### Typy złączenia tabel (join)

**Wprowadzenie**

**Złączenia** (join) są używane, gdy zachodzi konieczność uzyskania dostępu do więcej niż jednej tabeli lub więcej niż jednej krotki. Złączenia mogą być:

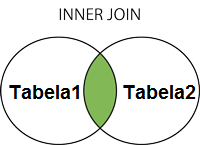
**wewnętrzne (INNER JOIN)** – tabela wynikowa będzie zawierać tylko takie elementy, dla których zostaną spełnione warunki złączenia wewnętrznego.

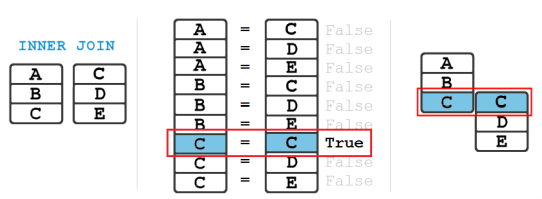
SELECT kolumna1, kolumna2, kolumnaN

FROM Tabela1

INNER JOIN Tabela2

ON Tabela1.kolumna1 = Tabela2.kolumna2;

Rys.1 Część wspólna z dwóch tabel

Rys.2 Wybierane są te rekordy, które mają odpowiedniki w obu tabelach

**zewnętrzne (OUTHER JOIN)** – zwracają również krotki niepasujące do warunku. Wyróżnia się trzy typy złączeń zewnętrznych:

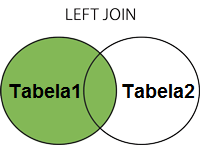
- **LEFT JOIN** (**LEFT OUTER JOIN) (lewostronne złączenie zewnętrzne)** – wynik złączenia zawiera wszystkie krotki pierwszej tabeli oraz wszystkie pasujące krotki drugiej tabeli.

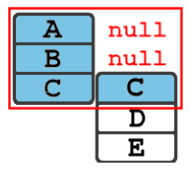
SELECT kolumna1, kolumna2, kolumnaN

FROM Tabela1

LEFT JOIN Tabela2

ON Tabela1.kolumna1 = Tabela2.kolumna2;

Rys.3 Wszystkie rekordy z pierwszej tabeli i tylko te rekordy z drugiej tabeli, które mają odpowiedniki w pierwszej tabeli

Rys.4 W wyniku LEFT JOIN zwracane są wszystkie rekordy z tabeli po lewej stronie (Tabela1) i wszystkie odpowiadające im rekordy z prawej tabeli (Tabela2). W rezultacie zwracana jest wartość nieokreślona NULL z prawej tabeli, jeśli nie ma odpowiednika w pierwszej tabeli.

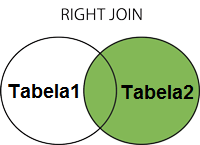
- **RIGHT JOIN** (**RIGHT OUTER JOIN) (prawostronne złączenie zewnętrzne)** – wynik złączenia zawiera wszystkie krotki drugiej tabeli oraz wszystkie pasujące krotki pierwszej tabeli

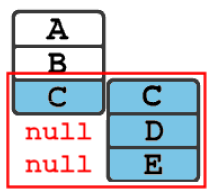
SELECT kolumna1, kolumna2, kolumnaN

FROM Tabela1

RIGHT JOIN Tabela2

ON Tabela1.kolumna1 = Tabela2.kolumna2;

Rys.5 Złączenie prawostronne RIGHT JOIN jest odwrotnością złączenia lewostronnego

Rys.6 Złączenie prawostronne RIGHT JOIN jest rzadziej używane niż lewostronne LEFT JOIN

- **FULL JOIN** (**FULL OUTER JOIN) (pełne złączenie zewnętrzne)** – wynik złączenia zawiera wszystkie krotki obu złączonych tabel (również te, które nie mają swoich odpowiedników)

SELECT kolumna1, kolumna2, kolumnaN

FROM Tabela1

FULL OUTER JOIN Tabela2

ON Tabela1.kolumna1 = Tabela2.kolumna2

WHERE warunki;

KlauzulaFULL OUTER JOINw dialekcie języka SQL jakim dysponuje system zarządzania bazami danych **MySQL** nie występuje, ale można próbować uzyskać identyczny efekt złączenia za pomocą następującej składni:

SELECT kolumna1, kolumna2, kolumnaN FROM Tabela1

LEFT JOIN Tabela2

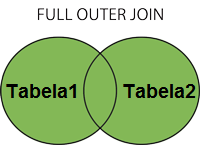
ON Tabela1.kolumna1 = Tabela2.kolumna2

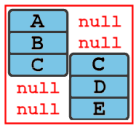
UNION

SELECT kolumna1, kolumna2, kolumnaN FROM Tabela1

RIGHT JOIN Tabela2

ON Tabela1.kolumna1 = Tabela2.kolumna2;

Rys.7 Złączenie zewnętrzne obustronne

Rys.8 Złączenie FULL JOIN zwraca wszystkie rekordy z lewej tabeli i wszystkie rekordy z prawej tabeli oraz te rekordy, które mają odpowiedniki w obu tabelach.

UWAGA: FULL JOIN może potencjalnie zwrócić bardzo duże zbiory danych!

**złączenie krzyżowe** (CROSS JOIN) – otrzymuje się kombinację każdej krotki pierwszej tabeli z każdą krotką drugiej tabeli (iloczyn kartezjański)

SELECT kolumna1, kolumna2 FROM Tabela1, Tabela2;

Wynik działania takiego złączenia w rzeczywistości jest bezużyteczny.

