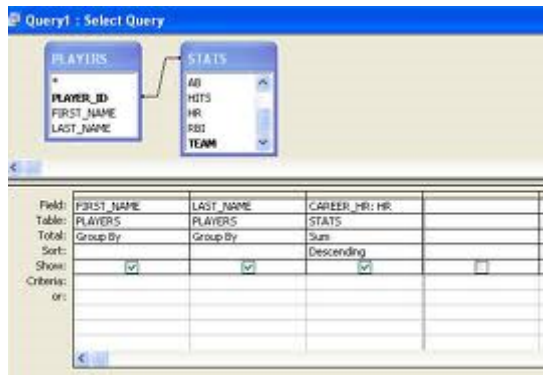


Programando SGBD's

Lubia Vinhas

Interfaces SGBD

GUI

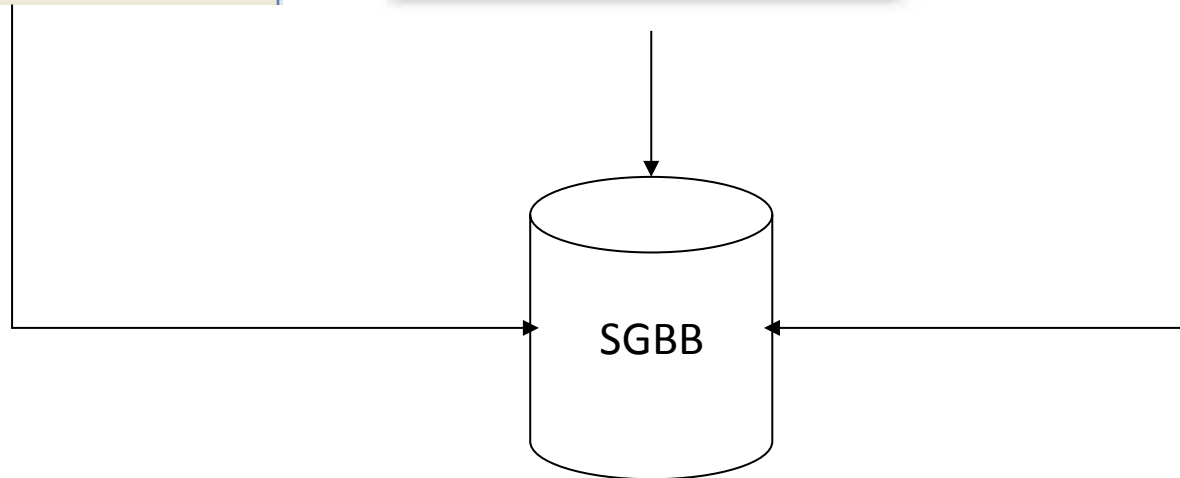


Prompt



API

```
int main()  
{  
  ...  
}
```



