

PHP + ORACLE

(w bardzo wielkim skrócie)

dr inż. Artur Gramacki

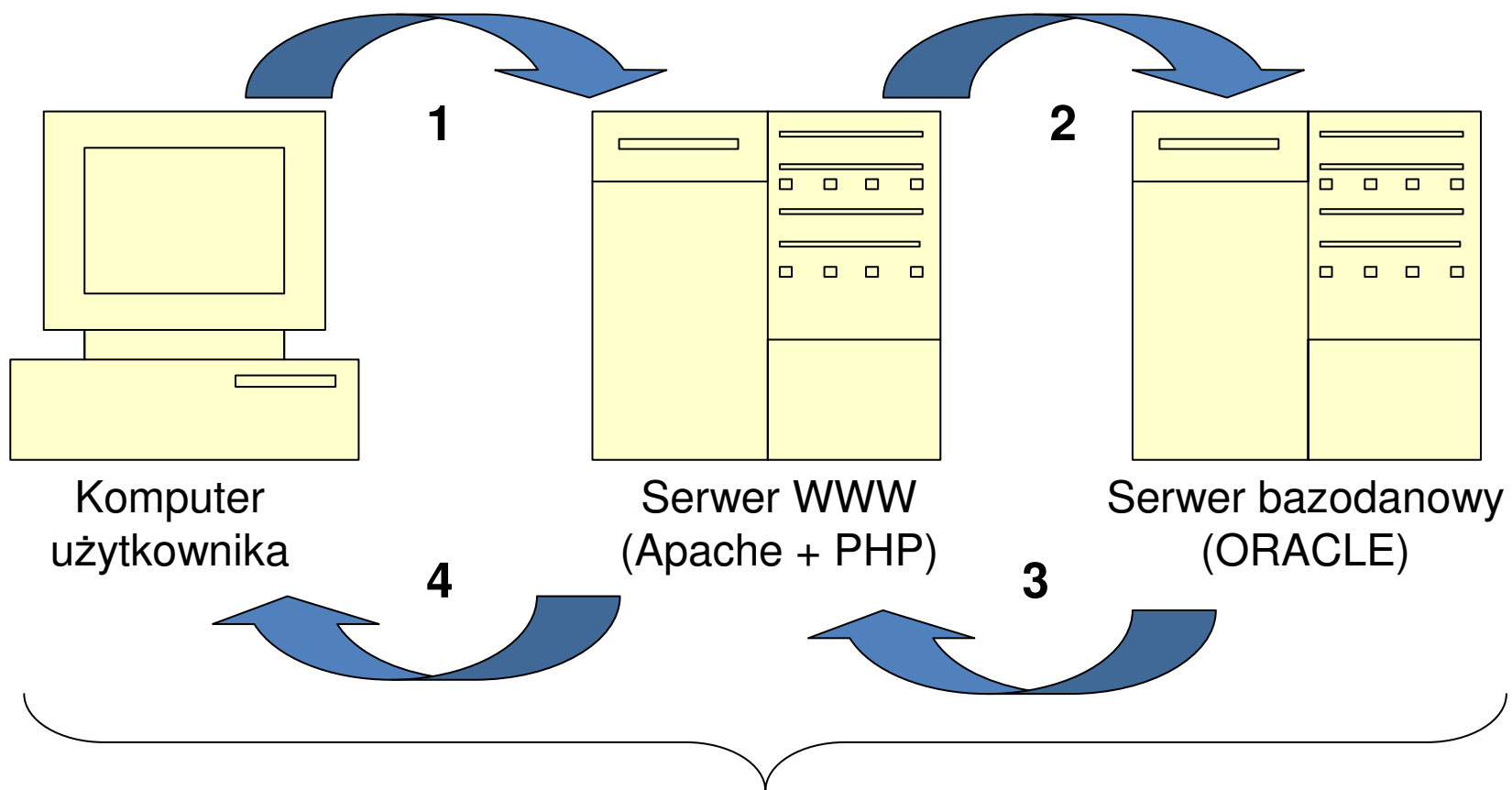
a.gramacki@issi.uz.zgora.pl

Instytut Sterowania i Systemów Informatycznych

Uniwersytet Zielonogórski

2015

Środowisko uruchamiania aplikacji WWW




Może być na jednym fizycznym komputerze,
na dwóch lub trzech komputerach

Co nam jest potrzebne?

- Konto na serwerze ORACLE
 - wystarczają minimalne uprawnienia pozwalające połączyć się do serwera i tworzyć obiekty w swoim schemacie (np. tabele, procedury)
- XAMPP
 - Apache
 - PHP
 - W chwili tworzenia niniejszej prezentacji używano XAMPP-a w wersji 5.6.12 ([xampp-portable-win32-5.6.12-0-VC11.7z](#))
- biblioteka OCI8 (Oracle Call Interface) dająca dostęp do bazy Oracle
 - najwygodniej używać: Oracle Instant Client

XAMPP, http://localhost

[Apache Friends](#)[Applications](#)[FAQs](#)[HOW-TO Guides](#)[PHPInfo](#)[phpMyAdmin](#)

 **XAMPP** Apache + MySQL + PHP + Perl

Welcome to XAMPP for Windows 5.6.12

You have successfully installed XAMPP on this system! Now you can start using Apache, MySQL, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#), adding yourself to the [Mailing List](#), and liking us on [Facebook](#), following our exploits on [Twitter](#), or adding us to your [Google+](#) circles.

XAMPP, panel kontrolny

The screenshot displays the XAMPP Control Panel v3.2.1 interface. The main window shows a list of modules (Apache, MySQL, FileZilla, Mercury, Tomcat) with their respective PID(s) and Port(s). The Apache module is highlighted in green, indicating it is running. The 'Actions' column for Apache shows 'Stop', 'Admin', 'Config', and 'Logs' buttons. The 'Logs' button is circled in blue, and a context menu is open over it, showing options like 'Apache (access.log)', 'Apache (error.log)', and 'PHP (php_error_log)'. The 'Logs' button is also circled in blue. The 'Netstat' window is open, showing a list of active sockets. The 'New socket' tab is selected, and the list shows several sockets, including those for 'httpd.exe' and 'svchost.exe'. The 'Logs' button in the XAMPP Control Panel is also circled in blue.

XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	3792 3192	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

20:37:10 [main] Checking for prerequisites
20:37:24 [main] All prerequisites found
20:37:24 [main] Initializing Modules
20:37:24 [main] The FileZilla module is disabled
20:37:24 [main] The Mercury module is disabled
20:37:24 [main] Starting Check-Timer
20:37:24 [main] Control Panel Ready
20:37:26 [Apache] Attempting to start Apache app...
20:37:26 [Apache] Status change detected: running

Netstat - TCP Listening sockets

Address	Port	PID	Name
0.0.0.0	80	5728	httpd.exe
0.0.0.0	443	752	svchost.exe
192.168.21.75	139	4	System
192.168.56.1	443	4	System
0.0.0.0	443	5728	httpd.exe
0.0.0.0	443	4	System
127.0.0.1	843	3192	Dropbox.exe
0.0.0.0	1521	2520	TNSLNR.EXE
0.0.0.0	4004	1488	svchost.exe
0.0.0.0	8080	2520	TNSLNR.EXE
0.0.0.0	17500	3192	Dropbox.exe
127.0.0.1	17600	3192	Dropbox.exe
127.0.0.1	19872	3192	Dropbox.exe
0.0.0.0	49152	752	svchost.exe
0.0.0.0	49153	1088	svchost.exe
0.0.0.0	49154	1192	svchost.exe
0.0.0.0	49155	916	lsass.exe
127.0.0.1	49156	2520	TNSLNR.EXE
192.168.21.75	49157	2052	omtsreco.exe
0.0.0.0	49160	1700	spoolsv.exe
0.0.0.0	49171	2560	oracle.exe
0.0.0.0	49172	908	services.exe
127.0.0.1	50002	3192	Dropbox.exe
127.0.0.1	50009	3192	Dropbox.exe
127.0.0.1	50010	3192	Dropbox.exe
0.0.0.0	50016	3380	java.exe
0.0.0.0	51402	2700	svchost.exe
192.168.21.75	51497	3192	Dropbox.exe
192.168.21.75	51609	3192	Dropbox.exe

