

PHP ORQALI MYSQL, MSSQL, POSTGRESQL, SQLITE BILAN ISHLASH

Mamatqodirov Mahammadali Mamatisakovich

Farg'ona Davlat Universiteti axborot texnologiyalari kafedrası katta o'qituvchisi

Murodjonova Lobaroy Ma'rufjon qizi

Farg'ona Davlat Universiteti 3-kurs talabasi

lobaroymurodjonova@gmail.com

Annotatsiya: Ushbu maqola PHP dasturlash tilida turli xil ma'lumotlar bazalari bilan ishlash usullarini o'rgatishga bag'ishlangan. Maqolada MySQL, MSSQL, PostgreSQL va SQLite ma'lumotlar bazalariga ulanish, ma'lumotlarni o'qish, kiritish, yangilash va o'chirish jarayonlari batafsil ko'rib chiqiladi. Shuningdek, PDO (PHP Data Objects) texnologiyasi yordamida turli ma'lumotlar bazalari bilan universal ishlash imkoniyatlari tahlil etiladi. Maqola web dasturlash sohasida faoliyat yurituvchi dasturchilar va talabalar uchun mo'ljallangan.

Kalit so'zlar: PHP, MySQL, MSSQL, PostgreSQL, SQLite, PDO, ma'lumotlar bazasi, SQL so'rovlar, CRUD operatsiyalari, ma'lumotlar bazasiga ulanish, web dasturlash, ma'lumotlarni saqlash.

Аннотация :Эта статья посвящена обучению методам работы с различными базами данных на языке программирования PHP. В статье подробно рассматриваются процессы подключения к базам данных MySQL, MSSQL, PostgreSQL и SQLite, чтения, введения, обновления и удаления данных. Также анализируются универсальные возможности работы с различными базами данных с помощью технологии PDO (PHP Data Objects). Статья предназначена для программистов и студентов, работающих в области веб-программирования.

Ключевые слова: PHP, MySQL, MSSQL, PostgreSQL, SQLite, PDO, база данных, SQL-запросы, операции CRUD, подключение к базе данных, веб-программирование, хранение данных.

Annotation: This article is dedicated to teaching methods for working with various databases in the PHP programming language. It thoroughly discusses the processes of connecting to MySQL, MSSQL, PostgreSQL, and SQLite databases, as well as reading, inserting, updating, and deleting data. It also analyzes the universal capabilities of working with different databases using PDO (PHP Data Objects) technology. The article is intended for programmers and students active in the field of web programming.

Keywords: PHP, MySQL, MSSQL, PostgreSQL, SQLite, PDO, database, SQL queries, CRUD operations, database connection, web programming, data storage.

KIRISH

Zamonaviy web ilovalarining asosiy qismini ma'lumotlar bazasi bilan ishlash tashkil etadi. PHP dasturlash tili turli xil ma'lumotlar bazalari bilan ishlash uchun keng imkoniyatlarni taqdim etadi. Eng mashhur ma'lumotlar bazalari orasida MySQL, PostgreSQL, SQLite va Microsoft SQL Server (MSSQL) ni ajratish mumkin.

Har bir ma'lumotlar bazasining o'ziga xos xususiyatlari, afzalliklari va qo'llanish sohalari mavjud. MySQL veb-ilovalar uchun eng keng tarqalgan tanlov hisoblanadi, PostgreSQL esa murakkab korporativ tizimlarda qo'llaniladi. SQLite esa kichik loyihalar va mobil ilovalar uchun ideal hisoblanadi.

PHP da ma'lumotlar bazasi bilan ishlashning ikki asosiy usuli mavjud: maxsus kengaytmalar (mysqli, pgsql) va universal PDO interfeysi. PDO yondashuvi ma'lumotlar bazasidan mustaqil kod yozish imkonini beradi va xavfsizlik jihatidan ham afzalroqdir.

ASOSIY QISM

MySQL bilan ishlash

MySQL PHP bilan eng ko'p ishlatiladigan ma'lumotlar bazasidir. MySQLi kengaytmasi yordamida MySQL bilan ishlash mumkin.

MySQL ga ulanish va ma'lumotni o'qish

php<?php

```
$servername = "localhost";

$username = "root";

$password = "";

$dbname = "test_db";

// MySQL ga ulanish

$conn = new mysqli($servername, $username, $password, $dbname);

// Ulanishni tekshirish

if ($conn->connect_error) {

    die("Ulanish xatosi: " . $conn->connect_error);

}

// Ma'lumotlarni o'qish

$sql = "SELECT id, firstname, lastname FROM users";

$result = $conn->query($sql);

if ($result->num_rows > 0) {

    while($row = $result->fetch_assoc()) {

        echo "ID: " . $row["id"]. " - Ism: " . $row["firstname"]. " " . $row["lastname"].

"<br>";

    }

} else {

    echo "0 natija";

}

$conn->close();
```

?>

MySQL ga ma'lumot kiritish

Ma'lumot kiritish uchun INSERT so'rovi ishlatiladi:

```
php<?php
```

```
$sql = "INSERT INTO users (firstname, lastname, email)
```

```
VALUES ('John', 'Doe', 'john@example.com')";
```

```
if ($conn->query($sql) === TRUE) {
```

```
    echo "Yangi yozuv muvaffaqiyatli qo'shildi";
```

```
} else {
```

```
    echo "Xato: " . $sql . "<br>" . $conn->error;
```

```
}
```

```
?>
```

PostgreSQL bilan ishlash

PostgreSQL kuchli va kengaytirilgan ma'lumotlar bazasi bo'lib, korporativ dasturlarda keng qo'llaniladi.

