Bazy danych II

Andrzej Grzybowski

Instytut Fizyki, Uniwersytet Śląski

Wykład 12

Zastosowanie PHP do programowania aplikacji baz danych MySQL

Wsparcie programowania w PHP baz danych MySQL

- Obsługa baz danych MySQL, od dawna dostępna w PHP, jest jednym z najpopularniejszych zastosowań tego języka. Te narzędzia, ze względu na darmowy dostęp do nich, są bardzo często wykorzystywane do programowania witryn WWW, które wymagają wsparcia przez bazy danych lub których możliwości wzrastają dzięki zastosowaniu baz danych.
- Oczywiście można próbować korzystać w skryptach PHP z baz danych MySQL poprzez ODBC, ale podobnie jak w przypadku Oracle'a znacznie bardziej uzasadnione jest wykorzystanie rozszerzenia PHP, specjalnie przeznaczonego do obsługi baz danych MySQL, które udostępnia zestaw specyficznych funkcji wspierających różne transakcje na tego typu bazach danych.

Wsparcie programowania w PHP baz danych MySQL

- Dokonując samodzielnej kompilacji PHP, należy podać opcję konfiguracyjną:
  - –with–mysql,

  która umożliwi PHP dostęp do baz danych MySQL. Jeżeli wybierzemy tę opcję bez podania ścieżki dostępu do MySQL, to PHP wykorzysta własne biblioteki klienta MySQL.

  Jeżeli podamy powyższą opcję w wersji:
  - –with–mysql=/sciezka_dostepu_do_MySQL,

  to wymusi na PHP użycie bibliotek zainstalowanych przez MySQL.
- Na przykład w przypadku systemów operacyjnych Unix lub Linux należy dążyć do kompilacji PHP przy opcji:
  - –with–mysql=/usr

  zamiast stosować bibliotekę systemową mysqlclient.

