Project Report: Analyzing Data From a Basketball Game Using MongoDB

Advanced Data Management and Decision Support Systems

Group Members:

Ozgur Gumus

Ghazaleh Monsef Shemordeh

Advisors:

Andrea Maurino

Federico Antonio Niccolò Amedeo Cabitza

Andrea Campagner

Milano, 2023

Objective

The primary objective of this project is to analyze data collected from a basketball game to derive insights into player performance, team strategies, and overall game dynamics. This involves the collection and analysis of sensor data, video footage, and game statistics. The specific queries to be addressed include identifying the highest-scoring player, determining the number of fouls committed by each player, tracking the positions on the court where a specific player spent the most time, counting the number of passes between two specific players, and determining the times during the game when a specific team had possession of the ball.

Introduction

Game 1 of the NBA Finals 2024 between the Boston Celtics and the Dallas Mavericks is picked for as a reference for this project. Parts of the data is taken from the game and some parts are generated with using reference from the real parts of the data in order to answer queries that are asked in the project.

What is MongoDB?

MongoDB is a NoSQL database known for its flexibility, scalability, and ease of use. Unlike traditional relational databases that use tables and rows, MongoDB stores data in flexible, JSON-like documents. This allows for a dynamic schema, meaning the structure of the data can change over time without requiring a predefined schema. MongoDB is designed to handle large volumes of diverse and complex data efficiently.

Why MongoDB is Ideal for This Project

1. Schema Flexibility

- Dynamic Schema: MongoDB’s flexible document-oriented storage easily handles various data types like player stats, game events, and sensor data, which are crucial for comprehensive basketball game analysis.

2. Scalability

- Horizontal Scaling: MongoDB's ability to distribute data across multiple servers allows it to efficiently manage the large volumes of real-time data generated during a basketball game, including high-resolution video and sensor data.

3. Rich Query Capabilities

- Powerful Queries: MongoDB supports complex queries and aggregations, enabling detailed analysis of player performance and team strategies. This is useful for answering specific questions like who scored the most points or the distribution of ball possession.

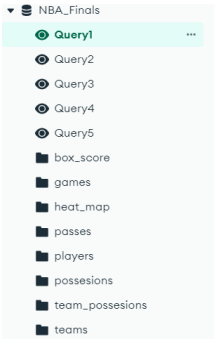
4. Real-Time Data Processing

- Live Insights: Integration with real-time data processing tools allows for live analysis of game data, providing immediate insights during the game, which can be valuable for coaching and strategy adjustments.

5. Ease of Use

- Developer-Friendly: MongoDB's simplicity and support for various programming languages make it easy to develop applications for collecting, storing, and analyzing basketball game data, facilitating rapid prototyping and adaptation to new data needs.

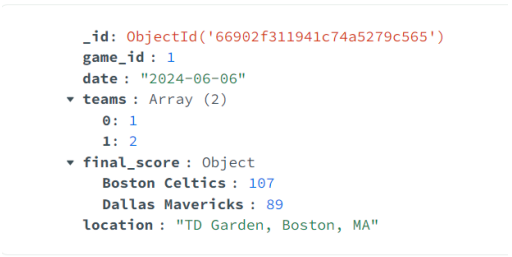
Data Model

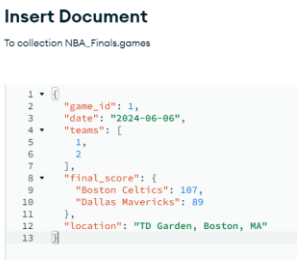


The data model for this project is designed to efficiently capture and organize the various aspects of an NBA basketball game. We have chosen Game 1 of the 2024 NBA Finals as our focus. Some of the collections are acquired from the real data of the game, while for some queries, we had to generate data. The collections for games, players, teams, possessions, and box_score are based on real data from the game. However, the collections for heat_map, passes, and team_possessions are generated by us to complete the queries for the project.

