

Temat: Złączenia. (baza nazwisko_firma)

Jeśli pobieramy dane z więcej niż jednej tabeli, w rzeczywistości wykonujemy tak zwane złączenie. W SQL istnieją instrukcje pozwalające na formalne wykonanie złączenia tabel - istnieje wiele typów złączeń.

Zapoznaj się z treścią wykładu 3.21_SQL_Zlaczzenia.pdf!

1. Złączenie typu CROSS JOIN – to złączenie krzyżowe, którego ogólna postać jest następująca:

```
SELECT kolumna1, kolumna2, ..., kolumnaN  
FROM tabela1 CROSS JOIN tabela2;
```

Wykonuje ono iloczyn kartezjański łączonych tabel, czyli łączy każdy wiersz *tabeli1* z każdym wierszem *tabeli2*. Jest to zatem odpowiednik instrukcji:

```
SELECT kolumna1, kolumna2, ..., kolumnaN  
FROM tabela1, tabela2;
```

Ćw.1

Wykonaj złączenie krzyżowe tabel *pracownicy* i *stanowiska*.

```
SELECT *  
FROM pracownicy CROSS JOIN stanowiska;
```

Ćw.1.1

Wykonaj złączenie krzyżowe tabel *stanowiska* i *pracownicy*. Porównaj wyniki obu zapytań.

```
SELECT *  
FROM stanowiska CROSS JOIN pracownicy;
```

Ćw.1.2

Wykonaj złączenie krzyżowe tabel *pracownicy* i *stanowiska*.

Wybierz tylko pola: *imie*, *nazwisko*, *nazwa* i *pensja*.

```
SELECT imie, nazwisko, nazwa, pensja  
FROM pracownicy CROSS JOIN stanowiska;
```

Wynik tego złączenia przedstawia rysunek obok.

imie	nazwisko	nazwa	pensja
Anna	Nowak	kierownik	4000.00
Anna	Nowak	asystent	3200.00
Anna	Nowak	sekreterka	2500.00
Anna	Nowak	pracownik	2800.00
Anna	Nowak	kierowca	2700.00
Jan	Kowalski	kierownik	4000.00
Jan	Kowalski	asystent	3200.00
Jan	Kowalski	sekreterka	2500.00
Jan	Kowalski	pracownik	2800.00
Jan	Kowalski	kierowca	2700.00
Kaja	Borecka	kierownik	4000.00
Kaja	Borecka	asystent	3200.00
Kaja	Borecka	sekreterka	2500.00
Kaja	Borecka	pracownik	2800.00
Kaja	Borecka	kierowca	2700.00
Maria	Borek	kierownik	4000.00
Maria	Borek	asystent	3200.00
Maria	Borek	sekreterka	2500.00
Maria	Borek	pracownik	2800.00
Maria	Borek	kierowca	2700.00

Ćw.1.3

Wykonaj złączenie krzyżowe tabel *stanowiska* i *pracownicy*.

Wybierz tylko pola: *nazwa*, *pensja*, *imie*, *nazwisko*.

```
SELECT imie, nazwisko, nazwa, pensja  
FROM stanowiska CROSS JOIN pracownicy;
```

2. Złączenie typu INNER JOIN – to złączenie warunkowe o postaci:

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

Słowo INNER w większości implementacji jest opcjonalne. W przypadku tego złączenia wyniku pojawią się tylko te wiersze z *tabeli1* i *tabeli2*, które spełniają warunki wymienione po klauzuli ON. Jest to więc odpowiednik instrukcji:

```
SELECT kolumna1, kolumna2, ..., kolumnaN
```

FROM *tabela1*, *tabela2*, ..., *tabelaN*
WHERE *warunki*;

Ćw.2

Wykonaj złączenie typu INNER JOIN pobierające dane z kolumn: imię, nazwisko, stanowisko i pensja z tabel *pracownicy* i *stanowiska* takie, aby każdemu pracownikowi była przypisana odpowiadająca mu płaca.

