SHALMON ANANDAS

ROLL NO. 91

RESEARCH PAPER REPORT

TABLE OF CONTENTS

Database Connection Technology

Abstract

This report focuses on database connection using three technologies: Java, Python, and PHP, and provides creative and proficient approaches for accessing large databases. The interplay of web and database is becoming a critical component of database education; integrating data across several systems is a major difficulty with current information technologies. Given the widespread use of database management systems, we'll talk about how to gain efficient and safe access to remote databases. This study delves into database technology, their implementation, and applications.

Introduction

Because of the enormous reliance of the corporate sector on online record maintenance in areas such as finance, transportation, education, culture, healthcare, leisure (theatre, concerts), and utilities such as water, gas, and electricity, database management technology has gained worldwide prominence. Database connection technologies have made it easier and more efficient to manage tasks. This article covers a variety of database connection technologies, including Python, Java, and PHP. It includes a quick overview of each of the languages, as well as how to connect to a database using each of them. It entails a simple step-by-step technique that works flawlessly and is also simple to implement.

Database connectivity in Java

Introduction to Java:

In 1995, Sun Microsystems released Java, a revolutionary programming language. Previously, the word "Java" could only have two meanings: an island or a coffee mix. Its syntax is derived from C, while its functionalities are derived from C++. As a result, one of the goals of Java development was to produce software that could be used in electrical devices. Java is an object-oriented programming language that provides a stable, secure, and portable platform. It is unique in that it is platform agnostic, meaning that its programmes can operate on a variety of systems like Linux, Microsoft Windows, Apple Macintosh, and others. In terms of performance, Java is one of the finest among high-level, fully interpreted scripting languages.

JDBC overview:

JDBC is a database access framework API made up of a set of interfaces and classes that allow Java programmes to communicate with databases. The JDBC driver converts low-level proprietary DBMS data into low-level data that the JBC API can understand. For interaction with a database management system, J2EE components employ a process.

How to connect to database:

1. Load JDBC driver
2. Open connection between J2EE and DBMS
3. Create and execute statement
4. Return data and error messages
5. Return transaction management routines
6. Terminate the connection.

Types of JDBC drivers:

1. Type 1 [JDBC ODBC bridge driver]:
    a. This is integrated in Java.
    b. Suitable to run with a local database.
    c. Needs the client to install database software.
    d. Can’t be used with applets
2. Type 2 [Native API partially Java enabled driver]:

    a. Client can connect directly to database server
    b. Provides adequate speed to access database.
    c. Needs configuration before installing native code on client side
    d. Dependent of DBMS.
    e. Code isn't very portable.
3. Type 3 [Net protocol fully Java enabled driver]:
    a. This is the most flexible configuration
    b. Can access multiple databases from one driver
    c. Independent of DBMS
    d. Needs an intermediate server to function which has to be configured by the vendor
4. Type 4 [Native protocol fully Java enabled driver]:
    a. This is the fastest way to communicate SQL queries to the DBMS
    b. This allows direct call to database without client pre-configuration
    c. A different driver needs to be loaded foreach DBMS
    d. Not appropriate for Applets

Database Connectivity in Python

Introduction to Python:

Guido Van Rossum invented the Python programming language. He named the language "Python" after the comic series "Monty Python's Flying Circus," which he loved. It's an object-oriented, interpreted, high-level programming language having uses in scripting, web programming, scientific computing, and artificial intelligence, among others. It is a widely used language, with companies such as Google, NASA, the CIA, and Disney using it. It has a basic syntax and is straightforward to learn for someone who is new to computer programming. Because the interpreter processes this language at runtime, there is no need to compile the programme before running it.

