

Obsługa bazy danych w PHP

Spis treści

| | |
|--|----|
| Opis możliwości połączeń | 1 |
| Połączenie z wybraną bazą danych | 2 |
| Zapytania do bazy danych | 3 |
| Biblioteka PDO | 11 |

Opis możliwości połączeń

W celu połączenia się z bazą danych z użyciem PHP możemy teoretycznie skorzystać z trzech możliwych tzw. rozszerzeń (czyli bibliotek) komunikacyjnych. Pierwszą z nich jest `mysql` (biblioteka jednak wycofana od PHP w wersji 7.0.0), `mysqli` (gdzie `i` oznacza improved, czyli wersję poprawioną, aktualną) albo możemy użyć rozszerzenia o nazwie PDO - PHP Data Objects (w pełni obiektowego).

API (ang. Application Programming Interface) - interfejs klas, metod, funkcji, zmiennych, parametrów, których aplikacja PHP używa w celu zrealizowania zaplanowanych przez programistę zadań (w naszym kontekście chodzi o komunikację z bazą danych).

Interfejs API może być albo `proceduralny` albo `obiektyowy`. Jeżeli jest `proceduralny`, to oznacza to, że operacje bazodanowe realizowane są przez odpowiednio przygotowane i wywołane `funkcje`. Każda funkcja wykonuje unikalne, dające się jednoznacznie wyróżnić działanie związane z bazą danych. Parametry przekazujemy do funkcji jako argumenty w nawiasie, w momencie ich wywołania.

Jeśli zaś interfejs API jest `obiektyowy`, to oznacza to, iż w aplikacji są tworzone `obiekty` (na podstawie "przepisu" `klas`), a operacje bazodanowe realizowane są poprzez `metody` (czyli `funkcje wewnątrz klas`) wywoływane na rzecz tychże stworzonych obiektów. Oprócz `metod`, obiekty mogą posiadać przypisane `atrybuty` (parametry, właściwości).

Rozszerzenie `mysql`

Dodatek wprowadzony w wersji 2.0 specyfikacji języka PHP, został zdeprecjonowany od PHP 5.5.0 oraz całkowicie usunięty od PHP 7.0.0. Nie zaleca się stosowania `mysql` we współczesnych projektach - zamiast tej biblioteki powinniśmy użyć `mysqli` lub PDO. Biblioteka posiada interfejs jedynie `proceduralny`, brak interfejsu `obiektyowego`.

Rozszerzenie `mysqli` (*i = ang. improved*)

Dodatek wprowadzony w wersji 5.0 specyfikacji języka PHP, usprawnił i zaktualizował komunikację do tej pory realizowaną biblioteką `mysql`. To rozszerzenie oferuje zarówno API `proceduralne` jak i `obiektywne`. W pełni aktualne, współczesne, zalecane do użycia w nowych projektach. Aczkolwiek w odróżnieniu od dodatku PDO (który potrafi obsługiwać różne silniki bazodanowe), `mysqli` współpracuje tylko z bazami MySQL.

Rozszerzenie PDO (*ang. PHP Data Objects*)

Dodatek wprowadzony również w wersji 5.0 specyfikacji języka PHP, zrealizowany w pełni `obiektywo`. W pełni aktualne, współczesne, zalecane do użycia w nowych projektach. Jako jedyne rozszerzenie potrafi oprócz MySQL obsługiwać różne silniki bazodanowe, w tym m.in.: PostgreSQL, Oracle, MS SQL, SQLite, IBM DB2, Firebird i wiele innych.¹

Przyjrzyjmy się teraz porównaniu rozszerzeń w postaci tabelarycznej:

| Rozszerzenie PHP | mysql | mysqli | PDO |
|---|---|--------------------|--------------------|
| Wprowadzono w wersji PHP | 2.0 | 5.0 | 5.0 |
| Wsparcie dla rozszerzenia | zdeprecjonowane od PHP 5.5.0, usunięte od PHP 7.0.0 | aktywnie wspierane | aktywnie wspierane |
| Zalecane dla nowych projektów | nie | tak | tak |
| Interfejs proceduralny | tak | tak | nie |
| Interfejs obiektowy | nie | tak | tak |
| Użycie różnych silników (driverów) SQL | nie | nie | tak |
| Wsparcie funkcji MySQL 5.1+ | nie | tak | większość |
| Prepared Statements po stronie serwera | nie | tak | tak |
| Prepared Statements po stronie klienta | nie | nie | tak |