PHP i ORACLE, opracowanie: Artur Gramacki

Pochodzenie przykładów

- Using PHP with Oracle Database 11g
 - http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/oow10/php_db/php_db.htm
- + własne modyfikacje / zmiany / uzupełnienia / komentarze

PHP, funkcja phpinfo()

```
<?php
    phpinfo();
?>
```

PHP Version 5.6.12	
	
System	Windows NT DOM 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
Build Date	Aug 6 2015 11:58:38
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-p3web" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com_dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	E:\xampp_5.6.12_Portable\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

Oci8 – diagnostyka

oci8

OCI8 Support	enabled
OCI8 DTrace Support	disabled
OCI8 Version	2.0.9
Revision	\$Id: f5a3ee1083d1ffa6adb5143efda6eafa210b8414 \$
Oracle Run-time Client Library Version	12.1.0.2.0
Oracle Compile-time Instant Client Version	12.1

Directive	Local Value	Master Value
oci8.connection_class	<i>no value</i>	<i>no value</i>
oci8.default_prefetch	100	100
oci8.events	Off	Off
oci8.max_persistent	-1	-1
oci8.old_oci_close_semantics	Off	Off
oci8.persistent_timeout	-1	-1
oci8.ping_interval	60	60
oci8.privileged_connect	Off	Off
oci8.statement_cache_size	20	20

Statistics	
Active Persistent Connections	0
Active Connections	0

Oci8 – konfiguracja

- Oracle Instant Client 12.1 32bits

- instantclient-basic-nt-12.1.0.2.0.zip
- instantclient-sqlplus-nt-12.1.0.2.0.zip
- tnsnames.ora

- php.ini

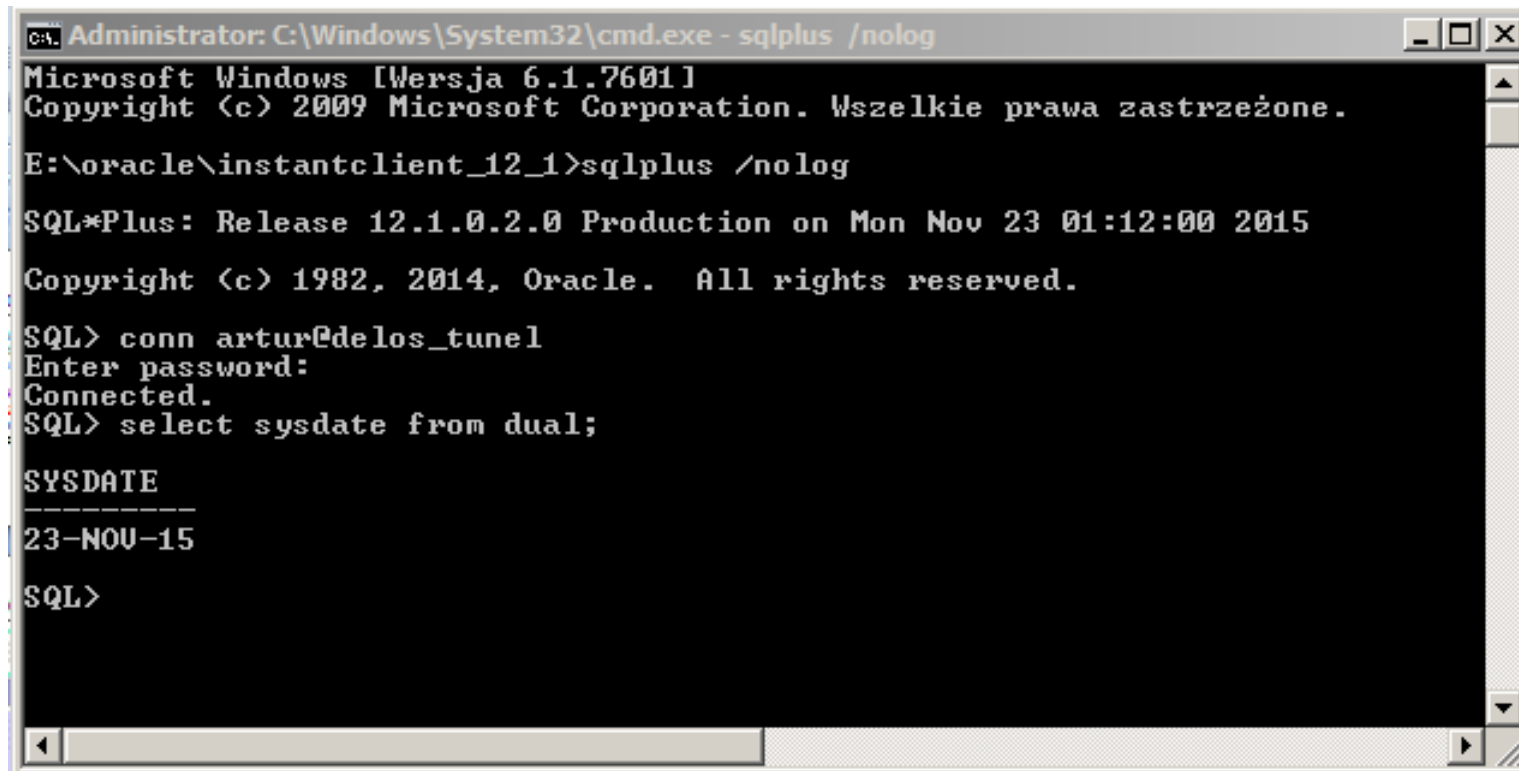
- ..\php\ext\php_oci8_12c.dll

```
;extension=php_oci8.dll          ; Use with Oracle 10gR2 Instant Client  
;extension=php_oci8_11g.dll      ; Use with Oracle 11gR2 Instant Client  
extension=php_oci8_12c.dll       ; dodane przez AG
```

- ..\xampp_5.6.12_Portable_start_xampp.bat

```
REM PHP musi widzieć biblioteki z instantclient-a  
REM więc ustawiamy zmienną PATH  
  
PATH=e:\oracle\instantclient_12_1_32bit  
xampp-control.exe
```

Oracle Instant Client



```
C:\Windows\System32\cmd.exe - sqlplus /nolog
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

E:\oracle\instantclient_12_1>sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Nov 23 01:12:00 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> conn artur@delos_tunel
Enter password:
Connected.
SQL> select sysdate from dual;

SYSDATE
-----
23-NOV-15

SQL>
```

tnsnames.ora

```
DELOS =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = delos.iie.uz.zgora.pl)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = orcl.iie.uz.zgora.pl)  
    )  
  )
```

Wersja do połączenia lokalnego

```
DELOS_TUNEL =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 9999))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = orcl.iie.uz.zgora.pl)  
    )  
  )
```

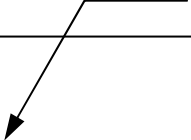
Wersja do połączenia za pomocą tunelu
SSH

Plik conn.inc

```
<?php
// define('USER', 'scott');
// define('PASS', 'tiger');
// define('CONN', 'delos.iie.uz.zgora.pl:1521/orcl.iie.uz.zgora.pl');

define('USER', 'scott');
define('PASS', 'tiger');
define('CONN', 'localhost:9999/orcl.iie.uz.zgora.pl');
?>
```

Oracle's Easy Connect
hostname + port + DB service name



Połączenie do bazy

```
<?php
require('conn.inc');
// Tworzymy połączenie do bazy Oracle.
$conn = oci_connect(USER, PASS, CONN);

if (!$conn) {
    $error = oci_error();
    echo $error['message'], "\n";
    exit;
} else {
    print "Podłączono do Oracle!";
}

// Zamykamy połączenie do bazy Oracle.
oci_close($conn);
```

Warning: oci_connect(): ORA-01017: invalid username/password; logon denied in **E:\xampp_5.6.12_Portable\htdocs\oracle2\ora_conn.php** on line **3**
ORA-01017: invalid username/password; logon denied

