**TRANSFER MARKET ANALYSIS 2022-23**

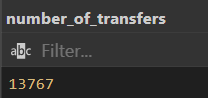
1. **TOTAL PLAYER TRANSFER COUNT:**

SELECT count(\*) as number\_of\_transfers FROM your\_table;

**EXPLANATION:**

* We use the SQL **"SELECT"** statement to retrieve specific data from a database table.
* The goal is to calculate the total number of player transfers during the project's defined period.
* The **"COUNT(\*)"** function is applied to count all rows or records within the "your\_table" database table. The result is assigned an alias **"number\_of\_transfers"** for clarity.
* It is a critical component of our analysis, offering insights into the extent of player transfers and shaping our understanding of market dynamics.

**OUTPUT:**

****

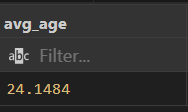
1. **AVERAGE AGE COMPUTATION FOR TRANSFERRED PLAYERS:**

SELECT AVG(TRANSFER\_FEE) AS AVG\_FEE FROM your\_table;

**EXPLANATION:**

* we utilize the **"SELECT"** statement to obtain specific data from the **"your\_table"** database table.
* The objective is to calculate the average transfer fee paid for player transfers during the designated period.
* To achieve this, we apply the **"AVG(TRANSFER\_FEE)"** function to compute the mean (average) value of the **"TRANSFER\_FEE"** column. The result is given the alias **"AVG\_FEE"** for clarity and ease of reference.
* This query contributes to our analysis by providing a key financial metric: the average transfer fee. It helps us understand the typical financial commitment clubs make when acquiring new talent in the football transfer market.

**OUTPUT:**



1. **CALCULATION OF AVERAGE TRANSFER TRANSACTION VALUE:**

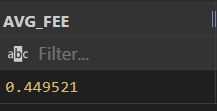
SELECT AVG(age) as avg\_age FROM your\_table

WHERE TRANSFER\_FEE > 0 ;

**EXPLANATION:**

* This SQL query focuses on calculating the average age of players who have been transferred and had a positive transfer fee associated with their transfers.
* The **"SELECT"** statement is used to retrieve specific data from the "your\_table" database table.
* We apply the **"AVG(age)"** function to calculate the mean (average) value of the "age" column for players who meet the specified condition.
* The **"WHERE TRANSFER\_FEE > 0"** clause filters the data, ensuring that only players with a positive transfer fee are included in the calculation.
* The result is given the alias **"avg\_age"** for clarity and ease of reference.
* This query is valuable for our analysis as it provides insights into the average age of transferred players who had financial transactions associated with their transfers. It helps us understand the age distribution of players involved in paid transfers, which can be essential for recruitment and talent strategy in football.

**OUTPUT:**



1. **TOTAL MARKET VALUATION OF TRANSFERRED PLAYERS:**

SELECT SUM(MARKET\_VALUE) AS TOTAL\_MARKET\_VALUE

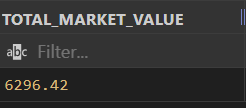
FROM your\_table

WHERE TRANSFER\_FEE > 0;

**EXPLANATION:**

* This SQL query is designed to compute the cumulative market value of players who have been transferred and had a positive transfer fee associated with their transfers.
* We use the **"SELECT"** statement to extract specific information from the **"your\_table"** database table.
* The **"SUM(MARKET\_VALUE)"** function is applied to sum up the values in the **"MARKET\_VALUE"** column for players who meet the condition specified in the **"WHERE"** clause.
* The **"WHERE TRANSFER\_FEE > 0"** condition ensures that only players involved in transfers with a positive transfer fee are considered for the calculation.
* The result is assigned the alias **"TOTAL\_MARKET\_VALUE"** for clarity and easy reference.
* This query is significant for our analysis as it provides a comprehensive view of the combined market value of players who were part of transfers with financial transactions. It allows us to assess the financial magnitude of player transfers during the specified period, contributing to a better understanding of the football transfer market's financial dynamics.