Połączenie z wybraną bazą danych

Bazy danych są zbiorem danych dla aplikacji internetowych. Można z nich pobrać informacje, które są prezentowane użytkownikowi strony. Do obsługi baz danych są wykorzystywane systemy zarządzania bazami danych. Najpopularniejszym systemem współpracującym z PHP jest MySQL.

Aby możliwa była współpraca języka PHP z systemem MySQL, musi istnieć baza danych, z którą można się połączyć.

Proces komunikacji z MySQL jest podzielony na kilka etapów:

- nawiązanie połączenia z MySQL i wybór bazy danych,
- utworzenie zapytania i jego wykonanie,
- otrzymanie rezultatów i wyświetlenie ich na stronie,
- rozłączenie z MySQL.

Do połączenia z bazą danych służy funkcja `mysqli_connect()`. Ma ona cztery argumenty: nazwę hosta, nazwę użytkownika, hasło użytkownika i nazwę bazy danych, której będziemy używali. Polecenie to jednocześnie dokonuje połączenia z serwerem bazy danych oraz wybiera bazę, z której będą pobierane dane.

```
$połączenie=mysqli_connect('localhost', 'root', '', 'sklep');
```

Nazwa hosta oznacza nazwę lub adres IP serwera, na którym umieszczono bazę danych.

Uwaga: Jeżeli użyjemy operatora `@` przed funkcją, to wypadku problemu z połączeniem nie zostanie wyświetlone ostrzeżenie wygenerowane przez PHP.

Do zakończenia połączenia z bazą danych służy funkcja `mysqli_close()` w postaci:

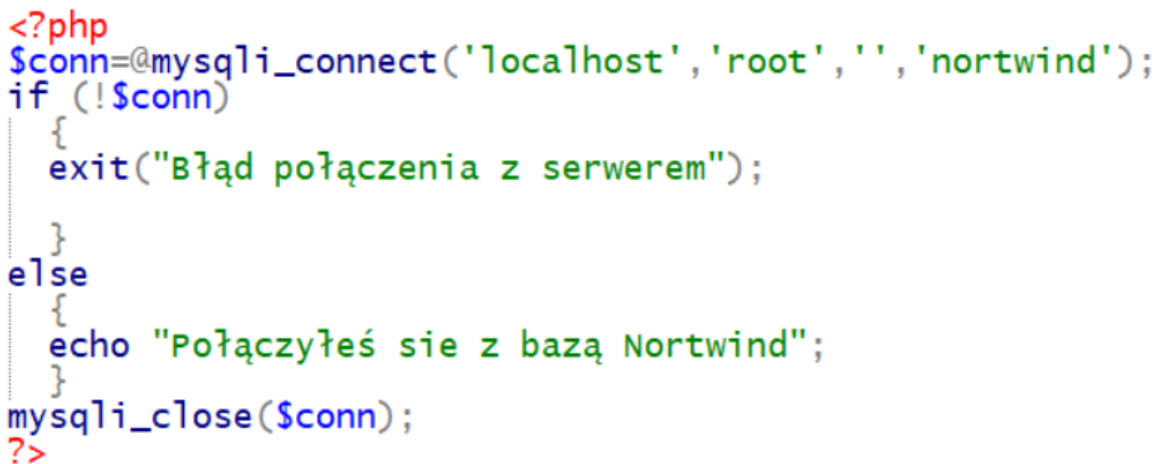
mysqli_close ([identyfikator_połączenia]).

Funkcja zwróci wartość TRUE, gdy operacja zamknięcia połączenia zakończy się powodzeniem.

