NINJA DATA

POWER BI DESKTOP, DAX

DAX Cheat Sheet Mistrz

Zmienne, przejście kontekstu, relacje, rankingi, tabele w DAX i sprawdzanie miar od AI.

ŚCIĄGA | NOWICJUSZ | WOJOWNIK | MISTRZ

01 ZMIENNE
VAR i RETURN

- **VAR**: nazwany krok obliczenia. Liczy się raz i łatwiej znaleźć błąd.

```
Zmiana RR % =
VAR Teraz = [Sprzedaż]
VAR RokTemu = [Sprzedaż PY]
RETURN DIVIDE(Teraz - RokTemu, RokTemu)
```

- **Zmienna to stała**: liczy się tam, gdzie ją zapisano. CALCULATE jej nie przeliczy.

```
VAR Sp = [Sprzedaż]
RETURN CALCULATE(Sp, Klienci[Kraj] = "PL")
// wynik = Sp, filtr nic nie zmienia
```

- **Szukanie błędu**: w RETURN wpisz na chwilę zmienną zamiast wyniku i sprawdź jeden krok.

02 PRZEJŚCIE KONTEKSTU
Wiersz staje się filtrem

- Miara w kolumnie obliczanej albo w iteratorze dostaje bieżący wiersz jako filtr (ukryty CALCULATE).

```
// kolumna obliczana w tabeli Klienci:
Sprzedaż klienta = [Sprzedaż]
// to samo bez miary:
Sprzedaż klienta 2 =
    CALCULATE(SUM(Sprzedaz[Wartosc]))
```

- **Bez CALCULATE**: suma całej tabeli przy każdym kliencie.

```
Źle = SUM(Sprzedaz[Wartosc])
```

WSKAZÓWKA Przejście kontekstu na tabeli faktów jest wolne i może liczyć podwójnie przy powtórzonych wierszach. Iteruj po kolumnie z kluczem, np. VALUES(Klienci[KlientID]).

03 SZYBKOŚĆ
Lżejsze miary

- **Miarę użytą kilka razy zapisz w zmiennej**: liczy się raz.

```
// wolniej, [Sprzedaż] liczy się dwa razy:
IF([Sprzedaż] > 0, [Sprzedaż] * 1.23)
VAR S = [Sprzedaż]  // szybciej
RETURN IF(S > 0, S * 1.23)
```

- **Liczby jako miary, nie kolumny obliczane w tabeli faktów**: lżejszy model.

04 RELACJE
Dane z innej tabeli

- **RELATED**: wartość z tabeli po stronie „jeden” (kolumna obliczana albo SUMX).

```
Marża = SUMX(Sprzedaz,
    Sprzedaz[Ilosc] * RELATED(Produkty[Marza]))
```

- **RELATEDTABLE**: wiersze z tabeli po stronie „wiele”.

```
Zamówienia klienta =
    COUNTROWS(RELATEDTABLE(Sprzedaz))
```

- **USERELATIONSHIP**: włączasz nieaktywną relację, np. po dacie wysyłki.

```
Sprzedaż wg wysyłki = CALCULATE([Sprzedaż],
    USERELATIONSHIP(Sprzedaz[DataWysylki],
                    Kalendarz[Date]))
```

- **CROSSFILTER**: kierunek filtrowania tylko w jednej mierze, zamiast relacji dwukierunkowej w modelu.

```
Produkty z zakupem = CALCULATE(
    COUNTROWS(Produkty),
    CROSSFILTER(Sprzedaz[ProduktID],
                Produkty[ProduktID], Both))
```

- **TREATAS**: filtr z tabeli bez relacji, np. cele w innym podziale niż sprzedaż.

```
Cel = CALCULATE(SUM(Cele[Kwota]),
    TREATAS(VALUES(Produkty[Kategoria]),
            Cele[Kategoria]))
```

- **LOOKUPVALUE**: wartość z tabeli bez relacji po kilku kolumnach, jak WYSZUKAJ.PIONOWO.

```
Kurs = LOOKUPVALUE(Kursy[Kurs],
    Kursy[Waluta], Sprzedaz[Waluta],
    Kursy[Data], Sprzedaz[Data])
```

- **Model gwiazdy**: fakty w środku, wymiary dookoła, relacje jeden do wielu w jedną stronę. Na nim DAX jest najprostszy.

