

## 1. Tworzenie tabeli

Aby stworzyć tabele w SQL-u należy użyć polecenia

```
CREATE TABLE nazwa_tabeli (nazwa_pola1 właściwości_pola1 , nazwa_pola2  
właściwości_pola2 , itd.)
```

Nazwa\_tabeli to wybrana przez nas nazwa dla danej tabeli. Nie może ona zawierać spacji, polskich liter ani innych znaków specjalnych. To samo dotyczy nazwy\_pola. Właściwość\_pola to dane które określają pole. Informacje o możliwych wartościach znajdziecie w dokumentacji bazy danych w której pracujecie.

Np.:

```
CREATE TABLE autor (autor_id INT NOT NULL AUTO_INCREMENT , imie  
VARCHAR(32) , nazwisko VARCHAR(32) );
```

Te polecenie stworzy tabele o nazwie autor. Tabela będzie miała trzy pola.

Pierwsze pole o nazwie autor\_id będzie numerem identyfikacyjnym autora, który nie będzie się powtarzał w żadnym innym rekordzie. Wartość INT określa w tym przypadku wartość liczbową przypisaną do tego pola. NOT NULL określa to, że pole nie może być puste, a AUTO\_INCREMENT powoduje iż baza danych automatycznie dla każdego wpisu będzie wstawiała w tym polu wartość, co zwolni nas z tego obowiązku. Jest to pole pozwalające na identyfikację każdego autora. Przypisany jednemu autorowi numer autor\_id nie powtórzy się nigdzie indziej.

Drugie pole ma nazwę imie, VARCHAR(32) informuje nas o tym, że w danym polu można wpisać tylko 32 znaki. Należy dobierać wartości pola tak aby były maksymalnie dopasowane do ilości znaków które chcemy w nie wprowadzić co w rezultacie zmniejszy rozmiar bazy danych. Niezbyt rozsądne byłoby przypisywanie temu polu wartości na przykład 100 znaków, osobiście nie znam kogoś o tak długim imieniu.

Trzecie pole nazwisko ma takie same cechy charakterystyczne jak pole imie. Zmieniła się tylko jego nazwa no i oczywiście to co do niego będziemy wprowadzać czyli nazwisko autora.

## 2. Umieszczanie danych w tabeli

Kiedy mamy już stworzoną tabelę musimy wprowadzić do niej dane. Robimy to za pomocą polecenia:

```
INSERT INTO nazwa_tabeli (pole1, pole2) VALUES ("wartość1" , "wartość2" , itd);
```

Wartości wprowadzamy w kolejności pól w tabeli którą poprzednio stworzyliśmy. Należy pamiętać aby umieszczać wartości w tzw. "uszkach" czyli wewnątrz znaczników "".

Np:

Dla stworzonej poprzednio przykładowej tabeli wprowadzamy dane.

```
INSERT INTO autor (autor_id, imie, nazwisko) VALUES ( " " , "Łukasz" , "Sosna");
```

Dane trafiają do odpowiednich pól przy czym pole pierwsze nie ma jako takiej wartości. To co wprowadzamy do tego pola to wartość NULL. Jak napisałem w poprzedniej lekcji pole autor\_id to pole typu AUTO\_INCREMENT i baza danych automatycznie uzupełni te pole kolejnym numerem. Próba wstawienia jakiegokolwiek innej wartości do tego pola niż NULL spowodowałaby błąd.

### 3. Wyświetlanie wyników

Kiedy mamy naszą tabelę i wypełniliśmy ją odpowiednią ilością danych czas na ich wyświetlenie. Sądzę, że będzie to najobszerniejsza część kursu gdyż jest mnóstwo sposobów sortowania i wyświetlania wyników.

Aby wyświetlić wpisy z tabeli używamy polecenia:

```
SELECT * FROM nazwa_tabeli
```

W ten sposób otrzymamy wszystkie rekordy w danej tabeli w kolejności w jakiej zostały do niej wprowadzone.

Np:

Aby wyświetlić listę autorów z utworzonej poprzednio tabeli używamy polecenia.

```
SELECT * FROM autor
```

W ten sposób na ekranie pojawią mi się wszystkie rekordy tej tabeli.

