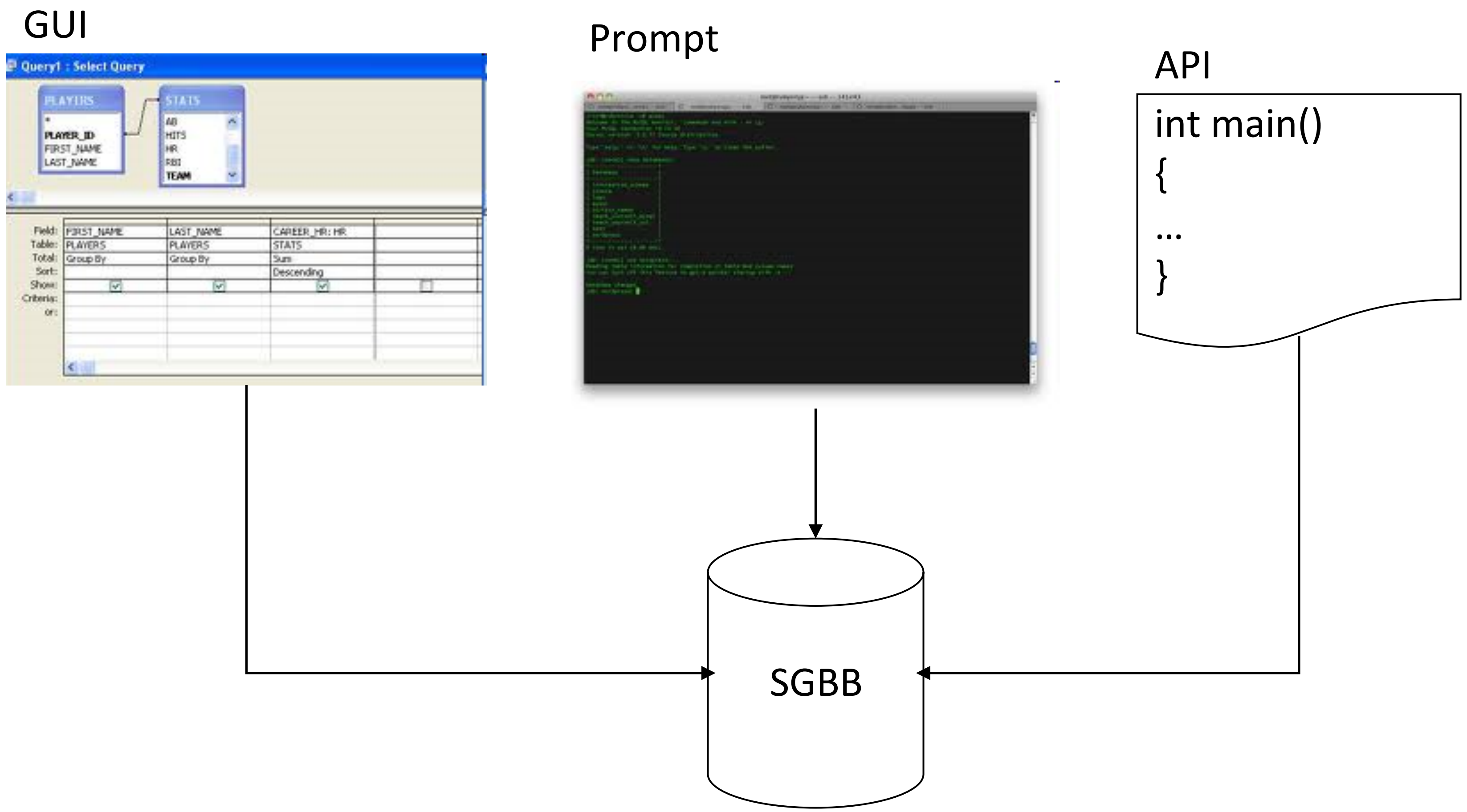
Programando em BD

Lubia Vinhas

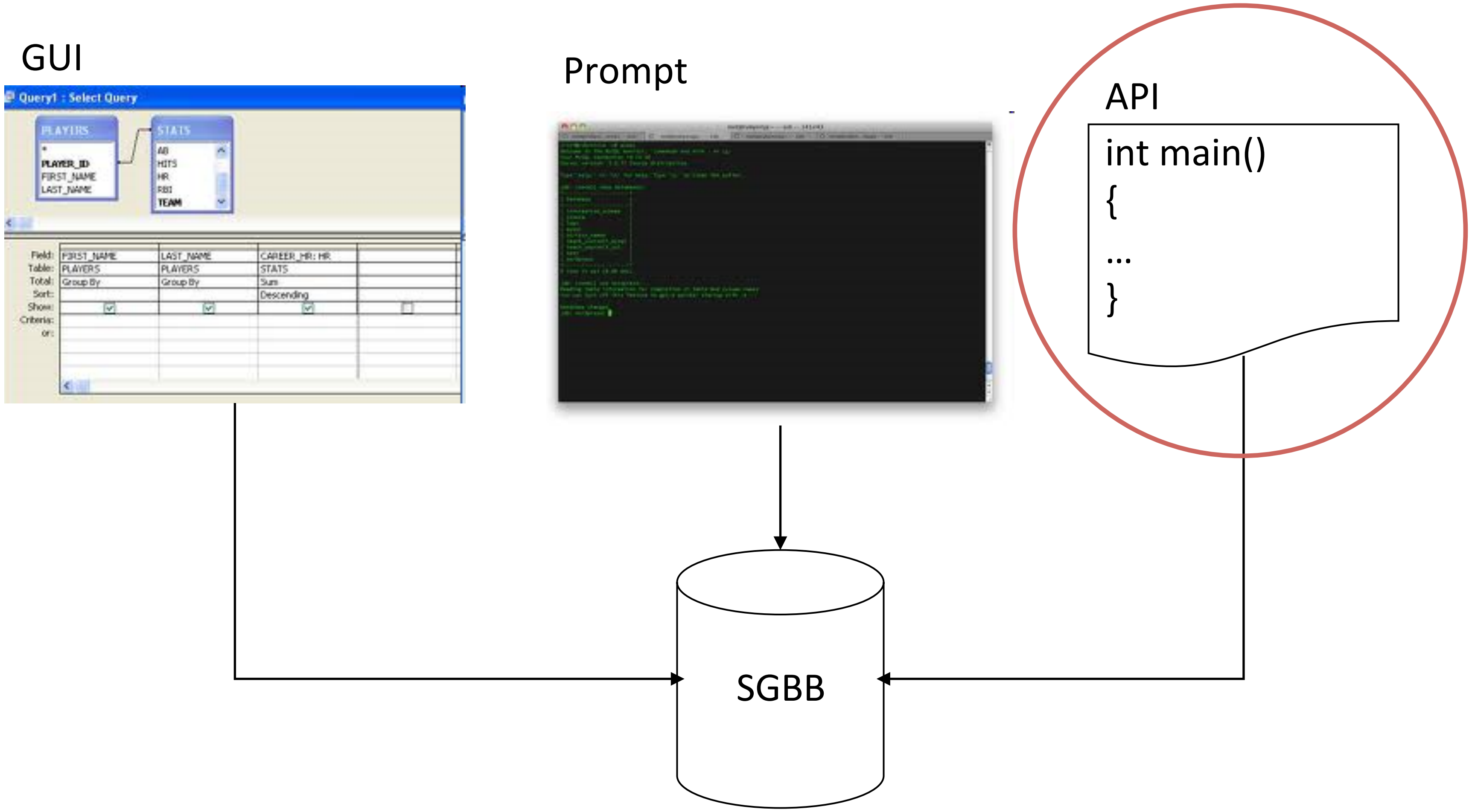
Interfaces SGBD



Field:	FIRST_NAME	LAST_NAME	CAREER_HR: HR	
Table:	PLAYERS	PLAYERS	STATS	
Total:	Group By	Group By	Sum	
Sort:			Descending	
Show:	✓	✓	✓	☐
Criteria:				
or:				

```
int main()
{
…
}
```

Interfaces SGBD



