CSE-5331-001-DBMS MODELS AND IMPLEMENTATION TECHNIQUES

Project 2 Phase 2

MongoDB Complex Object Creation and Querying

Pavithra Rathinasabapathy(1001698736)

Tharoon T Thiagarajan(1001704601)

List of mongoDB operations used:

We used to the following operations to create the complex nested document:

- $lookup
- $replaceRoot
- $mergeObjects
- arrayElemAt
- concatArrays

Question 1:

Merging Schedule_Results collection with Stadiums Collection and storing it in Game_Stadium:

```
db.Schedule_Results.aggregate([
{
    $lookup:
        {
            from: "Stadiums",
            localField:"SID",
            foreignField: "SID",
            as: "Game_Stadium"
        }
},
{   $unwind:"$Game_Stadium"   },
{
    $project:{
                MatchDate :1,
                TeamID1: 1,
                Team1_Score :1,
                TeamID2:1,
                Team2_Score :1,
                "Game_Stadium.SName" : 1,
                "Game_Stadium.SCity" :1,
            }
},
{$out:"Game_Stadium"}
])
```

Merging Game_Stadium collection with Team collection and storing it in Team_1:

```
db.Teams.aggregate([
{
    $lookup:
        {
            from:"Game_Stadium",
            localField:"TeamID",
            foreignField:"TeamID1",
            as: "Team1"
        }
},
{
    $project:{
                _id: 0,
                Team:1,
                TeamID:1,
                "Team1.MatchDate" :1,
                "Team1.TeamID1": 1,
                "Team1.Team1_Score" :1,
                "Team1.TeamID2":1,
                "Team1.Team2_Score" :1,
                "Team1.Game_Stadium.SName" : 1,
                "Team1.Game_Stadium.SCity" :1,
            }
},
{$out:"Team_1"}])
```

Merging Game_Stadium collection with Team collection and storing it in Team_2:

```
db.Teams.aggregate([
{
    $lookup:
        {
            from:"Game_Stadium",
            localField:"TeamID",
            foreignField:"TeamID2",
            as: "Team2"
        }
},
{
    $project:{
                _id: 0,
                Team:1,
                TeamID:1,
                "Team2.MatchDate" :1,
                "Team2.TeamID1": 1,
                "Team2.Team1_Score" :1,
                "Team2.TeamID2":1,
                "Team2.Team2_Score" :1,
                "Team2.Game_Stadium.SName" : 1,
                "Team2.Game_Stadium.SCity" :1,
            }
},
{$out:"Team_2"}
])
```

Merging Team_2 collection with Team_1 collection and storing in Combined Collection:

```
db.Team_2.aggregate([
{
$lookup:
{
from:"Team_1",
localField:"Team",
```

foreignField:"Team",

as: "Team_Combined"

}

},

{

$replaceRoot: { newRoot: { $mergeObjects: [{ $arrayElemAt: ["$Team_Combined", 0] }, "$$ROOT"] } }

