NINJA DATA

SKŁADNIA: SQL SERVER (T-SQL)

SQL Cheat Sheet Nowicjusz

Pierwsze zapytania: wybierasz kolumny, filtrujesz wiersze, sortujesz i liczysz.

ŚCIĄGA | NOWICJUSZ | WOJOWNIK | MISTRZ

01 PIERWSZE KROKI
Wybieranie danych

- **SELECT ***: wszystkie kolumny z tabeli. Dobre na start, w raportach wybieraj konkretne kolumny.

```
SELECT * FROM Produkty;
```

- **SELECT kolumny**: tylko to, czego potrzebujesz.

```
SELECT Nazwa, Cena FROM Produkty;
```

- **AS**: alias, czyli nowa nazwa kolumny w wyniku.

```
SELECT Cena * 1.23 AS CenaBrutto FROM Produkty;
```

- **DISTINCT**: każda wartość tylko raz.

```
SELECT DISTINCT Kategoria FROM Produkty;
```

- **TOP**: pierwsze N wierszy (w MySQL i PostgreSQL na końcu `LIMIT 10`).

```
SELECT TOP 10 * FROM Produkty ORDER BY Cena DESC;
```

WSKAZÓWKA Baza czyta zapytanie w innej kolejności, niż je piszesz: FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY. Dlatego alias z SELECT zadziała w ORDER BY, ale nie w WHERE.

02 SZKIELET
Kolejność pisania

- **Zawsze w tej kolejności**: brakujące części pomijasz.

```
SELECT   Kategoria, COUNT(*) AS Ile
FROM     Produkty
WHERE    Cena > 0
GROUP BY Kategoria
HAVING   COUNT(*) > 5
ORDER BY Ile DESC;
```

- **Komentarze**: baza je pomija, Ty za miesiąc wiesz, co robi zapytanie.

```
-- komentarz w jednej linii
/* komentarz
   w kilku liniach */
```

03 WHERE
Filtrowanie wierszy

- **= <> > < >= <=**: porównania. Tekst i daty w apostrofach.

```
SELECT * FROM Produkty WHERE Cena > 50;
SELECT * FROM Produkty WHERE Kategoria <> 'Dom';
```

- **AND, OR**: łączysz warunki. Przy mieszaniu dodaj nawiasy.

```
WHERE Cena > 50
  AND (Kategoria = 'Moda' OR Kategoria = 'Dom')
```

- **IN, NOT IN**: wartość z listy.

```
WHERE Kategoria IN ('Elektronika', 'Moda')
```

- **BETWEEN**: zakres razem z końcami.

```
WHERE Cena BETWEEN 50 AND 100
```

- **LIKE**: wzorzec tekstu. % to dowolny ciąg, _ jeden znak.

```
WHERE Nazwa LIKE 'Laptop%'    -- zaczyna się od
WHERE Nazwa LIKE '%Pro'       -- kończy się na
```

- **IS NULL, IS NOT NULL**: puste pola. `= NULL` nigdy nie działa.

```
WHERE Opis IS NULL
```

04 ORDER BY
Sortowanie

- **ASC, DESC**: rosnąco (domyślnie) albo malejąco.

```
SELECT Nazwa, Cena FROM Produkty
ORDER BY Cena DESC;
```

- **Kilka kolumn**: najpierw po pierwszej, remisy po drugiej.

```
ORDER BY Kategoria, Cena DESC
```

05 AGREGACJE

Liczenie i grupowanie

- **COUNT(*)**: liczba wierszy.

```
SELECT COUNT(*) AS Ile FROM Zamowienia;
```

- **COUNT(kolumna)**: tylko niepuste wartości. `COUNT(DISTINCT kolumna)` liczy unikalne.

```
SELECT COUNT(DISTINCT KlientID) AS Klienci
FROM Zamowienia;
```

- **SUM, AVG, MIN, MAX**: suma, średnia, najmniejsza i największa wartość.

```
SELECT SUM(Wartosc), AVG(Wartosc), MAX(Wartosc)
FROM Zamowienia;
```

- **GROUP BY**: jeden wiersz na grupę. Każda kolumna w SELECT jest w GROUP BY albo w funkcji agregującej.

```
SELECT Kategoria, SUM(Sprzedaz) AS Suma
FROM Produkty
GROUP BY Kategoria;
```

- **HAVING**: filtr po zgrupowaniu (WHERE filtruje wiersze przed).

```
SELECT Kategoria, SUM(Sprzedaz) AS Suma
FROM Produkty
GROUP BY Kategoria
HAVING SUM(Sprzedaz) > 1000;
```

06 ZBIORY

Łączenie wyników

- **UNION ALL**: skleja wyniki dwóch zapytań jeden pod drugim, z duplikatami.

```
SELECT Nazwa FROM Produkty_A
UNION ALL
SELECT Nazwa FROM Produkty_B;
```

- **UNION**: to samo, ale bez duplikatów (wolniejsze, bo baza je szuka).
- **EXCEPT**: wiersze z pierwszego zapytania, których nie ma w drugim.
- **INTERSECT**: tylko wiersze, które są w obu.

WSKAZÓWKA Oba zapytania mają tyle samo kolumn, w tej samej kolejności i o zgodnych typach.

07 TEKST

Funkcje tekstowe

- **UPPER, LOWER**: wielkie albo małe litery.

```
SELECT UPPER(Nazwa) FROM Produkty;
```

- **LEN**: długość tekstu (MySQL, PostgreSQL: `LENGTH`).
- **LEFT, RIGHT, SUBSTRING**: kawałek tekstu.

```
SELECT LEFT(Kod, 3), SUBSTRING(Kod, 4, 2)
FROM Produkty;
```

- **TRIM, REPLACE**: usuwasz spacje z brzegów, podmieniasz fragment.

```
SELECT REPLACE(TRIM(Nazwa), 'stary', 'nowy')
FROM Produkty;
```

- **CONCAT**: sklejasz teksty, NULL traktuje jak pusty tekst.

```
SELECT CONCAT(Imie, ' ', Nazwisko) AS Klient
FROM Klienci;
```

08 LICZBY I NULL

Obliczenia

- **+ - * /**: działania na kolumnach.

```
SELECT Cena * Ilosc AS Wartosc FROM Zamowienia;
```

- **Dzielenie liczb całkowitych**: wynik też jest całkowity. Pomnóż przez 1.0.

```
SELECT 7 / 2;        -- 3
SELECT 7 * 1.0 / 2;  -- 3.5
```

- **ROUND, FLOOR, CEILING**: zaokrąglenie, w dół, w górę.

```
SELECT ROUND(Cena * 1.23, 2) AS Brutto
FROM Produkty;
```

- **COALESCE**: pierwsza niepusta wartość, czyli zamiennik dla NULL.

```
SELECT COALESCE(CenaPromo, Cena) AS CenaKoncowa
FROM Produkty;
```

- **NULLIF**: chroni przed dzieleniem przez zero.

```
SELECT Zysk / NULLIF(Sprzedaz, 0) AS Marza
FROM Wyniki;
```

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