BUAN 6320 - Database Foundation for Business Analytics

Group 8 – Final Project Report

Dataset link (Football Data):

https://www.kaggle.com/datasets/mexwell/football-data-from-transfermarkt?select=players.csv

Content of the dataset:

The football dataset from Transfermarkt is a comprehensive and dynamically updated collection of football-related information spanning various seasons, competitions, clubs, and players. This rich dataset offers information for conducting in-depth analyses, uncovering trends, and gaining insights into the world of football. The dataset contains 8 csv files, each gives us different type of information including past results of football tournaments, past player performance and events. Brief information about data in each CSV file of the dataset is given below.

1. Competitions:
   - Types of competitions (league, cup, etc.).
   - Country names hosting the competitions.

2. Clubs:
   - Squad size.
   - Average age of players (average age of the squad).
   - Stadium details (capacity, last season played).
   - Free-kick statistics.

3. Club Games:
   - No primary key.
   - Goals scored by the home and away teams.
   - Indication of the winner (0 for home team, 1 for away team).
   - Host information linked to the Clubs table.

4. Games:
   - Game ID and season.
   - Match type (quarterfinals, semifinals, finals).
   - Date of the match.
   - Goals scored by the home and away clubs.
   - Stadium details and attendance.
   - Aggregate goals (Home: Opponent format).

5. Players:
- Player names.
- Last season played by each player.
- Current club ID.
- Player positions (attacker, defender, midfielder, goalkeeper).
- Highest market value in Euros.

6. Players Valuation:
- Valuation date (relevant to player valuation changes).
- Market values corresponding to the given date.

7. Appearance:
- Player name (related to individual games).
- Minutes played by each player.
- Red and yellow card information.
- Goals and assists by players.

8. Events in Games (Game Events):
- Minute of event occurrence.
- Types of events (substitution, goal).
- Details of player substitution (player in ID and player out ID).

Business Objective:

Leverage sophisticated football data analytics to strategically curate a diverse player pool, enhancing our European football team's on-field performance. The core objective is to elevate revenue streams and foster strategic growth by making well-informed decisions in the dynamic player acquisition landscape. The focal points encompass meticulous player scouting, insightful market trend analyses, judicious budget allocation, astute auction bidding strategies, and the overarching goal of building a resilient and competitive team for sustained success. This initiative seeks to establish our club as a formidable force, not only in sporting arenas but also in financial viability, all while fostering meaningful connections with fans through engaging and exciting team dynamics. Continuous scrutiny of data will be integral to evolving recruitment strategies, ensuring our club's enduring success in both local and international competitions.

Project Insights:
There are some questions that the club management has to ask itself to make sure that they have a pool of players to choose from in the upcoming auction. Questions such as:

1. What are the yearly goal records for each team? (We are choosing Southampton FC as their finances are dropped by 50%)

2. Top 20 player group by position based on average market value

3. Top 20 players for each player category ranked based on market value

4. What is the relationship between player appearances and their respective market values?

5. How is market valuation determined based on a player's position? To determine how much money, we need to put on players based on position.

