

Bazy danych II

Andrzej Grzybowski

Instytut Fizyki, Uniwersytet Śląski

Wykład 11

Zastosowanie PHP do programowania aplikacji
baz danych Oracle

Wsparcie programowania w PHP baz danych Oracle

- Oprócz możliwego dostępu do baz danych Oracle poprzez **ODBC** istnieją specjalne, bardziej optymalne rozszerzenia PHP, w znacznym stopniu wspomagające programowanie baz danych Oracle oparte na specyficznych sterownikach tej bazy danych.
- Najlepszym rozwiązaniem jest jednak korzystanie z odpowiedniego sterownika **OCI (Oracle Call-Interface)** dla danej wersji serwera Oracle.
- Podstawową zaletą PHP jest możliwość rozszerzenia go o obsługę **OCI8 (Oracle8 Call-Interface)**.
- Funkcje rozszerzenia **php_oci8** generalnie pozwalają na dostęp do serwera **Oracle 7** i **Oracle 8**, jednak w systemie operacyjnym **Windows** konieczny jest **Oracle 8.1**, aby używać biblioteki **php_oci8.dll**.

Wsparcie programowania w PHP baz danych Oracle

- Spośród możliwości `php_oci8` należy wyróżnić:
 - wiązanie globalnych i lokalnych zmiennych PHP z kolumnami tabel lub zmiennymi wiązanymi Oracle'a,
 - pełną obsługę abstrakcyjnych typów danych Oracle'a: dużych obiektów (LOB), plików (FILE) i unikalnych numerów wierszy (ROWID).
- Gdy wersja serwera Oracle, dla którego chcemy napisać aplikację, nie jest odpowiednia do stosowania rozszerzenia `php_oci8`, PHP oferuje jeszcze inne rozszerzenie `php_oracle`.
- Rozszerzenie `php_oracle` (pod `Windows` jest to biblioteka `php_oracle.dll`) stanowi standardowe wsparcie dostępu do serwerów Oracle z poziomu aplikacji PHP, ale ma mniej możliwości niż to oparte na sterowniku OCI8.

Typowy schemat korzystania z bazy danych przez aplikację

Połączenie z bazą danych



Tworzenie zapytania / polecenia



Interpretowanie (parsowanie) zapytania / polecenia



Wykonanie zapytania / polecenia



Ewentualne wykorzystanie wyników zapytania / polecenia
(np. pobranie i wyświetlenie wyników zapytania)



Zwolnienie pamięci zajmowanej przez zapytanie / polecenie



Zamknięcie połączenia z bazą danych

Podstawowe funkcje rozszerzenia PHP_OCI8

- **Połączenie z bazą danych** może być realizowane przez funkcje:
 - **ocilogon**(string **użytkownik**, string **hasło**, string **sid**)
 - tworzy (zwykłe) połączenie z bazą Oracle
 - **ocinlogon**(string **użytkownik**, string **hasło**, string **sid**)
 - tworzy następne (zwykłe) połączenie z bazą Oracle
 - **ociplogon**(string **użytkownik**, string **hasło**, string **sid**)
 - tworzy trwałe (persistent) połączenie z bazą Oracle

Identyfikator instancji bazy danych Oracle **sid** jest argumentem opcjonalnym i jeżeli nie zostanie podany, to nastąpi próba połączenia z bazą danych Oracle na podstawie wartości zmiennej środowiskowej **ORACLE_SID**.

Podstawowe funkcje rozszerzenia PHP_OCI8

- **Tworzenie zapytania / polecenia** może być realizowane generalnie na dwa sposoby:

- bez parametrów (zmiennych związanych), np.

```
$Query = "SELECT * FROM emp";
```

- z parametrami (zmiennymi związanymi), np.

```
$Query = "INSERT INTO dept (deptno,dname,loc) ";
```

```
$Query .= "VALUES (:nr_w,:nazwa_w,:lok_w)";
```

W tym przypadku konieczna jest interpretacja (parsowanie) zapytania / polecenia połączona z powiązaniem zmiennych wiązanych Oracle'a z odpowiednimi zmiennymi PHP przy pomocy funkcji **ocibindbyname**.

Podstawowe funkcje rozszerzenia PHP_OCI8

- Interpretowanie (parsowanie) zapytania / polecenia jest realizowane przez funkcję:

`ociparse(resource połączenie, string zapytanie)`

Funkcja `ociparse` tworzy na podstawie zapytania instrukcję dla danego połączenia z bazą danych.

- Jeżeli instrukcja, przygotowana przez funkcję `ociparse`, posiada parametry (zmiennie wiązane), to wymagają one powiązania ze zmiennymi PHP przez funkcję:

`ocibindbyname (resource instrukcja,
string zmienna_związana_Oracle'a,
odpowiednia_zmienna_PHP lub
referencja_do_odpowiedniej_zmiennej_PHP,
integer maksymalna_długość,
integer typ)`

Podstawowe funkcje rozszerzenia PHP_OCI8

Argument `maksymalna_długość` funkcji `ocibindbyname`, określający maksymalną długość wiązanych danych, posiada specjalne znaczenie w dwóch przypadkach:

- dla nieabstrakcyjnych typów danych, gdy przyjmuje wartość `-1`, to maksymalną długością jest długość powiązanej zmiennej PHP;
- dla abstrakcyjnych typów danych musi przyjmować wartość `-1`.