Każda standardowa sesja do bazy Oracle zostanie automatycznie zakończona z chwilą zakończenia działania skryptu (tu PHP)

Połączenie do bazy – inaczej

```
<?php
require('conn.inc');
// można też tak, tylko po co :-)

$connStr = '
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = localhost) (PORT = 9999))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = orcl.iie.uz.zgora.pl)
  )
)';

// Tworzymy połączenie do bazy Oracle.
$conn = oci_connect(USER, PASS, $connStr);

if (!$conn) {
  $error = oci_error();
  echo $error['message'], "\n";
  exit;
} else {
  print "Podłączono do Oracle!";
}

// Zamykamy połączenie do bazy Oracle.
oci_close($conn);
?>
```

Wtręt: apostrofy i cudzysłowy

- Pamiętaj, że w PHP
 - apostrofy służą do ograniczania tekstu, w którym interpreter PHP nie będzie szukał wywołań innych zmiennych
 - w przypadku cudzysłowów interpreter PHP będzie szukał wywołań zmiennych
 - tekst w apostrofach jest nieco szybciej „czytany” przez interpreter PHP ale różnice są naprawdę minimalne i w praktyce (pewnie) bardzo słabo zauważalne

```
<?php
    $test = "jakiś tekst";
    echo "$test";
    // Na ekranie wyświetli się: "jakiś tekst"
?>
```

```
<?php
    $test = "jakiś tekst";
    echo '$test';
    // na ekranie wyświetli się: "$test"
?>
```

Model demonstracyjny SCOTT

SQL> select * from emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

FK

SQL> select * from dept;

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

PK

Pobieranie danych

- Taki wynik chcemy otrzymać (zawartość tabeli EMP)

7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/07/13	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/07/13	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

Pobieranie danych

1. **Parse** the statement for execution
2. **Bind** data values (optional)
3. **Execute** the statement
4. **Fetch** the results from the database

Pobieranie danych

```
<?php
require('conn.inc');
$conn = oci_connect(USER, PASS, CONN);

$query = 'select * from emp';
$stmtid = oci_parse($conn, $query);
$r = oci_execute($stmtid);

print '<table border="1">';
while ($row = oci_fetch_array($stmtid, OCI_RETURN_NULLS + OCI_ASSOC)) {
    print '<tr>';
    foreach ($row as $item) {
        print '<td>'.($item !== null ? htmlentities($item, ENT_QUOTES) : '&nbsp;').</td>';
    }
    print '</tr>';
}
print '</table>';
?>
```

UWAGA: tu nie stawiamy średnika na końcu polecenia SQL. Widoczny śrdenik jest separatorem instrukcji w PHP

Zwraca wynik w postaci tablicy asocjacyjnej, wartości NULL też są zwracane

htmlentities – usuwa z tekstu ew. tagi html-owe i dzięki temu nie ma problemów z poprawnym wyświetlaniem w przeglądarce (np. tabele mogły by być błędnie formatowane)

Pobieranie danych

- [oci_fetch_all](#) — Fetches multiple rows from a query into a two-dimensional array
- [oci_fetch_array](#) — Returns the next row from a query as an associative or numeric array
- [oci_fetch_assoc](#) — Returns the next row from a query as an associative array
- [oci_fetch_object](#) — Returns the next row from a query as an object
- [oci_fetch_row](#) — Returns the next row from a query as a numeric array
- [oci_fetch](#) — Fetches the next row from a query into internal buffers

oci_fetch_array() Modes

Constant	Description
OCI_BOTH	Returns an array with both associative and numeric indices. This is the same as OCI_ASSOC + OCI_NUM and is the default behavior.
OCI_ASSOC	Returns an associative array.
OCI_NUM	Returns a numeric array.
OCI_RETURN_NULLS	Creates elements for NULL fields. The element values will be a PHP NULL .
OCI_RETURN_LOBS	Returns the contents of LOBs instead of the LOB descriptors.

Pobieranie danych

```
<?php
require('conn.inc');
$conn = oci_connect(USER, PASS, CONN);

$query = 'select * from emp';
$stmt = oci_parse($conn, $query);
$r = oci_execute($stmt);

print '<table border="1">';
while ($row = oci_fetch_array($stmt, OCI_ASSOC)) {
    print '<tr>';
    foreach ($row as $item) {
        print '<td>'.($item !== null ? htmlentities($item, true) : '&nbsp;');
    }
    print '</tr>';
}
print '</table>';
?>
```

Usunięto OCI_RETURN_NULLS

7369	SMITH	CLERK	7902	80/12/17	800	20	
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975	20	
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850	30	
7782	CLARK	MANAGER	7839	81/06/09	2450	10	
7788	SCOTT	ANALYST	7566	87/07/13	3000	20	
7839	KING	PRESIDENT	81/11/17	5000	10		
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/07/13	1100	20	
7900	JAMES	CLERK	7698	81/12/03	950	30	
7902	FORD	ANALYST	7566	81/12/03	3000	20	
7934	MILLER	CLERK	7782	82/01/23	1300	10	

Pobieranie danych

- Większa kontrola nad wyświetlaniem danych

```
<?php
require('conn.inc');
$conn = oci_connect(USER, PASS, CONN);

$query = 'select * from emp';
$stmt = oci_parse($conn, $query);
$rs = oci_execute($stmt);
```

```
print '<table border="1">';
while ($row = oci_fetch_array($stmt, OCI_NUM)) {
    print '<tr>' .
        '<td style="color:red"><b>' . $row[0] . '</td>' .
        '<td style="color:blue">' . $row[1] . '</td>' .
        '<td style="color:green">' . $row[2] . '</td>' .
        '</tr>';
}
print '</table>';
?>
```

Teraz zwraca tablicę
numeryczną

7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7876	ADAMS	CLERK
7900	JAMES	CLERK
7902	FORD	ANALYST
7934	MILLER	CLERK

Konieczne maskowanie
cudzysłów

```
// Chyba tu będzie wygodniejsza wersja z cudzysłowami.
$rs = oci_execute($stmt);
print '<table border="1">';
while ($row = oci_fetch_array($stmt, OCI_NUM)) {
    print "<tr>
        <td style=\"color:red\"><b> $row[0] </td>
        <td style=\"color:blue\"> $row[1] </td>
        <td style=\"color:green\"> $row[2] </td>
        </tr>";
}
print '</table>';
```

Zmienne wiązane

- Angielska nazwa: **bind variables**
- Pozwala uniknąć wielokrotnego parsowania
- Znacznie przyśpiesza wykonywanie się programów, które operują na „podobnych” danych
 - takie same polecenia SQL wykonywane z innymi parametrami nie wymagają ponownego parsowania (reparsowania) a proces parsowania nie jest natychmiastowy
- W pewnym zakresie zabezpiecza przed atakami typu **SQL Injection**

Zmienne wiązane

```
SELECT COUNT(*) FROM all_objects ;  
  
COUNT(*)  
-----  
56636
```

```
ALTER SYSTEM flush shared_pool;  
SET TIMING ON  
DECLARE  
  l_sql VARCHAR(100) := 'SELECT COUNT(*) FROM all_objects WHERE object_id = '  
  l_cnt NUMBER := 0;  
BEGIN  
  FOR i IN 1..1000 LOOP  
    EXECUTE IMMEDIATE l_sql || i INTO l_cnt;  
  END LOOP;  
END;  
/
```

ŹLE: 00:01:41.71

```
ALTER SYSTEM flush shared_pool;  
SET TIMING ON  
DECLARE  
  l_sql VARCHAR(100) := 'SELECT COUNT(*) FROM all_objects WHERE object_id = :id';  
  l_cnt NUMBER := 0;  
BEGIN  
  FOR i IN 1..1000 LOOP  
    EXECUTE IMMEDIATE l_sql INTO l_cnt USING i;  
  END LOOP;  
END;  
/
```

OK: 00:00:00.16

* autor przykładu: dr hab. inż. Mikołaj Morzy, Instytut Informatyki, Politechnika Poznańska