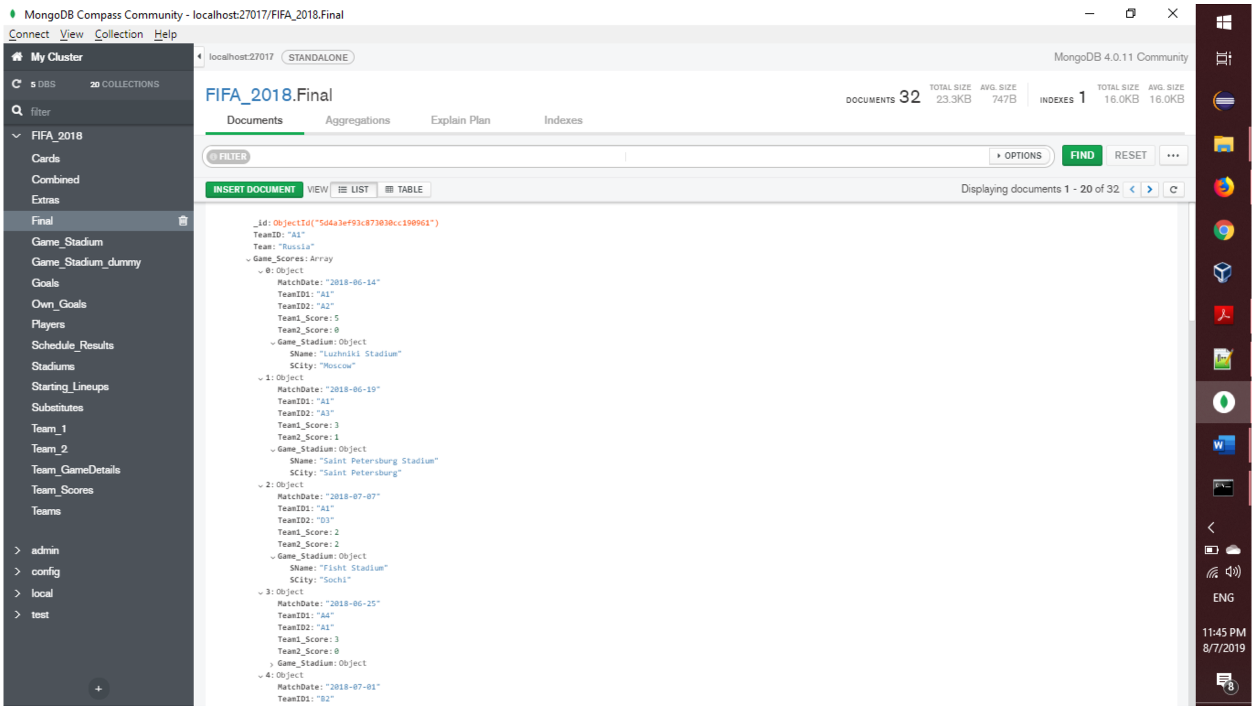
},

{ $project: { Team_Combined: 0 } },

{$out:"Combined"}

])

Complex Object:



```
_id: ObjectId("5d4a3ef93c873030cc190961")
TeamID: "A1"
Team: "Russia"
⌄Game_Scores: Array
  ⌄0: Object
      MatchDate: "2018-06-14"
      TeamID1: "A1"
      TeamID2: "A2"
      Team1_Score: 5
      Team2_Score: 0
    ⌄Game_Stadium: Object
        SName: "Luzhniki Stadium"
        SCity: "Moscow"
  ⌄1: Object
      MatchDate: "2018-06-19"
      TeamID1: "A1"
      TeamID2: "A3"
      Team1_Score: 3
      Team2_Score: 1
    ⌄Game_Stadium: Object
        SName: "Saint Petersburg Stadium"
        SCity: "Saint Petersburg"
  ⌄2: Object
      MatchDate: "2018-07-07"
      TeamID1: "A1"
      TeamID2: "D3"
      Team1_Score: 2
      Team2_Score: 2
    ⌄Game_Stadium: Object
        SName: "Fisht Stadium"
        SCity: "Sochi"
  ⌄3: Object
      MatchDate: "2018-06-25"
      TeamID1: "A4"
      TeamID2: "A1"
      Team1_Score: 3
      Team2_Score: 0
    ›Game_Stadium: Object
  ⌄4: Object
      MatchDate: "2018-07-01"
      TeamID1: "B2"
```

Sample Queries:

```
db.Final.find( { "Game_Scores": {$elemMatch:{ Team1_Score : { $gt:1 }}}} )

db.Final.find( { “Game_Scores.MatchDate”: {“  2018-06-24”}

db.Final.find ( { “Game_Scores.Game_Stadium.SName” : {“Luzhniki Stadium”} )

db.Final.find( {“Team”:”Russia”} )
```

Question 2:

Merge Schedule_Results collection with Stadium Collection and store it in Game_Stadium_Details:

```
db.schedule_result.aggregate ( [
{
$lookup:
{
from:"stadiums",
localField:"SID",
foreignField:"SID",
as:"Game_Stadium"
}
},
{
$project: {
	"_id": 0,
"GameID": 1,
"Groups": 1,
"MatchDate":1,
"SID" : 1,
	"TeamID1" : 1,
	"TeamID2" : 1,
	"Team1_Score" : 1,
	"Team2_Score" : 1,
	"Game_Stadium.SID" :1,
"Game_Stadium.SName" :1,
"Game_Stadium.SCity" :1
}
},
{
$out: "Game_Stadium_Details"
}
])
```

Merge Schedule_Results collection with Goals collection and store it in Game_Goal_Details:

```
db.schedule_result.aggregate ( [
{
$lookup:
{
from:"goals",
localField:"GameID",
foreignField:"GameID",
as:"Game_Goals"
}
},
{
$project: {
      "_id": 0,
"GameID": 1,
"Groups": 1,
"MatchDate":1,
"SID" : 1,
      "TeamID1" : 1,
      "TeamID2" : 1,
      "Team1_Score" : 1,
      "Team2_Score" : 1,
      "Game_Goals.GameID" :1,
"Game_Goals.TeamID" :1,
"Game_Goals.PlayerID" :1,
"Game_Goals.Time" :1
}
},
{
$out: "Game_Goal_Details"
}

])
```

Merge Rosters collection with Game_Stadium_Details collection and store it in Game_Player_Details:

