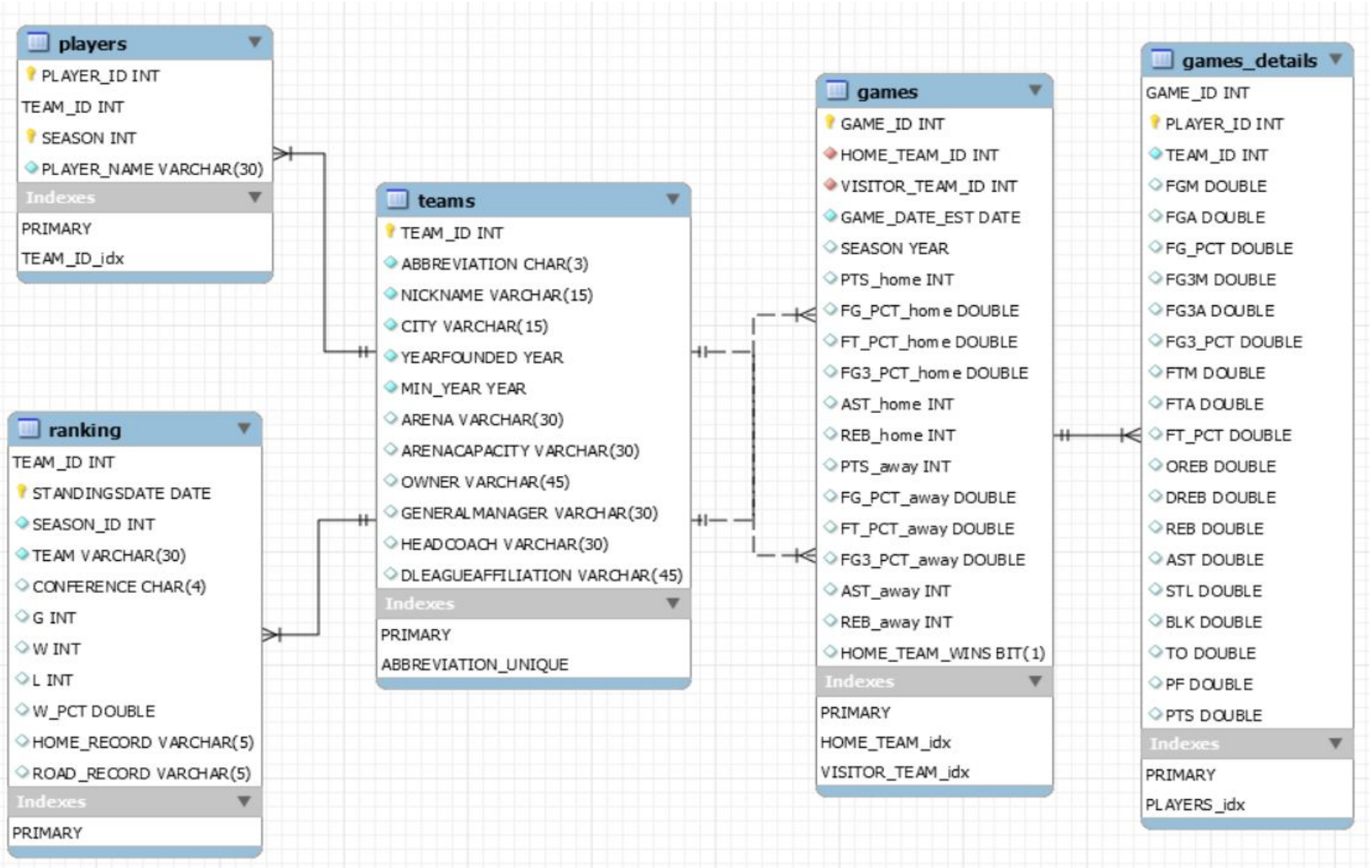
DATABASE SCHEMA (SQL -> NoSQL)