Interfaces SGBD

GUI

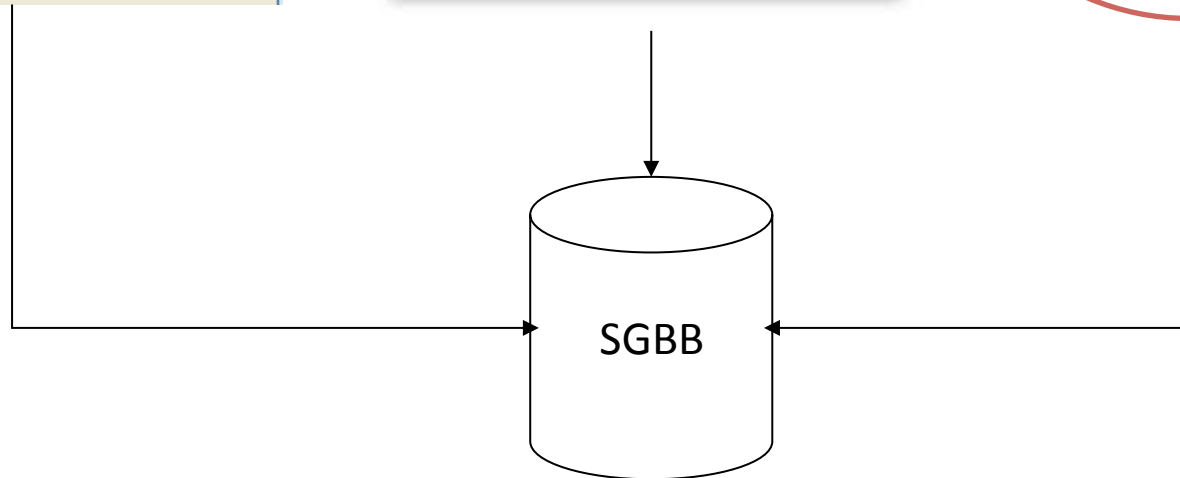


Prompt



API

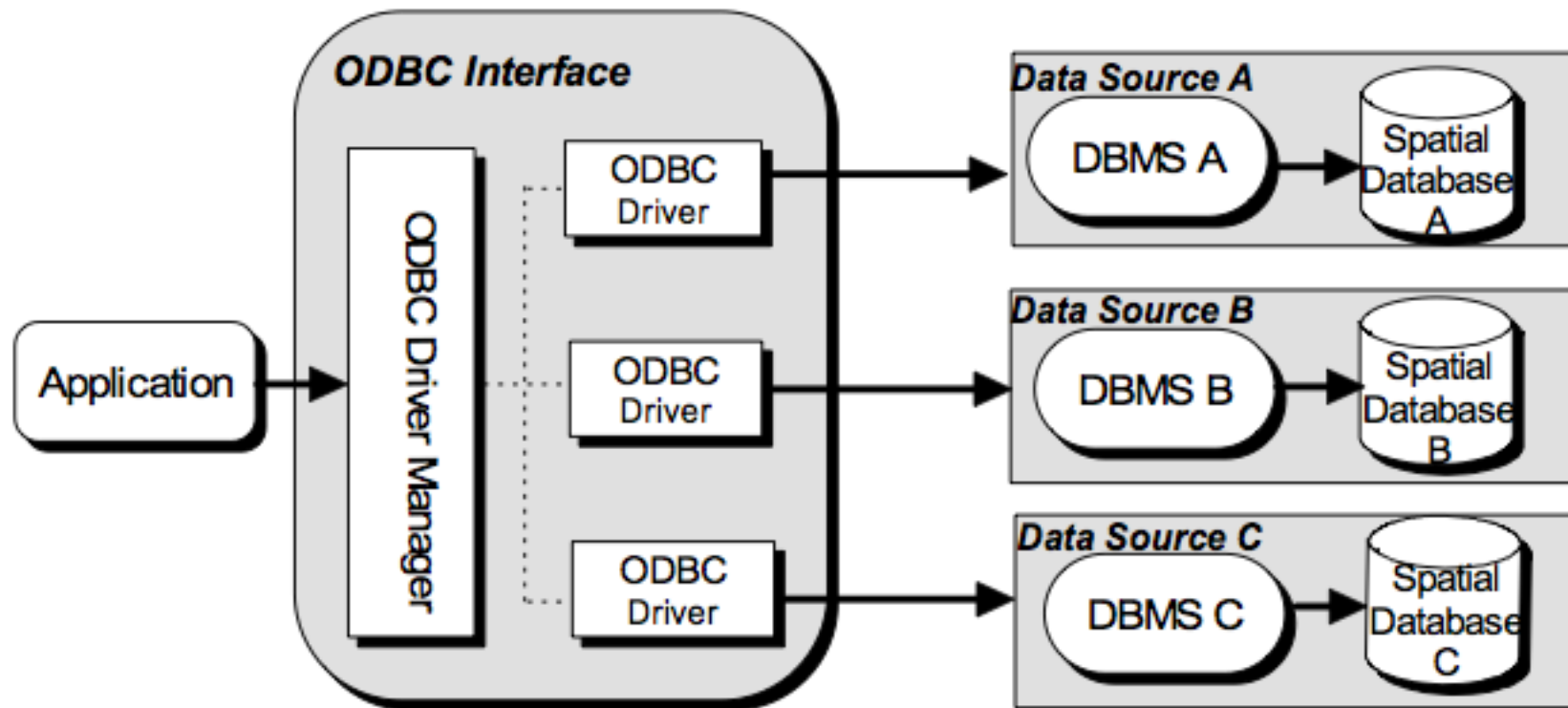
```
int main()
{
  ...
}
```



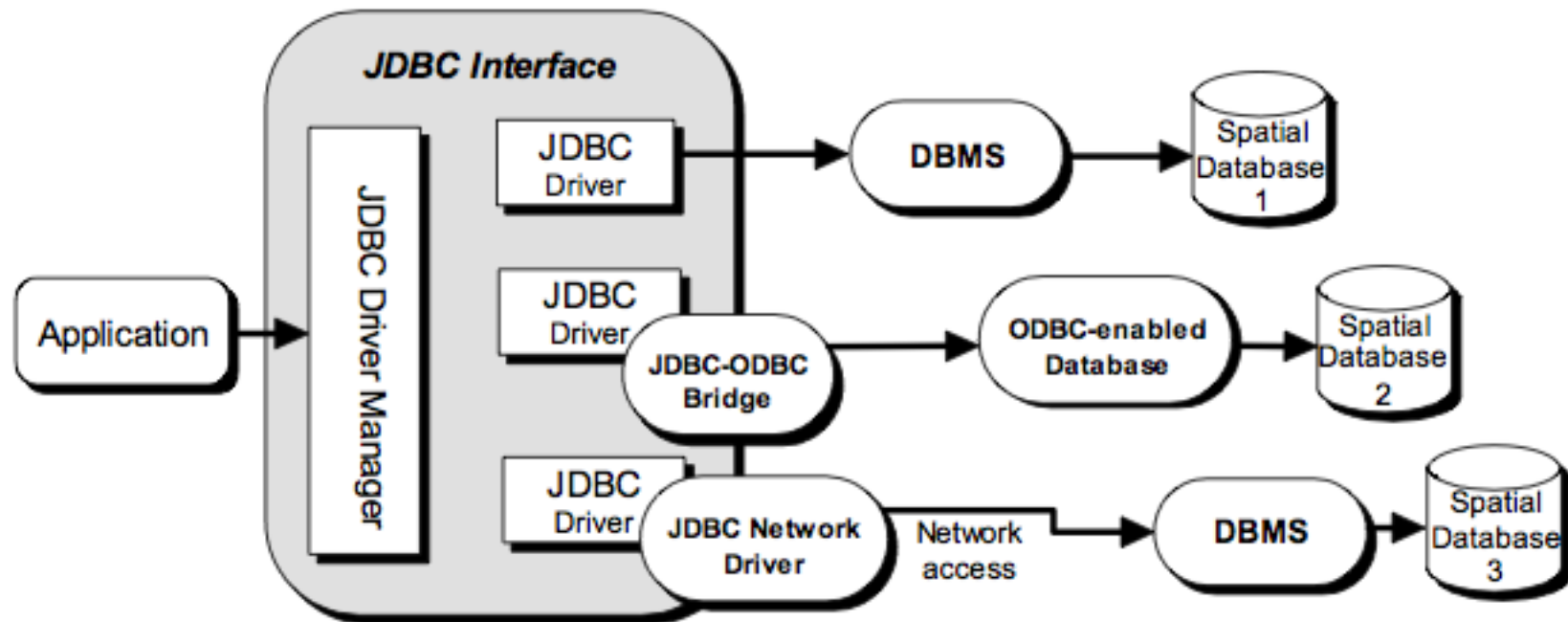
API para SGBD

- 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

Open Database Community (ODBC)



Java Database Connectivity (JDBC)



Exemplos



Small. Fast. Reliable.
Choose any three.