Poniżej prezentuje się ten sam skrypt pokazany jako zdjęcie i jako tekst:

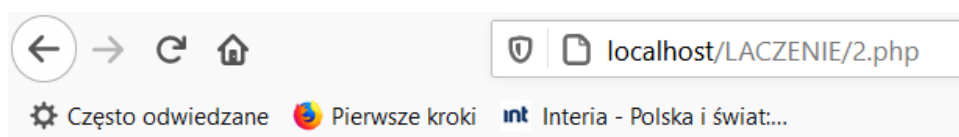
Przykład 2.php

```
<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{
    echo "Połączyłeś się z bazą Nortwind";
}
mysqli_close($conn);
?>
```



```
<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{
    echo "Połączyłeś się z bazą Nortwind";
}
mysqli_close($conn);
?>
```

Rysunek 1. Połączenie z serwerem bazy danych oraz zamknięcie połączenia



Połączyłeś się z bazą Nortwind

Po wywołaniu funkcji `mysqli_connect()` sprawdzane jest, czy zostało nawiązane połączenie. Jeśli tak, to wyświetli się komunikat „Połączyłeś się z bazą Nortwind”, w przeciwnym wypadku otrzymamy komunikat „Błąd połączenia z serwerem”. Na końcu nastąpiło zakończenie połączenia z bazą Nortwind.

Zapytania do bazy danych

Po nawiązaniu połączenia z wybraną bazą danych można wysyłać do niej zapytania. W tym celu stosujemy funkcję `mysqli_query()` w następującej postaci:

mysqli_query(identyfikator_połączenia, 'zapytanie')

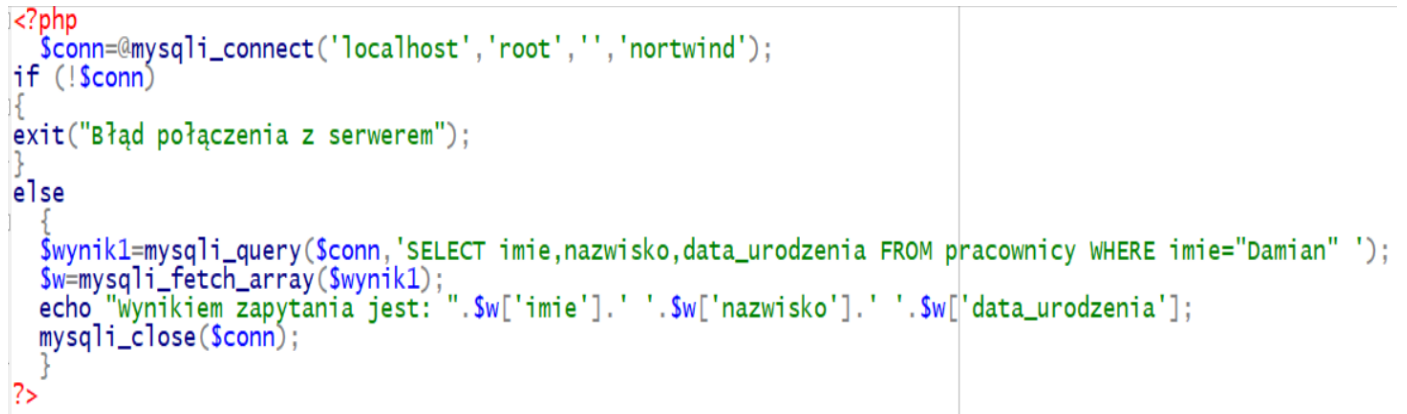
Zwracane wartości zależą od rodzaju zapytania. Jeśli zapytanie pobierało dane, to funkcja zwraca identyfikator do tych danych. Jeśli zapytanie nie pobiera danych, to funkcja zwraca wartość TRUE, w przeciwnym razie zwraca wartość FALSE.

Dla zapytań pobierających dane zwracany jest identyfikator zasobów. Do odczytania tego typu danych

można zastosować funkcję `mysqli_fetch_array ()`.

