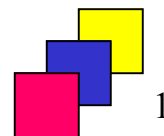


Język SQL. Rozdział 5.

Połączenia i operatory zbiorowe

Iloczyn kartezjański, połączenie równościowe, połączenie nierównościowe, połączenie zwrotne, połączenie zewnętrzne, składnia jawna połączeń, składnia niejawną połączeń, operatory zbiorowe.



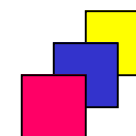
Iloczyn kartezjański

- Iloczyn kartezjański (ang. *cross-join*) dwóch relacji to zbiór wszystkich możliwych kombinacji krotek z obu relacji. W większości wypadków wystąpienie iloczynu kartezjańskiego sygnalizuje błąd w zapytaniu. Iloczyn kartezjański rzadko bywa przydatny.
- Iloczyn kartezjański w składni jawnej:

```
SELECT nazwisko, etat, nazwa  
FROM pracownicy CROSS JOIN etaty;
```

- Iloczyn kartezjański w składni niejawnej:

```
SELECT nazwisko, etat, nazwa  
FROM pracownicy, etaty;
```



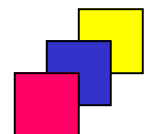
Połączenie (1)

- Zapytanie, którego wynikiem jest kombinacja tych rekordów z dwóch relacji, dla których spełniony jest **warunek połączeniowy**.
- Połączenie jest podzbiorem iloczynu kartezyjskiego.
- Ogólna postać:
 - składnia jawna

```
SELECT lista atrybutów  
FROM relacja_1 operator_połączenia relacja_2 warunek_połączeniowy  
WHERE pozostałe_warunki ...;
```

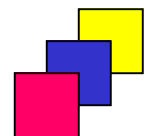
- składnia niejawna

```
SELECT lista atrybutów  
FROM relacja_1, relacja_2  
WHERE warunek_połączeniowy AND pozostałe_warunki ...;
```



Podział połączeń

- ze względu na postać warunku połączeniowego:
 - równościowe (ang. **equijoin**):
 - naturalne (ang. **natural join**)
 - zwykłe
 - nierównościowe (ang. **nonequijoin**)
- ze względu na sposób dopasowania rekordów łączonych relacji:
 - wewnętrzne (ang. **inner join**, **simple join**) – w zbiorze wynikowym tylko te rekordy z obu relacji, dla których spełniony jest warunek połączeniowy
 - zewnętrzne (ang. **outer join**) – w zbiorze wynikowym rekordy obu relacji, dla których spełniony jest warunek połączeniowy + pozostałe rekordy (z jednej lub obu relacji).
- ze względu na łączone relacje:
 - zwykłe
 - zwrotne (ang. **self join**)



Połączenie równościowe (1)

PRACOWNICY

ID_PRAC	NAZWISKO	ID_ZESP	
180	MAREK	10	
200	ZAKRZEWICZ	30	
210	BIAŁY	30	

ZESPOLY

ID_ZESP	NAZWA
10	ADMINISTRACJA
30	SYSTEMY EKSPERCKIE

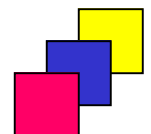
PRACOWNICY

ID_PRAC	NAZWISKO	ETAT	
180	BLAZEWICZ	PROFESOR	
200	SLOWINSKI	PROFESOR	
210	KOSZLAJDA	ADIUNKT	

ETATY

NAZWA	PLACA_MIN
PROFESOR	800
ADIUNKT	510

- Połączenie, w którym warunek połączeniowy wykorzystuje operator równości
- W wyniku połączenia równościowego otrzymujemy wszystkie krotki z obu relacji spełniające warunki połączenia.
- ang. *equijoin*



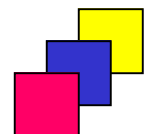
Połączenie równościowe (2)

- Operator połączenia w składni jawnej: **INNER JOIN ... ON**
 - słowo INNER można pominąć

```
SELECT lista_atrybutów  
FROM relacja_1 INNER JOIN relacja_2 ON warunek połączeniowy;
```

- Przykład:

```
SELECT nazwisko, placa_pod, nazwa, placa_min  
FROM pracownicy INNER JOIN etaty ON etat = nazwa;
```



Połączenie równościowe (3)

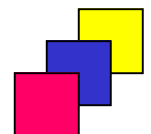
- Jeżeli w łączonych relacjach występują atrybuty o takich samych nazwach, to w zapytaniu muszą być poprzedzone nazwą relacji w celu uniknięcia dwuznaczności.

```
SELECT nazwisko, zespoly.id_zesp, pracownicy.id_zesp, nazwa
FROM pracownicy join zespoly
ON pracownicy.id_zesp = zespoly.id_zesp;
```

