Fórum Internacional Software Livre 8.0

PEAR::MDB2_Schema

Igor Feghali [ifeghali@php.net]

Dropline Fun Theme Wallpaper por Silvestre Herrera

Quem sou eu ?

- Graduando em Engenharia de Computação pela Universidade Federal do Espírito Santo (UFES)
- Participante do Google Summer of Code 2006 e 2007* pela organização PHP
- Autônomo no desenvolvimento de sistemas Web baseados em PHP
- Desenvolvedor PEAR há 10 meses e desenvolvedor do pacote MDB2_Schema

Introdução

Proponho apresentar brevemente o pacote MDB2_Schema do PEAR focando em suas principais aplicações e demonstrando como utilizá-lo.

Darei uma visão geral do Google Summer of Code 2006 expondo as contribuições que ele trouxe para o referido projeto.

Tempo estimado: 40m + perguntas

Conteúdo

- Visão geral do GSoC 2006
- Introdução ao MDB2_Schema
- Prática 1
- DML e DDL
- XML Schema
- Prática 2
- Carregando o pacote
- Prática 3

Google Summer of Code é um programa que oferece apoio financeiro à estudantes para desenvolverem projetos pessoais baseados em software de código livre.

http://code.google.com/soc/

Dropline Fun Theme Wallpaper por Silvestre Herrera

Google

Aconteceu no período de Junho a Setembro de 2006 contando com 630 estudantes e 1200 mentores distribuídos em 90 países. 82% dos estudantes receberam uma avaliação positiva de seus mentores ao final do programa.

Como funciona ?

- Organizações interessadas se cadastram
- Organizações são selecionadas
- Estudantes enviam projetos dentro do contexto de uma organização
- Estudantes são selecionados
- Estudantes selecionados começam a trabalhar em seus projetos
- Desempenho dos estudantes é avaliado pela organização

Participantes PHP finalistas

http://code.google.com/soc/php/about.html

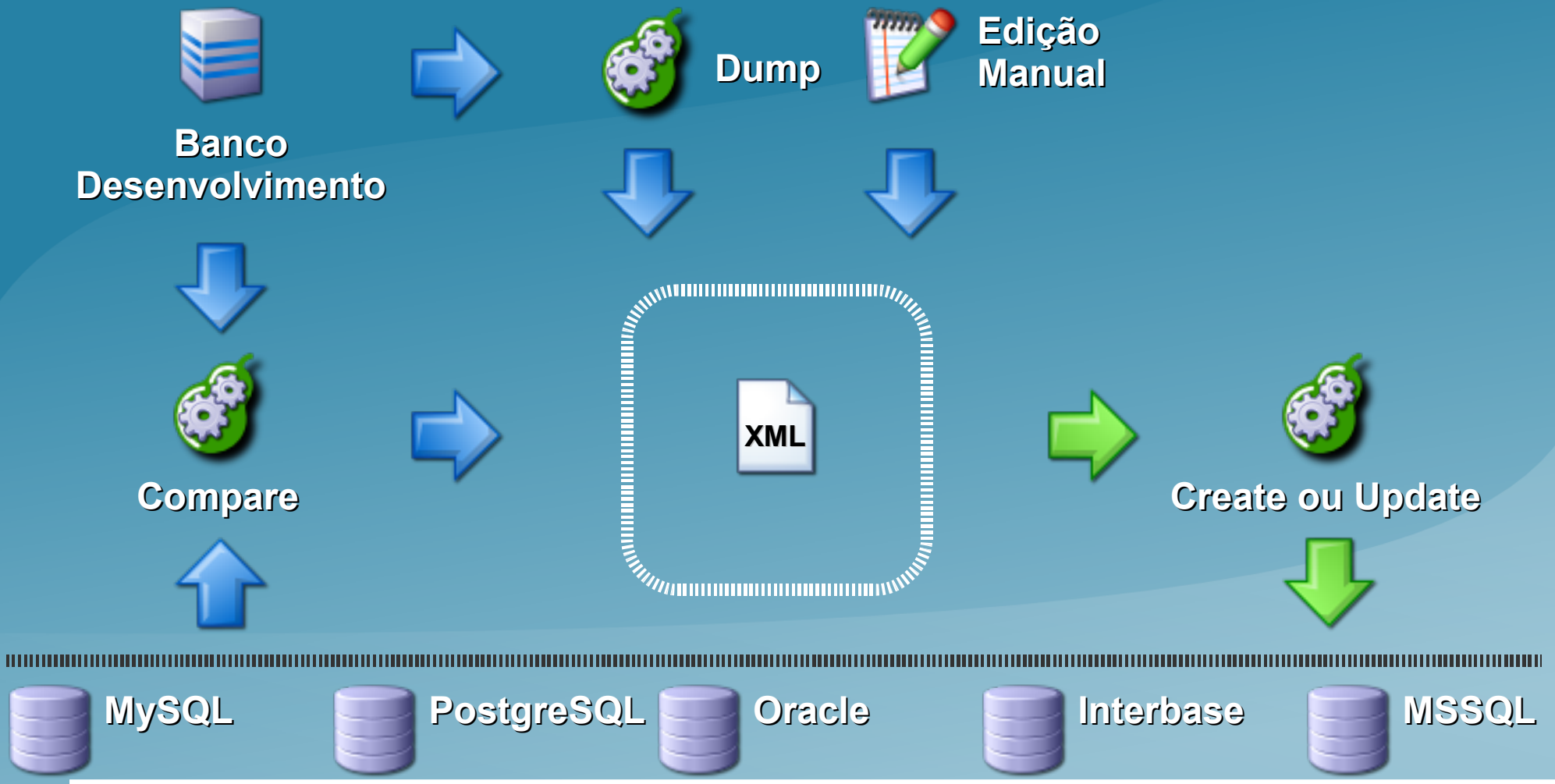
Dropline Fun Theme Wallpaper por Silvestre Herrera