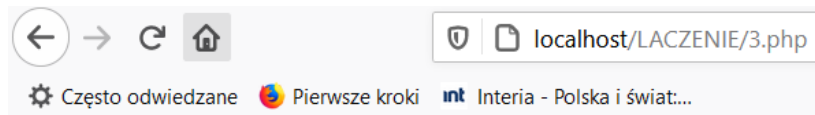
Przykład 3.php

```
<?php
    $conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
exit("Błąd połączenia z serwerem");
}
else
{
    $wynik1=mysqli_query($conn,'SELECT imie,nazwisko,data_urodzenia FROM
pracownicy WHERE imie="Damian" ');
    $w=mysqli_fetch_array($wynik1);
    echo "Wynikiem zapytania jest: ".$w['imie'].' '.$w['nazwisko'].'
'.$w['data_urodzenia'];
    mysqli_close($conn);
}
?>
```



```
<?php
    $conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
exit("Błąd połączenia z serwerem");
}
else
{
    $wynik1=mysqli_query($conn,'SELECT imie,nazwisko,data_urodzenia FROM pracownicy WHERE imie="Damian" ');
    $w=mysqli_fetch_array($wynik1);
    echo "Wynikiem zapytania jest: ".$w['imie'].' '.$w['nazwisko'].' '.$w['data_urodzenia'];
    mysqli_close($conn);
}
?>
```

Rysunek 2. Zwrócenie pojedynczego wiersza z zapytania



Wynikiem zapytania jest: Damian Biel 1977-12-09

Polecenie `mysqli_query()` wysłało zapytanie do bazy nortwind - wypisz imię, nazwisko i datę urodzenia pracownika o imieniu Damian. Zmienna `$w` jest tablicą asocjacyjną przechowującą pobrane dane z zapytania. Kolejnymi elementami tej tablicy są `$w [' imie ']`, `$w [' nazwisko ']` oraz `$w [' data_urodzenia']`.

Aby odczytać całą zawartość tabeli, należy wywołać funkcję w pętli. Gdy wszystkie dane zostaną odczytane, funkcja zwróci wartość `FALSE`.

Przykład 4.php

```

<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{

    $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by
nazwisko";
    $result=mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

    while($row = mysqli_fetch_array($result))
    {
        echo $row['imie']." ".$row['nazwisko']." ".$row['data_urodzenia']."<br>";
    }
    mysqli_close($conn);
}
?>
<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{

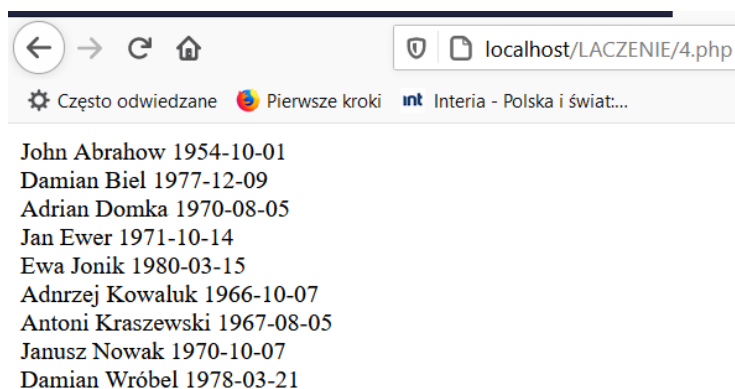
    $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by nazwisko";
    $result=mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

    while($row = mysqli_fetch_array($result))
    {
        echo $row['imie']." ".$row['nazwisko']." ".$row['data_urodzenia']."<br>";
    }
    mysqli_close($conn);
}
?>

```

Rysunek 3. Zastosowanie pętli do wyświetlenia zawartości tabeli

Skrypt wyświetli zawartość tablicy asocjacyjnej **\$row** w następującej postaci:



```

John Abraham 1954-10-01
Damian Biel 1977-12-09
Adrian Domka 1970-08-05
Jan Ewer 1971-10-14
Ewa Jonik 1980-03-15
Adnrzej Kowaluk 1966-10-07
Antoni Kraszewski 1967-08-05
Janusz Nowak 1970-10-07
Damian Wróbel 1978-03-21

```

Rysunek 4. Wynik zastosowania pętli do wyświetlenia zawartości tabeli

Do kolejnych przydatnych funkcji pracujących na bazie danych należy funkcja

mysqli_num_rows() . Zwraca ona liczbę wierszy znajdujących się w wyniku zapytania. Wartość tę można użyć do zbudowania pętli **FOR** odczytującej kolejne wiersze zwrócone przez zapytanie.


