

Język SQL – instrukcje definiowania zapytań (DQL). **Złączenia tabel.**

Lp.	Instrukcja	Opis i przykład użycia
ZŁĄCZENIA PIONOWE (SUMA, RÓŻNICA, ILOCZYN - CZĘŚĆ WSPÓLNA)		
1.	zapytanie1 UNION zapytanie2	<p>Łączy wyniki zapytań usuwając duplikaty. Wykonuje operację SUMOWANIA ZBIORÓW.</p> <p>Wyświetl imiona, nazwiska i pesele pracowników obu uczelni (usuń ewentualne duplikaty): <i>SELECT imie, nazwisko, pesel FROM uczelniaA</i> <i>UNION</i> <i>SELECT imie, nazwisko, pesel FROM uczelniaB</i></p>
2.	zapytanie1 UNION ALL zapytanie2	<p>Łączy wyniki zapytań. NIE usuwa duplikatów. Wykonuje operację SUMOWANIA ZBIORÓW.</p> <p>Wyświetl imiona, nazwiska i pesele pracowników obu uczelni: <i>SELECT imie, nazwisko, pesel FROM uczelniaA</i> <i>UNION</i> <i>SELECT imie, nazwisko, pesel FROM uczelniaB</i></p>
3.	zapytanie1 EXCEPT zapytanie2	<p>Zwraca wiersze, które znajdują się w wynikach pierwszego zapytania i jednocześnie nie znajdują się w wynikach drugiego zapytania. Wykonuje operację ODEJMOWANIA ZBIORÓW.</p> <p>Wyświetl którzy pracownicy uczelni "uczelniaA" nie pracują jednocześnie na uczelni "uczelniaB": <i>SELECT imie, nazwisko, pesel FROM uczelniaA</i> <i>EXCEPT</i> <i>SELECT imie, nazwisko, pesel FROM uczelniaB</i></p>
4.	zapytanie1 INTERSECT zapytanie2	<p>Zwraca wiersze, które znajdują się jednocześnie w wynikach pierwszego i drugiego zapytania. Usuwa duplikaty. Wykonuje operację wyznaczania CZĘŚCI WSPÓLNEJ ZBIORÓW.</p> <p>Wyświetl którzy pracownicy pracują jednocześnie na uczelni "uczelniaA" i uczelni "uczelniaB": <i>SELECT imie, nazwisko, pesel FROM uczelniaA</i> <i>INTERSECT</i> <i>SELECT imie, nazwisko, pesel FROM uczelniaB</i></p>
5.	zapytanie1 INTERSECT ALL zapytanie2	<p>Zwraca wiersze, które znajdują się jednocześnie w wynikach pierwszego i drugiego zapytania. NIE usuwa duplikaty. Wykonuje operację wyznaczania CZĘŚCI WSPÓLNEJ ZBIORÓW.</p>

ZŁĄCZENIA POZIOME: WEWNĘTRZNE		
6.	SELECT kolumna1, kolumna2, ..., kolumnaN, FROM nazwa_tabeli1 [INNER] JOIN nazwa_tabeli2 ON (wyrażenie_warunkowe) [WHERE warunek]	<p>Pobiera tylko odpowiadające sobie wiersze (ze względu na warunek złączenia) z dwóch tabel.</p> <p>Wyświetl imiona, nazwiska, płace i nazwy zajmowanych stanowisk wszystkich pracowników: <i>SELECT imie, nazwisko, nazwa, placa</i> <i>FROM pracownicy INNER JOIN stanowiska</i> <i>ON (pracownicy.stanowisko_id = stanowiska.id)</i></p> <p>To samo co wyżej, ale z użyciem aliasów: <i>SELECT imie, nazwisko, nazwa, placa</i> <i>FROM pracownicy p INNER JOIN stanowiska s</i> <i>ON (p.stanowisko_id = s.id)</i></p>
ZŁĄCZENIA POZIOME: ZEWNĘTRZNE		
4.	SELECT kolumna1, kolumna2, ..., kolumnaN, FROM nazwa_tabeli1 LEFT [OUTER] JOIN nazwa_tabeli2 ON (wyrażenie warunkowe) [WHERE warunek]	<p>Pobiera odpowiadające sobie wiersze z dwóch tabel. Jeżeli w pierwszej tabeli znajdują się wiersze, które nie mają (ze względu na warunek złączenia) swoich odpowiedników w drugiej tabeli, i tak zostaną uwzględnione w złączeniu.</p> <p>Wyświetl dane pracowników (imiona, nazwiska, nazwy stanowisk, płace), tak aby w wyniku pojawiły się również pracownicy, którym przypisano błędny nr stanowiska: <i>SELECT imie, nazwisko, nazwa, placa</i> <i>FROM pracownicy LEFT JOIN stanowiska</i> <i>ON (pracownicy.stanowisko_id = stanowiska.id)</i></p>
5.	SELECT kolumna1, kolumna2, ..., kolumnaN, FROM nazwa_tabeli1 RIGHT [OUTER] JOIN nazwa_tabeli2 ON (wyrażenie warunkowe) [WHERE warunek]	<p>Pobiera odpowiadające sobie wiersze z dwóch tabel. Jeżeli w drugiej tabeli znajdują się wiersze, które nie mają (ze względu na warunek złączenia) swoich odpowiedników w pierwszej tabeli, i tak zostaną uwzględnione w złączeniu.</p> <p>Wyświetl dane pracowników (imiona, nazwiska, nazwy stanowisk, płace), tak aby w wyniku pojawiły się również stanowiska, które nie zostały przypisane żadnym pracownikom: <i>SELECT imie, nazwisko, nazwa, placa</i> <i>FROM pracownicy LEFT JOIN stanowiska</i> <i>ON (pracownicy.stanowisko_id = stanowiska.id)</i></p>
6.	SELECT kolumna1, kolumna2, ..., kolumnaN, FROM nazwa_tabeli1 FULL [OUTER] JOIN nazwa_tabeli2 ON (wyrażenie warunkowe) [WHERE warunek]	<p>Jest kombinacją LEFT OUTER JOIN i RIGHT OUTER JOIN. Uwzględnia ono w wynikach zapytania zarówno takie wiersze z pierwszej tabeli, które nie mają swoich odpowiedników w drugiej, jak i takie wiersze z drugiej tabeli, które nie mają swoich odpowiedników w pierwszej.</p>

ZŁĄCZENIA POZIOME: KRZYŻOWE (iloczyn kartezjański)		
7.	SELECT kolumna1, kolumna2, ..., kolumnaN, FROM nazwa_tabeli1 CROSS JOIN nazwa_tabeli2	Wykonuje iloczyn kartezjański łączenia tabel. Łączy każdy wiersz tabeli "nazwa_tabeli1" z każdym wierszem tabeli "nazwa_tabeli2"