



SQL Cheat Sheet Basic

sklep.ninjadata.pl

Podstawowe zapytania do tabel

Wybór wszystkich kolumn z tabeli
`SELECT * FROM Produkty;`

Wybór konkretnych kolumn z tabeli
`SELECT Nazwa, Cena FROM Produkty;`

Filtracja danych

Filtruje rekordy, gdzie "Cena" jest większa niż 50
`SELECT * FROM Produkty WHERE Cena > 50;`

Wybiera rekordy, których "Nazwa" zaczyna się od "Laptop"
`SELECT * FROM Produkty WHERE Nazwa LIKE 'Laptop%';`

Wybiera rekordy, których "Nazwa" kończy się na "Pro"
`SELECT * FROM Produkty WHERE Nazwa LIKE '%Pro';`

Filtruje rekordy, gdzie cena jest większa niż 50 oraz kategoria to "Elektronika"
`SELECT * FROM Produkty WHERE Cena > 50 AND Kategoria = 'Elektronika';`

Filtruje rekordy, gdzie cena jest większa niż 50 lub kategoria to "Moda"
`SELECT * FROM Produkty WHERE Cena > 50 OR Kategoria = 'Moda';`

Wybiera rekordy, które mają kategorię "Elektronika" lub "Moda"
`SELECT * FROM Produkty WHERE Kategoria IN ('Elektronika', 'Moda');`

Filtruje rekordy, które nie mają kategorii "Spożywcze" ani "Dom"
`SELECT * FROM Produkty WHERE Kategoria NOT IN ('Spożywcze', 'Dom');`

Filtruje kategorie produktów, w których łączna sprzedaż przekracza 1000 zł
`SELECT Kategoria, SUM(Sprzedaz) AS LacznaSprzedaz FROM Produkty GROUP BY Kategoria HAVING SUM(Sprzedaz) > 1000;`

Funkcje NULL

Jeśli kolumna CenaPromocyjna jest NULL, zwraca wartość z kolumny Cena
`SELECT COALESCE(CenaPromocyjna, Cena) AS CenaFinalna FROM Produkty;`

Jeśli Cena i CenaPromocyjna są równe, zwraca NULL, w przeciwnym razie zwraca Cena
`SELECT NULLIF(Cena, CenaPromocyjna) AS Cena FROM Produkty;`

Filtruje rekordy, gdzie Opis jest pusty.
`SELECT * FROM Produkty WHERE Opis IS NULL;`

Filtruje rekordy, gdzie Opis nie jest pusty.
`SELECT * FROM Produkty WHERE Opis IS NOT NULL;`

Sortowanie

Sortuje wyniki według kolumny "Cena" rosnąco (ASC)
`SELECT Nazwa, Cena FROM Produkty ORDER BY Cena ASC;`

Sortuje wyniki według kolumny "Cena" malejąco (DESC)
`SELECT Nazwa, Cena FROM Produkty ORDER BY Cena DESC;`

Funkcje tekstowe

Pobiera ostatnie 3 znaki z kolumny Nazwa
`SELECT RIGHT(Nazwa, 3) AS OstatnieZnaki FROM Produkty;`

Pobiera pierwsze 5 znaków z kolumny Nazwa
`SELECT LEFT(Nazwa, 5) AS PierwszeZnaki FROM Produkty;`

Konwertuje wartości w kolumnie Kategoria na małe litery
`SELECT LOWER(Kategoria) AS MaleLitery FROM Produkty;`

Konwertuje wartości w kolumnie Kategoria na wielkie litery
`SELECT UPPER(Kategoria) AS WielkieLitery FROM Produkty;`

Pobiera 5 znaków, zaczynając od 3 pozycji w kolumnie Nazwa
`SELECT SUBSTRING(Nazwa, 3, 5) AS Podciag FROM Produkty;`

Zastępuje wszystkie wystąpienia "stary" w kolumnie Opis na "nowy"
`SELECT REPLACE(Opis, 'stary', 'nowy') AS ZmienionyOpis FROM Produkty;`

Usuwa białe znaki z kolumny Nazwa
`SELECT TRIM(Nazwa) AS BezBialychZnakow FROM Produkty;`

Łączy Nazwa i Kategoria z separatorem "-"
`SELECT CONCAT(Nazwa, ' - ', Kategoria) AS PelnaNazwa FROM Produkty;`