6. Attacker evaluating criteria

7. Midfielder evaluating criteria

8. How many goals does each player scored and what is the maximum goals per match scored by the player?

9. Successful passes by players based on position

10. Defender evaluating criteria

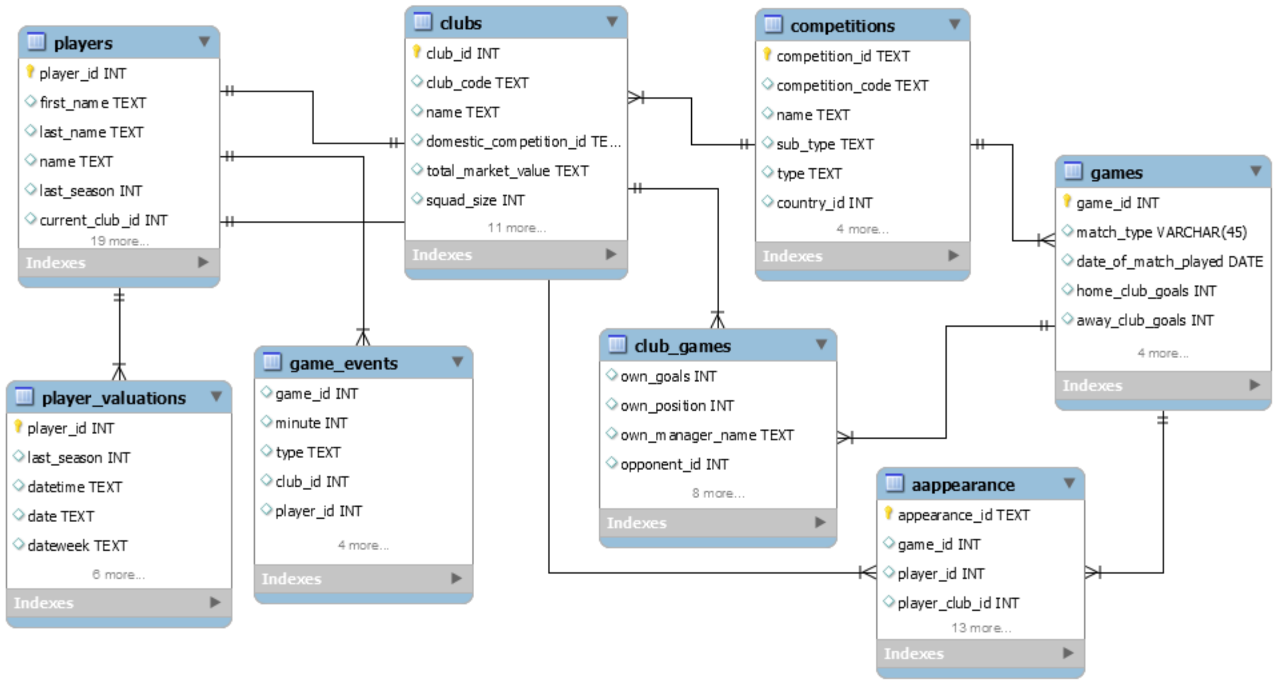
11. Evaluating criteria for goalkeeper

-- MySQL Workbench Forward Engineering

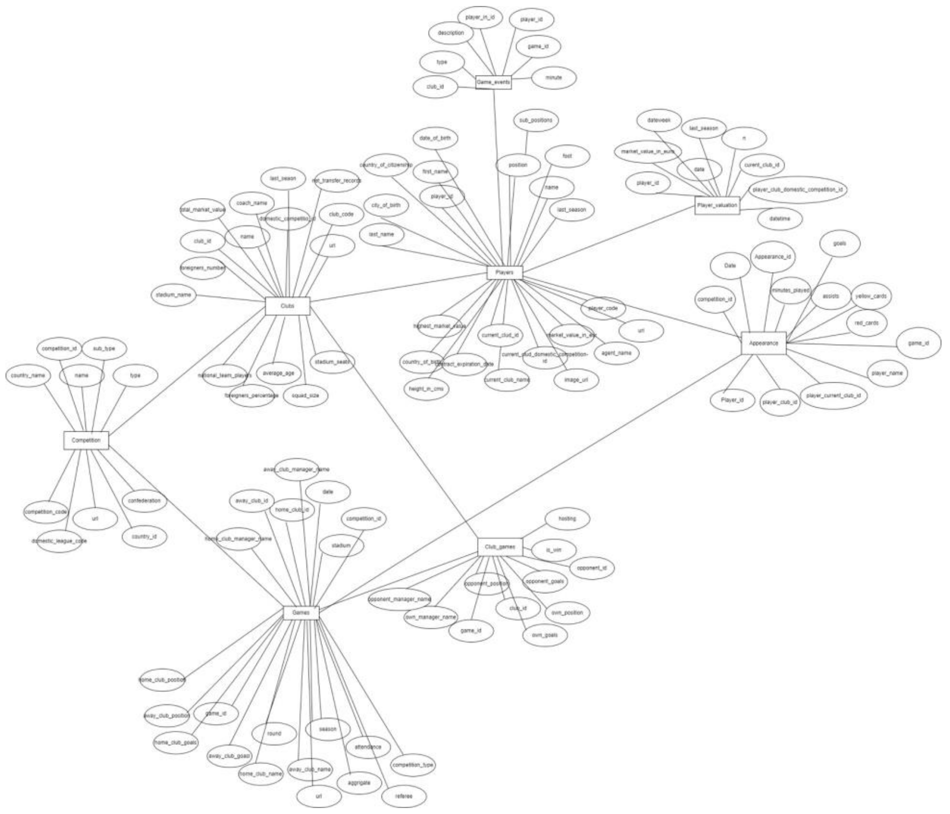
```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE
,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';



-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema football
-- -----------------------------------------------------



-- -----------------------------------------------------
-- Schema football
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `football` DEFAULT CHARACTER SET utf8mb4
```

```
COLLATE utf8mb4_0900_ai_ci ;
USE `football` ;



-- -----------------------------------------------------
-- Table `football`.`competitions`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`competitions` (
  `competition_id` TEXT NOT NULL,
  `competition_code` TEXT NULL DEFAULT NULL,
  `name` TEXT NULL DEFAULT NULL,
  `sub_type` TEXT NULL DEFAULT NULL,
  `type` TEXT NULL DEFAULT NULL,
  `country_id` INT NULL DEFAULT NULL,
  `country_name` TEXT NULL DEFAULT NULL,
  `domestic_league_code` TEXT NULL DEFAULT NULL,
  `confederation` TEXT NULL DEFAULT NULL,
  `url` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`competition_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;




-- -----------------------------------------------------
-- Table `football`.`games`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`games` (
  `game_id` INT NOT NULL,
  `match_type` VARCHAR(45) NULL,
  `date_of_match_played` DATE NULL,
  `home_club_goals` INT NULL,
  `away_club_goals` INT NULL,
  `stadium` VARCHAR(45) NULL,
  `attendance` INT NULL,
  `aggregate_goals` INT NULL,
  `competitions_competition_id` TEXT NOT NULL,
  PRIMARY KEY (`game_id`, `competitions_competition_id`),
  INDEX `fk_games_competitions_idx` (`competitions_competition_id` ASC) VISIBLE,
  CONSTRAINT `fk_games_competitions`
    FOREIGN KEY (`competitions_competition_id`)
    REFERENCES `football`.`competitions` (`competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;




-- -----------------------------------------------------
-- Table `football`.`clubs`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`clubs` (
  `club_id` INT NOT NULL,
  `club_code` TEXT NULL DEFAULT NULL,
  `name` TEXT NULL DEFAULT NULL,
  `domestic_competition_id` TEXT NULL DEFAULT NULL,
  `total_market_value` TEXT NULL DEFAULT NULL,
  `squad_size` INT NULL DEFAULT NULL,
  `average_age` DOUBLE NULL DEFAULT NULL,
  `foreigners_number` INT NULL DEFAULT NULL,
  `foreigners_percentage` DOUBLE NULL DEFAULT NULL,
  `national_team_players` INT NULL DEFAULT NULL,
  `stadium_name` TEXT NULL DEFAULT NULL,
  `stadium_seats` INT NULL DEFAULT NULL,
  `net_transfer_record` TEXT NULL DEFAULT NULL,
  `coach_name` TEXT NULL DEFAULT NULL,
  `last_season` INT NULL DEFAULT NULL,
  `url` TEXT NULL DEFAULT NULL,
  `competitions_competition_id` TEXT NOT NULL,
  PRIMARY KEY (`club_id`, `competitions_competition_id`),
  INDEX `fk_clubs_competitions1_idx` (`competitions_competition_id` ASC) VISIBLE,
  CONSTRAINT `fk_clubs_competitions1`
    FOREIGN KEY (`competitions_competition_id`)
    REFERENCES `football`.`competitions` (`competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;




-- -----------------------------------------------------
-- Table `football`.`players`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`players` (
  `player_id` INT NOT NULL,
  `first_name` TEXT NULL DEFAULT NULL,
  `last_name` TEXT NULL DEFAULT NULL,
  `name` TEXT NULL DEFAULT NULL,
  `last_season` INT NULL DEFAULT NULL,
  `current_club_id` INT NULL DEFAULT NULL,
  `player_code` TEXT NULL DEFAULT NULL,
  `country_of_birth` TEXT NULL DEFAULT NULL,
  `city_of_birth` TEXT NULL DEFAULT NULL,
  `country_of_citizenship` TEXT NULL DEFAULT NULL,
  `date_of_birth` TEXT NULL DEFAULT NULL,
  `sub_position` TEXT NULL DEFAULT NULL,
  `position` TEXT NULL DEFAULT NULL,
  `foot` TEXT NULL DEFAULT NULL,
  `height_in_cm` TEXT NULL DEFAULT NULL,
  `market_value_in_eur` TEXT NULL DEFAULT NULL,
  `highest_market_value_in_eur` INT NULL DEFAULT NULL,
  `contract_expiration_date` TEXT NULL DEFAULT NULL,
  `agent_name` TEXT NULL DEFAULT NULL,
  `image_url` TEXT NULL DEFAULT NULL,
  `url` TEXT NULL DEFAULT NULL,
  `current_club_domestic_competition_id` TEXT NULL DEFAULT NULL,
  `current_club_name` TEXT NULL DEFAULT NULL,
  `clubs_club_id` INT NOT NULL,
  `clubs_competitions_competition_id` TEXT NOT NULL,
  PRIMARY KEY (`clubs_club_id`, `clubs_competitions_competition_id`, `player_id`),
  CONSTRAINT `fk_players_clubs1`
    FOREIGN KEY (`clubs_club_id` , `clubs_competitions_competition_id`)
    REFERENCES `football`.`clubs` (`club_id` , `competitions_competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;




-- -----------------------------------------------------