Query1 : Select Query

PLAYERS: *, PLAYER_ID, FIRST_NAME, LAST_NAME

STATS: AB, HITS, HR, RBI, TEAM

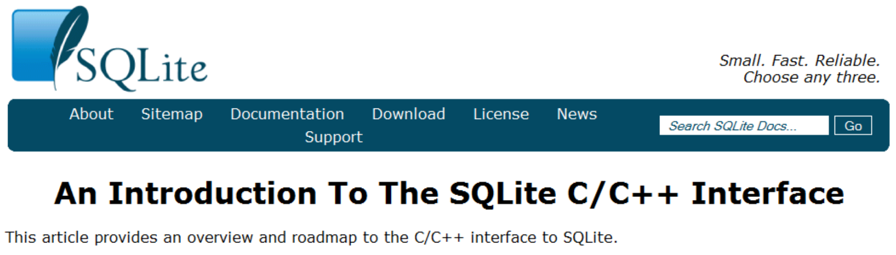
Field:	FIRST_NAME	LAST_NAME	CAREER_HR: HR	
Table:	PLAYERS	PLAYERS	STATS	
Total:	Group By	Group By	Sum	
Sort:			Descending	
Show:	✓	✓	✓	☐
Criteria:				
or:				

```
int main()
{
…
}
```