Zmienne wiązane

```
<?php
require('conn.inc');

function do_fetch($empno, $s) {
    print '<p>$empno is ' . $empno . '</p>';
    print '<table border="1">';
    while ($row = oci_fetch_array($s, OCI_RETURN_NULLS + OCI_ASSOC)) {
        print '<tr>';
        foreach ($row as $item) {
            print '<td>' . ($item ? htmlentities($item) : '&nbsp;') . '</td>';
        }
        print '</tr>';
    }
    print '</table>';
}

$c = oci_connect(USER, PASS, CONN);
$query = 'select * from emp where empno = :empno_bv';
$s = oci_parse($c, $query);

$empno = 101;
oci_bind_by_name($s, ":empno_bv", $empno);
oci_execute($s);
do_fetch($empno, $s);

$empno = 104;
oci_execute($s);
do_fetch($empno, $s);

?>
```

Używamy zmiennych wiązanych aby poprawić wydajność i ograniczyć ataki SQL Injection

Powtórne wykonanie tego samego SELECT-a bez konieczności jego powtórnego parsowania

Transakcje

```
<?php
    putenv('NLS_LANG=POLISH_POLAND.EE8MSWIN1250');
    require('conn.inc');

    $conn = oci_connect(USER, PASS, CONN) or die('Błąd połączenia');
    $query = "INSERT INTO tab VALUES (:id_bind_var, 'aęśćźźńół AĘŚĆŹŹŃÓŁ', sysdate)";
    $stmt = oci_parse($conn, $query) or die('Błąd wstawiania');

    $i = 1;
    oci_bind_by_name($stmt, ':id_bind_var', $i);
    // Zatwierdzanie zmian. Użyto trybu domyślnego OCI_COMMIT_ON_SUCCESS.
    oci_execute($stmt);

    $i = 2;
    oci_execute($stmt, OCI_NO_AUTO_COMMIT);
    // Zatwierdzanie zmian.
    oci_commit($conn);

    $i = 3;
    oci_execute($stmt, OCI_NO_AUTO_COMMIT);
    // Wycofanie transakcji.
    oci_rollback($conn);

    echo 'Sprawdź zawartość tabeli TAB.';

    oci_close($conn);
?>
```

Execution Modes

Constant	Description
OCI_COMMIT_ON_SUCCESS	Automatically commit all outstanding changes for this connection when the statement has succeeded. This is the default.
OCI_DESCRIBE_ONLY	Make query meta data available to functions like oci_field_name() but do not create a result set. Any subsequent fetch call such as oci_fetch_array() will fail.
OCI_NO_AUTO_COMMIT	Do not automatically commit changes. Prior to PHP 5.3.2 (PECL OCI8 1.4) use OCI_DEFAULT which is equivalent to OCI_NO_AUTO_COMMIT .

Pobieranie danych, większy przykład

```
<?php
putenv('NLS_LANG=POLISH_POLAND.EE8MSWIN1250');
require ('conn.inc');

$conn = oci_connect(USER, PASS, CONN) or die('Błąd połączenia');
$recordLimit = 5;
$whereFilter = 1000;
$stmt = oci_parse($conn,
    'SELECT * FROM (
        SELECT ename, job, sal FROM emp WHERE sal > :sal_bv ORDER BY sal DESC
    )
    WHERE ROWNUM <= :lim_bv') or die('Błąd polecenia SELECT');

oci_bind_by_name($stmt, ':sal_bv', $whereFilter);
oci_bind_by_name($stmt, ':lim_bv', $recordLimit);
oci_execute($stmt);

echo '<table border="1">';
while ($row = oci_fetch_array($stmt, OCI_NUM)) {
    echo
        '<tr>
            <td>' . $row[0] . '</td>
            <td>' . $row[1] . '</td>
            <td>' . $row[2] . '</td>
        </tr>';
}
echo '</table>';
?>
```

KING	PRESIDENT	5000
SCOTT	ANALYST	3000
FORD	ANALYST	3000
JONES	MANAGER	2975
BLAKE	MANAGER	2850

PL/SQL

- PL/SQL – język proceduralny, rozszerzenie SQL-a
- Procedury, funkcje, pakiety
- Kody przechowywane są w bazie ORACLE
 - bardzo wygodne w praktyce rozwiązanie
- Zwykle rozwiązanie to jest szybsze niż kody w PHP wykonujące analogiczne zadania

Kolejne demonstracyjne tabele

```
DROP TABLE tab;
DROP SEQUENCE myseq;

CREATE TABLE tab (
    id NUMBER,
    tekst VARCHAR2(20),
    data DATE,
    PRIMARY KEY (id)
);

CREATE SEQUENCE myseq START WITH 1000;

INSERT INTO tab(id, data) VALUES (myseq.NEXTVAL, sysdate);
INSERT INTO tab(id, data) VALUES (myseq.NEXTVAL, sysdate);
INSERT INTO tab(id, data) VALUES (myseq.NEXTVAL, sysdate);

COMMIT;

SELECT * FROM tab;
SELECT id, TO_CHAR(data, 'dd-mm-yyyy, hh24:mi:ss') data FROM tab;
```

ID	DATA
1000	26-11-2015, 21:19:57
1001	26-11-2015, 21:19:58
1002	26-11-2015, 21:19:58

PL/SQL, procedura

```
CREATE OR REPLACE PROCEDURE myproc (a_in NUMBER, b_in DATE) AS  
BEGIN  
    INSERT INTO tab(id, data) VALUES (a_in, b_in);  
END;  
/
```

```
SQL> execute myproc(1, sysdate);
```

Ukończono pomyślnie procedurę PL/SQL.

```
SQL> execute myproc(myseq.NEXTVAL, sysdate);
```

Ukończono pomyślnie procedurę PL/SQL.

PL/SQL, wywołanie procedury z PHP

```
<?php
require ('conn.inc');
$conn = oci_connect(USER, PASS, CONN) or die('Błąd połączenia');
$stmtid = oci_parse($conn, "BEGIN myproc(myseq.NEXTVAL, sysdate); END;");
oci_execute($stmtid);
echo "Wykonano procedurę składowaną.";
?>
```

```
<?php
require ('conn.inc');
$conn = oci_connect(USER, PASS, CONN) or die('Błąd połączenia');
$stmtid = oci_parse($conn, "BEGIN myproc(:bv1, :bv2); END;");
$v1 = 12345;
$v2 = '15/11/25';
oci_bind_by_name($stmtid, ":bv1", $v1);
oci_bind_by_name($stmtid, ":bv2", $v2);
$result = oci_execute($stmtid);
if ($result == false) {
    $error = oci_error($stmtid);
    echo $error['message'], "\n";
    exit;
} else {
    print "Wykonano procedurę składowaną.";
}
?>
```

PL/SQL, funkcja

```
CREATE OR REPLACE FUNCTION myfun (a_in NUMBER) RETURN NUMBER AS
BEGIN
    RETURN (a_in * 2);
END;
/
```

```
SQL> select myfun(123) FROM dual;
```

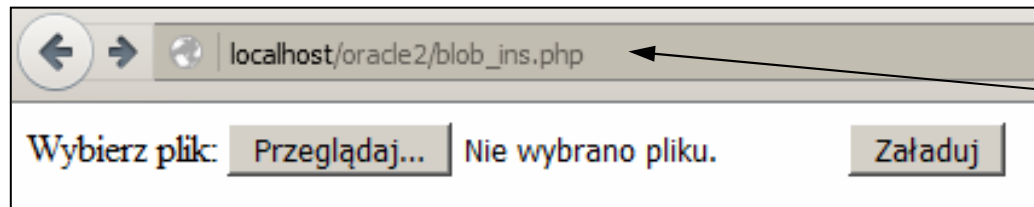
```
MYFUN(123)
-----
        246
```

```
<?php
require ('conn.inc');
$conn = oci_connect(USER, PASS, CONN) or die('Błąd połączenia');
$stmt = oci_parse($conn, "BEGIN :bv := myfun(123); END;");
oci_bind_by_name($stmt, ":bv", $v, 10);
$result = oci_execute($stmt);
if ($result == false) {
    $error = oci_error($stmt);
    echo $error['message'], "\n";
    exit;
} else {
    echo $v, "<br>\n";
    echo "Wykonano funkcję składowaną.";
}
?>
```