Przykład 5.php

```
<?php
    $conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
    {
    exit("Błąd połączenia z serwerem");
    }
    else
    {

        $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy
order by nazwisko";
        $result=mysqli_query($conn,$query1)
        or die('Błędne zapytanie');

        $ilosc=mysqli_num_rows($result);

        for($i=0;$i<$ilosc;$i++)
            {
                $row = mysqli_fetch_array($result);
                echo $row['imie']." ".$row['nazwisko']."
".$row['data_urodzenia']."<br>";
            }
        mysqli_close($conn);
    }
?>
```



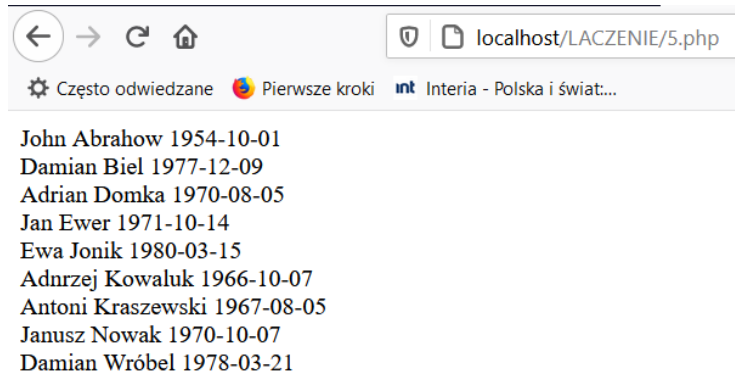
```
<?php
    $conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
    {
    exit("Błąd połączenia z serwerem");
    }
    else
    {

        $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by nazwisko";
        $result=mysqli_query($conn,$query1)
        or die('Błędne zapytanie');

        $ilosc=mysqli_num_rows($result);

        for($i=0;$i<$ilosc;$i++)
            {
                $row = mysqli_fetch_array($result);
                echo $row['imie']." ".$row['nazwisko']." ".$row['data_urodzenia']."<br>";
            }
        mysqli_close($conn);
    }
?>
```

Rysunek 5. Zastosowanie funkcji `mysqli_num_rows()`



The screenshot shows a web browser window with the address bar displaying 'localhost/LACZENIE/5.php'. Below the address bar, there are several browser tabs and a search bar. The main content of the page is a list of names and birth dates:

- John Abraham 1954-10-01
- Damian Biel 1977-12-09
- Adrian Domka 1970-08-05
- Jan Ewer 1971-10-14
- Ewa Jonik 1980-03-15
- Adnrzej Kowaluk 1966-10-07
- Antoni Kraszewski 1967-08-05
- Janusz Nowak 1970-10-07
- Damian Wróbel 1978-03-21

Skrypt wyświetla te same wyniki co skrypt z przykładu rysunek 4. Tym razem została zastosowana funkcja **mysqli_num_rows ()** zliczająca rekordy, by wyznaczyć liczbę powtórzeń w pętli.

Wyniki z przykładu rysunek 4 mogą zostać wyświetlone w postaci tabeli lub listy numerowanej albo wypunktowanej.

Wyświetlenie wyników w postaci tabeli:

Przykład 6.php

```
<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{

    $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by
nazwisko";
    $result=mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

    $ilosc=mysqli_num_rows($result);

    echo '<table border="1">';
    echo '<tr><th>Imie</th><th>Nazwisko</th><th>Data urodzenia</th></tr>';

    for($i=0;$i<$ilosc;$i++)
    {
        $row = mysqli_fetch_array($result);
        echo
"<tr><td>".$row['imie']. "</td><td>".$row['nazwisko']. "</td><td>".$row['data_urodzenia'
]. "</td></tr>";
    }
        echo '</ table >';
        mysqli_close($conn);
    }
?>
```

```

<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{

    $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by nazwisko";
    $result=mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

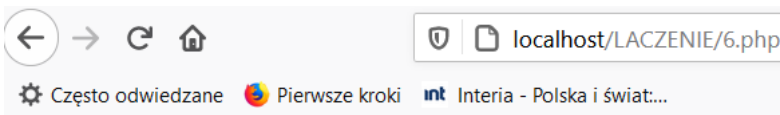
    $ilosc=mysqli_num_rows($result);

    echo '<table border="1">';
    echo '<tr><th>Imie</th><th>Nazwisko</th><th>Data urodzenia</th></tr>';

    for($i=0;$i<$ilosc;$i++)
    {
        $row = mysqli_fetch_array($result);
        echo "<tr><td>".$row['imie']."</td><td>".$row['nazwisko']."</td><td>".$row['data_urodzenia']."</td></tr>";
        echo '</ table >';
    }
    mysqli_close($conn);
}
?>