PEAR::MDB2_Schema

É uma ferramenta que permite ao usuário manter esquemas de bancos de dados em arquivos XML independentes de DBMS. Em caminho inverso, estes arquivos podem ser usados para criar, alterar e excluir bancos de dados, bem como manipular as informações ali contidas.

http://pear.php.net/package/MDB2_Schema/

Visão geral do pacote



Visão geral do pacote

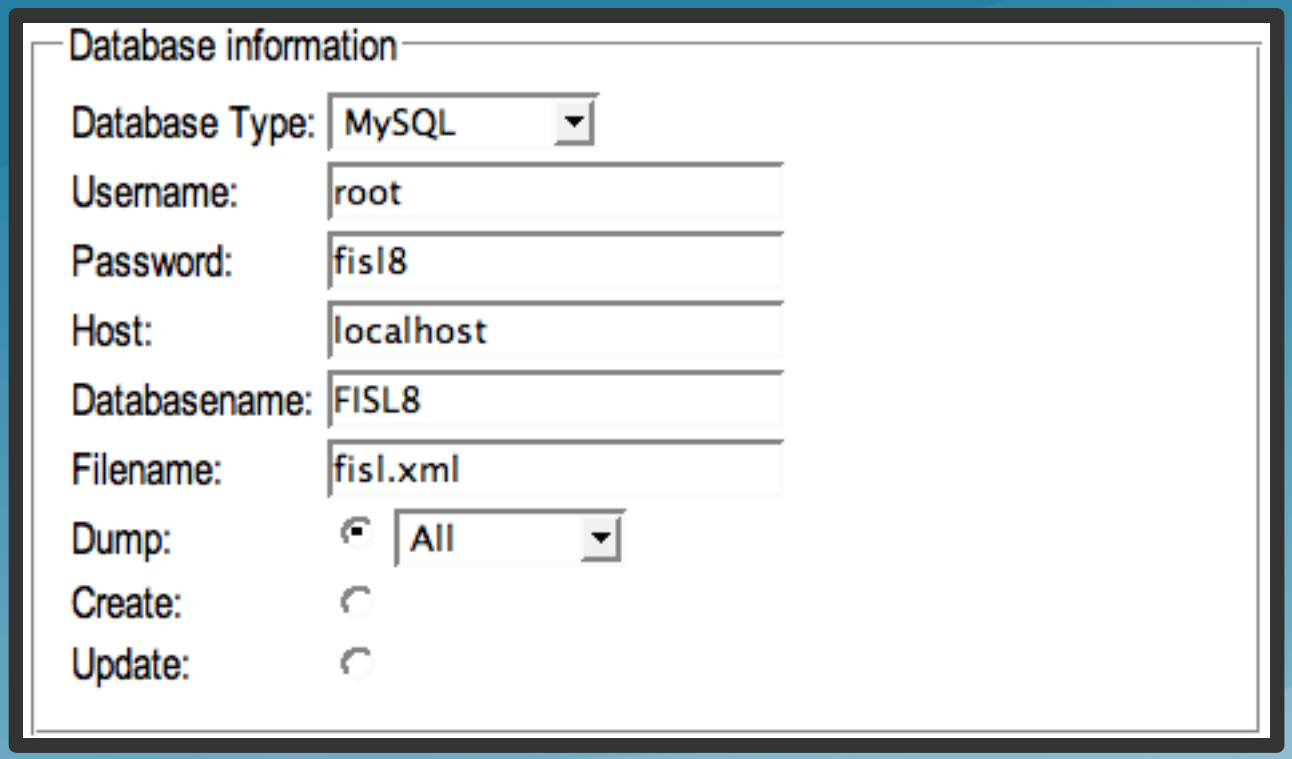
Aproveitando a camada de abstração criada pelo MDB2, uma das propostas do MDB2_Schema é ser uma ferramenta para auxiliar na atualização de esquema de banco de dados.

