Creating a Date Table with DAX

First, let's create a custom date table that will serve as the foundation for all time-based analysis:

dax

```
Date =
ADDCOLUMNS(
    CALENDAR(DATE(2018,1,1), DATE(2026,12,31)),
    "DateKey", FORMAT([Date], "YYYYMMDD"),
    "Year", YEAR([Date]),
    "Quarter", "Q" & FORMAT(QUARTER([Date]), "0"),
    "Month Number", MONTH([Date]),
    "Month", FORMAT([Date], "MMMM"),
    "Month Short", FORMAT([Date], "MMM"),
    "Month Year", FORMAT([Date], "MMM YYYY"),
    "Week Number", WEEKNUM([Date]),
    "Week Day Number", WEEKDAY([Date]),
    "Day Name", FORMAT([Date], "DDDD"),
    "Day Short", FORMAT([Date], "DDD"),
    "Is Weekend", IF(WEEKDAY([Date],2) > 5, TRUE(), FALSE()),
    "Is Current Month", IF(AND(YEAR([Date]) = YEAR(TODAY()), MONTH([Date]) = MONTH(TODAY())), TRUE(),
FALSE()),
    "Is Current Year", IF(YEAR([Date]) = YEAR(TODAY()), TRUE(), FALSE()),
    "YTD", IF(AND([Date] <= TODAY(), YEAR([Date]) = YEAR(TODAY())), TRUE(), FALSE()),
    "MTD", IF(AND([Date] <= TODAY(), YEAR([Date]) = YEAR(TODAY()), MONTH([Date]) = MONTH(TODAY())),
TRUE(), FALSE()),
    "QTD", IF(AND([Date] <= TODAY(), YEAR([Date]) = YEAR(TODAY()), QUARTER([Date]) = QUARTER(TODAY())),
TRUE(), FALSE()),
    "Day of Month", DAY([Date]),
    "Day of Year", DATEDIFF(DATE(YEAR([Date]),1,1), [Date], DAY) + 1
)
```

Core Business Metrics

Membership Metrics

dax

```
// Total Members
Total Members = COUNTROWS(MEMBERS)

// Active Members (has active subscription)
Active Members =
CALCULATE(
    COUNTROWS(MEMBERS),
    FILTER(
        SUBSCRIPTION,
        SUBSCRIPTION[status] = "Active"
    )
)

// Monthly Active Members (checked in within last 30 days)
Monthly Active Members =
CALCULATE(
    DISTINCTCOUNT(CHECKIN[member_id]),
    FILTER(
        CHECKIN,
        CHECKIN[check_in_date] >= TODAY() - 30
    )
)

// Inactive Members
Inactive Members = [Total Members] - [Active Members]

// New Members (joined in current month)
New Members MTD =
CALCULATE(
    COUNTROWS(MEMBERS),
    FILTER(
        MEMBERS,
        AND(
            MONTH(MEMBERS[join_date]) = MONTH(TODAY()),
            YEAR(MEMBERS[join_date]) = YEAR(TODAY())
        )
    )
)

// New Members (joined in current year)
New Members YTD =
CALCULATE(
    COUNTROWS(MEMBERS),
    FILTER(
        MEMBERS,
        AND(
            MEMBERS[join_date] >= DATE(YEAR(TODAY()), 1, 1),
            MEMBERS[join_date] <= TODAY()
        )
    )
)
```

Churn Metrics

dax

```
// Member Churn Count (Members who cancelled in selected period)
Member Churn Count =
CALCULATE(
    COUNTROWS(SUBSCRIPTION),
    FILTER(
        SUBSCRIPTION,
        AND(
            SUBSCRIPTION[status] = "Cancelled",
            SUBSCRIPTION[end_date] >= MIN(Date[Date]),
            SUBSCRIPTION[end_date] <= MAX(Date[Date])
        )
    )
)

// Member Churn Rate
Member Churn Rate =
DIVIDE(
    [Member Churn Count],
    [Total Members],
    0
)

// Monthly Churn Rate
Monthly Churn Rate =
VAR ChurnedThisMonth =
    CALCULATE(
        COUNTROWS(SUBSCRIPTION),
        FILTER(
            SUBSCRIPTION,
            AND(
                SUBSCRIPTION[status] = "Cancelled",
                MONTH(SUBSCRIPTION[end_date]) = MONTH(TODAY()),
                YEAR(SUBSCRIPTION[end_date]) = YEAR(TODAY())
            )
        )
    )
VAR ActiveStartOfMonth =
    CALCULATE(
        COUNTROWS(SUBSCRIPTION),
        FILTER(
            SUBSCRIPTION,
            SUBSCRIPTION[status] = "Active" ||
            (SUBSCRIPTION[status] = "Cancelled" &&
             MONTH(SUBSCRIPTION[end_date]) = MONTH(TODAY()) &&
             YEAR(SUBSCRIPTION[end_date]) = YEAR(TODAY()))
        ),
        LASTDATE(EOMONTH(TODAY(), -1))
    )
RETURN
    DIVIDE(ChurnedThisMonth, ActiveStartOfMonth, 0)
```