[About](#) [Sitemap](#) [Documentation](#) [Download](#) [License](#) [News](#)

An Introduction To The SQLite C/C++ Interface

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



Search Docu

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

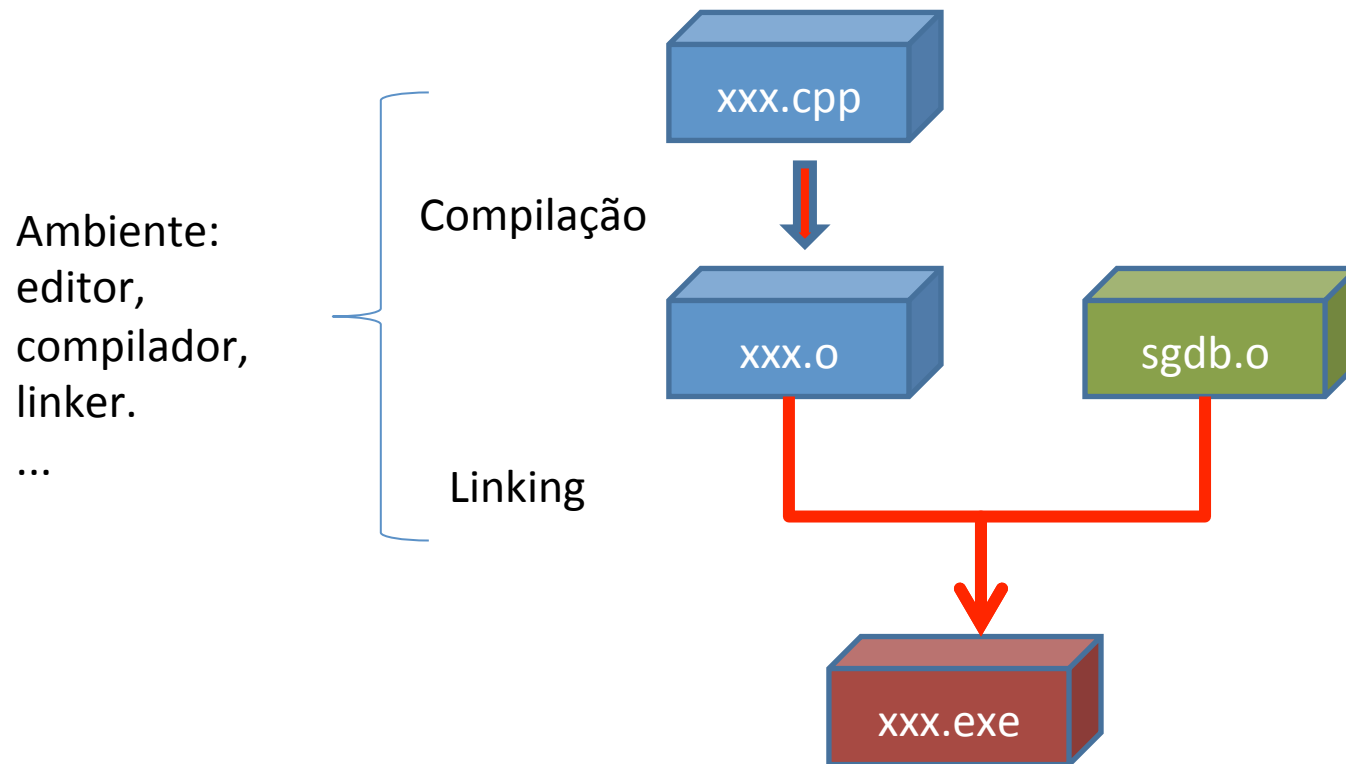
[Prev](#) [Fast Backward](#)

PostgreSQL 8.2.21 Documentation

Chapter 29. libpq - C Library

The screenshot shows the MySQL 5.0 Reference Manual website. The top navigation bar includes "Developer Zone", "Downloads", and "Documentation". Below this, there are links for "MySQL Server", "MySQL Enterprise", "MySQL Workbench", "MySQL Cluster", and "Topic Guide". The main content area is titled "MySQL 5.0 Reference Manual :: 19 Connectors and APIs". On the left, there is a "Documentation Library" section with a "Table of Contents" and links to manuals for MySQL 5.6, 5.5, 5.1, 5.0, and 3.23/4.0/4.1. The main content area lists chapters 19.1 through 19.14, including "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", and "19.14. MySQL Eiffel Wrapper". A search bar is located at the bottom left of the page.