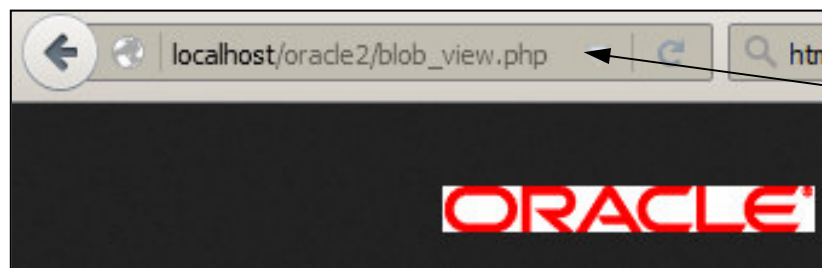
Alokacja dostatecznej
ilości pamięci do
przechowania danych

Użycie zmiennych typu LOB

```
DROP TABLE blob_tab;  
  
CREATE TABLE blob_tab (  
    blob_id NUMBER,  
    blob_data BLOB  
);
```



Ładowanie obrazka do bazy
(do kolumny typu BLOB)



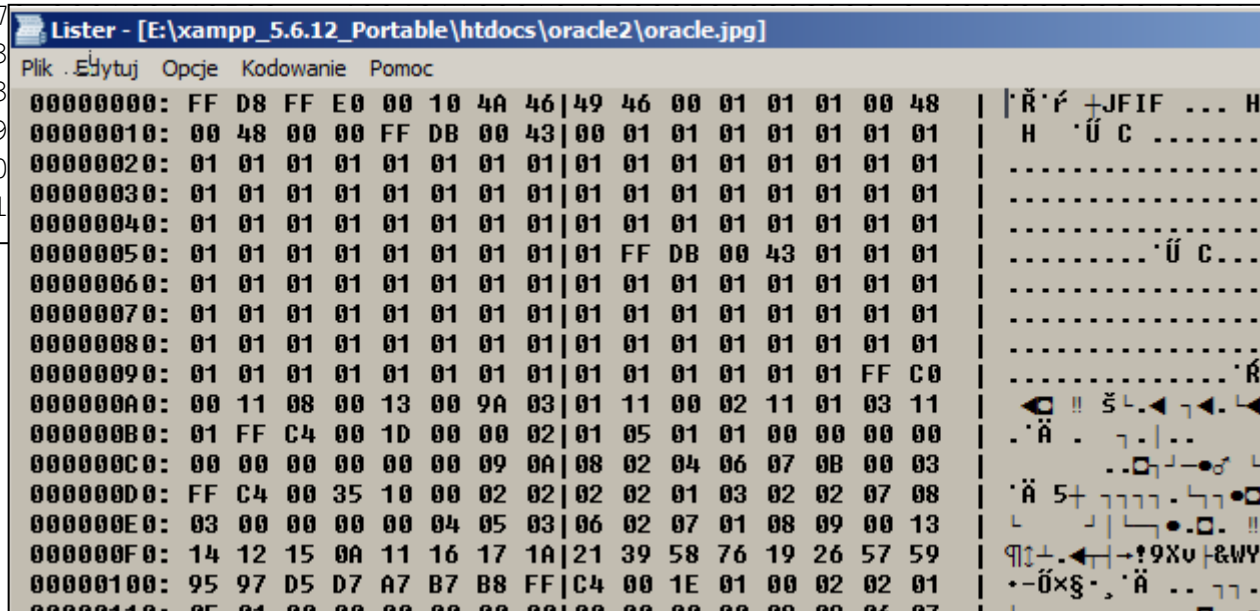
Pobranie obrazka z kolumny
BLOB i wyświetlenie go w
przeglądarce

Użycie zmiennych typu LOB

```
SQL> set long 500
SQL> SELECT blob_data FROM blob_tab WHERE blob_id = 1;
```



BLOB_DATA

[illegible]

Użycie zmiennych typu LOB, ładowanie

```
<?php
require ('conn.inc');
if (!isset($_FILES['lob_upload'])) {
    echo '
        <form action="blob_ins.php"
        method="POST" enctype="multipart/form-data">
        Wybierz plik: <input type="file" name="lob_upload">
        <input type="submit" value="Załaduj">
        </form> ';
} else {
    $myblob_id = 1;
    $conn = oci_connect(USER, PASS, CONN);
    $query = 'DELETE FROM blob_tab WHERE blob_id = :myblob_id';
    $stmt = oci_parse ($conn, $query);
    oci_bind_by_name($stmt, ':myblob_id', $myblob_id);
    $e = oci_execute($stmt);

    $lob = oci_new_descriptor($conn, OCI_D_LOB);
    $stmt = oci_parse($conn,
        'INSERT INTO blob_tab (blob_id, blob_data) ' .
        'VALUES (:myblob_id, empty_blob()) ' .
        'RETURNING blob_data INTO :blob_data');
    oci_bind_by_name($stmt, ':myblob_id', $myblob_id);
    oci_bind_by_name($stmt, ':blob_data', $lob, -1, OCI_B_BLOB);
    oci_execute($stmt, OCI_DEFAULT);
    if ($lob->savefile($_FILES['lob_upload']['tmp_name'])) {
        oci_commit($conn);
        echo "BLOB załadowany do bazy";
    } else {
        echo "Problem z załadowaniem do bazy BLOB-a";
    }
    $lob->free();
}
?>
```

Gdy nic nie załadowano, wyświetl
formularz (z polem typu file)

Gdy wybrano jakiś plik

Tworzy pusty deskryptor LOB-a

OCI_B_LOB: informujemy ORACLE,
jak ma traktować zmienną (tu: jako
BLOB-a)

Zapisuje zawartość pliku do BLOB-a.
Tymczasowa nazwa pliku, ustalana
wewnętrznie przez php

Użycie zmiennych typu LOB, wyświetlanie

```
<?php
require ('conn.inc');
$conn = oci_connect (USER, PASS, CONN);

$query = 'SELECT blob_data FROM blob_tab WHERE blob_id = :myblob_id';
$stmt = oci_parse ($conn, $query);
$myblob_id = 1;
oci_bind_by_name ($stmt, ':myblob_id', $myblob_id);
oci_execute ($stmt);
$arr = oci_fetch_array ($stmt, OCI_ASSOC);
$result = $arr['BLOB_DATA']->load();

header ("Content-type: image/JPEG");
echo $result;

oci_close($conn);
?>
```

UWAGA: w tym pliku nie ma prawa pojawić się żaden dodatkowy znak (spacja, cokolwiek) poza obszarem znaczników php (otwierającym i zamykającym)

Musi być dużymi literami, gdyż domyślnie ORACLE tworzy nazwy tabel i kolumn pisane z dużych liter

Wysłanie do przeglądarki nagłówka HTML informującego o rodzaju pliku (identyfikator MIME)

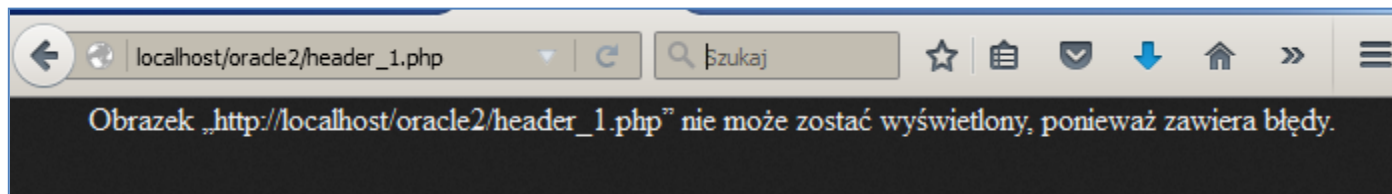
Dzięki temu przeglądarka umie prawidłowo go zinterpretować i wyświetlić

Wrzucenie na stronę strumienia bitów tworzących obrazek

Funkcja header()