Wsparcie programowania w PHP baz danych MySQL

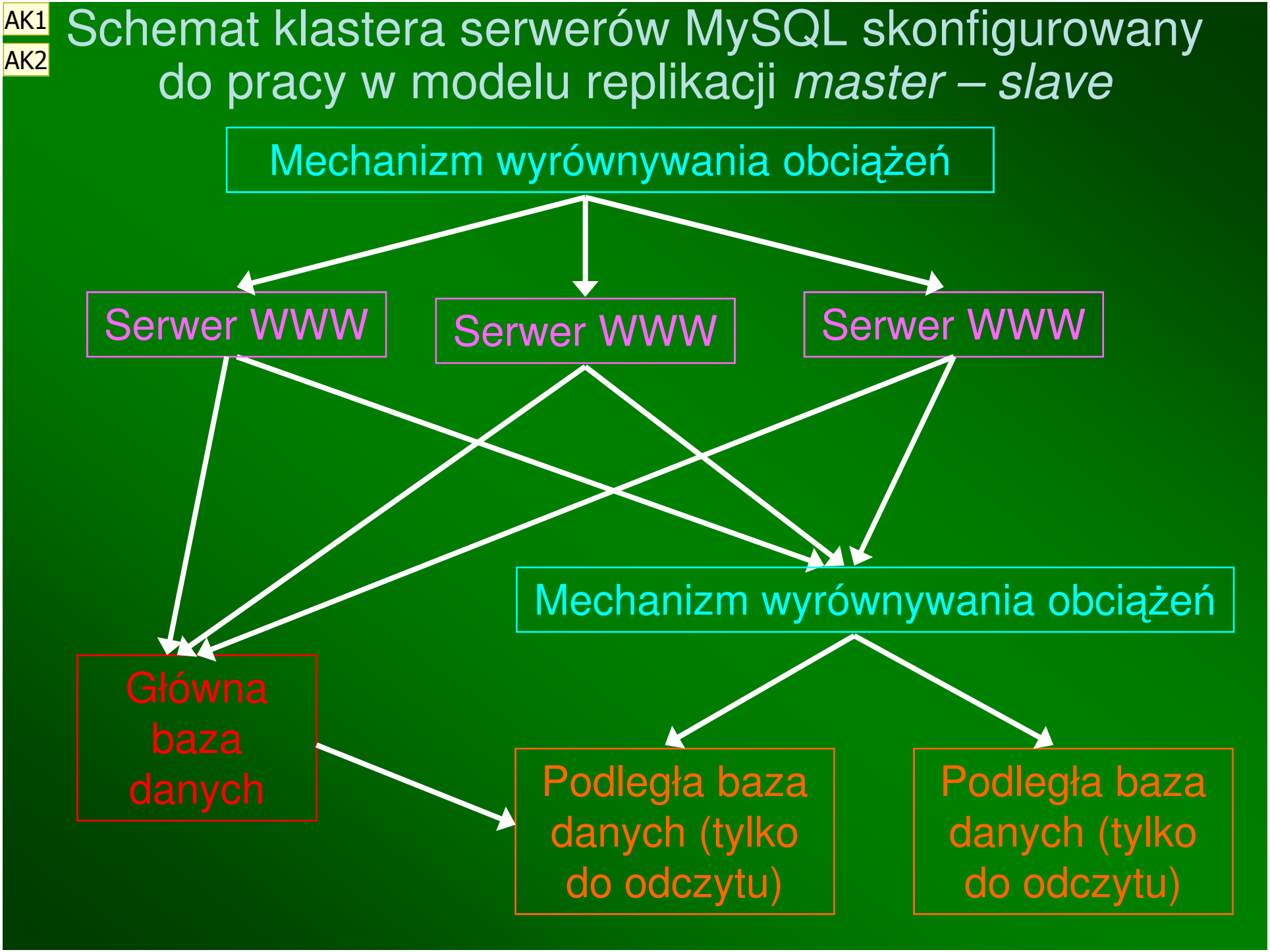
- Jednak koncepcje obsługi baz danych MySQL w PHP 4 i PHP 5 są różne:
  - W PHP 4 obsługa MySQL jest zawsze włączona i dlatego nawet jeżeli nie podamy opcji – –with–mysql, to zostanie użyta wbudowana biblioteka PHP (np. pod Windows będzie to libmysql.dll).
  - Natomiast w PHP 5 obsługa MySQL jest domyślnie wyłączona, aby ją włączyć należy użyć jednego z dwóch sposobów:
    - ✓ skompilować PHP z opcją – –with–mysql lub – –with–mysql=/sciezka_dostepu_do_MySQL
    - ✓ włączyć rozszerzenia PHP obsługujące MySQL w pliku konfiguracyjnym php.ini

Wsparcie programowania w PHP baz danych MySQL

- Na przykład PHP 5 przeznaczone dla systemu operacyjnego MS Windows zawiera także bibliotekę libmysql.dll, ale obsługa baz danych MySQL wymaga włączenia w pliku php.ini rozszerzeń:
  - php_mysql (zawartego w bibliotece php_mysql.dll), które oferuje podstawowy zestaw funkcji obsługi baz danych MySQL
  - i ewentualnie php_mysqli (zawartego w bibliotece php_mysqli.dll), które udostępnia nowe możliwości, jakie dają ostatnie wersje MySQL (4.1 i wyżej), np.
    - ✓ realizację zapytań z parametrami wiązanymi, chociaż sam MySQL 4.1 tego nie potrafi
    - ✓ efektywną, bo opartą na wbudowanych funkcjach MySQL, dystrybucję zapytań do baz danych typu *master – slave* (główny – podległy), w których podległa baza służy tylko do odczytu, a do głównej bazy danych dozwolony jest również zapis.

