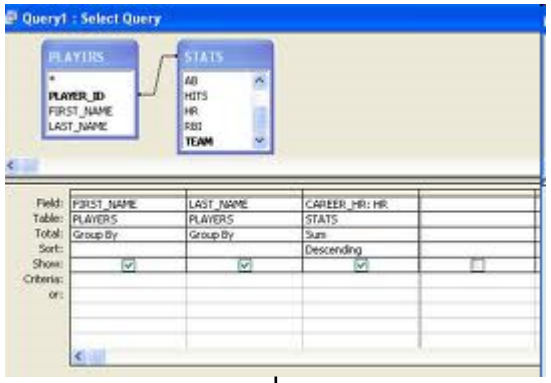
Programando em BD

Lubia Vinhas

Interfaces SGBD

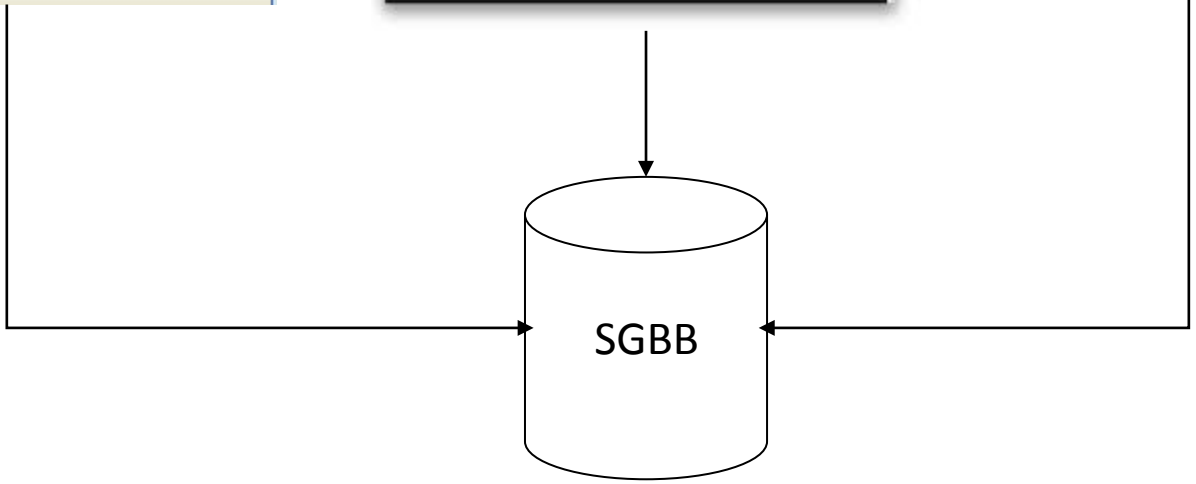
GUI



Prompt

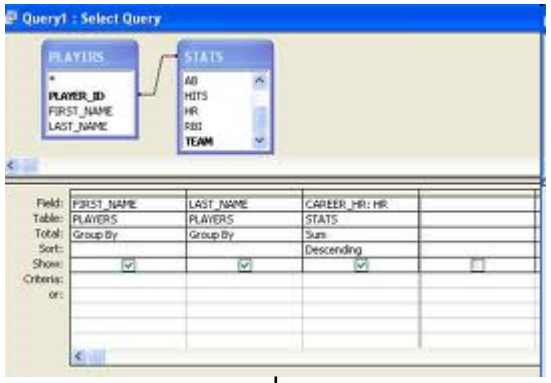
API

```
int main()
{
…
}
```



Interfaces SGBD

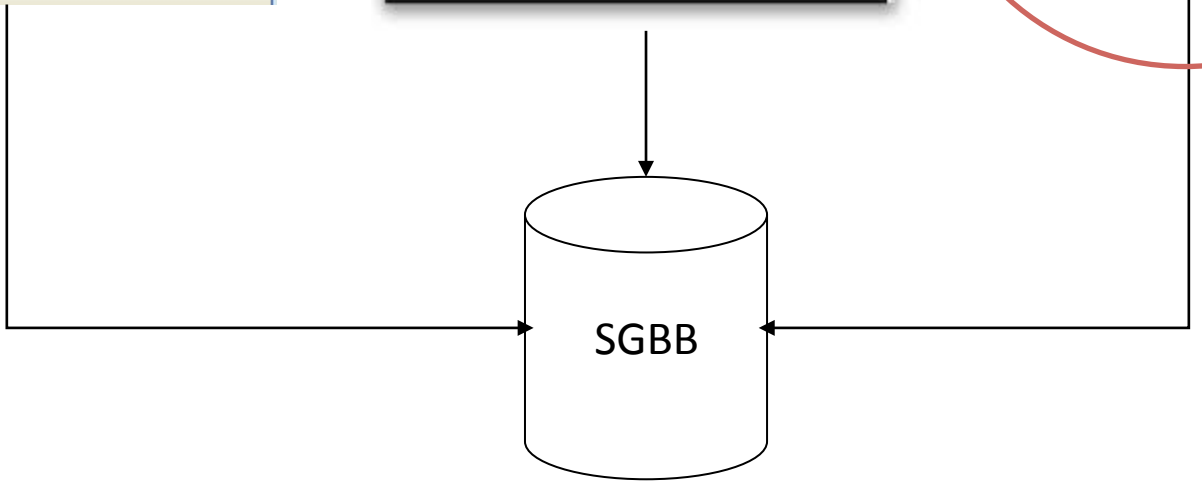
GUI



Prompt

API

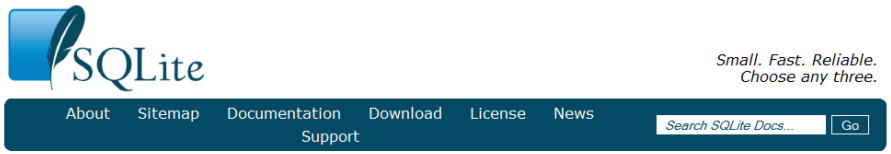
```
int main()
{
…
}
```

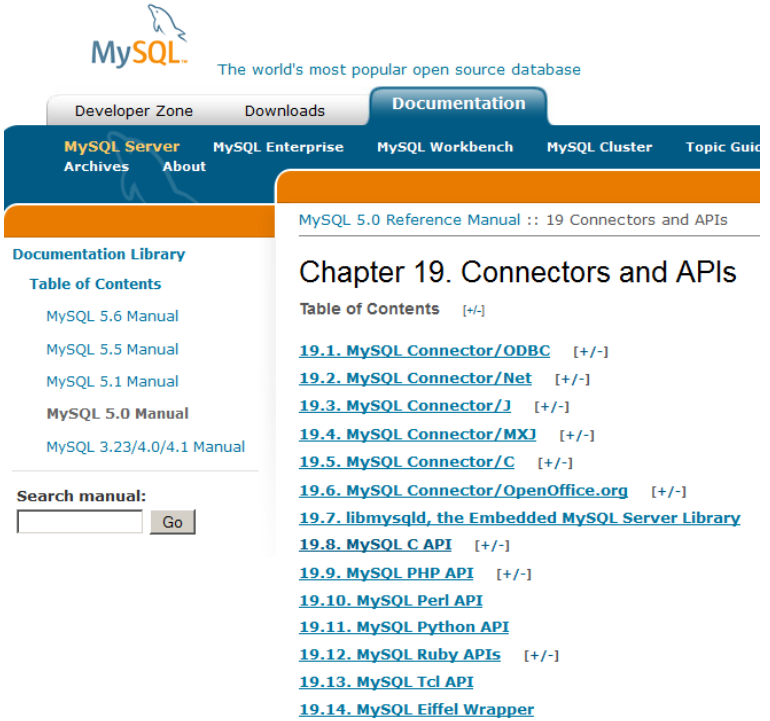


API para SGBD

- 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

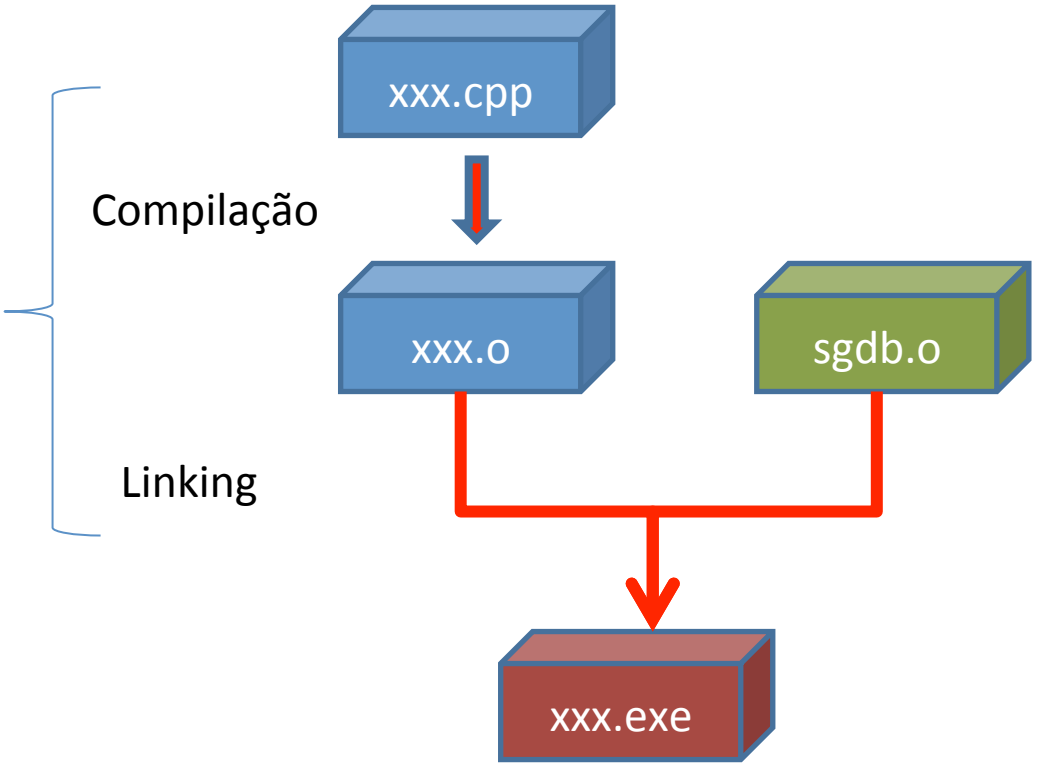

SQLite
Small. Fast. Reliable.
Choose any three.
About Sitemap Documentation Download License News
Support
Search SQLite Docs...
Go
An Introduction To The SQLite C/C++ Interface