How to establish connection to Database:

1. MySQL python connector needs to be installed
2. Import the connector
3. Establish connection to existing database
4. Create a cursor to go be a messenger between MySQL and Python
5. You can also do other Database management tasks using this connector like:
    a. Create a database in python
    b. Insert data into created database
    c. Update database
    d. Delete entries

Practical code to connect and execute commands on a database in Python:

1. Establishing connection:

```
>>> import mysql.connector
>>>conn=mysql.connector.connect(user='root',
password='123',
host='localhost',
database='family')
```

2. Make cursor

```
>>>pen=conn.cursor()
>>>pen.execute(“SHOW TABLES”)
>>>print(pen.fetchall())
```

3. Create database

```
>>>pen.execute(“CREATE DATABASE mypython”)
>>>pen.execute(“USE mypython”)
>>>pen.execute(“““CREATE TABLE customer
( idint primary key,
namevarchar(30),
emailvarchar(30),
ageint,
gender char(1))”””)
```

4. Insert Data in table

```
>>> import mysql.connector
>>>conn=mysql.connector.connect(user=’root’,
password=’123’,
host=’localhost’,
database=’family’)
>>>pen=conn.cursor ()
>>>pen.execute(“““INSERT
 INTO
 customer
VALUES(1,’JOE’,joe@gmail.com,22,’F’)”””)
>>>pen.execute(“““INSERT INTO customer VALUES(2,’JIMMY’,
jimmy@gmail.com,10,’F’)”””)
>>>pen.execute(“““INSERT INTO customer VALUES(3,’JACK’,
jack@gmail.com,28,’M’)”””)
>>>conn.commit()
>>>pen.execute(“SELECT * FROM customer”)
>>>print(pen.fetchall())
```

Python Connection Pooling:

You can pool a connection in python, this is basically a cache of the database which can be used later. The module “mysql.connector.pooling” handles pooling. The features of this module are:

1. Can open numerous connections and handles thread safety during connection.
2. Size of the pool is configurable
3. Connection pool can be named at creation time
4. Can handle multiple connection pools
5. Error handling if the pool is exhausted
6. Connection parameters can be restructured

Database Connectivity in PHP

Introduction to PHP:

PHP is a popular programming language in the software development industry. PHP, along with other open source projects MYSQL and Apache, is at the vanguard of Web2.0 and Service Oriented Architectures support technologies. For many people, the ability to connect with databases provided by a scripting language like PHP is the primary motivation for learning it. In this tutorial, I'll teach you how to connect PHP to a MYSQL database. Not only does PHP have a significant open source community, but it is also supported by IT giants like IBM, Oracle, and Microsoft. This document explains how to connect to a MySQL database using PHP.

1. **Creating a Database:**

   To manage a database you need admin privilege in MySQL. PHP uses mysql_query function to create a MySQL database, this function returns FALSE/TRUE.

   Syntax: `bool mysql_query_1( sql, connection);`

2. **Establish connection to existing database:**

   We first write a script to connect to the database to check if the connection succeeds. If the scripts fails, queries will also fail.

```
$user name=”your_username_1”;
$password=”your_password_1”;
$database=”your_database_1”;
```

3. **Connect PHP script to database:**

```
Mysql_connect($localhost,$username,$password)
```

4. **Select database:**

```
@mysql_select_db($database) or die($cant choose database)
```

5. **Running database_connectivity.php file:**

```
<?php
// Session Initialization
session_start();
// Data entered by user is stored in variables below
$user = $_REQUEST["username_1"];
```

```
$pass = $_REQUEST["password_1"];
// Initializing Connection Variable
$connection = mysql_connect("localhost",
"username_1","password_1")
or die("Failed! connecting to the server");
$db = mysql_select_db("DBUSERNAME", $connection) or die(“ Failed! selecting the database");
$forlogin = "select * from user";
$result = mysql_query($forlogin);
$flag = true;
while ($row = mysql_fetch_array1($result)) {
if ($user == $row['username'] && $password ==
$row['password']) {
     $_SESSION['currentuser_1'] = $user;
     header('Location: /folder1/page-to-goto-if-login-successful');
     $flag = false;
}
}
if ($flag == true)
// Unsuccessful login
header('Location:              /folder1/page-to-goto-if-unsuccessful-login'?myerror=mismatch');
?>
```

Conclusion

In this work, we used a variety of technologies to effectively connect to a database. Using Python, PHP, and Java, we can do MySQL database management operations. Various types of JDBC drivers have been discussed in Java. We are now able to manage big databases using Java, Python and PHP. Python has the advantage of being easy to construct short code that works efficiently, however PHP has the advantage of being intended for the creation of web apps.

References

1. Kumari, S., Rani, K. S., & Yadav, M. (2017). Database Connection Techonology. *International Journal of Advanced Research in Computer Science*, *8*(5). https://doi.org/https://doi.org/10.26483/ijarcs.v8i5