Lifetime Value Metrics

dax

```
// Average Member Tenure (in months)
Average Member Tenure =
VAR CurrentMembers =
    CALCULATETABLE(
        SUMMARIZE(
            MEMBERS,
            MEMBERS[member_id],
            "Tenure", DATEDIFF(MEMBERS[join_date], TODAY(), MONTH)
        ),
        FILTER(
            SUBSCRIPTION,
            SUBSCRIPTION[status] = "Active"
        )
    )
RETURN
    AVERAGEX(CurrentMembers, [Tenure])

// Average Monthly Revenue per Member
Avg Monthly Revenue per Member =
DIVIDE(
    CALCULATE(
        SUM(REVENUE[total_revenue]),
        DATESINPERIOD(Date[Date], TODAY(), -30, DAY)
    ),
    [Active Members],
    0
)

// Member Lifetime Value
Member Lifetime Value =
[Avg Monthly Revenue per Member] *
[Average Member Tenure] *
(1 - [Monthly Churn Rate])
```

Revenue Metrics

dax

```
// Total Revenue
Total Revenue = SUM(REVENUE[total_revenue])

// Subscription Revenue
Subscription Revenue = SUM(REVENUE[subscription_revenue])

// Other Revenue
Other Revenue = SUM(REVENUE[other_revenue])

// PT Revenue (Personal Training Revenue)
PT Revenue =
CALCULATE(
    SUM(REVENUE[other_revenue]),
    FILTER(
        REVENUE,
        REVENUE[revenue_type] = "PT"
    )
)

// Revenue per PT Hour
Revenue per PT Hour =
DIVIDE(
    [PT Revenue],
    CALCULATE(
        SUM(TRAINER[worked_hours]),
        FILTER(
            TRAINER,
            TRAINER[activity_type] = "PT"
        )
    ),
    0
)
```

```
// Revenue per Gym
Revenue per Gym =
DIVIDE(
    [Total Revenue],
    DISTINCTCOUNT(GYM[gym_id])
)

// Revenue YoY Growth
Revenue YoY Growth =
VAR CurrentYearRev =
    CALCULATE(
        [Total Revenue],
        FILTER(
            ALL(Date),
            Date[Year] = YEAR(TODAY())
        )
    )
VAR PrevYearRev =
    CALCULATE(
        [Total Revenue],
        FILTER(
            ALL(Date),
            Date[Year] = YEAR(TODAY()) - 1
        )
    )
RETURN
    DIVIDE(CurrentYearRev - PrevYearRev, PrevYearRev, 0)

// MTD Revenue
MTD Revenue =
CALCULATE(
    [Total Revenue],
    FILTER(
        ALL(Date),
        AND(
            Date[Year] = YEAR(TODAY()),
            Date[Month Number] = MONTH(TODAY()),
            Date[Day of Month] <= DAY(TODAY())
        )
    )
)

// QTD Revenue
QTD Revenue =
CALCULATE(
    [Total Revenue],
    FILTER(
        ALL(Date),
        Date[QTD] = TRUE()
    )
)

// Rolling 3-Month Revenue
Rolling 3-Month Revenue =
CALCULATE(
    [Total Revenue],
    DATESINPERIOD(Date[Date], TODAY(), -3, MONTH)
)
```

Conversion Funnel Metrics

dax

```
// Total Leads
Total Leads = COUNTROWS(LEADS)

// Contacted Leads
Contacted Leads =
CALCULATE(
    COUNTROWS(LEADS),
    LEADS[status] = "Contacted"
)
```

```
// Converted Leads
Converted Leads =
CALCULATE(
    COUNTROWS(LEADS),
    LEADS[status] = "Converted"
)

// Lost Leads
Lost Leads =
CALCULATE(
    COUNTROWS(LEADS),
    LEADS[status] = "Lost"
)

// Conversion Rate
Conversion Rate =
DIVIDE(
    [Converted Leads],
    [Total Leads],
    0
)

// Average Conversion Time (days)
Avg Conversion Time =
AVERAGEX(
    FILTER(
        LEADS,
        AND(
            LEADS[status] = "Converted",
            NOT(ISBLANK(LEADS[conversion_date]))
        )
    ),
    DATEDIFF(LEADS[lead_date], LEADS[conversion_date], DAY)
)
```

Trainer Performance Metrics

dax

```
// Total Trainer Hours
Total Trainer Hours =
SUMX(
    TRAINER,
    DATEDIFF(TRAINER[punch_in], TRAINER[punch_out], HOUR)
)

// PT Session Hours
PT Session Hours =
CALCULATE(
    [Total Trainer Hours],
    FILTER(
        TRAINER,
        TRAINER[activity_type] = "PT"
    )
)

// Trainer Idle Hours
Trainer Idle Hours =
CALCULATE(
    [Total Trainer Hours],
    FILTER(
        TRAINER,
        TRAINER[activity_type] = "Idle"
    )
)

// Trainer Utilization Rate
Trainer Utilization Rate =
DIVIDE(
    [PT Session Hours],
    [Total Trainer Hours],
    0
)
```

```
// Average PT Sessions per Trainer
Avg PT Sessions per Trainer =
DIVIDE(
    CALCULATE(
        COUNTROWS(TRAINER),
        FILTER(
            TRAINER,
            TRAINER[activity_type] = "PT"
        )
    ),
    DISTINCTCOUNT(TRAINER[trainer_id]),
    0
)

// Revenue per Trainer
Revenue per Trainer =
DIVIDE(
    [PT Revenue],
    DISTINCTCOUNT(TRAINER[trainer_id]),
    0
)
```