-- Table `football`.`aappearance`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`aappearance` (
  `appearance_id` TEXT NOT NULL,
  `game_id` INT NULL DEFAULT NULL,
  `player_id` INT NULL DEFAULT NULL,
  `player_club_id` INT NULL DEFAULT NULL,
  `player_current_club_id` INT NULL DEFAULT NULL,
  `date` TEXT NULL DEFAULT NULL,
  `player_name` TEXT NULL DEFAULT NULL,
  `competition_id` TEXT NULL DEFAULT NULL,
  `yellow_cards` INT NULL DEFAULT NULL,
  `red_cards` INT NULL DEFAULT NULL,
  `goals` INT NULL DEFAULT NULL,
  `assists` INT NULL DEFAULT NULL,
  `minutes_played` INT NULL DEFAULT NULL,
  `games_game_id` INT NOT NULL,
  `games_competitions_competition_id` TEXT NOT NULL,
  `players_clubs_club_id` INT NOT NULL,
  `players_clubs_competitions_competition_id` TEXT NOT NULL,
  PRIMARY KEY (`appearance_id`, `games_game_id`, `games_competitions_competition_id`,
`players_clubs_club_id`, `players_clubs_competitions_competition_id`),
  INDEX `fk_aappearance_games1_idx` (`games_game_id` ASC,
`games_competitions_competition_id` ASC) VISIBLE,
  INDEX `fk_aappearance_players1_idx` (`players_clubs_club_id` ASC,
`players_clubs_competitions_competition_id` ASC) VISIBLE,
  CONSTRAINT `fk_aappearance_games1`
    FOREIGN KEY (`games_game_id` , `games_competitions_competition_id`)
    REFERENCES `football`.`games` (`game_id` , `competitions_competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_aappearance_players1`
    FOREIGN KEY (`players_clubs_club_id` , `players_clubs_competitions_competition_id`)
    REFERENCES `football`.`players` (`clubs_club_id` , `clubs_competitions_competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;




-- -----------------------------------------------------
-- Table `football`.`club_games`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`club_games` (
  `own_goals` INT NULL DEFAULT NULL,
  `own_position` INT NULL DEFAULT NULL,
  `own_manager_name` TEXT NULL DEFAULT NULL,
  `opponent_id` INT NULL DEFAULT NULL,
  `opponent_goals` INT NULL DEFAULT NULL,
  `opponent_position` INT NULL DEFAULT NULL,
  `opponent_manager_name` TEXT NULL DEFAULT NULL,
  `hosting` TEXT NULL DEFAULT NULL,
  `is_win` INT NULL DEFAULT NULL,
  `games_game_id` INT NOT NULL,
  `games_competitions_competition_id` TEXT NOT NULL,
  `clubs_club_id` INT NOT NULL,
  PRIMARY KEY (`games_game_id`, `games_competitions_competition_id`, `clubs_club_id`),
  INDEX `fk_club_games_clubs1_idx` (`clubs_club_id` ASC) VISIBLE,
  CONSTRAINT `fk_club_games_games1`
    FOREIGN KEY (`games_game_id` , `games_competitions_competition_id`)
    REFERENCES `football`.`games` (`game_id` , `competitions_competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_club_games_clubs1`
    FOREIGN KEY (`clubs_club_id`)
    REFERENCES `football`.`clubs` (`club_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;




-- -----------------------------------------------------
-- Table `football`.`game_events`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`game_events` (
  `game_id` INT NULL DEFAULT NULL,
  `minute` INT NULL DEFAULT NULL,
  `type` TEXT NULL DEFAULT NULL,
  `club_id` INT NULL DEFAULT NULL,
  `player_id` INT NULL DEFAULT NULL,
  `description` TEXT NULL DEFAULT NULL,
  `player_in_id` INT NULL DEFAULT NULL,
  `players_clubs_club_id` INT NOT NULL,
  `players_clubs_competitions_competition_id` TEXT NOT NULL,
  PRIMARY KEY (`players_clubs_club_id`, `players_clubs_competitions_competition_id`),
  CONSTRAINT `fk_game_events_players1`
    FOREIGN KEY (`players_clubs_club_id` , `players_clubs_competitions_competition_id`)
    REFERENCES `football`.`players` (`clubs_club_id` , `clubs_competitions_competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;




-- -----------------------------------------------------
-- Table `football`.`player_valuations`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `football`.`player_valuations` (
  `player_id` INT NOT NULL,
  `last_season` INT NULL DEFAULT NULL,
  `datetime` TEXT NULL DEFAULT NULL,
  `date` TEXT NULL DEFAULT NULL,
  `dateweek` TEXT NULL DEFAULT NULL,
  `market_value_in_eur` INT NULL DEFAULT NULL,
  `n` INT NULL DEFAULT NULL,
  `current_club_id` INT NULL DEFAULT NULL,
  `player_club_domestic_competition_id` TEXT NULL DEFAULT NULL,
  `players_clubs_club_id` INT NOT NULL,
  `players_clubs_competitions_competition_id` TEXT NOT NULL,
  PRIMARY KEY (`players_clubs_club_id`, `players_clubs_competitions_competition_id`,
`player_id`),
  CONSTRAINT `fk_player_valuations_players1`
    FOREIGN KEY (`players_clubs_club_id` , `players_clubs_competitions_competition_id`)
    REFERENCES `football`.`players` (`clubs_club_id` , `clubs_competitions_competition_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Logical Model:



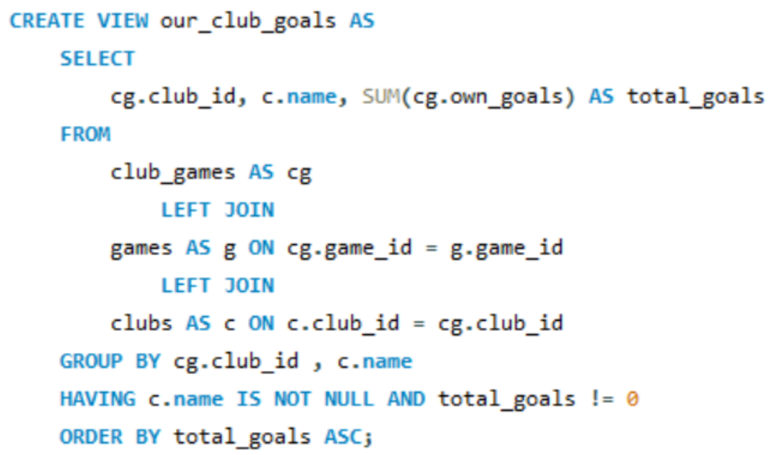
Conceptual Model:



Data Loading Concept Used: We Used Data Import method.

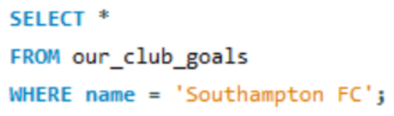
Final Outcome:

1. What are the yearly goal records for each team? (We are choosing Southampton FC as their finances are dropped by 50%)

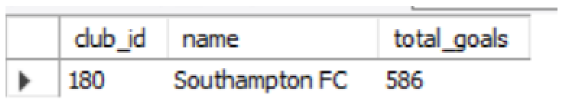