Ambiente de programação



APIs para SGBD

- **Objetos**: quais são os objetos que representam os diferentes componentes do SGBD
- **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

```
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
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

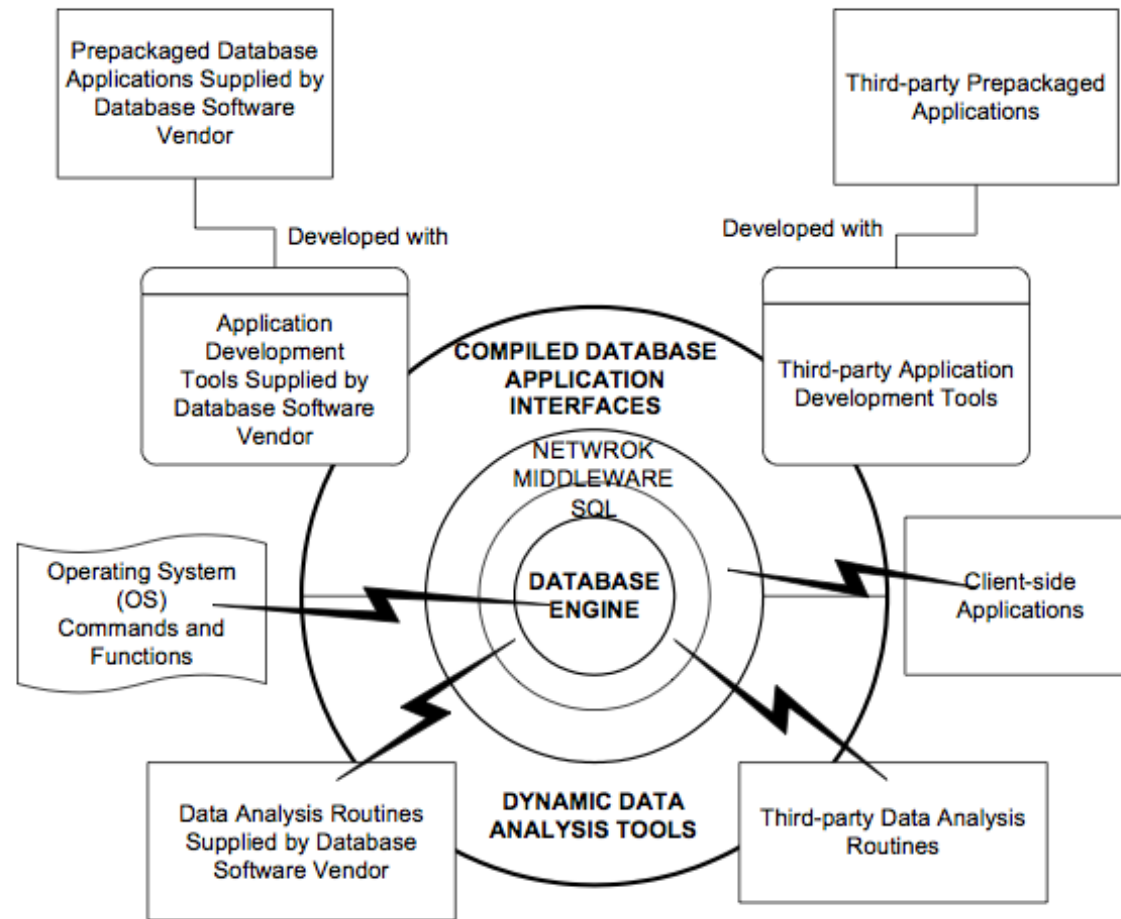
```
#!/usr/bin/python  
  