Gym Performance Metrics

dax

```
// Gym Health Score (composite KPI)
Gym Health Score =
VAR RetentionScore = 1 - [Monthly Churn Rate]
VAR UtilizationScore = [Trainer Utilization Rate]
VAR RevenueScore =
    DIVIDE(
        [Revenue per Gym],
        AVERAGEX(VALUES(GYM[gym_id]), [Revenue per Gym]),
        0
    )
RETURN
    (RetentionScore * 40 + UtilizationScore * 30 + RevenueScore * 30) / 100

// ROCE (Return on Capital Employed)
ROCE =
DIVIDE(
    [Total Revenue] - [Total Expenses],
    [Capital Employed],
    0
)

// Days Since Opening
Days Since Opening =
DATEDIFF(GYM[open_date], TODAY(), DAY)

// Gym Density (members per sq ft)
Gym Density =
DIVIDE(
    [Active Members],
    GYM[area_sqft],
    0
)
```

Forecasting Measures

dax

```
// Revenue Forecast (Simple Moving Average)
Revenue Forecast =
VAR HistoricalPeriods = 3
VAR CurrentPeriodRev =
    CALCULATE(
        [Total Revenue],
        DATESINPERIOD(Date[Date], TODAY(), -1, MONTH)
    )
VAR PrevPeriod1Rev =
    CALCULATE(
        [Total Revenue],
        DATESINPERIOD(Date[Date], DATEADD(TODAY(), -1, MONTH), -1, MONTH)
    )
VAR PrevPeriod2Rev =
    CALCULATE(
        [Total Revenue],
        DATESINPERIOD(Date[Date], DATEADD(TODAY(), -2, MONTH), -1, MONTH)
    )
RETURN
    (CurrentPeriodRev + PrevPeriod1Rev + PrevPeriod2Rev) / HistoricalPeriods

// Churn Forecast
Churn Forecast =
VAR HistoricalPeriods = 3
VAR CurrentPeriodChurn =
    CALCULATE(
        [Monthly Churn Rate],
        DATESINPERIOD(Date[Date], TODAY(), -1, MONTH)
    )
VAR PrevPeriod1Churn =
    CALCULATE(
        [Monthly Churn Rate],
        DATESINPERIOD(Date[Date], DATEADD(TODAY(), -1, MONTH), -1, MONTH)
    )
VAR PrevPeriod2Churn =
    CALCULATE(
        [Monthly Churn Rate],
        DATESINPERIOD(Date[Date], DATEADD(TODAY(), -2, MONTH), -1, MONTH)
    )
RETURN
    (CurrentPeriodChurn + PrevPeriod1Churn + PrevPeriod2Churn) / HistoricalPeriods
```

Additional Helper Measures for RLS and Reporting

dax

```
// Get Cluster ID (for RLS)
Cluster ID = SELECTEDVALUE(GYM[cluster_id])

// Is Current User Cluster Manager (for RLS)
Is Cluster Manager =
IF(
    USERNAME() = "cluster_manager@wtfgym.com" ||
    CONTAINSSTRING(USERNAME(), "cluster_"),
    TRUE(),
    FALSE()
)
```

Page-Specific DAX Measures

Executive Summary Page

dax

```
// Net Growth (New members minus churned members)
Net Member Growth =
CALCULATE(
    COUNTROWS(MEMBERS),
    FILTER(
        MEMBERS,
        MEMBERS[join_date] >= MIN(Date[Date]) &&
        MEMBERS[join_date] <= MAX(Date[Date])
    )
) - [Member Churn Count]

// YoY Net Growth %
YoY Net Growth % =
VAR CurrentYearGrowth =
    CALCULATE(
        [Net Member Growth],
        FILTER(
            ALL(Date),
            Date[Year] = YEAR(TODAY())
        )
    )
VAR PrevYearGrowth =
    CALCULATE(
        [Net Member Growth],
        FILTER(
            ALL(Date),
            Date[Year] = YEAR(TODAY()) - 1
        )
    )
RETURN
    DIVIDE(CurrentYearGrowth - PrevYearGrowth, ABS(PrevYearGrowth), 0)

// Cluster Performance Score
Cluster Performance Score =
AVERAGEX(
    VALUES(GYM[cluster_id]),
    [Gym Health Score]
)

// Top Performing Gym
Top Performing Gym =
VAR TopGym =
    TOPN(1, VALUES(GYM[gym_name]), [Gym Health Score], DESC)
RETURN
    SELECTEDVALUE(TopGym)

// Bottom Performing Gym
Bottom Performing Gym =
VAR BottomGym =
    TOPN(1, VALUES(GYM[gym_name]), [Gym Health Score], ASC)
RETURN
    SELECTEDVALUE(BottomGym)
```

