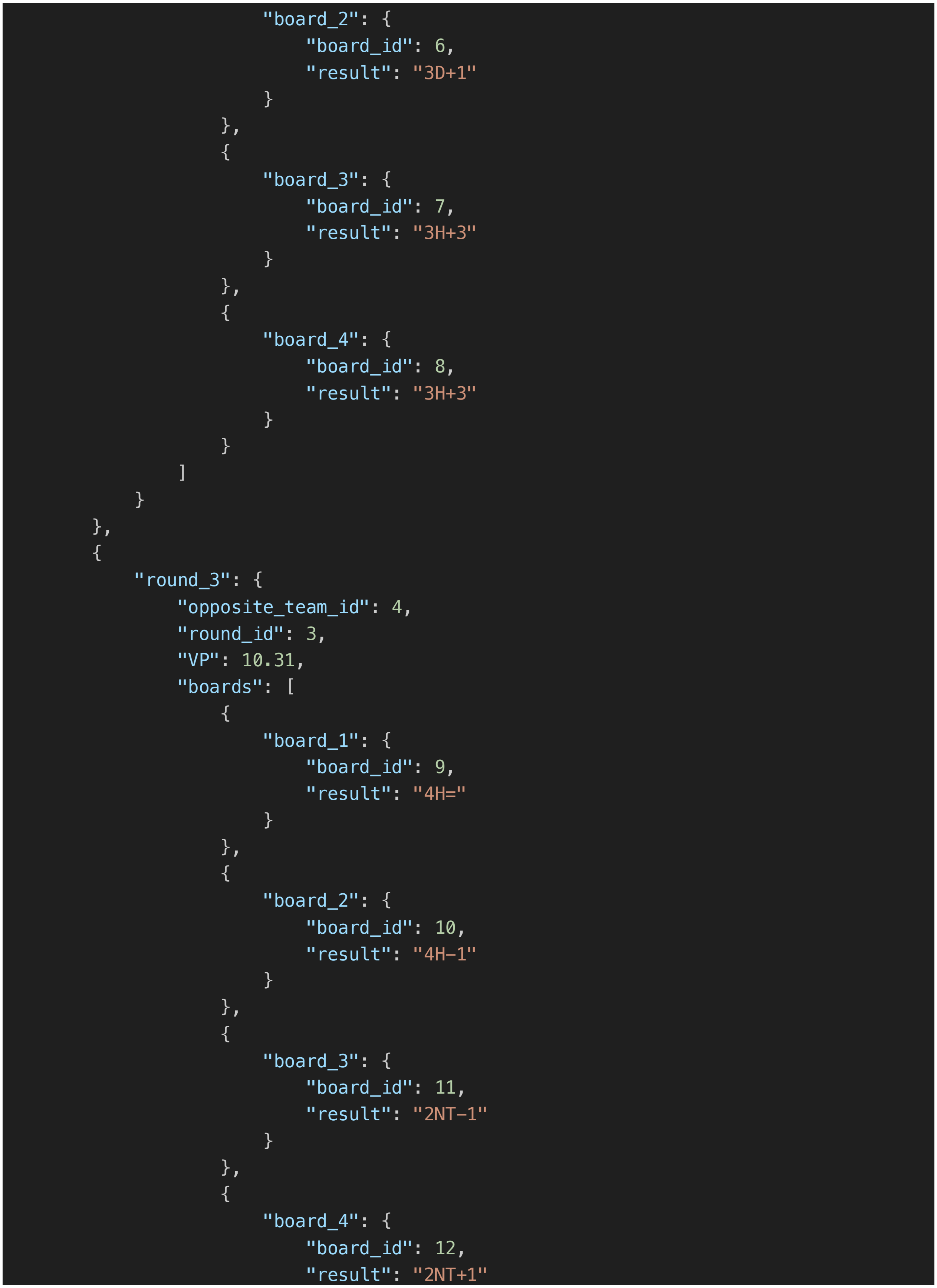
3. JSON examples:



```
{
    "team_id": 1,//select team table as main table, each team contains 4-6 players,
score of the team, details of every round.
    "team_name": "tiger",
    "score": 32.2,
    "players": [
        {
            "player_1": {
                "player_id": 1,
                "team_id": 1,
                "name": "Jim",
                "sex": "F",
                "contact": 4124198888
            }
        },
        {
            "player_2": {
                "player_id": 2,
                "team_id": 1,
                "name": "Jerry",
                "sex": "M",
                "contact": 4124198881
            }
        },
        {
            "player_3": {
                "player_id": 3,
                "team_id": 1,
                "name": "Max",
                "sex": "M",
                "contact": 4124198822
            }
        },
        {
            "player_4": {
                "player_id": 4,
                "team_id": 1,
                "name": "Carol",
                "sex": "M",
                "contact": 4124198811
            }
        }
    ],
```



```
    "rounds": [//”rounds” contains several rounds and in each round, it contains the
score “VP” and boards inside this round.
        {
            "round_1": {
                "opposite_team_id": 2,
                "round_id": 1,
                "VP": 10.31,
                "boards": [
                    {
                        "board_1": {
                            "board_id": 1,
                            "result": "3NT+1"
                        }
                    },
                    {
                        "board_2": {
                            "board_id": 2,
                            "result": "1S+3"
                        }
                    },
                    {
                        "board_3": {
                            "board_id": 3,
                            "result": "2S+1"
                        }
                    },
                    {
                        "board_4": {
                            "board_id": 4,
                            "result": "6D-1"
                        }
                    }
                ]
            }
        },
        {
            "round_2": {
                "opposite_team_id": 3,
                "round_id": 2,
                "VP": 10.31,
                "boards": [
                    {
                        "board_1": {
                            "board_id": 5,
                            "result": "6D+1"
                        }
                    },
                    {
```



```
                        "board_2": {
                            "board_id": 6,
                            "result": "3D+1"
                        }
                    },
                    {
                        "board_3": {
                            "board_id": 7,
                            "result": "3H+3"
                        }
                    },
                    {
                        "board_4": {
                            "board_id": 8,
                            "result": "3H+3"
                        }
                    }
                ]
            }
        },
        {
            "round_3": {
                "opposite_team_id": 4,
                "round_id": 3,
                "VP": 10.31,
                "boards": [
                    {
                        "board_1": {
                            "board_id": 9,
                            "result": "4H="
                        }
                    },
                    {
                        "board_2": {
                            "board_id": 10,
                            "result": "4H-1"
                        }
                    },
                    {
                        "board_3": {
                            "board_id": 11,
                            "result": "2NT-1"
                        }
                    },
                    {
                        "board_4": {
                            "board_id": 12,
                            "result": "2NT+1"
```

```
                        }
                    }
                ]
            }
        }
    ]
}
```