**OUTPUT:**



1. **IDENTIFICATION OF HIGHEST SINGLE-PLAYER TRANSFER FEE:**

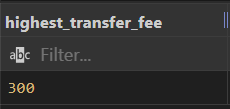
SELECT MAX(TRANSFER\_FEE) AS highest\_transfer\_fee

FROM your\_table;

**EXPLANATION:**

* This SQL query is constructed to find the maximum (highest) transfer fee ever paid for an individual player transfer.
* We employ the **"SELECT"** statement to retrieve specific information from the **"your\_table"** database table.
* The **"MAX(TRANSFER\_FEE)"** function is applied to identify the highest value in the **"TRANSFER\_FEE"** column, representing the highest transfer fee recorded.
* The **"AS highest\_transfer\_fee"** clause assigns an alias to the result for clear reference.
* This query is essential for our analysis as it reveals the highest financial transaction within the realm of sports, shedding light on an extraordinary figure in the football transfer market. It helps us understand the context and significance of this historic deal, which can have far-reaching implications and insights into the transfer market's dynamics.

**OUTPUT:**



1. **QUANTIFICATION OF FREE TRANSFERS:**

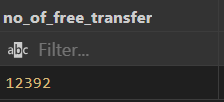
SELECT COUNT(\*) AS no\_of\_free\_transfer FROM your\_table

WHERE TRANSFER\_FEE = 0 ;

**EXPLANATION:**

* This SQL query is designed to count and quantify free transfers, which are player transitions that occur without any monetary exchange.
* We utilize the "SELECT" statement to retrieve specific information from the "your\_table" database table.
* The "COUNT(\*)" function is applied to tally the number of rows that meet the specified condition, which is where the "TRANSFER\_FEE" is equal to 0 (indicating free transfers).
* The result is assigned an alias using "AS no\_of\_free\_transfer" for easy reference and clarity.
* This query is vital for our analysis as it allows us to document and analyse instances where players move between clubs without any financial transactions. Understanding the prevalence and reasons behind free transfers is essential for gaining a comprehensive view of the football transfer market. It helps us explore factors such as contractual terminations, loan deals, or other unique circumstances contributing to these types of transfers.

**OUTPUT:**



1. **Transfer Volume Analysis by Month:**

SELECT

    day,

    COUNT(\*) AS TransferCount

FROM your\_table

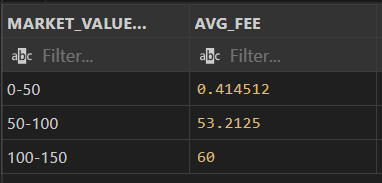
GROUP BY day

ORDER BY day;

**EXPLANATION:**

* In this SQL query, we aim to analyse the volume of player transfers on a month-by-month basis.
* We start with the **"SELECT"** statement to retrieve specific information from the **"your\_table"** database table.
* Two columns are selected for retrieval: "day" and the count of transfers denoted as **"Transfer Count."**
* The **"day"** column presumably contains date information, and we will extract and analyse data on a daily basis to later group it by month.
* The **"COUNT(\*)"** function is applied to count the number of rows within each grouping of **"day."** This effectively quantifies the number of transfers that occurred on each day.
* Next, we use the **"GROUP BY"** clause to group the data by the **"day"** column. This will aggregate the daily transfer counts into months.
* Finally, we add an **"ORDER BY"** clause to sort the results in ascending order by the **"day"** column.