Pode-se concentrar o desenvolvimento de um banco de dados em um gerenciador específico e, através do MDB2_Schema, sincronizar bancos de dados clientes rodando em outros gerenciadores.

Prática 1

```
CREATE DATABASE FISL8;

USE FISL8;

CREATE TABLE inscritos (Codigo INT, Nome
 CHAR(16));
```

Prática 1



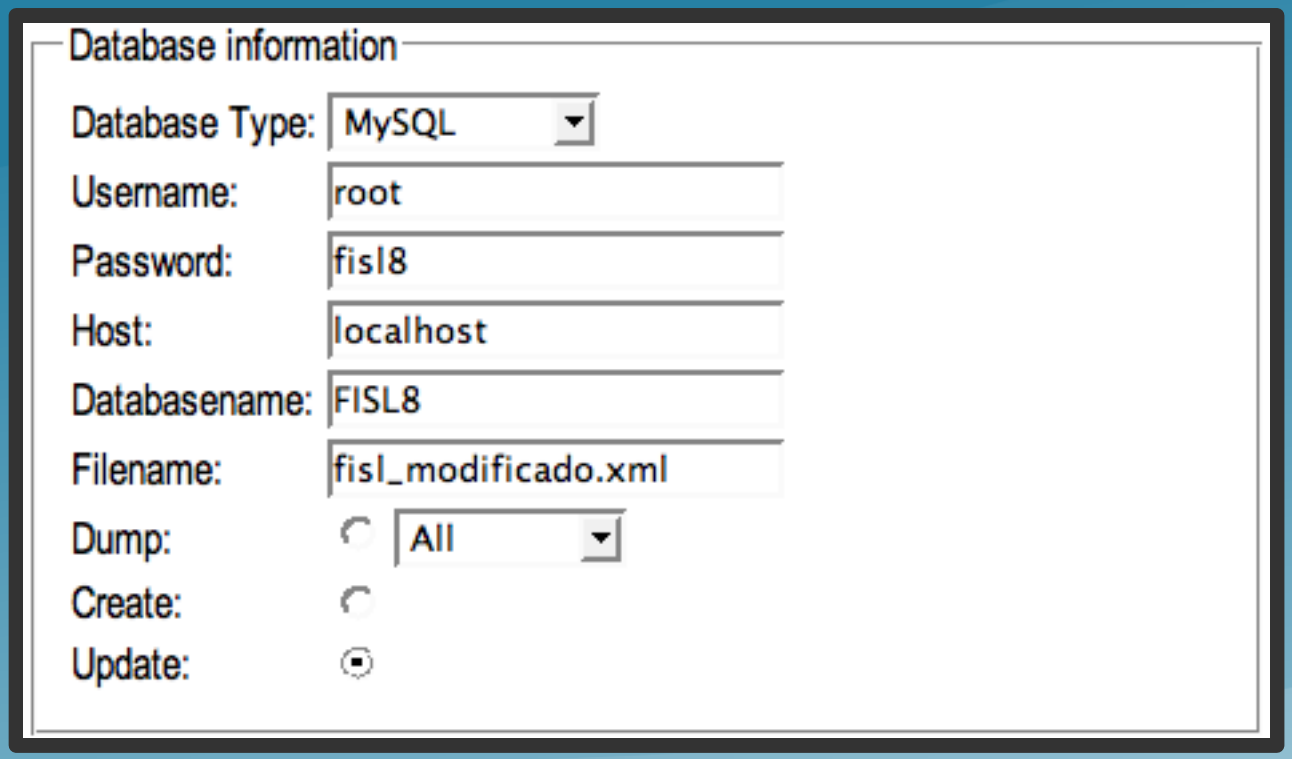
Prática 1

```
<database>
 <name>FISL8</name>
 <create>true</create>
 <overwrite>false</overwrite>

 <table>
  <name>inscritos</name>

  <declaration>

   <field>
    <name>Codigo</name>
    <type>integer</type>
    <length>4</length>
    <notnull>false</notnull>
    <default></default>
   </field>

   <field>
    <name>Nome</name>
    <type>text</type>
    <length>16</length>
    <notnull>false</notnull>
    <fixed>true</fixed>
    <default></default>
   </field>

  </declaration>

 </table>

</database>
```