This article provides an overview and roadmap to the C/C++ interface to SQLite.


MySQL
The world's most popular open source database
Developer Zone
Downloads
Documentation
MySQL Server
MySQL Enterprise
MySQL Workbench
MySQL Cluster
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 [+/-]
19.1. MySQL Connector/ODBC [+/-]
19.2. MySQL Connector/Net [+/-]
19.3. MySQL Connector/J [+/-]
19.4. MySQL Connector/MXJ [+/-]
19.5. MySQL Connector/C [+/-]
19.6. MySQL Connector/OpenOffice.org [+/-]
19.7. libmysqld, the Embedded MySQL Server Library
19.8. MySQL C API [+/-]
19.9. MySQL PHP API [+/-]
19.10. MySQL Perl API
19.11. MySQL Python API
19.12. MySQL Ruby APIs [+/-]
19.13. MySQL Tcl API
19.14. MySQL Eiffel Wrapper


PostgreSQL
Search Docu
Home → Documentation → Manuals → PostgreSQL 8.2
PostgreSQL 8.2.21 Documentation
Prev
Fast Backward
Chapter 29. libpq - C Library

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

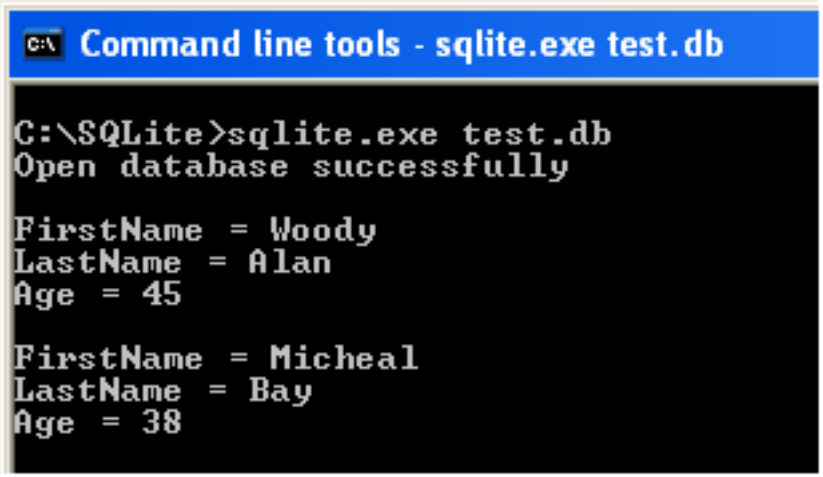
  `sqlite3_stmt`: representa um comando a ser submetido ao SGBD

- Funções:

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

Exemplo: SQLite C++

- `sqlite.cpp`



Exemplo: SQLite PHP

```php
<?php
$db = new SQLite3('mysqlitedb.db');

$db->exec('CREATE TABLE foo (bar STRING)');
$db->exec("INSERT INTO foo (bar) VALUES ('This is a test')");

$result = $db->query('SELECT bar FROM foo');
var_dump($result->fetchArray());
?>
```

```
  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();
```