SELECT *imie, nazwisko, nazwa, pensja*
FROM *pracownicy* INNER JOIN *stanowiska*
ON *stanowisko=stanowiska.id_stanowiska*;

imie	nazwisko	nazwa	pensja
Anna	Nowak	sekretarka	2500.00
Jan	Kowalski	kierownik	4000.00
Kaja	Borecka	asystent	3200.00
Maria	Borek	pracownik	2800.00
Marian	Marczak	pracownik	2800.00
Kajetan	Grota	kierowca	2700.00
Maria	Grota	asystent	3200.00

3. **Złączenie typu LEFT OUTER JOIN** – pozwala na uwzględnienie w wyniku danych, które nie mają swoich odpowiedników w złączonych tabelach. Oznacza to, że jeżeli w pierwszej tabeli znajdują się wiersze, które nie mają swoich odpowiedników w drugiej tabeli, i tak zostaną uwzględnione w złączeniu, a w miejsce pustych kolumn zostaną wstawione wartości NULL. Konstrukcja ta ma postać:

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

Słowo OUTER w większości implementacji jest opcjonalne. Aby sprawdzić, jak działa w praktyce, dodaj do tabeli *pracownicy* nowy wiersz:

INSERT INTO *pracownicy*
VALUES (8, 'Robert', 'Nowicki', '1985-11-01', '85110123456', 'Gdansk', 8);

Osobie tej został przypisany identyfikator nieistniejącego stanowiska (8). Jest to błąd polegający na niespójności danych w bazie, ale taka sytuacja może się zdarzyć w rzeczywistości. W takiej sytuacji złączenie typu INNER JOIN nie ujawni osoby o numerze 8, ponieważ w tabeli *stanowiska* nie ma stanowiska, któremu przypisano identyfikator 8. Inaczej będzie w przypadku złączenia LEFT INNER JOIN.

Ćw.3

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

SELECT *imie, nazwisko, nazwa, pensja*
FROM *pracownicy* LEFT OUTER JOIN *stanowiska*
ON *stanowisko=stanowiska.id_stanowiska*;

imie	nazwisko	nazwa	pensja
Anna	Nowak	sekretarka	2500.00
Jan	Kowalski	kierownik	4000.00
Kaja	Borecka	asystent	3200.00
Maria	Borek	pracownik	2800.00
Marian	Marczak	pracownik	2800.00
Kajetan	Grota	kierowca	2700.00
Maria	Grota	asystent	3200.00
Robert	Nowicki	NULL	NULL

W wyniku pojawiły się dane wszystkich pracowników. W kolumnach dotyczących stanowiska dla których nie ma odpowiedników w tabeli *stanowiska*, pojawiła się wartość NULL. W ten sposób otrzymaliśmy dane wszystkich pracowników, nawet jeśli były niespójne.

4. **Złączenie typu RIGHT OUTER JOIN** – jest przeciwieństwem LEFT OUTER JOIN. W tabeli wynikowej uwzględnia ono wiersze z drugiej tabeli, które nie mają swoich odpowiedników w pierwszej tabeli. Ma ono postać:

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

Słowo OUTER, tak jak w poprzednio opisywanych przypadkach, zazwyczaj jest opcjonalne. Aby sprawdzić działanie tego typu złączenia, dodaj do tabeli *stanowiska* stanowisko, do którego nie będzie przypisany żaden pracownik.

INSERT INTO *stanowiska* VALUES(6, 'informatyk', 4000.00);

Ćw.4

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

SELECT *imie, nazwisko, nazwa, pensja*
FROM *pracownicy* RIGHT OUTER JOIN *stanowiska*
ON *stanowisko=stanowiska.id_stanowiska*;

imie	nazwisko	nazwa	pensja
Jan	Kowalski	kierownik	4000.00
Kaja	Borecka	asystent	3200.00
Maria	Grota	asystent	3200.00
Anna	Nowak	sekretarka	2500.00
Maria	Borek	pracownik	2800.00
Marian	Marczak	pracownik	2800.00
Kajetan	Grota	kierowca	2700.00
NULL	NULL	informatyk	4000.00

Łatwo zauważyć, że RIGHT OUTER JOIN i LEFT OUTER JOIN to funkcjonalne odpowiedniki. Wystarczy zmienić kolejność tabel, aby uzyskać zapytanie pierwszego bądź drugiego typu.

Ćw.4.1

Użyj złączenia typu LEFT OUTER JOIN do uzyskania rezultatów takich jak w poprzednim ćwiczeniu.

5. **Złączenie typu FULL OUTER JOIN** – jest kombinacją LEFT OUTER JOIN i RIGHT OUTER JOIN. Nie jest ono obsługiwane przez bazę MYSQL. Ma ogólną postać:

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

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.

