NINJA DATA

SKŁADNIA: SQL SERVER (T-SQL)

SQL Cheat Sheet Wojownik

Łączysz tabele, liczysz na datach i budujesz własne kategorie. Codzienność analityka.

ŚCIĄGA | NOWICJUSZ | **WOJOWNIK** | MISTRZ

01 JOIN — Złączenia tabel

INNER JOIN: tylko wiersze z dopasowaniem w obu tabelach.

```
SELECT k.Nazwa, z.Data
FROM Klienci k
INNER JOIN Zamowienia z ON z.KlientID = k.ID;
```

LEFT JOIN: wszystko z lewej tabeli, z prawej tylko dopasowane (reszta to NULL).

```
FROM Klienci k
LEFT JOIN Zamowienia z ON z.KlientID = k.ID
```

LEFT JOIN bez dopasowania: np. klienci bez żadnego zamówienia.

```
FROM Klienci k
LEFT JOIN Zamowienia z ON z.KlientID = k.ID
WHERE z.KlientID IS NULL
```

RIGHT JOIN: lustro LEFT JOIN. Zwykle czytelniej zamienić tabele i dać LEFT.

FULL OUTER JOIN: wszystko z obu tabel, braki uzupełnione NULL.

```
FROM Klienci k
FULL OUTER JOIN Zamowienia z ON z.KlientID = k.ID
```

FULL OUTER bez części wspólnej: wiersze, które są tylko po jednej stronie.

```
WHERE k.ID IS NULL OR z.KlientID IS NULL
```

- **CROSS JOIN**: każdy wiersz z każdym, np. wszystkie miesiące razy wszystkie sklepy.

```
SELECT m.Miesiac, s.Sklep
FROM Miesiace m CROSS JOIN Sklepy s;
```

WSKAZÓWKA Po JOIN policz wiersze. Jeśli jest ich więcej niż w tabeli głównej, klucz po drugiej stronie się powtarza i sumy będą zawyżone.

02 CASE — Własne kategorie

- **CASE WHEN**: if w SQL. Warunki sprawdzane po kolei, pierwszy prawdziwy wygrywa.

```
SELECT Nazwa,
  CASE
    WHEN Cena >= 500 THEN 'Premium'
    WHEN Cena >= 100 THEN 'Standard'
    ELSE 'Budżet'
  END AS Segment
FROM Produkty;
```

- **CASE w agregacji**: kilka liczników w jednym zapytaniu.

```
SELECT
  SUM(CASE WHEN Status = 'Zwrot'
           THEN 1 ELSE 0 END) AS Zwroty,
  COUNT(*) AS Wszystkie
FROM Zamowienia;
```

- **IIF**: krótki CASE z jednym warunkiem (tylko SQL Server).

```
SELECT IIF(Ilosc > 0, 'Na stanie', 'Brak')
FROM Magazyn;
```

03 PODZAPYTANIA — Zapytanie w zapytaniu

- **W WHERE**: porównanie z wynikiem innego zapytania.

```
SELECT Nazwa, Cena FROM Produkty
WHERE Cena > (SELECT AVG(Cena) FROM Produkty);
```

- **IN (SELECT ...)**: lista wartości z innej tabeli.

```
SELECT Nazwa FROM Produkty
WHERE ID IN (SELECT ProduktID FROM Zamowienia
             WHERE Kraj = 'Polska');
```

- **W FROM**: najpierw liczysz, potem filtrujesz wynik.

```
SELECT * FROM (
  SELECT KlientID, SUM(Wartosc) AS Suma
  FROM Zamowienia GROUP BY KlientID
) AS t
WHERE t.Suma > 1000;
```

04 DATY
Operacje na datach

- **GETDATE, CAST AS date**: teraz z godziną, sama dzisiejsza data.

```
SELECT GETDATE(), CAST(GETDATE() AS date);
```

- **YEAR, MONTH, DAY, DATEPART**: części daty.

```
SELECT YEAR(Data) AS Rok,
       DATEPART(QUARTER, Data) AS Kwartal
FROM Zamowienia;
```

- **DATEADD**: dodajesz albo odejmujesz dni, miesiące, lata.

```
SELECT DATEADD(DAY, 7, Data)    AS ZaTydzien,
       DATEADD(MONTH, -1, Data) AS MiesiacTemu
FROM Zamowienia;
```

- **DATEDIFF**: różnica między datami w wybranej jednostce.

```
SELECT DATEDIFF(DAY, DataZamowienia, DataDostawy)
       AS Dni
FROM Zamowienia;
```

- **DATEFROMPARTS, EOMONTH**: pierwszy i ostatni dzień miesiąca.

```
SELECT DATEFROMPARTS(YEAR(Data), MONTH(Data), 1)
         AS Poczatek,
       EOMONTH(Data) AS Koniec
FROM Zamowienia;
```

- **Ostatnie 30 dni**: filtr, który działa szybko na dużej tabeli.

```
WHERE Data >= DATEADD(DAY, -30,
                      CAST(GETDATE() AS date))
```

WSKAZÓWKA Zakres dat pisz jako `>= początek AND < dzień po końcu`. Przy kolumnach z godziną BETWEEN gubi ostatni dzień.

05 TYPY
Konwersje

- **CAST**: zmiana typu, standard SQL.

```
SELECT CAST(Cena AS DECIMAL(10, 2)) FROM Import;
```

- **TRY_CAST**: jak CAST, ale przy złej wartości daje NULL zamiast błędu.

```
SELECT TRY_CAST(Kwota AS DECIMAL(10, 2))
FROM Import;
```

- **CONVERT**: daty z tekstu i do tekstu ze stylem (103 = dd/mm/rrrr).

```
SELECT CONVERT(date, '19/10/2026', 103);
```

- **FORMAT**: data albo liczba jako tekst do raportu.

```
SELECT FORMAT(Data, 'yyyy-MM') AS Miesiac
FROM Zamowienia;
```

06 TEKST W GRUPACH
Sklejanie i dzielenie

- **STRING_AGG**: lista wartości z wielu wierszy w jednej komórce.

```
SELECT KlientID,
  STRING_AGG(Produkt, ', ') AS Produkty
FROM Zamowienia
GROUP BY KlientID;
```

- **STRING_SPLIT**: jeden tekst na wiele wierszy.

```
SELECT value
FROM STRING_SPLIT('SQL,DAX,Python', ',');
```

- **CHARINDEX**: pozycja znaku, np. wyciągasz domenę z maila.

```
SELECT SUBSTRING(Email,
         CHARINDEX('@', Email) + 1, 100)
FROM Klienci;
```

07 WIDOKI
Zapisane zapytania

- **CREATE VIEW**: zapytanie zapisane pod nazwą, używasz jak tabeli.

```
CREATE VIEW v_SprzedazMiesieczna AS
SELECT FORMAT(Data, 'yyyy-MM') AS Miesiac,
       SUM(Wartosc) AS Sprzedaz
FROM Zamowienia
GROUP BY FORMAT(Data, 'yyyy-MM');
```

Chcesz mieć te komendy w głowie, a nie na kartce? **Fiszki SQL Podstawy** sklep.ninjadata.pl/produkt/sql-podstawy/