



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE

## Dostęp do baz danych

Piotr Wydrych

<http://kt.agh.edu.pl/~wydrych/>

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki  
Katedra Telekomunikacji

14 maja 2009

1/42

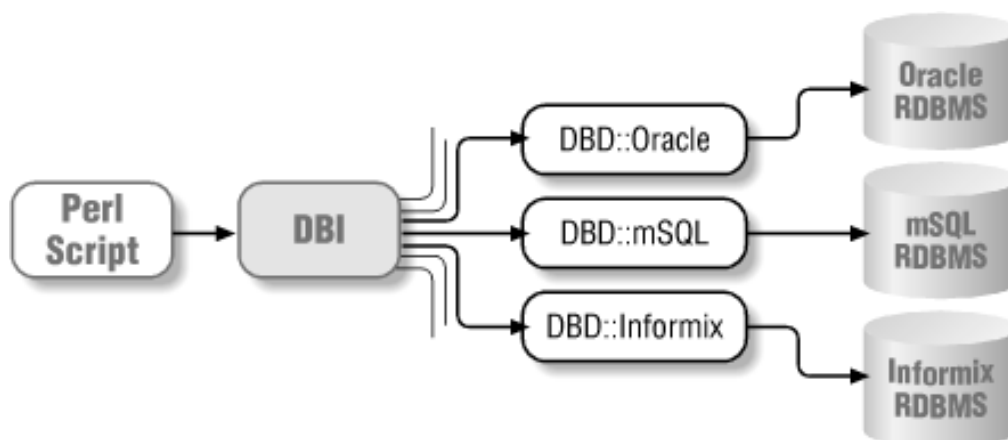


## AGH DBI — podstawowe informacje

- DataBase Interface.
- Standardowy interfejs dostępu do bazy danych niezależny od systemu zarządzania relacyjną bazą danych.

5/42

- DataBase Driver.
- Niezależne sterowniki do poszczególnych systemów (baz danych).
- Przykładowo: DBD::mysql, DBD::Oracle, DBD::Pg.
- Obecnie (listopad 2008) około 200 różnych dostępnych do pobrania z CPAN.



Źródło: Alligator Descartes, Tim Bunce: [Programming the Perl DBI](#). O'Reilly, 2000. ISBN 1-56592-699-4.

- Niektóre systemy operacyjne instalują DBI razem z Perlem.
- Debian, Ubuntu — pakiety libdbi-perl oraz libdbd-\*perl.
- FreeBSD, OpenBSD — p5-DBI oraz p5-DBD-\*.
- Solaris — CSWpmdbi oraz CSWpmdbd\*.
- Ogólnie — [dbi.perl.org](http://dbi.perl.org).

- + Łatwo zinterpretować gołym okiem.
- + Bez instalowania dodatkowych modułów.
- **Każdy** skrypt musi znać **dokładną** strukturę danych.
- Utrudniona aktualizacja.
- Brak transakcji.
- Brak powiązań, kluczy obcych, złączeń.

### Przykład

```
root:x:0:0:Super-User:/root:/bin/bash
```

- Dodawanie danych:

```
open F, '>>_/etc/passwd'  
and printf F '%s:...', @user{login, ...};
```

- Pobieranie danych:

```
open F, '<_/etc/passwd'  
and %data = map  
{ /(\w+):.../ ? ($1 => {uid => $2, ...}) : () } <F>;
```

- Usuwanie, aktualizacja = pobranie + zapis **całości**.

- + Łatwo zinterpretować gołym okiem.
- + Bez instalowania dodatkowych modułów.
- + Łatwe zapisywanie i pobieranie danych.
- Wady — jak dla zwykłego pliku.

- Zapisywanie danych:

```
open D, '>_data.dump'  
and print D Data::Dumper->Dump([$data], ['data']);
```

- Pobieranie danych:

```
open D, '<_data.dump'  
and eval join '', <D>;
```

- Wady i zalety DBI  $\approx$  wady i zalety baz danych.
- + Transakcje, złożenia, klucze, widoki itd.
- + Bezpieczeństwo i wydajność.
- Konieczność instalowania dodatkowego oprogramowania ...
- + ... które często i tak instalujemy!

### Wniosek

```
use DBI;
```

16/42

### connect

```
$dbh = DBI->connect($data_source, $user, $auth)  
or die $DBI::errstr;  
$dbh = DBI->connect($data_source, $user, $auth, \%attr)  
or die $DBI::errstr;
```

- `$data_source` — adres źródła danych zaczynający się od `dbi:STEROWNIK:`,  
np. `dbi:mysql:database=mydb:host=sunum.`
- `$user` i `$auth` — nazwa użytkownika i hasło.
- `\%attr` — właściwości połączenia:
  - Przykładowo `{ PrintError => 0 }`.
  - Podstawowe w dokumentacji DBI.
  - Pełna lista zależna od użytego DBD.