```
CREATE VIEW our_club_goals AS
    SELECT
        cg.club_id, c.name, SUM(cg.own_goals) AS total_goals
    FROM
        club_games AS cg
            LEFT JOIN
        games AS g ON cg.game_id = g.game_id
            LEFT JOIN
        clubs AS c ON c.club_id = cg.club_id
    GROUP BY cg.club_id , c.name
    HAVING c.name IS NOT NULL AND total_goals != 0
    ORDER BY total_goals ASC;
```

Final



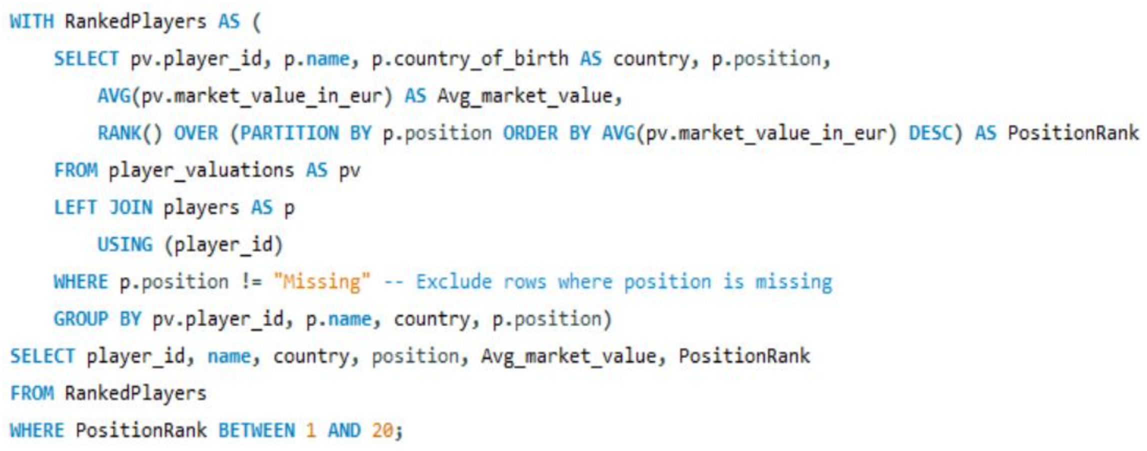
```
SELECT *
FROM our_club_goals
WHERE name = 'Southampton FC';
```



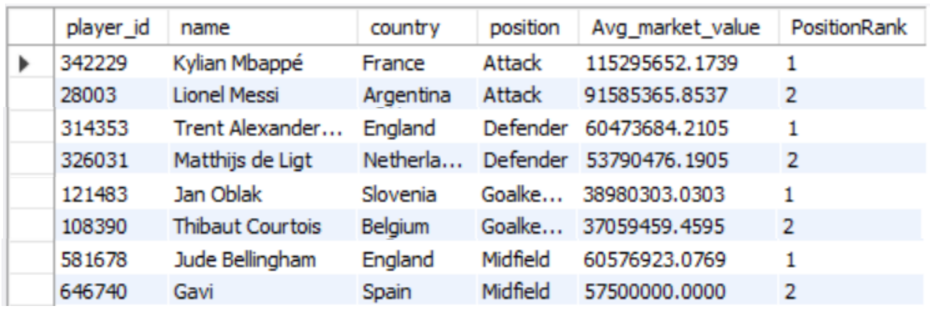
	club_id	name	total_goals
▶	180	Southampton FC	586

Recommendation: The total goals of South Hampton club (our club) are 586.

2. Top 20 player group by position based on average market value