Argument `typ` funkcji `ocibindbyname` jest opcjonalny:

- dla nieabstrakcyjnych typów danych, takich jak CHAR, VARCHAR2, NUMBER, DATE, nie podajemy go;
- dla abstrakcyjnych typów danych jego dozwolonymi wartościami są następujące predefiniowane stałe:

Podstawowe funkcje rozszerzenia PHP_OCI8

OCI_B_FILE – plik binarny

OCI_B_CFILE – plik tekstowy

OCI_B_BLOB – binarny duży obiekt

(LOB to akronim od Large Object)

OCI_B_CLOB – tekstowy duży obiekt

OCI_B_ROWID – identyfikator wiersza

(ROWID to akronim od ROW Identifier)

- Należy pamiętać, że przed wiązaniem ze zmienną o abstrakcyjnym typie danych należy alokować pamięć dla deskryptorów i wskaźników tego typu zmiennych, używając funkcji:

ocinewdescriptor(resource **połączenie**, integer **typ**)

Argument **typ** jest opcjonalny i może przyjmować wartości: **OCI_D_FILE**, **OCI_D_LOB**, **OCI_D_ROWID**, przy czym domyślnie przyjmowany jest deskryptor pliku (**OCI_D_FILE**).

Podstawowe funkcje rozszerzenia PHP_OCI8

- **Wykonanie zapytania / polecenia** jest realizowane przez funkcję:

ociexecute(resource **instrukcja**, integer **tryb**)

Argument **tryb** jest opcjonalny i może przyjmować predefiniowane wartości:

OCI_COMMIT_ON_EXECUTE – tryb domyślny, powodujący zatwierdzenie **instrukcji** po wykonaniu;

OCI_DEFAULT – powoduje, że transakcja **instrukcji** nie zostaje automatycznie zatwierdzona i może być jeszcze później wycofana, jeżeli dotyczy grupy poleceń DML (Data Manipulation Language) języka SQL.

Podstawowe funkcje rozszerzenia PHP_OCI8

- Transakcję można wycofać, jeżeli dotyczy **instrukcji** SQL z grupy DML (**insert,update,delete**), przez funkcję **ocirollback(resource połączenie)**

Funkcja próbuje wycofać wszystkie transakcje dla danego **połączenia**.

Jeżeli więcej **połączeń** zostało utworzonych przez funkcje **ocilogon** lub **ociplogon**, to połączenia te mogą nie być odrębne. Wówczas wycofanie może dotyczyć instrukcji, których nie chcemy anulować. Aby uniknąć tego typu problemów, warto stosować do kolejnych połączeń z bazą danych funkcji **ocinlogon**.

- Transakcję można zatwierdzić, jeżeli tego wymaga (tzn. należy do grupy instrukcji DML), przez funkcję **ocicommit(resource połączenie)**

która zatwierdza wszystkie transakcje dla **połączenia**.

Podstawowe funkcje rozszerzenia PHP OCI8

- Wykorzystanie wyników zapytania / polecenia może być zrealizowane na różne sposoby, które oparte są na kilku funkcjach pobierających wyniki zapytania i zapisujących je do wskazanych zmiennych PHP:

- `ociddefinebyname(resource instrukcja, string kolumna, zmienna_PHP, integer typ)`

Funkcja `ociddefinebyname` musi być wywołana przed odpowiednią funkcją `ociexecute(instrukcja)` i dokonuje przyporządkowania zmiennej PHP lub odwołaniu do niej kolumny tabeli bazy danych, występującej w instrukcji zwracanej przez funkcję `ociparse`.

Argument `typ` jest opcjonalny. Jest on wymagany i należy go używać w przypadku typów abstrakcyjnych. Dozwolone wartości typów abstrakcyjnych są takie same jak w przypadku.

Funkcję `ociddefinebyname` stosuje się w połączeniu z funkcją `ocifetch`.

Podstawowe funkcje rozszerzenia PHP_OCI8

- `ocifetch(resource instrukcja)` w połączeniu z funkcjami:
`ociresult(resource instrukcja, value kolumna)`
`ocinumcols(resource instrukcja)`

Funkcja `ocifetch` przygotowuje (wydobywa) kolejny wiersz danych do odczytu za pomocą funkcji `ociresult`, która zwraca z bieżącego wiersza wynikowego instrukcji wartość `kolumny` danej przez jej nazwę lub numer. Kolumny są numerowane od 1, a funkcja `ocinumcols` umożliwia poznanie liczby kolumn w instrukcji. Gdy funkcja `ocifetch` nie znajduje już kolejnego wiersza, to zwraca wartość `FALSE`.