WSKAZÓWKA Relacje dwukierunkowe w modelu spowalniają raport i dają niejednoznaczne ścieżki. Kierunek zmieniaj w mierze, nie w modelu.

05 WYBÓR
Co jest zaznaczone

- **SELECTEDVALUE**: wartość wybrana we fragmentatorze, przy wielu wyborach zapasowa.

```
Tytuł = "Sprzedaż: " &
    SELECTEDVALUE(Klienci[Kraj], "wszystkie")
```

- **HASONEVALUE**: czy w kontekście jest dokładnie jedna wartość.

```
Cena = IF(HASONEVALUE(Produkty[Nazwa]),
    MAX(Produkty[Cena]))
```

- **ISINSCOPE**: który poziom hierarchii liczy macierz. Inna logika dla sumy i dla wierszy.

```
Udział w kategorii % =
IF(ISINSCOPE(Produkty[Nazwa]),
    DIVIDE([Sprzedaż], CALCULATE([Sprzedaż],
        REMOVEFILTERS(Produkty[Nazwa]))))
```

- **ISFILTERED**: czy ktoś filtruje kolumnę, np. komunikat w karcie.

06 RANKINGI
Najlepsze i najgorsze

- **RANKX**: miejsce w rankingu. ALL, żeby porównać ze wszystkimi.

```
Ranking = RANKX(ALL(Produkty[Nazwa]),
    [Sprzedaż], , DESC, DENSE)
```

- **TOPN**: N najlepszych wierszy jako tabela.

```
Top 5 = CALCULATE([Sprzedaż],
    TOPN(5, ALL(Produkty[Nazwa]), [Sprzedaż]))
```

07 SEGMENTY
Grupy z miar

- Ilu klientów przekroczyło próg w wybranym okresie.

```
Klienci VIP = COUNTROWS(
    FILTER(VALUES(Klienci[KlientID]),
        [Sprzedaż] >= 10000))
```

- **Nowi klienci**: pierwszy zakup w historii wypada w wybranym okresie.

```
Nowi klienci = COUNTROWS(FILTER(
    VALUES(Sprzedaz[KlientID]),
    CALCULATE(MIN(Sprzedaz[Data]),
        REMOVEFILTERS(Kalendarz))
      >= MIN(Kalendarz[Date])))
```

08 TABELE W DAX
Wynik jako tabela

- **SUMMARIZECOLUMNS**: grupowanie z miarami, jak GROUP BY w SQL.

```
EVALUATE
SUMMARIZECOLUMNS(
    Produkty[Kategoria],
    "Sprzedaż", [Sprzedaż])
```

- **ADDCOLUMNS**: dokładasz kolumny do tabeli, często pod iterator.

```
ADDCOLUMNS(VALUES(Klienci[KlientID]),
    "@Sp", [Sprzedaż])
```

- **Tabela obliczana**: każdy z powyższych wyników zapisany w modelu (Modelowanie, Nowa tabela).

09 SPRAWDZANIE
Debugowanie miar

- **DAX query view**: piszesz zapytanie i od razu widzisz tabelę wyniku. Suma w tabeli ma się zgadzać z totalem na wykresie.

```
EVALUATE
TOPN(10,
    SUMMARIZECOLUMNS(Produkty[Nazwa],
        "Sprzedaż", [Sprzedaż]),
    [Sprzedaż], DESC)
```

- **Performance Analyzer**: który wykres liczy się najdłużej i jaki DAX wysyła.

10 DAX OD AI
Zanim wstawisz do raportu

- Copilot albo ChatGPT napisze miarę w kilka sekund. Bez błędu składni nie znaczy, że liczy dobrze.
- Sprawdź na jednej liczbie, którą znasz, np. sprzedaż z jednego miesiąca z Excela.
- Zapytaj AI, jakie filtry zdejmuje i dlaczego. Gdy nie umie wyjaśnić, nie bierz.
- Podaj AI nazwy tabel, kolumn i relacji. Bez tego zgaduje nazwy.

WSKAZÓWKA Miara od AI z FILTER na całej tabeli faktów albo z IF w IF? Poproś o prostszą wersję z CALCULATE i zmiennymi.

Sprawdź, czy Twój model jest gotowy na takie miary.
Checklista modelu danych sklep.ninjadata.pl/checklista-semantic-model/