DIFFERENCES:

1) replication of the PLAYER_NAME attribute in the games_details table

2) removing of indexes and keys

RELATIONAL DATABASE

TABLE FORMAT

PLAYER_ID	TEAM_ID	SEASON	PLAYER_NAME
244	1610612742	2010	Dee Brown
255	1610612746	2012	Grant Hill
283	1610612741	2009	Lindsey Hunter
406	1610612738	2010	Shaquille O'Neal
436	1610612748	2010	Juwan Howard
467	1610612742	2009	Jason Kidd
686	1610612759	2009	Antonio McDyess
689	1610612747	2010	Theo Ratliff
693	1610612737	2009	Joe Smith
703	1610612741	2010	Kurt Thomas
708	1610612738	2009	Kevin Garnett
711	1610612737	2011	Jerry Stackhouse
714	1610612738	2009	Michael Finley
739	1610612738	2009	Rasheed Wallace
947	1610612755	2009	Allen Iverson
948	1610612745	2011	Marcus Camby
951	1610612738	2009	Ray Allen

DOCUMENT-ORIENTED DATABASE

JSON FILE (DOCUMENT)

Key	Value	Type
(1) 628a7e46e2b06bb5b9b2d944	{ PLAYER_NAME : "Wesley Matthews", TEAM_ID : 1610612762, PLAYER_ID : 500032, SEASON : 2009 } (5 fields)	Document
_id	628a7e46e2b06bb5b9b2d944	ObjectId
PLAYER_NAME	Wesley Matthews	String
TEAM_ID	1610612762	Int32
PLAYER_ID	500032	Int32
SEASON	2.009 (2.0K)	Int32
(2) 628a7e46e2b06bb5b9b2d943	{ PLAYER_NAME : "Brian Hamilton", TEAM_ID : 1610612751, PLAYER_ID : 201646, SEASON : 2009 } (5 fields)	Document
(3) 628a7e46e2b06bb5b9b2d942	{ PLAYER_NAME : "Bennet Davis", TEAM_ID : 1610612751, PLAYER_ID : 201834, SEASON : 2009 } (5 fields)	Document
(4) 628a7e46e2b06bb5b9b2d941	{ PLAYER_NAME : "Warren Carter", TEAM_ID : 1610612752, PLAYER_ID : 201999, SEASON : 2009 } (5 fields)	Document
(5) 628a7e46e2b06bb5b9b2d940	{ PLAYER_NAME : "Lanny Smith", TEAM_ID : 1610612758, PLAYER_ID : 201831, SEASON : 2009 } (5 fields)	Document
(6) 628a7e46e2b06bb5b9b2d93f	{ PLAYER_NAME : "Dionte Christmas", TEAM_ID : 1610612755, PLAYER_ID : 1962936270, SEASON : 2009 } (5 fiel	Document
(7) 628a7e46e2b06bb5b9b2d93e	{ PLAYER_NAME : "Thomas Gardner", TEAM_ID : 1610612763, PLAYER_ID : 201242, SEASON : 2009 } (5 fields)	Document
(8) 628a7e46e2b06bb5b9b2d93d	{ PLAYER_NAME : "Sean Singletary", TEAM_ID : 1610612755, PLAYER_ID : 201606, SEASON : 2009 } (5 fields)	Document
(9) 628a7e46e2b06bb5b9b2d93c	{ PLAYER_NAME : "Chris Quinn", TEAM_ID : 1610612748, PLAYER_ID : 200809, SEASON : 2009 } (5 fields)	Document
(10) 628a7e46e2b06bb5b9b2d93b	{ PLAYER_NAME : "Melvin Ely", TEAM_ID : 1610612758, PLAYER_ID : 2408, SEASON : 2009 } (5 fields)	Document

