

DOCUMENTAÇÃO MEDIDAX DAX (POWER BI)

Nome da Medida: Aparicoes_Jogador

Descrição: Esta medida calcula o número de aparições de um jogador na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
CALCULATE(
    COUNTROWS('public fat_performance'),
    ALLEXCEPT('public fat_performance', 'public
fat_performance'[jogador_id])
)
```

Função CALCULATE: A função CALCULATE é utilizada para modificar o contexto de avaliação das expressões dentro dela, aplicando ou removendo filtros.

Função COUNTROWS: COUNTROWS é uma função que conta o número de linhas em uma tabela ou uma tabela resultante de uma expressão.

Função ALLEXCEPT: ALLEXCEPT remove todos os filtros exceto aqueles especificados, mantendo apenas o contexto de filtro na coluna 'jogador_id' da tabela 'public fat_performance'.

Parâmetros:

- **Tabela:** 'public fat_performance'
- **Coluna:** 'jogador_id'

Uso: Essa medida pode ser usada para determinar quantas vezes um jogador específico aparece na tabela de desempenho ('public fat_performance'). É útil para análises que requerem contagem de ocorrências específicas de um jogador ao longo do tempo ou em diferentes contextos de filtragem.

Nome da Medida: eFG%

Descrição: Esta medida calcula a porcentagem de efetividade de arremessos de um jogador na tabela 'public fat_performance', considerando arremessos de dois pontos (FG) e arremessos de três pontos (3P).

Expressão DAX:

```
dax
Copiar código
eFG% =
DIVIDE(
```

```

SUM('public fat_performance'[fg]) + 0.5 * SUM('public
fat_performance'[three_p]),
SUM('public fat_performance'[fga]),
0
)

```

Função DIVIDE: DIVIDE é uma função que divide dois números. Neste caso, é utilizada para calcular a porcentagem de efetividade de arremessos.

Parâmetros:

- **Numerador:** SUM('public fat_performance'[fg]) + 0.5 * SUM('public fat_performance'[three_p])
 - Soma dos arremessos de dois pontos mais a metade da soma dos arremessos de três pontos.
- **Denominador:** SUM('public fat_performance'[fga])
 - Soma dos arremessos de dois pontos tentados.

Uso: Esta medida é útil para avaliar a efetividade de arremessos de um jogador, levando em consideração a contribuição extra dos arremessos de três pontos, que são ponderados por um fator de 0.5 na fórmula.

Nome da Medida: eFG%_Com_Filtro

Descrição: Esta medida calcula a porcentagem de efetividade de arremessos de um jogador na tabela 'public fat_performance', considerando arremessos de dois pontos (FG) e arremessos de três pontos (3P), com um filtro aplicado para incluir apenas jogadores que têm pelo menos 50 aparições na tabela.

Expressão DAX:

```

dax
Copiar código
eFG%_Com_Filtro =
IF(
    [Aparicoes_Jogador] >= 50,
    DIVIDE(
        SUM('public fat_performance'[fg]) + 0.5 * SUM('public
fat_performance'[three_p]),
        SUM('public fat_performance'[fga]),
        0
    ),
    BLANK()
)

```

Função IF: IF é uma função condicional que retorna um valor se uma condição for verdadeira e outro valor se for falsa. Neste caso, determina se a contagem de aparições do jogador na tabela é maior ou igual a 50.

Função DIVIDE: DIVIDE é utilizada para calcular a porcentagem de efetividade de arremessos, seguindo a mesma lógica da medida anterior.

Parâmetros:

- **Condição:** [Aparicoes_Jogador] >= 50
 - Verifica se o número de aparições do jogador é maior ou igual a 50.
- **Valor se Verdadeiro:**

