

# Top 25 Mutual Fund KPIs with SQL and DAX

## 1. Total Assets Under Management (AUM)

**Definition:** Current total value of all investor holdings across all funds **Chart Type:** Card/KPI Visual

**SQL:**

```
sql

SELECT SUM(dollar_amount) as Total_AUM
FROM transactions
WHERE trade_class IN ('Sale', 'Exchange In')
AND trade_date = (SELECT MAX(trade_date) FROM transactions)
```

**DAX:**

```
dax

Total AUM =
CALCULATE(
    SUM(transactions[dollar_amount]),
    transactions[trade_class] IN {"Sale", "Exchange In"}
)
```

---

## 2. Net Sales Flow

**Definition:** Total sales minus total redemptions for the period **Chart Type:** Waterfall Chart

**SQL:**

```
sql

SELECT
    SUM(CASE WHEN trade_class = 'Sale' THEN dollar_amount ELSE 0 END) -
    SUM(CASE WHEN trade_class = 'Redemption' THEN dollar_amount ELSE 0 END) as Net_Sales_Flow
FROM transactions
WHERE trade_date BETWEEN @start_date AND @end_date
```

**DAX:**

```
dax
```

Net Sales Flow =

```
CALCULATE(SUM(transactions[dollar_amount]), transactions[trade_class] = "Sale") -  
CALCULATE(SUM(transactions[dollar_amount]), transactions[trade_class] = "Redemption")
```

### 3. Gross Sales

**Definition:** Total dollar amount of all sale transactions **Chart Type:** Line Chart (trending over time)

**SQL:**

```
sql  
  
SELECT SUM(dollar_amount) as Gross_Sales  
FROM transactions  
WHERE trade_class = 'Sale'  
AND trade_date BETWEEN @start_date AND @end_date
```

**DAX:**

```
dax  
  
Gross Sales =  
CALCULATE(  
    SUM(transactions[dollar_amount]),  
    transactions[trade_class] = "Sale"  
)
```

### 4. Total Redemptions

**Definition:** Total dollar amount of all redemption transactions **Chart Type:** Line Chart

**SQL:**

```
sql  
  
SELECT SUM(dollar_amount) as Total_Redemptions  
FROM transactions  
WHERE trade_class = 'Redemption'  
AND trade_date BETWEEN @start_date AND @end_date
```

**DAX:**

```
dax
```

```
Total Redemptions =  
CALCULATE(  
    SUM(transactions[dollar_amount]),  
    transactions[trade_class] = "Redemption"  
)
```

## 5. Redemption Rate

**Definition:** Percentage of redemptions relative to total AUM **Chart Type:** Gauge/KPI Card

**SQL:**

```
sql  
  
SELECT  
    (SUM(CASE WHEN trade_class = 'Redemption' THEN dollar_amount ELSE 0 END) /  
    SUM(CASE WHEN trade_class IN ('Sale', 'Exchange In') THEN dollar_amount ELSE 0 END)) * 100  
    as Redemption_Rate_Percent  
FROM transactions
```

**DAX:**

```
dax  
  
Redemption Rate =  
DIVIDE(  
    [Total Redemptions],  
    [Total AUM],  
    0  
) * 100
```

## 6. Sales Growth Rate (MoM)

**Definition:** Month-over-month percentage change in gross sales **Chart Type:** Line Chart with trend

**SQL:**

```
sql
```

```

WITH monthly_sales AS (
    SELECT
        YEAR(trade_date) as year,
        MONTH(trade_date) as month,
        SUM(dollar_amount) as monthly_total
    FROM transactions
    WHERE trade_class = 'Sale'
    GROUP BY YEAR(trade_date), MONTH(trade_date)
)
SELECT
    ((current_month.monthly_total - prev_month.monthly_total) / prev_month.monthly_total) * 100 as Growth_Rate
FROM monthly_sales current_month
JOIN monthly_sales prev_month ON current_month.month = prev_month.month + 1

```

## DAX:

```

dax

Sales Growth Rate MoM =
VAR CurrentMonthSales = [Gross Sales]
VAR PreviousMonthSales =
    CALCULATE(
        [Gross Sales],
        DATEADD('Date'[Date], -1, MONTH)
    )
RETURN
    DIVIDE(CurrentMonthSales - PreviousMonthSales, PreviousMonthSales, 0) * 100

```

## 7. Average Transaction Size

**Definition:** Average dollar amount per transaction **Chart Type:** Card/Column Chart by trade class

## SQL:

```

sql

SELECT AVG(dollar_amount) as Avg_Transaction_Size
FROM transactions
WHERE trade_class IN ('Sale', 'Redemption')
AND trade_date BETWEEN @start_date AND @end_date

