

Złączenia tabel w języku SQL

Model relacyjny zakłada, że dane, zawarte w bazie danych, podzielone są na tabele. Wydaje się zatem naturalne, że w pewnych sytuacjach konieczne będzie połączenie danych, zawartych w kilku tabelach, w celu uzyskania tabeli wynikowej. Proces taki nazywamy złączeniem (ang. *join*), ze względu na sposób łączenia dzieli się operacje złączeń na trzy grupy:

- złączenia wewnętrzne
- złączenia zewnętrzne
- samozłączenia

Podstawowe pytanie brzmi: w jaki sposób łączone są ze sobą informacje z dwóch tabel w bazie? Tabele mogą różnić się od siebie wszystkimi cechami, nie narzucanymi przez model relacyjny, a więc: ilością wierszy, ilością kolumn, oraz ich typami. Kluczem jest pojęcie, znane z algebry – iloczyn kartezjański. Operacje łączenia tabel zawierają w trakcie ich realizacji wstępny etap utworzenia nowej tabeli, będącej iloczynem kartezjańskim wierszy z łączonych tabel. Najprostsze zapytanie, pozwalające się o tym przekonać, ma postać:

SELECT * FROM tabela1, tabela2

Składnia powyższego zapytania oparta jest o standard języka *SQL-89*. Nowsza postać różnoważnego zapytania, w standardzie oznaczanym jako *SQL-92*, przedstawia się następująco:

SELECT * FROM tabela1 CROSS JOIN tabela2

Należy pamiętać o tym, że tabela będąca iloczynem kartezjańskim, ma ilość kolumn równą sumie ilości kolumn tabel źródłowych, a ilość wierszy równą iloczynowi ilości wierszy tabel źródłowych. Dla dużych tabel oznacza to ogromne obciążenie dla systemu komputerowego obsługującego bazę i w konsekwencji wolne przetwarzanie zapytań. Jeśli istnieje taka możliwość, to warto starać się minimalizować konieczność złączeń na dużych tabelach. Iloczyn kartezjański tabel używany jest następnie w operacjach złączeń. Złączenia więcej niż dwóch tabel oparte jest o rekurencyjne założenie, że skoro złączenie dwóch tabel jest tabelą, to np. złączenie trzech tabel to złączenie tabeli z wynikiem złączenia dwóch innych tabel itd...

Podstawowe, jak również najczęściej spotykane są *złączenia wewnętrzne*. Opierają się one na ograniczeniu iloczynu kartezjańskiego dwóch tabel źródłowych w oparciu o pewne relacje kolumn z tych tabel. Najczęściej pojawia się tu ograniczenie, oparte o relację pomiędzy kluczem głównym jednej z tabel, a odpowiednim kluczem obcym z drugiej tabeli. Schemat zapytania skonstruowanego w oparciu o powyższą relację dla kluczy jednokolumnowych, sformułowany w składni *SQL-89* ma postać:

SELECT * FROM tabela1, tabela2 WHERE klucz_główny = klucz_obcy

Podczas realizacji złączenia może pojawić się następujący problem: w łączonych tabelach klucz główny i obcy mają tę samą nazwę. W nowej składni *SQL* jest to wręcz pożądane, ale w starej stanowi problem, który można rozwiązać poprzez kwalifikowanie kolumn, w odniesieniu do tabel, za pomocą kropki:

SELECT * FROM tabela1, tabela2 WHERE tabela1.klucz_główny = tabela2.klucz_obcy

Dla złączeń wewnętrznych nowa składnia *SQL-92* udostępnia trzy warianty tworzenia zapytania:

1. Jeżeli chcemy dokonać złączenia tabel względem **WSZYSTKICH KOLUMN MAJĄCYCH TAKIE SAME NAZWY**, zapytanie ma postać:

SELECT * FROM tabela1 NATURAL JOIN tabela2

2. Jeżeli chcemy dokonać złączenia tabel względem **NIEKTÓRYCH KOLUMN, MAJĄCYCH TAKIE SAME NAZWY**, zapytanie ma postać:

```
SELECT * FROM tabela1 JOIN tabela2 USING(lista_nazw_kolumn)
```

3. Jeżeli nazwy kolumn, względem których chcemy dokonać złączenia tabel **MAJĄ RÓŻNE NAZWY**, stosuje się zapytanie postaci:

```
SELECT * FROM tabela1 JOIN tabela2 ON warunek_złączenia_tabel
```

Warunek złączenia tabel powinien zostać ograniczony wyłącznie do uzgodnienia reguł połączenia tabel, do realizacji operacji ograniczenia standardowo powinno się stosować klauzulę *WHERE*. Ponadto w bazie MySQL może okazać się konieczne użycie słów *INNER JOIN* zamiast słowa *JOIN*.