19/42



## AGH Zapytania bez zwracania danych

### do

```
$rows = $dbh->do($statement)           or die $dbh->errstr;  
$rows = $dbh->do($statement, \%attr) or die $dbh->errstr;  
$rows = $dbh->do($statement, \%attr, @bind_values)  
           or die $dbh->errstr;
```

- Przygotowuje i wykonuje zapytanie.
- Zwraca liczbę wierszy (np. skasowanych).
- Nie nadaje się do zapytań SELECT.
- `$statement` — zapytanie do wykonania.
- `@bind_values` — jeżeli zapytanie zawiera pytańniki, zamiast nich wstawiane są w bezpieczny sposób kolejne wartości z tablicy.

21/42



## AGH Zapytania bez zwracania danych Prosty przykład

### Przykład

```
$sql = 'CREATE_TABLE_dbmss_  
_name_VARCHAR(20)_PRIMARY_KEY,  
_newest_version_VARCHAR(20)_NOT_NULL,  
_web_page_VARCHAR(255)  
);';  
  
$dbh->do($sql)  
           or die $dbh->errstr;
```

22/42

### Przykład

```
$sql = 'UPDATE_dbmss_SET_newest_version=?  
_WHERE_name=?';  
  
$dbh->do($sql, undef, 5.1, "MySQL_('C.Ed.')" )  
    or die $dbh->errstr;
```

### Rzeczywiście wykonane zapytanie

```
UPDATE dbmss SET newest_version = '5.1'  
WHERE name = 'MySQL (\'C.Ed.\')';
```

Dwie metody:

- z odrębnym przygotowaniem zapytania,
- bez odrębnego przygotowania zapytania (czyli ze zintegrowanym przygotowaniem).



## AGH Przygotowywanie zapytania

### prepare

```
$sth = $dbh->prepare($statement)           or die $dbh->errstr;  
$sth = $dbh->prepare($statement, \%attr) or die $dbh->errstr;
```

- Przygotowuje zapytanie do późniejszego (być może wielokrotnego) wykonania.
- Zwraca uchwyt do przygotowanego zapytania.
- `$statement` — zapytanie do wykonania.

27/42



## AGH Wykonywanie przygotowanego zapytania

### execute

```
$rv = $sth->execute           or die $sth->errstr;  
$rv = $sth->execute(@bind_values) or die $sth->errstr;
```

- Rozpoczyna wykonywanie zapytania.
- Zwraca prawdę jeżeli zapytanie się powiodło.
- Raz przygotowane zapytanie można wykonywać wiele razy.
- Nie zwraca liczby wierszy ...
- ... ponieważ w momencie rozpoczęcia wykonywania zapytania nie wiadomo ile ich będzie!
- `@bind_values` — używane jeżeli podczas przygotowywania zapytania użyte zostały pytajniki.

28/42

#### fetchrow\_\*

```
$ary_ref = $sth->fetchrow_arrayref;  
$ary_ref = $sth->fetch;  
  
@ary = $sth->fetchrow_array;  
  
$hash_ref = $sth->fetchrow_hashref;  
$hash_ref = $sth->fetchrow_hashref($name);
```

- Zwracają odpowiednio referencję do tablicy, tablicę, referencję do hasha.
- Jeżeli nie ma więcej wierszy zwracają wartość niezdefiniowaną.
- \$name pozwala wybrać wielkość liter w kluczach hasha.