```
dax
Copiar código
DIVIDE (
    SUM('public fat_performance'[fg]) + 0.5 * SUM('public
fat_performance'[three_p]),
    SUM('public fat_performance'[fga]),
    0
)
```

- Calcula a porcentagem de efetividade de arremessos se a condição for verdadeira.
- **Valor se Falso:** BLANK()
 - Retorna um valor em branco se a condição não for satisfeita.

Uso: Esta medida é útil para avaliar a efetividade de arremessos de jogadores que tenham um número mínimo de aparições na tabela 'public fat_performance', fornecendo uma maneira de filtrar os dados para análises mais relevantes.

Nome da Medida: FG%

Descrição: Esta medida calcula a porcentagem de acertos de arremessos de dois pontos (FG) de um jogador na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
FG% = DIVIDE(SUM('public fat_performance'[fg]), SUM('public
fat_performance'[fga]), 0)
```

Função DIVIDE: DIVIDE é uma função que divide dois números. Neste caso, é utilizada para calcular a porcentagem de acertos de arremessos de dois pontos.

Parâmetros:

- **Numerador:** SUM('public fat_performance'[fg])
 - Soma dos arremessos de dois pontos convertidos em sucesso (FG).
- **Denominador:** SUM('public fat_performance'[fga])
 - Soma dos arremessos de dois pontos tentados.

Uso: Esta medida é útil para avaliar a precisão de arremessos de dois pontos (FG) de um jogador ao longo de um período de tempo, com base nos dados registrados na tabela 'public fat_performance'.

Aqui está a documentação para a medida FG % Com Filtro utilizando a expressão DAX fornecida:

Documentação do DAX

Nome da Medida: FG % Com Filtro

Descrição: Esta medida calcula a porcentagem de acertos de arremessos de três pontos (3P) de um jogador na tabela 'public fat_performance', aplicando um filtro para incluir apenas jogadores que tenham pelo menos 50 tentativas de arremessos de três pontos.

Expressão DAX:

```
dax
Copiar código
FG % Com Filtro =
IF(
    SUM('public fat_performance'[three_pa]) >= 50,
    DIVIDE(
        SUM('public fat_performance'[three_p]),
        SUM('public fat_performance'[three_pa]),
        0
    ),
    BLANK()
)
```

Função IF: IF é uma função condicional que retorna um valor se uma condição for verdadeira e outro valor se for falsa. Neste caso, verifica se o número de tentativas de arremessos de três pontos (3P) é maior ou igual a 50.

Função DIVIDE: DIVIDE é utilizada para calcular a porcentagem de acertos de arremessos de três pontos, seguindo a mesma lógica da medida anterior.

Parâmetros:

- **Condição:** SUM('public fat_performance'[three_pa]) >= 50
 - Verifica se o número de tentativas de arremessos de três pontos é maior ou igual a 50.
- **Valor se Verdadeiro:**

```
dax
Copiar código
DIVIDE(
    SUM('public fat_performance'[three_p]),
    SUM('public fat_performance'[three_pa]),
    0
)
```

- Calcula a porcentagem de acertos de arremessos de três pontos se a condição for verdadeira.
- **Valor se Falso:** BLANK()

- Retorna um valor em branco se a condição não for satisfeita.

Uso: Esta medida é útil para avaliar a precisão de arremessos de três pontos de jogadores que tenham um número mínimo de tentativas na tabela 'public fat_performance', fornecendo uma maneira de filtrar os dados para análises mais relevantes.

Nome da Medida: Jogador_Com_Mais_3PT

Descrição: Esta medida identifica o jogador que possui o maior número de arremessos de três pontos (3PT) na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Jogador_Com_Mais_3PT =
VAR TopPlayer =
    TOPN(1, 'public fat_performance', 'public
fat_performance'[three_p], DESC)
RETURN
    MAXX(TopPlayer, RELATED('public dim_jogador'[jogador]))
```

Função VAR: VAR é usada para definir uma variável local (TopPlayer neste caso), que armazena o jogador com o maior número de arremessos de três pontos.

Função TOPN: TOPN é usada para retornar o número especificado de linhas com base em uma classificação. Neste caso, retorna o jogador com o maior número de arremessos de três pontos ('public fat_performance'[three_p]) em ordem decendente (DESC).

Função MAXX: MAXX retorna o valor máximo de uma expressão avaliada para cada linha de uma tabela. Neste caso, retorna o nome do jogador relacionado ao jogador com o maior número de arremessos de três pontos.

Função RELATED: RELATED é usada para recuperar um valor de coluna relacionado em outra tabela. Aqui, recupera o nome do jogador ('public dim_jogador'[jogador]) relacionado ao jogador com o maior número de arremessos de três pontos.

Uso: Esta medida é útil para identificar rapidamente qual jogador teve o melhor desempenho em termos de arremessos de três pontos na tabela 'public fat_performance'.

Nome da Medida: Jogador_Com_Mais_Assists

Descrição: Esta medida identifica o jogador que possui o maior número de assistências (assists) na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Jogador_Com_Mais_Assists =
VAR TopPlayer =
    TOPN(1, 'public fat_performance', 'public fat_performance'[ast],
DESC)
RETURN
    MAXX(TopPlayer, RELATED('public dim_jogador'[jogador]))
```

