



Bazy danych w C#

Celem lekcji jest poruszenie tematu baz danych oraz wykonanie komend SQL przy pomocy języka programowania C#.



Przypomnienie



- **Rekord** - służy do przechowywania niemutowalnych (niezmienialnych) obiektów.
- **Słownik** - kolekcja przechowująca pary klucz wartość. Klucze w słownikach muszą być unikalne.
- **Hashsetu** używamy - kiedy chcemy trzymać unikalne elementy albo kiedy chcemy szybko sprawdzić czy element już istnieje w kolekcji.
- Czym się różni suma zbioru od różnicy i części wspólnej?
 - **Suma** zbiorów łączy dwa zbiory, zawierając wszystkie unikalne elementy.
 - **Różnica** zbiorów zawiera elementy z jednego zbioru, które nie są obecne w drugim zbiorze.
 - **Część wspólna** zbiorów zawiera elementy obecne w obu zbiorach.





- **SQL** (Structured Query Language) jest językiem zapytań służącym do zarządzania i manipulacji danymi w relacyjnych bazach danych. Używany jest do tworzenia, modyfikowania, zarządzania i przeszukiwania baz danych.



SELECT



- **Select** jest poleceniem używanych do wybierania z bazy danych.

SQL Statement:

[Get your own SQL server](#)

```
SELECT * FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blaiss Sea Delicatessen	Hanna Moos	Fraustorfer Str. 57	Mannheim	68306	Germany



CREATE TABLE



- **Create Table** jest poleceniem używanym do utworzenia nowej tabeli wewnątrz bazy danych.

```
1 -- create a table
2 CREATE TABLE students (
3   id INTEGER PRIMARY KEY,
4   name TEXT NOT NULL,
5   gender TEXT NOT NULL
6 );
```



INSERT



- **Insert** używamy do dodawania nowych rekordów do tabeli.

myCompiler Polski ▼ 🌙 Niedawne Logowanie

Wpisz tytuł...

SQL ▼

Uruchom

```
1 -- create a table
2 CREATE TABLE students (
3   id INTEGER PRIMARY KEY,
4   name TEXT NOT NULL,
5   gender TEXT NOT NULL
6 );
7 -- insert some values
8 INSERT INTO students VALUES (1, 'Ryan', 'M');
9 INSERT INTO students VALUES (2, 'Joanna', 'F');
10 INSERT INTO students VALUES (3, 'Gigant', 'M');
11 -- fetch some values
12 SELECT * FROM students;
```

Wyjście programu

1|Ryan|M
2|Joanna|F
3|Gigant|M

[Execution complete with exit code 0]



DELETE



- **Delete** używamy do usuwania rekordów z tabeli.

```
1 -- create a table
2 CREATE TABLE students (
3   id INTEGER PRIMARY KEY,
4   name TEXT NOT NULL,
5   gender TEXT NOT NULL
6 );
7 -- insert some values
8 INSERT INTO students VALUES (1, 'Ryan', 'M');
9 INSERT INTO students VALUES (2, 'Joanna', 'F');
10 INSERT INTO students VALUES (3, 'Gigant', 'M');
11 -- fetch some values
12
13 DELETE FROM students WHERE id=2;
14 SELECT * FROM students;
```

Wyjście programu

```
1|Ryan|M
3|Gigant|M
```

[Execution complete w:



WHERE



- **Where** używamy do filtrowania rekordów na podstawie warunków.



```
1 -- create a table
2 CREATE TABLE students (
3   id INTEGER PRIMARY KEY,
4   name TEXT NOT NULL,
5   gender TEXT NOT NULL
6 );
7 -- insert some values
8 INSERT INTO students VALUES (1, 'Ryan', 'M');
9 INSERT INTO students VALUES (2, 'Joanna', 'F');
10 INSERT INTO students VALUES (3, 'Gigant', 'M');
11 -- fetch some values
12 SELECT * FROM students WHERE name='Gigant';
```

Wyjście programu

```
3|Gigant|M
```

```
[Execution complete w:
```



ORDER BY



- **Order By** używamy do sortowania wyników zapytania w porządku rosnącym lub malejącym.

SQL ▼



```
1 -- create a table
2 CREATE TABLE students (
3   id INTEGER PRIMARY KEY,
4   name TEXT NOT NULL,
5   gender TEXT NOT NULL
6 );
7 -- insert some values
8 INSERT INTO students VALUES (1, 'Ryan', 'M');
9 INSERT INTO students VALUES (2, 'Joanna', 'F');
10 INSERT INTO students VALUES (3, 'Gigant', 'M');
11 -- fetch some values
12 SELECT * FROM students ORDER BY name;
```

Wyjście programu

```
3|Gigant|M
2|Joanna|F
1|Ryan|M
```

[Execution complete wi

