

## MAX oraz MIN

Funkcja **MIN ()** zwraca najmniejszą wartość wybranej kolumny.

Funkcja **MAX ()** zwraca największą wartość wybranej kolumny.



## Funkcje COUNT, AVG oraz SUM

Funkcja **COUNT ()** zwraca liczbę wierszy, które spełniają określone kryteria.

Funkcja **AVG ()** zwraca średnią wartość kolumny numerycznej.

Funkcja **SUM ()** zwraca całkowitą sumę kolumny numerycznej.



## Funkcje COUNT, AVG oraz SUM

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```



## GROUP BY

Instrukcja **GROUP BY** jest często używana z funkcjami agregującymi (COUNT, MAX, MIN, SUM, AVG), aby grupować zestaw wyników za pomocą jednej lub więcej kolumn.

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
ORDER BY column_name(s);
```



## HAVING

Klauzula **HAVING** została dodana do SQL, ponieważ nie można było użyć słowa kluczowego WHERE z funkcjami agregującymi.

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
HAVING condition  
ORDER BY column_name(s);
```



## RELACJA 1 : n

Relacja **jeden do wielu** istnieje, gdy jeden wiersz w tabeli A może być połączony z wieloma wierszami w tabeli B, ale jeden wiersz w tabeli B jest połączony z tylko jednym wierszem w tabeli A.

- Klienci mogą wykonywać wiele zamówień.
- Zamówienia mogą zawierać wiele elementów.
- Elementy mogą zawierać opisy w wielu językach.



## RELACJA 1 : 1

W relacyjnej bazie danych relacja **jeden do jednego** istnieje, gdy jeden wiersz w tabeli może być połączony tylko z jednym wierszem w innej tabeli i odwrotnie.

- Kraj ma tylko jedną stolicę, a stolica jest stolicą tylko jednego kraju



## RELACJA n : n

Relacje **wiele do wielu** są zazwyczaj realizowane za pomocą tabeli asocjacyjnej (znanej również jako tabela porównawcza).

- Każde zlecenie może zawierać wiele elementów. Każda pozycja może być również w wielu zamówieniach.





## INNER JOIN

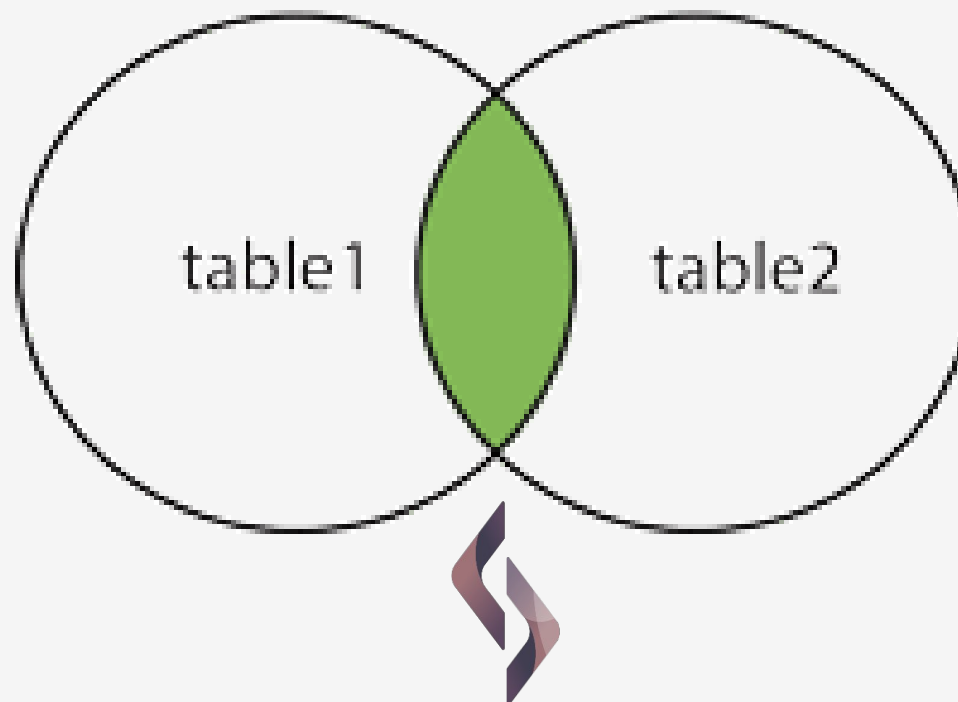
Słowo kluczowe **INNER JOIN** wybiera rekordy, które pasują do obu tabel.