- W klauzuli FROM można wprowadzić **aliasy** (alternatywne nazwy relacji) i używać aliasów zamiast nazw relacji. Jeśli alias został użyty, to nie wolno używać oryginalnej nazwy relacji.

```
SELECT nazwisko, p.id_zesp, z.id_zesp, nazwa
FROM pracownicy p JOIN zespoly z
ON p.id_zesp = z.id_zesp;
```

aliasy



Połączenie równościowe (4)

- Składnia niejawna:

```
SELECT nazwisko, placa_pod, nazwa  
FROM pracownicy, etaty  
WHERE etat = nazwa;
```

```
SELECT nazwisko, zespoly.id_zesp, nazwa  
FROM pracownicy, zespoly  
WHERE pracownicy.id_zesp = zespoly.id_zesp;
```

```
SELECT nazwisko, z.id_zesp, nazwa  
FROM pracownicy p, zespoly z  
WHERE p.id_zesp = z.id_zesp;
```

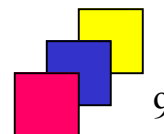

Połączenie naturalne (1)

- Jeśli nazwy atrybutów w warunku połączeniowym są takie same, wówczas połączenie równościowe nosi nazwę połączenia naturalnego (ang. *natural join*).
- Składnia jawna:
 - operator **NATURAL JOIN** – warunki równości dotyczą **wszystkich** par atrybutów o takich samych nazwach, należących do łączonych relacji.

```
SELECT lista_atrybutów  
FROM relacja_1 NATURAL JOIN relacja_2;
```

- operator **JOIN ... USING ()** – warunki równości dotyczą tylko atrybutów wymienionych po słowie **USING**

```
SELECT lista_atrybutów  
FROM relacja_1 JOIN relacja_2 USING (lista_atrybutów);
```



Połączenie naturalne (2)

- Przykłady:

```
SELECT nazwisko, id_zesp, nazwa  
FROM pracownicy NATURAL JOIN zespoly;
```

```
SELECT nazwisko, id_zesp, nazwa  
FROM pracownicy JOIN zespoly USING (id_zesp);
```

- W składni niejawnej brak osobnej konstrukcji dla połączenia naturalnego

```
SELECT nazwisko, z.id_zesp, nazwa  
FROM pracownicy p, zespoly z  
WHERE p.id_zesp = z.id_zesp;
```

Połączenie naturalne a równościowe – różnice

W wyniku połączenia naturalnego (zarówno NATURAL JOIN jak i JOIN ... USING ()) atrybut połączeniowy występuje tylko raz. W wyniku połączenia równościowego występują oba atrybuty połączeniowe z obu łączonych relacji.

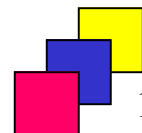
```
SELECT * FROM pracownicy NATURAL JOIN zespoly;
```

```
SELECT * FROM pracownicy JOIN zespoly USING (id_zesp);
```

ID_ZESP	ID_PRAC	NAZWISKO	ETAT	ID_SZEFA	ZATRUDNIO	PLACA_POD	PLACA_DOD	NAZWA	ADRES
10	100	WEGLARZ	DYREKTOR		01-JAN-68	1730	420.5	ADMINISTRACJA	PIOTROWO 3A
40	110	BLAZEWICZ	PROFESOR	10001	MAY-73	1350	210	ALGORYTMY	WLODKOWICA 16
30	120	SLOWINSKI	PROFESOR	10001	SEP-77	1070		SYSTEMY EKSPERCKIE	STRZELECKA 14

```
SELECT * FROM pracownicy JOIN zespoly  
ON pracownicy.id_zesp = zespoly.id_zesp;
```

ID_PRAC	NAZWISKO	ETAT	ID_SZEFA	ZATRUDNIO	PLACA_POD	PLACA_DOD	ID_ZESP	ID_ZESP	NAZWA	ADRES
100	WEGLARZ	DYREKTOR		01-JAN-68	1730	420.5	10	10	ADMINISTRACJA	PIOTROWO 3A
110	BLAZEWICZ	PROFESOR	10001	MAY-73	1350	210	40	40	ALGORYTMY	WLODKOWICA 16
120	SLOWINSKI	PROFESOR	10001	SEP-77	1070		30	30	SYSTEMY EKSPERCKIE	STRZELECKA 14



Połączenie nierównościone (1)