import sqlite3  
  
conn = sqlite3.connect('test.db')  
  
print "Opened database successfully";
```

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

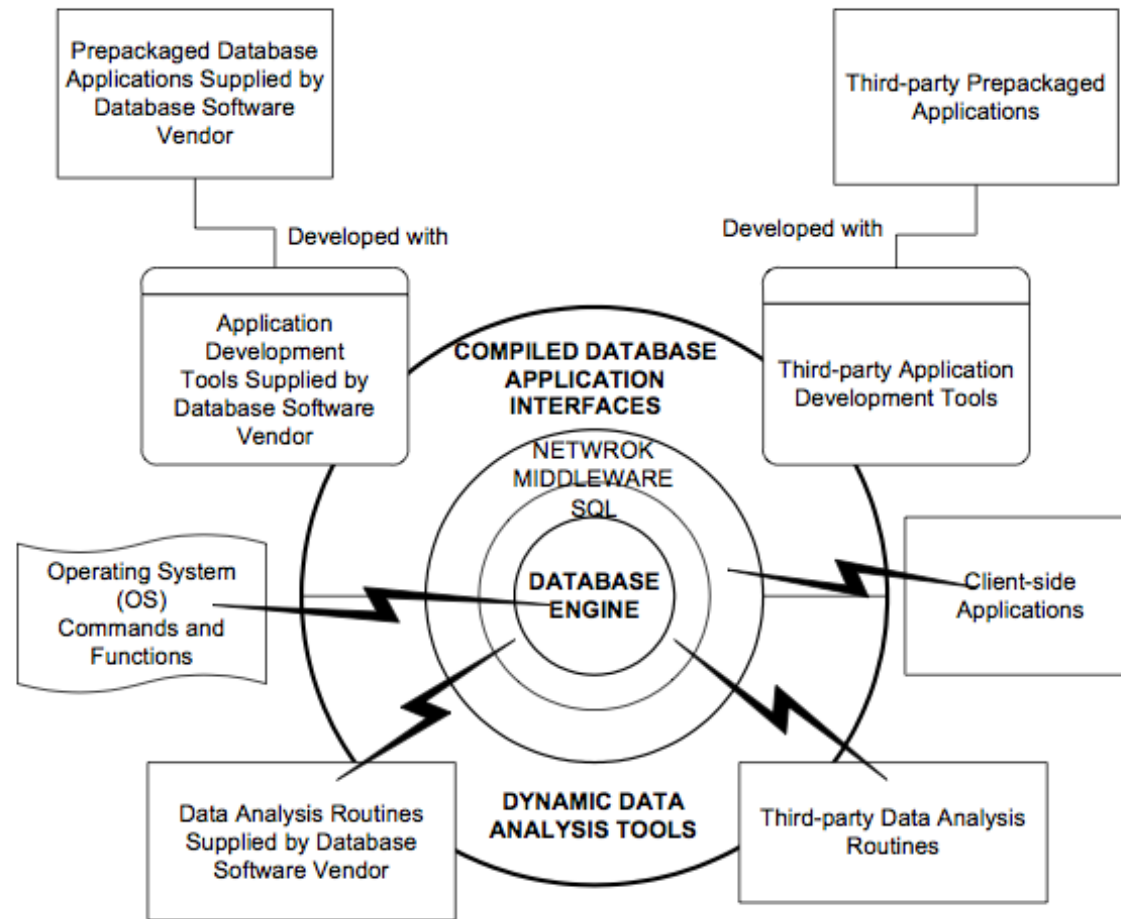
```
try {
    /*
    * Load the JDBC driver and establish a connection.
    */
    Class.forName("org.postgresql.Driver");
    String url = "jdbc:postgresql://localhost:5432/database";
    conn = DriverManager.getConnection(url, "postgres", "");
    /*
    * Add the geometry types to the connection. Note that you
    * must cast the connection to the postgres-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();
}
```


Database Software



Database Software



[Yeung and Hall, 2011]