1. games

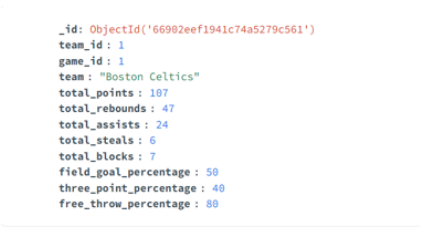
- **Purpose:** To store data about each game, in this case only 1 game, such as the date, participating teams, final score, and location.

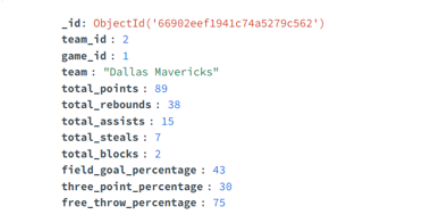




2. teams

- **Purpose:** To store aggregated statistics for each team participating in a game, such as total points, rebounds, assists, and other performance metrics. Boston Celtics and Dallas Mavericks are the teams that have participated in the Game 1 of NBA Finals 2024.





3. players

- **Purpose:** This data consists of the players who participated in the game, including their names, teams, and positions. It is taken from a real source and 23 players participated in the NBA Finals Game 1.

players

	_id Int32	name String	team String	position String
1	1	"Jayson Tatum"	"Boston Celtics"	"Forward"
2	2	"Jaylen Brown"	"Boston Celtics"	"Guard"
3	3	"Derrick White"	"Boston Celtics"	"Guard"
4	4	"Jrue Holiday"	"Boston Celtics"	"Guard"
5	5	"Al Horford"	"Boston Celtics"	"Center"
6	6	"Kristaps Porzingis"	"Boston Celtics"	"Forward"
7	7	"Sam Hauser"	"Boston Celtics"	"Forward"
8	8	"Payton Pritchard"	"Boston Celtics"	"Guard"
9	9	"Luke Kornet"	"Boston Celtics"	"Center"
10	10	"Svi Mykhailiuk"	"Boston Celtics"	"Guard"
11	11	"Oshae Brissett"	"Boston Celtics"	"Forward"


```
[{
  "_id": 1,
  "name": "Jayson Tatum",
  "team": "Boston Celtics",
  "position": "Forward"
},
{
  "_id": 2,
  "name": "Jaylen Brown",
  "team": "Boston Celtics",
  "position": "Guard"
},
{
  "_id": 3,
  "name": "Derrick White",
  "team": "Boston Celtics",
  "position": "Guard"
},
{
  "_id": 4,
  "name": "Jrue Holiday",
  "team": "Boston Celtics",
  "position": "Guard"
},
{
  "_id": 5,
```


4. box_score

- **Purpose:** This collection is designed to store comprehensive statistical data for each player in a game. This data includes points scored, rebounds, assists, steals, blocks, turnovers, personal fouls, true shooting percentage, and minutes played. These metrics are essential for evaluating individual contributions to the game and understanding player performance.


```
{
  "_id": 1,
  "game_id": 1,
  "points": 16,
  "rebounds": 11,
  "assists": 5,
  "steals": 0,
  "blocks": 1,
  "TS%": 47.4,
  "turnovers": 6,
  "personal_fouls": 1,
  "minutes": "42:09"
}
{
  "_id": 2,
  "game_id": 1,
  "points": 22,
  "rebounds": 6,
  "assists": 2,
  "steals": 3,
  "blocks": 3,
  "TS%": 65.3,
  "turnovers": 2,
  "personal_fouls": 3,
  "minutes": "37:14"
}
{
  "_id": 3,
```