Prática 1

```
$ diff fisl.xml fisl_modificado.xml
30a31,39
>     <field>
>       <name>Telefone</name>
>       <type>text</type>
>       <length>10</length>
>       <notnull>false</notnull>
>       <fixed>true</fixed>
>       <default></default>
>     </field>
```

Prática 1



Prática 1

```
Debug messages

query(1): SHOW COLUMNS FROM `inscritos`
query(1): SHOW COLUMNS FROM `inscritos` LIKE 'Codigo'
query(1): SHOW COLUMNS FROM `inscritos` LIKE 'Nome'
query(1): SHOW INDEX FROM `inscritos`
query(1): SHOW INDEX FROM `inscritos`
query(1): SHOW TABLES
query(1): SELECT Codigo, Nome FROM `inscritos`
query(1): SHOW DATABASES
begintransaction(1): Starting transaction/savepoint
query(1): START TRANSACTION
query(1): ALTER TABLE `inscritos` ADD `Telefone`
  CHAR(10) DEFAULT NULL
commit(1): Committing transaction/savepoint
query(1): COMMIT
dumpdatabasechanges(1): Added field 'Telefone'
```

DML e DDL

Até agora tudo é muito bonito...
mas e se a atualização estivesse removendo um campo ao invés de criá-lo ?

Para onde iriam os dados ?

DML e DDL

Terminologia

DDL – Data Definition Language
- CREATE
- ALTER
- DROP

DML – Data Manipulation Language
- INSERT
- UPDATE
- DELETE

O Desafio

Uma agenda de contatos

Versão 1.0

- Cada participante possui um único telefone

Versão 2.0

- Cada participante pode ter um número indefinido de telefones, através de uma tabela exclusiva.

O Desafio

```
CREATE TABLE telefones
    (CodigoInscrito INT, Telefone CHAR(10))

INSERT INTO telefones
    (SELECT Codigo, Telefone FROM inscritos)

ALTER TABLE inscritos
    DROP COLUMN Telefone
```

O Desafio

A adição de uma nova tabela, e a mudança na estrutura da tabela de participantes podem ser facilmente detectadas pelo MDB2_Schema.

O desafio está em atualizar um aplicativo, que já está em estado de produção, garantindo a permanência dos dados.