**złączenie w obrębie jednej tabeli**(SELF JOIN) – jest używane, gdy po obu stronach złączenia występuje ta sama tabela

SELECT kolumna1, kolumna2, kolumnaN

FROM Tabela1 T1, Tabela1 T2

WHERE warunek;

T1 i T2 są różnymi aliasami tej samej tabeli Tabela1

**Zadanie 1.**

Skorzystaj z bazy danych **szkola** i zbuduj zapytanie, w którym zostaną wyświetlone **nazwiska**, **imiona** uczniów z klasy 3ct. W edytorze tekstu wpisz odpowiednie informacje zgodnie z poniższą formatką. Uporządkuj rosnąco według nazwiska i imienia

|  |  |
| --- | --- |
| Użyte polecenie | SELECT nazwisko, imie, klasy.nazwa\_klasy FROM uczniowie INNER JOIN klasy ON uczniowie.id\_klasy = klasy.id\_klasy WHERE klasy.nazwa\_klasy = '3ct'; |
| Liczba rekordów spełniających warunek | 5 |
| Zrzut ekranu potwierdzający wykonanie zadania | |

**Zadanie 2.**

Skorzystaj z bazy danych **szkola** i zbuduj zapytanie, w którym zostaną wyświetlone nazwiska, imiona i nazwa klasy wszystkich uczniów, również tych uczniów, którzy przenieśli się z innej szkoły i nie zostali jeszcze przypisani do żadnej klasy. Uporządkuj rosnąco według nazwy klasy. W edytorze tekstu wpisz odpowiednie informacje zgodnie z poniższą formatką.

|  |  |
| --- | --- |
| Użyte polecenie | SELECT nazwisko, imie, klasy.nazwa\_klasy FROM uczniowie LEFT JOIN klasy ON uczniowie.id\_klasy=klasy.id\_klasy ORDER BY klasy.nazwa\_klasy; |
| Liczba rekordów spełniających warunek | 75 |
| Zrzut ekranu potwierdzający wykonanie zadania | |

**Zadanie 3.**

Skorzystaj z bazy danych **szkola** i zbuduj zapytanie, w którym zostaną wyświetlone: nazwa klasy, specjalizacja i liczba uczniów. Na wykazie mają znaleźć się też nowo utworzone klasy, w których jeszcze nie ma przypisanych uczniów. W edytorze tekstu wpisz odpowiednie informacje zgodnie z poniższą formatką.

|  |  |
| --- | --- |
| Użyte polecenie | SELECT nazwa\_klasy, specjalizacja, count(imie) FROM uczniowie RIGHT JOIN klasy ON uczniowie.id\_klasy=klasy.id\_klasy GROUP BY nazwa\_klasy; |
| Liczba rekordów spełniających warunek | 16 |
| Zrzut ekranu potwierdzający wykonanie zadania | |

**Zadanie 4.**

Skorzystaj z bazy danych **szkola** i zbuduj zapytanie, w którym zostanie wyświetlone: nazwisko, imię, nazwa\_klasy. Na wykazie maja znaleźć się również uczniowie, któzy nie zostali jeszcze przypisani do klas oraz klasy, w których nie ma jeszcze ani jednego ucznia.

W edytorze tekstu wpisz odpowiednie informacje zgodnie z poniższą formatką.

|  |  |
| --- | --- |
| Użyte polecenie | SELECT nazwisko, imie, klasy.nazwa\_klasy FROM uczniowie LEFT JOIN klasy ON uczniowie.id\_klasy=klasy.id\_klasy UNION SELECT nazwisko, imie, klasy.nazwa\_klasy FROM uczniowie RIGHT JOIN klasy ON uczniowie.id\_klasy=klasy.id\_klasy; |
| Liczba rekordów spełniających warunek | 77 |
| Zrzut ekranu potwierdzający wykonanie zadania | |

**Zadanie 5.**

Skorzystaj z bazy danych **szkola** i zbuduj zapytanie, w którym zostanie wyświetlone: nazwisko, imię, nazwa\_klasy. Wykonaj złączenie krzyżowe (CROSS JOIN) i zastanów się nad jego przydatnością.

W edytorze tekstu wpisz odpowiednie informacje zgodnie z poniższą formatką.

|  |  |
| --- | --- |
| Użyte polecenie | SELECT nazwisko, imie, nazwa\_klasy FROM uczniowie CROSS JOIN klasy ON uczniowie.id\_klasy=klasy.id\_klasy; |
| Liczba rekordów spełniających warunek | 72 |
| Zrzut ekranu potwierdzający wykonanie zadania | |

**Zadanie 6.**

Skorzystaj z bazy danych **szkola** i zbuduj zapytanie, w którym zostanie wyświetlone: nazwisko1, nazwisko2, miejscowość uczniów, którzy są z tych samych miejscowości. Uporządkuj według miejscowości. Spróbuj użyć złączenia SELF JOIN

W edytorze tekstu wpisz odpowiednie informacje zgodnie z poniższą formatką.

|  |  |
| --- | --- |
| Użyte polecenie | SELECT A.nazwisko AS nazwisko1, B.nazwisko AS nazwisko2, A.miejscowosc FROM uczniowie A, uczniowie B WHERE A.id\_ucznia <> B.id\_ucznia AND A.miejscowosc = B.miejscowosc ORDER BY A.miejscowosc; |
| Liczba rekordów spełniających warunek | 724 |
| Zrzut ekranu potwierdzający wykonanie zadania | |