```
WITH RankedPlayers AS (
    SELECT pv.player_id, p.name, p.country_of_birth AS country, p.position,
        AVG(pv.market_value_in_eur) AS Avg_market_value,
        RANK() OVER (PARTITION BY p.position ORDER BY AVG(pv.market_value_in_eur) DESC) AS PositionRank
    FROM player_valuations AS pv
    LEFT JOIN players AS p
        USING (player_id)
    WHERE p.position != "Missing" -- Exclude rows where position is missing
    GROUP BY pv.player_id, p.name, country, p.position)
SELECT player_id, name, country, position, Avg_market_value, PositionRank
FROM RankedPlayers
WHERE PositionRank BETWEEN 1 AND 20;
```

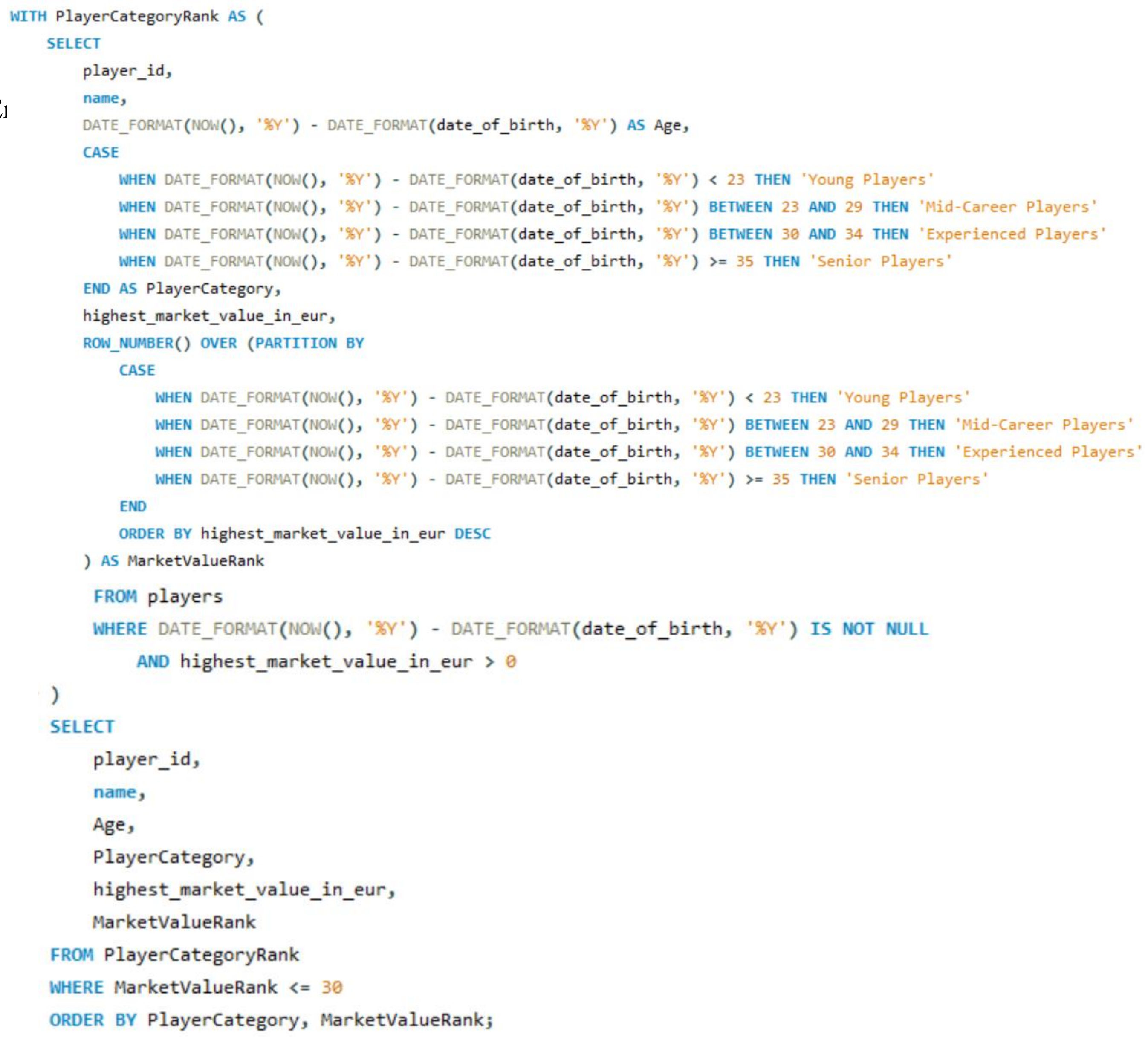


	player_id	name	country	position	Avg_market_value	PositionRank
▶	342229	Kylian Mbappé	France	Attack	115295652.1739	1
	28003	Lionel Messi	Argentina	Attack	91585365.8537	2
	314353	Trent Alexander...	England	Defender	60473684.2105	1
	326031	Matthijs de Ligt	Netherla...	Defender	53790476.1905	2
	121483	Jan Oblak	Slovenia	Goalke...	38980303.0303	1
	108390	Thibaut Courtois	Belgium	Goalke...	37059459.4595	2
	581678	Jude Bellingham	England	Midfield	60576923.0769	1
	646740	Gavi	Spain	Midfield	57500000.0000	2

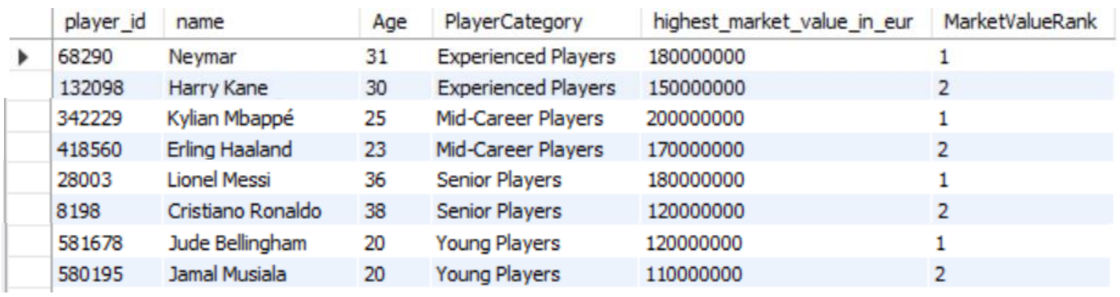
Final Recommendation: Here we found out and identified top 20 players based upon average market value. The category here is position on which the player plays in the field (Attack, Defence, Mid-field, Goalkeeper). Thus creating a pool of 80 players.

3. Top 20 players for each player category ranked based on market value

E



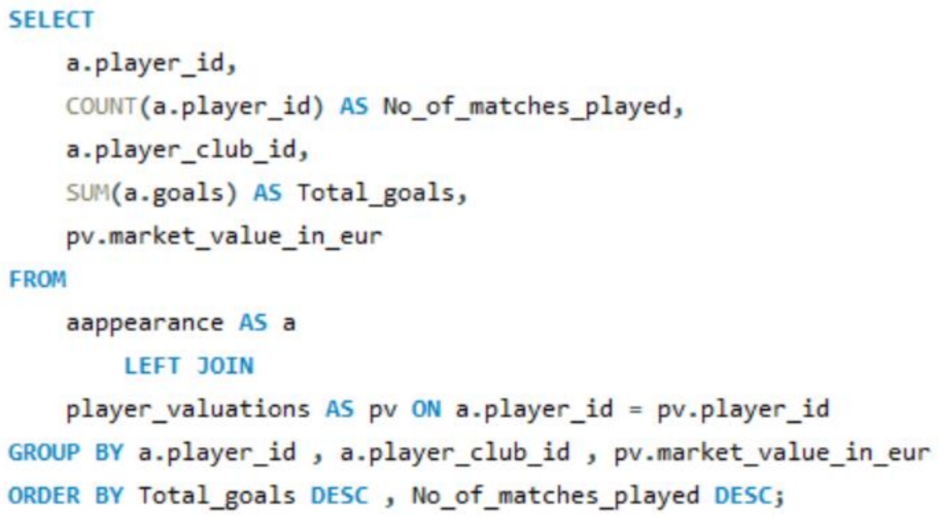
```
WITH PlayerCategoryRank AS (
    SELECT
        player_id,
        name,
        DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') AS Age,
        CASE
            WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') < 23 THEN 'Young Players'
            WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') BETWEEN 23 AND 29 THEN 'Mid-Career Players'
            WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') BETWEEN 30 AND 34 THEN 'Experienced Players'
            WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') >= 35 THEN 'Senior Players'
        END AS PlayerCategory,
        highest_market_value_in_eur,
        ROW_NUMBER() OVER (PARTITION BY
            CASE
                WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') < 23 THEN 'Young Players'
                WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') BETWEEN 23 AND 29 THEN 'Mid-Career Players'
                WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') BETWEEN 30 AND 34 THEN 'Experienced Players'
                WHEN DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') >= 35 THEN 'Senior Players'
            END
            ORDER BY highest_market_value_in_eur DESC
        ) AS MarketValueRank
        FROM players
        WHERE DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') IS NOT NULL
            AND highest_market_value_in_eur > 0
)
SELECT
    player_id,
    name,
    Age,
    PlayerCategory,
    highest_market_value_in_eur,
    MarketValueRank
FROM PlayerCategoryRank
WHERE MarketValueRank <= 30
ORDER BY PlayerCategory, MarketValueRank;
```



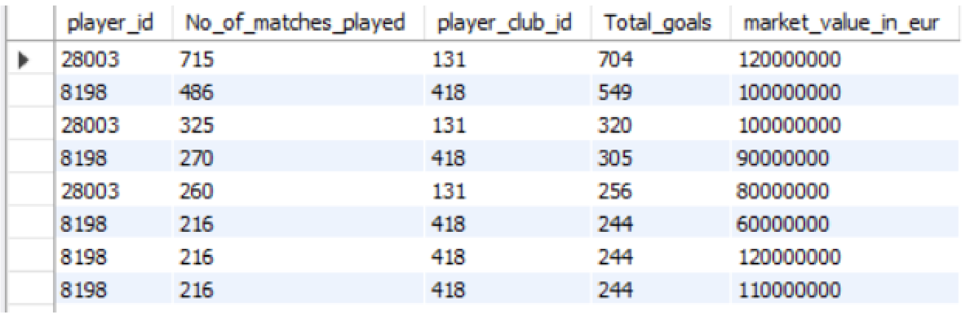
player_id	name	Age	PlayerCategory	highest_market_value_in_eur	MarketValueRank
68290	Neymar	31	Experienced Players	180000000	1
132098	Harry Kane	30	Experienced Players	150000000	2
342229	Kylian Mbappé	25	Mid-Career Players	200000000	1
418560	Erling Haaland	23	Mid-Career Players	170000000	2
28003	Lionel Messi	36	Senior Players	180000000	1
8198	Cristiano Ronaldo	38	Senior Players	120000000	2