Suporte à DML

Sim, então precisamos inserir DML em nosso XML. Mas como ?

http://2006.planet-soc.com/blog/238

Suporte à DML

Primeira abordagem

A consulta pode ser escrita diretamente no XML:

```
<query>INSERT ... </query>
<query>UPDATE ... WHERE ...</query>
<query>DELETE ... WHERE ...</query>
```

Suporte à DML

Mas espere, isso não amarraria nosso arquivo à um DBMS específico ?

E a filosofia de nosso pacote não é justamente privilegiar a portabilidade ?

Suporte à DML

Segunda abordagem

Deve haver uma representação XML completa, para que possamos construir consultas de uma forma portável.

Suporte à DML (XML Schema)

```
<table>
  <declaration/>
  <initialization>
    <insert/>*
    <update/>*
    <delete/>*
  </initialization>
</table>
```

mais informações na documentação do pacote

INSERT

```
<insert>
  {field}+
</insert>

<field>
  <name/>
  <null/> ou <value/> ou <column/> ou
 {function} ou {expression}
</field>
```

INSERT (exemplo)

```
<insert>
  <field>
    <name>foo</name>
    <value>1601</value>
  </field>
  <field>
    <name>creation</name>
    <function>
      <name>NOW</name>
    </function>
  </field>
</insert>
```

UPDATE

```
<update>
  {field}+
  <where>
    {expression}
  </where>?
</update>

<expression>
  <v/> ou <c/> ou {f} ou {e}
  <operator/>
  <v/> or <c/> or {f} or {e}
</expression>
```

UPDATE (exemplo)

```
<update>
  <field>
    <name>id</name>
    <expression>
      <column>id</column>
      <operator>MINUS</operator>
      <value>10</value>
    </expression>
  </field>
</update>
```

DELETE

```
<delete>
  <where>
    {expression}
  </where>
</delete>

<function>
  <name/>
  ( <v/> or <c/> or {f} or {e} )+
</function>
```

DELETE (exemplo)

```
<delete>
  <where>
    <expression>
      <expression>
        <column>id</column>
        <operator>GREATER THAN</operator>
        <value>10</value>
      </expression>
      <operator>AND</operator>
      <expression>
        <column>id</column>
        <operator>NOT EQUAL</operator>
        <value>15</value>
      </expression>
    </expression>
  </where>
</delete>
```