Função VAR: VAR é usada para definir uma variável local (TopPlayer neste caso), que armazena o jogador com o maior número de assistências.

Função TOPN: TOPN é usada para retornar o número especificado de linhas com base em uma classificação. Neste caso, retorna o jogador com o maior número de assistências ('public fat_performance'[ast]) em ordem decendente (DESC).

Função MAXX: MAXX retorna o valor máximo de uma expressão avaliada para cada linha de uma tabela. Neste caso, retorna o nome do jogador relacionado ao jogador com o maior número de assistências.

Função RELATED: RELATED é usada para recuperar um valor de coluna relacionado em outra tabela. Aqui, recupera o nome do jogador ('public dim_jogador'[jogador]) relacionado ao jogador com o maior número de assistências.

Uso: Esta medida é útil para identificar rapidamente qual jogador teve o melhor desempenho em termos de assistências na tabela 'public fat_performance'.

Nome da Medida: Jogador_Com_Mais_Blocks

Descrição: Esta medida identifica o jogador que possui o maior número de bloqueios (blocks) na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Jogador_Com_Mais_Blocks =
VAR TopPlayer =
    TOPN(1, 'public fat_performance', 'public fat_performance'[blk],
DESC)
RETURN
    MAXX(TopPlayer, RELATED('public dim_jogador'[jogador]))
```

Função VAR: VAR é usada para definir uma variável local (TopPlayer neste caso), que armazena o jogador com o maior número de bloqueios.

Função TOPN: TOPN é usada para retornar o número especificado de linhas com base em uma classificação. Neste caso, retorna o jogador com o maior número de bloqueios ('public fat_performance'[blk]) em ordem decendente (DESC).

Função MAXX: MAXX retorna o valor máximo de uma expressão avaliada para cada linha de uma tabela. Neste caso, retorna o nome do jogador relacionado ao jogador com o maior número de bloqueios.

Função RELATED: RELATED é usada para recuperar um valor de coluna relacionado em outra tabela. Aqui, recupera o nome do jogador ('public dim_jogador'[jogador]) relacionado ao jogador com o maior número de bloqueios.

Uso: Esta medida é útil para identificar rapidamente qual jogador teve o melhor desempenho em termos de bloqueios na tabela 'public fat_performance'.

Nome da Medida: Jogador_Com_Mais_Pontos

Descrição: Esta medida identifica o jogador que marcou o maior número de pontos (pts) na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Jogador_Com_Mais_Pontos =
VAR TopPlayer =
    TOPN(1, 'public fat_performance', 'public fat_performance'[pts],
DESC)
RETURN
    MAXX(TopPlayer, RELATED('public dim_jogador'[jogador]))
```

Função VAR: VAR é usada para definir uma variável local (TopPlayer neste caso), que armazena o jogador com o maior número de pontos.

Função TOPN: TOPN é usada para retornar o número especificado de linhas com base em uma classificação. Neste caso, retorna o jogador com o maior número de pontos ('public fat_performance'[pts]) em ordem descendente (DESC).

Função MAXX: MAXX retorna o valor máximo de uma expressão avaliada para cada linha de uma tabela. Neste caso, retorna o nome do jogador relacionado ao jogador com o maior número de pontos.

Função RELATED: RELATED é usada para recuperar um valor de coluna relacionado em outra tabela. Aqui, recupera o nome do jogador ('public dim_jogador'[jogador]) relacionado ao jogador com o maior número de pontos.