AK1
AK2

Schemat klastera serwerów MySQL skonfigurowany do pracy w modelu replikacji *master – slave*



Slajd 7

AK1 Alternatywą replikacji jest partycjonowanie (odpowiednie dla aplikacji silnie wykorzystujących zapis) jest podział jednego schematu logicznego pomiędzy kilka fizycznych baz danych według klucza głównego. Należy w tym modelu unikać zapytań korzystających z kilku baz danych.
Grzybowscy; 2005-05-29

AK2 Są dwa rodzaje replikacji:
1) master-master (główny-główny), który nie korzysta z współdzielonej pamięci masowej; w tym modelu ponoszone są koszty związane z synchronizacją transakcji oraz ich dwufazowym zatwierdzaniem w sieci (i dodatkowo kwestie spójność odczytu). Każda kopia bazy danych w tym nodelu musi być całkowicie zsynchronizowana z wszystkimi pozostałymi.
2) master-slave (główny-podległy) - tutaj aktualizacja często nie zachodzą nawet w czasie rzeczywistym, co najczęściej znacznie przyspiesza. Podstawową trudnością w tym modelu jest konieczność rozdzielenia operacji odczytu i zapisu.
Grzybowscy; 2005-05-29

Typowy schemat korzystania z bazy danych przez aplikację zostanie omówiony w ramach rozszerzenia PHP_mySQL

Podstawowe funkcje rozszerzenia PHP_MySQL

- Połączenie z bazą danych jest rozdzielone na dwa etapy:
  - połączenie z serwerem MySQL
  - opcjonalny wybór domyślnej bazy danych na tym serwerze

  Pierwszy etap może być realizowane przez funkcje:

  - mysql_connect(string host[:port], string użytkownik, string hasło, boolean nowe_połączenie, integer opcje)

    – tworzy (zwykłe) połączenie z serwerem MySQL, które jest automatycznie zamykane po zakończniu skryptu, ale może też być zamknięte przez funkcję mysql_close

  - mysql_pconnect(string host[:port], string użytkownik, string hasło, integer opcje)

    – tworzy trwałe (persistent) połączenie z serwerem MySQL, które nie jest zamykane z chwilą zakończenia skryptu, ale trwa tak długo, jak długo trwa proces serwera bazy danych.

Podstawowe funkcje rozszerzenia PHP_MySQL

- Wszystkie argumenty obydwu funkcji tworzących połączenie z bazą danych MySQL są opcjonalne. Domyślnie przyjmowane są:
  - ✓ host lokalny
  - ✓ nazwa użytkownika wykonującego skrypt, którym najczęściej jest *nobody*, czyli serwer sieciowy
  - ✓ łańcuch pusty dla hasła
  - ✓ brak argumentu nowe_połączenie funkcji mysql_connect
  - ✓ brak opcji
- Próba utworzenia istniejącego połączenia nie uda się.
- Argument nowe_połączenie musi być ustawiony na TRUE w celu nawiązania nowego połączenia.
- Jeżeli PHP nie potrafi nawiązać połączenia z bazą, to obydwie funkcje zwracają wartość FALSE.

Podstawowe funkcje rozszerzenia PHP_MySQL

- Argument opcje wpływa na charakter połączenia i może przyjmować następujące predefiniowne wartości oraz złożone z nich wyrażenia logiczne:

  MYSQL_CLIENT_COMPRESS – wymusza kompresję komunikacji pomiędzy serwerem a klientem

  MYSQL_CLIENT_IGNORE_SPACE – nakazuje serwerowi MySQL ignorowanie spacji po nazwach funkcji

  MYSQL_CLIENT_INTERACTIVE – stosuje tryb interaktywny, a nie oczekiwania, podczas oczekiwania na przekroczenie czasu operacji

  MYSQL_CLIENT_SSL – szyfruje komunikację pomiędzy klientem a serwerem za pomocą protokołu SSL

- Jeżeli użytkownik, który uzyskał połączenie z serwerem MySQL ma dostęp do więcej niż jedna bazy danych należy wybrać domyślną bazę danych, używając instrukcji USE MySQL lub funkcji rozszerzenia PHP_MySQL:

  mysql_select_db(string baza_danych,resource połączenie),

  która zwraca wartość logiczną zależną od powodzenia operacji.

Podstawowe funkcje rozszerzenia PHP_MySQL

- Tworzenie zapytania / polecenia przy założeniu wykorzystania rozszerzenia PHP_MySQL umożliwia jedynie realizację zapytań / poleceń
  - bez parametrów (zmiennych związanych), np.

