

Łączenie wyników zapytań

Język SQL zawiera mechanizmy pozwalające na łączenie wyników kilku zapytań. Czynność taka może zostać wykonana za pomocą instrukcji:

- UNION
- INTERSECT (nie jest obsługiwana przez bazę MySql)
- EXCEPT (nie jest obsługiwana przez bazę MySql)

Wszystkie mają podobną postać:

Zapytanie1 UNION | INTERSECT | EXCEPT zapytanie 2 [...UNION | INTERSECT | EXCEPT zapytanie N]

Ćwiczenie1.

Zaprojektuj dwie tabele, które będą zawierały listę pracowników dwóch uczelni:

Tabela 1 ma nazywać się *UczelniaA*

Tabela 2 ma nazywać się *UczelniaB*

Obie tabele będą zawierały następujące kolumny:

- Id
- Imię
- Nazwisko
- Pesel (typ char 11)

Do obu tabel należy wprowadzić po 3 rekordy, takie jak poniżej:

```
insert into uczelniaA values (1, 'Jan', 'Kowalski', '01234567890');  
insert into uczelniaA values (2, 'Adam', 'Nowak', '12345678901');  
insert into uczelniaA values (3, 'Anna', 'Wilk', '23456789012');
```

```
insert into uczelniaB values (1, 'Janina', 'Wolska', '34567890123');  
insert into uczelniaB values (2, 'Beata', 'Nowak', '45678901234');  
insert into uczelniaA values (1, 'Jan', 'Kowalski', '01234567890');
```

1. Instrukcja UNION

Zapytanie1 UNION [ALL] zapytanie 2 [...UNION zapytanie N]

Łączy wyniki zapytań *zapytanie1* i *zapytanie2*. Oba zapytania powinny zwracać taką samą liczbę kolumn o takich samych typach. Jeżeli słowo ALL zostanie pominięte, z wyniku zostaną usunięte duplikaty danych, czyli instrukcja zachowa się tak, jakby zostało użyte słowo DISTINCT (**SELECT słowo DISTINCT** Instrukcja eliminuje duplikaty (powtórzone dane) z

wyników zapytania). Jeśli słowo ALL zostanie użyte, w wynikach zostaną uwzględnione wszystkie wiersze z obu zapytań.

Przykład:

Wykonaj zapytanie, które wyświetli dane (imię, nazwisko i PESEL) wszystkich pracowników uczelni UczelniaA i UczelniaB.

```
select imie, nazwisko, pesel from uczelniaa  
union  
select imie, nazwisko, pesel from uczelniab;
```

Dodaj w odpowiednie miejsce w w/w instrukcji słowo ALL i sprawdź co się zmieni.

2. Instrukcja INTERSECT

Zapytanie1 INTERSECT [ALL] zapytanie 2 [...UNION zapytanie N]

Działa podobnie jak instrukcja UNION ale znaczenie jej jest inne. Jej wynikiem są takie wiersze, które występują jednocześnie w *zapytanie1* i *zapytanie2*.

3. Instrukcja EXCEPT

Zapytanie1 EXCEPT [ALL] zapytanie 2 [...UNION zapytanie N]

W wyniku działania tej funkcji zwrócone zostaną te wiersze, które znajdują się w wynikach pierwszego zapytania i jednocześnie nie znajdują się w wynikach drugiego zapytania. Możemy więc na przykład dowiedzieć się, którzy z pracowników uczelni A nie pracują jednocześnie na uczelni B.

Złączenia

Jeżeli pobieramy dane z więcej niż jednej tabeli, w rzeczywistości wykonujemy tak zwane złączenie. W SQL istnieją instrukcje pozwalające na formalne wykonanie złączenia tabel, co więcej – istnieje wiele typów złączeń. Najbardziej znane to:

- CROSS JOIN
- INNER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN (*nie jest obsługiwane przez bazę MySQL*)

1. Złączenie CROSS JOIN - złączenie krzyżowe.

Postać instrukcji:

SELECT kolumna1, kolumna2...kolumnaN **FROM** tabela1 **CROSS JOIN** tabela2

Jest to iloczyn kartezjański łączonych tabel, czyli łączy każdy wiersz tabeli *tabela1* z każdym wierszem tabeli *tabela2*. Jest to odpowiednik instrukcji:

SELECT kolumna1, kolumna2...kolumnaN **FROM** tabela1, tabela2

Przykład:

Złączenie krzyżowe tabel pracownicy, stanowiska.

SELECT* FROM pracownicy CROSS JOIN stanowiska;

2. Złączenie INNER JOIN – złączenie warunkowe.

Postać instrukcji:

SELECT kolumna1, kolumna2...kolumnaN **FROM** tabela1 **[INNER] JOIN** tabela2
ON wyrażenie_warunkowe.