```

Rysunek 6. Wyświetlenie wyników zapytania w postaci tabeli



| Imie | Nazwisko | Data urodzenia |
|---------|------------|----------------|
| John | Abraham | 1954-10-01 |
| Damian | Biel | 1977-12-09 |
| Adrian | Domka | 1970-08-05 |
| Jan | Ewer | 1971-10-14 |
| Ewa | Jonik | 1980-03-15 |
| Adnrzej | Kowaluk | 1966-10-07 |
| Antoni | Kraszewski | 1967-08-05 |
| Janusz | Nowak | 1970-10-07 |
| Damian | Wróbel | 1978-03-21 |

Wyświetlenie wyników w postaci listy:

Przykład 7.php

```

<?php
    $conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{

    $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by
nazwisko";
    $result=mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

    $ilosc=mysqli_num_rows($result);

    echo '<ol>';
    echo '<tr><th>Imie</th><th>Nazwisko</th><th>Data urodzenia</th></tr>';

    for($i=0;$i<$ilosc;$i++)
    {

```



```

        $row = mysqli_fetch_array($result);
        echo "<li>".$row['imie']." ".$row['nazwisko']."
".$row['data_urodzenia']."</li>";
    }
    echo '</ ol >';
    mysqli_close($conn);
}
?>

```

```

<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{
    $query1="SELECT imie,nazwisko,data_urodzenia,adres FROM pracownicy order by nazwisko";
    $result=mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

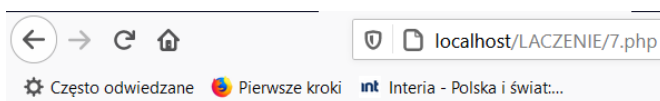
    $ilosc=mysqli_num_rows($result);

    echo '<ol>';
    echo '<tr><th>Imie</th><th>Nazwisko</th><th>Data urodzenia</th></tr>';

    for($i=0;$i<$ilosc;$i++)
    {
        $row = mysqli_fetch_array($result);
        echo "<li>".$row['imie']." ".$row['nazwisko']." ".$row['data_urodzenia']."</li>";
    }
    echo '</ ol >';
    mysqli_close($conn);
}
?>

```

Rysunek 7. Wyświetlenie wyników w postaci listy.



| Imie | Nazwisko | Data urodzenia |
|------|-------------------|----------------|
| 1. | John Abraham | 1954-10-01 |
| 2. | Damian Biel | 1977-12-09 |
| 3. | Adrian Domka | 1970-08-05 |
| 4. | Jan Ewer | 1971-10-14 |
| 5. | Ewa Jonik | 1980-03-15 |
| 6. | Adnrzej Kowaluk | 1966-10-07 |
| 7. | Antoni Kraszewski | 1967-08-05 |
| 8. | Janusz Nowak | 1970-10-07 |
| 9. | Damian Wróbel | 1978-03-21 |

Rysunek 8. Lista pracowników.

Dla zapytań modyfikujących funkcja **mysqli_query()** zwraca wartość TRUE lub FALSE.

Funkcja **mysqli_affected_rows()** wyświetla liczbę zmodyfikowanych rekordów w tabeli. Funkcja ma postać:

mysqli_affected_rows([identyfikator])

Uwaga: Jeśli w wywołaniu funkcji zostanie pominięty identyfikator połączenia z serwerem, to działanie funkcji będzie dotyczyło ostatnio otwartego połączenia.