API para SGBD

- Uma API é uma biblioteca de código em alguma linguagem de programação de forma que você possa criar um programa que converse com o banco
- Pra isso você precisa de um ambiente de edição, compilação e linking para aquela linguagem

Exemplos



SQLite

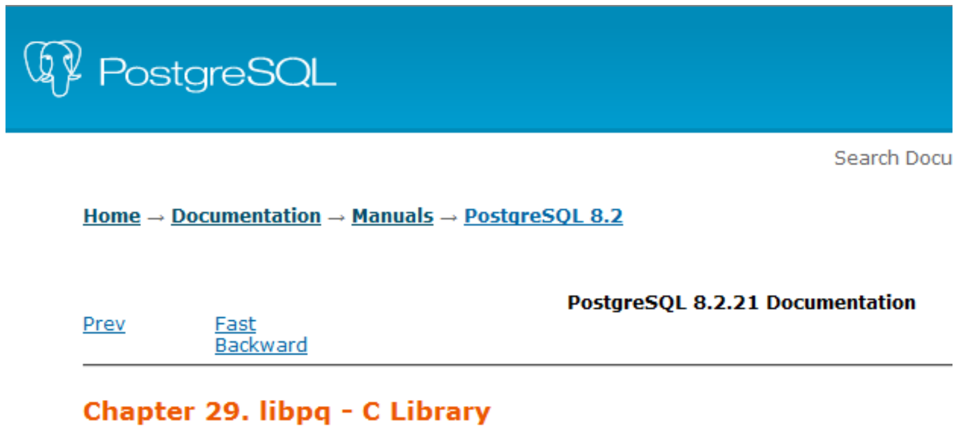
Small. Fast. Reliable. Choose any three.

About Sitemap Documentation Download License News Support

Search SQLite Docs... Go

An Introduction To The SQLite C/C++ Interface

This article provides an overview and roadmap to the C/C++ interface to SQLite.



PostgreSQL

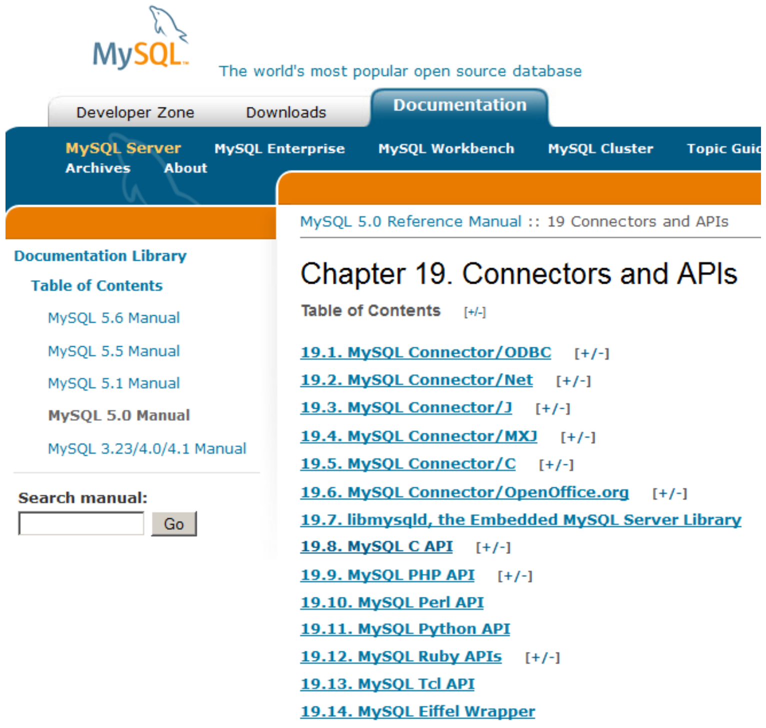
Search Docu

[Home](#) → [Documentation](#) → [Manuals](#) → [PostgreSQL 8.2](#)