_id	game_id	points	rebounds	assists	steals	blocks	TS%	turnovers	personal_f	minutes
1	1	16	11	5	0	1	47.4	6	1	42:09:00
2	1	22	6	2	3	3	65.3	2	3	37:14:00
3	1	15	2	5	1	0	63.1	2	2	35:10:00
4	1	12	8	5	1	0	60.7	0	1	34:54:00
5	1	10	7	3	0	2	62.5	0	2	29:48:00
6	1	20	6	0	0	3	72	1	1	20:34
7	1	8	4	1	0	0	100	0	2	16:02
8	1	0	2	2	0	0	0	1	0	15:30
9	1	2	1	0	0	1	100	0	1	03:19
10	1	2	0	0	0	0	100	0	1	02:40
11	1	0	0	0	0	0	0	0	0	00:42
12	1	30	10	1	2	0	53.2	4	4	38:14:00
13	1	12	3	2	2	0	31.6	3	2	36:29:00
14	1	14	8	1	0	0	51.3	0	0	36:07:00
15	1	5	6	0	1	0	27.8	0	1	28:47:00
16	1	8	3	1	1	0	103.1	0	0	14:19
17	1	3	3	0	0	0	37.5	0	0	20:17
18	1	2	0	1	0	1	100	1	0	18:29
19	1	2	5	1	1	0	53.2	2	0	18:29
20	1	13	3	1	0	0	66.6	0	1	10:41
21	1	0	0	0	1	0	0	0	1	07:42
22	1	0	2	1	0	0	0	1	1	05:17
23	1	0	0	0	0	0	0	1	0	04:58

- **_id:** A unique identifier for each player.
- **game_id:** The identifier of the game to which these statistics belong.
- **points:** Total points scored by the player.
- **rebounds:** Total rebounds grabbed by the player.
- **assists:** Total assists made by the player.
- **steals:** Total steals made by the player.
- **blocks:** Total blocks made by the player.
- **TS%:** True Shooting Percentage, a measure of shooting efficiency
- **turnovers:** Total turnovers committed by the player.
- **personal_fouls:** Total personal fouls committed by the player.
- **minutes:** Total minutes played by the player.


```
[{
  "_id": 1,
  "game_id": 1,
  "points": 16,
  "rebounds": 11,
  "assists": 5,
  "steals": 0,
  "blocks": 1,
  "TS%": 47.4,
  "turnovers": 6,
  "personal_fouls": 1,
  "minutes": "42:09"
},
{
  "_id": 2,
  "game_id": 1,
  "points": 22,
  "rebounds": 6,
  "assists": 2,
  "steals": 3,
  "blocks": 3,
  "TS%": 65.3,
  "turnovers": 2,
  "personal_fouls": 3,
  "minutes": "37:14"
},
{
  "_id": 3,
  "game_id": 1,
  "points": 15,
  "rebounds": 2,
  "assists": 5,
  "steals": 1,
  "blocks": 0,
  "TS%": 63.1,
```

5. heat_map

- **Purpose:** This collection is designed to track the amount of time each player spends in various positions on the court. The half-court is divided into 6 sections:

1. Left Wing
2. Right Wing
3. Left Corner
4. Right Corner
5. Paint (area near the basket)
6. Top (top of the key, beyond the three-point line)

This collection is created for Query 3, it's a generated data is based on players minutes and positions since we couldn't find a real data related to players active positions per playing minutes during the game. These times are generated by looking at the positions of the players and giving them weights according to it. Guards usually play outside of the 3pt line, center's are mostly on the paint… etc. We have created a python code to generate this data.

```python
# Distribute time based on position
def distribute_time(player):
    total_seconds = convert_to_seconds(player['minutes'])
    if player['position'] == 'Guard':
        distribution = {
            "Left Wing": 0.3,
            "Right Wing": 0.3,
            "Left Corner": 0.05,
            "Right Corner": 0.05,
            "Paint": 0.1,
            "Top": 0.2
        }
    elif player['position'] == 'Forward':
        distribution = {
            "Left Wing": 0.2,
            "Right Wing": 0.2,
            "Left Corner": 0.1,
            "Right Corner": 0.1,
            "Paint": 0.3,
            "Top": 0.1
        }
    elif player['position'] == 'Center':
        distribution = {
            "Left Wing": 0.05,
            "Right Wing": 0.05,
            "Left Corner": 0.1,
            "Right Corner": 0.1,
            "Paint": 0.6,
            "Top": 0.1
        }
    time_spent = {section: int(total_seconds * percentage) for section, percentage in distribution.items()}
    return time_spent
```

6. passes

- **Purpose:** This collection records each pass made during the game, including the players involved and the timing of the pass.