    $Query = ”SELECT * FROM emp”;
- Interpretowanie (parsowanie) zapytania / polecenia i wykonanie zapytania / polecenia jest realizowane przez jedną z dwóch funkcji zwracających zasoby będące wynikami:
  - mysql_query(string zapytanie, resource połączenie, integer buforowanie_wyników), która w przypadku poleceń z grupy DML (insert,update,delete) zwraca wartość logiczną zależną od powodzenia operacji
  - mysql_unbuffered_query(string zapytanie, resource połączenie, integer buforowanie_wyników)

Podstawowe funkcje rozszerzenia PHP_MySQL

- Obydwie funkcja interpretują i wykonują zapytanie w ramach połączenia, a jeśli ten argument jest pominięty, to domyślnie użyte jest ostatnio otwarte połączenie.
- Również opcjonalny argument buforowanie_wyników może przyjmować predefiniowane wartości:
  MYSQL_STORE_RESULT – oznacza wymóg buforowania wyników i jest wartością domyślną dla funkcji mysql_query,
  MYSQL_USE_RESULT – oznacza niebuforowaną transmisję wyników i jest wartością domyślną dla funkcji mysql_unbuffered_query.
  Należy pamiętać, że niebuforowana transmisja wyników pozwala odczytywać dane tylko wtedy, gdy są one potrzebne, ale wykonanie kolejnego zapytania na danym połączeniu powoduje utratę wyników zwróconych poprzez to połączenie wcześniej. Natomiast ograniczenie rozmiarów zbioru wyników można uzyskać za pomocą klauzuli LIMIT dostępnej w implementacji SQL zastosowanej w bazie danych MySQL.

Podstawowe funkcje rozszerzenia PHP_MySQL

- Wykorzystanie wyników zapytania / polecenia jest wspierane przez zestaw kilku funkcji, które udostępniają różne sposoby pobierania danych z wyniku zapytania i zapisywania ich w zmiennych PHP:
  - mysql_num_rows(resource wynik) (resource wynik) – zwraca liczbę wierszy w zbiorze wyników
  - mysql_data_seek(resource wynik,integer nr_wiersza)
    – przemieszcza wewnętrzny wskaźnik wierszy wyniku na wiersz o numerze nr_wiersza, licząc wiersze od 0
  - mysql_fetch_row(resource wynik) – zwraca tablicę reprezentującą wszystkie pola wiersza wyników, indeksowaną numerycznie, począwszy od 0. Każde wywołanie powoduje pobranie kolejnego wiersza aż do wyczerpania wszystkich wiersz, o czym funkcja informuje zwracając wartość logiczną FALSE.

Podstawowe funkcje rozszerzenia PHP_MySQL

- mysql_fetch_array(resource wynik,integer typ)
  - zwraca tablicę, która reprezentuje wszystkie pola wiersza wyniku aż do wyczerpania wierszy i wówczas zwraca wartość logiczną FALSE.
    Domyślnie wartość każdego pola jest przechowywana podwójnie w elemencie indeksowanym numerycznie i w innym elemencie indeksowanym nazwą pola.
    Można to zmienić ustawiając opcjonalny argument typ:
    MYSQL_NUM – wyłącznie indeksowanie numeryczne
    MYSQL_ASSOC – tylko indeksowanie nazwami pól
    MYSQL_BOTH – oznacza zachowanie domyślne
- mysql_fetch_assoc(resource wynik) – daje równoważny efekt jak mysql_fetch_array(wynik, MYSQL_ASSOC)
- mysql_fetch_object(resource wynik)
  - zwraca obiekt, którego właściwości reprezentują pola ze zbioru wyników. Każde wywołanie funkcji zwraca kolejny wiersz aż do wyczerpania wierszy i wówczas zwraca wartość logiczną FALSE.