PostgreSQL 8.2.21 Documentation

[Prev](#) [Fast Backward](#)

Chapter 29. libpq - C Library



MySQL

The world's most popular open source database

Developer Zone Downloads Documentation

MySQL Server MySQL Enterprise MySQL Workbench MySQL Cluster Topic Guid
Archives About

Documentation Library
Table of Contents
MySQL 5.6 Manual
MySQL 5.5 Manual
MySQL 5.1 Manual
MySQL 5.0 Manual
MySQL 3.23/4.0/4.1 Manual

Search manual: Go

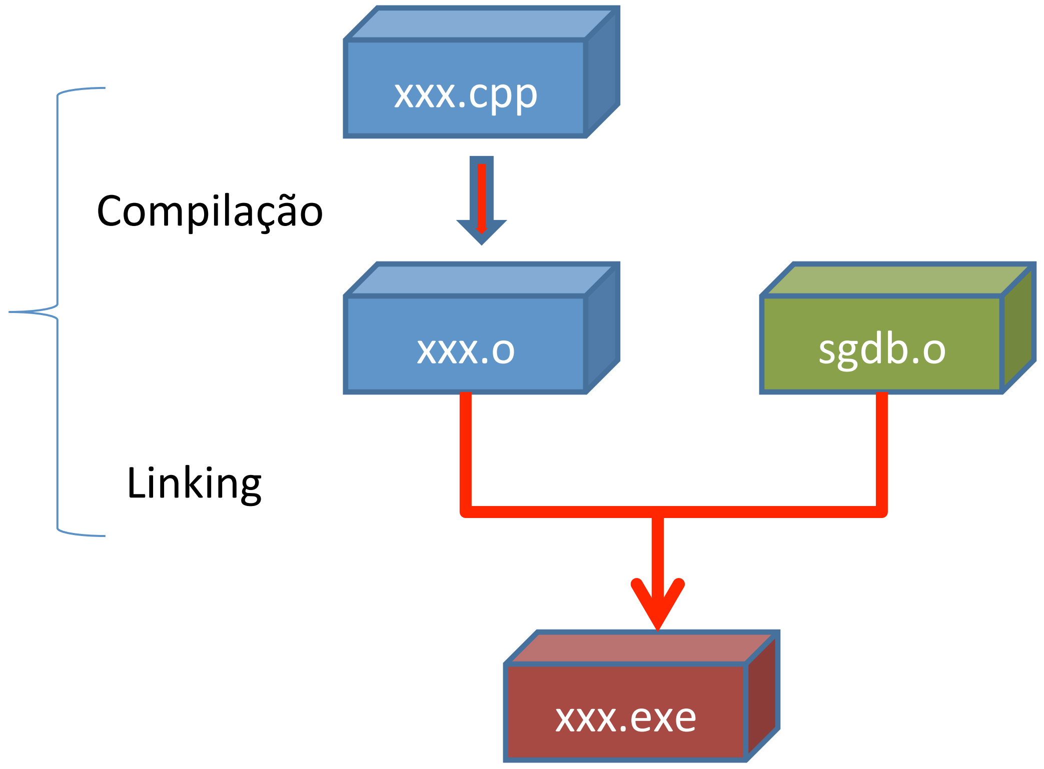
MySQL 5.0 Reference Manual :: 19 Connectors and APIs

Chapter 19. Connectors and APIs

Table of Contents [+/-]

API para SGBD

Ambiente:
editor,
compilador,
linker.
...



APIs para SGBD

- Objetos: quais são os objetos que representam os diferentes componentes do SBGD
- Funções: quais as funções sobre esses objetos

Exemplo: SQLite

- Objetos:

  `sqlite3` : representa uma conexão ao SGBD

  `sqlite3_stmt`: representa um comando a ser submetido ao SGBD

- Funções:

```
sqlite3_open()
sqlite3_prepare()
sqlite3_step()
sqlite3_column()
sqlite3_finalize()
sqlite3_close()
```

Exemplo: SQLite e C++

```
#include <stdio.h>
#include <sqlite3.h>

int main(int argc, char* argv[])
{
   sqlite3 *db;
   char *zErrMsg = 0;
   int rc;

   rc = sqlite3_open("test.db", &db);

   if( rc ){
      fprintf(stderr, "Can't open database: %s\n", sqlite3_errmsg(db));
      exit(0);
   }else{
      fprintf(stderr, "Opened database successfully\n");
   }
   sqlite3_close(db);
}
```

```
$gcc test.c -l sqlite3
$./a.out
Opened database successfully
```