Sales & PT Dashboard

dax

```
// Daily Revenue
Daily Revenue =
CALCULATE(
    [Total Revenue],
    FILTER(
        ALL(Date),
        Date[Date] = SELECTEDVALUE(Date[Date])
    )
)

// Weekly Revenue
Weekly Revenue =
CALCULATE(
    [Total Revenue],
    DATESINPERIOD(Date[Date], MAX(Date[Date]), -7, DAY)
)

// Revenue Target
Revenue Target =
// Simple target based on 10% growth from previous year
VAR PrevYearSameMonthRev =
    CALCULATE(
        [Total Revenue],
        SAMEPERIODLASTYEAR(Date[Date])
    )
RETURN
    PrevYearSameMonthRev * 1.1

// Revenue vs Target %
Revenue vs Target % =
DIVIDE(
    [Total Revenue],
    [Revenue Target],
    0
)
```

```
// PT Session Count
PT Session Count =
CALCULATE(
    COUNTROWS(TRAINER),
    FILTER(
        TRAINER,
        TRAINER[activity_type] = "PT"
    )
)

// Product Sales Revenue
Product Sales Revenue =
CALCULATE(
    SUM(REVENUE[other_revenue]),
    FILTER(
        REVENUE,
        REVENUE[revenue_type] = "Product"
    )
)
```

Retention & Churn Analysis Page

dax

```
// Recent Churners (Last 30 days)
Recent Churners =
CALCULATE(
    COUNTROWS(SUBSCRIPTION),
    FILTER(
        SUBSCRIPTION,
        AND(
            SUBSCRIPTION[status] = "Cancelled",
            SUBSCRIPTION[end_date] >= TODAY() - 30,
            SUBSCRIPTION[end_date] <= TODAY()
        )
    )
)

// At-Risk Members
At Risk Members =
CALCULATE(
    COUNTROWS(MEMBERS),
    FILTER(
        CHECKIN,
        CHECKIN[check_in_date] < TODAY() - 14 &&
        CHECKIN[check_in_date] >= TODAY() - 30
    )
)

// Reactivated Members
Reactivated Members =
VAR ReactivatedMembersTable =
    FILTER(
        SUBSCRIPTION,
        SUBSCRIPTION[status] = "Active" &&
        EARLIER(SUBSCRIPTION[member_id]) = SUBSCRIPTION[member_id] &&
        SUBSCRIPTION[start_date] > EARLIER(SUBSCRIPTION[end_date])
    )
RETURN
    COUNTROWS(ReactivatedMembersTable)

// Membership Length Categories
Membership Length Category =
VAR MembershipLength =
    SWITCH(
        TRUE(),
        [Average Member Tenure] < 3, "0-3 Months",
        [Average Member Tenure] < 6, "3-6 Months",
        [Average Member Tenure] < 12, "6-12 Months",
        [Average Member Tenure] < 24, "1-2 Years",
        "2+ Years"
    )
```

```
RETURN
    MembershipLength

// Dropout Rate by Subscription Type
Dropout Rate by Subscription Type =
DIVIDE(
    CALCULATE(
        [Member Churn Count],
        FILTER(
            ALL(MEMBERS),
            MEMBERS[subscription_plan] = SELECTEDVALUE(MEMBERS[subscription_plan])
        )
    ),
    CALCULATE(
        [Total Members],
        FILTER(
            ALL(MEMBERS),
            MEMBERS[subscription_plan] = SELECTEDVALUE(MEMBERS[subscription_plan])
        )
    ),
    0
)
```

Trainer Performance Page

dax

```
// Sessions Target per Trainer
Sessions Target per Trainer = 20 // Example value, adjust as needed

// Sessions vs Target %
Sessions vs Target % =
DIVIDE(
    [PT Session Count],
    [Sessions Target per Trainer] * DISTINCTCOUNT(TRAINER[trainer_id]),
    0
)

// Average Revenue per Session
Avg Revenue per Session =
DIVIDE(
    [PT Revenue],
    [PT Session Count],
    0
)

// Trainer Efficiency Score
Trainer Efficiency Score =
VAR RevPerHourScore =
    DIVIDE(
        [Revenue per PT Hour],
        AVERAGEX(ALL(TRAINER), [Revenue per PT Hour]),
        0
    ) * 50
VAR UtilizationScore = [Trainer Utilization Rate] * 50
RETURN
    RevPerHourScore + UtilizationScore

// Top Performing Trainer
Top Performing Trainer =
VAR TopTrainer =
    TOPN(1, VALUES(TRAINER[name]), [Trainer Efficiency Score], DESC)
RETURN
    SELECTEDVALUE(TopTrainer)

// Idle Hours Cost (Assuming hourly rate of $20)

Idle Hours Cost = [Trainer Idle Hours] * 20
```