```

## DAX:

```

dax

```

Average Transaction Size =  
`AVERAGE(transactions[dollar_amount])`

## 8. Top 10 Funds by AUM

**Definition:** Ranking of funds by total assets under management **Chart Type:** Horizontal Bar Chart

**SQL:**

```
sql

SELECT TOP 10
    fund_name,
    SUM(dollar_amount) as Fund_AUM
FROM transactions
WHERE trade_class IN ('Sale', 'Exchange In')
GROUP BY fund_name
ORDER BY Fund_AUM DESC
```

**DAX:**

```
dax

Fund AUM Rank =
RANKX(
    ALL(transactions[fund_name]),
    [Total AUM],
    ,
    DESC
)
```

## 9. Sales by Representative

**Definition:** Total sales generated by each sales representative **Chart Type:** Column Chart/Table

**SQL:**

```
sql
```

```

SELECT
    sales_representative,
    SUM(dollar_amount) as Rep_Sales
FROM transactions
WHERE trade_class = 'Sale'
    AND trade_date BETWEEN @start_date AND @end_date
GROUP BY sales_representative
ORDER BY Rep_Sales DESC

```

## DAX:

```

dax

Sales by Rep =
CALCULATE(
    SUM(transactions[dollar_amount]),
    transactions[trade_class] = "Sale"
)

```

## 10. Office Performance

**Definition:** Sales performance comparison across offices **Chart Type:** Clustered Column Chart

## SQL:

```

sql

SELECT
    office,
    SUM(CASE WHEN trade_class = 'Sale' THEN dollar_amount ELSE 0 END) as Office_Sales,
    SUM(CASE WHEN trade_class = 'Redemption' THEN dollar_amount ELSE 0 END) as Office_Redemptions
FROM transactions
WHERE trade_date BETWEEN @start_date AND @end_date
GROUP BY office

```

## DAX:

```

dax

Office Net Sales =
CALCULATE([Gross Sales] - [Total Redemptions])

```

## 11. Transaction Volume

**Definition:** Total number of transactions by type **Chart Type:** Donut Chart

## SQL:

```
sql

SELECT
    trade_class,
    COUNT(*) as Transaction_Count
FROM transactions
WHERE trade_date BETWEEN @start_date AND @end_date
GROUP BY trade_class
```

## DAX:

```
dax

Transaction Count =
COUNTROWS(transactions)
```

---

## 12. Exchange Activity Ratio

**Definition:** Ratio of exchange-in to exchange-out transactions **Chart Type:** Gauge

## SQL:

```
sql

SELECT
    SUM(CASE WHEN trade_class = 'Exchange In' THEN dollar_amount ELSE 0 END) /
    SUM(CASE WHEN trade_class = 'Exchange Out' THEN dollar_amount ELSE 0 END) as Exchange_Ratio
FROM transactions
```

## DAX:

```
dax

Exchange Ratio =
DIVIDE(
    CALCULATE(SUM(transactions[dollar_amount]), transactions[trade_class] = "Exchange In"),
    CALCULATE(SUM(transactions[dollar_amount]), transactions[trade_class] = "Exchange Out"),
    0
)
```

---

## 13. Daily Trading Volume

**Definition:** Daily aggregate trading activity **Chart Type:** Line Chart

## SQL:

```
sql

SELECT
    trade_date,
    SUM(dollar_amount) as Daily_Volume
FROM transactions
WHERE trade_class IN ('Sale', 'Redemption', 'Exchange In', 'Exchange Out')
GROUP BY trade_date
ORDER BY trade_date
```

## DAX:

```
dax

Daily Trading Volume =
CALCULATE(
    SUM(transactions[dollar_amount]),
    transactions[trade_class] IN {"Sale", "Redemption", "Exchange In", "Exchange Out"}
)
```

## 14. Fund Concentration Risk

**Definition:** Percentage of AUM in top 5 funds **Chart Type:** Stacked Bar Chart

## SQL:

```
sql

WITH fund_totals AS (
    SELECT
        fund_name,
        SUM(dollar_amount) as fund_aum,
        ROW_NUMBER() OVER (ORDER BY SUM(dollar_amount) DESC) as rank
    FROM transactions
    WHERE trade_class IN ('Sale', 'Exchange In')
    GROUP BY fund_name
)
SELECT
    (SUM(CASE WHEN rank <= 5 THEN fund_aum ELSE 0 END) / SUM(fund_aum)) * 100 as Top5_Concentration
FROM fund_totals
```

## DAX:

```
dax
```



```

Top 5 Fund Concentration =
VAR Top5Funds =
    TOPN(5, VALUES(transactions[fund_name]), [Total AUM], DESC)
VAR Top5AUM =
    CALCULATE([Total AUM], transactions[fund_name] IN Top5Funds)
RETURN
    DIVIDE(Top5AUM, [Total AUM], 0) * 100

