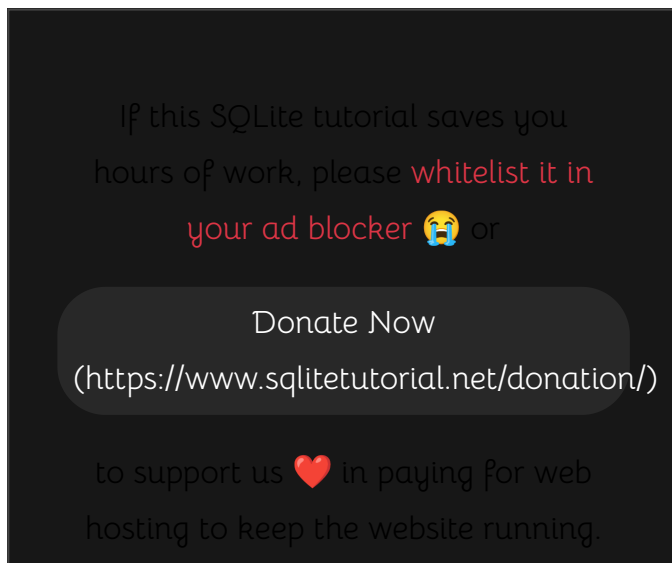


SQLite PHP: Inserting Data



Summary: in this tutorial, we will show you how to use PHP PDO to insert data into SQLite tables.

We will use the projects and tasks table that we created in the [creating table tutorial](https://www.sqlitetutorial.net/sqlite-php/create-tables/) (<https://www.sqlitetutorial.net/sqlite-php/create-tables/>) .

To insert data into a table, you follow these steps:

1. [Connect to the SQLite database](https://www.sqlitetutorial.net/sqlite-php/connect/) (<https://www.sqlitetutorial.net/sqlite-php/connect/>) by creating a new PDO instance with the path to the SQLite database file.
2. Compose an [INSERT statement](https://www.sqlitetutorial.net/sqlite-insert/) (<https://www.sqlitetutorial.net/sqlite-insert/>) . If you want to supply the values for the `INSERT` statement, you use the named placeholders in the form `:name` which `name` is the parameter that you will provide the value.
3. Call the `prepare()` method of the PDO object to prepare the `INSERT` statement for execution. This method returns a `PDOStatement` object.
4. Bind values to the parameters by calling the `bindValue()` method of the `PDOStatement` object.
5. Execute the statement by calling the `execute()` method of the `PDOStatement` object.
6. In case the table has a generated [autoincrement](https://www.sqlitetutorial.net/sqlite-autoincrement/) (<https://www.sqlitetutorial.net/sqlite-autoincrement/>) column as the [primary key](https://www.sqlitetutorial.net/sqlite-primary-key/) (<https://www.sqlitetutorial.net/sqlite-primary-key/>)

key/), you can get the inserted id by calling the `lastInsertId()` method of the PDO object.

The following `SQLiteInsert` class has two methods for inserting data into tables. The `insertProject()` method inserts a new project into the `projects` table and the `insertTask()` method inserts a new task into the `tasks` table.

```
<?php

namespace App;

/**
 * PHP SQLite Insert Demo
 */
class SQLiteInsert {

    /**
     * PDO object
     * @var \PDO
     */
    private $pdo;

    /**
     * Initialize the object with a specified PDO object
     * @param \PDO $pdo
     */
    public function __construct($pdo) {
        $this->pdo = $pdo;
    }

    /**
     * Insert a new project into the projects table
     * @param string $projectName
     * @return the id of the new project
     */
    public function insertProject($projectName) {
        $sql = 'INSERT INTO projects(project_name) VALUES(:project_name
```

```

$stmt = $this->pdo->prepare($sql);
$stmt->bindValue(':project_name', $projectName);
$stmt->execute();

return $this->pdo->lastInsertId();
}

/**
 * Insert a new task into the tasks table
 * @param type $taskName
 * @param type $startDate
 * @param type $completedDate
 * @param type $completed
 * @param type $projectId
 * @return int id of the inserted task
 */
public function insertTask($taskName, $startDate, $completedDate, $
    $sql = 'INSERT INTO tasks(task_name,start_date,completed_date,c
        . 'VALUES(:task_name,:start_date,:completed_date,:compl

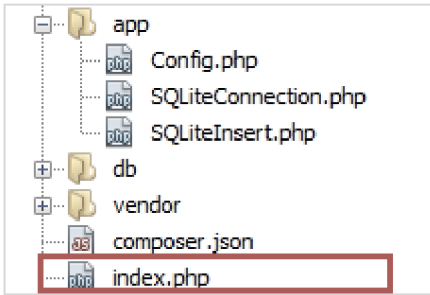
$stmt = $this->pdo->prepare($sql);
$stmt->execute([
    ':task_name' => $taskName,
    ':start_date' => $startDate,
    ':completed_date' => $completedDate,
    ':completed' => $completed,
    ':project_id' => $projectId,
]);

return $this->pdo->lastInsertId();
}
}

```

In the `insertTask()` method, instead of binding values, we pass the inserted values as an array to the `execute()` method.

We use the following code in the index.php file to insert projects and their associated tasks into the projects and tasks table.



```
<?php
```

```
require 'vendor/autoload.php';
```

```
use App\SQLiteConnection;
```

```
use App\SQLiteInsert;
```

```
$pdo = (new SQLiteConnection())->connect();
```

```
$sqlite = new SQLiteInsert($pdo);
```

```
// insert a new project
```

```
$projectId = $sqlite->insertProject('PHP SQLite Demo');
```

```
// insert some tasks for the project
```

```
$sqlite->insertTask('Prepare the sample database schema', '2016-06-01',
```

```
$sqlite->insertTask('Create new tables ', '2016-05-01', null, 0, $proje
```

```
$sqlite->insertTask('Insert some sample data', '2016-05-01', '2016-06-0
```

```
// insert a second project
```

```
$projectId = $sqlite->insertProject('Mastering SQLite');
```

```
// insert the tasks for the second project
```

```
$sqlite->insertTask('Go to sqlitetutorial.net', '2016-06-01', null, 0,
```

```
$sqlite->insertTask('Read all the tutorials.', '2016-06-01', null, 0, $
```

```
$sqlite->insertTask('Use Try It page to practice the SQLite commands.',
```

```
$sqlite->insertTask('Develop a simple SQLite-based application', '2016-
```

In this tutorial, we have shown you how to use the PDOStatement to insert data into the SQLite tables.