- Bardzo ważne: funkcja header() MUSI być wywołana jako pierwsza, zanim wyślemy do przeglądarki JAKIEKOLWIEK dane
 - np. każda spacja lub znak końca linii (których przecież nie widać zwykle na ekranie) też są traktowane jak pełnoprawne dane
 - tekst z dokumentacji (php.net/manual/en/function.header.php)
„It is a very common error to read code with include, or require, functions, or another file access function, and have spaces or empty lines that are output before header() is called.”


```
<!-- w tym miejscu, przed header(), nie może być __NIC__ -->
<?php
    header ("Content-type: image/JPEG");
    readfile('oracle.jpg');
?>
```



Potencjalny problem z wyświetlaniem zawartości BLOB-ów


1. PRZED oraz PO znacznikach php nie może być ŻADNYCH dodatkowych znaków. Nawet pojedyncza spacja lub znak końca linii powoduje, że próba wyświetlenia zawartości BLOB-a da błąd pokazany na poprzednim slajdu
2. Gdy pliki są kodowane w UTF8 muszą być zakodowane tak, aby nie zapisywał się tzw. BOM. Czyli np. w Notepad++, gdy chcemy zapisać plik w UTF8 musimy wybrać opcję "Encode in UTF8 without BOM" a nie "Encode in UTF8,, (na początku tak zakodowanego pliku wstawiane są trzy bajty danych, szesnastkowo: EF BB BF)

```
<?php
    require ('conn.inc');
    ...
    header ("Content-type: image/JPEG");
    echo $result;
?>
```



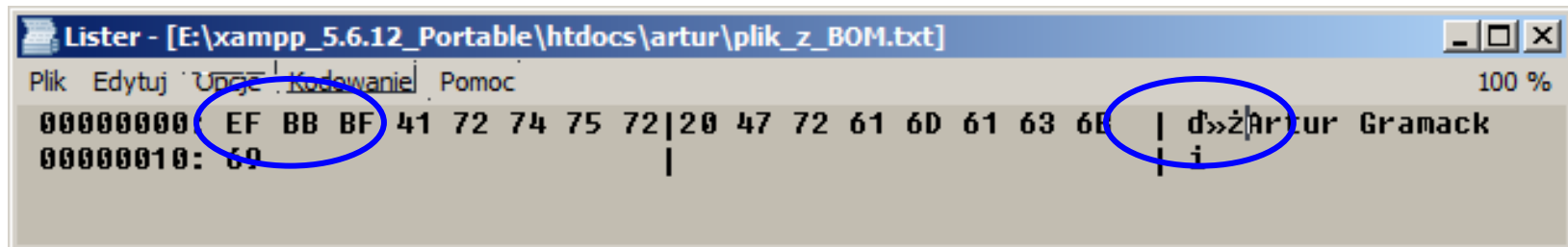
```
<?php
// define('USER', 'scott');
// define('PASS', 'tiger');
// define('CONN', 'delos.iie.uz.zgora.pl:1521/orcl.iie.uz.zgora.pl');

define('USER', 'scott');
define('PASS', 'tiger');
define('CONN', 'localhost:9999/orcl.iie.uz.zgora.pl');
?>
```



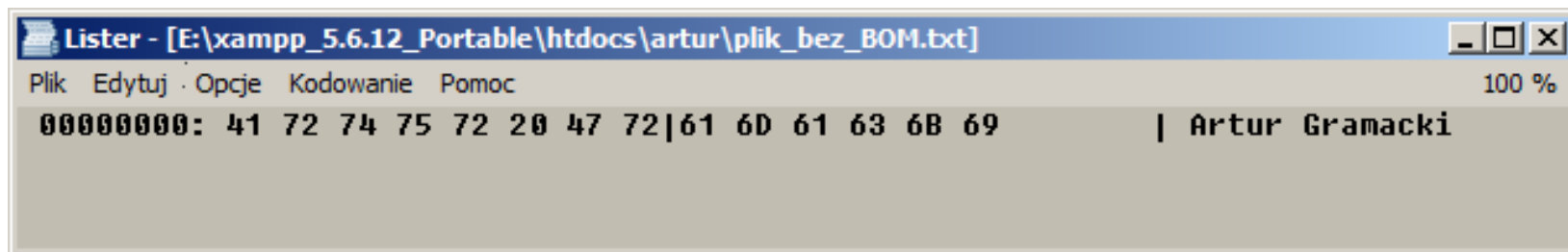
UTF8 z oraz bez BOM

- BOM (ang. **B**yte **O**rdery **M**ark), znacznik kolejności bajtów
 - https://pl.wikipedia.org/wiki/BOM_%28informatyka%29
- plik z BOM



```
Listy - [E:\xampp_5.6.12_Portable\htdocs\artur\plik_z_BOM.txt]
Plik Edytuj Opcje Kodowanie Pomoc 100 %
00000000: EF BB BF 41 72 74 75 72 20 47 72 61 60 61 63 68 69 | d>>żArtur Gramacki
00000010: 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 | i
```

- plik bez BOM



```
Listy - [E:\xampp_5.6.12_Portable\htdocs\artur\plik_bez_BOM.txt]
Plik Edytuj Opcje Kodowanie Pomoc 100 %
00000000: 41 72 74 75 72 20 47 72 61 60 61 63 68 69 | Artur Gramacki
```

Poprawne wyświetlanie znaków narodowych

- Plik źródłowy i wynik działania

```
1 <!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html; charset=UTF-8">
5 </head>
6 <body>
7   <h2>ąęśćźźńół ĄĘŚĆŹŹŃÓŁ</h2><br>
8   <?php
9     putenv('NLS_LANG=POLISH_POLAND.UTF8');
10    require ('conn.inc');
11
12    $conn = oci_connect(USER, PASS, CONN);
13    $stmt = oci_parse($conn, 'SELECT text FROM nls_test');
14    oci_execute($stmt);
15    while (($row = oci_fetch_row($stmt)) != false) {
16      echo $row[0];
17    }
18    oci_free_statement($stmt);
19    oci_close($conn);
20  ?>
21 </body>
22 </html>
```

ąęśćźźńół ĄĘŚĆŹŹŃÓŁ

ąęśćźźńół ĄĘŚĆŹŹŃÓŁ

length : 557 lines : 23 Ln : 23 Col : 1 Sel : 0 | 0 Dos\Windows ANSI as UTF-8

Poprawne wyświetlanie znaków narodowych

- W jakiej stronie kodowej ORACLE zapisuje dane?

```
C:\Windows\system32\cmd.exe
SQL>
SQL> SELECT * FROM V$NLS_PARAMETERS;

PARAMETER                                VALUE
-----
NLS_LANGUAGE                             POLISH
NLS_TERRITORY                             POLAND
NLS_CURRENCY                             zł
NLS_ISO_CURRENCY                         POLAND
NLS_NUMERIC_CHARACTERS                   'GREGORIAN'
NLS_CALENDAR                             RR/MM/DD
NLS_DATE_FORMAT                          POLISH
NLS_DATE_LANGUAGE                        POLISH
NLS_CHARACTERSET                         EE8MSWIN1250
NLS_SORT                                 POLISH
NLS_TIME_FORMAT                          HH24:MI:SSXFF
NLS_TIMESTAMP_FORMAT                     RR/MM/DD HH24:MI:SSXFF
NLS_TIME_TZ_FORMAT                       HH24:MI:SSXFF TZR
NLS_TIMESTAMP_TZ_FORMAT                  RR/MM/DD HH24:MI:SSXFF TZR
NLS_DUAL_CURRENCY                        zł
NLS_NCHAR_CHARACTERSET                   AL16UTF16
NLS_COMP                                 BINARY
NLS_LENGTH_SEMANTICS                     BYTE
NLS_NCHAR_CONV_EXCP                      FALSE

Wybrano wierszy: 19.
SQL>
```

Poprawne wyświetlanie znaków narodowych

- Zmienna NLS_LANG

- FAQ:

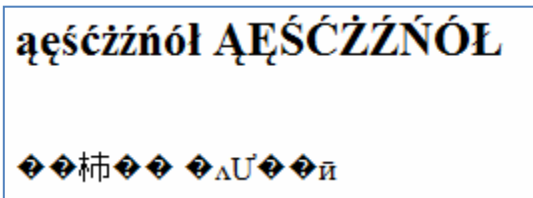
- (<http://www.oracle.com/technetwork/products/globalization/nls-lang-099431.html>)

- NLS_LANG = language_territory.charset

- charset : specifies the character set used by the client application. The character set defined with the NLS_LANG parameter does NOT CHANGE your client's character set. It is used to let Oracle know what character set you are USING on the client side, so Oracle can do the proper conversion.