Podstawowe funkcje rozszerzenia PHP_MySQL

- mysql_num_fields(resource wynik)
  - zwraca liczbę pól w zbiorze wyników
- mysql_field_seek(resource wynik, integer nr_pola)
  - przesuwa wewnętrzny wskaźnik pól, numerowanych przez PHP od 0, na pole określone numerem nr_pola
- mysql_fetch_field(resource wynik , integer nr_pola)
  - jest to funkcja zwracająca obiekt, którego właściwości zawierają różne informacje o polu o numerze nr_pola.

    Jeżeli opcjonalny argument nr_pola zostanie pominięty, to zwrócone zostaną informacje o kolejnym polu w zestawie.

Podstawowe funkcje rozszerzenia PHP_MySQL

Właściwości obiektu mysql_fetch_field

Właściwość	Opis
blob	TRUE, jeżeli pole jest dużym obiektem binarnym
max_length	Maksymalna długość
multiple_key	TRUE, jeżeli pole jest kluczem nieunikatowym
name	Nazwa pola (kolumny)
not_null	TRUE, jeżeli pole nie może być puste
numeric	TRUE, jeżeli pole jest numeryczne
primary_key	TRUE, jeżeli pole jest kluczem podstawowym
table	Nazwa tabeli
type	Typ pola
unique_key	TRUE, jeżeli pole jest kluczem unikatowym
unsigned	TRUE, jeżeli pole nie posiada znaku
zerofill	TRUE, jeżeli pole jest wypełnione zerami

Podstawowe funkcje rozszerzenia PHP_MySQL

- Zwolnienie pamięci zajmowanej przez zbiór wyników zapytania / polecenia odbywa się przy pomocy funkcji

  mysql_free_result(resource wynik)

  Funkcja zwalnia pamięć związaną z zasobem wynik utworzonym przy pomocy funkcji mysql_query lub mysql_unbuffered_query.

  Jednak trzeba zauważyć, że w zasadzie nie jest konieczne jej stosowanie, ponieważ po zakończeniu skryptu taka pamięć jest automatycznie zwalniana.

- Zamknięcie połączenia z bazą danych realizowane jest przez funkcję

  mysql_close(resource połączenie)

  Stosowania tej funkcji nie wymagają nietrwałe połączenia, które są zamykane po zakończeniu skryptu.
  Jeżeli opcjonalny parametr połączenie nie występuje, to domyślnie zamykane jest ostatnio otwarte połączenie z bazą danych.

Podstawowe funkcje rozszerzenia PHP_MySQL

- Obsługa błędów w operacjach na bazach danych MySQL jest wspierana przez dwie funkcje:
  - mysql_errno(resource połączenie)
    – zwraca numer błędu, dotyczący ostatniej operacji na bazie danych.
    Jeżeli opcjonalny argument połączenie jest pominięty, to użyte będzie ostatnio otwarte połączenie.
  - mysql_error(resource połączenie)
    – zwraca opis błędu, dotyczący ostatniej operacji na bazie danych.
    Jeżeli opcjonalny argument połączenie jest pominięty, to użyte będzie ostatnio otwarte połączenie.

Przykład zastosowania rozszerzenia PHP_MySQL

```
<?php
    // połączenie z serwerem
    $Connection = mysql_connection(”localhost”,”scott”,”tiger”);
     // wybór domyślnej bazy danych (może być pominięty w przypadku dostępu
     // przez użytkownika scott do tylko jednej bazy danych)
     mysql_select_db(”scottdb”, $Connection);
    // budownie zapytania
    $Query = ”SELECT EMPNO,ENAME,HIREDATE ”;
    $Query .= ”FROM EMP ”;
    $Query .= ”WHERE JOB = ‘CLERK’ ”;
    // interpretacja i wykonanie zapytania
    $Result = mysql_query($Connection,$Query);
    // pobranie wszystkich wierszy wyniku zapytania
    while ($row = mysql_fetch_object($Connection))
    {
       print(”Pracownik o numerze $row–>EMPNO i nazwisku $row–>ENAME ”);
        print(”został zatrudniony $row–>HIREDATE.<br>\n”);
    }
    // zwolenienie pamięci związanej ze zbiorem wyników
    mysql_free_result($Result);
    // zamknięcie połączenia
    mysql_close($Connection);
?>
```