**OUTPUT:**



**2. Relationship Between Average Transfer Fee, Age, and Position:**

SELECT AVG(TRANSFER\_FEE) AS AvgTransferFee, AVG(age) AS avg\_age, Positions

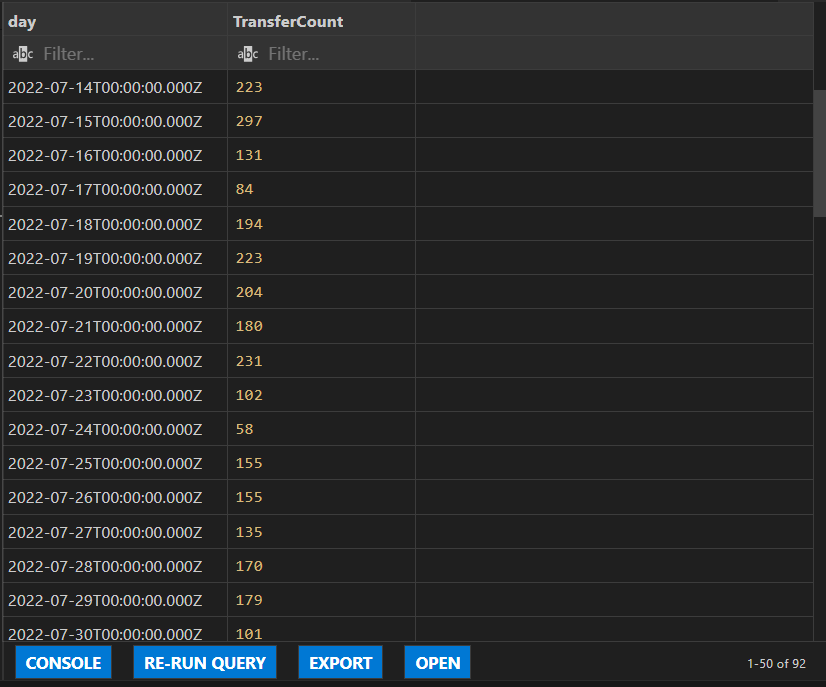
FROM your\_table

WHERE TRANSFER\_FEE > 0

GROUP BY Positions;

**EXPLANATION:**

* In this SQL query, we aim to calculate both the average transfer fee and the average age of players based on their positions.
* We start with the **"SELECT"** statement to retrieve specific information from the **"your\_table"** database table.
* Three columns are selected for retrieval: **"Positions,"** the average transfer fee denoted as **"AvgTransferFee,"** and the average age denoted as **"avg\_age."**
* The **"WHERE"** clause is used to filter the data. We want to consider only those records where the **"TRANSFER\_FEE"** is greater than 0, meaning we exclude records with zero transfer fees, as these might represent free transfers or loans.
* The **"GROUP BY"** clause is applied to group the data by the **"Positions"** column. This groups players by their respective positions, such as **"Goalkeeper," "Striker," "Midfield,"** etc.

**OUTPUT**

**3. Market Value vs. Transfer Fee Analysis:**

SELECT

    CASE

        WHEN MARKET\_VALUE >= 0 AND MARKET\_VALUE <= 50 THEN '0-50'

        WHEN MARKET\_VALUE > 50 AND MARKET\_VALUE <= 100 THEN '50-100'

        WHEN MARKET\_VALUE > 100 AND MARKET\_VALUE <= 150 THEN '100-150'

        ELSE 'Other'

    END AS MARKET\_VALUE\_RANGE,

    AVG(TRANSFER\_FEE) AS AVG\_FEE

FROM

    your\_table

WHERE

    TRANSFER\_FEE IS NOT NULL

    AND MARKET\_VALUE IS NOT NULL

GROUP BY

    CASE

        WHEN MARKET\_VALUE >= 0 AND MARKET\_VALUE <= 50 THEN '0-50'

        WHEN MARKET\_VALUE > 50 AND MARKET\_VALUE <= 100 THEN '50-100'

        WHEN MARKET\_VALUE > 100 AND MARKET\_VALUE <= 150 THEN '100-150'

        ELSE 'Other'

    END

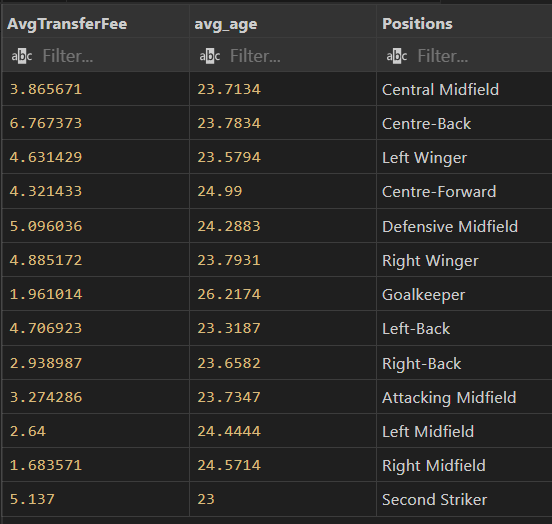
ORDER BY

    MIN(MARKET\_VALUE);

**EXPLANATION:**

* This SQL query is designed to calculate the average transfer fee for players categorized into different market value ranges.
* It starts with a **"SELECT"** statement to retrieve specific information from the **"your\_table"** database table.
* Two columns are selected for retrieval:
  + **"MARKET\_VALUE\_RANGE":** This is a calculated field that categorizes players into market value ranges. It uses the "CASE" statement to assign players to one of four ranges**: '0-50,' '50-100,' '100-150,' or 'Other,'** based on their market value.
  + **"AVG\_FEE":** This is the calculated average transfer fee for players within each market value range.
* In the **"WHERE"** clause, we apply filters to include only records where both **"TRANSFER\_FEE"** and **"MARKET\_VALUE"** are not null, ensuring that we work with valid data.
* The "GROUP BY" clause is used to group the data by the **"MARKET\_VALUE\_RANGE"** column. This groups players into the specified market value ranges.
* For each group (market value range), the query calculates the average transfer fee.