- Błędne stawienie zmiennej

- putenv('NLS_LANG=POLISH_POLAND.EE8MSWIN1250');

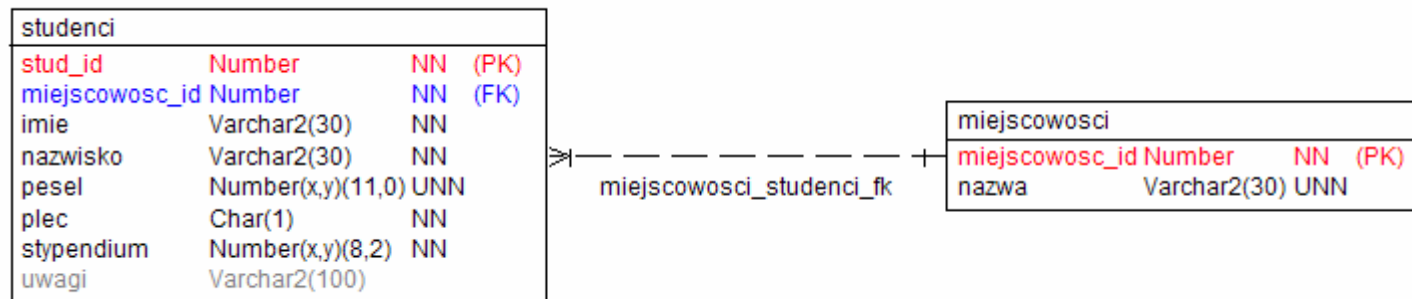


ąęśźńół ĄĘŚĆŹŹŃÓŁ

ąęśźńół ĄĘŚĆŹŹŃÓŁ

Ograniczenia bazodanowe

- Model
 - większość kolumn ma ograniczenie NOT NULL
 - kolumna 'plec' ma ograniczenie CHECK (nie widać na rysunku)
 - kolumna 'stypendium' ma ograniczenie DEFAULT (wartość domyślna = 0, nie widać na rysunku)
 - ograniczenie UNIQUE oznaczane jest literą 'U'
 - ograniczenie klucza obcego ma nadaną nazwę



Ograniczenia bazodanowe

```
DROP TABLE studenci;
DROP TABLE miejscowosci;

CREATE TABLE studenci (
    stud_id          NUMBER          NOT NULL,
    miejscowosc_id   NUMBER          NOT NULL,
    imie             VARCHAR2 (30)   NOT NULL,
    nazwisko         VARCHAR2 (30)   NOT NULL,
    pesel            NUMBER (11,0)   NOT NULL,
    plec             CHAR (1)        NOT NULL,
    stypendium       NUMBER (8,2)    DEFAULT 0 NOT NULL,
    uwagi           VARCHAR2 (100),
    CONSTRAINT studenci_pk PRIMARY KEY (stud_id),
    CONSTRAINT studenci_plec_ck CHECK (plec IN ('K', 'M')),
    CONSTRAINT studenci_pesel_uq UNIQUE (pesel)
);

CREATE TABLE miejscowosci (
    miejscowosc_id   NUMBER NOT NULL,
    nazwa            VARCHAR2 (30),
    CONSTRAINT miejscowosci_pk PRIMARY KEY (miejscowosc_id),
    CONSTRAINT miejscowosci_nazwa_uq UNIQUE (nazwa)
);

ALTER TABLE studenci
    ADD CONSTRAINT miejscowosci_studenci_fk
    FOREIGN KEY (miejscowosc_id)
    REFERENCES miejscowosci (miejscowosc_id);
```

Nazwy tabel i kolumn najlepiej bez cudzysłówów (cudzysłów powoduje, że ORACLE zachowuje wielkość liter. Raczej rzadko będziemy potrzebowali tą funkcjonalność)

W ORACLE długość wszelkich nazw jest ograniczona do 30 znaków. Czasami trzeba więc stosować (sensowne!) skróty

Ograniczenia bazodanowe

- Listowanie ograniczeń

```
SET LONG 30

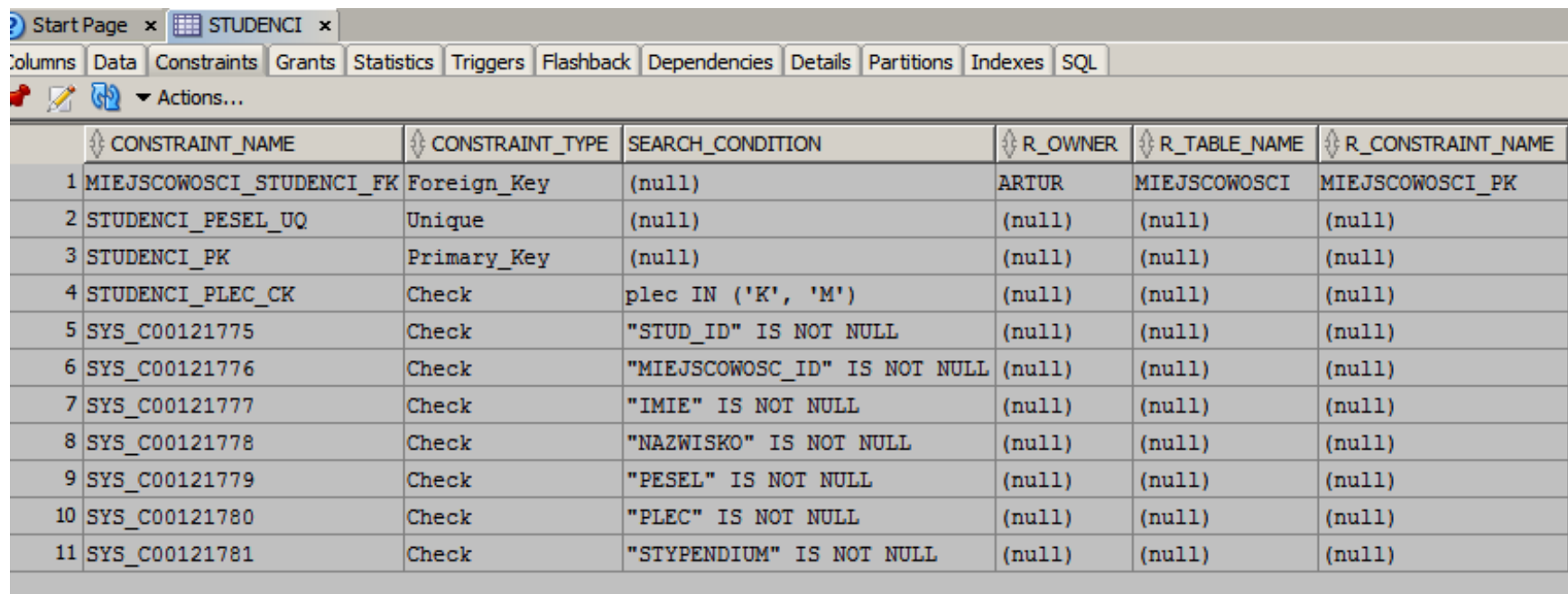
SELECT
    constraint_name, constraint_type, table_name, search_condition, r_constraint_name
FROM
    user_constraints
WHERE
    table_name IN ('STUDENCI', 'MIEJSCOWOSCI');
```

CONSTRAINT_NAME	C	TABLE_NAME	SEARCH_CONDITION	R_CONSTRAINT_NAME
SYS_C00121729	C	MIEJSCOWOSCI	"MIEJSCOWOSC_ID" IS NOT NULL	
MIEJSCOWOSCI_PK	P	MIEJSCOWOSCI		
MIEJSCOWOSCI_NAZWA_UQ	U	MIEJSCOWOSCI		
SYS_C00121719	C	STUDENCI	"STUD_ID" IS NOT NULL	
SYS_C00121720	C	STUDENCI	"MIEJSCOWOSC_ID" IS NOT NULL	
SYS_C00121721	C	STUDENCI	"IMIE" IS NOT NULL	
SYS_C00121722	C	STUDENCI	"NAZWISKO" IS NOT NULL	
SYS_C00121723	C	STUDENCI	"PESEL" IS NOT NULL	
SYS_C00121724	C	STUDENCI	"PLEC" IS NOT NULL	
SYS_C00121725	C	STUDENCI	"STYPENDIUM" IS NOT NULL	
STUDENCI_PLEC_CK	C	STUDENCI	plec IN ('K', 'M')	
STUDENCI_PK	P	STUDENCI		
STUDENCI_PESEL_UQ	U	STUDENCI		
MIEJSCOWOSCI_STUDENCI_FK	R	STUDENCI		MIEJSCOWOSCI_PK

14 rows selected.