PostgreSQL ga ulanish va ma'lumotlarni boshqarish

```
php<?php
```

```
$host = "localhost";
```

```
$port = "5432";
```

```
$dbname = "test_db";
```

```
$user = "postgres";
```

```
$password = "password";
```

```
// PostgreSQL ga ulanish

$dsn = "pgsql:host=$host;port=$port;dbname=$dbname;";

$pdo = new PDO($dsn, $user, $password);

// Ma'lumotlarni o'qish

$stmt = $pdo->query('SELECT * FROM users');

while ($row = $stmt->fetch()) {

    print_r($row);

}

?>
```

Ma'lumotlarni yangilash

UPDATE so'rovi yordamida mavjud ma'lumotlarni yangilash mumkin:

```
php<?php

$sql = "UPDATE users SET email = :email WHERE id = :id";

$stmt = $pdo->prepare($sql);

$stmt->execute([

    ':email' => 'newemail@example.com',

    ':id' => 1

]);

echo "Ma'lumot yangilandi";

?>
```

SQLite bilan ishlash

SQLite engil va fayl asosidagi ma'lumotlar bazasi bo'lib, kichik loyihalar uchun ideal hisoblanadi.

SQLite bilan asosiy operatsiyalar

```
php<?php
```

```
// SQLite ga ulanish
```

```
$pdo = new PDO('sqlite:example.db');
```

```
// Jadval yaratish
```

```
$sql = "CREATE TABLE IF NOT EXISTS users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL  
)";
```

```
$pdo->exec($sql);
```

```
// Ma'lumot qo'shish
```

```
$stmt = $pdo->prepare("INSERT INTO users (name, email) VALUES (?, ?)");
```

```
$stmt->execute(['Ali Valiyev', 'ali@example.com']);
```

```
?>
```

Ma'lumotlarni o'chirish

DELETE so'rovi yordamida keraksiz ma'lumotlarni o'chirish mumkin:

```
php<?php
```

```
$sql = "DELETE FROM users WHERE id = :id";
```

```
$stmt = $pdo->prepare($sql);
```

```
$stmt->execute([':id' => 1]);
```

```
echo "Ma'lumot o'chirildi";
```

```
?>
```

PDO orqali umumiy ma'lumotlar bazasi bilan ishlash

PDO (PHP Data Objects) turli ma'lumotlar bazalari bilan universal ishlash imkonini beradi. Bu yondashuv kod qayta ishlatilishi va xavfsizlik jihatidan afzalroqdir.