Funkcje numeryczne

Zwraca sumę wartości w kolumnie "Cena"
`SELECT SUM(Cena) FROM Produkty;`

Zwraca średnią wartość w kolumnie "Cena"
`SELECT AVG(Cena) FROM Produkty;`

Liczy wszystkie wiersze w tabeli "Produkty"
`SELECT COUNT(*) FROM Produkty;`

Zlicza liczbę niepustych wartości w kolumnie 'Nazwa'
`SELECT COUNT(Nazwa) FROM Produkty;`

Zwraca minimalną wartość w kolumnie "Cena"
`SELECT MIN(Cena) FROM Produkty;`

Zwraca maksymalną wartość w kolumnie "Cena"
`SELECT MAX(Cena) FROM Produkty;`

WHERE

LIKE

AND

OR

IN

NOT IN

HAVING

COALESCE

NULLIF

IS NULL

IS NOT NULL

ORDER BY

ASC/DSC

RIGHT

LEFT

LOWER

UPPER

SUBSTRING

REPLACE

TRIM

CONCAT

SUM

AVG

COUNT

MIN

MAX

Operatory zbioru

Łączy listę produktów z dwóch tabel, zachowując wszystkie duplikaty
`SELECT Nazwa, Cena FROM Produkty_A UNION ALL SELECT Nazwa, Cena FROM Produkty_B;`

Łączy listę produktów z dwóch tabel, eliminując duplikaty
`SELECT Nazwa, Cena FROM Produkty_A UNION SELECT Nazwa, Cena FROM Produkty_B;`

Wybiera produkty, które są w tabeli Produkty_A, ale nie występują w tabeli Produkty_B
`SELECT Nazwa, Cena FROM Produkty_A EXCEPT SELECT Nazwa, Cena FROM Produkty_B;`

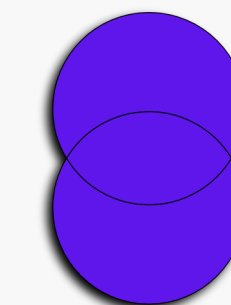
Wybiera produkty, które znajdują się zarówno w tabeli Produkty_A, jak i w tabeli Produkty_B
`SELECT Nazwa, Cena FROM Produkty_A INTERSECT SELECT Nazwa, Cena FROM Produkty_B;`

UNION ALL

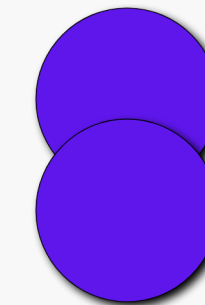
UNION

EXCEPT

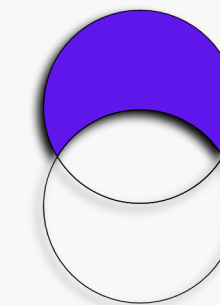
INTERSECT



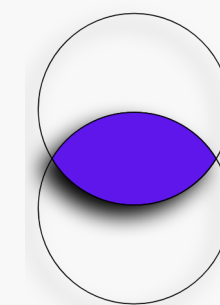
UNION ALL



UNION



EXCEPT



INTERSECT

Operacje na liczbach

Oblicza cenę po dodaniu podatku VAT (23%)
`SELECT Cena, Cena + (Cena * 0.23) AS CenaZVAT FROM Produkty;`

Oblicza cenę po odjęciu rabatu 10%
`SELECT Cena, Cena - (Cena * 0.10) AS CenaPoRabacie FROM Produkty;`

Oblicza całkowitą wartość zamówienia (Cena * Ilość)
`SELECT Produkt, Cena * Ilosc AS WartoscZamowienia FROM Zamowienia;`

Oblicza średnią cenę produktu w kategorii
`SELECT Kategoria, SUM(Cena) / COUNT(*) AS SredniaCena FROM Produkty GROUP BY Kategoria;`

Zaokrągla wartość zamówienia do dwóch miejsc po przecinku
`SELECT Produkt, ROUND(Cena * Ilosc, 2) AS WartoscZamowienia FROM Zamowienia;`

Zaokrągla koszt dostawy w dół
`SELECT Dostawca, FLOOR(KosztDostawy) AS KosztMinimalny FROM Dostawy;`

Zaokrągla koszt dostawy w górę
`SELECT Dostawca, CEIL(KosztDostawy) AS KosztMaksymalny FROM Dostawy;`

+

-

*

/

ROUND

FLOOR

CEIL