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2 ON table1.column_name =  
table2.column_name;
```



## INNER JOIN

INNER JOIN



## LEFT JOIN

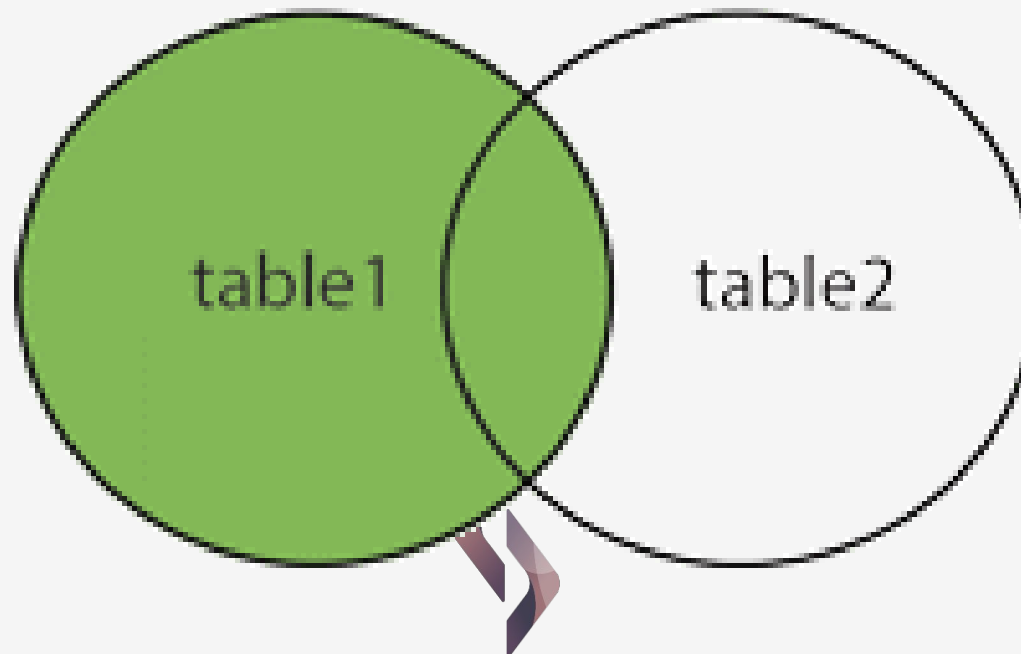
Słowo kluczowe **LEFT JOIN** zwraca wszystkie rekordy z lewej tabeli (table1) i dopasowane rekordy z prawej tabeli (table2). Wynik jest NULL z prawej strony, jeśli nie ma dopasowania.

```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2 ON table1.column_name =  
table2.column_name;
```



## LEFT JOIN

LEFT JOIN



## RIGHT JOIN

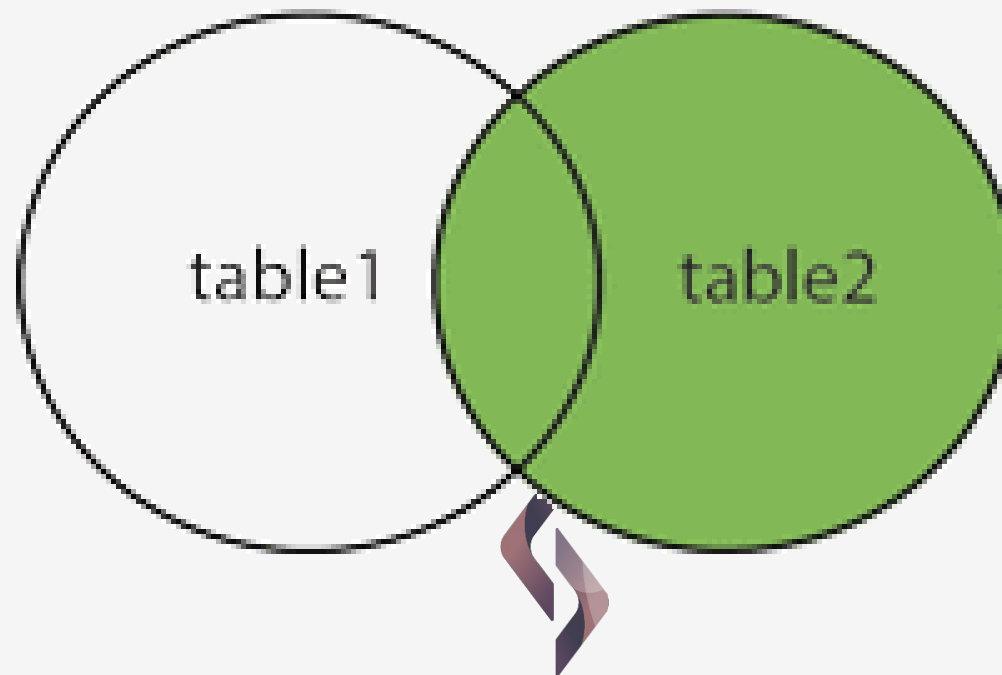
Słowo klucz **RIGHT JOIN** zwraca wszystkie rekordy z prawej tabeli (tabela2) i dopasowane rekordy z lewego tabeli (table1). Wynik jest NULL z lewej strony, gdy nie ma dopasowania.

```
SELECT column_name(s)  
FROM table1  
RIGHT JOIN table2 ON table1.column_name =  
table2.column_name;
```



## RIGHT JOIN

RIGHT JOIN



## FULL JOIN

Słowo kluczowe **FULL OUTER JOIN** zwraca wszystkie rekordy, gdy w tabelach po lewej (tabela 1) lub po prawej (tabela 2) zwraca się rekord.

```
SELECT column_name(s)  
FROM table1  
FULL OUTER JOIN table2 ON table1.column_name =  
table2.column_name;
```



## FULL JOIN

FULL OUTER JOIN