### 4. Sortowanie wyników

Jeżeli chcemy posortować wyniki uzyskane poprzez wyświetlenie wszystkich rekordów w bazie to musimy wybrać pole według którego chcemy sortować, a następnie sposób sortowania wyników.

```
SELECT * FROM nazwa_tabeli ORDER BY nazwa_pola sposób_sortowania
```

Nazwa\_pola jest polem według którego sortujemy, a sposób sortowania to:

- DESC - sortowanie od Z do A
- ASC - sortowanie od A do Z

Np:

Dla stworzonej poprzednio tabeli z autorami jeżeli chcieć by ją posortować w kolejności od Z do A i sortowanie miało by się odbywać względem wartości w polu nazwisko to polecenie wygląda następująco:

```
SELECT * FROM autor ORDER BY nazwisko DESC
```

W ten sposób wyświetlają nam się wszyscy autorzy w kolejności od Z do A

## 5. Uaktualnianie danych w tabeli

Aby zmienić wartość pola tabeli lub ją nad pisać używamy polecenia

```
UPDATE nazwa_tabeli SET nazwa_pola="wartość"
```

Nazwa\_pola to oczywiście pole którego wartość chcemy zmienić, a wartość to dane jakie do tego pola wpisujemy.

Np:

Dla tabeli autor wyglądało by to następująco

```
UPDATE autor SET imie="Łukasz"
```

Te polecenie zmieni imiona wszystkich autorów na Łukasz.

Jeżeli mamy na przykład wartość liczbową w danym polu i chcemy dodać do niej jeszcze inną wartość to należy wykonać to tak:

```
UPDATE nazwa_tabeli SET nazwa_pola="wartość+kolena_wartość"
```

Wtedy w polu o podanej nazwie do istniejącej już tam wartości zostanie dodana nowa.

Np:

Dodałem do tabeli autor pole o nazwie zarobek i wstawiłem do niego sumę jaką otrzymałem za napisanie jakiegoś artykułu. Pole te ma właściwość INT przez co będzie ono przyjmowało tylko liczby.

No więc chcąc dodać do wartości tego pola jeszcze dodatkową kwotę za napisanie kolejnego artykułu napisze polecenie:

```
UPDATE autor SET zarobek="zarobek+200"
```

Spowoduje to zwiększenie się wartości pola zarobek o 200.

## 6. Wyświetlanie warunkowe

Kiedys na pewno znajdzie potrzeba wyświetlenia rekordów odpowiadających odpowiednim warunkom. W tedy należy użyć konstrukcji.

```
SELECT * FROM nazwa_tabeli WHERE nazwa_pola="wartość"
```

Nazwa\_pola to oczywiście pole wobec którego chcemy posegregować dane, a wartość to odpowiednie dane zapisane w tym polu.

Np:

Gdybym z tabeli autor chciał wyświetlić wszystkich których zarobek wynosi 400 to użyłbym polecenia:

```
SELECT * FROM autor WHERE zarobek="400"
```

Otrzymam wszystkie rekordy spełniające powyższy warunek.

Jest jeszcze możliwe określenie kilku warunków dla danego pola. Dokonuje się tego za pomocą operatora AND.

```
SELECT * FROM nazwa_tabeli WHERE nazwa_pola="wartość" AND nazwa_pola="wartość"
```

Wyświetlą się tylko rekordy spełniające oba warunki jednocześnie.

Np:

Gdyby z tabeli autor chciałem wyświetlić wszystkich których zarobek wynosi 400 i na imię mają Łukasz użyłbym konstrukcji:

```
SELECT * FROM autor WHERE zarobek="400" AND imie="Łukasz"
```

Otrzymam tylko rekordy spełniające te warunki.

Jest jeszcze możliwe określenie kilku różnych warunków dla danego pola. Dokonuje się tego z pomocą operatora OR.

```
SELECT * FROM nazwa_tabeli WHERE nazwa_pola="wartość" OR nazwa_pola="wartość"
```

Wyświetlą się tylko rekordy które spełniają warunek pierwszy lub spełniają warunek drugi.

Np:

Gdyby z tabeli autor chciałem wyświetlić wszystkich których zarobek wynosi 400 lub na imię mają Łukasz użyłbym konstrukcji:

```
SELECT * FROM autor WHERE zarobek="400" OR imie="Łukasz"
```

Otrzymam tylko rekordy spełniające jeden z tych warunków.

## 7. Wyświetlanie rekordu zawierającego słowo

Jeżeli chcemy wyświetlić wszystkie rekordy zawierające jakiś wyraz lub jakąś literę, lub też część wyrazu, używamy polecenia:

```
SELECT * FROM nazwa_tabeli WHERE nazwa_pola LIKE "wartość"
```

Nazwa\_pola to oczywiście pole względem którego wartości szukamy. Jeżeli nie znamy pełnego wyrazu to jego brakujące części zastępujemy znakiem procentu "%"

Np:

Jeżeli początek wyrazu to "ba", a reszty nie znamy to taki sposób zapisu wygląda następująco "ba%".

Jeżeli koniec wyrazu to "za", a początku nie znamy to taki sposób zapisu wygląda następująco "%za".