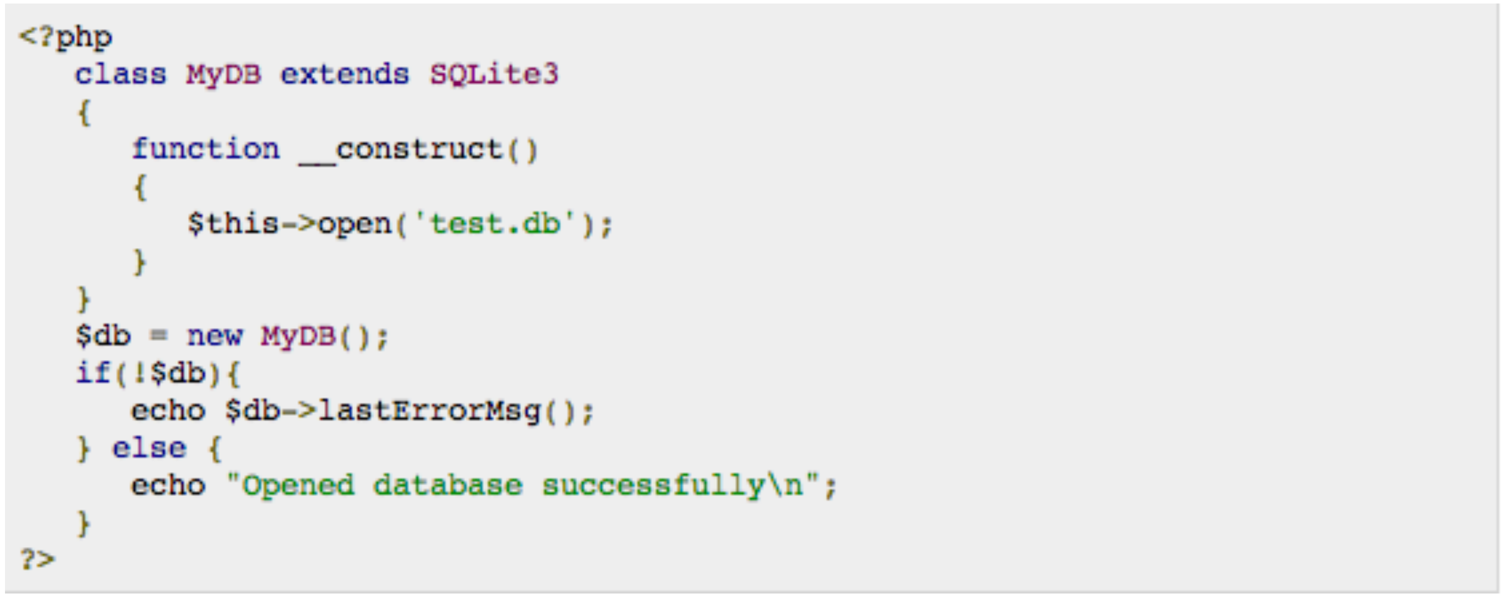
Exemplo: SQLite e JAVA

```java
import java.sql.*;

public class SQLiteJDBC
{
  public static void main( String args[] )
  {
    Connection c = null;
    try {
      Class.forName("org.sqlite.JDBC");
      c = DriverManager.getConnection("jdbc:sqlite:test.db");
    } catch ( Exception e ) {
      System.err.println( e.getClass().getName() + ": " + e.getMessage() );
      System.exit(0);
    }
    System.out.println("Opened database successfully");
  }
}
```