```

## 15. Settlement Efficiency

**Definition:** Percentage of transactions successfully settled **Chart Type:** KPI Card with trend

**SQL:**

```

sql

SELECT
    (COUNT(CASE WHEN trade_class = 'Settle' THEN 1 END) * 100.0 /
    COUNT(CASE WHEN trade_class IN ('Sale', 'Redemption') THEN 1 END)) as Settlement_Rate
FROM transactions

```

**DAX:**

```

dax

Settlement Rate =
VAR SettledTransactions = CALCULATE(COUNTROWS(transactions), transactions[trade_class] = "Settle")
VAR TotalTransactions = CALCULATE(COUNTROWS(transactions), transactions[trade_class] IN {"Sale", "Redemption"})
RETURN
    DIVIDE(SettledTransactions, TotalTransactions, 0) * 100

```

## 16. Year-to-Date Net Flows

**Definition:** Cumulative net flows from beginning of year **Chart Type:** Area Chart

**SQL:**

```

sql

SELECT
    SUM(CASE WHEN trade_class = 'Sale' THEN dollar_amount ELSE 0 END) -
    SUM(CASE WHEN trade_class = 'Redemption' THEN dollar_amount ELSE 0 END) as YTD_Net_Flows
FROM transactions
WHERE YEAR(trade_date) = YEAR(GETDATE())

```

## DAX:

```
dax

YTD Net Flows =
CALCULATE(
    [Net Sales Flow],
    DATESYTD('Date'[Date])
)
```

## 17. Client Retention Rate

**Definition:** Percentage of accounts with no redemptions in period **Chart Type:** Gauge

## SQL:

```
sql

SELECT
    (COUNT(DISTINCT CASE WHEN redemption_flag = 0 THEN investor_account_number END) * 100.0 /
     COUNT(DISTINCT investor_account_number)) as Retention_Rate
FROM (
    SELECT
        investor_account_number,
        MAX(CASE WHEN trade_class = 'Redemption' THEN 1 ELSE 0 END) as redemption_flag
    FROM transactions
    WHERE trade_date BETWEEN @start_date AND @end_date
    GROUP BY investor_account_number
) account_activity
```

## DAX:

```
dax

Client Retention Rate =
VAR AccountsWithRedemptions =
    CALCULATE(
        DISTINCTCOUNT(transactions[investor_account_number]),
        transactions[trade_class] = "Redemption"
    )
VAR TotalAccounts = DISTINCTCOUNT(transactions[investor_account_number])
RETURN
    DIVIDE(TotalAccounts - AccountsWithRedemptions, TotalAccounts, 0) * 100
```

## 18. Average Days to Settlement

**Definition:** Average time between trade date and settlement **Chart Type:** Line Chart with target line

### SQL:

```
sql

SELECT
    AVG(DATEDIFF(day, t1.trade_date, t2.trade_date)) as Avg_Days_to_Settlement
FROM transactions t1
JOIN transactions t2 ON t1.unique_transaction_id = t2.unique_transaction_id
WHERE t1.trade_class IN ('Sale', 'Redemption')
AND t2.trade_class = 'Settle'
```

### DAX:

```
dax

Avg Days to Settlement =
AVERAGEX(
    FILTER(
        transactions,
        transactions[trade_class] = "Settle"
    ),
    DATEDIFF(
        RELATED(transactions[trade_date]),
        transactions[trade_date],
        DAY
    )
)
```

---

## 19. Market Share by Firm

**Definition:** Percentage of total sales by firm **Chart Type:** Pie Chart

### SQL:

```
sql

SELECT
    firm,
    (SUM(dollar_amount) * 100.0 / (SELECT SUM(dollar_amount) FROM transactions WHERE trade_class = 'Sale')) as Market_Share
FROM transactions
WHERE trade_class = 'Sale'
GROUP BY firm
```

---

## DAX:

```
dax

Market Share by Firm =
DIVIDE(
    [Gross Sales],
    CALCULATE([Gross Sales], ALL(transactions[firm])),
    0
) * 100
```

## 20. Sales Momentum (3-Month Rolling)

**Definition:** 3-month rolling average of monthly sales **Chart Type:** Line Chart with multiple series

## SQL:

```
sql

SELECT
    trade_date,
    AVG(SUM(dollar_amount)) OVER (
        ORDER BY YEAR(trade_date), MONTH(trade_date)
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
    ) as Rolling_3M_Sales
FROM transactions
WHERE trade_class = 'Sale'
GROUP BY YEAR(trade_date), MONTH(trade_date), trade_date
```

## DAX:

```
dax

3M Rolling Sales =
CALCULATE(
    [Gross Sales],
    DATESINPERIOD('Date'[Date], LASTDATE('Date'[Date]), -3, MONTH)
) / 3
```