Ćw.5

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

SELECT *imie, nazwisko, pesel, nazwa, pensja*
FROM *pracownicy* FULL OUTER JOIN *stanowiska*
ON *stanowisko=stanowiska.id_stanowiska*;

6. Złączenia i klauzula WHERE

W złączeniach można również stosować klauzule 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, a serwer bazy danych zwróci ostateczny wynik.

Ćw.6

Napisz zapytanie wykorzystujące złączenie LEFT OUTER JOIN do wyświetlenia danych pracowników, dla których w kolumnie *stanowisko* znajdują się wartości niemające odpowiedników w tabeli *stanowiska*.

SELECT *imie, nazwisko, pesel, nazwa, pensja*
FROM *pracownicy* LEFT OUTER JOIN *stanowiska*
ON *stanowisko=stanowiska.id_stanowiska*
WHERE *nazwa IS NULL*;

imie	nazwisko	nazwa	pensja
Robert	Nowicki	NULL	NULL

Efekt działania tego zapytania przedstawiony jest na rysunku powyżej.

Ćw.6.1

Napisz zapytanie wykorzystujące dowolne złączenie do wyświetlenia nazw stanowisk, dla których w kolumnie *nazwisko* znajdują się wartości niemające odpowiedników w tabeli *stanowiska*.

7. Zrzuty ćwiczeń od 1 do 6.1 wraz z drugą częścią ćwiczeń prześlij przez zakładkę Zadania.

Temat: Operacje pionowe na zbiorach. Łączenie wyników zapytań.

Drugą grupą operacji na zbiorach, są operacje **pionowe**, czyli :

- UNION – suma (bez powtórzeń).
- UNION ALL – suma (z powtórzeniami).
- EXCEPT – odejmowanie zbiorów.
- INTERSECT – iloczyn (część wspólna).

Warunkiem podstawowym, któregośkolwiek ze sposobów operowania na zbiorach w sposób pionowy, jest podobna struktura tabel wejściowych. **Liczba kolumn w każdym zbiorze (kwerendzie), musi być identyczna oraz typy danych poszczególnych kolumn, muszą do siebie pasować.** Nazwy kolumn, nie mają znaczenia. W zbiorze wynikowym, atrybuty będą nazwane tak jak w pierwszej z kwerend.

1. W bazie danych **zapytania_nazwisko** przygotuj dwie nowe tabele. Będą one zawierały listę pracowników dwóch uczelni: UczelniaA i UczelniaB.

Uczelnia A

id	imie	nazwisko	pesel
1	Jan	Kowalski	01234567890
2	Adam	Nowak	12345678901
3	Anna	Wilk	23456789012

Uczelnia B

id	imie	nazwisko	pesel
1	Olga	Wolska	34567890123
2	Beata	Nowak	45678901234
3	Jan	Kowalski	01234567890

2. 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;
```

Dane Jana Kowalskiego, będącego pracownikiem obu uczelni, pojawiły się w wynikach zapytania tylko raz.

3. Użyj instrukcji UNION wyświetlającej dane wszystkich pracowników w taki sposób, aby w wynikach znalazły się wszystkie wiersze zapytań składowych.

```
SELECT imie, nazwisko, pesel FROM UczelniaA
UNION ALL
SELECT imie, nazwisko, pesel FROM UczelniaB;
```

4. Napisz zapytanie stwierdzające, którzy pracownicy pracują zarówno na UczelniA jak i na UczelniB.

```
SELECT imie, nazwisko, pesel FROM UczelniaA
INTERSECT
SELECT imie, nazwisko, pesel FROM UczelniaB;
```

(instrukcja nie jest obsługiwana przez bazę MySQL)

5. Napisz zapytanie pozwalające stwierdzić, którzy z pracowników UczelniA nie pracują jednocześnie na UczelniB.

```
SELECT imie, nazwisko, pesel FROM UczelniaA
EXCEPT
SELECT imie, nazwisko, pesel FROM UczelniaB;
```

(instrukcja nie jest obsługiwana przez bazy MySQL i ORACLE)

imie	nazwisko	nazwa	pensja
NULL	NULL	informatyk	4000.00

6. Z tabeli pracownicy wyświetl listę osób, których imiona zaczynają się na literę A i którzy zarabiają powyżej 2000 zł.
7. Zrzuty wszystkich ćwiczeń z tego PDF-a prześlij przez zakładkę Zadania.