

Mit SQLite3-Datenbanken arbeiten

Datenbank öffnen

```
<?php
// Pfad zur SQLite3-Datenbankdatei
$db_file = 'datenbank.sqlite';

// Verbindung zur SQLite3-Datenbank herstellen
try {
    $db = new SQLite3($db_file);
    echo "Verbindung erfolgreich hergestellt!<br>";
} catch (Exception $e) {
    die("Verbindung fehlgeschlagen: " . $e->getMessage());
}
?>
```

SQL-Befehl ausführen

```
<?php
// Fortsetzung des vorherigen Codes

// SQL-Befehl zum Erstellen einer Tabelle
$sql = "CREATE TABLE IF NOT EXISTS Benutzer (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    vorname TEXT NOT NULL,
    nachname TEXT NOT NULL,
    email TEXT,
    reg_date DATETIME DEFAULT CURRENT_TIMESTAMP
)";

if ($db->exec($sql)) {
    echo "Tabelle 'Benutzer' erfolgreich erstellt oder bereits
vorhanden.<br>";
} else {
    echo "Fehler beim Erstellen der Tabelle: " . $db->lastErrorMsg() .
"<br>";
}
?>
```

Daten einfügen

```
<?php
// Fortsetzung des vorherigen Codes

// Daten zum Einfügen
```

```
$vorname = "Max";
$nachname = "Mustermann";
$email = "max.mustermann@example.com";

// Prepared Statement zum Einfügen der Daten
$stmt = $db->prepare("INSERT INTO Benutzer (vorname, nachname, email) VALUES
(:vorname, :nachname, :email)");
$stmt->bindValue(':vorname', $vorname, SQLITE3_TEXT);
$stmt->bindValue(':nachname', $nachname, SQLITE3_TEXT);
$stmt->bindValue(':email', $email, SQLITE3_TEXT);

$result = $stmt->execute();

if ($result) {
    echo "Neuer Datensatz erfolgreich eingefügt.<br>";
} else {
    echo "Fehler beim Einfügen des Datensatzes: " . $db->lastErrorMsg() .
"<br>";
}
?>
```

Daten ändern

```
<?php
// Fortsetzung des vorherigen Codes

// Daten zum Aktualisieren
$id = 1; // ID des Datensatzes, den Sie aktualisieren möchten
$neuer_email = "max.neu@example.com";

// Prepared Statement zum Aktualisieren der Daten
$stmt = $db->prepare("UPDATE Benutzer SET email = :email WHERE id = :id");
$stmt->bindValue(':email', $neuer_email, SQLITE3_TEXT);
$stmt->bindValue(':id', $id, SQLITE3_INTEGER);

$result = $stmt->execute();

if ($result) {
    echo "Datensatz erfolgreich aktualisiert.<br>";
} else {
    echo "Fehler beim Aktualisieren des Datensatzes: " . $db->lastErrorMsg()
. "<br>";
}
?>
```

Daten löschen

```
<?php
```

```
// Fortsetzung des vorherigen Codes

// ID des Datensatzes, den Sie löschen möchten
$id = 1;

// Prepared Statement zum Löschen des Datensatzes
$stmt = $db->prepare("DELETE FROM Benutzer WHERE id = :id");
$stmt->bindValue(':id', $id, SQLITE3_INTEGER);

$result = $stmt->execute();

if ($result) {
    echo "Datensatz erfolgreich gelöscht.<br>";
} else {
    echo "Fehler beim Löschen des Datensatzes: " . $db->lastErrorMsg() .
    "<br>";
}
?>
```

Datentabelle auslesen und ausgeben

```
<?php
// Fortsetzung des vorherigen Codes

// SQL-Befehl zum Auslesen der Daten
$sql = "SELECT id, vorname, nachname, email, reg_date FROM Benutzer";
$result = $db->query($sql);

if ($result) {
    while ($row = $result->fetchArray(SQLITE3_ASSOC)) {
        echo "ID: " . $row["id"] . " - Name: " . $row["vorname"] . " " .
        $row["nachname"] . " - Email: " . $row["email"] . " - Registriert am: " .
        $row["reg_date"] . "<br>";
    }
} else {
    echo "Fehler beim Auslesen der Daten: " . $db->lastErrorMsg() . "<br>";
}
?>
```

Datenbank schließen

```
<?php
// Fortsetzung des vorherigen Codes

$db->close();
echo "Datenbankverbindung geschlossen.";
?>
```