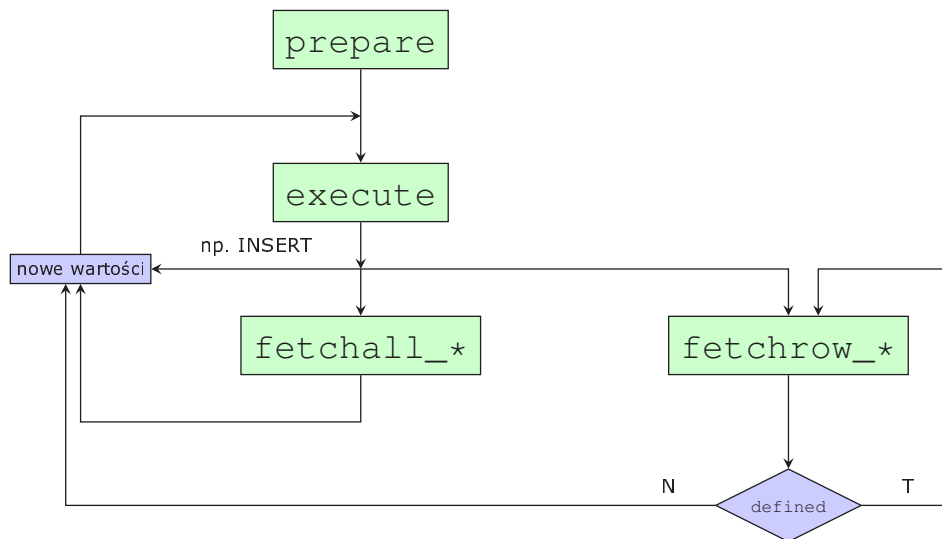
29/42

#### fetchall\_\*

```
$tbl_ary_ref = $sth->fetchall_arrayref;  
$tbl_ary_ref = $sth->fetchall_arrayref($slice);  
$tbl_ary_ref = $sth->fetchall_arrayref($slice, $max_rows);  
  
$hash_ref = $sth->fetchall_hashref($key_field);
```

- `fetchall_arrayref` zwraca referencję do tablicy referencji.
  - \$slice wybiera format wierszy (referencja do tablicy lub hasha) oraz ogranicza pobierane kolumny.
- `fetchall_hashref` zwraca referencję do hasha referencji do hashy.
  - Używa \$key\_field jako klucza.

30/42



### prepare, execute, fetchrow

```

$sth = $dbh->prepare(
    'SELECT_name,_newest_version_FROM_dbmss_WHERE_name_LIKE_?');

$sth->execute('%SQL') or die $sth->errstr;

printf "%s_%s\n", $ref->{name}, $ref->{newest_version}
while $ref = $sth->fetchrow_hashref;

```

### selectrow\_\*

```
@ary = $dbh->selectrow_array($statement);  
@ary = $dbh->selectrow_array($statement, \%attr);  
@ary = $dbh->selectrow_array($statement, \%attr, @bind_values);  
  
$ary_ref = $dbh->selectrow_arrayref(...);  
$hash_ref = $dbh->selectrow_hashref(...);
```

- Pobiera jeden wiersz (jedną krotkę).
- prepare + execute + raz wykonane fetchrow\_\*.
- \$statement może być wcześniej przygotowanym przez funkcję prepare zapytaniem.

### selectcol\_arrayref

```
$ary_ref = $dbh->selectcol_arrayref($statement);  
$ary_ref = $dbh->selectcol_arrayref($statement, \%attr);  
$ary_ref = $dbh->selectcol_arrayref($statement, \%attr,  
                                   @bind_values);
```



- Pobiera dane z jednej kolumny, ze wszystkich wierszy.
- prepare + execute + fetch w pętli.
- \$statement może być wcześniej przygotowanym przez funkcję prepare zapytaniem.
- Atrybut Columns umożliwia pobranie danych z innej niż pierwsza kolumny, a także z więcej niż jednej kolumny (nadal do jednej tabeli!)

### selectall\_\*

```
$ary_ref = $dbh->selectall_arrayref($statement);  
$ary_ref = $dbh->selectall_arrayref($statement, \%attr);  
$ary_ref = $dbh->selectall_arrayref($statement, \%attr,  
                                   @bind_values);  
  
$hash_ref = $dbh->selectall_hashref($statement, $key_field);  
$hash_ref = $dbh->selectall_hashref($statement, $key_field,  
                                   \%attr);  
$hash_ref = $dbh->selectall_hashref($statement, $key_field,  
                                   \%attr, @bind_values);
```

- `prepare + execute + fetchall_*`.
- Używając atrybut `Slice` pozwala wybrać format wierszy.

- `$sth->finish`
  - Informuje serwer, że z danego zapytania nie będą pobierane już żadne dane.
  - **Tylko** jeżeli skrypt nie pobrał wszystkich krotek.
- `$dbh->disconnect` or `warn $dbh->errstr`
  - Rozłącza połączenie z bazą danych.
  - Jeżeli `AutoCommit` jest wyłączone działanie jest zależne od silnika bazy danych.
  - Powinno zostać poprzedzone `$dbh->commit` lub `$dbh->rollback`.
- <http://search.cpan.org/~timb/DBI/DBI.pm>

-  T. Bunce.  
DBI - Database independent interface for Perl.  
<http://search.cpan.org/~timb/DBI/DBI.pm>.
-  A. Descartes and T. Bunce.  
Programming the Perl DBI.  
O'Reilly, 2000.