Jeżeli środek wyrazu to "az", a reszty nie znamy to taki sposób zapisu wygląda następująco "%az%". Zastępujemy litery początkowe i końcowe znakiem procenta "%" co pozwoli na wstawienie w ich miejsce dowolnej wartości.

Np:

Jeżeli z tabeli autor poszukujemy kogoś o imieniu zaczynającym się na "Łuka" i nie znamy reszty to takie polecenie wygląda następująco.

```
SELECT * FROM autor WHERE imie LIKE "Łuka%"
```

W ten sposób otrzymamy rekordy spełniające dany warunek.

Można także wymyślić sobie to, aby na przykład dane pole nie zawierało ciągu znaków. Postępujemy tak jak poprzednio z tym, że zmienia się część komendy. Przed LIKE wstawiamy operator NOT.

```
SELECT * FROM nazwa_tabeli WHERE nazwa_pola NOT LIKE "wartość"
```

## 8. Wartości pomiędzy

Jeżeli chcemy wyświetlić wszystkie rekordy w których w danym polu znajdują się wartości pomiędzy jakimiś przedziałami to używamy polecenia:

```
SELECT * FROM nazwa_tabeli WHERE nazwa_pola BETWEEN "wartość" AND "wartość"
```

Nazwa\_pola to pole wobec którego zamierzamy sortować.

Np:

Z tabeli autor mamy zamiar wyświetlić wszystkie osoby których zarobek mieści się pomiędzy 200 a 600. Aby to zrobić używamy polecenia:

```
SELECT * FROM autor WHERE zarobek BETWEEN "200" AND "600"
```

## 9. Liczenie ilości rekordów

Aby policzyć ile rekordów znajduje się w naszej bazie danych używamy polecenia:

```
SELECT COUNT(*) FROM nazwa_tabeli
```

Nazwa\_tabeli to tabela dla której stosujemy operację

Np:

Aby policzyć ile rekordów jest w naszej tabeli autor używamy polecenia:

```
SELECT COUNT(*) FROM autor
```

W odpowiedzi otrzymamy liczbę całkowitą.

## 10. Sumowanie wartości pola

Często zachodzi potrzeba sumowania wartości pola. Robimy to poleceniem:

```
SELECT SUM(nazwa_pola) FROM nazwa_tabeli
```

Nazwa\_pola to pole którego wartość będziemy sumować.

Np:

Aby zsumować zarobek wszystkich autorów w tabeli autor polecenie musi wyglądać następująco:

```
SELECT SUM(zarobek) FROM autor
```

W odpowiedzi otrzymamy wartość liczbową

## 11. Zmiana rozmiaru liter

Jeżeli zaistnieje potrzeba zmiany zapisanych w danym polu liter z małych na duże to używamy polecenia:

```
UPDATE nazwa_tabeli SET nazwa_pola=UPPER(nazwa_pola)
```

Nazwa\_pola to pole wobec którego chcemy zastosować tą funkcję.

Np:

Aby zmienić wielkość liter w naszej tabeli autor w polu nazwisko z małych na duże wpisujemy polecenie:

```
UPDATE autor SET nazwisko=UPPER(nazwisko)
```

W ten sposób wartość pola nazawisko została zamieniona na wielkie litery.

Można zrobić też na odwrót, to znaczy zmienić duże litery na małe, wystarczy tylko użyć polecenia LOWER

```
UPDATE nazwa_tabeli SET nazwa_pola=LOWER(nazwa_pola)
```

## 12 Kasowanie rekordów

Usuwanie rekordów dokonuje się za pomocą polecenia:

```
DELETE FROM nazwa_tabeli WHERE nazwa_pola LIKE "wartość"
```

Taka komenda usuwa rekordy z danej tabeli, które spełniają warunek.

Np:

Aby usunąć z tabeli autor wpis określający mnie należało by wpisać jedno z następujących poleceń:

```
DELETE FROM autor WHERE imie LIKE "Łukasz" lub
```

```
DELETE FROM autor WHERE nazwisko LIKE "Sosna" lub
```

```
DELETE FROM autor WHERE zarobek LIKE "400"
```

Należy przy tym uważać aby nie usunąć przypadkiem jakiegoś innego rekordu spełniającego ten warunek. Najlepiej więc posłużyć się ID danego rekordu. Jako iż wprowadziłem się do bazy jako pierwszą pozycje moje ID wynosi 0. Prawidłowe polecenie wygląda więc:

```
DELETE FROM autor WHERE id_autor LIKE "0"
```

To gwarantuje usunięcie tylko jednego wpisu. ID nie powtarza się w żadnym innym rekordzie. Baza danych na to nie pozwoli.

Udostępnione przez firmę [Netteria.NET](https://netteria.net)

Udostępniając nie zapomnij umieścić linku do serwisu <https://netteria.net>