581678	Jude Bellingham	20	Young Players	120000000	1
580195	Jamal Musiala	20	Young Players	110000000	2

Final Recommendation: Here we found and identified top 20 players in each category and ranked based upon highest market value in euros. We also categorized players in 4 groups based on their age. Players less than age 23 were categorized as “Young Players”. Players aged between 23 and 29 were categorized as “Mid-career players”. Players aged between 30 and 34 were categorized as “Experienced players”. Players aged more than 35 were categorized as “Senior Players”.

4. What is the relationship between player appearances and their respective market values?



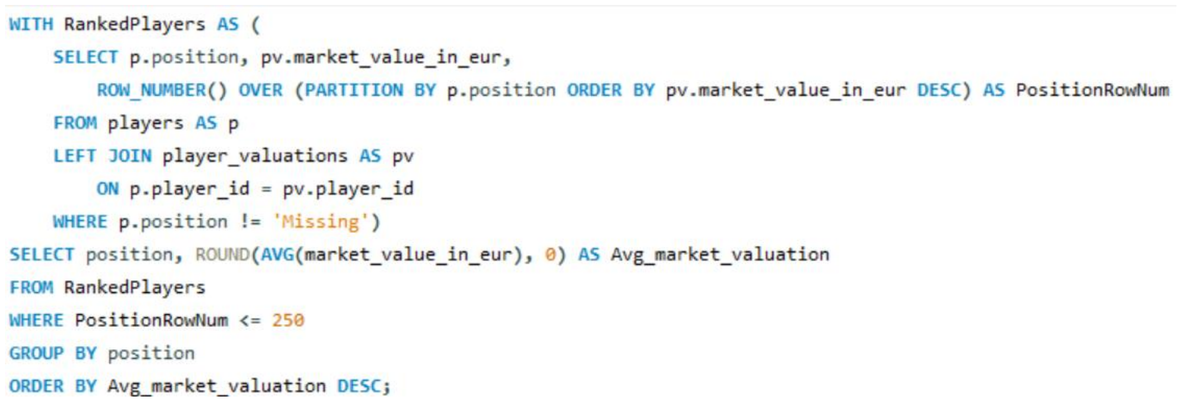
```
SELECT
    a.player_id,
    COUNT(a.player_id) AS No_of_matches_played,
    a.player_club_id,
    SUM(a.goals) AS Total_goals,
    pv.market_value_in_eur
FROM
    aappearance AS a
        LEFT JOIN
    player_valuations AS pv ON a.player_id = pv.player_id
GROUP BY a.player_id , a.player_club_id , pv.market_value_in_eur
ORDER BY Total_goals DESC , No_of_matches_played DESC;
```



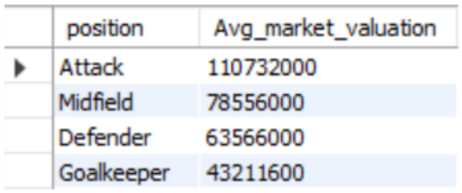
player_id	No_of_matches_played	player_club_id	Total_goals	market_value_in_eur
28003	715	131	704	120000000
8198	486	418	549	100000000
28003	325	131	320	100000000
8198	270	418	305	90000000
28003	260	131	256	80000000
8198	216	418	244	60000000
8198	216	418	244	120000000
8198	216	418	244	110000000

Final Recommendation: Relationship between player appearances and market values gives us an idea about how market value of a player changes over the years. It also gives us an idea how often has player switched clubs and the impact of that on his valuation.

5. How is market valuation determined based on a player's position? To determine how much money, we need to put on players based on position.



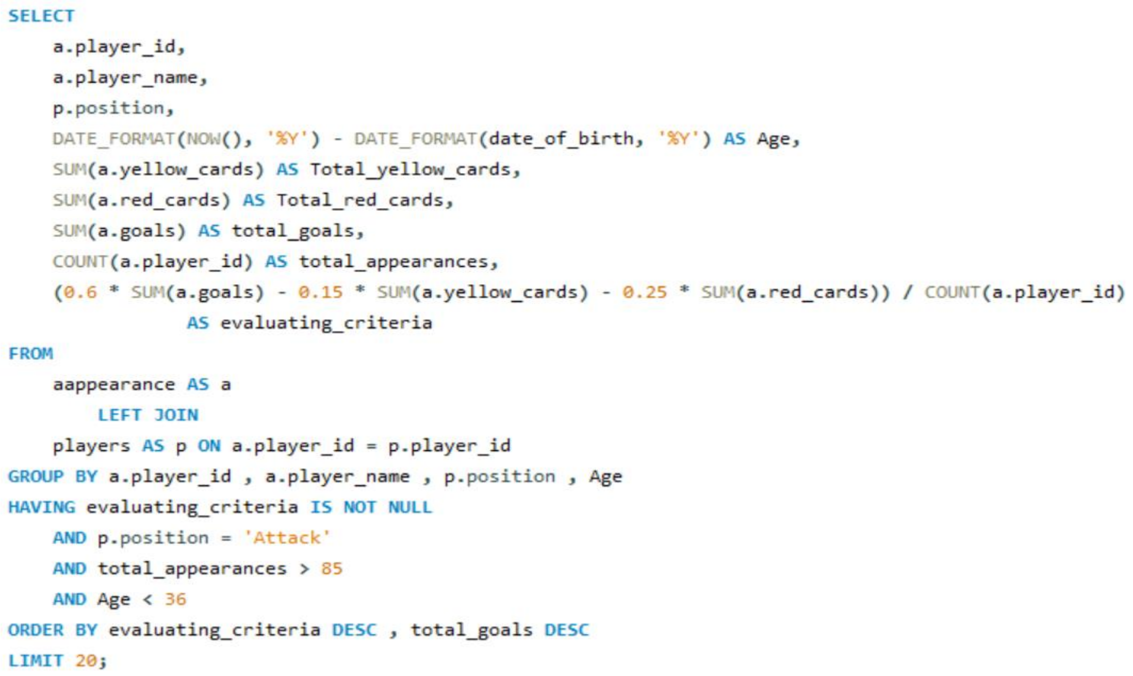
```
WITH RankedPlayers AS (
    SELECT p.position, pv.market_value_in_eur,
        ROW_NUMBER() OVER (PARTITION BY p.position ORDER BY pv.market_value_in_eur DESC) AS PositionRowNum
    FROM players AS p
    LEFT JOIN player_valuations AS pv
        ON p.player_id = pv.player_id
    WHERE p.position != 'Missing')
SELECT position, ROUND(AVG(market_value_in_eur), 0) AS Avg_market_valuation
FROM RankedPlayers
WHERE PositionRowNum <= 250
GROUP BY position
ORDER BY Avg_market_valuation DESC;
```



	position	Avg_market_valuation
▶	Attack	110732000
	Midfield	78556000
	Defender	63566000
	Goalkeeper	43211600

Final Recommendation: This gives us an idea about how average market valuation of each position type of player. Attack position type players are on an average more expensive than others. So it gives us an idea how do we want to allot a budget for players playing in a particular type of position.

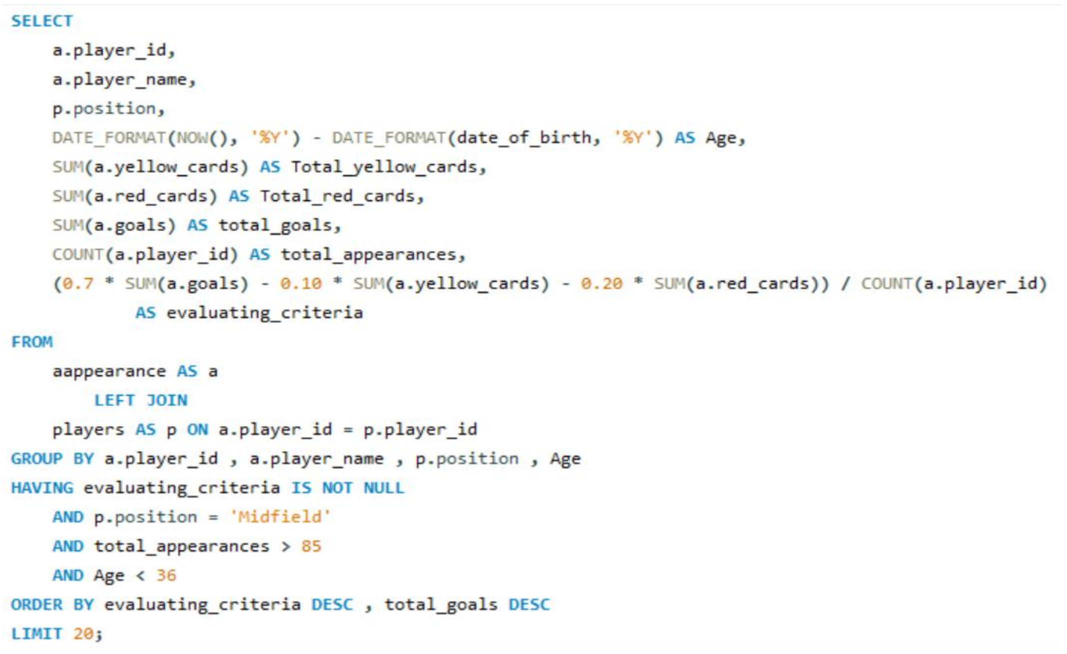
6. Attacker evaluating criteria



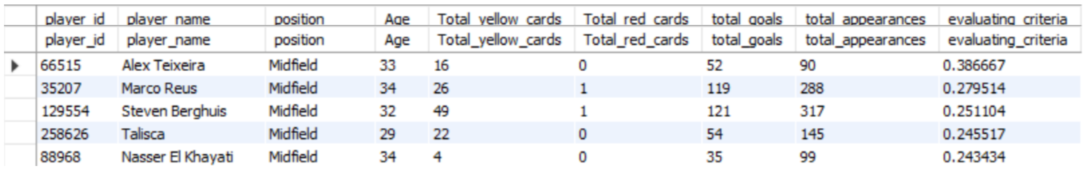
```
SELECT
    a.player_id,
    a.player_name,
    p.position,
    DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') AS Age,
    SUM(a.yellow_cards) AS Total_yellow_cards,
    SUM(a.red_cards) AS Total_red_cards,
    SUM(a.goals) AS total_goals,
    COUNT(a.player_id) AS total_appearances,
    (0.6 * SUM(a.goals) - 0.15 * SUM(a.yellow_cards) - 0.25 * SUM(a.red_cards)) / COUNT(a.player_id)
                AS evaluating_criteria
FROM
    aappearance AS a
        LEFT JOIN
    players AS p ON a.player_id = p.player_id
GROUP BY a.player_id , a.player_name , p.position , Age
HAVING evaluating_criteria IS NOT NULL
    AND p.position = 'Attack'
    AND total_appearances > 85
    AND Age < 36