Marketing Funnel Page

dax

```
// Lead to Member Conversion Time (days)
Lead to Member Conversion Time =
AVERAGEX(
    FILTER(
        LEADS,
        LEADS[status] = "Converted"
    ),
    DATEDIFF(LEADS[lead_date], LEADS[conversion_date], DAY)
)

// Lead Quality Score (based on conversion rate)
Lead Quality Score =
VAR ConvRate =
    DIVIDE(
        CALCULATE(
            COUNTROWS(LEADS),
            LEADS[status] = "Converted",
            SELECTEDVALUE(LEADS[lead_source]) = LEADS[lead_source]
        ),
        CALCULATE(
            COUNTROWS(LEADS),
            SELECTEDVALUE(LEADS[lead_source]) = LEADS[lead_source]
        ),
        0
    )
RETURN
    IF(ConvRate >= 0.5, "High", IF(ConvRate >= 0.25, "Medium", "Low"))

// Cost Per Acquisition (Mock data)
Cost Per Acquisition =
SWITCH(
    SELECTEDVALUE(LEADS[lead_source]),
    "Referral", 50,
    "Social Media", 120,
    "Website", 80,
    "Walk-in", 30,
    "Promotion", 150,
    100 // Default value
)

// Lead Source ROI
Lead Source ROI =
DIVIDE(
    [Member Lifetime Value] * [Converted Leads],
    [Cost Per Acquisition] * [Total Leads],
    0
)

// Lead Closing Rate by Source
Lead Closing Rate by Source =
DIVIDE(
    CALCULATE(
        COUNTROWS(LEADS),
        LEADS[status] = "Converted"
    ),
    CALCULATE(
        COUNTROWS(LEADS)
    ),
    0

)
```

Segmentation Measures for AI Insights

dax

```
// Member Frequency Category
Member Frequency Category =
VAR LastCheckinDate = MAX(CHECKIN[check_in_date])
VAR DaysSinceLastCheckin = DATEDIFF(LastCheckinDate, TODAY(), DAY)
RETURN
    SWITCH(
        TRUE(),
        DaysSinceLastCheckin <= 7, "Frequent",
        DaysSinceLastCheckin <= 14, "Regular",
        DaysSinceLastCheckin <= 30, "Occasional",
        "Inactive"
    )

// Member Value Segment
Member Value Segment =
VAR MemberLTV =
    CALCULATE(
        [Member Lifetime Value],
        ALLEXCEPT(MEMBERS, MEMBERS[member_id])
    )
RETURN
    SWITCH(
        TRUE(),
        MemberLTV >= PERCENTILE.INC(VALUES([Member Lifetime Value]), 0.8), "Premium",
        MemberLTV >= PERCENTILE.INC(VALUES([Member Lifetime Value]), 0.5), "Standard",
        "Basic"
    )

// Churn Risk Score
Churn Risk Score =
VAR LastCheckinDate = MAX(CHECKIN[check_in_date])
VAR DaysSinceLastCheckin = DATEDIFF(LastCheckinDate, TODAY(), DAY)
VAR AvgVisitFrequency =
    AVERAGEX(
        CHECKIN,
        DATEDIFF(CHECKIN[check_in_date], NEXT(CHECKIN[check_in_date], 1), DAY)
    )
VAR FrequencyScore =
    IF(
        DaysSinceLastCheckin > AvgVisitFrequency * 2,
        50,
        0
    )
VAR ContractEndingSoon =
    IF(
        DATEDIFF(TODAY(), SUBSCRIPTION[end_date], DAY) <= 30,
        30,
        0
    )
VAR RecentPTUsage =
    IF(
        CALCULATE(
            COUNTROWS(TRAINER),
            FILTER(
                TRAINER,
                TRAINER[activity_type] = "PT" &&
                TRAINER[punch_in] >= TODAY() - 30
            )
        ) = 0,
        20,
        0
    )
RETURN
    FrequencyScore + ContractEndingSoon + RecentPTUsage
```

What-If Parameter for Pricing Strategy (Bonus)

dax

```
// Create a What-If Parameter for Price Increase
Price Increase Parameter = 0.05 // Default 5%

// Projected Revenue with Price Increase
Projected Revenue =
[Total Revenue] * (1 + [Price Increase Parameter])

// Projected Churn with Price Increase
Projected Churn Rate =
[Monthly Churn Rate] * (1 + ([Price Increase Parameter] * 2))

// Projected Net Revenue Impact
Projected Net Revenue Impact =
[Projected Revenue] * (1 - [Projected Churn Rate]) -
```

```
[Total Revenue] * (1 - [Monthly Churn Rate])
```

Advanced Time Intelligence DAX Measures

dax

```
// Year to Date Revenue
YTD Revenue =
TOTALYTD(SUM(REVENUE[total_revenue]), Date[Date])

// Year to Date Revenue Previous Year
YTD Revenue PY =
TOTALYTD(SUM(REVENUE[total_revenue]), SAMEPERIODLASTYEAR(Date[Date]))

// YTD Revenue Growth %
YTD Revenue Growth % =
DIVIDE(
    [YTD Revenue] - [YTD Revenue PY],
    [YTD Revenue PY],
    0
)

// Quarter to Date Revenue
QTD Revenue =
TOTALQTD(SUM(REVENUE[total_revenue]), Date[Date])

// Quarter to Date Revenue Previous Year
QTD Revenue PY =
TOTALQTD(SUM(REVENUE[total_revenue]), SAMEPERIODLASTYEAR(Date[Date]))

// QTD Revenue Growth %
QTD Revenue Growth % =
DIVIDE(
    [QTD Revenue] - [QTD Revenue PY],
    [QTD Revenue PY],
    0
)

// Month to Date Revenue
MTD Revenue =
TOTALMTD(SUM(REVENUE[total_revenue]), Date[Date])

// Month to Date Revenue Previous Month
MTD Revenue PM =
TOTALMTD(SUM(REVENUE[total_revenue]), DATEADD(Date[Date], -1, MONTH))

// MTD Revenue Growth %
MTD Revenue Growth % =
DIVIDE(
    [MTD Revenue] - [MTD Revenue PM],
    [MTD Revenue PM],
    0
)
```