PRACOWNICY

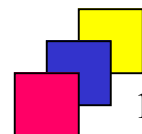
ID_PRAC	NAZWISKO	PLACA_POD
180	MAREK	410.2
200	ZAKRZEWICZ	208

ETATY

NAZWA	PLACA_MIN	PLACA_MAX
SEKRETARKA	270	450
STAŻYSTA	150	250



- Połączenie, w którym w warunku połączeniowym zastosowano inny niż = operator języka SQL.
- ang. **non-equijoin**



Połączenie nierównościowe (2)

- Przykłady:

```
SELECT nazwisko, nazwa, placa_pod, placa_min, placa_max  
FROM pracownicy JOIN etaty  
ON placa_pod BETWEEN placa_min AND placa_max;
```

```
SELECT nazwisko, nazwa, placa_pod, placa_min, placa_max  
FROM pracownicy, etaty  
WHERE placa_pod BETWEEN placa_min AND placa_max;
```

Połączenie zewnętrzne (1)

PRACOWNICY

ID_PRAC	NAZWISKO	ID_ZESP
180	MAREK	10
?	?	?

ZESPOLY

ID_ZESP	NAZWA
40	ALGORYTMY
50	BADANIA OPERACYJNE

<i>null</i>	<i>null</i>	<i>null</i>
-------------	-------------	-------------

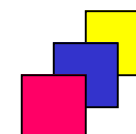
PRACOWNICY

ID_PRAC	NAZWISKO	ID_ZESP
180	MAREK	10
300	NOWICKI	99

ZESPOLY

ID_ZESP	NAZWA
50	BADANIA OPERACYJNE
?	?

<i>null</i>	<i>null</i>
-------------	-------------



Połączenie zewnętrzne (2)

- Przy połączeniu wewnętrznym wiersze z obu relacji nie posiadające odpowiedników spełniających warunek połączenia nie są wyświetlane.
- Połączenie zewnętrzne (ang. *outer join*) zachowuje wszystkie krotki z wybranej relacji, łącząc je z "wirtualnymi" krotkami z drugiej krotki. "Wirtualne" krotki są wypełnione wartościami pustymi.
- Operator w składni jawnej: **LEFT | RIGHT | FULL OUTER JOIN ... ON | USING ()**:
 - **LEFT** – połączenie lewostronne, do wyniku połączenia wewnętrznego obu relacji zostaną dodane te rekordy z relacji z lewej strony operatora, które nie połączyły się z rekordami relacji z prawej strony operatora
 - **RIGHT** – połączenie prawostronne, do wyniku połączenia wewnętrznego obu relacji zostaną dodane te rekordy z relacji z prawej strony operatora, które nie połączyły się z rekordami relacji z lewej strony operatora
 - **FULL** – połączenie obustronne, wynik połączenia wewnętrznego obu relacji zostanie uzupełniony o rekordy obu relacji, dla których nie został spełniony warunek połączeniowy

Połączenie zewnętrzne (3)

- Słowo OUTER jest opcjonalne.
- Przykłady:

```
SELECT nazwa, nazwisko, etat  
FROM pracownicy p RIGHT OUTER JOIN ZESPOLY z  
ON z.id_zesp = p.id_zesp;
```

```
SELECT nazwa, nazwisko, etat  
FROM PRACOWNICY p LEFT OUTER JOIN zespoly z  
ON z.id_zesp = p.id_zesp;
```

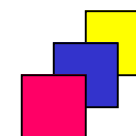
```
SELECT nazwa, nazwisko, etat  
FROM ZESPOLY FULL OUTER JOIN PRACOWNICY  
USING (id_zesp);
```


Połączenie zewnętrzne (4)

- Jeśli oprócz warunku połączeniowego w zapytaniu umieszczono dodatkowy warunek filtrujący na kolumnie z relacji, w której może "zabraknąć" rekordów, warunek powinien zostać umieszczony po operatorze AND w klauzuli FROM zapytania.

```
SELECT nazwa, nazwisko, etat  
FROM ZESPOLY z LEFT OUTER JOIN pracownicy p  
ON z.id_zesp = p.id_zesp AND placa_pod > 1000;
```

```
SELECT nazwa, nazwisko, etat  
FROM ZESPOLY z LEFT OUTER JOIN pracownicy p  
ON z.id_zesp = p.id_zesp  
WHERE placa_pod > 1000;
```

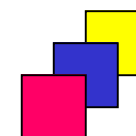


Połączenie zewnętrzne (5)