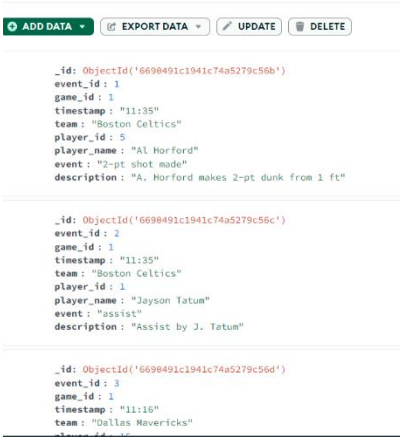
  We structured the passes data from the game, including details like passer, receiver, time, and quarter. Using Python, we stored this data in a DataFrame for easy analysis, calculating statistics such as the total number of passes between players. We used MongoDB to manage the data, inserting the pass data into a collection and performing query 4. This information helps in analyzing the flow of the game, identifying key passing patterns, and understanding how effectively the team moves the ball.

1. **_id**: A unique identifier for each record.
2. **pass_id**: A unique identifier for each pass event.
3. **game_id**: The identifier of the game during which the pass was made.
4. **event_type**: The type of event, in this case, "pass".
5. **timestamp**: The time at which the pass occurred during the game.
6. **quarter**: The quarter of the game in which the pass occurred.
7. **passing_player_id**: The identifier of the player who made the pass.
8. **receiving_player_id**: The identifier of the player who received the pass.

7. possesions

- **Purpose:** This collection tracks each significant event during the game that involves a player handling the ball, such as shots, assists, rebounds, fouls, and turnovers. This information is crucial for understanding player control, ball handling, and offensive efficiency. This data is not used for any query, but it is used for generating the necessary data for team possession query.

1. **_id**: A unique identifier for each player.
2. **event_id**: A unique identifier for each event in the game.
3. **game_id**: The identifier of the game during which the event occurred.
4. **timestamp**: The time at which the event occurred during the game.
5. **team**: The team to which the player belongs.
6. **player_id**: The identifier of the player involved in the event.
7. **player_name**: The name of the player involved in the event.
8. **event**: The type of event (e.g., shot made, assist, rebound).
9. **description**: A detailed description of the event.



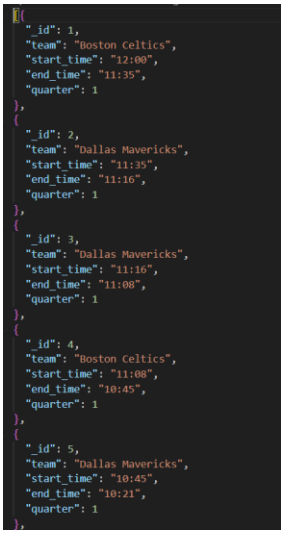
possesions

	timestamp String	team String	player_id Int32	player_name String	event String	description
1	"11:35"	"Boston Celtics"	5	"Al Horford"	"2-pt shot made"	"A. Horford m
2	"11:35"	"Boston Celtics"	1	"Jayson Tatum"	"assist"	"Assist by J.
3	"11:16"	"Dallas Mavericks"	15	"Derrick Jones Jr."	"3-pt shot made"	"D. Jones mak
4	"11:16"	"Dallas Mavericks"	16	"Daniel Gafford"	"assist"	"Assist by D.
5	"11:08"	"Boston Celtics"	4	"Jrue Holiday"	"3-pt shot made"	"J. Holiday m
6	"11:08"	"Boston Celtics"	3	"Derrick White"	"assist"	"Assist by D.
7	"10:45"	"Dallas Mavericks"	12	"Luka Doncic"	"2-pt shot made"	"L. Doncic ma
8	"9:47"	"Dallas Mavericks"	12	"Luka Doncic"	"defensive rebound"	"Defensive re
9	"9:46"	"Boston Celtics"	1	"Jayson Tatum"	"personal foul"	"Personal fou
10	"9:46"	"Dallas Mavericks"	12	"Luka Doncic"	"drawn foul"	"Foul drawn b
11	"9:30"	"Dallas Mavericks"	12	"Luka Doncic"	"2-pt shot missed"	"L. Doncic mi
12	"9:32"	"Dallas Mavericks"	16	"Daniel Gafford"	"offensive rebound"	"Offensive re
13	"9:32"	"Dallas Mavericks"	16	"Daniel Gafford"	"2-pt shot made"	"D. Gafford m
14	"9:19"	"Dallas Mavericks"	14	"P.J. Washington Jr."	"2-pt shot made"	"P. Washingto
15	"9:19"	"Dallas Mavericks"	12	"Luka Doncic"	"assist"	"Assist by L.