Advanced Segmentation DAX Measures

dax

```
// Member Segments for Cluster Analysis
Member Segment =
SWITCH(
    TRUE(),
    [Average Member Tenure] > 12 && [Monthly Active Members] = 1, "Loyal Active",
    [Average Member Tenure] > 12 && [Monthly Active Members] = 0, "Loyal At-Risk",
    [Average Member Tenure] <= 12 && [Monthly Active Members] = 1, "New Active",
    [Average Member Tenure] <= 12 && [Monthly Active Members] = 0, "New At-Risk",
    "Unknown"
)

// Gym Classification Based on Performance
Gym Classification =
SWITCH(
    TRUE(),
    [Gym Health Score] >= 85, "High Performing",
    [Gym Health Score] >= 70 && [Gym Health Score] < 85, "Average Performing",
    [Gym Health Score] < 70, "Under Performing",
    "Unclassified"
)

// Trainer Performance Category
Trainer Performance Category =
SWITCH(
    TRUE(),
    [Trainer Efficiency Score] >= 85, "Top Performer",
    [Trainer Efficiency Score] >= 70 && [Trainer Efficiency Score] < 85, "Average Performer",
    [Trainer Efficiency Score] < 70, "Under Performer",
    "Unclassified"
)
```

ROCE (Return on Capital Employed) Component Measures

dax

```
// Operational Profit (assuming 30% cost ratio)
Operational Profit =
[Total Revenue] * 0.7

// Capital Employed (mock data - typically would come from actual data)
Capital Employed =
VAR FixedAssetPerGym = 500000 // Example value
RETURN
    FixedAssetPerGym * DISTINCTCOUNT(GYM[gym_id])

// ROCE Calculation
ROCE =
DIVIDE(
    [Operational Profit],
    [Capital Employed],
    0
)

// ROCE Target
ROCE Target = 0.15 // 15% target return

// ROCE Achievement %
ROCE Achievement % =
DIVIDE(
    [ROCE],
    [ROCE Target],
    0
)
```

Advanced Forecasting Measures with Seasonality

dax

```
// Revenue Forecast with Seasonality
Revenue Forecast with Seasonality =
VAR BaselineForecast = [Revenue Forecast]
VAR CurrentMonth = MONTH(TODAY())
VAR SeasonalFactor =
    SWITCH(
        CurrentMonth,
        1, 0.9,  // January - post holiday slump
        2, 0.85, // February - lowest month
        3, 0.95, // March - slight improvement
        4, 1.0,  // April - average
        5, 1.05, // May - slight increase
        6, 1.1,  // June - summer increase
        7, 1.0,  // July - average
        8, 0.95, // August - slight decline
        9, 1.15, // September - back to school/work spike
        10, 1.1, // October - still strong
        11, 1.0, // November - average
        12, 0.9, // December - holiday season dip
        1.0     // Default
    )
RETURN
    BaselineForecast * SeasonalFactor

// Forecast Upper Bound (90% confidence)
Forecast Upper Bound = [Revenue Forecast with Seasonality] * 1.1

// Forecast Lower Bound (90% confidence)

Forecast Lower Bound = [Revenue Forecast with Seasonality] * 0.9
```

Membership Conversion and Retention Advanced Metrics

dax

```
// Trial to Paid Conversion Rate
Trial to Paid Conversion Rate =
DIVIDE(
    CALCULATE(
        COUNTROWS(SUBSCRIPTION),
        FILTER(
            SUBSCRIPTION,
            SUBSCRIPTION[subscription_plan] <> "Trial" &&
            SUBSCRIPTION[status] = "Active"
        )
    ),
    CALCULATE(
        COUNTROWS(SUBSCRIPTION),
        FILTER(
            SUBSCRIPTION,
            SUBSCRIPTION[subscription_plan] = "Trial"
        )
    ),
    0
)

// Membership Upgrade Rate
Membership Upgrade Rate =
DIVIDE(
    CALCULATE(
        COUNTROWS(SUBSCRIPTION),
        FILTER(
            SUBSCRIPTION,
            SUBSCRIPTION[subscription_plan] IN {"Premium", "Platinum"}
        )
    ),
    CALCULATE(
        COUNTROWS(SUBSCRIPTION),
        FILTER(
            SUBSCRIPTION,
            SUBSCRIPTION[subscription_plan] IN {"Basic", "Standard"}
        )
    ),
    0
)

// Member Retention Rate (inverse of churn)
Member Retention Rate = 1 - [Monthly Churn Rate]

// 3-Month Rolling Retention Rate
3-Month Rolling Retention Rate =
VAR Month1 =
    CALCULATE(
        [Member Retention Rate],
        DATEADD(Date[Date], -2, MONTH)
    )
VAR Month2 =
    CALCULATE(
        [Member Retention Rate],
        DATEADD(Date[Date], -1, MONTH)
    )
VAR Month3 = [Member Retention Rate]
RETURN
```

```
    (Month1 + Month2 + Month3) / 3
```