## 21. Exception Rate

**Definition:** Percentage of transactions marked as 'ignore' **Chart Type:** KPI Card with alert formatting

## SQL:

```
sql
```

```

SELECT
    (COUNT(CASE WHEN trade_class = 'Ignore' THEN 1 END) * 100.0 / COUNT(*)) as Exception_Rate
FROM transactions
WHERE trade_date BETWEEN @start_date AND @end_date

```

## DAX:

```

dax

Exception Rate =
DIVIDE(
    CALCULATE(COUNTROWS(transactions), transactions[trade_class] = "Ignore"),
    COUNTROWS(transactions),
    0
) * 100

```

## 22. NSCC Processing Volume

**Definition:** Volume of transactions by NSCC number **Chart Type:** Table/Matrix

## SQL:

```

sql

SELECT
    nsc_number,
    COUNT(*) as Transaction_Count,
    SUM(dollar_amount) as Total_Volume
FROM transactions
WHERE nsc_number IS NOT NULL
GROUP BY nsc_number
ORDER BY Total_Volume DESC

```

## DAX:

```

dax

NSCC Volume =
CALCULATE(
    SUM(transactions[dollar_amount]),
    NOT ISBLANK(transactions[nsc_number])
)

```

## 23. Fund Flow Velocity

**Definition:** Rate of change in fund flows (momentum indicator) **Chart Type:** Scatter Plot

**SQL:**

```
sql

WITH monthly_flows AS (
  SELECT
    fund_name,
    YEAR(trade_date) as year,
    MONTH(trade_date) as month,
    SUM(CASE WHEN trade_class = 'Sale' THEN dollar_amount ELSE -dollar_amount END) as net_flow
  FROM transactions
  WHERE trade_class IN ('Sale', 'Redemption')
  GROUP BY fund_name, YEAR(trade_date), MONTH(trade_date)
)
SELECT
  fund_name,
  (current_flow - lag_flow) / ABS(lag_flow) * 100 as Flow_Velocity
FROM (
  SELECT
    fund_name,
    net_flow as current_flow,
    LAG(net_flow) OVER (PARTITION BY fund_name ORDER BY year, month) as lag_flow
  FROM monthly_flows
) flow_comparison
```

**DAX:**

```
dax

Fund Flow Velocity =
VAR CurrentFlow = [Net Sales Flow]
VAR PreviousFlow =
  CALCULATE(
    [Net Sales Flow],
    DATEADD('Date'[Date], -1, MONTH)
  )
RETURN
  DIVIDE(CurrentFlow - PreviousFlow, ABS(PreviousFlow), 0) * 100
```

---

## 24. Peak Trading Day Volume

**Definition:** Highest single-day trading volume in period **Chart Type:** Card with historical comparison

**SQL:**

sql

```
SELECT MAX(daily_volume) as Peak_Daily_Volume
FROM (
    SELECT
        trade_date,
        SUM(dollar_amount) as daily_volume
    FROM transactions
    WHERE trade_class IN ('Sale', 'Redemption', 'Exchange In', 'Exchange Out')
    AND trade_date BETWEEN @start_date AND @end_date
    GROUP BY trade_date
) daily_totals
```

## DAX:

dax

```
Peak Trading Day =
MAXX(
    SUMMARIZE(
        transactions,
        'Date'[Date],
        "Daily Volume", [Daily Trading Volume]
    ),
    [Daily Volume]
)
```

## 25. Investor Account Growth

**Definition:** Net new investor accounts in period **Chart Type:** Column Chart with trend line

### SQL:

sql

```
WITH account_first_trade AS (
    SELECT
        investor_account_number,
        MIN(trade_date) as first_trade_date
    FROM transactions
    GROUP BY investor_account_number
)
SELECT
    COUNT(*) as New_Accounts
FROM account_first_trade
WHERE first_trade_date BETWEEN @start_date AND @end_date
```

## DAX:

dax

New Account Growth =

VAR CurrentPeriodAccounts =

```
CALCULATE(  
    DISTINCTCOUNT(transactions[investor_account_number]),  
    'Date'[Date] >= STARTOFMONTH('Date'[Date])  
)
```

VAR PreviousPeriodAccounts =

```
CALCULATE(  
    DISTINCTCOUNT(transactions[investor_account_number]),  
    DATEADD('Date'[Date], -1, MONTH)  
)
```

RETURN

CurrentPeriodAccounts - PreviousPeriodAccounts

## Implementation Notes:

1. **Date Parameters:** Replace `@start_date` and `@end_date` with appropriate date filters in your queries
2. **Performance:** Consider indexing on `trade_date`, `trade_class`, and `fund_name` for better query performance
3. **Data Quality:** Implement validation rules for the 'ignore' trade class to maintain data integrity
4. **Refresh Schedule:** Set up appropriate refresh schedules based on data update frequency
5. **Security:** Implement row-level security if different users need access to different firms/offices