PDO ning asosiy afzalliklari

Xavfsizlik: Prepared statements orqali SQL injection hujumlaridan himoya

Universallik: Turli ma'lumotlar bazalari bilan bir xil interfeys

Ob'ektga yo'naltirilganlik: Zamonaviy PHP dasturlash uslubi

Xato boshqaruvi: Kuchli exception handling mexanizmi

PDO bilan CRUD operatsiyalari

```
php<?php
```

```
class DatabaseManager {
```

```
    private $pdo;
```

```
    public function __construct($dsn, $username, $password) {
```

```
        try {
```

```
            $this->pdo = new PDO($dsn, $username, $password);
```

```
            $this->pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);
```

```
        } catch(PDOException $e) {
```

```
            throw new Exception("Ulanish xatosi: " . $e->getMessage());
```

```
    }  
}  
  
// Ma'lumot qo'shish (Create)  
  
public function create($table, $data) {  
    $columns = implode(',', array_keys($data));  
    $placeholders = ':' . implode(':', array_keys($data));  
    $sql = "INSERT INTO $table ($columns) VALUES ($placeholders)";  
    $stmt = $this->pdo->prepare($sql);  
    return $stmt->execute($data);  
}  
  
// Ma'lumotlarni o'qish (Read)  
  
public function read($table, $conditions = []) {  
    $sql = "SELECT * FROM $table";  
    if (!empty($conditions)) {  
        $whereClause = implode(' AND ', array_map(function($key) {  
            return "$key = :$key";  
        }, array_keys($conditions)));  
        $sql .= " WHERE $whereClause";  
    }  
    $stmt = $this->pdo->prepare($sql);  
    $stmt->execute($conditions);  
    return $stmt->fetchAll(PDO::FETCH_ASSOC);  
}
```

```
}

// Ma'lumotlarni yangilash (Update)

public function update($table, $data, $conditions) {

    $setClause = implode(' ', array_map(function($key) {

        return "$key = :$key";

    }, array_keys($data)));

    $whereClause = implode(' AND ', array_map(function($key) {

        return "$key = :where_$key";

    }, array_keys($conditions)));

    $sql = "UPDATE $table SET $setClause WHERE $whereClause";

// Where parametrlarini qayta nomlash

    $whereParams = [];

    foreach ($conditions as $key => $value) {

        $whereParams["where_$key"] = $value;

    }

    $params = array_merge($data, $whereParams);

    $stmt = $this->pdo->prepare($sql);

    return $stmt->execute($params);

}

// Ma'lumotlarni o'chirish (Delete)

public function delete($table, $conditions) {

    $whereClause = implode(' AND ', array_map(function($key) {
```

```
        return "$key = :$key";
    }, array_keys($conditions)));

    $sql = "DELETE FROM $table WHERE $whereClause";

    $stmt = $this->pdo->prepare($sql);

    return $stmt->execute($conditions);

    }

}

// Foydalanish misoli

$db = new DatabaseManager('mysql:host=localhost;dbname=test', 'root', 'password');

// Yangi foydalanuvchi qo'shish

$db->create('users', [

    'name' => Lobaroy Murodjonova,

    'email' => 'lobaroy@example.com'

]);

// Foydalanuvchilarni o'qish

$users = $db->read('users', ['name' => 'Lobaroy Murodjonova']);

// Malumotni yangilash

$db->update('users',

    ['email' => 'lobaroy.new@example.com'],

    ['name' => 'Lobaroy Murodjonova']

);

// Foydalanuvchini o'chirish
```

```
$db->delete('users', ['name' => 'Lobaroy Murodjonova']);
```

```
?>
```

Xavfsizlik choralar

Ma'lumotlar bazasi bilan ishlashda quyidagi xavfsizlik choralarini ko'rish muhim:

Prepared Statements: SQL injection hujumlaridan himoya uchun

Ma'lumotlarni validatsiya: Kiritilayotgan ma'lumotlarni tekshirish

Parollarni hash qilish: Foydalanuvchi parollarini xavfsiz saqlash

Ulanish ma'lumotlarini himoya qilish: Database credentials ni maxfiy saqlash

```
php<?php
```

```
// Xavfsiz parol hash qilish
```

```
$password = 'user_password';
```

```
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);
```

```
// Parolni tekshirish
```

```
if (password_verify($password, $hashedPassword)) {
```

```
    echo "Parol to'g'ri";
```

```
}
```

```
// Input validatsiya
```

```
function validateEmail($email) {
```

```
    return filter_var($email, FILTER_VALIDATE_EMAIL);
```

```
}
```

```
// XSS hujumlaridan himoya
```

```
function sanitizeOutput($data) {
```

```
return htmlspecialchars($data, ENT_QUOTES, 'UTF-8');  
  
} ?>
```

XULOSA: PHP dasturlash tilida turli ma'lumotlar bazalari bilan ishlash zamonaviy web ilovalar yaratishning ajralmas qismidir. MySQL, PostgreSQL, SQLite va boshqa ma'lumotlar bazalari har birining o'ziga xos afzalliklari mavjud. PDO texnologiyasi universal va xavfsiz yechim sifatida turli ma'lumotlar bazalari bilan ishlash imkonini beradi.

Muvaffaqiyatli ma'lumotlar bazasi bilan ishlash uchun CRUD operatsiyalarini to'g'ri amalga oshirish, xavfsizlik choralarini ko'rish va kodni qayta ishlatish tamoyillariga amal qilish zarur. Prepared statements, ma'lumotlarni validatsiya qilish va to'g'ri error handling mexanizmlarini qo'llash orqali ishonchli va xavfsiz web ilovalar yaratish mumkin.

Foydalanilgan Adabiyotlar:

PHP rasmiy hujjatlari. MySQL bilan ishlash.

URL: <https://www.php.net/manual/en/book.mysqli.php>

PHP Data Objects (PDO) rasmiy qo'llanma.

URL: <https://www.php.net/manual/en/book.pdo.php>

PostgreSQL PHP kengaytmasi hujjatlari.

URL: <https://www.php.net/manual/en/book.pgsql.php>

SQLite va PHP integratsiyasi.

URL: <https://www.php.net/manual/en/book.sqlite3.php>

Welling, L., Thomson, L. (2016). PHP and MySQL Web Development. 5th Edition. Addison-Wesley Professional.

Nixon, R. (2018). Learning PHP, MySQL & JavaScript. 5th Edition. O'Reilly Media.

Sklar, D., Trachtenberg, A. (2014). PHP Cookbook. 3rd Edition. O'Reilly Media.

OWASP SQL Injection Prevention Cheat Sheet.

URL:

[https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.htm](https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html)
l

MySQL rasmiy hujjatlari. Best Practices.

URL: <https://dev.mysql.com/doc/refman/8.0/en/>

PostgreSQL rasmiy hujjatlari. PHP bilan ishlash.

URL: <https://www.postgresql.org/docs/current/>