Podstawowe funkcje rozszerzenia PHP_OCI8

- `ocifetchinto(resource instrukcja,`
 `tablica_PHP` lub `odwołanie_do_tablicy_PHP,`
 `integer tryb)`

Funkcja pobiera kolejny wiersz danych z wykonanej `instrukcji` i umieszcza go w `tablicy_PHP`, która zostaje domyślnie poindeksowana liczbami całkowitymi, zaczynając od 1. Gdy funkcja `ocifetchinto` nie znajduje już wiersza, to zwraca wartość `FALSE`.

Opcjonalny argument `tryb` steruje sposobem indeksowania tablicy. Dostępne są następujące predefiniowane stałe wartości `trybu`:

`OCI_ASSOC` – zwraca kolumny indeksowane nazwami,

`OCI_NUM` – zwraca kolumny indeksowane liczbami,

`OCI_RETURN_LOBS` – zwraca wartości dużych obiektów zamiast ich deskryptorów,

`OCI_RETURN_NULLS` – tworzy elementy dla kolumn pusty, które można łączyć operatorami logicznymi.

Podstawowe funkcje rozszerzenia PHP_OCI8

- `ocifetchstatement(resource instrukcja, tablica_PHP lub odwołanie_do_tablicy_PHP)`

Funkcja umieszcza tablicę danych z wyniku wykonania instrukcji w tablicy PHP, która zostaje poindeksowana nazwami kolumn. Każdy z elementów tej tablicy jest tablicą poindeksowaną liczbami całkowitymi, począwszy od zera. Odpowiadające sobie elementy tych podtablic stanowią wartości kolejnych kolumn (atrybutów) w danym wierszu (krotce) tabeli (relacji). Ponadto `ocifetchstatement` zwraca ilość pobranych wierszy w wyniku działania instrukcji.

- Gdy używamy funkcji `ocifetchinto` lub `ocifetchstatement`, przydatna jest też funkcja

`implode(string separator, array tablica),`

która przekształca tablicę w łańcuch, w którym kolejne elementy tablicy są rozdzielone separatorem.

Podstawowe funkcje rozszerzenia PHP_OCI8

- Zwolnienie pamięci zajmowanej przez zapytanie / polecenie odbywa się przy pomocy funkcji

`ocifreestatement(resource instrukcja)`

Funkcja zwalnia pamięć związaną z instrukcją utworzoną przy pomocy funkcji `ociparse`.

- Jeżeli utworzono deskryptor dużego obiektu LOB przy pomocy funkcji `ocinewdescriptor` według schematu:

`$lob = ocinewdescriptor(połączenie, typ),`

to należy zwolnić zajmowaną przez niego pamięć, używając funkcji:

`ocifreedesc(object $lob)`

Podstawowe funkcje rozszerzenia PHP_OCI8

- Zamknięcie połączenia z bazą danych realizowane jest przez funkcję

`ocilogout(resource połączenie)`

Funkcja `ocilogout` zamyka `połączenie` z bazą danych.

- Obsługa błędów w operacjach na bazie danych Oracle jest wspierana przez funkcję

`ocierror(resource identyfikator),`

Funkcja w przypadku:

- wystąpienia błędu zwraca `tablicę asocjacyjną`, opisującą ostatni błąd wygenerowany przez bazę Oracle, która składa się z `dwóch elementów o indeksach "code" i "message"`, przechowujących odpowiednio kod błędu Oracle i opis tego błędu;

- braku błędu zwraca wartość `FALSE`.

Argument `identyfikator` jest opcjonalny i pozwala wskazać zasób (instrukcję lub połączenie), dla którego chcemy uzyskać informację o błędzie.

Przykład zastosowania rozszerzenia PHP_OCI8

```
<?php
// połączenie z bazą
$Connection = ocilogon("scott","tiger");
// budownie zapytania
$query = "SELECT EMPNO,ENAME,HIREDATE ";
$query .= "FROM EMP ";
$query .= "WHERE JOB = 'CLERK' ";
// interpretacja zapytania
$stmt = ociparse($Connection,$query);
// skojarzenie trzech kolumn ze zmiennymi
ocifindbyname($stmt,"EMPNO",$EmployeeNo);
ocifindbyname($stmt,"ENAME",$EmployeeName);
ocifindbyname($stmt,"HIREDATE",$EmployeeHiredate);
// wykonanie zapytania
ociexecute($stmt);
// pobranie wszystkich wierszy wyniku zapytania
while (ocifetch($stmt))
{
    print("Pracownik o numerze $EmployeeNo i nazwisku $EmployeeName ");
    print("został zatrudniony $EmployeeHiredate.<br>\n");
}
// zwolnienie polecenia
ocifreestatement($stmt);
// zamknięcie połączenia
ocilogout($Connection);
```

?>