Uso: Esta medida é útil para identificar rapidamente qual jogador marcou mais pontos na tabela 'public fat_performance'.

Nome da Medida: Jogador_Com_Mais_REB

Descrição: Esta medida identifica o jogador que possui o maior número de rebotes (REB) na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Jogador_Com_Mais_REB =
VAR TopPlayer =
    TOPN(1, 'public fat_performance', 'public fat_performance'[trb],
    DESC)
RETURN
    MAXX(TopPlayer, RELATED('public dim_jogador'[jogador]))
```

Função VAR: VAR é usada para definir uma variável local (TopPlayer neste caso), que armazena o jogador com o maior número de rebotes.

Função TOPN: TOPN é usada para retornar o número especificado de linhas com base em uma classificação. Neste caso, retorna o jogador com o maior número de rebotes ('public fat_performance'[trb]) em ordem descendente (DESC).

Função MAXX: MAXX retorna o valor máximo de uma expressão avaliada para cada linha de uma tabela. Neste caso, retorna o nome do jogador relacionado ao jogador com o maior número de rebotes.

Função RELATED: RELATED é usada para recuperar um valor de coluna relacionado em outra tabela. Aqui, recupera o nome do jogador ('public dim_jogador'[jogador]) relacionado ao jogador com o maior número de rebotes.

Uso: Esta medida é útil para identificar rapidamente qual jogador teve o melhor desempenho em termos de rebotes na tabela 'public fat_performance'.

Nome da Medida: Jogador_Com_Mais_STL

Descrição: Esta medida identifica o jogador que possui o maior número de roubos de bola (STL) na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Jogador_Com_Mais_STL =
VAR TopPlayer =
    TOPN(1, 'public fat_performance', 'public fat_performance'[stl],
    DESC)
RETURN
    MAXX(TopPlayer, RELATED('public dim_jogador'[jogador]))
```

Função VAR: VAR é usada para definir uma variável local (TopPlayer neste caso), que armazena o jogador com o maior número de roubos de bola.

Função TOPN: TOPN é usada para retornar o número especificado de linhas com base em uma classificação. Neste caso, retorna o jogador com o maior número de roubos de bola ('public fat_performance'[stl]) em ordem descendente (DESC).

Função MAXX: MAXX retorna o valor máximo de uma expressão avaliada para cada linha de uma tabela. Neste caso, retorna o nome do jogador relacionado ao jogador com o maior número de roubos de bola.

Função RELATED: RELATED é usada para recuperar um valor de coluna relacionado em outra tabela. Aqui, recupera o nome do jogador ('public dim_jogador'[jogador]) relacionado ao jogador com o maior número de roubos de bola.

Uso: Esta medida é útil para identificar rapidamente qual jogador teve o melhor desempenho em termos de roubos de bola na tabela 'public fat_performance'.

Nome da Medida: Media_Assistencias_por_Jogo

Descrição: Esta medida calcula a média de assistências por jogo na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Media_Assistencias_por_Jogo =
DIVIDE(
    SUM('public fat_performance'[ast]),
    COUNTROWS(
        FILTER(
            'public fat_performance',
            'public fat_performance'[ast] <> BLANK()
        )
    ),
    0
)
```

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma das assistências (SUM('public fat_performance'[ast])) pelo número de jogos onde assistências são registradas.

Função SUM: SUM é usada para somar os valores de uma coluna ('public fat_performance'[ast] neste caso, que representa as assistências).

Função COUNTROWS: COUNTROWS é usada para contar o número de linhas em uma tabela ou expressão de tabela. Aqui, conta o número de linhas onde há registros de assistências.

Função FILTER: FILTER é usada para retornar um conjunto de linhas que satisfazem determinados critérios de filtro. Neste caso, filtra as linhas na tabela 'public

fat_performance' onde as assistências não são vazias ('public fat_performance'[ast] <> BLANK()).

Uso: Esta medida é útil para calcular a média de assistências por jogo, considerando apenas os jogos em que assistências foram registradas na tabela 'public fat_performance'.

Nome da Medida: Media_Blocks_por_Jogo

Descrição: Esta medida calcula a média de bloqueios (blocks) por jogo na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Media_Blocks_por_Jogo =
DIVIDE(
    SUM('public fat_performance'[blk]),
    COUNTROWS(
        FILTER(
            'public fat_performance',
            'public fat_performance'[blk] <> BLANK()
        )
    ),
    0
)
```

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma dos bloqueios (SUM('public fat_performance'[blk])) pelo número de jogos onde bloqueios são registrados.