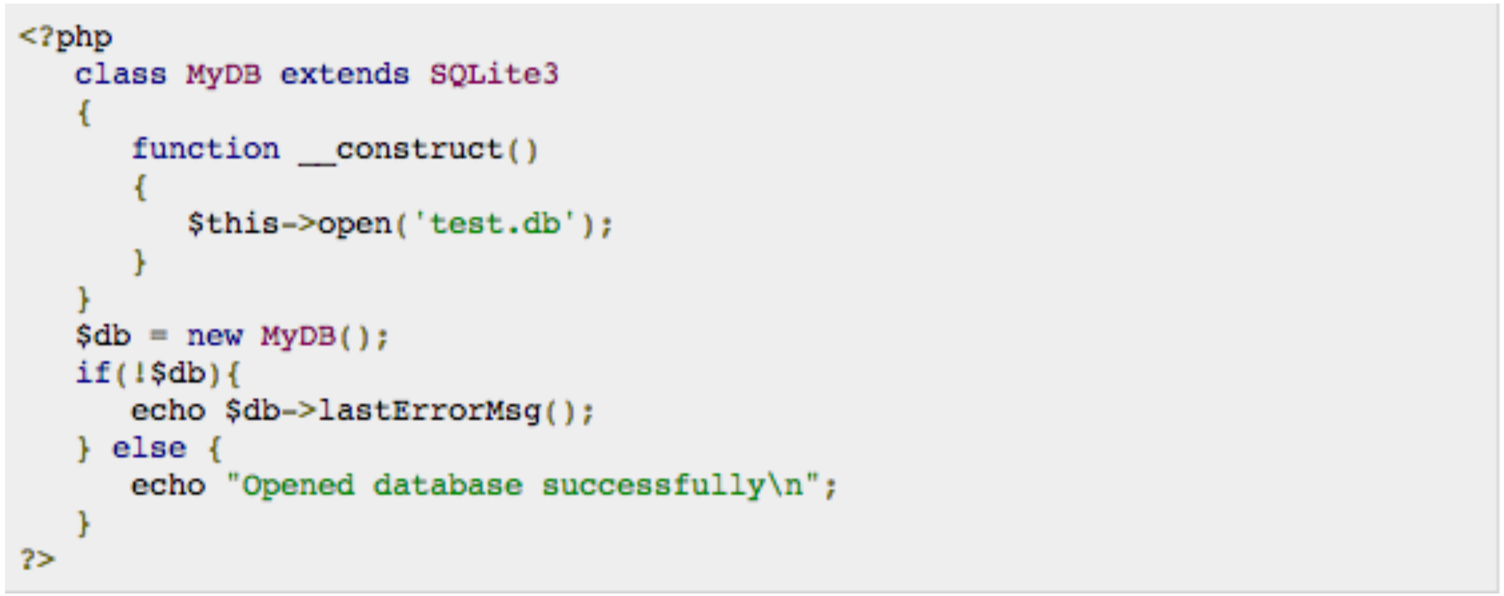
```
$javac SQLiteJDBC.java
$java -classpath ".:sqlite-jdbc-3.7.2.jar" SQLiteJDBC
Open database successfully
```

Exemplo: SQLite e PHP



```
<?php
   class MyDB extends SQLite3
   {
      function __construct()
      {
         $this->open('test.db');
      }
   }
   $db = new MyDB();
   if(!$db){
      echo $db->lastErrorMsg();
   } else {
      echo "Opened database successfully\n";
   }
?>
```

Exemplo: SQLite e PHP



```php
<?php
   class MyDB extends SQLite3
   {
      function __construct()
      {
         $this->open('test.db');
      }
   }
   $db = new MyDB();
   if(!$db){
      echo $db->lastErrorMsg();
   } else {
      echo "Opened database successfully\n";
   }
?>
```

Exemplo: SQLite e Python

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')

print "Opened database successfully";
```

```
$chmod +x sqlite.py
$./sqlite.py
Open database successfully
```

```
  try {
    /*
    * Load the JDBC driver and establish a connection.
    */
    Class.forName("org.postgresql.Driver");
    String url = "jdbc:postgresql://localhost:5432/database";
    conn = DriverManager.getConnection(url, "postgres", "");
    /*
    * Add the geometry types to the connection. Note that you
    * must cast the connection to the pgsql-specific connection
    * implementation before calling the addDataType() method.
    */
    ((org.postgresql.PGConnection)conn).addDataType("geometry",Class.forName("org.postgis.PGgeometry"));
    ((org.postgresql.PGConnection)conn).addDataType("box3d",Class.forName("org.postgis.PGbox3d"));

    /*
    * Create a statement and execute a select query.
    */
    Statement s = conn.createStatement();
    ResultSet r = s.executeQuery("select geom,id from geomtable");
    while( r.next() ) {
      /*
      * Retrieve the geometry as an object then cast it to the geometry type.
      * Print things out.
      */
      PGgeometry geom = (PGgeometry)r.getObject(1);
      int id = r.getInt(2);
      System.out.println("Row " + id + ":");
      System.out.println(geom.toString());
    }
    s.close();
    conn.close();
  }
catch( Exception e ) {
  e.printStackTrace();
```