8. team_possesions

- To create the `team_possessions` dataset, we extracted play-by-play events from the game data and identified key events indicating possession changes, such as made shots, missed shots followed by rebounds, and turnovers. Using Python, we calculated the start and end times for each possession by iterating through these events. we also accounted for quarter changes, resetting the start time at the beginning of each quarter. This process allowed me to accurately determine and document when each team had possession of the ball throughout the game.

- **Purpose:** This collection captures detailed information about when and for how long each team has possession of the ball during a basketball game. This data is critical for analyzing team control, understanding game flow, and evaluating overall team performance.



1. **_id**: A unique identifier for each record.
2. **team**: The name of the team having possession of the ball.
3. **start_time**: The timestamp when the team gained possession.
4. **end_time**: The timestamp when the team lost possession.
5. **quarter**: The quarter of the game in which the possession occurred.

team_possesions

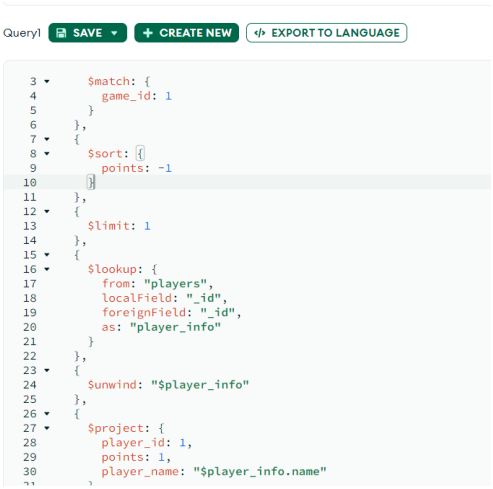
	_id Int32	team String	start_time String	end_time String	quarter Int32
1	1	"Boston Celtics"	"12:00"	"11:35"	1
2	2	"Dallas Mavericks"	"11:35"	"11:16"	1
3	3	"Dallas Mavericks"	"11:16"	"11:08"	1
4	4	"Boston Celtics"	"11:08"	"10:45"	1
5	5	"Dallas Mavericks"	"10:45"	"10:21"	1
6	6	"Boston Celtics"	"10:21"	"10:18"	1
7	7	"Boston Celtics"	"10:18"	"10:16"	1
8	8	"Boston Celtics"	"10:16"	"10:03"	1
9	9	"Boston Celtics"	"10:03"	"10:02"	1
10	10	"Boston Celtics"	"10:02"	"9:58"	1
11	11	"Boston Celtics"	"9:58"	"9:57"	1
12	12	"Dallas Mavericks"	"9:57"	"9:49"	1
13	13	"Dallas Mavericks"	"9:49"	"9:47"	1
14	14	"Dallas Mavericks"	"9:47"	"9:46"	1
15	15	"Boston Celtics"	"9:46"	"9:33"	1