Função SUM: SUM é usada para somar os valores de uma coluna ('public fat_performance'[blk] neste caso, que representa os bloqueios).

Função COUNTROWS: COUNTROWS é usada para contar o número de linhas em uma tabela ou expressão de tabela. Aqui, conta o número de linhas onde há registros de bloqueios.

Função FILTER: FILTER é usada para retornar um conjunto de linhas que satisfazem determinados critérios de filtro. Neste caso, filtra as linhas na tabela 'public fat_performance' onde os bloqueios não são vazios ('public fat_performance'[blk] <> BLANK()).

Uso: Esta medida é útil para calcular a média de bloqueios por jogo, considerando apenas os jogos em que bloqueios foram registrados na tabela 'public fat_performance'.

Nome da Medida: Media_Pontos_por_Jogo

Descrição: Esta medida calcula a média de pontos por jogo na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Media_Pontos_por_Jogo =
DIVIDE(
    SUM('public fat_performance'[pts]),
    COUNTROWS(
        FILTER(
            'public fat_performance',
            'public fat_performance'[pts] <> BLANK()
        )
    ),
    0
)
```

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma dos pontos (SUM('public fat_performance'[pts])) pelo número de jogos onde pontos são registrados.

Função SUM: SUM é usada para somar os valores de uma coluna ('public fat_performance'[pts] neste caso, que representa os pontos).

Função COUNTROWS: COUNTROWS é usada para contar o número de linhas em uma tabela ou expressão de tabela. Aqui, conta o número de linhas onde há registros de pontos.

Função FILTER: FILTER é usada para retornar um conjunto de linhas que satisfazem determinados critérios de filtro. Neste caso, filtra as linhas na tabela 'public fat_performance' onde os pontos não são vazios ('public fat_performance'[pts] <> BLANK()).

Uso: Esta medida é útil para calcular a média de pontos por jogo, considerando apenas os jogos em que pontos foram registrados na tabela 'public fat_performance'.

Nome da Medida: Media_Rebotes_por_Jogo

Descrição: Esta medida calcula a média de rebotes por jogo na tabela 'public fat_performance'.

Expressão DAX:

```
dax
Copiar código
Media_Rebotes_por_Jogo =
DIVIDE(
    SUM('public fat_performance'[trb]),
    COUNTROWS(
        FILTER(
```

```

        'public fat_performance',
        'public fat_performance'[trb] <> BLANK()
    )
),
0
)

```

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma dos rebotes (SUM('public fat_performance'[trb])) pelo número de jogos onde rebotes são registrados.

Função SUM: SUM é usada para somar os valores de uma coluna ('public fat_performance'[trb] neste caso, que representa os rebotes).

Função COUNTROWS: COUNTROWS é usada para contar o número de linhas em uma tabela ou expressão de tabela. Aqui, conta o número de linhas onde há registros de rebotes.

Função FILTER: FILTER é usada para retornar um conjunto de linhas que satisfazem determinados critérios de filtro. Neste caso, filtra as linhas na tabela 'public fat_performance' onde os rebotes não são vazios ('public fat_performance'[trb] <> BLANK()).

Uso: Esta medida é útil para calcular a média de rebotes por jogo, considerando apenas os jogos em que rebotes foram registrados na tabela 'public fat_performance'.

Nome da Medida: Media_Steals_por_Jogo2

Descrição: Esta medida calcula a média de roubos de bola por jogo na tabela 'public fat_performance', considerando apenas os jogadores que têm pelo menos 50 aparições.

Expressão DAX:

```

dax
Copiar código
Media_Steals_por_Jogo2 =
DIVIDE(
    SUM('public fat_performance'[stl]),
    COUNTROWS(
        FILTER(
            'public fat_performance',
            [Aparicoes_Jogador] >= 50
        )
    ),
    0
)

```

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma dos roubos de bola (SUM('public fat_performance'[stl])) pelo número de jogos onde o jogador tem pelo menos 50 aparições.

Função SUM: SUM é usada para somar os valores de uma coluna ('public fat_performance'[stl] neste caso, que representa os roubos de bola).

Função COUNTROWS: COUNTROWS é usada para contar o número de linhas em uma tabela ou expressão de tabela. Aqui, conta o número de jogos onde o jogador tem pelo menos 50 aparições, filtrando a tabela 'public fat_performance'.

Função FILTER: FILTER é usada para retornar um conjunto de linhas que satisfazem determinados critérios de filtro. Neste caso, filtra as linhas na tabela 'public fat_performance' onde [Aparicoes_Jogador] >= 50.

Uso: Esta medida é útil para calcular a média de roubos de bola por jogo apenas para jogadores que têm pelo menos 50 aparições, oferecendo insights sobre o desempenho médio de roubos de bola desses jogadores na tabela 'public fat_performance'.

Nome da Medida: TS%

Descrição: Esta medida calcula o True Shooting Percentage (TS%) na tabela 'public fat_performance'. O TS% é uma métrica estatística usada para avaliar a eficiência geral de arremessos de um jogador, levando em consideração arremessos de campo e lances livres.

Expressão DAX:

```
dax
Copiar código
TS% =
IF(
    COUNTROWS('public fat_performance') > 0,
    DIVIDE(
        SUM('public fat_performance'[pts]),
        2 * (SUM('public fat_performance'[fga]) + 0.44 * SUM('public fat_performance'[fta])),
        0
    ),
    BLANK()
)
```

Função IF: IF é usada para retornar um valor se uma condição for verdadeira e outro valor se for falsa. Neste caso, verifica se há mais de 0 linhas na tabela 'public fat_performance'. Se verdadeiro, executa o cálculo do TS%. Se falso, retorna BLANK().

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma dos pontos (SUM('public fat_performance'[pts])) por 2 vezes a soma dos arremessos de campo tentados (SUM('public fat_performance'[fga])) somado a 0.44 vezes a soma dos lances livres tentados (SUM('public fat_performance'[fta])).

Uso: Esta medida é útil para calcular o True Shooting Percentage (TS%), uma métrica que ajuda a entender a eficiência de arremessos de um jogador, considerando tanto arremessos de campo quanto lances livres.

Nome da Medida: TS%_Com_Filtro

Descrição: Esta medida calcula o True Shooting Percentage (TS%) na tabela 'public fat_performance', aplicando um filtro onde a soma dos pontos (SUM('public fat_performance'[pts])) deve ser maior ou igual a 700 para o cálculo ser realizado.

Expressão DAX:

```
dax
Copiar código
TS%_Com_Filtro =
IF(
    SUM('public fat_performance'[pts]) >= 700,
    DIVIDE(
        SUM('public fat_performance'[pts]),
        2 * (SUM('public fat_performance'[fga]) + 0.44 * SUM('public fat_performance'[fta])),
        0
    ),
    BLANK()
)
```

Função IF: IF é usada para retornar um valor se uma condição for verdadeira e outro valor se for falsa. Neste caso, verifica se a soma dos pontos é maior ou igual a 700. Se verdadeiro, executa o cálculo do TS%. Se falso, retorna BLANK().

Função DIVIDE: DIVIDE é usada para dividir dois valores. Neste caso, divide a soma dos pontos (SUM('public fat_performance'[pts])) por 2 vezes a soma dos arremessos de campo tentados (SUM('public fat_performance'[fga])) somado a 0.44 vezes a soma dos lances livres tentados (SUM('public fat_performance'[fta])).

Uso: Esta medida é útil para calcular o True Shooting Percentage (TS%) dos jogadores na tabela 'public fat_performance', considerando apenas aqueles cuja pontuação total atingiu um limiar específico (neste caso, 700 pontos).