Mais exemplos de expressões

datediff(`dataentrega`, `datapedido`) / 365

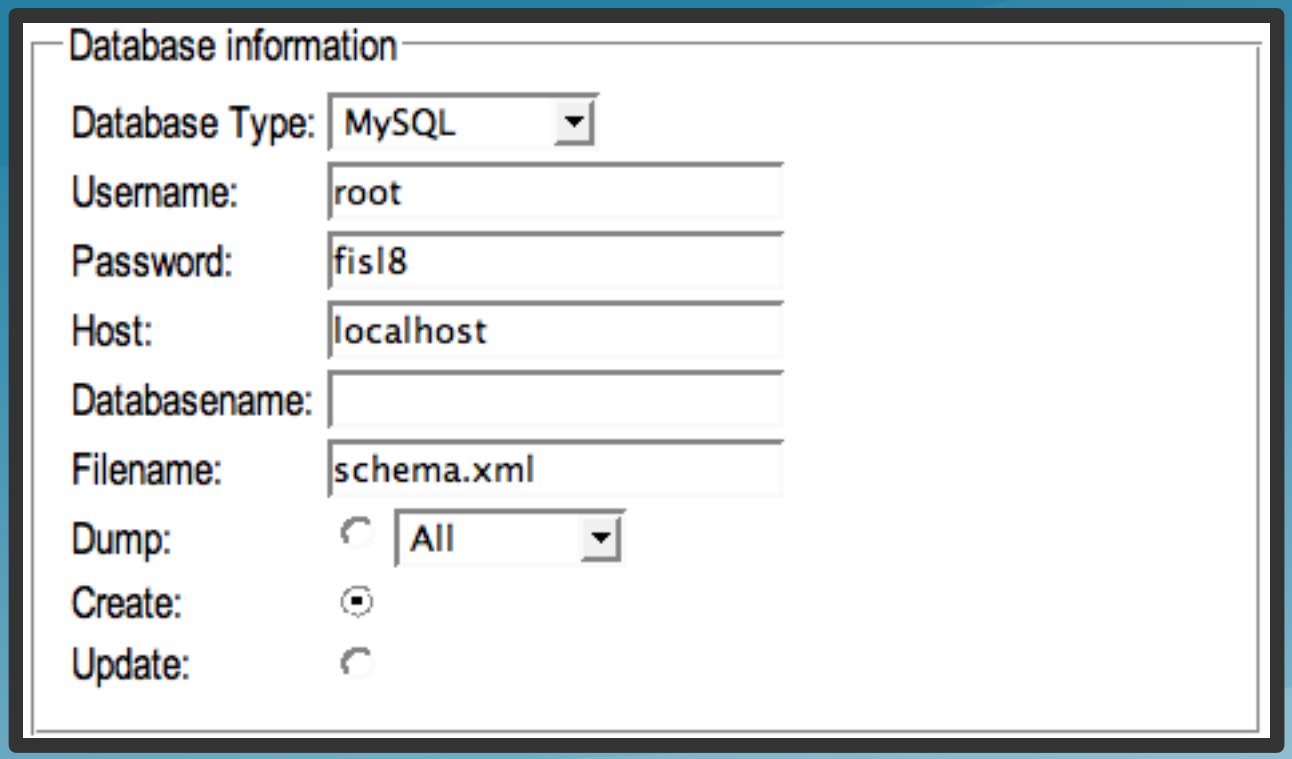
```
<expression>
  <function>
    <name>DATEDIFF</name>
    <column>dataentrega</column>
    <column>datapedido</column>
  </function>
  <operator>DIVIDED</operator>
  <value>365</value>
</expression>
```

Mais exemplos de expressões

(A ou B) e (C + 2)

```
<expression>
  <expression>
    <value>A</value>
    <operator>OR</operator>
    <value>B</value>
  </expression>
  <operator>AND</operator>
  <expression>
    <value>C</value>
    <operator>PLUS</operator>
    <value>2</value>
  </expression>
</expression>
```

Prática 2



Prática 2

```
bool(true)
Debug messages
query(1): SHOW DATABASES
query(1): CREATE DATABASE `MDB2Example`
begintransaction(1): Starting transaction/savepoint
query(1): START TRANSACTION
query(1): SHOW /*!50002 FULL*/ TABLES/*!50002 WHERE
  Table_type = 'BASE TABLE'*/
query(1): CREATE TABLE `People` (`id` INT UNSIGNED NOT
  NULL AUTO_INCREMENT PRIMARY KEY, `name` VARCHAR(128)
  DEFAULT 'unknown' NOT NULL, `age` INT UNSIGNED DEFAULT
  NULL, `birthdate` DATE DEFAULT NULL, `occupation`
  VARCHAR(128) DEFAULT NULL, `updated` DATETIME DEFAULT
  NULL, `aux` DOUBLE DEFAULT NULL)
```

Prática 2

```
query(1): INSERT INTO `People` (name, birthdate,
  occupation, updated) VALUES ('Igor', '1984-05-23',
  'engineer', CURRENT_TIMESTAMP)
query(1): INSERT INTO `People` (name, birthdate,
  occupation, updated) VALUES ('Heloisa', '1984-04-07',
  'lawyer', CURRENT_TIMESTAMP)
query(1): INSERT INTO `People` (name, birthdate,
  occupation, updated) VALUES ('Anne Sophie', '2004-01-
  16', 'engineer', CURRENT_TIMESTAMP)
query(1): INSERT INTO `People` (name, birthdate,
  occupation, updated) VALUES ('John', '2005-01-16',
  'philosopher', CURRENT_TIMESTAMP)
query(1): UPDATE `People` SET aux=(DATEDIFF(`updated`,
  `birthdate`) / '365'), age=floor(`aux`)
query(1): UPDATE `People` SET aux=NULL
```