Data Queries

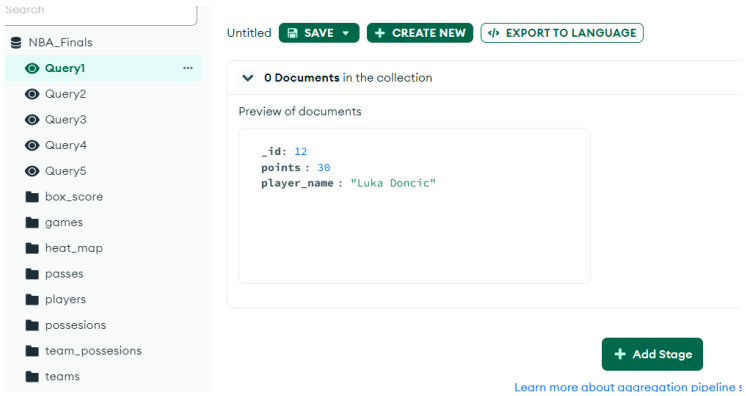
Query 1: Which player had the highest scoring in the game?

Steps:

1. **Match the Game**: Select records from the `box_score` collection where `game_id` is 1.
2. **Sort by Points**: Sort the records in descending order based on the `points` field.
3. **Limit the Results**: Limit the results to the top record (highest points).
4. **Join with Players Collection**: Use the `$lookup` stage to join with the `players` collection to fetch player details.
5. **Unwind the Joined Data**: Use `$unwind` to flatten the array returned by the `$lookup`.
6. **Project Fields**: Select and project relevant fields, including `player_id`, `points`, and `player_name`.



Result: Luka Doncic is the highest points along with their name and points scored.



Query 2: How many fouls did each player commit during the game?

Steps:

1. **Match the Game:** Select records from the box_score collection where game_id is 1.
2. **Join with Players Collection:** Use the $lookup stage to join with the players collection to fetch player details.
3. **Unwind the Joined Data:** Use $unwind to flatten the array returned by the $lookup.
4. **Project Fields:** Select and project relevant fields, including player_id, personal_fouls, and player_name.
5. **Sort by Fouls:** Sort the records in descending order based on the personal_fouls field.

Result: A list of all the players' number of fouls committed, sorted in descending order.

Query2

	_id Int32	personal_fouls Int32	player_name String
1	12	4	"Luka Doncic"
2	2	3	"Jaylen Brown"
3	5	2	"Al Horford"
4	3	2	"Derrick White"
5	13	2	"Kyrie Irving"
6	7	2	"Sam Hauser"
7	21	1	"Tim Hardaway Jr."
8	4	1	"Jrue Holiday"
9	9	1	"Luke Kornet"
10	10	1	"Svi Mykhailiuk"
11	20	1	"Jaden Hardy"
12	6	1	"Kristaps Porzingis"
13	22	1	"Dante Exum"

Query2

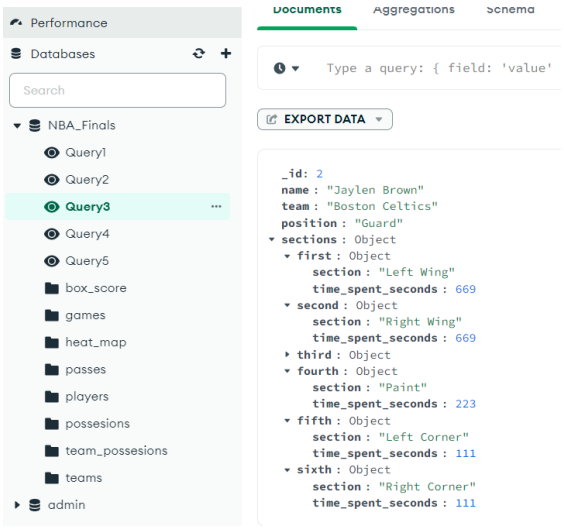
	_id Int32	personal_fouls Int32	player_name String
14	1	1	"Jayson Tatum"
15	15	1	"Derrick Jones Jr."
16	16	0	"Daniel Gafford"
17	8	0	"Payton Pritchard"
18	17	0	"Josh Green"
19	18	0	"Maxi Kleber"
20	19	0	"Dereck Lively II"
21	18	0	"Maxi Kleber"
22	19	0	"Dereck Lively II"
23	23	0	"Dwight Powell"

Query 3: Which positions on the court did a specific player spend the most time in?