4.
Provided a json file: teams.json
Use command: `mongoimport –d bridge –c teams mongodb://localhost:27017 --file=teams.json –jsonArray`

5.
1: Find teams with score lower than 30:

```
db.teams.find(
...   { "score": { $lt: 30 } },
...   { "team_name": 1, "_id": 0 }
... )
```

Aggregation

2. Count the Number of Players in Each Team and sort in descending order:

```
db.teams.aggregate([
...   { $project: { team_name: 1, _id: 0, numPlayers: { $size:
"$players" } } },
...   { $sort: { numPlayers: -1 } }
... ])
```

Complex search criterion

3: Find numbers of teams with Female Players or score >=30:

```
db.teams.find({ $or: [ { "score": { $gte: 30 } }, { "players":
{ $elemMatch: { "sex": "F" } } }] }).count()
```

Counting documents for an specific user

4: Counting documents for a specific player - count of boards for team_id=1:

```
db.teams.findOne({ "team_id": 1 }, { "rounds":
1 }).rounds.reduce((total, round) => total +
Object.keys(round).length, 0)
```

Updating a document based on a query parameter

5. Change the name of team 2 if it has 4 players:

```
db.teams.updateOne( { $and: [{ "team_id": 2 }, { $expr: { $eq:
[{ $size: "$players" }, 4] } }] }, { $set: { "team_name": "bass" } } )
```