1-2) return players who have scored at least 60 points in a game

MYSQL (SQL)

```sql
SELECT PLAYER_NAME, g.SEASON, PTS
FROM players p, games g, games_details gd
WHERE p.PLAYER_ID=gd.PLAYER_ID AND g.GAME_ID=gd.GAME_ID AND PTS>=60
GROUP BY gd.PLAYER_ID, gd.GAME_ID
ORDER BY PTS DESC
```

MONGO DB

1)

```
db.games_details.find({PTS:{$gte:60}},
                      {PLAYER_NAME:1,PTS:1,_id:0},
).sort({PTS:-1})
```

2)

```
db.games_details.aggregate([
    {$match:{PTS:{$gte:60}}},
    {$lookup:{
        from: "games",
        localField:"GAME_ID",
        foreignField:"GAME_ID",
        as:"games"
    }},
    {$unwind:"$games"},
    {$project:{PLAYER_NAME:1,"SEASON":"$games.SEASON",PTS:1,_id:0}},
    {$sort:{PTS:-1}}
])
```

3) for each team, return the number of days in which that team had a league winning percentage greater than 50%

SQL

```sql
SELECT NICKNAME, COUNT(*) AS TOT
FROM teams t JOIN ranking r ON t.TEAM_ID=r.TEAM_ID
WHERE W_PCT>0.5
GROUP BY r.TEAM_ID
ORDER BY TOT DESC
```

MONGO DB

```
db.ranking.aggregate([
    {$match:{W_PCT:{$gt:0.5}}},
    {$group:{_id:"$TEAM_ID",TOT:{$count:{}}}},
    {$lookup:{
        from:"teams",
        localField:"_id",
        foreignField:"TEAM_ID",
        as:"teams"
    }},
    {$unwind:"$teams"},
    {$project:{"NICKNAME":"$teams.NICKNAME",TOT:1,_id:0}},
    {$sort:{TOT:-1}}
])
```

4) return players who participated in at least one game in which the home team won by more than 30 points over the visiting team during the season 2021

SQL

```sql
SELECT DISTINCT PLAYER_NAME
FROM players p, games g, (SELECT PLAYER_ID, gd.GAME_ID
                          FROM games_details gd JOIN games g ON gd.GAME_ID=g.GAME_ID
                          WHERE SEASON=2021) AS gd1
WHERE p.PLAYER_ID=gd1.PLAYER_ID AND g.GAME_ID=gd1.GAME_ID AND PTS_home-PTS_away>30
```

MONGO DB

```
db.games.aggregate([
    {$project:{GAME_ID:1,SEASON:1,"DIFF":{$subtract:["$PTS_home","$PTS_away"]}}},
    {$match:{$and:[{SEASON:2021},{DIFF:{$gt:30}}]}},
    {$lookup:{
        from:"games_details",
        localField:"GAME_ID",
        foreignField:"GAME_ID",
        as:"gd"
    }},
    {$unwind:"$gd"},
    {$project:{"PLAYER_NAME":"$gd.PLAYER_NAME"}},
    {$group:{ _id:"$PLAYER_NAME"}}
])
```

5) return the number of points and the average points per game of all players who have played more than 300 games

SQL

```sql
SELECT PLAYER_NAME, gd.PLAYER_ID, COUNT(*) AS N_MATCHES, SUM(PTS) AS SUM_PTS, ROUND(AVG(PTS),2) AS AVERAGE_PTS
FROM games_details gd LEFT JOIN (SELECT * FROM players GROUP BY PLAYER_ID) p ON gd.PLAYER_ID=p.PLAYER_ID
GROUP BY gd.PLAYER_ID
HAVING N_MATCHES>300
```

MONGO DB

```
db.games_details.aggregate([
    {$group:{ _id:"$PLAYER_NAME",N_MATCHES:{$count:{}},SUM_PTS:{$sum:"$PTS"},AVERAGE_PTS:{$avg:"$PTS"}}},
    {$match:{N_MATCHES:{$gt:300}}},
])
```

6) return the names of the teams in which at least one player has scored more than 60 points in a game