**OUTPUT:**



1. **Total Transfers and Transfer Fees by Club:**

SELECT to\_league, to\_country, count(\*) as no\_transfer, SUM(TRANSFER\_FEE) as total\_transfer\_fee

FROM your\_table

GROUP BY to\_league, to\_country

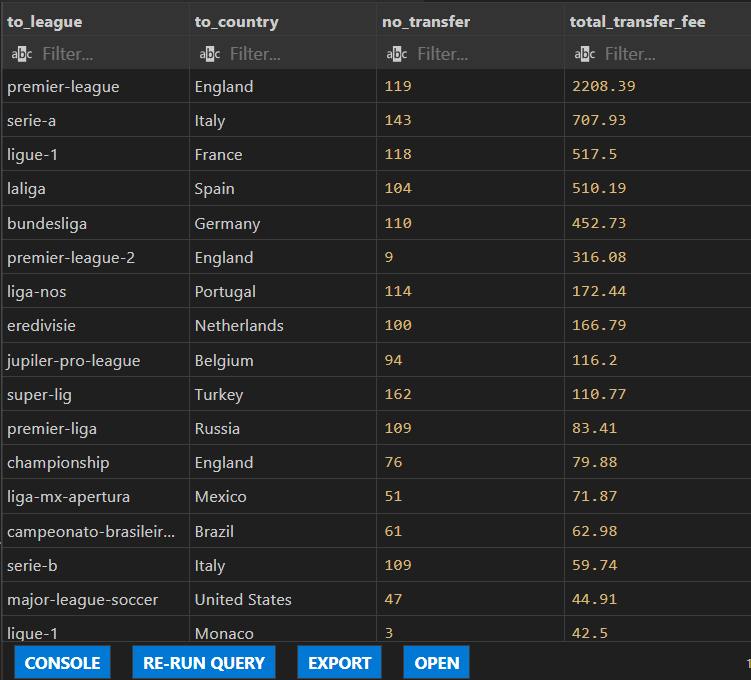
ORDER BY total\_transfer\_fee DESC

LIMIT 20;

**EXPLANATION:**

* This SQL query is designed to identify and rank the top 20 destination leagues and countries based on the number of player transfers and the total transfer fees incurred.
* It starts with a **"SELECT"** statement to retrieve specific information from the **"your\_table"** database table.
* Four columns are selected for retrieval:
  + **"to\_league":** This column represents the destination league where the player is transferred.
  + **"to\_country":** This column represents the destination country where the player is transferred.
  + **"no\_transfer":** This is the count of player transfers to each destination league and country combination. It indicates how many transfers occurred in each location.
  + **"total\_transfer\_fee":** This is the sum of transfer fees for all player transfers to each destination league and country combination. It represents the total financial commitment made to acquire players for that location.
* The **"GROUP BY"** clause is used to group the data by both **"to\_league"** and **"to\_country."** This means that the query will count and sum transfers for each unique combination of league and country.
* The results are then ordered by **"total\_transfer\_fee"** in descending order, ensuring that the top destinations with the highest total transfer fees are listed first.

**OUTPUT:**



**KEY INSIGHTS**

* Summer Transfer Surge: In the 2022-23 summer transfer window, the football world witnessed a significant surge in player transfers. July recorded the highest number of transfers at 9,623, marking an astounding 824.40% increase compared to September, which had the lowest number of transfers at 1,041.
* Monthly Activity: July dominated with 9,623 transfers, followed by August at 3,103 and September at 1,041, highlighting the stark contrast in activity.
* July's Prominence: July accounted for a substantial 69.90% of the total number of transfers during the summer window, underscoring its pivotal role.
* Monthly Breakdown: These figures provide a clear hierarchy in transfer activity, offering a comprehensive view of the dynamics during this period.
* Position Insights: Noteworthy position-related findings include Goalkeeper having the highest average age at 26.22, while Centre-Back recorded the highest average transfer fee at 6.77.
* League Dynamics: The analysis showed that the divergence between the sum of total transfer fees and the sum of the number of transfers was most pronounced in the Premier League, with total transfer fees being 774 units higher than the number of transfers. This highlights the unique financial dynamics within this league.