Ograniczenia bazodanowe

- Listowanie ograniczeń, SQL Developer



The screenshot shows the SQL Developer interface with the 'STUDENCI' table selected. The 'Constraints' tab is active, displaying a list of 11 constraints. The table has columns: CONSTRAINT_NAME, CONSTRAINT_TYPE, SEARCH_CONDITION, R_OWNER, R_TABLE_NAME, and R_CONSTRAINT_NAME.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	MIEJSCOWOSCI_STUDENCI_FK	Foreign_Key	(null)	ARTUR	MIEJSCOWOSCI	MIEJSCOWOSCI_PK
2	STUDENCI_PESSEL_UQ	Unique	(null)	(null)	(null)	(null)
3	STUDENCI_PK	Primary_Key	(null)	(null)	(null)	(null)
4	STUDENCI_PLEC_CHK	Check	plec IN ('K', 'M')	(null)	(null)	(null)
5	SYS_C00121775	Check	"STUD_ID" IS NOT NULL	(null)	(null)	(null)
6	SYS_C00121776	Check	"MIEJSCOWOSC_ID" IS NOT NULL	(null)	(null)	(null)
7	SYS_C00121777	Check	"IMIE" IS NOT NULL	(null)	(null)	(null)
8	SYS_C00121778	Check	"NAZWISKO" IS NOT NULL	(null)	(null)	(null)
9	SYS_C00121779	Check	"PESEL" IS NOT NULL	(null)	(null)	(null)
10	SYS_C00121780	Check	"PLEC" IS NOT NULL	(null)	(null)	(null)
11	SYS_C00121781	Check	"STYPENDIUM" IS NOT NULL	(null)	(null)	(null)

Ograniczenia bazodanowe

- Naruszanie ograniczeń

```
INSERT INTO miejscowosci VALUES (1, 'Zielona Góra');
INSERT INTO studenci VALUES (1, 1, 'Artur', 'Gramacki', '9999999999', 'M', 0, NULL);
COMMIT;
```

```
SQL> -- Naruszenie ograniczenia PK
SQL> INSERT INTO miejscowosci VALUES (1, 'Wrocław');
INSERT INTO miejscowosci VALUES (1, 'Wrocław')
*
ERROR at line 1:
ORA-00001: unique constraint (ARTUR.MIEJSCOWOSCI_PK) violated
```

```
SQL> -- Naruszenie ograniczenia CK
SQL> INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '9999999999', 'X', 0, NULL);
INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '9999999999', 'X', 0, NULL)
*
ERROR at line 1:
ORA-02290: check constraint (ARTUR.STUDENCI_PLEC_CK) violated
```

```
SQL> -- Naruszenie ograniczenia UQ
SQL> INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '9999999999', 'M', 0, NULL);
INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '9999999999', 'M', 0, NULL)
*
ERROR at line 1:
ORA-00001: unique constraint (ARTUR.STUDENCI_PESSEL_UQ) violated
```

Ograniczenia bazodanowe

- Naruszanie ograniczeń

```
INSERT INTO miejscowosci VALUES (1, 'Zielona Góra');
INSERT INTO studenci VALUES (1, 1, 'Artur', 'Gramacki', '9999999999', 'M', 0, NULL);
COMMIT;
```

```
SQL> -- Naruszenie ograniczenia FK
SQL> INSERT INTO studenci VALUES (2, 3, 'Artur', 'Gramacki', '1111111111', 'M', 0, NULL);
INSERT INTO studenci VALUES (2, 3, 'Artur', 'Gramacki', '1111111111', 'M', 0, NULL)
*
ERROR at line 1:
ORA-02291: integrity constraint (ARTUR.MIEJSCOWOSCI_STUDENCI_FK) violated - parent key not found
```

```
SQL> -- Naruszenie ograniczenia NN
SQL> INSERT INTO studenci VALUES (2, 1, NULL, NULL, '1111111111', 'M', 0, NULL);
INSERT INTO studenci VALUES (2, 1, NULL, NULL, '1111111111', 'M', 0, NULL)
*
ERROR at line 1:
ORA-01400: cannot insert NULL into ("ARTUR"."STUDENCI"."IMIE")
```


Ograniczenia bazodanowe

- Nazwy ograniczeń domyślne

```
DROP TABLE studenci;
DROP TABLE miejscowosci;

CREATE TABLE studenci (
    stud_id          NUMBER          NOT NULL PRIMARY KEY,
    miejscowosc_id   NUMBER          NOT NULL,
    imie             VARCHAR2 (30)   NOT NULL,
    nazwisko         VARCHAR2 (30)   NOT NULL,
    pesel            NUMBER (11,0)   NOT NULL UNIQUE,
    plec             CHAR (1)        NOT NULL CHECK (plec IN ('K', 'M')),
    stypendium        NUMBER (8,2)    DEFAULT 0 NOT NULL,
    uwagi            VARCHAR2 (100)
);

CREATE TABLE miejscowosci (
    miejscowosc_id   NUMBER NOT NULL PRIMARY KEY,
    nazwa            VARCHAR2 (30) UNIQUE
);

ALTER TABLE studenci
    ADD FOREIGN KEY (miejscowosc_id)
    REFERENCES miejscowosci (miejscowosc_id);
```

Ograniczenia bazodanowe

- Listowanie ograniczeń

CONSTRAINT_NAME	C	TABLE_NAME	SEARCH_CONDITION	R_CONSTRAINT_NAME
SYS_C00121771	C	MIEJSCOWOSCI	"MIEJSCOWOSC_ID" IS NOT NULL	
SYS_C00121772	P	MIEJSCOWOSCI		
SYS_C00121773	U	MIEJSCOWOSCI		
SYS_C00121763	C	STUDENCI	"IMIE" IS NOT NULL	
SYS_C00121764	C	STUDENCI	"NAZWISKO" IS NOT NULL	
SYS_C00121765	C	STUDENCI	"PESEL" IS NOT NULL	
SYS_C00121766	C	STUDENCI	"PLEC" IS NOT NULL	
SYS_C00121767	C	STUDENCI	"STYPENDIUM" IS NOT NULL	
SYS_C00121768	C	STUDENCI	plec IN ('K', 'M')	
SYS_C00121769	P	STUDENCI		
SYS_C00121770	U	STUDENCI		
SYS_C00121774	R	STUDENCI		SYS_C00121772
SYS_C00121761	C	STUDENCI	"STUD_ID" IS NOT NULL	
SYS_C00121762	C	STUDENCI	"MIEJSCOWOSC_ID" IS NOT NULL	

14 rows selected.

```
SQL> INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '99999999999', 'X', 0, NULL);
INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '99999999999', 'X', 0, NULL)
```

*

ERROR at line 1:

ORA-02290: check constraint (ARTUR.SYS_C00121768) violated

```
SQL> INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '99999999999', 'M', 0, NULL);
```

```
INSERT INTO studenci VALUES (2, 1, 'Artur', 'Gramacki', '99999999999', 'M', 0, NULL)
```

*

ERROR at line 1:

ORA-00001: unique constraint (ARTUR.SYS_C00121770) violated