Marketing ROI Advanced Metrics

dax

```
// Marketing Cost per Lead Source (mock data)
Marketing Cost per Lead Source =
SWITCH(
    SELECTEDVALUE(LEADS[lead_source]),
    "Referral", 2000,
    "Social Media", 5000,
    "Website", 3500,
    "Walk-in", 1000,
    "Promotion", 7500,
    3000 // Default
)

// Return On Marketing Investment (ROMI)
Return On Marketing Investment =
DIVIDE(
    [PT Revenue] + [Subscription Revenue] - [Marketing Cost per Lead Source],
    [Marketing Cost per Lead Source],
    0
)

// Customer Acquisition Cost (CAC)
Customer Acquisition Cost =
DIVIDE(
    [Marketing Cost per Lead Source],
    [Converted Leads],
    0
)

// CAC Payback Period (months)
CAC Payback Period =
DIVIDE(
    [Customer Acquisition Cost],
    [Avg Monthly Revenue per Member],
    0

)
```

Gym Utilization and Capacity Metrics

dax

```
// Peak Hour Utilization % (assuming capacity data)
Peak Hour Utilization % =
DIVIDE(
    CALCULATE(
        COUNTROWS(CHECKIN),
        FILTER(
            CHECKIN,
            HOUR(CHECKIN[check_in_time]) = SELECTEDVALUE(Hour[hour_number])
        )
    ),
    GYM[max_capacity],
    0
)

// Average Daily Check-ins
Average Daily Check-ins =
AVERAGEX(
    VALUES(Date[Date]),
    CALCULATE(
        COUNTROWS(CHECKIN)
    )
)

// High Traffic Days
High Traffic Day =
IF(
    CALCULATE(
        COUNTROWS(CHECKIN)
    ) > [Average Daily Check-ins] * 1.2,
    "High Traffic",
    "Normal"
)

// Low Traffic Hours
Low Traffic Hour =
IF(
    [Peak Hour Utilization %] < 0.3,
    "Low Traffic",
    "Normal"
)
```

Power BI Dashboard Implementation - Page by Page Guide

1. Executive Summary Page

Layout: 2x3 grid with header containing date slicer and gym filter

Key Visuals:

1. **Gym Health Score Card**
   - Large number visual showing [Gym Health Score]
   - Trend line showing score over time
   - Conditional formatting: <70 (red), 70-85 (yellow), >85 (green)
2. **Cluster Performance Map**
   - Filled map showing gym locations
   - Size: [Active Members]
   - Color: [Gym Health Score] with gradient
   - Tooltip: Show [Gym Name], [Revenue per Gym], [Monthly Churn Rate]
3. **Revenue Metrics**
   - Card visuals showing:
     - [Total Revenue]
     - [Revenue YoY Growth]
     - [MTD Revenue]
     - [Revenue vs Target %]
   - Small sparklines showing trends
4. **Member Metrics**
   - Line chart showing [Active Members] and [Monthly Active Members] over time
   - Add [Net Member Growth] as stacked column
   - Include reference line for previous year
5. **Churn Analysis**
   - Gauge showing [Monthly Churn Rate]
   - Target line at industry benchmark (e.g., 5%)
   - Small trend chart showing churn over time
6. **Top/Bottom Performers**
   - Small table showing top 3 and bottom 3 gyms by [Gym Health Score]
   - Include columns for [Revenue per Gym], [Churn Rate], [Trainer Utilization Rate]
   - Conditional formatting on all metrics

2. Sales & PT Dashboard

Layout: Single page with synchronized slicers (date, gym location, trainer)

Key Visuals:

1. **Revenue Trend**
   - Line chart showing daily revenue for selected period
   - Include moving average line
   - Incorporate [Revenue Target] as reference line

2. **PT Revenue Analysis**
    - Bar chart showing PT Revenue by trainer
    - Sort by highest to lowest
    - Add data label for [Revenue per PT Hour]
    - Include average line for comparison
3. **PT Session Calendar**
    - Matrix visual with days on columns and trainers on rows
    - Values showing session count with color gradient
    - Include row and column totals
    - Conditional formatting to highlight low utilization days
4. **Revenue Breakdown**
    - Donut chart showing:
        - [Subscription Revenue]
        - [PT Revenue]
        - [Product Sales Revenue]
    - Include % of total and absolute values
5. **Revenue Forecast**
    - Line chart with [Total Revenue] historical data
    - Add [Revenue Forecast] line extending 3 months
    - Include confidence interval bands (±10%)
    - Mark key seasonal events
6. **Revenue KPIs with Time Intelligence**
    - Card visuals with comparisons:
        - MTD vs. Previous MTD
        - QTD vs. Previous QTD
        - YoY Growth
        - Rolling 3-Month

3. Retention & Churn Analysis

Layout: Single page with member segmentation focus

Key Visuals:

1. **Member Status Overview**
    - Donut chart showing [Active Members] vs [Inactive Members]
    - Include count and percentage
    - Tooltip showing trend over time
2. **Churn Rate Analysis**
    - Line chart showing [Monthly Churn Rate] over time
    - Add column chart with [Recent Churners] count
    - Include benchmark line for industry average
3. **Key Influencers for Churn**
    - Key Influencers visual analyzing what factors influence churn
    - Target: [Member Churn Count]
    - Explanatory factors: Membership length, visit frequency, subscription type, trainer usage
    - Top segments view to identify high-risk groups
4. **Member Segmentation**
    - Scatter chart plotting:
        - X-axis: [Average Member Tenure]
        - Y-axis: Average check-ins per month
        - Size: [Member Lifetime Value]
        - Color: [Member Frequency Category]
5. **Membership Length Distribution**
    - Histogram showing distribution of membership length
    - Use [Membership Length Category] for binning
    - Include average line
    - Highlight churn probability for each segment
6. **Reactivation Funnel**
    - Funnel visual showing:
        - Churned members
        - Contacted for reactivation
        - Expressed interest
        - [Reactivated Members]
    - Conversion rates between stages

4. Trainer Performance Page

Layout: Single page with trainer-focused metrics

Key Visuals:

1. **PT Sessions Delivered**
   - Bar chart showing sessions by trainer
   - Include target line from [Sessions Target per Trainer]
   - Color bars based on achievement percentage
   - Sort by performance
2. **Revenue Generation**
   - Clustered column chart with:
     - [PT Revenue] by trainer
     - [Revenue per Trainer]
     - [Avg Revenue per Session]
   - Sort by highest revenue
3. **Trainer Utilization Heatmap**
   - Matrix visual with:
     - Hours of day on rows
     - Days of week on columns
     - Color gradient based on session count or utilization %
     - Highlight peak and low-demand periods
4. **Idle Hours Analysis**
   - Line chart showing [Trainer Idle Hours] by day
   - Add [Idle Hours Cost] as secondary axis
   - Include average line
   - Annotate with scheduling recommendations
5. **Trainer Efficiency Scorecard**
   - Table showing all trainers with:
     - [Trainer Efficiency Score]
     - [PT Session Count]
     - [Revenue per PT Hour]
     - [Trainer Utilization Rate]
   - Conditional formatting on all metrics
   - Top 1 highlighted as [Top Performing Trainer]
6. **PT Conversion Rate**
   - Gauge showing percentage of free PT sessions converted to paid packages
   - Targets: Low (30%), Medium (50%), High (70%)
   - Small trend line showing historical performance

5. Marketing Funnel Page

Layout: Single page with lead source effectiveness focus

Key Visuals:

1. **Lead Source Breakdown**
   - Pie chart showing lead distribution by source
   - Include count and percentage
   - Color by [Lead Quality Score]
2. **Conversion Funnel**
   - Funnel visual showing:
     - [Total Leads]
     - [Contacted Leads]
     - [Converted Leads]
     - Active members (retained beyond trial)
   - Percentage drop between each stage
3. **Time to Conversion**
   - Column chart showing [Lead to Member Conversion Time] by lead source
   - Sort by shortest to longest
   - Include overall average line
   - Annotate with optimization recommendations
4. **Cost Per Acquisition**
   - Bar chart showing [Cost Per Acquisition] by lead source
   - Include data labels
   - Add reference line for target CPA
5. **Marketing ROI Analysis**
   - Scatter plot with:
     - X-axis: [Cost Per Acquisition]
     - Y-axis: [Lead Source ROI]
     - Size: Number of leads
     - Color: [Lead Closing Rate by Source]
   - Quadrants labeled for analysis (High ROI/Low Cost, etc.)
6. **Lead Performance Over Time**
   - Area chart showing lead volume by source over time
   - Stacked by lead status

    - Include annotations for marketing campaigns
    - Secondary axis with [Conversion Rate] line

Implementation Tips

For Drillthrough Pages:

1. **Member Detail Drillthrough**
    - Create a separate page with member-specific metrics
    - Enable drillthrough from Member visuals
    - Include:
        - Member profile summary
        - Visit frequency chart
        - Subscription history
        - PT session history
        - Churn risk indicators
        - Recommended actions
2. **Trainer Detail Drillthrough**
    - Create a separate page with trainer-specific metrics
    - Enable drillthrough from Trainer visuals
    - Include:
        - Trainer profile summary
        - Session calendar
        - Revenue trend
        - Client satisfaction metrics
        - Utilization rate by hour/day
        - Performance vs. targets

For Bookmarks:

1. **Financial View**
    - Focus on revenue metrics, ROCE, LTV
    - Hide operational metrics
    - Emphasize financial KPIs
2. **Operational View**
    - Focus on member activity, trainer utilization
    - Hide financial details
    - Emphasize operational KPIs

Synchronized Slicers:

1. Create a separate "Slicer" page with:
    - Date range slider
    - Gym location dropdown/checkbox
    - Trainer multi-select
    - Subscription type filter
2. Use "Sync Slicers" feature to apply these across all pages
3. Add "Reset Filters" button using bookmark functionality