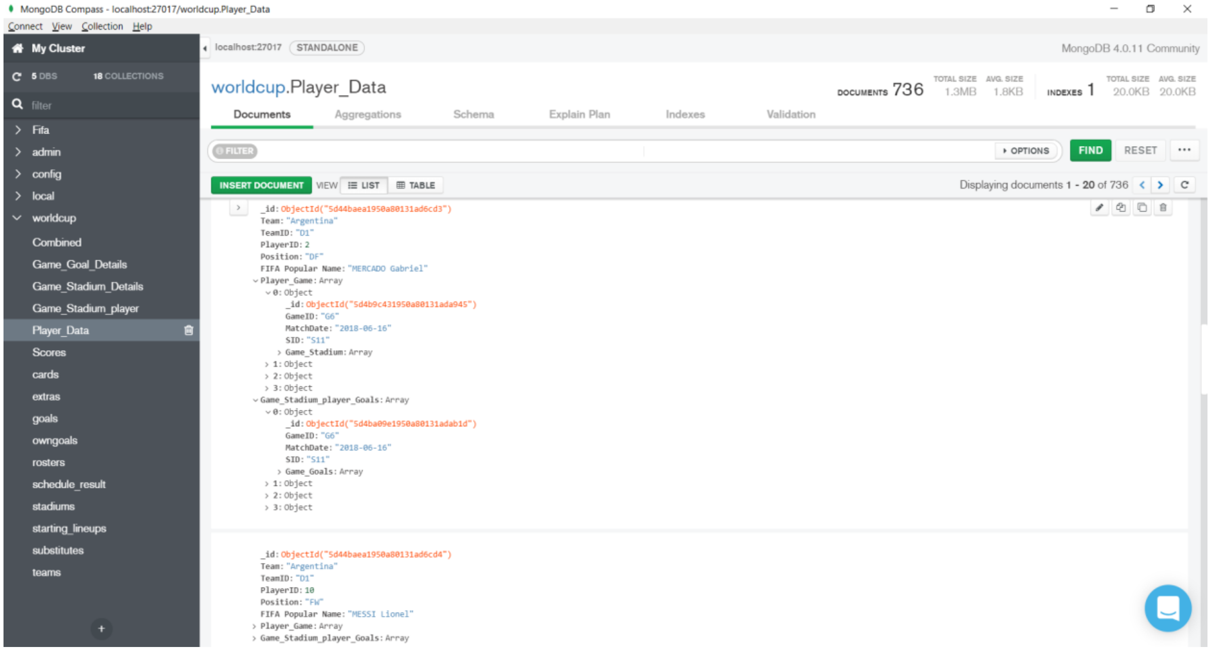
```
db.rosters.aggregate ( [
{
$lookup: {
        from: "Game_Stadium_Details",
        let: {
          teamID: "$TeamID",
        },
        pipeline: [
          { $match: {
              $expr: {
                $or: [
                  { $eq: [ "$TeamID1", "$$teamID" ] },
                  { $eq: [ "$TeamID2", "$$teamID" ] }
                ]
              }
            }
          }
        ],
as: "Player_Game"
}},
{
$project: {

  "Birth Date": 0,
  "Shirt Name": 0,
  "Club": 0,
  "Height": 0,
  "Weight":0,
  "Player_Game.Groups": 0,
  "Player_Game.TeamID1": 0,
  "Player_Game.TeamID2": 0,
  "Player_Game.Team1_Score": 0,
  "Player_Game.Team2_Score": 0,
}},
{
$out: "Game_Stadium_player"
}
])
```

Merge Game_Stadium_player collection with Game_Goal_Details collection and store it in Player_Data:

```
db.Game_Stadium_player.aggregate ( [
{
$lookup: {
          from: "Game_Goal_Details",
          let: {
            teamID: "$TeamID"
          },
          pipeline: [
            { $match: {
                $expr: {
                  $or: [
                    { $eq: [ "$TeamID1", "$$teamID" ] },
                    { $eq: [ "$TeamID2", "$$teamID" ] }
                  ]
                }
              }
            }
          ],
as: "Game_Stadium_player_Goals"
}},
{
$project: {
  "Game_Stadium_player_Goals.Groups": 0,
  "Game_Stadium_player_Goals.TeamID1": 0,
  "Game_Stadium_player_Goals.TeamID2": 0,
  "Game_Stadium_player_Goals.Team1_Score": 0,
  "Game_Stadium_player_Goals.Team2_Score": 0
  }},
{
 $out: "Player_Data1"
}])
```

Complex object:



```
_id: ObjectId("5d44baea1950a80131ad6cd3")
Team: "Argentina"
TeamID: "D1"
PlayerID: 2
Position: "DF"
FIFA Popular Name: "MERCADO Gabriel"
Player_Game: Array
  0: Object
    _id: ObjectId("5d4b9c431950a80131ada945")
    GameID: "G6"
    MatchDate: "2018-06-16"
    SID: "S11"
    Game_Stadium: Array
  1: Object
  2: Object
  3: Object
Game_Stadium_player_Goals: Array
  0: Object
    _id: ObjectId("5d4ba09e1950a80131adab1d")
    GameID: "G6"
    MatchDate: "2018-06-16"
    SID: "S11"
    Game_Goals: Array
  1: Object
  2: Object
  3: Object

_id: ObjectId("5d44baea1950a80131ad6cd4")
Team: "Argentina"
TeamID: "D1"
PlayerID: 10
Position: "FW"
FIFA Popular Name: "MESSI Lionel"
Player_Game: Array
Game_Stadium_player_Goals: Array
```

Sample Queries:

db.Player_Data.find({"FIFA Popular Name": "Lionel Messi"})