Prática 2

```
query(1): SHOW /*!50002 FULL*/ TABLES/*!50002 WHERE
  Table_type = 'BASE TABLE'*/
query(1): CREATE TABLE `Packages` (`id` INT UNSIGNED NOT
  NULL AUTO_INCREMENT PRIMARY KEY, `name` VARCHAR(128)
  DEFAULT '' NOT NULL, `summary` LONGTEXT NOT NULL,
  `creation` DATETIME DEFAULT NULL)
query(1): INSERT INTO `Packages` (id, name, summary,
  creation) VALUES ('11', 'PEAR', 'PEAR Base System',
  '2006-08-16')
query(1): INSERT INTO `Packages` (id, name, summary,
  creation) VALUES ('12', 'MDB2', 'Database Abstraction
  Layer', '2006-09-03')
query(1): INSERT INTO `Packages` (id, name, summary,
  creation) VALUES ('13', 'MDB2_Schema', 'XML Based
  Database Schema Manager', CURRENT_TIMESTAMP)
```

Prática 2

```
query(1): INSERT INTO `Packages` (id, name, summary,
  creation) VALUES ('14', 'XML_Parser', 'XML parsing class
  based on PHP\'s bundled expat', '2005-09-24')
query(1): INSERT INTO `Packages` (id, name) VALUES ('77',
  'I am supposed to be deleted')
query(1): UPDATE `Packages` SET id=(`id` - 10)
query(1): DELETE FROM `Packages` WHERE (`id` = 67)
commit(1): Committing transaction/savepoint
query(1): COMMIT
```

Carregando o pacote

Agora que já aprendemos o que é o MDB2_Schema, para quê serve e a estrutura básica dos arquivos com os quais ele trabalha, precisamos saber como carregá-lo e utilizá-lo em nosso aplicativo.

Carregando o pacote

```
require_once 'MDB2/Schema.php';

$options = array(
    'log_line_break' => '<br>',
    'idxname_format' => '%s',
    'debug' => true,
    'quote_identifier' => true,
    'force_defaults' => false,
    'portability' => false
);

$dsn = 'mysql://root:@localhost/DBname';
```

Carregando o pacote

```
$schema =& MDB2_Schema::factory($dsn,
 $options);
```

Carregando o pacote

```
$def = $schema->parseDatabaseDefinition();
if (PEAR::isError($def)) {
  $error = $def->getMessage();
} else {
  $op = $schema->createDatabase($def);
  if (PEAR::isError($op)) {
    $error = $op->getMessage();
  } else {
    var_dump($op);
  }
}
```

Carregando o pacote

```
if (isset($error) && $error) {
  echo $error;
} else {
  var_dump($def);
}

$schema->disconnect();
```

Principais métodos

getDefinitionFromDatabase

Faz engenharia reversa na base de dados atual e retorna uma estrutura vetorial processada.

Principais métodos

parseDatabaseDefinition

Lê o conteúdo de um arquivo MDB2 XML e retorna a definição do banco de dados em uma estrutura vetorial processada.

Principais métodos

dumpDatabase

Toma como entrada uma definição vetorial, anteriormente carregada, e produz a descrição XML correspondente.