ORDER BY evaluating_criteria DESC , total_goals DESC
LIMIT 20;
```

Final Recommendation: Here, we have created a criteria for player selection for the “Attack” position. We’ve given 60% weightage to goals scored. Similarly we have given 15% and 25% negative weightage to player who received yellow and red cards. Based upon this criteria we can select players that we are interested in.

7. Midfielder evaluating criteria



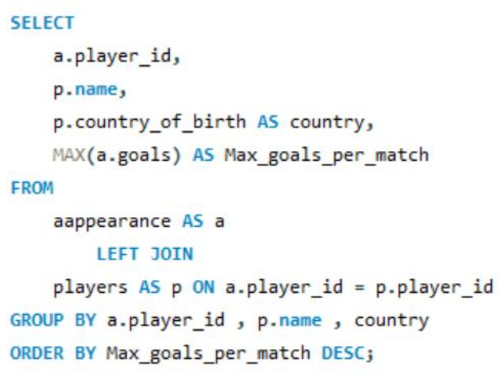
```
SELECT
    a.player_id,
    a.player_name,
    p.position,
    DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') AS Age,
    SUM(a.yellow_cards) AS Total_yellow_cards,
    SUM(a.red_cards) AS Total_red_cards,
    SUM(a.goals) AS total_goals,
    COUNT(a.player_id) AS total_appearances,
    (0.7 * SUM(a.goals) - 0.10 * SUM(a.yellow_cards) - 0.20 * SUM(a.red_cards)) / COUNT(a.player_id)
            AS evaluating_criteria
FROM
    aappearance AS a
        LEFT JOIN
    players AS p ON a.player_id = p.player_id
GROUP BY a.player_id , a.player_name , p.position , Age
HAVING evaluating_criteria IS NOT NULL
    AND p.position = 'Midfield'
    AND total_appearances > 85
    AND Age < 36
ORDER BY evaluating_criteria DESC , total_goals DESC
LIMIT 20;
```



player id	player name	position	Age	Total yellow cards	Total red cards	total goals	total appearances	evaluating criteria
player_id	player_name	position	Age	Total_yellow_cards	Total_red_cards	total_goals	total_appearances	evaluating_criteria
66515	Alex Teixeira	Midfield	33	16	0	52	90	0.386667
35207	Marco Reus	Midfield	34	26	1	119	288	0.279514
129554	Steven Berghuis	Midfield	32	49	1	121	317	0.251104
258626	Talisca	Midfield	29	22	0	54	145	0.245517
88968	Nasser El Khayati	Midfield	34	4	0	35	99	0.243434

Final Recommendation: Here, we have created a criterion for player selection for the “Midfield” position. We’ve given 70% weightage to goals scored. Similarly, we have given 10% and 20% negative weightage to player who received yellow and red cards. Based upon this criterion we can select players that we are interested in.

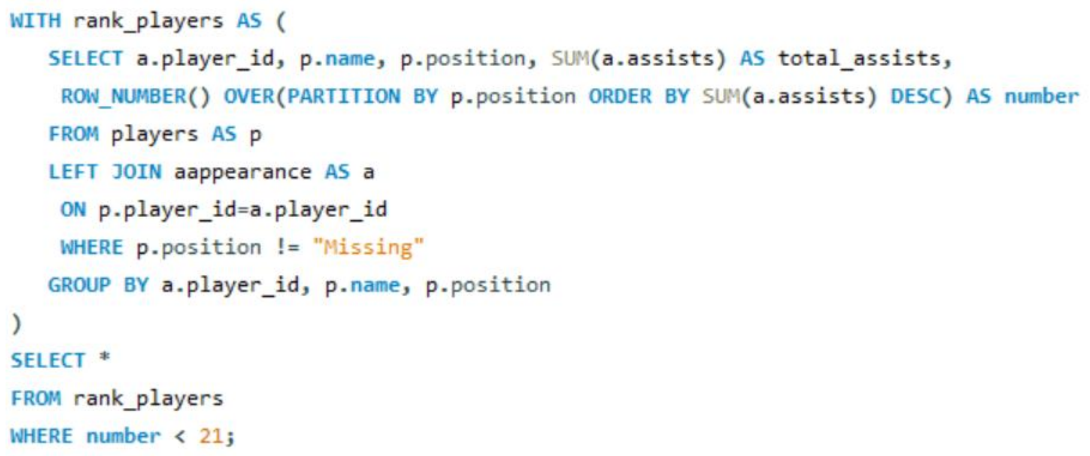
8. How many goals does each player scored and what is the maximum goals per match scored by the player?



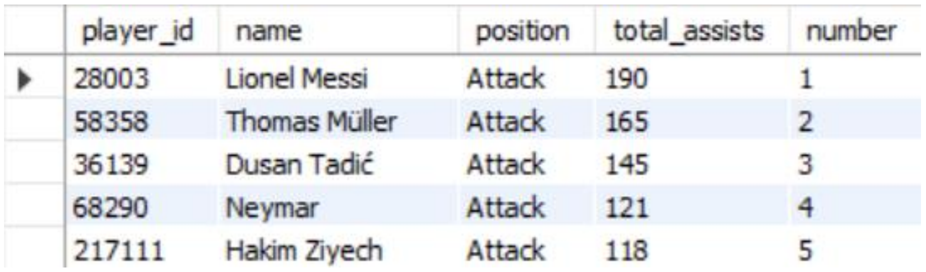
```
SELECT
    a.player_id,
    p.name,
    p.country_of_birth AS country,
    MAX(a.goals) AS Max_goals_per_match
FROM
    aappearance AS a
        LEFT JOIN
    players AS p ON a.player_id = p.player_id
GROUP BY a.player_id , p.name , country
ORDER BY Max_goals_per_match DESC;
```

Final Recommendation: This gives us an idea about maximum goals scored per match by a player. This tells us about the potential a player has.

9. . Successful passes by players based on position



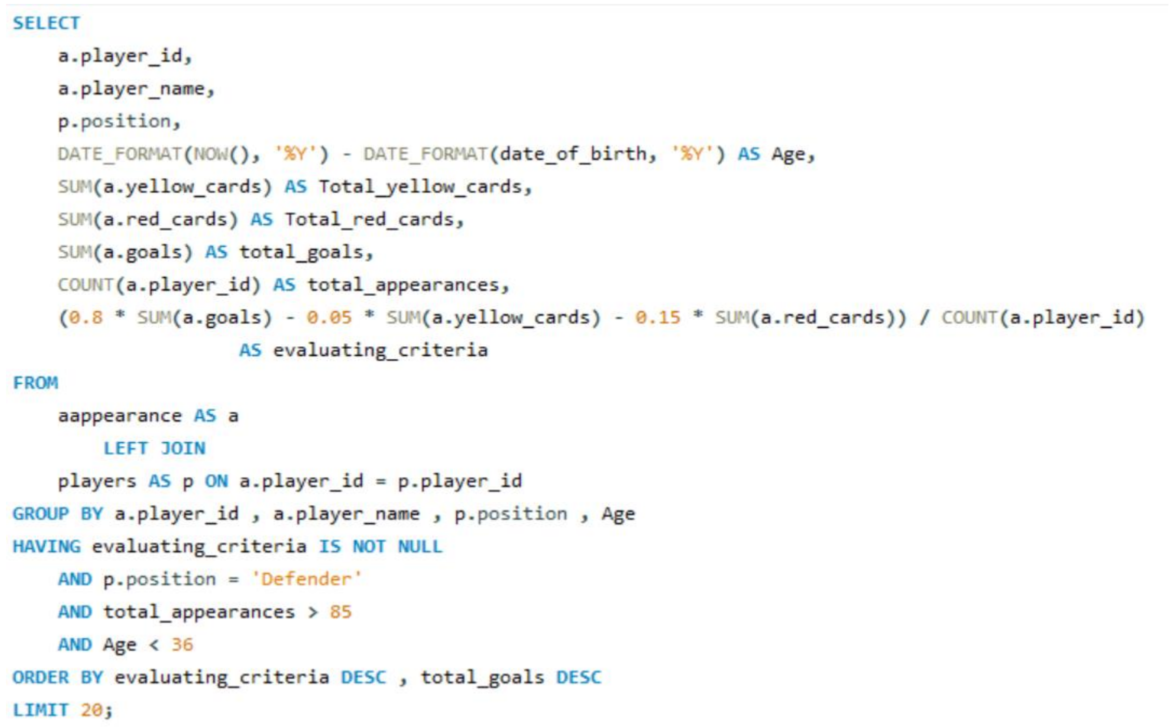
```
WITH rank_players AS (
   SELECT a.player_id, p.name, p.position, SUM(a.assists) AS total_assists,
    ROW_NUMBER() OVER(PARTITION BY p.position ORDER BY SUM(a.assists) DESC) AS number
   FROM players AS p
   LEFT JOIN aappearance AS a
    ON p.player_id=a.player_id
    WHERE p.position != "Missing"
   GROUP BY a.player_id, p.name, p.position
)
SELECT *
FROM rank_players
WHERE number < 21;
```



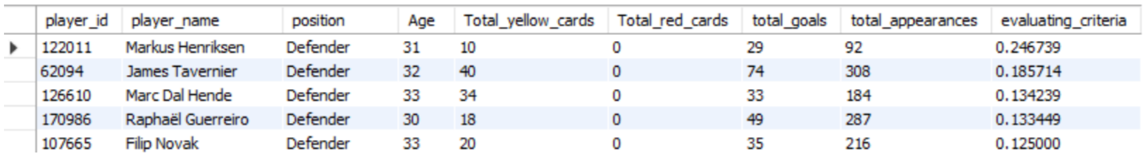
	player_id	name	position	total_assists	number
▶	28003	Lionel Messi	Attack	190	1
	58358	Thomas Müller	Attack	165	2
	36139	Dusan Tadić	Attack	145	3
	68290	Neymar	Attack	121	4
	217111	Hakim Ziyech	Attack	118	5

Final Recommendation: This shows us players with maximum number of assists. Assists are passes that converted to a successful goal.

10. Defender evaluating criteria