This query determines the amount of time a specific player spends in various positions on the court. We have picked the player Jaylen Brown for displaying the results. Jaylen Brown's players id is 2.

Steps:

1. **Match the Player:** Select records from the heat_map collection where _id is 2.
2. **Project Fields:** Project relevant fields, including name, team, position, and time_spent.
3. **Convert Object to Array:** Use $addFields to convert the time_spent object to an array for easier processing.
4. **Unwind the Array:** Use $unwind to flatten the total_time_spent array.
5. **Sort by Time Spent:** Sort the array in descending order based on time spent.
6. **Group and Restructure:** Group by _id and restructure the array back to an object with position names and time spent.



Result: Detailed information about the time Jayen Brown spent in different positions on the court. He has spent most of his time in Left- and Right-Wing positions and least in corners. Since he is a guard for the Boston Celtics, this generated dataset provides a realistic result.

Query 4: How many times was the ball passed between two specific players?

We have picked Payton Pritchard and Derrick White for this query to see how many passes they made in between each other.

Steps:

1. **Match the Passes:** Select records from the passes collection where passing_player_id is 3 and receiving_player_id is 8, or vice versa.
2. **Group and Count Passes:** Group by passing_player_id and receiving_player_id and count the total number of passes.
3. **Join with Players Collection:** Use $lookup to join with the players collection to fetch player details for both passing and receiving players.
4. **Unwind the Joined Data:** Use $unwind to flatten the arrays returned by the $lookup.
5. **Project Fields:** Select and project relevant fields, including passing_player_name, receiving_player_name, and total_passes

Result: The total number of passes between the Payton Pritchard and Derrick White is 8.

Query 5: At what times during the game did a specific team have possession of the ball?

This query retrieves all possessions of a specific team during the game. We have picked Dallas Mavericks to display when they had the ball possession.

Steps:

1. **Match the Team:** Select records from the team_possesions collection where team is "Dallas Mavericks".
2. **Project Fields:** Select and project relevant fields, including team, start_time, end_time, and quarter.

```
[
  {
    $match: {
      team: "Dallas Mavericks"
    }
  },
  {
    $project: {
      _id: 0,
      team: 1,
      start_time: 1,
      end_time: 1,
      quarter: 1
    }
  }
]
```

Result: A list of all possessions by Dallas Mavericks, including the start and end times and the quarters in which they occurred. During the game Dallas had the possession 105 times.

Query5

	team String	start_time String	end_time String	quarter Int32
1	"Dallas Mavericks"	"11:35"	"11:16"	1
2	"Dallas Mavericks"	"11:16"	"11:08"	1
3	"Dallas Mavericks"	"10:45"	"10:21"	1
4	"Dallas Mavericks"	"9:57"	"9:49"	1
5	"Dallas Mavericks"	"9:49"	"9:47"	1
6	"Dallas Mavericks"	"9:47"	"9:46"	1
7	"Dallas Mavericks"	"9:33"	"9:32"	1
8	"Dallas Mavericks"	"9:32"	"9:32"	1
9	"Dallas Mavericks"	"9:32"	"9:19"	1
10	"Dallas Mavericks"	"9:13"	"9:04"	1
11	"Dallas Mavericks"	"8:45"	"8:40"	1
12	"Dallas Mavericks"	"8:25"	"8:08"	1
13	"Dallas Mavericks"	"7:40"	"7:33"	1
14	"Dallas Mavericks"	"7:33"	"7:09"	1
15	"Dallas Mavericks"	"7:09"	"6:38"	1

References

- [1] Basketball Reference. Play-by-Play Data for NBA Finals Game 1, Boston Celtics vs. Dallas Mavericks. Retrieved from https://www.basketball-reference.com/boxscores/pbp/202406060BOS.html.
- [2] Chodorow, K. (2013). MongoDB: The Definitive Guide. O'Reilly Media. ISBN 1449344682.
- [3] Alamar, B. (2013). Sports Analytics: A Guide for Coaches, Managers, and Other Decision Makers. Columbia University Press. ISBN 0231162927.