Do tabeli **pracownicy** w bazie **nortwind** zostanie dodany jeden rekord:

Przykład 8.php

```
<?php
```

```

    $conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn) {
    exit("Błąd połączenia z serwerem");}
    else{
$query1="INSERT INTO pracownicy (imie, nazwisko, data_urodzenia)
VALUES ( 'Katarzyna', 'Dymna', '1980-11-15')";
mysqli_query($conn,$query1)
    or die('Błędne zapytanie');
    $ile_rekordow=mysqli_affected_rows($conn);
    echo "Liczba dodanych rekordów: $ile_rekordow";
    mysqli_close($conn);
    }
?>

```

```

<?php
$conn=@mysqli_connect('localhost','root','','nortwind');
if (!$conn)
{
    exit("Błąd połączenia z serwerem");
}
else
{
    $query1="INSERT INTO pracownicy (imie, nazwisko, data_urodzenia) VALUES ( 'Katarzyna', 'Dymna',
'1980-11-15')";
    mysqli_query($conn,$query1)
    or die('Błędne zapytanie');

    $ile_rekordow=mysqli_affected_rows($conn);

    echo "Liczba dodanych rekordów: $ile_rekordow";

    mysqli_close($conn);
}
?>

```

Rysunek 9. Do tabeli pracownicy w bazie nortwind zostanie dodany jeden rekord.

localhost/LACZENIE/8.php

Często odwiedzane Pierwsze kroki int Interia - Polska i świat...

Liczba dodanych rekordów: 1

✓ Pokazano wiersze 0 - 9 (10 ogółem, Wykonanie zapytania trwało 0,0000 sekund(y).)

SELECT * FROM 'pracownicy'

Profilowanie [Edytuj w linii] [Edytuj] [Wyjaśnij SQL] [Utw]

Pokaż wszystko | Liczba wierszy: 25 | Filtrowanie wierszy: Przeszukaj tę tabelę | Sortuj wg kłucza: Żaden

| | id_pracownik | imie | nazwisko | id_odzial | data_urodzenia | data_zatrudnienia | miasto | telefon | kraj | adres |
|--------------------------|--------------|-----------|------------|-----------|----------------|-------------------|---------|-----------|--------|----------------------|
| <input type="checkbox"/> | 1 | John | Abraham | 8 | 1954-10-01 | 1972-02-08 | Rzeszów | 787474747 | USA | NY,NEW STREET 7H |
| <input type="checkbox"/> | 2 | Ewa | Jonik | 1 | 1980-03-15 | 2003-08-01 | Kraków | 123636369 | Polska | ul. Różana 67 |
| <input type="checkbox"/> | 3 | Adnrzej | Kowaluk | 1 | 1966-10-07 | 1987-10-29 | Rzeszów | 1812545 | Polska | ul. Podkarpacka 6/45 |
| <input type="checkbox"/> | 4 | Damian | Biel | 13 | 1977-12-09 | 1999-10-29 | Leżajsk | 17224242 | Polska | ul. podkarpacka 8/2 |
| <input type="checkbox"/> | 5 | Antoni | Kraszewski | 11 | 1967-08-05 | 1983-10-29 | Łańcut | 178585877 | Polska | ul. Warszawska 8/7 |
| <input type="checkbox"/> | 6 | Janusz | Nowak | 4 | 1970-10-07 | 1992-10-21 | Rzeszów | 178785454 | Polska | ul. Hetmańska 111 |
| <input type="checkbox"/> | 7 | Jan | Ewer | 7 | 1971-10-14 | 1992-12-21 | Berlin | 2547474 | Niemcy | |
| <input type="checkbox"/> | 8 | Damian | Wróbel | 2 | 1978-03-21 | 1999-10-05 | Krosno | 798787874 | Polska | |
| <input type="checkbox"/> | 9 | Adrian | Domka | 1 | 1970-08-05 | 1992-10-21 | Rzeszów | 182525252 | Polska | |
| <input type="checkbox"/> | 10 | Katarzyna | Dymna | 0 | 1980-11-15 | 0000-00-00 | | | Polska | |

Rysunek 10. Dodany rekord.