```
SELECT
    a.player_id,
    a.player_name,
    p.position,
    DATE_FORMAT(NOW(), '%Y') - DATE_FORMAT(date_of_birth, '%Y') AS Age,
    SUM(a.yellow_cards) AS Total_yellow_cards,
    SUM(a.red_cards) AS Total_red_cards,
    SUM(a.goals) AS total_goals,
    COUNT(a.player_id) AS total_appearances,
    (0.8 * SUM(a.goals) - 0.05 * SUM(a.yellow_cards) - 0.15 * SUM(a.red_cards)) / COUNT(a.player_id)
                AS evaluating_criteria
FROM
    aappearance AS a
        LEFT JOIN
    players AS p ON a.player_id = p.player_id
GROUP BY a.player_id , a.player_name , p.position , Age
HAVING evaluating_criteria IS NOT NULL
    AND p.position = 'Defender'
    AND total_appearances > 85
    AND Age < 36
ORDER BY evaluating_criteria DESC , total_goals DESC
LIMIT 20;
```

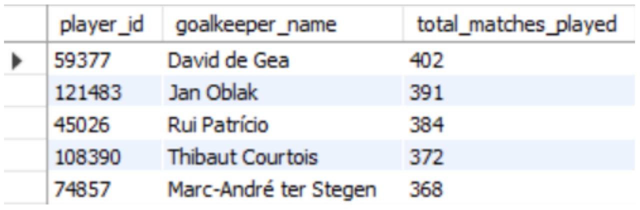


	player_id	player_name	position	Age	Total_yellow_cards	Total_red_cards	total_goals	total_appearances	evaluating_criteria
▸	122011	Markus Henriksen	Defender	31	10	0	29	92	0.246739
	62094	James Tavernier	Defender	32	40	0	74	308	0.185714
	126610	Marc Dal Hende	Defender	33	34	0	33	184	0.134239
	170986	Raphaël Guerreiro	Defender	30	18	0	49	287	0.133449
	107665	Filip Novak	Defender	33	20	0	35	216	0.125000

Final Recommendation: Here, we have created a criterion for player selection for the "Defender" position. We've given 80% weightage to goals scored. Similarly, we have given 5% and 15% negative weightage to player who received yellow and red cards. Based upon this criterion we can select players that we are interested in.

11. Goalkeeper Evaluating Criteria

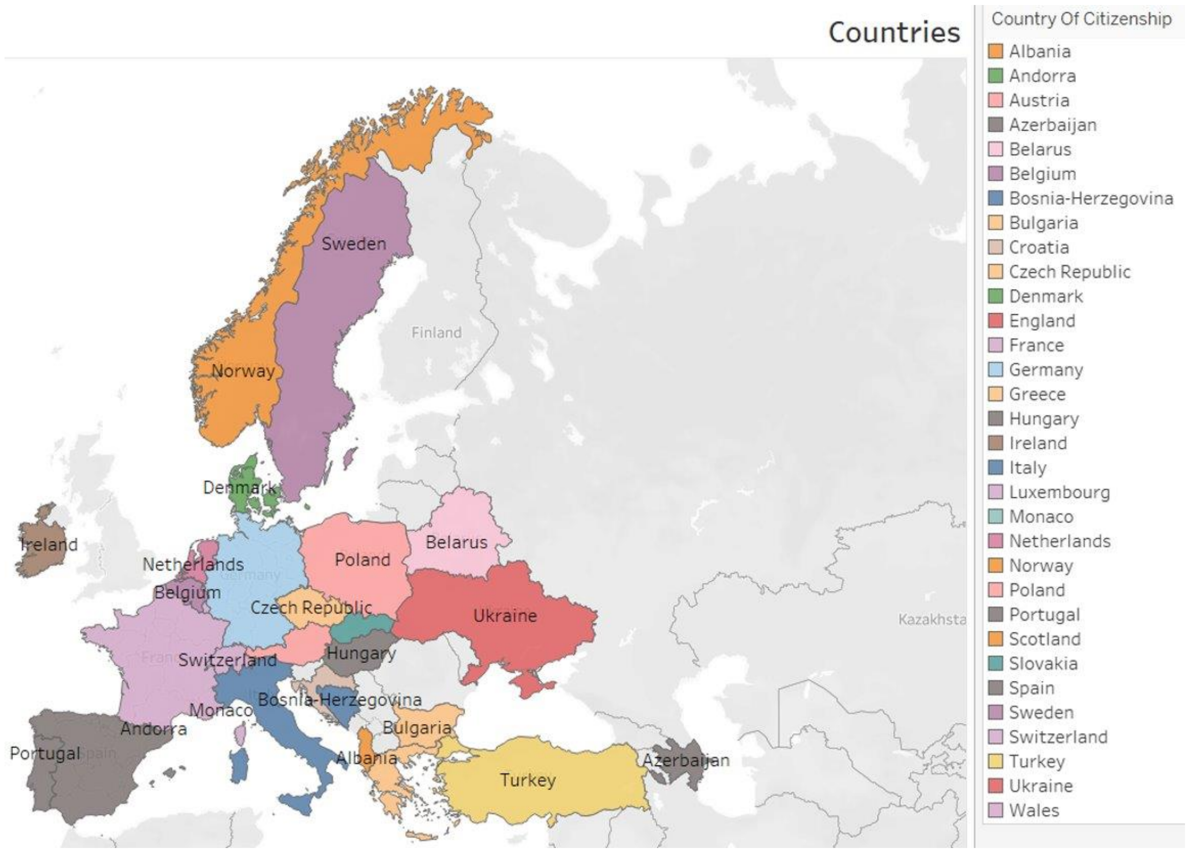
```
SELECT
    p.player_id,
    p.name AS goalkeeper_name,
    COUNT(a.player_id) AS total_matches_played
FROM
    players AS p
        LEFT JOIN
    aappearance AS a ON p.player_id = a.player_id
WHERE
    p.position = 'goalkeeper'
GROUP BY p.player_id , goalkeeper_name
ORDER BY total_matches_played DESC
LIMIT 20;
```

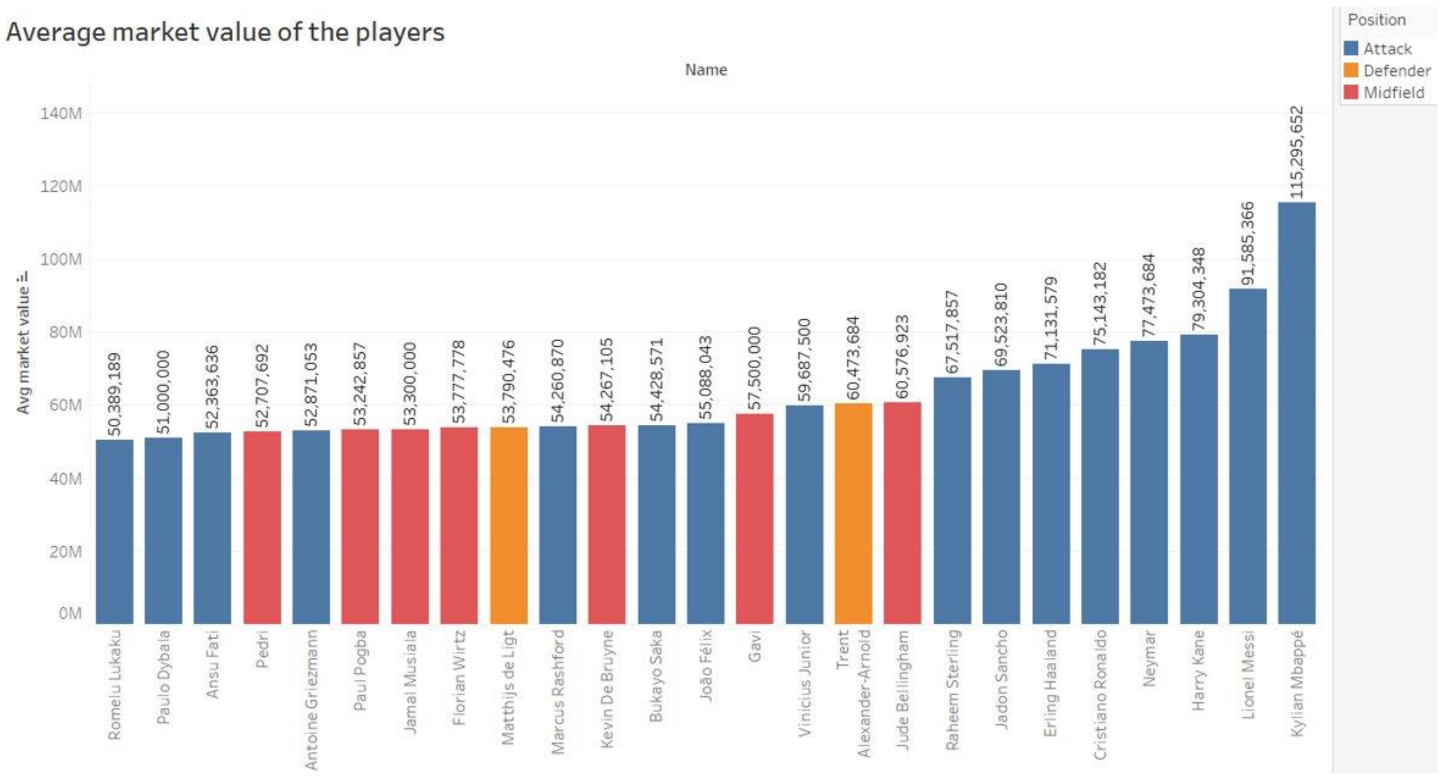


	player_id	goalkeeper_name	total_matches_played
▸	59377	David de Gea	402
	121483	Jan Oblak	391
	45026	Rui Patrício	384
	108390	Thibaut Courtois	372
	74857	Marc-André ter Stegen	368

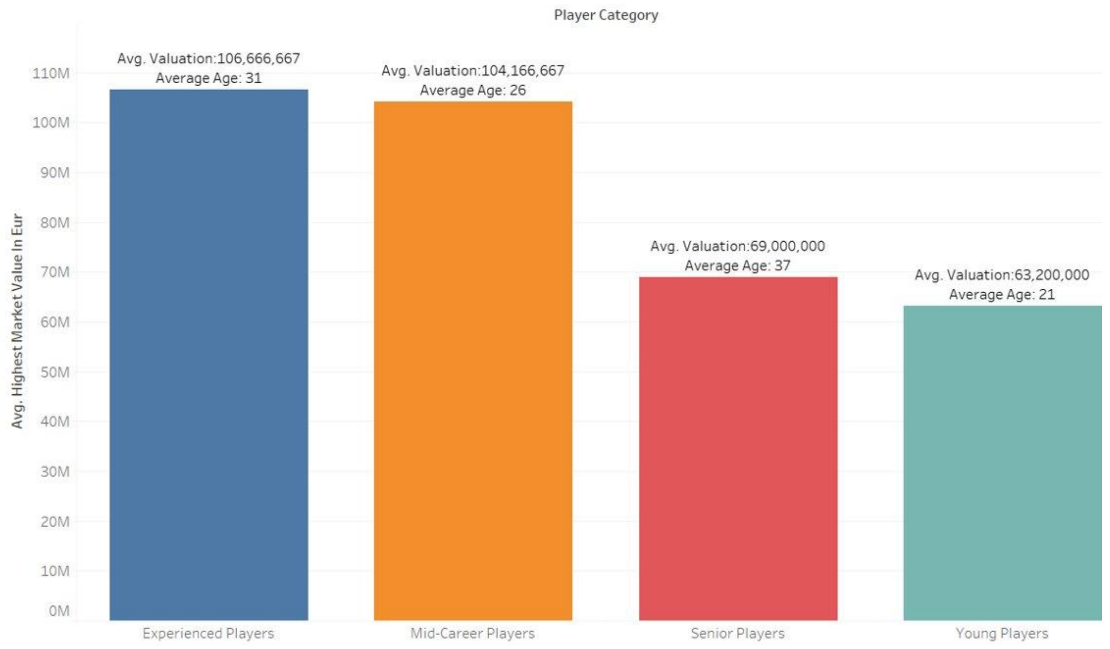
Final Recommendation: The goalkeeper is evaluated on the basis of number of matches played. Here data gives us total matches played by the player.

Visualization Charts:

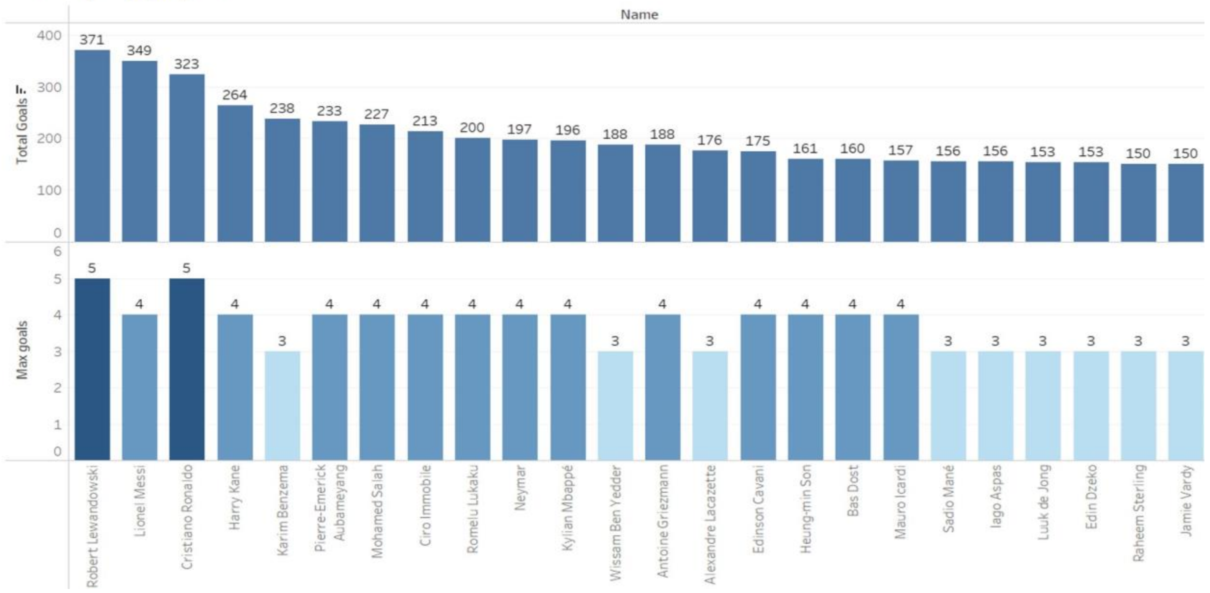




Player Category based on Valuation



Goals by top players



Conclusion:

To sum it up, our analysis of football data has given us great insights into improving our European football club's performance. By looking at player stats, team dynamics, and market trends, we've set the stage for a smart, data-driven approach to building our team. We used analytics tools like MYSQL, Tableau and historical data to make better decisions, ensuring we get the right players within our budget.

Our goal of creating a diverse player pool, participating in strategic auctions, and building a successful team has been addressed through this dataset. We've improved player scouting and auction strategies, making sure we spend wisely on players that fit our team's goals.

We also understand the importance of keeping our fans engaged. By building a competitive team, we aim to increase revenue and expand our club's influence locally and internationally. Looking ahead, using football data will continue to guide our decisions.