Principais métodos

updateDatabase

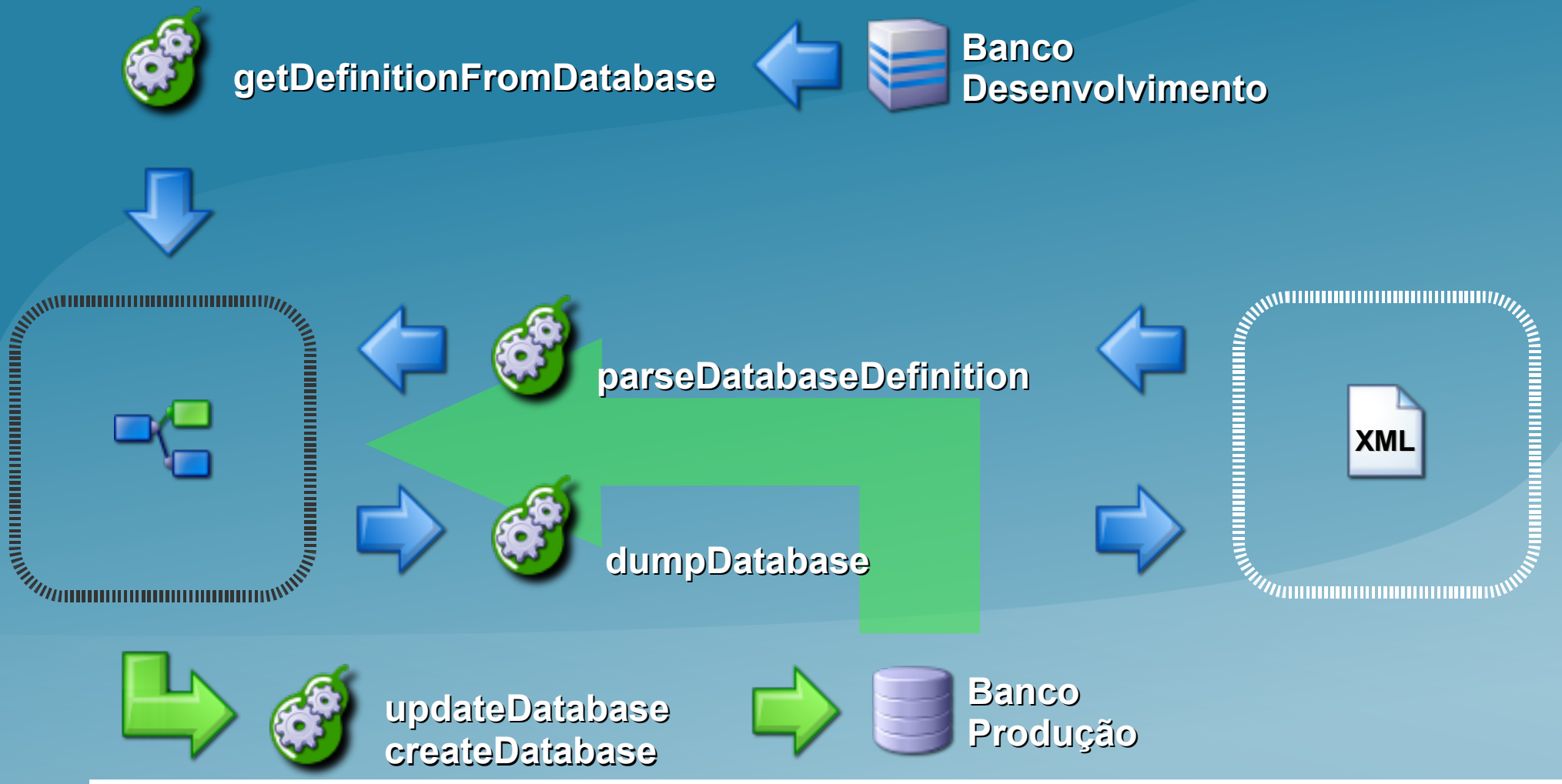
Compara duas versões de uma mesma base de dados e gera a diferença entre elas, atualizando a base atualmente instalada se desejado.

Principais métodos

createDatabase

Toma como entrada uma definição vetorial, anteriormente carregada, e cria a base de dados correspondente.

Principais Métodos: Ilustração



Prática 3

demonstrar a estrutura vetorial resultante do carregamento de uma base de dados

Você pode ajudar!

O MDB2_Schema é um aplicativo de código livre, distribuído sob a licença BSD.
Ele está em constante desenvolvimento e nós o encorajamos a propor correções caso você ache algo que não está funcionando como deveria.
Bugs podem ser submetidos através da web no site do PEAR.

O que ainda não está feito

- Parser não suporta elementos recursivos*
- Sub-consultas
- Cláusulas ORDER
- Chaves estrangeiras* (GSoC 2007)
- Views
- ...

Fim

ifeghali@php.net

dúvidas 

agradeço à minha amada Heloisa por seu apoio e dedicação