SQL

```sql
SELECT CITY, NICKNAME
FROM teams t
WHERE TEAM_ID IN (SELECT TEAM_ID
                  FROM players
                  WHERE PLAYER_ID IN (SELECT PLAYER_ID
                                      FROM games_details
                                      WHERE PTS>60))
ORDER BY CITY
```

MONGO DB

```
db.games_details.aggregate([
    {$match:{PTS:{$gt:60}}},
    {$lookup:{
        from:"players",
        localField:"PLAYER_ID",
        foreignField:"PLAYER_ID",
        as:"players"
    }},
    {$lookup:{
        from:"teams",
        localField:"TEAM_ID",
        foreignField:"TEAM_ID",
        as:"teams"
    }},
    {$unwind:"$teams"},
    {$group:{ _id:["$teams.CITY","$teams.NICKNAME"]}}
])
```

7) return games won by the Lakers in 2019 as home team

SQL

```sql
SELECT NICKNAME AS OPPONENT, GAME_DATE_EST, PTS_home, PTS_away
FROM games g, teams t
WHERE (HOME_TEAM_ID, VISITOR_TEAM_ID) IN (SELECT t1.TEAM_ID, t2.TEAM_ID
                                          FROM teams t1 CROSS JOIN teams t2
                                          WHERE t1.TEAM_ID!=t2.TEAM_ID AND t1.NICKNAME='Lakers' AND HOME_TEAM_WINS=1 AND SEASON=2019)
      AND g.VISITOR_TEAM_ID=t.TEAM_ID
```

MONGO DB

```
db.games.aggregate([
    {$lookup:{
        from:"teams",
        localField:"HOME_TEAM_ID",
        foreignField:"TEAM_ID",
        as:"teams1"
    }},
    {$lookup:{
        from:"teams",
        localField:"VISITOR_TEAM_ID",
        foreignField:"TEAM_ID",
        as:"teams2"
    }},
    {$unwind:"$teams1"},
    {$unwind:"$teams2"},
    {$project:{"HOME":"$teams1.NICKNAME","OPPONENT":"$teams2.NICKNAME",GAME_DATE_EST:1,PTS_home:1,PTS_away:1,HOME_TEAM_WINS:1,SEASON:1}},
    {$match:{HOME:"Lakers",HOME_TEAM_WINS:true,SEASON:2019}},
    {$project:{OPPONENT:1,GAME_DATE_EST:1,PTS_home:1,PTS_away:1,_id:0}}
])
```

8) return the player with the highest average points per game

SQL

```sql
SELECT PLAYER_NAME, ROUND(AVG(PTS),3) AS AVERAGE_PTS
FROM players p JOIN games_details gd ON p.PLAYER_ID=gd.PLAYER_ID
GROUP BY p.PLAYER_ID
HAVING AVG(PTS)>=ALL(SELECT AVG(PTS)
                     FROM players p JOIN games_details gd ON p.PLAYER_ID=gd.PLAYER_ID
                     GROUP BY gd.PLAYER_ID)
```

MONGO DB

```mongodb
db.games_details.aggregate([
    {$group:{ _id:"$PLAYER_NAME",AVERAGE_PTS:{$avg:"$PTS"}}},
    {$sort:{AVERAGE_PTS:-1}},
    {$limit:1}
])
```

9-10) return players and their total of 3-points scored in games in which they scored more than 30 points

MONGO DB

AGGREGATION

```
db.games_details.aggregate([
    {$match:{PTS:{$gt:30}}},
    {$group:{ _id:"$PLAYER_NAME",SHOTS3:{$sum:"$FG3M"}}}
])
```

MAP-REDUCE

```
db.games_details.mapReduce(
    function(){emit(this.PLAYER_NAME,this.FG3M); },
    function(key,values){return Array.sum(values); },
    {
        query:{PTS:{$gt:30}},
        out:"3SHOTS"
    }
)
```