W przypadku tego złączenia pojawiają się tylko te wiersze z tabel *tabela1* i *tabela2*, które spełniają warunki wymienione po klauzuli **ON**. Jest to więc odpowiednik instrukcji:

SELECT kolumna1, kolumna2...kolumnaN **FROM** tabela1, tabela2..tabelaN
WHERE warunki

Przykład:

Wykonaj złączenie **INNER JOIN** pobierające dane z kolumn: imię, nazwisko, placą z tabeli *pracownicy* i nazwa z tabeli *stanowiska*, takie, aby każdemu pracownikowi było przypisane odpowiadające mu stanowisko.

**select imię, nazwisko, nazwa, placą from pracownicy inner join stanowiska on
pracownicy.stanowisko_id=stanowiska.id_stanowiska;**

3. Złączenie LEFT OUTER JOIN

Postać instrukcji:

SELECT kolumna1, kolumna2...kolumnaN **FROM** tabela1 **LEFT [OUTER] JOIN**
tabela2 **ON** wyrażenie_warunkowe.

Złączenie to pozwala na uwzględnienie w wyniku danych, które nie mają swoich odpowiedników w złączanych tabelach. Oznacza to, że 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, a w miejsce pustych kolumn zostaną wstawione wartości NULL.

Dodaj do tabeli pracownicy rekord:

insert into pracownicy values (default, 'Robert', 'Nowicki', 1600,5);

Dane w rekordzie odwołują się do nr stanowiska, którego nie ma w tabeli stanowiska. W takiej sytuacji zapytanie typu **INNER JOIN** nie ujawni istnienia osoby o identyfikatorze 5. Inaczej będzie w przypadku polecenia **LEFT OUTER JOIN**.

Przykład:

Wyświetl dane pracowników wraz zajmowanymi przez nich stanowiskami, tak aby w wyniku pojawiły się również osoby, którym przypisano błędny nr stanowiska.

**select imie, nazwisko, nazwa from pracownicy left outer join stanowiska on
pracownicy.stanowisko_id=stanowiska.id_stanowiska;**

W pole nazwa stanowiska powinna pojawić się wartość NULL zamiast nazwy. W ten sposób można otrzymać dane wszystkich pracowników nawet jeśli są niespójne.

4. Złączenie RIGHT OUTER JOIN

Postać instrukcji:

SELECT kolumna1, kolumna2...kolumnaN **FROM** tabela1 **RIGHT [OUTER]**
JOIN tabela2 **ON** wyrażenie_warunkowe.

Złączenie to jest przeciwieństwem **LEFT OUTER JOIN**. W tabeli wynikowej uwzględnia ono wiersze z drugiej tabeli, które nie mają swoich odpowiedników w pierwszej.

Dodaj do tabeli stanowiska rekord:

insert into stanowiska values (6, 'informatyk');

Użyj zapytania typu **RIGHT OUTER JOIN** do złączenia tabel pracownicy i stanowiska.

**select imie, nazwisko, nazwa from pracownicy right outer join stanowiska on
pracownicy.stanowisko_id=stanowiska.id_stanowiska;**

Pola imie, nazwisko powinny mieć wpisaną wartość NULL.

Złączenie **RIGHT OUTER JOIN** można zastąpić złączeniem **LEFT OUTER JOIN** i odwrotnie. Wystarczy zamienić kolejność tabel, aby uzyskać zapytanie pierwszego lub drugiego typu.

Sprawdź jak to działa.

5. Złączenie FULL OUTER JOIN

Jest kombinacją **RIGHT OUTER JOIN** i **LEFT OUTER JOIN**. Uwzględnia 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.

6. Złączenia i klauzula WHERE

W złączeniach można również stosować klauzulę WHERE. W wyniku takiego zapytania najpierw zostanie wykonane złączenie spełniające warunki występujące po klauzuli ON, następnie z otrzymanych wierszy zostaną wyeliminowane te, które nie spełniają warunków występujących po klauzuli WHERE.

Przykład:

Zmodyfikuj zapytanie z przykładu w pkt.3 w taki sposób, aby znaleźć pracowników, dla których brakuje odpowiadającego wpisu w tabeli stanowiska.

```
select imie, nazwisko, nazwa from pracownicy left outer join stanowiska on  
pracownicy.stanowisko_id=stanowiska.id_stanowiska where nazwa is NULL;
```

To samo polecenie wykonaj dla przykładu z pkt. 4. Ma wyświetlić pola imie i nazwisko, które zawierają wartości puste.

Zadania do samodzielnego wykonania:

1. Na podstawie tabeli OCENY oraz tabeli UCZNIOWIE wyświetl alfabetyczną listę uczniów, którzy uzyskali ocenę 2. (Użyj odpowiedniego złączenia)
2. Utwórz wykorzystując odpowiednie polecenie listę wszystkich nauczycieli i uczniów w szkole.