Drugim typem złączeń są *złączenia zewnętrzne*. Są one pewnego rodzaju rozszerzeniem pojęcia złączenia wewnętrznego. Podczas gdy w złączeniu wewnętrznym w tabeli wynikowej umieszczane były wiersze, spełniające pewne kryterium, w złączeniu zewnętrznym dodatkowo dodawane są wiersze z jednej z tabel (a nawet obu), które zostałyby odrzucone w złączeniu wewnętrznym, ze względu na nie spełnienie kryterium złączenia, po odpowiednim uzupełnieniu wierszy wynikowych wartościami pustymi. Do czego można wykorzystać taką postać złączenia? Wyobraźmy sobie bazę danych o pracownikach firmy, która zawiera między innymi tabelę z danymi osobowymi pracowników oraz tabelę zawierającą rejestrację wypłat wynagrodzenia, połączone ze sobą odpowiednią relacją. Oczywiście, aby wyświetlić informację o tym, komu dokonano wypłaty wynagrodzenia wystarczy wykonać złączenie wewnętrzne obu tych tabel. Jak jednak znaleźć osoby, którym (być może przez pomyłkę) *nie wypłacono* wynagrodzenia? Jeżeli nie dokonano wypłaty, nie ma wpisu w tabeli z wypłatami, z którym można by było powiązać osobę! Pomocne tu staje się złączenie zewnętrzne, w którym oprócz umieszczenia w tabeli wynikowej danych o pracownikach, którzy wypłatę otrzymali, dodamy również wiersze z tabeli zawierającej dane personalne, nie mające powiązania z wierszami tabeli wypłat. Wiersze takie w procesie złączenia zewnętrznego mają puste (*NULL*) wpisy we wszystkich kolumnach, pochodzących z tabeli w wypłatami (właśnie dlatego, że z informacjami odnośnie osoby, która nie otrzymała wypłaty, nie są powiązane informacje o dokonanej wypłacie), co można łatwo odfiltrować realizując odpowiednie ograniczenie np. ***kwota IS NULL***, i otrzymując w ten sposób dane o pracownikach, którzy nie otrzymali jeszcze wynagrodzenia.

Złączenia zewnętrzne dzielimy na trzy kategorie:

- lewostronne
- prawostronne
- pełne

Jak nietrudno się domyślić, złączenie zewnętrzne lewostronne to takie, w którym do wyniku złączenia wprowadzone zostaną bezwarunkowo wszystkie wiersze z tabeli, stojącej po lewej stronie deklaracji złączenia, złączenie zewnętrzne prawostronne to analogicznie takie, które wprowadza do tabeli wynikowej wszystkie wiersze z tabeli, stojącej po prawej stronie deklaracji złączenia, a złączenie pełne to takie, które wprowadza do wyniku wszystkie wiersze obu tabel, bez względu na nałożone ograniczenie. Oczywiście w wierszu, dla którego nie jest spełniony warunek złączenia, wartości kolumn z "przeciwnieległej tabeli" wypełniane są znacznikami *NULL* w wierszu tabeli wynikowej.

Składnia zapytań dla złączeń zewnętrznych przedstawia się następująco:

```
SELECT * FROM tab1 LEFT OUTER JOIN tab2... – złączenie lewostronne zewnętrzne
```

```
SELECT * FROM tab1 RIGHT OUTER JOIN tab2... – złączenie prawostronne zewnętrzne
```

```
SELECT * FROM tab1 FULL OUTER JOIN tab2... – złączenie pełne zewnętrzne
```

a następnie należy podać predykat po klauzuli **ON**.

UWAGA! Baza danych *MySQL*, w zależności od wersji, może:

- nie realizować złączeń pełnych zewnętrznych
- pozwalać na użycie zamiast klauzuli *ON* wyrażenie klauzuli *USING(lista_kolumn)*
- traktować słowo *OUTER* jako opcjonalne

Trzecim i ostatnim rodzajem złączeń są tzw. *samołączenia*. Jak wskazuje nazwa, operacja ta powoduje utworzenie tabeli wynikowej w oparciu o iloczyn kartezjański tabeli z nią samą. Do czego może być wykorzystane takie złączenie? Przypuśćmy, że w bazie danych z pracownikami firmy jest tabela, w której zaznaczana jest obecność każdego pracownika w danym dniu. Kierownictwo postanowiło dać premię osobom, które pracowały zarówno pierwszego, jak trzeciego maja. Nie jest sensowne wydanie zapytania z predykatem:

data='2003-05-01' AND data='2003-05-03'

ponieważ wpis dotyczący daty nie może mieć *jednocześnie* dwóch wartości – w wyniku tego zapytanie z powyższym predykatem zawsze zwróci pustą tabelę wynikową. Można jednak wykonać złączenie wewnętrzne tej tabeli ze sobą, w oparciu o ograniczenie, że w danym wierszu identyfikatory pracownika (pochodzące z “obu” tabel) mają być równe. W wyniku otrzymamy tabelę, w której za pomocą selekcji możemy wybrać wiersze, w których “pierwsze” pole z datą ma wartość równą dacie pierwszego maja, a “drugie” pole z datą ma wartość równą dacie trzeciego maja. W ten sposób otrzyma się listę pracowników, którzy pracowali zarówno pierwszego, jak trzeciego maja.

Technicznie samołączenie realizowane jest prawie tak samo, jak złączenie wewnętrzne dwóch tabel, poza jednym szczegółem. Nie jest możliwe wprost złączenie tabeli z nią samą, gdyż spowoduje to konflikt nazw kolumn. Łącząc dwie różne tabele ze sobą, nawet jeśli w obu z nich znajdują się kolumny o tej samej nazwie, są one rozróżnialne, gdyż pochodzą z dwóch tabel o różnych nazwach i można się do nich odnieść, kwalifikując kolumnę nazwą tabeli, z której pochodzi np. Tab1.ID lub Tab2.ID. Gdy jednak również nazwy łączonych tabel są takie same, powoduje to konflikt. Rozwiązuje się go poprzez nadanie dwóch różnych nazw synonimicznych (aliasów) tabeli przy samołączeniu:

SELECT * FROM tabela AS tab1 JOIN tabela AS tab2 USING(pracownik) WHERE ...