

Perl Mongers im Ruhrgebiet



# Ruhr . pm

## Einfuehrung in Class::DBI

**Autor:** Simon Wilper

**E-Mail:** simon AT ruhr.pm.org

**Datum:** 10. April 2007

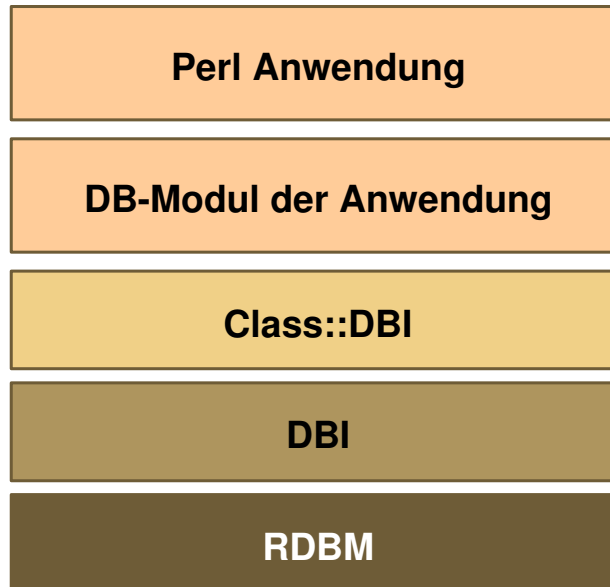
<http://ruhr.pm.org/>



# Ruhr . pm

## Was ist Class::DBI?

- Einfache Datenbankabstraktionsschicht
- »Mapping« von Datenbankobjekten auf Perl-Objekte





# Ruhr . pm

## Vor- / Nachteile

- + Strukturiertes Ansatz
- + Erhöhte Wartbarkeit
- + Einfache Erweiterbarkeit
- + 'Nachbau' von Datenbankeigenschaften
- "Cost of Abstraction"
- Initialaufwand



# Ruhr . pm

---

## Voraussetzungen

- DBI
- DBI-Treiber des verwendeten RDBM
- `Class::DBI` (mittels CPAN nachinstallieren)
- Vorhandene Datenbank / Tabellen



# Ruhr.pm

## Beispiel: Schritt 1: Adressdatenbank initialisieren

- Initialisieren des “Top Level Objekts”

```
#!/usr/bin/perl  
use warnings;  
use strict;
```

```
package Addresses::DBI;  
use base 'Class::DBI';
```

- Verbindung zum RDBM aufbauen

```
Addresses::DBI->connection(  
    'dbi:Pg:dbname=addresses', # DBName  
    '', # User  
    '', # Password  
    undef # Options  
);
```



# Ruhr.pm

## Schritt 2: Tabellenobjekt initialisieren

- Tabellenobjekt `Addresses::Names` erbt von `Addresses::DBI`

```
package Addresses::Names;  
use base 'Addresses::DBI';
```

- Tabellenname und Spalten zuweisen

```
Addresses::Names->table( 'names' );  
Addresses::Names->columns( All => qw(id first last email) );
```

Bei erster Spalte wird Primaerschlüssel angenommen



# Ruhr . pm

---

## Schritt 3: Fertig

- Folgende Methoden zur Manipulation stehen zur Verfügung:
  - retrieve / search
  - insert
  - update
  - delete



# Ruhr.pm

## Einfacher Retrieve

- Retrieve setzt Primaerschlüssel voraus

```
my $Row = Addresses::Names->retrieve( $id );

printf( "% 5d %-20s %-20s %-30s\n",
        $Row->id,
        $Row->first,
        $Row->last,
        $Row->email
    }
```





# Ruhr.pm

## Search

- **Iterator:** Guenstig, um grosse Datenmengen abzufragen

```
my $Iterator = Addresses::Names->retrieve_all;
my $Iterator = Addresses::Names->search( id => '26123' );
my $Iterator = Addresses::Names->search_like( id => '26%' );

while ( my $Row = $Iterator->next ) {
    printf( "% 5d %-20s %-20s %-30s\n",
            $Row->id,
            $Row->first,
            $Row->last,
            $Row->email
    );
}
```