- Składnia niejawna – operator (+) w warunku połączeniowym wskazuje relację, w której "zabraknie" rekordów do połączenia z drugą relacją (a więc to relacja w warunku po stronie bez (+) zachowa wszystkie swoje rekordy).
- Przykłady:

```
SELECT z.id_zesp, z.nazwa, p.nazwisko  
FROM ZESPOLY z, pracownicy p  
WHERE p.id_zesp (+) = z.id_zesp;
```

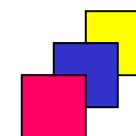
```
SELECT z.id_zesp, z.nazwa, p.nazwisko  
FROM zespoly z, PRACOWNICY p  
WHERE p.id_zesp = z.id_zesp (+);
```



Połączenie zewnętrzne (6)

- Ograniczenia:
 - jeśli warunek połączeniowy jest złożony, operator (+) musi zostać zastosowany do wszystkich części składowych warunku.
 - warunek związany z połączeniem zewnętrznym nie może używać operatora IN ani nie może być łączony z innymi warunkami za pomocą operatora OR
 - w składni niejawnej brak możliwość wykonania obustronnego połączenia zewnętrznego.

```
SELECT z.id_zesp, z.nazwa, p.nazwisko  
FROM ZESPOLY z, pracownicy p  
WHERE p.id_zesp (+) = z.id_zesp  
AND p.placa_pod (+) > 1000;
```



Połączenie zwrotne (1)



- Połączenie (ang. **self join**), w którym relacja łączy się "sama z sobą" (występuje kilkukrotnie w klazuli FROM).
- Użycie aliasów w połączeniu zwrotnym jest obowiązkowe.

Połączenie zwrotne (2)

```
SELECT p.nazwisko, s.nazwisko  
FROM pracownicy p, pracownicy s  
WHERE p.id_szefa = s.id_prac;
```

```
SELECT p.nazwisko, s.nazwisko  
FROM pracownicy p JOIN pracownicy s  
ON p.id_szefa = s.id_prac;
```

Połączenia wielu relacji

- Połączenie jest operacją binarną, jeśli w zapytaniu więcej niż dwie relacje, wówczas połączenie jest realizowane parami, np.: relacja A z relacją B, wynik połączenia z relacją C, itd.
- Łącząc **N** relacji należy podać minimum **N-1** warunków połączenia.

```
SELECT p.nazwisko, z.nazwa, e.nazwa, e.placa_min, e.placa_max
FROM pracownicy p JOIN zespoły z ON p.id_zesp = z.id_zesp
      JOIN etaty e ON p.placa_pod BETWEEN placa_min AND
      placa_max;
```

```
SELECT p.nazwisko, z.nazwa, e.nazwa, e.placa_min, e.placa_max
FROM pracownicy p, zespoły z, etaty e
WHERE p.id_zesp = z.id_zesp
      AND p.placa_pod BETWEEN placa_min AND placa_max;
```



Operatory zbiorowe

UNION

- suma zbiorów
- eliminuje duplikaty

UNION ALL

- suma zbiorów
- nie eliminuje duplikatów

INTERSECT

- część wspólna zbiorów
- eliminuje duplikaty

MINUS (standard SQL: EXCEPT)

- różnica zbiorów
 - eliminuje duplikaty
-
- Operatory zbiorowe nazywamy również operatorami złączeń pionowych (*vertical join*) ponieważ łączą kolumny zbiorów wynikowych.

```
SELECT etat
FROM pracownicy
WHERE id_zesp = 20
      MINUS
SELECT etat
FROM pracownicy
WHERE id_zesp = 30
ORDER BY etat;
```

```
SELECT nazwisko
FROM pracownicy
      UNION
SELECT nazwa
FROM zespoly;
```

Reguły stosowania operatorów zbiorowych

- W łączonych operatorami zbiorowymi klauzulach SELECT musi wystąpić ta sama liczba atrybutów.
- Typy odpowiednich atrybutów różnych klauzul SELECT muszą być zgodne.
- W wyniku zapytania pojawiają się nazwy atrybutów wyłącznie z pierwszej klauzuli SELECT.
- Klauzula ORDER BY może być użyta tylko jako ostatnia klauzula zapytania.
- Polecenia SELECT są wykonywane w kolejności ich wystąpienia (od góry do dołu), nawiasy umożliwiają zmianę domyślnej kolejności wykonywania łączonych operatorami zbiorowymi poleceń SELECT.
- **UWAGA:** w celu zapewnienia zgodności ze standardem SQL w przyszłych wydaniach Oracle operator INTERSECT będzie miał największy priorytet.

