

Przedmiot: **Pracownia baz danych**

Klasa: **3TI**

Temat: **Ćwiczenia powtórzeniowe**

Czas przewidziany na realizację: **45 minut** [2019.02.18]

Zakres: **Łączenie tabel, łączenie wyników zapytań [instrukcje UNION, INTERSECT, EXCEPT]**

Krótką powtórka z teorii

* Złączenia tabel *

Złączenia (JOIN) są używane, gdy zachodzi konieczność uzyskania dostępu do więcej niż jednej tabeli lub więcej niż jednej krotki.

Złączenia mogą być:

1) **wewnętrzne** (INNER JOIN) — tabela wynikowa będzie zawierać tylko takie elementy, dla których zostaną spełnione warunki złączenia wewnętrznego. Mogą to być:

- *równozłączenia* — powstają na podstawie operatora porównania;
- *nierównozłączenia* (THETA JOIN) — powstają na podstawie operatora innego niż porównania, np. znaku mniejszości, większości, (zaprzeczenia), BETWEEN

2) **zewnętrzne** (OUTER JOIN) — zwracają również krotki niepasujące do warunku. Wyróżnia się trzy typy złączeń zewnętrznych:

- LEFT OUTER JOIN (lewostronne złączenie zewnętrzne) — wynik złączenia zawiera wszystkie krotki pierwszej tabeli oraz wszystkie pasujące krotki drugiej tabeli;
- RIGHT OUTER JOIN (prawostronne złączenie zewnętrzne) — wynik złączenia zawiera wszystkie krotki drugiej tabeli oraz wszystkie pasujące krotki pierwszej tabeli;
- FULL OUTER JOIN (pełne złączenie zewnętrzne) — wynik złączenia zawiera wszystkie krotki obu złączonych tabel (również te, które nie mają swoich odpowiedników)

3) **złączenia krzyżowe** (CROSS JOIN) — otrzymuje się kombinację każdej krotki pierwszej tabeli z każdą krotką drugiej tabeli (iloczyn kartezjański)

4) **złączenie w obrębie jednej tabeli** (SELF JOIN) — jest używane, gdy po obu stronach złączenia występuje ta sama tabela. W tym wypadku używa się aliasów.

* Łączenie wyników zapytań *

W języku SQL istnieją mechanizmy, które pozwalają łączyć wyniki kilku zapytań. Do połączenia zapytań można użyć jednej z trzech instrukcji: UNION, INTERSECT, EXCEPT.

[1] Instrukcja **UNION** łączy wyniki zapytań i ma postać: *zapytanie1 UNION zapytanie2*. Warunkiem wykonania instrukcji jest, aby łączone zapytania miały taką samą liczbę kolumn oraz aby typy kolumn były takie same. Ta instrukcja domyślnie powoduje usunięcie powtarzających się wierszy. W celu zachowania w wyniku wszystkich wierszy należy użyć instrukcji **UNION ALL**.

[2] Instrukcja **INTERSECT** zwraca część wspólną wyników dwóch zapytań i ma postać: *zapytanie1 INTERSECT zapytanie2*. Tutaj również obydwa zapytania powinny zwracać taką samą liczbę kolumn o takich samych typach.

[3] Instrukcja **EXCEPT** zwraca te wiersze, które wystąpiły w wyniku pierwszego zapytania, ale nie było ich w wyniku drugiego zapytania. Zmiana kolejności zapytań w tej instrukcji spowoduje zmianę wyniku.

Ćwiczenia powtórkowe

UWAGA!

Odwiedź moją stronę: <http://marekkwiatkowski.eu>

W najnowszym wpisie znajduje się ten dokument w formacie PDF i bazy danych o nazwie „ksiegarnia_internetowa” i „firma” w formacie SQL do pobrania. Pobierz i zaimportuj te bazy danych, będą one nam potrzebne do dalszych ćwiczeń.

Ćwiczenie nr 1 [łączenie dwóch tabel] – na dobry początek!

Wykonaj następujące zapytanie SQL na bazie danych 'ksiegarnia_internetowa':

```
SELECT zamowienia.zamowienie_id, zamowienia.wartosc, zamowienia.data
FROM klienci, zamowienia
WHERE klienci.nazwisko='Józef Koń'
AND klienci.klient_id=zamowienia.klient_id;
```

Co uzyskałeś? Jakie możesz wyciągnąć z tego wnioski?

Mała uwaga - przyjrzyj się tabelom 'klienci' oraz 'zamowienia' i porównaj wyniki wyszukiwania z danymi znajdującymi się w tych tabelach. Dla 'Józef Koń', 'klient_id=5'. W tabeli 'zamowienia', 'klient_id' o wartości równej 5 występuje 2 razy i właśnie te rekordy zostały pokazane w wyniku wyszukiwania.

Ćwiczenie nr 2 – [łączenie kilku tabel] –no to jeszcze jedno dość proste ćwiczenie

Wykonaj następujące zapytanie SQL na bazie danych o nazwie 'ksiegarnia_internetowa':

```
SELECT klienci.nazwisko
FROM klienci, zamowienia, ksiazki_zamowione, ksiazki
WHERE klienci.klient_id=zamowienia.klient_id
AND zamowienia.zamowienie_id=ksiazki_zamowione.zamowienie_id
AND ksiazki_zamowione.isbn=ksiazki.isbn
AND ksiazki.tytul LIKE '%C++%';
```

Co uzyskałeś? Jakie możesz wyciągnąć z tego wnioski?

Mała uwaga - powiązaliśmy dane z 4 tabel. Aby je powiązać za pomocą połączenia typu *equi-join*, należało zastosować 3 różne warunki połączenia. W przypadku łączenia tabel, prawidłowością jest, że na każdą ich parę przypada jeden warunek. W rezultacie, **suma wszystkich nałożonych warunków połączenia, jest o jeden mniejsza, niż liczba łączonych tabel.**

Ćwiczenie nr 3 [zastosowanie instrukcji UNION, INTERSECT I EXCEPT – łączenie wyników zapytań]

Pracując na bazie danych o nazwie 'firma' wykonaj następujące działania, **pamiętaj o sygnalizowaniu wykonania poszczególnych podpunktów zadania prowadzącemu zajęcia.**

- 1) Wyświetl nazwisko, imię, pensję przeliczoną na euro (sprawdź aktualny kurs dolara do euro w Internecie), dział, stan pracowników, którzy zarabiają powyżej 2000 Euro. Uporządkuj dane według malejącej pensji.
- 2) Dopisz siebie do tabeli 'personel', dodaj również informacje w polach: nazwisko, imię, pensja, data_zatr, zwolniony.
- 3) Wyświetl nazwiska oraz imiona pracowników (z tabeli 'personel') oraz klientów (z tabeli 'klient'). Uporządkuj alfabetycznie dane na podstawie pola 'nazwisko'.

Zadanie domowe:

Korzystając z bazy danych o nazwie 'firma.sql' otwórz ją w darmowym systemie PostgreSQL, do pobrania z mojej strony. Wyświetl identyfikatory stanów, w których mieszczą się siedziby i dostawców i klientów. Wykonaj zrzut ekranu, wpisz w dokumencie tekstowym użyte polecenie, liczbę rekordów i wyślij odpowiedź w skompresowanym folderze na adres marek@teachers.org bądź poprzez stronę internetową <http://marekkwiatkowski.eu> i zawartą na niej opcję „Wyślij plik!” [czas wysyłki - do środy 20.02.2019].