# Ruhr . pm

## Insert

- Automatisch zu generierende Werte werden erkannt

```
my $Data = {  
    first => 'Dummy Vorname',  
    last  => 'Dummy Nachname',  
    email => 'Dummy@nowhere.org'  
};
```

```
my $Row = Addresses::Names->insert( $Data );
```



# Ruhr . pm

## Update

- Jedes Datensatzobjekt verfügt ueber Accessor-Methoden

```
my $Row = Addresses::Names->retrieve( $id );
```

```
$Row->set( first => 'New Dummy Name' );
```

```
$Row->update;
```



# Ruhr . pm

## Delete

- delete()-Methode des Datensatzobjekts aufrufen

```
my $Row = Addresses::Names->retrieve( $id );  
$Row->delete;  
  
print "Row $id deleted\n";
```



# Ruhr . pm

---

## Benchmark

- Ausgaber aller 27'000 Datensätze
- In der Class::DBI Version mit `retrieve_all()`
- In der klassischen Version mit `while fetchrow_hashref()`



# Ruhr.pm

## Profiler Class::DBI

Total Elapsed Time = **34.87068 Seconds**

User+System Time = 27.18068 Seconds

Exclusive Times

%Time	ExclSec	CumulS	#Calls	sec/call	Csec/c	Name
35.8	9.742	10.201	26999	0.0004	0.0004	Class::DBI::_fresh_init
9.40	2.554	3.886	917978	0.0000	0.0000	Class::DBI::Column::name_lc
8.73	2.374	6.260	917969	0.0000	0.0000	Class::DBI::Column::__ANON__
4.90	1.332	1.332	917998	0.0000	0.0000	Class::Accessor::Fast::__ANON__
4.63	1.259	21.959	134995	0.0000	0.0002	Class::DBI::get
3.84	1.044	16.245	26999	0.0000	0.0006	Class::DBI::_flesh
3.43	0.933	0.933	27000	0.0000	0.0000	DBI::st::execute
3.29	0.895	0.895	405022	0.0000	0.0000	Class::Data::Inheritable::__ANON__
2.95	0.803	0.803	26999	0.0000	0.0000	DBD::Pg::db::_ping
2.80	0.762	2.123	27000	0.0000	0.0001	Class::DBI::SQL::Transformer::_do_transformation
2.74	0.744	2.309	242991	0.0000	0.0000	Class::DBI::_attribute_exists
2.70	0.735	2.167	27000	0.0000	0.0001	DBIx::ContextualFetch::st::_untaint_execute
2.69	0.731	4.045	27000	0.0000	0.0001	Class::DBI::transform_sql
2.42	0.657	1.513	80998	0.0000	0.0000	Class::DBI::primary_column
2.39	0.650	1.595	134995	0.0000	0.0000	Class::DBI::ColumnGrouper::find_column



# Ruhr.pm

## Profiler Klassischer fetchrow\_hashref()

Total Elapsed Time = 1.573362 Seconds

User+System Time = 0.583362 Seconds

Exclusive Times

%Time	ExclSec	Cumuls	#Calls	sec/call	Csec/c	Name
40.6	0.237	0.392	27000	0.0000	0.0000	DBI::st::fetchrow_hashref
14.0	0.082	0.082	27003	0.0000	0.0000	DBI::common::FETCH
12.3	0.072	0.072	27000	0.0000	0.0000	DBI::st::fetch
1.71	0.010	0.010	1	0.0100	0.0100	DBD::Pg::db::_login
1.71	0.010	0.010	2	0.0050	0.0050	DBI::setup_driver
1.71	0.010	0.010	13	0.0008	0.0007	DBI::BEGIN
1.71	0.010	0.029	4	0.0024	0.0072	main::BEGIN
0.00	0.000	0.000	1	0.0000	0.0000	DynaLoader::bootstrap_inherit
0.00	0.000	0.000	1	0.0000	0.0000	Exporter::Heavy::heavy_export
0.00	0.000	0.000	1	0.0000	0.0000	Exporter::Heavy::heavy_require_version
0.00	-	-0.000	1	-	-	DBI::dr::disconnect_all
0.00	-	-0.000	1	-	-	DBD::Pg::st::_prepare
0.00	-	-0.000	1	-	-	List::Util::bootstrap
0.00	-	-0.000	1	-	-	DBI::DBI_tie::TIEHASH
0.00	-	-0.000	1	-	-	DBI::_firesafe::BEGIN